Time-Series Forecasting with Hybrid Federated Learning

A Personalized Approach to Collaboration

J.R. Vega Sanchez

TUDelft

Time-Series Forecasting with Hybrid Federated Learning

A Personalized Approach to Collaboration

by

J.R. Vega Sanchez

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday November 19, 2024 at 11:00 AM.

Student number:	4204207	
Project duration:	November, 2023 – November, 2024	
Thesis committee:	Dr. Lydia Y. Chen,	Faculty EEMCS, TU Delft, University supervisor
	Dr. Rihan Hai,	Faculty EEMCS, TU Delft
	Dr. Thiago Guzella,	ASML, Company supervisor
	Aditya Shankar	Faculty EEMCS, TU Delft, Supervisor

TU Delft

Preface

This thesis has been a long journey which took a lot of effort to complete. I could not have done this without my supervisors Dr. Lydia Y. Chen, Dr. Thiago Guzella and Aditya Shankar. I want to thank Dr. Lydia Y. Chen for giving me the opportunity to work on this research project and believing in me from the start, Dr. Thiago Guzella from ASML B.V. for sharing the same passion for this project and guiding me with critical feedback, giving me the ability to polish this project, and Aditya Shankar for guiding me through this project on a weekly basis and being available whenever I needed him. I would not have been able to complete this research at this level without the help and guidance from all of you.

I also want to thank Frédérique van Tilborg for supporting me along this journey. She took care of me when during the most challenging times and encouraged me along the way.

Julio Vega Sanchez
The Hague, Netherlands
November 10, 2024

Contents

1

Research Paper

1.1. Research Paper

This first chapter contains the research paper. It is a condensed version of the thesis. The thesis starts from Chapter 2.

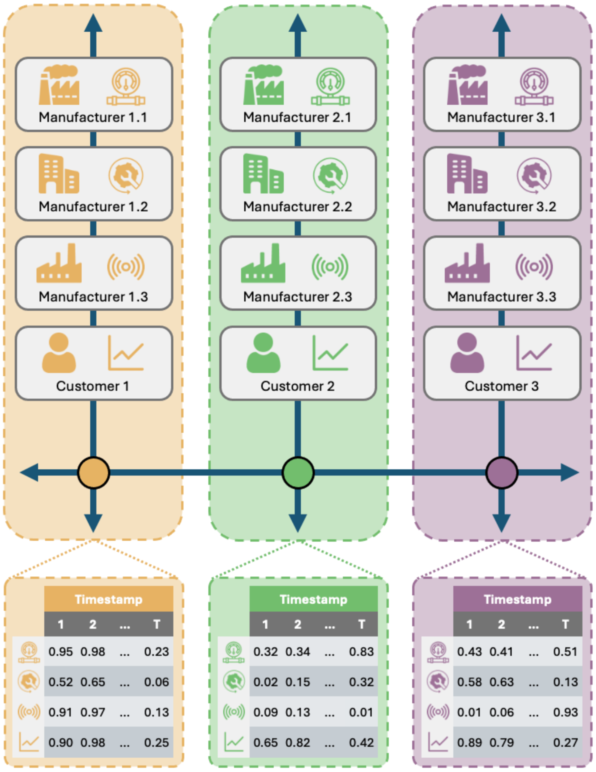
Time-Series Forecasting with Hybrid Federated Learning: A Personalized Approach to Collaboration

Abstract—**Collaborative efforts in Predictive Maintenance and Control can be beneficial for manufacturers and customers in industrial environments. However, these efforts are challenged by the need for multi-dimensional sharing of information about the same type (horizontal) and piece (vertical) of equipment, privacy restrictions and the presence of heterogeneous data distributions across participants. Existing solutions have addressed some of these challenges for forecasting or different purposes but there lacks a comprehensive approach that handles all of them for time series forecasting. To solve this problem, we introduce Time-series-based Personalized Hybrid Federated Learning (TPHFL), a hybrid federated learning (FL) strategy that combines Horizontal FL and Vertical FL to enable multi-level knowledge exchange while preserving data privacy. All participants use a personalization mechanism to make predictions that better suit their underlying data distribution. Our approach employs a distributed model to handle vertical privacy constraints and addresses data heterogeneity across equipment through a personalisation mechanism. Through extensive experiments on four public and one industry-specific datasets, we show that TPHFL outperforms independent learning scenarios by 27.2%, providing a strong incentive for parties to collaborate. We demonstrate the effectiveness of personalisation by showing an accuracy improvement of up to 42.7% when comparing TPHFL with personalisation to TPHFL without personalisation, and 32.7% when comparing traditional HFL methods to HFL with personalisation. Additionally, we evaluate a different configuration for personalisation and perform a detailed hyperparameter analysis to better understand the behaviour of TPHFL in different contexts.**

I. Introduction

Time series forecasting is relevant in Predictive Maintenance and Control (PMC), where temporal data and models are utilized to monitor and estimate the current health state of equipment, predict future behaviour for early problem flagging, or schedule maintenance [1], [2] In industrial environments, information related to one piece of equipment is scattered over different data sources and often siloed due to privacy concerns, making it challenging to integrate and leverage them for predictive models [3], [4]. Moreover, the deployment of this equipment is geographically distributed across multiple locations, gathering data in various operational contexts with heterogeneous data distributions and following similar privacy concerns [5]. Leveraging data from multiple locations and data sources offers the potential to significantly enhance the predictive performance of models, enabling more accurate forecasting and more effective PMC.

The problem scenario in Figure 1 illustrates the complexities of managing and integrating data from distributed equipment. Three clusters hold three manufacturers and one customer that collect unique performance measurements of distinct machines. Three contain sensory data owned by various manufacturers, and one holds performance data owned by the customer. Each customer wants to enhance the predictions of its performance data by utilizing the data from different manufacturers and customers. The utilization of data requires a two-level knowledge exchange. On the first level, we need to exchange knowledge related to the same piece of equipment within the cluster and on the second level, we need to exchange knowledge related to the same types of equipment between clusters. However, privacy restrictions inhibit sharing at both levels and even if we could address these concerns, the heterogeneous data distributions of clusters with data from machines in different operational contexts complicate information exchange.



	1	2	...	T
	0.95	0.98	...	0.23
	0.52	0.65	...	0.06
	0.91	0.97	...	0.13
	0.90	0.98	...	0.25

	1	2	...	T
	0.32	0.34	...	0.83
	0.02	0.15	...	0.32
	0.09	0.13	...	0.01
	0.65	0.82	...	0.42

	1	2	...	T
	0.43	0.41	...	0.51
	0.58	0.63	...	0.13
	0.01	0.06	...	0.93
	0.89	0.79	...	0.27

Fig. 1: Problem scenario: three clusters with manufacturers and customers collect performance measurements. Utilization of all data requires knowledge exchange within, and between clusters.

Existing solutions Federated Learning (FL), more specifically Horizontal FL (HFL), addresses the privacy concerns between clusters by training a global model while only sharing local model updates [6], [7], as shown in Figure 2a. This method allows for the exchange of information between clusters of data belonging to the same machine whilst preserving horizontal privacy constraints. Still, it does not account for the heterogeneous task profiles and could lead to the generalization of predictive models. Multi-Task FL (MTFL) and HFL-based personalization methods address the data heterogeneity by considering the modelling of machine-specific data to be a unique task and balancing task-specific (i.e. cluster-specific) and global knowledge, or by customizing a shared model to adapt to machine-specific data, respectively [8], [9]. However, these methods do not overcome the privacy restrictions within the clusters.

Alternatively, Vertical FL (VFL) addresses these restrictions within clusters by training separate models for each party that differ in accustomed features [10], as shown in Figure 2b. Different models are trained separately for manufacturers and customers and exchange knowledge to improve the predictive capabilities. Since this method does not overcome privacy restrictions between clusters, HFL combined with VFL accommodates both but does not allow for heterogeneous task profiles [11]. All existing solutions lack a comprehensive approach that effectively handles knowledge exchange between and within clusters and accounts for heterogeneous profiles.

Challenges Existing solutions solve part of the problems for time-series forecasting or provide a solution for non-temporal data. However, a comprehensive approach that effectively handles all problems for time-series forecasting does not exist. Together with its importance in practical applications such as PMC, this shows the need for further research into collaborative time-series predictions.

The primary challenges in this domain include:

- **Horizontal knowledge exchange**: Between clusters knowledge is siloed due to privacy constraints requiring inter-task knowledge exchange between parties of different tasks while preserving data privacy. Here, we consider modelling data of different machines as a unique task, similar to the definitions in MTFL [8], [12], [13].
- **Heterogeneous task profiles**: The distinctive characteristics of each task must be accounted for when sharing knowledge between them as they may exhibit diverse profiles of time-series data due to different operational contexts. Classic HFL methods struggle to maintain accuracy when dealing with such non-IID data, which adds complexity to FL implementations [14].
- **Vertical knowledge exchange for sequential data**: Within tasks, knowledge is distributed between multiple parties and siloed due to privacy constraints requiring intra-task exchange between parties of the same task whilst preserving data privacy. We are particularly interested in time-series-based solutions which contain sequential data. Traditional models fail to capture temporal dependencies and specialized sequential models overcome these challenges [15].

Contributions In this paper, we present a novel approach to time-series forecasting in predictive maintenance and control by introducing a Hybrid FL strategy: Time-series-based Personalized Hybrid Federated Learning (TPHFL). Our contributions are as follows:

1) **Hybrid FL strategy**: TPHFL integrates both horizontal and vertical dimensions in FL, facilitating knowledge exchange within tasks (intra-task) and between tasks (inter-task) We use a hierarchical solution strategy that approaches this problem at two different levels. First, each task is assigned a task model horizontally aggregated by a Federator [7]. Second, each task model operates as a distributed model with distinct entry points for each feature, enabling vertically distributed features in each task. Our strategy provides privacy to a certain extent by preserving the locality of data, laying a critical groundwork for future privacy-preserving solutions.
2) **Time-series-based memorization**: In TPHFL, clients train a global model that generalizes to all tasks. For each task, the model is adapted to the task-specific environment by a personalization mechanism which utilizes the task-specific training data to refine the predictions and better align with the underlying data distribution [16].
3) **Time-series-based hierarchical model**: A deep learning model containing sequential model components has private entry points for each party that produces intermediary representations. These are concatenated and fed through an upper layer. This architecture contains sequential modules to facilitate temporal data and can be split into multiple components making it compatible with the Hybrid FL strategy.

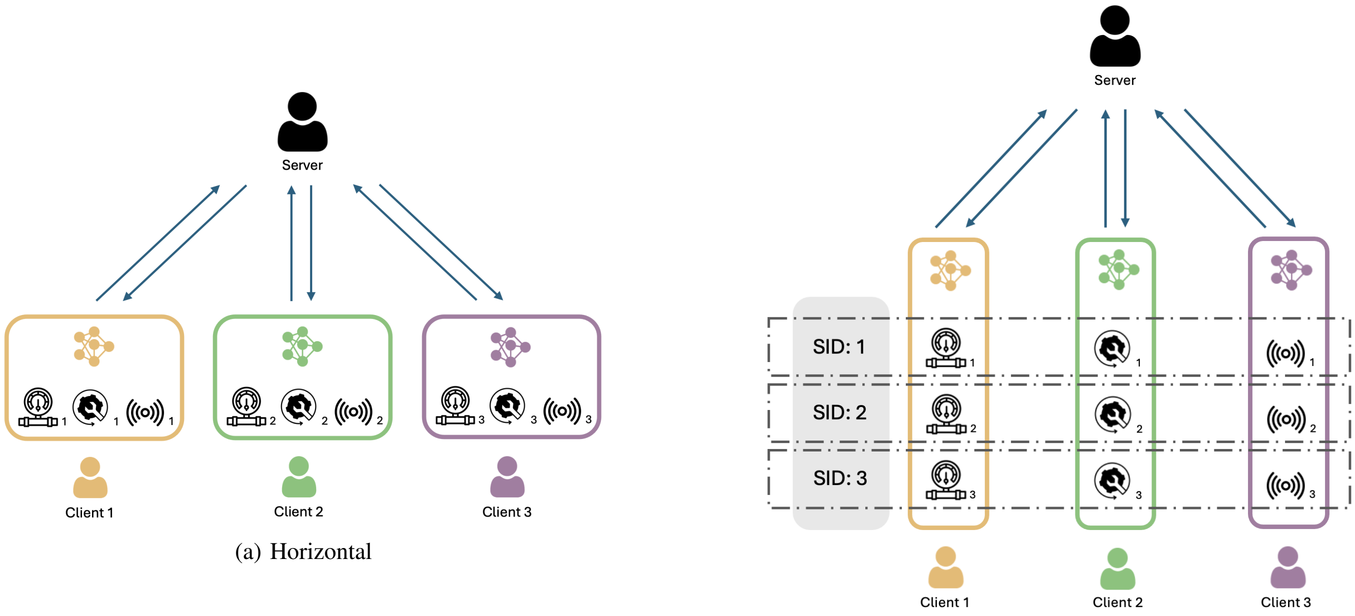
II. Related Work

In this section, we discuss background knowledge on time series forecasting, FL and MTFL.

A. *Time series forecasting*

Time-series forecasting involves predicting future values based on previously observed values in a sequence over time. The primary challenge in time-series forecasting lies in capturing temporal dependencies and patterns, which can be complicated by trends, seasonality, and irregularities [17]. Seasonal Autoregressive Integrated Moving Average with eXogenous inputs (SARIMAX) has been used widely for time-series forecasting, and generalizes other forecaster such as ARIMAX, ARIMA and SARIMA [18]–[20]. These linear models are suitable for small datasets and have a low time complexity but often struggle with complex, non-linear relationships within the data [21].

In recent years, deep learning models have become increasingly popular in recent years for time-series forecasting due to their ability to capture complex temporal dependencies [22], [23]. Recurrent Neural Networks (RNNs) were one of the first neural architectures designed for sequential data, enabling the network to maintain an evolving hidden state to capture



(a) Horizontal

(b) Vertical

Fig. 2: Two types of Federated Learning. Each client holds different features or Sample IDs (SID)
.

temporal patterns [24], [25]. However, traditional RNNs have limited ability to model long-term dependencies effectively because of the vanishing gradient problem [26]. Long Short-Term Memory (LSTM) addresses this limitation by incorporating memory cells and gating mechanisms to selectively retain relevant information over longer time sequences, making them more robust for complex time-series tasks [23], [27]. Gated Recurrent Units (GRUs), a simplified variant of LSTMs, offer similar performance while reducing computational complexity by merging some of the gating mechanisms [23], [28]. More recently, attention-based methods such as the Transformer [29] enhance interpretability and performance in time-series forecasting. These methods rely on attention mechanisms to dynamically focus on different parts of the input sequence, providing high accuracy and insights into the underlying temporal relationships [30]. However, due to the increased model complexity, these models risk overfitting, especially with small datasets [31].

B. Horizontal Federated Learning

HFL allows N clients with different samples and the same features to collaboratively train a machine-learning model without sharing their input data [6] (Figure 2a). Instead, they share locally computed model updates, such as gradients or model parameters, with a central server. In FedAvg [7], clients share model weights with a central server, the Federator, which aggregates these, updates and redistributes the aggregated model back to the clients [7], [32]:

$$\theta_{\text{Global}} = \frac{1}{N}\sum_{n=1}^{N}\theta_n \tag{1}$$

HFL is particularly advantageous for maintaining data privacy, as sensitive information remains localized at clients' sites.

C. Heterogeneous data distributions in HFL

To tackle data heterogeneity in FL, various solutions have been proposed, particularly MTFL and personalization [12], [13]. MTFL is a federated adaptation of Multi-Task Learning (MTL) [33], [34] where different related tasks can be learned jointly, allowing knowledge to be shared between tasks. MTFL extends MTL by treating clients as a unique task. Shared-private attention mechanisms address data heterogeneity by selectively focusing on relevant information across tasks using attention-based models [35]–[37]. In these methods, clients train one model with one private attention layer for each client and one shared attention layer, balancing task-specific and shared knowledge, and improving the performance of individual tasks. However, the large model complexity leads to overfitting and the models lean more towards generalization [31].

Clustering techniques group clients with similar data distributions or tasks together [38]–[40]. Each cluster trains a separate model, allowing for information exchange between similar clients. Secondary information such as model weights can be shared between the clusters in training a cluster model. However, cluster models will always generalize to their clients, which is a problem if the tasks do not align perfectly.

Personalization extends the HFL paradigm and customizes global models individually for clients with unique data distributions [12], [13]. Traditional personalization methods, such as those employed in FedProx, FedPer, and FedRep, adapt a global model to local data distributions [**?**], [41]. FedProx extends the FedAvg method by introducing a proximal term used during training to ensure the local models do not deviate significantly from the global model, allowing the client to balance personalization and generalization. However, FedProx shows limited improvement over FedAvg. FedPer and FedRep the model into global and local components. In these methods, we train the global part collectively and local components independently for each client. Although these approaches allow for personalized adaptation, they can fail to capture critical features within the global model that may benefit all clients.

Memorization-based approaches such as KnnPer [16] provide a personalized solution by using local memorization of training samples for predictions. Instead of focusing on global model updates, KnnPer relies on a non-parametric method, where each client makes predictions by leveraging its training data directly through a k-nearest neighbors (KNN) approach. This approach selects the k most similar samples and uses the sample labels to make new predictions allowing the model to adapt to each client's unique data distribution without global parameter updates.

D. Vertical Federated Learning

VFL addresses scenarios where different clients hold different features of the same samples [10] (Figure 2b). In MMVFL, a central server aligns the locally predicted labels of clients through aggregation [42]. This allows for the effective transfer of private labels but is limited to closed-form models and is not suitable for non-linear models. Through Secure Multi-Party Computation (SMPC) each client trains a local model with the encrypted data from their peers [43]. However, both these methods are not suitable for non-linear models.

In split learning, all clients train a segmented model of which each client has access to only a portion of the model [44], [45]. This allows participants to share only intermediate representations (instead of raw data), preserving privacy while enabling model training across vertically partitioned data. This method has been designed for classification purposes but can be altered to work for sequential data.

While there are Hybrid FL solutions that combine VFL and HFL approaches [11], they are often ill-suited for time-series data due to its focus classification models and datasets.

III. Framework

We introduce the problem definition in this section, followed by a step-by-step overview of TPHFL.

1) Problem definition: The proposed architecture targets time-series forecasting problems involving heterogeneous data distributions across tasks. Specifically, we consider N tasks with M parties that predict a univariate time series given endogenous feature $X_{n,1}$ and exogenous features $X_{n,j}, \forall j \in [2, M]$. Each party $i \in [M]$ for task n owns the samples for feature $X_{n,i}$. All input features are uni-variate and have the same time window W but can, for simplicity, be considered one multi-variate input vector $X_n \in \mathbb{R}^{M \times W}$ belonging to task n. For each task n, the model predicts one or multiple future time steps for the endogenous feature. $Y_n \in \mathbb{R}^P$ represents the predictions, the so-called target, where P is the length of the output time window and owned by party 1 for task n. We assume that in each task common samples are identified and aligned by privacy-preserving mechanisms [48]–[50].

Method	H	V	DH	TS	NN
FedAvg [7]	✓	✗	✗	✗	✓
[32]	✓	✗	✗	✓	✓
SplitNN [44]	✗	✓	✗	✗	✓
MMVFL [42]	✗	✓	✗	✗	✓
STV [43]	✗	✓	✗	✓	✗
[11]	✓	✓	✗	✗	✓
FedProx [41]	✓	✗	✓	✗	✓
FedPer [46]	✓	✗	✓	✗	✓
FedRep [47]	✓	✗	✓	✗	✓
FATHOM [35]	✓	✗	✓	✓	✓
MTL-Trans [36]	✓	✗	✓	✓	✓
MSJF [37]	✓	✗	✓	✓	✓
MOCHA [38]	✓	✗	✓	✗	✓
[40]	✓	✗	✓	✗	✓
KnnPer [16]	✓	✗	✓	✗	✓
TPHFL	✓	✓	✓	✓	✓

TABLE I: Horizontal (H), Vertical (V), Data Heterogeneity (DH), Time Series (TS), Neural Networks (NN)

Notation	Meaning
N	number of tasks
M	number of parties and features in each task
W	size of input time window
P	size of output time window
H	size of hidden dimension
$X_n \in \mathbb{R}^{M \times W}$	input vector for task n
$X_{n,m} \in \mathbb{R}^{W}$	feature m of input vector for task n
$Y_n \in \mathbb{R}^{P}$	prediction vector for task n
θ_n	model of task n
$\theta_{n,m}$	m-th component of model of task n
$h_m \in \mathbb{R}^{W \times H}$	hidden vector produced by component $m \in [M]$
$h \in \mathbb{R}^{M \times W \times H}$	concatenation of vectors h_m for $m \in [M]$
$h' \in \mathbb{R}^{W \times H}$	hidden state vector produced by component $M + 1$
$\phi(X)$	intermediary representation for input vector X

TABLE II: Notations

The objective of this work is to improve time series prediction through personalization while preserving data locality by ensuring that features are kept private within or between tasks. To achieve this, we first aim to develop a global model that generalizes well across all tasks: To achieve this, we first aim to develop a global model that generalizes well across all tasks:

$$\min_{\theta_{\text{Global}}} \sum_{n=1}^{N} \mathcal{L}_n(\theta_{\text{Global}}) \tag{2}$$

After a consensus on the model by all tasks, we use it in a personalization algorithm that allows for predictions better suited for the tasks underlying data distribution.

A trusted third party takes the role of the Federator, responsible for securely collecting the model weights of each party, aggregating them, and redistributing them to the correct parties. Furthermore, it is responsible for initializing the weights of θ_{Global} and sharing them with each party.

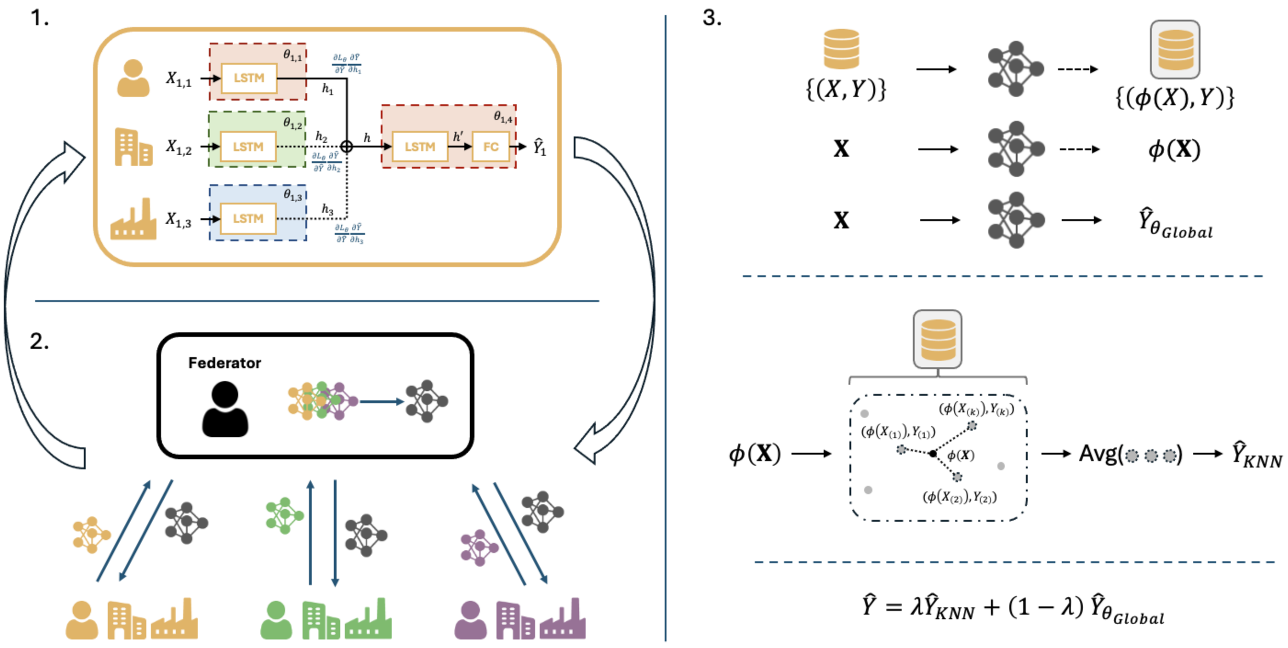

1.
$X_{1,1}$
$X_{1,2}$
$X_{1,3}$
LSTM
LSTM
LSTM
$\theta_{1,1}$
$\theta_{1,2}$
$\theta_{1,3}$
h_1
h_2
h_3
$\frac{\partial L_\theta}{\partial \hat{Y}} \frac{\partial \hat{Y}}{\partial h_1}$
$\frac{\partial L_\theta}{\partial \hat{Y}} \frac{\partial \hat{Y}}{\partial h_2}$
$\frac{\partial L_\theta}{\partial \hat{Y}} \frac{\partial \hat{Y}}{\partial h_3}$
h
$\theta_{1,4}$
LSTM
h'
FC
$\hat{Y}_1$
2.
Federator
3.
$\{(X, Y)\}$
$\{(\phi(X), Y)\}$
$\mathbf{X}$
$\phi(\mathbf{X})$
$\mathbf{X}$
$\hat{Y}_{\theta_{Global}}$
$\phi(\mathbf{X})$
$(\phi(X_{(k)}), Y_{(k)})$
$(\phi(X_{(1)}), Y_{(1)})$
$\phi(X)$
$(\phi(X_{(2)}), Y_{(2)})$
Avg( )
$\hat{Y}_{KNN}$
$\hat{Y} = \lambda \hat{Y}_{KNN} + (1 - \lambda)\, \hat{Y}_{\theta_{Global}}$

Fig. 3: TPHFL Framework in three incremental steps: training, optimization and personalization

2) TPHFL overview: An overview of TPHFL is given in Figure 3. The method consists of three steps: training, optimization and personalization. The first two are described in Algorithm 1, the latter in Algorithm 2. We will discuss each step individually.

Training. In the initial step, the parties with the features and labels for task n collaboratively train a distributed model for multiple epochs. The task model θ_n contains multiple components:

$$\theta_n = \{\theta_{n,1}, \theta_{n,2}, ..., \theta_{n,M+1}\} \tag{3}$$

$\theta_{n,1}$ to $\theta_{n,M}$ comprise single LSTM units at the beginning of the model, whereas $\theta_{n,M+1}$ contains an LSTM and Fully Connected (FC) layer. We chose an LSTM because of its ability to capture long-term dependencies at a moderate level of model complexity. Each party $m \in [M]$ for task n has private ownership over $\theta_{n,M}$ and party 1 has additional ownership over $\theta_{n,M+1}$ meaning that only the designated party can read and write the given model weights.

To collaboratively train θ_n, the Federator initiates the training process for all parties across tasks in lines 6 and 8 of Algorithm 1. The Federator calls party 1 separately to handle the flow of data through the final model component. Each party processes their data through their private LSTM with an input window of size 1 in line 18. This way, each piece of input data $X_{n,m} \in \mathbb{R}^W$ is transformed to hidden states $h_m \in \mathbb{R}^{W \times H}$. Each party shares these states with party 1 in line 19, who concatenates them in line 22, producing $h \in \mathbb{R}^{M \times W \times H}$. This state serves as input for the LSTM in $\theta_{n,M+1}$ with input window $M \times H$ that transforms h to $h' \in \mathbb{R}^{W \times H}$. The new hidden state is fed through the FC layer to produce prediction $\hat{Y}_n$. During back-propagation in line 23, party 1 calculates the loss and gradient for $\theta_{n,M+1}$ necessary for updating the model parameters:

$$\nabla \theta_{n,M+1} = \frac{\partial \mathcal{L}_{\theta_n}}{\partial \theta_{n,M+1}} = \frac{\partial \mathcal{L}_{\theta_n}}{\partial \hat{Y}_n} \frac{\partial \hat{Y}_n}{\partial \theta_{n,M+1}} \tag{4}$$

Parties 1 to M calculate the gradients for $\theta_{n,1}$ to $\theta_{n,M}$ individually with:

$$\nabla \theta_{n,m} = \frac{\partial \mathcal{L}_{\theta_n}}{\partial \theta_{n,m}} = \frac{\partial \mathcal{L}_{\theta_n}}{\partial \hat{Y}_n} \frac{\partial \hat{Y}_n}{\partial h_m} \frac{\partial h_m}{\partial \theta_{n,m}} \tag{5}$$

Party 1 calculates the gradient for $\theta_{n,M+1}$ and $\theta_{n,1}$ and the derivatives $\frac{\partial \mathcal{L}_{\theta_n}}{\partial \hat{Y}_n}$ and $\frac{\partial \hat{Y}_n}{\partial h_m}$ for $m \in [2, M]$. It sends the derivatives to the correct parties in line 24 so they can complete their gradient calculations in line 27.

Optimization. Each party shares its model component with the Federator, which is responsible for aggregating these components using the FedAvg algorithm (Equation 1). The Federator identifies each model component it receives and aggregates all $\theta_{n,i}$ separately. It saves the new model weights in lines 6 and 8 based on task and party index. This approach simulates the aggregation of the full task models θ_n by aggregating components independently in line 12, creating a global model that is the same for all tasks while only exchanging components. Through this process, the Federator facilitates collaboration and information exchange between tasks.

We alternate training and optimization across multiple training epochs. After each local epoch, each task shares its

Algorithm 1: Training and optimization

```
Data: $X_{n,m}$ on party $(n,m)$ fed in batches $B_{n,m}$, and
      $Y_n$ on party $(n,1)$
Param: Global model parameters $\theta_{Global}$, tasks $N$,
       distributed features $M$, rounds $R$, epochs $E$
Result: Trained distributed models $\theta_{Global}$
Federator executes:
    Initialize $\theta_{Global}$;
    for $r = 1, ..., R$ do
        for $(n,m) \in [N] \times [M]$ do
            if $m == 1$ then
                $[\theta_{n,m}, \theta_{n,M+1}] \leftarrow$
                  Train$(n, m, \theta_{Global,m}, \theta_{Global,M+1})$;
            else
                $\theta_{n,m} \leftarrow$
                  Train$(n, m, \theta_{Global,m}, None)[0]$;
            end
        end
        for $m \in [M+1]$ do
            $\theta_{Global,m} = \frac{1}{N}\sum_N \theta_{n,m}$
        end
    end
Train$(n, m, \theta_m, \theta_{M+1})$
    for $e = 1, ..., E$ do
        for $b \in B_{n,m}$ do
            $h_m \leftarrow \theta_m(b)$ ;
            send($h_m$ to party $(n,1)$);
            if $m == 1$ then
                await($h_m$ for $m \in [M]$);
                $\hat{Y} \leftarrow \theta_{M+1}(\bigoplus_M h_m)$;
                $\theta_{M+1} \leftarrow \theta_{M+1} - \nabla\mathcal{L}_{\theta_{M+1}}$;
                send($\frac{\partial\theta}{\partial\hat{Y}}\frac{\partial\hat{Y}}{\partial h_m}$ to $(n,m), \forall m \in [M]$)
            end
            await($\frac{\partial\mathcal{L}_\theta}{\partial\hat{Y}}\frac{\partial\hat{Y}}{\partial h_m}$ from party 1);
            $\theta_m \leftarrow \theta_m - \nabla\theta_m$;
        end
    end
    return $[\theta_m, \theta_{M+1}]$
```

Algorithm 2: Personalization

```
   Data: Dataset $S_n$ on task $n$
   Param: Distributed features $M$
   Result: Predictions $\hat{Y}$
1  Each task $n$ executes:
2      $D_n \leftarrow \emptyset$ for $m \in [M]$ do
3          if $m == 1$ then
4              $D_n \leftarrow$ TransformData$(n, m, S_n)$;
5          else
6              TransformData$(n, m, S_n)$;
7          end
8      end
9      At inference on $\mathbf{X}$ return $\hat{Y}$ with transformed data
        $D_n$ and Equation 10 ;
10 TransformData$(n, m, S)$
11     $D \leftarrow \emptyset$ ;
12     for $(X, Y) \in S$ do
13         for $m \in [M]$ do
14             $h_m = \theta_m(X_m)$;
15             send($h_m$ to party $(n,m)$);
16             if $n == 1$ then
17                 await($h_m$ for $m \in [M]$);
18                 $\phi(X) \leftarrow \bigoplus_M h_m$;
19                 $D \leftarrow D \cup (\phi(X), Y)$;
20             end
21         end
22     end
23     return $D$
```

model components with the Federator and receives updated components. We continue this process until the task model performances have converged or after a fixed amount of training epochs, reaching a consensus on the global model.

Personalization. After achieving consensus on the final global model, having completed training and optimization, each task uses its task model θ_n for memorization-based personalization. In this approach, we select the most similar training samples during inference using KNN and use the accompanied labels Y for memorization-based predictions. For similarity measurements, we transform the observation X in the training samples to intermediary representations $\phi(X)$ as these contain significant information about the model's input interpretation. This representation can be any state inside the distributed model. In our case, we use h as an intermediary representation.

Typically, we transform our data beforehand as this can be computationally expensive. In lines 4 and 6, we call the transformation algorithm for each party of task n, of which party 1 saves this transformed data in a new dataset. All parties collaboratively transform the data, as each party will send their hidden states to party 1 in line 15. Party 1 saves ϕ together with the labels in line 19.

During inference in line 9, we select the k most similar samples from our transformed dataset:

$$N_k(\mathbf{X}) = \{(\phi(X_{(1)}), Y_{(1)}), ..., (\phi(X_{(k)}), Y_{(k)})\} \quad (6)$$

where the ordering is determined by intermediary distances:

$$d(\phi(X_{(1)}), \mathbf{X}) \leq ... \leq d(\phi(X_{(k)}), \mathbf{X}) \quad (7)$$

$(\phi(X_{(1)}), Y_{(1)})$ is the i-th nearest neighbour for task n and sample X. The distance metric d, typically chosen as Euclidean distance or another similarity measure, plays a key role in determining the influence of each neighbour label on the final prediction.

These neighbours are most similar to our input and can be used to make memorization-based predictions:

$$d_{(i)}(\mathbf{X}) = d(\phi(X_{(i)}), \mathbf{X}) \quad (8)$$

$$\hat{Y}_{KNN} = \frac{\sum_{i=1}^{k} K(d_{(i)}(\mathbf{X}))Y_{(i)}}{\sum_{i=1}^{k} K(d_{(i)}(\mathbf{X}))} \tag{9}$$

where K is a kernel. We combine these predictions with global model predictions $\hat{Y}_{\theta_{Global}}$, the output of the global model.

$$\hat{Y} = \lambda \hat{Y}_{KNN} + (1-\lambda)\hat{Y}_{\theta_{Global}} \tag{10}$$

where $\lambda \in [0,1]$ is a weight parameter balancing the contributions of both losses for task n.

IV. Experiments

We evaluate the forecasting of TPHFL against scenarios with different forms of data locality and collaborative capabilities. Additionally, we conduct experiments with a different hidden representation ϕ and perform hyper-parameter analysis. We first discuss the experimental settings.

A. Experimental settings

We use four public datasets for the experiments: Air quality [51], Solar power [52], Crypto [53] and Rossman Sales [54]. Additionally, we used an industry-specific dataset to predict a specific parameter from sensor values in semiconductor manufacturing. Further details on the public datasets are given in Appendix A. We briefly go over the baselines used for evaluation and more specific settings.

	Collaboration	
Data locality	Vertical	Hybrid
None	-	Centralized
Horizontal	Independent	FedAvg, TPHFL-H
Vertical	-	Centralized+
Hybrid	Independent+	TPHFL-NP, TPHFL

TABLE III: Baseline methods with different forms of data locality and collaboration.

B. Baseline

We compare TPHFL to scenarios that differ in data locality and collaborative capabilities shown in Table III. We show a schematic overview for each baseline method in Appendix B. All methods enable vertical collaboration by default because we are training on multivariate time series data because the goal is not to demonstrate that using more features improves predictive performance. Instead, we focus on how different collaboration configurations impact the model's performance under privacy constraints. We will discuss the methods in order of privacy restrictions:

None: No privacy restrictions, allowing all data to be combined freely.

- **Centralized**: In this case, horizontal collaboration is introduced by centralizing and concatenating all training data and using it to train a single LSTM.

Horizontal: Parties share data within tasks, not between tasks.

- **Independent**: We allow vertical collaboration by letting each task train a separate LSTM on its multivariate time series data, with no exchange of information between tasks.
- **FedAvg**: We allow hybrid collaboration by letting each task train a separate LSTM and sharing the model weights with a Federator, responsible for aggregating the models from each task.
- **TPHFL-Horizontal**: This method builds upon FedAvg by adding the memorization-based personalization algorithm, similar to TPHFL. Different from TPHFL, TPHFL-H uses a single LSTM per task.

Vertical: Parties share data between tasks, not within tasks.

- **Centralized+**: This method is similar to its counterpart without privacy restrictions (Centralized) but employs the distributed model instead of a single LSTM to maintain vertical data locality.

Hybrid: Both dimensions of data locality are in place.

- **Independent+**: This method is similar to its counterpart with horizontal privacy restrictions (Independent) but employs a distributed model instead of a single LSTM to maintain vertical data locality.
- **TPHFL-NoPersonalization**: This variant of TPHFL omits the personalization algorithm.
- **TPHFL**: Our proposed method enables hybrid collaboration while maintaining horizontal and vertical data locality.

Independent and Centralized serve as expected upper-bound and lower-bound, respectively. We want TPHFL to show a decrease in Mean Absolute Error (MAE), meaning an increase in performance, compared with Independent, showing that there is an incentive for participants to share knowledge with other tasks. Centralized is the lower bound because it is an ideal scenario without data privacy constraints, allowing for less computational complexity and better performance.

C. Metrics and Setup

We compare the performance of TPHFL and the different scenarios using the Mean Absolute Error (MAE). For all datasets, the missing values were interpolated and replaced with 0 if there were no neighbouring values. We normalized all data for consistent comparison.

The LSTMs used in the experiments have two layers, a hidden size of 20 and a dropout of 0.2. We train the models in 30 epochs, with a batch size of 32, a learning rate of 0.001 and a weight decay of 0.001.

We conduct the training process by splitting the data into training data (80%) and test data (20%). We use a fixed input window of 32 and a variable prediction window of 1, 2, 4, 8 and 16. We construct the samples using a sliding window of 1 timestep.

To compare our method with other scenarios, we calculate the MAE for the different prediction windows and average all outcomes. For our solution, we run different values for k 1, 3, 5, 7 and 10 and choose the best-performing. In every

Dataset	Independent	Centralised	TPHFL	Rel. Imp.
AirQuality	3.60 +/- 0.07	2.77 +/- 0.01	2.96 +/- 0.01	17.8%
Industry	4.28 +/- 0.63	2.48 +/- 0.14	3.23 +/- 0.38	24.4%
Sales	2.96 +/- 0.01	3.02 +/- 0.02	4.40 +/- 0.06	-48.7%
Crypto	2.64 +/- 0.14	1.69 +/- 0.04	2.40 +/- 0.04	8.9%
Solar	2.17 +/- 0.03	1.42 +/- 0.01	1.58 +/- 0.04	27.2%

TABLE IV: Average MAE and standard deviation for methods with no privacy constraints and TPHFL. We show the relative improvement of TPHFL over Independent (rel. imp.) in percentages.

experiment, we choose the most optimal λ per task for which we can get the lowest MAE. In the following paragraphs, we will discuss the results for 6 tasks (except if stated differently) and the best-performing k. In Appendix C, we included more extensive results for 2, 4 and 6 tasks with different prediction window sizes.

D. Forecasting results

Table IV shows that TPHFL performs better than Independent for four out of five datasets with an increase in performance as high as 27.2%. A performance increase was not feasible for the Sales dataset due to insufficient training samples and data quality. The Crypto dataset has a limited performance increase compared to its peers because two exogenous features are too similar to the endogenous feature and contain limited valuable information to improve the predictions (see Figure 7).

The relative improvement reflects the average performance enhancement across all prediction windows. However, our experiments have shown that this improvement is not uniform; some prediction windows exhibit significantly higher gains than others. One explanation for this is the seasonality in the temporal data, where repeating patterns over specific intervals can have varying impacts for different windows. Windows that coincide with seasonal trends may benefit from the model's ability to capture these patterns.

In Table V, we compare three methods without vertical privacy constraints to their counterparts that enforce vertical restrictions. Most experiments show declines in performance that can be attributed to using a distributed model in place of a single LSTM. We expect this outcome because increased model complexity typically leads to a trade-off in predictive performance. Centralized shows a smaller reduction or improvements in performance because it uses a larger dataset, which is crucial when training more complex models to mitigate the negative impact on accuracy. TPHFL can improve its performance by increasing the number of samples, something we learned from the Centralized experiments. However, this may not always be possible due to the limited availability of temporal data. Sensory data may only be available for a certain period, and receiving more samples requires the machine to operate for a longer periodremove ASML context, more general.

In Table VI, we compare two methods that incorporate horizontal collaboration and personalization with their counterpart that do not use personalization. The results demonstrate the effectiveness of the personalization algorithm, improving the accuracy in the horizontal and hybrid data privacy domain. This improvement is especially pronounced for TPHFL because TPHFL-NP struggles to fit the data due to the complexity of the distributed model and the limited amount of data. Personalization helps mitigate these challenges, leading to a larger performance gap than FedAvg and TPHFL-V, methods that are better suited to fit the model effectively without personalization.

Dataset	Independent+		Centralised+		TPHFL-H	
	MAE	Imp.	MAE	Imp.	MAE	Imp.
AirQuality	3.72 +/- 0.03	-3.3%	2.80 +/- 0.01	-0.9%	2.58 +/- 0.00	-14.8%
Industry	5.76 +/- 0.91	-34.7%	2.62 +/- 0.23	-6.0%	2.85 +/- 0.34	-13.5%
Sales	5.17 +/- 0.07	-74.8%	3.21 +/- 0.09	-6.3%	2.83 +/- 0.02	-55.3%
Crypto	3.91 +/- 0.41	-48.3%	1.61 +/- 0.03	4.5%	1.67 +/- 0.02	-43.7%
Solar	2.67 +/- 0.07	-23.1%	1.40 +/- 0.01	1.4%	1.38 +/- 0.02	-14.3%

TABLE V: Average MAE and standard deviation for different methods with vertical restrictions (Independent+, Centralized+), and Horizontal restrictions (TPHFL-H). We compare the performance increase for methods if we introduce vertical privacy restrictions: Independent to Independent+, Centralized to Centralized+ and TPHFL-H to TPHFL.

Dataset	FedAvg	Imp.	TPHFL-NP	Imp.
AirQuality	2.91 +/- 0.02	11.3%	3.49 +/- 0.03	15.0%
Industry	3.44 +/- 0.42	17.2%	4.29 +/- 0.55	24.6%
Sales	3.43 +/- 0.02	17.3%	7.68 +/- 0.09	42.7%
Crypto	1.94 +/- 0.04	14.0%	2.94 +/- 0.05	18.3%
Solar	2.05 +/- 0.03	32.7%	2.58 +/- 0.06	38.8%

TABLE VI: Average MAE and standard deviation for different methods with horizontal and hybrid privacy constraints. We show the improvement of introducing the personalization mechanism: FedAvg to TPHFL-H and TPHFL-NP to TPHFL.

Lastly, we show in Figure 4 the MAE for at least one method from each previous comparison. We show the results for three datasets and a different number of tasks. When comparing TPHFL with methods without privacy restrictions, we observe that Independent consistently serves as an upper bound, while Centralized almost always acts as a lower bound. The anomalies are caused by Centralized combining all data, sometimes at the expense of task-specific performance due to overfitting or loss of task nuances.

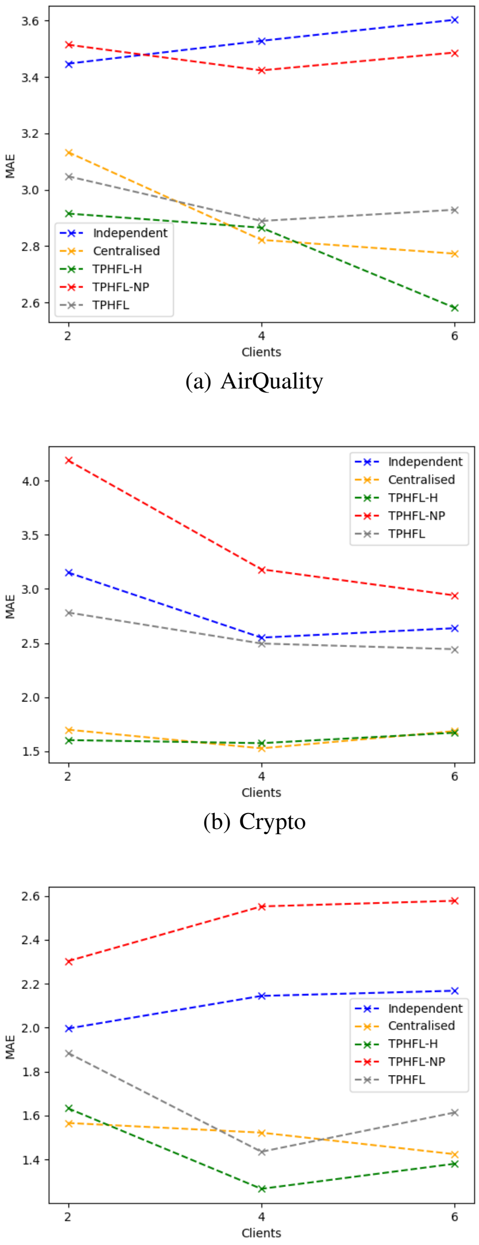
TPHFL-H consistently shows an improvement over TPHFL, as expected, since using a distributed model in TPHFL introduces complexity that often leads to a loss in accuracy. Finally, TPHFL-NP consistently underperforms compared to TPHFL, which aligns with our expectations, as the absence of personalization limits the model's ability to fine-tune itself to task-specific data distributions.

E. Different hidden representation

In the personalization step of TPHFL, the intermediary representation $\phi(X)$ can be any output within the model. In

Dataset	2 tasks		4 tasks		6 tasks	
	TPHFL-I2	TPHFL	TPHFL-I2	TPHFL	TPHFL-I2	TPHFL
AirQuality	**11.6%**	9.7%	**18.1%**	17.4%	**18.7%**	17.76%
Industry	39.2%	**39.5%**	37.1%	**37.8%**	23.2%	**24.5%**
Sales	-23.8%	**-19.8%**	-38.6%	**-18.4%**	-79.9%	**-48.7%**
Crypto	11.7%	**14.2%**	2.1%	**4.2%**	7.4%	**8.9%**
Solar	5.6%	**6.9%**	33.1%	**34.7%**	25.6%	**27.2%**

TABLE VII: Relative improvement over Independent for TPHFL with different forms of intermediary representations. TPHFL uses h as TPHFL-I2 uses h' as intermediary representation.



(a) AirQuality

(b) Crypto

(c) Solar

Fig. 4: Average error for different prediction windows from three datasets.

Table VII, we compare TPHFL with a variant that uses h' as an intermediary state, referred to as TPHFL-I2. The results indicate that this variant performs better for the AirQuality dataset, suggesting that, for this dataset, the hidden outputs from the upper model contain more valuable information for the samples than those generated by the private LSTM. A possible explanation is that the endogenous features for each task in the AirQuality dataset exhibit a higher correlation than other datasets, leading to task models leaning more toward global generalization rather than local specialization. The hidden states produced by the upper model LSTM are better suited for capturing these global patterns, making them more suitable as intermediary representations for this dataset.

However, the key takeaway is that choosing intermediary representations can be highly dataset-dependent as many unique combinations of outputs could be used for this purpose. One must carefully evaluate the different possibilities to find a representation that maximizes the predictive performance of TPHFL.

F. *Hyper-parameter analysis*

As mentioned earlier, we can tune two hyperparameters: the value of k and the hyperparameter λ. Our experiments revealed that varying k has little impact on the performance of TPHFL, as demonstrated in Figure 5. In this figure, we plotted the MAE for all datasets (excluding Sales due to its consistently low performance) across different values of k. The results show that the error remains nearly constant, indicating that the choice of k does not significantly affect the method's performance.

Different values of λ significantly affect the performance of each task model. In Figure 6, we use the Solar dataset as an example to illustrate this. We plot the MAE of three different strategies for selecting λ: setting a single global value for all tasks, choosing an optimal λ for each task individually, and selecting the optimal λ on a sample-by-sample basis. For reference, we included the centralized and independent as upper-bound and lower-bound, respectively. MM is plotted against different values for λ, while the other methods remain static because they either are not dependent on λ or always select an optimal value, leaving no room for tuning λ.

The optimal global λ is set at 0.4 for the Solar dataset, meaning that we interpolate 40% KNN predictions and 60% global predictions. We can reduce the MAE further by setting the parameter on a task basis, demonstrating that the optimal

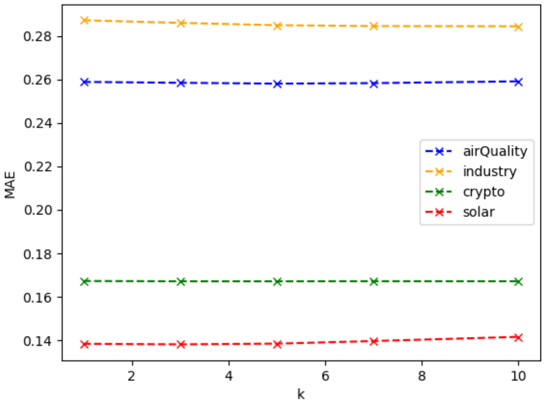


Fig. 5: Average MAE for different values of k in TPHFL.

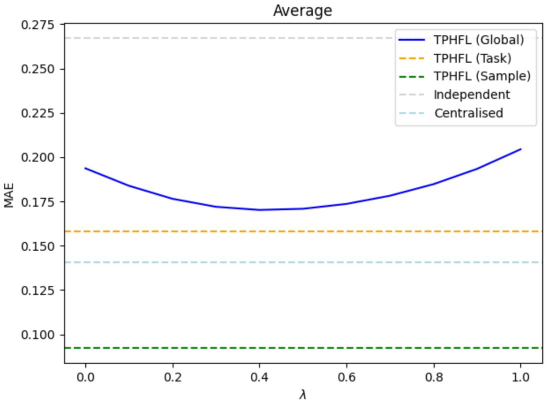


Fig. 6: Average MAE for different values of λ in TPHFL and Solar dataset.

task-based λ varies from task to task, allowing tasks to balance private and global information independently, improving overall performance. The balance between private and global information varies across tasks with heterogeneous data distributions. Consequently, the optimal λ depends on the specific characteristics of each task. Tasks with more private information tend to select a higher λ, while tasks with less private information lean toward a lower value.

The MAE of sample-based TPHFL falls significantly below the expected lower bound of Centralized because, on a sample-by-sample basis, the optimal λ often turns out to be an extreme value—either 0.0 or 1.0. In other words, it is best to rely entirely on KNN or global model predictions for most inferences, or in terms of data distribution, the model either fully prioritizes private information or global knowledge exchange, depending on the specific sample. When choosing an optimal λ at the task level, the model balances out these extreme cases, finding a middle ground. Further investigation is required to uncover a direct relationship between the characteristics of individual samples and their corresponding optimal λ values, providing more insight into the prioritization of private or global information.

V. Conclusion

In this thesis, we proposed a novel FL framework TPHFL designed to tackle the challenges of time-series forecasting in distributed, privacy-sensitive industrial environments. By integrating both HFL and VFL approaches, our model facilitates multi-level knowledge sharing while preserving data locality by not sharing private data between different parties, laying a critical groundwork for future, more robust privacy-preserving solutions. Experiments on several real-world datasets demonstrate the effectiveness of our method, showing a significant improvement in predictive performance over traditional independent models and further enhancing results from horizontal collaboration through a personalization algorithm.

For future research, there are multiple areas to investigate. Our method does not have any formal guarantees necessary for deploying a method like this. It is essential to strengthen security guarantees without sacrificing too much model performance. This could involve implementing formal privacy guarantees or leveraging privacy-preserving techniques such as homomorphic encryption [55] or differential privacy [56] which have been proven to be useful in federated setttings [57], [58]. We recommend exploring the possibilities of multivariate forecasting. Our method only predicts one endogenous feature. Predicting multiple interrelated variables could yield richer insights and more accurate forecasts in practical scenarios by capturing the interactions between variables. Methods such as FATHOM have shown that multivariate predictions are a viable option [35]. Future work could explore architectures that enable soft predictions, allowing each participating party to generate localized predictions. Soft predictions would enable parties without direct access to labels to make approximated predictions, typically aligned with predictions from party 1. Techniques such as label sharing in MMVFL or SMPC could facilitate secure, parallel training across parties, potentially allowing each party to independently refine and validate predictions [42], [43]. Lastly, Future research could investigate strategies for dynamically optimizing λ on a sample-specific basis, potentially developing algorithms to adapt λ based on real-time data characteristics. Another approach could involve analyzing the sensitivity of λ to different data distributions, enabling a better understanding of its role and refining it into a more flexible parameter within the model.

References

[1] K. S. Kiangala and Z. Wang, "An effective predictive maintenance framework for conveyor motors using dual time-series imaging and convolutional neural network in an industry 4.0 environment," *Ieee Access*, vol. 8, pp. 121 033–121 049, 2020.

[2] C.-Y. Lin, Y.-M. Hsieh, F.-T. Cheng, H.-C. Huang, and M. Adnan, "Time series prediction algorithm for intelligent predictive maintenance," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2807–2814, 2019.

[3] D. L. Andersen, C. S. A. Ashbrook, and N. B. Karlborg, "Significance of big data analytics and the internet of things (iot) aspects in industrial development, governance and sustainability," *International Journal of Intelligent Networks*, vol. 1, pp. 107–111, 2020.

[4] S. Sun, X. Zheng, J. Villalba-Díez, and J. Ordieres-Meré, "Data handling in industry 4.0: Interoperability based on distributed ledger technology," *Sensors*, vol. 20, no. 11, p. 3046, 2020.

[5] J. Ahn, Y. Lee, N. Kim, C. Park, and J. Jeong, "Federated learning for predictive maintenance and anomaly detection using time series data distribution shifts in manufacturing processes," *Sensors*, vol. 23, no. 17, p. 7331, 2023.

[6] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

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

[8] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias, "Three types of incremental learning," *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1185–1197, 2022.

[9] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 12, pp. 9587–9603, 2022.

[10] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y.-Q. Zhang, and Q. Yang, "Vertical federated learning: Concepts, advances, and challenges," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[11] X. Zhang, W. Yin, M. Hong, and T. Chen, "Hybrid federated learning: Algorithms and implementation," *arXiv preprint arXiv:2012.12420*, 2020.

[12] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 12, pp. 9587–9603, 2022.

[13] D. Gao, X. Yao, and Q. Yang, "A survey on heterogeneous federated learning," *arXiv preprint arXiv:2210.04505*, 2022.

[14] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[15] F. M. Shiri, T. Perumal, N. Mustapha, and R. Mohamed, "A comprehensive overview and comparative analysis on deep learning models: Cnn, rnn, lstm, gru," *arXiv preprint arXiv:2305.17473*, 2023.

[16] O. Marfoq, G. Neglia, R. Vidal, and L. Kameni, "Personalized federated learning through local memorization," in *International Conference on Machine Learning*. PMLR, 2022, pp. 15 070–15 092.

[17] G. P. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," *European journal of operational research*, vol. 160, no. 2, pp. 501–514, 2005.

[18] J. Korstanje, *The SARIMAX Model*. Berkeley, CA: Apress, 2021, pp. 125–131. [Online]. Available: https://doi.org/10.1007/978-1-4842-7150-68

[19] J. D. Hamilton, *Time series analysis*. Princeton university press, 2020.

[20] R. H. Shumway, D. S. Stoffer, R. H. Shumway, and D. S. Stoffer, "Arima models," *Time series analysis and its applications: with R examples*, pp. 75–163, 2017.

[21] V. I. Kontopoulou, A. D. Panagopoulos, I. Kakkos, and G. K. Matsopoulos, "A review of arima vs. machine learning approaches for time series forecasting in data driven networks," *Future Internet*, vol. 15, no. 8, p. 255, 2023.

[22] S.-I. Ao and H. Fayek, "Continual deep learning for time series modeling," *Sensors*, vol. 23, no. 16, p. 7167, 2023.

[23] F. M. Shiri, T. Perumal, N. Mustapha, and R. Mohamed, "A comprehensive overview and comparative analysis on deep learning models: Cnn, rnn, lstm, gru," *arXiv preprint arXiv:2305.17473*, 2023.

[24] L. R. Medsker, L. Jain *et al.*, "Recurrent neural networks," *Design and Applications*, vol. 5, no. 64-67, p. 2, 2001.

[25] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[26] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[27] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[28] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation. arxiv 2014," *arXiv preprint arXiv:1406.1078*, 2020.

[29] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[30] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[31] Y. Bengio, I. Goodfellow, and A. Courville, *Deep learning*. MIT press Cambridge, MA, USA, 2017, vol. 1.

[32] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6348–6358, 2020.

[33] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE transactions on knowledge and data engineering*, vol. 34, no. 12, pp. 5586–5609, 2021.

[34] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, pp. 41–75, 1997.

[35] Y. Chen, Y. Ning, Z. Chai, and H. Rangwala, "Federated multi-task learning with hierarchical attention for sensor data analytics," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[36] Z. Chen, E. Jiaze, X. Zhang, H. Sheng, and X. Cheng, "Multi-task time series forecasting with shared attention," in *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2020, pp. 917–925.

[37] T. Ma and Y. Tan, "Multiple stock time series jointly forecasting with multi-task learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[38] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Advances in neural information processing systems*, vol. 30, 2017.

[39] Y. Kim, E. Al Hakim, J. Haraldson, H. Eriksson, J. M. B. da Silva, and C. Fischione, "Dynamic clustering in federated learning," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[40] H. Zhang, M. Tao, Y. Shi, X. Bi, and K. B. Letaief, "Federated multi-task learning with non-stationary and heterogeneous data in wireless networks," *IEEE Transactions on Wireless Communications*, 2023.

[41] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[42] F. Siwei and Y. Han, "Multi-participant multi-class vertical federated learning," 2020.

[43] A. Shankar, L. Y. Chen, J. Decouchant, D. Gkorou, and R. Hai, "Share your secrets for privacy! confidential forecasting with vertical federated learning," *arXiv preprint arXiv:2405.20761*, 2024.

[44] I. Ceballos, V. Sharma, E. Mugica, A. Singh, A. Roman, P. Vepakomma, and R. Raskar, "Splitnn-driven vertical partitioning," *arXiv preprint arXiv:2008.04137*, 2020.

[45] J. Chen and A. Zhang, "Fedmsplit: Correlation-adaptive federated multi-task learning across multimodal split networks," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 87–96.

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

[47] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International conference on machine learning*. PMLR, 2021, pp. 2089–2099.

[48] L. Lu and N. Ding, "Multi-party private set intersection in vertical federated learning," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2020, pp. 707–714.

[49] O. Goldreich, "Secure multi-party computation," *Manuscript. Preliminary version*, vol. 78, no. 110, pp. 1–108, 1998.

[50] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," *arXiv preprint arXiv:1711.10677*, 2017.

[51] S. Chen, "Beijing multi-site air quality," https://archive.ics.uci.edu/dataset/501/beijing+multi+site+air+quality+data, 2019, uCI Machine Learning Repository, Dataset.

[52] "Solar power generation," https://www.nrel.gov/grid/solar-power-data.html, 2006, dataset.

[53] A. T. et al., “G-research crypto forecasting,” https://kaggle.com/competitions/g-research-crypto-forecasting, 2021, kaggle, Datatset.

[54] F. Knauer and W. Cukierski, “Rossmann store sales,” https://kaggle.com/competitions/rossmann-store-sales, 2015, kaggle, Dataset.

[55] C. Dwork, “Differential privacy,” in *International colloquium on automata, languages, and programming*. Springer, 2006, pp. 1–12.

[56] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, “A survey on homomorphic encryption schemes: Theory and implementation,” *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1–35, 2018.

[57] A. El Ouadrhiri and A. Abdelhadi, “Differential privacy for deep and federated learning: A survey,” *IEEE access*, vol. 10, pp. 22 359–22 380, 2022.

[58] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, “{BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning,” in *2020 USENIX annual technical conference (USENIX ATC 20)*, 2020, pp. 493–506.

Appendix A
Datasets

We use four public datasets in our evaluation: Air quality [51], Solar power [52], Crypto [53] and Rossman Sales [54]. The average correlations between features in each datasets can be found in Figure 7.

All datasets follow a similar preprocessing protocol. We select samples in a given time frame, interpolate missing values and set the remaining missing values to 0. All data is normalized for consistent comparison.

A. Air Quality

The Air Quality dataset contains hourly data of different sensory measurements by twelve weather stations in Beijing. There are approximately 35000 samples of temporal data with 11 attributes under which gasses, temperature or wind direction. For our experiments, we specifically pick four attributes PM2.5, PM10, NO2 and CO, of six weather stations Aotizhongxin, Dingling, Gucheng, Huairou, Tiantan and Wanshouxigong. The data of each weather station is used for a separate task. During preprocessing, we select approximately samples in a two-month period. PM2.5 is the endogeneous feature, all other features are exogenous.

B. Rossman Sales

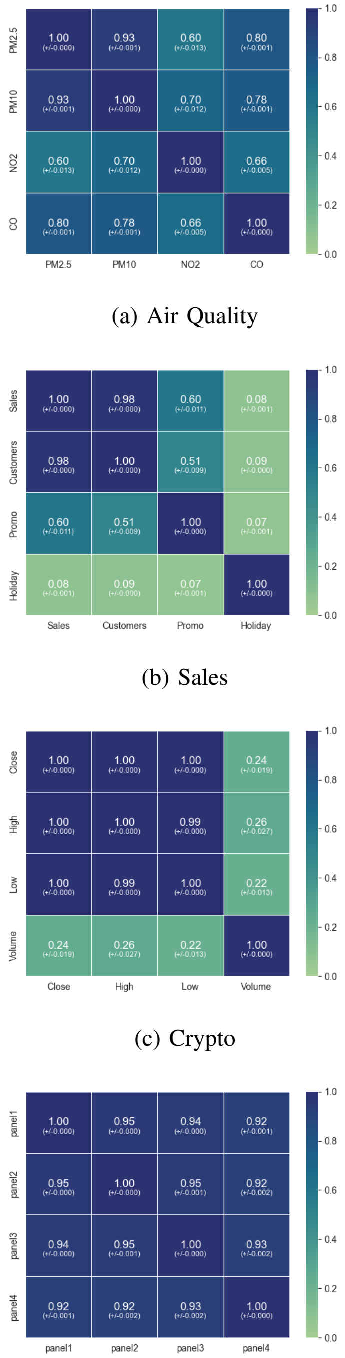
Rossman Sales contains historical sales date for 1115 Rossman stores. The data was measured on a daily basis on contains around 920 samples per store. We specifically selecte four attributes: Sales, Customers, Promo (indicating if there is a promotion), and Holiday of stores 1 to 6. Sales serves as the endogeneous feature. The last attribute is a combination of SchoolHoliday and StateHoliday which we combined during preprocessing. Specific configurations of batch size, input- and training window resulted in the use of approximately 720 samples in the experiments.

C. Crypto

The Crypto dataset contains historical trading data for different cryptocurrencies. The dataset contains a different number of samples for each currency since the initiated at different moments in history. All measurements were done per minute. We selected Close as endogenous feature; and Open, Close and Volume as exogenous features of six assets: Binance Coin, Bitcoin, Bitcoin Cash, Cardano, Dogecoin and EOS.IO. During pre-processing we select approximately 1600 samples in a two-month period and resample the data into hourly data - considering the correct aggregation function for each column.

D. Solar

Solar contains energy production measurements in MW for different solar panels located at multiple solar fields. The power consumption is measured every five minutes. We construct tasks by selecting multiple solar panels in one solar field and treating them as a task. We choose solar fields in Alabama, Florida, Illinois, Kansas, Massachusetts and Maine. We selected approximately 1800 in a 7 day period. We select one panel as endogenous feature and use other panels as exogenous features.



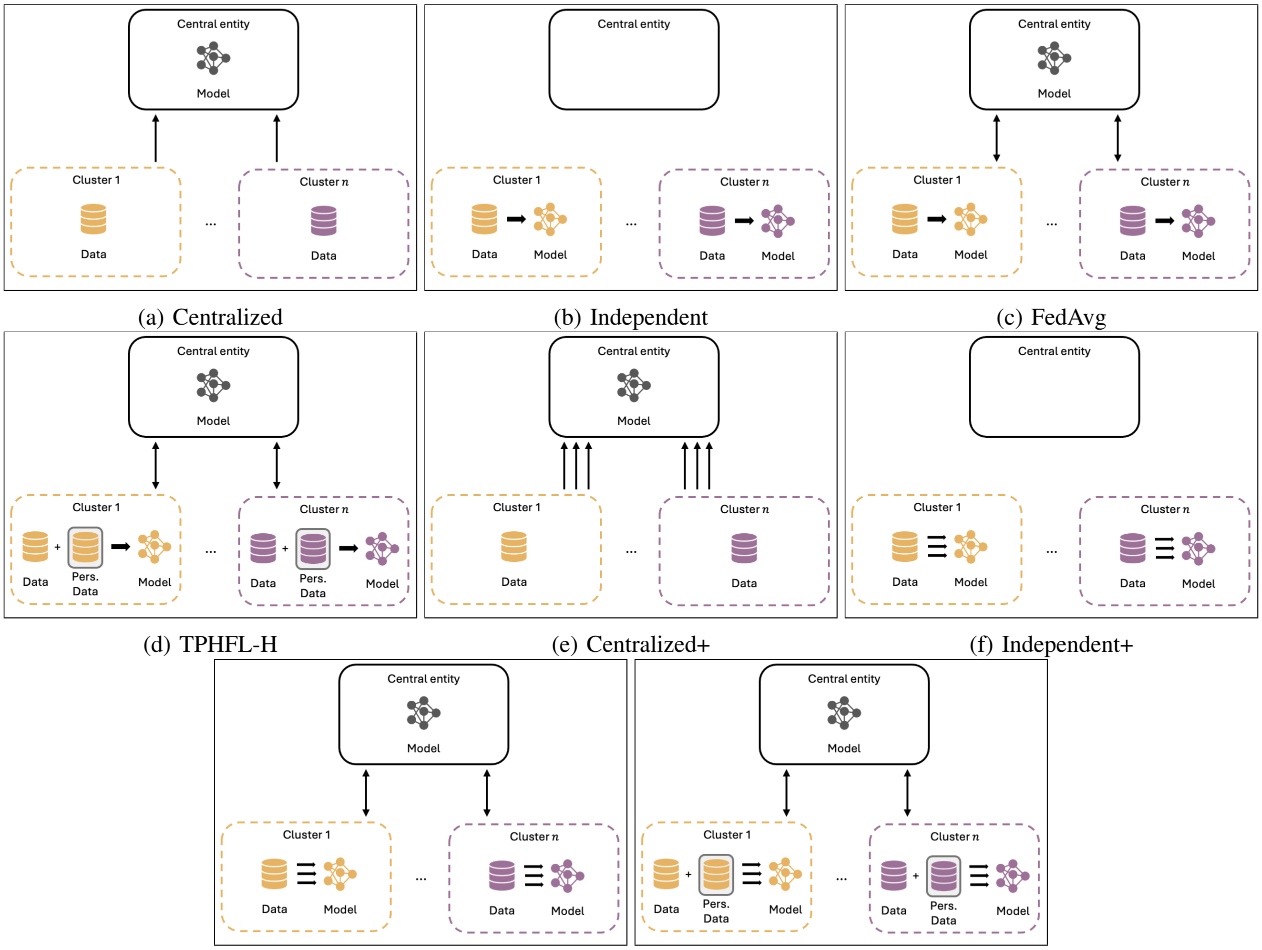
(a) Air Quality

(b) Sales

(c) Crypto

(d) Solar

Fig. 7: Correlation within tasks, each box contains the average correlation value and variance.



(a) Centralized

(b) Independent

(c) FedAvg

(d) TPHFL-H

(e) Centralized+

(f) Independent+

(g) TPHFL-NP

(h) TPHFL

Fig. 8: Schematic figures of different baselines. Multiple arrows in the last four subfigures indicate distributed features.

APPENDIX B
BASELINE

In Figure 8, we show schematic overviews of all baseline methods. In Section IV-B we discussed different forms of data locality and collaboration. These forms translate to the following configurative choices:

Data locality:

- **None**: each cluster can share its data with the central entity, which trains one model with all data (Figures 8a).
- **Horizontal**: each cluster does not share any information with others (Figures 8b) or shares its model weights with the central entity, serving as the Federator, and receives updated model weights (Figures 8c, 8d).
- **Vertical**: each cluster shares distributed features with the central entity, which trains one model with all data (Figures 8e).
- **Hybrid**: we combine the configurations of Horizontal and Vertical data locality. Each cluster trains its model using distributed features but does not share any information with the central entity (Figure 8f) or trains with distributed features and shares only model weights (Figures 8g, 8h).

For collaboration, with exclusively Vertical collaboration we do not exchange any information between the clusters (Figures 8b, 8f). In all other cases, there is exchange between clusters.

APPENDIX C
EXPERIMENTS

We compare TPHFL to all baseline methods, in three separate tables. In Table VIII, which is an expanded version of Table IV, we compare TPHFL to Independent and Centralized and show the relative improvement of TPHFL compared to Independent. In Table IX, which is an expansion of Table V, we compare three methods without vertical privacy constraints to their counterparts that enforce vertical restrictions. In Table X, which is an expansion of Table V, we compare two methods that incorporate horizontal collaboration and personalization with their counterpart that do not use personalization. Lastly, in Table XI, which is an expansion of Table VII, we show the improvements of TPHFL-I2 and TPHFL over Independent.

We conducted these experiments additionally for 2 and 4 tasks. The corresponding results can be found in Tables XII to XIX.

Dataset	PW	Independent	Centralized	TPHFL	Imp.
AirQuality	1	2.38 +/- 0.02	1.55 +/- 0.00	1.91 +/- 0.00	19.6%
	2	2.21 +/- 0.01	1.90 +/- 0.01	2.00 +/- 0.01	9.5%
	4	4.42 +/- 0.02	3.60 +/- 0.01	3.84 +/- 0.01	13.2%
	8	6.68 +/- 0.24	5.41 +/- 0.04	5.29 +/- 0.01	20.8%
	16	3.60 +/- 0.07	2.77 +/- 0.01	2.96 +/- 0.01	17.8%
	Avg.	4.17 +/- 0.46	1.30 +/- 0.02	2.71 +/- 0.11	34.9%
Industry	1	4.76 +/- 0.58	2.05 +/- 0.06	2.62 +/- 0.09	44.9%
	2	4.44 +/- 1.26	3.54 +/- 0.43	4.42 +/- 1.14	0.4%
	4	3.45 +/- 0.34	2.57 +/- 0.07	3.20 +/- 0.26	7.2%
	8	4.56 +/- 0.51	2.92 +/- 0.11	3.19 +/- 0.29	29.9%
	16	4.28 +/- 0.63	2.48 +/- 0.14	3.23 +/- 0.38	24.4%
	Avg.	2.38 +/- 0.01	2.53 +/- 0.03	3.24 +/- 0.06	-36.3%
Sales	1	2.64 +/- 0.01	2.59 +/- 0.01	3.61 +/- 0.04	-36.6%
	2	3.15 +/- 0.00	3.22 +/- 0.02	4.34 +/- 0.03	-37.7%
	4	3.65 +/- 0.01	3.58 +/- 0.00	5.32 +/- 0.02	-45.8%
	8	2.98 +/- 0.01	3.17 +/- 0.02	5.50 +/- 0.14	-84.7%
	16	2.96 +/- 0.01	3.02 +/- 0.02	4.40 +/- 0.06	-48.7%
	Avg.	2.24 +/- 0.14	1.19 +/- 0.02	2.09 +/- 0.03	6.9%
Crypto	1	1.97 +/- 0.09	1.17 +/- 0.02	1.95 +/- 0.03	1.0%
	2	2.41 +/- 0.11	1.21 +/- 0.02	2.24 +/- 0.05	7.0%
	4	2.89 +/- 0.16	2.06 +/- 0.07	2.63 +/- 0.05	9.0%
	8	3.67 +/- 0.19	2.80 +/- 0.10	3.10 +/- 0.04	15.6%
	16	2.64 +/- 0.14	1.69 +/- 0.04	2.40 +/- 0.04	8.9%
	Avg.	1.40 +/- 0.01	0.71 +/- 0.00	0.99 +/- 0.02	29.5%
Solar	1	1.81 +/- 0.02	1.44 +/- 0.02	1.43 +/- 0.03	21.1%
	2	1.67 +/- 0.02	1.13 +/- 0.00	1.34 +/- 0.02	19.4%
	4	2.60 +/- 0.04	1.80 +/- 0.02	1.86 +/- 0.04	28.3%
	8	3.36 +/- 0.07	2.04 +/- 0.02	2.27 +/- 0.08	32.5%
	16	2.17 +/- 0.03	1.42 +/- 0.01	1.58 +/- 0.04	27.2%

TABLE VIII: Average MAE and standard deviation for methods with no privacy constraints and TPHFL. The relative improvement of TPHFL over Independent (Rel. Imp.) is shown in percentages.

Dataset	PW	Independent	Independent+	Imp.	Centralized	Centralized+	Imp.	TPHFL-H	TPHFL	Imp.
AirQuality	1	2.32 +/- 0.06	2.72 +/- 0.03	-16.8%	1.40 +/- 0.01	1.58 +/- 0.01	-12.6%	1.31 +/- 0.00	1.78 +/- 0.00	-35.9%
	2	2.38 +/- 0.02	2.51 +/- 0.01	-5.6%	1.55 +/- 0.00	1.85 +/- 0.01	-19.6%	1.52 +/- 0.00	1.91 +/- 0.00	-25.3%
	4	2.21 +/- 0.01	2.46 +/- 0.03	-11.2%	1.90 +/- 0.01	2.19 +/- 0.01	-15.1%	1.91 +/- 0.00	2.00 +/- 0.01	-4.9%
	8	4.42 +/- 0.02	4.69 +/- 0.03	-6.2%	3.60 +/- 0.01	3.45 +/- 0.01	4.3%	3.43 +/- 0.01	3.84 +/- 0.01	-12.0%
	16	6.68 +/- 0.24	6.23 +/- 0.06	6.7%	5.41 +/- 0.04	4.92 +/- 0.03	9.1%	4.74 +/- 0.02	5.29 +/- 0.01	-11.6%
	Avg.	3.60 +/- 0.07	3.72 +/- 0.03	-3.3%	2.77 +/- 0.01	2.80 +/- 0.01	-0.9%	2.58 +/- 0.00	2.96 +/- 0.01	-14.8%
Industry	1	4.17 +/- 0.46	4.73 +/- 0.32	-13.4%	1.30 +/- 0.02	1.34 +/- 0.01	-2.8%	2.21 +/- 0.07	2.71 +/- 0.11	-22.5%
	2	4.76 +/- 0.58	5.68 +/- 0.37	-19.4%	2.05 +/- 0.06	1.80 +/- 0.04	12.4%	2.61 +/- 0.15	2.62 +/- 0.09	-0.7%
	4	4.44 +/- 1.26	6.66 +/- 2.08	-49.9%	3.54 +/- 0.43	3.34 +/- 0.58	5.5%	3.55 +/- 0.99	4.42 +/- 1.14	-24.6%
	8	3.45 +/- 0.34	6.18 +/- 1.24	-79.3%	2.57 +/- 0.07	3.07 +/- 0.26	-19.4%	2.79 +/- 0.27	3.20 +/- 0.26	-14.8%
	16	4.56 +/- 0.51	5.55 +/- 0.52	-21.6%	2.92 +/- 0.11	3.58 +/- 0.25	-22.6%	3.07 +/- 0.24	3.19 +/- 0.29	-4.1%
	Avg.	4.28 +/- 0.63	5.76 +/- 0.91	-34.7%	2.48 +/- 0.14	2.62 +/- 0.23	-6.0%	2.85 +/- 0.34	3.23 +/- 0.38	-13.5%
Sales	1	2.38 +/- 0.01	4.33 +/- 0.09	-82.1%	2.53 +/- 0.03	2.39 +/- 0.02	5.6%	2.27 +/- 0.02	3.24 +/- 0.06	-42.8%
	2	2.64 +/- 0.01	4.72 +/- 0.07	-78.4%	2.59 +/- 0.01	3.46 +/- 0.38	-33.7%	2.47 +/- 0.01	3.61 +/- 0.04	-46.1%
	4	3.15 +/- 0.00	4.75 +/- 0.07	-50.8%	3.22 +/- 0.02	3.36 +/- 0.01	-4.5%	3.03 +/- 0.01	4.34 +/- 0.03	-43.0%
	8	3.65 +/- 0.01	6.83 +/- 0.03	-87.1%	3.58 +/- 0.00	3.79 +/- 0.01	-5.9%	3.70 +/- 0.02	5.32 +/- 0.02	-43.6%
	16	2.98 +/- 0.01	5.24 +/- 0.09	-76.1%	3.17 +/- 0.02	3.04 +/- 0.03	4.1%	2.69 +/- 0.03	5.50 +/- 0.14	-104.0%
	Avg.	2.96 +/- 0.01	5.17 +/- 0.07	-74.8%	3.02 +/- 0.02	3.21 +/- 0.09	-6.3%	2.83 +/- 0.02	4.40 +/- 0.06	-55.3%
Crypto	1	2.24 +/- 0.14	3.50 +/- 0.43	-55.9%	1.19 +/- 0.02	1.48 +/- 0.03	-24.5%	1.32 +/- 0.01	2.09 +/- 0.03	-58.0%
	2	1.97 +/- 0.09	3.32 +/- 0.39	-68.7%	1.17 +/- 0.02	1.21 +/- 0.02	-3.6%	1.40 +/- 0.02	1.95 +/- 0.03	-39.1%
	4	2.41 +/- 0.11	2.98 +/- 0.14	-23.5%	1.21 +/- 0.02	1.28 +/- 0.02	-5.7%	1.57 +/- 0.03	2.24 +/- 0.05	-42.9%
	8	2.89 +/- 0.16	4.37 +/- 0.52	-51.1%	2.06 +/- 0.07	1.65 +/- 0.02	20.0%	1.84 +/- 0.03	2.63 +/- 0.05	-43.0%
	16	3.67 +/- 0.19	5.40 +/- 0.55	-47.0%	2.80 +/- 0.10	2.43 +/- 0.07	13.2%	2.22 +/- 0.01	3.10 +/- 0.04	-39.3%
	Avg.	2.64 +/- 0.14	3.91 +/- 0.41	-48.3%	1.69 +/- 0.04	1.61 +/- 0.03	4.5%	1.67 +/- 0.02	2.40 +/- 0.04	-43.7%
Solar	1	1.40 +/- 0.01	1.70 +/- 0.02	-21.2%	0.71 +/- 0.00	0.77 +/- 0.01	-7.8%	0.83 +/- 0.01	0.99 +/- 0.02	-19.5%
	2	1.81 +/- 0.02	2.51 +/- 0.08	-38.8%	1.44 +/- 0.02	1.19 +/- 0.02	17.5%	1.16 +/- 0.02	1.43 +/- 0.03	-23.4%
	4	1.67 +/- 0.02	2.10 +/- 0.02	-25.5%	1.13 +/- 0.00	0.99 +/- 0.01	12.1%	1.03 +/- 0.01	1.34 +/- 0.02	-30.9%
	8	2.60 +/- 0.04	3.08 +/- 0.06	-18.4%	1.80 +/- 0.02	1.82 +/- 0.01	-1.1%	1.69 +/- 0.03	1.86 +/- 0.04	-10.0%
	16	3.36 +/- 0.07	3.97 +/- 0.18	-18.0%	2.04 +/- 0.02	2.25 +/- 0.02	-10.5%	2.20 +/- 0.05	2.27 +/- 0.08	-3.2%
	Avg.	2.17 +/- 0.03	2.67 +/- 0.07	-23.1%	1.42 +/- 0.01	1.40 +/- 0.01	1.4%	1.38 +/- 0.02	1.58 +/- 0.04	-14.3%

TABLE IX: Average MAE and standard deviation for different methods with Vertical restrictions and Horizontal restrictions. We compare the performance increase for methods if we introduce vertical privacy restrictions.

Dataset	PW	FedAvg	TPHFL-H	Imp.	TPHFL-NP	TPHFL	Imp.
AirQuality	1	1.77 +/- 0.01	1.31 +/- 0.00	26.2%	2.57 +/- 0.06	1.78 +/- 0.00	30.9%
	2	2.06 +/- 0.04	1.52 +/- 0.00	26.1%	2.42 +/- 0.02	1.91 +/- 0.00	21.1%
	4	2.00 +/- 0.00	1.91 +/- 0.00	4.3%	2.23 +/- 0.02	2.00 +/- 0.01	10.0%
	8	3.63 +/- 0.01	3.43 +/- 0.01	5.7%	4.41 +/- 0.01	3.84 +/- 0.01	12.9%
	16	5.09 +/- 0.02	4.74 +/- 0.02	6.9%	5.81 +/- 0.04	5.29 +/- 0.01	9.0%
	Avg.	2.91 +/- 0.02	2.58 +/- 0.00	11.3%	3.49 +/- 0.03	2.96 +/- 0.01	15.0%
Industry	1	2.66 +/- 0.14	2.21 +/- 0.07	16.7%	3.50 +/- 0.14	2.71 +/- 0.11	22.4%
	2	3.16 +/- 0.21	2.61 +/- 0.15	17.7%	3.53 +/- 0.14	2.62 +/- 0.09	25.6%
	4	3.79 +/- 1.15	3.55 +/- 0.99	6.4%	5.27 +/- 1.46	4.42 +/- 1.14	16.1%
	8	3.83 +/- 0.31	2.79 +/- 0.27	27.2%	4.70 +/- 0.58	3.20 +/- 0.26	31.9%
	16	3.74 +/- 0.29	3.07 +/- 0.24	17.9%	4.43 +/- 0.44	3.19 +/- 0.29	28.0%
	Avg.	3.44 +/- 0.42	2.85 +/- 0.34	17.2%	4.29 +/- 0.55	3.23 +/- 0.38	24.6%
Sales	1	2.67 +/- 0.01	2.27 +/- 0.02	14.8%	7.29 +/- 0.11	3.24 +/- 0.06	55.5%
	2	2.92 +/- 0.01	2.47 +/- 0.01	15.4%	7.41 +/- 0.10	3.61 +/- 0.04	51.3%
	4	3.64 +/- 0.01	3.03 +/- 0.01	16.6%	7.86 +/- 0.08	4.34 +/- 0.03	44.8%
	8	4.53 +/- 0.03	3.70 +/- 0.02	18.3%	8.29 +/- 0.02	5.32 +/- 0.02	35.8%
	16	3.39 +/- 0.03	2.69 +/- 0.03	20.5%	7.54 +/- 0.12	5.50 +/- 0.14	27.1%
	Avg.	3.43 +/- 0.02	2.83 +/- 0.02	17.3%	7.68 +/- 0.09	4.40 +/- 0.06	42.7%
Crypto	1	1.71 +/- 0.05	1.32 +/- 0.01	22.7%	2.60 +/- 0.10	2.09 +/- 0.03	19.7%
	2	1.79 +/- 0.08	1.40 +/- 0.02	21.7%	2.78 +/- 0.06	1.95 +/- 0.03	30.0%
	4	1.74 +/- 0.04	1.57 +/- 0.03	9.9%	2.97 +/- 0.03	2.24 +/- 0.05	24.4%
	8	2.10 +/- 0.03	1.84 +/- 0.03	12.2%	2.86 +/- 0.03	2.63 +/- 0.05	7.9%
	16	2.38 +/- 0.00	2.22 +/- 0.01	6.6%	3.50 +/- 0.03	3.10 +/- 0.04	11.3%
	Avg.	1.94 +/- 0.04	1.67 +/- 0.02	14.0%	2.94 +/- 0.05	2.40 +/- 0.04	18.3%
Solar	1	1.49 +/- 0.01	0.83 +/- 0.01	44.5%	1.63 +/- 0.02	0.99 +/- 0.02	39.3%
	2	1.67 +/- 0.03	1.16 +/- 0.02	30.8%	2.24 +/- 0.04	1.43 +/- 0.03	36.3%
	4	1.43 +/- 0.01	1.03 +/- 0.01	28.3%	2.20 +/- 0.04	1.34 +/- 0.02	38.8%
	8	2.53 +/- 0.06	1.69 +/- 0.03	32.9%	2.96 +/- 0.05	1.86 +/- 0.04	37.0%
	16	3.13 +/- 0.06	2.20 +/- 0.05	29.8%	3.86 +/- 0.15	2.27 +/- 0.08	41.2%
	Avg.	2.05 +/- 0.03	1.38 +/- 0.02	32.7%	2.58 +/- 0.06	1.58 +/- 0.04	38.8%

TABLE X: Average MAE and standard deviation for different methods with Horizontal and Hybrid privacy constraints. We show the improvement of introducing the personalization mechanism.

Dataset	PW	Independent	TPHFL-I2	Imp.	Independent	TPHFL	Imp.
AirQuality	1	2.38 +/- 0.02	1.90 +/- 0.00	20.0%	2.38 +/- 0.02	1.91 +/- 0.00	19.6%
	2	2.21 +/- 0.01	2.02 +/- 0.01	8.8%	2.21 +/- 0.01	2.00 +/- 0.01	9.5%
	4	4.42 +/- 0.02	3.76 +/- 0.01	15.0%	4.42 +/- 0.02	3.84 +/- 0.01	13.2%
	8	6.68 +/- 0.24	5.19 +/- 0.01	22.3%	6.68 +/- 0.24	5.29 +/- 0.01	20.8%
	16	3.60 +/- 0.07	2.93 +/- 0.00	18.7%	3.60 +/- 0.07	2.96 +/- 0.01	17.8%
	Avg.	4.17 +/- 0.46	2.84 +/- 0.12	31.8%	4.17 +/- 0.46	2.71 +/- 0.11	34.9%
Industry	1	4.76 +/- 0.58	2.81 +/- 0.10	40.9%	4.76 +/- 0.58	2.62 +/- 0.09	44.9%
	2	4.44 +/- 1.26	4.16 +/- 1.05	6.3%	4.44 +/- 1.26	4.42 +/- 1.14	0.4%
	4	3.45 +/- 0.34	3.31 +/- 0.29	4.1%	3.45 +/- 0.34	3.20 +/- 0.26	7.2%
	8	4.56 +/- 0.51	3.29 +/- 0.32	27.9%	4.56 +/- 0.51	3.19 +/- 0.29	29.9%
	16	4.28 +/- 0.63	3.28 +/- 0.38	23.2%	4.28 +/- 0.63	3.23 +/- 0.38	24.4%
	Avg.	2.38 +/- 0.01	3.70 +/- 0.09	-55.7%	2.38 +/- 0.01	3.24 +/- 0.06	-36.3%
Sales	1	2.64 +/- 0.01	3.76 +/- 0.05	-42.3%	2.64 +/- 0.01	3.61 +/- 0.04	-36.6%
	2	3.15 +/- 0.00	4.83 +/- 0.03	-53.4%	3.15 +/- 0.00	4.34 +/- 0.03	-37.7%
	4	3.65 +/- 0.01	7.60 +/- 0.01	-108.3%	3.65 +/- 0.01	5.32 +/- 0.02	-45.8%
	8	2.98 +/- 0.01	6.72 +/- 0.15	-125.9%	2.98 +/- 0.01	5.50 +/- 0.14	-84.7%
	16	2.96 +/- 0.01	5.32 +/- 0.07	-79.9%	2.96 +/- 0.01	4.40 +/- 0.06	-48.7%
	Avg.	2.24 +/- 0.14	2.11 +/- 0.02	5.8%	2.24 +/- 0.14	2.09 +/- 0.03	6.9%
Crypto	1	1.97 +/- 0.09	1.99 +/- 0.03	-1.0%	1.97 +/- 0.09	1.95 +/- 0.03	1.0%
	2	2.41 +/- 0.11	2.39 +/- 0.05	0.8%	2.41 +/- 0.11	2.24 +/- 0.05	7.0%
	4	2.89 +/- 0.16	2.65 +/- 0.05	8.4%	2.89 +/- 0.16	2.63 +/- 0.05	9.0%
	8	3.67 +/- 0.19	3.08 +/- 0.04	16.2%	3.67 +/- 0.19	3.10 +/- 0.04	15.6%
	16	2.64 +/- 0.14	2.44 +/- 0.04	7.4%	2.64 +/- 0.14	2.40 +/- 0.04	8.9%
	Avg.	1.40 +/- 0.01	1.01 +/- 0.01	28.2%	1.40 +/- 0.01	0.99 +/- 0.02	29.5%
Solar	1	1.81 +/- 0.02	1.45 +/- 0.02	20.0%	1.81 +/- 0.02	1.43 +/- 0.03	21.1%
	2	1.67 +/- 0.02	1.35 +/- 0.03	19.2%	1.67 +/- 0.02	1.34 +/- 0.02	19.4%
	4	2.60 +/- 0.04	1.90 +/- 0.04	27.0%	2.60 +/- 0.04	1.86 +/- 0.04	28.3%
	8	3.36 +/- 0.07	2.37 +/- 0.07	29.5%	3.36 +/- 0.07	2.27 +/- 0.08	32.5%
	16	2.17 +/- 0.03	1.61 +/- 0.04	25.6%	2.17 +/- 0.03	1.58 +/- 0.04	27.2%

TABLE XI: Average MAE and standard deviation for Independent, TPHFL-I2 and TPHFL. We show the improvement over Independent.

Dataset	PW	Independent	Centralized	TPHFL	Imp.
AirQuality	1	2.13 +/- 0.03	1.91 +/- 0.05	2.09 +/- 0.00	1.8%
	2	2.17 +/- 0.00	1.80 +/- 0.01	2.11 +/- 0.00	2.9%
	4	2.16 +/- 0.00	2.03 +/- 0.00	2.21 +/- 0.00	-2.5%
	8	4.57 +/- 0.02	4.01 +/- 0.00	3.78 +/- 0.00	17.2%
	16	6.21 +/- 0.01	5.91 +/- 0.01	5.37 +/- 0.00	13.6%
	Avg.	3.45 +/- 0.01	3.13 +/- 0.02	3.11 +/- 0.00	9.7%
Industry	1	6.65 +/- 0.33	3.59 +/- 0.20	4.25 +/- 0.13	36.1%
	2	8.17 +/- 0.46	3.91 +/- 0.13	2.36 +/- 0.02	71.2%
	4	2.25 +/- 0.10	3.06 +/- 0.25	1.74 +/- 0.02	22.6%
	8	3.52 +/- 0.38	2.04 +/- 0.06	3.54 +/- 0.36	-0.5%
	16	7.19 +/- 0.28	4.12 +/- 0.37	4.91 +/- 0.06	31.7%
	Avg.	5.56 +/- 0.31	3.34 +/- 0.20	3.36 +/- 0.12	39.5%
Sales	1	2.58 +/- 0.01	2.37 +/- 0.01	2.77 +/- 0.02	-7.6%
	2	2.66 +/- 0.01	2.70 +/- 0.01	2.97 +/- 0.03	-11.4%
	4	3.32 +/- 0.00	3.16 +/- 0.00	3.44 +/- 0.02	-3.8%
	8	3.88 +/- 0.00	3.62 +/- 0.00	4.78 +/- 0.00	-23.0%
	16	2.98 +/- 0.00	2.80 +/- 0.00	4.52 +/- 0.05	-51.6%
	Avg.	3.09 +/- 0.00	2.93 +/- 0.00	3.70 +/- 0.02	-19.8%
Crypto	1	3.36 +/- 0.38	1.50 +/- 0.02	2.19 +/- 0.09	34.9%
	2	1.96 +/- 0.12	1.74 +/- 0.05	2.03 +/- 0.06	-3.2%
	4	2.24 +/- 0.07	1.14 +/- 0.01	1.63 +/- 0.02	27.3%
	8	3.31 +/- 0.24	1.82 +/- 0.05	2.60 +/- 0.16	21.5%
	16	4.90 +/- 0.26	2.29 +/- 0.02	5.10 +/- 0.29	-4.1%
	Avg.	3.15 +/- 0.21	1.70 +/- 0.03	2.71 +/- 0.12	14.2%
Solar	1	1.33 +/- 0.01	1.05 +/- 0.01	1.20 +/- 0.01	9.3%
	2	1.64 +/- 0.01	1.21 +/- 0.00	1.73 +/- 0.01	-5.5%
	4	1.58 +/- 0.00	1.20 +/- 0.00	1.54 +/- 0.02	2.5%
	8	2.19 +/- 0.03	2.03 +/- 0.05	2.14 +/- 0.03	2.4%
	16	3.24 +/- 0.14	2.35 +/- 0.03	2.68 +/- 0.07	17.4%
	Avg.	2.00 +/- 0.04	1.57 +/- 0.02	1.86 +/- 0.03	6.9%

TABLE XII: 2 tasks: Average MAE and standard deviation for methods with no privacy constraints and TPHFL. The relative improvement of TPHFL over Independent (Rel. Imp.) is shown in percentages.

Dataset	PW	Independent	Centralized	TPHFL	Imp.
AirQuality	1	2.36 +/- 0.05	1.51 +/- 0.02	2.09 +/- 0.00	11.7%
	2	2.25 +/- 0.01	1.70 +/- 0.01	1.93 +/- 0.00	14.0%
	4	2.18 +/- 0.00	1.82 +/- 0.00	1.97 +/- 0.00	9.9%
	8	4.49 +/- 0.02	3.81 +/- 0.00	3.81 +/- 0.00	15.3%
	16	6.36 +/- 0.10	5.28 +/- 0.02	4.78 +/- 0.01	24.7%
	Avg.	3.53 +/- 0.04	2.82 +/- 0.01	2.91 +/- 0.00	17.4%
Industry	1	4.56 +/- 0.64	1.59 +/- 0.02	2.59 +/- 0.09	43.1%
	2	5.91 +/- 0.77	2.23 +/- 0.03	2.76 +/- 0.04	53.2%
	4	2.61 +/- 0.13	1.77 +/- 0.03	1.96 +/- 0.01	24.9%
	8	3.04 +/- 0.28	2.72 +/- 0.13	2.69 +/- 0.16	11.5%
	16	5.08 +/- 0.86	2.39 +/- 0.06	3.17 +/- 0.25	37.6%
	Avg.	4.24 +/- 0.54	2.14 +/- 0.05	2.64 +/- 0.11	37.8%
Sales	1	2.61 +/- 0.01	2.66 +/- 0.01	2.44 +/- 0.01	6.7%
	2	2.79 +/- 0.01	2.83 +/- 0.01	2.58 +/- 0.01	7.5%
	4	3.42 +/- 0.00	3.40 +/- 0.01	3.64 +/- 0.01	-6.2%
	8	3.93 +/- 0.01	3.55 +/- 0.02	5.71 +/- 0.00	-45.2%
	16	3.13 +/- 0.01	2.88 +/- 0.01	4.46 +/- 0.02	-42.4%
	Avg.	3.18 +/- 0.01	3.06 +/- 0.01	3.77 +/- 0.01	-18.4%
Crypto	1	2.48 +/- 0.27	1.07 +/- 0.01	2.20 +/- 0.04	11.1%
	2	1.75 +/- 0.07	1.30 +/- 0.02	1.69 +/- 0.01	3.0%
	4	1.96 +/- 0.05	1.20 +/- 0.01	1.95 +/- 0.03	0.6%
	8	2.74 +/- 0.16	1.65 +/- 0.02	2.60 +/- 0.06	5.0%
	16	3.82 +/- 0.25	2.42 +/- 0.02	3.77 +/- 0.09	1.4%
	Avg.	2.55 +/- 0.16	1.53 +/- 0.02	2.44 +/- 0.05	4.2%
Solar	1	1.38 +/- 0.02	1.16 +/- 0.01	0.91 +/- 0.02	33.6%
	2	1.88 +/- 0.06	1.48 +/- 0.01	1.25 +/- 0.03	33.2%
	4	1.78 +/- 0.02	1.13 +/- 0.01	1.16 +/- 0.02	34.8%
	8	2.39 +/- 0.07	1.75 +/- 0.02	1.60 +/- 0.04	32.8%
	16	3.30 +/- 0.13	2.09 +/- 0.03	2.07 +/- 0.08	37.4%
	Avg.	2.14 +/- 0.06	1.52 +/- 0.02	1.40 +/- 0.04	34.7%

TABLE XIII: 4 tasks: Average MAE and standard deviation for methods with no privacy constraints and TPHFL. The relative improvement of TPHFL over Independent (Rel. Imp.) is shown in percentages.

Dataset	PW	Independent	Independent+	Imp.	Centralized	Centralized+	Imp.	TPHFL-H	TPHFL	Imp.
AirQuality	1	2.13 +/- 0.03	2.77 +/- 0.02	-30.1%	1.91 +/- 0.05	1.88 +/- 0.01	1.8%	1.81 +/- 0.00	2.09 +/- 0.00	-15.4%
	2	2.17 +/- 0.00	2.39 +/- 0.01	-10.3%	1.80 +/- 0.01	1.97 +/- 0.01	-9.2%	1.84 +/- 0.00	2.11 +/- 0.00	-14.6%
	4	2.16 +/- 0.00	2.30 +/- 0.01	-6.6%	2.03 +/- 0.00	2.00 +/- 0.00	1.8%	1.90 +/- 0.00	2.21 +/- 0.00	-16.2%
	8	4.57 +/- 0.02	4.49 +/- 0.00	1.7%	4.01 +/- 0.00	4.06 +/- 0.00	-1.1%	3.60 +/- 0.01	3.78 +/- 0.00	-4.9%
	16	6.21 +/- 0.01	5.76 +/- 0.01	7.3%	5.91 +/- 0.01	5.49 +/- 0.02	7.0%	5.42 +/- 0.00	5.37 +/- 0.00	0.9%
	Avg.	3.45 +/- 0.01	3.54 +/- 0.01	-2.7%	3.13 +/- 0.02	3.08 +/- 0.01	1.8%	2.92 +/- 0.00	3.11 +/- 0.00	-6.7%
Industry	1	6.65 +/- 0.33	5.91 +/- 0.20	11.1%	3.59 +/- 0.20	6.36 +/- 1.43	-77.1%	4.34 +/- 0.20	4.25 +/- 0.13	2.2%
	2	8.17 +/- 0.46	6.17 +/- 0.54	24.6%	3.91 +/- 0.13	7.11 +/- 1.67	-81.7%	4.15 +/- 0.20	2.36 +/- 0.02	43.2%
	4	2.25 +/- 0.10	3.58 +/- 0.22	-59.1%	3.06 +/- 0.25	2.13 +/- 0.05	30.2%	1.38 +/- 0.02	1.74 +/- 0.02	-26.7%
	8	3.52 +/- 0.38	4.82 +/- 0.83	-37.0%	2.04 +/- 0.06	4.72 +/- 0.14	-131.6%	2.79 +/- 0.35	3.54 +/- 0.36	-26.8%
	16	7.19 +/- 0.28	6.05 +/- 0.21	15.9%	4.12 +/- 0.37	6.98 +/- 0.33	-69.3%	3.86 +/- 0.03	4.91 +/- 0.06	-27.3%
	Avg.	5.56 +/- 0.31	5.31 +/- 0.40	4.5%	3.34 +/- 0.20	5.46 +/- 0.72	-63.3%	3.30 +/- 0.16	3.36 +/- 0.12	-1.7%
Sales	1	2.58 +/- 0.01	5.37 +/- 0.03	-108.3%	2.37 +/- 0.01	2.79 +/- 0.01	-17.9%	1.97 +/- 0.01	2.77 +/- 0.02	-40.6%
	2	2.66 +/- 0.01	5.65 +/- 0.02	-112.1%	2.70 +/- 0.01	2.72 +/- 0.01	-0.6%	2.29 +/- 0.02	2.97 +/- 0.03	-29.7%
	4	3.32 +/- 0.00	6.14 +/- 0.03	-85.2%	3.16 +/- 0.00	3.57 +/- 0.00	-12.9%	2.92 +/- 0.01	3.44 +/- 0.02	-18.0%
	8	3.88 +/- 0.00	7.81 +/- 0.01	-101.1%	3.62 +/- 0.00	4.87 +/- 0.05	-34.6%	3.47 +/- 0.00	4.78 +/- 0.00	-37.6%
	16	2.98 +/- 0.00	6.11 +/- 0.01	-104.8%	2.80 +/- 0.00	3.35 +/- 0.00	-19.5%	2.42 +/- 0.00	4.52 +/- 0.05	-86.6%
	Avg.	3.09 +/- 0.00	6.22 +/- 0.02	-101.5%	2.93 +/- 0.00	3.46 +/- 0.02	-18.1%	2.62 +/- 0.01	3.70 +/- 0.02	-41.4%
Crypto	1	3.36 +/- 0.38	5.36 +/- 1.25	-59.4%	1.50 +/- 0.02	1.74 +/- 0.06	-16.1%	1.20 +/- 0.01	2.19 +/- 0.09	-82.6%
	2	1.96 +/- 0.12	4.24 +/- 0.74	-116.2%	1.74 +/- 0.05	2.00 +/- 0.10	-14.8%	1.18 +/- 0.00	2.03 +/- 0.06	-72.2%
	4	2.24 +/- 0.07	3.14 +/- 0.04	-40.4%	1.14 +/- 0.01	1.42 +/- 0.02	-24.2%	1.27 +/- 0.01	1.63 +/- 0.02	-27.6%
	8	3.31 +/- 0.24	5.88 +/- 0.96	-77.9%	1.82 +/- 0.05	2.65 +/- 0.14	-46.0%	1.55 +/- 0.01	2.60 +/- 0.16	-67.2%
	16	4.90 +/- 0.26	7.75 +/- 0.58	-58.2%	2.29 +/- 0.02	4.28 +/- 0.04	-86.7%	2.81 +/- 0.07	5.10 +/- 0.29	-81.2%
	Avg.	3.15 +/- 0.21	5.27 +/- 0.71	-67.3%	1.70 +/- 0.03	2.42 +/- 0.07	-42.4%	1.60 +/- 0.02	2.71 +/- 0.12	-68.8%
Solar	1	1.33 +/- 0.01	1.40 +/- 0.01	-5.7%	1.05 +/- 0.01	1.05 +/- 0.01	0.4%	1.13 +/- 0.01	1.20 +/- 0.01	-6.2%
	2	1.64 +/- 0.01	2.22 +/- 0.04	-35.4%	1.21 +/- 0.00	1.37 +/- 0.00	-13.1%	1.48 +/- 0.01	1.73 +/- 0.01	-16.8%
	4	1.58 +/- 0.00	1.95 +/- 0.02	-23.2%	1.20 +/- 0.00	1.21 +/- 0.00	-0.8%	1.12 +/- 0.00	1.54 +/- 0.02	-37.9%
	8	2.19 +/- 0.03	2.87 +/- 0.13	-30.9%	2.03 +/- 0.05	2.10 +/- 0.02	-3.4%	1.98 +/- 0.03	2.14 +/- 0.03	-7.8%
	16	3.24 +/- 0.14	3.50 +/- 0.09	-7.9%	2.35 +/- 0.03	2.66 +/- 0.05	-13.3%	2.45 +/- 0.06	2.68 +/- 0.07	-9.3%
	Avg.	2.00 +/- 0.04	2.39 +/- 0.06	-19.6%	1.57 +/- 0.02	1.68 +/- 0.02	-7.0%	1.63 +/- 0.02	1.86 +/- 0.03	-13.8%

TABLE XIV: 2 tasks: Average MAE and standard deviation for different methods with Vertical restrictions and Horizontal restrictions. We compare the performance increase for methods if we introduce vertical privacy restrictions.

Dataset	PW	Independent	Independent+	Imp.	Centralized	Centralized+	Imp.	TPHFL-H	TPHFL	Imp.
AirQuality	1	2.36 +/- 0.05	2.82 +/- 0.03	-19.2%	1.51 +/- 0.02	1.65 +/- 0.00	-9.0%	1.84 +/- 0.00	2.09 +/- 0.00	-13.2%
	2	2.25 +/- 0.01	2.56 +/- 0.01	-13.9%	1.70 +/- 0.01	1.74 +/- 0.00	-2.7%	1.73 +/- 0.00	1.93 +/- 0.00	-11.4%
	4	2.18 +/- 0.00	2.32 +/- 0.01	-6.2%	1.82 +/- 0.00	1.78 +/- 0.00	2.1%	1.92 +/- 0.00	1.97 +/- 0.00	-2.3%
	8	4.49 +/- 0.02	4.84 +/- 0.02	-7.8%	3.81 +/- 0.00	3.68 +/- 0.01	3.4%	3.61 +/- 0.00	3.81 +/- 0.00	-5.5%
	16	6.36 +/- 0.10	6.08 +/- 0.03	4.3%	5.28 +/- 0.02	5.14 +/- 0.03	2.7%	5.22 +/- 0.01	4.78 +/- 0.01	8.4%
	Avg.	3.53 +/- 0.04	3.72 +/- 0.02	-5.5%	2.82 +/- 0.01	2.80 +/- 0.01	0.9%	2.87 +/- 0.00	2.91 +/- 0.00	-1.7%
Industry	1	4.56 +/- 0.64	4.47 +/- 0.45	1.9%	1.59 +/- 0.02	3.09 +/- 0.16	-94.3%	1.78 +/- 0.00	2.59 +/- 0.09	-45.2%
	2	5.91 +/- 0.77	5.40 +/- 0.45	8.7%	2.23 +/- 0.03	3.09 +/- 0.13	-38.6%	1.93 +/- 0.03	2.76 +/- 0.04	-43.1%
	4	2.61 +/- 0.13	3.55 +/- 0.17	-36.0%	1.77 +/- 0.03	2.61 +/- 0.12	-48.0%	1.50 +/- 0.01	1.96 +/- 0.01	-31.0%
	8	3.04 +/- 0.28	4.32 +/- 0.61	-42.0%	2.72 +/- 0.13	3.84 +/- 0.40	-41.3%	1.99 +/- 0.08	2.69 +/- 0.16	-35.4%
	16	5.08 +/- 0.86	5.24 +/- 0.46	-3.1%	2.39 +/- 0.06	4.81 +/- 0.38	-101.4%	2.14 +/- 0.05	3.17 +/- 0.25	-48.1%
	Avg.	4.24 +/- 0.54	4.60 +/- 0.43	-8.4%	2.14 +/- 0.05	3.49 +/- 0.24	-63.1%	1.87 +/- 0.04	2.64 +/- 0.11	-41.1%
Sales	1	2.61 +/- 0.01	5.42 +/- 0.04	-107.5%	2.66 +/- 0.01	2.63 +/- 0.01	1.3%	2.10 +/- 0.01	2.44 +/- 0.01	-15.8%
	2	2.79 +/- 0.01	5.57 +/- 0.03	-99.3%	2.83 +/- 0.01	3.17 +/- 0.02	-11.7%	2.35 +/- 0.01	2.58 +/- 0.01	-9.8%
	4	3.42 +/- 0.00	6.21 +/- 0.02	-81.5%	3.40 +/- 0.01	3.66 +/- 0.01	-7.4%	2.92 +/- 0.01	3.64 +/- 0.01	-24.7%
	8	3.93 +/- 0.01	7.81 +/- 0.03	-98.6%	3.55 +/- 0.02	3.95 +/- 0.02	-11.3%	3.51 +/- 0.00	5.71 +/- 0.00	-62.8%
	16	3.13 +/- 0.01	6.40 +/- 0.03	-104.3%	2.88 +/- 0.01	3.05 +/- 0.01	-5.9%	2.47 +/- 0.01	4.46 +/- 0.02	-81.0%
	Avg.	3.18 +/- 0.01	6.28 +/- 0.03	-97.6%	3.06 +/- 0.01	3.29 +/- 0.02	-7.3%	2.67 +/- 0.01	3.77 +/- 0.01	-41.1%
Crypto	1	2.48 +/- 0.27	4.03 +/- 0.89	-62.7%	1.07 +/- 0.01	1.62 +/- 0.03	-52.4%	1.26 +/- 0.01	2.20 +/- 0.04	-74.5%
	2	1.75 +/- 0.07	3.19 +/- 0.50	-82.5%	1.30 +/- 0.02	1.44 +/- 0.01	-10.5%	1.30 +/- 0.01	1.69 +/- 0.01	-30.2%
	4	1.96 +/- 0.05	2.52 +/- 0.06	-28.8%	1.20 +/- 0.01	1.54 +/- 0.02	-28.6%	1.34 +/- 0.01	1.95 +/- 0.03	-45.0%
	8	2.74 +/- 0.16	4.47 +/- 0.73	-63.0%	1.65 +/- 0.02	2.21 +/- 0.04	-34.2%	1.65 +/- 0.01	2.60 +/- 0.06	-58.1%
	16	3.82 +/- 0.25	6.00 +/- 0.69	-57.0%	2.42 +/- 0.02	3.08 +/- 0.07	-27.2%	2.32 +/- 0.02	3.77 +/- 0.09	-62.5%
	Avg.	2.55 +/- 0.16	4.04 +/- 0.58	-58.6%	1.53 +/- 0.02	1.98 +/- 0.04	-29.6%	1.57 +/- 0.01	2.44 +/- 0.05	-55.2%
Solar	1	1.38 +/- 0.02	1.68 +/- 0.02	-22.0%	1.16 +/- 0.01	0.88 +/- 0.01	23.7%	0.78 +/- 0.01	0.91 +/- 0.02	-17.7%
	2	1.88 +/- 0.06	2.44 +/- 0.08	-30.2%	1.48 +/- 0.01	1.52 +/- 0.01	-2.8%	1.06 +/- 0.02	1.25 +/- 0.03	-18.6%
	4	1.78 +/- 0.02	1.97 +/- 0.02	-10.3%	1.13 +/- 0.01	1.38 +/- 0.01	-21.8%	0.93 +/- 0.01	1.16 +/- 0.02	-25.7%
	8	2.39 +/- 0.07	2.95 +/- 0.08	-23.5%	1.75 +/- 0.02	1.53 +/- 0.02	12.3%	1.57 +/- 0.05	1.60 +/- 0.04	-2.4%
	16	3.30 +/- 0.13	4.07 +/- 0.26	-23.3%	2.09 +/- 0.03	2.15 +/- 0.02	-2.8%	2.01 +/- 0.06	2.07 +/- 0.08	-2.9%
	Avg.	2.14 +/- 0.06	2.62 +/- 0.09	-22.2%	1.52 +/- 0.02	1.49 +/- 0.02	1.9%	1.27 +/- 0.03	1.40 +/- 0.04	-10.5%

TABLE XV: 4 tasks: Average MAE and standard deviation for different methods with Vertical restrictions and Horizontal restrictions. We compare the performance increase for methods if we introduce vertical privacy restrictions.

Dataset	PW	FedAvg	TPHFL-H	Imp.	TPHFL-NP	TPHFL	Imp.
AirQuality	1	2.23 +/- 0.01	1.84 +/- 0.00	17.6%	2.35 +/- 0.01	2.11 +/- 0.00	10.4%
	2	2.10 +/- 0.00	1.90 +/- 0.00	9.5%	2.30 +/- 0.01	2.21 +/- 0.00	3.8%
	4	4.19 +/- 0.00	3.60 +/- 0.01	14.0%	4.46 +/- 0.01	3.78 +/- 0.00	15.3%
	8	5.73 +/- 0.01	5.42 +/- 0.00	5.4%	5.56 +/- 0.01	5.37 +/- 0.00	3.4%
	16	3.26 +/- 0.01	2.92 +/- 0.00	10.6%	3.51 +/- 0.02	3.11 +/- 0.00	11.5%
	Avg.	5.74 +/- 0.24	4.34 +/- 0.20	24.4%	4.77 +/- 0.36	4.25 +/- 0.13	11.1%
Industry	1	5.94 +/- 0.06	4.15 +/- 0.20	30.2%	3.74 +/- 0.39	2.36 +/- 0.02	37.1%
	2	2.30 +/- 0.08	1.38 +/- 0.02	40.3%	3.86 +/- 0.06	1.74 +/- 0.02	54.8%
	4	3.95 +/- 0.76	2.79 +/- 0.35	29.3%	6.78 +/- 0.45	3.54 +/- 0.36	47.8%
	8	5.88 +/- 0.12	3.86 +/- 0.03	34.4%	4.87 +/- 0.11	4.91 +/- 0.06	-0.8%
	16	4.76 +/- 0.25	3.30 +/- 0.16	30.7%	4.81 +/- 0.27	3.36 +/- 0.12	30.1%
	Avg.	2.50 +/- 0.01	1.97 +/- 0.01	21.0%	6.25 +/- 0.03	2.77 +/- 0.02	55.6%
Sales	1	2.73 +/- 0.01	2.29 +/- 0.02	16.3%	6.40 +/- 0.03	2.97 +/- 0.03	53.6%
	2	3.31 +/- 0.01	2.92 +/- 0.01	11.9%	7.27 +/- 0.02	3.44 +/- 0.02	52.6%
	4	4.14 +/- 0.00	3.47 +/- 0.00	16.1%	8.16 +/- 0.01	4.78 +/- 0.00	41.4%
	8	2.95 +/- 0.00	2.42 +/- 0.00	17.8%	6.76 +/- 0.04	4.52 +/- 0.05	33.0%
	16	3.13 +/- 0.01	2.62 +/- 0.01	16.3%	6.97 +/- 0.02	3.70 +/- 0.02	46.9%
	Avg.	1.41 +/- 0.02	1.20 +/- 0.01	15.1%	3.72 +/- 0.14	2.19 +/- 0.09	41.1%
Crypto	1	1.23 +/- 0.01	1.18 +/- 0.00	4.6%	3.63 +/- 0.09	2.03 +/- 0.06	44.3%
	2	1.70 +/- 0.01	1.27 +/- 0.01	25.2%	3.12 +/- 0.07	1.63 +/- 0.02	47.9%
	4	1.99 +/- 0.09	1.55 +/- 0.01	21.8%	4.18 +/- 0.37	2.60 +/- 0.16	37.9%
	8	3.20 +/- 0.06	2.81 +/- 0.07	12.2%	6.30 +/- 0.10	5.10 +/- 0.29	19.1%
	16	1.91 +/- 0.04	1.60 +/- 0.02	16.0%	4.19 +/- 0.16	2.71 +/- 0.12	35.4%
	Avg.	1.33 +/- 0.01	1.13 +/- 0.01	14.5%	1.65 +/- 0.01	1.20 +/- 0.01	26.9%
Solar	1	1.67 +/- 0.02	1.48 +/- 0.01	11.3%	2.23 +/- 0.04	1.73 +/- 0.01	22.4%
	2	1.47 +/- 0.01	1.12 +/- 0.00	24.1%	1.86 +/- 0.01	1.54 +/- 0.02	17.0%
	4	2.39 +/- 0.04	1.98 +/- 0.03	17.1%	2.43 +/- 0.04	2.14 +/- 0.03	12.0%
	8	2.91 +/- 0.11	2.45 +/- 0.06	15.9%	3.35 +/- 0.13	2.68 +/- 0.07	20.1%
	16	1.95 +/- 0.04	1.63 +/- 0.02	16.4%	2.30 +/- 0.05	1.86 +/- 0.03	19.3%

TABLE XVI: 2 tasks: Average MAE and standard deviation for different methods with Horizontal and Hybrid privacy constraints. We show the improvement of introducing the personalization mechanism.

Dataset	PW	FedAvg	TPHFL-H	Imp.	TPHFL-NP	TPHFL	Imp.
AirQuality	1	2.24 +/- 0.06	1.84 +/- 0.00	17.7%	2.76 +/- 0.12	2.09 +/- 0.00	24.4%
	2	1.95 +/- 0.01	1.73 +/- 0.00	11.3%	2.32 +/- 0.01	1.93 +/- 0.00	16.9%
	4	2.00 +/- 0.00	1.92 +/- 0.00	3.9%	2.15 +/- 0.01	1.97 +/- 0.00	8.7%
	8	3.96 +/- 0.01	3.61 +/- 0.00	8.9%	4.26 +/- 0.01	3.81 +/- 0.00	10.6%
	16	5.59 +/- 0.01	5.22 +/- 0.01	6.6%	5.62 +/- 0.04	4.78 +/- 0.01	15.0%
	Avg.	3.15 +/- 0.02	2.87 +/- 0.00	9.0%	3.42 +/- 0.04	2.91 +/- 0.00	14.9%
Industry	1	2.30 +/- 0.05	1.78 +/- 0.00	22.6%	3.72 +/- 0.22	2.59 +/- 0.09	30.4%
	2	2.75 +/- 0.04	1.93 +/- 0.03	29.7%	3.68 +/- 0.22	2.76 +/- 0.04	24.9%
	4	1.95 +/- 0.04	1.50 +/- 0.01	23.3%	2.88 +/- 0.06	1.96 +/- 0.01	31.8%
	8	3.16 +/- 0.37	1.99 +/- 0.08	37.1%	4.77 +/- 0.64	2.69 +/- 0.16	43.5%
	16	2.88 +/- 0.09	2.14 +/- 0.05	25.8%	4.52 +/- 0.70	3.17 +/- 0.25	29.8%
	Avg.	2.61 +/- 0.12	1.87 +/- 0.04	28.4%	3.91 +/- 0.37	2.64 +/- 0.11	32.6%
Sales	1	2.52 +/- 0.01	2.10 +/- 0.01	16.4%	6.18 +/- 0.04	2.44 +/- 0.01	60.5%
	2	2.87 +/- 0.01	2.35 +/- 0.01	18.0%	6.54 +/- 0.05	2.58 +/- 0.01	60.5%
	4	3.48 +/- 0.00	2.92 +/- 0.01	16.1%	7.30 +/- 0.03	3.64 +/- 0.01	50.2%
	8	4.19 +/- 0.01	3.51 +/- 0.00	16.4%	8.03 +/- 0.02	5.71 +/- 0.00	28.8%
	16	3.03 +/- 0.00	2.47 +/- 0.01	18.5%	6.78 +/- 0.05	4.46 +/- 0.02	34.2%
	Avg.	3.22 +/- 0.01	2.67 +/- 0.01	17.0%	6.96 +/- 0.04	3.77 +/- 0.01	45.9%
Crypto	1	1.48 +/- 0.01	1.26 +/- 0.01	14.7%	2.80 +/- 0.14	2.20 +/- 0.04	21.4%
	2	1.44 +/- 0.00	1.30 +/- 0.01	9.9%	2.56 +/- 0.06	1.69 +/- 0.01	33.9%
	4	1.44 +/- 0.01	1.34 +/- 0.01	6.8%	3.08 +/- 0.06	1.95 +/- 0.03	36.7%
	8	1.97 +/- 0.01	1.65 +/- 0.01	16.4%	2.92 +/- 0.07	2.60 +/- 0.06	10.9%
	16	2.56 +/- 0.01	2.32 +/- 0.02	9.3%	4.54 +/- 0.12	3.77 +/- 0.09	17.0%
	Avg.	1.78 +/- 0.01	1.57 +/- 0.01	11.5%	3.18 +/- 0.09	2.44 +/- 0.05	23.2%
Solar	1	1.23 +/- 0.01	0.78 +/- 0.01	36.8%	1.74 +/- 0.04	0.91 +/- 0.02	47.5%
	2	1.78 +/- 0.05	1.06 +/- 0.02	40.6%	2.13 +/- 0.06	1.25 +/- 0.03	41.2%
	4	1.30 +/- 0.01	0.93 +/- 0.01	28.9%	1.99 +/- 0.01	1.16 +/- 0.02	41.5%
	8	2.64 +/- 0.13	1.57 +/- 0.05	40.6%	3.06 +/- 0.13	1.60 +/- 0.04	47.6%
	16	3.26 +/- 0.12	2.01 +/- 0.06	38.4%	3.84 +/- 0.16	2.07 +/- 0.08	46.2%
	Avg.	2.04 +/- 0.06	1.27 +/- 0.03	37.9%	2.55 +/- 0.08	1.40 +/- 0.04	45.1%

TABLE XVII: 4 tasks: Average MAE and standard deviation for different methods with Horizontal and Hybrid privacy constraints. We show the improvement of introducing the personalization mechanism.

Dataset	PW	Independent	TPHFL-I2	Imp.	Independent	TPHFL	Imp.
AirQuality	1	2.13 +/- 0.03	1.99 +/- 0.00	6.5%	2.13 +/- 0.03	2.09 +/- 0.00	1.8%
	2	2.17 +/- 0.00	2.02 +/- 0.00	7.0%	2.17 +/- 0.00	2.11 +/- 0.00	2.9%
	4	2.16 +/- 0.00	2.19 +/- 0.00	-1.8%	2.16 +/- 0.00	2.21 +/- 0.00	-2.5%
	8	4.57 +/- 0.02	3.77 +/- 0.01	17.4%	4.57 +/- 0.02	3.78 +/- 0.00	17.2%
	16	6.21 +/- 0.01	5.26 +/- 0.00	15.3%	6.21 +/- 0.01	5.37 +/- 0.00	13.6%
	Avg.	3.45 +/- 0.01	3.05 +/- 0.00	11.6%	3.45 +/- 0.01	3.11 +/- 0.00	9.7%
Industry	1	6.65 +/- 0.33	4.21 +/- 0.15	36.7%	6.65 +/- 0.33	4.25 +/- 0.13	36.1%
	2	8.17 +/- 0.46	2.92 +/- 0.10	64.3%	8.17 +/- 0.46	2.36 +/- 0.02	71.2%
	4	2.25 +/- 0.10	1.74 +/- 0.02	22.6%	2.25 +/- 0.10	1.74 +/- 0.02	22.6%
	8	3.52 +/- 0.38	3.37 +/- 0.29	4.2%	3.52 +/- 0.38	3.54 +/- 0.36	-0.5%
	16	7.19 +/- 0.28	4.64 +/- 0.03	35.4%	7.19 +/- 0.28	4.91 +/- 0.06	31.7%
	Avg.	5.56 +/- 0.31	3.38 +/- 0.12	39.2%	5.56 +/- 0.31	3.36 +/- 0.12	39.5%
Sales	1	2.58 +/- 0.01	2.76 +/- 0.01	-7.0%	2.58 +/- 0.01	2.77 +/- 0.02	-7.6%
	2	2.66 +/- 0.01	2.89 +/- 0.03	-8.6%	2.66 +/- 0.01	2.97 +/- 0.03	-11.4%
	4	3.32 +/- 0.00	3.37 +/- 0.02	-1.5%	3.32 +/- 0.00	3.44 +/- 0.02	-3.8%
	8	3.88 +/- 0.00	5.09 +/- 0.00	-31.0%	3.88 +/- 0.00	4.78 +/- 0.00	-23.0%
	16	2.98 +/- 0.00	4.99 +/- 0.05	-67.3%	2.98 +/- 0.00	4.52 +/- 0.05	-51.6%
	Avg.	3.09 +/- 0.00	3.82 +/- 0.02	-23.8%	3.09 +/- 0.00	3.70 +/- 0.02	-19.8%
Crypto	1	3.36 +/- 0.38	2.28 +/- 0.08	32.1%	3.36 +/- 0.38	2.19 +/- 0.09	34.9%
	2	1.96 +/- 0.12	2.07 +/- 0.06	-5.6%	1.96 +/- 0.12	2.03 +/- 0.06	-3.2%
	4	2.24 +/- 0.07	1.71 +/- 0.01	23.4%	2.24 +/- 0.07	1.63 +/- 0.02	27.3%
	8	3.31 +/- 0.24	2.64 +/- 0.15	20.2%	3.31 +/- 0.24	2.60 +/- 0.16	21.5%
	16	4.90 +/- 0.26	5.21 +/- 0.25	-6.3%	4.90 +/- 0.26	5.10 +/- 0.29	-4.1%
	Avg.	3.15 +/- 0.21	2.78 +/- 0.11	11.7%	3.15 +/- 0.21	2.71 +/- 0.12	14.2%
Solar	1	1.33 +/- 0.01	1.22 +/- 0.01	8.1%	1.33 +/- 0.01	1.20 +/- 0.01	9.3%
	2	1.64 +/- 0.01	1.74 +/- 0.01	-6.4%	1.64 +/- 0.01	1.73 +/- 0.01	-5.5%
	4	1.58 +/- 0.00	1.57 +/- 0.02	0.9%	1.58 +/- 0.00	1.54 +/- 0.02	2.5%
	8	2.19 +/- 0.03	2.20 +/- 0.03	-0.2%	2.19 +/- 0.03	2.14 +/- 0.03	2.4%
	16	3.24 +/- 0.14	2.70 +/- 0.07	16.7%	3.24 +/- 0.14	2.68 +/- 0.07	17.4%
	Avg.	2.00 +/- 0.04	1.89 +/- 0.03	5.6%	2.00 +/- 0.04	1.86 +/- 0.03	6.9%

TABLE XVIII: 2 tasks: Average MAE and standard deviation for Independent, TPHFL-I2 and TPHFL. We show the improvement over Independent.

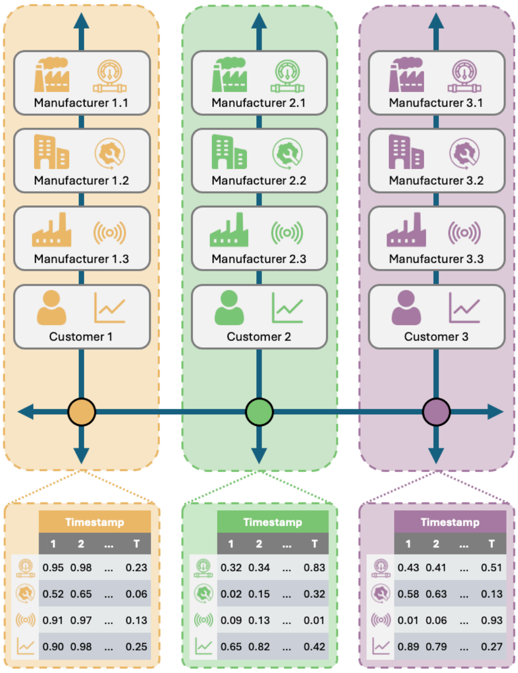
Dataset	PW	Independent	TPHFL-I2	Imp.	Independent	TPHFL	Imp.
AirQuality	1	2.36 +/- 0.05	2.04 +/- 0.00	13.5%	2.36 +/- 0.05	2.09 +/- 0.00	11.7%
	2	2.25 +/- 0.01	1.93 +/- 0.00	14.3%	2.25 +/- 0.01	1.93 +/- 0.00	14.0%
	4	2.18 +/- 0.00	1.97 +/- 0.00	9.8%	2.18 +/- 0.00	1.97 +/- 0.00	9.9%
	8	4.49 +/- 0.02	3.72 +/- 0.01	17.2%	4.49 +/- 0.02	3.81 +/- 0.00	15.3%
	16	6.36 +/- 0.10	4.79 +/- 0.00	24.6%	6.36 +/- 0.10	4.78 +/- 0.01	24.7%
	Avg.	3.53 +/- 0.04	2.89 +/- 0.00	18.1%	3.53 +/- 0.04	2.91 +/- 0.00	17.4%
Industry	1	4.56 +/- 0.64	2.76 +/- 0.12	39.3%	4.56 +/- 0.64	2.59 +/- 0.09	43.1%
	2	5.91 +/- 0.77	2.91 +/- 0.05	50.7%	5.91 +/- 0.77	2.76 +/- 0.04	53.2%
	4	2.61 +/- 0.13	1.90 +/- 0.01	27.3%	2.61 +/- 0.13	1.96 +/- 0.01	24.9%
	8	3.04 +/- 0.28	2.60 +/- 0.17	14.5%	3.04 +/- 0.28	2.69 +/- 0.16	11.5%
	16	5.08 +/- 0.86	3.16 +/- 0.21	37.7%	5.08 +/- 0.86	3.17 +/- 0.25	37.6%
	Avg.	4.24 +/- 0.54	2.67 +/- 0.11	37.1%	4.24 +/- 0.54	2.64 +/- 0.11	37.8%
Sales	1	2.61 +/- 0.01	2.64 +/- 0.01	-0.9%	2.61 +/- 0.01	2.44 +/- 0.01	6.7%
	2	2.79 +/- 0.01	2.73 +/- 0.02	2.4%	2.79 +/- 0.01	2.58 +/- 0.01	7.5%
	4	3.42 +/- 0.00	3.89 +/- 0.02	-13.6%	3.42 +/- 0.00	3.64 +/- 0.01	-6.2%
	8	3.93 +/- 0.01	6.57 +/- 0.01	-67.1%	3.93 +/- 0.01	5.71 +/- 0.00	-45.2%
	16	3.13 +/- 0.01	6.21 +/- 0.01	-98.1%	3.13 +/- 0.01	4.46 +/- 0.02	-42.4%
	Avg.	3.18 +/- 0.01	4.41 +/- 0.01	-38.6%	3.18 +/- 0.01	3.77 +/- 0.01	-18.4%
Crypto	1	2.48 +/- 0.27	2.23 +/- 0.04	10.1%	2.48 +/- 0.27	2.20 +/- 0.04	11.1%
	2	1.75 +/- 0.07	1.73 +/- 0.01	1.1%	1.75 +/- 0.07	1.69 +/- 0.01	3.0%
	4	1.96 +/- 0.05	2.14 +/- 0.04	-9.1%	1.96 +/- 0.05	1.95 +/- 0.03	0.6%
	8	2.74 +/- 0.16	2.63 +/- 0.06	4.1%	2.74 +/- 0.16	2.60 +/- 0.06	5.0%
	16	3.82 +/- 0.25	3.76 +/- 0.10	1.8%	3.82 +/- 0.25	3.77 +/- 0.09	1.4%
	Avg.	2.55 +/- 0.16	2.50 +/- 0.05	2.1%	2.55 +/- 0.16	2.44 +/- 0.05	4.2%
Solar	1	1.38 +/- 0.02	0.96 +/- 0.02	30.3%	1.38 +/- 0.02	0.91 +/- 0.02	33.6%
	2	1.88 +/- 0.06	1.28 +/- 0.03	31.5%	1.88 +/- 0.06	1.25 +/- 0.03	33.2%
	4	1.78 +/- 0.02	1.19 +/- 0.02	33.4%	1.78 +/- 0.02	1.16 +/- 0.02	34.8%
	8	2.39 +/- 0.07	1.62 +/- 0.04	32.3%	2.39 +/- 0.07	1.60 +/- 0.04	32.8%
	16	3.30 +/- 0.13	2.13 +/- 0.07	35.5%	3.30 +/- 0.13	2.07 +/- 0.08	37.4%
	Avg.	2.14 +/- 0.06	1.44 +/- 0.04	33.1%	2.14 +/- 0.06	1.40 +/- 0.04	34.7%

TABLE XIX: 4 tasks: Average MAE and standard deviation for Independent, TPHFL-I2 and TPHFL. We show the improvement over Independent.

2 Introduction

2.1. Problem Scenario

Time series forecasting is relevant in Predictive Maintenance and Control (PMC), where temporal data and models are utilized to monitor and estimate the current health state of equipment, predict future behaviour for early problem flagging, or schedule maintenance [31, 36] In industrial environments, information related to one piece of equipment is scattered over different data sources and often siloed due to privacy concerns, making it challenging to integrate and leverage them for predictive models [6, 55]. Moreover, the deployment of this equipment is geographically distributed across multiple locations, gathering data in various operational contexts with heterogeneous data distributions and following similar privacy concerns [2]. Leveraging data from multiple locations and data sources offers the potential to significantly enhance the predictive performance of models, enabling more accurate forecasting and more effective PMC.



	Timestamp			
	1	2	...	T
	0.95	0.98	...	0.23
	0.52	0.65	...	0.06
	0.91	0.97	...	0.13
	0.90	0.98	...	0.25

	Timestamp			
	1	2	...	T
	0.32	0.34	...	0.83
	0.02	0.15	...	0.32
	0.09	0.13	...	0.01
	0.65	0.82	...	0.42

	Timestamp			
	1	2	...	T
	0.43	0.41	...	0.51
	0.58	0.63	...	0.13
	0.01	0.06	...	0.93
	0.89	0.79	...	0.27

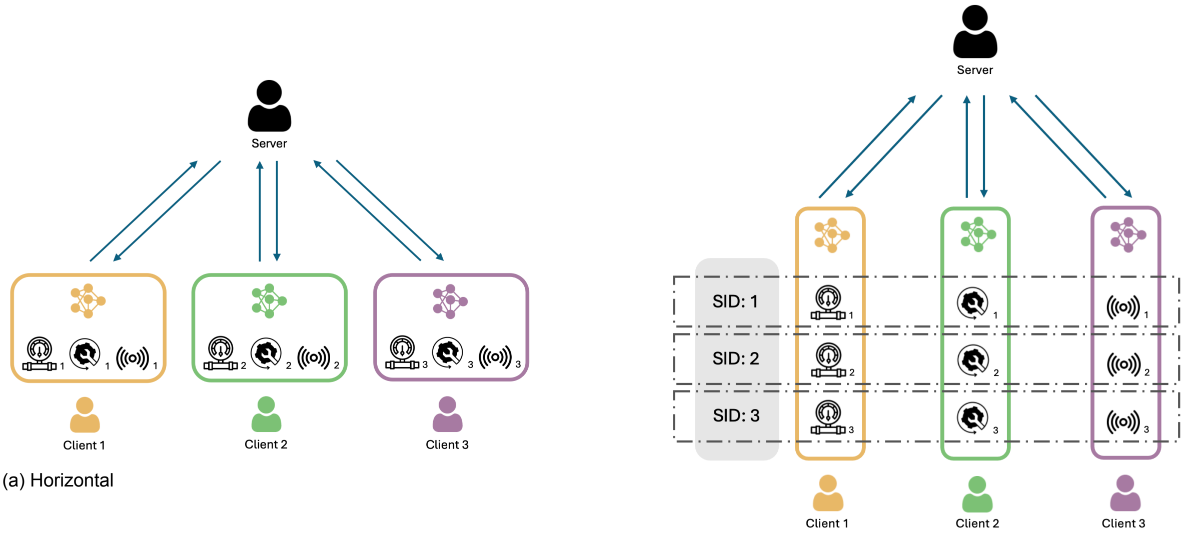
Figure 2.1: Problem scenario: three clusters with manufacturers and customers collect performance measurements. Utilization of all data requires knowledge exchange within, and between clusters.

The problem scenario in Figure 2.1 illustrates the complexities of managing and integrating data from distributed equipment. Three clusters hold three manufacturers and one customer that collect unique performance measurements of distinct machines. Three contain sensory data owned by various manufacturers, and one holds performance data owned by the customer. Each customer wants to enhance the predictions of its performance data by utilizing the data from different manufacturers and customers. The utilization of data requires a two-level knowledge exchange. On the first level, we need to exchange knowledge related to the same piece of equipment within the cluster and on the second level, we need to exchange knowledge related to the same types of equipment between clusters. However, privacy restrictions inhibit sharing at both levels and even if we could address these concerns, the heterogeneous data distributions of clusters with data from machines in different operational contexts complicate information exchange.

2.2. Existing solutions

Federated Learning (FL), more specifically Horizontal FL (HFL), addresses the privacy concerns between clusters by training a global model while only sharing local model updates [65, 42], as shown in Figure 2.2a. This method allows for the exchange of information between clusters of data belonging to the same machine whilst preserving horizontal privacy constraints. Still, it does not account for the heterogeneous task profiles and could lead to the generalization of predictive models. Multi-Task FL (MTFL) and HFL-based personalization methods address the data heterogeneity by considering the modelling of machine-specific data to be a unique task and balancing task-specific (i.e. cluster-specific) and global knowledge, or by customizing a shared model to adapt to machine-specific data, respectively [60, 56]. However, these methods do not overcome the privacy restrictions within the clusters.

Alternatively, Vertical FL (VFL) addresses these restrictions within clusters by training separate models for each party that differ in accustomed features [37], as shown in Figure 2.2b. Different models are trained separately for manufacturers and customers and exchange knowledge to improve the predictive capabilities. Since this method does not overcome privacy restrictions between clusters, HFL combined with VFL accommodates both but does not allow for heterogeneous task profiles [69]. All existing solutions lack a comprehensive approach that effectively handles knowledge exchange between and within clusters and accounts for heterogeneous profiles.



(a) Horizontal

(b) Vertical

Figure 2.2: Two types of Federated Learning. Each client holds different features or Sample IDs (SID)

.

2.3. Challenges

Existing solutions solve part of the problems for time-series forecasting or provide a solution for non-temporal data. However, a comprehensive approach that effectively handles all problems for time-series forecasting does not exist. Together with its importance in practical applications such as PMC, this shows the need for further research into collaborative time-series predictions.

The primary challenges in this domain include:

- **Horizontal knowledge exchange**: Between clusters knowledge is siloed due to privacy constraints requiring inter-task knowledge exchange between parties of different tasks while preserving data privacy. Here, we consider modelling data of different machines as a unique task, similar to the definitions in MTFL [57, 22, 60].
- **Heterogeneous task profiles**: The distinctive characteristics of each task must be accounted for when sharing knowledge between them as they may exhibit diverse profiles of time-series data due to different operational contexts. Classic HFL methods struggle to maintain accuracy when dealing with such non-IID data, which adds complexity to FL implementations [71].
- **Vertical knowledge exchange for sequential data**: Within tasks, knowledge is distributed between multiple parties and siloed due to privacy constraints requiring intra-task exchange between parties of the same task whilst preserving data privacy. We are particularly interested in time-series-based solutions which contain sequential data. Traditional models fail to capture temporal dependencies and specialized sequential models overcome these challenges [48].

2.4. Research questions

This thesis presents a novel approach to time-series forecasting in PMC by introducing a Hybrid FL strategy: Time-series-based Personalized Hybrid Federated Learning (TPHFL). Our contributions address the following research questions:

1. *How can we enable effective horizontal knowledge exchange between tasks in FL while preserving data privacy across clusters?* This question addresses the need for inter-task collaboration in FL, specifically in contexts where tasks represent different machines or systems. Each task must maintain data privacy while benefiting from shared insights across clusters.
2. *How can FL models be adapted to handle heterogeneous task profiles in time-series data, where each task's data may have distinct non-IID characteristics?* This question tackles the complexity of handling diverse time-series profiles within federated learning. The challenge lies in maintaining model performance and robustness despite variability in task characteristics and operational contexts, which classic HFL methods struggle to address.
3. *How can we enable vertical knowledge exchange for time-series data across parties with different feature sets while preserving privacy and capturing sequential dependencies?* This question focuses on intra-task collaboration in FL, where multiple parties hold different features of the same time-series data. The goal is to ensure effective knowledge sharing that respects privacy constraints and leverages sequential dependencies in the data.

3 Background

In this section, we discuss background knowledge on time series forecasting, FL and MTFL.

3.1. Time Series forecasting

3.1.1. Definitions

Time series can have different numbers of variables observed and measured over time and come in two forms: univariate and multivariate series. Univariate time series consists of measurements of only one variable and can be used in modelling to predict future values [10]. However, it can not measure the potential relationships between the variable and other influencing factors. Mathematically, we define univariate time series as a sequence of observations and predictions over time:

$$X = \{x_t\}_{t=1}^{T_1} = \{x_1, x_2, ..., x_{T_1}\} \tag{3.1}$$

$$\hat{Y} = \{\hat{y}_{T_1+t}\}_{t=1}^{T_2} = \{\hat{y}_{T_1+1}, \hat{y}_{T_1+2}, ..., \hat{y}_{T_1+T_2}\} \tag{3.2}$$

Here, x_t is a scalar value representing the observation at time t and $\hat{y}_t$ a value representing predictions of the observations at time t. X and $\hat{Y}$ have two different sizes T_1 and T_2 for their time series, allowing the observations to have a different size than the predictions. The observations in X end at time step T_1, and we predict the adjacent timesteps in $\hat{Y}$ starting at time step $T_1 + 1$

In contrast to univariate time series, multivariate time series contains measurements of two or more variables and allows for the analysis of relationships between them. When modelling the series, we typically distinguish between endogenous and exogenous variables [21, 44] . Endogenous variables are the variables in question, the ones we would like to predict, whereas exogenous variables enrich the endogenous ones. Mathematically, we define multivariate time series as:

$$X = \{X_n\}_{n=1}^{N} \tag{3.3}$$

$$\hat{Y} = \{\hat{Y}_m\}_{m=1}^{M} \tag{3.4}$$

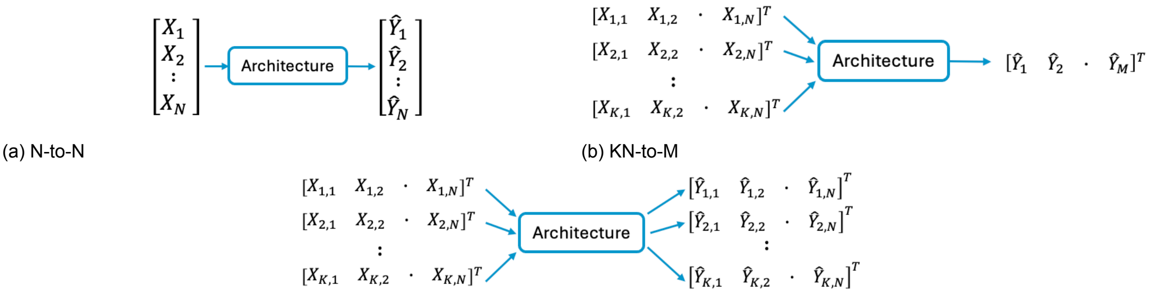
$$X_n = \{x_{n,t}\}_{t=1}^{T_1} \tag{3.5}$$

$$\hat{Y}_m = \{\hat{y}_{m,T_1+t}\}_{t=1}^{T_2} \tag{3.6}$$

In this context, X_n represents the observed values of variable n over time, while $\hat{Y}_m$ represents the m-th variable of the predictions. The number of observed time series N and predicted time series M may differ, as we forecast only part of the observed series or different variables.

In the literature, we found three configurations of multivariate series used for modelling, as shown in Figure 3.1. In these configurations, the observations serve as input and the predictions as output.

1. **N-to-N configuration**: The number of input series N equals the number of predicted time series N [40, 18, 15].
2. **KN-to-M configuration**: A larger set of input series $K \times N$ is used to predict a single set of M time series [25].


$[X_1 \; X_2 \; : \; X_N]$
Architecture
$[\hat{Y}_1 \; \hat{Y}_2 \; : \; \hat{Y}_N]$
$[X_{1,1} \; X_{1,2} \; \cdot \; X_{1,N}]^T$
$[X_{2,1} \; X_{2,2} \; \cdot \; X_{2,N}]^T$
$[X_{K,1} \; X_{K,2} \; \cdot \; X_{K,N}]^T$
Architecture
$[\hat{Y}_1 \; \hat{Y}_2 \; \cdot \; \hat{Y}_M]^T$
Architecture
$[\hat{Y}_{1,1} \; \hat{Y}_{1,2} \; \cdot \; \hat{Y}_{1,N}]^T$
$[\hat{Y}_{2,1} \; \hat{Y}_{2,2} \; \cdot \; \hat{Y}_{2,N}]^T$
$[\hat{Y}_{K,1} \; \hat{Y}_{K,2} \; \cdot \; \hat{Y}_{K,N}]^T$

(a) N-to-N

(b) KN-to-M

(c) KN-to-KM

Figure 3.1: Different configurations for modelling multivariate time series

3. **KN-to-KM configuration**: K multivariate time series are used to predict K series, each with a varying number of variables [62, 14].

3.1.2. Linear Models

Time-series forecasting involves predicting future values based on previously observed values in a sequence over time. The primary challenge in time-series forecasting lies in capturing temporal dependencies and patterns, which can be complicated by trends, seasonality, and irregularities [67]. Autoregressive Integrated Moving Average (ARIMA) has been used widely for time-series forecasting [50]. The prediction process in this model involves fitting time-series data to autoregressive (AR) and moving average (MA) components. The AR component predicts future values based on past values, while the MA component models the error as a linear combination of past errors.

ARIMA with eXogenous inputs (ARIMAX) extends on ARIMA by incorporating exogenous variables that might influence the time series [26]. ARIMAX predicts the output $\hat{y}_t$ at time t using a polynomial that combines p past values y_{t-i} for $i \in [p]$, q past errors ϵ_{t-j} for $j \in [q]$, current error ϵ_t and some constant μ:

$$\hat{y}_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \sum_{k=1}^{m} \beta_k x_{t-k} + \epsilon_t \tag{3.7}$$

Estimating the coefficients μ, ϕ_i and θ_j is done by methods such as MLE [30, 54] or LS [27].

Seasonal ARIMA (SARIMA) and Seasonal ARIMAX (SARIMAX) are an extension on ARIMA and ARIMAX, respectively, by introducing the ability to include seasonality in the estimation, allowing the model to account for recurring patterns [34]. Linear models, in general, are suitable for small datasets and have a low time complexity but often struggle with complex, non-linear relationships within the data.

3.1.3. Deep Learning Models

Deep learning models have become increasingly popular in recent years for time-series forecasting due to their ability to capture complex temporal dependencies [7, 49]. Different from linear models, deep learning models excel in learning non-linear relationships. In deep learning, we typically build a neural network by combining multiple building blocks that transform the input data. At the core, there are two crucial operations for these blocks:

- **Linear transformation**: a neural network processes data through transformations where input data x is multiplied by weight matrix W and added to bias b. The outcome z can defined as:

$$z = Wx + b \tag{3.8}$$

- **Activation functions**: after applying the linear transformation, the activation function introduces non-linearity, allowing the model to capture more complex relationships. Common activation functions include:

- **Sigmoid**: Squeezes the values between 0 and 1

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.9}$$

- **Tanh (hyperbolic tangent)**: Squeezes the values between -1 and 1

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{3.10}$$

The neural network is optimised by adjusting the weights and biases through an algorithm such as stochastic gradient descent [5] . The network can make predictions by feeding data through each block and using the final output as a prediction. Certain combinations of model components with specific transformations are good for capturing temporal dependencies. We will discuss four of them.

Recurrent Neural Networks

Recurrent Neural Networks (RNNs) (Figure 3.2a) were one of the first neural architectures designed for sequential data, enabling the network to maintain an evolving hidden state to capture temporal patterns [43, 64]. RNNs work by sustaining a hidden state that is updated at each time step based on the current input and previous hidden state, allowing the model to retain information from preceding time steps to capture temporal dependencies in sequential data. The equation for RNN is defined by:

$$h_t = tanh(W_h h_{t-1} + W_x x_t + b_h) \tag{3.11}$$

$$y_t = W_y h_t + b_y \tag{3.12}$$

Here, h_t, x_t and y_t are the hidden state, input and predicted output at time t, respectively, and W_h, W_x, W_y, b_h and b_y learnable weights and biases. However, traditional RNNs have limited ability to model long-term dependencies effectively because of the vanishing gradient problem [29]. Gradients that capture long-term dependencies become increasingly small during backpropagation.

Long Short-Term Memory

Long Short-Term Memory (LSTM) (Figure 3.2b) addresses this limitation by incorporating memory cells and gating mechanisms to selectively retain relevant information over longer time sequences, making them more robust for complex time-series tasks [49, 47]. LSTMs introduce memory cells and gates (input gate, forget gate and output gate) that store knowledge and trigger parts in the network. These triggers selectively retrain or discard previous information. We construct each gate with the same equation:

$$g_t = \sigma(W_g \cdot [h_{t-1}, x_t] + b_g) \tag{3.13}$$

Here, the gate g_t activates the linear transformation of the previous hidden state h_{t-1} and current input x_t. The forget gates, input gates and output gates f_t, i_t and x_t, are used to (de)activate cells:

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3.14}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{3.15}$$

$$h_t = x_t \cdot tanh(C_t) \tag{3.16}$$

In these equations, C_t is the cell state and $\tilde{C}_t$ the candidate cell state at time t. $\tilde{C}_t$ follows a similar equation as the gates but with a different activation function. The current cell state C_t is a combination of the previous state and candidate state C_{t-1} and $\tilde{C}_t$ that are (partially) included depending on the forget gate and input gate. This cell captures the long-term dependencies. The hidden state is the product of the candidate state and the output gate and captures the short-term dependencies.

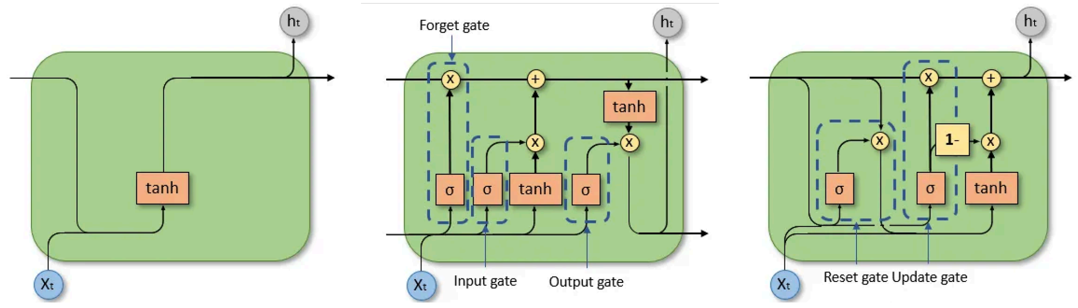
Gated Recurrent Units

Gated Recurrent Units (GRUs) (Figure 3.2c), a simplified variant of LSTMs, offer similar performance while reducing computational complexity by merging some of the gating mechanisms [49, 16]. They eliminate the memory cells in LSTMs and combine the input and forget gate into a single update gate z_t, and replacing the output gate by reset gate r_t (both gates follow Equation 3.13). We mathematically define the GRU as:

$$\tilde{h}_t = tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t] + b_h) \tag{3.17}$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \tag{3.18}$$

Here, the candidate hidden state $\tilde{h}_t$ follows a transformation and activation, including the reset gate r_t. The hidden state h_t is a combination of the previous states and candidate states that depend on a single update gate z_t.



(a) RNN

(b) LSTM

(c) GRU

Figure 3.2: Different types of sequential deep learning models [3]

Attention-based

Attention-based methods enhance interpretability and performance in time-series forecasting. These methods rely on attention mechanisms to dynamically focus on different parts of the input sequence, providing high accuracy and insights into the underlying temporal relationships [45].

An example is the transformer [59], of which the core component is the multi-head self-attention mechanism, described as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.19}$$

Here, Q, K and V are query, key and value matrices that represent the input sequence, d_k is the dimensionality of the keys, and softmax is an activation function that squeezes values between 0 and 1 for cases where more than 1 class is involved. These methods can have high-performance gains compared to LSTMs or GRUs. However, due to the increased model complexity, these models risk overfitting, especially with small datasets [9].

3.2. Federated learning

In this section, we will first discuss HFL and VFL. We will not go into detail on VFL because we will discuss the principles of VFL and common methodologies in Chapter 4.2. To overcome heterogeneous data distributions in FL, we will research two methods: MTFL and personalization.

3.2.1. Horizontal Federated Learning

HFL allows N clients with different samples and the same features to collaboratively train a machine-learning model without sharing their input data [65]. Instead, they share locally computed model updates, such as gradients or model parameters, with a central server. In FedAvg [42], clients share model weights with a central server, the Federator, which aggregates these, updates and redistributes the aggregated model back to the clients:

$$\theta_{\text{Global}} = \frac{1}{N}\sum_{n=1}^{N}\theta_n \tag{3.20}$$

By aggregating local models, we try to find a model θ that minimizes the loss over the samples of all clients:

$$\min_{\theta}\mathcal{L}_{\theta}(S) = \sum_{n=1}^{N}\frac{1}{|P_n|}\sum_{i\in P_n} f(\theta; x_i, y_i) \tag{3.21}$$

In this equation, dataset $S = \{(x_i, y_i)\}_{i=1}^{D}$ contains D samples that is partitioned, with P_n the set of indices belonging to party n, and f the loss of prediction on sample (x_i, y_i) given some model θ.

HFL is particularly advantageous for maintaining data privacy, as sensitive information remains localized at each clients' site. Liu et al. used FedAvg for anomaly detection in time series by using a combination of an attention mechanism and LSTM as a model [38].

3.2.2. Vertical Federated Learning

VFL addresses scenarios where different clients hold different features of the same samples. In this approach, multiple clients train a partitioned model on a set of distributed features across [37]. This configuration is commonly encountered in industries like healthcare, finance, and marketing, where organizations might collaborate to build a comprehensive predictive model but cannot share raw data due to privacy concerns, legal regulations (e.g., GDPR [23]), or competitive interests. In VFL, collaborators train a joint model in which we want to minimize the loss while preserving the privacy of data among the different parties:

$$\min_{\theta}\mathcal{L}_{\theta}(S) = \frac{1}{D}\sum_{i=1}^{D}\mathcal{L}(\psi_N(\theta_1(x_{i,1}), \ldots, \theta_N(x_{i,N})); y_i) \tag{3.22}$$

In this equation, the joint model θ can be decomposed into local models θ_n for $n \in [N]$ and global module ψ_N. Feature vector x_i is distributed across N parties, each having their private share $x_{i,n}$ and local model G_i for $n \in [N]$. Party N has the label information y_i and access to the global module ψ_N.

3.2.3. Multi-task Federated Learning

To tackle data heterogeneity in FL, various solutions have been proposed, particularly MTFL and personalization [57, 22]. In the literature, there is no clear distinction between the two techniques. We will discuss the techniques preceding MTFL and personalization to decouple these terms.

MTFL is a federated adaptation of Multi-Task Learning (MTL) [70, 11]. MTL learns different related tasks jointly, allowing knowledge to be shared between tasks. For example, imagine we want to classify dogs and cats. Classifying different animals can be considered distinct tasks, but since they are both animals, they are related and can share knowledge to improve the classification accuracy. MTFL extends MTL by treating clients as a unique task, creating the potential to capture client relationships.

MTL is often associated with incremental learning [57, 60]. Though these terms are often exchanged, incremental learning refers to a single model adapting to changing data distributions - the data distribution evolves - making it useful in cases where data may change over time due to various factors, such as drift in data or the introduction of new data sources [60]. In our context, we use MTL for scenarios where we learn existing tasks with unique data distributions, making it different from incremental learning.

Van de Ven et al. [60] distinguish three incremental learning scenarios for learning different mappings. We can also use these mappings in MTL because they tell us how we transform our task-specific input into labels. In the mappings, $\mathcal{X}$ denotes the input space, $\mathcal{Y}$ the within-context label space and $\mathcal{C}$ the context space:

1. **Task-incremental learning**: $f : \mathcal{X} \times \mathcal{C} \rightarrow \mathcal{Y}$ Task-incremental learning allows a model to receive the input data and a task label. It outputs a within-context label from a shared label space. The label can mean the same or be different for the different tasks. An example is the classification of

the genders of various animals. The model receives the task label of the animal to help classify the gender.

2. **Domain-incremental learning**: $f : \mathcal{X} \rightarrow \mathcal{Y}$ In domain-incremental learning, there are no task labels. The model receives the input data and labels them independent of the task. The model does not receive the task label and determines the gender.

3. **Class-incremental learning**: $f : \mathcal{X} \rightarrow \mathcal{C} \times \mathcal{Y}$ Lastly, in class-incremental learning, the model must infer the task label based on the input, allowing the output space to grow to every possible combination of within-context and task labels. The model receives a data sample and determines to which task it belongs. The model then labels each sample with the type of animal and gender.

Our emphasis is on domain-incremental scenarios where multiple tasks are trained concurrently. To demonstrate how objective functions can differ from traditional HFL, we take the shared-private attention mechanism as an example. In this mechanism each client has a shared component and local component. The shared information allows for generalization while using task-specific parameters to balance global and private information [14, 15, 40]. Mathematically, we can extend Equation 3.21 to an optimization function that takes into account the shared representations and task-specific parameters:

$$\min_{\Theta,h} \mathcal{L}_{\Theta,h}(D) = \sum_{n=1}^{N} \frac{1}{|P_n|} \sum_{i \in P_n} f(\theta_n; h; x_i, y_i) \tag{3.23}$$

In this equation, we try to minimize the loss of all tasks by finding the optimal shared representation h and model parameters θ_k for each task. The model parameters serve as task-specific parameters. For each task k, the loss has a prediction function f_k that depends on task-specific input x_k, representation h and parameters θ_k, and the target prediction y_k. We will discuss the mechanism in more detail in Chapter 4.1.1.

3.2.4. Personalization

Personalization extends the HFL paradigm and customizes global models individually for clients with unique data distributions [57, 22]. The personalized model better suits the underlying client distributions. Unlike earlier MTFL methods, its objective is not to learn distinct tasks but to adapt a global model, such as the model in Equation 3.20, to local data distributions, i.e. generalization is followed by personalization.

FedProx extends the FedAvg method, addressing data heterogeneity by allowing clients to train their local models independently [35]. In Figure 3.3, we see how a client trains its model for multiple training epochs locally, sending its model weights to a Federator afterwards. It receives the aggregated model weights and trains this new model for numerous training epochs locally. The method introduces a proximal term used during training to ensure the local models do not deviate significantly from the global model, allowing the client to balance personalization and generalization. The objective function for each client can be formulated as:

$$\min_{\Theta} \mathcal{L}_{\Theta}(D, \theta_{\text{Global}}) = \sum_{n=1}^{N} \frac{1}{|P_n|} \sum_{i \in P_n} (f(\theta_n; x_i, y_i) + \frac{\mu}{2} ||\theta_n - \theta_{\text{Global}}||^2) \tag{3.24}$$

In this equation, $\mathcal{L}_k$ is the loss function for client k, w is the local model's parameters, w_g is the global model's parameters, and μ is a regularization term. While FedProx mitigates client drift and improves convergence under heterogeneous distributions, the performance increase over FedAvg is limited.

FedPer and FedRep approach personalization differently by splitting the model into global and local components [8, 17]. Both methods train the global part collectively and local components independently for each client. Figure 3.4 shows a schematic overview of FedRep that clearly shows this distinction between global and local components. The objective of this model looks similar to the objective for MTFL with shared-private attention mechanisms in Equation 3.23. However, the difference between the two methods lies in how we create shared/global parts and local parts. In personalization approaches, we split the weights of one model (global/local), whereas in MTFL, we partition models and representation (shared/local).

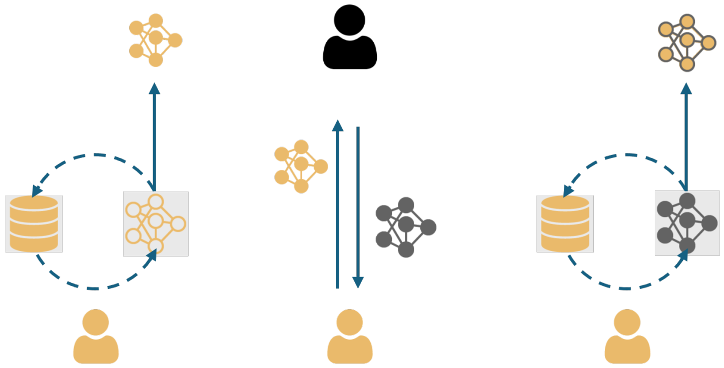


Figure 3.3: Schematic overview of FedProx: clients train their local model, share it with a Federator and retrain the global model for personalization

Both methods choose the model's head, or representation layer, as a local component. The difference between both methods is that FedPer updates the global and local components simultaneously, resulting in the same number of updates for both components, whereas FedRep separates the local updates from the global updates. The method performs local updates for multiple rounds and global updates only once during each epoch. However, critical personalized features may not be encapsulated adequately by this component.

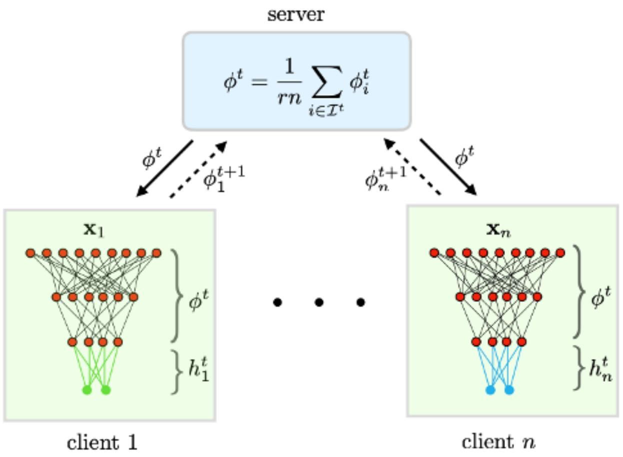


Figure 3.4: Scematic overview of FedRep: client models are split into global and local components. We only share the global components with a central server [17]

.

4 Methodology

In this chapter, we discuss our methodology approach developed to tackle the primary challenges of collaborative learning for time-series forecasting: horizontal knowledge exchange, heterogeneous data distributions and vertical knowledge exchange. Our goal is to design an architecture that enables learning across clients while preserving data locality and ensuring adaptability to each unique client data characteristic.

We structure the methodology in three parts. First, we explore methods to handle horizontal knowledge exchange for heterogeneous data distributions. We examine shared-private attention mechanisms [14, 15, 40], clustering techniques [52, 32, 68], and memorization-based personalization [41]. Second, we investigate approaches for vertical knowledge exchange such as MMVFL [51], Secure Multi-Party Computation (SMPC) [46] and Split Learning [12, 15] to enable collaboration of clients with different feature sets. Finally, we present our integrated solution, which combines several methods into a unified framework.

4.1. Solving horizontal knowledge exchange for heterogeneous data

To address the challenge of horizontal knowledge exchange among clients with heterogeneous data distributions, we evaluated multiple methods that allow knowledge sharing while accommodating unique client data patterns. This section investigates two MTFL approaches: shared-private attention mechanisms and clustering techniques. Our investigation highlights the limitations of these initial approaches, including issues with model complexity, overfitting, and insufficient task-specific learning. Recognizing these challenges, we explore an alternative solution: memorization-based personalization, achieving more effective knowledge exchange for time-series data.

4.1.1. Initial approach: MTFL

We focus on two MTFL strategies, shared-private attention mechanisms and clustering techniques, treating their clients as unique tasks and allowing selective knowledge sharing across clients. Shared-private attention leverages shared components for generalization while retaining private components for local adaptation, and clustering organizes clients into groups with similar data profiles, creating cluster-specific models. In both methods, we use unique models or model components for different tasks.

Shared-private attention

In the literature, we found three methods that utilize shared-private attention: FATHOM, MTL-Trans and MSJF [14, 15, 40]. These mechanisms are a powerful approach in FL designed to address the challenge of task heterogeneity, where different clients or tasks may have distinct data distributions but also share some common underlying structures. By leveraging a combination of shared and private attention layers, these models can balance learning global representations (shared across clients) and local task-specific representations (unique to each client), improving the performance and adaptability of the model. As an example of a shared-private attention mechanism, we show the architecture of FATHOM in Figure 4.1. A shared-private attention architecture contains two parts:

1. **Shared attention**: Known as global attention; this component captures the common global representations across all tasks. The shared attention is one component shared by all tasks and trained by all tasks collectively.

2. **Private attention**: Known as task-specific attention; this component captures the client-specific local representations. The private attention layer computes its weights based only on local client data, allowing for a personalized representation that does not rely on shared data.

The architecture combines shared and private attention by concatenating the latent representations of both attentions or by applying one attention mask to the representation of the other. The integrated representation is fed through a final layer to make predictions. All clients collectively train the architecture model as a whole, meaning that the model behaves as a single, unified entity across all clients, limiting the ability to perform additional local iterations. While shared-private attention models heterogeneous data, it has the risk of overfitting due to increased architectural complexity [9]. Furthermore, it is impossible to perform additional local training cycles for each client because the client depends on the input of other clients to complete the cycle.

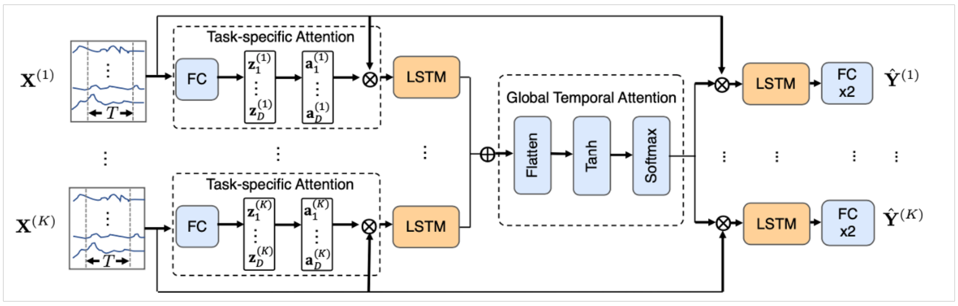


Figure 4.1: The architecture of FATHOM [14]

Clustering

Clustering offers a solution for heterogeneous data distributions by grouping clients with similar data distributions or tasks into clusters [72]. Typically, the similarity depends on statistical distribution, model weights or predictions. Each cluster learns a cluster-specific model that better aligns with the underlying data distributions of the clients in each group. We demonstrate this with four clients in Figure 4.2, each with a distinct data distribution. All four are members of one of two clusters with a cluster-specific model. Clients do not exchange any knowledge with clients of other clusters. However, the exchange of model-specific information between clusters is permitted, e.g. model weights.

In the literature, we found three variants of clustering in FL: static clustering, dynamic clustering, and hierarchical clustering. Static clustering groups clients based on initial data similarities and maintains these clusters throughout the training process, as seen in early methods like MOCHA [52]. However, this variant does not adapt to changing data distributions, which is specifically relevant for temporal data. Dynamic clustering adapts over time by adjusting cluster membership based on evolving data distributions, overcoming the limitations of static clustering in non-stationary environments [32]. Finally, hierarchical clustering organizes clients into multi-level clusters, allowing for more granular control and scalable aggregation, enhancing model performance across diverse clients [68].

Experiments and conclusions

Shared-private attention and clustering have trouble balancing task-specific and global information, as shown in an evaluation of experiments for both methods in Appendix A and B, respectively. Our implementation of various configurations of shared-private attention could not capture valuable shared patterns with the global attention unit due to overfitting or the model leaning more towards generalization. Additionally, these complex architectures had trouble fitting the data due to the limited number of data samples available. With clustering, we could not find a configuration of clustered clients in which all clients benefited. These clusters are over-generalized due to a wrong choice of similarity metric or clusters not being able to preserve task-specific information for dissimilar tasks. Cluster models will

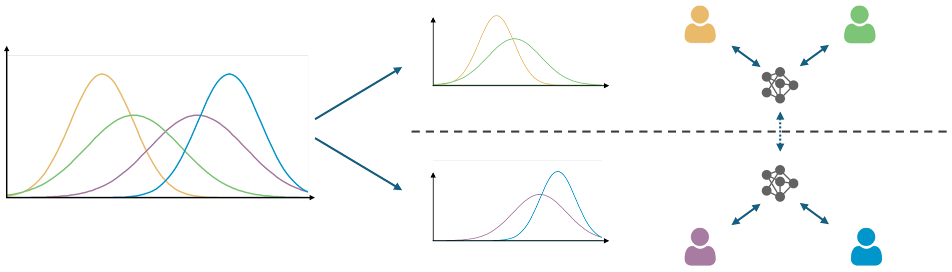


Figure 4.2: Clustering in FL: we cluster clients with similar data distributions and train one model per cluster. Clusters are allowed to exchange model-specific information.

always generalize to their clients, which is not a problem if the tasks align perfectly. Otherwise, these models will lose task-specific knowledge in the training process .

We must explore different methods for sharing horizontal knowledge with heterogeneous data to overcome these challenges. Shared-private attention could be optimized by increasing the amount of training data, but the availability of samples in predictive maintenance can be scarce for cases where sensor measurements are expensive to produce. We could choose a different similarity metric for clustering, but that does not overcome the generalization of cluster models. Therefore, we will look into more lightweight models that do not require large datasets and do not suffer from potential generalization by cluster models, global models or global components.

4.1.2. Solution: memorization-based personalization

Personalization is an approach that customizes global models for individual clients to better suit their underlying data distributions. Different from earlier MTFL methods, its objective is not to learn distinct tasks but to adapt a global model, e.g. in FedProx and FedPer [35, 8]. Memorization-based approaches provide a specific way of personalization by allowing clients to leverage their local data more directly for predictions [41]. The global model is not adapted to local data or split into private-shared partitions. Memorization techniques leave the global model as is and focus on using non-parametric methods that store and utilize client-specific data samples for inference. This approach is particularly well-suited for cases with high data heterogeneity, where adapting a global model may not sufficiently capture the unique patterns present in clients' data.

KnnPer [41] is a memorization-based method that uses k-nearest neighbour (KNN) to make personalized predictions. KnnPer memorizes the clients' training samples and compares new inputs with memorized samples to make predictions. This way, KnnPer, or memorization in general, bypasses the need for global model parameter updates by using stored data points to make new predictions.

The so-called KNN-prediction of a new input x is made by finding the k nearest neighbours in the clients' training data and averaging their labels (a simplified version of the original formula):

$$\hat{y}_{KNN} = \frac{\sum_{\{\phi(x'),y\}\in N_k} K(d(\phi(x),\phi(x'))) \cdot y}{\sum_{\phi(x')\in N_k} K(d(\phi(x),\phi(x')))} \tag{4.1}$$

Here, $\hat{y}_{KNN}$ is the KNN-prediction, $N_k(x)$ are the k nearest neighbours for x based on representation ϕ. $N_k(x)$ contains the representations and labels y of the neighbours. d is a distance metric for two representations ϕ. The authors use the hidden state of an LSTM as ϕ.

This formulation allows KnnPer to make predictions by comparing the input to previously encountered examples, bypassing the need for complex global model training. The distance metric d, typically chosen as Euclidean distance or another similarity measure, plays a key role in determining the influence of neighbours' labels on the final prediction.

This method has several advantages. By using client-specific data directly for predictions, memorization methods like KnnPer offer a high degree of personalization, making them well-suited for heterogeneous data. Additionally, memorization focuses on stored local data and naturally adapts to changing data distributions. As new data arrives, it can be directly stored and used for future predictions without requiring model re-training, which is particularly useful in environments where data evolves.

Experiments and conclusions

In the experiments in Appendix C, we evaluate traditional personalization mechanisms FedProx and FedPer, and an implementation of memorization-based mechanism KnnPer suitable for sequential data. From the traditional methods, only FedProx showed improvements for univariate datasets. We did not measure these improvements for multi-variate datasets due to the limited number of training epochs. KnnPer, on the other hand, showed significant improvements over the traditional personalization methods for multi-variate datasets. By leveraging local memorization of training samples, KnnPer provides personalized predictions better aligned with client-specific data distributions, making it particularly effective for heterogeneous time-series data.

Memorization is highly effective for horizontal knowledge exchange with privacy constraints. Clients do not share their private data and can have heterogeneous data distributions. Having found a solution for horizontal knowledge exchange with heterogeneous data, we need to find a method that provides vertical knowledge exchange and merge it into our current solution.

4.2. Solving vertical knowledge exchange

VFL addresses scenarios where different clients hold different features of the same samples. In this approach, we train a global model on a set of distributed features across multiple clients [37]. We encounter this configuration commonly in industries like healthcare, finance, and marketing, where organizations might collaborate to build a comprehensive predictive model but cannot share raw data due to privacy concerns, legal regulations (e.g., GDPR), or competitive interests. The training protocol for VFL generally follows two steps [61, 37]:

1. **Entity alignment**: In VFL, clients possess different feature sets for the same entities (e.g. users, customers or patients). Since each client holds only a subset of features for these entities, it is important to align them before training any model. First, clients identify matching entities because they are usually not publicly available due to privacy constraints. Techniques such as Private Set Intersection [39] or SMPC [24] perform alignment between clients without disclosing any private information.

2. **Distributed training**: After identifying common samples and aligning them, clients can collaboratively train the model in a distributed fashion. We will discuss different approaches for doing this.

4.2.1. Modelling approaches

MMVFL is a method that securely transfers labels between clients without sharing raw data, particularly in scenarios where clients hold different features of the same data samples (Figure 4.3). In MMVFL, each client trains a local model using their respective features, and the central server aggregates the predicted labels from all clients. This approach allows label sharing across clients while maintaining privacy [51]. However, MMVFL is limited to closed-form models and is unsuitable for more complex models like neural networks, making it unsuitable for our solution.

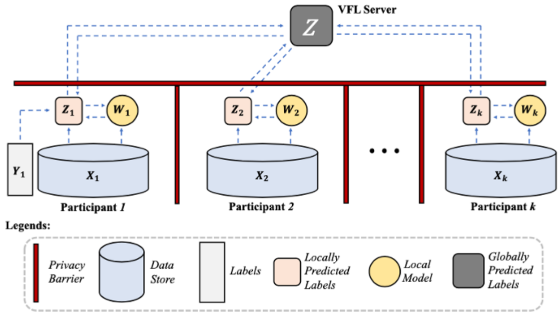


Figure 4.3: Schematic of MMVFL [51]

SMPC is a cryptographic technique that enables multiple parties to collaboratively compute a function over their inputs while keeping those inputs private. In the context of VFL, SMPC allows the training process to proceed securely by encrypting the data from each client and only sharing encrypted features for sequential linear models [46]. In Figure 4.4, we show that each client trains its local model using (secretly shared) encrypted features, ensuring that no client can access the raw data of others during model training. While this method is promising in privacy-preserving VFL, it is unsuitable for non-linear models. The exchange of secretly shared features requires the local models to behave deterministically, a property non-linear models do not have.

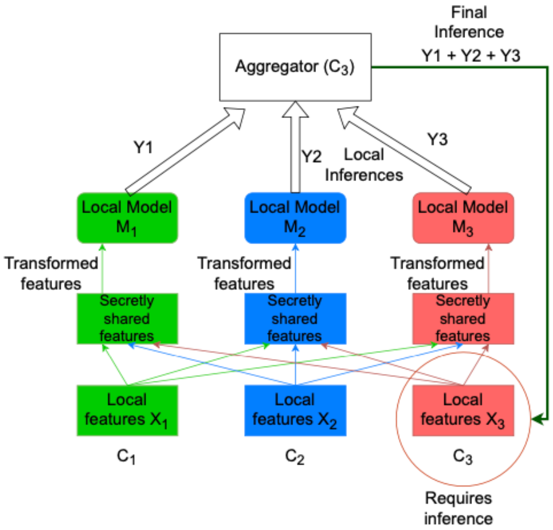


Figure 4.4: Schematic overview of SMPC-based method STV [46]

Split Learning is another approach that has gained traction in VFL [12, 15]. In split learning, the model consists of multiple segments, with each client responsible for training only a portion of the model. Figure 4.5 shows entry points for different features. An independent model processes each feature and shares the intermediary feature with a central entity. The entity processes this intermediate output and calculates the loss to finish the forward propagation. The entity allows for the completion of the training process by sending the gradients concerning the loss of each independent feature model. This method enables clients to collaborate on training complex models (i.e. neural networks) while preserving data privacy.

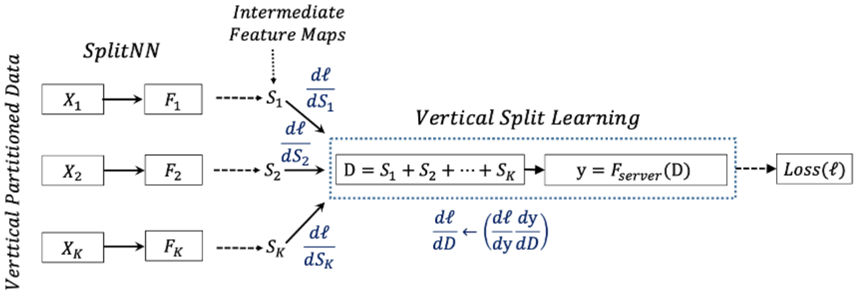


Figure 4.5: Schematic overview of split learning[12]

4.2.2. Solution: Hierarchical time-series based modelling

Through our exploration of VFL approaches, split learning emerges as the most promising solution for enabling vertical knowledge exchange in time-series models while preserving privacy. It offers the flexibility to handle non-linear time-series models like RNNs, GRUs, or LSTMs, which are necessary for capturing the temporal dependencies and complex patterns in time-series data.

To fully adapt split learning to time-series models, we must investigate how to optimize split learning for sequential dependencies inherent in time-series data. The model complexity of a suitable architecture must work for small datasets. We found an architecture in the literature that stacks private and shared sequential models suitable for split learning [63].

In Figure 4.6, we see our interpretation of this architecture using LSTM as sequential models since these are suitable for capturing both long-term and short-term dependencies. This architecture assigns a private LSTM to each client. They produce hidden states that serve as intermediary features. These features are concatenated and used as input for the shared model. Likewise, to the example in Figure 4.5, this architecture also limits the clients from sharing intermediary features and receiving their gradients needed for back-propagation.

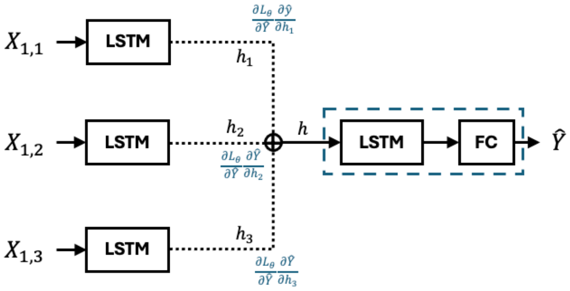

$X_{1,1}$
LSTM
$\frac{\partial L_\theta}{\partial \hat{Y}} \frac{\partial \hat{Y}}{\partial h_1}$
h_1
$X_{1,2}$
LSTM
h_2
$\frac{\partial L_\theta}{\partial \hat{Y}} \frac{\partial \hat{Y}}{\partial h_2}$
h
LSTM
FC
$\hat{Y}$
$X_{1,3}$
LSTM
h_3
$\frac{\partial L_\theta}{\partial \hat{Y}} \frac{\partial \hat{Y}}{\partial h_3}$

Figure 4.6: Distributed model

We will merge the distributed model into our memorization-based solution to solve all challenges for horizontal knowledge exchange with heterogeneous data.

4.3. Combining the solutions for all challenges

We introduce the problem definition in this section, followed by a step-by-step overview of TPHFL.

Problem definition

The proposed architecture targets time-series forecasting problems involving heterogeneous data distributions across tasks. Specifically, we consider N tasks with M parties that predict a univariate time series given endogenous feature $X_{n,1}$ and exogenous features $X_{n,j}, \forall j \in [2, M]$. Each party $i \in [M]$ for task n owns the samples for feature $X_{n,i}$. All input features are uni-variate and have the same time window W but can, for simplicity, be considered one multi-variate input vector $X_n \in \mathbb{R}^{M \times W}$ belonging to task n. For each task n, the model predicts one or multiple future time steps for the endogenous feature. $Y_n \in \mathbb{R}^P$ represents the predictions, the so-called target, where P is the length of the output time window and owned by party 1 for task n. We assume that in each task common samples are identified and aligned by privacy-preserving mechanisms [39, 24, 28].

Notation	Meaning
N	number of tasks
M	number of parties and features in each task
W	size of input time window
P	size of output time window
H	size of hidden dimension
$X_n \in \mathbb{R}^{M \times W}$	input vector for task n
$X_{n,m} \in \mathbb{R}^W$	feature m of input vector for task n
$Y_n \in \mathbb{R}^P$	prediction vector for task n
θ_n	model of task n
$\theta_{n,m}$	m-th component of model of task n
$h_m \in \mathbb{R}^{W \times H}$	hidden vector produced by component $m \in [M]$
$h \in \mathbb{R}^{M \times W \times H}$	concatenation of vectors h_m for $m \in [M]$
$h' \in \mathbb{R}^{W \times H}$	hidden state vector produced by component $M + 1$
$\phi(X)$	intermediary representation for input vector X

Table 4.1: Notations

The objective of this work is to improve time series prediction through personalization while preserving data locality by ensuring that features are kept private within or between tasks. To achieve this, we first aim to develop a global model that generalizes well across all tasks:

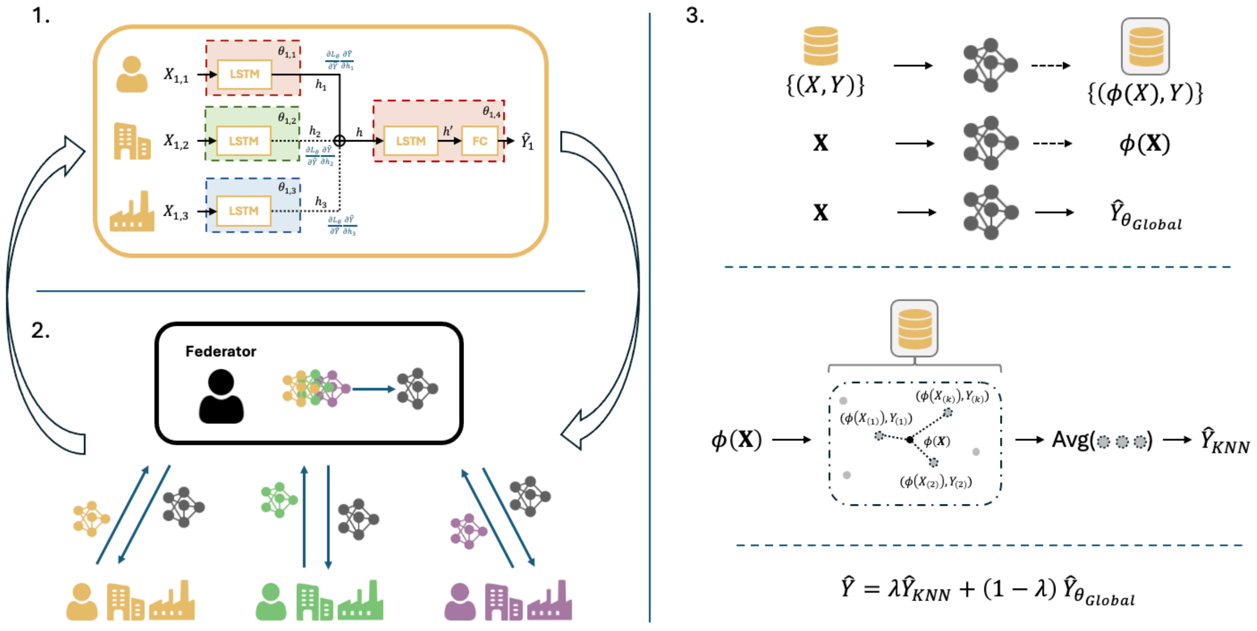


Figure 4.7: TPHFL Framework in three incremental steps: training, optimization and personalization

$$\min_{\theta_{\text{Global}}} \sum_{n=1}^{N} \mathcal{L}_n(\theta_{\text{Global}}) \tag{4.2}$$

After a consensus on the model by all tasks, we use it in a personalization algorithm that allows for predictions better suited for the tasks underlying data distribution.

A trusted third party takes the role of the Federator, responsible for securely collecting the model weights of each party, aggregating them, and redistributing them to the correct parties. Furthermore, it is responsible for initializing the weights of θ_{Global} and sharing them with each party.

TPHFL overview

An overview of TPHFL is given in Figure 4.7. The method consists of three steps: training, optimization and personalization. The first two are described in Algorithm 1, the latter in Algorithm 2. We will discuss each step individually.

Training. In the initial step, the parties with the features and labels for task n collaboratively train a distributed model for multiple epochs. The task model θ_n contains multiple components:

$$\theta_n = \{\theta_{n,1}, \theta_{n,2}, ..., \theta_{n,M+1}\} \tag{4.3}$$

$\theta_{n,1}$ to $\theta_{n,M}$ comprise single LSTM units at the beginning of the model, whereas $\theta_{n,M+1}$ contains an LSTM and Fully Connected (FC) layer. We chose an LSTM because of its ability to capture long-term dependencies at a moderate level of model complexity. Each party $m \in [M]$ for task n has private ownership over $\theta_{n,M}$ and party 1 has additional ownership over $\theta_{n,M+1}$ meaning that only the designated party can read and write the given model weights.

To collaboratively train θ_n, the Federator initiates the training process for all parties across tasks in lines 6 and 8 of Algorithm 1. The Federator calls party 1 separately to handle the flow of data through the final model component. Each party processes their data through their private LSTM with an input window of size 1 in line 18. This way, each piece of input data $X_{n,m} \in \mathbb{R}^{W}$ is transformed to hidden states $h_m \in \mathbb{R}^{W \times H}$. Each party shares these states with party 1 in line 19, who concatenates them in line 22, producing $h \in \mathbb{R}^{M \times W \times H}$. This state serves as input for the LSTM in $\theta_{n,M+1}$ with input window $M \times H$ that transforms h to $h' \in \mathbb{R}^{W \times H}$. The new hidden state is fed through the FC layer to produce prediction $\hat{Y}_n$. During back-propagation in line 23, party 1 calculates the loss and gradient for $\theta_{n,M+1}$ necessary for updating the model parameters:

$$\nabla\theta_{n,M+1} = \frac{\partial\mathcal{L}_{\theta_n}}{\partial\theta_{n,M+1}} = \frac{\partial\mathcal{L}_{\theta_n}}{\partial\hat{Y}_n}\frac{\partial\hat{Y}_n}{\partial\theta_{n,M+1}} \tag{4.4}$$

Parties 1 to M calculate the gradients for $\theta_{n,1}$ to $\theta_{n,M}$ individually with:

$$\nabla\theta_{n,m} = \frac{\partial\mathcal{L}_{\theta_n}}{\partial\theta_{n,m}} = \frac{\partial\mathcal{L}_{\theta_n}}{\partial\hat{Y}_n}\frac{\partial\hat{Y}_n}{\partial h_m}\frac{\partial h_m}{\partial\theta_{n,m}} \tag{4.5}$$

Party 1 calculates the gradient for $\theta_{n,M+1}$ and $\theta_{n,1}$ and the derivatives $\frac{\partial\mathcal{L}_{\theta_n}}{\partial\hat{Y}_n}$ and $\frac{\partial\hat{Y}_n}{\partial h_m}$ for $m \in [2, M]$. It sends the derivatives to the correct parties in line 24 so they can complete their gradient calculations in line 27.

Optimization. Each party shares its model component with the Federator, which is responsible for aggregating these components using the FedAvg algorithm (Equation 3.20). The Federator identifies each model component it receives and aggregates all $\theta_{n,i}$ separately. It saves the new model weights in lines 6 and 8 based on task and party index. This approach simulates the aggregation of the full task models θ_n by aggregating components independently in line 12, creating a global model that is the same for all tasks while only exchanging components. Through this process, the Federator facilitates collaboration and information exchange between tasks.

We alternate training and optimization across multiple training epochs. After each local epoch, each task shares its model components with the Federator and receives updated components. We continue this process until the task model performances have converged or after a fixed amount of training epochs, reaching a consensus on the global model.

Personalization. After achieving consensus on the final global model, having completed training and optimization, each task uses its task model θ_n for memorization-based personalization. In this approach, we select the most similar training samples during inference using KNN and use the accompanied labels Y for memorization-based predictions. For similarity measurements, we transform the observation X in the training samples to intermediary representations $\phi(X)$ as these contain significant information about the model's input interpretation. This representation can be any state inside the distributed model. In our case, we use h as an intermediary representation.

Typically, we transform our data beforehand as this can be computationally expensive. In lines 4 and 6, we call the transformation algorithm for each party of task n, of which party 1 saves this transformed data in a new dataset. All parties collaboratively transform the data, as each party will send their hidden states to party 1 in line 15. Party 1 saves ϕ together with the labels in line 19.

During inference in line 9, we select the k most similar samples from our transformed dataset:

$$N_k(\mathbf{X}) = \{(\phi(X_{(1)}), Y_{(1)}), \phi(X_{(2)}), Y_{(2)}), \ldots, (\phi(X_{(k)}), Y_{(k)})\} \tag{4.6}$$

where the ordering is determined by intermediary distances:

$$d(\phi(X_{(1)}), \mathbf{X}) \leq d(\phi(X_{(2)}), \mathbf{X}) \leq \ldots \leq d(\phi(X_{(k)}), \mathbf{X}) \tag{4.7}$$

$(\phi(X_{(1)}), Y_{(1)})$ is the i-th nearest neighbour for task n and sample X. The distance metric d, typically chosen as Euclidean distance or another similarity measure, plays a key role in determining the influence of each neighbour label on the final prediction.

These neighbours are most similar to our input and can be used to make memorization-based predictions:

$$d_{(i)}(\mathbf{X}) = d(\phi(X_{(i)}), \mathbf{X}) \tag{4.8}$$

$$\hat{Y}_{KNN} = \frac{\sum_{i=1}^{k} K(d_{(i)}(\mathbf{X}))Y_{(i)}}{\sum_{i=1}^{k} K(d_{(i)}(\mathbf{X}))} \tag{4.9}$$

where K is a kernel. We combine these predictions with global model predictions $\hat{Y}_{\theta_{Global}}$, the output of the global model.

$$\hat{Y} = \lambda \hat{Y}_{KNN} + (1 - \lambda)\hat{Y}_{\theta_{Global}} \tag{4.10}$$

where $\lambda \in [0, 1]$ is a weight parameter balancing the contributions of both losses for task n.

Algorithm 1: Training and optimization

```
**Data:** $X_{n,m}$ on party $(n,m)$ fed in batches $B_{n,m}$, and $Y_n$ on party $(n,1)$
**Param:** Global model parameters $\theta_{Global}$, tasks $N$, distributed features $M$, rounds $R$, epochs $E$
**Result:** Trained distributed models $\theta_{Global}$
**Federator executes:**
    Initialize $\theta_{Global}$;
    **for** $r = 1, \dots, R$ **do**
        **for** $(n,m) \in [N] \times [M]$ **do**
            **if** $m == 1$ **then**
                $[\theta_{n,m}, \theta_{n,M+1}] \leftarrow$ Train$(n, m, \theta_{Global,m}, \theta_{Global,M+1})$;
            **else**
                $\theta_{n,m} \leftarrow$ Train$(n, m, \theta_{Global,m}, None)[0]$;
            **end**
        **end**
        **for** $m \in [M + 1]$ **do**
            $\theta_{Global,m} = \frac{1}{N} \sum_N \theta_{n,m}$
        **end**
    **end**
**Train($n, m, \theta_m, \theta_{M+1}$)**
    **for** $e = 1, \dots, E$ **do**
        **for** $b \in B_{n,m}$ **do**
            $h_m \leftarrow \theta_m(b)$ ;
            send($h_m$ to party $(n, 1)$);
            **if** $m == 1$ **then**
                await($h_m$ for $m \in [M]$);
                $\hat{Y} \leftarrow \theta_{M+1}(\bigoplus_M h_m)$;
                $\theta_{M+1} \leftarrow \theta_{M+1} - \nabla \mathcal{L}_{\theta_{M+1}}$;
                send($\frac{\partial \theta}{\partial \hat{Y}} \frac{\partial \hat{Y}}{\partial h_m}$ to $(n, m), \forall m \in [M]$)
            **end**
            await($\frac{\partial \mathcal{L}_\theta}{\partial \hat{Y}} \frac{\partial \hat{Y}}{\partial h_m}$ from party 1);
            $\theta_m \leftarrow \theta_m - \nabla \theta_m$;
        **end**
    **end**
    return $[\theta_m, \theta_{M+1}]$
```

Algorithm 2: Personalization

```
Data: Dataset $S_n$ on task $n$
Param: Distributed features $M$
Result: Predictions $\hat{Y}$
Each task $n$ executes:
    $D_n \leftarrow \emptyset$ for $m \in [M]$ do
        if $m == 1$ then
            $D_n \leftarrow$ TransformData$(n, m, S_n)$;
        else
            TransformData$(n, m, S_n)$;
        end
    end
    At inference on X return $\hat{Y}$ with transformed data $D_n$ and Equation 4.10 ;
TransformData($n, m, S$)
    $D \leftarrow \emptyset$ ;
    for $(X, Y) \in S$ do
        for $m \in [M]$ do
            $h_m = \theta_m(X_m)$;
            send($h_m$ to party $(n, m)$);
            if $n == 1$ then
                await($h_m$ for $m \in [M]$);
                $\phi(X) \leftarrow \oplus_M h_m$;
                $D \leftarrow D \cup (\phi(X), Y)$;
            end
        end
    end
    return $D$
```

5 Experiments

The content of this chapter overlaps with the content of the research paper. We evaluate the forecasting of TPHFL against scenarios with different forms of data locality and collaborative capabilities. Additionally, we conduct experiments with a different hidden representation ϕ and perform hyper-parameter analysis. We first discuss the experimental settings.

5.1. Experimental settings

We use four public datasets for the experiments: Air quality [13], Solar power [53], Crypto [4] and Rossman Sales [33]. Additionally, we used an industry-specific dataset to predict a specific parameter from sensor values in semiconductor manufacturing. Further details on the public datasets are given in Appendix D. We briefly go over the baselines used for evaluation and more specific settings.

	Collaboration	
Data locality	Vertical	Hybrid
None	-	Centralized
Horizontal	Independent	FedAvg, TPHFL-H
Vertical	-	Centralized+
Hybrid	Independent+	TPHFL-NP, TPHFL

Table 5.1: Baseline methods with different forms of data locality and collaboration.

5.1.1. Baseline

We compare TPHFL to scenarios that differ in data locality and collaborative capabilities shown in Table 5.1. We show a schematic overview for each baseline method in Appendix E. All methods enable vertical collaboration by default because we are training on multivariate time series data because the goal is not to demonstrate that using more features improves predictive performance. Instead, we focus on how different collaboration configurations impact the model's performance under privacy constraints. We will discuss the methods in order of privacy restrictions:

None: No privacy restrictions, allowing all data to be combined freely.

- **Centralized**: In this case, horizontal collaboration is introduced by centralizing and concatenating all training data and using it to train a single LSTM.

Horizontal: Parties share data within tasks, not between tasks.

- **Independent**: We allow vertical collaboration by letting each task train a separate LSTM on its multivariate time series data, with no exchange of information between tasks.

- **FedAvg**: We allow hybrid collaboration by letting each task train a separate LSTM and sharing the model weights with a Federator, responsible for aggregating the models from each task.

- **TPHFL-Horizontal**: This method builds upon FedAvg by adding the memorization-based personalization algorithm, similar to TPHFL. Different from TPHFL, TPHFL-H uses a single LSTM per task.

Vertical: Parties share data between tasks, not within tasks.

- **Centralized+**: This method is similar to its counterpart without privacy restrictions (Centralized) but employs the distributed model instead of a single LSTM to maintain vertical data locality.

Hybrid: Both dimensions of data locality are in place.

- **Independent+**: This method is similar to its counterpart with horizontal privacy restrictions (Independent) but employs a distributed model instead of a single LSTM to maintain vertical data locality.
- **TPHFL-NoPersonalization**: This variant of TPHFL omits the personalization algorithm.
- **TPHFL**: Our proposed method enables hybrid collaboration while maintaining horizontal and vertical data locality.

Independent and Centralized serve as expected upper-bound and lower-bound, respectively. We want TPHFL to show a decrease in Mean Absolute Error (MAE), meaning an increase in performance, compared with Independent, showing that there is an incentive for participants to share knowledge with other tasks. Centralized is the lower bound because it is an ideal scenario without data privacy constraints, allowing for less computational complexity and better performance.

5.1.2. Metrics and Setup

We compare the performance of TPHFL and the different scenarios using the Mean Absolute Error (MAE). For all datasets, the missing values were interpolated and replaced with 0 if there were no neighbouring values. We normalized all data for consistent comparison.

The LSTMs used in the experiments have two layers, a hidden size of 20 and a dropout of 0.2. We train the models in 30 epochs, with a batch size of 32, a learning rate of 0.001 and a weight decay of 0.001.

We conduct the training process by splitting the data into training data (80%) and test data (20%). We use a fixed input window of 32 and a variable prediction window of 1, 2, 4, 8 and 16. We construct the samples using a sliding window of 1 timestep.

To compare our method with other scenarios, we calculate the MAE for the different prediction windows and average all outcomes. For our solution, we run different values for k 1, 3, 5, 7 and 10 and choose the best-performing. In every experiment, we choose the most optimal λ per task for which we can get the lowest MAE. In the following paragraphs, we will discuss the results for 6 tasks (except if stated differently) and the best-performing k. In Appendix F, we included more extensive results for 2, 4 and 6 tasks with different prediction window sizes.

5.2. Forecasting results

Table 5.2 shows that TPHFL performs better than Independent for four out of five datasets with an increase in performance as high as 27.2%. A performance increase was not feasible for the Sales dataset due to insufficient training samples and data quality. The Crypto dataset has a limited performance increase compared to its peers because two exogenous features are too similar to the endogenous feature and contain limited valuable information to improve the predictions (see Figure D.1).

The relative improvement reflects the average performance enhancement across all prediction windows. However, our experiments have shown that this improvement is not uniform; some prediction windows exhibit significantly higher gains than others. One explanation for this is the seasonality in the temporal data, where repeating patterns over specific intervals can have varying impacts for different windows. Windows that coincide with seasonal trends may benefit from the model's ability to capture these patterns.

In Table 5.3, we compare three methods without vertical privacy constraints to their counterparts that enforce vertical restrictions. Most experiments show declines in performance that can be attributed to using a distributed model in place of a single LSTM. We expect this outcome because increased model

Dataset	Independent	Centralised	TPHFL	Rel. Imp.
AirQuality	3.60 +/- 0.07	2.77 +/- 0.01	2.96 +/- 0.01	17.8%
Industry	4.28 +/- 0.63	2.48 +/- 0.14	3.23 +/- 0.38	24.4%
Sales	2.96 +/- 0.01	3.02 +/- 0.02	4.40 +/- 0.06	-48.7%
Crypto	2.64 +/- 0.14	1.69 +/- 0.04	2.40 +/- 0.04	8.9%
Solar	2.17 +/- 0.03	1.42 +/- 0.01	1.58 +/- 0.04	27.2%

Table 5.2: Average MAE and standard deviation for methods with no privacy constraints and TPHFL. We show the relative improvement of TPHFL over Independent (rel. imp.) in percentages.

complexity typically leads to a trade-off in predictive performance. Centralized shows a smaller reduction or improvements in performance because it uses a larger dataset, which is crucial when training more complex models to mitigate the negative impact on accuracy. TPHFL can improve its performance by increasing the number of samples, something we learned from the Centralized experiments. However, this may not always be possible due to the limited availability of temporal data. Sensory data may only be available for a certain period, and receiving more samples requires the machine to operate for a longer period.

	Independent+		Centralised+		TPHFL-H	
Dataset	MAE	Imp.	MAE	Imp.	MAE	Imp.
AirQuality	3.72 +/- 0.03	-3.3%	2.80 +/- 0.01	-0.9%	2.58 +/- 0.00	-14.8%
Industry	5.76 +/- 0.91	-34.7%	2.62 +/- 0.23	-6.0%	2.85 +/- 0.34	-13.5%
Sales	5.17 +/- 0.07	-74.8%	3.21 +/- 0.09	-6.3%	2.83 +/- 0.02	-55.3%
Crypto	3.91 +/- 0.41	-48.3%	1.61 +/- 0.03	4.5%	1.67 +/- 0.02	-43.7%
Solar	2.67 +/- 0.07	-23.1%	1.40 +/- 0.01	1.4%	1.38 +/- 0.02	-14.3%

Table 5.3: Average MAE and standard deviation for different methods with vertical restrictions (Independent+, Centralized+), and Horizontal restrictions (TPHFL-H). We compare the performance increase for methods if we introduce vertical privacy restrictions: Independent to Independent+, Centralized to Centralized+ and TPHFL-H to TPHFL.

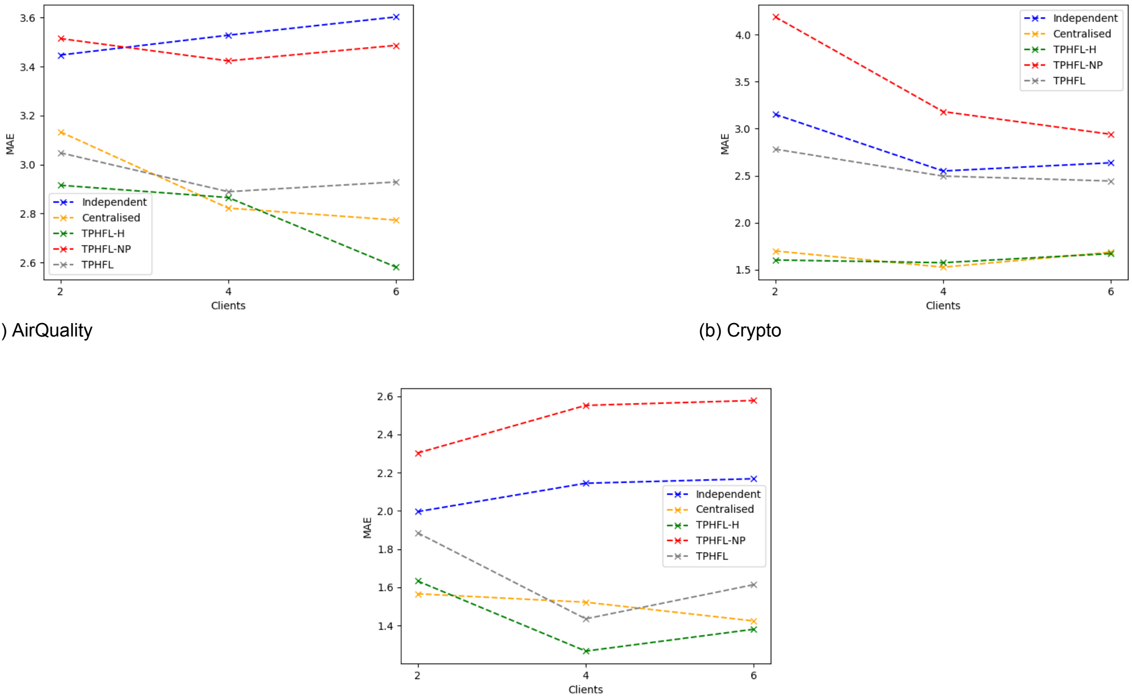
In Table 5.4, we compare two methods that incorporate horizontal collaboration and personalization with their counterpart that do not use personalization. The results demonstrate the effectiveness of the personalization algorithm, improving the accuracy in the horizontal and hybrid data privacy domain. This improvement is especially pronounced for TPHFL because TPHFL-NP struggles to fit the data due to the complexity of the distributed model and the limited amount of data. Personalization helps mitigate these challenges, leading to a larger performance gap than FedAvg and TPHFL-V, methods that are better suited to fit the model effectively without personalization.

Dataset	FedAvg	Imp.	TPHFL-NP	Imp.
AirQuality	2.91 +/- 0.02	11.3%	3.49 +/- 0.03	15.0%
Industry	3.44 +/- 0.42	17.2%	4.29 +/- 0.55	24.6%
Sales	3.43 +/- 0.02	17.3%	7.68 +/- 0.09	42.7%
Crypto	1.94 +/- 0.04	14.0%	2.94 +/- 0.05	18.3%
Solar	2.05 +/- 0.03	32.7%	2.58 +/- 0.06	38.8%

Table 5.4: Average MAE and standard deviation for different methods with horizontal and hybrid privacy constraints. We show the improvement of introducing the personalization mechanism: FedAvg to TPHFL-H and TPHFL-NP to TPHFL.

Lastly, we show in Figure 5.1 the MAE for at least one method from each previous comparison. We show the results for three datasets and a different number of tasks. When comparing TPHFL with methods without privacy restrictions, we observe that Independent consistently serves as an upper bound, while Centralized almost always acts as a lower bound. The anomalies are caused by Centralized combining all data, sometimes at the expense of task-specific performance due to overfitting or loss of task nuances.

TPHFL-H consistently shows an improvement over TPHFL, as expected, since using a distributed model in TPHFL introduces complexity that often leads to a loss in accuracy. Finally, TPHFL-NP consistently underperforms compared to TPHFL, which aligns with our expectations, as the absence of personalization limits the model's ability to fine-tune itself to task-specific data distributions.



(a) AirQuality

(b) Crypto

(c) Solar

Figure 5.1: Average error for different prediction windows from three datasets.

Dataset	2 tasks		4 tasks		6 tasks	
	TPHFL-l2	TPHFL	TPHFL-l2	TPHFL	TPHFL-l2	TPHFL
AirQuality	**11.58%**	9.72%	**18.11%**	17.40%	**18.70%**	17.76%
Sales	-23.79%	**-19.84%**	-38.59%	**-18.44%**	-79.90%	-48.72%
Crypto	11.74%	**14.16%**	2.13%	**4.17%**	7.36%	**8.93%**
Solar	5.56%	**6.90%**	33.07%	**34.71%**	25.55%	**27.19%**
Industry	39.23%	**39.54%**	37.10%	**37.84%**	23.23%	**24.45%**

Table 5.5: Relative improvement over Independent for TPHFL with different forms of intermediary representations. TPHFL uses h as TPHFL-l2 uses h' as intermediary representation.

5.3. Different hidden representation

In the personalization step of TPHFL, the intermediary representation $\phi(X)$ can be any output within the model. In Table 5.5, we compare TPHFL with a variant that uses h' as an intermediary state, referred to as TPHFL-I2. The results indicate that this variant performs better for the AirQuality dataset, suggesting that, for this dataset, the hidden outputs from the upper model contain more valuable information for the samples than those generated by the private LSTM. A possible explanation is that the endogenous features for each task in the AirQuality dataset exhibit a higher correlation than other datasets, leading to task models leaning more toward global generalization rather than local specialization. The hidden states produced by the upper model LSTM are better suited for capturing these global patterns, making them more suitable as intermediary representations for this dataset.

However, the key takeaway is that choosing intermediary representations can be highly dataset-dependent as many unique combinations of outputs could be used for this purpose. One must carefully evaluate the different possibilities to find a representation that maximizes the predictive performance of TPHFL.

5.4. Hyper-parameter analysis

As mentioned earlier, we can tune two hyperparameters: the value of k and the hyperparameter λ. Our experiments revealed that varying k has little impact on the performance of TPHFL, as demonstrated in Figure 5.2. In this figure, we plotted the MAE for all datasets (excluding Sales due to its consistently low performance) across different values of k. The results show that the error remains nearly constant, indicating that the choice of k does not significantly affect the method's performance.

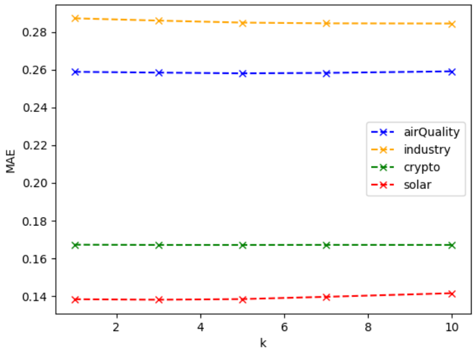


Figure 5.2: Average MAE for different values of k in TPHFL.

Different values of λ significantly affect the performance of each task model. In Figure 5.3, we use the Solar dataset as an example to illustrate this. We plot the MAE of three different strategies for selecting λ: setting a single global value for all tasks, choosing an optimal λ for each task individually, and selecting the optimal λ on a sample-by-sample basis. For reference, we included the centralized and independent as upper-bound and lower-bound, respectively. MM is plotted against different values for λ, while the other methods remain static because they either are not dependent on λ or always select an optimal value, leaving no room for tuning λ.

The optimal global λ is set at 0.4 for the Solar dataset, meaning that we interpolate 40% KNN predictions and 60% global predictions. We can reduce the MAE further by setting the parameter on a task basis, demonstrating that the optimal task-based λ varies from task to task, allowing tasks to balance private and global information independently, improving overall performance. The balance between private and global information varies across tasks with heterogeneous data distributions. Consequently, the optimal λ depends on the specific characteristics of each task. Tasks with more private information tend to select a higher λ, while tasks with less private information lean toward a lower value.

The MAE of sample-based TPHFL falls significantly below the expected lower bound of Centralized because, on a sample-by-sample basis, the optimal λ often turns out to be an extreme value—either 0.0 or 1.0. In other words, it is best to rely entirely on KNN or global model predictions for most inferences, or in terms of data distribution, the model either fully prioritizes private information or global knowledge

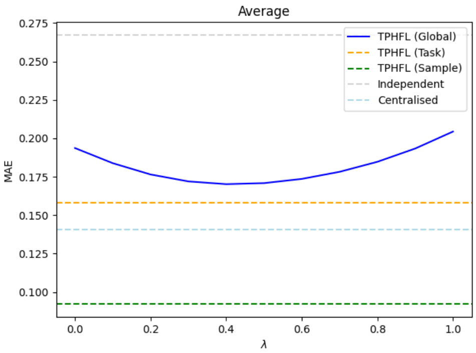


Figure 5.3: Average MAE for different values of λ in TPHFL and Solar dataset.

exchange, depending on the specific sample. When choosing an optimal λ at the task level, the model balances out these extreme cases, finding a middle ground. Further investigation is required to uncover a direct relationship between the characteristics of individual samples and their corresponding optimal λ values, providing more insight into the prioritization of private or global information.

6 Conclusions

In this thesis, we proposed a novel FL framework TPHFL, designed to tackle the challenges of time-series forecasting in distributed, privacy-sensitive industrial environments. By integrating both HFL and VFL approaches, our model facilitates multi-level knowledge sharing while preserving data locality by not sharing private data between different parties, laying a critical groundwork for future, more robust privacy-preserving solutions. Our contributions are as follows:

1. **Hybrid FL strategy**: TPHFL integrates both horizontal and vertical dimensions in FL, facilitating knowledge exchange within tasks (intra-task) and between tasks (inter-task). We use a hierarchical solution strategy that approaches this problem at two levels. First, each task is assigned a task model horizontally aggregated by a Federator [42]. Second, each task model operates as a distributed model with distinct entry points for each feature, enabling vertically distributed features in each task. Our strategy provides privacy to a certain extent by preserving the locality of data, laying a critical groundwork for future privacy-preserving solutions.
2. **Time-series-based memorization**: In TPHFL, clients train a global model that generalizes to all tasks. For each task, the model is adapted to the task-specific environment by a personalization mechanism which utilizes the task-specific training data to refine the predictions and better align with the underlying data distribution [41].
3. **Time-series-based hierarchical model**: A deep learning model containing sequential model components has private entry points for each party that produces intermediary representations. These are concatenated and fed through an upper layer. This architecture contains sequential modules to facilitate temporal data and is split into multiple components, making it compatible with the Hybrid FL strategy.

Experiments on several real-world datasets demonstrate the effectiveness of our method, showing a significant improvement in predictive performance over traditional independent models and further enhancing results from horizontal collaboration through a personalization algorithm.

For future research, several promising avenues remain unexplored or could benefit from deeper investigation to further enhance the robustness, applicability, and precision of the proposed methodology. These areas include:

1. **Formal Confidentiality Guarantees**: While our method effectively limits data sharing, it lacks formal privacy guarantees—a critical requirement for secure deployment, especially in sensitive domains like healthcare, finance, or industrial IoT. Future work should prioritize the integration of formal confidentiality frameworks that provide quantifiable privacy assurances without compromising model accuracy. Approaches such as differential privacy [19], which adds controlled noise to protect individual data points, or homomorphic encryption [1], which enables computation on encrypted data, can help achieve robust privacy. These methods have been proven useful in federated settings [20, 66]. Research into how these techniques impact time-series forecasting

performance and computational load would be beneficial, as balancing security with scalability is key for practical deployment.

2. **Multivariate Forecasting**: Currently, our method is limited to predicting a single endogenous variable, which restricts its use in multivariate time-series forecasting applications in PMC. Predicting multiple interrelated variables could yield richer insights and more accurate forecasts in practical scenarios by capturing the interactions between variables. Methods such as FATHOM have shown that multivariate predictions are a viable option [14]. Future studies could investigate extending the framework to support multivariate predictions, examining the challenges this presents, such as increased model complexity and computational requirements.

3. **Soft Predictions for Distributed Models**: The current implementation restricts prediction capabilities to a single designated party (party 1), limiting the flexibility of predictions within a distributed framework. Future work could explore architectures that enable soft predictions, allowing each participating party to generate localized predictions. Soft predictions would enable parties without direct access to labels to make approximated predictions, typically aligned with predictions from party 1. One approach could involve the development of a shared upper layer accessible to all participants, enabling distributed predictions while safeguarding label confidentiality. Techniques such as label sharing in MMVFL or SMPC could facilitate secure, parallel training across parties, potentially allowing each party to independently refine and validate predictions [51, 46].

4. **Optimizing Sample-based λ for Improved Prediction**: In our final results, the hyperparameter λ demonstrated a significant performance impact when tuned appropriately for the dataset. However, λ was set globally rather than dynamically, which may limit the method's adaptability to diverse or shifting data. Future research could investigate strategies for dynamically optimizing λ on a sample-specific basis, potentially developing algorithms to adapt λ based on real-time data characteristics. Exploring methods for automated tuning or even learning-based approaches, where λ adapts based on historical forecasting performance, could lead to considerable gains in accuracy and robustness. Another approach could involve analyzing the sensitivity of λ to different data distributions, enabling a better understanding of its role and refining it into a more flexible parameter within the model.

In summary, these recommendations for future research highlight areas where further advancements can enhance the scalability, privacy, and predictive accuracy of federated learning for time-series forecasting. Integrating privacy guarantees, extending model capabilities to multivariate time series, enabling distributed predictions, and dynamically optimizing hyperparameters will be essential steps towards refining the applicability of this approach in complex real-world environments.

Bibliography

[1] Abbas Acar et al. “A survey on homomorphic encryption schemes: Theory and implementation”. In: *ACM Computing Surveys (Csur)* 51.4 (2018), pp. 1–35.

[2] Jisu Ahn et al. “Federated learning for predictive maintenance and anomaly detection using time series data distribution shifts in manufacturing processes”. In: *Sensors* 23.17 (2023), p. 7331.

[3] AIML. *Compare the different Sequence models (RNN, LSTM, GRU, and Transformers)*. 2024. URL: `https://aiml.com/compare-the-different-sequence-models-rnn-lstm-gru-and-transformers/`.

[4] A. Ticchi et al. *G-Research Crypto Forecasting*. `https://kaggle.com/competitions/g-research-crypto-forecasting`. Kaggle, Datatset. 2021.

[5] Shun-ichi Amari. “Backpropagation and stochastic gradient descent method”. In: *Neurocomputing* 5.4-5 (1993), pp. 185–196.

[6] Daniel Lee Andersen, Christine Sarah Anne Ashbrook, and Neil Bang Karlborg. “Significance of big data analytics and the internet of things (IoT) aspects in industrial development, governance and sustainability”. In: *International Journal of Intelligent Networks* 1 (2020), pp. 107–111.

[7] Sio-Iong Ao and Haytham Fayek. “Continual deep learning for time series modeling”. In: *Sensors* 23.16 (2023), p. 7167.

[8] Manoj Ghuhan Arivazhagan et al. “Federated learning with personalization layers”. In: *arXiv preprint arXiv:1912.00818* (2019).

[9] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*. Vol. 1. MIT press Cambridge, MA, USA, 2017.

[10] George EP Box et al. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[11] Rich Caruana. “Multitask learning”. In: *Machine learning* 28 (1997), pp. 41–75.

[12] Iker Ceballos et al. “Splitnn-driven vertical partitioning”. In: *arXiv preprint arXiv:2008.04137* (2020).

[13] S. Chen. *Beijing Multi-Site Air Quality*. `https://archive.ics.uci.edu/dataset/501/beijing+multi+site+air+quality+data`. UCI Machine Learning Repository, Dataset. 2019.

[14] Yujing Chen et al. “Federated multi-task learning with hierarchical attention for sensor data analytics”. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020, pp. 1–8.

[15] Zekai Chen et al. “Multi-task time series forecasting with shared attention”. In: *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2020, pp. 917–925.

[16] Kyunghyun Cho et al. “Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv 2014”. In: *arXiv preprint arXiv:1406.1078* (2020).

[17] Liam Collins et al. “Exploiting shared representations for personalized federated learning”. In: *International conference on machine learning*. PMLR. 2021, pp. 2089–2099.

[18] Jinliang Deng et al. “A multi-view multi-task learning framework for multi-variate time series forecasting”. In: *IEEE Transactions on Knowledge and Data Engineering* 35.8 (2022), pp. 7665–7680.

[19] Cynthia Dwork. “Differential privacy”. In: *International colloquium on automata, languages, and programming*. Springer. 2006, pp. 1–12.

[20] Ahmed El Ouadrhiri and Ahmed Abdelhadi. “Differential privacy for deep and federated learning: A survey”. In: *IEEE access* 10 (2022), pp. 22359–22380.

[21] GH Fisher. “Endogenous and exogenous investment in macro-economic models”. In: *The Review of Economics and Statistics* (1953), pp. 211–220.

[22] Dashan Gao, Xin Yao, and Qiang Yang. “A survey on heterogeneous federated learning”. In: *arXiv preprint arXiv:2210.04505* (2022).

[23] General Data Protection Regulation GDPR. “General data protection regulation”. In: *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC* (2016).

[24] Oded Goldreich. “Secure multi-party computation”. In: *Manuscript. Preliminary version* 78.110 (1998), pp. 1–108.

[25] Vibhor Gupta et al. “Continual learning for multivariate time series tasks with variable input dimensions”. In: *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2021, pp. 161–170.

[26] James D Hamilton. *Time series analysis*. Princeton university press, 2020.

[27] Edward J Hannan and Laimonis Kavalieris. “A method for autoregressive-moving average estimation”. In: *Biometrika* 71.2 (1984), pp. 273–280.

[28] Stephen Hardy et al. “Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption”. In: *arXiv preprint arXiv:1711.10677* (2017).

[29] Sepp Hochreiter. “The vanishing gradient problem during learning recurrent neural nets and problem solutions”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02 (1998), pp. 107–116.

[30] Richard H Jones. “Maximum likelihood fitting of ARMA models to time series with missing observations”. In: *Technometrics* 22.3 (1980), pp. 389–395.

[31] Kahiomba Sonia Kiangala and Zenghui Wang. “An effective predictive maintenance framework for conveyor motors using dual time-series imaging and convolutional neural network in an industry 4.0 environment”. In: *Ieee Access* 8 (2020), pp. 121033–121049.

[32] Yeongwoo Kim et al. “Dynamic clustering in federated learning”. In: *ICC 2021-IEEE International Conference on Communications*. IEEE. 2021, pp. 1–6.

[33] F. Knauer and W. Cukierski. *Rossmann Store Sales*. `https://kaggle.com/competitions/rossmann-store-sales`. Kaggle, Dataset. 2015.

[34] Joos Korstanje. “The SARIMAX Model”. In: *Advanced Forecasting with Python: With State-of-the-Art-Models Including LSTMs, Facebook’s Prophet, and Amazon’s DeepAR*. Berkeley, CA: Apress, 2021, pp. 125–131. ISBN: 978-1-4842-7150-6. DOI: `10.1007/978-1-4842-7150-6_8`. URL: `https://doi.org/10.1007/978-1-4842-7150-68`.

[35] Tian Li et al. “Federated optimization in heterogeneous networks”. In: *Proceedings of Machine learning and systems* 2 (2020), pp. 429–450.

[36] Chin-Yi Lin et al. “Time series prediction algorithm for intelligent predictive maintenance”. In: *IEEE Robotics and Automation Letters* 4.3 (2019), pp. 2807–2814.

[37] Yang Liu et al. “Vertical federated learning: Concepts, advances, and challenges”. In: *IEEE Transactions on Knowledge and Data Engineering* (2024).

[38] Yi Liu et al. “Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach”. In: *IEEE Internet of Things Journal* 8.8 (2020), pp. 6348–6358.

[39] Linpeng Lu and Ning Ding. “Multi-party private set intersection in vertical federated learning”. In: *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE. 2020, pp. 707–714.

[40] Tao Ma and Ying Tan. “Multiple stock time series jointly forecasting with multi-task learning”. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020, pp. 1–8.

[41] Othmane Marfoq et al. “Personalized federated learning through local memorization”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 15070–15092.

[42] Brendan McMahan et al. “Communication-efficient learning of deep networks from decentralized data”. In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[43] Larry R Medsker, Lakhmi Jain, et al. “Recurrent neural networks”. In: *Design and Applications* 5.64-67 (2001), p. 2.

[44] Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci. *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

[45] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. “A review on the attention mechanism of deep learning”. In: *Neurocomputing* 452 (2021), pp. 48–62.

[46] Aditya Shankar et al. “Share Your Secrets for Privacy! Confidential Forecasting with Vertical Federated Learning”. In: *arXiv preprint arXiv:2405.20761* (2024).

[47] Alex Sherstinsky. “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network”. In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306.

[48] Farhad Mortezapour Shiri et al. “A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU”. In: *arXiv preprint arXiv:2305.17473* (2023).

[49] Farhad Mortezapour Shiri et al. “A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU”. In: *arXiv preprint arXiv:2305.17473* (2023).

[50] Robert H Shumway et al. “ARIMA models”. In: *Time series analysis and its applications: with R examples* (2017), pp. 75–163.

[51] Feng Siwei and Yu Han. *Multi-participant multi-class vertical federated learning*. 2020.

[52] Virginia Smith et al. “Federated multi-task learning”. In: *Advances in neural information processing systems* 30 (2017).

[53] *Solar power generation*. `https://www.nrel.gov/grid/solar-power-data.html`. NREL, Dataset. 2006.

[54] Fallaw Sowell. “Maximum likelihood estimation of stationary univariate fractionally integrated time series models”. In: *Journal of econometrics* 53.1-3 (1992), pp. 165–188.

[55] Shengjing Sun et al. “Data handling in industry 4.0: Interoperability based on distributed ledger technology”. In: *Sensors* 20.11 (2020), p. 3046.

[56] Alysa Ziying Tan et al. “Towards personalized federated learning”. In: *IEEE transactions on neural networks and learning systems* 34.12 (2022), pp. 9587–9603.

[57] Alysa Ziying Tan et al. “Towards personalized federated learning”. In: *IEEE transactions on neural networks and learning systems* 34.12 (2022), pp. 9587–9603.

[58] Artur Trindade. *ElectricityLoadDiagrams20112014*. UCI Machine Learning Repository. 2015.

[59] A Vaswani. “Attention is all you need”. In: *Advances in Neural Information Processing Systems* (2017).

[60] Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. “Three types of incremental learning”. In: *Nature Machine Intelligence* 4.12 (2022), pp. 1185–1197.

[61] Kang Wei et al. “Vertical federated learning: Challenges, methodologies and experiments”. In: *arXiv preprint arXiv:2202.04309* (2022).

[62] Huanlai Xing et al. “An efficient federated distillation learning system for multitask time series classification”. In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–12.

[63] Yang Yan et al. “Multi-participant vertical federated learning based time series prediction”. In: *Proceedings of the 8th International Conference on Computing and Artificial Intelligence*. 2022, pp. 165–171.

[64] Yong Yu et al. “A review of recurrent neural networks: LSTM cells and network architectures”. In: *Neural computation* 31.7 (2019), pp. 1235–1270.

[65] Chen Zhang et al. “A survey on federated learning”. In: *Knowledge-Based Systems* 216 (2021), p. 106775.

[66] Chengliang Zhang et al. “{BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning”. In: *2020 USENIX annual technical conference (USENIX ATC 20)*. 2020, pp. 493–506.

[67] G Peter Zhang and Min Qi. “Neural network forecasting for seasonal and trend time series”. In: *European journal of operational research* 160.2 (2005), pp. 501–514.

[68] Hongwei Zhang et al. “Federated multi-task learning with non-stationary and heterogeneous data in wireless networks”. In: *IEEE Transactions on Wireless Communications* (2023).

[69] Xinwei Zhang et al. “Hybrid federated learning: Algorithms and implementation”. In: *arXiv preprint arXiv:2012.12420* (2020).

[70] Yu Zhang and Qiang Yang. “A survey on multi-task learning”. In: *IEEE transactions on knowledge and data engineering* 34.12 (2021), pp. 5586–5609.

[71] Yue Zhao et al. “Federated learning with non-iid data”. In: *arXiv preprint arXiv:1806.00582* (2018).

[72] Hangyu Zhu et al. “Federated learning on non-IID data: A survey”. In: *Neurocomputing* 465 (2021), pp. 371–390.

Glossary

FL Federated Learning
ARIMA Autoregressive Intergated Moving Average
ARIMAX Autoregressive Intergated Moving Average with eXogeneous inputs
FC Fully Connected
GRU Gated Recurrent Units
HFL Horizontal Federated Learning
IID Independent and Identically Distributed
LSTM Long-Short Term Memory
MAE Mean Absolute Error
MTFL Multi-Task Federated Learning
MTL Multi-Task Federated Learning
PMC Predictive Maintenance and Control
GDPR General Data Protection Regularization
TPFL Time-series-based Personalized Federated Learning
RNN Recurrent Neural Networks
SARIMA Seasonal Autoregressive Intergated Moving Average
SARIMAX Seasonal Autoregressive Intergated Moving Average with eXogeneous inputs
SID Sampled ID
SMPC Secure Multi-Party Computation
VFL Vertical Federated Learning

A

Shared-Private Attention Experiments

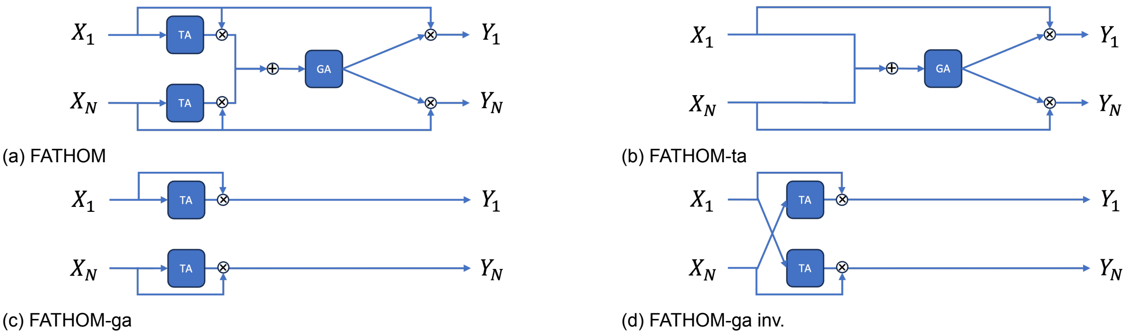
We evaluated an implementation of FATHOM that we developed using the public dataset Air Quality. For this dataset, we only model a selected univariate time series to exclude any possible interference of the model performance by the use of a (more complex) multivariate time series. Further details on the public datasets and selection of univariate series are given in Appendix D.

We conducted different types of experiments on this implementation, including testing multiple prediction windows and hyperparameters and evaluating diverse architectures to mitigate possible limitations of the architecture. However, during late experiments, we discovered that the one key reason for the limited predictive performance of our implementation was the use of attention units. This made most of our experiments irrelevant, and we will not discuss these in this thesis. We have selected a few experiments that did give us valuable insights for the continuation of the research and will discuss them in the following paragraphs.

To measure the contribution of global- and task-specific attention, we compare our implementation of FATHOM to different variants shown in Figure A.1, including a variant that does not apply task attention (FATHOM-ta, i.e. without task attention) and a variant that does not apply global attention (FATHOM-ga). In Figure A.2, we show that the performance of FATHOM-ta was similar to that of FATHOM whereas FATHOM-ga shows an improvement over FATHOM. We expect FATHOM to perform best, as this method balances task-specific and global knowledge. However, the global attention unit was unable to capture valuable shared patterns and even interfered with the performance of FATHOM.

A possible explanation is that the different tasks did not contain any valuable information concerning different clients. However, if we invert the task-attention (FATHOM-ga inv.), that is, we use the data from a different task to create a mask for the current task, we see that there is an improvement over FATHOM. This improvement demonstrates that each task contains information that can be of value to the other task. The global attention unit is unable to capture these due to overfitting or the model leaning more towards generalization at the cost of preserving task-specific knowledge.

Further experiments showed that the attention units in FATHOM did not work as we expected. We evaluated a modification of the architecture called FATHOM+ in which we replaced the task- and global



(a) FATHOM

(b) FATHOM-ta

(c) FATHOM-ga

(d) FATHOM-ga inv.

Figure A.1: Different modifications of FATHOM

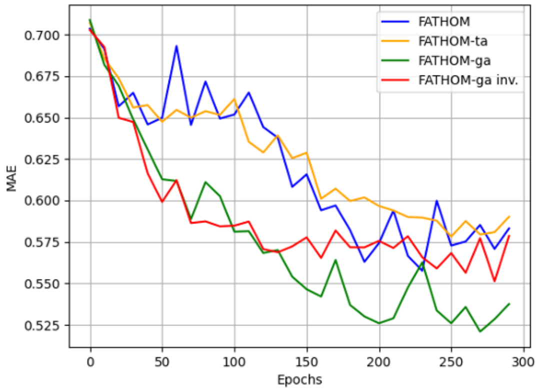


Figure A.2: Comparison of different modifications of FATHOM

attention with a task-specific attention mechanism that calculates the mask based on the intermediary representations of all tasks, as shown in Figure A.3. We compared FATHOM+ to a version of itself that excluded the attention units from updating the model weights during back-propagation, leaving the units static during the whole training process. The results in Figure A.4 show that the inclusion of the attention unit results in a degradation of the model performance due to the limited amount of training data available or the units overfitting on the training data. This unexpected behaviour also appeared in previous variants of FATHOM, showing that the attention units did not work for the datasets.

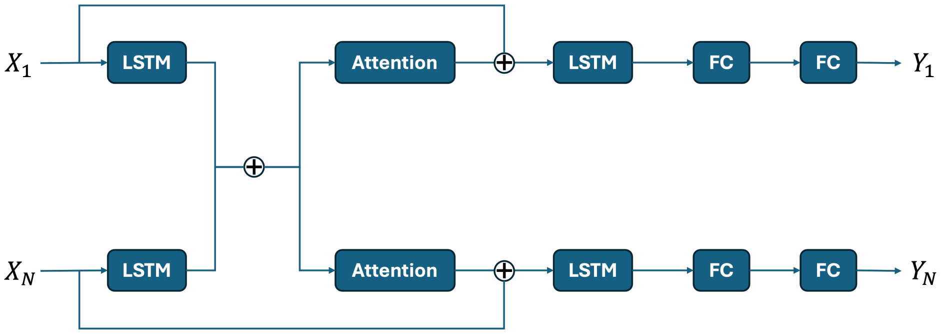


Figure A.3: Implementation of FATHOM+

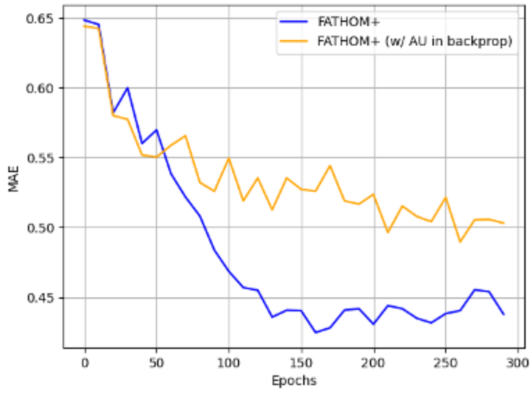


Figure A.4: Comparison of different modifications of FATHOM

B Clustering Attention Experiments

We evaluated static clustering on the public Electricity dataset containing multiple tasks with univariate time series. Further details on the public dataset are given in Appendix D. We based the similarity of different clients on the correlation parameter and manually selected which client belongs to which cluster. For simplicity, we create two distinctive cases of static clustering:

1. **Hard clustering**: we cluster clients and train one model per cluster. We exchange no knowledge between the clusters.

$$\theta_k = \frac{1}{|C_k|} \sum_{n \in C_k} \theta_n \tag{B.1}$$

2. **Soft clustering**: we train one model per cluster and allow for knowledge exchange with model weights. The new model weights are a weighted average of each cluster model based on the similarity with the current cluster.

$$\theta_k = \frac{1}{\sum_{k' \in K} s(k, k')} \sum_{k' \in K} s(k, k') \cdot \theta_{k'} \tag{B.2}$$

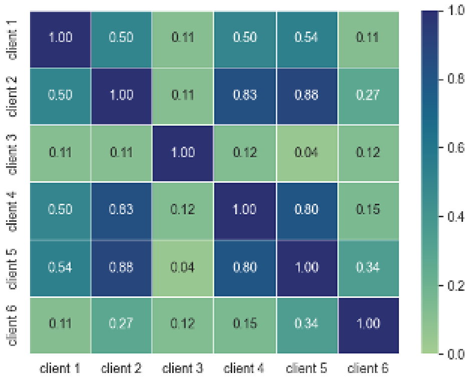
In Figure B.1, we show the correlation values between clients of the dataset. We can see in the heatmap that clients 2, 4 and 5 are highly correlated. We make one cluster with these three clients and independent clusters for all other clients. During the experiments, we allow each cluster to train its model independently for 10 training epochs, after which we share knowledge between the cluster model using Equation B.2. We repeat this cycle four times, resulting in 40 training epochs with four moments of cluster knowledge exchange.

In Figure B.1, we show the results per client for four different scenarios: Independent, Centralized, Hard- and Soft clustering. In Independent, we train one model per client and do not exchange any information between the clients; in Centralized, we train one model for all clients.

We first compare the Independent with the Centralized scenario. On average, Centralized performs worse than Independent; per client, we see performance gains for clients 1 and 3 and performance degradation for all other clients. The shared model trained in the Centralized scenario should generalize to all, but due to the heterogeneous data distributions, this model serves only two clients, whilst all other clients are better off using a client-specific model.

Hard clustering could potentially overcome the performance degradations of most clients by grouping those with similar data distributions. With our configuration of clusters, we overcome the performance degradation for all applicable clients. For client 6, this falls in line with the expectation as we expect a similar MAE as for the Independent scenario. However, Hard Clustering does not give us any performance gains in cluster 1 (as we would expect), and we have lost the performance gains for clients 1 and 3.

Soft clustering could solve this problem by keeping clusters partially isolated while sharing knowledge between clusters. Clients 1 and 3 benefited from a generalized model in which knowledge is fully



	client 1	client 2	client 3	client 4	client 5	client 6
client 1	1.00	0.50	0.11	0.50	0.54	0.11
client 2	0.50	1.00	0.11	0.83	0.88	0.27
client 3	0.11	0.11	1.00	0.12	0.04	0.12
client 4	0.50	0.83	0.12	1.00	0.80	0.15
client 5	0.54	0.88	0.04	0.80	1.00	0.34
client 6	0.11	0.27	0.12	0.15	0.34	1.00

Figure B.1: Correlation Electricity

shared, and cluster 1 could potentially improve its performance. Looking at the results (when comparing to Independent), we see that this was unsolved. There is a performance gain for client 1, whereas all other clients have degraded in performance.

Cluster	Client	PW	Independent	Centralised	Hard Clustering	Soft Clustering
-	Average	1	**2.77 +/- 0.11**	2.84 +/- 0.14	2.82 +/- 0.13	2.86 +/- 0.13
		2	3.16 +/- 0.13	**3.13 +/- 0.13**	3.14 +/- 0.13	3.26 +/- 0.15
		4	**3.85 +/- 0.19**	3.76 +/- 0.14	3.86 +/- 0.20	3.90 +/- 0.19
		8	**4.19 +/- 0.13**	4.55 +/- 0.10	4.31 +/- 0.15	4.34 +/- 0.13
		16	**5.44 +/- 0.45**	6.06 +/- 0.26	5.60 +/- 0.46	5.62 +/- 0.37
		Avg.	**3.88 +/- 0.20**	4.07 +/- 0.15	3.95 +/- 0.22	3.99 +/- 0.19
Cluster 1	Client2	1	**1.61 +/- 0.00**	1.69 +/- 0.00	1.67 +/- 0.00	2.94 +/- 0.00
		2	**1.94 +/- 0.00**	1.99 +/- 0.00	2.06 +/- 0.00	3.28 +/- 0.00
		4	**2.46 +/- 0.00**	2.50 +/- 0.00	2.64 +/- 0.00	4.23 +/- 0.00
		8	**2.89 +/- 0.00**	3.49 +/- 0.00	3.36 +/- 0.00	4.80 +/- 0.00
		16	**3.92 +/- 0.00**	5.04 +/- 0.00	4.14 +/- 0.00	5.37 +/- 0.00
		Avg.	**2.57 +/- 0.00**	2.94 +/- 0.00	2.77 +/- 0.00	4.13 +/- 0.00
	Client4	1	2.92 +/- 0.00	4.16 +/- 0.00	**2.75 +/- 0.00**	4.72 +/- 0.00
		2	3.37 +/- 0.00	4.67 +/- 0.00	**3.14 +/- 0.00**	4.84 +/- 0.02
		4	4.02 +/- 0.00	5.56 +/- 0.00	**3.90 +/- 0.00**	5.12 +/- 0.00
		8	**4.52 +/- 0.00**	5.58 +/- 0.00	4.84 +/- 0.00	6.27 +/- 0.00
		16	**5.30 +/- 0.00**	9.01 +/- 0.00	5.49 +/- 0.00	7.30 +/- 0.00
		Avg.	**4.02 +/- 0.00**	5.80 +/- 0.00	**4.02 +/- 0.00**	5.65 +/- 0.00
	Client5	1	1.91 +/- 0.00	2.84 +/- 0.00	**1.87 +/- 0.00**	1.88 +/- 0.00
		2	2.22 +/- 0.00	3.18 +/- 0.00	2.15 +/- 0.00	2.22 +/- 0.00
		4	**2.71 +/- 0.00**	3.95 +/- 0.00	2.74 +/- 0.00	**2.71 +/- 0.00**
		8	3.37 +/- 0.00	5.02 +/- 0.00	**3.34 +/- 0.00**	3.57 +/- 0.00
		16	**3.73 +/- 0.00**	5.84 +/- 0.00	3.81 +/- 0.00	4.42 +/- 0.00
		Avg.	2.79 +/- 0.00	4.17 +/- 0.00	**2.78 +/- 0.00**	2.96 +/- 0.00
Cluster 2	Client1	1	4.66 +/- 0.01	4.62 +/- 0.00	4.86 +/- 0.02	**1.73 +/- 0.00**
		2	4.95 +/- 0.00	4.61 +/- 0.00	4.66 +/- 0.00	**2.09 +/- 0.00**
		4	5.67 +/- 0.02	5.03 +/- 0.00	5.54 +/- 0.04	**2.44 +/- 0.00**
		8	6.13 +/- 0.02	5.86 +/- 0.00	6.26 +/- 0.01	**3.07 +/- 0.00**
		16	7.35 +/- 0.00	6.90 +/- 0.01	7.48 +/- 0.01	**4.02 +/- 0.00**
		Avg.	5.75 +/- 0.01	5.40 +/- 0.00	5.76 +/- 0.02	**2.67 +/- 0.00**
Cluster 3	Client3	1	3.50 +/- 0.00	**1.93 +/- 0.00**	3.50 +/- 0.00	3.93 +/- 0.00
		2	4.28 +/- 0.00	**2.21 +/- 0.00**	4.27 +/- 0.00	4.88 +/- 0.00
		4	5.58 +/- 0.02	**2.88 +/- 0.00**	5.19 +/- 0.01	6.15 +/- 0.01
		8	4.98 +/- 0.00	**4.10 +/- 0.00**	4.99 +/- 0.00	5.03 +/- 0.00
		16	9.11 +/- 0.00	**5.70 +/- 0.00**	9.09 +/- 0.00	8.96 +/- 0.00
		Avg.	5.49 +/- 0.01	**3.36 +/- 0.00**	5.41 +/- 0.00	5.79 +/- 0.00
Cluster 4	Client6	1	2.01 +/- 0.00	**1.80 +/- 0.00**	1.90 +/- 0.00	1.93 +/- 0.00
		2	2.21 +/- 0.00	**2.11 +/- 0.00**	2.17 +/- 0.00	2.22 +/- 0.00
		4	2.64 +/- 0.00	2.65 +/- 0.00	**2.63 +/- 0.00**	2.75 +/- 0.00
		8	**3.25 +/- 0.00**	3.27 +/- 0.00	3.29 +/- 0.00	3.33 +/- 0.00
		16	3.24 +/- 0.00	3.84 +/- 0.00	**3.23 +/- 0.00**	3.64 +/- 0.00
		Avg.	2.67 +/- 0.00	2.73 +/- 0.00	**2.64 +/- 0.00**	2.77 +/- 0.00

Table B.1: Clustering

C Personalization Experiments

We evaluated FedProx and FedPer using multiple datasets with univariate time series. Additionally, we evaluated FedProx and KnnPer using datasets with multivariate time series. Further details on these datasets are given in Appendix D. We compare all methods to Independent, Centralized and FedAvg. We ran experiments for FedProx with three different values for μ: 0.01, 0.001 and 0.0. We ran all univariate experiments for 40 epochs, and multivariate experiments for 30 epochs. In the FedProx experiments, we only aggregate after 10 epochs. We discuss the univariate and multivariate experiments in different subsections.

In Table C.1, our experiments show that FedProx consistently outperforms FedPer and FedAvg, highlighting the potential of personalization, particularly because its design allows clients to train their local models to align better alignment with local data distributions. However, these improvements over FedAvg are limited for all datasets, showing that this method can only adapt the model to the local distribution to a limited extent. The global model's shared parameters still dominate, leading to suboptimal personalization for client data.

Furthermore, FedProx can not improve over Independent and Centralized for highly heterogeneous dataset Electricity. In this dataset, we see that Independent performs better than Centralized on average because the distributions are too dissimilar for training one model. We expected that FedProx would perform better because it accounts for these heterogeneous distributions. However, this did not work because FedProx still has too much global information.

FedPer did not consistently outperform FedAvg and, in two cases, showed worse performance. The reason for this is likely the simultaneous update of the global and local components in FedPer, which can lead to overfitting in the local layers while the global layers' remain not under-optimized. This simultaneous update mechanism may prevent the global model from capturing useful general patterns, while the local components may not have enough flexibility to fully adapt to the diverse local data distributions.

The results for the experiments with multivariate datasets are shown in Table C.2. Different from the univariate experiments, we see that FedProx does not improve on FedAvg in three of five cases (on average). One reason for this is that FedProx aggregates the model only four times throughout the entire training process, which is likely insufficient for more complex models dealing with multivariate data. Multivariate time series tend to have richer and more complex relationships across different variables, making more frequent aggregation necessary to capture the temporal dependencies between these variables effectively.

KnnPer consistently performs better than the other methods in most datasets. KnnPer outperforms the Independent baseline in all cases, demonstrating significant performance improvements even when clients are highly heterogeneous. This suggests that KnnPer's non-parametric approach provides a more personalized and accurate model for clients' data distribution. Furthermore, in four out of five cases, KnnPer outperforms the Centralized model, highlighting that its local memorization technique is particularly effective in preserving the unique characteristics of clients' data while still benefiting from collaborative training. This allows KnnPer to make predictions that align more closely with the underlying data distributions at the local level, avoiding the pitfalls of generalization seen in Centralized approaches.

Dataset	PW	Independent	Centralised	FedAvg	FedProx $(\mu = 0.01)$	FedProx $(\mu = 0.001)$	FedProx $(\mu = 0.0)$	FedPer
AirQuality	1	1.82 +/- 0.02	1.28 +/- 0.00	1.32 +/- 0.00	1.62 +/- 0.00	**1.21 +/- 0.00**	1.23 +/- 0.00	2.08 +/- 0.03
	2	1.71 +/- 0.01	**1.41 +/- 0.00**	1.63 +/- 0.00	1.82 +/- 0.00	1.44 +/- 0.00	1.45 +/- 0.00	1.96 +/- 0.01
	4	1.76 +/- 0.00	**1.50 +/- 0.00**	1.72 +/- 0.00	1.84 +/- 0.00	1.60 +/- 0.00	1.60 +/- 0.00	1.92 +/- 0.00
	8	3.76 +/- 0.01	3.28 +/- 0.00	3.29 +/- 0.01	4.03 +/- 0.01	3.51 +/- 0.01	**3.49 +/- 0.01**	3.76 +/- 0.01
	16	5.39 +/- 0.02	**4.43 +/- 0.01**	5.19 +/- 0.02	5.25 +/- 0.01	5.01 +/- 0.02	5.06 +/- 0.02	5.51 +/- 0.01
	Avg.	2.89 +/- 0.01	**2.38 +/- 0.00**	2.63 +/- 0.01	2.91 +/- 0.01	2.55 +/- 0.01	2.57 +/- 0.01	3.05 +/- 0.01
Electricity	1	**2.81 +/- 0.13**	2.84 +/- 0.14	2.84 +/- 0.12	2.97 +/- 0.14	2.86 +/- 0.13	2.82 +/- 0.13	2.83 +/- 0.12
	2	3.17 +/- 0.14	**3.13 +/- 0.13**	3.19 +/- 0.12	3.42 +/- 0.15	3.19 +/- 0.12	3.19 +/- 0.14	3.19 +/- 0.13
	4	3.80 +/- 0.18	**3.76 +/- 0.14**	3.93 +/- 0.18	4.04 +/- 0.19	3.79 +/- 0.14	3.78 +/- 0.15	3.88 +/- 0.20
	8	**4.23 +/- 0.14**	4.55 +/- 0.10	4.37 +/- 0.12	4.46 +/- 0.12	**4.23 +/- 0.11**	**4.23 +/- 0.13**	4.28 +/- 0.14
	16	**5.48 +/- 0.45**	6.06 +/- 0.26	5.70 +/- 0.29	5.92 +/- 0.36	5.66 +/- 0.41	5.56 +/- 0.41	5.51 +/- 0.48
	Avg.	**3.90 +/- 0.21**	4.07 +/- 0.15	4.01 +/- 0.17	4.16 +/- 0.19	3.94 +/- 0.18	3.92 +/- 0.19	3.94 +/- 0.21
Solar	1	1.00 +/- 0.01	**0.63 +/- 0.00**	1.29 +/- 0.00	1.11 +/- 0.01	1.09 +/- 0.01	1.11 +/- 0.01	1.04 +/- 0.02
	2	1.43 +/- 0.01	**0.78 +/- 0.00**	1.54 +/- 0.01	1.53 +/- 0.01	1.43 +/- 0.01	1.50 +/- 0.01	1.48 +/- 0.01
	4	1.31 +/- 0.00	**0.89 +/- 0.00**	1.12 +/- 0.00	1.33 +/- 0.00	1.15 +/- 0.00	1.20 +/- 0.00	1.38 +/- 0.01
	8	2.07 +/- 0.01	**1.74 +/- 0.01**	2.19 +/- 0.01	2.03 +/- 0.01	1.99 +/- 0.01	1.95 +/- 0.01	2.11 +/- 0.01
	16	2.57 +/- 0.02	2.48 +/- 0.01	2.59 +/- 0.02	**2.38 +/- 0.02**	2.39 +/- 0.02	**2.38 +/- 0.02**	2.41 +/- 0.02
	Avg.	1.68 +/- 0.01	**1.30 +/- 0.01**	1.74 +/- 0.01	1.68 +/- 0.01	1.61 +/- 0.01	1.63 +/- 0.01	1.69 +/- 0.01
Industry	1	2.70 +/- 0.47	**0.88 +/- 0.03**	2.53 +/- 0.63	2.74 +/- 0.73	2.61 +/- 0.66	2.52 +/- 0.64	2.89 +/- 0.56
	2	1.55 +/- 0.14	**0.76 +/- 0.02**	1.32 +/- 0.14	1.32 +/- 0.10	1.29 +/- 0.11	1.33 +/- 0.11	1.58 +/- 0.16
	4	1.96 +/- 0.06	**1.57 +/- 0.04**	2.01 +/- 0.02	1.92 +/- 0.03	1.80 +/- 0.02	1.80 +/- 0.03	2.44 +/- 0.11
	8	1.90 +/- 0.18	**1.52 +/- 0.07**	3.21 +/- 0.80	2.59 +/- 0.79	2.65 +/- 0.75	2.68 +/- 0.74	2.98 +/- 0.82
	16	2.30 +/- 0.33	**2.20 +/- 0.38**	2.79 +/- 0.64	2.40 +/- 0.63	2.43 +/- 0.72	2.43 +/- 0.71	2.45 +/- 0.35
	Avg.	2.08 +/- 0.23	**1.39 +/- 0.11**	2.37 +/- 0.45	2.19 +/- 0.46	2.16 +/- 0.45	2.15 +/- 0.45	2.47 +/- 0.40

Table C.1: Results Personalization univariate

Overall, the results indicate that for multivariate time series, the complexity of the relationships between variables requires more sophisticated methods for capturing both global and local information. FedProx's global model aggregation struggles to adapt to these complexities due to its limited aggregation frequency, whereas KnnPer excels by focusing on client-specific patterns through memorization, making it more robust in such scenarios.

Dataset	PW	Independent	Centralised	FedAvg	FedProx ($\mu = 0.01$)	FedProx ($\mu = 0.001$)	FedProx ($\mu = 0.0$)	KnnPer
AirQuality	1	2.32 +/- 0.06	1.40 +/- 0.01	1.77 +/- 0.01	2.43 +/- 0.08	2.05 +/- 0.06	1.95 +/- 0.02	**1.31 +/- 0.00**
	2	2.38 +/- 0.02	1.55 +/- 0.00	2.06 +/- 0.04	2.51 +/- 0.03	2.19 +/- 0.02	2.10 +/- 0.01	**1.52 +/- 0.00**
	4	2.21 +/- 0.01	**1.90 +/- 0.01**	2.00 +/- 0.00	2.33 +/- 0.01	2.16 +/- 0.01	2.16 +/- 0.01	1.91 +/- 0.00
	8	4.42 +/- 0.02	3.60 +/- 0.01	3.63 +/- 0.01	4.21 +/- 0.02	3.98 +/- 0.01	3.95 +/- 0.01	**3.43 +/- 0.01**
	16	6.68 +/- 0.24	5.41 +/- 0.04	5.09 +/- 0.02	5.70 +/- 0.01	5.89 +/- 0.07	5.91 +/- 0.05	**4.74 +/- 0.02**
	Avg.	3.60 +/- 0.07	2.77 +/- 0.01	2.91 +/- 0.02	3.44 +/- 0.03	3.25 +/- 0.03	3.21 +/- 0.02	**2.58 +/- 0.00**
Industry	1	4.17 +/- 0.46	**1.30 +/- 0.02**	2.66 +/- 0.14	3.62 +/- 0.25	3.43 +/- 0.23	3.30 +/- 0.23	2.21 +/- 0.07
	2	4.76 +/- 0.58	**2.05 +/- 0.06**	3.16 +/- 0.21	4.31 +/- 0.25	4.02 +/- 0.19	4.02 +/- 0.20	2.61 +/- 0.15
	4	4.44 +/- 1.26	**3.54 +/- 0.43**	3.79 +/- 1.15	4.37 +/- 1.70	4.59 +/- 1.66	4.49 +/- 1.60	3.55 +/- 0.99
	8	3.45 +/- 0.34	**2.57 +/- 0.07**	3.83 +/- 0.31	4.06 +/- 0.55	4.13 +/- 0.55	4.15 +/- 0.50	2.79 +/- 0.27
	16	4.56 +/- 0.51	**2.92 +/- 0.11**	3.74 +/- 0.29	4.39 +/- 0.41	4.29 +/- 0.37	4.30 +/- 0.38	3.07 +/- 0.24
	Avg.	4.28 +/- 0.63	**2.48 +/- 0.14**	3.44 +/- 0.42	4.15 +/- 0.63	4.09 +/- 0.60	4.05 +/- 0.58	2.85 +/- 0.34
Sales	1	2.38 +/- 0.01	2.53 +/- 0.03	2.67 +/- 0.01	2.60 +/- 0.00	2.49 +/- 0.00	2.47 +/- 0.01	**2.27 +/- 0.02**
	2	2.64 +/- 0.01	2.59 +/- 0.01	2.92 +/- 0.01	2.83 +/- 0.01	2.77 +/- 0.01	2.77 +/- 0.01	**2.47 +/- 0.01**
	4	3.15 +/- 0.00	3.22 +/- 0.02	3.64 +/- 0.01	3.81 +/- 0.02	3.54 +/- 0.01	3.50 +/- 0.00	**3.03 +/- 0.01**
	8	3.65 +/- 0.01	**3.58 +/- 0.00**	4.53 +/- 0.03	7.38 +/- 0.06	4.43 +/- 0.02	4.42 +/- 0.03	3.70 +/- 0.02
	16	2.98 +/- 0.01	3.17 +/- 0.02	3.39 +/- 0.03	7.03 +/- 0.18	3.28 +/- 0.02	3.28 +/- 0.02	**2.69 +/- 0.03**
	Avg.	2.96 +/- 0.01	3.02 +/- 0.02	3.43 +/- 0.02	4.73 +/- 0.05	3.30 +/- 0.01	3.29 +/- 0.01	**2.83 +/- 0.02**
Crypto	1	2.24 +/- 0.14	**1.19 +/- 0.02**	1.71 +/- 0.05	1.87 +/- 0.05	1.97 +/- 0.09	1.94 +/- 0.07	1.32 +/- 0.01
	2	1.97 +/- 0.09	**1.17 +/- 0.02**	1.79 +/- 0.08	1.96 +/- 0.05	1.79 +/- 0.05	1.82 +/- 0.06	1.40 +/- 0.02
	4	2.41 +/- 0.11	**1.21 +/- 0.02**	1.74 +/- 0.04	2.06 +/- 0.03	2.00 +/- 0.06	1.93 +/- 0.05	1.57 +/- 0.03
	8	2.89 +/- 0.16	2.06 +/- 0.07	2.10 +/- 0.03	2.03 +/- 0.03	2.32 +/- 0.05	2.28 +/- 0.05	**1.84 +/- 0.03**
	16	3.67 +/- 0.19	2.80 +/- 0.10	2.38 +/- 0.00	2.66 +/- 0.02	2.58 +/- 0.03	2.44 +/- 0.01	**2.22 +/- 0.01**
	Avg.	2.64 +/- 0.14	1.69 +/- 0.04	1.94 +/- 0.04	2.11 +/- 0.04	2.13 +/- 0.05	2.08 +/- 0.05	**1.67 +/- 0.02**
Solar	1	1.40 +/- 0.01	**0.71 +/- 0.00**	1.49 +/- 0.01	1.67 +/- 0.02	1.32 +/- 0.03	1.51 +/- 0.02	0.83 +/- 0.01
	2	1.81 +/- 0.02	1.44 +/- 0.02	1.67 +/- 0.03	1.71 +/- 0.02	1.83 +/- 0.02	1.86 +/- 0.03	**1.16 +/- 0.02**
	4	1.67 +/- 0.02	1.13 +/- 0.00	1.43 +/- 0.01	1.59 +/- 0.02	1.48 +/- 0.02	1.61 +/- 0.02	**1.03 +/- 0.01**
	8	2.60 +/- 0.04	1.80 +/- 0.02	2.53 +/- 0.06	2.57 +/- 0.03	2.38 +/- 0.04	2.27 +/- 0.04	**1.69 +/- 0.03**
	16	3.36 +/- 0.07	**2.04 +/- 0.02**	3.13 +/- 0.06	3.47 +/- 0.07	3.17 +/- 0.06	3.34 +/- 0.11	2.20 +/- 0.05
	Avg.	2.17 +/- 0.03	1.42 +/- 0.01	2.05 +/- 0.03	2.20 +/- 0.03	2.04 +/- 0.03	2.12 +/- 0.05	**1.38 +/- 0.02**

Table C.2: Results Personalization multivariate

D
Datasets

We use four public datasets in our evaluation: Electricity [58], Air Quality [13], Solar Power [53], Crypto [4] and Rossman Sales [33]. Electricity, Air Quality and Solar were used for uni-variate experiments. Air Quality, Solar Power, Crypto and Rossman Sales were used for multivariate experiments. The average correlations between features in each multi-variate dataset can be found in Figure D.1.

All datasets follow a similar preprocessing protocol. We select samples in a given time frame, interpolate missing values and set the remaining missing values to 0. All data is normalized for consistent comparison. For the uni-variate experiments, the endogenous feature serves as the uni-variate time series.

D.1. Electricity

The Electricity dataset contains measurements per 15 min of electricity consumption at different locations in kWh. There are approximately 140,000 samples with one attribute. We select samples in a 1.5-month period for experiments.

D.2. Air Quality

The Air Quality dataset contains hourly data of different sensory measurements by twelve weather stations in Beijing. There are approximately 35000 samples of temporal data with 11 attributes under which gasses, temperature or wind direction. For our experiments, we specifically pick four attributes PM2.5, PM10, NO2 and CO, of six weather stations Aotizhongxin, Dingling, Gucheng, Huairou, Tiantan and Wanshouxigong. The data of each weather station is used for a separate task. During preprocessing, we select approximately samples in a two-month period. PM2.5 is the endogeneous feature, all other features are exogenous.

D.3. Rossman Sales

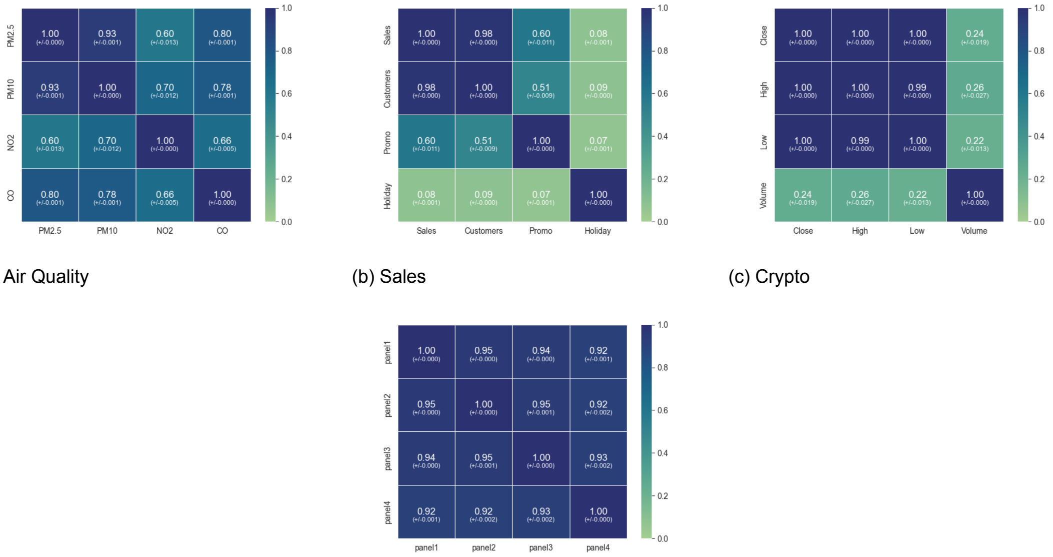
Rossman Sales contains historical sales data for 1115 Rossman stores. The data was measured on a daily basis on contains around 920 samples per store. We specifically selected four attributes: Sales, Customers, Promo (indicating if there is a promotion), and Holiday of stores 1 to 6. Sales serves as the endogenous feature. The last attribute is a combination of SchoolHoliday and StateHoliday which we combined during preprocessing. Specific configurations of batch size, input- and training window resulted in the use of approximately 720 samples in the experiments.

D.4. Crypto

The Crypto dataset contains historical trading data for different cryptocurrencies. The dataset contains a different number of samples for each currency since the initiated at different moments in history. All measurements were done per minute. We selected Close as endogenous feature; and Open, Close and Volume as exogenous features of six assets: Binance Coin, Bitcoin, Bitcoin Cash, Cardano, Dogecoin and EOS.IO. During pre-processing we select approximately 1600 samples in a two-month period and resample the data into hourly data - considering the correct aggregation function for each column.

D.5. Solar

Solar contains energy production measurements in MW for different solar panels located at multiple solar fields. The power consumption is measured every five minutes. We construct tasks by selecting multiple solar panels in one solar field and treating them as a task. We choose solar fields in Alabama, Florida, Illinois, Kansas, Massachusetts and Maine. We selected approximately 1800 in a 7 day period. We select one panel as an endogenous feature and use other panels as exogenous features.



(a) Air Quality

(b) Sales

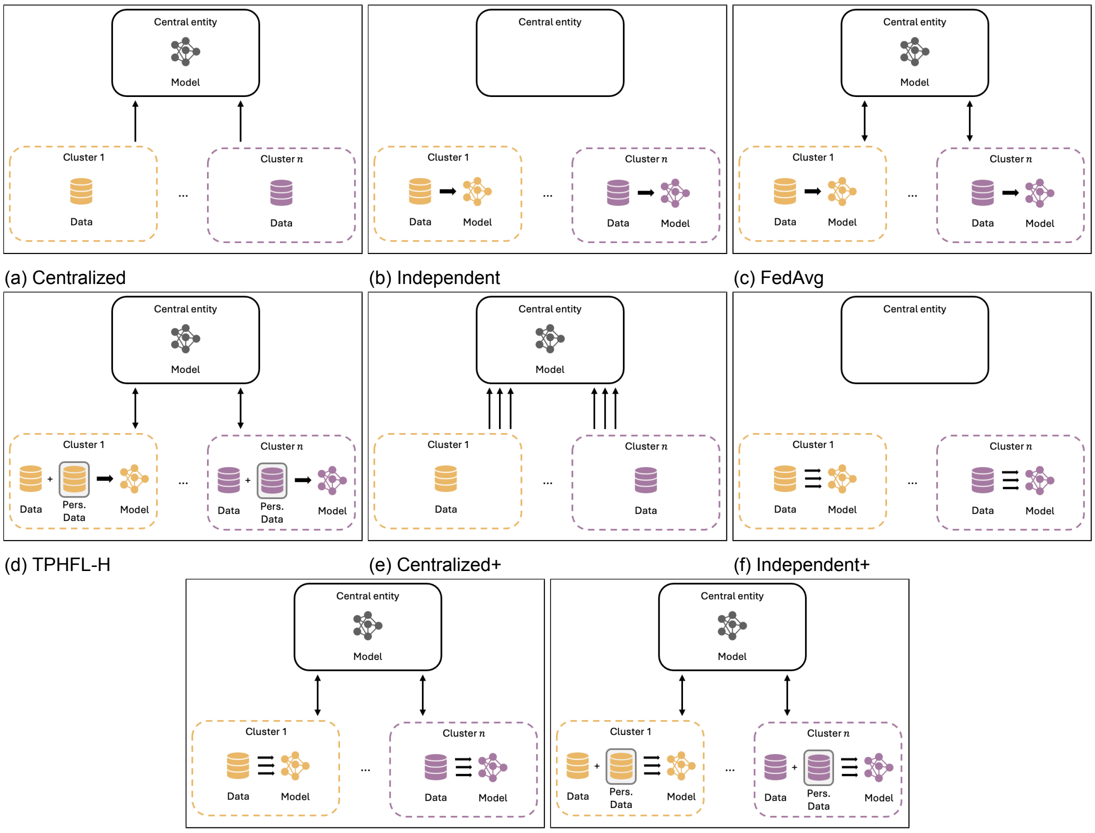
(c) Crypto

(d) Solar

Figure D.1: Correlation within tasks, each box contains the average correlation value and variance.

E

Baselines



(a) Centralized

(b) Independent

(c) FedAvg

(d) TPHFL-H

(e) Centralized+

(f) Independent+

(g) TPHFL-NP

(h) TPHFL

Figure E.1: Schematic figures of different baselines. Multiple arrows in the last four subfigures indicate distributed features.

In Figure E.1, we show schematic overviews of all baseline methods. In Section 5.1.1 we discussed different forms of data locality and collaboration. These forms translate to the following configurative choices:

Data locality:

- **None**: each cluster can share its data with the central entity, which trains one model with all data (Figures E.1a).
- **Horizontal**: each cluster does not share any information with others (Figures E.1b) or shares its model weights with the central entity, serving as the Federator, and receives updated model weights (Figures E.1c, E.1d).

- **Vertical**: each cluster shares distributed features with the central entity, which trains one model with all data (Figures E.1e).
- **Hybrid**: we combine the configurations of Horizontal and Vertical data locality. Each cluster trains its model using distributed features but does not share any information with the central entity (Figure E.1f) or trains with distributed features and shares only model weights (Figures E.1g, E.1h).

For collaboration, with exclusively Vertical collaboration, we do not exchange any information between the clusters (Figures E.1b, E.1f). In all other cases, there is an exchange between clusters.

F

Additional experiments

We compare TPHFL to all baseline methods, in three separate tables. In Table F.1, which is an expanded version of Table 5.2, we compare TPHFL to Independent and Centralized and show the relative improvement of TPHFL compared to Independent. In Table F.2, which is an expansion of Table 5.3, we compare three methods without vertical privacy constraints to their counterparts that enforce vertical restrictions. In Table F.3, which is an expansion of Table 5.3, we compare two methods that incorporate horizontal collaboration and personalization with their counterpart that do not use personalization. Lastly, in Table F.4, which is an expansion of Table 5.5, we show the improvements of TPHFL-l2 and TPHFL over Independent.

We conducted these experiments additionally for 2 and 4 tasks. The corresponding results can be found in Tables F.5 to F.12.

Dataset	PW	Independent	Centralized	TPHFL	Imp.
AirQuality	1	2.38 +/- 0.02	1.55 +/- 0.00	1.91 +/- 0.00	19.6%
	2	2.21 +/- 0.01	1.90 +/- 0.01	2.00 +/- 0.01	9.5%
	4	4.42 +/- 0.02	3.60 +/- 0.01	3.84 +/- 0.01	13.2%
	8	6.68 +/- 0.24	5.41 +/- 0.04	5.29 +/- 0.01	20.8%
	16	3.60 +/- 0.07	2.77 +/- 0.01	2.96 +/- 0.01	17.8%
	Avg.	4.17 +/- 0.46	1.30 +/- 0.02	2.71 +/- 0.11	34.9%
Industry	1	4.76 +/- 0.58	2.05 +/- 0.06	2.62 +/- 0.09	44.9%
	2	4.44 +/- 1.26	3.54 +/- 0.43	4.42 +/- 1.14	0.4%
	4	3.45 +/- 0.34	2.57 +/- 0.07	3.20 +/- 0.26	7.2%
	8	4.56 +/- 0.51	2.92 +/- 0.11	3.19 +/- 0.29	29.9%
	16	4.28 +/- 0.63	2.48 +/- 0.14	3.23 +/- 0.38	24.4%
	Avg.	2.38 +/- 0.01	2.53 +/- 0.03	3.24 +/- 0.06	-36.3%
Sales	1	2.64 +/- 0.01	2.59 +/- 0.01	3.61 +/- 0.04	-36.6%
	2	3.15 +/- 0.00	3.22 +/- 0.02	4.34 +/- 0.03	-37.7%
	4	3.65 +/- 0.01	3.58 +/- 0.00	5.32 +/- 0.02	-45.8%
	8	2.98 +/- 0.01	3.17 +/- 0.02	5.50 +/- 0.14	-84.7%
	16	2.96 +/- 0.01	3.02 +/- 0.02	4.40 +/- 0.06	-48.7%
	Avg.	2.24 +/- 0.14	1.19 +/- 0.02	2.09 +/- 0.03	6.9%
Crypto	1	1.97 +/- 0.09	1.17 +/- 0.02	1.95 +/- 0.03	1.0%
	2	2.41 +/- 0.11	1.21 +/- 0.02	2.24 +/- 0.05	7.0%
	4	2.89 +/- 0.16	2.06 +/- 0.07	2.63 +/- 0.05	9.0%
	8	3.67 +/- 0.19	2.80 +/- 0.10	3.10 +/- 0.04	15.6%
	16	2.64 +/- 0.14	1.69 +/- 0.04	2.40 +/- 0.04	8.9%
	Avg.	1.40 +/- 0.01	0.71 +/- 0.00	0.99 +/- 0.02	29.5%
Solar	1	1.81 +/- 0.02	1.44 +/- 0.02	1.43 +/- 0.03	21.1%
	2	1.67 +/- 0.02	1.13 +/- 0.00	1.34 +/- 0.02	19.4%
	4	2.60 +/- 0.04	1.80 +/- 0.02	1.86 +/- 0.04	28.3%
	8	3.36 +/- 0.07	2.04 +/- 0.02	2.27 +/- 0.08	32.5%
	16	2.17 +/- 0.03	1.42 +/- 0.01	1.58 +/- 0.04	27.2%

Table F.1: Average MAE and standard deviation for methods with no privacy constraints and TPHFL. The relative improvement of TPHFL over Independent (Rel. Imp.) is shown in percentages.

Dataset	PW	Independent	Independent+	Imp.	Centralized	Centralized+	Imp.	TPHFL-H	TPHFL	Imp.
AirQuality	1	2.32 +/- 0.06	2.72 +/- 0.03	-16.8%	1.40 +/- 0.01	1.58 +/- 0.01	-12.6%	1.31 +/- 0.00	1.78 +/- 0.00	-35.9%
	2	2.38 +/- 0.02	2.51 +/- 0.01	-5.6%	1.55 +/- 0.00	1.85 +/- 0.01	-19.6%	1.52 +/- 0.00	1.91 +/- 0.00	-25.3%
	4	2.21 +/- 0.01	2.46 +/- 0.03	-11.2%	1.90 +/- 0.01	2.19 +/- 0.01	-15.1%	1.91 +/- 0.00	2.00 +/- 0.01	-4.9%
	8	4.42 +/- 0.02	4.69 +/- 0.03	-6.2%	3.60 +/- 0.01	3.45 +/- 0.01	4.3%	3.43 +/- 0.01	3.84 +/- 0.01	-12.0%
	16	6.68 +/- 0.24	6.23 +/- 0.06	6.7%	5.41 +/- 0.04	4.92 +/- 0.03	9.1%	4.74 +/- 0.02	5.29 +/- 0.01	-11.6%
	Avg.	3.60 +/- 0.07	3.72 +/- 0.03	-3.3%	2.77 +/- 0.01	2.80 +/- 0.01	-0.9%	2.58 +/- 0.00	2.96 +/- 0.01	-14.8%
Industry	1	4.17 +/- 0.46	4.73 +/- 0.32	-13.4%	1.30 +/- 0.02	1.34 +/- 0.01	-2.8%	2.21 +/- 0.07	2.71 +/- 0.11	-22.5%
	2	4.76 +/- 0.58	5.68 +/- 0.37	-19.4%	2.05 +/- 0.06	1.80 +/- 0.04	12.4%	2.61 +/- 0.15	2.62 +/- 0.09	-0.7%
	4	4.44 +/- 1.26	6.66 +/- 2.08	-49.9%	3.54 +/- 0.43	3.34 +/- 0.58	5.5%	3.55 +/- 0.99	4.42 +/- 1.14	-24.6%
	8	3.45 +/- 0.34	6.18 +/- 1.24	-79.3%	2.57 +/- 0.07	3.07 +/- 0.26	-19.4%	2.79 +/- 0.27	3.20 +/- 0.26	-14.8%
	16	4.56 +/- 0.51	5.55 +/- 0.52	-21.6%	2.92 +/- 0.11	3.58 +/- 0.25	-22.6%	3.07 +/- 0.24	3.19 +/- 0.29	-4.1%
	Avg.	4.28 +/- 0.63	5.76 +/- 0.91	-34.7%	2.48 +/- 0.14	2.62 +/- 0.23	-6.0%	2.85 +/- 0.34	3.23 +/- 0.38	-13.5%
Sales	1	2.38 +/- 0.01	4.33 +/- 0.09	-82.1%	2.53 +/- 0.03	2.39 +/- 0.02	5.6%	2.27 +/- 0.02	3.24 +/- 0.06	-42.8%
	2	2.64 +/- 0.01	4.72 +/- 0.07	-78.4%	2.59 +/- 0.01	3.46 +/- 0.38	-33.7%	2.47 +/- 0.01	3.61 +/- 0.04	-46.1%
	4	3.15 +/- 0.00	4.75 +/- 0.07	-50.8%	3.22 +/- 0.02	3.36 +/- 0.01	-4.5%	3.03 +/- 0.01	4.34 +/- 0.03	-43.0%
	8	3.65 +/- 0.01	6.83 +/- 0.03	-87.1%	3.58 +/- 0.00	3.79 +/- 0.01	-5.9%	3.70 +/- 0.02	5.32 +/- 0.02	-43.6%
	16	2.98 +/- 0.01	5.24 +/- 0.09	-76.1%	3.17 +/- 0.02	3.04 +/- 0.03	4.1%	2.69 +/- 0.03	5.50 +/- 0.14	-104.0%
	Avg.	2.96 +/- 0.01	5.17 +/- 0.07	-74.8%	3.02 +/- 0.02	3.21 +/- 0.09	-6.3%	2.83 +/- 0.02	4.40 +/- 0.06	-55.3%
Crypto	1	2.24 +/- 0.14	3.50 +/- 0.43	-55.9%	1.19 +/- 0.02	1.48 +/- 0.03	-24.5%	1.32 +/- 0.01	2.09 +/- 0.03	-58.0%
	2	1.97 +/- 0.09	3.32 +/- 0.39	-68.7%	1.17 +/- 0.02	1.21 +/- 0.02	-3.6%	1.40 +/- 0.02	1.95 +/- 0.03	-39.1%
	4	2.41 +/- 0.11	2.98 +/- 0.14	-23.5%	1.21 +/- 0.02	1.28 +/- 0.02	-5.7%	1.57 +/- 0.03	2.24 +/- 0.05	-42.9%
	8	2.89 +/- 0.16	4.37 +/- 0.52	-51.1%	2.06 +/- 0.07	1.65 +/- 0.02	20.0%	1.84 +/- 0.03	2.63 +/- 0.05	-43.0%
	16	3.67 +/- 0.19	5.40 +/- 0.55	-47.0%	2.80 +/- 0.10	2.43 +/- 0.07	13.2%	2.22 +/- 0.01	3.10 +/- 0.04	-39.3%
	Avg.	2.64 +/- 0.14	3.91 +/- 0.41	-48.3%	1.69 +/- 0.04	1.61 +/- 0.03	4.5%	1.67 +/- 0.02	2.40 +/- 0.04	-43.7%
Solar	1	1.40 +/- 0.01	1.70 +/- 0.02	-21.2%	0.71 +/- 0.00	0.77 +/- 0.01	-7.8%	0.83 +/- 0.01	0.99 +/- 0.02	-19.5%
	2	1.81 +/- 0.02	2.51 +/- 0.08	-38.8%	1.44 +/- 0.02	1.19 +/- 0.02	17.5%	1.16 +/- 0.02	1.43 +/- 0.03	-23.4%
	4	1.67 +/- 0.02	2.10 +/- 0.02	-25.5%	1.13 +/- 0.00	0.99 +/- 0.01	12.1%	1.03 +/- 0.01	1.34 +/- 0.02	-30.9%
	8	2.60 +/- 0.04	3.08 +/- 0.06	-18.4%	1.80 +/- 0.02	1.82 +/- 0.01	-1.1%	1.69 +/- 0.03	1.86 +/- 0.04	-10.0%
	16	3.36 +/- 0.07	3.97 +/- 0.18	-18.0%	2.04 +/- 0.02	2.25 +/- 0.02	-10.5%	2.20 +/- 0.05	2.27 +/- 0.08	-3.2%
	Avg.	2.17 +/- 0.03	2.67 +/- 0.07	-23.1%	1.42 +/- 0.01	1.40 +/- 0.01	1.4%	1.38 +/- 0.02	1.58 +/- 0.04	-14.3%

Table F.2: Average MAE and standard deviation for different methods with Vertical restrictions and Horizontal restrictions. We compare the performance increase for methods if we introduce vertical privacy restrictions.

Dataset	PW	FedAvg	TPHFL-H	Imp.	TPHFL-NP	TPHFL	Imp.
AirQuality	1	1.77 +/- 0.01	1.31 +/- 0.00	26.2%	2.57 +/- 0.06	1.78 +/- 0.00	30.9%
	2	2.06 +/- 0.04	1.52 +/- 0.00	26.1%	2.42 +/- 0.02	1.91 +/- 0.00	21.1%
	4	2.00 +/- 0.00	1.91 +/- 0.00	4.3%	2.23 +/- 0.02	2.00 +/- 0.01	10.0%
	8	3.63 +/- 0.01	3.43 +/- 0.01	5.7%	4.41 +/- 0.01	3.84 +/- 0.01	12.9%
	16	5.09 +/- 0.02	4.74 +/- 0.02	6.9%	5.81 +/- 0.04	5.29 +/- 0.01	9.0%
	Avg.	2.91 +/- 0.02	2.58 +/- 0.00	11.3%	3.49 +/- 0.03	2.96 +/- 0.01	15.0%
Industry	1	2.66 +/- 0.14	2.21 +/- 0.07	16.7%	3.50 +/- 0.14	2.71 +/- 0.11	22.4%
	2	3.16 +/- 0.21	2.61 +/- 0.15	17.7%	3.53 +/- 0.14	2.62 +/- 0.09	25.6%
	4	3.79 +/- 1.15	3.55 +/- 0.99	6.4%	5.27 +/- 1.46	4.42 +/- 1.14	16.1%
	8	3.83 +/- 0.31	2.79 +/- 0.27	27.2%	4.70 +/- 0.58	3.20 +/- 0.26	31.9%
	16	3.74 +/- 0.29	3.07 +/- 0.24	17.9%	4.43 +/- 0.44	3.19 +/- 0.29	28.0%
	Avg.	3.44 +/- 0.42	2.85 +/- 0.34	17.2%	4.29 +/- 0.55	3.23 +/- 0.38	24.6%
Sales	1	2.67 +/- 0.01	2.27 +/- 0.02	14.8%	7.29 +/- 0.11	3.24 +/- 0.06	55.5%
	2	2.92 +/- 0.01	2.47 +/- 0.01	15.4%	7.41 +/- 0.10	3.61 +/- 0.04	51.3%
	4	3.64 +/- 0.01	3.03 +/- 0.01	16.6%	7.86 +/- 0.08	4.34 +/- 0.03	44.8%
	8	4.53 +/- 0.03	3.70 +/- 0.02	18.3%	8.29 +/- 0.02	5.32 +/- 0.02	35.8%
	16	3.39 +/- 0.03	2.69 +/- 0.03	20.5%	7.54 +/- 0.12	5.50 +/- 0.14	27.1%
	Avg.	3.43 +/- 0.02	2.83 +/- 0.02	17.3%	7.68 +/- 0.09	4.40 +/- 0.06	42.7%
Crypto	1	1.71 +/- 0.05	1.32 +/- 0.01	22.7%	2.60 +/- 0.10	2.09 +/- 0.03	19.7%
	2	1.79 +/- 0.08	1.40 +/- 0.02	21.7%	2.78 +/- 0.06	1.95 +/- 0.03	30.0%
	4	1.74 +/- 0.04	1.57 +/- 0.03	9.9%	2.97 +/- 0.03	2.24 +/- 0.05	24.4%
	8	2.10 +/- 0.03	1.84 +/- 0.03	12.2%	2.86 +/- 0.03	2.63 +/- 0.05	7.9%
	16	2.38 +/- 0.00	2.22 +/- 0.01	6.6%	3.50 +/- 0.03	3.10 +/- 0.04	11.3%
	Avg.	1.94 +/- 0.04	1.67 +/- 0.02	14.0%	2.94 +/- 0.05	2.40 +/- 0.04	18.3%
Solar	1	1.49 +/- 0.01	0.83 +/- 0.01	44.5%	1.63 +/- 0.02	0.99 +/- 0.02	39.3%
	2	1.67 +/- 0.03	1.16 +/- 0.02	30.8%	2.24 +/- 0.04	1.43 +/- 0.03	36.3%
	4	1.43 +/- 0.01	1.03 +/- 0.01	28.3%	2.20 +/- 0.04	1.34 +/- 0.02	38.8%
	8	2.53 +/- 0.06	1.69 +/- 0.03	32.9%	2.96 +/- 0.05	1.86 +/- 0.04	37.0%
	16	3.13 +/- 0.06	2.20 +/- 0.05	29.8%	3.86 +/- 0.15	2.27 +/- 0.08	41.2%
	Avg.	2.05 +/- 0.03	1.38 +/- 0.02	32.7%	2.58 +/- 0.06	1.58 +/- 0.04	38.8%

Table F.3: Average MAE and standard deviation for different methods with Horizontal and Hybrid privacy constraints. We show the improvement of introducing the personalization mechanism.

Dataset	PW	Independent	TPHFL-I2	Imp.	Independent	TPHFL	Imp.
AirQuality	1	2.38 +/- 0.02	1.90 +/- 0.00	20.0%	2.38 +/- 0.02	1.91 +/- 0.00	19.6%
	2	2.21 +/- 0.01	2.02 +/- 0.01	8.8%	2.21 +/- 0.01	2.00 +/- 0.01	9.5%
	4	4.42 +/- 0.02	3.76 +/- 0.01	15.0%	4.42 +/- 0.02	3.84 +/- 0.01	13.2%
	8	6.68 +/- 0.24	5.19 +/- 0.01	22.3%	6.68 +/- 0.24	5.29 +/- 0.01	20.8%
	16	3.60 +/- 0.07	2.93 +/- 0.00	18.7%	3.60 +/- 0.07	2.96 +/- 0.01	17.8%
	Avg.	4.17 +/- 0.46	2.84 +/- 0.12	31.8%	4.17 +/- 0.46	2.71 +/- 0.11	34.9%
Industry	1	4.76 +/- 0.58	2.81 +/- 0.10	40.9%	4.76 +/- 0.58	2.62 +/- 0.09	44.9%
	2	4.44 +/- 1.26	4.16 +/- 1.05	6.3%	4.44 +/- 1.26	4.42 +/- 1.14	0.4%
	4	3.45 +/- 0.34	3.31 +/- 0.29	4.1%	3.45 +/- 0.34	3.20 +/- 0.26	7.2%
	8	4.56 +/- 0.51	3.29 +/- 0.32	27.9%	4.56 +/- 0.51	3.19 +/- 0.29	29.9%
	16	4.28 +/- 0.63	3.28 +/- 0.38	23.2%	4.28 +/- 0.63	3.23 +/- 0.38	24.4%
	Avg.	2.38 +/- 0.01	3.70 +/- 0.09	-55.7%	2.38 +/- 0.01	3.24 +/- 0.06	-36.3%
Sales	1	2.64 +/- 0.01	3.76 +/- 0.05	-42.3%	2.64 +/- 0.01	3.61 +/- 0.04	-36.6%
	2	3.15 +/- 0.00	4.83 +/- 0.03	-53.4%	3.15 +/- 0.00	4.34 +/- 0.03	-37.7%
	4	3.65 +/- 0.01	7.60 +/- 0.01	-108.3%	3.65 +/- 0.01	5.32 +/- 0.02	-45.8%
	8	2.98 +/- 0.01	6.72 +/- 0.15	-125.9%	2.98 +/- 0.01	5.50 +/- 0.14	-84.7%
	16	2.96 +/- 0.01	5.32 +/- 0.07	-79.9%	2.96 +/- 0.01	4.40 +/- 0.06	-48.7%
	Avg.	2.24 +/- 0.14	2.11 +/- 0.02	5.8%	2.24 +/- 0.14	2.09 +/- 0.03	6.9%
Crypto	1	1.97 +/- 0.09	1.99 +/- 0.03	-1.0%	1.97 +/- 0.09	1.95 +/- 0.03	1.0%
	2	2.41 +/- 0.11	2.39 +/- 0.05	0.8%	2.41 +/- 0.11	2.24 +/- 0.05	7.0%
	4	2.89 +/- 0.16	2.65 +/- 0.05	8.4%	2.89 +/- 0.16	2.63 +/- 0.05	9.0%
	8	3.67 +/- 0.19	3.08 +/- 0.04	16.2%	3.67 +/- 0.19	3.10 +/- 0.04	15.6%
	16	2.64 +/- 0.14	2.44 +/- 0.04	7.4%	2.64 +/- 0.14	2.40 +/- 0.04	8.9%
	Avg.	1.40 +/- 0.01	1.01 +/- 0.01	28.2%	1.40 +/- 0.01	0.99 +/- 0.02	29.5%
Solar	1	1.81 +/- 0.02	1.45 +/- 0.02	20.0%	1.81 +/- 0.02	1.43 +/- 0.03	21.1%
	2	1.67 +/- 0.02	1.35 +/- 0.03	19.2%	1.67 +/- 0.02	1.34 +/- 0.02	19.4%
	4	2.60 +/- 0.04	1.90 +/- 0.04	27.0%	2.60 +/- 0.04	1.86 +/- 0.04	28.3%
	8	3.36 +/- 0.07	2.37 +/- 0.07	29.5%	3.36 +/- 0.07	2.27 +/- 0.08	32.5%
	16	2.17 +/- 0.03	1.61 +/- 0.04	25.6%	2.17 +/- 0.03	1.58 +/- 0.04	27.2%

Table F.4: Average MAE and standard deviation for Independent, TPHFL-I2 and TPHFL. We show the improvement over Independent.

Dataset	PW	Independent	Centralized	TPHFL	Imp.
AirQuality	1	2.13 +/- 0.03	1.91 +/- 0.05	2.09 +/- 0.00	1.8%
	2	2.17 +/- 0.00	1.80 +/- 0.01	2.11 +/- 0.00	2.9%
	4	2.16 +/- 0.00	2.03 +/- 0.00	2.21 +/- 0.00	-2.5%
	8	4.57 +/- 0.02	4.01 +/- 0.00	3.78 +/- 0.00	17.2%
	16	6.21 +/- 0.01	5.91 +/- 0.01	5.37 +/- 0.00	13.6%
	Avg.	3.45 +/- 0.01	3.13 +/- 0.02	3.11 +/- 0.00	9.7%
Industry	1	6.65 +/- 0.33	3.59 +/- 0.20	4.25 +/- 0.13	36.1%
	2	8.17 +/- 0.46	3.91 +/- 0.13	2.36 +/- 0.02	71.2%
	4	2.25 +/- 0.10	3.06 +/- 0.25	1.74 +/- 0.02	22.6%
	8	3.52 +/- 0.38	2.04 +/- 0.06	3.54 +/- 0.36	-0.5%
	16	7.19 +/- 0.28	4.12 +/- 0.37	4.91 +/- 0.06	31.7%
	Avg.	5.56 +/- 0.31	3.34 +/- 0.20	3.36 +/- 0.12	39.5%
Sales	1	2.58 +/- 0.01	2.37 +/- 0.01	2.77 +/- 0.02	-7.6%
	2	2.66 +/- 0.01	2.70 +/- 0.01	2.97 +/- 0.03	-11.4%
	4	3.32 +/- 0.00	3.16 +/- 0.00	3.44 +/- 0.02	-3.8%
	8	3.88 +/- 0.00	3.62 +/- 0.00	4.78 +/- 0.00	-23.0%
	16	2.98 +/- 0.00	2.80 +/- 0.00	4.52 +/- 0.05	-51.6%
	Avg.	3.09 +/- 0.00	2.93 +/- 0.00	3.70 +/- 0.02	-19.8%
Crypto	1	3.36 +/- 0.38	1.50 +/- 0.02	2.19 +/- 0.09	34.9%
	2	1.96 +/- 0.12	1.74 +/- 0.05	2.03 +/- 0.06	-3.2%
	4	2.24 +/- 0.07	1.14 +/- 0.01	1.63 +/- 0.02	27.3%
	8	3.31 +/- 0.24	1.82 +/- 0.05	2.60 +/- 0.16	21.5%
	16	4.90 +/- 0.26	2.29 +/- 0.02	5.10 +/- 0.29	-4.1%
	Avg.	3.15 +/- 0.21	1.70 +/- 0.03	2.71 +/- 0.12	14.2%
Solar	1	1.33 +/- 0.01	1.05 +/- 0.01	1.20 +/- 0.01	9.3%
	2	1.64 +/- 0.01	1.21 +/- 0.00	1.73 +/- 0.01	-5.5%
	4	1.58 +/- 0.00	1.20 +/- 0.00	1.54 +/- 0.02	2.5%
	8	2.19 +/- 0.03	2.03 +/- 0.05	2.14 +/- 0.03	2.4%
	16	3.24 +/- 0.14	2.35 +/- 0.03	2.68 +/- 0.07	17.4%
	Avg.	2.00 +/- 0.04	1.57 +/- 0.02	1.86 +/- 0.03	6.9%

Table F.5: 2 tasks: Average MAE and standard deviation for methods with no privacy constraints and TPHFL. The relative improvement of TPHFL over Independent (Rel. Imp.) is shown in percentages.

Dataset	PW	Independent	Centralized	TPHFL	Imp.
AirQuality	1	2.36 +/- 0.05	1.51 +/- 0.02	2.09 +/- 0.00	11.7%
	2	2.25 +/- 0.01	1.70 +/- 0.01	1.93 +/- 0.00	14.0%
	4	2.18 +/- 0.00	1.82 +/- 0.00	1.97 +/- 0.00	9.9%
	8	4.49 +/- 0.02	3.81 +/- 0.00	3.81 +/- 0.00	15.3%
	16	6.36 +/- 0.10	5.28 +/- 0.02	4.78 +/- 0.01	24.7%
	Avg.	3.53 +/- 0.04	2.82 +/- 0.01	2.91 +/- 0.00	17.4%
Industry	1	4.56 +/- 0.64	1.59 +/- 0.02	2.59 +/- 0.09	43.1%
	2	5.91 +/- 0.77	2.23 +/- 0.03	2.76 +/- 0.04	53.2%
	4	2.61 +/- 0.13	1.77 +/- 0.03	1.96 +/- 0.01	24.9%
	8	3.04 +/- 0.28	2.72 +/- 0.13	2.69 +/- 0.16	11.5%
	16	5.08 +/- 0.86	2.39 +/- 0.06	3.17 +/- 0.25	37.6%
	Avg.	4.24 +/- 0.54	2.14 +/- 0.05	2.64 +/- 0.11	37.8%
Sales	1	2.61 +/- 0.01	2.66 +/- 0.01	2.44 +/- 0.01	6.7%
	2	2.79 +/- 0.01	2.83 +/- 0.01	2.58 +/- 0.01	7.5%
	4	3.42 +/- 0.00	3.40 +/- 0.01	3.64 +/- 0.01	-6.2%
	8	3.93 +/- 0.01	3.55 +/- 0.02	5.71 +/- 0.00	-45.2%
	16	3.13 +/- 0.01	2.88 +/- 0.01	4.46 +/- 0.02	-42.4%
	Avg.	3.18 +/- 0.01	3.06 +/- 0.01	3.77 +/- 0.01	-18.4%
Crypto	1	2.48 +/- 0.27	1.07 +/- 0.01	2.20 +/- 0.04	11.1%
	2	1.75 +/- 0.07	1.30 +/- 0.02	1.69 +/- 0.01	3.0%
	4	1.96 +/- 0.05	1.20 +/- 0.01	1.95 +/- 0.03	0.6%
	8	2.74 +/- 0.16	1.65 +/- 0.02	2.60 +/- 0.06	5.0%
	16	3.82 +/- 0.25	2.42 +/- 0.02	3.77 +/- 0.09	1.4%
	Avg.	2.55 +/- 0.16	1.53 +/- 0.02	2.44 +/- 0.05	4.2%
Solar	1	1.38 +/- 0.02	1.16 +/- 0.01	0.91 +/- 0.02	33.6%
	2	1.88 +/- 0.06	1.48 +/- 0.01	1.25 +/- 0.03	33.2%
	4	1.78 +/- 0.02	1.13 +/- 0.01	1.16 +/- 0.02	34.8%
	8	2.39 +/- 0.07	1.75 +/- 0.02	1.60 +/- 0.04	32.8%
	16	3.30 +/- 0.13	2.09 +/- 0.03	2.07 +/- 0.08	37.4%
	Avg.	2.14 +/- 0.06	1.52 +/- 0.02	1.40 +/- 0.04	34.7%

Table F.6: 4 tasks: Average MAE and standard deviation for methods with no privacy constraints and TPHFL. The relative improvement of TPHFL over Independent (Rel. Imp.) is shown in percentages.

Dataset	PW	Independent	Independent+	Imp.	Centralized	Centralized+	Imp.	TPHFL-H	TPHFL	Imp.
AirQuality	1	2.13 +/- 0.03	2.77 +/- 0.02	-30.1%	1.91 +/- 0.05	1.88 +/- 0.01	1.8%	1.81 +/- 0.00	2.09 +/- 0.00	-15.4%
	2	2.17 +/- 0.00	2.39 +/- 0.01	-10.3%	1.80 +/- 0.01	1.97 +/- 0.01	-9.2%	1.84 +/- 0.00	2.11 +/- 0.00	-14.6%
	4	2.16 +/- 0.00	2.30 +/- 0.01	-6.6%	2.03 +/- 0.00	2.00 +/- 0.00	1.8%	1.90 +/- 0.00	2.21 +/- 0.00	-16.2%
	8	4.57 +/- 0.02	4.49 +/- 0.00	1.7%	4.01 +/- 0.00	4.06 +/- 0.00	-1.1%	3.60 +/- 0.01	3.78 +/- 0.00	-4.9%
	16	6.21 +/- 0.01	5.76 +/- 0.01	7.3%	5.91 +/- 0.01	5.49 +/- 0.02	7.0%	5.42 +/- 0.00	5.37 +/- 0.00	0.9%
	Avg.	3.45 +/- 0.01	3.54 +/- 0.01	-2.7%	3.13 +/- 0.02	3.08 +/- 0.01	1.8%	2.92 +/- 0.00	3.11 +/- 0.00	-6.7%
Industry	1	6.65 +/- 0.33	5.91 +/- 0.20	11.1%	3.59 +/- 0.20	6.36 +/- 1.43	-77.1%	4.34 +/- 0.20	4.25 +/- 0.13	2.2%
	2	8.17 +/- 0.46	6.17 +/- 0.54	24.6%	3.91 +/- 0.13	7.11 +/- 1.67	-81.7%	4.15 +/- 0.20	2.36 +/- 0.02	43.2%
	4	2.25 +/- 0.10	3.58 +/- 0.22	-59.1%	3.06 +/- 0.25	2.13 +/- 0.05	30.2%	1.38 +/- 0.02	1.74 +/- 0.02	-26.7%
	8	3.52 +/- 0.38	4.82 +/- 0.83	-37.0%	2.04 +/- 0.06	4.72 +/- 0.14	-131.6%	2.79 +/- 0.35	3.54 +/- 0.36	-26.8%
	16	7.19 +/- 0.28	6.05 +/- 0.21	15.9%	4.12 +/- 0.37	6.98 +/- 0.33	-69.3%	3.86 +/- 0.03	4.91 +/- 0.06	-27.3%
	Avg.	5.56 +/- 0.31	5.31 +/- 0.40	4.5%	3.34 +/- 0.20	5.46 +/- 0.72	-63.3%	3.30 +/- 0.16	3.36 +/- 0.12	-1.7%
Sales	1	2.58 +/- 0.01	5.37 +/- 0.03	-108.3%	2.37 +/- 0.01	2.79 +/- 0.01	-17.9%	1.97 +/- 0.01	2.77 +/- 0.02	-40.6%
	2	2.66 +/- 0.01	5.65 +/- 0.02	-112.1%	2.70 +/- 0.01	2.72 +/- 0.01	-0.6%	2.29 +/- 0.02	2.97 +/- 0.03	-29.7%
	4	3.32 +/- 0.00	6.14 +/- 0.03	-85.2%	3.16 +/- 0.00	3.57 +/- 0.00	-12.9%	2.92 +/- 0.01	3.44 +/- 0.02	-18.0%
	8	3.88 +/- 0.00	7.81 +/- 0.01	-101.1%	3.62 +/- 0.00	4.87 +/- 0.05	-34.6%	3.47 +/- 0.00	4.78 +/- 0.00	-37.6%
	16	2.98 +/- 0.00	6.11 +/- 0.01	-104.8%	2.80 +/- 0.00	3.35 +/- 0.00	-19.5%	2.42 +/- 0.00	4.52 +/- 0.05	-86.6%
	Avg.	3.09 +/- 0.00	6.22 +/- 0.02	-101.5%	2.93 +/- 0.00	3.46 +/- 0.02	-18.1%	2.62 +/- 0.01	3.70 +/- 0.02	-41.4%
Crypto	1	3.36 +/- 0.38	5.36 +/- 1.25	-59.4%	1.50 +/- 0.02	1.74 +/- 0.06	-16.1%	1.20 +/- 0.01	2.19 +/- 0.09	-82.6%
	2	1.96 +/- 0.12	4.24 +/- 0.74	-116.2%	1.74 +/- 0.05	2.00 +/- 0.10	-14.8%	1.18 +/- 0.00	2.03 +/- 0.06	-72.2%
	4	2.24 +/- 0.07	3.14 +/- 0.04	-40.4%	1.14 +/- 0.01	1.42 +/- 0.02	-24.2%	1.27 +/- 0.01	1.63 +/- 0.02	-27.6%
	8	3.31 +/- 0.24	5.88 +/- 0.96	-77.9%	1.82 +/- 0.05	2.65 +/- 0.14	-46.0%	1.55 +/- 0.01	2.60 +/- 0.16	-67.2%
	16	4.90 +/- 0.26	7.75 +/- 0.58	-58.2%	2.29 +/- 0.02	4.28 +/- 0.04	-86.7%	2.81 +/- 0.07	5.10 +/- 0.29	-81.2%
	Avg.	3.15 +/- 0.21	5.27 +/- 0.71	-67.3%	1.70 +/- 0.03	2.42 +/- 0.07	-42.4%	1.60 +/- 0.02	2.71 +/- 0.12	-68.8%
Solar	1	1.33 +/- 0.01	1.40 +/- 0.01	-5.7%	1.05 +/- 0.01	1.05 +/- 0.01	0.4%	1.13 +/- 0.01	1.20 +/- 0.01	-6.2%
	2	1.64 +/- 0.01	2.22 +/- 0.04	-35.4%	1.21 +/- 0.00	1.37 +/- 0.00	-13.1%	1.48 +/- 0.01	1.73 +/- 0.01	-16.8%
	4	1.58 +/- 0.00	1.95 +/- 0.02	-23.2%	1.20 +/- 0.00	1.21 +/- 0.00	-0.8%	1.12 +/- 0.00	1.54 +/- 0.02	-37.9%
	8	2.19 +/- 0.03	2.87 +/- 0.13	-30.9%	2.03 +/- 0.05	2.10 +/- 0.02	-3.4%	1.98 +/- 0.03	2.14 +/- 0.03	-7.8%
	16	3.24 +/- 0.14	3.50 +/- 0.09	-7.9%	2.35 +/- 0.03	2.66 +/- 0.05	-13.3%	2.45 +/- 0.06	2.68 +/- 0.07	-9.3%
	Avg.	2.00 +/- 0.04	2.39 +/- 0.06	-19.6%	1.57 +/- 0.02	1.68 +/- 0.02	-7.0%	1.63 +/- 0.02	1.86 +/- 0.03	-13.8%

Table F.7: 2 tasks: Average MAE and standard deviation for different methods with Vertical restrictions and Horizontal restrictions. We compare the performance increase for methods if we introduce vertical privacy restrictions.

Dataset	PW	Independent	Independent+	Imp.	Centralized	Centralized+	Imp.	TPHFL-H	TPHFL	Imp.
AirQuality	1	2.36 +/- 0.05	2.82 +/- 0.03	-19.2%	1.51 +/- 0.02	1.65 +/- 0.00	-9.0%	1.84 +/- 0.00	2.09 +/- 0.00	-13.2%
	2	2.25 +/- 0.01	2.56 +/- 0.01	-13.9%	1.70 +/- 0.01	1.74 +/- 0.00	-2.7%	1.73 +/- 0.00	1.93 +/- 0.00	-11.4%
	4	2.18 +/- 0.00	2.32 +/- 0.01	-6.2%	1.82 +/- 0.00	1.78 +/- 0.00	2.1%	1.92 +/- 0.00	1.97 +/- 0.00	-2.3%
	8	4.49 +/- 0.02	4.84 +/- 0.02	-7.8%	3.81 +/- 0.00	3.68 +/- 0.01	3.4%	3.61 +/- 0.00	3.81 +/- 0.00	-5.5%
	16	6.36 +/- 0.10	6.08 +/- 0.03	4.3%	5.28 +/- 0.02	5.14 +/- 0.03	2.7%	5.22 +/- 0.01	4.78 +/- 0.01	8.4%
	Avg.	3.53 +/- 0.04	3.72 +/- 0.02	-5.5%	2.82 +/- 0.01	2.80 +/- 0.01	0.9%	2.87 +/- 0.00	2.91 +/- 0.00	-1.7%
Industry	1	4.56 +/- 0.64	4.47 +/- 0.45	1.9%	1.59 +/- 0.02	3.09 +/- 0.16	-94.3%	1.78 +/- 0.00	2.59 +/- 0.09	-45.2%
	2	5.91 +/- 0.77	5.40 +/- 0.45	8.7%	2.23 +/- 0.03	3.09 +/- 0.13	-38.6%	1.93 +/- 0.03	2.76 +/- 0.04	-43.1%
	4	2.61 +/- 0.13	3.55 +/- 0.17	-36.0%	1.77 +/- 0.03	2.61 +/- 0.12	-48.0%	1.50 +/- 0.01	1.96 +/- 0.01	-31.0%
	8	3.04 +/- 0.28	4.32 +/- 0.61	-42.0%	2.72 +/- 0.13	3.84 +/- 0.40	-41.3%	1.99 +/- 0.08	2.69 +/- 0.16	-35.4%
	16	5.08 +/- 0.86	5.24 +/- 0.46	-3.1%	2.39 +/- 0.06	4.81 +/- 0.38	-101.4%	2.14 +/- 0.05	3.17 +/- 0.25	-48.1%
	Avg.	4.24 +/- 0.54	4.60 +/- 0.43	-8.4%	2.14 +/- 0.05	3.49 +/- 0.24	-63.1%	1.87 +/- 0.04	2.64 +/- 0.11	-41.1%
Sales	1	2.61 +/- 0.01	5.42 +/- 0.04	-107.5%	2.66 +/- 0.01	2.63 +/- 0.01	1.3%	2.10 +/- 0.01	2.44 +/- 0.01	-15.8%
	2	2.79 +/- 0.01	5.57 +/- 0.03	-99.3%	2.83 +/- 0.01	3.17 +/- 0.02	-11.7%	2.35 +/- 0.01	2.58 +/- 0.01	-9.8%
	4	3.42 +/- 0.00	6.21 +/- 0.02	-81.5%	3.40 +/- 0.01	3.66 +/- 0.01	-7.4%	2.92 +/- 0.01	3.64 +/- 0.01	-24.7%
	8	3.93 +/- 0.01	7.81 +/- 0.03	-98.6%	3.55 +/- 0.02	3.95 +/- 0.02	-11.3%	3.51 +/- 0.00	5.71 +/- 0.00	-62.8%
	16	3.13 +/- 0.01	6.40 +/- 0.03	-104.3%	2.88 +/- 0.01	3.05 +/- 0.01	-5.9%	2.47 +/- 0.01	4.46 +/- 0.02	-81.0%
	Avg.	3.18 +/- 0.01	6.28 +/- 0.03	-97.6%	3.06 +/- 0.01	3.29 +/- 0.02	-7.3%	2.67 +/- 0.01	3.77 +/- 0.01	-41.1%
Crypto	1	2.48 +/- 0.27	4.03 +/- 0.89	-62.7%	1.07 +/- 0.01	1.62 +/- 0.03	-52.4%	1.26 +/- 0.01	2.20 +/- 0.04	-74.5%
	2	1.75 +/- 0.07	3.19 +/- 0.50	-82.5%	1.30 +/- 0.02	1.44 +/- 0.01	-10.5%	1.30 +/- 0.01	1.69 +/- 0.01	-30.2%
	4	1.96 +/- 0.05	2.52 +/- 0.06	-28.8%	1.20 +/- 0.01	1.54 +/- 0.02	-28.6%	1.34 +/- 0.01	1.95 +/- 0.03	-45.0%
	8	2.74 +/- 0.16	4.47 +/- 0.73	-63.0%	1.65 +/- 0.02	2.21 +/- 0.04	-34.2%	1.65 +/- 0.01	2.60 +/- 0.06	-58.1%
	16	3.82 +/- 0.25	6.00 +/- 0.69	-57.0%	2.42 +/- 0.02	3.08 +/- 0.07	-27.2%	2.32 +/- 0.02	3.77 +/- 0.09	-62.5%
	Avg.	2.55 +/- 0.16	4.04 +/- 0.58	-58.6%	1.53 +/- 0.02	1.98 +/- 0.04	-29.6%	1.57 +/- 0.01	2.44 +/- 0.05	-55.2%
Solar	1	1.38 +/- 0.02	1.68 +/- 0.02	-22.0%	1.16 +/- 0.01	0.88 +/- 0.01	23.7%	0.78 +/- 0.01	0.91 +/- 0.02	-17.7%
	2	1.88 +/- 0.06	2.44 +/- 0.08	-30.2%	1.48 +/- 0.01	1.52 +/- 0.01	-2.8%	1.06 +/- 0.02	1.25 +/- 0.03	-18.6%
	4	1.78 +/- 0.02	1.97 +/- 0.02	-10.3%	1.13 +/- 0.01	1.38 +/- 0.01	-21.8%	0.93 +/- 0.01	1.16 +/- 0.02	-25.7%
	8	2.39 +/- 0.07	2.95 +/- 0.08	-23.5%	1.75 +/- 0.02	1.53 +/- 0.02	12.3%	1.57 +/- 0.05	1.60 +/- 0.04	-2.4%
	16	3.30 +/- 0.13	4.07 +/- 0.26	-23.3%	2.09 +/- 0.03	2.15 +/- 0.02	-2.8%	2.01 +/- 0.06	2.07 +/- 0.08	-2.9%
	Avg.	2.14 +/- 0.06	2.62 +/- 0.09	-22.2%	1.52 +/- 0.02	1.49 +/- 0.02	1.9%	1.27 +/- 0.03	1.40 +/- 0.04	-10.5%

Table F.8: 4 tasks: Average MAE and standard deviation for different methods with Vertical restrictions and Horizontal restrictions. We compare the performance increase for methods if we introduce vertical privacy restrictions.

Dataset	PW	FedAvg	TPHFL-H	Imp.	TPHFL-NP	TPHFL	Imp.
AirQuality	1	2.23 +/- 0.01	1.84 +/- 0.00	17.6%	2.35 +/- 0.01	2.11 +/- 0.00	10.4%
	2	2.10 +/- 0.00	1.90 +/- 0.00	9.5%	2.30 +/- 0.01	2.21 +/- 0.00	3.8%
	4	4.19 +/- 0.00	3.60 +/- 0.01	14.0%	4.46 +/- 0.01	3.78 +/- 0.00	15.3%
	8	5.73 +/- 0.01	5.42 +/- 0.00	5.4%	5.56 +/- 0.01	5.37 +/- 0.00	3.4%
	16	3.26 +/- 0.01	2.92 +/- 0.00	10.6%	3.51 +/- 0.02	3.11 +/- 0.00	11.5%
	Avg.	5.74 +/- 0.24	4.34 +/- 0.20	24.4%	4.77 +/- 0.36	4.25 +/- 0.13	11.1%
Industry	1	5.94 +/- 0.06	4.15 +/- 0.20	30.2%	3.74 +/- 0.39	2.36 +/- 0.02	37.1%
	2	2.30 +/- 0.08	1.38 +/- 0.02	40.3%	3.86 +/- 0.06	1.74 +/- 0.02	54.8%
	4	3.95 +/- 0.76	2.79 +/- 0.35	29.3%	6.78 +/- 0.45	3.54 +/- 0.36	47.8%
	8	5.88 +/- 0.12	3.86 +/- 0.03	34.4%	4.87 +/- 0.11	4.91 +/- 0.06	-0.8%
	16	4.76 +/- 0.25	3.30 +/- 0.16	30.7%	4.81 +/- 0.27	3.36 +/- 0.12	30.1%
	Avg.	2.50 +/- 0.01	1.97 +/- 0.01	21.0%	6.25 +/- 0.03	2.77 +/- 0.02	55.6%
Sales	1	2.73 +/- 0.01	2.29 +/- 0.02	16.3%	6.40 +/- 0.03	2.97 +/- 0.03	53.6%
	2	3.31 +/- 0.01	2.92 +/- 0.01	11.9%	7.27 +/- 0.02	3.44 +/- 0.02	52.6%
	4	4.14 +/- 0.00	3.47 +/- 0.00	16.1%	8.16 +/- 0.01	4.78 +/- 0.00	41.4%
	8	2.95 +/- 0.00	2.42 +/- 0.00	17.8%	6.76 +/- 0.04	4.52 +/- 0.05	33.0%
	16	3.13 +/- 0.01	2.62 +/- 0.01	16.3%	6.97 +/- 0.02	3.70 +/- 0.02	46.9%
	Avg.	1.41 +/- 0.02	1.20 +/- 0.01	15.1%	3.72 +/- 0.14	2.19 +/- 0.09	41.1%
Crypto	1	1.23 +/- 0.01	1.18 +/- 0.00	4.6%	3.63 +/- 0.09	2.03 +/- 0.06	44.3%
	2	1.70 +/- 0.01	1.27 +/- 0.01	25.2%	3.12 +/- 0.07	1.63 +/- 0.02	47.9%
	4	1.99 +/- 0.09	1.55 +/- 0.01	21.8%	4.18 +/- 0.37	2.60 +/- 0.16	37.9%
	8	3.20 +/- 0.06	2.81 +/- 0.07	12.2%	6.30 +/- 0.10	5.10 +/- 0.29	19.1%
	16	1.91 +/- 0.04	1.60 +/- 0.02	16.0%	4.19 +/- 0.16	2.71 +/- 0.12	35.4%
	Avg.	1.33 +/- 0.01	1.13 +/- 0.01	14.5%	1.65 +/- 0.01	1.20 +/- 0.01	26.9%
Solar	1	1.67 +/- 0.02	1.48 +/- 0.01	11.3%	2.23 +/- 0.04	1.73 +/- 0.01	22.4%
	2	1.47 +/- 0.01	1.12 +/- 0.00	24.1%	1.86 +/- 0.01	1.54 +/- 0.02	17.0%
	4	2.39 +/- 0.04	1.98 +/- 0.03	17.1%	2.43 +/- 0.04	2.14 +/- 0.03	12.0%
	8	2.91 +/- 0.11	2.45 +/- 0.06	15.9%	3.35 +/- 0.13	2.68 +/- 0.07	20.1%
	16	1.95 +/- 0.04	1.63 +/- 0.02	16.4%	2.30 +/- 0.05	1.86 +/- 0.03	19.3%

Table F.9: 2 tasks: Average MAE and standard deviation for different methods with Horizontal and Hybrid privacy constraints. We show the improvement of introducing the personalization mechanism.

Dataset	PW	FedAvg	TPHFL-H	Imp.	TPHFL-NP	TPHFL	Imp.
AirQuality	1	2.24 +/- 0.06	1.84 +/- 0.00	17.7%	2.76 +/- 0.12	2.09 +/- 0.00	24.4%
	2	1.95 +/- 0.01	1.73 +/- 0.00	11.3%	2.32 +/- 0.01	1.93 +/- 0.00	16.9%
	4	2.00 +/- 0.00	1.92 +/- 0.00	3.9%	2.15 +/- 0.01	1.97 +/- 0.00	8.7%
	8	3.96 +/- 0.01	3.61 +/- 0.00	8.9%	4.26 +/- 0.01	3.81 +/- 0.00	10.6%
	16	5.59 +/- 0.01	5.22 +/- 0.01	6.6%	5.62 +/- 0.04	4.78 +/- 0.01	15.0%
	Avg.	3.15 +/- 0.02	2.87 +/- 0.00	9.0%	3.42 +/- 0.04	2.91 +/- 0.00	14.9%
Industry	1	2.30 +/- 0.05	1.78 +/- 0.00	22.6%	3.72 +/- 0.22	2.59 +/- 0.09	30.4%
	2	2.75 +/- 0.04	1.93 +/- 0.03	29.7%	3.68 +/- 0.22	2.76 +/- 0.04	24.9%
	4	1.95 +/- 0.04	1.50 +/- 0.01	23.3%	2.88 +/- 0.06	1.96 +/- 0.01	31.8%
	8	3.16 +/- 0.37	1.99 +/- 0.08	37.1%	4.77 +/- 0.64	2.69 +/- 0.16	43.5%
	16	2.88 +/- 0.09	2.14 +/- 0.05	25.8%	4.52 +/- 0.70	3.17 +/- 0.25	29.8%
	Avg.	2.61 +/- 0.12	1.87 +/- 0.04	28.4%	3.91 +/- 0.37	2.64 +/- 0.11	32.6%
Sales	1	2.52 +/- 0.01	2.10 +/- 0.01	16.4%	6.18 +/- 0.04	2.44 +/- 0.01	60.5%
	2	2.87 +/- 0.01	2.35 +/- 0.01	18.0%	6.54 +/- 0.05	2.58 +/- 0.01	60.5%
	4	3.48 +/- 0.00	2.92 +/- 0.01	16.1%	7.30 +/- 0.03	3.64 +/- 0.01	50.2%
	8	4.19 +/- 0.01	3.51 +/- 0.00	16.4%	8.03 +/- 0.02	5.71 +/- 0.00	28.8%
	16	3.03 +/- 0.00	2.47 +/- 0.01	18.5%	6.78 +/- 0.05	4.46 +/- 0.02	34.2%
	Avg.	3.22 +/- 0.01	2.67 +/- 0.01	17.0%	6.96 +/- 0.04	3.77 +/- 0.01	45.9%
Crypto	1	1.48 +/- 0.01	1.26 +/- 0.01	14.7%	2.80 +/- 0.14	2.20 +/- 0.04	21.4%
	2	1.44 +/- 0.00	1.30 +/- 0.01	9.9%	2.56 +/- 0.06	1.69 +/- 0.01	33.9%
	4	1.44 +/- 0.01	1.34 +/- 0.01	6.8%	3.08 +/- 0.06	1.95 +/- 0.03	36.7%
	8	1.97 +/- 0.01	1.65 +/- 0.01	16.4%	2.92 +/- 0.07	2.60 +/- 0.06	10.9%
	16	2.56 +/- 0.01	2.32 +/- 0.02	9.3%	4.54 +/- 0.12	3.77 +/- 0.09	17.0%
	Avg.	1.78 +/- 0.01	1.57 +/- 0.01	11.5%	3.18 +/- 0.09	2.44 +/- 0.05	23.2%
Solar	1	1.23 +/- 0.01	0.78 +/- 0.01	36.8%	1.74 +/- 0.04	0.91 +/- 0.02	47.5%
	2	1.78 +/- 0.05	1.06 +/- 0.02	40.6%	2.13 +/- 0.06	1.25 +/- 0.03	41.2%
	4	1.30 +/- 0.01	0.93 +/- 0.01	28.9%	1.99 +/- 0.01	1.16 +/- 0.02	41.5%
	8	2.64 +/- 0.13	1.57 +/- 0.05	40.6%	3.06 +/- 0.13	1.60 +/- 0.04	47.6%
	16	3.26 +/- 0.12	2.01 +/- 0.06	38.4%	3.84 +/- 0.16	2.07 +/- 0.08	46.2%
	Avg.	2.04 +/- 0.06	1.27 +/- 0.03	37.9%	2.55 +/- 0.08	1.40 +/- 0.04	45.1%

Table F.10: 4 tasks: Average MAE and standard deviation for different methods with Horizontal and Hybrid privacy constraints. We show the improvement of introducing the personalization mechanism.

Dataset	PW	Independent	TPHFL-l2	Imp.	Independent	TPHFL	Imp.
AirQuality	1	2.13 +/- 0.03	1.99 +/- 0.00	6.5%	2.13 +/- 0.03	2.09 +/- 0.00	1.8%
	2	2.17 +/- 0.00	2.02 +/- 0.00	7.0%	2.17 +/- 0.00	2.11 +/- 0.00	2.9%
	4	2.16 +/- 0.00	2.19 +/- 0.00	-1.8%	2.16 +/- 0.00	2.21 +/- 0.00	-2.5%
	8	4.57 +/- 0.02	3.77 +/- 0.01	17.4%	4.57 +/- 0.02	3.78 +/- 0.00	17.2%
	16	6.21 +/- 0.01	5.26 +/- 0.00	15.3%	6.21 +/- 0.01	5.37 +/- 0.00	13.6%
	Avg.	3.45 +/- 0.01	3.05 +/- 0.00	11.6%	3.45 +/- 0.01	3.11 +/- 0.00	9.7%
Industry	1	6.65 +/- 0.33	4.21 +/- 0.15	36.7%	6.65 +/- 0.33	4.25 +/- 0.13	36.1%
	2	8.17 +/- 0.46	2.92 +/- 0.10	64.3%	8.17 +/- 0.46	2.36 +/- 0.02	71.2%
	4	2.25 +/- 0.10	1.74 +/- 0.02	22.6%	2.25 +/- 0.10	1.74 +/- 0.02	22.6%
	8	3.52 +/- 0.38	3.37 +/- 0.29	4.2%	3.52 +/- 0.38	3.54 +/- 0.36	-0.5%
	16	7.19 +/- 0.28	4.64 +/- 0.03	35.4%	7.19 +/- 0.28	4.91 +/- 0.06	31.7%
	Avg.	5.56 +/- 0.31	3.38 +/- 0.12	39.2%	5.56 +/- 0.31	3.36 +/- 0.12	39.5%
Sales	1	2.58 +/- 0.01	2.76 +/- 0.01	-7.0%	2.58 +/- 0.01	2.77 +/- 0.02	-7.6%
	2	2.66 +/- 0.01	2.89 +/- 0.03	-8.6%	2.66 +/- 0.01	2.97 +/- 0.03	-11.4%
	4	3.32 +/- 0.00	3.37 +/- 0.02	-1.5%	3.32 +/- 0.00	3.44 +/- 0.02	-3.8%
	8	3.88 +/- 0.00	5.09 +/- 0.00	-31.0%	3.88 +/- 0.00	4.78 +/- 0.00	-23.0%
	16	2.98 +/- 0.00	4.99 +/- 0.05	-67.3%	2.98 +/- 0.00	4.52 +/- 0.05	-51.6%
	Avg.	3.09 +/- 0.00	3.82 +/- 0.02	-23.8%	3.09 +/- 0.00	3.70 +/- 0.02	-19.8%
Crypto	1	3.36 +/- 0.38	2.28 +/- 0.08	32.1%	3.36 +/- 0.38	2.19 +/- 0.09	34.9%
	2	1.96 +/- 0.12	2.07 +/- 0.06	-5.6%	1.96 +/- 0.12	2.03 +/- 0.06	-3.2%
	4	2.24 +/- 0.07	1.71 +/- 0.01	23.4%	2.24 +/- 0.07	1.63 +/- 0.02	27.3%
	8	3.31 +/- 0.24	2.64 +/- 0.15	20.2%	3.31 +/- 0.24	2.60 +/- 0.16	21.5%
	16	4.90 +/- 0.26	5.21 +/- 0.25	-6.3%	4.90 +/- 0.26	5.10 +/- 0.29	-4.1%
	Avg.	3.15 +/- 0.21	2.78 +/- 0.11	11.7%	3.15 +/- 0.21	2.71 +/- 0.12	14.2%
Solar	1	1.33 +/- 0.01	1.22 +/- 0.01	8.1%	1.33 +/- 0.01	1.20 +/- 0.01	9.3%
	2	1.64 +/- 0.01	1.74 +/- 0.01	-6.4%	1.64 +/- 0.01	1.73 +/- 0.01	-5.5%
	4	1.58 +/- 0.00	1.57 +/- 0.02	0.9%	1.58 +/- 0.00	1.54 +/- 0.02	2.5%
	8	2.19 +/- 0.03	2.20 +/- 0.03	-0.2%	2.19 +/- 0.03	2.14 +/- 0.03	2.4%
	16	3.24 +/- 0.14	2.70 +/- 0.07	16.7%	3.24 +/- 0.14	2.68 +/- 0.07	17.4%
	Avg.	2.00 +/- 0.04	1.89 +/- 0.03	5.6%	2.00 +/- 0.04	1.86 +/- 0.03	6.9%

Table F.11: 2 tasks: Average MAE and standard deviation for Independent, TPHFL-l2 and TPHFL. We show the improvement over Independent.

Dataset	PW	Independent	TPHFL-I2	Imp.	Independent	TPHFL	Imp.
AirQuality	1	2.36 +/- 0.05	2.04 +/- 0.00	13.5%	2.36 +/- 0.05	2.09 +/- 0.00	11.7%
	2	2.25 +/- 0.01	1.93 +/- 0.00	14.3%	2.25 +/- 0.01	1.93 +/- 0.00	14.0%
	4	2.18 +/- 0.00	1.97 +/- 0.00	9.8%	2.18 +/- 0.00	1.97 +/- 0.00	9.9%
	8	4.49 +/- 0.02	3.72 +/- 0.01	17.2%	4.49 +/- 0.02	3.81 +/- 0.00	15.3%
	16	6.36 +/- 0.10	4.79 +/- 0.00	24.6%	6.36 +/- 0.10	4.78 +/- 0.01	24.7%
	Avg.	3.53 +/- 0.04	2.89 +/- 0.00	18.1%	3.53 +/- 0.04	2.91 +/- 0.00	17.4%
Industry	1	4.56 +/- 0.64	2.76 +/- 0.12	39.3%	4.56 +/- 0.64	2.59 +/- 0.09	43.1%
	2	5.91 +/- 0.77	2.91 +/- 0.05	50.7%	5.91 +/- 0.77	2.76 +/- 0.04	53.2%
	4	2.61 +/- 0.13	1.90 +/- 0.01	27.3%	2.61 +/- 0.13	1.96 +/- 0.01	24.9%
	8	3.04 +/- 0.28	2.60 +/- 0.17	14.5%	3.04 +/- 0.28	2.69 +/- 0.16	11.5%
	16	5.08 +/- 0.86	3.16 +/- 0.21	37.7%	5.08 +/- 0.86	3.17 +/- 0.25	37.6%
	Avg.	4.24 +/- 0.54	2.67 +/- 0.11	37.1%	4.24 +/- 0.54	2.64 +/- 0.11	37.8%
Sales	1	2.61 +/- 0.01	2.64 +/- 0.01	-0.9%	2.61 +/- 0.01	2.44 +/- 0.01	6.7%
	2	2.79 +/- 0.01	2.73 +/- 0.02	2.4%	2.79 +/- 0.01	2.58 +/- 0.01	7.5%
	4	3.42 +/- 0.00	3.89 +/- 0.02	-13.6%	3.42 +/- 0.00	3.64 +/- 0.01	-6.2%
	8	3.93 +/- 0.01	6.57 +/- 0.01	-67.1%	3.93 +/- 0.01	5.71 +/- 0.00	-45.2%
	16	3.13 +/- 0.01	6.21 +/- 0.01	-98.1%	3.13 +/- 0.01	4.46 +/- 0.02	-42.4%
	Avg.	3.18 +/- 0.01	4.41 +/- 0.01	-38.6%	3.18 +/- 0.01	3.77 +/- 0.01	-18.4%
Crypto	1	2.48 +/- 0.27	2.23 +/- 0.04	10.1%	2.48 +/- 0.27	2.20 +/- 0.04	11.1%
	2	1.75 +/- 0.07	1.73 +/- 0.01	1.1%	1.75 +/- 0.07	1.69 +/- 0.01	3.0%
	4	1.96 +/- 0.05	2.14 +/- 0.04	-9.1%	1.96 +/- 0.05	1.95 +/- 0.03	0.6%
	8	2.74 +/- 0.16	2.63 +/- 0.06	4.1%	2.74 +/- 0.16	2.60 +/- 0.06	5.0%
	16	3.82 +/- 0.25	3.76 +/- 0.10	1.8%	3.82 +/- 0.25	3.77 +/- 0.09	1.4%
	Avg.	2.55 +/- 0.16	2.50 +/- 0.05	2.1%	2.55 +/- 0.16	2.44 +/- 0.05	4.2%
Solar	1	1.38 +/- 0.02	0.96 +/- 0.02	30.3%	1.38 +/- 0.02	0.91 +/- 0.02	33.6%
	2	1.88 +/- 0.06	1.28 +/- 0.03	31.5%	1.88 +/- 0.06	1.25 +/- 0.03	33.2%
	4	1.78 +/- 0.02	1.19 +/- 0.02	33.4%	1.78 +/- 0.02	1.16 +/- 0.02	34.8%
	8	2.39 +/- 0.07	1.62 +/- 0.04	32.3%	2.39 +/- 0.07	1.60 +/- 0.04	32.8%
	16	3.30 +/- 0.13	2.13 +/- 0.07	35.5%	3.30 +/- 0.13	2.07 +/- 0.08	37.4%
	Avg.	2.14 +/- 0.06	1.44 +/- 0.04	33.1%	2.14 +/- 0.06	1.40 +/- 0.04	34.7%

Table F.12: 4 tasks: Average MAE and standard deviation for Independent, TPHFL-I2 and TPHFL. We show the improvement over Independent.