Federated Incomplete Tabular Data Prediction with Missing Complementarity

Yan Zhang*[1Δ], Shuwei Liang*[2Δ], Xiaoye Miao*[3+], Yangyang Wu*[4], Jianwei Yin†[5]
*Center for Data Science, Zhejiang University, Hangzhou, China
†College of Computer Science, Zhejiang University, Hangzhou, China
{nameyzhang[1], lsw5221[2], miaoxy[3], zjuwuyy[4]}@zju.edu.cn, zjuyjw[5]@cs.zju.edu.cn

ABSTRACT

Tabular data is abundant and crucial across both industry and academia. Federated learning (FL) offers a promising solution for the analysis of tabular data distributed across multiple organizations, without the need to share the privacy information of each client. Existing federated tabular data prediction methods optimize performance and privacy leakage under the completeness assumption of tabular data. They are not applicable in real-world scenarios that are struggling with missing values in tabular data. In this paper, we propose a novel federated prediction framework for incomplete tabular data, named DARN, which leverages the *missing complementarity* to directly optimize prediction performance without relying on the imputed values. It is especially beneficial when clients exhibit heterogeneity in missing data distributions, and the pairwise observed data are complementary. Specifically, each client trains a *missing distribution learning model* to capture the distribution of locally incomplete data. To assist in this, we present a *missing-aware transformer block* with a novel missing-aware attention mechanism to represent incomplete tabular data directly. The server calculates the personalized weights of the prediction models by combining *missing complementary score* and *observed sample size score*, thereby maximizing the utility of the available data. Extensive experiments on four publicly available real-world datasets demonstrate that DARN outperforms state-of-the-art methods with 25.80% improvement in both classification and regression tasks.

PVLDB Reference Format:
Yan Zhang, Shuwei Liang, Xiaoye Miao, Yangyang Wu, Jianwei Yin. Federated Incomplete Tabular Data Prediction with Missing Complementarity. PVLDB, 18(10): 3531 - 3544, 2025. doi:10.14778/3748191.3748213

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/LS5221/DARN.

1 INTRODUCTION

Tabular data, which constitutes over 70% of global data and is commonly stored in databases or spreadsheets [21], is abundant and crucial across both industries and academia. It is often distributed across multiple organizations, making centralized data sharing impractical due to privacy and regulatory concerns. Federated learning (FL) [17, 49] collaboratively trains a global model on distributed data without the need for raw data sharing [1, 40]. It allows organizations to utilize their tabular data while ensuring privacy and compliance. Consequently, FL is particularly beneficial for various prediction tasks related to tabular data, including healthcare [41], urban computing [6, 28], recommender systems [22], and so forth.

Federated prediction task involves the collaborative training of predictive models across distributed clients, enabling predictions to be made without directly sharing private data. This approach offers a scientifically sound and practical foundation for decision-making while preserving privacy. Existing federated tabular data prediction methods can be categorized into two branches. The first branch consists of machine learning-based methods, such as gradient boosting decision tree (GBDT) [16, 29, 30, 58], random forests [32, 44, 48], and XGBoost [48, 51]. The second branch is deep learning-based methods, where the representatives include generative adversarial networks (GAN) [12, 57] and contrastive learning [19]. All of these prediction methods achieve high performance and minimal privacy leakage under the assumption that the tabular data possessed by each client is *complete*.

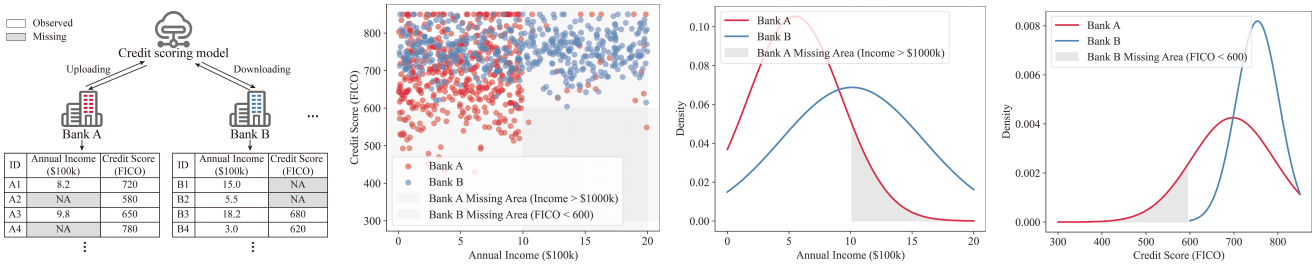
However, in real-world scenarios, *incomplete* tabular data is ubiquitous due to various factors, such as human error during data processing, machine malfunctions, respondents' refusal to answer certain questions, and privacy constraints [36]. As a result, the presence of missing values in tabular data hinders researchers from conducting comprehensive analyses. Thus, it is critical and challenging to propose an effective federated prediction framework on incomplete tabular data.

Example 1: Figure 1(a) shows a toy example of a financial scenario. Due to limited data and the need for privacy protection, all banks aim to collaboratively train a federated credit scoring model using incomplete tabular data from each institution [15, 18]. There are two banks: Bank *A*, which specializes in serving freelancers (*e.g.,* Livi Bank), and Bank *B*, which focuses on new immigrants (*e.g.,* Chime Bank) [27]. High-income freelancers at Bank *A* exhibit systematic missing data due to privacy concerns and income volatility; however, their credit scores are complete. Conversely, new immigrants at Bank *B* lack local credit history, resulting in missing low credit scores despite complete income profiles. As a result, Bank *A* exhibits systematic gaps in high-income records (> \$1000k), while Bank *B* lacks low credit scores (FICO[1] < 600).

Δ Equal Contribution. + Corresponding authors.

Proceedings of the VLDB Endowment, Vol. 18, No. 10 ISSN 2150-8097.
doi:10.14778/3748191.3748213

[1]The FICO score is widely used in credit scoring systems and serves as a key criterion for the approval of loans and credit cards. The higher the score, the lower the risk and the better the credit, and vice versa.



(a) FL with incomplete tabular data (b) Annual income vs. credit score (c) Annual income distribution (d) Credit scores distribution

Figure 1: Illustration of missing data complementarity in federated learning with incomplete tabular data prediction.

There is a group of related studies that impute incomplete data by aggregating all users' data for centralized learning. The core idea behind these centralized imputation methods is to replace missing values with estimates generated through various techniques, including statistical methods [3, 13], machine learning ones [9, 56], and deep learning ones [34]. However, they cannot be directly applied in a federated learning scenario for two major reasons. *First*, they require access to data from all users to estimate the missing values, which introduces privacy and security risks, as user data is often highly sensitive. *The second reason* is that, these methods basically operate under the ideal assumptions of missing completely at random (MCAR) [50] and missing at random (MAR) [23]. In other words, they do imputations that only rely on observed data. They fail to effectively deal with the more complex mechanism of missing not at random (MNAR) [31], where the probability of missing data is related to the value itself, and the pattern of missingness cannot be fully captured by other observed variables.

A *step-by-step* approach is typically followed for federated incomplete tabular prediction, in which missing values are imputed, and the estimated values are subsequently used to perform various federated prediction tasks. However, this approach has three limitations. *First*, these federated imputation methods for tabular data [33, 60] are not capable of handling more complex missing data mechanisms, *i.e.,* MNAR. *Second*, it overlooks valuable information that could be derived from downstream tasks, such as task-specific features or feedback that could refine the imputation process [8, 11]. As a result, this may lack targeted adjustments, thereby impacting the accuracy and effectiveness of the task. *Third*, errors incurred during the imputation phase can propagate and amplify during the prediction stage, further deteriorating overall performance. This cumulative error effect reduces the reliability and accuracy of model predictions. Overall, the step-by-step approach for federated prediction models with incomplete tabular data is suboptimal.

Inspired by previous works [37, 52, 59], this paper aims to train federated prediction models over incomplete tabular data directly without relying on estimated values. However, the following two challenges need to be addressed.

Challenge 1: How can the available data be maximally utilized to construct prediction models without introducing estimation error? In the FL architecture, each participant presents an individual entity. The missing data patterns are heterogeneous across clients due to differences in data collection methods, environmental factors, user behaviors, and so on. In the aforementioned toy example, Bank *A* lacks records of high-income individuals but has complete credit scores, while Bank *B* faces the opposite issue. This creates an opportunity for *missing complementarity*. It implies that for specific features, the missing parts of local data can be recovered by the corresponding parts of other clients. As shown in Figure 1(b), the complete global knowledge distribution can be jointly modeled by the data of Bank *A* and Bank *B*. Specifically, as shown in Figure 1(c), Bank *B*'s observed annual income data distribution can potentially assist in imputing *A*'s missing annual income data. Similarly, as shown in Figure 1(d), Bank *B*'s missing low-score segment is filled by Bank *A*'s comprehensive data spanning FICO. The missing data mechanism is categorized as MNAR because the probability of missingness is directly related to the unobserved data values (*e.g.,* high income or low credit score). It motivates us to construct personalized prediction models by leveraging missing complementarity to improve model performance.

Challenge 2: How can incomplete tabular data be represented to capture the missing data distribution accurately? Tabular data representation is the process of converting structured data into a format that machine learning models can effectively use to capture patterns and relationships. The quality of this representation directly influences task processing, model training, and final prediction performance. However, existing studies on tabular data representation primarily focus on full-knowledge data analysis problems [2, 55]. Moreover, some encoding techniques for incomplete tabular data require supplementary imputation algorithms, which may introduce bias [20]. Thus, it is crucial to represent missing tabular data in a way that accurately captures the missing data distribution without the need for additional imputation operations.

In this paper, we propose a novel framework, named federated incomplete tabular data prediction with missing complementarity (DARN). It enables personalized federated prediction by leveraging the complementarity of missing data distributions, without relying on imputed values. Specifically, to address the first challenge, each client maintains a *missing distribution learning model* to learn the distribution of local incomplete data. We also introduce a missing complementarity score calculated based on the dissimilarity between different pairs of missing distributions. By combining the missing complementarity score with the observed sample size score, the personalized weight for the prediction model is derived, maximizing the utilization of the observed data. To address the second challenge, we propose a missing-aware transformer block that incorporates a missing-aware attention mechanism. This block is shared between the prediction and imputation models, allowing it to directly represent incomplete data while capturing the

missing pattern without the need for imputation. The shared block ensures that there is no additional computation or communication overhead for the model. The representation is further employed in the prediction model training and missing distribution learning to enhance overall performance. In addition, we analyze DARN as a privacy-preserving framework, which can be enhanced through the application of differential privacy [47]. In summary, the key contributions of this paper are as follows:

- We propose a novel federated incomplete tabular data prediction framework, called DARN, which trains personalized prediction models directly on missing data without relying on the estimated values. It leverages the complementarity of missing data to maximize the utility of the available information.
- We introduce the missing complementarity score, which is calculated based on the dissimilarity between the missing distributions. We combine it with the observed sample size score to calculate the personalized weight for optimizing the prediction model.
- We present a missing-aware transformer block incorporating a missing-aware attention mechanism to directly represent incomplete tabular data and capture complex missing patterns for enhanced utility.
- Extensive experiments on four publicly available datasets in six scenarios demonstrate the superiority of DARN over state-of-the-art methods in both classification and regression tasks. We also evaluate the effectiveness of DARN using two real-world incomplete tabular datasets.

The remainder of this paper is organized as follows. In Section 2, we review the related work. In Section 3, two key concepts and the problem statement are described. The proposed framework DARN is elaborated in Section 4. The experimental results are reported in Section 5. Finally, we conclude the paper in Section 6.

2 RELATED WORK

In this section, we provide an overview of the related studies on tabular data prediction and incomplete data imputation under the federated environment, respectively.

Federated tabular data prediction. Tabular data is a typical structure that organizes and stores information in a tabular form, commonly used in various fields [22, 41], *e.g.,* finance, healthcare, and recommender systems. Existing federated tabular data prediction methods contain machine learning-based and deep learning-based ones. The machine learning-based methods include gradient boosting decision tree (GBDT) [16, 29, 30, 58], random forests [32, 44, 48], and XGBoost [48, 51]. For example, the study in [48] proposes a novel solution for privacy-preserving vertical decision tree training and prediction, ensuring that no intermediate information is disclosed beyond what the clients have agreed to release. Furthermore, a significant number of deep learning-based prediction models for federated tabular data have also been developed. For example, some studies [12, 57] leverage GANs to generate synthetic tabular data, which in turn helps build more effective downstream global models. Contrastive learning is used to create more common feature representations across different data silos [19]. However, the above two types of studies only consider the complete data scenarios. It limits their applicability in the real world, which is struggling with ubiquitous incomplete tabular data.

Federated incomplete data imputation. It focuses on effectively imputing missing data through collaborative learning while preserving data privacy. Existing federated missing data imputation methods can be classified into three branches: GAN-based methods, expectation maximization (EM)-based methods, and multiple imputation (MI)-based methods. In particular, the GAN-based methods [33, 60] leverage GAN to capture complex data distributions and generate synthetic data that closely resembles real-world data. These synthetic datasets are then used to enhance the robustness of models across clients. The EM-based methods [11] iteratively perform expectation steps (to estimate missing values) and maximization steps (to optimize model parameters). It progressively predicts missing values in distributed data. The MI-based methods [8] use statistical models to impute missing data based on observational data and then transmit statistical summaries generated from the imputed data for global modeling and inference. However, all of the aforementioned methods impute missing values in federated learning under the MAR or MCAR assumptions, which rely on observed data. Moreover, if using the federated imputation as a prior step of federated prediction tasks over missing tabular data, the accuracy of downstream prediction models is highly dependent on the accuracy of estimated values. Thus, existing federated imputation methods neither address more complex missing data mechanisms, *i.e.,* MNAR, nor effectively help federated prediction over incomplete tabular data.

3 PRELIMINARIES

In this section, we present two key concepts: incomplete tabular data and standard federated learning. Then, we describe the problem we studied in this paper. Table 1 summarizes the frequently used notations throughout the paper.

Incomplete tabular data. Incomplete tabular data, also known as missing tabular data, can be categorized into the following three types based on the mechanism and cause of the missing data [42]: (i) missing completely at random (MCAR) [50], where missing values are unrelated to both observed and missing components; (ii) missing at random (MAR) [23], in which the likelihood of missing data depends on other observed variables but remains unrelated to the missing values; and (iii) missing not at random (MNAR) [31], indicating that the missing data is only related to the missing value themselves. For example, as shown in Figure 1, higher-income participants often withhold their income, resulting in missing values that are directly related to the measured variable. Among the three missing data mechanisms, MNAR is the most complex and representative. Therefore, this paper primarily focuses on analyzing incomplete tabular data under the MNAR mechanism.

Let us consider an incomplete tabular dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i = (x_{i1}, \cdots, x_{id})$ represents a sample with $x_{ij} \in \mathcal{X}_j$ taking values from a d-dimensional space, and $y_i \in \mathbb{Y}$ denotes the corresponding label. We focus on cases where feature values are missing, but the label is complete, *i.e.,* scenarios in which one or more feature values x_{ij} are not observed for certain samples $\mathbf{x}_i$. To systematically encode the missing data within D, we introduce a mask matrix $\mathbf{M}$, defined as $\mathbf{M} = (\mathbf{m}_1, \cdots, \mathbf{m}_N)^\top$. Each mask vector $\mathbf{m}_i = (m_{i1}, \cdots, m_{id})$ corresponds to a sample $\mathbf{x}_i$ and indicates the presence or absence of its feature values. The elements of $\mathbf{M}$ are

Table 1: Notation description.

Notation	Description
K	the number of clients in the federated learning system
N	the total number of samples from all K clients
T	the total rounds of collaborative communication
E	the number of local training epochs
λ	the noise strength of the Laplace distribution
D_k	the local private dataset of the k-th client
O_k	the total number of observed features across all samples of the k-th client
C_i	the observed sample size score of the i-th client
S_{ij}	the pair complementary score of the i-th and j-th client
$\mathbf{X}, \hat{\mathbf{X}}, \mathbf{X}'$	the incomplete, reconstructed and masked attribute matrix, *i.e.,* $\mathbf{X}, \hat{\mathbf{X}}, \mathbf{X}' \in \mathbb{R}^{n\times d}$
$\mathbf{Y}, \hat{\mathbf{Y}}$	the true and predicted label matrix, *i.e.,* $\mathbf{Y}, \hat{\mathbf{Y}} \in \mathbb{Y}^{n\times 1}$
$\mathbf{M}, \hat{\mathbf{M}}$	the mask matrix and transformed mask matrix of $\mathbf{X}$, *i.e.,* $\mathbf{M}, \hat{\mathbf{M}} \in \mathbb{R}^{n\times d}$
$\mathbf{M}'$	the predicted missing probability matrix, *i.e.,* $\mathbf{M}' \in \mathbb{R}^{n\times d}$
$\mathbf{R}$	the random mask matrix, *i.e.,* $\mathbf{R} \in \{0,1\}^{n\times d}$
$\mathbf{A}$	the attention score matrix
$\mathbf{H}$	the high-dimensional representation vector, *i.e.,* $\mathbf{H} \in \mathbb{R}^{d\times e}$
$\mathcal{P}, \theta^P$	the prediction model and its parameters
$\mathcal{I}, \theta^I$	the imputation model and its parameters
$\mathcal{M}, \theta^M$	the missing distribution learning model and its parameters

defined as follows:

$$m_{ij} = \begin{cases} 1, & \text{if } x_{ij} \text{ is observed,} \\ 0, & \text{if } x_{ij} \text{ is missing,} \end{cases} \tag{1}$$

where $i \in \{1, \cdots, N\}$ and $j \in \{1, \cdots, d\}$. The mask matrix $\mathbf{M}$ functions as an indicator for the availability of each feature across all samples in D.

Standard federated learning. In this paper, we consider a typical horizontally partitioned federated learning setup, involving a central server and a total of K local clients. The goal is to collaboratively train a high-performance FL global model parameterized by θ^* under server cooperation without sharing their local dataset. For each client $k \in [K]$ holds a private complete dataset $D_k = \{(\mathbf{x}_i^k, y_i^k) | i = 1, 2, \cdots, n_k\}$, where $\mathbf{x}_i^k$ and y_i^k denote the i-th input and label of k-th client, and $[K]$ represents a collection of client indices. $N = \sum_{k=1}^{K} n_k$ is the total number of samples from all clients. Assume that all participants are honest but curious, meaning they adhere to the protocol but attempt to learn information from the received messages. Mathematically, the goal objective of standard FL is to minimize the loss of all K clients as follows:

$$\theta^* = \arg\min_{\theta} \sum_{k=1}^{K} w_k \mathcal{L}(\theta_k),$$

where $\mathcal{L}(\theta_k) = \mathbb{E}_{(\mathbf{x},y)\in D_k} \ell(\mathbf{x}, y; \theta_k)$ is the empirical loss of k-th client, $\ell(\cdot;\cdot)$ is the supervised loss for client tasks, and w_k is the weight for the k-th client's loss such that $w_k \geq 0$ and $\sum_{k=1}^{K} w_k = 1$.

Problem definition. Based on the above concepts, we assume the existence of a central server and K clients in this paper. Each client has an incomplete tabular dataset $D_k = (\mathbf{X}_k, \mathbf{Y}_k)$, where $\mathbf{X}_k \in \mathbb{R}^{n_k\times d}$ is incomplete attribute matrix, $\mathbf{Y}_k \in \mathbb{Y}^{n_k\times 1}$ is label matrix. A mask matrix $\mathbf{M}_k$ of $\mathbf{X}_k$ represents the state of missing features, where $\mathbf{M}_k \in \mathbb{R}^{n_k\times d}$ is same as Eq. 1. Each client has n_k samples and the same feature spaces. Therefore, the total number of all clients' samples is $N = \sum_{k=1}^{K} n_k$. We consider two types of prediction tasks including classification $\mathbb{Y} \in \{1, \cdots, C\}$ and regression $\mathbb{Y} \in \mathbb{R}$.

The problem objective is to obtain an optimal set of local prediction model parameters $\mathcal{P} = \{\theta_1^P, \theta_2^P, \cdots, \theta_K^P\}$, imputation model parameters $\mathcal{I} = \{\theta_1^I, \theta_2^I, \cdots, \theta_K^I\}$ and missing distribution model parameters $\mathcal{M} = \{\theta_1^M, \theta_2^M, \cdots, \theta_K^M\}$ that minimizes the supervised loss, reconstruction loss and missing distribution learning loss across all K clients, *i.e.,*

$$\arg\min_{\theta} \sum_{k=1}^{K} w_k \left(\mathcal{L}_{\mathsf{sup}}(\theta_k^P) + \alpha \mathcal{L}_{\mathsf{rec}}(\theta_k^I) + \mathcal{L}_{\mathsf{prob}}(\theta_k^M) \right), \tag{2}$$

where $\mathcal{L}_{\mathsf{sup}}(\theta_k^P) = \mathbb{E}_{(\mathbf{x},y)\in D_k, \mathbf{m}\in\mathbf{M}_k} \ell(\mathcal{P}(\mathbf{x}, \mathbf{m}|\theta_k^P), y)$ is the supervised loss, $\mathcal{L}_{\mathsf{rec}}(\theta_k^I) = \mathbb{E}_{(\mathbf{x},y)\in D_k, \mathbf{x}'\in\mathbf{X}'_k, \hat{\mathbf{m}}\in\hat{\mathbf{M}}_k} \ell(\mathcal{I}(\mathbf{x}', \hat{\mathbf{m}}|\theta_k^I), \mathbf{x})$ is the reconstruction loss, and $\mathcal{L}_{\mathsf{prob}}(\theta_k^M) = \mathbb{E}_{\hat{\mathbf{x}}\in\hat{\mathbf{X}}_k, \mathbf{m}\in\mathbf{M}_k} \ell(\mathcal{M}(\hat{\mathbf{x}}|\theta_k^M), \mathbf{m})$ is the missing distribution learning loss. $\mathbf{X}'$ and $\hat{\mathbf{X}}$ are masked and reconstructed attribute matrix of $\mathbf{X}$, repectively. $\hat{\mathbf{M}}$ is the transformed mask matrix of $\mathbf{M}$, α is a hyperparameter of weight, and w_k is the weight of client k in the global optimization process.

4 DARN FRAMEWORK

In this section, we introduce our proposed federated incomplete tabular data prediction model, DARN, which enables privacy-preserving prediction across multiple clients using incomplete data directly. This model fully leverages the information from each client's missing data distribution, maximizing the utility of available data.

4.1 Framework Overview

The overall framework of DARN is shown in Figure 2. The framework enables personalized federated prediction by leveraging the complementarity of heterogeneous missing data distributions, thereby eliminating errors introduced by imputation.

It primarily comprises two core components: (i) *missing distribution learning for prediction model* in the local training phase ②, and (ii) *personalized weight averaging* in the server aggregation phase ④. Each client employs a shared missing-aware transformer block to encode incomplete and masked incomplete data into high-dimensional embeddings, respectively. Subsequently, distinct multilayer perceptron (MLP) networks are utilized to generate predicted labels and reconstructed data. The reconstructed data is then fed into the missing distribution learning model, which is equipped with a logistic regression network to estimate the missing distribution. Lastly, the missing distribution and the observed sample size for each client are transmitted to the server. After receiving the information from clients, the server calculates personalized weights for each client's prediction model based on the complementarity of the missing distribution and the amount of observed data. Personalized averaging is then performed using these weights. The optimal personalized prediction models are derived after T collaborative communication rounds or when a convergence criterion is satisfied.

Overall, this framework learns the distribution of missing data and leverages the complementarity of missing data during model training, fully utilizing the available data from each client. It, in turn, enhances the collaborative training effect in federated learning. As a result, DARN accelerates model convergence and boosts the

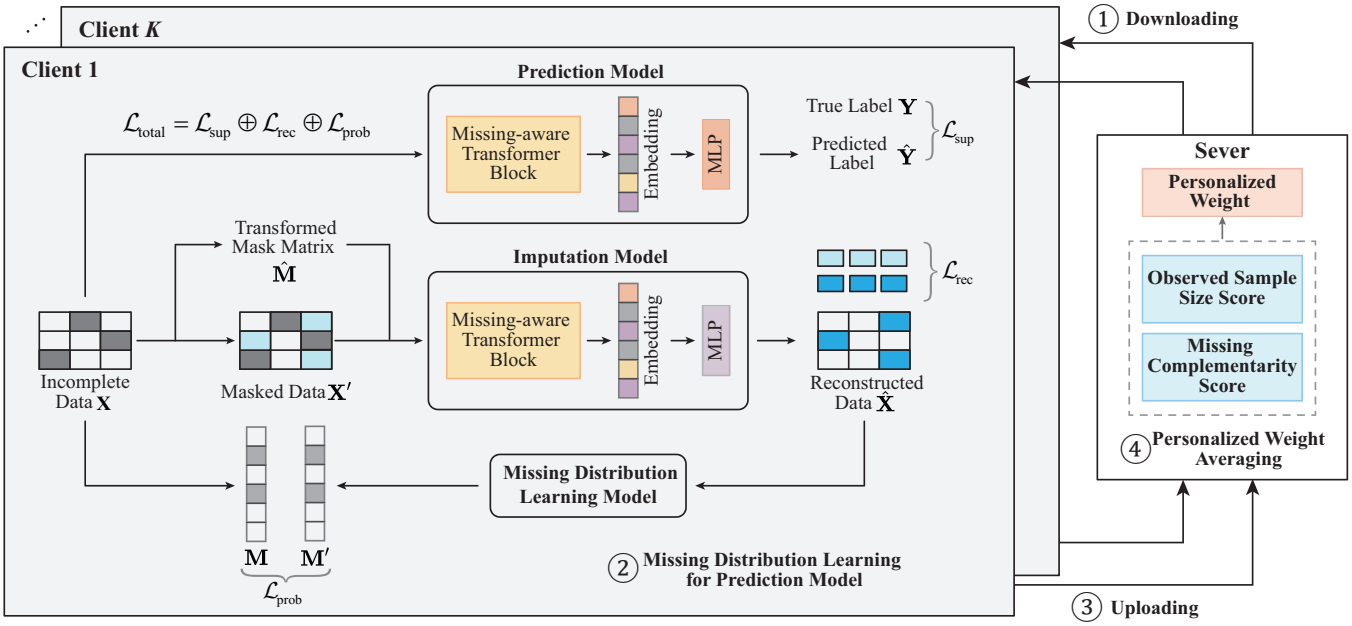


Figure 2: The overview framework of DARN.

performance of downstream tasks, while simultaneously ensuring privacy without introducing additional bias.

4.2 Missing Distribution Learning for Prediction Model

A straightforward approach to handling incomplete data in federated learning is to jointly impute the missing values across clients [33, 60]. Once the data is imputed, downstream prediction tasks can be performed using the completed dataset. However, it has two main limitations: (i) missing data imputation often introduces additional errors, as the process may rely on assumptions and estimates that are not always accurate, and (ii) averaging imputation model parameters disregards each client's unique missing data distributions. As a result, the performance of a global model trained on imputed data for specific downstream tasks may be suboptimal. To this end, DARN leverages the complementarity of each client's missing data distribution to perform prediction tasks directly, without relying on imputed data. The advantage of our method is that it avoids potential bias and error accumulation associated with data imputation, particularly when data is highly incomplete or the missing mechanism is complex (*i.e.,* MNAR).

Learning each client's missing data distribution is essential for obtaining a high-performance prediction model with incomplete data. We achieve this by designing an *imputation model* with the shared missing-aware transformer block and a *missing distribution learning model*. Based on the heterogeneity missing distribution, high-performance *prediction models* can be derived.

Missing-aware transformer-based imputation model. To avoid the imputation error, we incorporate the transformer to represent the incomplete tabular data directly. However, traditional transformer models require supplementary imputation algorithms to handle missing data [20]. Even when padding vectors are used to replace embeddings for missing values, it can still negatively impact the calculation of attention scores. To address this, we propose a missing-aware transformer block with a novel missing-aware attention mechanism. It enables the model to learn the distributions of both observed data and the missing states of incomplete data, enhancing the generation of effective representations.

Specifically, we first mask the incomplete data $\mathbf{X} \in \mathbb{R}^{n \times d}$ with a random mask matrix $\mathbf{R}$ at a rate of ρ to get masked attribution matrix $\mathbf{X}'$. The random mask matrix representing the masking status of each value is denoted by $\mathbf{R} \in \{0,1\}^{n \times d} = (\mathbf{r}_1, \cdots, \mathbf{r}_n)^\top$, where each vector $\mathbf{r}_i = (r_{i1}, \cdots, r_{id})$ corresponds to a sample $\mathbf{x}_i$. In particular, $r_{ij} = 1$ means that the j-th feature of $\mathbf{x}_i$ is masked, otherwise $r_{ij} = 0$. For the masked matrix $\mathbf{X}'$, the missing-aware transformer block first computes the query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ matrices using linear transformations. Meanwhile, based on the mask matrix $\mathbf{M}$, we generate the transformed mask matrix $\hat{\mathbf{M}}$, where each element $\hat{m}_{ij} \in \hat{\mathbf{M}}$ takes a value from $\{1, -\infty\}$. In detail, $\hat{m}_{ij} = 1$ (resp. $-\infty$) *iff* the $m_{ij} = 1$ (resp. 0). Then, the attention matrix $\mathbf{A}$ can be calculated as follows:

$$\mathbf{A} = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} + \hat{\mathbf{M}}\right)\mathbf{V},$$

where d_k is the dimension of the $\mathbf{K}$, affecting the scaling in the dot-product computation.

The main idea of the missing-aware attention mechanism is that special treatment is applied to missing values during the calculation of the attention matrix. It uses the transformed mask matrix $\hat{\mathbf{M}}$, which adds $-\infty$ to the attention scores corresponding to the missing values. The attention scores taking values of $-\infty$ are ignored after the *softmax* operation since they equal zero. This mechanism enables the transformer block to learn and represent incomplete data directly, without imputation, effectively eliminating the influence of missing values on data representation.

ID	Annual Income ($100k)	Credit Score (FICO)
A1	8.2	720
A2	NA	580
A3	9.8	650
A4	NA	780

(a) Incomplete data $\mathbf{X}$

ID	Annual Income ($100k)	Credit Score (FICO)
A1	8.2	NA
A2	NA	580
A3	NA	650
A4	NA	NA

(b) Masked data $\mathbf{X}'$

ID	Annual Income ($100k)	Credit Score (FICO)
A1	8.2	700
A2	11.0	580
A3	10.0	650
A4	12.5	800

(c) Reconstructed data $\hat{\mathbf{X}}$

Figure 3: An illustrative workflow of the imputation model.

Next, the attention matrix $\mathbf{A}$ is fed into normalization, residuals and feed-forward layers successively, similar to a traditional transformer, to produce the high-dimensional representation $\mathbf{H} \in \mathbb{R}^{d \times e}$, with d, e denoting the number of features and the embedding dimension, respectively. Finally, the high-dimensional representation $\mathbf{H}$ inputs an MLP to get the reconstructed data $\hat{\mathbf{X}}$.

We provide a running example to illustrate the workflow of the imputation model, as shown in Figure 3. It is designed to capture the underlying data distribution more accurately and produce more precise imputation results, thereby providing more effective guidance for training the missing distribution learning model. The objective of the imputation model $\mathcal{I}(\cdot|\theta^I)$ is to minimize the reconstruction loss, *i.e.,* the *mean absolute error* between the original values of the masked data and the reconstructed values. Taking the example from Figure 3, we calculate the mean absolute error for {9.8, 10.0}, {720, 700}, and {780, 800}. Consequently, the reconstruction loss of the k-th client can be expressed as follows:

$$\mathcal{L}_{\mathsf{rec}}(\theta_k^I) = \frac{1}{\sum_{i=1}^{n_k}\sum_{j=1}^{d} m_{ij} \cdot r_{ij}} \sum_{i=1}^{n_k}\sum_{j=1}^{d} m_{ij} \cdot r_{ij} \cdot \ell(x_{ij}, \hat{x}_{ij}), \quad (3)$$

where r_{ij} represents the value in the random mask matrix $\mathbf{R}$, and m_{ij} is the feature missing state of x_{ij}. $\hat{x}_{ij}$ is the predicted value estimated by the missing-aware transformer-based imputation model with the input of the masked values $\mathbf{X}'$ and the transformed mask matrix $\hat{\mathbf{M}}$, *i.e.,* $\hat{\mathbf{X}} = \mathcal{I}(\mathbf{X}', \hat{\mathbf{M}}|\theta^I)$. $\ell(x_{ij}, \hat{x}_{ij})$ represents the absolute error in the j-th feature of the i-th sample between $\mathbf{X}$ and $\hat{\mathbf{X}}$.

Missing distribution learning model. Based on the reconstructed data obtained through the missing-aware transformer-based imputation model, we design a missing distribution learning model to capture the unique missing distribution for each client. Specifically, we feed the reconstructed data $\hat{\mathbf{X}}$ into the missing distribution learning model, which consists of a logistic regression network. This network generates a missing probability matrix $\mathbf{M}'$, *i.e.,* $\mathbf{M}' = \mathcal{M}(\hat{\mathbf{X}}|\theta^M)$. By comparing the missing probability matrix $\mathbf{M}'$ with the mask matrix $\mathbf{M}$, the missing distribution learning model is trained. The parameters of the missing distribution learning model represent the client's unique missing distribution. Our empirical evaluation also validates the effectiveness of approximating the missing distribution using logistic regression. The binary cross-entropy loss is used to measure the difference between $m_{ij} \in \mathbf{M}$ and $m'_{ij} \in \mathbf{M}'$ for the k-th client as follows:

$$\mathcal{L}_{\mathsf{prob}}(\theta_k^M) = -\sum_{i=1}^{n_k}\sum_{j=1}^{d}\Big(m_{ij}\log(m'_{ij}) + (1-m_{ij})\log(1-m'_{ij})\Big), \quad (4)$$

where m_{ij} and m'_{ij} represent the feature missing state of $\mathbf{X}$ and $\hat{\mathbf{X}}$, repectively.

Missing-aware transformer-based prediction model. Leveraging the learned distribution of missing data, these clients, which are highly complementary, collaboratively train personalized prediction models. The prediction model utilizes the shared missing-aware transformer block from the imputation model to generate a comprehensive data embedding. Subsequently, this model employs an additional MLP to predict the label $\hat{\mathbf{Y}}$ using the derived data embedding, *i.e.,* $\hat{\mathbf{Y}} = \mathcal{P}(\mathbf{X}, \mathbf{M}|\theta^P)$. We apply a supervised learning loss, $\mathcal{L}_{\mathsf{sup}}$, to update the learnable weights in this MLP. Here, we also use the cross-entropy loss to calculate the difference between the predicted label and the ground truth for the k-th client as follows:

$$\mathcal{L}_{\mathsf{sup}}(\theta_k^P) = -\Big(\mathrm{Y}\log(\hat{\mathrm{Y}}) + (1-\mathrm{Y})\log(\mathbf{1}-\hat{\mathrm{Y}})\Big). \quad (5)$$

4.3 Personalized Weight Averaging

To maximize each client's available data for direct prediction without relying on estimated data, we leverage the missing complementary score and observed sample size score to calculate personalized weights, resulting in high-performance prediction models. Formally, let O_k denote the total number of observed features across all samples (*i.e.,* cells) for the k-th client, and let θ_k^M represent the missing distribution (*i.e.,* the parameters of the missing distribution learning model) for the k-th client. Specifically, we calculate the observed sample size score by:

$$C_k = \frac{O_k}{\max\{O_1, O_2, \cdots, O_K\}}. \quad (6)$$

We measure the *cosine similarity* between two missing distributions (θ_i^M of i-th client and θ_j^M of j-th client) to denote the pair missing complementary score as follows:

$$S_{ij} = \frac{1}{2}\left(1 - \frac{\sum_{k=1}^{d}\theta_{ik}^M\theta_{jk}^M}{\sqrt{\sum_{k=1}^{d}(\theta_{ik}^M)^2}\cdot\sqrt{\sum_{k=1}^{d}(\theta_{jk}^M)^2}}\right), \quad (7)$$

where d refers to the dimensionality of the features in the input tabular data. The value of S lies within the interval $[0, 1]$. If the missing distributions θ_i^M and θ_j^M of two clients are very similar, the cosine similarity approaches 1, causing S_{ij} to approach 0, which indicates weak complementarity. Conversely, if the missing distribution of the two clients differs significantly, the cosine similarity decreases, and S_{ij} approaches 1, reflecting strong complementarity.

In addition, the number of complete samples is also important to the prediction model. The less missing data there is, the more information a client can provide. Therefore, these clients with high complementary scores and larger observed cell sizes can collaboratively train high-performance prediction models with greater personalized weights. To achieve this, we incorporate the missing complementary scores S and the observed sample size scores C into our personalized model aggregation process to get the personalized prediction model. For each pair of client i and client j ($i, j \in [K], i \neq j$), the weight of i-th client is calculated by:

$$w_{ij} = \beta C_j + (1-\beta)S_{ij}, \quad (8)$$

where β is a hyperparameter to adjust the importance degree of observed sample size scores or missing complementary scores. Then, the personalized prediction model and imputation model for i-th

Algorithm 1: The DARN Algorithm

Input: K clients, collaborative communication rounds T, local training epochs E, incomplete datasets D_k and its mask matrix $\mathbf{M}_k$, initialized prediction model $\mathcal{P}$ and its parameters θ^P, initialized imputation model $\mathcal{I}$ and its parameters θ^I, and initialized missing mechanism model $\mathcal{M}$ and its parameters θ^M, batch size b

Output: The optimal personalized prediction model $\{\theta_1^{P*}, \cdots, \theta_K^{P*}\}$

Server executes: */* Run on central server */*

1: initialize θ_0^P, θ_0^I and θ_0^M
2: **for** each round t from 0 to $(T-1)$ **do**
3: **for** each client $k \in [K]$ **in parallel do**
4: $\{\theta_{k,t+1}^P, \theta_{k,t+1}^I, \theta_{k,t+1}^M, O_k\} \leftarrow$ ClientUpdate $(\theta_{k,t}^P, \theta_{k,t}^I)$
5: calculate observed sample size scores with Eq. 6
6: calculate missing complementary scores with Eq. 7
7: calculate personalized weight with Eq. 8
8: update prediction and imputation model with Eq. 9
9: **return** $\{\theta_1^{P*}, \cdots, \theta_K^{P*}\}$

Client executes: */* Run on client k */*

10: **Function** ClientUpdate $(\theta_{k,t}^P, \theta_{k,t}^I)$**:**
11: $\theta_0^P, \theta_0^I \leftarrow$ *deepcopy* $(\theta_{k,t}^P, \theta_{k,t}^I)$
12: calculate the number of observed cells O_k
13: **for** each local epoch e from 0 to $(E-1)$ **do**
14: $\{\mathbf{X}, \mathbf{Y}\} \leftarrow$ SampleBatch (D_k, b)
15: get predicted label $\hat{\mathbf{Y}} = \mathcal{P}(\mathbf{X}, \mathbf{M}|\theta_e^P)$
16: get mask incomplete data $\mathbf{X}'$ with $\mathbf{R}$
17: get transformed mask matrix $\hat{\mathbf{M}}$
18: get reconstructed data $\hat{\mathbf{X}} = \mathcal{I}(\mathbf{X}', \hat{\mathbf{M}}|\theta_e^I)$
19: get predicted missing probability matrix $\mathbf{M}' = \mathcal{M}(\hat{\mathbf{X}}|\theta_e^M)$
20: calculate supervised loss $\mathcal{L}_{\mathsf{sup}}$ with Eq. 5
21: calculate reconstruction loss $\mathcal{L}_{\mathsf{rec}}$ with Eq. 3
22: calculate missing distribution learning loss $\mathcal{L}_{\mathsf{pro}}$ with Eq. 4
23: update the $\theta_{e+1}^P, \theta_{e+1}^I, \theta_{e+1}^M$ with Eq. 2
24: $\theta_{t+1}^P = \theta_E^P$, $\theta_{t+1}^I = \theta_E^I$, $\theta_{t+1}^M = \theta_E^M$
25: **return** $\theta_{k,t+1}^P, \theta_{k,t+1}^I, \theta_{k,t+1}^M, O_k$

client can be refined by the personalized weight as follows:

$$\theta_i^P = \gamma\theta_i^P + (1-\gamma)\frac{1}{K-1}\sum_{j\neq i, j\in[K]} w_{ij}\theta_j^P,$$
$$\theta_i^I = \gamma\theta_i^I + (1-\gamma)\frac{1}{K-1}\sum_{j\neq i, j\in[K]} w_{ij}\theta_j^I, \tag{9}$$

where γ is a hyperparameter to adjust the personalized weight.

4.4 Algorithm Overall Procedure

The training process of DARN is presented in Algorithm 1. Initially, the central server initializes a global prediction model θ_0^P, an imputation model θ_0^I and a missing distribution learning model θ_0^M. In the t-th interaction, the server sends the prediction model θ_t^P, the imputation model θ_t^I and the missing distribution learning model θ_t^M to each client $k \in [K]$. For k-th client, after downloading θ_t^P, θ_t^I from the server, it has three operations: (i) get predicted label $\hat{\mathbf{Y}}$ using local incomplete data $\mathbf{X}$ and corresponding mask matrix $\mathbf{M}$ (Line 15); (ii) random mask incomplete data to get $\mathbf{X}'$ with random mask matrix $\mathbf{R}$ and get transformed mask matrix $\hat{\mathbf{M}}$, leverage $\mathbf{X}'$ and $\hat{\mathbf{M}}$ as input to get reconstructed data $\hat{\mathbf{X}}$ (Lines 16-18); and (iii) get predicted missing probability matrix $\mathbf{M}'$ (Line 19). Then, the supervised loss $\mathcal{L}_{\mathsf{sup}}$, reconstruction loss $\mathcal{L}_{\mathsf{rec}}$, and missing distribution learning loss $\mathcal{L}_{\mathsf{pro}}$ are calculated (Lines 20-22). By minimizing the loss function with Eq. 2, the parameters of the prediction model θ_{t+1}^P, imputation model θ_{t+1}^I and missing distribution learning model θ_{t+1}^M can be updated. These parameters, $\theta_{k,t+1}^P, \theta_{k,t+1}^I, \theta_{k,t+1}^M$, and O_k, are sent to the server. After receiving the parameters from all clients, the server calculates the observed sample size score with Eq. 6 (Line 5) and missing complementary score with Eq. 7 (Line 6). Combining the above two scores, the personalized weight can be calculated with Eq. 8 (Line 7). Finally, the server can update the prediction model and imputation model according to personalized weight with Eq. 9 (Line 8). The whole process repeats until convergence or meets the predefined requirements. In this way, the optimal personalized prediction models $\{\theta_1^{P*}, \cdots, \theta_K^{P*}\}$ are constructed, maximizing the utility of observed data by leveraging missing complementarity without relying on estimated values.

4.5 Privacy Protection Enhanced DARN

We argue that DARN adheres to the standard FL training protocol by transmitting model parameters rather than local training data, thus protecting each client's local training data from exposure to other parties, including the FL server [35]. Specifically, the transmitted parameters are the prediction models θ^P, imputation models θ^I, missing distribution learning models θ^M, and the number of observed cells O. The number of observed cells is an aggregated statistic that does not reveal any raw data or sensitive loss distribution of individual samples. However, the parameters of $\theta^P, \theta^I, \theta^M$, and O may still pose a risk of revealing sensitive user information if malicious entities attempt to infer private details. To mitigate this risk, we incorporate the local differential privacy (DP) [46] technique into our method.

Definition 1. Federated (ϵ, δ)-Differential Privacy with Laplace Noise. *Let $\mathcal{R}: D_1 \times \cdots \times D_K \to \mathcal{Y}$ be a randomized mechanism in a federated system with K clients. $\mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent federated datasets $D = (D_1, \cdots, D_k, \cdots, D_K)$ and $D' = (D_1, \cdots, D_k', \cdots, D_K)$ differing in at most one data record of any single client D_k, and for all measurable subsets $Y \subseteq \mathcal{Y}$:*

$$\Pr[\mathcal{R}(D) \in Y] \leq e^{\epsilon} \cdot \Pr[\mathcal{R}(D') \in Y] + \delta. \tag{10}$$

To enforce this guarantee, each client k perturbs its shared parameters $\theta_k = \{\theta_k^P, \theta_k^I, \theta_k^M, O_k\}$ in each communication round as $\tilde{\theta}_k = \theta_k + \eta_k$, where $\eta_k \sim \text{Laplace}(0, \Delta f/\epsilon)$. Here, ϵ denotes privacy budget and Δf denotes the L_1-*sensitivity* of the parameter update function f, defined as: $\Delta f = \max_{D_k, D_k'} \|f(D_k) - f(D_k')\|$, where D_k and D_k' are adjacent local datasets differing by one record. The global model, obtained by aggregating $\{\tilde{\theta}_k\}_{k=1}^K$, preserves ϵ-DP due to the post-processing immunity of differential privacy. The experimental results in Table 10 demonstrate that by adjusting the intensity of noise, $\lambda = \Delta f/\epsilon$, we can control the privacy protection capability of DARN. Specifically, increasing the noise intensity enhances the effectiveness of privacy protection.

Table 2: Dataset statistics in the experiments.

Name	# Samples	# Features	Task
Bank	45,211	16	Classification
Higgs	98,050	29	Classification
Covertype	581,012	54	Classification
Gas	4,178,504	56	Regression
Beers	2,410	11	Classification
Mobility	2,268,105	77	Regression

5 EXPERIMENTS

In this section, we evaluate the performance of our proposed framework, DARN, against six state-of-the-art federated tabular data prediction methods in both classification and regression tasks. All methods are implemented in Python. The experiments are conducted in an Intel Core 2.90GHz server with 3 × A40 48GB (GPUs) and 256GB RAM, running on the Ubuntu 18.04 system.

5.1 Experimental Setup

Datasets. Four publicly available real-world datasets are utilized to evaluate the effectiveness of DARN: Bank [38], Higgs [5], Covertype [7] and Gas [26]. Two real-world *incomplete* datasets, Beers [39] and Mobility [45], are employed to test its effectiveness and applicability, with average missing rates of 16% and 30.62%, respectively. The characteristics of these datasets are shown in Table 2. For each dataset, 10% is randomly selected for testing, 10% for validation, and the remaining 80% is used for training. The missing rate of the dataset is denoted by R. To simulate an independent and identically distributed (IID) scenario, all training data are randomly assigned to all clients. Following [24], we model the non-IID case by incorporating *Dirichlet sampling* (*i.e.,* $\mathrm{Dir}(\varphi)$) to capture label distribution skew across clients, where φ indicates heterogeneity degree. A small φ means high heterogeneity.

Missing pattern simulation. We simulate MNAR patterns, as discussed in [10], because they frequently occur in real-world scenarios and pose significant challenges to address. In this paper, we categorize the MNAR missing patterns into two types: MNAR-High and MNAR-Low. To simulate MNAR, an attribute f_m is first selected. The value x_{im} of a sample $\mathbf{x}_i$ is missing with a probability of $\mathrm{P}_m(x_{im})$, *i.e.,*

$$\mathrm{P}_m(x_{im}) = \Phi(x_{im}) / \sum_{i=1}^{n} \Phi(x_{im}),$$

where $\Phi(x_{im})$ represents the ranking of x_{im} in f_m. In the MNAR-High setting, higher values of f_m correspond to higher rankings, whereas in the MNAR-Low setting, lower values of f_m result in higher rankings. We define the missing pattern as $\mathcal{M}_1 = \mathcal{M}(C_1, r_1)$ and $\mathcal{M}_2 = \mathcal{M}(C_2, r_2)$, where r_1 and r_2 represent the missing rates of each client, and the classes $C_1, C_2 \in \{H, L\}$ denote High (H) and Low (L), respectively.

Federated learning scenarios with missing data. To capture the heterogeneity of missing data patterns and address a common situation of missing data across clients, we define the following six scenarios, each varying in complementarity, generalizability, or missing mechanisms. Let the $\mathcal{M}$-set represent the set of all missing patterns, defined by varying missing rates and class values.

- *Balanced complementarity (BC) scenario*: For each feature, half of the clients adhere to the missing pattern $\mathcal{M}(H, 0.5)$, while the other half follow $\mathcal{M}(L, 0.5)$. This configuration represents an ideal case of balanced complementarity, where DARN is expected to achieve optimal performance.
- *Complete complementarity (CC) scenario*: For each feature, half of the clients are randomly selected and provided with a missing pattern drawn from the $\mathcal{M}$-set. The complementary missing pattern is assigned for the same feature in the remaining clients. For example, if a feature in the selected clients follows $\mathcal{M}(H, 0.4)$, the other clients are assigned $\mathcal{M}(L, 0.6)$.
- *Partial complementarity (PC) scenario*: For each feature, half of the clients are randomly selected and assigned a missing pattern from the $\mathcal{M}$-set, denoted as $\mathcal{M}_1(C_1, r_1)$. For the same feature in the remaining clients, a partially complementary pattern $\mathcal{M}_2(C_2, r_2)$ is applied, satisfying $C_1 \neq C_2$ and $r_1 \neq 1 - r_2$.
- *Single-sided complementarity (SSC) scenario*: In this scenario, half of the clients are randomly selected and assigned a pattern from the $\mathcal{M}$-set, denoted as $\mathcal{M}_1(C_1, r_1)$. For the same feature, the remaining clients are assigned a single-sided complementarity pattern $\mathcal{M}_2(C_2, r_2)$, meaning the patterns partially vary from, but do not fully complement, those of the selected clients, satisfying $C_1 = C_2$ and $r_1 \neq r_2$.
- *Completely random (CR) scenario*: In this scenario, each client is randomly assigned a pattern from the $\mathcal{M}$-set for each feature, which may result in fully independent and uncoordinated missing patterns across all clients.
- *Mixed missing mechanism (MMM) scenario*: In this scenario, each client is randomly assigned one missing data mechanism from MCAR, MAR, and MNAR with equal probability, and the missing rate is set to 50% for all clients.

Baselines. In the experiments, we evaluate DARN against six state-of-the-art baseline methods, comprising three machine learning-based methods (F-XGBoost [51], F-GBDT [30] and F-RF [48]), three deep learning-based variants employed in FL (F-MLP [20], F-TabNet [4], and F-SAINT [43]), and two variants of DARN (Central-DARN and Local-DARN). Since F-XGBoost is the only method capable of directly handling incomplete tabular data, we employ four advanced data imputation methods: F-Mean [14], F-MIWAE [34], F-NMIWAE [25], and F-GAIN [54] to impute missing values for the other methods. Notably, federated versions of all these deep learning prediction and imputation methods are implemented using the FedAVG [35] algorithm, adapted from their respective local implementations. Central-DARN refers to a centralized learning framework where all data is stored in a single location, without any privacy preservation. Local-DARN involves all clients independently training their local models without parameter aggregation, and we report the average performance metrics across all these clients.

Metrics. To evaluate the performance of all methods, we use *AUC* and standard *accuracy* to evaluate classification performance. AUC quantifies the area under the receiver operating characteristic curve based on the prediction results, whereas accuracy represents the proportion of correctly classified samples relative to the total number of samples in the dataset. For the regression task, we employ *root mean square error (RMSE)* and *R-squared (R2)*. RMSE quantifies the mean squared magnitude of the prediction errors, whereas R2 indicates the proportion of the variance in the dependent variable that is explained by the independent variables. A lower RMSE value

Table 3: The prediction performance under the BC scenario.

Datasets		Bank		Higgs		Covertype		Gas	
Models		Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	RMSE ↓	R2 ↑
F-XGBoost		0.843 ± 0.004	0.723 ± 0.003	0.649 ± 0.003	0.707 ± 0.002	0.707 ± 0.005	0.933 ± 0.003	103.436 ± 2.551	0.994 ± 0.001
F-GBDT	F-Mean	0.831 ± 0.003	0.710 ± 0.004	0.635 ± 0.003	0.685 ± 0.001	0.692 ± 0.003	0.911 ± 0.001	116.968 ± 3.019	0.993 ± 0.001
	F-MIWAE	0.815 ± 0.002	0.695 ± 0.003	0.618 ± 0.004	0.666 ± 0.005	0.678 ± 0.003	0.897 ± 0.004	–	–
	F-NMIWAE	0.818 ± 0.003	0.698 ± 0.002	0.621 ± 0.005	0.669 ± 0.003	0.683 ± 0.001	0.901 ± 0.002	–	–
	F-GAIN	0.834 ± 0.002	0.715 ± 0.003	0.638 ± 0.001	0.689 ± 0.003	0.697 ± 0.004	0.919 ± 0.003	107.149 ± 5.147	0.994 ± 0.001
F-RF	F-Mean	0.814 ± 0.002	0.691 ± 0.002	0.619 ± 0.004	0.659 ± 0.007	0.664 ± 0.002	0.885 ± 0.003	–	–
	F-MIWAE	0.802 ± 0.003	0.680 ± 0.002	0.608 ± 0.004	0.646 ± 0.005	0.655 ± 0.001	0.869 ± 0.003	–	–
	F-NMIWAE	0.805 ± 0.002	0.684 ± 0.002	0.610 ± 0.001	0.648 ± 0.002	0.659 ± 0.003	0.874 ± 0.001	–	–
	F-GAIN	0.819 ± 0.003	0.697 ± 0.002	0.624 ± 0.003	0.665 ± 0.004	0.671 ± 0.001	0.894 ± 0.001	–	–
F-MLP	F-Mean	0.824 ± 0.003	0.702 ± 0.003	0.623 ± 0.004	0.666 ± 0.003	0.677 ± 0.003	0.894 ± 0.001	133.398 ± 7.269	0.990 ± 0.001
	F-MIWAE	0.812 ± 0.002	0.690 ± 0.002	0.617 ± 0.004	0.656 ± 0.002	0.659 ± 0.007	0.882 ± 0.002	–	–
	F-NMIWAE	0.816 ± 0.003	0.695 ± 0.002	0.621 ± 0.001	0.660 ± 0.002	0.668 ± 0.005	0.890 ± 0.003	–	–
	F-GAIN	0.828 ± 0.002	0.708 ± 0.002	0.633 ± 0.005	0.684 ± 0.005	0.683 ± 0.002	0.899 ± 0.002	125.699 ± 4.937	0.992 ± 0.001
F-TabNet	F-Mean	0.837 ± 0.002	0.715 ± 0.003	0.629 ± 0.003	0.674 ± 0.003	0.717 ± 0.002	0.937 ± 0.001	86.418 ± 6.841	0.995 ± 0.001
	F-MIWAE	0.825 ± 0.003	0.705 ± 0.002	0.623 ± 0.001	0.667 ± 0.001	0.704 ± 0.001	0.928 ± 0.002	–	–
	F-NMIWAE	0.829 ± 0.002	0.710 ± 0.002	0.624 ± 0.003	0.671 ± 0.001	0.712 ± 0.003	0.937 ± 0.002	–	–
	F-GAIN	0.840 ± 0.002	0.720 ± 0.002	0.634 ± 0.003	0.685 ± 0.004	0.728 ± 0.002	0.942 ± 0.001	78.175 ± 3.184	0.995 ± 0.001
F-SAINT	F-Mean	0.840 ± 0.003	0.720 ± 0.002	0.637 ± 0.003	0.688 ± 0.002	0.723 ± 0.002	0.941 ± 0.001	72.491 ± 5.497	0.996 ± 0.001
	F-MIWAE	0.828 ± 0.002	0.708 ± 0.002	0.625 ± 0.002	0.672 ± 0.003	0.709 ± 0.003	0.933 ± 0.004	–	–
	F-NMIWAE	0.832 ± 0.003	0.713 ± 0.002	0.627 ± 0.005	0.674 ± 0.002	0.713 ± 0.005	0.938 ± 0.003	–	–
	F-GAIN	0.843 ± 0.002	0.725 ± 0.002	0.644 ± 0.006	0.702 ± 0.004	0.734 ± 0.002	0.947 ± 0.003	67.198 ± 1.487	0.997 ± 0.001
Central-DARN		0.856 ± 0.003	0.744 ± 0.002	0.643 ± 0.001	0.700 ± 0.003	0.727 ± 0.002	0.943 ± 0.002	74.164 ± 7.928	0.997 ± 0.001
Local-DARN		0.852 ± 0.002	0.740 ± 0.003	0.639 ± 0.001	0.696 ± 0.001	0.721 ± 0.002	0.940 ± 0.003	71.948 ± 6.156	0.997 ± 0.001
DARN		**0.878 ± 0.001**	**0.781 ± 0.001**	**0.662 ± 0.001**	**0.721 ± 0.001**	**0.770 ± 0.002**	**0.967 ± 0.001**	**40.147 ± 2.009**	**0.999 ± 0.001**

Table 4: The prediction performance under the CC scenario.

Datasets	Bank		Higgs		Covertype		Gas	
Models	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	RMSE ↓	R2 ↑
F-XGBoost	0.832 ± 0.003	0.712 ± 0.002	0.651 ± 0.005	0.710 ± 0.005	0.703 ± 0.002	0.929 ± 0.003	99.681 ± 4.651	0.994 ± 0.001
F-GBDT	0.820 ± 0.002	0.699 ± 0.003	0.640 ± 0.004	0.692 ± 0.002	0.695 ± 0.003	0.917 ± 0.005	111.519 ± 5.941	0.993 ± 0.001
F-RF	0.805 ± 0.003	0.681 ± 0.002	0.621 ± 0.001	0.662 ± 0.002	0.668 ± 0.006	0.891 ± 0.003	–	–
F-MLP	0.814 ± 0.002	0.692 ± 0.002	0.631 ± 0.005	0.682 ± 0.004	0.689 ± 0.003	0.902 ± 0.004	127.581 ± 3.654	0.992 ± 0.001
F-TabNet	0.826 ± 0.003	0.706 ± 0.002	0.632 ± 0.004	0.686 ± 0.003	0.721 ± 0.004	0.939 ± 0.003	81.651 ± 0.001	0.995 ± 0.001
F-SAINT	0.829 ± 0.002	0.711 ± 0.003	0.643 ± 0.003	0.701 ± 0.002	0.732 ± 0.001	0.944 ± 0.002	71.948 ± 4.738	0.997 ± 0.001
Central-DARN	0.847 ± 0.002	0.735 ± 0.003	0.641 ± 0.001	0.694 ± 0.003	0.727 ± 0.003	0.941 ± 0.004	68.417 ± 6.185	0.997 ± 0.001
Local-DARN	0.843 ± 0.003	0.731 ± 0.002	0.637 ± 0.002	0.689 ± 0.002	0.720 ± 0.003	0.939 ± 0.002	76.779 ± 3.617	0.997 ± 0.001
DARN	**0.868 ± 0.002**	**0.772 ± 0.002**	**0.658 ± 0.002**	**0.717 ± 0.001**	**0.767 ± 0.002**	**0.964 ± 0.002**	**45.164 ± 3.698**	**0.999 ± 0.001**

Table 5: The prediction performance under the PC scenario.

Datasets	Bank		Higgs		Covertype		Gas	
Models	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	RMSE ↓	R2 ↑
F-XGBoost	0.821 ± 0.002	0.702 ± 0.002	0.649 ± 0.003	0.708 ± 0.002	0.701 ± 0.001	0.926 ± 0.002	107.982 ± 0.669	0.994 ± 0.001
F-GBDT	0.810 ± 0.003	0.691 ± 0.002	0.636 ± 0.004	0.688 ± 0.003	0.695 ± 0.002	0.915 ± 0.002	114.648 ± 1.233	0.994 ± 0.001
F-RF	0.797 ± 0.002	0.673 ± 0.002	0.622 ± 0.001	0.661 ± 0.001	0.664 ± 0.003	0.886 ± 0.001	–	–
F-MLP	0.805 ± 0.002	0.684 ± 0.002	0.626 ± 0.001	0.676 ± 0.001	0.684 ± 0.001	0.894 ± 0.001	131.495 ± 2.541	0.992 ± 0.001
F-TabNet	0.817 ± 0.002	0.698 ± 0.002	0.634 ± 0.005	0.684 ± 0.004	0.724 ± 0.003	0.940 ± 0.001	86.176 ± 2.481	0.995 ± 0.001
F-SAINT	0.820 ± 0.002	0.703 ± 0.002	0.636 ± 0.001	0.693 ± 0.002	0.728 ± 0.001	0.942 ± 0.002	73.486 ± 4.561	0.997 ± 0.001
Central-DARN	0.838 ± 0.002	0.727 ± 0.002	0.640 ± 0.004	0.693 ± 0.004	0.725 ± 0.003	0.941 ± 0.002	70.165 ± 7.169	0.997 ± 0.001
Local-DARN	0.834 ± 0.002	0.723 ± 0.002	0.637 ± 0.002	0.690 ± 0.003	0.714 ± 0.002	0.936 ± 0.002	79.146 ± 5.532	0.997 ± 0.001
DARN	**0.859 ± 0.001**	**0.764 ± 0.001**	**0.653 ± 0.001**	**0.713 ± 0.002**	**0.754 ± 0.001**	**0.957 ± 0.002**	**52.194 ± 5.024**	**0.999 ± 0.001**

signifies superior prediction performance, whereas higher values correspond to better performance for the other three metrics.

Implementation details. The total number of clients K in the FL system is set to 10. Each federated imputation algorithm is configured with a learning rate of 0.3 and is run for a total of $T = 100$. For F-MIWAE and F-NMIWAE, the sampling size is set to 10, and their corresponding local models are MIWAE and non-MIWAE, respectively, both grounded in the importance-weighted autoencoder framework. In F-Mean, each client sends its local mean to the server, which computes the global mean by averaging the received local means and then sends it back to the clients. In F-GAIN, the local model is GAIN, with both the generator and discriminator implemented as two-layer fully connected networks. For all federated prediction methods, the learning rate is set to 0.001, and $T = 100$. In F-XGBoost, the maximum tree depth is fixed at 5. For two attention-based federated prediction methods (F-TabNet and F-SAINT), the embedding size is set to 32, the number of attention heads is 4, the dropout rate is 0.5, and the transformer depth is 6. In DARN, the model is trained with hyperparameters $\alpha = 0.5$, $\beta = 0.8$ and $\gamma = 0.5$. We restrict the missing rate r to a moderate range of 0.3 to 0.7. This ensures sufficient data for learning while maintaining a substantial missing rate to assess its impact across various scenarios effectively. For the non-IID setting, we set $\varphi = 0.1$ to simulate high heterogeneity. An early stopping strategy [53] is employed for all methods, which halts training if the validation loss does not improve for seven consecutive epochs, thereby mitigating overfitting. Each set of experiments is repeated five times with independent random seeds to ensure reliability, and the results, along with their margin of error, are presented.

5.2 Overall Performance

Effectiveness. We assess the effectiveness of federated prediction methods across the six scenarios, as shown in Tables 3-9. The best results are highlighted in **bold**. It is observed that F-RF is unable

Table 6: The prediction performance under the SSC scenario.

Datasets	Bank		Higgs		Covertype		Gas	
Models	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	RMSE ↓	R2 ↑
F-XGBoost	0.813 ± 0.003	0.693 ± 0.003	0.646 ± 0.002	0.706 ± 0.002	0.689 ± 0.006	0.913 ± 0.005	109.781 ± 5.517	0.994 ± 0.001
F-GBDT	0.802 ± 0.004	0.682 ± 0.003	0.634 ± 0.007	0.687 ± 0.010	0.684 ± 0.005	0.913 ± 0.003	117.982 ± 1.981	0.993 ± 0.001
F-RF	0.789 ± 0.003	0.665 ± 0.002	0.618 ± 0.008	0.656 ± 0.006	0.651 ± 0.002	0.871 ± 0.003	–	–
F-MLP	0.796 ± 0.003	0.676 ± 0.002	0.623 ± 0.003	0.671 ± 0.002	0.673 ± 0.007	0.886 ± 0.004	135.714 ± 4.897	0.992 ± 0.001
F-TabNet	0.808 ± 0.002	0.689 ± 0.002	0.628 ± 0.003	0.675 ± 0.001	0.710 ± 0.001	0.931 ± 0.002	90.641 ± 3.983	0.994 ± 0.001
F-SAINT	0.811 ± 0.002	0.694 ± 0.002	0.638 ± 0.001	0.695 ± 0.001	0.722 ± 0.001	0.936 ± 0.002	76.415 ± 6.614	0.997 ± 0.001
Central-DARN	0.829 ± 0.003	0.718 ± 0.002	0.643 ± 0.001	0.697 ± 0.002	0.726 ± 0.004	0.941 ± 0.005	64.517 ± 2.148	0.998 ± 0.001
Local-DARN	0.825 ± 0.002	0.714 ± 0.002	0.638 ± 0.002	0.693 ± 0.003	0.717 ± 0.001	0.938 ± 0.003	68.492 ± 4.738	0.997 ± 0.001
DARN	**0.851 ± 0.001**	**0.755 ± 0.001**	**0.648 ± 0.001**	**0.707 ± 0.001**	**0.731 ± 0.001**	**0.947 ± 0.001**	**55.134 ± 6.517**	**0.999 ± 0.001**

Table 7: The prediction performance under the CR scenario.

Datasets	Bank		Higgs		Covertype		Gas	
Models	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	RMSE ↓	R2 ↑
F-XGBoost	0.838 ± 0.003	0.718 ± 0.002	0.642 ± 0.001	0.698 ± 0.002	0.705 ± 0.003	0.929 ± 0.002	105.564 ± 3.477	0.994 ± 0.001
F-GBDT	0.826 ± 0.002	0.705 ± 0.003	0.632 ± 0.006	0.687 ± 0.002	0.696 ± 0.006	0.918 ± 0.004	103.189 ± 3.655	0.993 ± 0.001
F-RF	0.811 ± 0.002	0.688 ± 0.002	0.618 ± 0.003	0.655 ± 0.004	0.676 ± 0.001	0.901 ± 0.002	–	–
F-MLP	0.820 ± 0.003	0.699 ± 0.002	0.626 ± 0.004	0.673 ± 0.005	0.688 ± 0.005	0.899 ± 0.002	114.487 ± 7.246	0.993 ± 0.001
F-TabNet	0.832 ± 0.002	0.710 ± 0.003	0.631 ± 0.005	0.675 ± 0.003	0.719 ± 0.001	0.936 ± 0.003	82.791 ± 4.489	0.994 ± 0.001
F-SAINT	0.835 ± 0.003	0.715 ± 0.003	0.641 ± 0.001	0.698 ± 0.001	0.737 ± 0.002	0.948 ± 0.003	69.486 ± 3.332	0.997 ± 0.001
Central-DARN	0.852 ± 0.003	0.739 ± 0.002	0.638 ± 0.003	0.692 ± 0.002	0.733 ± 0.004	0.946 ± 0.003	71.912 ± 6.166	0.998 ± 0.001
Local-DARN	0.849 ± 0.002	0.736 ± 0.003	0.636 ± 0.004	0.688 ± 0.003	0.723 ± 0.002	0.939 ± 0.004	73.984 ± 8.728	0.997 ± 0.001
DARN	**0.873 ± 0.001**	**0.777 ± 0.001**	**0.660 ± 0.002**	**0.720 ± 0.001**	**0.773 ± 0.001**	**0.971 ± 0.001**	**38.624 ± 1.137**	**0.999 ± 0.001**

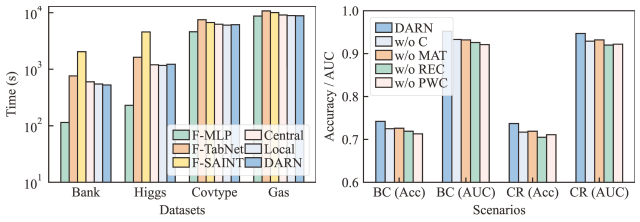
Table 8: The prediction performance under the MMM scenario.

Datasets	Bank		Higgs		Covertype		Gas	
Models	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	RMSE ↓	R2 ↑
F-XGBoost	0.824 ± 0.003	0.704 ± 0.002	0.628 ± 0.003	0.683 ± 0.002	0.691 ± 0.004	0.914 ± 0.003	108.245 ± 3.500	0.993 ± 0.002
F-GBDT	0.808 ± 0.003	0.689 ± 0.002	0.619 ± 0.005	0.672 ± 0.003	0.682 ± 0.002	0.903 ± 0.004	115.892 ± 3.700	0.993 ± 0.002
F-RF	0.789 ± 0.002	0.668 ± 0.002	0.605 ± 0.002	0.641 ± 0.003	0.662 ± 0.003	0.886 ± 0.002	–	–
F-MLP	0.810 ± 0.002	0.694 ± 0.001	0.613 ± 0.001	0.659 ± 0.004	0.673 ± 0.004	0.892 ± 0.003	132.782 ± 7.300	0.991 ± 0.002
F-TabNet	0.811 ± 0.002	0.693 ± 0.001	0.621 ± 0.006	0.661 ± 0.005	0.705 ± 0.003	0.921 ± 0.003	87.345 ± 4.600	0.994 ± 0.002
F-SAINT	0.817 ± 0.001	0.709 ± 0.002	0.625 ± 0.002	0.683 ± 0.003	0.713 ± 0.002	0.931 ± 0.003	74.123 ± 4.800	0.996 ± 0.002
Central-DARN	0.815 ± 0.002	0.701 ± 0.002	0.623 ± 0.004	0.681 ± 0.004	0.709 ± 0.004	0.929 ± 0.003	72.345 ± 7.300	0.996 ± 0.002
Local-DARN	0.806 ± 0.002	0.693 ± 0.002	0.620 ± 0.003	0.678 ± 0.004	0.698 ± 0.003	0.924 ± 0.004	81.234 ± 8.800	0.996 ± 0.002
DARN	**0.866 ± 0.001**	**0.769 ± 0.001**	**0.645 ± 0.002**	**0.705 ± 0.003**	**0.758 ± 0.002**	**0.956 ± 0.003**	**51.456 ± 5.200**	**0.998 ± 0.001**

to generalize for regression tasks, and its performance is marked as " – ". Additionally, some entries are labeled as " – " when the runtime exceeds 10^5 seconds. Since all baselines achieve nearly the best results using F-GAIN for imputation, as shown in Table 3, we exclusively use F-GAIN for imputation across all scenarios.

We have the following observations. First, it can be observed that DARN consistently outperforms all baseline methods, achieving higher prediction performance (*i.e.,* Accuracy, AUC, and R2) and lower errors (*i.e.,* RMSE). Specifically, DARN achieves an average improvement of 36.91%, 34.32%, 26.72%, 25.80%, 40.85%, and 28.20% for BC, CC, PC, SSC, CR and MMM scenarios, respectively, across all metrics and datasets. These imputation-based methods exhibit poor performance because the introduction of imputation techniques can introduce bias, potentially compromising the final prediction accuracy. Although F-XGBoost can handle incomplete tabular data directly, its performance is suboptimal, particularly on large datasets (*e.g.,* Covertype and Gas). This is because large datasets tend to increase the depth of individual trees, causing the model to fit finer details of the data and become more prone to overfitting. In contrast, DARN excels due to its effective capture of each client's unique missing data distribution and the use of a complementary missing distribution to construct personalized federated prediction models directly, without relying on estimated values. It also has advantages for large datasets due to the increased availability of observed information.

Second, DARN outperforms two variants of centralized learning-based DARN, *i.e.,* Central-DARN and Local-DARN. For example,



(a) Efficiency evaluation (b) Ablation study

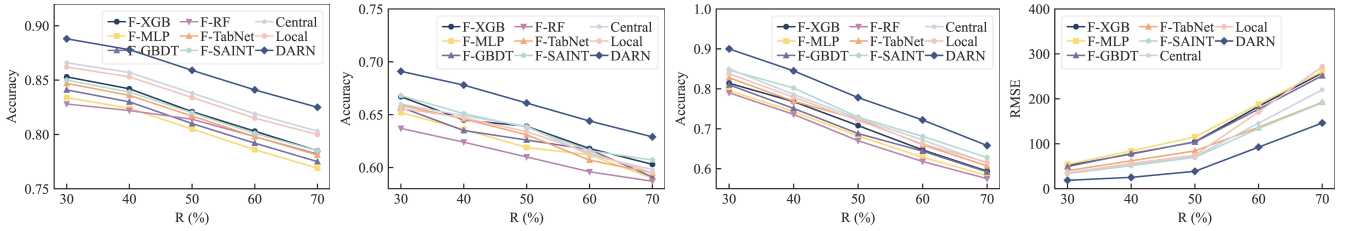
Figure 4: The efficiency evaluation and ablation study.

in the BC scenario, as shown in Table 3, DARN achieves an average performance improvement of 42.43% over Central-DARN and 40.77% over Local-DARN. This is because DARN is designed for a distributed framework and calculates personalized weights for each client based on the similarity of pairs of clients' missing distributions and their observed sample sizes. However, Central-DARN processes all data through a single centralized node, treating all incomplete tabular data samples as equally important for the prediction model. The Local-DARN is similar to Central-DARN but with fewer training samples. It demonstrates that our proposed method is explicitly designed for collaborative scenarios involving missing complementarity.

Third, DARN exhibits a decreasing advantage in prediction performance as the level of missing pattern complementarity decreases. In scenarios with high to moderate complementarity (*i.e.,* BC, CC,

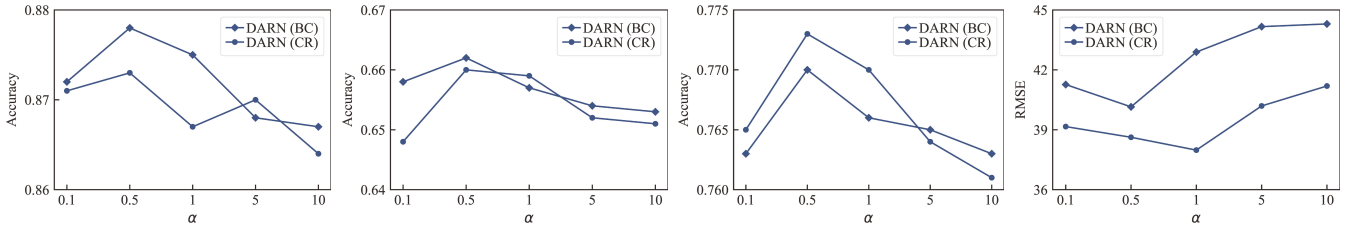
Table 9: The prediction performance under the non-IID setting.

Scenarios	BC		CC		PC		SSC		CR	
Models	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑
F-XGBoost	0.685 ± 0.004	0.903 ± 0.003	0.672 ± 0.002	0.891 ± 0.002	0.663 ± 0.003	0.881 ± 0.001	0.655 ± 0.005	0.872 ± 0.004	0.678 ± 0.003	0.898 ± 0.002
F-GBDT	0.674 ± 0.005	0.892 ± 0.004	0.662 ± 0.003	0.880 ± 0.003	0.652 ± 0.002	0.872 ± 0.002	0.645 ± 0.004	0.863 ± 0.003	0.669 ± 0.004	0.887 ± 0.003
F-RF	0.656 ± 0.002	0.873 ± 0.002	0.645 ± 0.003	0.861 ± 0.002	0.634 ± 0.001	0.853 ± 0.001	0.627 ± 0.002	0.844 ± 0.002	0.650 ± 0.002	0.868 ± 0.002
F-MLP	0.665 ± 0.003	0.883 ± 0.002	0.654 ± 0.002	0.871 ± 0.003	0.646 ± 0.001	0.863 ± 0.001	0.638 ± 0.003	0.854 ± 0.002	0.659 ± 0.003	0.878 ± 0.002
F-TabNet	0.694 ± 0.002	0.911 ± 0.003	0.683 ± 0.003	0.899 ± 0.002	0.674 ± 0.002	0.891 ± 0.001	0.666 ± 0.001	0.882 ± 0.002	0.688 ± 0.002	0.906 ± 0.003
F-SAINT	0.709 ± 0.003	0.925 ± 0.002	0.698 ± 0.002	0.913 ± 0.003	0.689 ± 0.001	0.905 ± 0.002	0.681 ± 0.002	0.896 ± 0.002	0.703 ± 0.003	0.920 ± 0.003
Central-DARN	0.705 ± 0.003	0.920 ± 0.002	0.698 ± 0.002	0.910 ± 0.003	0.690 ± 0.002	0.902 ± 0.002	0.682 ± 0.003	0.893 ± 0.002	0.703 ± 0.003	0.918 ± 0.002
Local-DARN	0.701 ± 0.002	0.917 ± 0.003	0.694 ± 0.003	0.907 ± 0.002	0.686 ± 0.002	0.899 ± 0.002	0.678 ± 0.002	0.890 ± 0.002	0.699 ± 0.002	0.915 ± 0.003
DARN	**0.742 ± 0.001**	**0.952 ± 0.001**	**0.732 ± 0.002**	**0.940 ± 0.002**	**0.723 ± 0.001**	**0.932 ± 0.001**	**0.715 ± 0.001**	**0.923 ± 0.001**	**0.737 ± 0.001**	**0.947 ± 0.001**



(a) Bank (b) Higgs (c) Covertype (d) Gas

Figure 5: The prediction performance of tabular data prediction algorithms vs. missing rate R.



(a) Bank (b) Higgs (c) Covertype (d) Gas

Figure 6: The prediction performance of DARN vs. weight hyperparameter α.

and PC scenarios), DARN significantly outperforms baseline methods and improves slightly in the low complementarity scenario (*i.e.,* SSC scenario). Even in the CR scenario and practical setting (*i.e.,* MMM scenario) with random missing patterns or missing mechanisms for each client, DARN performs well, as missing complementarity is also present in these scenarios to some extent. By calculating the personalized weight, DARN can further leverage the observed samples to their maximum potential instead of imputation. Considering that low and moderate levels of missing complementarity are likely to occur frequently in real-world scenarios, DARN may demonstrate higher practical applicability.

Finally, DARN also remains effective in the non-IID setting. The results on the Covertype dataset in Table 9 show that DARN consistently achieves the highest prediction performance compared to all other baseline methods across all scenarios. As an example, in the BC scenario, the average test accuracy of DARN is 4.65% higher than that of the best-performing baseline F-SAINT. This is because F-XGBoost and imputation-based methods face greater challenges in the non-IID setting. The skewed label distribution can introduce processing bias or additional imputation errors, further degrading prediction performance. In contrast, DARN aims to directly represent incomplete data and leverage the observed information to its fullest extent, without introducing additional biases due to the heterogeneous distribution.

Efficiency. Since FL involves multiple communication rounds with all clients, efficiency is essential when training personalized models to handle incomplete data. We compare the runtime of DARN with these deep learning-based methods to assess the efficiency of our approach, as our method falls within this category. These deep learning methods all employ F-GAIN for handling incomplete data, which is the best-performing federated imputation algorithm discussed earlier. The average runtime for each method across all scenarios and datasets is reported in Figure 4(a). The results indicate that DARN demonstrates competitive (or even lower) runtime compared to these deep learning methods. This is because these deep learning-based methods (F-TabNet and F-SAINT), which combine prediction with imputation, incur significant computational overhead. An exception is F-MLP, which demonstrates relatively high efficiency due to its lower model complexity; however, its performance is not comparable to that of attention-based models. In contrast, DARN only requires fitting an additional logistic regression model to learn the missing data distribution for calculating personalized weights. Its runtime for each dataset is comparable to that of Central-DARN and Local-DARN. These results demonstrate that our model offers significantly higher efficiency, making it better suited to meet the demands of real-world applications.

5.3 Ablation Study

We conduct ablation studies to evaluate the effectiveness of different components in DARN under the non-IID setting using the following four strategies. The experimental results on the Covertype dataset, focusing on BC and CR scenarios, are presented in Figure 4(b).

Table 10: The performance of DARN vs. noise λ.

Datasets	Covertype		Bank	
λ	Accuracy ↑	AUC ↑	Accuracy ↑	AUC ↑
0	**0.770 ± 0.002**	**0.967 ± 0.001**	**0.878 ± 0.001**	**0.781 ± 0.001**
0.1	0.766 ± 0.002	0.962 ± 0.001	0.874 ± 0.002	0.777 ± 0.001
0.2	0.762 ± 0.003	0.958 ± 0.003	0.870 ± 0.001	0.773 ± 0.002
0.3	0.759 ± 0.002	0.956 ± 0.003	0.867 ± 0.002	0.770 ± 0.001
0.4	0.755 ± 0.001	0.953 ± 0.001	0.863 ± 0.001	0.766 ± 0.002
0.5	0.756 ± 0.001	0.955 ± 0.001	0.864 ± 0.001	0.767 ± 0.001

- ***w/o C***: This strategy does not consider the observed data size score when calculating the personalized weight.
- ***w/o MAT***: This strategy uses a traditional transformer block instead of the missing-aware transformer for both the prediction and imputation models. As a result, they do not incorporate a missing-aware attention mechanism and instead rely on the values estimated by the imputation model.
- ***w/o REC***: This strategy does not incorporate reconstruction error during the local model update process.
- ***w/o PWC***: This strategy does not employ the personalized weight calculation approach. Instead, it directly utilizes FedAVG for parameter updates.

The results indicate that the observed data size score, the missing-aware transformer block, reconstruction loss, and the personalized weight calculation strategy all positively influence prediction performance. Specifically, the average performance decreases by 2.19%, 2.04%, 3.20%, and 3.29%, respectively, when these components are removed. It highlights that the personalized weight calculation strategy contributes most significantly to DARN.

5.4 Parameter Evaluation

Effect of R. We investigate the robustness of DARN against a range of missing rates, from 30% to 70%. From the results on the four datasets under the CR scenario, as shown in Figure 5, we observe that DARN consistently outperforms all baseline methods. As the missing rate R increases, the prediction accuracy decreases for all methods due to the reduced availability of observed data. Notably, the prediction accuracy of DARN decreases at a much slower rate. This advantage can be attributed to the missing distribution learning module, which is a strength in the missing-aware transformer block. Additionally, the personalized weight averaging strategy optimizes the use of the available data. As a result, DARN effectively mitigates the adverse effects of higher missing rates.

Effect of α. We investigate how DARN performs with different weights for the reconstruction loss in Eq. 2. The parameter α is designed to train these three models in a balanced manner, while also partially mitigating reconstruction errors. Figure 6 illustrates the impact of the hyperparameter α on the performance of DARN on the four datasets under BC and CR scenarios. As shown, DARN achieves its optimal performance, indicated by higher accuracy or lower RMSE, when α is set to 0.5.

Effect of λ. We validate the privacy-utility tradeoff of DARN enhanced with the DP technique, as defined in Eq. 10. Specifically, we vary the Laplacian noise strength added to the shared parameters, denoted by $\lambda = \Delta f/\epsilon$, from 0 to 0.5 in increments of 0.1, and conduct a set of experiments on the Bank and Covertype datasets under the BC scenario. Tuning the noise intensity allows control of privacy protection strength, with higher noise levels offering stronger privacy guarantees. As shown in Table 10, performance deteriorates as the noise strength λ increases, though the degradation remains minimal when λ is not excessively large. This demonstrates that our method not only provides robust privacy protection but also meets the essential security requirements for practical deployment.

Table 11: The performance over real-world datasets.

Datasets	*Beers*		*Mobility*	
Models	Accuracy ↑	AUC ↑	RMSE ↓	R2 ↑
F-XGBoost	0.228	0.605	81.04	0.539
F-GBDT	0.220	0.591	84.75	0.521
F-RF	0.208	0.586	88.43	0.504
F-MLP	0.222	0.594	78.12	0.546
F-TabNet	0.231	0.611	74.39	0.557
F-SAINT	0.238	0.629	70.83	0.553
Central-DARN	0.245	0.646	66.72	0.571
Local-DARN	0.237	0.631	71.57	0.555
DARN	**0.253**	**0.648**	**63.41**	**0.587**

5.5 Case Study

We further verify the superiority of the DARN on two *real-world incomplete datasets*—the Beers dataset and the Mobility dataset. Specifically, the Beers dataset is a real-world dataset sourced through web scraping and manually cleaned by the dataset owner, with an average missing rate of 16%. It is a multi-class classification task involving the prediction of 16 different beer styles. The Mobility dataset, which tracks COVID-19 community mobility, shows how the length of stay at various locations changes relative to a baseline in a specific region, with an average missing rate of 30.62%. It is a regression task to predict the number of new cases confirmed after a positive test. From Table 11, we can observe that our method consistently outperforms baseline approaches on two real-world datasets, achieving an average improvement of 3.92% on the Beers dataset and 10.35% on the Mobility dataset. The results not only demonstrate that missing complementarity exists to a certain degree in real-world incomplete datasets, but also confirm the robustness and applicability of DARN in real-world scenarios.

6 CONCLUSION

In this paper, we introduce a novel federated prediction framework for incomplete tabular data, called DARN. This framework leverages missing complementarity to construct personalized federated prediction models without relying on imputed values. Each client trains a model to learn the unique missing data distribution and uploads its parameters, along with the observed sample size. The central server calculates missing complementarity scores and observed sample size scores to determine the personalized weights for the prediction models. Furthermore, we present a missing-aware transformer block to represent incomplete tabular data accurately. We incorporate differential privacy techniques into DARN to enhance privacy. Extensive experiments on four publicly available real-world datasets and two real-world incomplete datasets verify the superiority and robustness of DARN.

ACKNOWLEDGMENTS

This work was supported by the Leading Goose R&D Program of Zhejiang (No. 2024C01109), the NSFC (No. 62372404), and the Fundamental Research Funds for the Central Universities (No. 226-2024-00030). Xiaoye Miao is the corresponding author of the work.

REFERENCES

[1] 2018. General data protection regulation. *Intouch* 25 (2018), 1–5.

[2] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Benjamin J. Lengerich, Rich Caruana, and Geoffrey E. Hinton. 2021. Neural additive models: Interpretable machine learning with neural nets. In *NeurIPS*. 4699–4711.

[3] Naomi S Altman. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician* 46, 3 (1992), 175–185.

[4] Sercan Ö Arik and Tomas Pfister. 2021. TabNet: Attentive interpretable tabular learning. In *AAAI*, Vol. 35. 6679–6687.

[5] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. 2014. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications* 5, 1 (2014), 4308.

[6] Jingjing Bao, Celimuge Wu, Yangfei Lin, Lei Zhong, Xianfu Chen, and Rui Yin. 2023. A scalable approach to optimize traffic signal control with federated reinforcement learning. *Scientific Reports* 13, 1 (2023), 19184.

[7] Jock Blackard. 1998. Covertype. UCI Machine Learning Repository.

[8] Changgee Chang, Yi Deng, Xiaoqian Jiang, and Qi Long. 2020. Multiple imputation for analysis of incomplete data in distributed health data networks. *Nature Communications* 11, 1 (2020), 5467.

[9] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *SIGKDD*. 785–794.

[10] Adriana Fonseca Costa, Miriam Seoane Santos, Jastin Pompeu Soares, and Pedro Henriques Abreu. 2018. Missing data imputation via denoising autoencoders: The untold story. In *IDA*. 87–98.

[11] Aymeric Dieuleveut, Gersende Fort, Eric Moulines, and Geneviève Robin. 2021. Federated-EM with heterogeneity mitigation and variance reduction. In *NeurIPS*. 29553–29566.

[12] Shaoming Duan, Chuanyi Liu, Peiyi Han, Tianyu He, Yifeng Xu, and Qiyuan Deng. 2022. Fed-TDA: Federated tabular data augmentation on Non-IID data. *CoRR* abs/2211.13116 (2022).

[13] Alireza Farhangfar, Lukasz A. Kurgan, and Witold Pedrycz. 2007. A novel framework for imputation of missing values in databases. *IEEE Trans. Syst. Man Cybern. Part A* 37, 5 (2007), 692–709.

[14] Alireza Farhangfar, Lukasz A Kurgan, and Witold Pedrycz. 2007. A novel framework for imputation of missing values in databases. *IEEE Trans. Syst. Man Cybern. Part A* 37, 5 (2007), 692–709.

[15] Raquel Florez-Lopez. 2010. Effects of missing data in credit risk scoring. A comparative analysis of methods to achieve robustness in the absence of sufficient data. *J. Oper. Res. Soc.* 61, 3 (2010), 486–501.

[16] Fangcheng Fu, Yingxia Shao, Lele Yu, Jiawei Jiang, Huanran Xue, Yangyu Tao, and Bin Cui. 2021. VF^{2}Boost: Very fast vertical federated gradient boosting for cross-enterprise learning. In *SIGMOD*. 563–576.

[17] Dawei Gao, Daoyuan Chen, Zitao Li, Yuexiang Xie, Xuchen Pan, Yaliang Li, Bolin Ding, and Jingren Zhou. 2023. FS-Real: A real-world cross-device federated learning platform. *Proc. VLDB Endow.* 16, 12 (2023), 4046–4049.

[18] Geoffrey R Gerdes and Xuemei Liu. 2019. Improving response quality with planned missing data: An application to a survey of banks. In *The Econometrics of Complex Survey Data*. Vol. 39. 237–258.

[19] Achmad Ginanjar, Xue Li, and Wen Hua. 2024. Contrastive federated learning with tabular data silos. *arXiv preprint arXiv:2409.06123* (2024).

[20] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. In *NeurIPS*. 18932–18943.

[21] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. 2022. Why do tree-based models still outperform deep learning on typical tabular data?. In *NeurIPS*, Vol. 35. 507–520.

[22] Xinrui He, Shuo Liu, Jacky Keung, and Jingrui He. 2024. Co-clustering for federated recommender system. In *WWW*. 3821–3832.

[23] Daniel F Heitjan and Srabashi Basu. 1996. Distinguishing "missing at random" and "missing completely at random". *The American Statistician* 50, 3 (1996), 207–213.

[24] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335* (2019).

[25] Niels Bruun Ipsen, Pierre-Alexandre Mattei, and Jes Frellsen. 2021. Not-MIWAE: Deep generative modelling with missing not at random data. *arXiv preprint arXiv:2006.12871* (2021).

[26] Burgu Javier. 2019. https://archive.ics.uci.edu/ml/datasets/Gas+sensor+array+temperature+modulation. (2019).

[27] Trevor Kirby. 2023. An evolution in the business of banking: The neobank partnership. *Loy. U. Chi. J. Reg. Compl.* 11 (2023), 28.

[28] Pengyu Li, Chengwei Guo, Yanxia Xing, Yingji Shi, Lei Feng, and Fanqin Zhou. 2024. Core network traffic prediction based on vertical federated learning and split learning. *Scientific Reports* 14, 1 (2024), 4663.

[29] Qinbin Li, Zeyi Wen, and Bingsheng He. 2020. Practical federated gradient boosting decision trees. In *AAAI*. 4642–4649.

[30] Xiaochen Li, Yuke Hu, Weiran Liu, Hanwen Feng, Li Peng, Yuan Hong, Kui Ren, and Zhan Qin. 2022. OpBoost: A vertical federated tree boosting framework based on order-preserving desensitization. *Proc. VLDB Endow.* 16, 2 (2022), 202–215.

[31] Roderick J Little. 2021. Missing data assumptions. *Annual Review of Statistics and Its Application* 8, 1 (2021), 89–107.

[32] Samuel Maddock, Graham Cormode, Tianhao Wang, Carsten Maple, and Somesh Jha. 2022. Federated boosted decision trees with differential privacy. In *CCS*. 2249–2263.

[33] Paul Joe Maliakel, Shashikant Ilager, and Ivona Brandic. 2024. FLIGAN: Enhancing federated learning with incomplete data using GAN. In *EdgeSys*. 1–6.

[34] Pierre-Alexandre Mattei and Jes Frellsen. 2019. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *ICML*, Vol. 97. 4413–4423.

[35] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*. 1273–1282.

[36] Xiaoye Miao, Yangyang Wu, Lu Chen, Yunjun Gao, and Jianwei Yin. 2023. An experimental survey of missing data imputation algorithms. *IEEE Trans. Knowl. Data Eng.* 35, 7, 6630–6650.

[37] Sitao Min, Hafiz Asif, Xinyue Wang, and Jaideep Vaidya. 2025. Cafe: Improved federated data imputation by leveraging missing data heterogeneity. *IEEE Trans. Knowl. Data Eng.* 37, 5 (2025), 2266–2281.

[38] Rita P. Moro, S. and P. Cortez. 2014. Bank marketing. UCI Machine Learning Repository.

[39] Wei Ni, Xiaoye Miao, Xiangyu Zhao, Yangyang Wu, Shuwei Liang, and Jianwei Yin. 2024. Automatic data repair: Are we ready to deploy? *Proc. VLDB Endow.* 17, 10 (2024), 2617–2630.

[40] Stuart L Pardau. 2018. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J. Tech. L. & Pol'y* 23 (2018), 68.

[41] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus H. Maier-Hein, Sébastien Ourselin, Micah J. Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. 2020. The future of digital health with federated learning. *NPJ Digit. Medicine* 3, 1 (2020), 119.

[42] Donald B Rubin. 1976. Inference and missing data. *Biometrika* 63, 3 (1976), 581–592.

[43] Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. 2021. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342* (2021).

[44] Xiaoqing Tan, Chung-Chou H. Chang, Ling Zhou, and Lu Tang. 2022. A tree-based model averaging approach for personalized treatment effect estimation from heterogeneous data sources. In *ICML*, Vol. 162. 21013–21036.

[45] Oscar Wahltinez, K Murphy, M Brenner, et al. 2020. COVID-19 open-data: Curating a fine-grained, global-scale data repository for SARS-CoV-2. 2020. *Work in progress* (2020).

[46] Qinyong Wang, Hongzhi Yin, Tong Chen, Junliang Yu, Alexander Zhou, and Xiangliang Zhang. 2022. Fast-adapting and privacy-preserving federated recommender system. *VLDB J.* 31, 5 (2022), 877–896.

[47] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans. Inf. Forensics Secur.* 15 (2020), 3454–3469.

[48] Yuncheng Wu, Shaofeng Cai, Xiaokui Xiao, Gang Chen, and Beng Chin Ooi. 2020. Privacy preserving vertical federated learning for tree-based models. *Proc. VLDB Endow.* 13, 11 (2020), 2090–2103.

[49] Yuncheng Wu, Naili Xing, Gang Chen, Tien Tuan Anh Dinh, Zhaojing Luo, Beng Chin Ooi, Xiaokui Xiao, and Meihui Zhang. 2023. Falcon: A privacy-preserving and interpretable vertical federated learning system. *Proc. VLDB Endow.* 16, 10 (2023), 2471–2484.

[50] Jing Xia, Shengyu Zhang, Guolong Cai, Li Li, Qing Pan, Jing Yan, and Gangmin Ning. 2017. Adjusted weight voting algorithm for random forests in handling missing values. *Pattern Recognit.* 69 (2017), 52–60.

[51] Wei Xu, Hui Zhu, Yandong Zheng, Fengwei Wang, Jiaqi Zhao, Zhe Liu, and Hui Li. 2024. ELXGB: An efficient and privacy-preserving XGBoost for vertical federated learning. *IEEE Trans. Serv. Comput.* 17, 3 (2024), 878–892.

[52] Kunda Yan, Sen Cui, Abudukelimu Wuerkaixi, Jingfeng Zhang, Bo Han, Gang Niu, Masashi Sugiyama, and Changshui Zhang. 2024. Balancing similarity and complementarity for federated learning. In *ICML*. 55739–55758.

[53] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. 2007. On early stopping in gradient descent learning. *Constructive Approximation* 26, 2 (2007), 289–315.

[54] Jinsung Yoon, James Jordon, and Mihaela Schaar. 2018. GAIN: Missing data imputation using generative adversarial nets. In *ICML*. 5675–5684.

[55] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. 2020. VIME: Extending the success of self- and semi-supervised learning to tabular domain. In *NeurIPS*, Vol. 33. 11033–11043.

[56] Yazhong Zhang, Hanbing Zhang, Zhenying He, Yinan Jing, Kai Zhang, and X. Sean Wang. 2021. Parrot: A progressive analysis system on large text collections. *Data Sci. Eng.* 6, 1 (2021), 1–19.

[57] Zilong Zhao, Han Wu, Aad van Moorsel, and Lydia Y. Chen. 2023. GTV: Generating tabular data via vertical federated learning. *CoRR* abs/2302.01706 (2023).

[58] Yifeng Zheng, Shuangqing Xu, Songlei Wang, Yansong Gao, and Zhongyun Hua. 2023. Privet: A privacy-preserving vertical federated learning service for gradient boosted decision tables. *IEEE Trans. Serv. Comput.* 16, 5 (2023), 3604–3620.

[59] Yizhen Zheng, He Zhang, Vincent Cheng-Siong Lee, Yu Zheng, Xiao Wang, and Shirui Pan. 2023. Finding the missing-half: Graph complementary learning for homophily-prone and heterophily-prone graphs. In *ICML*. 42492–42505.

[60] Xu Zhou, Xiaofeng Liu, Gongjin Lan, and Jian Wu. 2021. Federated conditional generative adversarial nets imputation method for air quality missing data. *Knowl. Based Syst.* 228 (2021), 107261.