electronics

MDPI

Article

Privacy-Preserving Vertical Federated KNN Feature Imputation Method

Wenyou Du [1,2,*], Yichen Wang [1], Guanglei Meng [1,*] and Yuming Guo [1]

[1] College of Automation, Shenyang Aerospace University, Shenyang 110136, China; wangyichen1@stu.sau.edu.cn (Y.W.); guoyuming108@163.com (Y.G.)
[2] Chinese Academy of Sciences Shenyang Institute of Computing Technology Co., Ltd., Shenyang 110168, China
* Correspondence: wen-you.du@sau.edu.cn (W.D.); mengguanglei@sau.edu.cn (G.M.)

Abstract: Federated learning stands as a pivotal component in the construction of data infrastructure. It significantly fortifies the safety and reliability of data circulation links, facilitating credible sharing and openness among diverse subjects. The presence of missing data poses a pervasive and challenging issue in the implementation of federated learning. Current research on imputation missing values predominantly concentrates on centralized methods and horizontal federation scenarios. However, there is a notable absence of exploration in the context of vertical federated application scenarios. In this paper, the problem of missing imputation in vertical federated learning is investigated and a novel vertical federated k-nearest neighbors (KNN) imputation method is proposed. Extensive experiments are conducted using publicly available data sets to compare existing imputation methods, the results demonstrate the effectiveness and progress of our approach.

Keywords: federated learning; missing data; vertical federated imputation; k-nearest neighbors

check for updates

Citation: Du, W.; Wang, Y.; Meng, G.; Guo, Y. Privacy-Preserving Vertical Federated KNN Feature Imputation Method. *Electronics* **2024**, *13*, 381. https://doi.org/10.3390/electronics13020381

Academic Editor: Myung-Sup Kim

Received: 8 December 2023
Revised: 6 January 2024
Accepted: 15 January 2024
Published: 17 January 2024

1. Introduction

Over the last few years, there has been an escalating demand for the protection of personal data privacy. In response, the European Union, the United States, and China have introduced various relevant policies to make clear provisions on the collection, storage, and use of personal information such as the General Data Protection Regulation (GDPR) implemented by the EU on 25 May 2018 [1] and Consumer Privacy Bill of Rights in the U.S. [2].

At the same time, artificial intelligence technologies, including machine learning, computer vision, natural language processing and deep learning, develop rapidly. These artificial intelligence methods are built on the foundation of big data [3], but as the legal environment is becoming more and more severe, the data cannot be directly traded, and in many cases, the scale of the data obtained before modeling is insufficient to meet training requirements. This may include a smaller amount of data, the absence of labels, or the absence of features, resulting in the formation of the data isolated islands [4]. Privacy computing is an effective way to deal with data isolated islands, and there are three mainstream privacy computing methods: multi-party secure computing [5,6], federated learning, and trusted execution environments [7]. Among them, the concept of federated learning is a machine learning model based on distributed data sets proposed by Google [8–10]. Reference [4] categorizes federated learning into horizontal federated learning, vertical federated learning and federated transfer learning based on the distribution of training data in the data feature space and sample ID space among different parties as showed in Figure 1.

Among them, vertical federated learning is applicable to the cases that data sets of different parties share the same sample ID space but differ in feature space. For diverse business purposes, datasets owned by different enterprises often have distinct feature spaces. Despite the dissimilarities in data features, they come from the same customer

group and show the status of customers from different levels. Using the heterogeneous data distributed across these enterprises, superior machine learning models can be built. Due to data security laws and regulations, direct data sharing among enterprises is precluded. Even intra-enterprise data may not be shared among disparate departments. Consequently, only a few large Internet companies have real "big data", and most small and micro enterprises and small and medium-sized enterprises are faced with the problem of scarce feature dimensions, which restricts the application of artificial intelligence technology. The advent of vertical federated learning resolves the challenge of jointly modeling heterogeneous data across enterprises without necessitating the exchange or compromise of private data. Based on vertical federated learning, many algorithms have been developed to enable federated learning. Reference [11] proposes a novel lossless privacy-preserving tree-boosting system known as SecureBoost in the setting of federated learning. Reference [12] proposes a quasi-Newton method based vertical federated learning framework for logistic regression within the additively homomorphic encryption scheme. References [13,14] improve classical neural network models such as backpropagation neural networks with minimal computation and communication costs while ensuring lossless model accuracy.

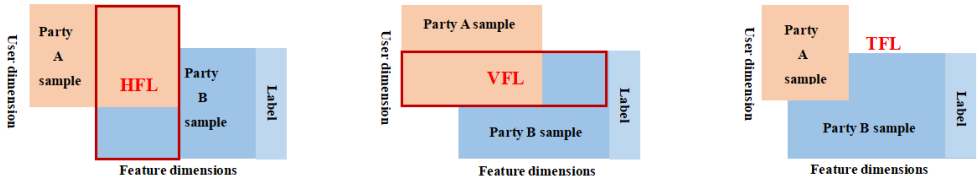

Figure 1. Federated Learning Classification.

Nevertheless, the vertical federated methods mentioned above rely on data sets devoid of missing values. In instances where data sets include missing values, it impedes the modeling process, necessitating the imputation of missing values in the data set. Currently, there are some researches on missing values imputation algorithms in federated learning scenarios. Reference [15] proposes a conditional GAN imputation method under a federated learning framework to solve the data sets that come from diverse data-owners without sharing. Reference [16] proposes a novel federated learning method, a light weight yet effective autoencoder-based model is employed to address the examined problem, modified properly to capture the temporal dependencies of the time series data. On the one hand, all the above methods are horizontal federated missing values imputation methods, but the research on vertical federated missing values imputation methods is still lacking. On the other hand, the current popular federated learning platform such as FATE [17] only provides simple imputation methods such as maximum imputation (MAX), minimum imputation (MIN), and mean imputation (MEAN). These simple imputation methods have limited effect on model improvement. Therefore, it is necessary to study and explore more complex methods suitable for vertical federated missing values imputation.

Compared to federated imputation, centralized imputation methods, which can also be referred to as traditional imputation, contains various of methods. In general, the missing imputation techniques can be categorized into two types, namely, statistical-based imputation techniques and machine learning-based imputation techniques [18,19]. In the area of statistically based imputation techniques, most widely used statistical techniques include MEAN [20], expectation management (EM) [21], linear regression (LR) [22], least squares (LS) [23] and principal component analysis (PCA) [24]. Reference [25] summarizes the fundamental principle of expectation management (EM), linear regression (LR) and least squares (LS) methods. Reference [26] describes normal-model multiple imputation (MI) and maximum likelihood methods. Machine learning-based imputation techniques are developing fast, most widely used machine techniques include k-nearest neighbors (KNN) [27], Random forest (RF) [28], Decision tree (DT) [29] and Clustering [30].

However, all above statistical and machine learning-based imputation algorithms are limited to their own data, and the process of imputation is carried out independently.

The correlation between the data sets of different parties that can bring improvement to data imputation is ignored. Therefore, in this paper, the centralized KNN imputation algorithm is improved and a federated KNN imputation method based on vertical federated learning is proposed. This approach can impute the missing values of all parties' data sets in vertical federated scenarios and encrypt the intermediate results of the transmittal using homomorphic encryption algorithms to protect the privacy and security.

The remainder of this paper is organized as follows. Section 2 shows problem statement and Section 3 provides an introduction to the vertical federated KNN imputation method proposed in this paper, followed by comparative experiments to validate the effectiveness of the method in Section 4. Lastly, conclusions and future works are given in Section 5.

2. Problem Definition

In this paper, it is assumed that multiple parties hold their respective data sets that share the same sample ID space but differ in feature space. Each party is precluded from directly utilizing the data sets of other parties for training the missing data imputation model, primarily owing to concerns related to privacy, data security, and competitive considerations. In this section, the problem definition is divided into two parts to describe the missing data formulation and vertical federated imputation settings.

2.1. Missing Data Formulation

Consider that the data comes from m-dimensional space $\mathbf{F}$. Each sample x_i in the non-missing data set $\mathbf{X} = \{x_1; x_2; ...x_i; ...x_n\}(|\mathbf{X}| = n, x_i \in \mathbb{R}^m)$ is a random variable in $\mathbf{F}$. When it occurs that data is missing, the observed data $\mathbf{X_{obs}}$ contains missing data, which is represented by NaN. Let $\mathbf{M}$ represents the mask matrix, which indicates whether the observed data point in $\mathbf{X}$ exists or misses, if $x_{i,j}$ exists, $m_{i,j}$ is 0 , otherwise is 1. The following is the relationship between $\mathbf{X}$, $\mathbf{M}$ and $\mathbf{X_{obs}}$ with an example:

$$\mathbf{X_{obs_{i,j}}} = \begin{cases} x_{i,j}, & \text{if } m_{i,j} = 0, \\ NaN, & \text{if } m_{i,j} = 1. \end{cases} \tag{1}$$

$$\mathbf{X} = \begin{bmatrix} 4 & 3 & 5 & 7 \\ 15 & 6 & 12 & 20 \\ 9 & 8 & 3 & 9 \end{bmatrix} \tag{2}$$

$$\mathbf{M} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \tag{3}$$

$$\mathbf{X_{obs}} = \begin{bmatrix} 4 & NaN & 5 & 7 \\ 15 & 6 & NaN & 20 \\ 9 & NaN & NaN & 9 \end{bmatrix} \tag{4}$$

Therefore, the relation between $\mathbf{X_{obs}}$, $\mathbf{M}$ and $\mathbf{X}$ can be presented by:

$$\mathbf{X_{obs}} \odot \mathbf{M} = \mathbf{X} \odot \mathbf{M} \tag{5}$$

where $\odot$ denotes element-wise multiplication.

2.2. Vertical Federated Imputation Settings

Suppose that there are N_p data holders or training parties $\mathbf{P} = \{P_1, \ldots, P_{N_p}\}$ that are not allowed to use the other data holder's data sets directly to train an imputation model. Consider that the samples of data sets are intersected, and the alignment of the samples corresponds by ID numbers. Sample alignment by ID yields n intersected samples. The features of the data are different for each party, so the data dimensions may be different.

The data quality of different holders is diverse and when data missing occurs, there may be different degrees of data missing in each party. The objective of vertical federated imputation is to collaborate all participants' feature spaces and produces a collaborated imputation model without leaking privacy data.

3. Vertical Federated k-Nearest Neighbors Feature Imputation

In this section, the proposed method will be described in detail. The first part introduces the basic principle of the centralized KNN imputation algorithm. The second part discusses the application scenarios and system architecture. The third part presents the KNN imputation algorithm under a federated learning architecture and outlines the specific execution steps for missing values imputation. The last part describes the implementation process.

3.1. Fundamentals

The basic principle of the centralized KNN imputation algorithm is introduced first, as the inspiration for the proposed method primarily stems from it. The KNN imputation algorithm can be defined as the process of identifying K samples in each data set that are spatially similar or closely located by measuring distances. These K samples are subsequently utilized to impute the values of the missing data points. When encountering missing features, the distance is computed by disregarding the missing features and assigning greater importance to the non-missing features.

Considering the distance between two samples x_p and x_q, each potentially containing missing values, is defined as:

$$d(x_p, x_q) = \frac{1}{S}\sum_{s=1}^{S}(x_{p,s} - x_{q,s})^2, \mid \mathbf{M}(p,s) \neq 1 \ \ \& \ \ \mathbf{M}(q,s) \neq 1, \tag{6}$$

where s represents non-missing features in both x_p and x_q satisfying the condition behind the $\mid$. S represents the total number.

Assuming the o-th feature of the p-th sample $x_{p,o}$ is missing, let $x_{k,o}^K$ represents the K nearest samples without missing the o-th feature. Then, the missing value of $x_{p,o}$ can be imputed using Equation (7):

$$\hat{x}_{p,o} = \frac{\sum_{k=1}^{K} x_{k,o}^K}{K} \tag{7}$$

3.2. Architecture

Based on the typical vertical federated learning architecture, the vertical federated imputation architecture is designed as shown in Figure 2, where the vertical federated modeling of two parties is taken as an example.

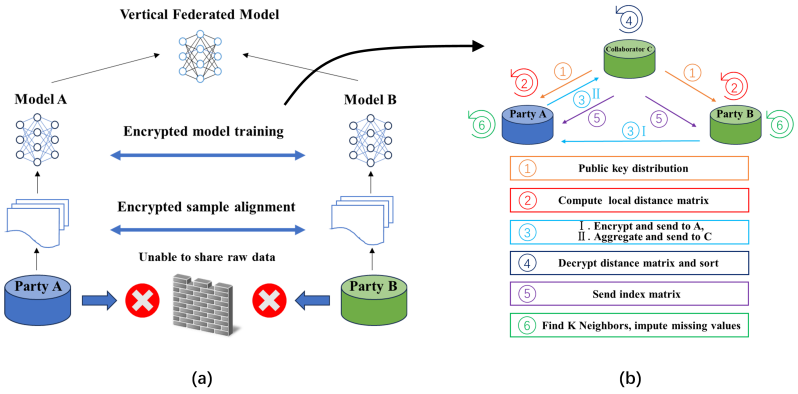

Figure 2. Architecture of vertical federated k-nearest neighbors imputation method. (**a**) Vertical federated learning framework. (**b**) Specific implementation of encrypted model training.

The architecture of federated learning is shown in Figure 2a, suppose that Party A and Party B, namely guest and host, want to train a vertical federated model using their respective data, both Party A's data and Party B's data contains some different features with the same sample ID. Both parties' data sets contain missing values, so the missing feature values need to be imputed before federated modeling such as vertical linear regression, vertical secure boosting tree etc. For data privacy and security reasons, Party A and Party B cannot exchange data directly. Figure 2b is the specific implementation of encrypted model training in Figure 2a. The equations and symbols are explained in the next section.

3.3. Algorithm

In order to ensure the confidentiality of the data, a trusted third-party collaborator C is introduced to coordinate the intermediate interaction results, and party C can be played by an authoritative organization such as the government or replaced by secure computing node such as Intel Software Guard Extensions [31]. The federated KNN imputation algorithm executes the following six steps:

Step 1: Collaborator C generates public key *public_key* and private key *private_key* of homomorphic encryption, distributes *public_key* to the parties.

Step 2: Each party computes the local distance matrix $\mathbf{D}_{P_i}$:

$$\mathbf{D}_{P_i}(p,q) = d(x_p, x_q), P_i \in \mathbf{P}, \mathbf{D} \in \mathbb{R}^{n \times n} \tag{8}$$

The property of symmetry of the distance matrix is utilized to improve computational efficiency. According to Equation (6), The elements at symmetric positions in the upper and lower triangular matrices have equal values, i.e.:

$$d(x_p, x_q) = d(x_q, x_p) \tag{9}$$

Therefore, calculations are performed only for the elements in the upper triangular matrix, reducing computational workload. The calculation of different elements can be conducted independently, allowing for parallel computation to further enhance computational efficiency.

Step 3: Each party encrypts $\mathbf{D}_{P_i}$ to obtain a local encryption distance matrix $[\![\mathbf{D}_{P_i}]\!]$ using *public_key*, where the $[\![\cdot]\!]$ represents homomorphic encryption. Then, each party transmits $[\![\mathbf{D}_{P_i}]\!]$ to party A, A aggregates all parties' local encryption distance matrix to obtain the global encryption distance matrix $[\![\mathbf{D}_{com}]\!]$ and transmits it to C:

$$[\![\mathbf{D}_{com}]\!] = \sum_{i=1}^{N_p} [\![\mathbf{D}_{P_i}]\!], P_i \in \mathbf{P} \tag{10}$$

Step 4: Collaborator C uses the private key to decrypt $[\![\mathbf{D}_{com}]\!]$ to obtain the global distance matrix $\mathbf{D}_{com}$:

$$\mathbf{D}_{com} = private\_key([\![\mathbf{D}_{com}]\!]) \tag{11}$$

Then, C takes the square root of each value in the $\mathbf{D}_{com}$, sorts the results by row in descending order to obtain the index matrix $\mathbf{D}_{index}$ and transmits $\mathbf{D}_{index}$ to each party, where $sort()$ represents the sorting of row vectors in descending order and return the index:

$$\mathbf{D}_{index} = sort(\sqrt{\mathbf{D}_{com}}) \tag{12}$$

During sorting, the distance of the current sample to itself is not taken into consideration.

Step 5: Collaborator C sends $\mathbf{D}_{index}$ to each party.

Step 6: Each party computes the mean and imputes missing value. Obtain the missing locations (p, o) from mask $\mathbf{M}$ where the value is 1, and for each missing feature $x_{p,o}$, compute the mean of the feature values of the corresponding K-nearest neighbors samples without missing the o-th feature and replace NaN values with the mean:

$$\hat{\mathbf{X}}_{\mathbf{obs}}(p,o) = \frac{1}{K}\sum_{\lambda=1}^{K} \mathbf{X}_{\mathbf{obs}}(\mathbf{D}_{index}(p,\lambda),o), \mid \mathbf{M}(\mathbf{D}_{index}(p,\lambda),o) \neq 1, \tag{13}$$

where λ satisfies the condition behind $\mid$. The steps are summarized as Algorithm 1.

Algorithm 1 Federated process. $\mathbf{P}$ represents party set, $[\![\cdot]\!]$ represents homomorphic encryption algorithm, Decry() represents decryptsion algorithms, Sort() represents distance sorting.

Party executes:
for each *party* $n \in \mathbf{P}$ **do**
  for each $x_p, x_q \in \mathbf{X_{obs}}$ **do**
    $\mathbf{D} \leftarrow \mathbf{D}(p,q) = d(x_p, x_q)$
  end for
  for $\mathbf{D}(p,q) \in \mathbf{D}$ **do**
    $[\![\mathbf{D}]\!] \leftarrow [\![\mathbf{D}(p,q)]\!]$
  end for
end for
$[\![\mathbf{D}_{com}]\!] \leftarrow$ Aggregate all $[\![\mathbf{D}]\!]$
Transmit $[\![\mathbf{D}_{com}]\!]$, Receive $\mathbf{D}_{index}$
for each $x_{p,o}$(i.e., $\mathbf{X_{obs}}(p,o)$), *where* $Mask_{po} = 1$ **do**
  Impute using Equation (13)
end for
Coordinator executes:
Receive $[\![\mathbf{D}_{com}]\!]$
$\mathbf{D}_{com} \leftarrow Decry([\![\mathbf{D}_{com}]\!])$
$\mathbf{D}_{index} \leftarrow Sort(\mathbf{D}_{com})$
Transmit $\mathbf{D}_{index}$

The contributions of this algorithm can be summarized as follows. It specifically addresses missing value imputation in a vertical federated learning setting. In such scenarios, multiple parties collaborate to perform computations while keeping their data locally stored and secure. The method's capability to impute missing values across all parties' datasets is also a crucial contribution as it can help each party to enrich its own feature space.

3.4. Implementation

Figure 3 gives an implementation of the above algorithm with an example of two-party federated modeling.

In terms of the complexity of the algorithm, it is illustrated in terms of time and space. The time complexity of the algorithm is $O(n^2 \log_2 n)$, which can be mainly broken up as (1) computing the distance matrix $\mathbf{D}$, the complexity is $\frac{n(n-1)}{2}$, (2) encrypting the distance matrix $\mathbf{D}$, which is n^2, (3) decrypting the distance matrix $[\![\mathbf{D}]\!]$, which is n^2, (4) computing the mean value, impute the missing values, which takes $rn^2 + rn^2 \log_2 n + rn$ time in total. The spatial complexity of the algorithm is $O(n^2)$. The user must store the distance matrix $\mathbf{D}$, encryption distance matrix $[\![\mathbf{D}]\!]$, and the index matrix $\mathbf{D}_{index}$, which are $O(n^2)$ in total.

The work on data privacy protection can be summarized into three aspects. Firstly, the calculation of distance matrix is conducted locally by each party, ensuring that the raw data remain within the local environment, and the original data of each party remain opaque. Secondly, the choice of distance matrix as intermediate exchange results helps prevent reverse inference and untrustworthy behavior by participating parties. Finally, utilizing homomorphic encryption for calculations ensures that, while obtaining the results of distance ranking, any party is unable to decrypt and access the raw data or even the distance matrix of other parties.

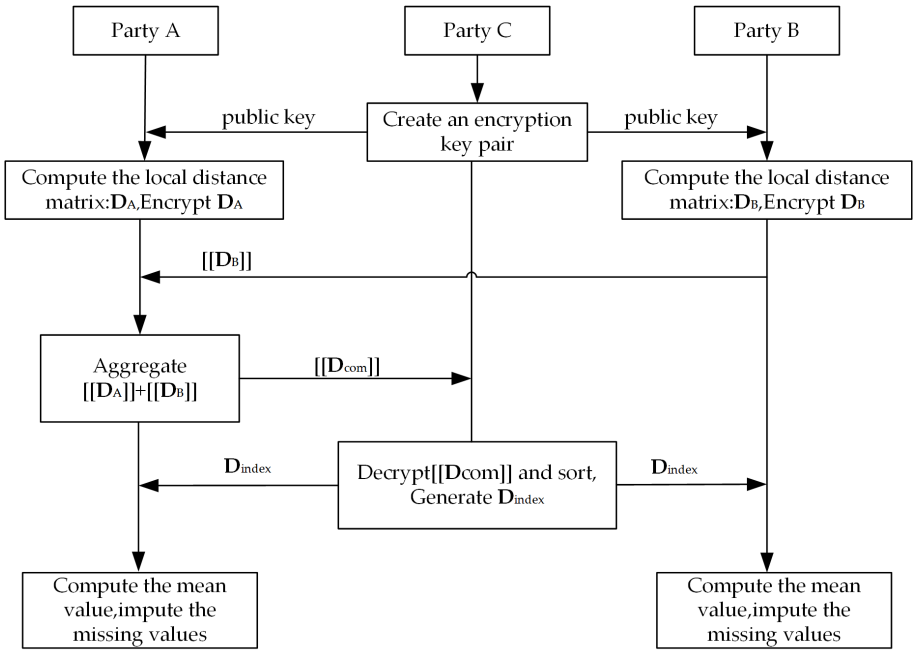

Figure 3. The process of federated KNN imputation algorithm.

The involvement of Coordinator C is optional. If the intermediate result of the exchange does not involve privacy or if the participant is semi-honest, Coordinator C is not required. However, if privacy is implicated, Coordinator C becomes necessary. The challenge lies in balancing privacy protection and efficiency. Striving for extreme security compromises learning efficiency or renders it impractical. Conversely, an unwavering pursuit of efficiency may result in privacy breaches. Thus, the decision to employ Coordinator C hinges on striking a balance between safety and efficiency. If the distance matrix is considered non-sensitive, it can be transmitted between parties using RSA encryption without the need for homomorphic encryption. This approach reduces communication frequency and, consequently, minimizes the time required. The participants can also include multiple guest parties.

4. Experimental Section

4.1. Environment

In this section, all the experiments are performed based on an open-source federated learning framework namely FATE [17]. This framework can be deployed using Kubernetes, docker compose or the standalone way, which are suitable for production environment, model development environment and algorithm development environment, respectively. As the purpose of this paper is to develop a new imputation algorithm, FATE is deployed in the standalone way. The 1.10 version of FATE is deployed in CentOS 7 system. Fateboard is also deployed to exhibit the result of modeling. Fate-flow is deployed to achieve federated job&task scheduling and monitoring. The programming language is python and the IDE of VS code is used as the main development tool. In the aspect of the hardware, a personal computer with the Intel(R) Core (TM) I5-11400CPU, 16 GB running memory, 512 GB SSD and 6 GB graphics memory is adopted to perform the experiment.

4.2. Data

The permanent Magnet synchronous motor (PMSM) data set is used for missing data imputation experiments, which is provided by the open-source project of FATE. The data set is collected from a PMSM deployed on the test bench and contains one label (speed) and 11 features. Set the two parties of vertical federated learning as party A and party B, namely guest and host, and when this data set is used for vertical federated learning, it is divided into two data sets. The guest has a label and four features, while the host has

seven features. Assuming that the four features owned by the guest cannot predict the speed well, we hope to add the host's data to enhance the model, and during the modeling process, the data from both parties does not leave their respective domains, protecting data privacy. In fact, it is also the biggest motivation of vertical federated learning, where the party with the label does not have enough features to establish an effective model to predict the label, and it is necessary to introduce effective features held by other parties to improve the model. This situation is widely present in fields such as finance and healthcare. Actually, the partitioning of the data set exactly simulates this situation, which can be exhibited by a simple linear regression for example. Perform four-fold cross validation on 800 samples for speed prediction. Using only the four features of the guest party obtains a validation RMSE of 0.709 and adding seven features of the host side obtains a validation RMSE of 0.097, indicating that the features of the host side can effectively improve the model. Therefore, this data is suitable for the application scenarios of vertical federated learning. The partitioned guest and host data sets named "motor_ hetero_ guest.csv" and "motor_ hetero_ host.csv", respectively, can be downloaded [32]. Features and their descriptions are shown in Tables 1 and 2.

Table 1. Data set of Party A.

Features	Description	Features Type
idx	138 h to 185 h of records	int
pm	Permanent magnet temperature (in °C) measured with thermocouples	float
stator_yoke	Stator yoke temperature (in °C) measured with thermocouples	float
stator_tooth	Stator tooth temperature (in °C) measured with thermocouples	float
stator_winding	Stator winding temperature (in °C) measured with thermocouples	float
motor_speed (label)	Motor speed (in rpm)	float

Table 2. Data set of Party B.

Features	Description	Features Type
idx	138 h to 185 h of records	int
ambient	Feed pump flow	float
coolant	Coolant temperature (in °C)	float
u_d	Voltage d-component measurement in dq-coordinates	float
u_q	Voltage q-component measurement in dq-coordinates (in V)	float
torque	Motor torque (in Nm)	float
i_d	Current d-component measurement in dq-coordinates	float
i_q	Current q-component measurement in dq-coordinates	float

4.3. Results

To verify the effectiveness of our method, four comparative experiments are designed. In the first experiment, three statistical imputation techniques provided by the FATE platform are considered for comparison: MAX, MIN, and MEAN. In the second experiment, the proposed vertical federated imputation approach is compared with a typical centralized imputation method. Then, in the third experiment, the lossless performance of the federated method is verified. Finally, the KNN imputation method is applied to the regression task to verify its contribution to regression task.

According to Little and Rubin [25], there are three types of missingness mechanisms that can cause an incomplete data set. They are missing completely at random (MCAR), missing at random (MAR), and not missing at random (NMAR) [33]. Assuming that the missing values in the data set do not depend on either the known values or the missing data, the MCAR mechanism is used for the experiments.

4.3.1. Federated Comparative Experiment

In this experiment, the proposed vertical federated imputation method is compared with the existing missing imputation methods. As far as we have learned, no one has researched about vertical federated imputation method before, and constant value imputation methods such as MAX, MIN, and MEAN are most frequently used in the vertical federated situation currently. So, the above three methods are compared. In the aspect of data missing generation, MCAR mechanism is adopted to generate an incomplete data set with missing rates of 10%, 20%, and 30% randomly. This process is repeated 10 times to obtain the averaged evaluation. For the imputed data, we test the validity by comparing the original values to the imputed values.

The experimental results are as Figure 4, where the results with 10% missing rate appear in the first row, and the remaining rows indicate the evaluation results with 20%, 30% missing rates. The 'missing value index' in Figure 4 refers to the positional index of an element within matrix **M**, signifying the ordinal placement of a value equal to 1.

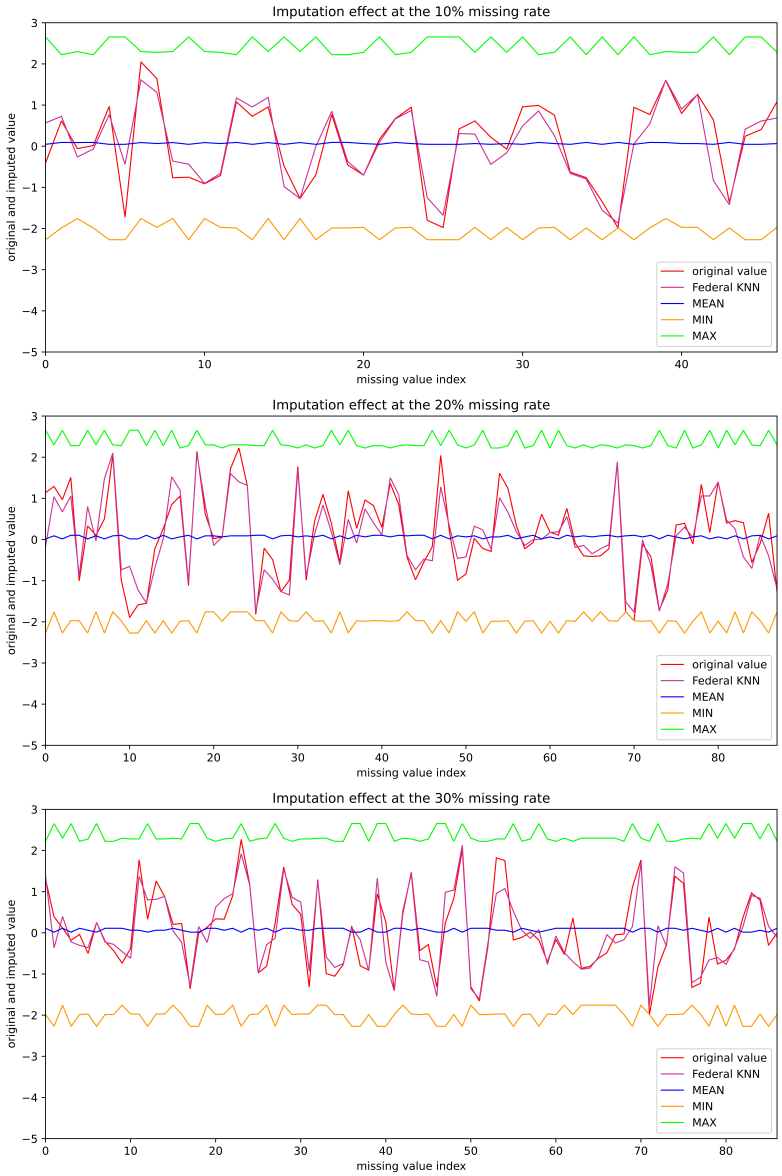

Figure 4. Comparison of original and imputed values across different missing rates.

Overall, among the methods studied, the proposed federated KNN imputation method yields imputed values that are closer to the original distribution of the data. As more values

are removed, the data matrix becomes sparser (i.e., fewer complete values are available for training), which in turn degrades interpolation performance. That is, all techniques tend to use denser data matrices to produce more accurate imputation, which is consistent with previous research [34]. Root mean squared error (RMSE) is selected as the performance metric to further compare the imputation effect of the proposed method with the other three methods. Each experiment is conducted 10 times, results are shown in Table 3 and displayed in Figures 5–7.

Table 3. RMSE performance of the proposed method compared to the benchmarks.

Missing Rate	Federated KNN	MAX	MIN	MEAN
10%	0.3677	2.4136	2.3402	0.9861
20%	0.3358	2.1008	2.1515	0.8188
30%	0.3883	2.5465	2.2252	0.9650

Figure 5. RMSE at the 10% missing rate.

Figure 6. RMSE at the 20% missing rate.

Figure 7. RMSE at the 30% missing rate.

We perform 10 evaluations using different random seeds to remove different parts of the original values. As can be seen from Figures 5–7, the RMSE values of the proposed method are significantly lower than the RMSE values of the other three methods. Among the remaining three methods, MAX and MIN have the highest RMSE values across different missing rates. Compared to MEAN, our method has lower RMSE values across different missing rates.

Constant value imputation methods are to impute all missing values by using fixed values that meet certain conditions. These methods are not sensitive to the variation of feature values, and do not consider the correlation between samples and features. Compared with constant value imputation methods, the superiority of the vertical federated KNN imputation method lies in the following aspects. Firstly, it leverages the similarity between samples to impute missing values, providing potentially more accurate estimates compared to simple constant imputation methods, as it considers the overall data structure of different parties. Secondly, it not only takes into account information from the feature with missing values but also considers relationships between other features. This helps capture the complex structure and patterns in the data. Lastly, it is a non-parametric method, making no assumptions about the distribution of data. This flexibility is advantageous when dealing with diverse types of data and problems.

4.3.2. Centralized Comparative Experiment

In this experiment, we compare the proposed method with traditional centralized imputation algorithms including MEAN, LR, KNN, and RF to verify the enhancement of the imputation effect by heterogeneous data sets from two parties. In conducting experiments with the centralized imputation algorithm, only the missing data set from party A is utilized for imputation. For experiments with the proposed federated algorithm, both data sets of party A and party B are employed to impute the missing data set of party A. Compare the imputation effect of the data set of party A under the five imputation methods. Similarly, the missing rates are increased from 5% to 40% at the intervals of 5%, and each experiment is conducted three times under each missing rate, using RMSE as the performance metric. The average RMSE of three repeated experiments are given in Table 4 and the deviation is shown in Figure 8.

Table 4. Comparison of imputation performances, FKNN vs. centralized methods.

Missing Rate	FKNN	MEAN	LR	KNN	RF
5%	0.38509	1.06481	1.01250	0.83104	1.07105
10%	0.33951	1.01277	0.95623	0.86715	1.02745
15%	0.36231	1.04159	0.99785	0.92863	1.06034
20%	0.43012	1.02356	0.97037	0.94683	1.03123
25%	0.44781	1.05121	1.01357	1.05514	1.10138
30%	0.48925	1.04509	1.00616	1.05207	1.08145
35%	0.51976	1.02456	0.97432	1.08242	1.07756
40%	0.53624	1.04242	0.99425	1.11375	1.06162

"FKNN" standards for federated KNN imputation method.

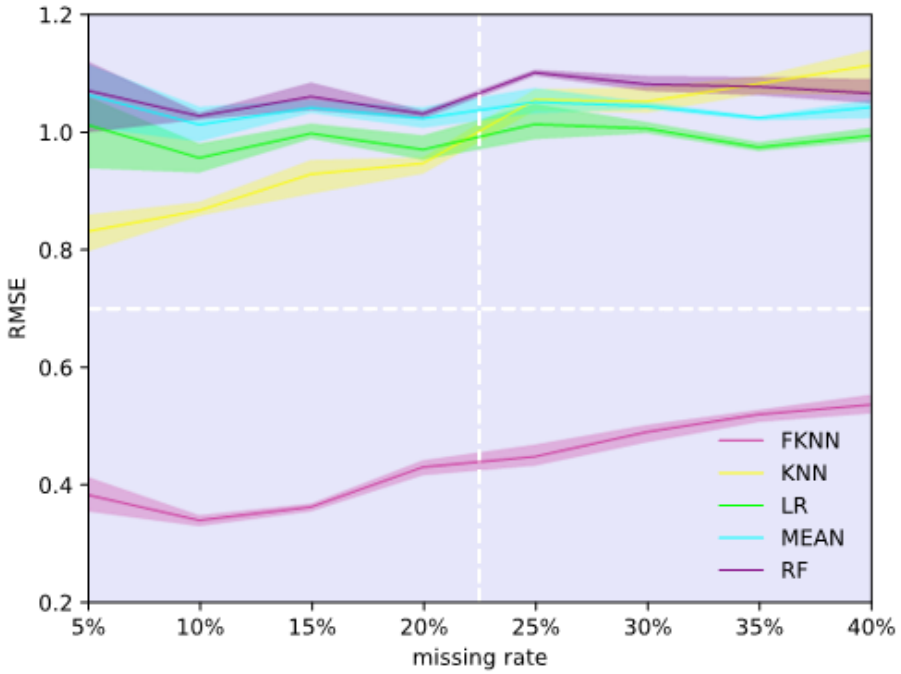

Figure 8. Average RMSE and its deviations at different missing rates in centralized comparative experiment.

The results of the vertical federated KNN imputation method are notably superior to other traditional data imputation methods in all cases of missing rates. The performance of the proposed method and the centralized KNN imputation algorithm gradually decreases when the missing rate exceeds 10%. As the missing rate increases, the experimental metrics of different methods collectively deteriorate and gradually converge. This phenomenon is attributed to the increasing difficulty in imputation as the missing rate rises.

The centralized algorithms can only be applied to one side of the data set, and the effectiveness of the algorithms will be reduced when the one-sided data set has fewer features. Compared with the centralized imputation algorithms, the proposed algorithm expands the feature space by using multi-party data sets, and provides more basis for the similarity of samples. By combining data sets from different sources for imputation, the proposed federated missing data imputation algorithm contributes to enhancing the model's generalization performance. This is because they can capture a broader range of data features and patterns.

4.3.3. Lossless Testing

When the distance matrix is encrypted and decrypted, the calculation results may be affected. This experiment is to compare the federated KNN imputation method with the centralized KNN algorithm to verify the lossless nature. By combining the local data of

the two parties, the centralized imputation experiment is conducted, the imputation effect of partial missing values is shown in Figure 9. Paired-sample *t*-test is performed on the imputation results of the two methods across different missing rates, and the results are shown in Table 5.

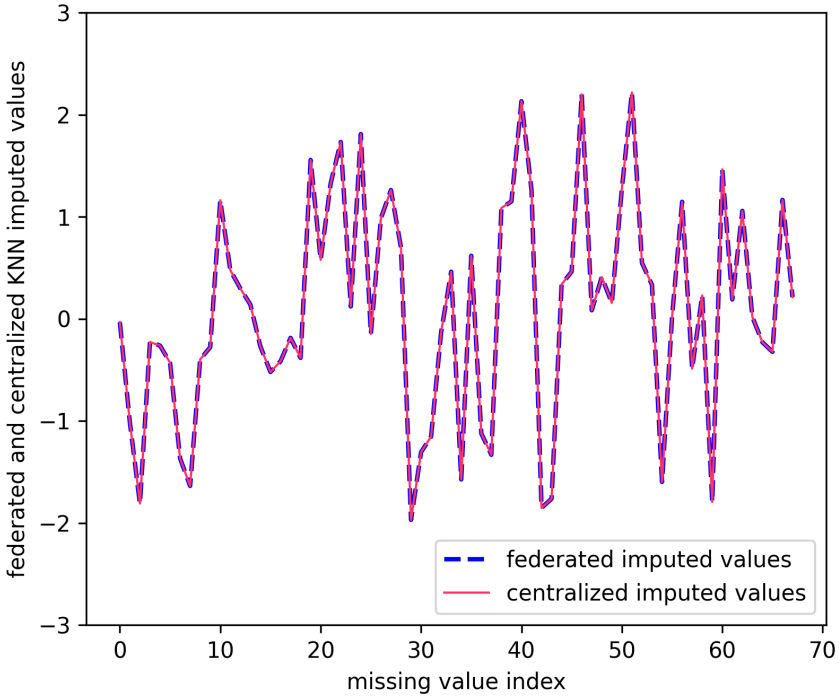

Figure 9. Comparison of imputed values of two methods.

Table 5. Paired sample *t*-test across different missing rates (α = 0.05).

Missing Rates	**5%**	**10%**	**15%**	**20%**	**25%**	**30%**	**35%**	**40%**
p value	0.6412	0.4416	0.7110	0.4836	0.1732	0.3608	0.9499	0.5784

It can be seen from the Figure 9 that the federated KNN imputation results are completely consistent with the centralized KNN imputation results, indicating that the federated KNN imputation method can almost realize lossless federated imputation while protecting local data set.

Furthermore, to demonstrate that the lossless is credible, the *t*-test is conducted. Within each missing rate, repeated experiments are conducted and the two RMSE vectors for both FKNN and centralized KNN are obtained, which is used for paired-sample *t*-test under the significance level of 0.05. As shown in Table 5, all *p* values are much greater than 0.05 proving that the two RMSE vectors have no significant difference and the lossless is proved to be credible.

4.3.4. Contribution to Regression

Another experiment to verify the effectiveness of the federated KNN method is to use the imputed data for modeling tasks, and compare the corresponding modeling performances with other imputation methods. In this experiment, federated KNN imputation method is selected as the basic experimental group, and the MAX, MIN, and MEAN imputation methods are selected as the comparison group to carry out the vertical federated linear regression modeling tasks, respectively.

The imputed data set is utilized for linear regression modeling to predict motor speed. The 80% of the data set is used as the training set and the rest as the test set. During the testing process, there are basically two methods for imputation. The first method is to carry out the completely same process as training process on testing data set. The other method

is to impute with well imputed training data set. Considering the scale of testing set may be small and cannot impute by itself, the second way is adopted. Explainable variance and RMSE are selected as the regression performance metrics. For each method, 10 sets of randomized experiments are conducted, and the average performance metrics from these 10 sets are taken as the final values. The algorithm of vertical federated linear regression realized inside FATE is adopted and the whole training and testing processes are shown in the Figure 10.

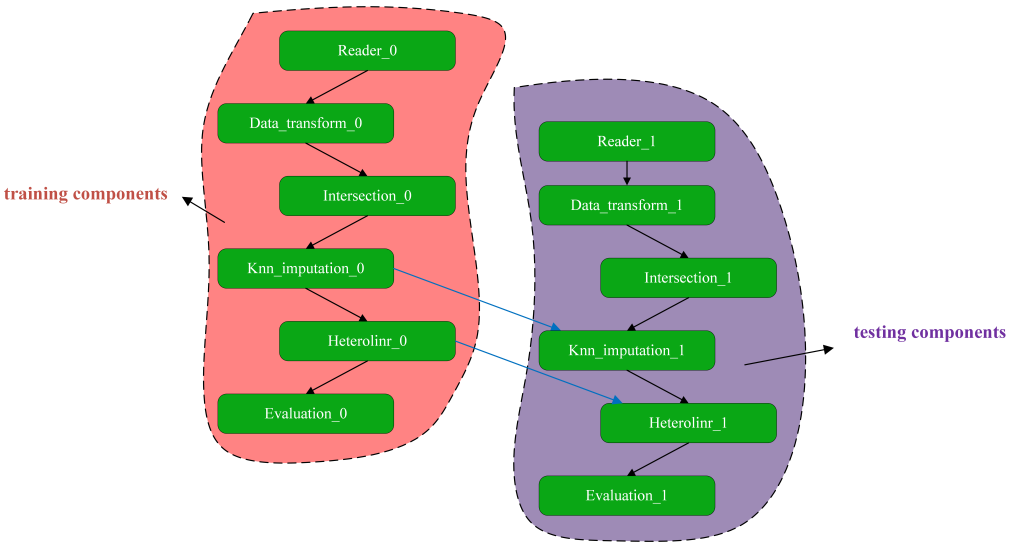

Figure 10. Linear regression modeling processes.

Both processes utilize the Reader component to read the data, and the processed data enters the DataTransform component for format conversion. After completing the data conversion, both sides' data is intersected through the Intersection component to align the data samples based on the ID. The KnnImputation component is then employed to impute any missing values in the data. Following feature imputation, the HeteroLinR linear regression component is used for vertical federated regression modeling. Finally, the results are passed to the Evaluation component to compute the explainable variance and RMSE of the imputation. The experimental results are as follows.

In Figure 11, the left side depicts a chart comparing explained variances. The horizontal axis represents the missing rate, while the vertical axis indicates the values of explained variance. For each missing rate, the explained variances of the four methods are plotted together in a group. It can be observed that, within each group, the FKNN method consistently achieves the maximum value. The right side is a chart comparing RMSE values. The horizontal axis represents the missing rate, while the vertical axis displays the values of RMSE. For each level of missing data, the RMSE values of the four methods are plotted together as a group. It can be observed that, within each group, the FKNN method consistently achieves the minimum value.

In the Figures 12–14, the left side comprises box plots illustrating explained variance, while the right side depicts box plots for RMSE. The upper and lower edges of the box represent the middle 50% range of the ten dots, with the midpoint indicating the mean. Additionally, dots are plotted and methods are distinguished by different colors. It can be observed that FKNN exhibits optimal and significant mean performance. However, the performance gap between the MEAN method and the FKNN method is relatively small. The reasons are as follows.

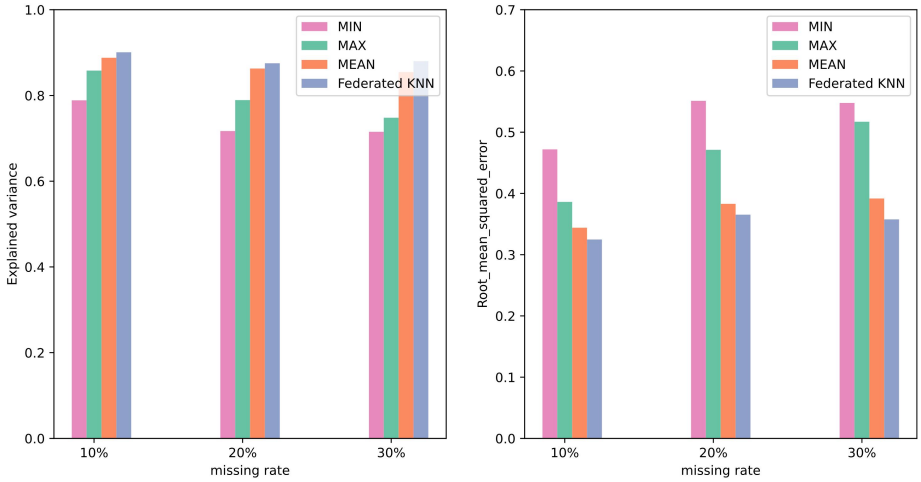

Figure 11. Bar charts of explainable variance and RMSE across different missing rates (The data of Party A contain missing values).

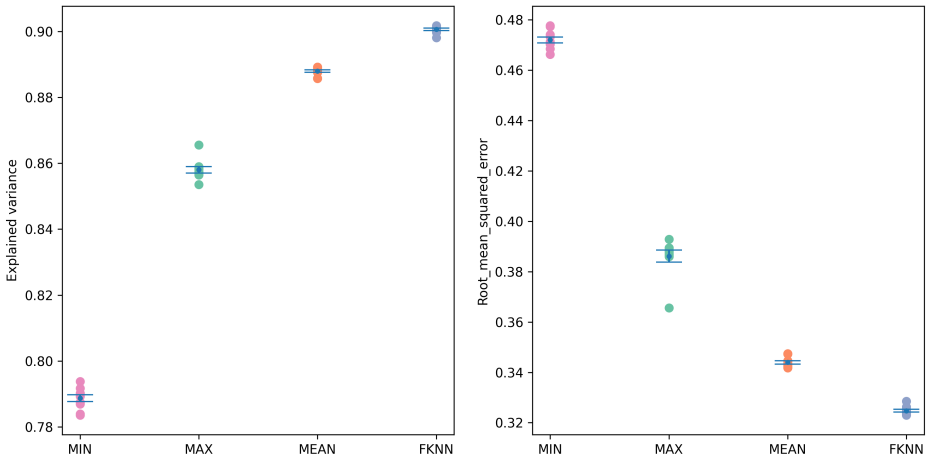

Figure 12. Box plots of explainable variance and RMSE with 10% missing rate (The data of Party A contain missing values).

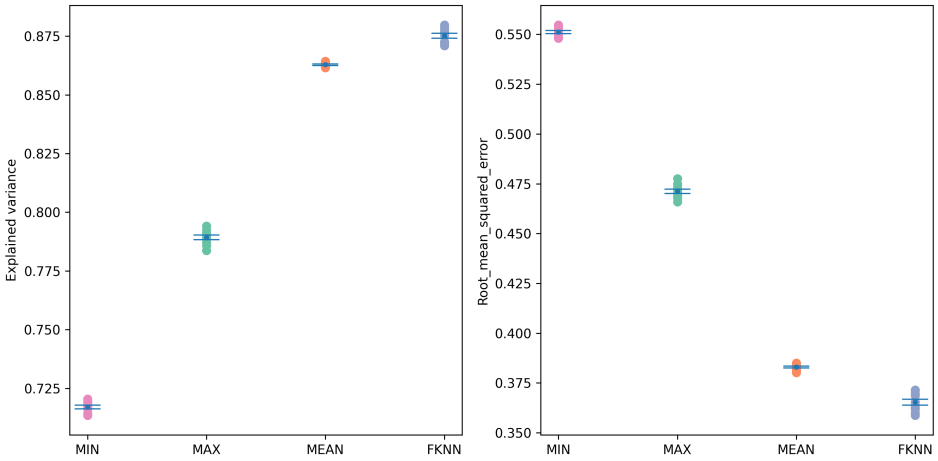

Figure 13. Box plots of explainable variance and RMSE with 20% missing rate (The data of Party A contain missing values).

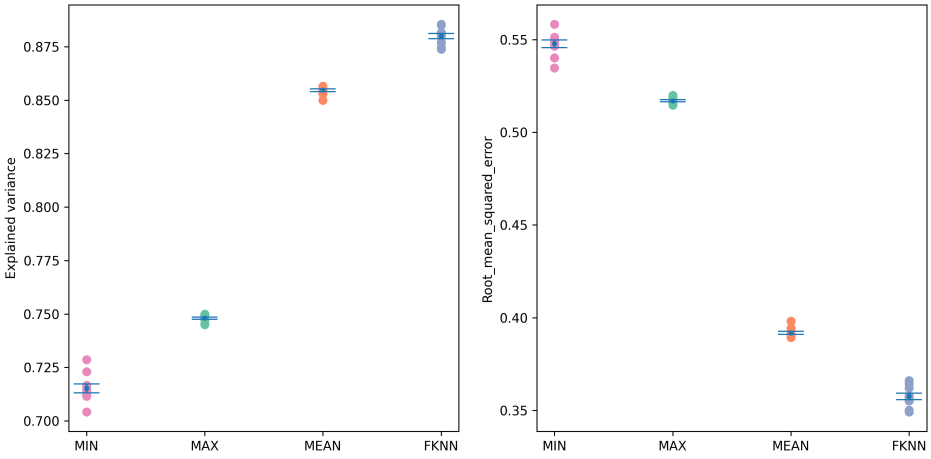

Figure 14. Box plots of explainable variance and RMSE with 30% missing rate (The data of Party A contain missing values).

In this experiment, it is assumed that Party A's data contain missing values, while Party B's data are devoid of any missing values. According to the experimental results, even though the imputed Party A's data exhibit a lower imputation RMSE, the contribution to regression model performance is limited. This is because, as it is settled, Party A's data set contributes less to the label prediction while Party B's data set contributes more to the label prediction. Subsequently, the experiment is modified once again, assuming that Party B's data set also contains missing values. Under the same missing rate with Party A, the experiment is conducted once more, and the results are illustrated as follows.

In Figure 15, the differences between the four methods, whether in terms of explained variance or RMSE, have increased compared to Figure 11, making the distinctions more pronounced. For example, at the 10% missing rate, the difference of the explained variance between FKNN and MEAN is about 0.04, which is larger than 0.01 in Figure 11. In the Figures 16–18, FKNN consistently achieves optimal mean performance. Compared to the Figures 12–14, it can be observed that the interquartile range of each box plot is further expanded at the upper and lower edges, and the distribution of the 10 dots becomes more dispersed. This is because that the missing data from the Party B has a significant impact on the linear regression model. Generally speaking, the proposed federated KNN imputation method makes a significant contribution to linear regression modeling tasks.

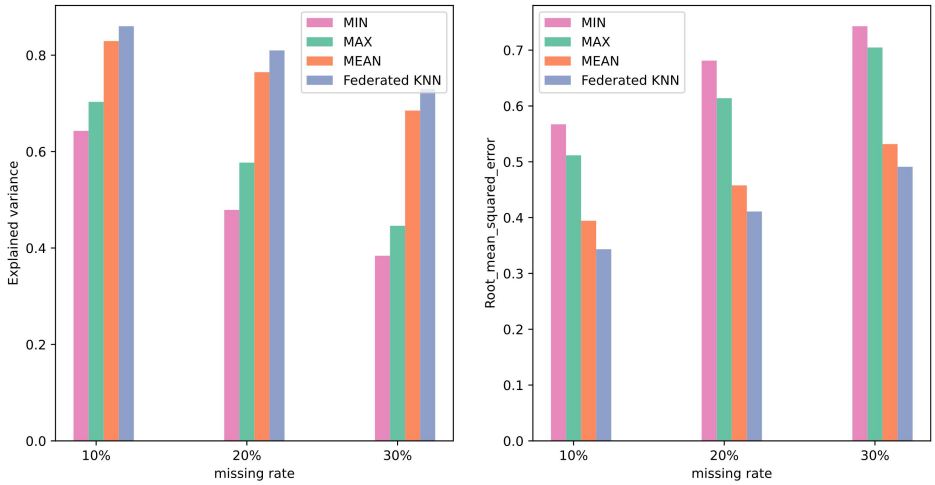

Figure 15. Bar charts of explainable variance and RMSE across different missing rates (The data of Party A and Party B contain missing values).

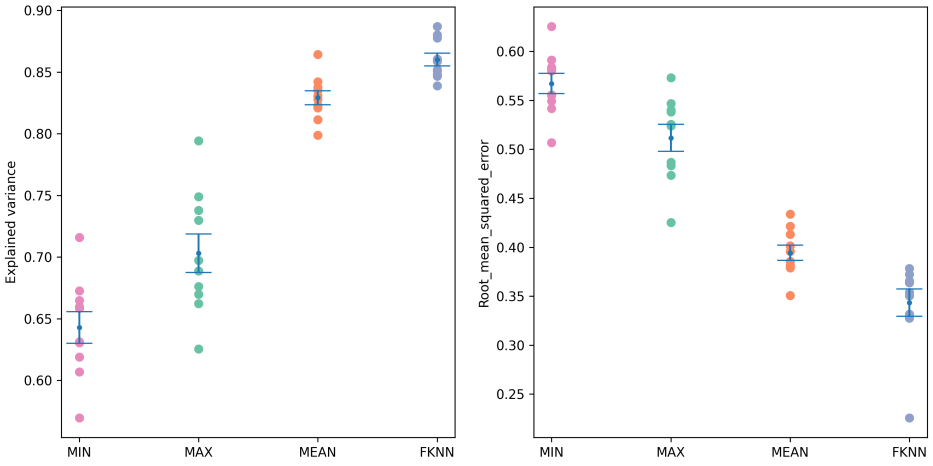

Figure 16. Box plots of explainable variance and RMSE with 10% missing rat (The data of Party A and Party B contain missing values).

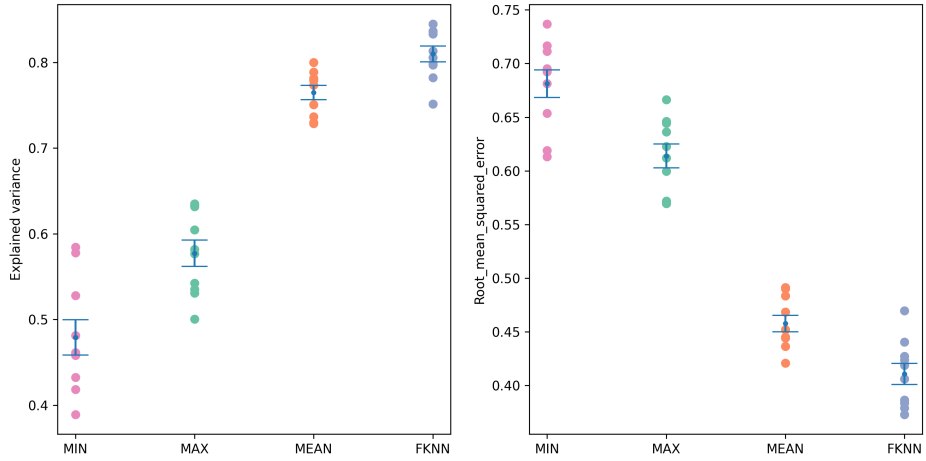

Figure 17. Box plots of explainable variance and RMSE with 20% missing rate (The data of Party A and Party B contain missing values).

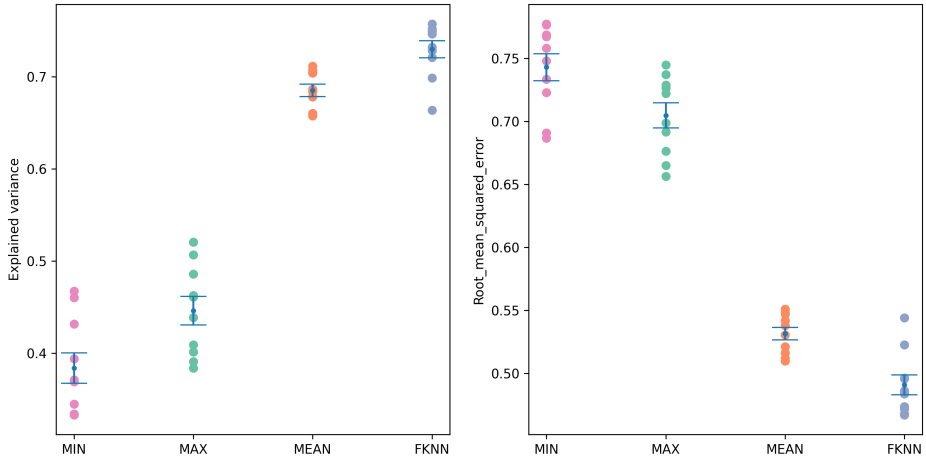

Figure 18. Box plots of explainable variance and RMSE with 30% missing rate (The data of Party A and Party B contain missing values).

Based on the results, it is evident that compared to the scenario where only Party A's data contain missing values, the imputed Party B's data show a more noticeable improvement in regression model performance. This indicates, from the perspective of

predicting the guest label, that the quality of Party B's data is superior to that of Party A's data. This aligns with the original intention of vertical federated modeling, which is to leverage different features from other parties to obtain a better machine learning model.

Taking into consideration the results from the above two experiments, the proposed missing imputation method achieves the maximum value of explainable variance, and the minimum value of RMSE for the listed 10%, 20%, and 30% missing rate cases, compared to the other three basic methods. In the case of 10% missing rate, all four methods achieve their respective maximum explainable variance values, minimum RMSE values.

Overall, data imputation contributes significantly to regression modeling, with the proposed FKNN imputation method demonstrating the most evident modeling contribution. This contribution is also dependent on the features themselves, such as the features of Party B contributing more to the regression model than those of Party A. In such cases, imputing missing data for Party B can enhance the performance of the regression model more effectively.

5. Conclusions

Currently, there is a lack of research on vertical federated data imputation methods. A vertical federated data imputation method is proposed which considers the correlation between data from different parties, aiming to enhance imputation accuracy. The proposed method does not require the exchange of raw data during computation; instead, only the exchange of distance matrix and index matrix is necessary, mitigating the risk of privacy leakage. To further address concerns of potential data leakage or dishonest participants attempting to recover original data through distance matrix during network communication, homomorphic encryption is employed. Party A aggregates the encrypted global distance matrix and submits it to the coordinating party for decryption. Throughout this process, no party can decrypt the local distance matrices of other parties, ensuring the privacy of their data while completing the imputation of missing data based on the global distance matrix.

The imputation method comparison experiment, centralized comparison experiment, lossless testing experiment, and linear regression modeling experiment are designed to verify the validity. According to the comparative evaluation results, our approach exhibits consistently better performance than several simple imputation methods in different performance measurement and prediction tasks. The merits of our method can be attributed to several factors. Firstly, feature values among similar samples exhibit a closer proximity than to other samples. This closeness is dictated by the realistic physical meaning of the features. Therefore, imputing missing values using feature values from similar samples is more aligned with the true values. Secondly, by federating host parties, feature space is expanded, and the sample similarity calculation is more credible to improve its imputation performance.

In future work, enhancements can be made to the proposed method. Similar to other KNN methods, the vertical federated KNN imputation method proposed in this paper also faces computational time constraints. Due to the high computational complexity and spatial complexity of the algorithm, it cannot handle large sample data sets at present. As the size of data set increases, the computational time grows exponentially. There are two directions for improvement in the future. One direction is to federalize other centralized imputation algorithms such as LR and RF, the difficulty is how to deal with the problem of missing values in the modeling process. Another improvement direction is to use approximate KNN algorithm to shorten the computation time and improve the computation efficiency.

Author Contributions: Methodology, W.D.; Software, Y.W.; Validation, Y.W.; Resources, W.D. and G.M.; Writing—original draft, Y.W.; Writing—review & editing, Y.G.; Project administration, W.D.; Funding acquisition, G.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by National Natural Science Foundation of China (61903262), Natural Science Foundation of Liao Ning province (2020-BS-176), and PhD Start-up Fund of Shen Yang Aerospace University (19YB21).

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://github.com/FederatedAI/FATE/tree/master/examples/data.

Conflicts of Interest: Author Wenyou Du was employed by the company Chinese Academy of Sciences Shenyang Institute of Computing Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Regulation (EU) 2016/679 of the European Parliament and of the Council 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 (General Data Protection Regulation). Available online: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679 (accessed on 14 January 2024).
2. Gaff, B.M.; Sussman, H.E.; Geetter, J. Privacy and big data. *Computer* **2014**, *47*, 7–9. [CrossRef]
3. Misra, N.N.; Dixit, Y.; Al-Mallahi, A. IoT, big data, and artificial intelligence in agriculture and food industry. *IEEE Internet Things J.* **2020**, *9*, 6305–6324. [CrossRef]
4. Yang, Q.; Liu, Y.; Chen, T. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 1–19. [CrossRef]
5. Yao, A.C.C. How to generate and exchange secrets. In Proceedings of the 27th Annual Symposium on Foundations of Computer Science, Toronto, ON, Canada, 27–29 October 1986; pp. 162–167.
6. Clifton, C.; Kantarcioglu, M.; Vaidya, J. Tools for privacy preserving distributed data mining. *ACM Sigkdd Explor. Newsl.* **2002**, *4*, 28–34. [CrossRef]
7. Dai, W.; Jin, H.; Zou, D. TEE: A virtual DRTM based execution environment for secure cloud-end computing. In Proceedings of the 17th ACM Conference on Computer and Communications Security, Dubai, United Arab Emirates, 4–8 October 2010; pp. 663–665.
8. Konen, J.; Mcmahan, H.B.; Ramage, D. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *arXiv* **2016**, arXiv:1610.02527.
9. Mcmahan, H.B.; Moore, E.; Ramage, D. Federated learning of deep networks using model averaging. *arXiv* **2016**, arXiv:1602.05629.
10. Konečný, J.; McMahan, H.B.; Yu, F.X. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
11. Cheng, K.; Fan, T.; Jin, Y. Secureboost: A lossless federated learning framework. *IEEE Intell. Syst.* **2021**, *36*, 87–98. [CrossRef]
12. Yang, K.; Fan, T.; Chen, T. A quasi-newton method based vertical federated learning framework for logistic regression. *arXiv* **2019**, arXiv:1912.00513.
13. Zhang, Q.; Wang, C.; Wu, H. GELU-Net: A Globally Encrypted, Locally Unencrypted Deep Neural Network for Privacy-Preserved Learning. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, SE, USA, 13–19 July 2018; pp. 3933–3939.
14. Zhang, Y.; Zhu, H. Additively Homomorphical Encryption based Deep Neural Network for Asymmetrically Collaborative Machine Learning. *arXiv* **2020**, arXiv:2007.06849.
15. Zhou, X.; Liu, X.; Lan, G. Federated conditional generative adversarial nets imputation method for air quality missing data. *Knowl.-Based Syst.* **2021**, *228*, 107261. [CrossRef]
16. Gkillas, A.; BLalos, A.S. Missing Data Imputation for Multivariate Time series in Industrial IoT: A Federated Learning Approach. In Proceedings of the 2022 IEEE 20th International Conference on Industrial Informatics (INDIN), Perth, Australia, 25–28 July 2022; pp. 87–94.
17. FATE (Federated AI Technology Enabler). Available online: https://github.com/FederatedAI/FATE (accessed on 28 November 2023).
18. Aittokallio, T. Dealing with missing values in large-scale studies: Microarray data imputation and beyond. *Briefings Bioinform.* **2010**, *11*, 253–264. [CrossRef] [PubMed]
19. García-Laencina, P.J.; Sancho-Gómez, J. Pattern classification with missing data: A review. *Neural Comput. Appl.* **2010**, *19*, 263–282. [CrossRef]
20. Armitage, E.G.; Godzien, J.; Alonso-Herranz, V. Missing value imputation strategies for metabolomics data. *Electrophoresis* **2015**, *10*, 3050–3060. [CrossRef] [PubMed]
21. Aussem, A.; de Morais, S.R. A conservative feature subset selection algorithm with missing data. *Neurocomputing* **2010**, *73*, 585–590. [CrossRef]
22. De Souto, M.C.; Jaskowiak, P.A.; Costa, I.G. Impact of missing data imputation methods on gene expression clustering and classification. *BMC Bioinform.* **2015**, *16*, 1–9. [CrossRef]

23. Hron, K.; Templ, M.; Filzmoser, P. Imputation of missing values for compositional data using classical and robust methods. *Comput. Stat. Data Anal.* **2010**, *54*, 3095–3107. [CrossRef]
24. Liu, C.C.; Dai, D.Q.; Yan, H. The theoretic framework of local weighted approximation for microarray missing value estimation. *Pattern Recognit.* **2010**, *43*, 2993–3002. [CrossRef]
25. Little, R.J.; Rubin, D.B. *Statistical Analysis with Missing Data*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2019; p. 793.
26. Graham, J.W. Missing Data Analysis: Making It Work in the Real World. *Annu. Rev. Psychol.* **2008**, *60*, 549–576. [CrossRef]
27. Ding, Y.; Ross, A. A comparison of imputation methods for handling missing scores in biometric fusion. *Pattern Recognit.* **2012**, *45*, 919–933. [CrossRef]
28. Kapelner, A.; Bleich, J. Prediction with missing data via Bayesian additive regression trees. *Can. J. Stat.* **2015**, *43*, 224–239. [CrossRef]
29. Ding, Y.F.; Simonoff, J.S. An investigation of missing data methods for classification trees applied to binary response data. *J. Mach. Learn. Res.* **2010**, *11*, 142–149.
30. Li, D.; Gu, H.; Zhang, L. A fuzzy c-means clustering algorithm based on nearest-neighbor intervals for incomplete data. *Expert Syst. Appl.* **2010**, *37*, 6942–6947. [CrossRef]
31. Bahmani, R.; Barbosa, M.; Brasser, F. Secure multiparty computation from SGX. In *International Conference on Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 477–497.
32. Kaggle (Electric Motor Temperature). Available online: https://www.kaggle.com/datasets/wkirgsn/electric-motor-temperature (accessed on 26 April 2021).
33. Lin, W.; Tsai, C.F. Missing value imputation: A review and analysis of the literature (2006–2017). *Artif. Intell. Rev.* **2020**, *53*, 1487–1509. [CrossRef]
34. Troyanskaya, O.; Cantor, M.; Sherlock, G. Missing value estimation methods for DNA microarrays. *Bioinformatics* **2001**, *17*, 520–525. [CrossRef]