MUSE: a trustworthy vertical federated feature selection framework

Ji, X.; Wang, C.; Gadyatskaya, O.; Zhao, F.; Mao, Z.; Xi, W.

MUSE: A Trustworthy Vertical Federated Feature Selection Framework

Xinyuan Ji, Chenfei Wang, Olga Gadyatskaya, Fei Zhao, Zixiang Mao, and Wei Xi

***Abstract*—Vertical federated feature selection can select effective features and avoid overfitting in vertical federated learning. However, existing privacy-preserving techniques for vertical federated feature selection are limited to selecting task-related features and cannot reduce redundant features among clients, resulting in performance loss. This article introduces a mutual information-based federated feature selection (MUSE) framework to address these issues. In the MUSE framework, the correlation of cross-device feature–feature and feature–class is estimated by our defined privacy-preserving mutual information, called federated mutual information (FMI). To compute FMI, we propose the anonymous bin matching (ABM) algorithm, which only uses the intersection size of bins rather than bin elements to avoid *sample-IDs* leakage. With FMI, MUSE can support the minimized dependency feature selection criteria for removing redundant features. Additionally, we propose the local feature preselection to reduce the computation cost of FMI. It is theoretically and experimentally proved as a close approximation of the global optimum under certain constraints. We evaluate the effectiveness of our MUSE framework on various datasets. The experimental results demonstrate that our methods consistently outperform the state-of-the-art federated feature selection methods across most datasets. Moreover, our method shows potential in multimodal data as well.**

***Index Terms*—Anonymous bin matching (ABM), federated mutual information (FMI), vertical federated feature selection.**

I. Introduction

FEDERATED learning (FL) is a decentralized machine learning technique that allows different clients to collaboratively train a model without sharing their raw data [1], [2], [3], [4], [5], [6]. However, there are a lot of noisy and redundant features in the real-world datasets [7], which may result in a decrease in the performance of federated learning models in the case of insufficient sample size [8], [9], [10], [11], [12], [13]. As a solution, federated feature selection can collaboratively select effective features and reduce the noisy features in a secure manner [14], [15], [16].

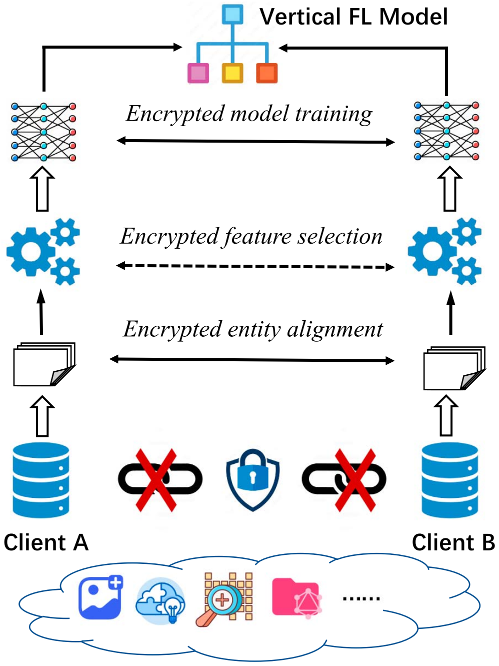


Fig. 1. Flowchart for a two-client vertical FL process. It is a special case for multiclients vertical FL. Vertical federated feature selection can be contained in the vertical FL process. Encrypted entity alignment is needed to confirm the common users of both clients before vertical federated feature selection.

As a type of federated feature selection, *vertical federated feature selection* has strong application requirements, e.g., the fields of text mining [17], [18], information retrieval [9], bioinformatics [19], and industrial applications [20], [21], [22], [23]. As shown in Fig. 1, the vertical federated feature selection (encrypted feature selection) process is done after encrypted entity alignment [24] to confirm the common users (or sample space) of different clients. It can choose effective features from multiple clients with the same *sample-IDs* without raw data sharing [25], [26], [27]. So far, several studies of vertical federated feature selection have been conducted [25], [28], [29].

Nevertheless, the state-of-the-art vertical federated feature selection methods show limitations. Specifically, they focus

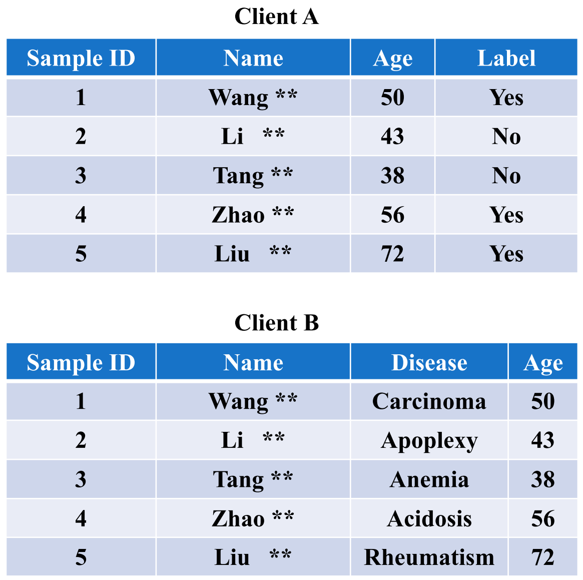
Received 22 May 2024; revised 7 August 2024 and 12 September 2024; accepted 16 September 2024. Date of publication 7 November 2025; date of current version 2 October 2025. This work was supported in part by the National Key R&D Program of China under Grant 2023YFF0905300 and in part by the National Natural Science Foundation of China (NSFC) under Grant 62176205. *(Corresponding author: Wei Xi.)*

Xinyuan Ji is with the Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China, and also with Leiden Institute of Advanced Computer Science, Leiden University, 2333CC Leiden, The Netherlands.

Chenfei Wang, Fei Zhao, Zixiang Mao, and Wei Xi are with the Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xian 710049, China (e-mail: xiwei@xjtu.edu.cn).

Olga Gadyatskaya is with Leiden Institute of Advanced Computer Science, Leiden University, 2333CC Leiden, The Netherlands.

Digital Object Identifier 10.1109/TCSS.2024.3465272



Client A

Sample ID	Name	Age	Label
1	Wang **	50	Yes
2	Li **	43	No
3	Tang **	38	No
4	Zhao **	56	Yes
5	Liu **	72	Yes

Client B

Sample ID	Name	Disease	Age
1	Wang **	Carcinoma	50
2	Li **	Apoplexy	43
3	Tang **	Anemia	38
4	Zhao **	Acidosis	56
5	Liu **	Rheumatism	72

Fig. 2. Illustration of redundant features in vertical federated learning. “Name” and “Age” are the redundant features among clients A and B.

on task-related features but cannot reduce redundant features among clients due to privacy or confidentiality concerns when measuring the relationship of cross-device features. As shown in Fig. 2, there are redundant features on different clients, such as “Name” and “Age.” These redundant features not only increase computational complexity but can also negatively impact the generalization ability of the model, resulting in overfitting of the FL model when dealing with a small number of samples. Therefore, it is very challenging to select task-related features and reduce the redundant features among clients simultaneously.

In this article, we propose the MUSE framework, a privacy-preserving vertical federated feature selection approach designed to address the limitations of existing methods. MUSE aims to select task-related features while minimizing redundant features across clients. In the MUSE framework, anonymous bin matching (ABM) is proposed to measure the correlation of cross-device features without leaking the client’s data privacy. Specifically, ABM computes the mutual information of cross-device features, referred to as federated mutual information (FMI). Crucially, ABM only reveals the size of the intersection between two bins, rather than the elements within the intersection (i.e., *sample-IDs*). Based on ABM, existing feature selection MI-based feature selection criteria such as mRMR [30], MIFS [31], and diversity maximization distance (DD) [32] can be instantiated into MUSE to further eliminate redundant features. Compared to existing methods focused on selecting task-related features with privacy preservation, MUSE addresses the privacy of both selecting task-related and eliminating redundant features. Unfortunately, frequent FMI computations during feature selection result in a high communication and computation overhead. Therefore, we propose the local feature preselection step to preliminarily select features in every client, thereby avoiding unnecessary FMI computations among different clients.

Our contributions can be summarized as follows:

1) We propose a trustworthy framework for vertical federated feature selection. It can select task-related features, as well as reduce redundant features among clients to efficiently improve the accuracy of the FL model.
2) MUSE achieves no *sample-IDs* leakage and lightweight local feature preselection. ABM is proposed to compute federated mutual information without *sample-IDs* leakage from bin sets. Moreover, we theoretically prove that the proposed local feature preselection can achieve an approximation of the global optimal solution with high probability.
3) With an extensive empirical study of diverse datasets, MUSE can effectively improve model accuracy in most cases and reduce the redundancy degree compared to state-of-the-art methods. In addition, local feature preselection reduces the number of FMI computations by more than 20×.

The rest of this article is structured as follows. We describe the related works in Section II. Then, we defined the problem of vertical federated feature selection and detailed the proposed MUSE framework in Section III. In Section IV, we present the performance analyses. Finally, we conclude and discuss the article in Sections VI and V. For the symbols used in this paper, see Table I.

II. Related Work

FL is currently the dominant framework for distributed training of machine learning models under communication and privacy constraints [33], [34]. Many works in the literature focus on the optimization of communication and performance of the global model under the assumption that data is of high quality, without noisy and redundant features in it [35], [36]. Different from the above assumption, we consider the challenge in real applications, when there might be a lot of noisy and redundant features in the data. In this section, we introduce the background of our work.

A. MI-Based Centralized Feature Selection Criteria

Different criteria $J(.)$ are used for adding or removing features when searching for a feature subset in centralized feature selection. These criteria can assess the correlation between candidate features and labels, or among features themselves. Assume $\mathcal{S}$ denotes the currently selected feature set, which is initially empty. Here, $f_j \in \mathcal{S}$ denotes a selected feature within $\mathcal{S}$. $\mathcal{L}$ represents the vector of class labels. Generally, the feature selection criteria $J(.)$ can be defined as (1), where $DIST(\cdot, \cdot)$ represents the correlation distance of two variables, and it varies according to different criteria

$$J(f_k) = \sum_{f_j \in \mathcal{S}} DIST(f_k, f_j) + DIST(f_k, \mathcal{L}). \tag{1}$$

The higher the value of $J(f_k)$, the more important the candidate feature f_k is. After computing the value of $J(.)$ for all candidate

TABLE I
NOTATION DESCRIPTION

Notation	Description
$J(\cdot)$	Feature selection criterion used to score candidate features
$J_{MIFS/mRMR/DD}(\cdot)$	MIFS/mRMR/DD feature selection criteria in the MUSE framework
$\mathcal{S}$	Selected feature subset, containing the set of all selected features
$\mathcal{L}$	Class label vector
K	Total number of clients
N	Total number of samples
$\mathcal{D}/\mathcal{D}_k$	Total feature set/feature set on the *k*th client
d_k	Total number of features on the *k*th client
f_k/f_j	The *k*th column feature f_k or the *j*th column feature f_j
R	Number of selected features/feature selection round
$\mathcal{T}$	Candidate feature set
LMI	Local mutual information
FMI	Federated mutual information
$\mathbf{M}$	Global mutual information matrix

features, the features with the highest feature scores are chosen and added to $\mathcal{S}$. The process is repeated until the desired number of selected features is obtained.

There are different feature selection criteria based on MI. For example, *Mutual Information Feature Selection* (MIFS) [31] and *Minimal Redundancy Maximal Relevance* (mRMR) [30] have been proposed that consider the feature relevance and redundancy at the same time, as shown in (2) and (3). $I(.)$ is the mutual information function

$$J_{MIFS}\left(f_k\right) = I\left(f_k;\mathcal{L}\right) - \beta \sum_{f_j\in\mathcal{S}} I\left(f_k;f_j\right) \tag{2}$$

$$J_{mRMR}\left(f_k\right) = I\left(f_k;\mathcal{L}\right) - \frac{1}{|\mathcal{S}|} \sum_{f_j\in\mathcal{S}} I\left(f_k;f_j\right). \tag{3}$$

In addition, there is another feature selection criteria defined in (4) used for the diversity maximization distance (DD) problem that considers redundancy and relevance [32]

$$J_{DD}(f_k)=\sum_{f_j\in\mathcal{S}}\left(\lambda Red(f_k,f_j)+(1-\lambda)\frac{Rel(f_k,\mathcal{L})+Rel(f_j,\mathcal{L})}{2}\right) \tag{4}$$

$$Red(f_k,f_j) = 1 - \frac{I(f_k;f_j)}{H(f_k,f_j)} \tag{5}$$

$$Rel(f_k,\mathcal{L}) = \frac{I(f_k;\mathcal{L})}{\sqrt{H(f_k)H(\mathcal{L})}} \tag{6}$$

where *Red* and *Rel* are, respectively, the redundancy degree between two features and the relevance of features f_k or f_j with the class labels $\mathcal{L}$, and λ is a regularization factor. The *Red* and *Rel* are defined in (5) and (6), where $H(.)$ is the entropy function. The *Red* value is close to zero when two features are similar.

Many other MI-based feature selection criteria can simultaneously consider the feature relevance and redundancy, such as Normalized MIFS [37], MIFS-ND [38], and FCBF [39], and others. Although these centralized feature selection criteria based on mutual information support the effective removal of redundant features, they often reveal the original data privacy when measuring feature correlation through mutual information calculation. Therefore, it is necessary to propose a mutual information calculation method without revealing privacy.

The main purpose of this article is to instantiate these centralized feature selection criteria based on mutual information computation into the privacy feature selection framework based on mutual information proposed by us, so as to ensure the mutual information computation among the features of some data that do not want to be leaked. Focusing on achieving a privacy-preserving federated feature selection framework, we only instantiate the methods mRMR, MIFS, DD into MUSE as MUSE (mRMR), MUSE (MIFS), and MUSE (DD), although other MI-based feature selection methods also can be instantiated into our framework.

B. Private Set Intersection

Private set intersection (PSI) is a secure multiparty computation technique that allows two parties holding sets to compare encrypted versions of these sets to privately compute the set intersection [40], [41]. One of the possible ways to implement PSI is *Oblivious Transfer* (OT) [42]. It is a protocol between two parties, in which the sending party transfers some data to the receiving party, without knowing what data has been received by the receiver (the sender is thus oblivious to the sent data).

A communication- and computation-efficient PSI based on the multipoint oblivious pseudorandom function (multipoint OPRF-PSI) protocol was proposed [43], [44], where the sender learns a pseudorandom function (PRF) key k and the receiver can obliviously evaluate the outputs from two parties. Motivated by multipoint OPRF-PSI, we proposed the anonymous bin matching (ABM) to compute federated mutual information (FMI). Multipoint OPRF PSI aims to get intersections for two parties, which leads to *sample-IDs* leakage from bin sets. Different from multipoint OPRF-PSI, we propose ABM to compute FMI without *sample-IDs* leakage (presented in Section III-B2).

III. METHODOLOGY

A. Problem Definition

In this section, we formulate the problem of vertical federated feature selection. Let $\mathcal{D}$ denote a collection of features on all clients, where each feature is N-dimensional, such as

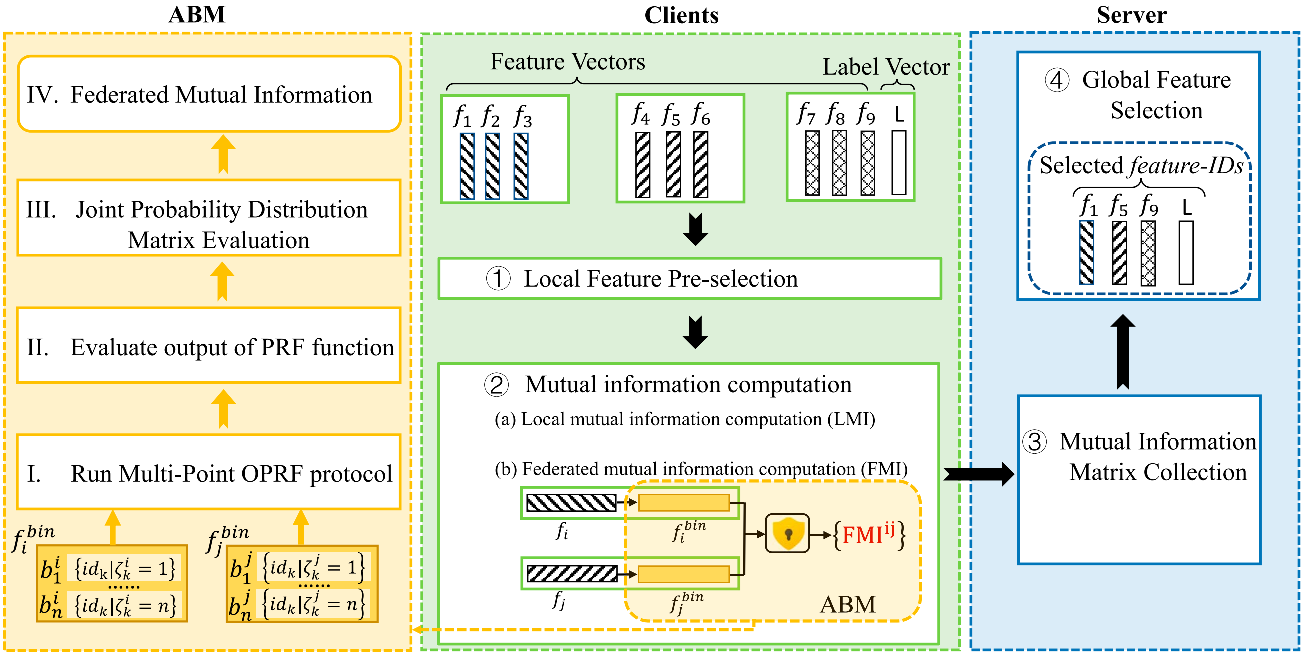

ABM
Clients
Server
IV. Federated Mutual Information
III. Joint Probability Distribution Matrix Evaluation
II. Evaluate output of PRF function
I. Run Multi-Point OPRF protocol
f_i^{bin}
b_1^i $\{id_k|\zeta_k^i=1\}$
......
b_n^i $\{id_k|\zeta_k^i=n\}$
f_j^{bin}
b_1^j $\{id_k|\zeta_k^j=1\}$
......
b_n^j $\{id_k|\zeta_k^j=n\}$
Feature Vectors
Label Vector
f_1 f_2 f_3
f_4 f_5 f_6
f_7 f_8 f_9 L
① Local Feature Pre-selection
② Mutual information computation
(a) Local mutual information computation (LMI)
(b) Federated mutual information computation (FMI)
f_i
f_i^{bin}
$\{$FMI$^{ij}\}$
f_j
f_j^{bin}
ABM
④ Global Feature Selection
Selected *feature-IDs*
f_1 f_5 f_9 L
③ Mutual Information Matrix Collection

Fig. 3. Overview of our MUSE framework. The first step for every client is to make a local feature preselection (Step 1). After that, every client computes: (a) mutual information of local features; (b) federated mutual information, then sends them to the server (Step 2). The server collects all mutual information from clients into a mutual information matrix (Step 3). Finally, the server will select the features according to the mutual information matrix (Step 4). In this setting, the server can only learn which features will be selected, but not the values of features.

feature vector $f = \{\zeta_1, \zeta_2, \ldots, \zeta_N\}$. Here N denotes the number of samples. In the vertical FL setting, the data $\mathcal{D}$ and class labels $\mathcal{L} = \{y_1, y_2, \ldots, y_N\}$ are not at a central location but are distributed across K clients with no raw data exchange, i.e. $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$. Here, $\mathcal{D}_k = \{f_1^k, f_2^k, \ldots, f_{d_k}^k\} \in \mathbb{R}^{N \times d_k}$, for $k \in \{1, \ldots, K\}$, where d_k denotes the number of features at the kth client. Besides, we assume that the class label vector $\mathcal{L}$ is held at only one client. In real-world scenarios, there are a large number of noisy features or redundant features among clients. The vertical federated feature selection aims to select a feature subset $\mathcal{S} = \{f_1, f_2, \ldots, f_R\} \subset \mathcal{D}$. R denotes the number of selected features. After that, the vertical federated learning model w can be learned based on the selected features. Specifically, we have the following definition:

Definition 1 (Vertical Federated Feature Selection): The selected feature subset $\mathcal{S} = \{f_1, f_2, \ldots, f_R\} \subset \mathcal{D}$ is selected from the features $\mathcal{D}_k = \left\{f_1^k, f_2^k, \ldots, f_{d_k}^k\right\} \in \mathbb{R}^{N \times d_k}$ on client k, for $k \in \{1, \ldots, K\}$, s.t. $Acc(f(w; \mathcal{S})) \geq Acc(f(w; \mathcal{D}))$.

B. MUSE Framework

Our proposed MUSE framework is shown in Fig. 3. The framework operates under the assumption that all parties involved in the computation are semihonest (honest but curious). An auxiliary server is required to compute federated mutual information (FMI) values and select features. The framework includes four main steps: Local Feature Preselection, Mutual Information Computation, Mutual Information Matrix Collection, and Global Feature Selection. Mutual Information Computation includes *Local Mutual Information Computation* and *Federated Mutual Information Computation*, where Local Mutual Information Computation aims to compute mutual information of features on clients, and Federated Mutual Information Computation aims to privately compute mutual information of features across clients.

In the first step, every client k performs a "Local Feature Preselection" (detailed in Section III-B3) from their local data $\mathcal{D}_k$. The selected feature index subset is denoted as $\mathcal{S}_k$, where $|\mathcal{S}_k| = R$. In the second step, the clients compute the local and federated mutual information corresponding to $\mathcal{S}_k$ and send these values to the server. The computation methods for Local Mutual Information (LMI) are described in Section III-B1, and the FMI computation is outlined in Algorithm 2 in Section III-B2.

In the third step, the server collects all the mutual information of $\mathcal{S}_k$, for $k \in \{1, \ldots, K\}$ into a mutual information matrix $\mathbf{M}$. Finally, in the fourth step, the server selects the features based on the mutual information matrix $\mathbf{M}$. Specifically, the server initializes the index set of global candidate features as $\mathcal{T} = \bigcup_{k=1}^{k} \mathcal{S}_k$ and the index set of global selected features $\mathcal{S} = \emptyset$. Using a greedy search strategy, the server adds the feature with the maximum value according to the criterion $J(\cdot)$. As mentioned in Section II, $J(\cdot)$ is the function of mutual information of features. The complete pseudo-code of our framework is given in Algorithm 1.

1) Local Mutual Information Computation: To compute $J(\cdot)$ in the MUSE framework, it is necessary to compute the mutual information of features on local clients, referred to as local mutual information (LMI). Given two discrete random

Algorithm 1: The MUSE framework

```
Input: the $K$ clients are indexed by $k$; the number of selected features or rounds $R$; the feature selection criteria $J(\cdot)$.
Output: the index set of selected features $\mathcal{S}$.
Client Executes: /* Run on client $k$ */
if $L$ in client $k$: then
    $\mathrm{I}_{kk'}^{i\mathcal{L}} = \mathrm{LMI}(f_i^k; \mathcal{L}), \forall f_i^k \in \mathcal{D}_k, k' = k$;
end
else
    /*Call Algorithm 2*/
    $\mathrm{I}_{kk'}^{i\mathcal{L}} = \mathrm{FMI}(f_i^k; \mathcal{L}), \forall f_i^k \in \mathcal{D}_k, \mathcal{L}$ in client $k'$;
end
$\mathrm{I}_{kk'}^{ij} = \mathrm{LMI}(f_i^k; f_j^{k'}), \forall f_i^k, f_j^{k'} \in \mathcal{D}_k, i < j, k = k'$;
/* $\mathcal{S}_k \leftarrow$ pre-selects $R$ features (if $R > d_k$: $R = d_k$) */
initializes $\mathcal{T}_k \leftarrow \mathcal{D}_k$;
initializes $\mathcal{S}_k = \emptyset$;
for each local round $r_k$ = 1: R do
    $f_i \leftarrow \mathrm{argmax}_{f_i \in \mathcal{T}_k} J(f_i)$;
    /*add $f_i$ to $\mathcal{S}_k$*/
    $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup f_i$;
    $\mathcal{T}_k \leftarrow \mathcal{T}_k \backslash f_i$;
end
/*Call Algorithm 2*/
$\mathrm{I}_{kk'}^{ij} = \mathrm{FMI}(f_i^k; f_j^{k'}), \forall f_i^k \in \mathcal{S}_k, f_j^{k'} \in \mathcal{S}_{k'}, k < k'$;
Server Executes:
$M = (\mathrm{I}_{ij})_{|\bigcup_{k=1}^{K} \mathcal{S}_k| \times |\bigcup_{k=1}^{K} \mathcal{S}_k \bigcup \{\mathcal{L}\}|}, \mathrm{I}_{ij} = MI(f_i, f_j)$
where $f_i$, $f_j \in |\bigcup_{k=1}^{K} \mathcal{S}_k \bigcup \{\mathcal{L}\}|$, specifically, $f_j = \mathcal{L}$ as label vector.
initializes $\mathcal{T} \leftarrow \bigcup_{k=1}^{K} \mathcal{S}_k$;
initializes $\mathcal{S} = \emptyset$;
for each round $r$ = 1: R do
    $f_i \leftarrow \mathrm{argmax}_{f_i \in \mathcal{T}} J(f_i)$;
    /*add $f_i$ to $\mathcal{S}$*/
    $\mathcal{S} \leftarrow \mathcal{S} \cup f_i$;
    $\mathcal{T} \leftarrow T \backslash f_i$;
end
returns $\mathcal{S}$ to $clients$;
```

variables x and y, their mutual information is defined based on their probabilistic density functions $p(x)$, $p(y)$, and $p(x, y)$ as follows:

$$I(x; y) = \sum_{x_i \in x} \sum_{y_j \in y} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)\, p(y_j)}. \tag{7}$$

It is straightforward to compute mutual information by estimating the probability distribution through counting the number of samples that fall into the intersections of bins.

It means that mutual information computation is independent of the specific values of features, but it depends on the number of samples in the intersection of bins. For example,

Algorithm 2: Anonymous Bin Matching (ABM)

```
Input: Server $\boldsymbol{S}$, $f_1^{bin}$ and $f_2^{bin}$ from clients $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$
Output: FMI($f_1^{bin}$; $f_2^{bin}$)
(1) Run Multi-Point OPRF
    $\boldsymbol{P_1} \leftarrow F_{key}(f_1^{bin}) = \{F_{key}(b_i^1) | i \in [n]\}$,
    $F_{key}(b_i^1) = \{F_{key}(id_j) | (\zeta_j^1 = i) \wedge (j \in [N])\}$;
    $\boldsymbol{P_2} \leftarrow$ secret key of PRF $key$;
(2) Evaluate the output of the PRF function
    $\boldsymbol{P_2} \leftarrow F_{key}(f_2^{bin}) = \{F_{key}(b_i^2) | i \in [n]\}$,
    $F_{key}(b_i^2) = \{F_{key}(id_j) | (\zeta_j^1 = i) \wedge (j \in [N])\}$;
(3) Joint Probability Distribution Matrix Evaluation
    $\boldsymbol{P_1}$, $\boldsymbol{P_2}$ sends $F_{key}(f_1^{bin})$ and $F_{key}(f_2^{bin})$ to $\boldsymbol{S}$;
    $\boldsymbol{S}$ gets a joint probability distribution matrix M according to Eqn 8;
(4) Federated Mutual Information Output
    $\boldsymbol{S}$ computes FMI($f_{bin}^1$; $f_{bin}^2$) according to Eqn 9.
Return FMI($f_{bin}^1$; $f_{bin}^2$)
```

assume that two feature vectors $f_1 = \{\zeta_1^1, \zeta_2^1, \ldots, \zeta_N^1\}$ and $f_2 = \{\zeta_1^2, \zeta_2^2, \ldots, \zeta_N^2\}$ are discretized into n bins, where ζ_i^1 and ζ_i^2 ($\forall i \in [N]$) separately take a value from the finite set $[n] = \{1, \ldots, n\}$. Besides, the N samples are identified by *sample-IDs* $= \{id_1, \ldots, id_N\}$, so f_1 can be reconstructed as $f_1^{\mathrm{bin}} = \{b_i^1 \mid i \in [n]\}$, where $b_i^1 = \{id_j \mid (\zeta_j^1 = i) \wedge (j \in [N])\}$ (f_2 can be constructed in the same way as $f_2^{\mathrm{bin}} = \{b_i^2 \mid i \in [n]\}$, where $b_i^2 = \{id_j \mid (\zeta_j^2 = i) \wedge (j \in [N])\}$). For two binned features, we get a joint probability distribution matrix

$$\mathbb{P} = \left\{ p_{ij}, p_{ij} = \frac{|b_i^1 \bigcap b_j^2|}{N} \right\}, i, j \in [n] \tag{8}$$

in which $\sum_{i=1}^n |b_i^1| = \sum_{j=1}^n |b_j^2| = N$, where $|.|$ denotes the size of a set. Thus, $p(b_i^1) = \sum_{j=1}^n p_{ij}$, $p(b_i^1, b_j^2) = p_{ij}$, and $p(b_j^2) = \sum_{i=1}^n p_{ij}$. So, formally, the mutual information of two feature vectors f_1 and f_2 can be defined as

$$\begin{aligned} I(f_1; f_2) &= \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} \log \left(\frac{p_{ij}}{\sum_{j=1}^n p_{ij} \sum_{i=1}^n p_{ij}} \right) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{|b_i^1 \bigcap b_j^2|}{N} \log \frac{N\, |b_i^1 \bigcap b_j^2|}{\sum_{j=1}^n |b_i^1 \bigcap b_j^2| \sum_{i=1}^n |b_i^1 \bigcap b_j^2|}. \end{aligned} \tag{9}$$

2) Federated Mutual Information Computation: As described in Section III-B1, the local mutual information between two features can be computed according to (9). However, private information may be leaked when computing the mutual information of two features from different clients. For example, the discretized features include the *sample-IDs* in bins. To address this issue, we propose the ABM algorithm to compute mutual information of cross-device features, referred to as FMI,

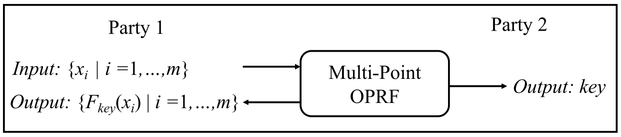

Party 1
Party 2
Input: $\{x_i \mid i = 1,\ldots,m\}$
Output: $\{F_{key}(x_i) \mid i = 1,\ldots,m\}$
Multi-Point OPRF
Output: key

Fig. 4. Multipoint OPRF protocol.

without potential information leakage. The proposed ABM algorithm primarily relies on the multipoint oblivious pseudorandom function (OPRF) protocol [44], where the detailed privacy proof guarantees the effectiveness of privacy for ABM. As shown in Fig. 4, a multipoint OPRF is a pseudorandom function with the following properties.

a) Two parties compute: Output $= F_{key}$(Input).

b) The first party knows the set of inputs $\{x_i | i = 1, \ldots, m\}$ and learns the set of outputs $\{F_{key}(x_i) | i = 1, \ldots, m\}$ but does not learn the secret key key.

c) The second party, only knows the secret key, but does not learn either inputs $\{x_i | i = 1, \ldots, m\}$, nor the output $\{F_{key}(x_i) | i = 1, \ldots, m\}$.

Based on the multipoint OPRF protocol, the PSI can be achieved easily. The second party just needs to evaluate the PRF function on every element $\{y_i | i = 1, \ldots, m\}$ in its set and send all the PRF values $\{F_{key}(y_i) | i = 1, \ldots, m\}$ to the first party. By comparing these PRF values, the first party can easily figure out the intersection of the two sets.

For two cross-device binned features $f_1^{\text{bin}} = \left\{b_i^1 | i \in [n]\right\}$ and $f_2^{\text{bin}} = \left\{b_i^2 | i \in [n]\right\}$, the FMI can be computed according to following steps.

a) Two parties run a multipoint OPRF. Based on the multipoint OPRF protocol, Party 2 holds the key, and Party 1 holds the PRF values $F_{key}(f_1^{\text{bin}}) = \left\{F_{key}(b_i^1) | i \in [n]\right\}$, $F_{key}(b_i^1) = \left\{F_{key}(id_j) | (\zeta_j^1 = i) \wedge (j \in [N])\right\}$.

b) Party 2 evaluates the PRF function on f_2^{bin} as the PRF values $F_{key}(f_2^{\text{bin}}) = \left\{F_{key}(b_i^2) | i \in [n]\right\}$, $F_{key}(b_i^2) = \left\{F_{key}(id_j) | (\zeta_j^1 = i) \wedge (j \in [N])\right\}$.

c) Two parties send all the PRF values to the server. By comparing these PRF values, the server can compute a joint probability distribution matrix as (8).

d) The FMI output of two cross-device features according to (9).

The detail of ABM can refer to Algorithm 2. It can compute the FMI for two cross-device discretized vectors without common elements leakage, which means that the server can only get the cardinality of the intersection but knows nothing about *sample-IDs* in the bin set of discretized vectors.

3) Local Feature Preselection: Excessive FMI computations within the MUSE framework can lead to substantial overheads in both communication and computation. Consequently, we propose the local feature preselection method. This approach aims to curtail the quantity of FMI computations involved in global feature selection, thereby mitigating the strain on resources and enhancing efficiency.

Every client k should first calculate the feature-class FMI (only if L is not in client k, otherwise calculate the feature-class LMI) and feature–feature LMI to ready for the local feature preselection (Line 4–Line 9 of Algorithm 1). Then client k greedily selects the R features as the selected feature subset S_k based on the feature selection criteria $J(\cdot)$. With the local feature preselection step, there will only be up to $C_K^2 R^2$ feature–feature FMI computed for the global feature selection. However, without local feature preselection, there will be $\sum_{i=1}^{K} \sum_{j=i+1}^{K} d_i \times d_j$ feature–feature FMI computation. Therefore, the local feature preselection step can remove unrelated and redundant features in advance, and effectively reduce the number of FMI computations for the global feature selection process. We also prove that the local feature preselection step ensures an acceptable approximation of the global optimal solution.

Theorem 1: For any constant $0 < \epsilon \leq (1/4)$, Algorithm 1 with local feature preselection is a $(1 - \epsilon/4)$-approximation algorithm for centralized feature selection problem with a high probability (e.g., probability $1 - e^{\Omega(-\epsilon R)}$) with $K \geq (6/\epsilon)$ clients.

Lemma 1: There are at most $\epsilon R/3$ features in OPT_k for every $1 \leq k \leq K$ with probability at least $1 - e^{\Omega(-\epsilon R)}$. □

Proof: Let OPT be the optimum set of R features. Let OPT_k be $OPT \cap \mathcal{D}_k$, the optimum solution features that are sent to machine k. We expect R/K features in each set OPT_k, and $|\text{OPT}_k| = \sum_{f \in \text{OPT}} f = R$. Then, according to Theorem 4 in [45], $\Pr\left[|\text{OPT}_k| > (1 + \delta)\mu\right] < e^{-(\delta^2/2+\delta)\mu}$, where μ is R/K and δ is set to $(K\epsilon/3) - 1$, so we have $(1 + \delta)\mu = (\epsilon R/3)$. Since K is at least $6/\epsilon$, δ is at least 1. We conclude that $\Pr\left[|\text{OPT}_k| > (\epsilon R/3)\right] < e^{-(1+\delta/6)\mu} = e^{-R\epsilon/18}$, where the $\Pr\left[|\text{OPT}_k| > (\epsilon R/3)\right]$ converges to zero with R growth. So $\Pr\left[|\text{OPT}_k| \leq (\epsilon R/3)\right] < 1 - e^{-R\epsilon/18} = 1 - e^{\Omega(-\epsilon R)}$. ■

Lemma 2: There exists a set A of R features among the selected features $\cup_{k=1}^{K} \mathcal{S}_k$ that represent the R features of OPT, and the distance of each optimum feature and its representative substitutions in A is upper bounded by $(1 + \epsilon)\tau$. □

Proof: We create a set $A \subset \cup_{k=1}^{K} \mathcal{S}_k$ of R features. We add each feature $f \in \text{OPT} \cap (\cup_{k=1}^{K} \mathcal{S}_k)$ to set A. For every feature $f \in \text{OPT}$ and $\left(f \notin \cup_{k=1}^{K} \mathcal{S}_k\right)$, we will find a selected feature close to it. Let $\mathcal{S}_k = \{f_1, \ldots, f_R\}$ be the features that machine k selected with the same order (f_1 is selected first, and f_R is selected last). According to the greedy search strategy, we have the following inequalities for any feature $f \in (\text{OPT} \cap \mathcal{D}_k) \setminus \mathcal{S}_k$:

$$\begin{gathered} \text{DIST}\,(f, f_1) \leq \text{DIST}\,(f_2, f_1) \\ \text{DIST}\,(f, f_1) + \text{DIST}(f, f_2) \leq \text{DIST}(f_3, f_1) + \text{DIST}(f_3, f_2) \\ \cdots \\ \sum_{i=1}^{R-1} \text{DIST}\,(f, f_i) \leq \sum_{i=1}^{R-1} \text{DIST}\,(f_R, f_i)\,. \end{gathered} \tag{10}$$

Summing the above inequalities implies that

$$\begin{aligned} \sum_{i=1}^{R-1} (R - i)\,\text{DIST}\,(f, f_i) &\leq \sum_{i=1}^{R-1} \sum_{j=i+1}^{R} \text{DIST}\,(f_i, f_j) \\ &= \text{DIV}\,(\mathcal{S}_k) \end{aligned} \tag{11}$$

where $\mathrm{DIV}(\mathcal{S})$ (and $\mathrm{DIV}(\mathcal{S}_k)$) can be defined as follows:

$$\mathrm{DIV}(\mathcal{S}) = \frac{1}{2} \sum_{f \in \mathcal{S}} \sum_{f' \in S} \mathrm{DIST}(f, f'). \tag{12}$$

On the left of (11), we have $\binom{R}{2}$ distances from f to features in $\mathcal{S}_k$, and on the right of (11), it is the diversity of set $\mathcal{S}_k$. Let τ be the maximum average distance of pairs of features in selected sets, $\max_{1 \le k \le K}(\mathrm{Div}\,(\mathcal{S}_k)/\binom{R}{2})$. Then we can get $(\sum_{i=1}^{R-1}(R-i)\,\mathrm{DIST}\,(f, f_i)\binom{R}{2}) \le ((\mathrm{Div}\,(\mathcal{S}_k)\binom{R}{2}) = \tau)$, where τ is the upper bound. Furthermore, we aim to verify that the distance of f to at least $(\epsilon|\mathcal{S}_k|/3)$ features in $\mathcal{S}_k$ is upper bounded by $(1+\epsilon)\tau$. Otherwise, there are $a > (1-\epsilon/3)R$ features in $\mathcal{S}_k$ with distance more than $(1+\epsilon)\tau$ from f. In the left part of (11), at least $\binom{a}{2}$ of the $\binom{R}{2}$ distances are greater $(1+\epsilon)\tau$. So we have the following lower bound on the left side of (11):

$$\begin{aligned}
\tbinom{R}{2}(1+\epsilon)\tau \ge \Bigg(& \frac{((1-\epsilon/3)R)((1-\epsilon/3)R-1)}{2} \times (1+\epsilon)\tau \\
& = \tbinom{R}{2}\tau \times \frac{(1-\epsilon/3)R}{R} \times \frac{(1-\epsilon/3)R-1}{R-1} \times (1+\epsilon) \\
& \ge \tbinom{R}{2}\tau \times (1-\epsilon/3) \times \left(1-\epsilon/3-\frac{\epsilon/3}{2}\right) \times (1+\epsilon) \\
& > \tbinom{R}{2}\tau \ge \mathrm{DIV}\,(\mathcal{S}_k)\Bigg)
\end{aligned} \tag{13}$$

where the first inequality holds by the lower bound on a, and the second to the last inequality holds since $\epsilon \le 1/4$. Combining the above lower bound on $\sum_{i=1}^{R-1}(R-i)\,\mathrm{DIST}(f, f_i)$ with (11) implies a contradiction. So there should be at least $\epsilon R/3$ features in $\mathcal{S}_k$ with distance at most $(1+\epsilon)\tau$ from f. Since there are at most $\epsilon R/3$ features in OPT_k with high probability, we can find one distinct representative feature $f' \in \mathcal{S}_k$ for each feature $f \in OPT_k \setminus \mathcal{S}_k$ to add to A.

We conclude that there exists a set A of R features among the selected features $\cup_{k=1}^{K}\mathcal{S}_k$ that represent the R features of OPT, and the distance of each optimum feature and its representative in A is upper bounded by $(1+\epsilon)\tau$. ■

Now we are ready to prove Theorem 1.

Proof: Using the triangle inequality, we know that $(\mathrm{DIV}(A) = \sum_{p',q' \in A} \mathrm{DIST}(p', q')) \ge \sum_{p,q \in \mathrm{OPT}} \mathrm{DIST}(p, q) - \mathrm{DIST}\,(p, p') - \mathrm{DIST}\,(q, q') \ge \mathrm{DIV}(\mathrm{OPT}) - 2\binom{R}{2}(1+\epsilon)\tau$ where p' and q' are the representatives of p and q. We know that the greedy algorithm is a centralized 1/2-approximation for diversity maximization [46]. So we can find a set $\mathcal{S}$ with diversity at least half of diversity of A on $\cup_{k=1}^{K}\mathcal{S}_k$, $1/2\mathrm{DIV}(A) \le 1/2(\mathrm{DIV}(\mathrm{OPT}) - 2\binom{R}{2}(1+\epsilon)\tau)$. Finally, we take the maximum diversity of this selected set and all m selected sets $\{\mathcal{S}_k\}_{k=1}^{K}$, the diversity of the final output set will be at least $\max\left\{\binom{R}{2}\tau, (\mathrm{DIV}(\mathrm{OPT})/2) - (1+\epsilon)\binom{R}{2}\tau\right\}$ which is at least $(\mathrm{DIV}(\mathrm{OPT})/4(1+\epsilon)) \ge (1-\epsilon/4)$. ■

IV. Experiments

We experimentally evaluate MUSE with four goals in mind. The first two goals of the experiments are to compare the performance of MUSE with the state-of-the-art baselines, including the prediction accuracy of models based on the selected features and the redundancy of selected features. Our third goal is to explore the effect of the bin number on MUSE. Our fourth goal is to demonstrate the reduction in the number of FMI computations with the local feature preselection step. Note that we show the effectiveness of the local feature preselection step by the number of FMI computations but not the running time. The key factor for overheads is the number of FMI computations.

TABLE II
Summary of the Datasets

# Dataset	# Features	# Instances	# Classes
Colon	2000	62	2
Lung	3312	203	5
Lymphoma	4026	96	9
NCI9	9712	60	9
Pixraw10P	10 000	100	10
RELATHE	4322	1427	2
Srbct	2308	83	4
TOX-171	5748	171	4
USPS	256	9298	10
Yale	1024	165	15

A. Experimental Setup

Our experiments have been performed on a workstation using an Apple M1 CPU with 8 cores at 3.2 GHz, 16 GB, and Matlab R2021a.

1) Datasets and Models: We use ten benchmark datasets[1] with different sample sizes and feature numbers from different fields. These datasets include six biological datasets, two face image datasets, one text dataset, and one handwritten image dataset. Table II illustrates the information of these datasets. To simulate the federated feature selection setting, we distribute various number of features across $K = 20$ clients following a log–normal distribution $\mathcal{N}(\mu, \sigma)$ with $\mu = 4, \sigma = 2$. The labels are only located in the active party. Following this, we generate overlapping features across clients by setting $\alpha = 0.5$, $\beta = 0.5$, where α is the rate of the clients we randomly select to generate overlapping features, β is the ratio of the number of overlapping features to the total number of features at the clients. All experiments are performed with support vector machines (SVM) and k nearest neighbors (KNN) as the classifiers, and the tenfold cross-validation (CV) is used for different datasets. The parameter defaults are the initial setting unless indicated otherwise. We set $k = 3$ in the KNN classifier and we use the LIBSVM package [47] with the regularization factor $c = 1$, and the linear kernel function.

2) Our Method and Baselines: We instantiate the MI-based feature selection criterion: mRMR, MIFS, DD to into MUSE. Therefore, there are three corresponding instantiated methods of MUSE. We consider the original feature set *OFS* (without generated overlapped features), and all features participating *AFP* (with generated overlapping features) as baselines. For

[1] All datasets are available at https://jundongl.github.io/scikit-feature/datasets.html [9].

TABLE III
Prediction Accuracy of Various Methods Evaluated by SVM

	Colon	Lymphoma	NCI9	Pixraw10P	RELATHE	Srbct	TOX-171	USPS	Yale	W/T/L
OFS	35.00	36.67	13.33	90.00	53.80	97.50	93.53	94.90	71.88	4/0/5
AFP	56.67	53.33	21.67	86.00	53.80	96.25	66.47	91.44	16.88	6/1/2
SFFS	65.00	78.89	28.33	**92.00**	53.80	80.00	49.41	84.90	56.88	7/1/1
FATE-IV	63.33	70.00	28.33	43.00	62.46	83.75	51.18	59.76	41.88	9/0/0
FATE-Lasso	**70.00**	76.67	13.33	**92.00**	53.80	90.00	**74.12**	92.62	58.75	5/0/4
MUSE (mRMR)	65.00	**88.89**	**33.33**	86.00	**68.80**	**95.00**	71.76	91.31	**66.25**	–
MUSE (MIFS)	35.00	70.00	13.33	84.00	53.80	87.50	66.47	93.63	38.12	8/0/1
MUSE (DD)	35.00	82.22	13.33	87.00	53.80	85.00	68.24	**94.41**	61.25	7/0/2

Note: MUSE (mRMR), MUSE (MIFS), and MUSE (DD) are the vertical federated feature selection framework MUSE under mRMR, MIFS, and DD criteria, respectively. The results with the highest accuracy are indicated in bold for each dataset (column).

reference, we also compare MUSE with three state-of-the-art methods of vertical federated feature selection methods: *SFFS* [25] and *FATE-IV* [48], and *FATE-Lasso* [48]. We summarized our method and the baselines as follows.

a) *MUSE(mRMR):* instantiated mRMR criteria into MUSE.

b) *MUSE(MIFS):* instantiated MIFS criteria into MUSE. We set hyperparameter $\beta = 0.8$.

c) *MUSE(DD):* instantiated DD criteria into MUSE. The regularization factor of MUSE(DD) (λ) is set to be 0.5 in all of our experiments.

d) *OFS:* The model is trained on the original feature set, and there are no overlapping features in it.

e) *AFP:* The model is trained on the generated feature set, where the generated overlapping features are included.

f) *SFFS:* A federated feature selection algorithm based on secure multiparty computation. It is designed for vertical FL to reduce the ineffective features with privacy protection.

g) *FATE-IV:* A information value (IV) [49] based feature selection method in the FATE framework. The IV values related to the label vector are calculated for each feature, and the top-k IV value features are selected.

h) *FATE-Lasso:* Every feature will get a coefficient to represent the degree of importance when the Lasso regression model is trained in FATE, and the features are sorted with absolute values of coefficients. We set hyperparameter $\lambda = 10^{-2}$.

We discretize continuous variables to bins with $n = 15$ in MUSE and FATE-IV.

3) Metrics: In this article, we use test accuracy and redundancy as the evaluation metric to measure the performance of our proposed MUSE framework. The higher accuracy denotes the better performance of the selected features. The lower redundancy denotes the more compact is the selected features. We need to explain some indicators in detail as follows.

The test accuracy to evaluate the performance is by constructing a classifier on the training dataset with selected features and finally getting predicted performance on the test dataset, where Accuracy $=$ (# of correct predictions Precision/# of test dataset).

To quantify thfe degree of redundancy, we begin by computing the Spearman correlation matrix $\mathbf{S}$ [50] for the selected features. Each element of $\mathbf{S}$ is determined according to the below equation, where "cov" represents the covariance of two features, and "σ" denotes the standard deviation

$$s_{ij} = \frac{\text{cov}(f_i, f_j)}{\sigma f_i \sigma f_j}. \tag{14}$$

Subsequently, we can measure redundancy by calculating the squared Euclidean distance between the Spearman correlation matrix $\mathbf{S}$ and the Identity matrix $\mathbf{E}$, as depicted in the below equation

$$d(\mathbf{S}, \mathbf{E}) = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} (s_{ij} - e_{ij})^2}. \tag{15}$$

The identity matrix $\mathbf{E}$ signifies the optimal scenario where the selected features exhibit no redundancy. A greater distance between $\mathbf{S}$ and $\mathbf{E}$ indicates higher redundancy among the selected features, whereas a closer distance implies lower redundancy.

B. Performance Comparison

We compare the accuracy of MUSE with the baseline methods on all datasets. We report the average cross-validation classification accuracy of each method with the number of selected features $|\mathcal{S}| = 50$. Table III shows the results evaluated by the SVM classifier, and Table IV shows the evaluated results by the KNN classifier.

The results with the highest accuracy are indicated in bold for each dataset (column). In the last column of Tables III and IV, the performance of each method is compared with MUSE (mRMR) in the number of datasets that MUSE (mRMR) has won, tied, or lost (W/T/L), respectively. OFS represents the approach where the model is trained on raw data, while AFP serves as a method trained on vertically federated data with numerous overlapping features. From Tables III and IV, we observe that in most cases, the performance of AFP is significantly lower than that of OFS, except on datasets colon, lymphoma, and NCI9 trained by SVM. This suggests that overlapping features may adversely affect the model's performance,

TABLE IV
PREDICTION ACCURACY OF VARIOUS METHODS EVALUATED BY KNN

	Colon	Lymphoma	NCI9	Pixraw10P	RELATHE	Srbct	TOX-171	USPS	Yale	W/T/L
OFS	71.67	90.00	33.33	91.00	75.35	78.75	79.41	96.35	52.50	5/0/4
AFP	58.33	52.22	25.00	90.00	56.76	77.50	72.35	91.28	18.75	7/0/2
SFFS	68.33	71.11	26.67	89.00	56.76	73.75	50.00	80.47	45.62	8/0/1
FATE-IV	68.33	64.44	**40.00**	41.00	65.49	86.25	57.06	55.36	43.75	8/1/0
FATE-Lasso	70.00	74.44	18.33	**90.00**	61.41	78.75	**74.71**	91.16	45.00	6/0/3
MUSE (mRMR)	**80.00**	**86.67**	**40.00**	87.00	**79.08**	**92.50**	73.53	90.85	**53.12**	–
MUSE (MIFS)	71.67	72.22	23.33	77.00	79.01	86.25	68.82	94.43	36.25	8/0/1
MUSE (DD)	70.00	77.78	23.33	88.00	74.44	71.25	62.94	**95.59**	50.00	7/0/2

Note: MUSE (mRMR), MUSE (MIFS), and MUSE (DD) are the vertical federated feature selection framework MUSE under mRMR, MIFS, and DD criteria, respectively. The results with the highest accuracy are indicated in bold for each dataset (column).

TABLE V
REDUNDANCY COMPARISON

	Colon	Lymphoma	NCI9	Pixraw10P	RELATHE	Srbct	TOX-171	USPS	Yale	W/T/L
SFFS	2.16	**1.83**	**1.70**	6.89	2.65	4.78	4.35	7.58	4.27	7/0/2
FATE-IV	4.09	11.71	5.83	19.78	2.93	14.49	20.08	10.82	21.73	9/0/0
FATE-Lasso	2.17	1.89	1.86	10.76	2.86	5.26	4.38	4.43	4.42	7/1/1
MUSE (mRMR)	1.91	1.89	1.90	**5.31**	2.41	**4.71**	**4.27**	**4.01**	**4.16**	–
MUSE (MIFS)	2.59	2.82	2.31	23.93	**1.98**	20.16	14.61	9.31	14.11	8/0/1
MUSE (DD)	**1.83**	1.91	1.82	15.04	2.27	19.33	13.37	8.23	8.21	6/0/3

Note: The results with the highest accuracy are indicated in bold for each dataset (column).

emphasizing the importance of reducing both overlapping and redundant features. However, there are instances where AFP outperformed OFS. This could be attributed to factors such as the beneficial combination of overlapping features for training SVM. Furthermore, the best performance of all the feature selection methods on each dataset is not consistently superior to that of OFS, as observed in cases such as Srbct, TOX-171, USPS, and Yale.

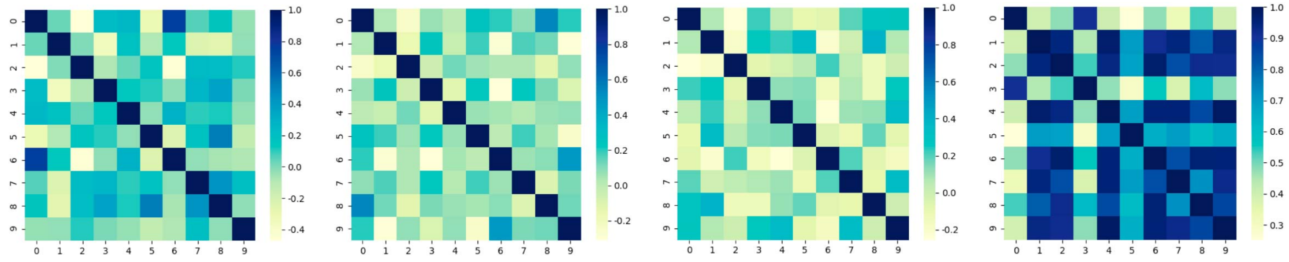
This suggests that the number of selected features may not be sufficient to adequately represent all features. To address this issue, one potential solution is to increase the threshold for the number of selected features to $|\mathcal{S}| > 50$. Additionally, not all MUSE methods outperform other baseline methods. For instance, while MUSE (mRMR) generally exhibits better performance than the baselines, MUSE (MIFS) and MUSE (DD) demonstrate comparable performance. This can be attributed to the fact that MUSE primarily serves as a framework for mutual information-based feature selection, enabling such methods to conduct feature selection in the vertical federated setting without privacy leakage. Nevertheless, the feature selection criterion remains pivotal in the performance of selected features.

C. Redundancy Comparison

We analyze the redundancy of selected features across all methods, focusing on the top 10 features selected by each method. Table V presents the redundancy scores computed by (15) for all methods on different datasets. The lowest redundancy scores are highlighted in bold for each dataset. In the last column of Table V, the performance of each method is compared with MUSE (mRMR) in the number of datasets that MUSE (mRMR) has won, tied, or lost (W/T/L), respectively. MUSE (mRMR) consistently demonstrates lower redundancy across most datasets, likely contributing to its superior performance compared to other methods. Conversely, MUSE (DD) and MUSE (MIFS) exhibit higher redundancy levels, possibly due to suboptimal hyperparameters for mitigating redundancy in this dataset. Among the baseline methods, FATE-IV stands out for its high redundancy, attributed to its neglect of feature interactions or redundant features, notably evident in Colon, Lymphoma, NCI9, TOX-171, USPS, RELATHE, and Yale datasets. In contrast, FATE-Lasso and SFFS show lower redundancy, indicating a more balanced selection of features with consideration for both feature correlation and task effectiveness. To illustrate the relationship of selected features, we visualize the Spearman correlation matrix S of selected features on the USPS dataset, as shown in Fig. 5. The darker the color in the heat map, the stronger the redundancy among the selected features.

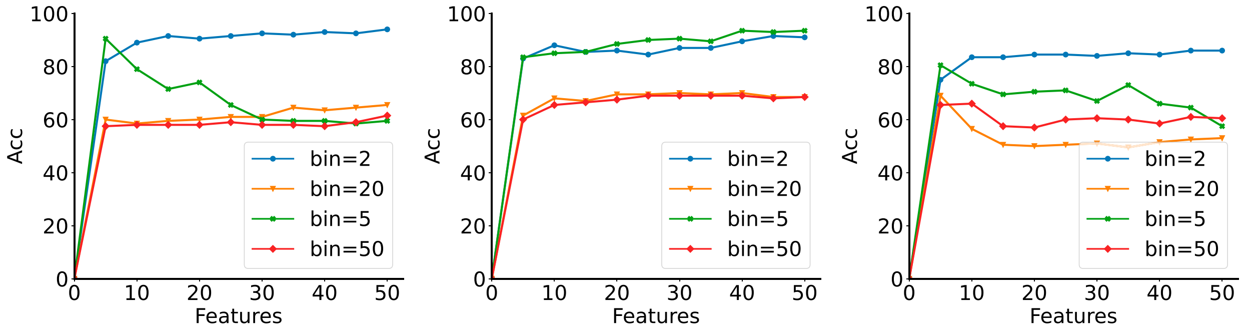
D. Effect of the Bin Number

In this section, we test how the bin number n in our algorithm influences the performance of MUSE. We run this part of the experiments on the Lung dataset to observe how changing n affects the classification results of selected features. We test all MUSE methods by cross-validation with the SVM and KNN classifiers. We change the selected features $|S|$ from 0 to 50 at intervals of 5 with a fixed value $n = 2, 5, 20, 50$. Fig. 6 shows the results evaluated by the SVM classifier and Fig. 7 shows the evaluated results evaluated by the KNN classifier.



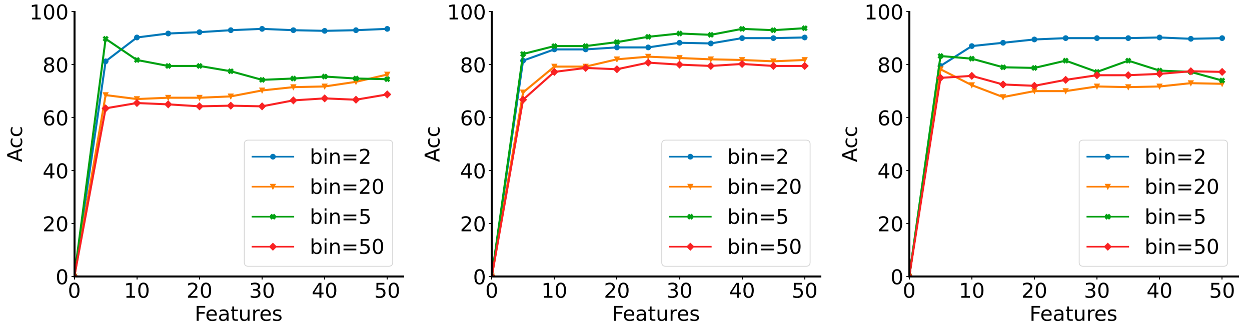
(a) (b) (c) (d)

Fig. 5. Heat-map for different methods in the USPS dataset. It shows the redundancy of different feature selection methods. The darker the color in the heat map, the stronger the redundancy among the selected features. (a) SFFS. (b) FATE-Lasso. (c) MUSE (mRMR). (d) FATE-IV.



(a) (b) (c)

Fig. 6. Effect of bin number n to MUSE evaluated by SVM in the lung dataset. (a) MUSE (MIFS). (b) MUSE (mRMR). (c) MUSE (DD).



(a) (b) (c)

Fig. 7. Effect of bin number n to MUSE evaluated by KNN in the lung dataset. (a) MUSE (MIFS). (b) MUSE (mRMR). (c) MUSE (DD).

The analysis from Figs. 6 and 7 suggest that, for most feature selection methods within the MUSE framework, setting the parameter number of bins $n = 2$ yields superior performance when evaluated using both KNN and SVM classifiers compared to other values. This observation indicates that $n = 2$ is the preferred choice for the dataset under consideration over larger or smaller values of n. However, MUSE (mRMR) deviates from this trend. It demonstrates optimal performance when the number of bins is set to $n = 5$, outperforming other bin numbers. Interestingly, the performance of MUSE (mRMR) seems less sensitive to changes in the number of bins compared to MUSE (DD) and MUSE (MIFS). This suggests a higher level of stability for MUSE (mRMR) across different bin numbers. On the other hand, MUSE (MIFS) and MUSE (DD) consistently demonstrate similar performance across different bin numbers. This uniformity in results may be attributed to the utilization of fixed hyperparameters (i.e. λ, β) aimed at balancing redundancy and relevance within their feature selection criteria.

In summary, the analysis highlights the significance of parameter tuning, particularly the choice of the number of bins, to the effectiveness of the MUSE framework. For example, with an appropriate parameter n, the prediction performance for selected features will increase significantly.

TABLE VI
Ablation Study Results of MUSE Evaluated by SVM

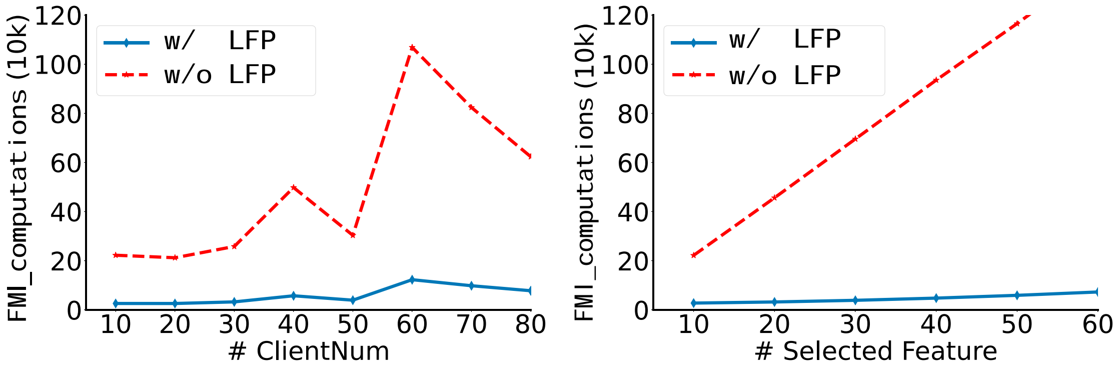
	Colon	Lymphoma	NCI9	Pixraw10P	RELATHE	Srbct	TOX-171	USPS	Yale
MUSE (mRMR)	65.00	88.89	33.33	86.00	68.80	95.00	71.76	91.31	66.25
MUSE (mRMR) w/o LFP	65.00	88.89	33.33	60.00	68.80	95.00	72.94	91.41	67.50
MUSE (MIFS)	35.00	70.00	13.33	84.00	53.80	87.50	66.47	93.63	38.12
MUSE (MIFS) w/o LFP	70.00	80.00	43.33	58.00	53.80	93.75	74.71	93.71	64.38
MUSE (DD)	35.00	82.22	13.33	87.00	53.80	85.00	68.24	94.41	61.25
MUSE_DD w/o LFP	41.67	80.00	18.33	61.00	53.80	96.25	77.65	94.37	59.38

Note: MUSE and MUSE without local feature selection (MUSE w/o LFP) are compared.

TABLE VII
Ablation Study Results of MUSE Evaluated by KNN

	Colon	Lymphoma	NCI9	Pixraw10P	RELATHE	Srbct	TOX-171	USPS	Yale
MUSE (mRMR)	80.00	86.67	40.00	87.00	79.08	92.50	73.53	90.85	53.12
MUSE (mRMR) w/o LFP	80.00	88.89	40.00	60.00	79.08	92.50	72.94	90.99	53.12
MUSE (MIFS)	71.67	72.22	23.33	77.00	79.01	86.25	68.82	94.43	36.25
MUSE (MIFS) w/o LFP	71.67	87.78	41.67	62.00	79.08	91.25	78.24	94.52	46.88
MUSE (DD)	70.00	77.78	23.33	88.00	74.44	71.25	62.94	95.59	50.00
MUSE_DD w/o LFP	73.33	90.00	20.00	57.00	78.03	93.75	81.18	95.64	52.50

Note: MUSE and MUSE without local feature selection (MUSE w/o LFP) are compared.



(a)

(b)

Fig. 8. Number of FMI computations in MUSE with local feature preselection (w/ LFP) and without local feature preselection (w/o LFP) in different settings. (a) With parameter of clients. (b) Selected features.

E. Ablation Study

In this section, we run an ablation study to prove the effectiveness of local feature preselection (LFP) from two sides. The first side is to validate the effects of LFP on the performance of MUSE. The second side is to validate the effect of LFP on communication reduction.

1) Effects of LFP on Performance: We compare the performance of MUSE "with LFP" and "without LFP" on different datasets, as shown in Table VI. The results indicate that, compared with MUSE (MIFS) and MUSE (DD), "with LFP" achieves a prediction accuracy more similar to "without LFP" in MUSE (mRMR), except on the Pixraw10P dataset. The significant performance difference between "with LFP" and "without LFP" in MUSE on Pixraw10P clearly demonstrates that LFP is not suitable for this particular dataset. Furthermore, there is a notable performance gap between "with LFP" and "without LFP" in both MUSE (MIFS) and MUSE (DD), indicating that LFP's effectiveness varies across different feature selection methods in MUSE and datasets.

2) Effects of LFP on Communication Efficiency: To assess the impact of LFP on communication efficiency in MUSE, we compare the total number of FMI computations with/without LFP on the NCI9 dataset. Note that the amounts of FMI computations of each method in MUSE are the same because the amounts of FMI computations are not related to feature criteria. We analyze some possible factors affecting the number of FMI computations: the number of clients and the number of selected features. We show the number of FMI computations with/without LFP based on these factors in Fig. 8. Specifically, we analyze the number of FMI computations in MUSE with or without LFP while varying the number of clients and the number of selected features.

Fig. 8 shows that LFP can achieve significantly fewer FMI computations. Moreover, both “with LFP” and “without LFP” exhibit a similar trend in the number of FMI computations as the number of selected features or the number of clients increases. Specifically, Fig. 8(a) shows that the number of FMI computations increases linearly with the number of clients. In contrast, Fig. 8(b) demonstrates that the number of FMI computations does not increase linearly with the number of clients. This is because the number of FMI computations is also influenced by the distribution of features across clients. From Fig. 8, we can conclude that the communication and computational overhead of MUSE without LFP could become significant as the number of clients or the number of features increases. However, LFP can improve the scalability of MUSE to handle large-scale federated learning environments.

V. Discussion

In summary, our experimental results demonstrate that MUSE outperforms the state-of-the-art methods in the Accuracy of most datasets. Moreover, MUSE can select less redundant features, compared with other state-of-the-art methods. To enhance the practical deployment of MUSE, this section provides insights into real-world applications and limitations of MUSE.

A. Applications

The MUSE framework is particularly well-suited for vertical federated learning (VFL) scenarios where data is distributed across different parties with distinct feature sets but shared user IDs. These scenarios are common in industries such as finance, healthcare, and telecommunications. For instance:

1) *Finance:* Multiple financial institutions may wish to collaborate to improve fraud detection algorithms. Each institution holds different types of data (e.g., transaction history, credit scores, and personal information) for the same set of customers. Using MUSE, these institutions can securely compute mutual information and select relevant features without exposing sensitive customer data, thereby enhancing the overall fraud detection model.
2) *Healthcare:* Hospitals and research institutions often hold diverse patient data (e.g., medical history, genetic information, and treatment plans). By using MUSE, these entities can collaborate to identify key predictors of diseases or treatment outcomes, while preserving patient privacy and complying with regulations such as health insurance portability and accountability act (HIPAA).
3) *Telecommunications:* Telecom companies can benefit from MUSE to improve customer service and predict churn by combining their proprietary feature data selected by MUSE (e.g., call records, internet usage, and service requests) with feature data selected by MUSE from other sources without risking customer data privacy.

B. Limitations

In this study, we have demonstrated the effectiveness of our methods across various datasets. However, MUSE does have some limitations, including two aspects.

1) Given the rising importance of multimodal data, which integrates information from multiple sources such as text, images, and audio, there is significant potential for MUSE in this area. However, handling multimodal data introduces complexity in feature selection and redundancy reduction across different modalities. While our methods have shown strong performance in selecting task-related features and reducing redundant features, extending our approach to multimodal datasets will require addressing these complexities and optimizing our techniques for heterogeneous data sources.
2) MUSE utilizes a mutual information (MI)-based measurement to support various MI feature selection criteria. There are performance differences between these MI feature selection criteria. From the experimental results of mRMR, MIFS, and DD, the performance of MUSE under the mRMR criterion is best, which can be seen in the experiments why FATE-Lasso shows competitive performance and even outperforms MUSE (DD) and MUSE (MIFS) in both metrics in our experiments. This highlights criteria such as minimum redundancy maximum relevance are effective. In contrast, others may not yield the same improvements.

VI. Conclusion and Future Work

This article reveals that noisy and redundant features could affect the performance of models in vertical FL. To address this issue, we propose a privacy-preserving vertical federated feature selection framework named MUSE, which selects a compact and effective feature set to enhance the accuracy of FL models. MUSE can identify high task-related features and eliminate redundant ones without causing data leakage.

This is achieved by estimating the correlation of cross-device feature–feature and feature-class relationships using our defined privacy-preserving mutual information metric, called FMI. In detail, the ABM algorithm is first proposed to compute FMI while avoiding *sample-IDs* leakage. Second, the LFP is proposed to reduce the number of FMI computations, thereby improving the efficiency of MUSE.

MUSE is evaluated on ten different types of datasets and compared with five baseline methods. The experimental results demonstrate that MUSE exhibits a significant improvement in the classification performance of the model compared to state-of-the-art methods in most cases and effectively reduces the redundant features as well. Moreover, experimental results indicate that LFP in MUSE is an effective strategy for reducing the number of FMI computations.

Our work assumes clean labels for the classification task. However, noisy or missing labels are common in real-world data due to various factors such as human error, data corruption, and incomplete data collection processes. In the future, we aim to address these issues to improve the robustness of the MUSE framework.

References

[1] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, and E. A. Ivanov, "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst.*, 2019, pp. 374–388.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, pp. 50–60, 2020.

[3] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, pp. 46–51, 2020.

[4] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, pp. 1–210, 2021.

[5] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Comput. Ind. Eng.*, vol. 149, p. 106854, 2020.

[6] L. T. Yang, R. Zhao, D. Liu, W. Lu, and X. Deng, "Tensor-empowered federated learning for cyber-physical-social computing and communication systems," *IEEE Commun. Surv. Tut.*, vol. 25, pp. 1909–1940, 2023.

[7] W. Gong, J. Liu, and Z. Yang, "Fast and reliable unknown tag detection in large-scale RFID systems," in *Proc. 17th ACM Int. Symp. Mobile Ad hoc Netw. Comput.*, 2016, pp. 141–150.

[8] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," *Data Class. Algorithms Appl.*, pp. 37–64, 2014.

[9] J. Li et al., "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, pp. 1–45, 2017.

[10] V. Kumar and S. Minz, "Feature selection: a literature review," *SmartCR*, vol. 4, pp. 211–229, 2014.

[11] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[12] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, pp. 131–156, 1997.

[13] Q. Lin et al., "Has multimodal learning delivered universal intelligence in healthcare? A comprehensive survey," 2024, *arXiv:2408.12880*.

[14] I. Ahmed, G. Jeon, and F. Piccialli, "From artificial intelligence to explainable artificial intelligence in industry 4.0: A survey on what, how, and where," *IEEE Trans. Ind. Inform.*, vol. 18, no. 8, pp. 5031–5042, 2022.

[15] P. Cassará, A. Gotta, and L. Valerio, "Federated feature selection for cyber-physical systems of systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9937–9950, 2022.

[16] S. Banerjee, E. Elmroth, and M. Bhuyan, "FED-FIS: A novel information-theoretic federated feature selection for learning stability," in *Proc. Conf. Neural Inf. Process.*, 2021, pp. 480–487.

[17] Q. Lin et al., "Contrastive graph representations for logical formulas embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 35, pp. 3563–3574, 2021.

[18] F. Xu et al., "Symbol-LLM: Towards foundational symbol-centric interface for large language models," in *Proc. 62st Annu. Meeting Assoc. Comput. Linguistics*, 2024, pp. 13091–13116.

[19] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectron.*, 2015, pp. 1200–1205.

[20] C. Liu, D. Jiang, and W. Yang, "Global geometric similarity scheme for feature selection in fault diagnosis," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3585–3595, 2014.

[21] F.-Y. Wang and Y. Wang, "Parallel ecology for intelligent and smart cyber–physical–social systems," *IEEE Trans. Comput. Social Syst.*, vol. 7, pp. 1318–1323, 2020.

[22] K. Liu, Q. Ma, W. Gong, X. Miao, and Y. Liu, "Self-diagnosis for detecting system failures in large-scale wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 13, pp. 5535–5545, 2014.

[23] Z. Zhou, X. Zhou, B. Guo, S. Wang, and T. He, "Multi-sensor data-driven route prediction in instant delivery with a 3-conversion network," *ACM Trans. Sensor Networks*, vol. 20, no. 2, pp. 1–21, Mar. 2024.

[24] Y. He, X. Tan, J. Ni, L. T. Yang, and X. Deng, "Differentially private set intersection for asymmetrical ID alignment," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 3479–3494, 2022.

[25] F. Pan, D. Meng, Y. Zhang, H. Li, and X. Li, "Secure federated feature selection for cross-feature federated learning," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020.

[26] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, pp. 1–19, 2019.

[27] K. Cheng et al., "Secureboost: A lossless federated learning framework," *IEEE Intell. Syst.*, vol. 36, no. 6, pp. 87–98, Nov./Dec. 2021.

[28] M. Banerjee and S. Chakravarty, "Privacy preserving feature selection for distributed data using virtual dimension," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 2281–2284.

[29] X. Li, R. Dowsley, and D. C. M., "Privacy-preserving feature selection with secure multiparty computation," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 6326–6336.

[30] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 1226–1238, 2005.

[31] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, pp. 537–550, 1994.

[32] S. Zadeh, M. Ghadiri, V. Mirrokni, and M. Zadimoghaddam, "Scalable feature selection via distributed diversity maximization," in *Proc. Assoc. Advancement Artif. Intell.*, 2017, pp. 2876–2883.

[33] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[34] Z. Zhu, J. Zhu, J. Liu, and Y. Liu, "Federated bandit: A gossiping approach," in *Proc. ACM Meas. Anal. Comput. Syst.*, 2021, pp. 3–4.

[35] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, pp. 3710–3722, 2021.

[36] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-IID data," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, pp. 3400–3413, 2019.

[37] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Trans. Neural Networks*, vol. 20, pp. 189–201, 2009.

[38] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "MIFS-ND: A mutual information-based feature selection method," *Expert Syst. Appl.*, vol. 41, pp. 6371–6385, 2014.

[39] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 856–863.

[40] H. Chen, K. Laine, and P. Rindal, "Fast private set intersection from homomorphic encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1243–1255.

[41] B. Pinkas, T. Schneider, G. Segev, and M. Zohner, "Phasing: Private set intersection using permutation-based hashing," in *Proc. 24th USENIX Secur. Symp. (USENIX Secur. 15)*, 2015, pp. 515–530.

[42] M. O. Rabin, "How to exchange secrets with oblivious transfer," *Cryptol. ePrint Arch.*, p. 187, 2005.

[43] V. Kolesnikov, R. Kumaresan, M. Rosulek, and N. Trieu, "Efficient batched oblivious PRF with applications to private set intersection," in *Proc. 2016 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 818–829.

[44] M. Chase and P. Miao, "Private set intersection in the internet setting from lightweight oblivious PRF," in *Proc. Conf. Annu. Int. Cryptol.*, 2020, pp. 34–63.

[45] M. Goemans, "Chernoff bounds, and some applications," 2015. [Online]. Available: http://math.mit.edu/goemans/18310S15/chernoff-notes.pdf

[46] B. Birnbaum and K. J. Goldman, "An improved analysis for a greedy remote-clique algorithm using factor-revealing LPS," *Algorithmica*, vol. 55, pp. 42–59, 2009.

[47] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[48] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "Fate: An industrial grade platform for collaborative learning with data protection," *J. Mach. Learn. Res.*, vol. 22, pp. 1–6, 2021.

[49] R. A. Howard, "Information value theory," *IEEE Trans. Syst. Sci. Cybern.*, vol. 2, pp. 22–26, 1966.

[50] J. Hauke and T. Kossowski, "Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data," *Quaest. Geogr.*, vol. 30, no. 2, pp. 87–93, 2011.

Xinyuan Ji received the B.S. degree from Shanxi University, Taiyuan, China, and the M.S. degree from Shaanxi Normal University, Xian, China, in 2016 and 2019, respectively. She is currently working toward the Ph.D. degree with Xi'an Jiaotong University, Xian, China, and Leiden University, Leiden, The Netherlands, all in computer science and technology.

Her research interests include distributed computing, data mining, machine learning, and privacy & security.

Chenfei Wang received the B.S. degree in computer science & technology and the M.S. degree in electrical engineering & automation from Xi'an Jiaotong University, Xian, China, in 2018, and 2021, respectively.

Currently, he is working with Baidu Kuike Science and Technology Building, Beijing, China. His research interests include federated learning and machine learning.

Olga Gadyatskaya received the Ph.D. degree in mathematics from Novosibirsk State University, Novosibirsk, Russia.

She is an Associate Professor with Leiden Institute of Advanced Computer Science, Leiden University. Prior to joining Leiden University, Leiden, The Netherlands, she was a Research Associate with the University of Luxembourg, Luxembourg, and a Postdoc with the University of Trento, Trento, Italy. Her research focuses on cybersecurity, and her research interests include security risk management, mobile security, secure AI systems, and security decision-making in organizations.

Fei Zhao received the B.S. degree from Qingdao University, China, in 2019. He is currently working toward the M.S. degree with Xi'an Jiaotong University, Xian, China, both in software engineering.

His research interests include privacy computing and federated learning.

Zixiang Mao received the B.S. degree in computer science and technology from Dalian University of Technology, Dalian, China, in 2019. He is currently working toward the M.S. degree with the Department of Computer Science and Technology, Xi'an Jiaotong University, Xian, China.

His research interests include federated learning and graph neural networks.

Wei Xi received the Ph.D. degree in computer science from Xi'an Jiaotong University, Xian, China, in 2014.

Currently, he is an Associate Professor with the School of Computer Science and Technology, Xi'an Jiaotong University. His research interests include the Internet of Things, artificial intelligence, and network security.

He is a member of CCF and ACM.