Secure Federated Feature Selection For Cross-Feature Federated Learning

Fucheng Pan[1,2], **Dan Meng**[2], **Yu Zhang**[2], **Hongyu Li**[2],**Xiaolin Li**[2,*]
[1]East China Normal University,ShangHai, China
[2]AI Institute,Tongdun Technology, China

Abstract

Cross-feature federated learning(FL) aims to combine all the features from different parties to train a global model with better performance, while without compromising user privacy and security. However, existing works rarely consider the feature selection among different parties, which is an important and inevitable process for machine learning. Consequently, it not only introduces feature noise during the training process, but also reduces the effectiveness for FL. To address these problems, in this paper, we propose a secure federated feature selection method (SFFS) for cross-feature FL. SFFS allows features in all parties to be selected effectively and enables flexible and effective models to be build. Rather than previous methods, SFFS implements a federated feature selection algorithm based on secure multi-party computation. To the best of our knowledge, it is the first study that considers both security and all parties' features for federated feature selection. As a plug-and-play algorithm, SFFS can be easily integrated into various cross-feature FL models. Extensive experiments show great prospects of SFFS.

1 Introduction

The success of machine learning rests on the availability of massive amounts of data. However, it limits machine learning's capability to deal with real-world applications, where data has been isolated across different organizations and data privacy has been emphasized. To address the above challenge problem, Google introduced a federated learning (FL) system [1], in which a global machine learning model is updated by the distributed participants while their raw data must be kept locally. As a new distributed and privacy-preserving machine learning (ML) paradigm, FL is well suited for the "data island" scenarios and has been attracting growing attention.

According to the distribution of data, FL methods can be classified into three types: cross-sample FL [2, 3, 4], cross-feature FL [5, 7, 8], and hybrid FL [9, 10, 6]. Cross-sample FL consider the scenario where each party has data with different sample IDs but shares many common features. Therefore, they can collaboratively learn a joint mapping from the feature space to the label space. Unlike cross-sample FL, in cross-feature FL, multiple parties handle data with the same sample IDs, but each party has its own feature set and only one party holds labels. This is a common phenomenon in e-commerce, financial, and healthcare applications. For example, an e-commerce company may want to predict a customers credit using her/his historical transactions from multiple financial institutions, and a healthcare company wants to evaluate the health condition of a particular patient using his/her clinical data from various hospitals. So by combining multi-view features of customer or patient, companies can establish a more accurate prediction model.

When compared with cross-sample FL, cross-feature FL has its unique properties and challenges, such as model modification, and security design. Thus, there arise solutions for solving efficiency

[*] Corresponding Author

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

and data privacy problems in cross-feature FL [11, 12]. Nevertheless, on the one hand, these methods often ignore performing federated feature selection before constructing machine learning models. In this case, the feature space used locally for modeling will be inconsistent with the global feature space, which often introduces feature noise, resulting in a decrease in accuracy and excessive parameter transmission. On the other hand, different from cross-sample FL, cross-feature FL' labels always are holded in one party. So it is impossible to select effective local feature through supervised learning in every parties's FL training process.

In this paper, we propose secure federated feature selection method (SFFS) for cross-feature FL to solve the above problems. It leverages global features to select suitable features without compromising user privacy and security under a federated paradigm. Briefly, we summarize our main contributions into three folds:

- To the best of our knowledge, we introduce SFFS for cross-feature FL for the first time, which effectively avoids feature noise and improves the robustness of cross-feature FL.
- We formalize the research problem of SFFS in a privacy-preserving setting to provide solutions for federation problems beyond the scope of existing cross-feature FL approaches.
- Through extensive experiments in different cross-feature FL approaches, including cross-feature Logistic Regression (CF-LR) [32], cross-feature XGBoost (CF-XGB) [33], and cross-feature Neural Network (CF-NN) [1], the effectiveness of SFFS has been proved in many aspects.

2 Related Work

Privacy is one of the essential properties of FL. Thus, Secure Multi-party Computation (SMC) has been widely used in FL contributes to its security property. SMC models involve multiple parties and provide security proof in a well-defined simulation framework to guarantee complete zero knowledge, that is, each party knows nothing except its input and output. Zero knowledge is highly desirable, but this desired property usually requires complicated computation protocols and may not be achieved efficiently. In certain scenarios, partial knowledge disclosure may be considered acceptable if security guarantees are provided. It is possible to build a security model with SMC under lower security requirements in exchange for efficiency [13]. Recently, a study [14] used the SMC framework to train machine-learning models with two parties under semi-honest assumptions. SMC protocols are also used in [15] for model training and verified without revealing sensitive data. One of the state-of-the-art SMC frameworks is Sharemind [16], which is an efficient general-purpose computation system to process sensitive data and make large-scale share computing feasible in practice. Besides, the authors of [17] proposed a 3PC model [18, 19] with a honest majority and considered security under both semi-honest and malicious assumptions. These works ensure participants data to be secretly shared among non-collusion participants, which reveals an important way to protect privacy.

Feature selection plays a fundamental role in tradition machine learning tasks. It can be divided into three types, namely the wrapper methods [20, 21, 22], the embedded methods [23, 24, 25], and the filter methods [26, 27, 28]. Wrapper methods follow heuristic guidelines to select a subset of features with the best prediction performance. As the number of possible subsets may be tremendously large and new classifiers should be established for each updated subset, the wrapper methods are generally computationally expensive. Embedded methods, although faster than wrapper methods, are still computationally heavy, and feature selection results depend on selected learning algorithm [29]. Filter feature selection methods score and rank feature candidates in accordance with a certain criterion, and extract one feature at a time to form a subset with a predefined dimension. They are computationally cheaper and do not rely on particular predictors. In FL setting, computational cost is one of the primary concern. Intuitively, filter methods are always more suitable for FL tasks, which can achieve a balance between computational cost and efficiency. On the other hand, consider global features from different parties, features need to be secretly shared. Fortunately, SMC paves the way for federated feature selection methods by secrete sharing.

In this paper, we take advantages of SMC and filter feature selection methods from security and efficiency aspects for federated feature selection, which is a crucial process in cross-feature FL. To the best of our knowledge, SFFS is the first method that enables cross-feature FL to benefit from feature selection.

3 The Proposed Approach

In this section, we present details of our proposed approach-**Security Federated Feature Selection** (SFFS). Firsty, we give some preliminaries definition for SFFS. Secondly, more details of SFFS using SMC will be given.

3.1 Preliminaries Definition

To simplify the description, in this paper, we only consider three parities for cross-feature FL. It is worth noting that, the proposed SFFS can be extended to multiple parties.

Suppose we have three parties, party A, B, and C. Party A and B are the data provider, which has the dataset $D_A := \{(x_A^i, y_A^i)\}_{i=1}^{N_A}$, and $D_B := \{x_B^i\}_{i=1}^{N^B}$ respectively. Besides, $x_A^i \in \mathbb{R}^a$ and $x_B^i \in \mathbb{R}^b$. We also assume that there exists a co-occurrence samples set $D_C := \{(x_A^i, x_B^i, y_A^i)\}_{i=1}^{N_C}$, defining $X_A \in \mathbb{R}^{N_C \times a}$, $X_B \in \mathbb{R}^{N_C \times b}$, and $Y_A \in \mathbb{R}^{N_C \times 1}$ as the matrix of data and label. Without loss of generality, we assume all labels are in party A, but all the deduction in this paper can be adapted to the case where labels hold by party B. Party C plays the role of helping party A an B to generate random matrices and transfer parameters. We ensure D_A and D_B are separately held by two parties locally, and cannot be exposed to each other. Given the above settings, the objective here is to design a SFFS for cross-feature FL, by using SMC while with no risk of exposing data to each other.

In SFFS, we use feature importance as the criterion for FL features selection. We briefly introduce the related concepts in this part. It is well known that not all features are effective for building a classification model. Furthermore, not all features are of equal importance. Since linear model is as widely used method to measure feature importance, our goal is to select valuable features with the help of linear model. Suppose there are m features, and the linear model can be written as $y = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 + \cdots + \Theta_m x_m$. Obviously, if an independent feature $x_j, j = 1, 2, \cdots, m$ has little or no significant effect on constructing the predicting model, its coefficient Θ_j should close to 0. In order to obtain the importance of each feature, we use F-statistic to calculate feature importance based on its coefficient Θ_j. Considering that the relevant proof is complicated and beyond the scope of this paper, please refer to [30, 31] for more details.

3.2 SFFS Using Secure Multiparty Computation

In this section, we introduce our proposed method in details. As mentioned before, filter methods have been widely adopted by feature selection in tradition machine learning tasks. In SFFS, we gradually remove features with small F-statistic. Before calculating the statistics, we need to estimate the model parameters. when all data $\{X, Y\}$ is in a party, We use least squares method to estimate model parameters $\Theta = (X^T X)^{-1} X^T Y$. So, obvious, The key to the above calculation is how to calculate the covariance matrix $(X^T X)^{-1}$ and $X^T Y$. However, here, we explore a more challenge scenario, in which party A and B cannot expose its own data to each other, but can still fulfill feature selection among all features. So in two parties A, B, $X^T Y$ and $X^T Y$ can be describe as

$$
\begin{aligned}
X^T X &= \begin{bmatrix} X_A^T \\ X_B^T \end{bmatrix} \begin{bmatrix} X_A, & X_B \end{bmatrix} = \begin{bmatrix} X_A^T X_A & X_A^T X_B \\ X_B^T X_A & X_B^T X_B \end{bmatrix} \\
X^T Y &= \begin{bmatrix} X_A^T \\ X_B^T \end{bmatrix} Y_A = \begin{bmatrix} X_A^T Y_A \\ X_B^T Y_A \end{bmatrix}
\end{aligned}
\tag{1}
$$

As shown in Eq. (1), A and B can get $X_A^T X_A$ and $X_B^T X_B$ independently. But $X_A^T X_B$ and $X_B^T Y_A$ must requires collaborative calculation of A and B. So SFFS designs a subtle multi-party security protocol to safely calculate under the premise that both parties A and B do not expose their own data $X_A^T Y_A$ and $X_B^T X_B$. Algorithm 1 shows the working overflow of SFFS and describe the specific operations of each party.

As shown in Algorithm 1, SSFS uses the idea of secret sharing and first splits feature data matrix of each participant separately. Into matrix fragments, the splitting requirement is that the sum of

Algorithm 1 SFFS Using Secure Matrix Multiplication.

```
Require: $\{(x_A^i, y_A^i)\}_{i=1}^{N_C} = X^A$ with dimensions $N_C \times (a+1)$, $Y_A = \{y_A^i\}_{i=1}^{N_C}$ with dimensions
    $N_C \times 1$ ; $\{x_B^i\}_{i=1}^{N_C} = X_B$ with dimensions $N_C \times b$
Ensure: $F_j$.the F statistics of each feature
$\boldsymbol{C}$ do
generate random matrix $R_a$, $r_a$ and $r_b$, where $R_a \in \mathbb{R}^{N_C \times (a+1)}$, $r_a \in \mathbb{R}^{(a+1) \times b}$, $R_b \in \mathbb{R}^{N_C \times b}$,
    $r_b \in \mathbb{R}^{(a+1) \times b}$, $r_a + r_b = R_a^T R_b$ and send A and B
$\boldsymbol{A}$ do
calculate $X_A^{''} = [X_A, Y_A]$, $X_A^{'} = X_A^{''} + R_a$, and send $X_A^{'}$ to B
$\boldsymbol{B}$ do
generate random matrix $V_b \in \mathbb{R}^{(a+1) \times b}$
compute $X_B^{'} = X_B + R_b$, $\gamma = (X_A^{'})^T X_B + r_b - V_b$, and send $V_b$, $X_B^{'}$, $\gamma$ to A
$\boldsymbol{A}$ do
calculate $V_a = \gamma + r_a - R_a X_B^{'}$, $V = V_a + V_b = [X_A Y_A]^T X_B$
$\boldsymbol{B}$ do
$X_B^T X_B$ and send $X_B^T X_B$ to A
$\boldsymbol{A}$ do
obtain $\Theta = \chi\kappa$ using Eq. (2) and (3)
denote $\Theta = [\Theta_A, \Theta_B]$, and send $\Theta_B$ to B
$\boldsymbol{B}$ do
send $X_B \Theta_B$ to A
$\boldsymbol{A}$ do
get $Y^{'}$ by Eq. (4) and $\Omega = (Y_A - Y^{'})^2$
obtain $F_j$. $j = 1, 2, \cdots, a+b$ using Eq. (5)
return $F_j$
```

matrix fragments is equal to the characteristic data matrix. To meet the requirements, we use trusted third party C generates random matrices R_a, r_a and r_b, where $R_a \in \mathbb{R}^{N_C \times (a+1)}$, $r_a \in \mathbb{R}^{(a+1) \times b}$, $R_b \in \mathbb{R}^{N_C \times b}$, $r_b \in \mathbb{R}^{(a+1) \times b}$, $r_a + r_b = R_a^T R_b$. Besides, party C sends R_a, r_a to party A, and sends R_b, r_b to party B. So A can calculate matrix fragments $X_A^{'}$ and send to party B. Correspondingly, B also calculate matrix fragments $X_B^{'}$, γ and generates random matrix V_b. in these protocols, we obtains $V_a = \gamma + r_a - R_a X_B^{'}$, and obtains intermediate results $V = V_a + V_b = [X_A Y_A]^T X_B$, where $X_A^T X_B = V[: -1, :]$, $Y X_B^T = V[-1, :]$, Next, A can calculate $X_A^T X_A$, and $X_B^T X_B$ to get the variance-covariance matrix χ and the least square estimation $\Theta = \chi\kappa$ using Eq.(2) and Eq.(3).

$$\chi = \begin{bmatrix} X_A^T X_A & X_A^T X_B \\ X_B^T X_A & X_B^T X_B \end{bmatrix} = \begin{bmatrix} X_A^T X_A & V[: -1, :] \\ V[: -1, :]^T & X_B^T X_B \end{bmatrix} \tag{2}$$

$$\kappa = \begin{bmatrix} X_A^T \\ X_B^T \end{bmatrix} Y_A = \begin{bmatrix} X_A^T Y_A \\ X_B^T Y_A \end{bmatrix} = \begin{bmatrix} X_A^T Y_A \\ V[-1, :]^T \end{bmatrix} \tag{3}$$

Finally, SFFS successfully obtained estimated parameters $\Theta = \chi\kappa$ through secret sharing. Θ can be splited for $[\Theta_A, \Theta_B]$, So party A can obtain the label $Y^{'}$ using Eq.(4).

$$Y^{'} = X_A \Theta_A + X_B \Theta_B \tag{4}$$

Party A obtains the mean square error $\Omega = (Y_A - Y^{'})^2$, where Y_A and $Y^{'}$ are the the real and predicted label. At this time, SFFS uses F-statistic to measure each feature's importance F_j. $j = 1, 2, \cdots, a+b$.

$$F_j = \frac{(\Theta_j)^2 / \chi_{jj}}{\Omega / (N_c - s - 1)} \tag{5}$$

where N_c is the number of samples, $s = a+b$ is the number of total features, χ_{jj} is the corresponding diagonal element in the variance-covariance matrix χ. SFFS will verifies whether the minimum

Table 1: Description Of Datasets

Datasets	Instances	Features	classes	Type	Field
RELATHE	1427	4322	2	Discrete	Text data
arcene	200	10000	2	Continuous	Other data
lymphoma	96	4026	9	Discrete	Biological data
GLI-85	85	22283	2	Continuous	Biological data
SMK_CAN_187	187	19993	2	Continuous	Biological data
Prostate-GE	102	5966	2	Continuous	Biological data
ALLAML	72	7129	2	Continuous	Biological data
CLL-SUB-111	111	11340	3	Continuous	Biological data
TOX-171	171	5748	4	Continuous	Biological data

Table 2: Average Accuracy(mean±std) for CF-LR, CF-XGB, CF-NN in different datasets(%)

Datasets	CF-LR		CF-XGB		CF-NN	
	with SFFS	without SFFS	with SFFS	without SFFS	with SFFS	without SFFS
RELATHE	**87.39 ± 0.23**	85.46 ± 0.45	**86.87± 0.22**	86.51 ± 0.34	**90.02 ± 0.32**	89.32 ± 0.52
arcene	**77.6± 1.20**	74.17± 1.08	**74.52± 1.13**	71.67 ± 1.12	**89.77± 1.27**	89.50 ± 0.97
lymphoma	**85.09 ± 0.96**	79.31 ± 1.15	**84.34 ± 1.03**	79.31 ± 1.09	**91.28 ± 1.54**	87.05 ± 1.08
GLI-85	**89.36 ± 1.28**	86.27 ± 1.16	**80.52 ± 1.19**	74.51± 1.24	**87.89 ± 1.05**	83.05 ± 0.86
SMK_CAN_187	**72.26 ± 1.23**	66.37 ± 1.42	**74.36 ± 1.56**	67.26 ± 1.44	**77.43 ± 1.58**	76.40 ± 1.24
Prostate-GE	**98.89 ± 0.56**	98.39 ± 0.75	**92.86 ± 1.58**	90.32± 0.25	**92.78 ± 0.62**	92.68 ± 0.76
ALLAML	**94.56 ± 0.78**	86.36 ± 0.67	**96.75 ± 0.56**	90.91 ± 0.67	**95.78 ± 0.82**	96.55 ± 0.28
CLL-SUB-111	**69.18 ± 2.52**	61.19± 2.32	**82.15 ± 1.56**	78.26 ± 1.36	**78.92 ± 2.08**	74.40 ± 1.85
TOX-171	**83.45 ± 1.38**	81.55 ± 2.16	**83.67 ± 1.37**	82.86 ± 1.76	**78.45 ± 1.25**	73.33 ± 1.67

value of the F statistic satisfies the F test. If so, the whole process of SFFS is done. Otherwise, party A or B should delete the specific feature, and repeats above process.

From the above steps, SFFS can pick out suitable features securely and efficiently to further fulfill FL task.

4 Experiment Evalution

4.1 Datasets And Experiment Setup

In order to verify our proposed SFFS, we apply SFFS to three classical cross-feature FL models, including cross-feature Logistic Regression (CF-LR) [32], cross-feature XGBoost (CF-XGB) [33], and cross-feature Neural Network (CF-NN) [1]. We conduct experiments based on the above three FL models with and without SFFS on 9 real-world high-dimensional datasets, which come from different fields [34]. Table 1 illustrates the information of these datasets, such as name, size, data type, and so on. In our experiments, we assume there are three parities for cross-feature FL, besides only party A holds label. For a fair comparison, we divide features into two parts randomly for every dataset, and assign each part to party A and B. We also randomly choose 70% samples for training, and the rest for testing.

All the experiments are performed on Ubuntu 16 operating system, 3.4 GHz Intel Core CPU, 150GB memory, Tesla P40 GPU with 24GB memory. We use iBond platform [9] to realize the cross-feature FL models with our proposed SFFS. The codes will be available soon.

4.2 Experimental Results And Analyses

As shown in Table 2, CF-LR, CF-XGB, CF-NN are executed in different datasets. The experiment compared the results of using SFFS and not using SFFS. We use the average accuracy of multiple rounds as the evaluation metric. Aiming to fully demonstrate the performance of our proposed algorithm, we give the mean ± standard deviation accuracy rates for each tested algorithm. It can be observed from Table 2 that SFFS significantly improves the performance of every cross-feature FL model on all the 9 tested datasets. For example, the average accuracy of CF-LR with SFFS is 0.4% ~5% higher than without SFFS. This reveals the importance of considering feature selection for cross-feature FL. With the increment of parties, the number of global features will increase explosively, reasonable and safe feature selection methods can effectively improve the stability of cross-feature FL training. For example, SMK_CAN_187's feature dimension is as high as 19993, but SFFS still yields 2~7% improvement. The results also indicate the robustness of SFFS. Although CF-XGB and CF-NN have strong ability to select features, it still brings these models great gain with

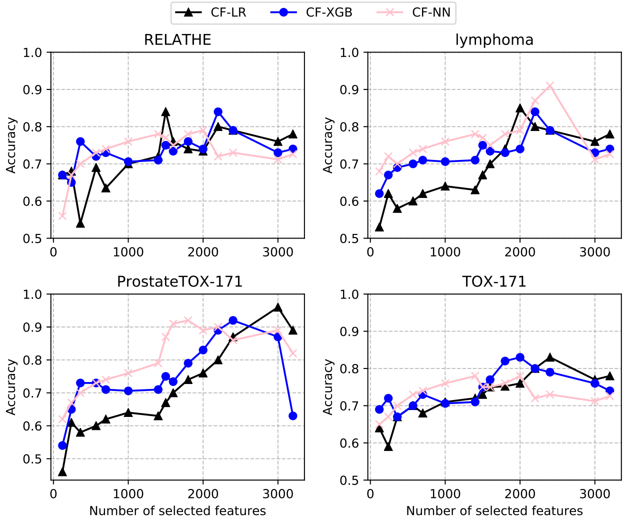

Figure 1: The influence of the number of the selected features for different algorithms and different datasets. CF-LR, CF-XGB, CF-NN with SFFS are applied to four datasets including RELATHE, lymphoma, and Prostate.

Table 3: Proportion of SFFS training time and Increased Accuracy for CF-LR, CF-XGB, CF-NN in different datasets)

Datasets	CF-LR		CF-XGB		CF-NN	
	SFFS / train	Increased Accuracy	SFFS / train	Increased Accuracy	SFFS / train(%)	Increased Accuracy
RELATHE	2%	**3%**	3%	**4%**	2%	**2%**
arcene	2.5%	**4%**	4%	**4.2%**	3.6%	**4.6%**
lymphoma	3.4%	**3%**	4.6%	**5.3%**	3.2%	**4.8%**
GLI-85	2.6%	**4.2%**	3.4%	**3.2%**	1.9%	**3%**
SMK_CAN_187	2.8%	**3.8%**	2.6%	**4.5%**	3.6%	**3.5 %**
Prostate-GE	2.8%	**3.5%**	4.5%	**3.6%**	2.7%	**4.1%**
ALLAML	2.3%	**3.8%**	2.6%	**3.1%**	1.6%	**3.8%**
CLL-SUB-111	1.2%	**3.4%**	2.4%	**3.2%**	3.8%	**4.6%**
TOX-171	3.8%	**4.7%**	2.9%	**4.8%**	3.2%	**4.5%**

the help of SFFS. For SMK_CAN_187 dataset, the accuracy of CF-XGB with SFFS is 7% higher than without SFFS. For GLI-85, CF-NN with SFFS obtains $87.89\% \pm 1.05\%$ while CF-NN without SFFS only reaches $83.05\% \pm 0.86\%$.

We also analyzed the influence of the number of selected features for different FL models on various datasets. Specificly, CF-LR, CF-XGB, and CF-NN with SFFS are applyed to four datasets, including RELATHE, lymphoma, Prostate, TOX-171. The total number of features in these datasets varies from 4000 and 6000. The results are shown in Fig. 1, indicating that with the increase of feature number, the accuracy varies a lot. These observations indicate a problem that is often overlooked in cross-feature FL. It is not the more features FL used, the better performance FL will be. Instead, some invalid features will turn into feature noise, which will cause a decrease in the accuracy. From this prospect, it is necessary to use SFFS before training a classification model.

Finally, we consider another core problem for cross-feature FL, namely efficiency and effectiveness. Will SFFS ensure data privacy at the expense of training time? As shown in Table 3, the time spent using SFFS accounts for only 2%~4% of the total training time, but the accuracy is improved by 2%-4%. This not only shows the efficiency of SFFS, but also shows that SFFS can achieved a balance between accuracy and time consumption.

5 Conclusions

In this paper, we propose a Secure Federated Feature Selection Method (SFFS) for cross-feature FL. SFFS is the first study that considers global features' efficiency and effectiveness for cross-feature FL, without compromising data privacy. Experimental results show solid evidence that SFFS a general privacy-preserving federated feature selection method that is not restricted to specific models, which can improve the robustness of cross-feature FL.

References

[1] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," 2016.

[2] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[3] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems*, 2017, pp. 5330–5340.

[4] P. P. Liang, T. Liu, L. Ziyin, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," *arXiv preprint arXiv:2001.01523*, 2020.

[5] Y. Liu, Y. Kang, X. Zhang, L. Li, Y. Cheng, T. Chen, M. Hong, and Q. Yang, "A communication efficient vertical federated learning framework," *arXiv preprint arXiv:1912.11187*, 2019.

[6] T. Chen, X. Jin, Y. Sun, and W. Yin, "Vafl: a method of vertical asynchronous federated learning," *arXiv preprint arXiv:2007.06081*, 2020.

[7] Y. Hu, D. Niu, J. Yang, and S. Zhou, "Fdml: A collaborative machine learning framework for distributed features," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2232–2240.

[8] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[9] H. Li, D. Meng, and X. Li, "Knowledge federation: Hierarchy and unification," in *IEEE international conference on knowledge graph*, 2020.

[10] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intelligent Systems*, 2020.

[11] J. Vaidya and C. Clifton, "Privacy-preserving decision trees over vertically partitioned data," in *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 2005, pp. 139–152.

[12] J. Vaidya, B. Shafiq, W. Fan, D. Mehmood, and D. Lorenzi, "A random decision tree framework for privacy-preserving data mining," *IEEE transactions on dependable and secure computing*, vol. 11, no. 5, pp. 399–411, 2013.

[13] W. Du, Y. S. Han, and S. Chen, "Privacy-preserving multivariate statistical analysis: Linear regression and classification," in *Proceedings of the 2004 SIAM international conference on data mining*. SIAM, 2004, pp. 222–233.

[14] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 19–38.

[15] N. Kilbertus, A. Gascón, M. J. Kusner, M. Veale, K. P. Gummadi, and A. Weller, "Blind justice: Fairness with encrypted sensitive attributes," *arXiv preprint arXiv:1806.03281*, 2018.

[16] D. Bogdanov, S. Laur, and J. Willemson, "Sharemind: A framework for fast privacy-preserving computations," in *European Symposium on Research in Computer Security*. Springer, 2008, pp. 192–206.

[17] P. Mohassel and P. Rindal, "Aby3: A mixed protocol framework for machine learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 35–52.

[18] J. Furukawa, Y. Lindell, A. Nof, and O. Weinstein, "High-throughput secure three-party computation for malicious adversaries and an honest majority," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2017, pp. 225–255.

[19] P. Mohassel, M. Rosulek, and Y. Zhang, "Fast and secure three-party computation: The garbled circuit approach," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 591–602.

[20] H. Zhou, Y. Zhang, Y. Zhang, and H. Liu, "Feature selection based on conditional mutual information: minimum conditional relevance and minimum conditional redundancy," *Applied Intelligence*, vol. 49, no. 3, pp. 883–896, 2019.

[21] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[22] S. Maldonado, R. Weber, and F. Famili, “Feature selection for high-dimensional class-imbalanced data sets using support vector machines,” *Information sciences*, vol. 286, pp. 228–246, 2014.

[23] J. A. Baranauskas, O. P. Netto, S. R. Nozawa, and A. A. Macedo, “A tree-based algorithm for attribute selection,” *Applied Intelligence*, vol. 48, no. 4, pp. 821–833, 2018.

[24] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[25] L. Wang, J. Zhu, and H. Zou, “Hybrid huberized support vector machines for microarray classification and gene selection,” *Bioinformatics*, vol. 24, no. 3, pp. 412–419, 2008.

[26] J. Che, Y. Yang, L. Li, X. Bai, S. Zhang, and C. Deng, “Maximum relevance minimum common redundancy feature selection for nonlinear data,” *Information Sciences*, vol. 409, pp. 68–86, 2017.

[27] F. Li, Z. Zhang, and C. Jin, “Feature selection with partition differentiation entropy for large-scale data sets,” *Information Sciences*, vol. 329, pp. 690–700, 2016.

[28] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt, “Feature selection via dependence maximization,” *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1393–1434, 2012.

[29] J. R. Vergara and P. A. Estévez, “A review of feature selection methods based on mutual information,” *Neural computing and applications*, vol. 24, no. 1, pp. 175–186, 2014.

[30] D. Causeur, C.-F. Sheu, E. Perthame, and F. Rufini, “A functional generalized f-test for signal detection with applications to event-related potentials significance analysis,” *Biometrics*, vol. 76, no. 1, pp. 246–256, 2020.

[31] J. F. Harper, “Peritz’f test: basic program of a robust multiple comparison test for statistical analysis of all differences among group means,” *Computers in biology and medicine*, vol. 14, no. 4, pp. 437–445, 1984.

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

[33] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, and Q. Yang, “Secureboost: A lossless federated learning framework,” *arXiv preprint arXiv:1901.08755*, 2019.

[34] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, “Feature selection: A data perspective,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.