SCIENCE CHINA
Information Sciences

• Supplementary File •

Supplementary material for "Federated local causal structure learning"

Kui YU[1†*], Chen RONG[1†], Hao WANG[1], Fuyuan CAO[2] & Jiye LIANG[2]

[1]*School of Computer and Information, Hefei University of Technology, Hefei* 230601*, China;*
[2]*School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China*

Appendix A Privacy protections for federated learning

The privacy protection of data, as accomplished by the FedLCS algorithm, is dependent on the federated learning framework. Within the federated framework, there is no need to share or access the raw data of each client. Instead, only the intermediate parameters including local skeletons, the separation sets, and the p-values are transmitted to the server for computation. Subsequently, the outcomes will be sent to each clients, completing the information sharing among clients. In addition, identifying v-structures and using Meek rule for edge orientations is only implemented at the server without touching clients' data, further reducing the risk of private information exposure.

Appendix B Experiments of consistent separation sets

A correct separation set is key to identifying v-structures. To validate the learning strategy of consistent separation set in the FLSori subroutine, we produce the FedLCS-InterSec algorithm. The difference between FedLCS-InterSec and FedLCS lies in that in the FLSori subroutine, we replace the consistent sets with the highest p-values with the intersection of the separation sets obtained from the FLSori subroutine. For instance, for the undirected triple $X_i - T - X_j$, during the subroutine of federated extension of local causal skeleton, each client will obtain the following separation sets of X_i and X_j, $sepset(X_i, X_j)_1, sepset(X_i, X_j)_2, \ldots, sepset(X_i, X_j)_N$. We consider $sepset(X_i, X_j) = \bigcap_{i=1}^{N} sepset(X_i, X_j)_i$ as the final separation set of X_i and X_j. Then we compare FedLCS-InterSec and FedLCS, and with the experimental results shown in Figure B1, B2, and B3, FedLCS is superior to FedLCS-InterSec in all experimental datasets.

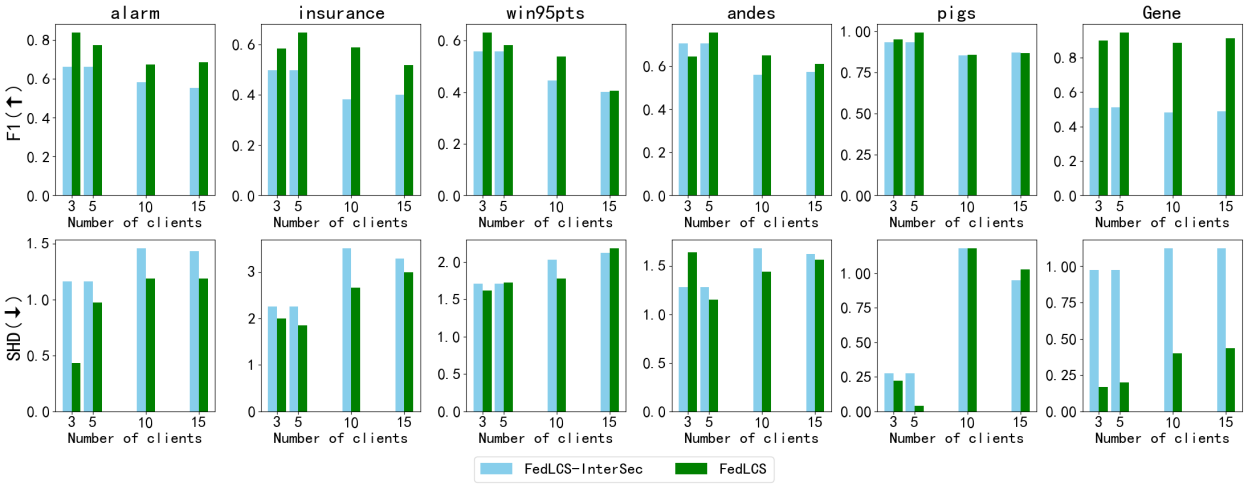

Figure B1 Comparison of FedLCS and FedLCS-InterSec on benchmark datasets.

Appendix C Experiments of parameter analysis

In this section, we conduct the following experiments to analyze the impacts of these parameters on FedLCS.

* Corresponding author (email: yukui@hfut.edu.cn)
† Kui YU and Chen RONG contributed equally to this work.

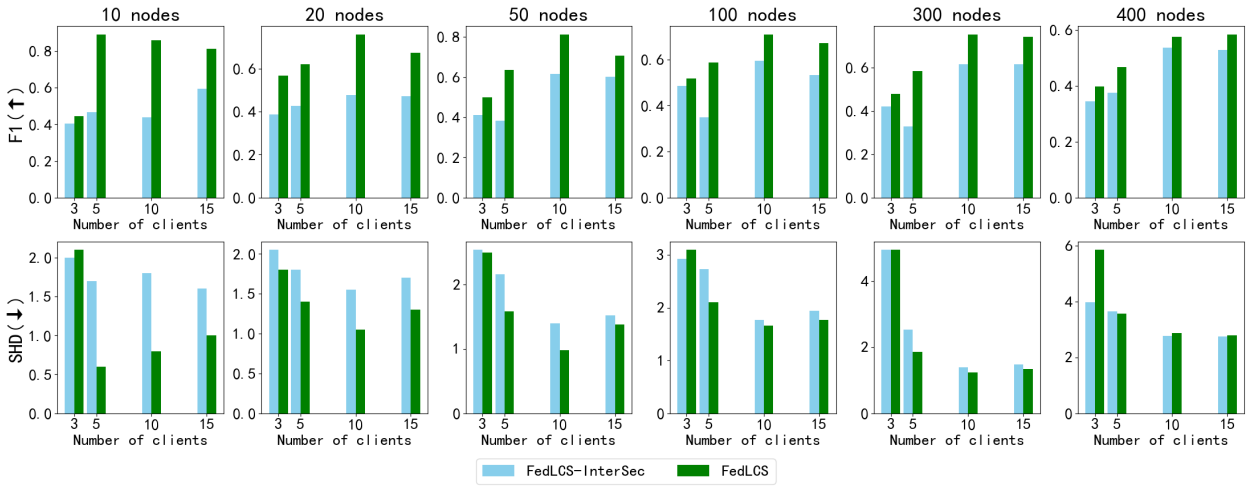

Figure B2 Comparison of FedLCS and FedLCS-InterSec on synthetic datasets.

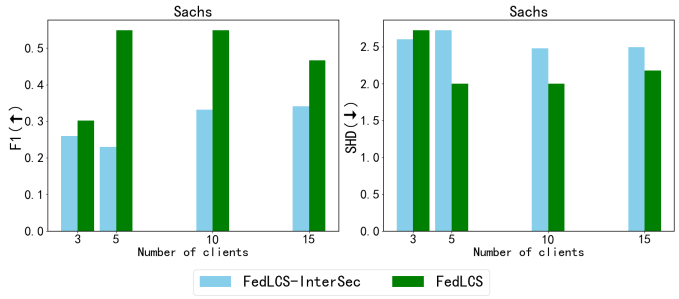

Figure B3 Comparison of FedLCS and FedLCS-InterSec on the Sachs dataset.

Appendix C.1 The layer-wise strategy parameter

The layer-wise strategy is used in the FLSke subroutine, in this section we conduct the following experiment to explore the impact of number of the layers on the FedLCS.

First, for a dataset, we run the FedLCS algorithm for each variable and store the values of layers when the learning procedure converges. We record the number of variables with the same values of layers. Then we display the percentage of the number of variables under each layer value in Figure C1. The results show that in the benchmark datasets, most variables can fully converge when the value of layer is up to 6, while in the synthetic datasets, the convergence is achieved when the value of layer is up to 8. Considering the time efficiency of FedLCS, we choose 6 as the layer parameter in the FLSke subroutine.

Appendix C.2 The ratio parameter

In the FLSke subroutine, in Eq.(2), an appropriate ratio value needs to be selected for determining whether an edge is kept or not during the process of skeleton aggregation stage. We conduct experiments within the ratio varying from 0.2 to 0.9 and observe the F1 and SHD values using different datasets. The results are shown as Figure C2. The F1 value and SHD value are normalized using the min-max normalization, scaling them to the interval of $(0, 1)$ for easy understanding.

Figure C2 indicates that in general the F1 value first increases and then decreases with the increasing ratio value, while the SHD value first decreases and then increases. We can see that when the ratio value reaches 0.4, FedLCS achieves the maximum F1 value and the minimum SHD value on the majority of datasets. This suggests that it is appropriate to set the ratio value 0.4 for achieving good performance of FedLCS.

Appendix C.3 The extension depth parameter

In the extension-and-backtracking orientation phase, we need to give a depth threshold to limit the extension depth of the learned local skeleton. We set the default value of the extension depth threshold to 2. The Figure C3 shows the impact of different extension depth thresholds on the performance of FedLCS.

In Figure C3 , when the threshold increases from 1 to 2, there is a significant performance improvement of FedLCS on both F1 and SHD metrics. However, the performance improvement becomes small when the threshold increases from 2 to 3. when the threshold is above 3, the performance of FedLCS will not change any more. The running time significantly increases with the threshold increasing on the datasets with more than 100 nodes. Based on these empirical results, we choose 2 as the extension depth threshold in this paper.

Appendix D Results on real dataset

We compare FedLCS with its rivals on the real Sachs dataset, with the results shown in Figure D1. It is evident that FedLCS outperforms its rivals on the real dataset. And when the number of clients is 5 or 10, the performance of FedLCS is significantly

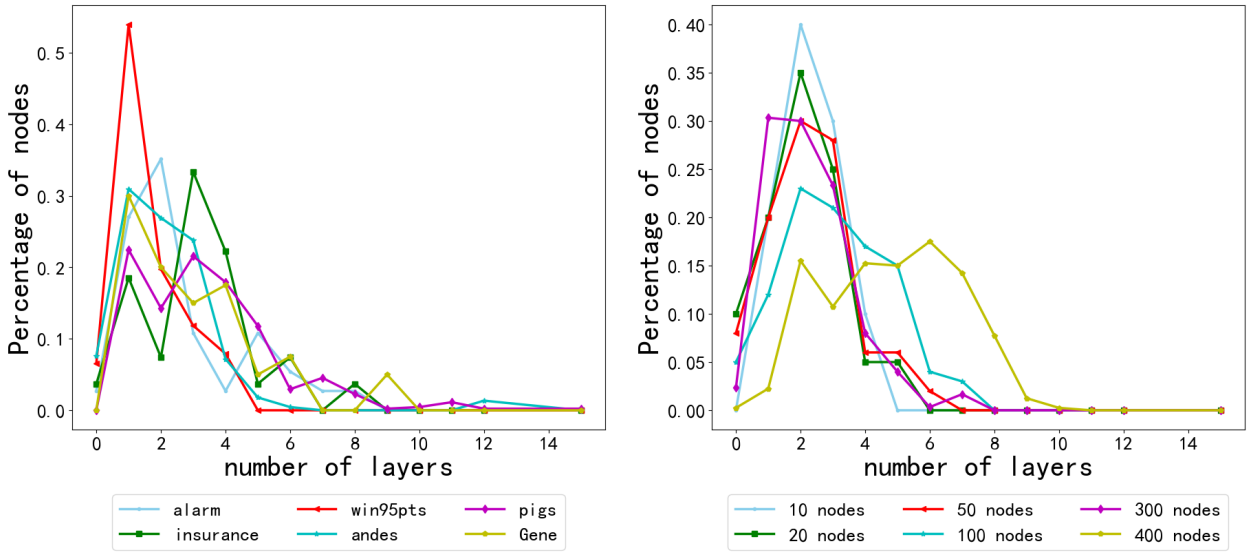

Figure C1 Value of layers and the convergence of FedLCS.

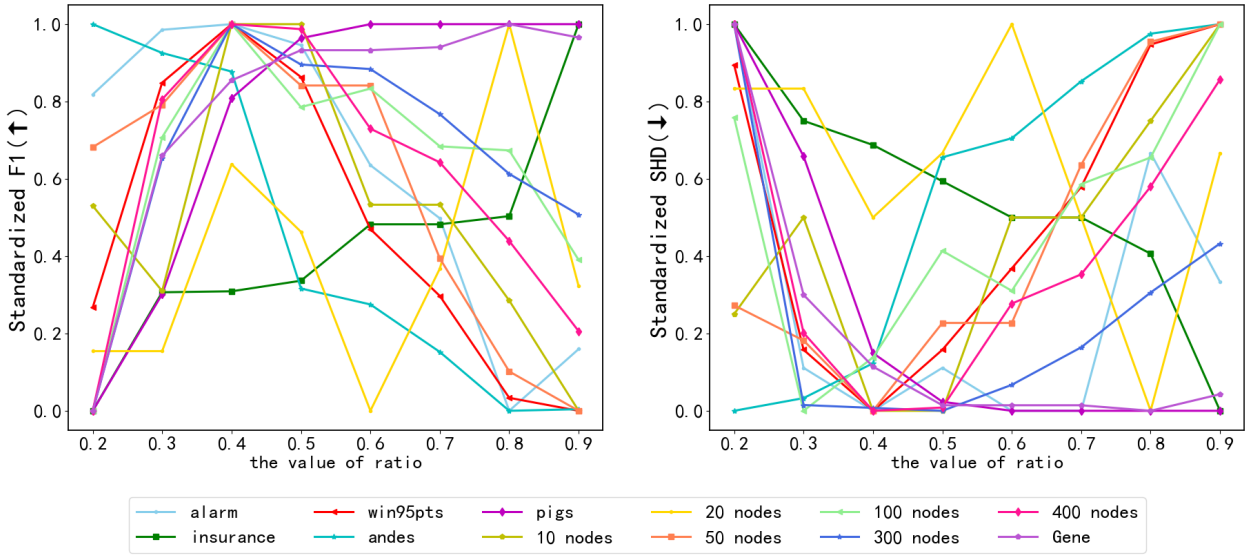

Figure C2 Impact of the ratio parameter

better than that of the other algorithms.

Appendix E Complexity analysis

For constraint-based causal structure learning algorithms, the execution time of CI test is usually the main consumption of the overall running time of the algorithm. Therefore, the mainstream method of measuring the time complexity of such algorithms is through the number of CI tests. Next, we will analyze the time complexity of the FedLCS algorithm.

Firstly, on the client side, each client independently learns the local skeleton, with a time complexity of $O(|PC| \cdot 2^{|PC|-1})$ (where $|PC|$ represents the maximum size of the PC for all nodes). Given that there are N clients, the time complexity for learning skeletons on the client side is $O(N \cdot |PC| \cdot 2^{|PC|-1})$.

On the server side, the time cost is distributed across two processes: aggregation through voting and the identification of v-structures. In the former process, it is necessary to traverse the adjacency matrices of N skeletons to calculate the EF scores. This process does not involve CI tests, and therefore, its computational cost is comparatively negligible. In the latter process, the goal is to find all possible triples formed by the PC sets of the target variables, and then perform CI tests on these triples individually. We assume that the number of triples to be identified is S, and the average degree (i.e., average PC set length) in Table 4 is rounded up, so the maximum average PC set length is 4, then the maximum number of the possible triples is $2 * C_4^2 = 12$ (i.e., $S \leqslant 12$). Therefore, the time complexity of one communication from client to server is $O(N \cdot |PC| \cdot 2^{|PC|-1} + S)$.

Finally, in the FLEori subroutine, according to the experimental parameters, the target variables undergo a two-layer outward skeleton expansion learning, which means learning an undirected path of length 2 connecting to the target variable. Thus, the overall time complexity of the FedLCS algorithm should be $O(|PC|^2 \cdot (N \cdot 2^{|PC|-1} + S)) \sim O(|PC|^3 \cdot N \cdot 2^{|PC|})$. Compared to

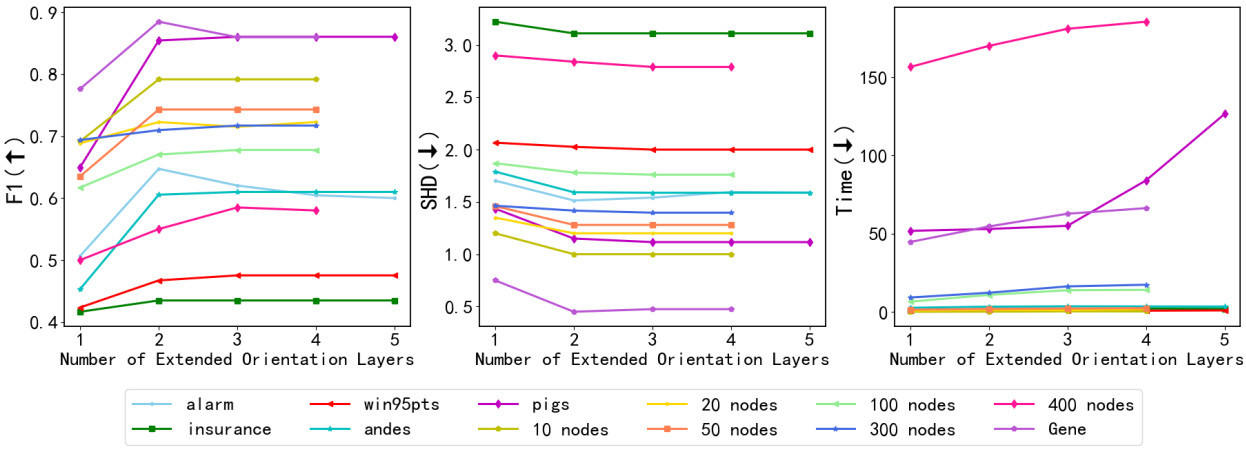

Figure C3 Impact of the extension depth parameter

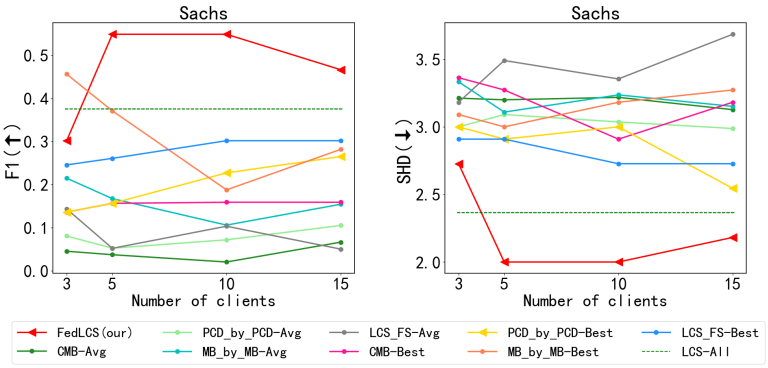

Figure D1 Results on real datasets.

algorithms like PCD-by-PCD and CMB, the FedLCS algorithm introduces an additional factor of N due to the process of concurrent learning across N clients.