8th November 2023

Jie Tang, PhD

Editor in Chief, IEEE Transactions on Big Data

Email: jietang@tsinghua.edu.cn

**Subject: Detailed Response to Reviewers for TBD-2023-06-0274.R1**

Dear Prof. Tang,

Thank you very much for organizing the review and giving us the opportunity to revise our article “Multi-party Federated Recommendation Based on Semi-supervised Learning”. The review comments have been fully taken into account and we have put full effort to address the comments raised by the reviewers in this revision.

We hope that our revision will satisfy the requirements of the reviewers, and make the article acceptable for publication in the IEEE Transactions on Big Data journal.

Below please find our detailed response organized by the required structure.

Sincerely Yours,

Xin Liu, Jiuluan Lv, Feng Chen, Qingjie Wei, Hangxuan He, and Ying Qian

Chongqing University of Posts and Telecommunications

Editor:

1. a) The entire comments made by the Associate Editor

During the second round of review, a reviewer proposed a critical question regarding the tradeoff between computation efficiency and the performance. please seriously answer the following question in the response letter which will be reviewed by AE, otherwise it may be rejected.

"The computational time expenditure is excessively substantial. I express my utmost gratitude to the author for their supplementary experiments pertaining to computational complexity (Table 3). Despite the significant enhancement in performance, the tenfold increase in computational efficiency evidently fails to satisfy the pursuit of a superior model. Hence, I opine that the author should optimize the methodology and resubmit it once more."

1. b) Response to Associate Editor

We have carefully reviewed our method and found that the issue did not lie in the method itself but rather in our implementation of the bagging process. The original implementation did not utilize parallel execution, which led to a substantial increase in computational time. This has been detailed in the Appendix. To address this, we have refactored our training code and have successfully reduced the computational time of our method by several times. Now, the computational time of our method is lower than that of VF\_2StepGBDT and is approximately 2-3 times higher than that of VF\_GBDT and VF\_Bagging\_GBDT.

We have revised the method section 3.2.2（2）to clarify that our method uses parallel execution. The modification involves adding the following paragraph to page 5 of the manuscript:

‘**The inner for loop, from line 5 to line 11 of Algorithm 1, is actually a bagging process. The bagging process contains steps for sampling, training, and predicting, repetitively for iterations. In each iteration, these steps are independent from those in other iterations.** **On this basis, we adopt parallel processing to execute the bagging procedure, therefore minimizing the overall time consumption of the algorithm.**’

We have revised the RQ3 section 4.2.3. We re-conducted experiments on Table 3 and recorded the runtime using parallel execution. In the segment comparing the runtimes of different methods, we analyzed the reasons why the runtime of our method is lower than that of VF\_2Step\_GBDT and why it is higher than VF\_GBDT and VF\_Bagging\_GBDT. The modifications are as follows:

**‘On the Credit dataset, our method had a runtime of 30086.33s, while the runtimes of VF\_GBDT, VF\_Bagging\_GBDT, and VF\_2Step\_GBDT were 12025.47s, 15791.59s, and 46954.19s respectively. Owing to the bootstrap technique, VFPU\_GBDT uses a smaller balanced dataset in each round of training, so the runtime of VFPU\_GBDT is lower than VF\_2Step\_GBDT for the same number of rounds. VFPU\_GBDT is approximately two to three times more time-consuming than VF\_GBDT and VF\_Bagging\_GBDT in terms of the metric of runtime(s). This is because VFPU\_GBDT adopts a more cautious strategy in selecting reliable positive samples through multiple iterations and selecting only a small portion in each iteration. So, there is a trade-off between the training time and the accuracy of the recommendation, and this time overhead is absolutely acceptable since the accuracy is significantly improved. When executing the bagging process of Algorithm 1, we optimized it by using parallel processing to reduce the time consumption. For details on the serial implementation and parallel implementation, as well as the comparison of their runtimes, please refer to the Appendix.’**

Reviewer #1:

1. a) The entire comments made by the Reviewer 1

Summary:

The problem solved by this work is promising, but the computational cost of ten times beyond the baseline models still does not meet the requirements.

Strength:

1. Identification of UDD-PU Learning Problem: The manuscript successfully identifies and addresses the Unlabeled-Data-Deficient PU (UDD-PU) learning problem, which is a novel challenge in the context of vertical federated learning. By recognizing this problem, the manuscript fills a gap in the existing literature and proposes a solution to tackle it.

2. Proposal of VFPU Algorithm: The manuscript introduces the VFPU algorithm, which enables multiple parties to collaboratively train a machine learning model in a privacy-preserving manner. The algorithm involves random sampling, training with balanced positive and negative samples, and selecting reliable positive samples iteratively. The proposed algorithm is innovative and demonstrates comparable performance to non-federated methods, outperforming other federated semi-supervised learning approaches.

3. Experimental Results and Performance Evaluation: The manuscript presents experimental results that validate the effectiveness of the VFPU algorithm. The comparative analysis shows that VFPU performs well in terms of recommendation accuracy and outperforms existing federated semi-supervised learning methods. These results provide empirical evidence of the algorithm's strength and contribute to the understanding of its performance in real-world scenarios.

Weakness:

What are the three weakest aspects of this manuscript?: The computational time expenditure is excessively substantial. I express my utmost gratitude to the author for their supplementary experiments pertaining to computational complexity (Table 3). Despite the significant enhancement in performance, the tenfold increase in computational efficiency evidently fails to satisfy the pursuit of a superior model. Hence, I opine that the author should optimize the methodology and resubmit it once more.

1. b) Response to Reviewer 1

Thank you for your feedback and suggestions, especially regarding the tradeoff between computational efficiency and performance. We have taken this seriously and have optimized the code implementation of our method accordingly.

Upon careful review of our method, we have found that the issue did not lie in the method itself but rather in our implementation of the bagging process. The original implementation did not utilize parallel execution, which led to a substantial increase in computational time. This has been detailed in the appendix.

To address this, we have restructured our training code and have successfully reduced the computational time of our method by several times. Now, the computational time of our method is lower than that of VF\_2StepGBDT, and is approximately 2-3 times higher than that of VF\_GBDT and VF\_Bagging\_GBDT.

We have revised the method section 3.2.2（2）to clarify that our method uses parallel execution. The modification involves adding the following paragraph to page 5 of the manuscript:

‘**The inner for loop, from line 5 to line 11 of Algorithm 1, is actually a bagging process. The bagging process contains steps for sampling, training, and predicting, repetitively for iterations. In each iteration, these steps are independent from those in other iterations.** **On this basis, we adopt parallel processing to execute the bagging procedure, therefore minimizing the overall time consumption of the algorithm.**’

We have revised the RQ3 section 4.2.3. We re-conducted experiments on Table 3 and recorded the runtime using parallel execution. In the segment comparing the runtimes of different methods, we analyzed the reasons why the runtime of our method is lower than that of VF\_2Step\_GBDT and why it is higher than VF\_GBDT and VF\_Bagging\_GBDT. The modifications are as follows:

**‘On the Credit dataset, our method had a runtime of 30086.33s, while the runtimes of VF\_GBDT, VF\_Bagging\_GBDT, and VF\_2Step\_GBDT were 12025.47s, 15791.59s, and 46954.19s respectively. Owing to the bootstrap technique, VFPU\_GBDT uses a smaller balanced dataset in each round of training, so the runtime of VFPU\_GBDT is lower than VF\_2Step\_GBDT for the same number of rounds. VFPU\_GBDT is approximately two to three times more time-consuming than VF\_GBDT and VF\_Bagging\_GBDT in terms of the metric of runtime(s). This is because VFPU\_GBDT adopts a more cautious strategy in selecting reliable positive samples through multiple iterations and selecting only a small portion in each iteration. So, there is a trade-off between the training time and the accuracy of the recommendation, and this time overhead is absolutely acceptable since the accuracy is significantly improved. When executing the bagging process of Algorithm 1, we optimized it by using parallel processing to reduce the time consumption. For details on the serial implementation and parallel implementation, as well as the comparison of their runtimes, please refer to the Appendix.’**

Reviewer #2:

No more comments.