This document provides clear instructions and a step-by-step guide to reproduce the experimental results reported in the ICSOC2025 paper: Privacy-Preserving Revenue Prediction in Service-Oriented Industrial Supply Chains. All related code, datasets, and documentation are hosted on GitHub: https://github.com/windflowerHu/DPC-GFRBM/

1. Repository Structure

The repository contains the following key components:

1.1 Main Entry Functions

- **test_ASOC.m**: Baseline method CFC-GFRBM
- **test_ASOC_NOCOLLAB.m**: Baseline method ISO-GFRBM
- **test_DBSCAN.m**: Baseline methods DBSCAN-GFRBM, DF-GFRBM, and FCM-GFRBM
- **test_R1.m**: Proposed method DPC-GFRBM
- **test_real_R1**.m: Proposed method DPC-GFRBM on the fortune1000_2024 dataset

1.2 Utility Functions

- **data_normalize.m / dataNormalization.m**: Dataset preprocessing and initialization
- **split_data.m**: Partition dataset into local alliances
- **FCM.m**: Standard fuzzy C-means clustering function

1.3 Baseline Model Functions

- **gran_model_train.m**: Train prototypes for CFC-GFRBM
- **nocollab_model_train.m**: Train prototypes for ISO-GFRBM
- **gran_model_test.m**: Compute prediction results for CFC-GFRBM and ISO-GFRBM
- **compute_PGJ_onedim.m**: Compute granular prototypes for CFC-GFRBM and ISO-GFRBM
- **dbscan_model1.m**: Granular outputs for DBSCAN-GFRBM
- **dbscan_model2.m**: Granular outputs for DF-GFRBM
- **dbscan_model3.m**: Granular outputs for FCM-GFRBM
- **dbscan_test.m**: Membership matrix for DBSCAN-GFRBM, DF-GFRBM, FCM-GFRBM

1.4 Proposed Model Functions (DPC-GFRBM)

- **step1_local_dpc.m**: Local clustering using DPC within alliances

- **step1_local_fcm.m**: Local clustering using FCM within alliances
- **step2_local_gran.m**: Granularization after local FCM clustering
- **step2_local_gran_dpc.m**: Granularization after local DPC clustering
- **step3_global_fcm.m**: Aggregate local granular prototypes to build global prototypes
- **step4_new_gran_y.m**: Establish mapping between global membership prototypes and interval outputs
- **step5_gran_test.m**: Compute membership matrix for the testing dataset

1.5 Evaluation Functions

- **index_gran.m**: Granular prediction metric (V)

---

2. Datasets

- **Training sets**: ./data/UCI/train/
  Each UCI dataset is stored as a .txt file.
- **Testing sets**: ./data/UCI/test/
  Corresponding test data for UCI datasets, in .txt format.
- **Real industrial dataset**: ./data/real_income/fortune1000_2024
  A real dataset containing enterprise income records.

Note: Detailed dataset descriptions and download instructions are provided in Section 4.1 of the main paper.

---

3. Execution Instructions

Step 1: Configure Parameters

Open one of the entry files (test_ASOC.m, test_ASOC_NOCOLLAB.m, test_DBSCAN.m, test_R1.m or test_real_R1.m) and specify:

1. **Dataset name** (e.g., Airfoil Self Noise, fortune1000_2024, etc.)
2. **Number of alliances (P)**
3. **Number of clusters (n_cluster)**

Step 2: Run the Algorithm

Execute one of the entry files in MATLAB:

```
run('test_R1.m'); % Example: run proposed DPC-GFRBM
```

This will automatically:

- Load and preprocess the dataset
- Partition data across alliances
- Train prototypes using the selected method
- Evaluate test set performance

Step 3: Collect Results

The script will output the evaluation metric **V (Granular Prediction Metric).**