Supplementary Algorithm S1 **Grid search hyperparameter tuning**

|  |
| --- |
| **Algorithm 1 .** Grid search hyperparameter tuning |
|  **Start** |
|  | **Input:**Model: a machine learning model (XGBoost classifier in this study) to be trained and testedHyperparameters: a dictionary of hyperparameter ranges to search throughX\_train, y\_train: training data and labelsX\_test, y\_test: testing data and labels |
|  | **Output:**Best hyperparameters: a dictionary containing the best hyperparameters found during the grid searchBest score: the best score achieved by the model during the grid search |
| **1** | Define the hyperparameters to be searched over**hyperparameters\_to\_search** = Hyperparameters.keys () |
| **2** | Generate all possible combinations of hyperparameters**parameter\_combinations** = cartesian\_product (\*Hyperparameters.values ()) |
| **3** | Train and test the model on each combination of hyperparameters**best\_score** = None**best\_hyperparameters** = None |
| **4** | **for** parameters in parameter\_combinations: |
| **5** |  |  Train the model on the training data using the current hyperparameters model.set\_params(\*\*parameters) model.fit (X\_train, y\_train) |
| **6** |  |  Evaluate the model on the testing data using the current hyperparameters **score** = model.score (X\_test, y\_test) |
| **7** |  |  Update the best hyperparameters and score if the current score is better |
| **8** |  |  **if** best\_score is None or score > best\_score: |
| **9** |  |  |  **best\_score** = score **best\_hyperparameters** = parameters |
| **10** |  |  **End if** |
| **11** | **End for** |
| **12** | Return the best hyperparameters and scorereturn **best\_hyperparameters**, **best\_score**Retrain the classifier on the entire dataset using the optimal combination of hyperparameters |
| **13** | **End**  |