1. **Bayesian optimization algorithm (BOA) implementation code:**
2. Define the optimization function (including K-fold cross-validation):

def objective(trial):

cv = \*

X = excel()

wb = xlrd2.open\_workbook('\*\*\*\*\*\*\*')

sheet = wb.sheet\_by\_name('\*\*\*\*\*')

y = sheet.col\_values(\*\*)

X\_train, X\_test, y\_train, y\_test = sklearn.model\_selection.train\_test\_split(X, y, test\_siz =0.2, shuffle=True, random\_state=2)

sfolder = KFold(n\_splits=cv, shuffle=True, random\_state=7)

XGBDT\_n\_estimators = trial.suggest\_int('XGBDT\_n\_estimators', 1, 500, log=True)

XGBDT\_learning\_rate = trial.suggest\_float('XGBDT\_learning\_rate', 0.001, 0.2)

XGBDT\_max\_depth = trial.suggest\_int('XGBDT\_max\_depth', 1, 50, log=True)

regressor\_obj = XGBRegressor(n\_estimators=XGBDT\_n\_estimators, learning\_rate=XGBDT\_learning\_rate, max\_depth=XGBDT\_max\_depth)

R2 = 0

for train\_index, val\_index in sfolder.split(X\_train):

X\_train\_ture = []

y\_train\_ture = []

X\_val = []

y\_val = []

for idx in train\_index:

X\_train\_ture.append(X\_train[idx])

y\_train\_ture.append(y\_train[idx])

for idx in val\_index:

X\_val.append(X\_train[idx])

y\_val.append(y\_train[idx])

X\_train\_ture\_backup = copy.deepcopy(X\_train\_ture)

X\_val\_backup = copy.deepcopy(X\_val)

model = regressor\_obj.fit(X\_train\_ture\_backup, y\_train\_ture)

y\_val\_pred = model.predict(X\_val\_backup)

R2 += sklearn.metrics.r2\_score(y\_val, y\_val\_pred)

R2 /= cv

return R2

1. Optimization in the main program:

opt\_num = \*\*\*\*\*\*\*\*

study = optuna.create\_study(direction='maximize')

study.optimize(objective, n\_trials=opt\_num)

best\_parameters = study.best\_params

1. **Training model using the best combination of hyperparameters:**

model = XGBRegressor(n\_estimators=best\_parameters['XGBDT\_n\_estimators'], learning rate=best\_parameters['XGBDT\_learning\_rate'], max\_depth=best\_parameters['XGBDT\_max\_depth']) # Use this item when optimizing

model = XGBRegressor(n\_estimators=\*\*\*\*, learning\_rate=\*\*\*\*\*\*, max\_depth=\*\*\*\*) # Use this item when running the model

model.fit(X\_train\_backup, y\_train)

y\_pred = model.predict(X\_backup)

y\_train\_pred = model.predict(X\_train\_backup)

y\_test\_pred = model.predict(X\_test\_backup)

*Note: \* in the code indicates the file name or variable value*