Supplementary Table S1 The Best Combination of hyperparameter for Different Machine Learning Algorithms

Algorithms	Function	Parameter Range	Optimum Parameter
AdaBoost	AdaBoostClassifier	**learning_rate** = [0.1, 1, 0.01, 0.001], **n_estimators** = [50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180]	**learning_rate** = 0.1, **n_estimators** = 180
Bagging	BaggingClassifier	**n_estimators =** [10, 30, 50, 70, 80, 120, 130, 140, 145, 150, 160, 170, 175, 180, 185], **max_features =** range (1, 10)	**n_estimators = 150, max_features = 8**
Decision Tree	DecisionTreeClassifier	**criterion** = ['gini', 'entropy'], **max_depth** = range (1, 30), **max_features** = [21, 22, 23, 24, 25, 26, 28, 29, 30, 'auto']	**criterion** = 'entropy', **max_depth** = 14, **max_features** = 29
LDA	LinearDiscriminantAnalysis	**solver** = ['eigen', 'lsqr'], **shrinkage** = [0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 'auto']	**solver**='eigen, **shrinkage** = 0.03
Quadratic Discriminant	QuadraticDiscriminantAnalysis	**reg_param =** np.linspace(0, 1, 10), **tol =** [0.0001, 0.001, 0.01, 0.1, 1]	**reg_param =** 0.88, **tol =** 0.0001
Random Forest	RandomForestClassifier	**Criterion** = ['gini', 'entropy'], **max_depth** = range (1, 10), **n_estimators** = [50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180]	**criterion** = 'entropy', **max_depth** = 8, **n_estimators** = 150

SVM	SVC	**Cs =** [0.0001, 0.001, 0.01, 0.1, 1, 2, 3, 4, 5, 10], **gamma =** [0.0001, 0.001, 0.01, 0.1, 1], kernel = ['rbf', ' linear ']	**Cs =** 0.001, **gamma =** 0.001, **kernel =** 'linear'
XGBoost	XGBClassifier	**n_estimators =** [50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180], **learning_ rate =** [0.1, 1, 0.01, 0.001]	**estimators** = 100, **learning_rate** = 0.01