Table 1. Hyper parameters for Logistic Regression.

Parameter	Description	Range
penalty	The norm used in the penalization.	["l2", "l1"]
C	Inverse of regularization strength.	range(0.001, 10, 0.001)
class_weight	Weights associated with classes in the form {class_label: weight}.	[None, "balanced"]

* The function range(start, end, step=1) is used.

Table 2. Hyper parameters for K Nearest Neighbor.

Parameter	Description	Range
n_neighbors	Number of neighbors to use.	range(1, 51)
weights	Weight function used in prediction.	["uniform"]
metric	The distance metric to use for the tree.	["minkowski", "euclidean", "manhattan"]

Table 3. Hyper parameters for Naivers Bayes.

Parameter	Description	Range
var_smoothing	Portion of the largest variance of all features that is added to variances for calculation stability.	[0.000000001, 0.000000002, 0.000000005, 0.000000008, 0.000000009, 0.0000001, 0.0000002, 0.0000003, 0.0000005, 0.0000007, 0.0000009, 0.00001, 0.001, 0.002, 0.003, 0.004, 0.005, 0.007, 0.009, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.1, 1]

Table 4. Hyper parameters for Support Vector Machine (Linear).

Parameter	Description	Range
penalty	The penalty (aka regularization term) to be used.	["elasticnet", "l2", "l1"],
l1_ratio	The Elastic Net mixing parameter.	range(0.0000000001, 1, 0.01)
alpha	Constant that multiplies the regularization term.	[0.0000001, 0.000001, 0.0001, 0.001, 0.01, 0.0002, 0.002, 0.02, 0.0005, 0.005, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5]
fit_intercept	Whether the intercept should be estimated or not.	[True, False]
learning_rate	The learning rate schedule for training.	["constant", "invscaling", "adaptive", "optimal"]
eta0	The initial learning rate for the ‘constant’, ‘invscaling’ or ‘adaptive’ schedules.	[0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5]

Table 5. Hyper parameters for Support Vector Machine (rbf).

Parameter	Description	Range
C	Regularization parameter.	range(0, 50, 0.01)
class_weight	Weights associated with classes.	[None, "balanced"]

Table 6. Hyper parameters for Linear Discriminant Analysis.

Parameter	Description	Range
solver	Solver to use.	["lsqr", "eigen"]
shrinkage	Shrinkage parameter.	["empirical", "auto", 0.0001, 0.001, 0.01, 0.0005, 0.005, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]

Table 7. Hyper parameters for Quadratic Discriminant Analysis.

Parameter	Description	Range
reg_param	Regularization parameter for the per-class covariance estimates.	range(0, 1, 0.01)

Table 8. Hyper parameters for Ridge Classifier.

Parameter	Description	Range
normalize	Data normalization or not.	[True, False]
alpha	Regularization strength.	range(0.01, 10, 0.01)
fit_intercept	Whether to calculate the intercept.	[True, False]

Table 9. Hyper parameters for Decision Tree.

Parameter	Description	Range
max_depth	The maximum depth of the tree.	range(1, 16, 1)
max_features	The number of features to consider when looking for the best split.	[1.0, "sqrt", "log2"]
min_samples_leaf	The minimum number of samples required to be at a leaf node.	[2, 3, 4, 5, 6]
min_samples_split	The minimum number of samples required to split an internal node.	[2, 5, 7, 9, 10]
criterion	The function to measure the quality of a split.	["gini", "entropy"]
min_impurity_decrease	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.	[0, 0.0001, 0.001, 0.01, 0.0002, 0.002, 0.02, 0.0005, 0.005, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5]

Table 10. Hyper parameters for Random Forest.

Parameter	Description	Range
n_estimators	The number of trees in the forest.	range(10, 300, 10)
max_depth	The maximum depth of the tree.	range(1, 11, 1)

min_impurity_decrease	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.	[0, 0.0001, 0.001, 0.01, 0.0002, 0.002, 0.02, 0.0005, 0.005, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5]
max_features	The number of features to consider when looking for the best split.	[1.0, "sqrt", "log2"]
bootstrap	Whether bootstrap samples are used when building trees.	[True, False]
criterion	The function to measure the quality of a split.	["gini", "entropy"]
class_weight	Weights associated with classes.	[None, "balanced", "balanced_subsample"]
min_samples_split	The minimum number of samples required to split an internal node.	[2, 5, 7, 9, 10]
min_samples_leaf	The minimum number of samples required to be at a leaf node.	[2, 3, 4, 5, 6]

Table 11. Hyper parameters for Extra Trees.

Parameter	Description	Range
n_estimators	The number of trees in the forest.	range(10, 300, 10)
criterion	The function to measure the quality of a split.	["gini", "entropy"]
max_depth	The maximum depth of the tree.	range(1, 11, 1)
min_impurity_decrease	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.	[0, 0.0001, 0.001, 0.01, 0.0002, 0.002, 0.02, 0.0005, 0.005, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5]
max_features	The number of features to consider when looking for the best split.	[1.0, "sqrt", "log2"]
bootstrap	Whether bootstrap samples are used when building trees.	[True, False]
min_samples_split	The minimum number of samples required to split an internal node.	[2, 5, 7, 9, 10]
min_samples_leaf	The minimum number of samples required to be at a leaf node.	[2, 3, 4, 5, 6]
class_weight	Weights associated with classes.	[None, "balanced", "balanced_subsample"]

Table 12. Hyper parameters for AdaBoost.

Parameter	Description	Range
n_estimators	The maximum number of estimators.	range(10, 300, 10)
learning_rate	Weight applied to each classifier at each boosting iteration.	[0.0000001, 0.000001, 0.0001, 0.001, 0.01, 0.0005, 0.005, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5]

algorithm	Boosting algorithm to use.	["SAMME", "SAMME.R"]

Table 13. Hyper parameters for Gradient Boosting Classifier.

Parameter	Description	Range
n_estimators	The number of boosting stages to perform.	range(10, 300, 10)
learning_rate	Learning rate shrinks the contribution of each tree by learning_rate.	[0.0000001, 0.000001, 0.0001, 0.001, 0.01, 0.0005, 0.005, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5]
subsample	The fraction of samples to be used for fitting the individual base learners.	range(0.2, 1, 0.05)
min_samples_split	The minimum number of samples required to split an internal node.	[2, 4, 5, 7, 9, 10]
min_samples_leaf	The minimum number of samples required to be at a leaf node.	[1, 2, 3, 4, 5]
max_depth	The maximum depth of the individual regression estimators.	range(1, 11, 1)
min_impurity_decrease	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.	[0, 0.0001, 0.001, 0.01, 0.0002, 0.002, 0.02, 0.0005, 0.005, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5]
max_features	The number of features to consider when looking for the best split.	[1.0, "sqrt", "log2"]

Table 14. Hyper parameters for Extreme Gradient Boosting.

Parameter	Description	Range
n_estimators	The number of trees.	range(10, 300, 10)
learning_rate	Step size shrinkage used in update to prevents overfitting.	[0.0000001, 0.000001, 0.0001, 0.001, 0.01, 0.0005, 0.005, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5]
subsample	Subsample ratio of the training instances.	[0.2, 0.3, 0.5, 0.7, 0.9, 1]
max_depth	Maximum depth of a tree.	range(1, 11, 1)
colsample_bytree	Subsample ratio of columns when constructing each tree.	[0.5, 0.7, 0.9, 1]
min_child_weight	Minimum sum of instance weight (hessian) needed in a child.	[1, 2, 3, 4]
reg_alpha	L1 regularization term on weights.	[0.0000001, 0.000001, 0.0001, 0.001, 0.01, 0.0005, 0.005, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.7, 1, 2, 3, 4, 5, 10]
reg_lambda	L2 regularization term on weights.	[0.0000001, 0.000001, 0.0001, 0.001, 0.01, 0.0005, 0.005, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.7,

		1, 2, 3, 4, 5, 10]
scale_pos_weight	Control the balance of positive and negative weights, useful for unbalanced classes.	range(0, 50, 0.1)

Table 15. Hyper parameters for Light Gradient Boosting.

Parameter	Description	Range
num_leaves	max number of leaves in one tree.	[2, 4, 6, 8, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 256]
learning_rate	Shrinkage rate.	[0.0000001, 0.000001, 0.0001, 0.001, 0.01, 0.0005, 0.005, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5]
n_estimators	The number of trees.	range(10, 300, 10)
min_split_gain	The minimal gain to perform split.	[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
reg_alpha	L1 regularization term on weights.	[0.0000001, 0.000001, 0.0001, 0.001, 0.01, 0.0005, 0.005, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.7, 1, 2, 3, 4, 5, 10]
reg_lambda	L2 regularization term on weights.	[0.0000001, 0.000001, 0.0001, 0.001, 0.01, 0.0005, 0.005, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.7, 1, 2, 3, 4, 5, 10]
feature_fraction	The ratio of the subset of features	range(0.4, 1, 0.1)
bagging_fraction	The ratio of the data selected randomly without resampling.	range(0.4, 1, 0.1)
bagging_freq	Frequency for bagging.	[0, 1, 2, 3, 4, 5, 6, 7]
min_child_samples	Minimal number of data in one leaf.	range(1, 100, 5)

Table 16. Hyper parameters for CatBoost.

Parameter	Description	Range
eta	The learning rate.	[0.0000001, 0.000001, 0.0001, 0.001, 0.01, 0.0005, 0.005, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5]
depth	Depth of the tree.	range(1, 12)
n_estimators	The maximum number of trees.	range(10, 300, 10)
random_strength	The amount of randomness to use for scoring splits when the tree structure is selected.	range(0, 0.8, 0.1)
l2_leaf_reg	Coefficient at the L2 regularization term of the cost function.	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 50, 100, 200]

Table 17. Shared hyper parameters for deep learning models.

Parameter	Description	Range
batch_size	Batch size for training.	[256, 512, 1024]
time_steps	Time steps used as inputs.	[30, 60, 90]
lr	Learning rate of the optimizer.	[0.1, 0.01, 0.001, 0.0001]
epochs	Number of epochs for training.	[10, 50, 100]
patience	Patience of early stopping.	[5, 10]
optimizer	Optimizer for training.	[Adam, SGD, RMSProp]

Table 18. Model-related hyper parameters for MLP.

Parameter	Description	Range
layers	The number of layers and neurons in the hidden layers.	[100, 100, 100], [200, 200, 200], [500, 500, 500]]
fc_dropout	Dropout rate in the fully connected layers.	[0.1, 0.2, 0.5]
use_bn	Use batch normalization or not.	[True, False]

Table 19. Model-related hyper parameters for RNN, LSTM and GRU.

Parameter	Description	Range
n_layers	The number of RNN layers.	[1, 2, 3, 4, 5]
hidden_size	The number of neurons in the hidden layers.	[50, 100, 200]
bidirectional	Bidirectional RNN or not.	[True, False]

Table 20. Model-related hyper parameters for FCN.

Parameter	Description	Range
layers	The number of layers and neurons in the hidden layers.	[[64, 128, 64], [128, 256, 128], [256, 512, 256]]
kss	Kernel filter size for convolution.	[[11, 9, 7], [9, 7, 5], [7, 5, 3]]

Table 21. Model-related hyper parameters for TCN.

Parameter	Description	Range
layers	The number of layers and neurons in the hidden layers.	[[25, 25, 25, 25, 25, 25, 25, 25], [25, 25, 25, 25, 25, 25], [25, 25, 25, 25]]
ks	Kernel filter size for convolution.	[7, 5, 3]
conv_dropout	Dropout rate for convolution.	[0.0, 0.1, 0.2, 0.5]

Table 22. Model-related hyper parameters for InceptionTime.

Parameter	Description	Range
nf	The number of filters.	[16, 32, 64]

Table 23. Model-related hyper parameters for XceptionTime.

Parameter	Description	Range
nf	The number of filters.	[8, 16, 32]

Table 24. Model-related hyper parameters for XCM.

Parameter	Description	Range
nf	The number of filters.	[64, 128, 256]
fc_dropout	Dropout rate in the fully connected layers.	[0.0, 0.1, 0.2, 0.5]
bn	Use batch normalization or not.	[True, False]