Table 1. Exact hyper parameters used for Logistic Regression.

Parameter	Description	Value
penalty	The norm used in the penalization.	"l2"
C	Inverse of regularization strength.	4.212
class_weight	Weights associated with classes in the form {class_label: weight}.	None

Table 2. Exact hyper parameters used for K Nearest Neighbor.

Parameter	Description	Value
n_neighbors	Number of neighbors to use.	5
weights	Weight function used in prediction.	"uniform"
metric	The distance metric to use for the tree.	"euclidean"

Table 3. Exact hyper parameters used for Naivers Bayes.

Parameter	Description	Value
var_smoothing	Portion of the largest variance of all features that is added to variances for calculation stability.	2e-07

Table 4. Exact hyper parameters used for Support Vector Machine (Linear).

Parameter	Description	Value
penalty	The penalty (aka regularization term) to be used.	"elasticnet"
l1_ratio	The Elastic Net mixing parameter.	0.91
alpha	Constant that multiplies the regularization term.	0.01
fit_intercept	Whether the intercept should be estimated or not.	True
learning_rate	The learning rate schedule for training.	"adaptive"
eta0	The initial learning rate for the ‘constant’, ‘invscaling’ or ‘adaptive’ schedules.	0.2

Table 5. Exact hyper parameters used for Support Vector Machine (rbf).

Parameter	Description	Value
C	Regularization parameter.	1.0
class_weight	Weights associated with classes.	"balanced"

Table 6. Exact hyper parameters used for Linear Discriminant Analysis.

Parameter	Description	Value
solver	Solver to use.	"lsqr"
shrinkage	Shrinkage parameter.	0.0005

Table 7. Exact hyper parameters used for Quadratic Discriminant Analysis.

Parameter	Description	Value
reg_param	Regularization parameter for the per-class covariance estimates.	0.0

Table 8. Exact hyper parameters used for Ridge Classifier.

Parameter	Description	Value
normalize	Data normalization or not.	False
alpha	Regularization strength.	5.43
fit_intercept	Whether to calculate the intercept.	True

Table 9. Exact hyper parameters used for Decision Tree.

Parameter	Description	Value
max_depth	The maximum depth of the tree.	3
max_features	The number of features to consider when looking for the best split.	1.0
min_samples_leaf	The minimum number of samples required to be at a leaf node.	3
min_samples_split	The minimum number of samples required to split an internal node.	9
criterion	The function to measure the quality of a split.	"gini"
min_impurity_decrease	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.	0.0005

Table 10. Exact hyper parameters used for Random Forest.

Parameter	Description	Value
n_estimators	The number of trees in the forest.	230
max_depth	The maximum depth of the tree.	9
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
max_features	The number of features to consider when looking for the best split.	"sqrt"
bootstrap	Whether bootstrap samples are used when building trees.	True
criterion	The function to measure the quality of a split.	"entropy"
class_weight	Weights associated with classes.	None
min_samples_split	The minimum number of samples required to split an internal node.	10
min_samples_leaf	The minimum number of samples required to be at a leaf node.	6

Table 11. Exact hyper parameters used for Extra Trees.

Parameter	Description	Value
n_estimators	The number of trees in the forest.	230
criterion	The function to measure the quality of a split.	"entropy"
max_depth	The maximum depth of the tree.	9
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
max_features	The number of features to consider when looking for the best split.	"sqrt"
bootstrap	Whether bootstrap samples are used when building trees.	True
min_samples_split	The minimum number of samples required to split an internal node.	10
min_samples_leaf	The minimum number of samples required to be at a leaf node.	6
class_weight	Weights associated with classes.	None

Table 12. Exact hyper parameters used for AdaBoost.

Parameter	Description	Value
n_estimators	The maximum number of estimators.	130
learning_rate	Weight applied to each classifier at each boosting iteration.	0.494
algorithm	Boosting algorithm to use.	"SAMME.R"

Table 13. Exact hyper parameters used for Gradient Boosting Classifier.

Parameter	Description	Value
n_estimators	The number of boosting stages to perform.	180
learning_rate	Learning rate shrinks the contribution of each tree by learning_rate.	0.089
subsample	The fraction of samples to be used for fitting the individual base learners.	0.65
min_samples_split	The minimum number of samples required to split an internal node.	5
min_samples_leaf	The minimum number of samples required to be at a leaf node.	2
max_depth	The maximum depth of the individual regression estimators.	6
min_impurity_decrease	A node will be split if this split induces a decrease of the impurity greater than	0.002

	or equal to this value.	
max_features	The number of features to consider when looking for the best split.	1.0

Table 14. Exact hyper parameters used for Extreme Gradient Boosting.

Parameter	Description	Value
n_estimators	The number of trees.	60
learning_rate	Step size shrinkage used in update to prevents overfitting.	0.129
subsample	Subsample ratio of the training instances.	0.9
max_depth	Maximum depth of a tree.	11
colsample_bytree	Subsample ratio of columns when constructing each tree.	1
min_child_weight	Minimum sum of instance weight (hessian) needed in a child.	1
reg_alpha	L1 regularization term on weights.	5
reg_lambda	L2 regularization term on weights.	0.2
scale_pos_weight	Control the balance of positive and negative weights, useful for unbalanced classes.	15.1

Table 15. Exact hyper parameters used for Light Gradient Boosting.

Parameter	Description	Value
num_leaves	max number of leaves in one tree.	60
learning_rate	Shrinkage rate.	0.2
n_estimators	The number of trees.	240
min_split_gain	The minimal gain to perform split.	0.2
reg_alpha	L1 regularization term on weights.	4
reg_lambda	L2 regularization term on weights.	1
feature_fraction	The ratio of the subset of features	0.9
bagging_fraction	The ratio of the data selected randomly without resampling.	0.5
bagging_freq	Frequency for bagging.	6
min_child_samples	Minimal number of data in one leaf.	95

Table 16. Exact hyper parameters used for CatBoost.

Parameter	Description	Value
eta	The learning rate.	0.001
depth	Depth of the tree.	6
n_estimators	The maximum number of trees.	210
random_strength	The amount of randomness to use for scoring splits when the tree structure is selected.	0.6

l2_leaf_reg	Coefficient at the L2 regularization term of the cost function.	6

Table 17. Exact hyper parameters used for MLP.

Parameter	Description	Value
batch_size	Batch size for training.	1024
time_steps	Time steps used as inputs.	90
lr	Learning rate of the optimizer.	0.1
epochs	Number of epochs for training.	21
patience	Patience of early stopping.	5
optimizer	Optimizer for training.	SGD
layers	The number of layers and neurons in the hidden layers.	[500, 500, 500]
fc_dropout	Dropout rate in the fully connected layers.	0.2
use_bn	Use batch normalization or not.	False

Table 18. Exact hyper parameters used for RNN.

Parameter	Description	Value
batch_size	Batch size for training.	256
time_steps	Time steps used as inputs.	90
lr	Learning rate of the optimizer.	0.0001
epochs	Number of epochs for training.	21
patience	Patience of early stopping.	5
optimizer	Optimizer for training.	RMSProp
n_layers	The number of RNN layers.	2
hidden_size	The number of neurons in the hidden layers.	50
bidirectional	Bidirectional RNN or not.	False

Table 19. Exact hyper parameters used for LSTM.

Parameter	Description	Value
batch_size	Batch size for training.	512
time_steps	Time steps used as inputs.	90
lr	Learning rate of the optimizer.	0.001
epochs	Number of epochs for training.	11
patience	Patience of early stopping.	5
optimizer	Optimizer for training.	Adam
n_layers	The number of RNN layers.	4
hidden_size	The number of neurons in the hidden layers.	100
bidirectional	Bidirectional RNN or not.	True

Table 20. Exact hyper parameters used for GRU.

Parameter	Description	Value
batch_size	Batch size for training.	1024
time_steps	Time steps used as inputs.	90
lr	Learning rate of the optimizer.	0.0001
epochs	Number of epochs for training.	27
patience	Patience of early stopping.	5
optimizer	Optimizer for training.	RMSProp
n_layers	The number of RNN layers.	3
hidden_size	The number of neurons in the hidden layers.	50
bidirectional	Bidirectional RNN or not.	True

Table 21. Exact hyper parameters used for FCN.

Parameter	Description	Value
batch_size	Batch size for training.	1024
time_steps	Time steps used as inputs.	90
lr	Learning rate of the optimizer.	0.0001
epochs	Number of epochs for training.	25
patience	Patience of early stopping.	5
optimizer	Optimizer for training.	RMSProp
layers	The number of layers and neurons in the hidden layers.	[128, 256, 128]
kss	Kernel filter size for convolution.	[11, 9, 7]

Table 22. Exact hyper parameters used for TCN.

Parameter	Description	Value
batch_size	Batch size for training.	512
time_steps	Time steps used as inputs.	90
lr	Learning rate of the optimizer.	0.0001
epochs	Number of epochs for training.	70
patience	Patience of early stopping.	10
optimizer	Optimizer for training.	Adam
layers	The number of layers and neurons in the hidden layers.	[25, 25, 25, 25]
ks	Kernel filter size for convolution.	7
conv_dropout	Dropout rate for convolution.	0.2

Table 23. Exact hyper parameters used for InceptionTime.

Parameter	Description	Value
batch_size	Batch size for training.	1024
time_steps	Time steps used as inputs.	90
lr	Learning rate of the optimizer.	0.001
epochs	Number of epochs for training.	47

patience	Patience of early stopping.	5
optimizer	Optimizer for training.	SGD
nf	The number of filters.	32

Table 24. Exact hyper parameters used for XceptionTime.

Parameter	Description	Value
batch_size	Batch size for training.	256
time_steps	Time steps used as inputs.	90
lr	Learning rate of the optimizer.	0.1
epochs	Number of epochs for training.	10
patience	Patience of early stopping.	10
optimizer	Optimizer for training.	SGD
nf	The number of filters.	32

Table 25. Exact hyper parameters used for XCM.

Parameter	Description	Value
batch_size	Batch size for training.	1024
time_steps	Time steps used as inputs.	60
lr	Learning rate of the optimizer.	0.01
epochs	Number of epochs for training.	9
patience	Patience of early stopping.	10
optimizer	Optimizer for training.	RMSProp
nf	The number of filters.	64
fc_dropout	Dropout rate in the fully connected layers.	0.0
bn	Use batch normalization or not.	True