Instructions to replicate ML classification results.

Machine Learning (ML) Models were implemented using WEKA software Version 3.9.

The software can be free downloaded at this link: http://www.cs.waikato.ac.nz/ml/weka/downloading.html . A complete documentation about WEKA software is available at this link: http://www.cs.waikato.ac.nz/ml/weka/ .

Classification results that are reported in the paper were obtained following these steps:

1) Open WEKA 3.9 and choose the *Explorer* application.
2) From the *Preprocess Panel* open the file "**ML Training data - 40 subjects.arff**" (data from 40 participants). This file contains the data of the 5 features that the authors used in the paper to classify subjects as liars or truth-tellers (see "Classifiers performance" section in the manuscript). The variable named "mind_codition" is the class to predict (truth-teller vs liar).
3) Switch in *Classify Panel* and choose Cross-validation as *Test option* setting the number of folds to 10.
4) To replicate the paper results, from the *Classifier* list choose Logistic, SMO, Random Forest, LMT and run the models. The analyses were carried out using the default parameters of classifiers. The details on the default ML classifiers parameters are reported in **Annex 1**.
5) To test the 4 model obtained by the 10-fold cross-validation on the validation sample (20 new participants), select Supplied test set as *Test option*. Upload the file "**ML Training data - 20 subjects.arff**" as supplied test set. Then, to replicate the paper results on validation sample, choose Random Forest, Logistic, SMO, LMT from the *Classifier* list and run the models. The analyses were carried out using the default parameters of classifiers. The details on the default ML classifiers parameters are reported in **Annex 1**.

Annex 1 - Details on ML classifiers parameters

The analyses were carried out in WEKA (Version 3.8) using the default parameters. Note that there is no fine tuning of the parameters in order to increase the classification accuracy.

Random Forest:
weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

- seed -- The random number seed to be used. = 1
- storeOutOfBagPredictions -- Whether to store the out-of-bag predictions. = FALSE
- numExecutionSlots -- The number of execution slots (threads) to use for constructing the ensemble. = 1
- bagSizePercent -- Size of each bag, as a percentage of the training set size. = 100
- numDecimalPlaces -- The number of decimal places to be used for the output of numbers in the model. = 2
- batchSize -- The preferred number of instances to process if batch prediction is being performed. More or fewer instances may be provided, but this gives implementations a chance to specify a preferred batch size. = 100
- printClassifiers -- Print the individual classifiers in the output. = FALSE
- numIterations -- The number of iterations to be performed. = 100
- debug -- If set to true, classifier may output additional info to the console. = FALSE
- outputOutOfBagComplexityStatistics -- Whether to output complexity-based statistics when out-of-bag evaluation is performed. = FALSE
- breakTiesRandomly -- Break ties randomly when several attributes look equally good. = FALSE
- doNotCheckCapabilities -- If set, classifier capabilities are not checked before classifier is built (Use with caution to reduce runtime). = FALSE
- maxDepth -- The maximum depth of the tree, 0 for unlimited. = 0
- calcOutOfBag -- Whether the out-of-bag error is calculated. = FALSE
- numFeatures -- Sets the number of randomly chosen attributes. If 0, int(log_2(#predictors) + 1) is used. = 0

Logistic:
weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

- numDecimalPlaces -- The number of decimal places to be used for the output of numbers in the model. = 4
- batchSize -- The preferred number of instances to process if batch prediction is being performed. More or fewer instances may be provided, but this gives implementations a chance to specify a preferred batch size. = 100
- debug -- Output debug information to the console. = FALSE
- ridge -- Set the Ridge value in the log-likelihood. = 1.0E-8
- useConjugateGradientDescent -- Use conjugate gradient descent rather than BFGS updates; faster for problems with many parameters. = FALSE
- maxIts -- Maximum number of iterations to perform. = -1
- doNotCheckCapabilities -- If set, classifier capabilities are not checked before classifier is built (Use with caution to reduce runtime). = FALSE

SMO:
weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4"

- buildCalibrationModels -- Whether to fit calibration models to the SVM's outputs (for proper probability estimates). = FALSE
- numFolds -- The number of folds for cross-validation used to generate training data for calibration models (-1 means use training data). = -1
- randomSeed -- Random number seed for the cross-validation. = 1
- c -- The complexity parameter C. = 1.0
- numDecimalPlaces -- The number of decimal places to be used for the output of numbers in the model. = 2
- batchSize -- The preferred number of instances to process if batch prediction is being performed. More or fewer instances may be provided, but this gives implementations a chance to specify a preferred batch size. = 100
- kernel -- The kernel to use. Polykernel –C 250007 -E 1.0
- checksTurnedOff -- Turns time-consuming checks off - use with caution. = FALSE
- debug -- If set to true, classifier may output additional info to the console. = FALSE
- filterType -- Determines how/if the data will be transformed. = Normalized training data
- toleranceParameter -- The tolerance parameter (shouldn't be changed). = 0.001
- calibrator -- The calibration method to use. = Logistic
- doNotCheckCapabilities -- If set, classifier capabilities are not checked before classifier is built (Use with caution to reduce runtime). = FALSE
- epsilon -- The epsilon for round-off error (shouldn't be changed). 1.0E-12

LMT:

weka.classifiers.trees.LMT -I -1 -M 15 -W 0.0

- splitOnResiduals -- Set splitting criterion based on the residuals of LogitBoost. There are two possible splitting criteria for LMT: the default is to use the C4.5 splitting criterion that uses information gain on the class variable. The other splitting criterion tries to improve the purity in the residuals produces when fitting the logistic regression functions. The choice of the splitting criterion does not usually affect classification accuracy much, but can produce different trees. = FALSE
- useAIC -- The AIC is used to determine when to stop LogitBoost iterations. The default is not to use AIC. = FALSE
- numDecimalPlaces -- The number of decimal places to be used for the output of coefficients. = 2
- batchSize -- The preferred number of instances to process if batch prediction is being performed. More or fewer instances may be provided, but this gives implementations a chance to specify a preferred batch size. = 100
- weightTrimBeta -- Set the beta value used for weight trimming in LogitBoost. Only instances carrying (1 - beta)% of the weight from previous iteration are used in the next iteration. Set to 0 for no weight trimming. The default value is 0. = 0.0
- doNotMakeSplitPointActualValue -- If true, the split point is not relocated to an actual data value. This can yield substantial speed-ups for large datasets with numeric attributes. = FALSE
- debug -- If set to true, classifier may output additional info to the console. = FALSE
- numBoostingIterations -- Set a fixed number of iterations for LogitBoost. If >= 0, this sets a fixed number of LogitBoost iterations that is used everywhere in the tree. If < 0, the number is cross-validated. = -1
- fastRegression -- Use heuristic that avoids cross-validating the number of Logit-Boost iterations at every node. When fitting the logistic regression functions at a node, LMT has to determine the number of LogitBoost iterations to run. Originally, this number was cross-validated at every node in the tree. To save time, this heuristic cross-validates the number only

once and then uses that number at every node in the tree. Usually this does not decrease accuracy but improves runtime considerably. = TRUE

- minNumInstances -- Set the minimum number of instances at which a node is considered for splitting. The default value is 15. = 15
- doNotCheckCapabilities -- If set, classifier capabilities are not checked before classifier is built (Use with caution to reduce runtime). = FALSE
- errorOnProbabilities -- Minimize error on probabilities instead of misclassification error when cross-validating the number of LogitBoost iterations. When set, the number of LogitBoost iterations is chosen that minimizes the root mean squared error instead of the misclassification error. = FALSE
- convertNominal -- Convert all nominal attributes to binary ones before building the tree. This means that all splits in the final tree will be binary. = FALSE