Table S1. ML algorithm full parameters.

Classifier algorithm	Algorithm's hyper-parameters for CWT dataset	Algorithm's hyper-parameters for RAW dataset
ANN	Number of learnable parameters= 8.4k, Solver= 'ADAM', Mini batch size= 64, Shuffle= 'every-epoch', Max epoch= 50, Learn Rate= 0.01, LR drop period= 10, LR drop factor= 0.9, Output network= 'best validation loss'. *Layers:* *Sequence input (540);* *Fully connected (16);* *Batch normalization;* *Fully connected (5);* *Batch normalization;* *Softmax;* *Classification;*	Number of learnable parameters= 463, Solver= 'ADAM', Mini batch size= 64, Shuffle= 'every-epoch', Max epoch= 50, Learn Rate= 0.01, LR drop period= 10, LR drop factor= 0.9, Output network= 'best validation loss'. *Layers:* *Sequence input (20);* *Fully connected (16);* *Batch normalization;* *Fully connected (5);* *Batch normalization;* *Softmax;* *Classification;*
DT	Max. epochs= 50, Number of learnable parameters= 2, Max. Num. splits= 10, Min. Parent size= 2, Num. grid divisions= 10, KFold= 10, Surrogate= 'off' Bayesian optimization of split criterion and min. leaf size.	Max. epochs= 50, Number of learnable parameters= 2, Max. Num. splits= 10, Min. Leaf size= 1, Min. Parent size= 2, Num. grid divisions= 10, KFold= 10, Surrogate= 'off' Bayesian optimization of split criterion.
SVM	Coding= 'onevsone', Learners= 'SVM', Number of learnable parameters=6, Max epoch= 7, KFold= 4, Class prior probabilities= 'empirical' Bayesian optimization of Box constraint and kernel size.	Coding= 'onevsone', Learners= 'SVM', Number of learnable parameters=6, Max epoch= 7, KFold= 4, Class prior probabilities= 'empirical' Bayesian optimization of Box constraint and kernel size.
NB model	Data distribution= 'gaussian', Kernel smoother type= 'normal', Support= 'unbounded', KFold= 10, Prior probabilities= 'empirical'.	Data distribution= 'gaussian', Kernel smoother type= 'normal', Support= 'unbounded', KFold= 10, Prior probabilities= 'empirical'.
LSTM NN	Number of learnable parameters= 410k, LSTM Solver= 'SGDM', Mini batch size= 16, Shuffle= 'every-epoch', Max epoch= 60, Learn Rate= 0.05, Output network= 'best validation loss'.	Number of learnable parameters= 6.9k LSTM Solver= 'SGDM', Mini batch size= 16, Shuffle= 'every-epoch', Max epoch= 60, Learn Rate= 0.05, Output network= 'best validation loss'.
LSTM NN	*Layers:* *Sequence input (540);* *biLSTM (64);* *Relu;* *Batch normalization;* *Dropout (0.5);* *biLSTM (64);* *Relu;* *Batch normalization;* *Dropout (0.5);* *Fully connected (5);* *Batch normalization;* *Softmax;* *Classification;*	*Layers:* *Sequence input(20);* *biLSTM (12);* *Relu;* *Batch normalization;* *Dropout (0.5);* *biLSTM (12);* *Relu;* *Batch normalization;* *Dropout (0.5);* *Fully connected (5);* *Batch normalization;* *Softmax;* *Classification;*
AE+Softmax	Number of bottleneck neurons= 100, Number of softmax neurons= 5,	Number of bottleneck neurons= 12, Number of softmax neurons= 5,

AE+Softmax	Number of learnable parameters= 109k, Solver= 'Scaled conjugate gradient', Encoder Max epoch= 400, Softmax Max epoch= 400, Encoder transfer function= logsig,, Decoder transfer function= logsig, L2 weight regularization= 0.001, Loss function= MSE sparse, Sparsity proportion= 0.05, Sparsity regularization= 1.	Number of learnable parameters= 522, Solver= 'Scaled conjugate gradient', Encoder Max epoch= 400, Softmax Max epoch= 400, Encoder transfer function= logsig,, Decoder transfer function= logsig, L2 weight regularization= 0.001, Loss function= MSE sparse, Sparsity proportion= 0.05, Sparsity regularization= 1.
AE+biLSTM NN	Encoder bottleneck neurons= 100, Number of learnable parameters= 171k, Encoder Max epoch= 400, Encoder transfer function= logsig, Decoder transfer function= logsig, L2 weight regularization= 0.001, Loss function= MSE sparse, Sparsity proportion= 0.05, Sparsity regularization= 1, LSTM Solver= 'ADAM', Mini batch size= 8, Shuffle= 'every-epoch', Max epoch= 75, Learn Rate= 0.05, Output network= 'best validation loss'. *Layers:* *Sequence input (540);* *biLSTM (32);* *Relu;* *Batch normalization;* *Dropout (0.5);* *biLSTM (32);* *Relu;* *Batch normalization;* *Dropout (0.5);* *Fully connected (5);* *Batch normalization;* *Softmax;* *Classification;*	Encoder bottleneck neurons= 12, Number of learnable parameters= 60k, Encoder Max epoch= 400, Encoder transfer function= logsig, Decoder transfer function= logsig, L2 weight regularization= 0.001, Loss function= MSE sparse, Sparsity proportion= 0.05, Sparsity regularization= 1, LSTM Solver= 'ADAM', Mini batch size= 8, Shuffle= 'every-epoch', Max epoch= 75, Learn Rate= 0.05, Output network= 'best validation loss'. *Layers:* *Sequence input (20);* *biLSTM (32);* *Relu;* *Batch normalization;* *Dropout (0.5);* *biLSTM (32);* *Relu;* *Batch normalization;* *Dropout (0.5);* *Fully connected (5);* *Batch normalization;* *Softmax;* *Classification;*
CNN+RNN	Num. of learnable parameters= 9.4M, Solver= 'RMSPROP, Mini batch size= 64, Shuffle= 'every-epoch', Max epoch= 50, Learn Rate= 0.01, LR drop period= 10, LR drop factor= 0.9, Output network= 'best validation loss'. *Layers:* *Sequence input(540);* *1D convolution (filt.Size=64; num.filt.=256);* *1D MaxPool (8);* *Batch normalization;* *Relu;* *1D convolution (filt.size=32; num.filt.=64);* *1D MaxPool (8);* *Batch normalization;* *Relu;* *1D convolution (filt.Size=16; num.filt.=32);* *1D MaxPool (8);* *Batch normalization;* *Relu;* *Fully connected (8);* *Batch normalization;* *Fully connected (5);* *Batch normalization;* *Softmax;* *Classification;*	Num. of learnable parameters= 15k, Solver= 'RMSPROP, Mini batch size= 64, Shuffle= 'every-epoch', Max epoch= 50, Learn Rate= 0.01, LR drop period= 10, LR drop factor= 0.9, Output network= 'best validation loss'. *Layers:* *Sequence input(20);* *1D convolution (filt.Size=8; num.filt.=16);* *1D MaxPool (6);* *Batch normalization;* *Relu;* *1D convolution (filt.size=16; num.filt.=16);* *1D MaxPool (4);* *Batch normalization;* *Relu;* *1D convolution (filt.Size=32; num.filt.=16);* *1D MaxPool (4);* *Batch normalization;* *Relu;* *Fully connected (8);* *Batch normalization;* *Fully connected (5);* *Batch normalization;* *Softmax;* *Classification;*

CNNA+RNN	Num. of learnable parameters= 47M, Parallel CNN streams= 5, Solver= 'RMSPROP, Mini batch size= 64, Shuffle= 'every-epoch', Max epoch= 50, Learn Rate= 0.01, LR drop period= 10, LR drop factor= 0.9, Output network= 'best validation loss'. *Layers:* *Sequence input(540);* *1D convolution (filt.Size=64; num.filt.=256);* *1D MaxPool (8);* *Batch normalization;* *Relu;* *1D convolution (filt.size=32; num.filt.=64);* *1D MaxPool (8);* *Batch normalization;* *Relu;* *1D convolution (filt.Size=16; num.filt.=32);* *1D MaxPool (8);* *Batch normalization;* *Relu;* *ABOVE LAYERS ARE USED FIVE TIMES, WIRED IN PARALLEL* *Add (5);* *Fully connected (8);* *Batch normalization;* *Fully connected (5);* *Batch normalization;* *Softmax;* *Classification;*	Num. of learnable parameters= 75k, Parallel CNN streams= 5, Solver= 'RMSPROP, Mini batch size= 64, Shuffle= 'every-epoch', Max epoch= 50, Learn Rate= 0.01, LR drop period= 10, LR drop factor= 0.9, Output network= 'best validation loss'. *Layers:* *Sequence input(20);* *1D convolution (filt.Size=8; num.filt.=16);* *1D MaxPool (6);* *Batch normalization;* *Relu;* *1D convolution (filt.size=16; num.filt.=16);* *1D MaxPool (4);* *Batch normalization;* *Relu;* *1D convolution (filt.Size=32; num.filt.=16);* *1D MaxPool (4);* *Batch normalization;* *Relu;* *ABOVE LAYERS ARE USED FIVE TIMES, WIRED IN PARALLEL* *Fully connected (8);* *Batch normalization;* *Fully connected (5);* *Batch normalization;* *Softmax;* *Classification;*