**Glossary of Acronyms and Other Commonly Used Terms**

**Acronyms**

* **AD**:All data
* **BPTT**: Backpropagation through time
* **D**: Duchenne (subject)
* **DC**: Data cube
* **DMD**: Duchenne muscular dystrophy
* **FFNN**: Feedforward neural network
* **FRC**: Feature-reduced via concatenation
* **HC**: Healthy control (subject)
* **GOSH**: Great Ormond Street Hospital
* **JA**: Joint angle
* **LSTM**: Long short-term memory
* **MAE**: Mean average error
* **MD**: Muscular dystrophy
* **MPS**: Model predictions set
* **MSE**: Mean squared error
* **NMB**: Natural movement behaviour
* **NSAA**: North Start Ambulatory Assessment
* **PCA**: Principal component analysis
* **RMSE**: Root mean squared error
* **RNN**: Recurrent neural network

**Commonly Used Terms**

* **Activity time**: Refers to the start and end times of an observable NSAA activity within a source ‘.mat’ file.
* **Acts**: Output type that corresponds to the model classifying an input sequence with a series of 17 numbers, each of which containing either a 0, 1, or 2 (i.e. not continuous values) that corresponds to the models guess as to the individual acts scores it believes the single sequence corresponds to.
* **AD (referring to file)**: The most common type of source ‘.mat’ file that we work with that contains all the captured information by the body suit for a single subject. This contrasts with ‘JA’ and ‘DC’, which only contain joint angle measurements captured by the suit.
* **AD (referring to input type or measurement)**: Short hand (that is recorded within ‘model\_predictions.csv’ and referenced elsewhere) for computed statistical values; hence, in places where measurements include ‘AD’ along with other raw measurements, for example, simply take this to mean ‘computed statistical values’.
* **Additional directory (or ‘add dir’)**: Refers to the process of RNN models using more than one source directory for the files to train the model. For example, a model might use processed (e.g. raw-measurement-extracted) files from the ‘NSAA’ directory and the ‘6minwalk-matfiles’ additional directory to use as ‘x’ and ‘y’ data in the training/testing process.
* **Alternative directory (or ‘alt dir’)**: Refers to the ‘model\_predictor.py’ script using models trained on directories that are not the same as the directory of the assessing file(s). For example, ‘model\_predictor.py’ might assess subject ‘D4’s files from ‘NSAA’ on models trained on ‘allmatfiles’ directory files.
* **Annotator**: Anyone who has made active contributions to the Google annotations sheet regarding inputting start and end times of activities for the subjects by observing videos of the NSAA assessments or otherwise.
* **Argument**: Console arguments that are provided along with the name of the script being invoked by the user (or by a batch script) that change what the script does or otherwise sets values that impact what is produced by the script.
* **Computed statistical values**: Values that are computed over each of the source ‘.mat’ files and over all of the measurements that include means, variances, eigenvalues of covariance matrices, etc., over all the features and several combinations of features. The results are stored in separate ‘.csv’ files as ‘<subject short name>\_stat\_features.csv’, which are expected to be operated on by ‘ft\_sel\_red.py’ to reduce dimensionality.
* **Complete-act file**: In contrast to a single-act file, this generally refers to any file input to an RNN model for any measurement that does not originate from a single-act file (e.g. ‘D4\_acceleration.csv’).
* **Console output**: Output of various scripts that is printed directly to whatever console was used to invoke the scripts. Contains information about the results of building models, assessing subjects (that is also written to ‘model\_predictions.csv’), and current progress of executing a longer script such as ‘comp\_stat\_vals.py’.
* **Curse of dimensionality**: Refers to the consequences of using higher-dimensionality data that make learning from said data increasing more difficult as the dimensionality increases; feature selection/reduction techniques are generally chosen in order to combat these effects.
* **Data pipeline**: The part of the system that ingests data in various forms and produces data in other forms that link together; generally includes ‘comp\_stat\_vals.py’, ‘ft\_sel\_red.py’, ‘mat\_act\_div.py’, ‘ext\_raw\_measures.py’, and ‘rnn.py’.
* **Data file type**: Refers to the different forms the data from the suit can take that includes ‘AD’, ‘JA’ and ‘DC’.
* **Dhc**: Output type that corresponds to the model classifying on a single output node that outputs a ‘D’ or ‘HC’ value based on whether the model believes the inputted sequence comes from a ‘D’ or ‘HC’ file, respectively.
* **Discard proportion**: The process of discarding a proportion of the frames from within a single sequence of data, generally used with a large sequence overlap and a very long sequence. For example, if the discard proportion is set to ‘0.9’ and we have a sequence of length ‘600’, we keep only every 10th frame within the sequence to end up with a new sequence of length ‘60’.
* **Downsampling**: The process of removing a large amount of data from a data set to end up with a smaller data set whose ‘y’ labels are evenly distributed. For example, if we have 6 ‘x’ sequences of ‘y’ labels [‘1’, ‘1’, ‘2’, ‘2’, ‘2’, ‘3’], we would observe that the least frequent label amount is 1 (for ‘3’), so we would randomly sample (with replacement) 1 sample of each ‘x’ sequence with corresponding ‘y’ label to reduce to a size of 3 with labels [‘1’, ‘2’, ‘3’].
* **D/HC classification**: The act of classifying either a single sequence or a complete file/subject as being (from) either a ‘D’ subject (i.e. a subject with DMD) or (from) an ‘HC’ subject (i.e. a healthy control subject).
* **Experiment set**: Set of experiments that generally involve several models being built with different input types or model hyperparameter settings to assess whether or not the models improve on the test set with respect to these changes.
* **Familiar/non-familiar**: Refers to models that have either seen the respective subject during the model training process or have not, respectively.
* **Feature concat**: The act of, instead of building separate models for different input types, concatenating horizontally the input data prior to building a single model for all the input types. In other words, instead of having data shapes for three models for three input types of ‘(a, b, c)’, ‘(a, b, d)’, and ‘(a, b, e)’, we instead get a single model with a data shape of ‘(a, b, c+d+e)’.
* **Feature-reduced via concatenation**: Refers to a process within ‘ft\_sel\_red.py’ where the dimensionality is reduced over all the input files in the same way (i.e. all files are projected to the same feature subspace), as opposed to the standard way ‘ft\_sel\_red.py’ where each file has their dimensions reduced in turn.
* **Features**: Generally refers to the dimensions of a certain measurement that is determined either by the settings of the suit (i.e. 66 is set by the suit because it measures 22 joint angles in three dimensions) or determined by ‘ft\_sel\_red.py’.
* **Frame**: A single row of a sequence or data prior to being divided into sequences, this is a single vector with a number of values equal to the number of features of the measurement we are concerned with. This can be a single row of data within a source ‘.mat’ file, a row within a sequence that is to be fed into a model, or a row of data of a complete matrix of ‘x’ data.
* **Indiv**: Output type that corresponds to the model regressing on a single output node that outputs a continuous value between 0 or 2 for input data from single-act files.
* **Individual NSAA Score**: A single-activity NSAA score, as determined by a specialist, which is either a 0, 1, or 2, with a 0 being the subject being unable to complete the activity, a 1 being the subject completes the activity with difficult, and a 2 being the subject completing the activity as if they didn’t have DMD.
* **Input file**: (see ‘Data file type’; not to be confused with ‘Input type’).
* **Input type**: The different types of data that can be used to train a model and is thus mostly synonymous with the definition of ‘measurement’.
* **Intermediate data**: Data that exists as a product of one script and that is consumed by another script. For example, raw measurement data that is produced by ‘ext\_raw\_measures.py’ and subsequently used by ‘rnn.py’ is a form of intermediate data that is in contrast to the raw data of the source ‘.mat’ files.
* **Left-out subject**: A subject where none of their data was used as part of the training process, and hence the assessment on this subject by ‘model\_predictor.py’ as part of a model predictions set is equivalent to the model performing on a brand-new, never seen before subject (i.e. how it would function in production).
* **Local directory**: The location where all the data that is used and created by the system is stored for the user. The location of this is generally set by modifying the ‘local\_dir’ variable within ‘settings.py’.
* **Measurement**: Refers to a specific type of data that can be extracted from the suit data source files that contain the suit data from the subjects carrying out assessments (e.g. NSAA or 6-minute walk) or natural movement behaviour. Can be either a raw measurement that is taken directly from the source ‘.mat’ via the ‘ext\_raw\_measure.py’ script (e.g. ‘jointAngle’ or ‘acceleration’) or computed statistical values that are created via the ‘comp\_stat\_vals.py’ script.
* **Metric**: Usually refers to a calculation done to assess the performance of an experiment or model prediction for a given output type; metrics can include ‘MAE of overall NSAA scores in the test set’ and ‘the percent of correctly predicted sequences for D/HC classification’.
* **Model**: Used as a noun, this refers to a single TensorFlow-built model that’s built by ‘rnn.py’ on one or more input types (i.e. measurements such as joint angles or computed statistical values) and to target one specific output type, such as regression to the overall NSAA score (‘overall’) or classifying D/HC labels (‘dhc’). These models are trained on sequenced data, stored in directories within ‘rnn\_models’, and are loaded via ‘model\_predictor.py’ to predict on (generally) unseen complete subjects.
* **Model directory**: The location with respect to the local directory (‘local\_dir’ of ‘settings.py’) of the models created by various runs of ‘rnn.py’. The path to this is given as ‘<local directory>\output\_files\rnn\_models’.
* **Model predictions set**: A set of predictions for (generally) various left-subjects with respect to different model variations to investigate generalization performance. In other cases, refers to the process of utilizing ‘model\_predictor.py’ in other ways, such as with the ‘test\_altdirs.py’ script.
* **Natural movement behaviour**: The suit capture data of subjects moving in non-assessment environments doing activities such as playing, eating, and otherwise interacting with others, among other tasks; this is in contrast to the specific assessments of NSAA and 6-minute walks.
* **NSAA**: Generally refers to the assessment involving 17 activities that subjects carry out for specialists to score each of on a 0 to 2 discrete scale, the accumulation of which is the overall NSAA score.
* **Output type**: The different types of output that we can configure the RNN models to produce that includes ‘overall’, ‘dhc’, ‘acts’, and ‘indiv’.
* **Overall**: Output type that corresponds to the model regressing on a single output node that outputs a continuous value between 0 or 34 for the model’s prediction of the overall NSAA score of the file that the inputted sequence comes from.
* **Overall NSAA score**: The summation of the individual activity scores for a given subject that will therefore range between 0 (i.e. none of the activities being able to have been completed) or 34 (i.e. all activities completely adequately, in the case of ‘HC’ subjects).
* **Preprocessing**: The steps undertaken by either ‘rnn.py’ and/or ‘ft\_sel\_red.py’ to carry out a variety of steps to prepare the data for input into the model including (but not limited to) pulling all the necessary data from the specified directories, creating sequences with specified length and overlap, and dividing the data into training and testing components. Generally this is all done by the ‘preprocessing()’ function in ‘rnn.py’.
* **Production**: Refers to the act of using this system in its ‘final’ state (i.e. with all experiments completed) and using this as a tool for new subject assessment.
* **Project directory**: Refers to the location of the directory that contains all parts of the project that does NOT include the raw data needed (which instead are stored within the local directory) that includes the source code, documentation, batch scripts, code README files, etc.
* **Raw measurement**: A measurement that is directly extracted from source ‘.mat’ files and subsequently fed directly into models, as opposed to computed statistical values that are instead fed through ‘comp\_stat\_vals.py’ and ‘ft\_sel\_red.py’.
* **Reference documents**: A collection of documents that provides necessary information to provide the data with ‘y’ labels along with storing the results of experiment sets and model predictions sets.
* **Sampling rate**: The rate at which the suit samples data from each of its sensors for the different measurements; this is fixed at 60Hz, i.e. 60 rows of data produced each second for every measurement.
* **Script**: Generally refers to either a Python file (with ‘.py’ extension) that encompasses the core parts of the system or batch files (with ‘.cmd’ extension) that are used to run model prediction sets or carry out other tasks that involve the calling of Python files.
* **Sequence**: A two-dimensional grouping of data that the RNN will treat as a single sample that is fed in, with a number of rows equal to the sample length and the number of columns equal to the number of features that the input type in question contains (e.g. 66 if we’re using joint angle data).
* **Sequence length**: The number of frames of values within a single sequence. For example, if the sequence length is set to 10 for position data (with 69 features), then each sequence has a shape of ‘(10, 69)’.
* **Sequence overlap**: The proportion of frames that overlap with the adjacent sequence. For example, with a sequence overlap of 0.2, then for a sequence of shape ‘(10, 69)’, the bottom 2 rows of ‘69’ values will be the first two frames of data for the next sequence also of shape ‘(10, 69)’.
* **Single-act concat**: The process of ‘mat\_act\_div.py’ extracting the single-act files from the source ‘.mat’ files as normal but, instead of keeping them as individual files, vertically concatenating these files. This creates complete-act files (as the files were originally) but now with any data from ‘between’ the activities removed.
* **Single-act file**: The product of the ‘mat\_act\_div.py’ script, these are ‘.mat’ files that contain only the movement data of a single activity of the NSAA assessment. For example, the source ‘.mat’ file ‘D4\_nsaa.mat’ would have 17 (generally speaking) new files created by ‘mat\_act\_div.py’ with names such as ‘D4\_act<activity number>\_nsaa.mat’. See the relevant section in the system overview for ‘mat\_act\_div.py’ for more information as to how this is done and for what reasons.
* **Source ‘.mat’ file(s)**: These are the MATLAB data files in their initial states that are provided to us directly as part of the project that have been directly translated from their ‘.mvnx’ equivalent files that are produced by the suits. Hence, these are the files for each subject we are concerned with prior to extracting any raw measurements or computed statistical values.
* **Specialist**: Any medical specialist involved in the process of human analysis of the severity of DMD for each of the subjects.
* **Standard approach**: Used in the context of model prediction sets, this tends to refer to the ‘best’ approach to building models and have them generalize to ‘left-out’ subjects. This often means MPS 3, which for many MPS’s is determined to be the currently optimal setup and hence is used as the default comparison to whichever approach we are currently assessing.
* **Subject**: Refers to the participants of the wider Imperial and Great Ormond Street Hospital DMD study that includes ~ 20 ‘D’ subjects (i.e. subjects with varying degrees of DMD) and ~10 ‘HC’ subjects (healthy control subjects that don’t have DMD).
* **Subject assessment**: Can either refer to the assessment made by specialists with the subject carrying out the NSAA or 6-minute walk assessments, or assessment made by the ‘model\_predictor.py’ script on a subject or specific file, the results of which appear in the ‘model\_predictions.csv’ file.
* **Subject short name**: The names that the subjects go by in this project to make it easier to refer to them at various points in the project and to keep certain aspects of anonymity intact; e.g. ‘D4’, ‘D16’, ‘HC2’, etc.
* **System**: The complete set of scripts and supporting documents that are used to carry out all the experiment sets and model predictions sets and that encompasses the entirety of the work done for this project; does not include the local computational resources the user uses to execute any parts of the system, as the system is intended to be fairly platform independent.
* **Vector sample**: (see ‘Frame’).
* **‘X’ data**: The sequences themselves of the data being fed into the RNN models. After preprocessing, this will be of shape ‘(a, b, c)’, where ‘a’ is the number of sequences to train and test the model on (note: this is considering the data before the train and test split process), ‘b’ is the sequence length, and ‘c’ is the number of features for the given input type.
* **‘Y’ data**: The labels that correspond to each sequence of data that can be either binary values (for the ‘dhc’ output type), continuous values between 0 and 2 (for the ‘indiv’ output type), continuous values between 0 and 34 (for the ‘overall’ output type), or lists of 17 values each of either 0, 1, or 2 (for the ‘acts output type). Note that the number of these labels is the same as the number of sequences of data in ‘x’ data’.