**Final Evaluation Results**

**Conclusions from Experiment Sets and Model Predictions Sets**

Now that all the experimentation with the system and corresponding experiment sets and model predictions sets have been completed, it’s worth conducting an overview of the primary takeaways from the various aspects of experimentation. As these covered different parts and applications of the system, some of the conclusions we can draw are somewhat disconnected from others, and there isn’t a particularly unifying concept surrounding many of them. However, all conclusions drawn can fall under the umbrella of drawing insights about how RNNs can learn from human movement data and what is needed from the data in order to effectively do so. It should also be noted that some of the experimentation results are contradicted by later results obtained due to the modification of the system (e.g. MPS 4’s claim that aggregating the measurements for all output types is an ideal method is later contradicted by MPS 20’s claim that sensor magnetic field is the best measurement type to use on its own); as these later results are obtained by extensive system modification to obtain more accurate file assessments, the results from the later experimentation are generally asserted to be the most likely to be true.

The following are some of the more significant takeaways from the experimentation which most likely have the potential to impact other studies and that hold the most significance in the building of a file assessment system within the project, along with the experiment sets and/or model predictions sets that support these claims. Many of these help justify architectural decisions made in creating the final models to be used as ‘production’ models to assess new files (more on this in the following section), whilst others serve as insights into the DMD data (and hence the subjects themselves) which may prove to be useful to specialists:

* The most important activities within the NSAA assessment (whose usefulness in assessing a subject for their true overall NSAA score is greater than other activities) seems to be those using the left-leg in more muscular-demanding activities and also involving the subject standing upright throughout the activity.
  + MPS 23
* When it comes to assessing a left-out subject, the ‘V2’ versions of subjects, or assessing files trained on ‘alt directories’, the use of only the sensor magnetic field data is preferable to the use of any other raw measurement data, computed statistical values, or any other combination of said measurements.
  + MPS 20, 21, 22
* It’s very possible, and possibly even ideal, to assess models built on one data set (e.g. ‘NSAA’) with files from a different data set (e.g. ‘NMB’) than was used to train the model, as these show very accurate overall NSAA score predictions upon the assessed file which the models have not seen before.
  + MPS 22
* While we do not have the true overall and single-act NSAA scores for the ‘V2’ files within the data sets, we do see that models that are trained on ‘V1’ files and tested on ‘V2’ files (e.g. ‘D4V2-sensorMagneticField.csv’) show results that seem very plausible (e.g. on average predicting a drop in NSAA performance for the subject over the 6-month period between the versions) and therefore the system could be easily adapted to monitor subjects’ progress over time given regular or semi-regular suit data inputs.
  + MPS 21
* The use of only sensor magnetic field data from the NSAA data set with additional files from the NMB data set, the full data set being used for the training set, and computing an aggregated overall NSAA score over all output types gives the models the best ability to generalize its assessments to new subjects.
  + MPS 16, 18, 19, 20
* The addition of noise to the data set, the extraction of only a single sequence from the assessing file to assess the file with, the feature reduction of raw measurement data, the addition of 6-minute walk data to train NSAA-based models, the concatenation of features for multiple raw measurements, the downsampling of the NSAA data set, the reduction of NSAA source ‘.mat’ files to only their 17 activity components, the removal of the outlier subject from the data set, and the changing of how the computed statistical values have their features reduced all have either no effect or an outright negative effect on the models’ ability to generalize to unseen subjects.
  + MPS 5, 8, 9, 10, 11, 13, 14, 15, 17
* Optimal model performance for this type of movement data was found to be using a very large sequence length for raw measurements (e.g. sequence length of 600 which covered a 10 second time-context window) but with a discard proportion of 0.9 (i.e. only keeping every 10th row of the 600 to be inputted into the RNN) and a sequence overlap of 0.9 (so 90% of each sequence overlaps with the next sequence), so we get a context window of DMD movement of 10 seconds while keeping the sequence input into the models as limited as possible with a high overlap so the possibility of an NSAA activity not being completely captured by a sequence is limited.
  + ES 10
* A sequence length for computed statistical values was also found to be ideal at 10 with a high sequence overlap of 0.9 (that increased the total amount of sequences we could use and increased the likelihood of capturing complete activities in a single sequence) and features of the reduced-dimensionality data set to 30 which (as this corresponds to a 10 second time-context window per sequence) shows therefore that, for any type of measurement, a 10 second context window is ideal for this particular type of movement data.
  + ES 4, 7, 8, 9
* Building models and then assessing them on 20% of the data from the total data set but with no specific subject left-out of the data set (i.e. the test set therefore being data that the models haven’t seen before but from subjects it has seen before) shows incredibly accurate results for all 4 output types in both classification- and regression-based tasks, with the position, sensor magnetic field, joint angle, and joint angle XZY measurements being the far more useful raw measurements to draw from the ‘AD’ files to build models from than other measurements like velocity and acceleration to train models; however, this degree of performance doesn’t translate to as good performance on left-out subjects or to new versions of existing subjects.
  + ES 1, 2, 3

**Final Models Directory**

With all the experimentation now complete and the optimal setups of models ascertained (including the different hyperparameters, input types to use, etc.), we wished to extract the models from the set of all created models that we have used for the project. In total, there were 762 models created, many of which were only used for a single experiment set or model predictions set. Therefore, in any effort to separate the models created for experimentation purposes and those being used for ‘production’ (in other words, the models called upon by a user assessing a file in it’s final state), we copied the necessary models from the ‘rnn\_models’ directory located within the local directory and added them to another directory, ‘rnn\_models\_final’, within the project directory under ‘<project directory>\source\’. This was done for several reasons:

* It allows one to use the system completely free of the local directory (thus not requiring the large amount of data to download or models to setup via the data pipeline and the ‘rnn.py’ script). Thus, a user would only need the models within ‘rnn\_models\_final’ to assess files using ‘model\_predictor.py’ and/or the wrapper ‘assess\_nsaa\_nmb\_file.py’ script.
* As the project directory contains the majority of what constitutes the project deliverables, we add in the chosen models as another deliverable in the form of a distinct sub-directory; these are the deliverables as a consequence of many experiment sets and model predictions sets where we looked for the best possible model setups to solve the problems set out in the ‘Aims and Objectives’ section of the ‘Project Overview’ chapter.
* We wouldn’t necessarily want to include all created models in ‘rnn\_models\_final’ as these will not be accessed via the ‘assess\_nsaa\_nmb\_file.py’ script, as by definition they would not be the best models to assess with via ‘model\_predictor.py’ (as they have been proven to be the inferior option via the various experiment sets and model predictions sets); furthermore, they are a form of intermediate data and thus predominantly lie within the scope of the local directory, along with other forms of intermediate data (such as computed statistical values and extracted raw measurements).

With regards to the models themselves, they broadly fall under three categories, which encompass all 48 models contained within ‘rnn\_models\_final’:

1. Models trained on all the subjects (i.e. no left-out subjects) contained within the NSAA data set, with supplementary data from NMB (maximum of 3 files per subject), all of the data available used for training, a sequence length of 600, a sequence overlap of 0.9, a discard proportion of 0.9, and the number of training epochs set to 20. One model is trained per combination of input types (joint angle, sensor magnetic field, position, and AD) and output types (‘dhc’, ‘overall’, and ‘acts’) for 12 total models.
2. Models trained on all the subjects (i.e. no left-out subjects) except for files of ‘V2’ subjects (e.g. a file containing ‘D4V2’ would not be used) contained within the NSAA data set, with supplementary data from NMB (maximum of 3 files per subject), all of the data available used for training, a sequence length of 600, a sequence overlap of 0.9, a discard proportion of 0.9, and the number of training epochs set to 20. One model is trained per combination of input types (joint angle, sensor magnetic field, position, and AD) and output types (‘dhc’, ‘overall’, and ‘acts’) for 12 total models.
3. Models trained exclusively on either the NSAA or NMB data set with no subjects left-out, which is then useful in analyzing subject files from an alternative directory. All of the data available in either data set is used for training, along with a sequence length of 600, a sequence overlap of 0.9, a discard proportion of 0.9, and the number of training epochs set to 20. One model is trained per directory used for model training (NSAA and NMB), per combination of input types (joint angle, sensor magnetic field, position, and AD) and output types (‘dhc’, ‘overall’, and ‘acts’) for 24 total models.

It should be noted that categories 2 and 3 will have been created and used by the user undertaking model predictions sets 21 and 23, respectively, while models from category 1 will have been created independently by running the ‘models\_no\_leftout.cmd’ script found in ‘<project directory>\source\batch\_files’. This is because, while we have very similar models that will have been created in MPS 20, these all contain one subject left-out of the training set (so as to assess this subject on the models to test generalization ability). If we wish to test subject files in production, we want to use models that have been trained on all available training data that we currently have; hence, the ‘models\_no\_leftout.cmd’ script is run to create these models, which constitutes category 1.

**Utilizing the Final Models to Assess New NSAA/NMB Files**

Having chosen the best performing models to generalize towards new files and that have been trained on all available NSAA data we have (along with being supplemented by NMB data where necessary), we were now in a position to create a final Python script that was properly able to utilize these models. While ‘model\_predictor.py’ was written in order to assess a variety of files on different models with different arguments set, the ‘assess\_nsaa\_nmb\_files.py’ script was written to exclusively operate using only several of the models built in total (specifically, a selection of those in ‘rnn\_models\_final’). The script itself calls ‘model\_predictor.py’ with the setting of certain arguments based on user input supplied to ‘assess\_nsaa\_nmb\_files.py’. Additionally, it also does a lot of the preparation of the data that is to be fed through ‘model\_predictor.py’. In other words, ‘assess\_nsaa\_nmb\_file.py’ is a wrapper that sits over many of the other scripts that, based on user supplied input at runtime (as opposed to through arguments), runs the appropriate parts of the data pipeline and loads the models to use from ‘rnn\_models\_final’ to assess the file. For further information about the script and an exact breakdown of how it works, see the section on ‘assess\_nsaa\_nmb\_file.py’ within the ‘Script Ecosystem Overview’ chapter.

While the majority of the Python and associated batch scripts would be used by other users if they were intending to either replicate results outlined in this report or to continue the work already done by the project, the ‘assess\_nsaa\_nmb\_file.py’ script has a slightly different focus in that it’s intended to be able to used as a tool in a production setting. Below, we consider several use-cases of the script where the project directory has been obtained, the Python language (and requisite packages) have been installed to run the script, and the user has one or more source ‘.mat’ files that they wish to analyze:

* Given the ‘.mat’ file data of a brand-new subject (either of natural movement or an NSAA assessment) to the research initiative, one could use the script to give the system’s opinions regarding the subject’s overall and individual NSAA scores (the D/HC will most likely already be known to the user).
  + In the case where it is of an NMB data file, it would allow the user to possibly avoid the requirement of doing an NSAA assessment for the subject if the subject isn’t able or willing to do so but can instead provide them with natural movement data obtained with the suit.
  + In the case where it is an NSAA file, it would be able to be assess the subject who’s file it corresponds to and give a supplementary opinion regarding their scores (e.g. if the assessor is unsure about certain scores or for whatever reason has not provided them).
* Given a new ‘.mat’ file of a new ‘version’ of an existing subject (e.g. the suit data of a subject’s 2nd assessment, 6-months apart from their previous one, at the hospital), it could provide an easy way for the assessor to see how much worse the subject has gotten between their true previous scores (i.e. for their ‘V1’ visit) and what the model predicts as their scores from their new ‘V2’ visit.
* If the user was unsure about the results they achieved either from the system or from their own assessment, they could opt to test the ‘.mat’ file on models trained on an alternative directory to that of the source ‘.mat’ file so as to receive a different opinion from a different set of models.