Reducing the Dimensionality of Sensor Data for the Sit-to-Stand Transfer

03 SitToStand

Mu'izz Asif, Luca Bennett, Joshua Hunt, Vedang Joshi and Isobel Russell

EMAT30005 Mathematical and Data Modelling

Department of Engineering Mathematics, University of Bristol

May 9, 2021

1 Introduction and Background

Assistive technology is any kind of device or system that enables a person to perform a function they struggle with when alone due to disability, restricted mobility or other impairments. Of the people in need of assistive technology, only 1 in 10 receive it [1]. Thus there is a clear motive for development in this sector of healthcare. Assistive technology, while typically used by the elderly, is also widely used by any individuals with difficulties performing tasks such as students with learning difficulties [2]. There are many areas in which assistive technology can improve quality of life for an individual including reading disabilities [3], partial vision loss [4], navigation difficulties [5], and mobility impairments [6].

Although not as fast-growing as that of Japan and Italy, the UK has an ageing population. It is projected that there will be an additional 7.5 million people aged 65 and over in 50 years [7]. A higher proportion of elderly people gives rise to a greater number of age-related disabilities and thus an increased demand for support. Introducing more assistive technology into domestic environments would allow the elderly to be able to live independently for longer. This would be beneficial to the individual and also from a socioeconomic standpoint in that it would reduce pressure on residential homes.

Lack of mobility in particular, be it age-related or not, can have a huge negative impact on a person's quality of life as it can cause further problems with both physical and mental health [8]. Some individuals with mobility issues are unable to work due to their disability and become locked into poverty, while others suffer from loneliness due to their inability to attend social activities.

Before assistive technology can be made more readily available, it must be further developed. The Bristol Robotics Laboratory at the University of the West of England is currently working on assuring safety in assistive robots. They have built a prototype of a robot that would assist people with a disability that prevents them from being able to manoeuvre between a seated and standing position unsupervised. Assisting a person with this specific movement is particularly important to focus on, since it is frequently required in day-to-day life. The robot is suspended from the ceiling and can move in a 2D plane. It consists of two arms with handles for the user to grip which supports the weight of the user during the sit-to-stand transfer. As the user stands the robot moves with them, providing mechanical assistance and reducing the workload on their body. Currently, the robot requires a large system of approximately 50 sensors, including a suit of sensors (Xsens) which the user must wear.

The system of sensors on the participant and the robot generates large quantities of sensor data. This sensor data is comprised of features such as orientation and position. This data is large in scope and difficult to analyse without reducing the dimensionality of the dataset. Some features carry more important information regarding the sit-to-stand transfer than others.

The aim of this work is to identify the features that best represent the entire dataset so that we can reduce the dimensionality whilst retaining the necessary information. Statistical methods are used to group features with similar information content into categories and machine learning (ML) methods are used to score the relative importance of features. We employ two different ML methods: an Extremely Random decision tree model (Embedded ML method) and a Sequential Forward Selection model (Wrapper ML method). We obtain importance scores for all the features for all the trials for data from two participants, and use these importance scores to determine which features to select, for the individual participants.

2 Data Inspection

The dataset contains time series data for 40 trials split equally over two participants ('WAN' and 'PRA'), with each trial consisting of 192 features. In a single trial, participants are recorded as they perform the sit-to-stand transfer leaning on the arms of the robot for support. Throughout the run, the sensors collect data at a frequency of 100Hz; some sensors have many individual features whilst others have just one.

There are 5 different sensor types: balance board, force cell, lidar, linear encoder and Xsens. The balance board has 4 pressure sensors, one on each corner. The force cells are placed on the handles of the machine and the data represents the user pulling and pushing on the grips. lidar sensors measure the distance from the robot to the user. Linear encoders measure the distance moved by the robot. The Xsens suit contains 23 sensors attached to different points on the human body, each of which records their position and orientation in 3D space. This maps out the position of the user as they use the machine and undergo the sit-to-stand transfer. The three figures in Appendix C, taken from an animation of the Xsens data throughout a full trial, clearly display the participant in three distinct stages. We can see this from the change in the angles at vertices 16 and 20 (the knees of the participant).

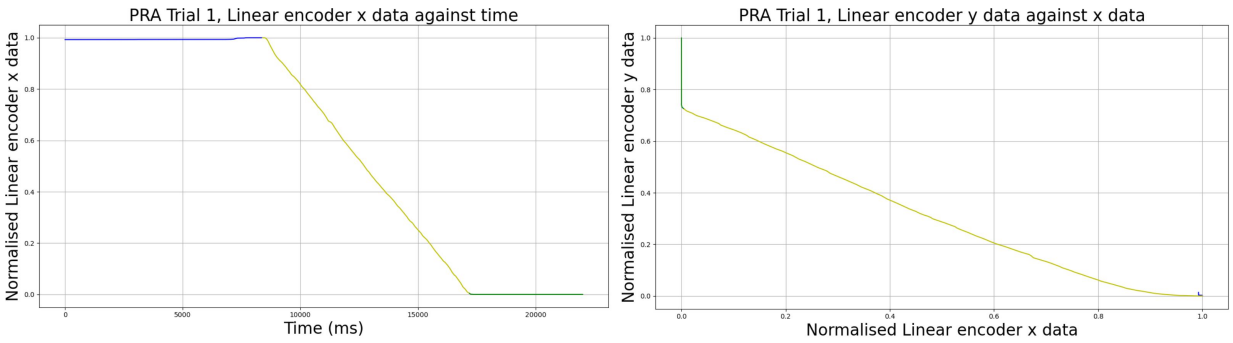
Given the variety of units within the data, we decided to normalise the values to a common scale to enable us to make more direct comparisons. Due to our aim requiring us to compare the different features we concluded the relative values were more important than the magnitudes so this step was justified.

To get an initial understanding of the data, we focused on separating the transfer into distinguishable stages. This allowed us to splice the data into smaller sets across a shorter time frame to isolate the time period when the robot was in motion. These stages are applied to the other sets of sensor data, giving insight into how their dynamics change depending on the motion of the robot.

The linear encoder data represents the location of the robot over time, with (x, y) pairs of coordinates showing its location in the robot's plane of motion. Hence, the velocity of the robot can be determined by the gradient with respect to time of the x and y linear encoder data respectively. It can be seen in Figure 1a that there are three distinct stages in each transfer, with a significant difference in velocity between each stage.

The changes in velocity, which can be seen in Figure 1a as variations in the gradient, are used to determine the stages present in the transfer. The data is first passed through a Savitzky–Golay filter [9]. This smooths the data, increasing the precision without distorting the signal, allowing the analysis to more accurately identify the changes in velocity which define the stages. The acceleration of the robot is calculated, then the locations of significant changes are found, which allows for changes in velocity and hence the stages to be determined. These stages are plotted in different colours in Figure 1, colour coded to the stages shown in Figure 3.

By plotting the linear encoder x data against the y data as seen in Figure 1b, we were able to observe how the robot moves in the plane during the transfer. Figure 1b shows the movement of the robot during the three stages previously identified. Initially the robot is completely stationary, then it moves upwards and away from the user, then finally it moves vertically upwards completing the sit-to-stand transfer. For any given participant and trial, the linear encoder data can be analysed to determine the time-frame of the three distinct stages. Only the second stage is useful when considering the transfer as that is when the velocity is non-zero and the transfer is actually occurring. Hence, the linear encoder data can be used to find the points at which to crop the data for all of the sensors such that they only contain the second stage. By doing this, the sensor



(a) (b)

Figure 1: Plots visualising Linear encoder data for participant PRA, trial 1. The stages of the transfer are colour coded to the stages shown in figure 3.

features which are most meaningful in describing the sit-to-stand transfer can be found, allowing the number of features to be reduced.

3 Feature Reduction

From the visualisations of the time series data, we can qualitatively identify that some lines vary together and assume they carry similar information. This can be exploited to inform feature selection, in that we can group features by how similar they are, under some similarity or distance measure. This is conducted over trials per participant, comparing 190 features to each other (excluding the 2 features from linear encoders). An appropriate threshold for the metric is selected to divide between features that are similar enough to be interchangeable, and those that are not. The specific measures we used are Pearson correlation coefficient [10], Spearman's rank correlation coefficient [11], and the distance extracted from dynamic time warping mappings.

3.1 Statistical Similarity

In order to reduce the number of features necessary to fully describe the transfer, we used two statistical tests (Pearson correlation coefficient and Spearman's rank correlation coefficient) to quantify the statistical dependence of one feature on another. This dependence represents the correlation between the two features. Every feature was tested against every other feature, generating a matrix of scores which measures the dependence of each feature on every other feature.

The assumptions differ between the two statistical tests; Pearson assumes that the two features have a linear correlation whilst Spearmans's assumes a monotonic correlation, whether linear or not [12]. The data given by the features does not hold fully to either assumption, but shares sufficient relationships such that the scores given by the tests can be used for meaningful comparison between features.

Both statistical tests were implemented in Python using the SciPy ecosystem of packages [13]. Both tests produce a score in the range interval $[-1, 1]$, with a score of 0 representing zero correlation between the two features and a score of ± 1 representing a perfect correlation (the sign indicating whether the correlation is positive or negative). As we were solely interested in the magnitude of correlation between features, our scores were modified as follows: $\beta = 1 - |\alpha|$, where β is the

modified score and α is the raw score. Hence, the modified scores fall in the interval $[0, 1]$, where a score of 0 represents a perfect correlation and a score of 1 represents zero correlation between the two features. Hence, our modified score can be considered the 'distance' between two features.

With both statistical metrics, we can attain a matrix of distances between any two features. Only the values below the leading diagonal need to be calculated, due to the distances being reflexive and the values on the diagonal having zero distance - i.e. the distance of a feature to itself. Two features can be identified as being similar enough to be interchangeable if the distance between them falls under some threshold.

Consider all pairs of features which have a distance falling under some threshold, indicated in the matrix. They can be grouped further by those that share a feature, thereby producing partitions of the set of features where every feature only has distance less than the threshold with features within the same set. This is conceptually analogous to a network representation. Each feature can be thought of as a node, and edge weightings are the corresponding distances between any two features. Given some threshold value, we delete the edges with weighting greater than the threshold, creating a number of separated networks. The set of nodes (features) of each unconnected network represents a set of features that can be grouped together, where one could be selected to represent the whole group of features.

Therefore, the number and size of the groupings of features hinge on the selection of an appropriate threshold value. A high threshold value allows for greater distances between features to be designated as interchangeable, producing fewer, larger partitions, and vice versa. We can map different threshold values against the number of partitions they create and then select an optimal threshold value by using the the elbow/knee method as described in [14]. The elbow method finds the elbow points (also known as the knee points) on a graph which are points of significant change. It is used in contexts where there is a trade-off to pick an optimal value for some parameter. The trade-off in this particular scenario is selecting a low enough threshold to capture a high amount of distinctive behaviour (indicated by the number of groups of features), but high enough so a useful amount of feature reduction takes place. A benefit to using this method is that it selects a value that avoids under-fitting and over-fitting, in our case this means it avoids dividing the data into too few and too many groups respectively. The implementation of the 'kneed' package in Python was used.

3.2 Dynamic Time Warping (DTW)

A more tailored approach to time series data can be taken, which accounts for the time component. Different approaches for measuring how the similarity, or conversely the 'distance', between two sets of sequences exist, such as mean similarity and peak similarity [15]. These are often combined with different definitions of distances, such as Euclidean and other L_p-norm measures. Dynamic time warping (as described in [16]) is a particular approach to measure the similarity (and mapping) of two temporal sequences of data, using a many-to-many mapping. A 'distance' measurement can be extracted, which highlights how different two sequences are: robust to noise, shifts and stretches. A key assumption of DTW is that one of the two time series which are being compared is a combination of nonlinear transformations of the other.

To implement and automate the comparison between different features in the dataset, the Python implementation 'dtaidistance' [17] was used for speed and the alignment optimisation purposes of the mappings, so that mappings are not made between large periods of time, rather more locally. As with the statistical approach, the process is repeated to identify groups of similar fea-

tures for feature reduction as well as optimisation of the threshold value using the elbow/knee point.

3.3 Comparison of Statistical Methods and DTW

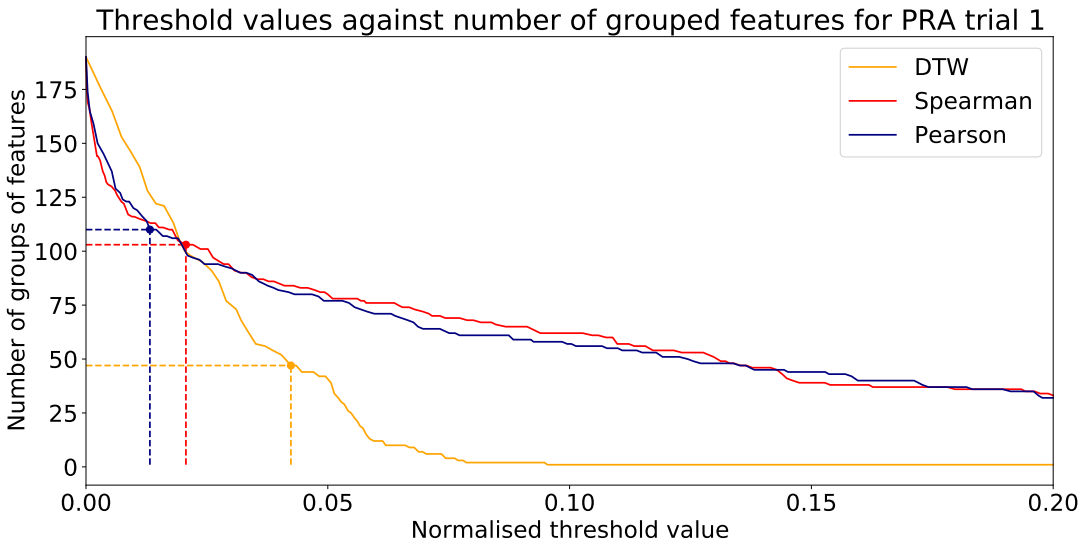


Figure 2: A normalised plot showing the knee point for DTW, Spearmans and Pearsons for participant 'PRA' trial 1. DTW has knee point 1.242 which is normalised to 0.0424 with 47 groups. Spearmans rank has a knee point of 0.021 with 103 groups and Pearsons has a knee point of 0.013 with 110 groups.

This comparison examines the feature reduction applied to the case of participant PRA, trial 1, the results of which have been visualised in the appendix. The results for other participants and trials are largely similar, with some negligible differences.

Taking the case of participant PRA, trial 1, the Pearson and Spearman statistical methods for grouping features are nearly identical in their results as seen in Figure 2. The Spearman method proved slightly more effective in reducing the features, grouping the 190 features into 103 groups compared to the Pearson method's 110 at the knee point threshold value. This result is to be expected as Spearman's assumption of a monotonic correlation [18] between the features is more accurate than Pearson's assumption of a linear correlation [10]. However, the difference of only 7 groups between the two methods can be considered small enough to be negligible. In comparison, DTW performed significantly better than either of the statistical methods, grouping the 190 features into 47 groups at the knee point threshold value.

Considering the breakdown of group membership for all three methods in the case of participant PRA, trial 1, the largest group is comprised solely of Xsens features as seen in Appendix H (Figures 10a, 10b, and 10c). This largest group is significantly larger than all other groups for all three methods, although this difference in size is more pronounced in the statistical methods than in DTW. As with their results in grouping, the sensor breakdown of the groups is largely identical for the two statistical methods, with some slight differences. Whilst the second largest

group for the statistical methods is similar to the first and is only made up of Xsens features, it is significantly different for DTW. It is shown in Figure 10c that the majority of its members are lidar features. This is very different from the two statistical methods who split the lidar features into many smaller groups as seen in Figures 10a and 10b.

The distribution of distances across all features for participant PRA, trial 1 for the Spearman method as seen in Figure 7 is approximately uniform, with the exception of a significant spike in frequency for distances close to 0. Figure 8 shows that the distance distribution for the Pearson method is very similar to the Spearman method with a spike close to 0, followed by a near uniform distribution. However, they diverge for distance values $\gtrapprox 0.7$ with Spearman remaining uniform and Pearson experiencing an increase in frequency. The distance distribution for DTW shown in Figure 9 shares the same spike for distance values close to 0, however, it has a positively skewed unimodal distribution for the remainder of the distance values.

4 Grouped Features: ML Selection Methods

There are many studies in the literature which integrate feature selection methodologies to reduce the complexities of problems and reduce noise in time series data. Maurer *et al* [19] used feature selection techniques (a combination of decision trees, Naive-Bayes classifiers and nearest neighbour classifiers) in a study identifying a user's activity whilst wearing a multi-sensor platform worn on different body positions, to classify 6 different body positions. They obtained a classification accuracy of 87%. In a study modelling human activities using sensor data, Lester *et al* [20] extracted the top 50 informative features from the raw data features. We used several statistical methods to group the features together, we looked at using some wrapper and embedded approaches to select features based on these feature groups.

There are three main approaches to feature selection of which we will be considering an embedded and a wrapper approach. We do not consider filter methods in our investigation as multiple studies show that wrapper methods continuously outperform filter methods [21]. Embedded methods employ learning algorithms of which feature selection is a part, whereas wrapper methods wrap around particular learning algorithms which are used to determine the selected feature subsets to derive the final classifier [22]. We trained both ML models for every trial, and obtained rankings for the entire feature set.

4.1 Embedded ML Approach: Extra-Trees Model

The Extremely Randomised Trees (or Extra-trees) algorithm was developed in order to minimise over-fitting to a random forest model. The Extra-trees model is consistent with the classical top-down approach to building decision trees just like a random forest model. Fundamentally, this approach differs from a random forest model, in the way the tree nodes are split: instead of searching for the most discriminative split thresholds, they are drawn at random for each feature. The best of these chosen thresholds is used to split the sample [23]. The model also differs from a random trees model by using the entire training sample (this minimises the bias) rather than a bootstrap replica (assigned accuracy scores as sample estimates) as implemented in a random forest model [24]. These aspects of the model have motivated its use in our methodology.

We used a Python library (Scikit-learn [25]) implementation of the Extra-trees classifier, to train our model. We used the Gini-impurity (measure of frequency of incorrect labels if a feature is

randomly labelled according to the distribution of labels in the data-set [26]) on our classifier. We also set the minimum number of samples required to split a node to 2. During the training of the model, the classifier performs an implicit feature selection, the outcome of which may be ranked according to relative importance of features, also known as the Gini importance [27]. This feature importance is also calculated by using a Scikit-learn implementation.

The distribution of scores across all 20 trials was consistent for both participants. All the distributions may be considered to be multimodal, with clusters of features with importance scores between 0.07 and 0.08 which can be seen in Figure 4. All distributions contain long trains below scores of 0.06 with all scores having a frequency less than 5.

4.2 Wrapper ML Approach: Sequential Forward Selection Algorithm

Sequential Forward Selection (SFS) is an heuristic search algorithm which starts off with zero selected features. The first feature is selected ($S_1 = f_i$) and given an importance score using a particular criterion function - the most common of which are Mallow's C_p and Akaike's criteria [28]. Then pairs of important features are created with S_1, of which the best one is selected using the same criterion to form $S_2 = \{f_i, f_j\}$. In the next step, sets of three features are formed using S_2 and the best 3 features are selected to form $S_3 = \{f_i, f_j, f_k\}$. This cycle is repeated till there is no improvement to the performance of the previously selected feature [29]. The aim is that the extended feature set minimises the classification error as compared to any new feature selected. SFS is mainly chosen for its fast and simple implementation.

We use the Python library, 'mlxtend' [30], for implementation of the SFS algorithm. This library employs the inbuilt Mallow's C_p criterion function to evaluate performances of features. We use the k-nearest neighbour (kNN) algorithm as the classifier around which the SFS algorithm is wrapped. kNN is a non-parametric learning algorithm primarily used for regression and classification. As kNN classifies data on instances, the entire computation takes place at the classification point meaning it is a very time and computationally effective algorithm of order O(1) [31]. Multiple studies demonstrate the effectiveness of wrapper methods with kNN classifiers [32], when used with time series data analogous to our dataset. In our implementation, the number of neighbours for the k-neighbour queries is 3, with all nodes in the neighbourhood weighted equally. We also employ the standard Euclidean metric, to compute the distances of the nearest neighbours.

Figure 6 shows the distribution of scores across the first 10 trials, for participant 'PRA'. The distribution of scores across all 20 trials was consistent for both participants. All the distributions may be considered to be multimodal, with clusters of features with importance scores between 0.06 and 0.065. The existence of long tails implies the existence of clusters of features with the same or similar importance scores. This is better represented in the histogram in Figure 4.

4.3 Trial-wise Blend of Methods

From the ML models (Extra Trees and SFS) and the different distance measures (DTW, Pearson and Spearman), groups of related features and a ranking on all features is identified, per trial, per participant. These individual trial results can be amalgamated with a scoring scheme to highlight the most important features across all trials per participant, combining the relevant trials' grouping of features and ML ranking.

For each trial, per participant, the rankings can be used to select the most important feature for each group. The highest ranking feature from each group can be selected to represent the

whole group. This set of selected features can be ordered again based on the same rankings, in descending order. For each trial, the most important selected feature, is given a score based on its position (indices starting at 0, with the best selected feature having a zero score). The scores for the different selected features are summed across trials, per participant with the feature with the lowest total score indicating it is the most important feature across the trials. The approach of the most important feature in a set of selected features having the lowest score rather than the highest, aims to prevent skewed results in trials where more groups of features were identified. This is conducted for each configuration of distance measure and ML model, from which the top n features can be selected for comparison between participants. If a feature is never selected to represent a group of features, it is not considered amongst the top features (despite having a score of 0).

5 Results & Discussion

Table 1 gives an example of the results obtained from the amalgamation of the feature reduction and selection techniques. For the other results please refer to Appendix F.

Table 1: Top 10 features for PRA, WAN, using DTW, SFS

PRA		WAN	
Sensor	Score	Sensor	Score
Lidar_field_ranges3	4	Xsens_field_poses3_position_z	11
Lidar_field_ranges351	8	Lidar_field_ranges356	13
Xsens_field_poses3_position_z	11	Lidar_field_ranges354	16
Lidar_field_ranges356	12	Lidar_field_ranges353	20
Lidar_field_ranges354	16	Lidar_field_ranges351	21
Lidar_field_ranges353	18	Lidar_field_ranges355	24
Lidar_field_ranges1	19	Lidar_field_ranges1	26
Lidar_field_ranges349	20	Lidar_field_ranges352	28
Lidar_field_ranges350	20	Lidar_field_ranges0	28
Lidar_field_ranges5	20	Lidar_field_ranges359	28

The results show that the Xsens sensors consistently provided the most meaningful features. The only exception to this was the DTW measure combined with the SFS model which mostly recommends the lidar sensors. It is logical that the Xsens data is likely to be the most meaningful in indicating the stages of the transfer as they make up 84% of the features. This is despite the groupings aiming to mitigate some of this bias, by selecting only one feature from a group of similar features, of which the largest groups are mostly composed of Xsens features (as indicated in Appendix H).

The two statistical methods both produce very similar results, though there is some difference depending on the ML approach used. Comparing Table 3 with 4 (SFS) and Table 6 with 7 (Extra Trees) shows that using the same ML approach produces the same list of top 10 features, with some slight differences in score and ordering. Though the methods differ when the ML approach is changed, this can be solely attributed to differences in the ML models. For both statistical methods, all of the top 10 features were Xsens features, as the breakdown of group composition by

sensor in Figures 10a and 10b would suggest, with Xsens features dominating the largest group in both cases.

More specifically, from the tables of the top 10 features in each combination of methods (shown in Appendix F) we see that often the shoulder and shoulder blade sensors on the right side are favoured by the Extra trees model. With the SFS method however, the height of the third spinal sensor proved to be very significant.

With such a small sample size of 2 participants, it is difficult to say whether this would be reproducible across all new users and more trials and so further trials would have to be conducted with a greater number of participants to test this.

The knee point selection method relies on a distinct change in the derivative. Applying this to our data results in Figure 2. Our model heavily relies on the selected threshold value where a small change in threshold can cause a large change in the number of groups. Exploring alternative optimisation approaches such as applying a smoothing filter to the threshold graph before applying the knee point method may drastically change the result. This would reduce the sensitivity to noise in the data and could provide a more stable result.

The results support the need to select a knee point for feature reduction. It is conducted on a per trial basis and informs the feature groupings yet produces homogeneity in the results across participants. It appears to optimise each trial to the same, appropriate level. Other schemes could be designed to provide different levels of aggression in the feature reduction.

It is clear there is much more variation in the rankings between ML models than within individual ML models. We can infer from this that the ML models contribute more to differences in the top 10 features than the two statistical models and DTW. SFS highlights the differences between participants more than Extra trees.

Our results were often consistent between the two participants suggesting that the importance of the features is not dependant on the participant. This is especially apparent in Tables 4 and 5 (as shown in Appendix F) where the features and ordering are identical.

The main limitation of SFS is that it is unable to remove features that become redundant after the addition of other features. SFS assumes that a single interval model is used consistently throughout. When considering specific feature performance, at the beginning, it does not consider combinations of features as a starting point. This means it may not result in the optimal solution.

On the other hand, the results for DTW (Tables 2 and 5) differ from the statistical methods. When using the Extra Trees ML approach with DTW, it produces a similar result to the statistical methods, with the top 10 list differing by only a small number of features and being solely comprised of Xsens features. However, when the SFS ML approach is used, the results (Table 2) drastically differ from both the statistical methods and the other ML approach combined with DTW. The top 10 features are instead mostly composed of lidar features. This is supported by the breakdown of group composition by feature for DTW in Figure 10c in which the second largest group is mostly made up of lidar features.

The methodology to merge the results of feature reduction and selection appears to yield consistent and plausible results. It aims to select the best representation of each distinctive behaviour captured by groupings on a per trial basis. This is through selecting the highest ranking feature within the groupings of features per trial, and then ranking those selected features by the same ranking. A similar approach could build on this idea to be more robust to new ML models and different distance measures.

Looking at the score values within the top 10 features tables in Appendix F, we can see that

overall the Extra trees model produces much lower scores than the SFS model. This could have two inferences based on the frequency of the low scoring features being initially selected from the groupings. For features that are selected in only a few trials, their scores may improperly represent their importance. This is because no penalties are applied for importantly ranked features with low frequency appearances in the selected features. Conversely, if the feature was ranked many times and still maintained a low score, the value will be more meaningful and so we can be more confident of its accuracy. To quantify the significance of low scoring features, a valuable development of this method would be to track the number of times a feature is selected across trials, as well as its ranking when it appears.

It is also vital to note that the rankings between tables are not directly comparable. For example, two features that have both been ranked first in importance may not have equal significance with regards to the sit-to-stand transfer.

We assumed that the robot moves in an entirely synchronised fashion with the participant throughout the transfer, our method relies heavily on the linear encoder data. This data was the basis of our decision to crop the features.

The application of the methods that we have designed could extend to prediction of the transfer. This could be implemented by accounting for the data prior to the transfer to identify the point at which the transfer occurs. The process involving feature reduction and selection then would identify key features that include some description of the transition from sitting to the transfer.

6 Conclusion

We have developed a suite of methods to evaluate feature importance on a trial and participant-wise basis. We introduced a new method to select distinctive behaviours assessed during the grouping process on a trial-wise basis. The Xsens suit proved to be most fruitful in terms of picking out features across different trials and participants. However, the SFS-DTW combination resulted in the lidar sensors being ranked as more important than other sensors in the dataset. The ML methods give slightly contrasting results: the Extra-trees model highly ranks the right shoulder and the shoulder blade sensors as compared to the SFS model, where the height of the third spinal sensor was favoured. We discovered that the ML methods contribute more to variations in the rankings for the top features than the statistical models and DTW. In the ML models, Extra Trees performed better than SFS in terms of differences in feature rankings. The homogeneity between features selected across participants validates our use of the knee point for feature reduction. More general results are hard to draw with only 2 participants and feature difference between 2 participants are of limited use. Our method is also limited by the lack of penalties for importantly ranked features, with low frequency appearances in the features selected per trial. A similar methodology could be applied to data prior to the transfer. It would enable extended functionality that could identify features which predict the sit-to-stand transfer.

References

[1] World Health Organization et al. Improving access to assistive technology: report by the secretariat. *Geneva: WHO*, 2016.

[2] Karen E. Forgrave. Assistive technology: Empowering students with learning disabilities. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*, 75(3):122–126, January 2002.

[3] Ingvar Lundberg. The computer as a tool of remediation in the education of students with reading disabilities—a theory-based approach. *Learning Disability Quarterly*, 18(2):89–99, 1995.

[4] Ramiro Velázquez. Wearable assistive devices for the blind. In *Wearable and autonomous biomedical devices and systems for smart environment*, pages 331–349. Springer, 2010.

[5] AM Howard Freiberger. Mobility aids for the blind. *Bulletin of prosthetics research*, 22:73, 1974.

[6] Uta R Roentgen, Gert Jan Gelderblom, Mathijs Soede, and Luc P De Witte. Inventory of electronic mobility aids for persons with visual impairments: A literature review. *Journal of Visual Impairment & Blindness*, 102(11):702–724, 2008.

[7] Sarah Coates, Priya Tanna, and Eleanor Scott-Allen. Overview of the uk population: August 2019. *Office for National Statistics: Newport, UK*, 2019.

[8] Shirley Musich, Shaohung S. Wang, Joann Ruiz, Kevin Hawkins, and Ellen Wicker. The impact of mobility limitations on health outcomes among older adults. *Geriatric Nursing*, 39(2):162–169, March 2018.

[9] Yanping Liu, Bo Dang, Yue Li, Hongbo Lin, and Haitao Ma. Applications of savitzky-golay filter for seismic random noise reduction. *Acta Geophysica*, 64(1):101–124, feb 2016.

[10] TR Derrick, BT Bates, and JS Dufek. Evaluation of time-series data sets using the pearson product-moment correlation coefficient. *Medicine and science in sports and exercise*, 26(7):919—928, July 1994.

[11] Spearman rank correlation coefficient. In *The Concise Encyclopedia of Statistics*, pages 502–505. Springer New York.

[12] Anthony J. Bishara and James B. Hittner. Testing the significance of a correlation with nonnormal data: Comparison of pearson, spearman, transformation, and resampling approaches. *Psychological Methods*, 17(3):399–417, sep 2012.

[13] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa,

Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[14] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pages 166–171. IEEE, 2011.

[15] Carmelo Cassisi, Placido Montalto, Marco Aliotta, Andrea Cannata, and Alfredo Pulvirenti. Similarity measures and dimensionality reduction techniques for time series data mining. *Advances in data mining knowledge discovery and applications'(InTech, Rijeka, Croatia, 2012,*, pages 71–96, 2012.

[16] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:, 1994.

[17] Wannes Meert, Kilian Hendrickx, and Toon Van Craenendonck. wannesm/dtaidistance v2.0.0, August 2020.

[18] Thomas D. Gauthier. Detecting trends using spearman's rank correlation coefficient. *Environmental Forensics*, 2(4):359–362, 2001.

[19] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, pages 4 pp.–116, 2006.

[20] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *IJCAI*, volume 5. Citeseer, 2005.

[21] George H John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Elsevier, 1994.

[22] X. Chen and J. C. Jeong. Enhanced recursive feature elimination. In *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pages 429–435, 2007.

[23] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[24] L Breiman. Arcing classifiers (technical report). *University of California, Department of Statistics*, 1996.

[25] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[26] Ting Zhi, Hongbin Luo, and Ying Liu. A gini impurity-based interest flooding attack defence mechanism in ndn. *IEEE Communications Letters*, 22(3):538–541, 2018.

[27] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[28] Beata Walczak. *Wavelets in chemistry*. Elsevier, 2000.

[29] David W. Aha and Richard L. Bankert. A comparative evaluation of sequential feature selection algorithms. In *Learning from Data*, pages 199–206. Springer New York, 1996.

[30] Sebastian Raschka. Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack. *The Journal of Open Source Software*, 3(24), April 2018.

[31] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[32] Aiguo Wang, Ning An, Guilin Chen, Lian Li, and Gil Alterovitz. Accelerating wrapper-based feature selection with k-nearest-neighbor. *Knowledge-Based Systems*, 83:81–91, July 2015.

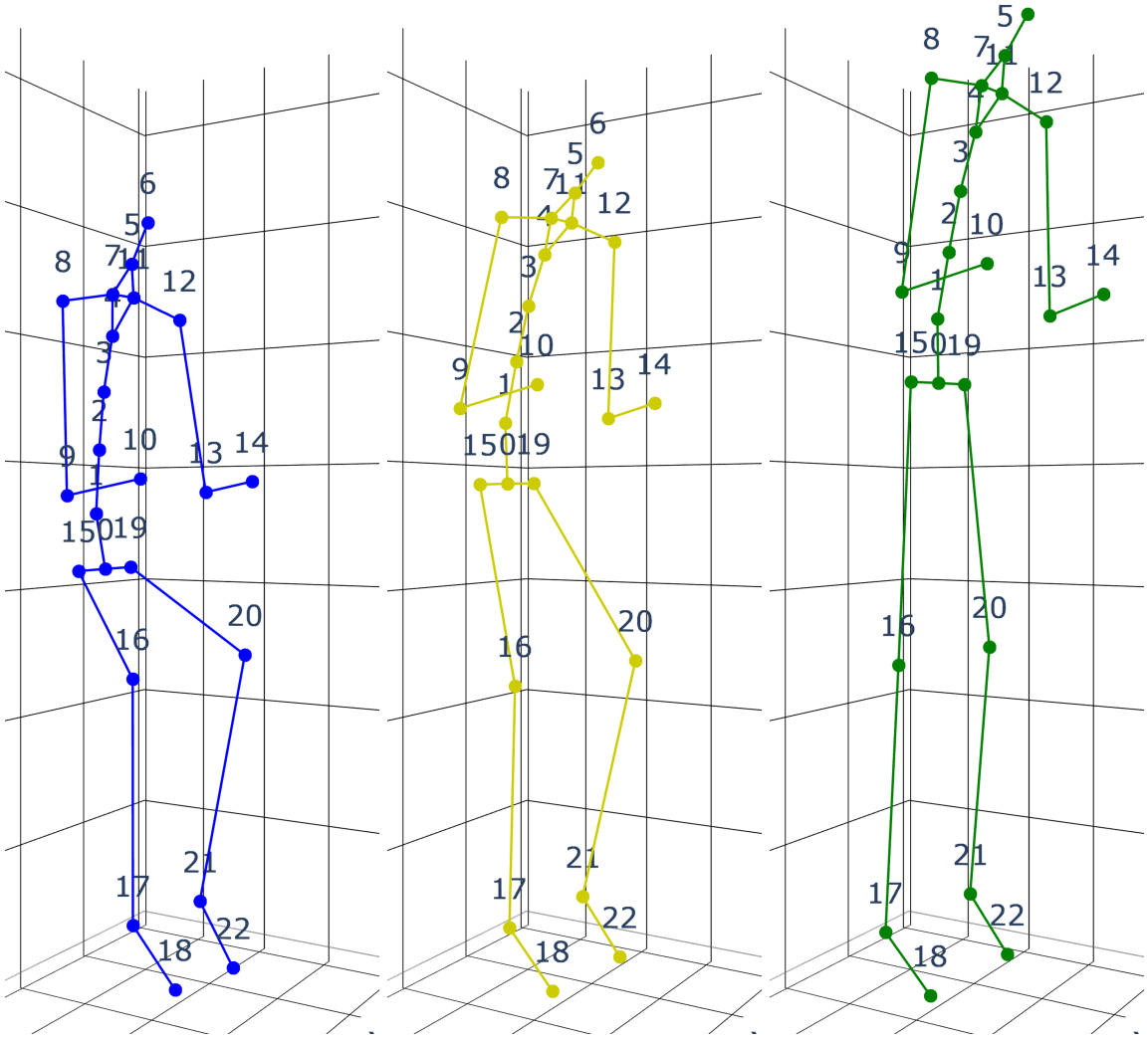
A EDI statement

We quickly split our problem into several sub-problems, which were able to be tackled by 1 or 2 people. This meant the workload was distributed equitably from the beginning. Naturally some of the approaches weren't as successful as others however we quickly re-distributed ourselves based on what we felt were our strong areas. This adaptability ultimately lead to the successful completion of our project.

B COVID mitigation statement

We were unable to meet in person with some members not returning to Bristol after the Christmas period, although we had no problem collaborating using online platforms. We used Facebook messenger for time sensitive communication as well as Microsoft Teams for group meetings. This meant that we were able to effectively keep in contact. We had some issue with internet speeds during online meetings, although it was quickly and easily overcome and as such it didn't have any adverse effects on our progress. Overall we successfully mitigated the disruption due to COVID.

C Stickman



(a) Xsens sensors while sitting before the sit to stand transfer

(b) Xsens sensors during the sit to stand transfer

(c) Xsens sensors while standing after the sit to stand transfer

Figure 3: A Figure showing the Xsens sensor data for participant WAN, trial 1 visualised in 3D space. The joints are labelled according to their number in the dataset.

D Extra Trees Histogram

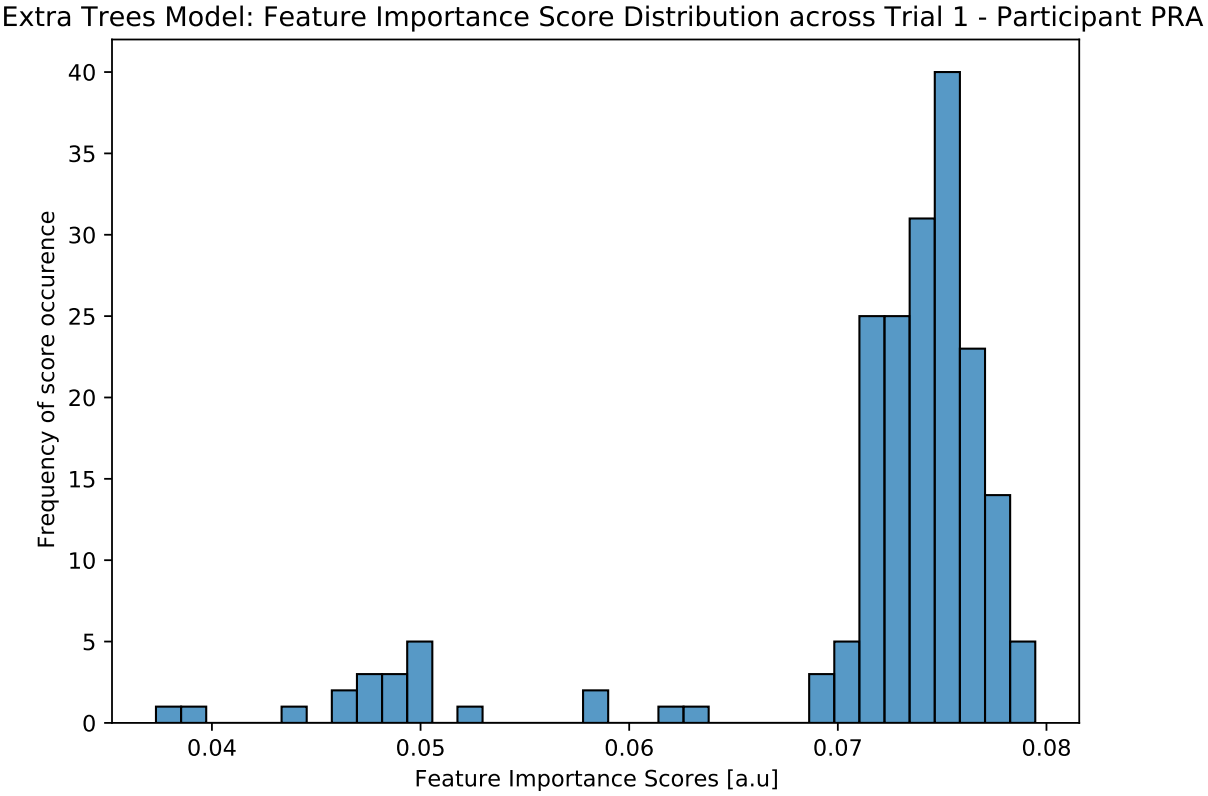


Figure 4: A histogram of the feature importance scores for the Extra Trees model for trial 1 for participant 'PRA'. The horizontal axis shows the feature importance scores in arbitrary units.

E Violin Plots

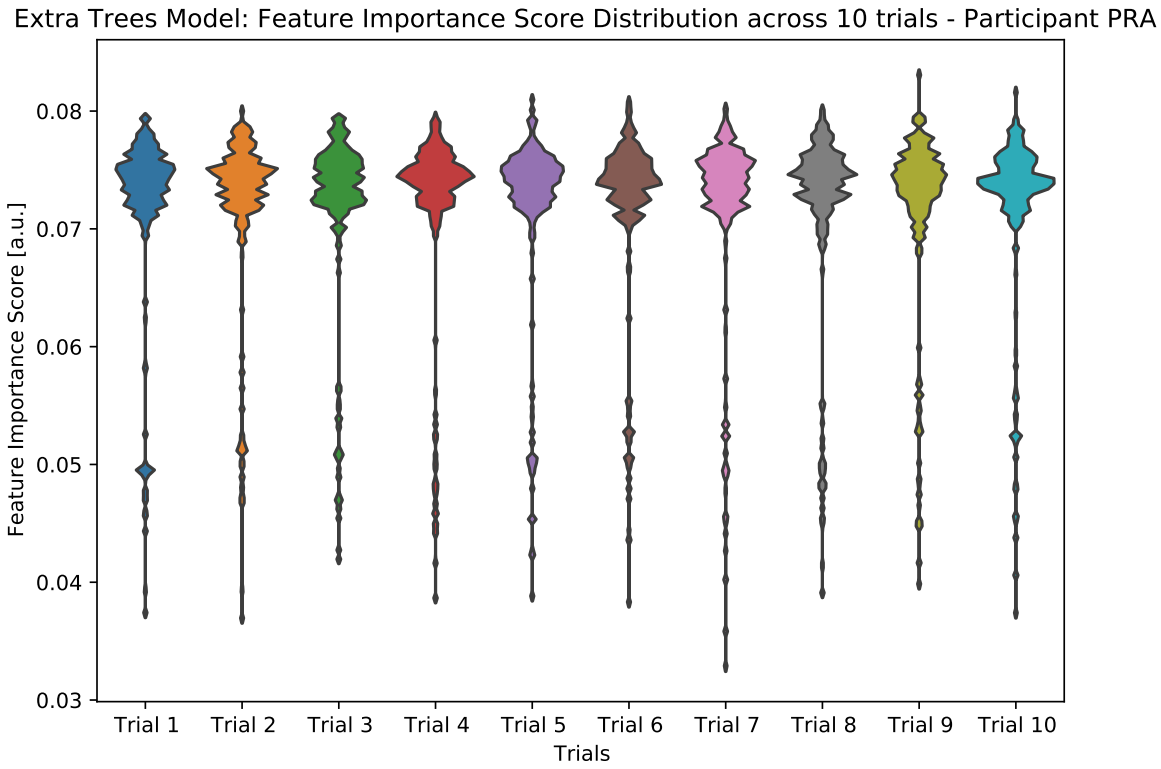


Figure 5: A violin plot of the feature importance scores for the Extra Trees model (participant 'PRA') for 10 trials. The horizontal axis shows the trial numbers. The vertical axis shows the distribution of scores across the trial and may be compared to Figure 4 for similarities. The feature importance scores have arbitrary units. Smoothing bandwidth = 0.02

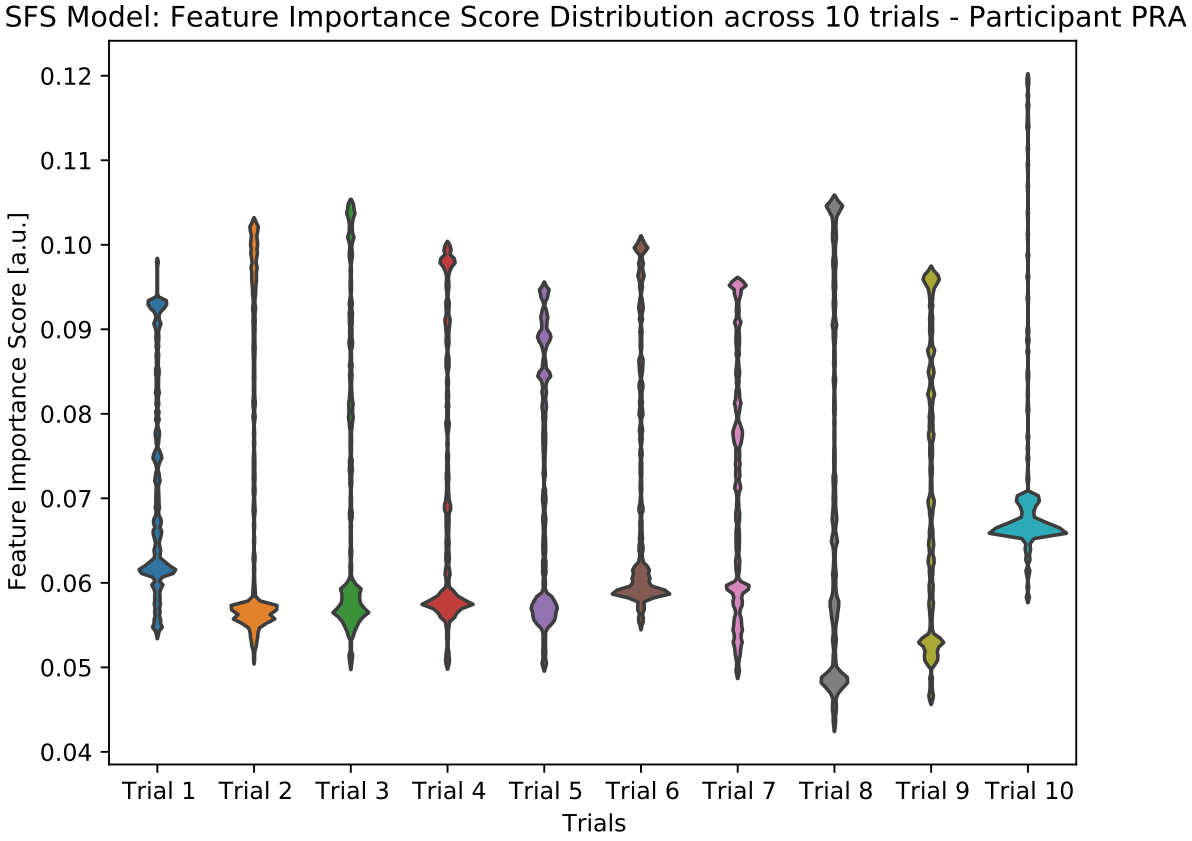


Figure 6: A violin plot of the feature importance scores for the SFS model (participant 'PRA') for 10 trials. The horizontal axis shows the trial numbers. The vertical axis shows the distribution of scores across the trial and may be compared to Figure 4 for similarities. The feature importance scores have arbitrary units. Smoothing bandwidth = 0.02

F Result Tables

Table 2: Top 10 features for PRA, WAN, using DTW, SFS - shown in section 5 as Table 1

PRA		WAN	
Sensor	Score	Sensor	Score
Lidar_field_ranges3	4	Xsens_field_poses3_position_z	11
Lidar_field_ranges351	8	Lidar_field_ranges356	13
Xsens_field_poses3_position_z	11	Lidar_field_ranges354	16
Lidar_field_ranges356	12	Lidar_field_ranges353	20
Lidar_field_ranges354	16	Lidar_field_ranges351	21
Lidar_field_ranges353	18	Lidar_field_ranges355	24
Lidar_field_ranges1	19	Lidar_field_ranges1	26
Lidar_field_ranges349	20	Lidar_field_ranges352	28
Lidar_field_ranges350	20	Lidar_field_ranges0	28
Lidar_field_ranges5	20	Lidar_field_ranges359	28

Table 3: Top 10 features for PRA, WAN, using Spearman, SFS

PRA		WAN	
Sensor	Score	Sensor	Score
Xsens_field_poses3_position_z	11	Xsens_field_poses3_position_z	11
Xsens_field_poses4_orientation_w	32	Xsens_field_poses1_position_z	37
Xsens_field_poses1_position_y	36	Xsens_field_poses6_position_y	40
Xsens_field_poses1_position_z	37	Xsens_field_poses10_position_y	42
Xsens_field_poses6_position_y	40	Xsens_field_poses15_orientation_z	42
Xsens_field_poses10_position_y	42	Xsens_field_poses2_position_z	43
Xsens_field_poses15_orientation_z	42	Xsens_field_poses1_position_y	65
Xsens_field_poses2_position_z	43	Xsens_field_poses4_position_y	91
Xsens_field_poses4_position_y	91	Xsens_field_poses12_position_z	95
Xsens_field_poses12_position_z	95	Xsens_field_poses2_position_y	121

Table 4: Top 10 features for PRA, WAN, using Pearson, SFS

PRA		WAN	
Sensor	Score	Sensor	Score
Xsens_field_poses3_position_z	11	Xsens_field_poses1_position_z	37
Xsens_field_poses1_position_z	37	Xsens_field_poses6_position_y	40
Xsens_field_poses6_position_y	40	Xsens_field_poses10_position_y	42
Xsens_field_poses10_position_y	42	Xsens_field_poses15_orientation_z	42
Xsens_field_poses15_orientation_z	42	Xsens_field_poses2_position_z	43
Xsens_field_poses2_position_z	43	Xsens_field_poses3_position_z	44
Xsens_field_poses1_position_y	65	Xsens_field_poses4_position_y	91
Xsens_field_poses4_position_y	91	Xsens_field_poses1_position_y	94
Xsens_field_poses12_position_z	95	Xsens_field_poses12_position_z	95
Xsens_field_poses2_position_y	121	Xsens_field_poses2_position_y	121

Table 5: Top 10 features for PRA, WAN, using DTW, Extra Trees

PRA		WAN	
Sensor	Score	Sensor	Score
Xsens_field_poses14_position_z	0	Xsens_field_poses14_position_z	0
Xsens_field_poses7_position_y	0	Xsens_field_poses7_position_y	0
Xsens_field_poses8_position_z	1	Xsens_field_poses8_position_z	1
Xsens_field_poses4_position_z	2	Xsens_field_poses4_position_z	2
Xsens_field_poses13_position_z	3	Xsens_field_poses13_position_z	3
Xsens_field_poses12_position_y	3	Xsens_field_poses12_position_y	3
Xsens_field_poses19_orientation_z	4	Xsens_field_poses19_orientation_z	4
Xsens_field_poses11_position_y	4	Xsens_field_poses11_position_y	4
Xsens_field_poses2_position_y	5	Xsens_field_poses2_position_y	5
Xsens_field_poses11_position_x	7	Xsens_field_poses8_position_y	6

Table 6: Top 10 features for PRA, WAN, using Spearman, Extra Trees

PRA		WAN	
Sensor	Score	Sensor	Score
Xsens_field_poses7_position_y	0	Xsens_field_poses7_position_y	0
Xsens_field_poses8_position_z	1	Xsens_field_poses8_position_z	1
Xsens_field_poses14_position_z	2	Xsens_field_poses14_position_z	2
Xsens_field_poses4_position_z	2	Xsens_field_poses4_position_z	2
Xsens_field_poses19_position_y	2	Xsens_field_poses19_position_y	2
Xsens_field_poses15_position_z	3	Xsens_field_poses15_position_z	3
Xsens_field_poses19_orientation_z	4	Xsens_field_poses19_orientation_z	4
Xsens_field_poses13_position_z	6	Xsens_field_poses13_position_z	6
Xsens_field_poses12_position_y	6	Xsens_field_poses12_position_y	6
Xsens_field_poses8_position_y	6	Xsens_field_poses8_position_y	13

Table 7: Top 10 features for PRA, WAN, using Pearson, Extra Trees

PRA		WAN	
Sensor	Score	Sensor	Score
Xsens_field_poses7_position_y	0	Xsens_field_poses7_position_y	0
Xsens_field_poses8_position_z	1	Xsens_field_poses8_position_z	1
Xsens_field_poses4_position_z	2	Xsens_field_poses4_position_z	2
Xsens_field_poses14_position_z	4	Xsens_field_poses14_position_z	4
Xsens_field_poses19_orientation_z	4	Xsens_field_poses19_orientation_z	4
Xsens_field_poses19_position_y	4	Xsens_field_poses15_position_z	6
Xsens_field_poses15_position_z	6	Xsens_field_poses12_position_y	9
Xsens_field_poses13_position_z	9	Xsens_field_poses19_position_y	10
Xsens_field_poses12_position_y	9	Xsens_field_poses13_position_z	12
Xsens_field_poses8_position_y	13	Xsens_field_poses11_position_z	12

G Distance Distribution

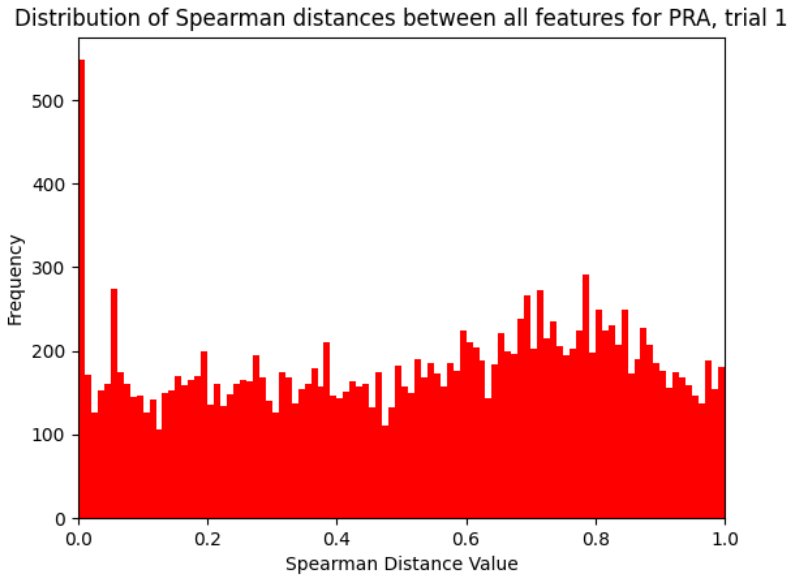


Figure 7: A plot showing the distribution of distances between all features for the Spearman statistical method for participant PRA, trial 1.

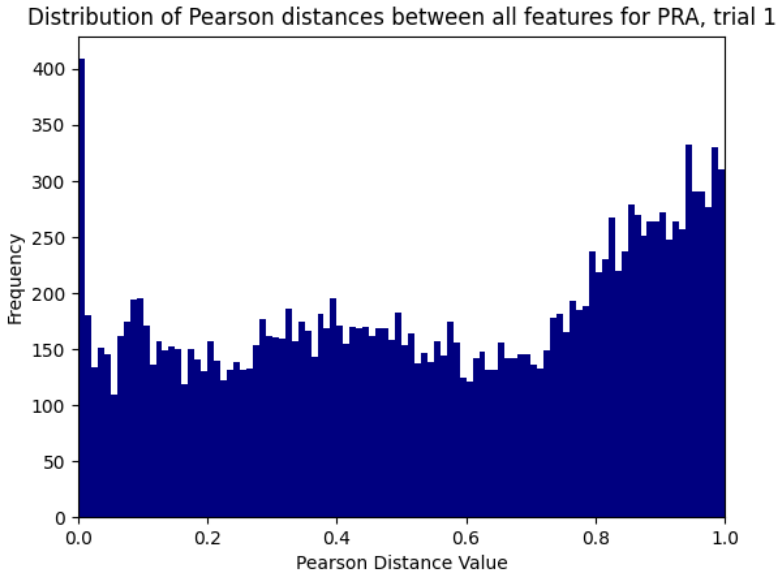


Figure 8: A plot showing the distribution of distances between all features for the Pearson statistical method for participant PRA, trial 1.

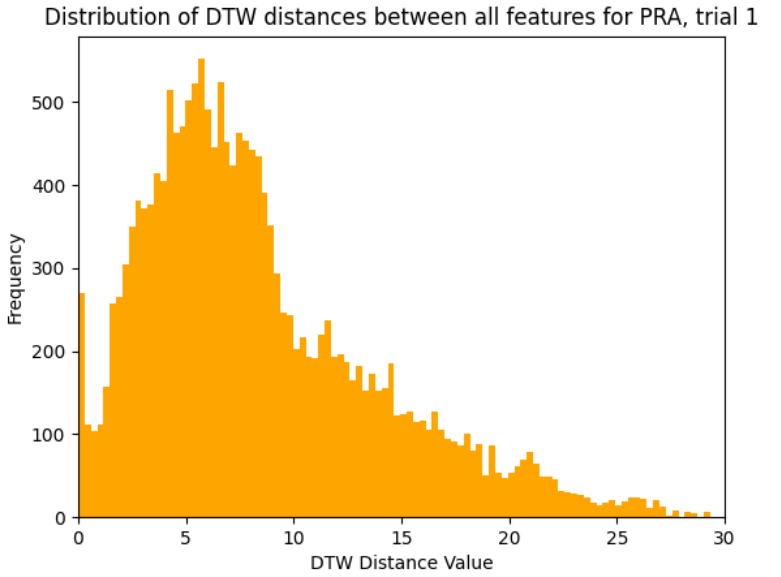
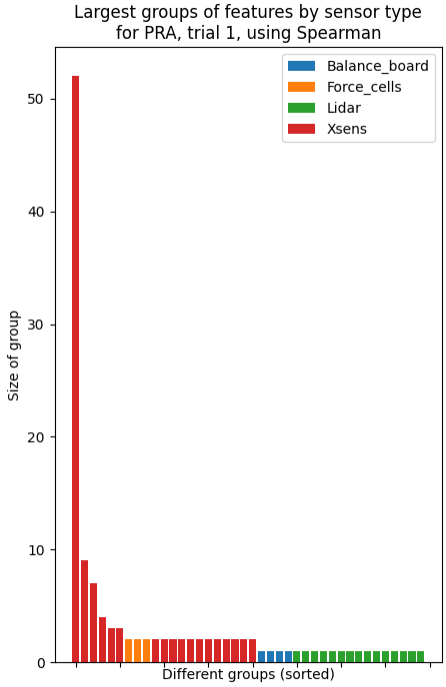
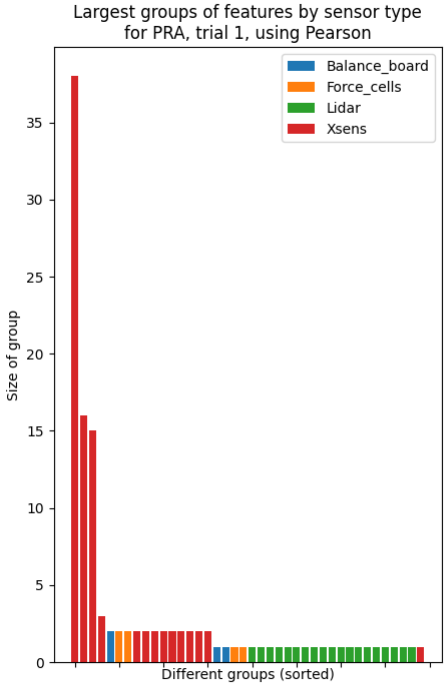


Figure 9: A plot showing the distribution of distances between all features for the DTW method for participant PRA, trial 1.

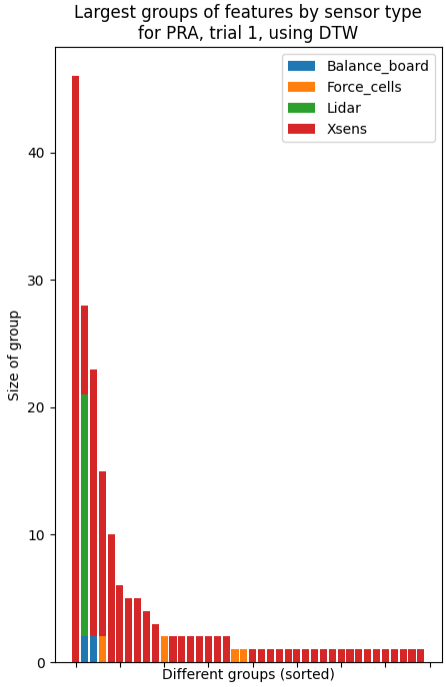
H Sensor Breakdowns



(a)



(b)



(c)

Figure 10: Plots showing the composition of sensor type per group for the top 40 groups in PRA trial 1 for Spearman (a), Pearson (b) and DTW (c).