CSE 572: DATA MINING

Assignment 2
Spring 2018

GROUP 2

Dwarsala Venkata Manoj Reddy	1213185010
Dipika Ramaswamy	1213237517
Kiran Teja Settipalli	1213019858
Rohit Polisetti	1212933824
Srinivas Puranam	1211168267

Table of Contents

1. Phase 1:

Data was collected in this part. Every person had performed certain signs that are used in the American Sign Language to create data sets. Data was collected by using sensors which have 28 data streams in total, which are – 3 from accelerometer, 4 from gyroscope, 3 from orientation and 8 from EMG sensors. Some of the signs that we have performed are ABOUT, AND, CAN, COP, DEAF, DECIDE, FATHER, FIND, GO OUT, HEARING, etc.

2. Phase 2:

2.1 Task 1:

We have initially used data from phase 1 collected by different groups. They are arranged in a set of folders called data sets. The data collected from different teams in phase-1 is arranged in different folders called data sets. Each data set contains files of time-series data for each gesture. The time series of each action are stored column wise. In task1 of phase-2, we need to select a set of datasets, read through all the files and produce 10 different excel files for the given 10 gestures.

The matlab code for task1 can be found in 'task1.m' file.

Here, we have selected 5 data-sets. For each gesture, we loop through the data sets one at a time and read the time-series data of each gesture in that particular data set. Then the time-series data of the gesture is stored in a matrix and copied the matrix data into the separate excel file for each gesture. Lastly each output is stored in an output folder with each file name starting with "output_". The following is the snippet of how the data looks for each gesture after task-1 has been performed.

About ALX 0.32 0.23 0.243 0.242 0.344 ……..

About ALY 0.31 0.31 0.254 0.282 0.324 ……..

|

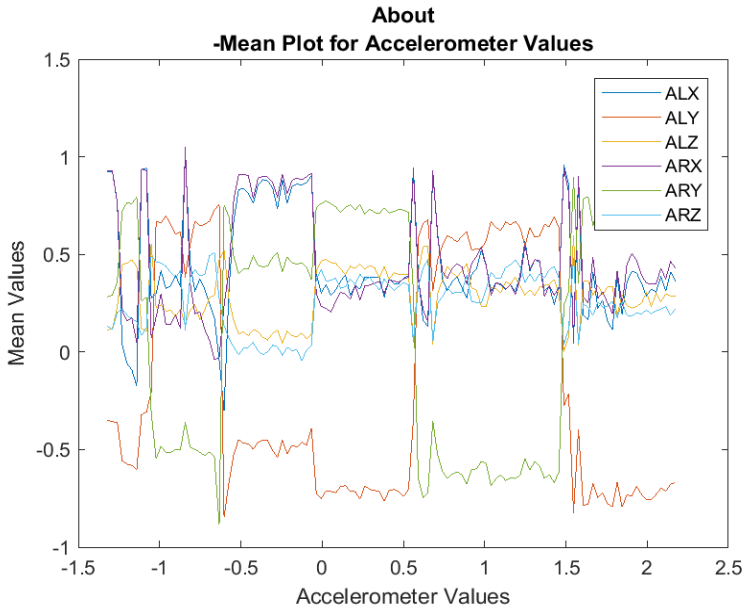
2.2 Task 2: Feature Extraction

In this task, the excel files for different actions that have been generated using task-1 are used. These files are used to construct the data matrix for each action. Then, a set of feature extraction methods are applied on the data matrix that best differentiates all the actions.

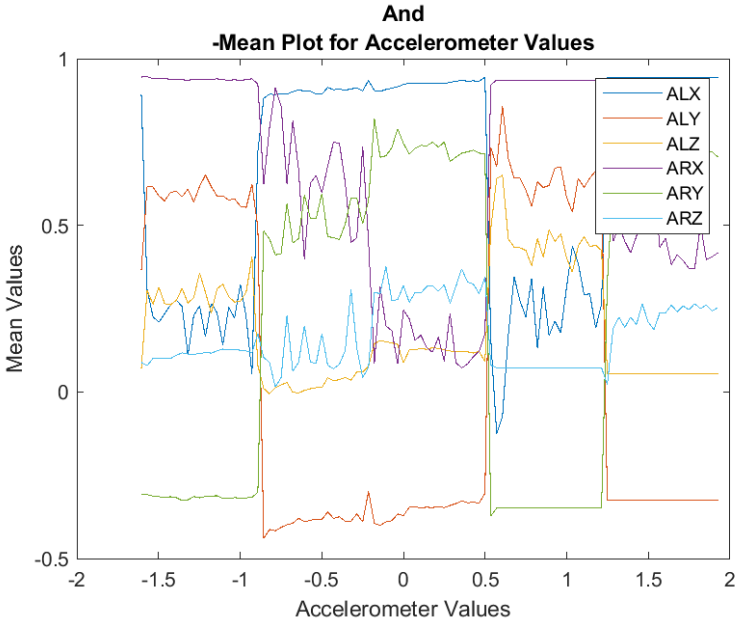
Following feature extraction methods are used:
1. Mean
2. Fast Fourier Transform
3. Range
4. Standard Deviation
5. Root Mean Square

The extracted features are presented on a graph to visualize the distinction among all the actions. We also, overlapped features for a set of similar actions and presented on a graph for better understanding of patterns among the features. In this document, for each feature extraction method, we first discussed about how the feature extraction is applied for each action. We then explained the reason for selecting such feature extraction method. Then mentioned the path for the Matlab code. After that, discussed about the plotted features on the graph for each action. Then, explained whether the initial intuition about the features holds true or not.

2.2.1 Mean:



Mean plot for 'About'



Mean plot for 'And'

a) Mean is a statistical measure that gives the average of a set of data points in a collection. For each of the ten gestures, we have selected their respective accelerometer sensor values as features to analyze how the mean values sensors vary across all the gestures. Since mean indicates the central value at every step of every gesture, it needs to progress in a straight-forward manner, with peaks and dips in the curves denoting sharp hand movements in the corresponding axes. As we are plotting left x,y,z and right x,y,z accelerometer sensors separately, we can also observe the correlation between every left and its right counterpart .

b) Mean, being a measure of centrality in a dataset can help us observe the nature of hand movements demanded by every gesture. The time-series plot of the mean values can help us gauge whether certain gestures are similar in their acting patterns and if the movements consist of sudden turns resulting in sharp spikes or falls in the graph.

c) Matlab code has been attached in the folder as “Task2MeanFFTRange.m”.

d) The function ‘nanmean’ was used in Matlab to calculate the mean of the six accelerometer sensor values ALX, ALY, ALZ, ARX, ARY and ARZ without taking into account all ‘Not a Number’ (Nan) values in the dataset. The screenshots of ‘nanmean’ function on all the ten gestures are provided in the folder ‘Mean’ in the submission. The X-axis contains labels in the range covered by the minimum and the maximum accelerometer values in the dataset while the Y-axis contains the results of ‘mean’ on the six sensor values.

The plots of the other gestures are placed in the folder ‘Mean’ under ‘Plots’.

e) Observing the graphs for the mean of the ten gestures and seeing obvious results in the patterns generated, we can say that our initial intuition to analyze the data using ‘mean’ as a measure proved insightful. In each of the ten graphs, it is evident that every left and right axis’ curve moves either in synchronization or asynchronization with each other since the left and right curves for each axis occurs either parallel to each other or as mirror images to each other. The accelerometer values range from -3 to +5 on the X-axis on the whole whereas the Y-axis mean ranges from -0.5 to 1.5. The X-axes, left and right, regularly have higher mean values as they are usually seen on the top half of the graph while the Z-axes means stay constant in the middle of the graph. These trends justify the selection of mean as a feature selection method.

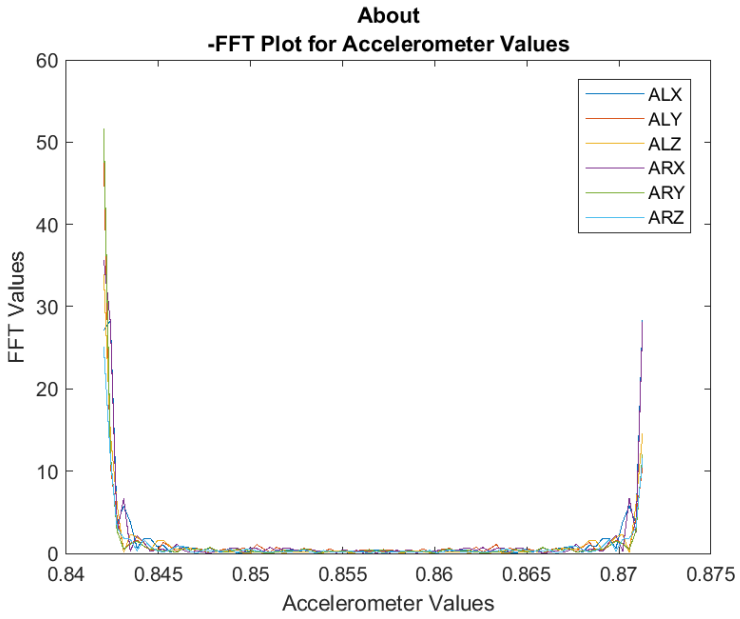
2.2.2 Fast Fourier Transform:

a) Fast Fourier Transform is an algorithm that deconstructs a time-domain signal into its frequency components. We have used the ‘fft’ function from Matlab to perform FFT on each of the six accelerometer sensors ALX, ALY, ALZ, ARX, ARY, ARZ. From the peaks and dips in the curves plotted, we can understand about the nature of actions along specific axes.

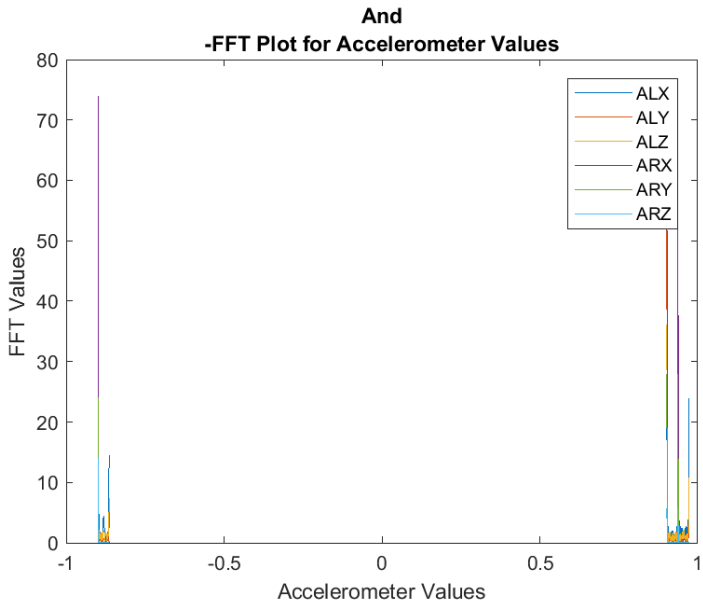
b) FFT as a feature extraction method can help us compare the ten gestures and observe deviations in the frequency domain. This is why we proceeded with FFT as a feature extraction method.

c) Matlab code has been attached in the folder as “Task2MeanFFTRange.m”.

d) The function ‘fft’ was used in Matlab to calculate the fast fourier transform of every row in the dataset. This was done taking the six accelerometer sensors ALX, ALY, ALZ, ARX, ARY and ARZ as features. The screenshots of ‘fft’ on all the ten gestures are provided in the folder ‘FFT’ in the submission. The X-axis contains labels in the range covered by the minimum and the maximum accelerometer values in the dataset while the Y-axis contains the results of ‘fft’ on the six sensor values.



FFT plot for ‘About’

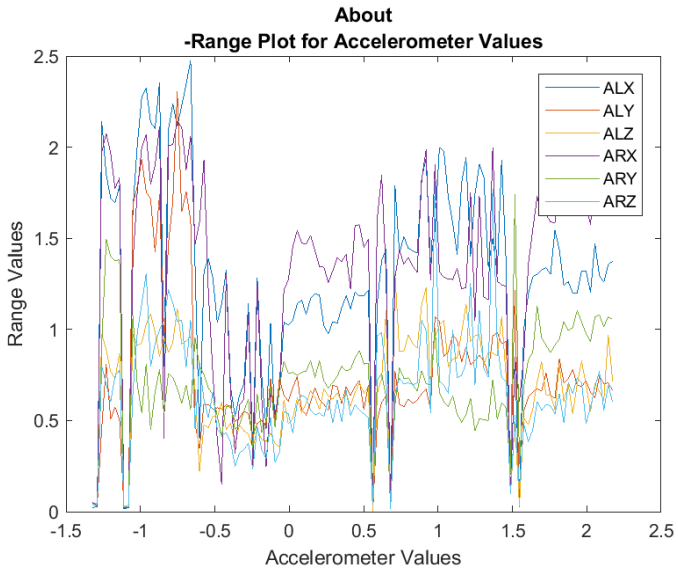


FFT plot for ‘And’

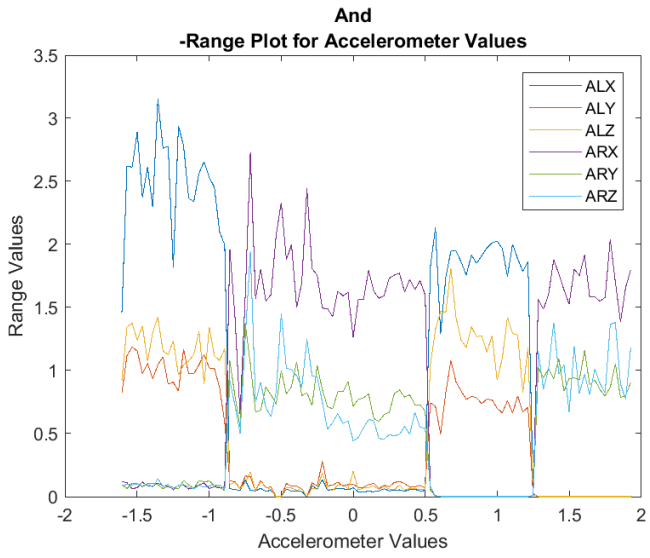
The plots of the other gestures are placed in the folder ‘FFT’ under ‘Plots’.

e) From the graphs we can observe that certain actions like ‘can’, ’find’ and ‘deaf’ have more peak frequencies compared to others where the curves are stabilized over time ,not allowing for sharp gradient turns .The X-axis accelerometer values for all the gestures lies between -1 and +3 while the FFT values vary in the range contained by 0 and 80. With the exceptions of ‘About’ and ‘Can’, every gesture has higher frequencies recorded along X-axis, while frequencies along Y-axis and Z-axis are much lower. The gestures ‘can’, and ‘About’ attain their maximum frequencies at around 50 while the other gestures have maximum values greater than or equal to 70. Having given such interesting insights into the action patterns, our intuition to go with FFT for feature extraction proved to be correct.

2.2.3 Range:



Range plot for ‘About’



Range plot for ‘And’

a) Range is an aggregate of two measure of data representation, namely the minimum and maximum values of a dataset. Range can be defined as the difference between the highest and the least value in a dataset. As a heuristic in data mining, range can help us understand the domain of each feature quantitatively and hence the spread of the data values. Because of this similarity, range is closely associated with standard deviation which also indicates the spread of data points. We find the range of the six accelerometer sensors and plot the trends accordingly.

b) Range as a feature extraction method can help us understand the extremities in the values of each feature taken. The difference between minimum and maximum is indicative of the overall variance in the feature in the whole dataset. It gives the possible range of values the feature could take, denoting the spread of data for that feature.
c) Matlab code has been attached in the folder as "Task2MeanFFTRange.m".

d) The functions 'max' and 'min' were used in Matlab to calculate the difference between the two for every row in the dataset. This was done taking the six accelerometer sensors ALX, ALY, ALZ, ARX, ARY and ARZ as features. The screenshots of 'range' on all the ten gestures are provided in the folder 'Range' in the submission. The X-axis contains labels in the range covered by the minimum and the maximum accelerometer values in the dataset while the Y-axis contains the results of 'range' on the six sensor values.

The plots of the other gestures are placed in the folder 'Range' under 'Plots'.

e) From the graphs, we can observe that the overall minimum and maximum values of each gesture lies in between the range covered by -2 and +4.5 along the X-axis while the Range values in the Y-axis are contained in the interval between -3 and +5. Overall, the ALX, ALY and ALZ values have a higher range compared to the right accelerometer x, y, z sensors. This means that these actions have high steady feature values and they maintain this interval while the right sensors have low values and take on high values suddenly that causes the curves to spike suddenly and tower in the graphs. The gestures 'And', 'cop', 'deaf', 'father', 'find' have sudden peaks in the curve of ARX feature that indicates that ARX is amplified at certain points when performing the gestures. 'About', 'can' and 'deaf' plots show similar curves in the ranges of the six features such that each feature varies only in the strength of the action and not on the nature of the action. Our intuition to select 'range' as a feature extraction method gave us many insights.

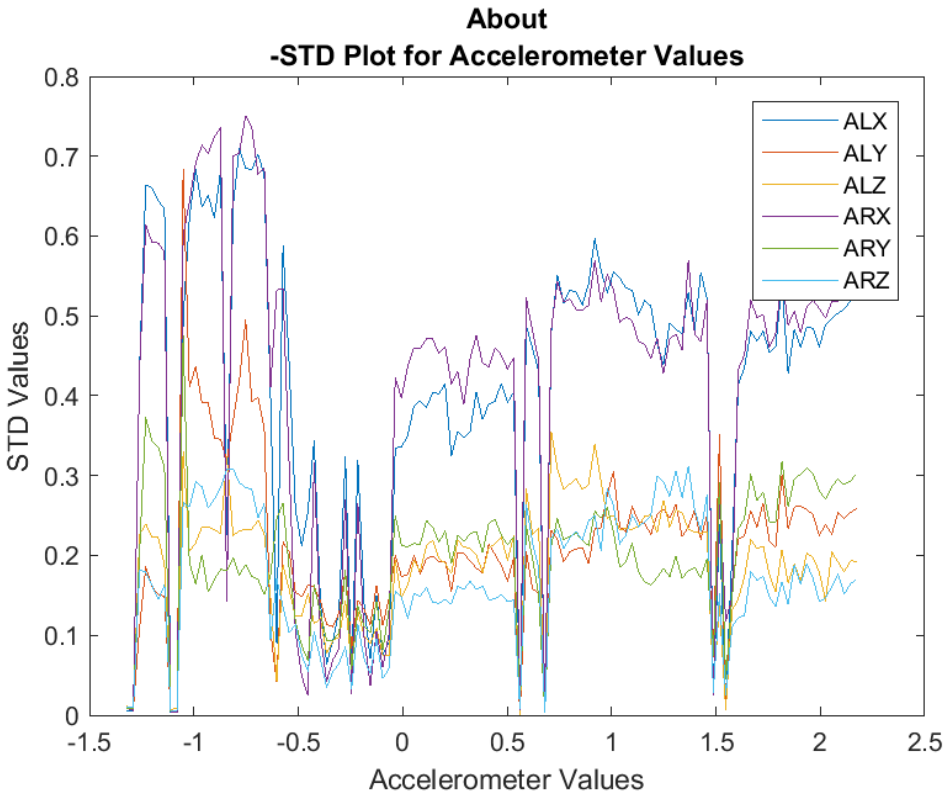
2.2.4 Standard Deviation:

a) Standard Deviation for a group of data points denotes the degree of variation among them. For each action, we have selected the accelerometer values of left and right hands. Each hand has data streams along x, y and z axes. The data file that has been generated in task-1 is used to generate the data matrix for each action. From the data matrix, we have selected the data streams of accelerometer values and applied standard deviation method. The obtained standard deviation values for each data stream along a time-series data are stored in an array. Then we have plotted a graph for standard deviation values Vs accelerometer values along each x, y and z axes of both hands.

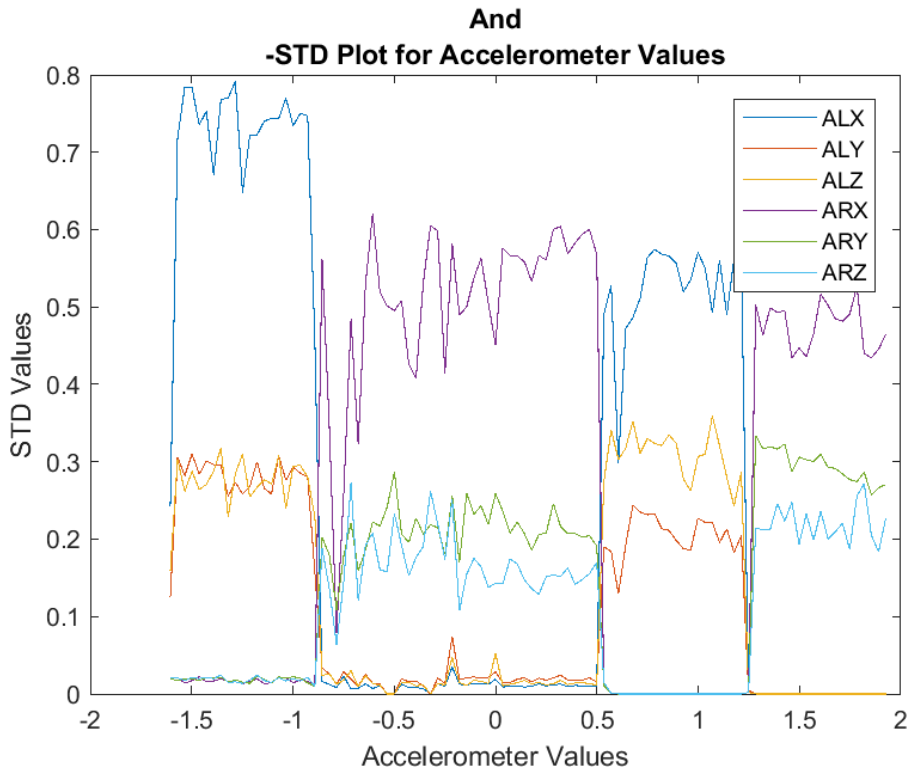
b) As standard deviation represents the amount of variation for a group of data, using this feature extraction method will help us differentiate each action based on the amount of variation existing among them. Each action involves different hand movements. For example, performing Father action, requires only right hand and hence the plot contains a lot of spikes for right hand accelerometer values because of high deviation. Whereas, for performing can action both hands are used and we observe a lot of spikes for both hands accelerometer values denoting high deviation among them. Thus, standard deviation as a method of feature extraction will help us differentiate actions clearly.

c) Matlab code for standard deviation can be found in "Task2-RMS-STD.m" file.

d) Matlab function 'std' is used to calculate the standard deviation of six accelerometer values ALX, ALY, ALZ, ARX, ARY, ARZ. The plotted figures are saved in the folder named 'STD' in the submission. The X-axis contains labels in the range covered by the minimum and the maximum accelerometer values of the dataset, while the Y-axis contains the results of 'Standard deviation' of the six sensor values.



STD plot for 'About'



STD plot for 'And'

The plots of the other gestures are placed in the folder 'STD' under 'Plots'.

e) Observing the plots for standard deviation of all the ten gestures and seeing obvious deviations of patterns in each gesture, we say that our initial intuition to analyze the data using 'standard deviation' as a measure proved insightful. From the data we can see that there a lot of spikes for both left and right axes. Some gestures like Father, Cop have more spikes on right axes compared to left. While for other actions, like Can and Decide have more spikes on both the axes. These trends justify the selection of standard deviation as a feature extraction method.

2.2.5 Root Mean Squares:

a) Root Mean Squares is used to measure the magnitude of a data set. If a data set contains a lot of negative and positive values, then computing average won't give us the desired result as negative values are cancelled by positive values. Thus, to take average efficiently and compute magnitude 'RMS' is the best approach. For each action, we have selected the accelerometer values of left and right hands. Each hand has data streams along x, y and z axes. The data file that has been generated in task-1 is used to generate the data matrix for each action.

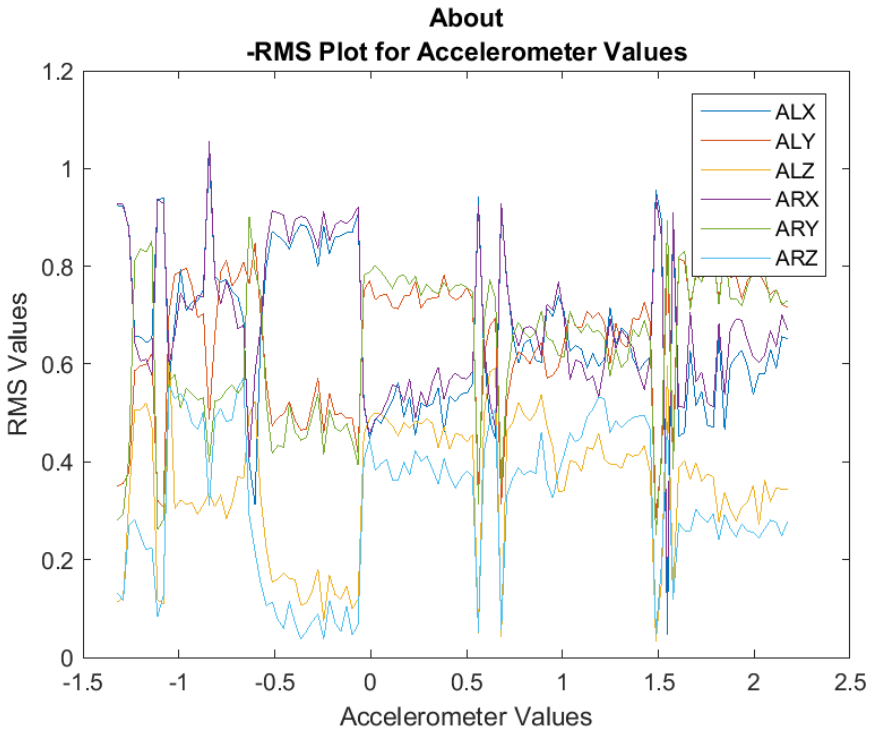
From the data matrix, we have selected the data streams of accelerometer values and applied Root Mean Squares method. The obtained RMS values for each data stream along a time-series data are stored in an array. Then we have plotted a graph for RMS values Vs accelerometer values along each x, y and z axes of both hands.

b) As root mean squares method measure the amount of magnitude for each actions data set, using this feature extraction method will help us differentiate each action clearly. The time-series plot of the RMS values can help us gauge whether certain gestures are similar in their acting patterns and if the movements consist of sudden turns resulting in sharp spikes or falls in the graph.

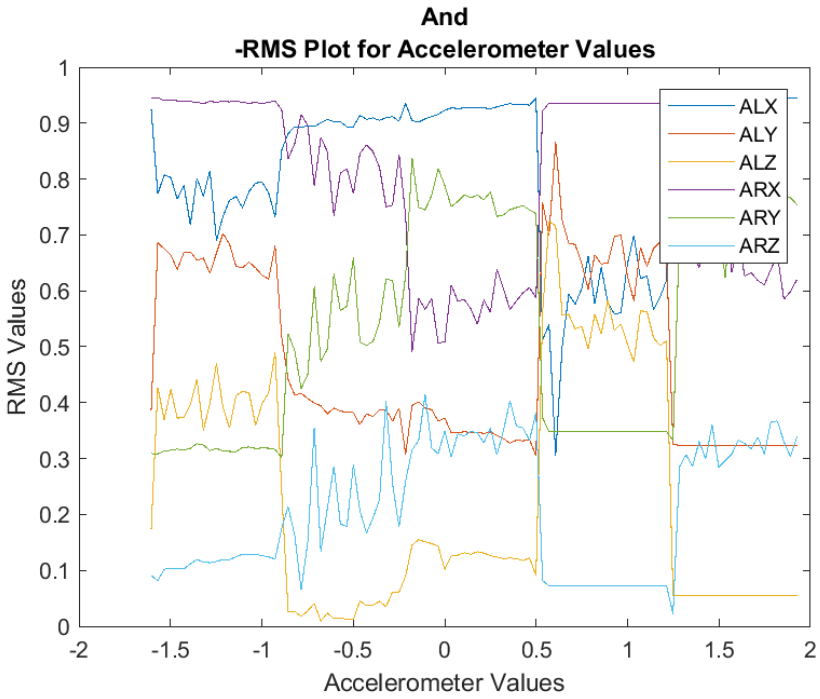
For example, actions like Father, Cop uses only right hands and hence magnitude of their data sets will be similar, where as for other actions which used both hands like can, decide may contain different magnitudes based on their hand movements and we will observe a lot of spiked or falls.

c) Matlab code for Root-Mean- Squares can be found in "Task2-RMS- STD.m" file.

d) Matlab function 'rms' is used to calculate the Root-Mean- Squares of six accelerometer values ALX, ALY, ALZ, ARX, ARY, ARZ. The plotted figures are saved in the folder named 'RMS' in the submission. The X-axis contains labels in the range covered by the minimum and the maximum accelerometer values of the dataset, while the Y-axis contains the results of 'Root-Mean-Squares' of the six sensor values.



RMS plot for 'About'



RMS plot for 'And'

The plots of the other gestures are placed in the folder 'RMS' under 'Plots'.

e) Observing the plots for Root-Mean- Squares of all the ten gestures and seeing obvious spikes or falls of patterns in each gesture, we say that our initial intuition to analyze the data using 'RMS' as a measure proved insightful. From the data we can see that there a lot of spikes for both left and right axes. Some gestures like Father, Cop have more spikes on right axes compared to left. While other actions, like "Can" and "Decide" have more spikes on both the axes. These trends justify the selection of 'Root-Mean- Squares' as a feature extraction method.

The initial intuition about why we selected these features holds true.

2.3 Task 3: Feature Selection

In this task we have reduced the feature to a lower dimension using PCA. PCA finally keeps only those features which show maximum distance between two classes. The PCA function which is already present in MATLAB is used here.

2.3.1 Arranging the Feature Matrix

The data matrix is built separately for all the 10 gestures and finally concatenated to form a single matrix which is then given as an input to PCA. We are giving a single matrix as input to PCA because we want the PCA to give us a set of eigen vectors which are common for all the gestures and hence these vectors will be used to transform the data of all 10 gestures into new feature space. The features for all the sensors are calculated and appended. All the 34 sensors are the

rows of this data matrix. The streams of data from the five different features (Mean, RMS, Range, FFT, Standard Deviation) are the columns of this data matrix. The template of the data matrix of a single gesture is shown below:

Sensors	Mean	FFT (Peak Value)	RMS	Range	STD	Mean	FFT (Peak Value)	RMS	Range	STD
ALX										
ALY										
ALZ										

Figure: Template of data matrix of a single gesture

Since we have worked on 5 data sets, there are 20 files of each gesture which in turn results in 100 such data records. So, there are 100 columns for each feature which makes the number of columns to 500. Hence, the size of the data matrix of a single gesture is 34 * 500.

Now all the 10 data matrices corresponding to the 10 gestures are concatenated column wise into a single matrix of size 34 * 5000. This is the final data matrix which is used to apply PCA.

2.3.2 Execution of Principal Component Analysis (PCA)

After PCA was executed we got multiple principal components out of which the top 5 principal components are shown below. These help us get the maximum variance in the new feature space.

PC1	PC2	PC3	PC4	PC5
0.012162	-0.00308	0.006869	-0.00293	-0.00986
0.003779	0.000197	0.002544	-0.00091	-2.4E-06
0.00331	-0.0016	0.001769	-0.00062	-0.00164
0.00788	0.008032	0.006604	0.004751	9.19E-05
0.002985	0.002704	0.00209	0.001223	0.000238
0.001491	0.002897	0.001296	0.00251	0.000494
0.114188	-0.07478	0.310911	-0.32691	-0.42262
0.112384	-0.05594	0.269686	-0.19295	-0.18101
0.083307	-0.01654	0.310642	-0.09371	-0.10897
0.075776	-0.04328	0.273742	-0.23792	0.519092
0.085649	-0.0427	0.26573	-0.14215	0.58046
0.075166	-0.02868	0.232877	-0.14911	0.228952
0.071497	-0.00479	0.194602	-0.04207	-0.1152
0.111206	-0.04179	0.396238	-0.16033	-0.2912
0.049609	0.126327	0.160401	0.353598	-0.00905
0.048777	0.086368	0.174659	0.231467	0.084414
0.04152	0.070929	0.140461	0.209341	0.085086
0.041273	0.088424	0.167615	0.263203	-0.01061

0.054061	0.108252	0.187113	0.31846	-0.01338
0.041542	0.059781	0.142381	0.207033	0.010483
0.059219	0.070966	0.17655	0.277696	-0.06181
0.054693	0.098658	0.229805	0.35608	-0.02468
0.498734	-0.23577	-0.14722	0.079729	0.007523
0.498734	-0.23577	-0.14722	0.079733	0.007524
0.498734	-0.23577	-0.14722	0.079733	0.007524
0.23081	0.499885	-0.09259	-0.1415	-0.00017
0.23081	0.499885	-0.09259	-0.1415	-0.00017
0.23081	0.499885	-0.09259	-0.1415	-0.00017
0.039953	-0.01127	0.043192	-0.02245	0.001279
0.020152	0.008332	0.023592	-0.01266	0.004429
0.035039	0.01858	0.036506	-0.02354	-0.01354
0.026599	0.033084	0.015443	0.007698	-0.00921
0.021685	0.010329	0.01286	-0.00557	-0.00181
0.032996	0.011794	0.03349	-0.01453	0.005402

Table: Top 5 principal components

Since there are more dimensions to plot, visualizing the data is very difficult in a graph such as a biplot. Hence, we used a spider plot to show the eigen vectors. The spider plot consisting of the top 5 principal components is as follows:

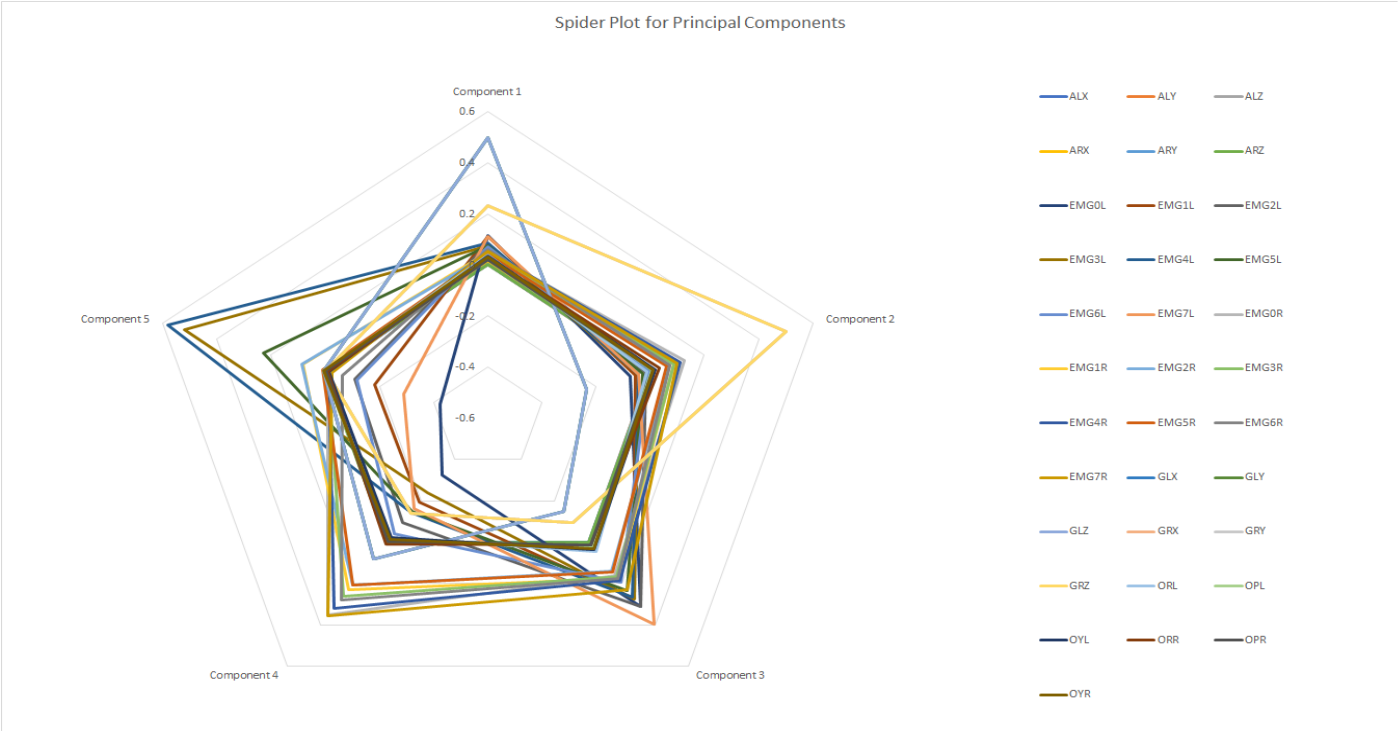


Figure: Spider Plot for Principal Components

2.3.3 Inference from Eigen Vectors

The percentage variability of the eigen vectors are found after execution of PCA. Almost, all the variance is depicted by the top three principal components. The percentage variance due to the top few principal components in the decreasing order are shown below.

Percentage Variance
66.46362
23.7395
4.470762
1.658676
0.67659
0.455863
0.398082
0.351867
0.345182
0.266318
0.225581

So, we inferred that PCA can be used to reduce the dimensions drastically. Almost 95% variance can be determined by the first 3 principal components.

2.3.4 Plots (Before and After PCA)

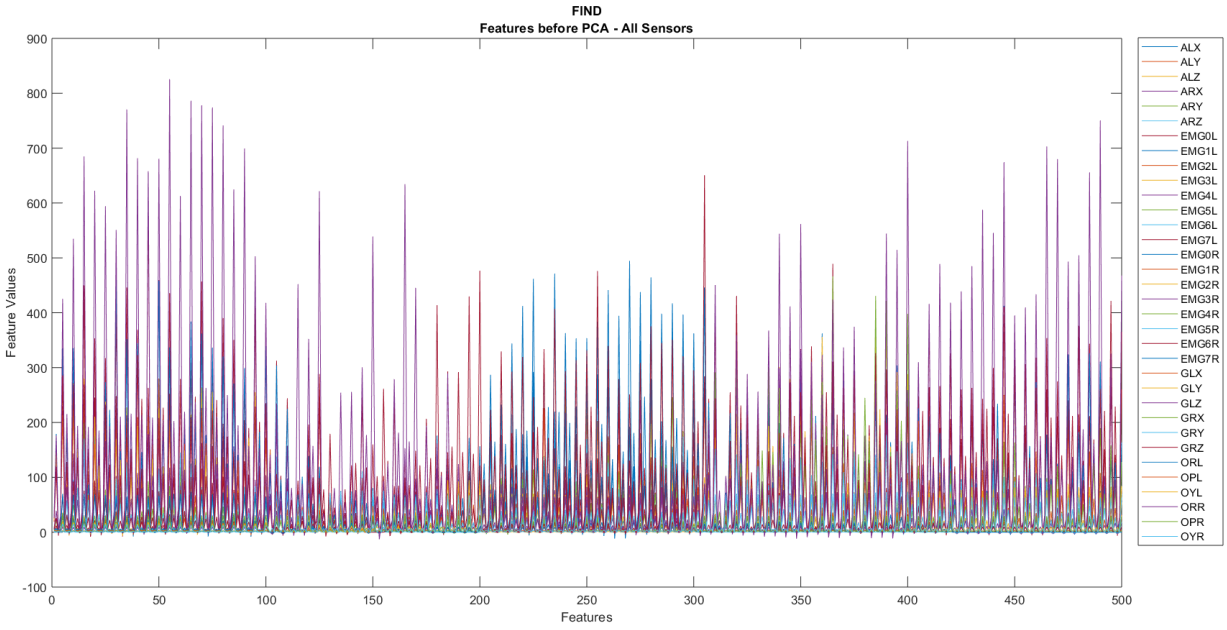


Figure: Features before PCA (FIND)

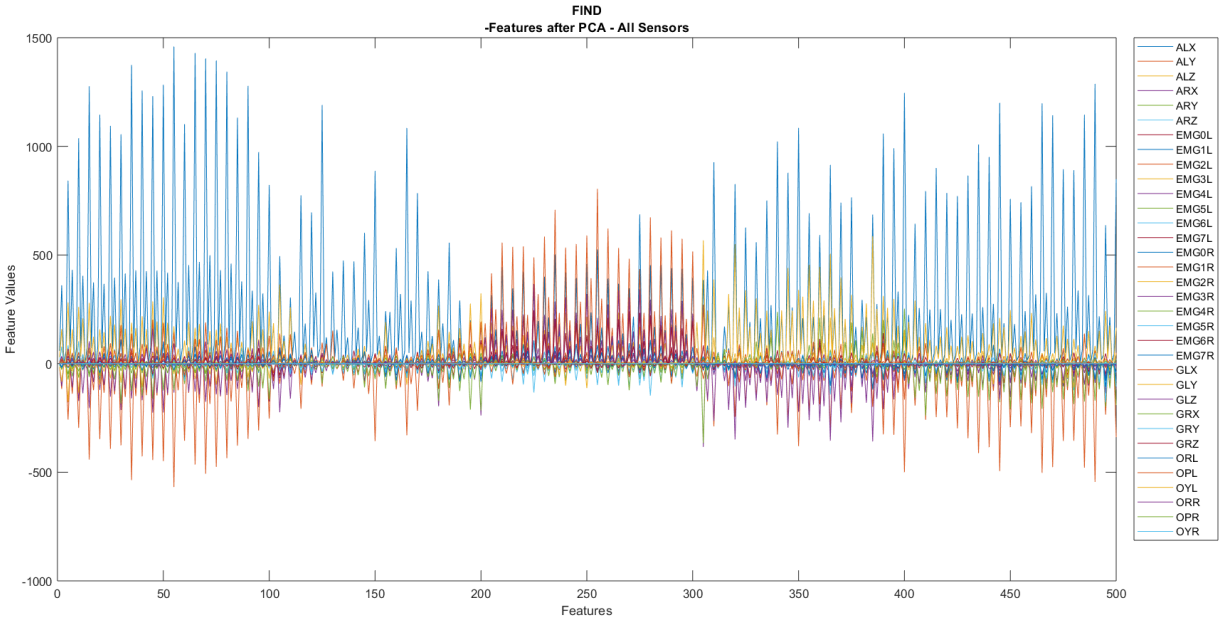


Figure: Features after PCA (FIND)

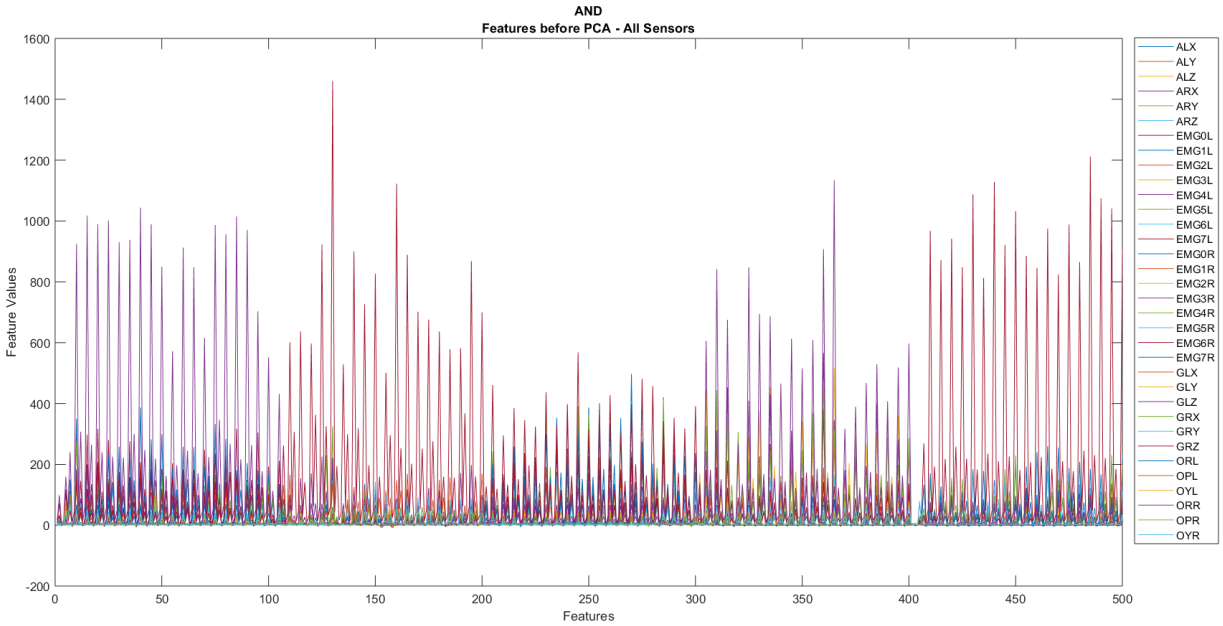


Figure: Features before PCA (AND)

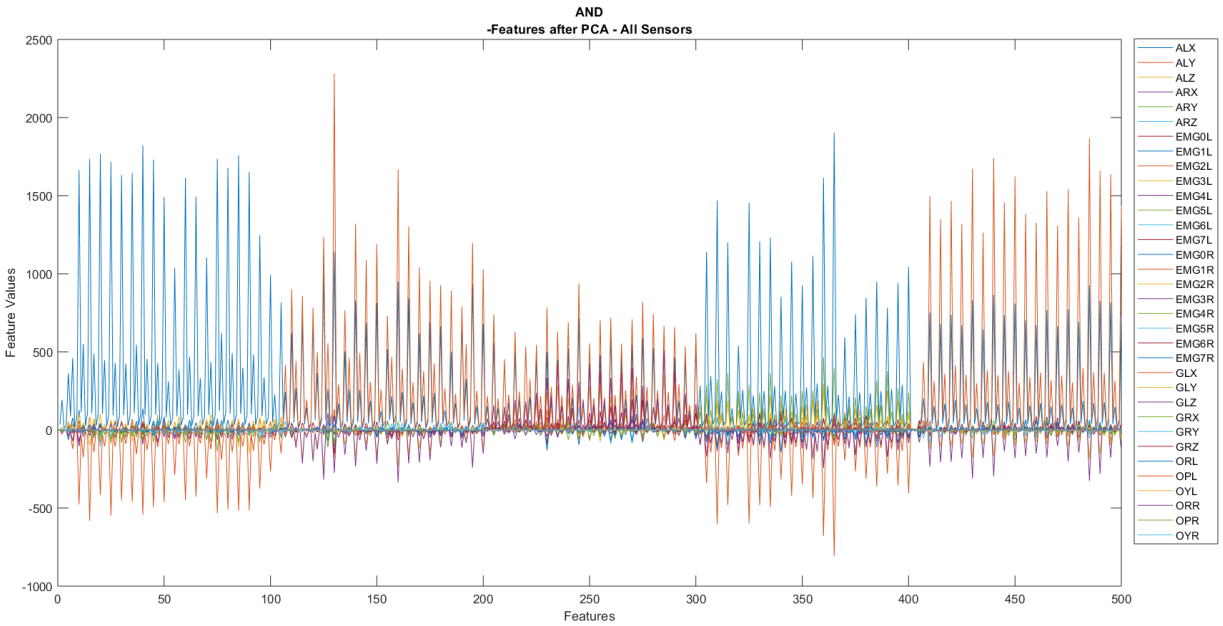


Figure: Features after PCA (AND)

We have plotted the feature matrices before and after PCA and obtained the above results. These are the graphs for two gestures, “FIND” and “AND”. The plots for other gestures are included in the zip file submitted along with the report. We observed that after applying PCA, the new feature space is spread out over a wider range. As it is seen in the above graphs, the feature values for “FIND” lie between 0 to 800 before PCA but in the new feature space, they lie between -500 to 1500. This clearly shows the greater variance achieved after applying PCA.

2.3.5 Conclusion

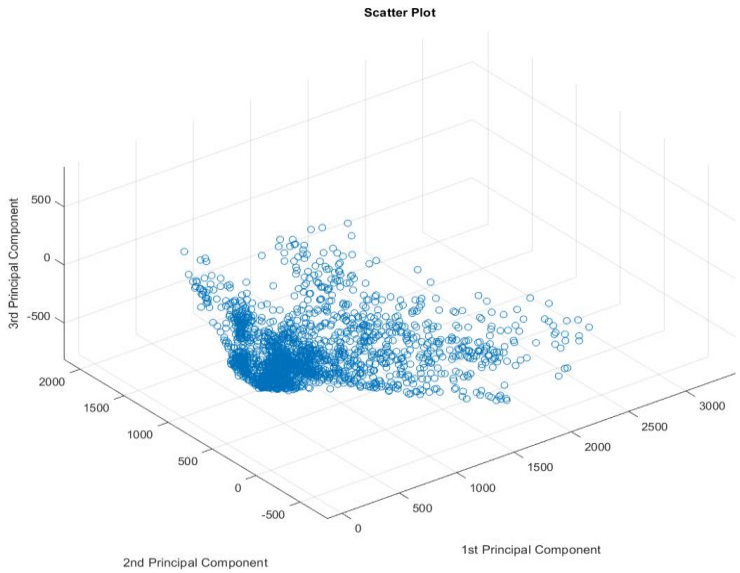


Figure: Scatter plot of data across Principal Components

We plotted a scatter plot using the feature data across the top three principal components. The above scatter plot gives us a visual representation how well the first three principal components can be used to determine variance. We can infer from the above figure that the largest variability is along the first principal component. This is the largest possible variance among all the others. The variability along the second principal component is the second highest among all the principal components.

We would like to conclude that performing PCA was helpful to transform the data into new feature space where there is maximum variance. As explained above, the first 3 principal components itself attribute to about 95% variance. Hence, using just Nx3 data matrix, we can obtain very high variance and at the same time save a lot of space.