Human Activity Recognition Dataset - A Comparative Study

Shriya Rai
Rachit Jas

University of Minnesota
Driven to Discover℠

Outline

- Introduction
- Data Analysis
- Predictive Models with Feature Extractors
- Standalone Models with Automated Feature Extraction
- Conclusion

Quick Data Overview

- **30 volunteers'** activity series data collected from Samsung SII
- **Age bracket** of volunteers: 19-48 years
- **Total samples** = 7352, **Feature Vector** = 561
- **6 classes**: Walking, Standing, Sitting, Laying, Walking Upstairs, Walking Downstairs
- **2 sensors** - accelerometer + gyroscope (giving 3 dimensional data)
- **Challenging Data**
  - Multiclass problem, multivariate time series data for each patient
  - Total activity data for each volunteer non-constant
  - Each activity sequence length for volunteer non-constant
- **Useful for**:
  - For further health studies (for example, collecting data to detect possible sleeping period of people)

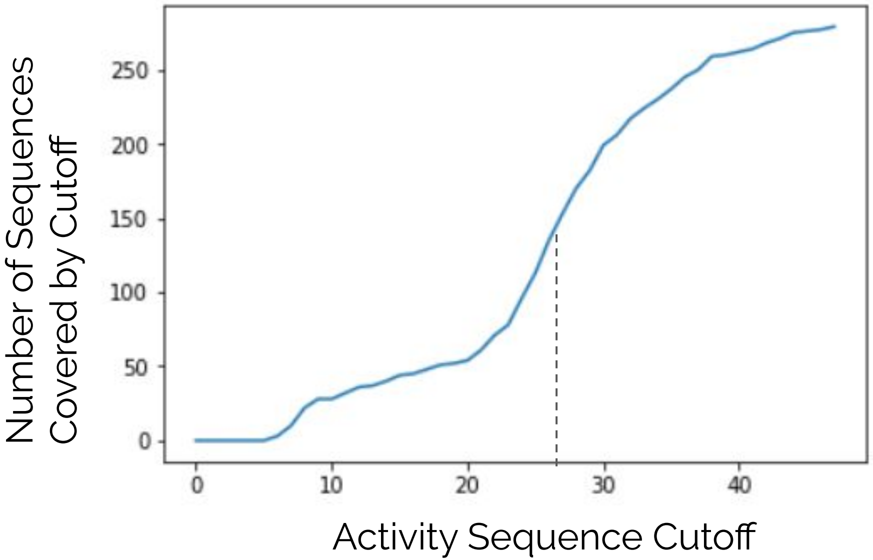
Data Transformations

- 80% of the time went in preparing the data!

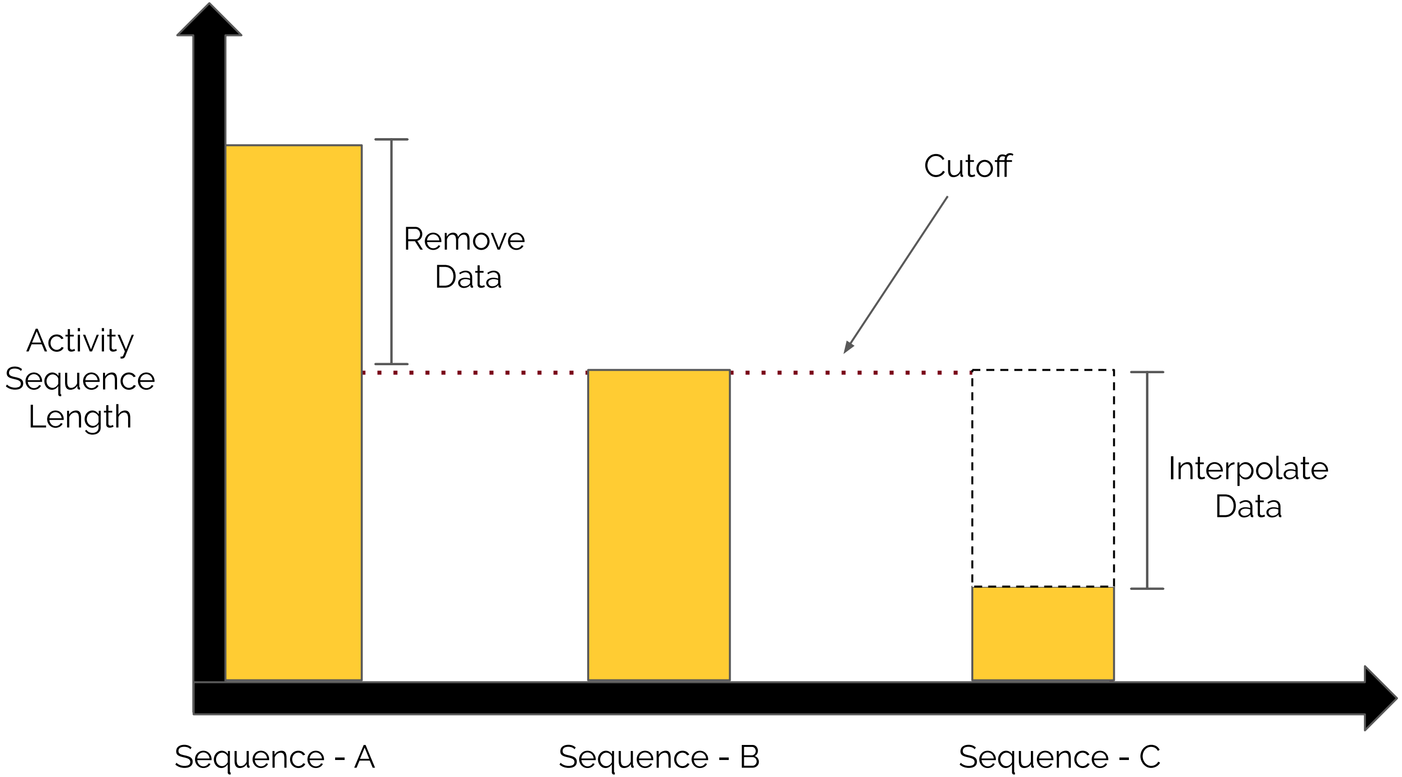
- Standardization
- **Series Data Interpolation (Order 3 Spline)**
- **Removing Extra Data**
- Cutoff = "juggle"' between
  data loss v.s. **Interpolation error**
  (obs. data)   (missing data)
- **Cutoff** = based on central measures
  of tendency: **mean**, median

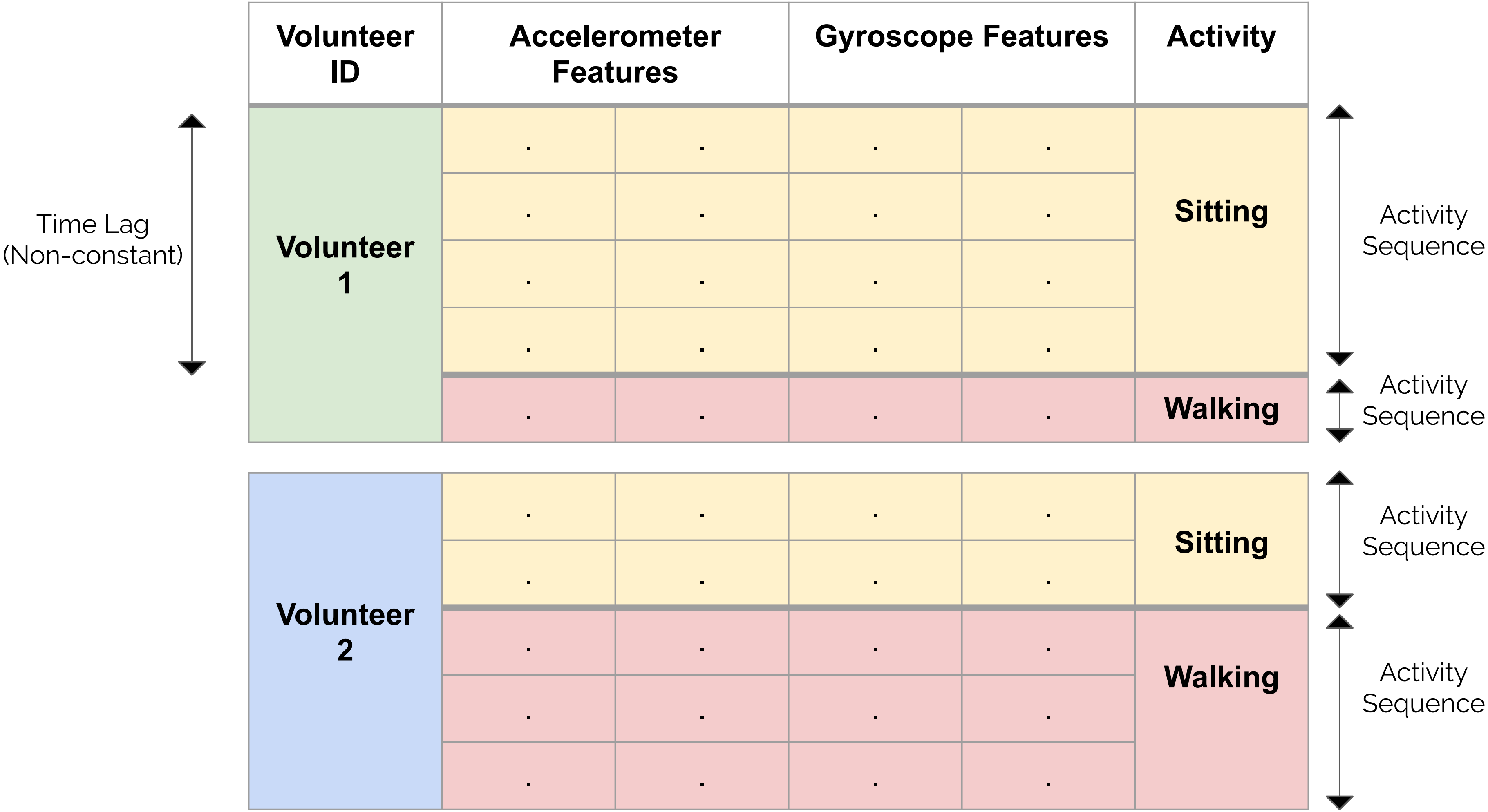
- 280 Activity Sequences, each
  being a (cutoff x 561) 14586 input vec

# of sequences captured by Cutoff (Lag)



Activity Sequence Length Problem





Volunteer ID	Accelerometer Features		Gyroscope Features		Activity
Volunteer 1	.	.	.	.	Sitting
	.	.	.	.	
	.	.	.	.	
	.	.	.	.	
	.	.	.	.	Walking
Volunteer 2	.	.	.	.	Sitting
	.	.	.	.	
	.	.	.	.	Walking
	.	.	.	.	
	.	.	.	.	

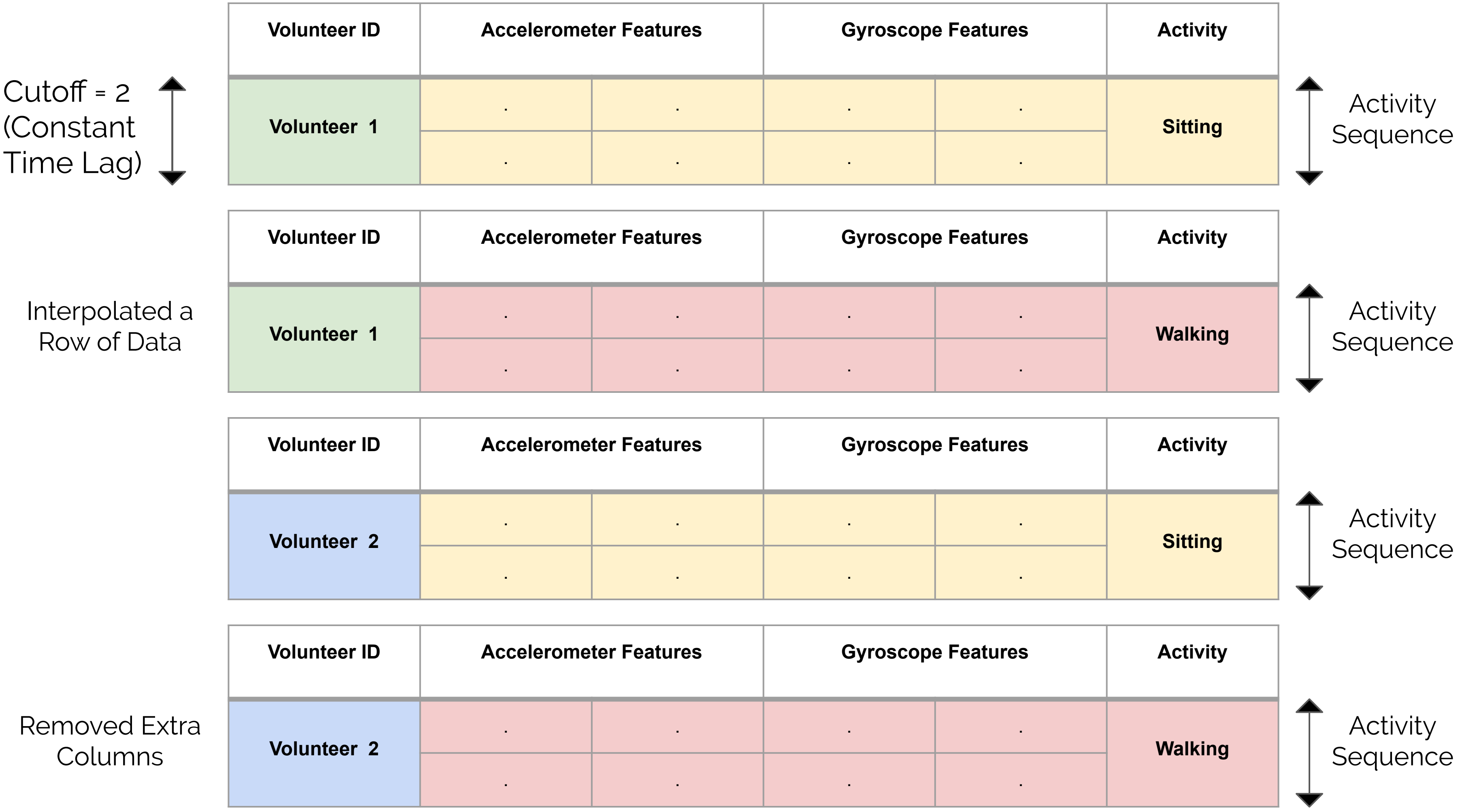
Time Lag
(Non-constant)

Activity Sequence

Activity Sequence

Activity Sequence

Activity Sequence



Cutoff = 2 (Constant Time Lag)

Volunteer ID	Accelerometer Features		Gyroscope Features		Activity
Volunteer 1	.	.	.	.	Sitting
	.	.	.	.	

Activity Sequence

Interpolated a Row of Data

Volunteer ID	Accelerometer Features		Gyroscope Features		Activity
Volunteer 1	.	.	.	.	Walking
	.	.	.	.	

Activity Sequence

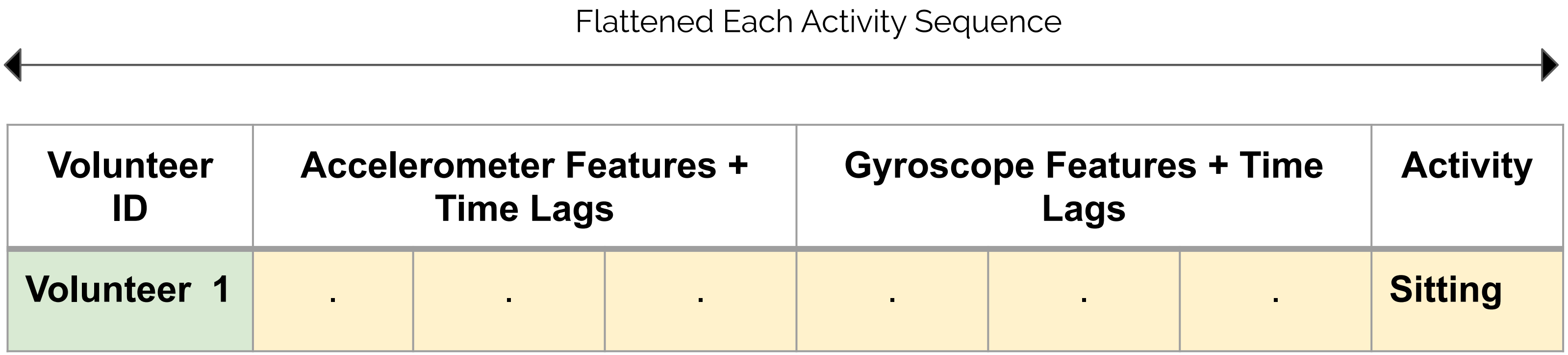
Volunteer ID	Accelerometer Features		Gyroscope Features		Activity
Volunteer 2	.	.	.	.	Sitting
	.	.	.	.	

Activity Sequence

Removed Extra Columns

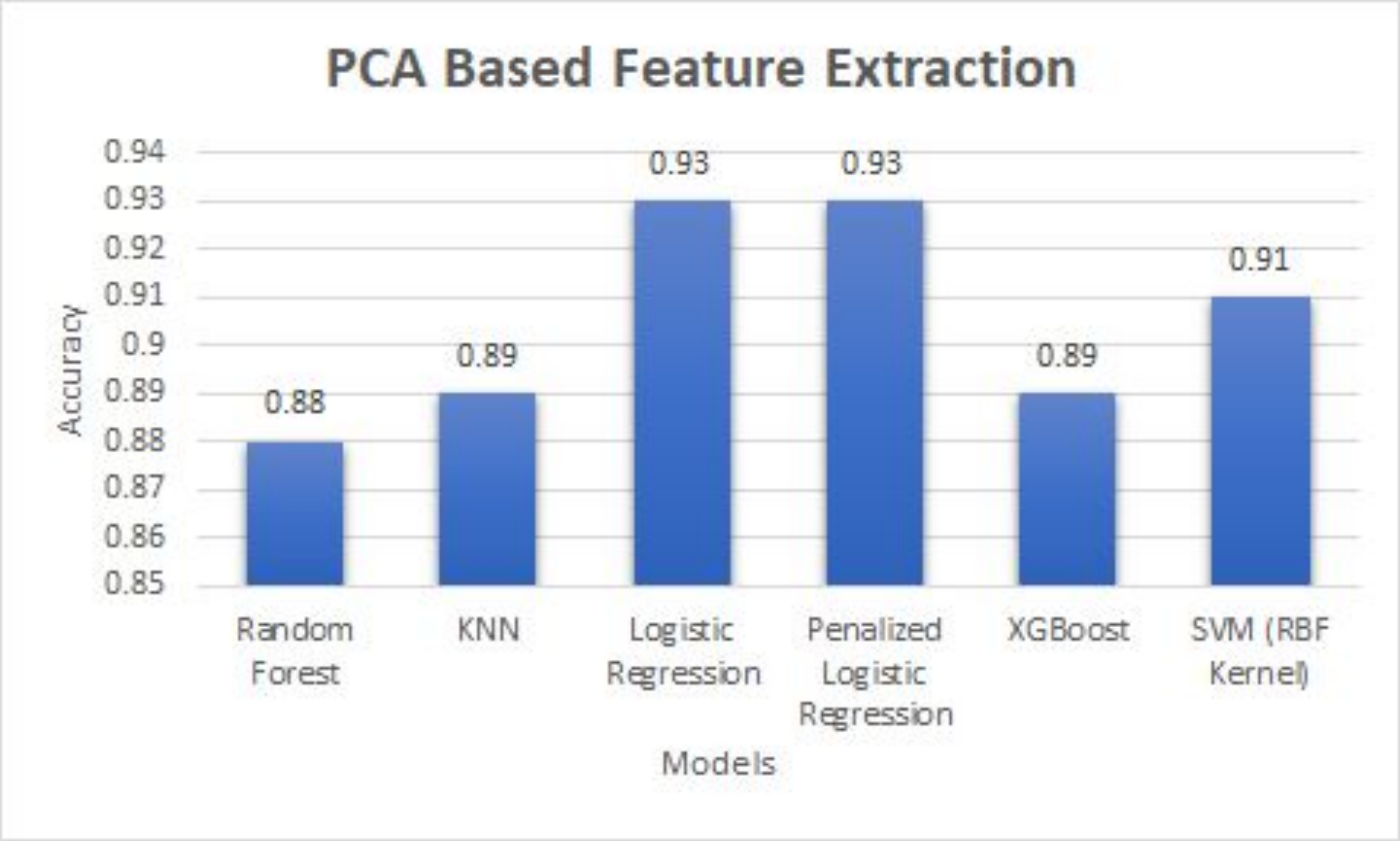
Volunteer ID	Accelerometer Features		Gyroscope Features		Activity
Volunteer 2	.	.	.	.	Walking
	.	.	.	.	

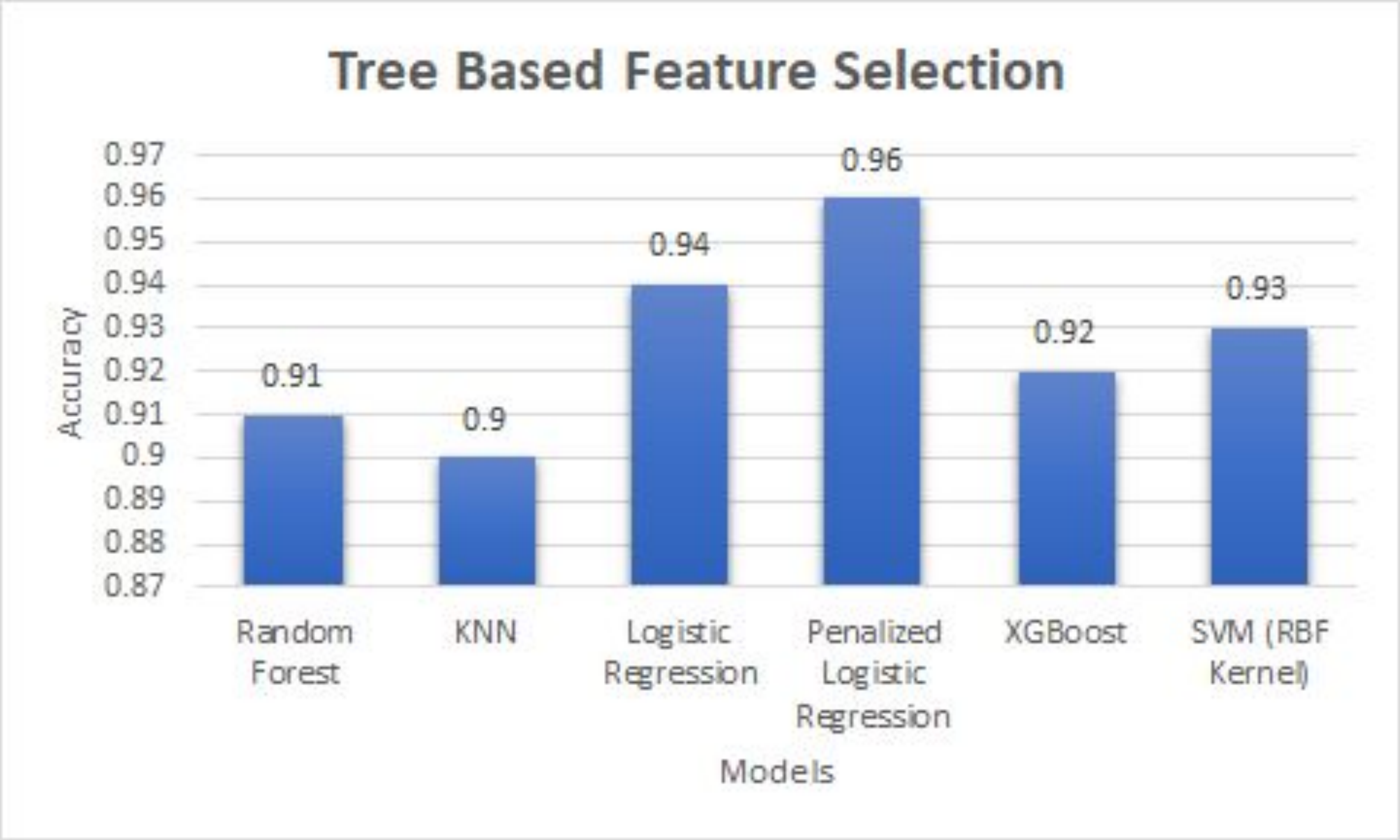
Activity Sequence

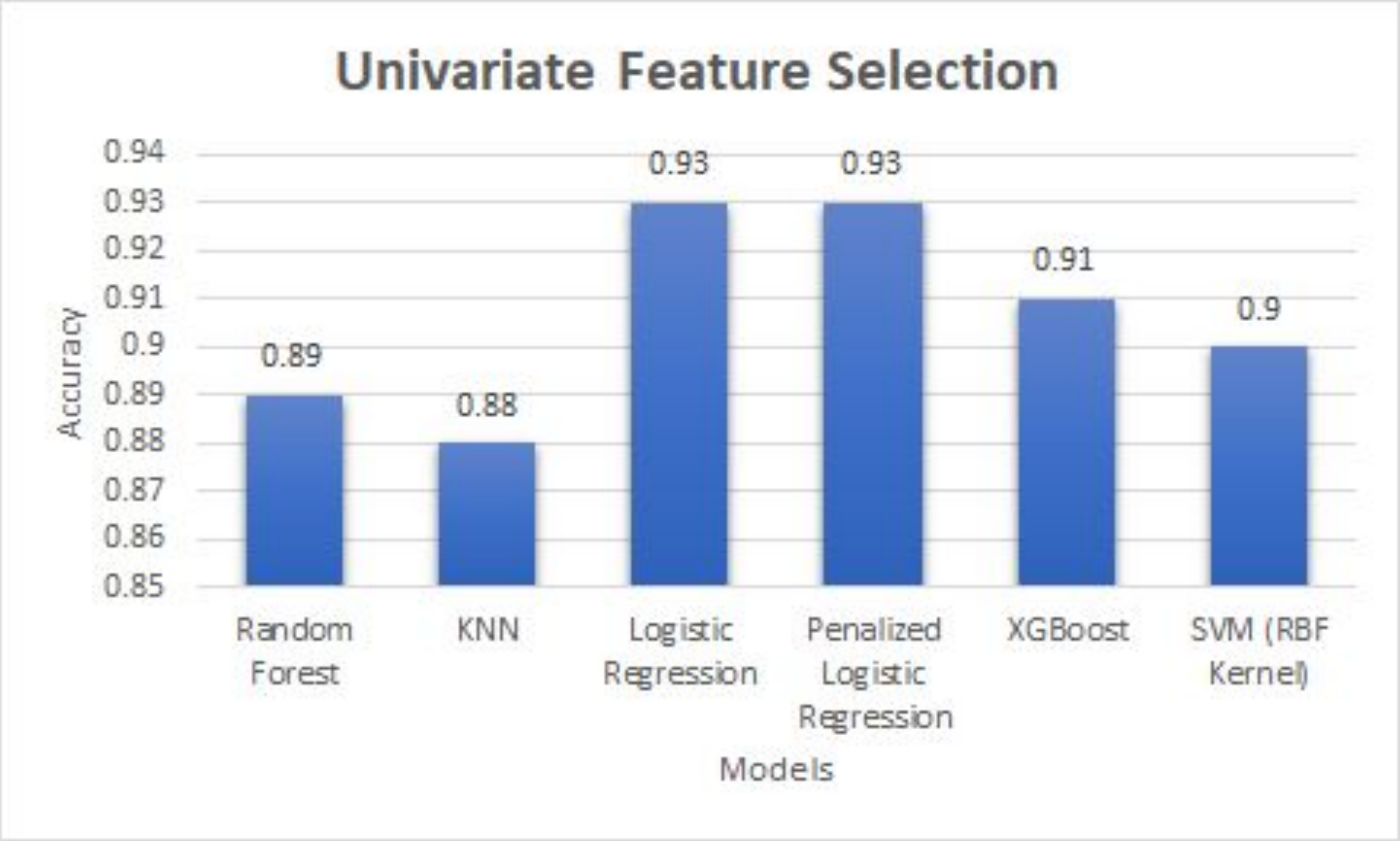


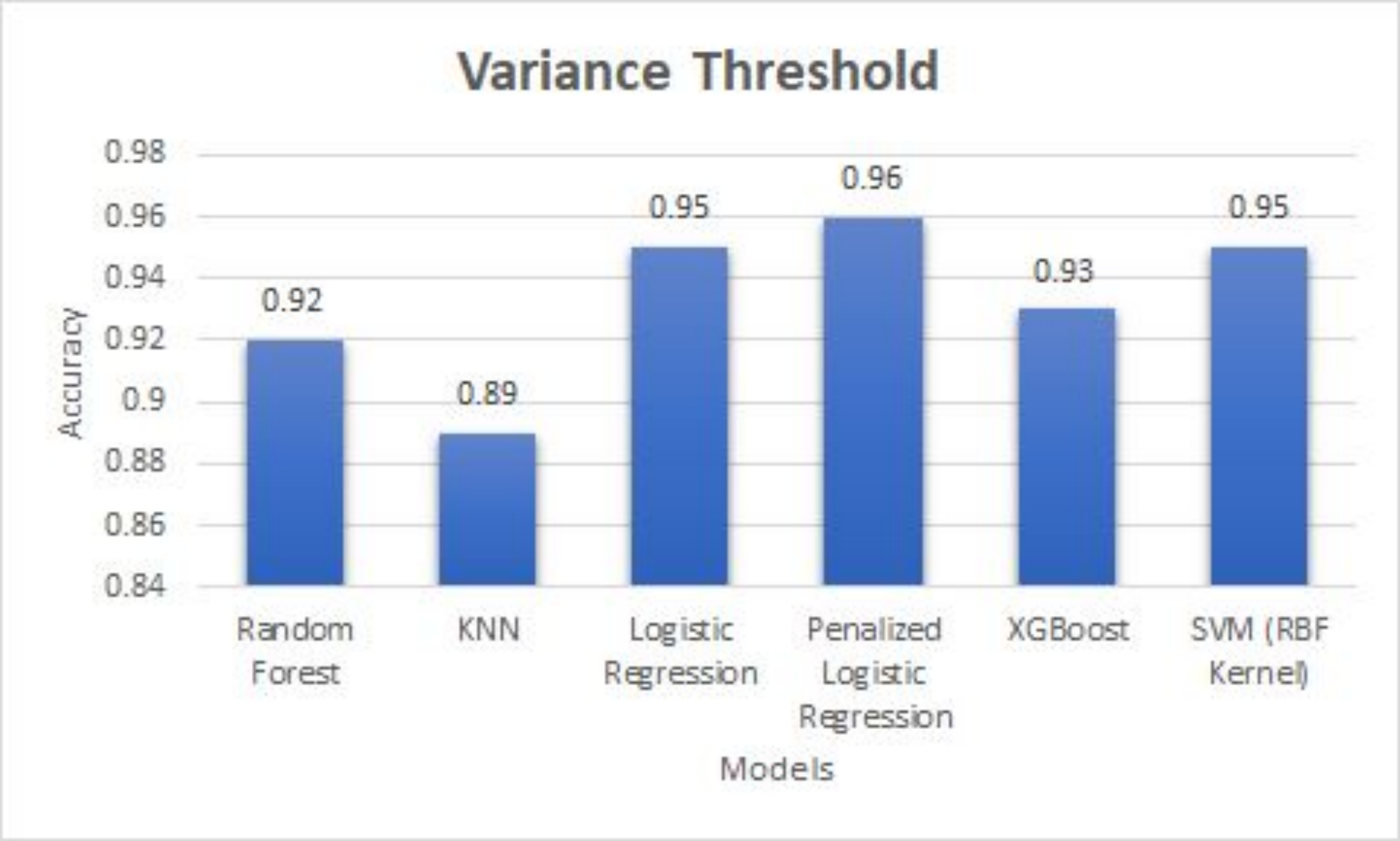
Flattened Each Activity Sequence

Volunteer ID	Accelerometer Features + Time Lags			Gyroscope Features + Time Lags			Activity
Volunteer 1	.	.	.	.	.	.	Sitting


PCA Based Feature Extraction
Accuracy
0.94
0.93
0.92
0.91
0.9
0.89
0.88
0.87
0.86
0.85
0.88
0.89
0.93
0.93
0.89
0.91
Random Forest
KNN
Logistic Regression
Penalized Logistic Regression
XGBoost
SVM (RBF Kernel)
Models


Tree Based Feature Selection
Accuracy
0.97
0.96
0.95
0.94
0.93
0.92
0.91
0.9
0.89
0.88
0.87
0.91
0.9
0.94
0.96
0.92
0.93
Random Forest
KNN
Logistic Regression
Penalized Logistic Regression
XGBoost
SVM (RBF Kernel)
Models


Univariate Feature Selection
Accuracy
0.94
0.93
0.92
0.91
0.9
0.89
0.88
0.87
0.86
0.85
0.89
0.88
0.93
0.93
0.91
0.9
Random Forest
KNN
Logistic Regression
Penalized Logistic Regression
XGBoost
SVM (RBF Kernel)
Models


Variance Threshold
Accuracy
0.98
0.96
0.94
0.92
0.9
0.88
0.86
0.84
0.92
0.89
0.95
0.96
0.93
0.95
Random Forest
KNN
Logistic Regression
Penalized Logistic Regression
XGBoost
SVM (RBF Kernel)
Models

Automated Feature Extraction (based on Neural Networks)

Optimizer: Adam

Loss: Cross Entropy

Model	Layers	Architecture	Details

CNN	6 + 1	Conv/Max ->
		Dropout ->
		Conv/Max ->
		Dropout ->
		Fully Connected ->
		Fully Connected ->
		Softmax
			Layers = 7
			Kernel Size = 5
			Stride = 1
			Padding = 0
			Pool Size = 2

RNN	4 + 1	RNN Unit -> RNN Unit -> RNN Unit -> RNN Unit -> Softmax	Hidden Dimensions = 100 Output Dimensions = 6

Interesting Observations

- CNN with Dropout: **0.95** Accuracy, Slower
- CNN without Dropout: **0.94** Accuracy, Slightly Faster

- RNN with Dropout: **0.98**, Slightly Slower
- RNN: **0.996** Accuracy (1 miss-classified sample from 2947 samples), Fastest and greater than **0.993 (accuracy as claimed in Kaggle!)**

Final Conclusions

- Rooms for improvement:
  - Imputation Error needs to be studied more - spline order might change with time
  - Still less data for training, (only 7532 samples => 280 sequences)
  - True infinite data distribution could be imbalanced

- Are complex algorithms worth the computational time wait?
  - Penalized Logistic Regression took **couple of minutes** to train (**96% accuracy**) !
  - CNN took **2 hours** for training (**95% acc**), RNN took **30 minutes for training** (**99.6% acc**)

- High Dimensional data (14586) => expected SVM to perform better