Human Activities Recognition (HAR): Data Sampling Machine Learning Analysis

Sean Mullane (spm9r), Sri Vemulapalli (sv2fr), Tommy Jun (tbj2cu)

I. INTRODUCTION

Smartphone sensor data has been shown to be a useful input for Human Activity Recognition modeling. One challenge to ensuring adoption of smartphone-based HAR applications is ensuring that the application that collects the data does not use so much power that the phone's battery life is adversely affected. Energy use increases in proportion to the frequency and duration of data collection. However model accuracy can be adversely affected by the wrong choice of data collection frequency and data collection duration.

In order to make an infomed decision on what tradeoff to make between accuracy and efficiency, we modeled the effects of various data collection frequencies and temporal window sizes using data from UC Irvine's Human Activities Recognition (HAR) research. Through this exploration, energy expended by data-gathering hardware can be optimized on a cost-to-performance ratio.

The metrics analyzed are calculated by applying classification-based machine learning algorithms on a feature set derived from the most recent public dataset from HAR, Smartphone-Based Recognition of Human Activities and Postural Transitions (2015).

II. DATA PROCESSING

A. Initial Processing

Python is solely used in the data processing stage for its many publicly available packages which proved to be useful for feature extraction. The raw data from HAR[1] consists of readings from accelerometers and gyroscopes mounted on 30 participants during several sessions where they were asked to perform a number of physical activities over a period of time. The data was recorded at a frequency of 50 Hz with varying time frames.

Along with these readings are indicators of the different types physical activities the participants performed (Table I). Furthermore, these activities are categorized into ambulatory and non-ambulatory activities.

To remove noise, a median filter (n=3) is first applied to the raw data. This removes several outliers which may have been recorded during sudden movements or unexpected shifting of the hardware itself. Then a Butterworth filter (f_c=0.3Hz) is passed through the accelerometer readings in order to extract features based on gravitational acceleration. Other methods taken into consideration include a preliminary Butterworth filter (f_c=20Hz) on both the accelerometer and gyroscope data immediately after the median filter in order to further reduce noise; however, due to conflicting issues with downsampling to lower frequencies, too much information would have been lost.

Regarding downsampling, in order to emulate variations in frequency the data is then downsampled by retaining every n-th reading (resulting in 50Hz, 25Hz, 10Hz, and 5Hz). For variations in sampling window, data is grouped into bins of m seconds (resulting in groups of 1, 2, 3, 5, and 10 seconds). Transient periods between switching activities often resulted in partially full bins.

These bins were discarded not only because some calculations were not useful but also because of the inherent ambiguity of factor level while alternating activities. For each permutation of frequency and window size, 576 features from the time and frequency domain are calculated. In particular, the frequency domain calculations are made by applying a fast-Fourier transformation (FFT) over the time domain data.

A list of all of the features is available on the GitHub repository[2].

This results in 20 individual files of 576 features (explanatory and response variables) with varying magnitudes of observation inversely related to the sampling window. Prior to analysis these files are pre-processed by first removing any incomplete or ambiguous features. In particular, entropy calculations were dropped due to extreme values that provided no usable information.

Additionally, smaller groups of data at smaller observation windows contained far too few data points for certain calculations to be valid such as the higher-order autoregressive coefficients or frequency bins. For reference, the smallest group of accelerometer data contained 5 data points (in the 5Hz and 1 second frame).

The data is further processed by scaling all of the features to between the range -1 and 1. The sets are partitioned into training and testing sets (75%/25%, respectively), and the explanatory variables are separated from the response variable. This results 80 data sets (pairs of 20 test and 20 training sets) which are used in the modeling stage.

B. Modeling

The machine learning models are written in either Python and R using *scikitlearn*[3] and *CARET*[4]. Each language provide different strengths and a wide selection of packages to choose from. Different classification algorithms were considered but the four in particular selected for the scope of this study are: Logistic Regression, Decision Trees, SVM, and K-NN. These models are trained on each of the 20 sets produced in the initial processing stage and several metrics are calculated (Precision, Recall, F-Measure, and Accuracy). Each of the models are tuned to maximize accuracy.

The models trained in R were Boosted Logistic Regression, SVM, K-NN, and GLMnet. These models performed similarly to their python counterparts where applicable. These models were trained using 10-fold cross-validation which allowed us to use a hypothesis test to determine significance of the effect of frequency and window size on the overall accuracy of the models.

Regarding these metrics in general, both average values across all factors (i.e. Average Precision) and individual factors (i.e. Average Precision while standing still) are provided. These metrics can be used not only in order to determine the changes in accuracy due to a variation in either variables but also make comparisons between factors.

1) Logistic Regression: Because logistic regression is scalable to large data sets with several features, it was the starting point for the modeling stage. Since the results of this analysis is expected to be expanded upon in the future, logistic regression is the best model to use for quick analysis which can be iterated as new features are added. The primary parameter included in Python's *scikitlearn* package is *C*, the inverse of regularization strength. Through grid search methods, a value of *C*=10 was deduced to be optimal for high accuracy across most of the datasets with the exception of smaller sample windows which suggested optimal accuracy at weaker regularization up to *C*=100. While quick, this model does not provide adequate insight into potential nonlinear relationships that may exist between the features.

2) Decision Trees: Decision tree classification is suitable for multi-ouput problems, and it requires little data preparation. DTs can handle numerical as well as categorical data. However, it can create over complex trees and can be biased if some classes dominate. Since there are a lot of features, it is possible that there is some overlapping of classes. The model is created with gini index to minimize misclassification, and the branches are created using best split.

3) SVM: SVMs are similar to logistic regression models in that they both scale well with many features; however, SVMs are far superior in capturing the nonlinear relationships between factors. Multiclass support vector classification (SVC) is supported in Python with two default kernels to choose from. Although the linear kernel performed well, the radial based function (RBF) was superior in initial tests. This may be indicative of significant nonlinear relationships within the data. The model is tuned with parameters *C*, the penalty for the error term, at *C*=1000 and γ, the kernel coefficient, at γ=0.0001.

4) K-NN: In contrast to SVM and logistic regression models is k-nearest neighbors. While training efficiency of this model does not scale well with large numbers of features the results are still quite useful. Different sizes for the number of neighbors were tested but *N*=7 using Manhattan distance produced the best results similar to that of the SVM and logistic regression models.

5) GLMnet: GLMnet, also known as ElasticNet, is a variation on logistic regression that includes a component that allows regularization to be tuned smoothly between L_1 (Lasso) and L_2 (Ridge). This makes a GLMnet model particularly well suited to high-dimensional situations where only a subset of predictors are likely to add lift to the final model.

Activities by Factor Level

Factor Level	Activity	Ambulatory
1	Walking	Yes
2	Walking Upstairs	Yes
3	Walking Downstairs	Yes
4	Sitting	No
5	Standing	No
6	Laying	No
7	Standing to Sitting	No
8	Sitting to Standing	No
9	Sitting to Lying	No
10	Lying to Sitting	No
11	Standing to Lying	No
12	Lying to Standing	No

TABLE I

ACTIVITIES INDICATED BY FACTORS FOUND IN *activity_labels.txt*

III. RESULTS

The primary metric of interest for this investigation is accuracy and each model was optimized around it. While secondary metrics such as precision, F-measure (F1 score), and recall were considered, accuracy was preferred because of its ease of applicability in labeled data. Even still, the weighted averages of these secondary metrics are recorded and may provide further insight into the performance between different models. These weighted averages take into consideration of the relative frequency of each activity, as each activity was performed at varying durations. Other factors such as training time are taken into consideration for practical use however not quantified. These metrics are listed on table sets IV, VI, VIII, and X. Note that for multiclass data the accuracy is equal to the average recall. Recall is still a useful metric as it can be separated into its component factors.

Metrics were calculated for ambulatory activities (factor levels 1 to 3) and non-ambulatory activities (4-12). The mean values for each of these are also listed in their respective tables.

IV. Analysis

To determine whether there is a statistically significant difference between the highest-performing combination of frequency and window size we used 10-fold cross-validation to create samples of model performance for each combination and performed a one-sided hypothesis test between the highest-performing sample and each other sample. Since variance generally increased with window size due to correspondingly lower numbers of observations (see table II), we chose Welch's unequal-variance t-test for this purpose. The results can be seen in tables XI through XVIII. Because this is a multiple-comparisons situation we applied the Bonferroni correction to a p-value of 0.05 so that we required a p-value of approximately 0.0026 to achieve significance.

Standard Deviation of Model Accuracy (10-Fold Cross-Validation)
Model: SVM Linear

	Sample Window (s)				
Freq (Hz)	1	2	3	5	10
5	0.004	0.0063	0.0084	0.0076	0.0193
10	0.0042	0.0037	0.0043	0.0092	0.0146
25	0.0025	0.0066	0.0065	0.004	0.0166
50	0.0033	0.0049	0.0051	0.0071	0.0125

TABLE II
Standard Deviation of Models

In general, Logistic Regression and SVM were the highest performing models with K-NN and GLMnet also strong contenders. The decision tree model, while not bad, significantly under-performed compared to the other models. Across all model types, with the exception of SVM with a rbf kernel, increases in sampling frequency increased model performance. SVM-rbf saw peak performance at the 25Hz level. However, SVM-linear, Boosted Logistic and K-NN saw performance at 25Hz that did not differ statistically from peak performance at 50Hz. Regarding sampling windows, each model performed best when the window was between 2 and 3 seconds. The overall best model for accuracy is logistic regression at levels 25Hz and 2s (*acc*=0.9289). However, taking into consideration that increased sampling frequency corresponds with increased energy usage the SVM performed at similar levels at 25Hz and 3s (*acc*=0.9219).

There were also significant differences in metrics between the ambulatory and non-ambulatory activities where ambulatory activities were calculated to have significantly higher scores across all metrics. This is especially apparent in the decision tree model (Table VI) where metrics for ambulatory activities scored at least 0.8328 and non-ambulatory activities scored at most 0.5587. The lowest scores belong to activity groups 7 to 12, where the activities are transitions between standing, sitting, and lying—with the lowest scores associated with the transition to sitting (7: Standing to Sitting and 10: Lying to Sitting).

Logistic Regression
Metric: Accuracy

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.8042	0.8436	0.8505	0.8730	0.8296
10	0.8576	0.8941	0.8976	0.8864	0.8667
25	0.8990	0.9201	0.9249	0.9109	0.8938
50	0.9076	**0.9289**	0.9242	0.9060	0.8988

Logistic Regression
Metric: Weighted Average Precision

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.8098	0.8519	0.8579	0.8765	0.8285
10	0.8642	0.8936	0.9017	0.8880	0.8641
25	0.9004	0.9209	0.9251	0.9111	0.8968
50	0.9078	**0.9295**	0.9275	0.9119	0.9046

Logistic Regression
Metric: Weighted Average F-Measure

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.8022	0.8427	0.8477	0.8727	0.8265
10	0.8562	0.8926	0.8962	0.8844	0.8623
25	0.8977	0.9186	0.9225	0.9095	0.8886
50	0.9065	**0.9281**	0.9231	0.9049	0.8950

Logistic Regression
Metric: Weighted Average Recall

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.8042	0.8436	0.8505	0.8730	0.8296
10	0.8576	0.8941	0.8976	0.8864	0.8667
25	0.8990	0.9201	0.9249	0.9109	0.8938
50	0.9076	**0.9289**	0.9242	0.9060	0.8988

TABLE III

FOUR METRICS FOR LOGISTIC REGRESSION

Decision Tree
Metric: Accuracy

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.7109	0.6863	0.7135	0.7521	0.6321
10	0.7420	0.7466	0.7437	0.7717	0.7728
25	0.7839	0.8098	0.7283	0.7998	0.7062
50	**0.8236**	0.8103	0.7828	0.8083	0.8000

Decision Tree
Metric: Weighted Average Precision

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.7147	0.7058	0.7190	0.7730	0.6397
10	0.7445	0.7690	0.7586	0.7958	0.7832
25	0.7912	0.8105	0.7868	0.8070	0.7412
50	**0.8253**	0.8195	0.7968	0.8187	0.8068

Decision Tree
Metric: Weighted Average F-Measure

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.7115	0.6923	0.7146	0.7598	0.6313
10	0.7419	0.7542	0.7450	0.7795	0.7734
25	0.7858	0.8089	0.7462	0.8009	0.7008
50	**0.8239**	0.8099	0.7847	0.8092	0.7990

Decision Tree
Metric: Weighted Average Recall

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.7109	0.6863	0.7135	0.7521	0.6321
10	0.7420	0.7466	0.7437	0.7717	0.7728
25	0.7839	0.8098	0.7283	0.7998	0.7062
50	**0.8236**	0.8103	0.7828	0.8083	0.8000

TABLE V

FOUR METRICS FOR DECISION TREE

Logistic Regression
Frequency: 50Hz, Sample Window: 2s

Activity	Metric Precision	F-Measure	Recall
1	0.9662	0.9727	0.9795
2	0.9801	0.9641	0.9486
3	0.9495	0.9687	0.9887
Ambulatory	0.9653	0.9685	0.9723
4	0.9156	0.9097	0.9038
5	0.9098	0.9161	0.9224
6	0.9732	0.9820	0.9909
7	0.7200	0.6792	0.6429
8	0.9375	0.8571	0.7895
9	0.4516	0.5091	0.5833
10	0.7308	0.7170	0.7037
11	0.7857	0.6027	0.4889
12	0.7000	0.7636	0.8400
Non-Ambulatory	0.7916	0.7707	0.7628

TABLE IV

LOGISTIC REGRESSION: C=10

Decision Tree
Frequency: 50Hz, Sample Window: 1s

Activity	Metric Precision	F-Measure	Recall
1	0.8486	0.8361	0.8240
2	0.8577	0.8249	0.7945
3	0.8134	0.8454	0.8800
Ambulatory	0.8399	0.8355	0.8328
4	0.8394	0.8369	0.8343
5	0.8394	0.8429	0.8465
6	0.9801	0.9780	0.9759
7	0.2500	0.2857	0.3333
8	0.4839	0.3896	0.3261
9	0.4137	0.3902	0.3692
10	0.3898	0.3898	0.3898
11	0.3721	0.4051	0.4444
12	0.4308	0.4667	0.5091
Non-Ambulatory	0.5555	0.5539	0.5587

TABLE VI

DECISION TREE

SVM

Metric: Accuracy

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.8236	0.8593	0.8513	0.8632	0.8247
10	0.8647	0.8956	0.8969	0.8779	0.8469
25	0.9118	0.9186	**0.9219**	0.9072	0.8519
50	0.9140	0.9196	0.9161	0.9035	0.8765

SVM

Metric: Weighted Average Precision

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.8221	0.8610	0.8534	0.8634	0.8107
10	0.8661	0.8962	0.8983	0.8791	0.8427
25	0.9101	0.9186	**0.9220**	0.9083	0.8495
50	0.9117	0.9187	0.9177	0.9050	0.8780

SVM

Metric: Weighted Average F-Measure

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.8215	0.8575	0.8485	0.8608	0.8152
10	0.8627	0.8944	0.8947	0.8748	0.8398
25	0.9100	0.9169	**0.9192**	0.9045	0.8434
50	0.9118	0.9179	0.9152	0.9019	0.8702

SVM

Metric: Weighted Average Recall

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.8236	0.8593	0.8513	0.8632	0.8247
10	0.8647	0.8956	0.8969	0.8779	0.8469
25	0.9118	0.9186	**0.9219**	0.9072	0.8519
50	0.9140	0.9196	0.9161	0.9035	0.8765

TABLE VII

FOUR METRICS FOR SVM

SVM

Frequency: 25Hz, Sample Window: 3s

Activity	Metric Precision	F-Measure	Recall
1	0.9333	0.9561	0.9800
2	0.9684	0.9558	0.9436
3	0.9590	0.9689	0.9791
Ambulatory	0.9534	0.9603	0.9676
4	0.9078	0.9235	0.9397
5	0.9103	0.9142	0.9181
6	0.9829	0.9850	0.9871
7	0.9000	0.7826	0.6923
8	1.0000	0.6364	0.4667
9	0.3846	0.2702	0.2083
10	0.6667	0.7272	0.8000
11	0.3871	0.4615	0.5714
12	0.8750	0.6829	0.5600
Non-Ambulatory	0.7794	0.7093	0.6826

TABLE VIII

SVM: C=1000, γ=0.0001, KERNEL=RBF

K-Nearest Neighbors

Metric: Accuracy

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.7086	0.7956	0.8122	0.8376	0.7704
10	0.8047	0.8691	0.8667	0.8523	0.7901
25	0.8770	0.8897	0.8962	0.8559	0.7852
50	0.8723	**0.9044**	0.8918	0.8608	0.8074

K-Nearest Neighbors

Metric: Weighted Average Precision

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.7159	0.8016	0.8179	0.8369	0.7634
10	0.8107	0.8715	0.8661	0.8497	0.7888
25	0.8755	0.8886	0.8913	0.8482	0.7693
50	0.8737	**0.9021**	0.8865	0.8598	0.7951

K-Nearest Neighbors

Metric: Weighted Average F-Measure

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.7035	0.7885	0.8063	0.8332	0.7574
10	0.8006	0.8655	0.8593	0.8441	0.7759
25	0.8728	0.8846	0.8885	0.8437	0.7622
50	0.8690	**0.9009**	0.8857	0.8528	0.7930

K-Nearest Neighbors

Metric: Weighted Average Recall

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.7086	0.7956	0.8122	0.8376	0.7704
10	0.8047	0.8691	0.8667	0.8523	0.7901
25	0.8770	0.8897	0.8962	0.8559	0.7852
50	0.8723	**0.9044**	0.8918	0.8608	0.8074

TABLE IX

FOUR METRICS FOR K-NEAREST NEIGHBORS

K-Nearest Neighbors

Frequency: 50Hz, Sample Window: 2s

Activity	Metric Precision	F-Measure	Recall
1	0.9320	0.9584	0.9863
2	0.9199	0.9213	0.9228
3	0.9520	0.9609	0.9699
Ambulatory	0.9346	0.9469	0.9597
4	0.9043	0.8911	0.8782
5	0.8590	0.8844	0.9114
6	0.9701	0.9774	0.9848
7	0.8750	0.6364	0.5000
8	0.6923	0.5625	0.4737
9	0.3793	0.4151	0.4583
10	0.7692	0.7547	0.7407
11	0.6923	0.5070	0.4000
12	0.7059	0.5714	0.4800
Non-Ambulatory	0.7608	0.6889	0.6475

TABLE X

K-NN: N=7 USING MANHATTAN DISTANCE

Mean Accuracy of Models
Model: SVM Linear

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.8221	0.8347	0.8566	0.8512	0.8284
10	0.8684	0.8946	0.8980	0.8864	0.8654
25	0.9128	0.9187	**0.9213**	0.9093	0.8933
50	0.9152	**0.9265**	**<u>0.9277</u>**	**0.9206**	0.8957

TABLE XI
MEAN ACCURACY (10-FOLD CROSS-VALIDATION)

Welch's Two-Sample t-Test
Model: SVM Linear

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0000	0.0018	0.0134	0.0000	0.0000
50	0.0000	0.3138	NA	0.0102	0.0000

TABLE XII
UNCORRECTED P-VALUES

Mean Accuracy of Models
Model: K-NN

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.7242	0.7586	0.7897	0.8007	0.7570
10	0.7812	0.8251	0.8397	0.8418	0.8040
25	0.8434	0.8703	**0.8720**	0.8599	0.7972
50	0.8548	**0.8733**	**<u>0.8800</u>**	0.8623	0.8004

TABLE XIII
MEAN ACCURACY (10-FOLD CROSS-VALIDATION)

Welch's Two-Sample t-Test
Model: K-NN

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0000	0.0009	0.0097	0.0000	0.0000
50	0.0000	0.0160	NA	0.0000	0.0000

TABLE XIV
UNCORRECTED P-VALUES

Mean Accuracy of Models
Model: LogitBoost

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.8382	0.8599	0.8792	0.8789	0.8682
10	0.8839	0.8986	0.9035	0.9127	0.8832
25	0.9137	**0.9227**	**0.9232**	**0.9204**	0.9046
50	0.9189	**0.9247**	**<u>0.9257</u>**	**0.9218**	0.9077

TABLE XV
MEAN ACCURACY (10-FOLD CROSS-VALIDATION)

Welch's Two-Sample t-Test
Model: LogitBoost

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0001	0.0000
25	0.0000	0.0953	0.1663	0.0409	0.0000
50	0.0000	0.2291	NA	0.0776	0.0001

TABLE XVI
UNCORRECTED P-VALUES

Mean Accuracy of Models
Model: GLMnet

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.7921	0.8191	0.8412	0.8367	0.8255
10	0.8326	0.8532	0.8607	0.8611	0.8467
25	0.8617	**0.8681**	0.8789	0.8741	0.8431
50	0.8629	0.8715	**<u>0.8821</u>**	**0.8696**	0.8599

TABLE XVII
MEAN ACCURACY (10-FOLD CROSS-VALIDATION)

Welch's Two-Sample t-Test
Model: GLMnet

Freq (Hz)	Sample Window (s) 1	2	3	5	10
5	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0000	0.0000	0.1193	0.0004	0.0000
50	0.0000	0.0000	NA	0.0029	0.0002

TABLE XVIII
UNCORRECTED P-VALUES

V. DISCUSSION

To further advance this research, tests must be run to measure energy usage of hardware. For the majority of the results of this study, increased sampling rates resulted in an increase in accuracy, although the difference between 25Hz and 50Hz was typically not significant, indicating diminishing returns with increased sampling rate. This supports Bieber et al's assertion[7] that 32Hz should be sufficient to model human activity.

With more information on energy usage, the metrics derived from these models can be scaled appropriately in order to determine the optimal sampling frequency and window on a case-by-case basis. Both linear and rbf kernel based SVMs performed most other models by a moderate margin, but more sophisticated models were not tested in this study and there may be further gains to be achieved by other models. Nevertheless the relationship between frequency, window size, and model performance was consistent across the models we tested.

One point of interest arises from the low accuracy scores for the non-ambulatory activities, in particular those which involve movement to the sitting position. It is unknown whether or not this is due to the positioning of the hardware itself on the participant causing noisy data or if these activities are inherently difficult to predict based on the provided features. Other HAR studies [5] have gathered data at frequencies as high as 100Hz but have not recorded as many different activities. Since these models cap out at the maximum of 50Hz, further research could be performed on data at even higher frequencies, if possible. However since we see diminishing returns from 25Hz to 50Hz it is doubtful whether a further increased sampling rate would be beneficial. One caveat regarding significance testing is that since our samples were not independent, the Bonferroni correction may be overly conservative. A more precise correction to the multiple comparisons problem might reveal significance levels that we did not achieve and therefore reveal an advantage to higher sampling rates.

Another point of interest in our data is that for window sizes greater than 3s the models we trained showed diminished accuracy. While it is possibly that this is a direct result of the reduced count of observations generated by our resampling of the raw data (from 2,220 up to 22,427 rows), we speculate that the longer windows can contain a greater variety of motion within a given activity type and this may result in a noisier signal.

While this study focused on a particular dataset, the code is robust enough to work with any accelerometer and gyroscope data which is similar to UCIs research. Provided more time, the code could be optimized for performance and also generalized to accept readings from other hardware sources as well.

REFERENCES

[1] UCI HAR Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set
http://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions

[2] GitHub Repository
https://github.com/seanmullane/DS6999_HAR

[3] Scikitlearn package in Python for Machine Learning
http://scikit-learn.org/stable/

[4] Caret package in R for Machine Learning
http://topepo.github.io/caret/index.html

[5] Niazi, Anzah Hayat Khan. A Study in Human Activity Recognition: Hierarchical Classification and Statistical Analysis. B.Eng, National University of Science and Technology, Islamabad, Pakistan, 2013.
https://www.ai.uga.edu/sites/default/files/theses/niazi_anzah_h_201608_ms.pdf

[6] Reyes-Ortiz, Jorge-L., et al. Transition-Aware Human Activity Recognition Using Smartphones. Neurocomputing, vol. 171, 2016, pp. 754767., doi:10.1016/j.neucom.2015.07.085.
https://www.sciencedirect.com/science/article/pii/S0925231215010930

[7] Gerald Beiber, Jrg Voskamp, and Bodo Urban. Activity recognition for everyday life on mobile phones. International Conference on Universal Access in Human-Computer Interaction, pages 289296, 2009.