Offline Machine Learning for Human Activity Recognition with Smartphone

Yanjia Zhang
Department of Mathematics & Statistics
University of South Florida
Tampa, FL, USA
Email: yzhang17@usf.edu

Kandethody M. Ramachandran
Department of Mathematics & Statistics
University of South Florida
Tampa, FL, USA
Email: ram@usf.edu (Corresponding author)

***Abstract*—Wearable sensors in smart-phone and wrist tracking devices are widely used in the activity tracking and body monitoring with a low cost. Human activity recognition (HAR) is one of the important applications. Activities identification then is the core part of HAR. In this report, we present a comparison with several popular offline machine learning methods using smartphone data and try to find the most effective model for analyzing these sensor data. The methods include Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Linear Discriminant Analysis (LDA), Decision Tree (DT), Random Forest (RF), and Artificial Neural Network (ANN). Two datasets with different transition methods from smartphone are used. The data includes Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing, Lying, and Jogging. The first dataset has the first 6 activities and shows that the SVM with linear kernel and LDA have the highest test average accuracy of 96.4% and 96.2%, respectively. Decision Tree performs worse than others with test average accuracy of 86.0%. While the second dataset excludes the Lying, but has jogging, and shows that LDA and DT are the most appropriate algorithms, the test average accuracy of these are 100%. KNN is the worse one with 74.3% test average accuracy. Based on all the results, LDA might be the best one for these sensors data. Moreover, the transition method used to reduce noise and extra information in the second data might be better than that in the first one. It has lower dimensions and better classification performance. In order to get these improved accuracy rates, in this paper, we used grid search, multi-fold cross validation, and dimensional reduction method. In addition to just doing the comparison, we also proposed a two-layer method for activity identification. This method is more flexible of choosing classifiers for activities.**

I. Introduction

Human activity recognition (HAR) with wearable sensor is an area that focuses on automatically identifying human activities based on transmitted sensor data. Since wearable devices present a convenient and noninvasive way to record physiological data from users with reduced manual intervention and a low cost, HAR has been successfully applied to several areas. For example, it is frequently used for health monitoring, sport training, and recreational activities recording. Based on the sensors in the smartphone, such as accelerometer and gyroscope, movements are identified or grouped into the sequences of activities with a fixed time window transition. For example, walking, standing, walking stairs, sitting and lying. Many machine learning algorithms have been used over years, including supervised or semi-supervised ways.

The two main classification methodologies are applied to the data. The first one is parametric, such as the multivariate linear regression, Bayesian methods, etc. Some additional conditions are usually needed to meet the assumptions, such as normality and non-collinearity. These assumptions are often violated with the sensor data, as Figure 1 shows the shape from the sensor coordinates, they are not following the normality assumption. This means that these methods will lose power to do classifications. The second one is non-parametric, which is distributions free and has less assumptions, such as Support Vector Machine, K-Nearest Neighbor, Neural Network and non-parametric multiplicative regression. These methods are more suitable for dealing with the data that we do not know the relationship between the response and the independence variables or the shape among variables. Considering the large size of the sensor data, the high dimension and the uncertainty of the relationship between variables, we tend to use non-parametric methods to create models and evaluate the prediction on the test data.

The data used in this report are downloaded from UCI Machine Learning website [1] and the WISDM dataset [2] which is available in public domain. The UCI data experiments were carried out with 30 volunteers with ages 19 to 48 years old. During the experiment, the volunteers were

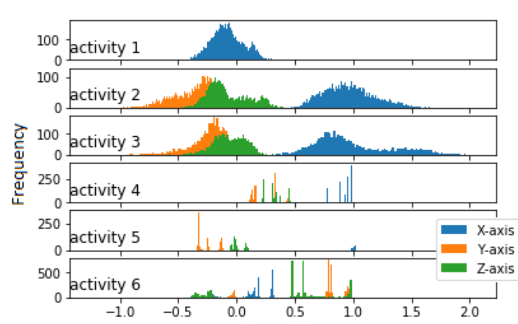

Fig. 1. Histogram for Activities

TABLE I
TOTAL DATA STRUCTURE SUMMARY

	Walk	Up	Down	Sit	Stand	Lay	Jog
UCI (%)	16.5	14.8	13.5	17.3	19.0	18.8	
WISDM (%)	38.4	11.7	9.8	5.7	4.6		30.0

wearing a Samsung Galaxy S II on the waist. The experiment collected the data in 3-axial linear acceleration and angular velocity at a constant rate of 50HZ. The data was modified by applying several filters and same specific hertz rates to remove the noise. After this, a Fast Fourier Transform (FFT) was applied to these signal data. Finally, there are a vector of 561 features in each record. They performed six basic activities, walking, walking-upstairs, walking-downstairs, sitting, standing, and lying. The first three are called dynamic activities and the rest are called static activities. In brief, this data includes $10,411$ total number of observations, Table I shows the details of the percentages for each activity. The subjects were randomly selected into two groups, 70% of them in the training group and the rest in the testing group. In this case, we have 7352 records for training data and 2947 for testing data.

From the data, we have the sequences of the coordinate's streams, as shown in Figure 2. It is easy to notice that there are big differences between dynamic and static activities. Sensor coordinates values vary with dynamic activities while static ones have more flat lines. The bottom row in Figure 2 shows the activities, 1,2, and 3 represent the dynamic activities walking, upstairs, and downstairs, and 4,5, and 6 represent the static ones, sitting, standing, and lying.

The WISDM dataset [2] had 36 volunteers with Android-based smartphones in their front pants pockets and they were asked to perform 6 activities for specific periods of time under monitoring, including walking, jogging, walking upstairs, walking downstairs, sitting, and standing. The data was recorded with 20Hz, lower than the UCI data (50Hz). The researcher used arffmagic program to transfer the raw data

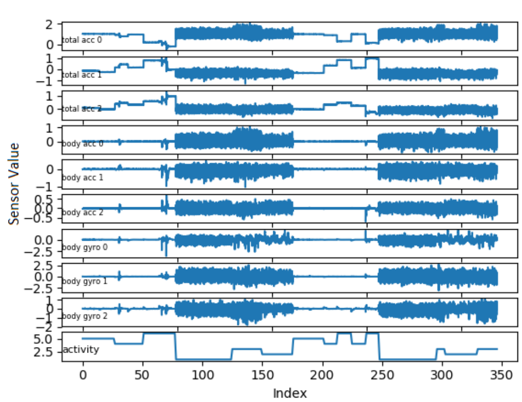

Fig. 2. Coordinates Values of Activities

with a 10 second window size to new data with 43 features. These features include the difference between maximum and minimum, average sensor value, time between peaks, standard deviation, variance, and average resultant acceleration. Since the average sensor value of X-axis is 0, we remove this feature in the analysis. In brief, this data includes 5,424 total number of observations. The percentage of each activity is shown in table I. As we did for UCI data, we split training and test data by 70% and 30% of the subjects, respectively. Since the researchers did not ask all the volunteers to perform all 6 activities, some subjects might have less activities.

II. RELATED WORKS

The work of human activity recognition based on the sensors can be traced back to 1990s [3]. Sharma, Lee, and Chuang[4] applied neural networks (ANN) for a chest worn wireless sensor dataset and achieved 83.95% accuracy. Wu [6] used K Nearest Neighbors (KNN) as the best classifier with iPod Touch data, but the results show that it fails to effectively classify similar activities as well. Anguita [7] used 561 transformed features to classify six different activities using a one vs. all support vector machine (SVM) and obtained as high as 89% accuracy. Kwapisz, Weiss, and Moore [2] from the WISDM Lab used Multilayer Perceptron and they got a best accuracy of 91.7%. Zhang, Wu, and Luo [8] point out that the combination of the Hidden Markov Model and the Deep Neural Network (HMM-DNN) has a higher accuracy compared with Gaussian mixture method, Random Forest, and their combination with HMM. The accuracy of HMM-DNN is 93.5%. Guo, Liu, and Chen [9] performed a two layer and multi-strategy framework for sensor smartphone data and the result shows a 95.71% average accuracy. Besides, Ronao ad Cho [10] applied deep learning neural networks (DNN) to both raw sensor data and Fast Fourier Transformed smartphone data. Their work shows that the data with the transformed information provides average accuracy rate of 95.75%, which is 1% higher than the results from the raw data. Nakano and Chakraborty [11] point out that the convolutional neural network (CNN) has better performance in identifying dynamic activities than other methods. The average accuracy is 98% with classifying walking, walking upstairs and walking downstairs. Ignatov [12] used CNN for the accelerometer data from smartphone. They obtain a 97.63% average accuracy with the statistical features. Besides, there are also some online works, which treat the data as streaming. Na, Ramachandran, and Ji [13] used the Online Bayesian Kernel Segmentation method for classifying 6 activities. The result shows a 92% average accuracy rate with the new segmentation data instead of fixed window data. In the paper, she first did the segmentation for new windows and then applied filters with these windows. Zhang and Ramachandran [14] used Very Fast Decision Tree method for online classification with the original window transformed data. The result shows that the overall accuracy 85.9%. The advantage of this online method is that after inputting a user's own data to the model from lab data, the new model will be more personal. Since not all the data and features are the same, it is hard to compare which method is better than others. But the challenge for most of the methods is the difficulty in discriminating between similar activities, especially for sitting and standing, walking upstairs and walking downstairs.

III. Methods

In this paper we compare performance of some of the most popular machine learning methods for the smartphone based data, including Support Vector Machine with three different kernels, K-Nearest Neighbor with different number of neighbors, Artificial Neural Network with 1 and 2 layers with different numbers of neurons, Linear Discriminant Analysis. Also, we compare these methods with the dimension reduction through Principle Components Analysis. For Support Vector Machine (SVM), Decision Tree and Random Forest methods, we used the grid search and 5-folds cross-validation to find the best hyper-parameters, since these parameters in the model cannot be estimated from the training data.

SVM is an algorithm that finds classification boundaries so that categories are divided by a clear gap that is as wide as possible [15]. With the labels, the algorithm will output an optimal hyperplane which categorizes the data into different groups. There are three commonly used kernels, include linear kernel, Radial basis function (rbf) kernel, and polynomial kernel. To satisfy the assumption of SVM, we need to standardize the data before we apply the SVM. The common parameters we need to define are regularization parameter (C), Degree of the polynomial kernel function (degree), Kernel coefficient for rbf and polynomial kernel (gamma). In the experiments, we set C from 0.001 to 10 by 10, degree from 2 to 5 by 1, and gamma from 0.001 to 1 by 10.

K-Nearest Neighbor (KNN) is a non-parametric method that assign the class to a point by taking the majority of votes of its K neighbors [16]. KNN is based on the feature similarity, i.e., the more closely out-of-sample features resemble the training set, the more likely they are to be classified to a certain group. A characteristic of KNN is that it is sensitive to the local structure of the data. We apply KNN here to see if there are any large distances between any two classes. The number of neighbors starts from 5 to 20.

Artificial Neural Network (ANN) extracts linear combinations of the inputs as derived features, and then model the target as a nonlinear function of these features and evolves to encompass a large class of models and learning methods [17]. Figure 3 shows a neural network algorithm with one layer and 5 neurons. Each neuron has an associated weight vector, which is assigned on the basis of its relative importance to the inputs. With the activation function, the neurons output the non-linear results. The advantage of using this method is that ANN has a flexibility to capture the nonlinearities in the data.

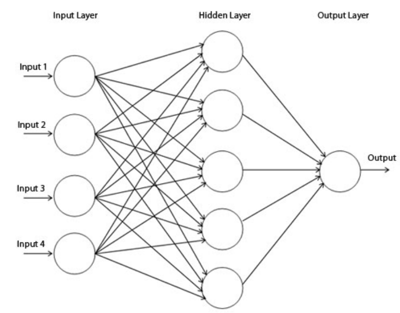

Fig. 3. Single Neural Network

Linear Discriminant Analysis (LDA) is most commonly used as data classifier and dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. This method maximizes the ratio of between-class variance to the within-class variance in any particular data set thereby guaranteeing maximal separability [18]. We use this method because of the high accuracy rate of the linear-SVM.

Decision Tree (DT) and Random Forest (RF) are other two non-linear methods. We try Decision Tree here to see that if the greedy method can find a good cut-point for these continuous variables and select the better variables to do the split. Random Forest (RF) is an ensemble learning method that operates by constructing a large number of decision trees at training time and outputting the class that is the mode of the classes of the individual trees [19]. Since Random Forest do the voting and regardless of the collinearity, we also try this method. The most important parameter for these two are the branch split rules and the stop criteria. We use the Gini index for the measure of split quality because of less computation, the minimum number of samples for split is from 2 to 10 by 1, the maximum depth of the tree is from 3 to 10 by 1, and the number of trees in the forest is from 20 to 100.

Principal Component Analysis (PCA) is a dimensionality-reduction technique that is often used to transform a high-dimensional dataset into a lower dimensional subspace. PCA finds the principal components of the dataset by transforming the data into a new coordinate system. In the new subspace, the first axis corresponds to the first principal component, which is the component that explains the greatest amount of the variance in the data. Considering the UCI data has 561 variables, it is very unlikely that all the variables are independent. To overcome this problem, in this paper we also implemented PCA and analyzed the resulting data. Considering the small number of the features (42) in the WISDM data, we did not apply PCA to it.

TABLE II
NUMBER OF OBSERVATIONS IN TRAINING DATA

Activities	Walk	Up	Down	Sit	Stand	Lay	Jog
UCI	1226	1073	987	1293	1423	1413	
WISDM	1529	447	375	190	170		1200

IV. EXPERIMENTS AND RESULTS

We apply these machine learning methods to 70% of the subjects to train our model with grid searching for best parameters. The number of observations for each activity in training data is shown in Table II. The UCI data has balanced groups, while WISDM data has different training observations for each activity. This is an unbalanced dataset, but here we will conduct the experiment same as if it was a balanced data. As we see from the table, Upstairs, Downstairs, sitting and Standing have less observations in WISDM data than in UCI data.

Since we used grid search for some of the parameters, Table III also shows some of the setting, for example, the SVM with polynomial kernel (Poly-SVM), the best degree for the model is 2, the KNN with 10 neighbors, and the ANN with different layers and units. And we show the accuracy rate of the test data in Table III to compare the performances of the models and make a rough conclusion of the relations between different activities. In general, models with WISDM data perform better than those with UCI data, except the SVM with linear (Linear-SVM) and Poly-SVM and KNN with 10 neighbors. From this table, we can see that LDA has the best performance with both datasets, 96.2% and 100%, for UCI and WISDM data respectively. This means that the different activities exist linear classifiers based on the feature combination. But the best classifier for these two datasets are different, the best performance for UCI dataset is 96.4% from Linear-SVM. This is much better than the result of 89.3% in [7], which also used SVM methods. The reason might be because of the setting of hyper-parameters, which we got from grid search and cross validation. While the average test accuracy for WISDM data is 100% from LDA and DT. This is also better than the result of 91.7% from [2], which used Multilayer Perceptron. This is because we checked the data structure and features and found the feature named average sensor value of X-axis is a constant 0. Thus, we removed it before the analysis. The result shows that better data pre-process might improve the model performance. The DT and RF perform much better with the WISDM data than with UCI data. This might have many reasons. Comparing these two datasets, there are two main differences. Firstly, the way of collecting and transforming of raw sensor data. Secondly, the body locations for the sensors placed. The KNN does not perform as good as others with both datasets. In other words, the Euclidean distance used in KNN might not be suitable for this type of data, which means the variance within each activity might vary. Moreover, this comparison also shows that the neural network does not always "win". It depends on the data type. Besides, increasing the layers and the neurons does not improve the accuracy much in the neural network, which also means that the dataset might have linear relationships among activities. Both linear and non-linear methods can do the perfect identification for WISDM data with a small amount of observations, this might mean that the data captured the characteristics for each activity very well.

TABLE III
METHODS COMPARISON WITH AVERAGE ACCURACY (%)

Methods	UCI	WISDM
Linear-SVM	**96.4**	89.9
RBF-SVM	95.3	97.9
Poly-SVM (2)/(3)	93.7	90.8
KNN (10)	88.5	74.3
LDA	**96.2**	**100**
DT	86.0	**100**
RF	92.5	99.1
One-layer-NN (30)	94.6	97.3
One-layer-NN (50)	94.4	97.3
Two-layer-NN (30, 6)	95.2	98.3

TABLE IV
UCI TEST AVERAGE ACCURACY FOR ACTIVITIES

Methods	Walk	Up	Down	Sit	Stand	Lay
Linear-SVM	0.99	0.97	0.97	0.87	0.97	1.0
RBF-SVM	0.97	0.97	0.92	0.90	0.96	1.0
Poly-SVM (2)	0.99	0.94	0.84	0.89	0.95	1.0
KNN (10)	0.98	0.88	0.73	0.81	0.93	0.94
LDA	0.99	0.98	0.96	0.88	0.96	1.0
DT	0.88	0.78	0.83	0.77	0.88	1.0
RF	0.97	0.90	0.84	0.88	0.94	1.0
One-layer-NN (30)	0.99	0.94	0.93	0.89	0.95	0.97
One-layer-NN (50)	0.99	0.95	0.92	0.87	0.96	0.96
Two-layer-NN (30, 6)	0.99	0.94	0.93	0.88	0.97	1.0

Table IV and Table V show the prediction details from each method. These two tables show that most of the methods can successfully identify Walking, Lying and Jogging with more than 95% accuracy. They are also able to identify the Upstairs and Downstairs with a decent accuracy, around 90%. The biggest challenge is to identify the sitting and standing. These two activities have the similar pattern and very minimum difference with the raw sensor data. Thus, they are difficult to be identified from each other. For example, the result from Linear-SVM for the UCI data shows that, 59 out of 66 misclassified sitting are identified as standing, and all of the misclassified standing are identified as sitting. The same thing to the LDA, 62 out of 63 misclassified sitting cases are identified as standing and all of the 23 misclassified standing are identified as sitting.

Then, we use PCA to reduce the data dimensions for the UCI data. Since the WISDM only has 42 features, PCA is no necessary. Figure 5 shows that the proportion of variance explained by each component is less than 0.1% after the first 40 principle components. And **??** shows that the cumulative proportion of variance explained by the components are close to 1 (99.99978%) after the first 200 principle components.

TABLE V
WISDM Test Average Accuracy for Activities

Methods	Walk	Up	Down	Sit	Stand	Jog
Linear-SVM	1.0	0.54	0.69	0.84	0.97	1.0
RBF-SVM	0.99	0.99	0.99	0.91	0.79	1.0
Poly-SVM (3)	0.96	0.70	0.89	0.68	0.91	1.0
KNN (10)	0.93	0.41	0.31	0.28	0.39	0.98
LDA	1.0	1.0	1.0	1.0	1.0	1.0
DT	1.0	1.0	1.0	1.0	1.0	1.0
RF	1.0	0.99	0.99	0.92	0.97	1.0
One-layer-NN (30)	1.0	0.99	0.99	0.73	0.91	1.0
One-layer-NN (50)	1.0	1.0	0.99	0.75	0.93	1.0
Two-layer-NN (30, 6)	1.0	0.99	0.99	0.84	0.93	1.0

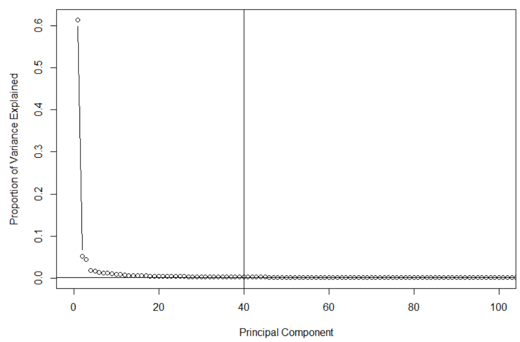

Fig. 4. Proportion of Variance

Then we can have a conclusion that the first 200 principle components are sufficiently explained the data information and we can reduce the data dimension from 561 to 200.

As we can see from Table VI, the PCA method successfully reduces the data dimension to less than half of the original data dimensions without losing the features variance. Again, the linear classifiers have better performance, since the highest accuracy rates are from Linear-SVM with 200 principle components and the LDA with 250 and 300 principle components, with accuracy 96.06%, 96.36%, and 96.53%, respectively. Additionally, Poly-SVM and KNN do not perform as well as others. From this, it is reasonable to think that PCA reduces the data dimensions but does not change the data structure and the relationships among categories. However, there is little improvement for the average accuracy, which still around 96%. In this point of view, PCA reduces the dimensions effectively without loss in performance.

Table IV and Table V also imply that most of the algorithms are good at identify dynamic activities but have worse performance with static ones. To create an algorithm which has more flexibility to adapt different algorithms, we propose a two layers method for UCI data, as shown in Figure 6. By experiments, we selected the LDA as the first layer binary classifier. There are two reasons. First, LDA gives a good result for identifying dynamic and static activities. It has 100%

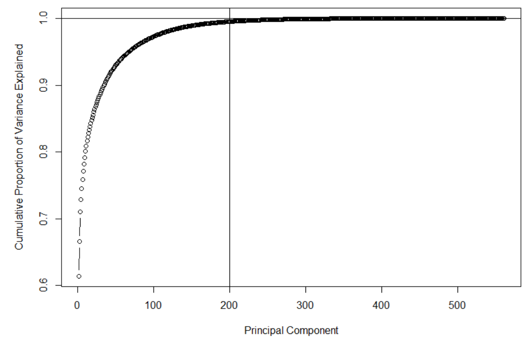

Fig. 5. Cumulative Proportion of Variance

average test accuracy rate with 1 out of 1609 static cases misclassified as dynamic. Secondly, LDA has stable result. All the parameters are from training data, while linear-SVM have hyper-parameters. We used Linear-SVM and Poly-SVM for dynamic and static classification, respectively. These selections were based on the performance of the algorithm with the single group data. Table VII shows that the average test accuracy is improved a little by using different classifiers. The biggest challenge is Sitting and Standing identification. The problem might imply that the transition of these two activities sensor data might not be appropriate. It loses the power to extra characteristic for these two.

V. Conclusion

Comparing results from two different datasets, LDA turns out to be the best choice for both datasets. Thus, we might claim that all of these activities are linearly separated. It is also obvious that the methods perform better with the WISDM data than with UCI data. The reasons are not clear yet. But we might guess that the position of the sensor might be one. UCI data is collected with the smartphone on

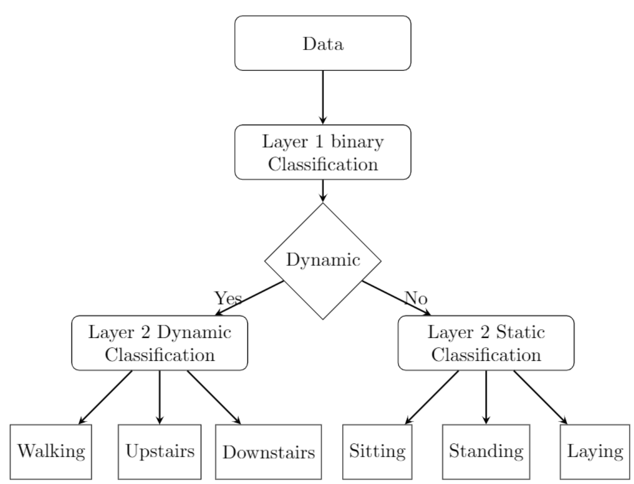

Fig. 6. Two Layers Method Flowchart

TABLE VI
COMPARISON WITH PCA METHODS

Methods	# of Components	Avg. accuracy (%)	Methods	# of Components	Avg.accuracy (%)
Linear-SVM	50	91.49	LDA	100	93.69
	100	94.43		150	95.16
	150	96.06		200	95.96
	200	96.46		250	96.36
	250	96.06		300	96.53
RBF-SVM	100	96.43	KNN(10)	100	89.89
	150	94.73		150	90.02
	200	93.52		200	90.39
Poly-SVM(2)	100	92.59	KNN(20)	100	90.15
	150	92.39		150	90.62
	200	91.32		200	90.45
One-Layer-NN(30)	100	94.55	Two-Layer-NN(30,10)	100	95.45
	150	95.02		150	95.10
	200	95.78		200	96.14

TABLE VII
UCI TEST AVERAGE ACCURACY FOR ACTIVITIES TWO LAYERS METHOD

Methods	Two Layers Method						
Activity	Walk	Up	Down	Sit	Stand	Lay	Recall %
Walk	**492**	2	2	0	0	0	99
Up	20	**451**	0	0	0	0	96
Down	4	9	**407**	0	0	0	97
Sit	0	1	0	**464**	43	0	91
Stand	0	0	0	19	**537**	0	97
Lay	0	0	0	2	0	**543**	100
Precision%	95	97	100	96	93	100	**96.6**

waist while WISDM data is collected from the front pants pockets. The leg movements might be more sensitive to the sensors than that on the waist. Another guess is the way of collecting and transforming the raw data. UCI data had higher frequency 50Hz and first applied median and butterworth filters to reduce noise and used FFT to transform the signal data, while WISDM lab collected in 20Hz and used the original signal data with arffmagic program. Based on this result, we might have a conclusion that for human activities recognition with smartphone sensors, the arffmagic program is better that the FFT and might be in the lab experiment, noise reduction might not be necessary. Besides, without resorting to complicated methods, simple LDA is sufficient for data analysis.

REFERENCES

[1] UCI Machine Learning Web `https://archive.ics.uci.edu/ml/datasets`

[2] J. R. Kwapisz, G. M. Weiss and S. A. Moore. *Activity Recognition Using Cell Phone Accelerometers*. Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data (at KDD-10), Washington DC,2012

[3] F. Friedrich, M. Smeja, and J. Fahrenberg. *Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring.* Computers in human behavior 15.5 (1999): 571-583.

[4] A. Sharma, Y. Lee, and W. Chung. *High accuracy human activity monitoring using neural network*. 2008 Third International Conference on Convergence and Hybrid Information Technology. Vol. 1. IEEE, 2008.

[5] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. *Cell phone-based biometric identification*. 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS). IEEE, 2010.

[6] W. Wu, et al. *Classification accuracies of physical activities using smartphone motion sensors*. Journal of medical Internet research 14.5 (2012).

[7] D. Anguita, et al. *Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine*. International workshop on ambient assisted living. Springer, Berlin, Heidelberg, 2012.

[8] L. Zhang, X. Wu, and D. Luo. *Human activity recognition with HMM-DNN model*. 2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI* CC). IEEE, 2015.

[9] Q. Guo, B. Liu, and C. Chen. *A two-layer and multi-strategy framework for human activity recognition using smartphone*. 2016 IEEE International Conference on Communications (Icc). IEEE, 2016.

[10] C. Ronao, and S. Cho. *Human activity recognition with smartphone sensors using deep learning neural networks*. Expert systems with applications 59 (2016): 235-244.

[11] K. Nakano, and B. Chakraborty. *Effect of dynamic feature for human activity recognition using smartphone sensors*. 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST). IEEE, 2017.

[12] A. Ignatov. *Real-time human activity recognition from accelerometer data using Convolutional Neural Networks*. Applied Soft Computing 62 (2018): 915-922.

[13] S. Na, K. M. Ramachandran, and M. Ji. *Real time Activity Recognition using Smartphone Accelerometer*.

[14] Y. Zhang, and K. M. Ramachandran. *Human Activity Recognition with Streaming Smartphone Data*. 2019 Global Conference for Advancement in Technology (GCAT). IEEE, 2019.

[15] C. Cortes, and V. Vapnik. *Support-vector networks*. Machine learning 20.3 (1995): 273-297.

[16] N. S. Altman. *An introduction to kernel and nearest-neighbor nonparametric regression*. The American Statistician 46.3 (1992): 175-185.

[17] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*. Vol. 1. New York: Springer series in statistics, 2001.

[18] J. Ye, R. Janardan, and Q. Li. *Two-dimensional linear discriminant analysis*. Advances in neural information processing systems. 2005.

[19] T. K. Ho. *Random decision forests*. Document analysis and recognition, 1995., proceedings of the third international conference on. Vol. 1. IEEE, 1995.