Approaches of Human Activity Recognition Using Smartphones

1. Introduction

Human Activity Recognition (HAR) aims to identify the actions carried out by a person given a set of observations of him/herself and the surrounding environment. Recognition can be accomplished by exploiting the information retrieved from various sources such as environmental [1] or body-worn sensors [2, 3]. Smartphones are bringing up new research opportunities for human-centered applications where the user is a rich source of context information and the phone is the firsthand sensing tool.

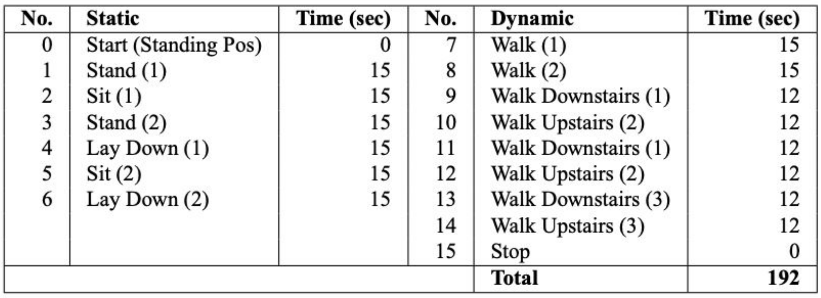
In this project, the data from sensors is collected to indicate six activities. In order to distinguish these activities, I implement several approaches of HAR refering the paper from Anguita et al.[4]. The methods of SVM and Random Forest are applied on the dataset. Moreover, I address the Principal Component Analysis (PCA) to reduce the dimension of dataset.

All accuracy results of classification among these approaches are above 0.97, which are better than the results from original paper a little bit. The classifiers can tell the difference between movement and static status, but they still make ambiguity among dynamic and non-dynamic actions, e.g. there are some false positive values among “walking”, “walking_donwstairs” and “walking_upstairs”.

2. Data Collection and Preparation

2.1 Source of dataset:

Experiments were carried out with a group of 30 volunteers within an age bracket of 19-48 years while wearing a smartphone (Samsung Galaxy) on the waist. Each person performed six activities: “Walking”, “Walking upstairs”, “Walking downstairs”, “Sitting”, “Standing”, and “Laying”. Table 1 shows experiment protocol details.



No.	Static	Time (sec)	No.	Dynamic	Time (sec)
0	Start (Standing Pos)	0	7	Walk (1)	15
1	Stand (1)	15	8	Walk (2)	15
2	Sit (1)	15	9	Walk Downstairs (1)	12
3	Stand (2)	15	10	Walk Upstairs (2)	12
4	Lay Down (1)	15	11	Walk Downstairs (1)	12
5	Sit (2)	15	12	Walk Upstairs (2)	12
6	Lay Down (2)	15	13	Walk Downstairs (3)	12
			14	Walk Upstairs (3)	12
			15	Stop	0
				Total	**192**

Table 1: Protocol of activities for the HAR Experiment.

2.2 Total number of records, number of variables

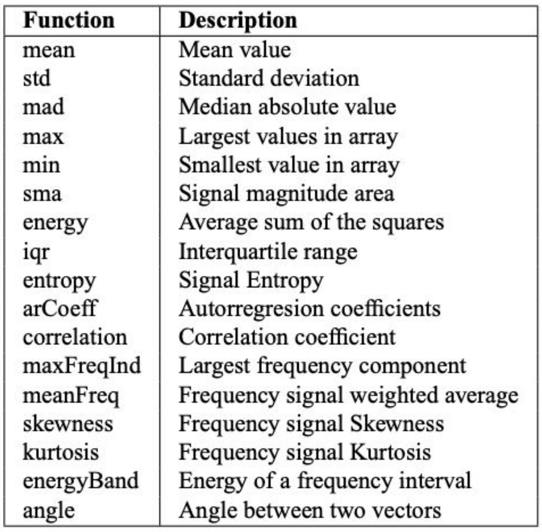
The data collected consists of 10299 records, representing six activities observations of 30 volunteers. Each observation corresponds to a 2.5s segment from a volunteer.
A total of 561 different features are extracted to describe each activity window from the raw accelerometer and gyroscope signals.

Columns	Features	Data type
1-561	Values of signals from sensors	Float
562	Index of subject	Integer
563	Label of activity	object

Table 2: Features of dataset

2.3 Feature Extraction

According to the original paper, besides the standard measures used in HAR literature [11] such as the mean, correlation, signal magnitude area (SMA) and autoregression coefficients [12] were employed for the feature mapping, the set of features was also employed in order to improve the learning performance, including energy of different frequency bands, frequency skewness, and angle between vectors (e.g. mean body acceleration and y vector). Table 3 contains the list of all the measures applied to the time and frequency domain signals.



Function	Description
mean	Mean value
std	Standard deviation
mad	Median absolute value
max	Largest values in array
min	Smallest value in array
sma	Signal magnitude area
energy	Average sum of the squares
iqr	Interquartile range
entropy	Signal Entropy
arCoeff	Autorregresion coefficients
correlation	Correlation coefficient
maxFreqInd	Largest frequency component
meanFreq	Frequency signal weighted average
skewness	Frequency signal Skewness
kurtosis	Frequency signal Kurtosis
energyBand	Energy of a frequency interval
angle	Angle between two vectors

Table 3: List of measures for computing feature vectors.

In this dataset, it already contains extracted features from the raw signal data. Therefore there isn't any works about feature extraction.

2.4 Data preprocessing

To implement the multiclass classification, I map the categorical value of “Activity” into the numerical value. Table 4 shows the corresponding relationship between two data types.

Activity	Corresponding labels
Laying	0
Sitting	1
Standing	2
Walking	3
Walking downstairs	4
Walking upstairs	5

Table 4: Corresponding relationship of labels

2.5 Split into training and testing data sets

Since the purpose of this project is to predict the human activity, we should ignore the action from specific person, therefore I drop the feature of “subject” and remain 561 features.

In order to train and predict models, the dataset has been randomly partitioned into two independent sets, where 70% of the data were selected for training and the remaining 30% for testing.

3. Exploratory Analysis

This dataset has wonderful quality, there isn’t any missing or None value, and all values are normalized between -1 and 1, which is easily learning for the models.

The observations are broken down into six activity groups equally, the data is balanced. Table 5 and Figure 1 show the equality of each activity.

Activity	Count number
Laying	1944
Sitting	1777
Standing	1906
Walking	1722
Walking downstairs	1406
Walking upstairs	1544

Table 5: Count number of each activity

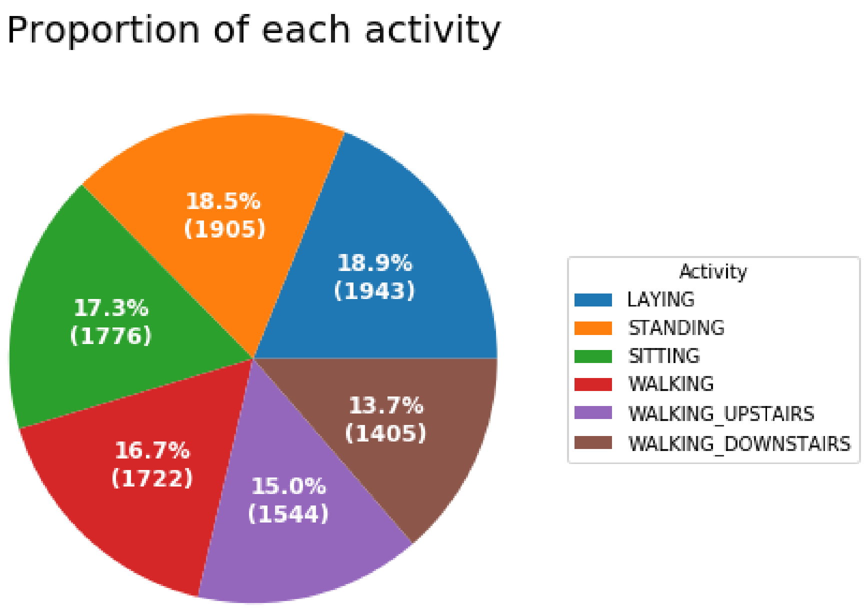


Figure 1: Proportion of each activity

4. Methods

4.1 Statistical / Machine learning methods described

Since the referred approach [4] used the Support Vector Machine (SVM) to implement classification, I decide to use it again and compare the results. Furthermore, I apply another popular logistic regression method “Random Forest” in this project.

3.1.1 Support Vector Machine

In machine learning, support-vector machines (SVMs, also support-vector networks[5]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall.

Different from the simple binary classification, this dataset has a high dimensional feature space. In 1992, Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik suggested a way to create nonlinear classifiers by applying the kernel trick (originally proposed by Aizerman et al.[6]) to maximum-margin hyperplanes.[7] The resulting algorithm is formally similar, except that every dot product is replaced by a nonlinear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The transformation may be nonlinear and the transformed space high-dimensional; although the classifier is a hyperplane in the transformed feature space, it may be nonlinear in the original input space. Some common kernels are “Polynomial”, “Gaussian”, and “Sigmoid”. I use these three kernels and choose the best result.

4.1.2 Random Forest

Random forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.[8][9] Random decision forests correct for decision trees' habit of overfitting to their training set [10].

4.2 Apply and evaluate two models

“Sci-kit learn (sklearn)” is one of the most popular library for machine learning. It contains not only training methods, but also evaluation tools.

To assess the performance of models, I use “Accuracy” to evaluate the whole performance, besides, for classification, I usually introduce the confusion matrix and f1-score to evaluate the model more precisely.

4.2.1 SVM models

Firstly, I apply the SVM with “Polynomial”, “Gaussian”, and “Sigmoid” kernels on dataset and get the evaluation of them.

From the Table 6 we can see that the best accuracy of kernel “Polynomial” is as high as 0.99. The accuracy of kernel “Gaussian” is 0.969, which is a pretty good result too. The worst one is kernel “Sigmoid”, the accuracy is only 0.859.

Kernel	**Polynomial**	**Gaussian**	**Sigmoid**
Accuracy	0.99	0.969	0.859

Table 6: Accuracy of each SVM kernel

Kernel	**Activity**	**Precision**	**Recall**	**F1-score**	**Support**
Polynomial	Laying	1.00	1.00	1.00	563
	Sitting	0.98	0.97	0.98	555
	Standing	0.98	0.98	0.98	567
	Walking	1.00	0.99	1.00	529
	Walking downstairs	0.99	1.00	0.99	425
	Walking upstairs	1.00	0.99	1.00	451
Gaussian	Laying	1.00	1.00	1.00	563

	Sitting	0.93	0.93	0.93	555
	Standing	0.93	0.93	0.93	567
	Walking	0.99	1.00	0.99	529
	Walking downstairs	0.98	0.99	0.98	425
	Walking upstairs	0.99	0.98	0.98	451
Sigmoid	Laying	0.97	1.00	0.99	563
	Sitting	0.98	0.46	0.62	555
	Standing	0.66	0.98	0.79	567
	Walking	0.91	0.93	0.92	529
	Walking downstairs	0.91	0.83	0.86	425
	Walking upstairs	0.91	0.96	0.93	451

Table 7: Evaluation summary of SVM

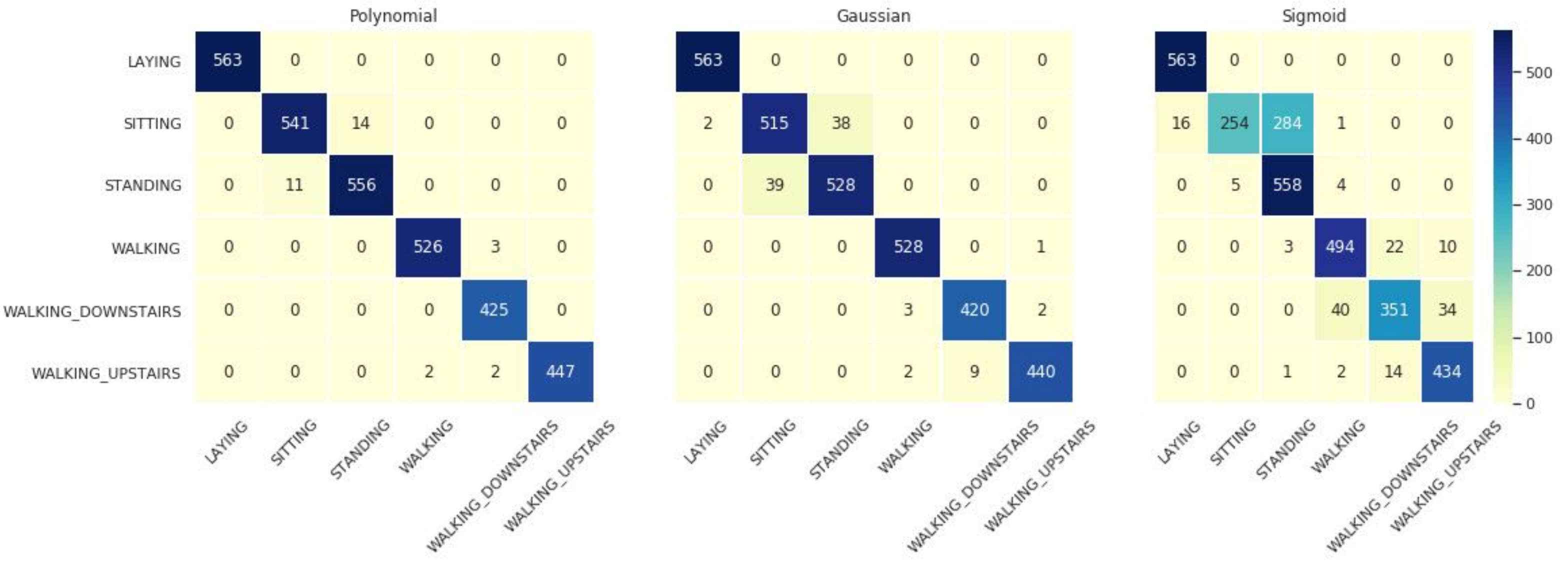


Figure 2: The confusion matrix of three kernels

From Table 7 and Figure 2, we can conclude that both kernels of “Polynomial” and “Gaussian” perform excellent on predicting “Laying” and “Walking”, they only make limited mistakes on distinguish the status of movement and stable, e.g., there are a couple of mis-classification between “Sitting” and “Standing”.

For the kernel “Sigmoid”, it can’t tell the status of “Sitting” from “Standing”, more than half of “Sitting” are recognized as “Standing”. This kernel also make a few mistake at recognizing “Walking downstairs” and “Walking upstairs”.

Based on the evaluation, I believe both the kernels of “Polynomial” and “Gaussian” work well in this classification job.

4.2.2 Model of Random Forest

After training and predicting, the evaluation of Random Forest is as good as SVM. The total accuray is 0.975. Table 8 shows the summary of evaluation.

Model	Activity	Precision	Recall	F1-score	Support
Random Forest	Laying	1.00	1.00	1.00	563
	Sitting	0.96	0.95	0.96	555
	Standing	0.95	0.96	0.96	567
	Walking	0.99	0.98	0.99	529
	Walking downstairs	0.97	0.97	0.97	425
	Walking upstairs	0.97	0.98	0.97	451

Table 8: Evaluation summary of Random Forest

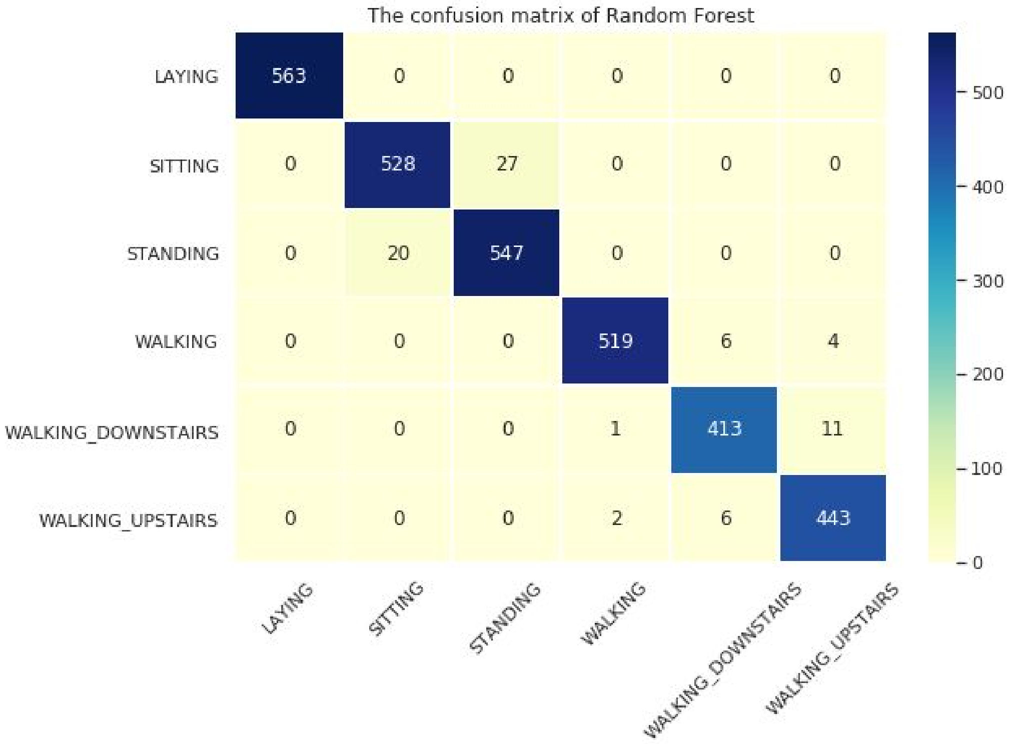


Figure 3: The confusion matrix of Random Forest

Similar to SVM, the confusion matrix of Random Forest shows the model performs well at recognizing “Laying” and “Walking” whereas makes several mistakes at distinguishing “Sitting” and “Standing”.

4.2.3 Tuning models
To get better result, I implement some fine tuning methods to adjust the hyperparameters of the model. Since the result of “Sigmoid” kernel is terrible, I only tune the other two kernels.

After tuning, the accuracy of “Gaussian” kernel is promoted to 0.989, which has an improvement of 2.06%, the mistakes of “Sitting” and “Standing” are less than before. For the other kernel of “Polynomial”, the accuracy is still 0.99 even I adjust the parameter of “degree”, and the mistakes values are less than previous very slightly, it seems get the ceiling of classification. Table 9 shows the summary after tuning.

Kernel	Activity	Precision	Recall	F1-score	Support
Polynomial	Laying	1.00	1.00	1.00	563
	Sitting	0.98	0.97	0.98	555
	Standing	0.98	0.98	0.98	567
	Walking	1.00	0.99	1.00	529
	Walking downstairs	0.99	1.00	1.00	425
	Walking upstairs	1.00	0.99	1.00	451
Gaussian	Laying	1.00	1.00	1.00	563
	Sitting	0.97	0.97	0.97	555
	Standing	0.98	0.97	0.97	567
	Walking	1.00	1.00	1.00	529
	Walking downstairs	1.00	1.00	1.00	425
	Walking upstairs	1.00	0.99	1.00	451

Table 9: Evaluation summary of SVM after fine tuning

For another model, Random Forest, the accuracy increases to 0.98, 0.005pp higher than the default result. Table 10 shows the summary after fine tuning.

Model	Activity	Precision	Recall	F1-score	Support
Random Forest	Laying	1.00	1.00	1.00	563
	Sitting	0.97	0.96	0.97	555
	Standing	0.96	0.98	0.97	567

	Walking	1.00	0.98	0.99	529
	Walking downstairs	0.97	0.98	0.98	425
	Walking upstairs	0.97	0.99	0.98	451

Table 10: Evaluation summary of Random Forest after fine tuning

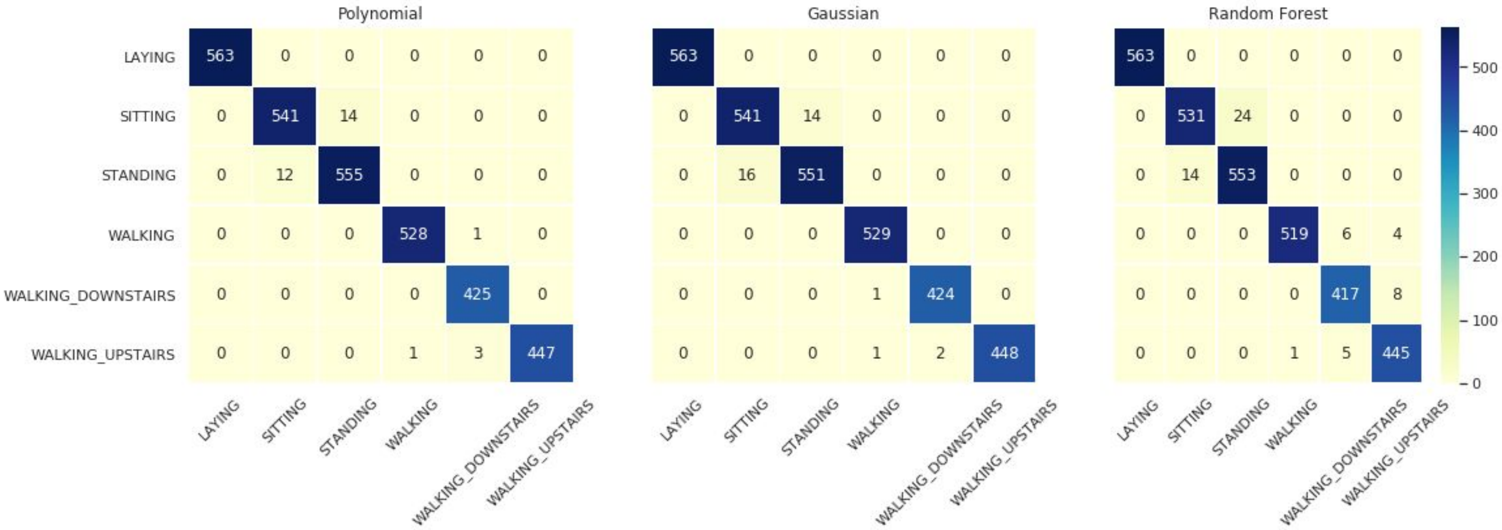


Figure 4: Confusion matrix of models after fine tuning

Overall, the best three models for predicting HAR in this project are SVM with “Polynomial” and “Gaussian” kernels and the Random Forest. Figure 4 shows the confusion matrix of these three models after fine tuning. We can see that all models are better than previous, the ambiguous mistakes arranged around “Sitting” and “Standing”

4.3 Feature Importance and correlation

The dataset contains 561 features, it causes high complexity of models due to high dimension. Moreover, it’s impossible that all features have same importance.

If we want to reduce the dimension of dataset, a good start is to know the importance of each feature. I extract the feature importance by a method from Random Forest library, and find that first 220 features contain 90% importance. Figure 5 shows top 10 features.

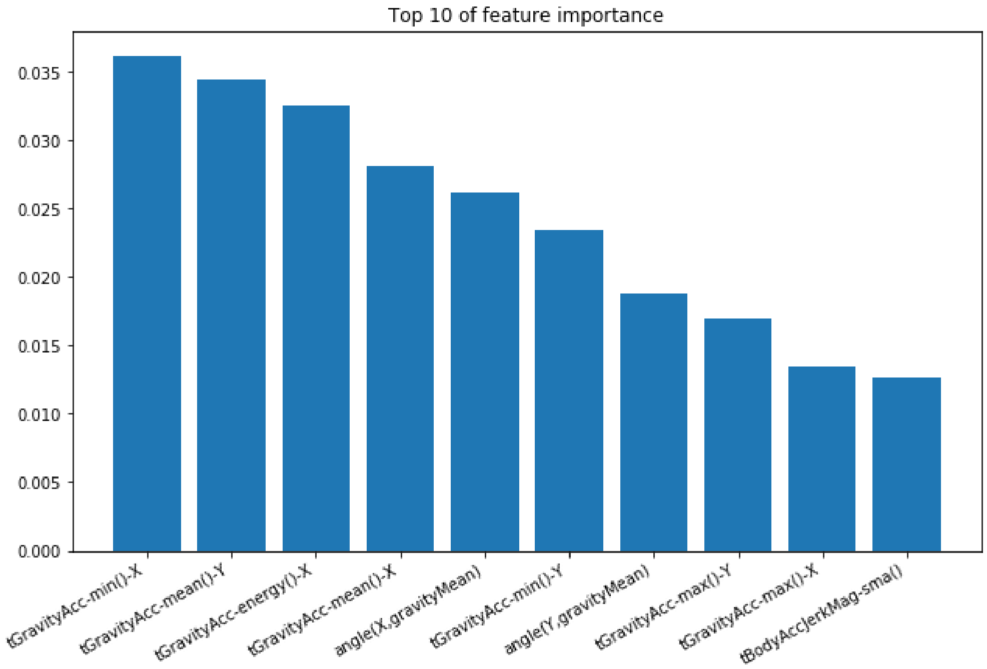


Figure 5: Top 10 of feature importance

If we only keep the first 220 features, we can reduce half of features and remain 90% complement of information.

There is a problem during choosing features, even we've already reduced the features to 220, there are still a lot of features are correlated each other. Figure 6 shows the correlation among top 10 features.

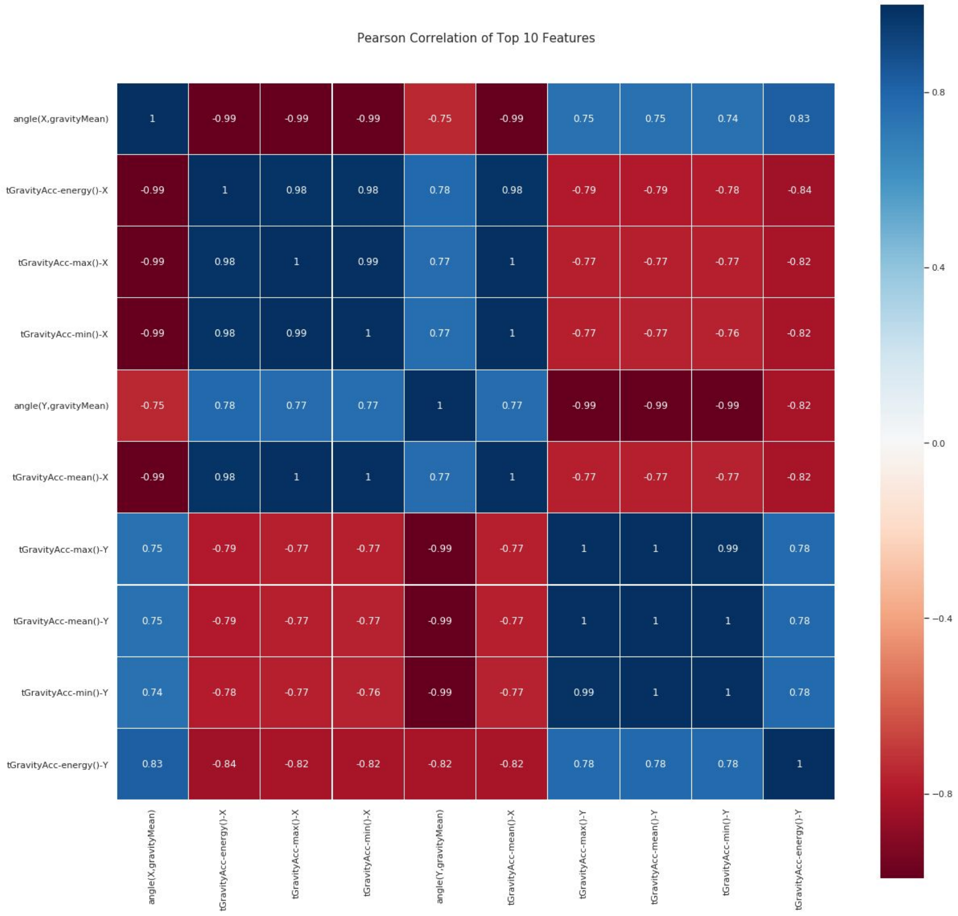


Figure 6: Pearson Correlation of Top 10 Features

From Figure 6 we can see there are so many features are correlated either positive or negative, therefore I omit the features which have more than 95% correlation with others and there are 76 features left.

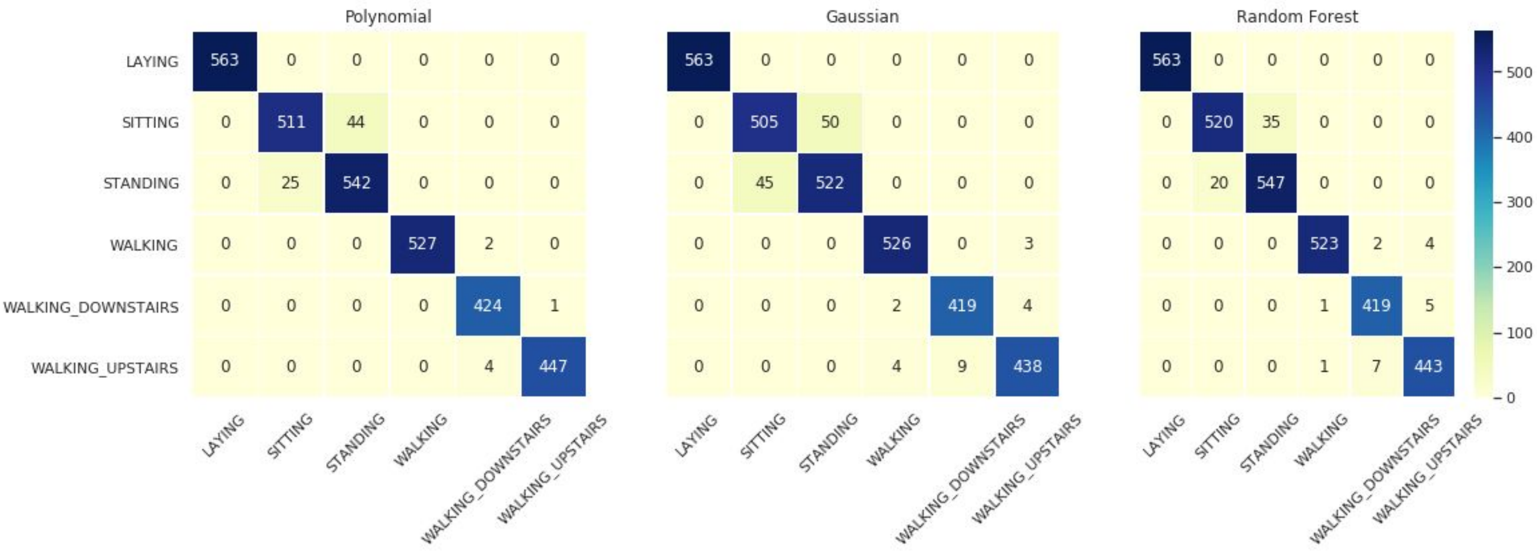
4.4 Applying models on reduced dataset

After applying the same models on the reduced dataset, we get the evaluation of each model. Table 11 shows the accuracy and summary of each model.

Model	Accuracy	Activity	Precision	Recall	F1-score	Support
SVM with Polynomial kernel	0.975	Laying	1.00	1.00	1.00	563
		Sitting	0.95	0.92	0.94	555
		Standing	0.92	0.96	0.94	567
		Walking	1.00	1.00	1.00	529
		Walking downstairs	0.99	1.00	0.99	425
		Walking upstairs	1.00	0.99	0.99	451

SVM with Gaussian kernel	0.962	Laying	1.00	1.00	1.00	563
		Sitting	0.92	0.91	0.91	555
		Standing	0.91	0.92	0.92	567
		Walking	0.99	0.99	0.99	529
		Walking downstairs	0.98	0.99	0.98	425
		Walking upstairs	0.98	0.97	0.98	451
Random Forest	0.976	Laying	1.00	1.00	1.00	563
		Sitting	0.96	0.94	0.95	555
		Standing	0.94	0.96	0.95	567
		Walking	1.00	0.99	0.99	529
		Walking downstairs	0.98	0.99	0.98	425
		Walking upstairs	0.98	0.98	0.98	451

Table 11 Accuracy and summary of reduced dataset


Polynomial

	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	563	0	0	0	0	0
SITTING	0	511	44	0	0	0
STANDING	0	25	542	0	0	0
WALKING	0	0	0	527	2	0
WALKING_DOWNSTAIRS	0	0	0	0	424	1
WALKING_UPSTAIRS	0	0	0	0	4	447

Gaussian

	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	563	0	0	0	0	0
SITTING	0	505	50	0	0	0
STANDING	0	45	522	0	0	0
WALKING	0	0	0	526	0	3
WALKING_DOWNSTAIRS	0	0	0	2	419	4
WALKING_UPSTAIRS	0	0	0	4	9	438

Random Forest

	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	563	0	0	0	0	0
SITTING	0	520	35	0	0	0
STANDING	0	20	547	0	0	0
WALKING	0	0	0	523	2	4
WALKING_DOWNSTAIRS	0	0	0	1	419	5
WALKING_UPSTAIRS	0	0	0	1	7	443

Figure 7: Confusion matrix of models after reducing dataset

From Table 11 and Figure 7 we can see that the quality of classification with reduced dataset decreases 1.5-3pp due to missing information. Although there is a tradeoff to choose computational resource or accuracy, I think we can get the benefit from rapid training time and just loss little accuracy.

Objective 2: (40%) feature importance
For one of the chosen models,

a)Vary the number of features used in the prediction (e.g. from a few to 561), and compute the resulting accuracy
b)Determine the number of features required to obtain 80% , 90% accuracy

4.5 Applying PCA to reduce the dimension of dataset

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. If a multivariate dataset is visualised as a set of coordinates in a high-dimensional data space (1 axis per variable), PCA can supply the user with a lower-dimensional picture, a projection of this object when viewed from its most informative viewpoint. This is done by using only the first few principal components so that the dimensionality of the transformed data is reduced.

We can easily implement PCA by Python, I make a set of number of principal components and evaluate the accuracy of them.

I should note that after dimensionality reduction, there usually isn't a particular meaning assigned to each principal component. The new components are just the main dimensions of variation.

4.5.1 Plot accuracy vs number of principal components

After implementing the PCA with three models, the accuracy of model of Random Forest is greater than others. If the proportion of principal components exceed 30%, the accuracy holds at the highest level. The accuracy of SVM with Polynomial kernel is so interesting that as the increasing of number of principal components, it shows the trend of decreasing.

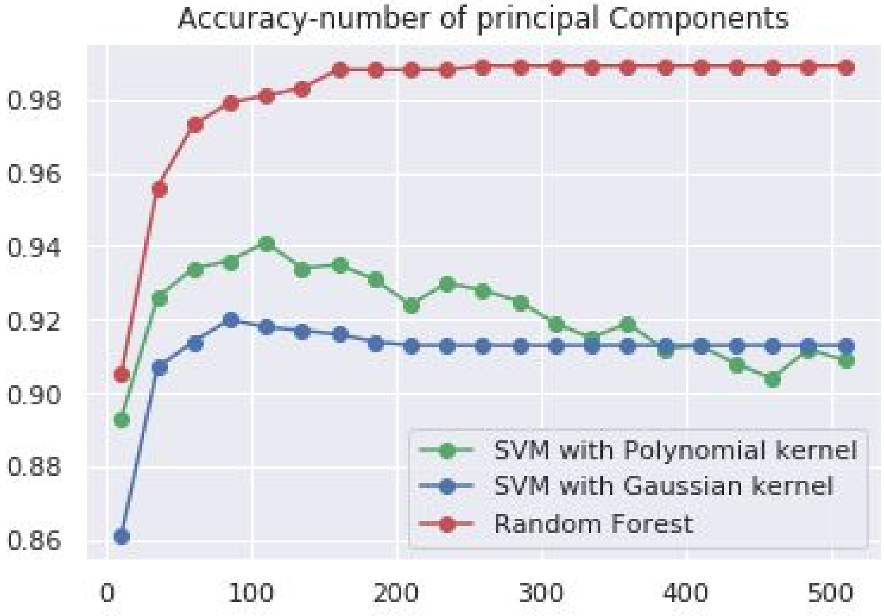


Figure 8: Accuracy vs number of principal components

4.5.2 Determine minimum number of PCA

To determine minimum number of PCA components required to obtain 90% accuracy, I make a “for loop” to iterate the number of components from 1 to 50 by the step of 1. For Random Forest, the minimum number is 9; for SVM with Polynomial kernel, the minimum number is 12; and for SVM with Gaussian kernel, the minimum number is 27.

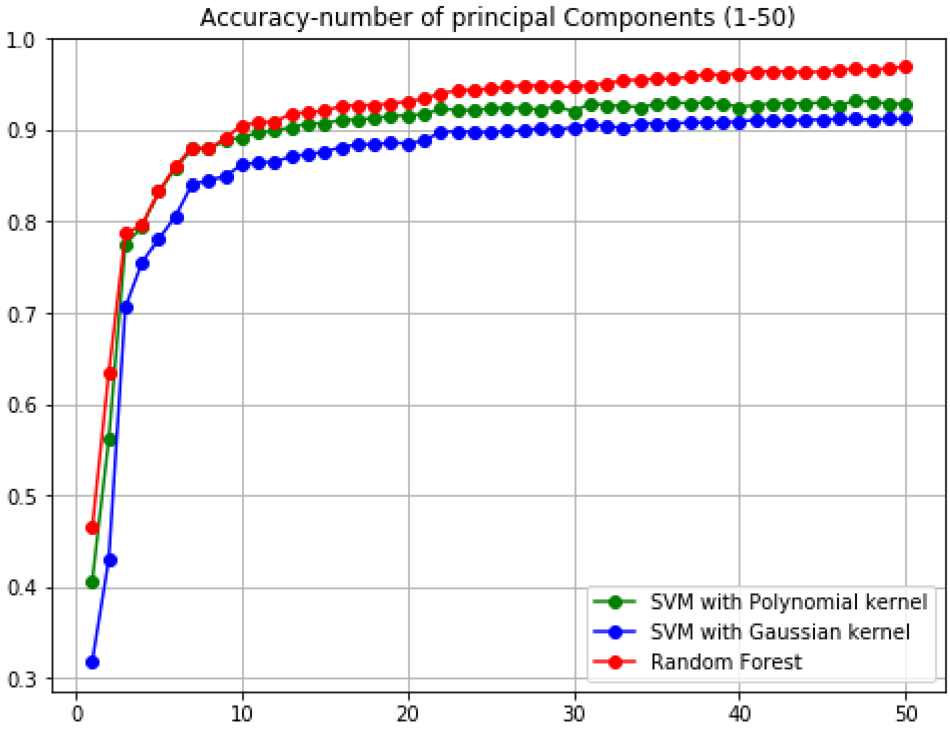


Figure 9: Accuracy vs number of principal component (1-50)

5. Results (40 %)

In this project, I apply SVM with three kernels and Random Forest on the dataset to predict the HAR. After fine tuning, the classification accuracy of SVM with Polynomial and Gaussian kernel and Random Forest is 98%-99%, which are better than the results from original paper a little bit.

From the perspective of confusion matrix, all models perform excellent at distinguishing movement from static status, and there isn’t any false positive between these two states. however, they always make a couple of mistakes to tell the difference between “Sitting” and “Standing”.

Then I implement two approaches to reduce the dimension of dataset. For the first method, I sort the features by their importance, and keep 90% of information complement. There are so many correlated features, I omit features of greater than 95% correlation and the dataset is reduced to 76 features. The accuracy of three models is lower than it of full dataset slightly.

The second method of reducing dimension is PCA. Comparing to the previous method, the effectiveness is worse, especially for SVM, the accuracy is 6% lower. Among the accuracy of three models after implementing PCA, the best one is Random Forest.

If we want to achieve the accuracy of 90%, the minimum number of principal component will be around 10-30.

Overall, these models can make excellent classification of HAR, even work on the dataset with compressed dimension.

References

[1] R. Poppe. Vision-based human motion analysis: An overview. Computer Vision and Image Understanding, 108(1-2):4–18, 2007.

[2] P. Lukowicz, J.A. Ward, H. Junker, M. Stager, G. Tr ¨ oster, A. Atrash, and T. Starner. Recognizing workshop activity using body worn microphones and accelerometers. Proceedings of the 2nd Int Conference Pervasive Computing, pages 18–22, 2004.

[3] D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell, and B.G. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE Transactions on Information Technology in Biomedicine, 10(1):156–167, 2006.

[4] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

[5] Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks". Machine Learning. 20 (3): 273–297. CiteSeerX 10.1.1.15.9362. doi:10.1007/BF00994018.

[6] Aizerman, Mark A.; Braverman, Emmanuel M. & Rozonoer, Lev I. (1964). "Theoretical foundations of the potential function method in pattern recognition learning". Automation and Remote Control. 25: 821–837.

[7] Boser, Bernhard E.; Guyon, Isabelle M.; Vapnik, Vladimir N. (1992). "A training algorithm for optimal margin classifiers". Proceedings of the fifth annual workshop on Computational learning theory – COLT '92. p. 144. CiteSeerX 10.1.1.21.3818. doi:10.1145/130385.130401. ISBN 978-0897914970.

[8] Ho, Tin Kam (1995). Random Decision Forests. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. Archived from the original on 17 April 2016. Retrieved 5 June 2016.

[9] Ho TK (1998). "The Random Subspace Method for Constructing Decision Forests" . IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (8): 832–844. doi: 10.1109 /34.709601.

[10] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2008). The Elements of Statistical Learning (2nd ed.). Springer. ISBN 0-387-95284-5.

[11] J.Y. Yang, J.S. Wang, and Y.P. Chen. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. Pattern recognition letters, 29(16):2213– 2220, 2008.

[12] A.M. Khan, Y.-K. Lee, S.Y. Lee, and T.-S. Kim. Human activity recognition via an accelerometerenabled-smartphone using kernel discriminant analysis. In Proceedings of the 5th International Conference on Future Information Technology, pages 1–6, 2010.