

Analysis and Effect of Feature Selection with Generic Classifier over Activity Recognition Dataset for Daily Life Activity Recognition

Supratip Ghose

Faculty, Dept. of CSE

University of Information Technology & Sciences
Pragati Sarani, Baridha View, Dhaka Bangladesh

ABSTRACT

In recent years, the convergence in sensing environments and pervasive computing has opened a scope for human activity recognition research. In this paper, we form populations from smartphone accelerometer and gyroscope sensor training-based data for Human Activity Recognition (HAR). In the same vein, we described in this work an Activity Recognition database, assembled from the recorded activities of 24 subjects doing Activities of Daily Living (ADL). The paper then aggregated this time-series data into features and subsets of features were selected using two filters based, classifier-independent feature selection methods. We used 10-fold cross-validation strategy to validate the experimentation. Evaluation of the variation of generic decision tree classifiers showed that the feature subsets bring forth acceptable performances than classification with the entire feature set resulting in productive computer overhead in the reduced feature subset. Therefore, classifier-independence feature set should be useful for developing and improving HAR systems across and within populations.

General Terms

Activity Recognition, Data Mining, Classification, Sensors.

Keywords

Feature Selection, activity recognition, classification model, accelerometer, sensors.

1. INTRODUCTION

The Activity Recognition umbrella approaches aim to infer from the sensor data the actions performed by a person regarding a set of observations of him/herself and the surrounding environment. Recent achievements in the field of machine learning and data mining have enabled activity recognition research using smart homes sensing data to play a direct role in improving the quality of life. With these trends, the motion sensors in the form Accelerometry is currently the most exploited technology in physical activity monitoring [7]. Physical activity means either static postures, such as standing, sitting, lying or dynamic motions, such as walking, running, stair climbing [13]. Human activity recognition (HAR) [6], deploying wearable sensor in the form of accelerometer and gyroscope can retrieve possible information from the body postures and movements. Automatically recognizing human activities in an experimental setting leads many applications in areas such as smart pervasive environments in [1, 2, 7] and healthcare [5]. Smartphones with accelerometer and gyroscope sensors are a convenient, minimally invasive, low-cost approach for mobility monitoring. Apparently, we divided this work into typical three principal components [4] to implement the activity recognition task using supervised learning. In the low-

level sensing module, we first collected accelerometer data from twenty-four users. The system lets the user performing activities such as strolling, running, ascending, descending, Relaxing (sitting inhaling), and Relaxing (standing exhaling). In the feature selection and processing module, we then aggregated this raw time series accelerometer data into examples using WEKA data mining tools [17]. We provide an in-depth research analysis that could benefit from mining sensor data from these powerful mobile devices and to employ these data through feature selection and machine learning related to HAR. Activity monitoring systems using sensors embedded in modern smartphones (accelerometers, gyroscopes, etc.) avoid the need for wearable components in user body. Therefore, such systems are non-intrusive and have high acceptance in everyday life. To help discriminate between activities and feature selection and classification for predicting activities, we use WEKA as a machine learning tools for data mining. WEKA is a machine learning workbench that supports many activities of machine learning practitioners, and it can be easily extended or customized with new classifiers, clusters, attribute selection methods and other components.

Android platform supports different broad categories of Accelerometry in the form Android-based gadgets and provides API for application development. We procured a substantial portion of the data for our research through a triaxial motion sensor Accelerometer [4, 8]. These particular types of accelerometer return a real-valued stream of tuples of real numbers acceleration along the x, y and z-axes from which, the open-source Android API determined velocity and displacement. In this paper, we identified trainable window data (extracted features) by windowing an activity signal and fed into HAR using smartphones. This setting allows us to collect various populations independent of the classifier. This objective was achieved by examining a diverse dataset, using two different populations in a feature subset selection module independent of the classification module. Identifying feature subsets that improve activity classification will improve mobility monitoring models for use in future classifiers.

Recognizing activities based on sensor reading is challenging because sensor data are inherently noisy, and human activities appear in a non-deterministic fashion. A critical issue is to develop appropriate activity models that map low-level sensor features to high-level concepts. Probabilistic models gain more popularity as sensor reading are usually noisy, and human activities show stochastic nature. Typically, static classification and temporal classification are the two forms of Probabilities model. In static classification, typical static classifiers first extract a variety of features from sensor readings, and then activity classification happens with a static classifier. Interesting works reported in the literature over the

years, concerned, among other aspects of training-based generic classifiers like Naïve Bayes [3], decision tree [4,11,13]. Moreover developments such as K-nearest neighbor (kNN) used in [3, 8, 15], and Support Vector Machine (SVM) used in [8] are popular. The HAR signal processing involves the acquisition, representation of sensor data, segmentation (windowing) for time frame representation of sensor data, dimensionality reduction (feature selection), and classification [7]. Features are application dependent data from data window, usually calculated by a process of windowing in the form of signal magnitude, area, etc. from signal steps or full acceleration [8]. HAR approaches exist in the literature as a package of the limited sampling of sensor data, preprocessing techniques, defined feature subset and classification algorithms and feature set and post-processing result. Therefore, extracting meaningful information to guide HAR algorithm development is complicated. Wyatt et al. [14] built their activity models by a bootstrap method, in, the accuracy may drop apparently with the inclusion of activity model in real-world scenarios. In contrast, our sensing module segmented data with for every user with 10-second example duration that does not eventually lead to a drop in accuracy. Consequently, in our system, four out of six activities show accuracy over 90%. Moreover, regarding hardware implementation they use a multimodal sensor, whereas, we have not used that extensive architecture.

The existing observed methodologies in the literature of Activity recognition datasets serve to propose machine learning algorithms and choose the optimal one for having relatively better results for their particular domain. Cho et. al. [6] deployed an embedded image sensor in single tri-axial accelerometer, worn at the user's waist, to identify nine activities. However, Multi-sensor approaches leads to incorporation of multi-type phenomena into devices. Our approach from existing literature revolves around the idea that an accelerometer the sensor of main interest in this paper is needed to recognize most daily activities. Thus, our method offers a practical implementation for the collection of the dataset in different dimensions. Also, the characterizing aspect of the study is to throw light on the effect of feature selection. Therefore, to identify feature subset as the motion or activity occurs to different dimensions of the data in real time with mobility monitoring is the objective [10]. The system should be facilitated in such way so as to choose the informative optimal feature subset for the distinct feature spaces with large samples of datasets in HAR. Having justified our approach for the practically practical application, in next few sections, the rest of this paper appears as follows. We describe our activity model and describe the process for addressing the activity recognition task, including data collection. Next, in section 2, we go on for data preprocessing, and data transformation to start with experimentation with the test bed data and with several known supervised learning algorithms. Toward the end, in Section 3, we conclude the work and write our future aspirations with the system.

2. DATA COLLECTION AND FEATURE CONSTRUCTION

In this study, we take into account six body movements: strolling, running, going up stairs, going down stairs, Relaxing (sitting inhaling), and Relaxing (standing exhaling). We selected these movements, because, most of these actions involve repetitive motions, and we believe this should also make the actions easier to recognize. Preprocessing, windowing, feature extraction were the carefully handcrafted setting that was required for the subjects carrying an Android-

based smartphone for the purpose of the data collection. Subsequently, it was controlled by a small Android application [20]. This application, provides a simple graphical user interface for recording the user's name, start-stop the data collection, and label the activity being performed [16]. The motion sensor devices (Android Operating System based mobile) generates data in every millisecond. In all cases, we performed data acquisition for windowing with 5 minutes keeping in mind the occlusions and interference of sensors. This act gives us 30 thousand examples of raw data for an activity performed by each subject. While collecting test data, we address this bias problem by using subjects completely unaffiliated with any of the researchers, as such, he may not perform an activity in a manner favoring the recognition system. An accelerometer—tied with the body pinpoints the projection along its sensitive axis of the special force f exerted. Next, each reading in those 30000 examples contained a stream of tuples of real numbers representing the acceleration in x, y and z-direction with 60 Hz. This window the Acquisition results in three-component vectors of samples stored in *matrix B* the

$$B_i = [X Y Z]$$

There are two types of acceleration components. Firstly, the linear acceleration component a , due to body motion, and secondly, the gravitational acceleration component, g —both projected along the sensitive axis of the accelerometer and results in the special force. In common parlance, the zero-frequency (DC) component relates the low-frequency component of the acceleration signal while the AC component relates the high-frequency component of the signal [15]. The DC component takes care of the static posture while the counterpart does activities with a dynamic motion like walking, shaking head. The feature variables of interest in the existing literature constructed from the raw sensor data within sliding windows with finite and constant width are called data frames. In our paper, the construction of features is specific to problems and popular among practitioners observed in the literature. The signal average from the data samples in each frame measures the DC component of the feature vector. The DC component of the acceleration thus forms the body postures by the gravity director. The variance constitutes the average of the squared samples within each frame. The signal energy and distribution of the frequency domain are other popular choices. We extract features from the data window in the form of the frequency of steps or overall acceleration as these measurements help us to distinguish between running, strolling, and standing. Our experiments consist of two major tasks: Data Preprocessing and Feature Selection. Both tasks are described follow:

3. EXPERIMENTATION

3.1 Data Preprocessing

Data preprocessing technique involves transforming raw, real world data often characterized by incomplete, inconsistent, error-prone and lacking in certain behaviors or trends into an understandable format. We solicit to prepare the data with the objective of Data preprocessing with the input data for discriminant classification state. To transform the collected raw data into discrete weighted average value in X, Y and Z direction format, next few subsections talks on process segmentation of the acceleration data.

```

1 @relation ARMD_final
2
3 @attribute "UNIQUE_ID" numeric
4 @attribute "user" ("33",
5 @attribute "X0" numeric
6 @attribute "X1" numeric
7 @attribute "X2" numeric
8 @attribute "X3" numeric
9 @attribute "X4" numeric
10 @attribute "X5" numeric
11 @attribute "X6" numeric
12 @attribute "X7" numeric
13 @attribute "X8" numeric
14 @attribute "X9" numeric
15 @attribute "Y0" numeric
16 @attribute "Y1" numeric
17 @attribute "Y2" numeric
18 @attribute "Y3" numeric
19 @attribute "Y4" numeric
20 @attribute "Y5" numeric
21 @attribute "Y6" numeric
22 @attribute "Y7" numeric
23 @attribute "Y8" numeric
24 @attribute "Y9" numeric
25 @attribute "Z0" numeric
26 @attribute "Z1" numeric
27
28 @attribute "Z3" numeric
29 @attribute "Z4" numeric
30 @attribute "Z5" numeric
31 @attribute "Z6" numeric
32 @attribute "Z7" numeric
33 @attribute "Z8" numeric
34 @attribute "Z9" numeric
35 @attribute "XAVG" numeric
36 @attribute "YAVG" numeric
37 @attribute "ZAVG" numeric
38 @attribute "XPEAK" numeric
39 @attribute "YPEAK" numeric
40 @attribute "ZPEAK" numeric
41 @attribute "XABSOLDEV" numeric
42 @attribute "YABSOLDEV" numeric
43 @attribute "ZABSOLDEV" numeric
44 @attribute "XSTANDDEV" numeric
45 @attribute "YSTANDDEV" numeric
46 @attribute "ZSTANDDEV" numeric
47 @attribute "RESULTANT" numeric
48 @attribute class( "Strolling", "Running", "A
49
50 @data
51 1,33,0.04,0.09,0.14,0.12,0.11,0.1,0.08,0.13,0.
52 2,33,0.12,0.12,0.06,0.07,0.11,0.1,0.11,0.09,0.
53 3,33,0.14,0.09,0.11,0.09,0.09,0.11,0.12,0.08,0.

```

Fig 1: Feature construction in our dataset

To collect data for every 10-second segment, we divide the collected 30 thousand raw data into three equal bins. Each of these bins contains approximately 1000 of raw data samples.

Firstly, we added three temporary attributes (bin_col_x, bin_col_y, bin_col_z) for every axis. Taking into account of these attributes, Firstly, we determine the achievable range of values for each axis (maximum-minimum). We framed the range into ten equal frequency bins with the attributes and then estimated what fraction of values fell within each of the bins. Secondly, we generated the following attributes.

We generated a total of forty summary features, although these are all variants of just six basic features. These features are:

Binned Discretization: we here divide the range into ten equal bins. Later the other features are taken as the Mean value of the accelerations: Average acceleration value distributed at three axes, Standard Deviation of the acceleration, Average absolute difference records the duration of action, the Average resultant acceleration due to RMS of the acceleration value that is recorded in Figure 1.

3.1.1 Binned Discretization

The range of maximum and minimum value for equal binning were chosen. Subsequently, we divided this range into ten equal size bins. Here, the value of accelerometers has three variants with average acceleration values distributed at the three axes, and average orientation pointed at three directions give thirty attributes for three axes (ten bins for each axis).

3.1.2 Mean Value of the acceleration

This mean value of accelerometers has three variants with average acceleration values distributed at the three axes, and average orientation pointed in three directions. It has three attributes corresponding to XAVG, YAVG, and ZAVG respectively. With average acceleration values distributed at the three axes and average orientation pointed at three directions give thirty attributes for three axes (ten bins for each axis).

3.1.3 Standard Deviation

It is the standard deviation of each of the acceleration and orientation values within the duration. There are three features in this of this type. A counterpart of data variance helps discriminate between presence and absence of motion. Standard Deviation

3.1.4 Average absolute difference

This kind of feature variant is defined as the difference of readings between action duration defined in section 3 and the mean value over those readings for each axes. There are three features of this feature variant.

3.1.5 Average Resultant Acceleration

This feature is the outcome due to the variance. It is deduced by taking square roots of the sum of the values of each axis squared

$$\sqrt{x_i^2 + y_i^2 + z_i^2}$$

Moreover, the making an average. There is one feature corresponding this feature variant.

The In our system, each activity accompanies a different number of examples generated for each user. These variations are due to the physical limitations of spent time for each action of each user and correspondingly impacts time. Another feature we consider is the length between successive peaks. This feature is important as it characterizes the periodic patterns of repeated waves in motion for the substantial period and in this situation activities are easier to detect. Then simple heuristic constraints the offset to pick the minimum peaks on each axis that satisfy the offset value. We define the offset as a comprehensive measure as the third quartile of the highest peaks represented in the sample duration. In the case of the absence of offset, the offset can be lowered to find at least three peaks to form an average among the time between successive peaks. This heuristic, although simple, is worthwhile for our purpose. Here, bin defines the continuously converted data into discrete sets of similar values. Then WEKA preprocessing tool is used to divide the raw data into 10-second segments. The duration of each segment is termed as the Action Duration (AD). We chose a ten-second length AD because we saw that it provided sufficient time to capture several repetitions of the motions involved in some of the six actions.

From every AD, we chose all example for per person and activity and then divide the acceleration data of each axis. We generated a total number of forty summary features in Figure 1.

3.2 Feature Selection

We Data representation, coming out from smartphone sensor data, regarding feature variables, have been identified and evaluated for HAR [10, 11]. The feasibility related to the achievement of the optimal feature set is minimal because of the high computational costs connected to search the desired feature from large sample space.

Feature selection from the signal with high information content is necessary to identify the salient candidate features in high dimension feature space. This step of Data Mining detects and discards the demonstrated extra pruned features that perform less significantly to cause a correct response of the classifier. A feature is considered a statistical candidate in its evaluation if removing it decreases the prediction power. Moreover, the feature is considered redundant if another relevant feature exists with similar predicting power. We can categorize Feature selection methods as wrapper, filter, or embedded techniques [11, 17]. Filter methods consider data's correlation characteristics, information theory to evaluate features independent of a classifier [19]. A wrapper method wraps around a classifier using bias of the particular classifier. Thus, both of the latter two methods generate classifier

specific results used for the task. Hence, weights of features or in special, feature subset selection may only be useful using that particular classifier during filtering. Therefore, the latter two methods may produce feature weights or ranked list of features suitable only for a particular task in hand.

ReliefF [11] is an extension of Relief family of the algorithm is lately the popular filter method among researchers that ranks features by weighting them with the values that distinguish between the instances based on the nearest relevance points. For each instance as a representation of examples, the algorithm finds the closest context (data point from the same class) and nearest misses (data points from different classes). Thus, it can be applied to contextual information problem returned as ranked features list in the subset and can correctly estimate the quality of relevance among attributes in problems with strong dependence between attributes.

Table 1. Feature subset in feature space for Classifier

DataSet	RFTree	FCBF
Data Population	2,40,43,42,44, 41,39, 1,45,34,37,35, 22,36, 12,32,20,17,21 ,18,16, 13,15,9, 4,28,11,3, 27,19,10, 30,14,6,31, 23,8,7, 5,29,25,24, 26,33,38	40,45, 41,19,36

The Fast Correlation-Based Filter (FCBF) method takes into account the high correlation between features and the concerned classes and selects the dominant features from its redundant neighbors. Symmetrical uncertainty serves as the factor for the subsets of features, and the algorithm can detect inessential features and henceforth faster selection with a high level of dimensionality reduction is achieved.

Pragmatically these two filter methods churn out multiple features from a common data set. Specifically, Relief-F comes as handy in case data interdependencies could happen, for example, interdependencies between fields in case of from the same accelerometer sensor data. FCBF is desirable as it draws features that have high correlation and distracts features that are uncorrelated with each other. In the space of highly correlated features, it is necessary to prune irrelevant and unnecessary features and identify features that can be used in a body to increase performance. FCBF usually avoid feature overhead by choosing smaller subsets than Relief-F, thus saves computational cost. It is noticeable that fewer features are beneficial and ideal in a wearable system. Table 1 shows the two feature selection methods

4. ACTIVITY CLASSIFICATION

Activity Classification is a form of data analysis that can be used to induce models from the training set to label attributes from classes. Moreover, it can be used to predict future data

trends or target attribute over test data by finding the accuracy. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.

Intuitively for choosing classifier, the manual intervention for handcrafting of classifier parameters sometimes is harmful to achieving good generalization properties of the classifier itself. We used generic classification algorithms from the WEKA data mining suite [17] to induce models for predicting the user actions. To verify if the features selected are informative, we use different classification methods for classifying the six activities. We compare the classification results obtained using several generic classifiers of Decision trees and probabilistic model [17]. It includes Decision Trees, a Random Forest of 10 Decision Trees, Bagging of 10 Decision Trees. We considered Adaboost using decision trees as base classifiers. The data set has been created using the 46 features selected by the RELIEF. In each case, ten runs were chosen as default settings and crafted the parameters for good generalization characteristics. We use 10-fold cross-validation [19] validation stratification procedure. It ensures that each fold contains approximately the same proportions of different classes for all experiments as a measure of finding accuracy.

A decision tree is a flowchart-like tree structure, in which each internal node performs a test on an attribute. Each branch represents an outcome of the trial, and each leaf node holds a class label. J48 is the version of decision tree implemented in WEKA. After applying the C4.5 Decision trees (J48) in our data set with another classifier, we found the resulting output in figure 2.

The RF [11] belongs to one of the newest algorithms and is a non-probabilistic decision tree based classifier. The algorithm generates some decision trees. For each tree, the algorithm considers only a random subset of the available data. Additionally at each node, the split happens with only a random subset of all features. The final trees do not show any pruning. Each tree experience classification operation and a majority vote over all trees decide which class label is assigned. The RF classifier is considered to provide resistance to over-fitting that guarantees generalization to new data.

Bagging [17] can do classification and regression depending on the base learner with the ultimate aim to reduce variance. Bagging provides a sample "with replacement". We can choose here the bag size. For example, a bag size of 100% is going to sample the training set to get another set the same size. That means we are going to get different sets of the same size every time we sample, but each set might contain repeats of the original training. The datasets that are marked with asterisks are used in bias-variance estimation experimental study. Figure2 presents for each data set the zero-one loss, which is obtained by the ten test folds to give an averaged accuracy of an algorithm. Statistically, we interpret win/draw/loss record (W/D/L) from the result in the figure. The record counts the number of data sets for which one algorithm performs better, equally well or worse than the other on a given measure.

Dataset	(1) meta.Ada	(2) funct	(3) trees	(4) trees	(5) meta.
'activity_recognition_lab(100)	91.79	89.84	92.51	88.81 *	90.68
	(v/ /*)	(0/1/0)	(0/1/0)	(0/0/1)	(0/1/0)

Key:

- (1) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -1178107808933117974
- (2) functions.Logistic '-R 1.0E-8 -M -1' 3932117032546553727
- (3) trees.RandomForest '-I 10 -K 0 -S 1 -num-slots 1' 1116839470751428698
- (4) trees.J48 '-C 0.25 -M 2' -217733168393644444
- (5) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -115879962237199703

Fig 2: Result of WEKA experimenter for the performance comparison of different classifiers.

Correctly Classified Instances	1349	93.0345 %							
Incorrectly Classified Instances	101	6.9655 %							
Kappa statistic	0.8991								
Mean absolute error	0.0432								
Root mean squared error	0.1325								
Relative absolute error	18.7472 %								
Root relative squared error	39.0611 %								
Coverage of cases (0.95 level)	99.5172 %								
Mean rel. region size (0.95 level)	26.9655 %								
Total Number of Instances	1450								
=== Detailed Accuracy By Class ===									
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.980	0.016	0.976	0.980	0.978	0.963	0.998	0.997	Strolling
	0.984	0.009	0.982	0.984	0.983	0.974	0.996	0.996	Running
	0.795	0.034	0.738	0.795	0.765	0.737	0.969	0.809	Ascending_Up
	0.702	0.026	0.774	0.702	0.736	0.706	0.973	0.823	Descending_down
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Relaxing_sitting_inhali
	0.944	0.001	0.944	0.944	0.944	0.944	1.000	0.994	Relaxing_standing_exhal
Weighted Avg.	0.930	0.016	0.930	0.930	0.930	0.914	0.991	0.957	

Fig 3: Random Forest classifier without Attribute selection

Correctly Classified Instances	1339	92.3448 %							
Incorrectly Classified Instances	111	7.6552 %							
Kappa statistic	0.8891								
Mean absolute error	0.0448								
Root mean squared error	0.1354								
Relative absolute error	19.4497 %								
Root relative squared error	39.9053 %								
Coverage of cases (0.95 level)	99.6552 %								
Mean rel. region size (0.95 level)	27.4828 %								
Total Number of Instances	1450								
=== Detailed Accuracy By Class ===									
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.976	0.019	0.973	0.976	0.975	0.957	0.995	0.995	Strolling
	0.986	0.008	0.984	0.986	0.985	0.977	0.998	0.997	Running
	0.712	0.035	0.712	0.712	0.712	0.677	0.974	0.826	Ascending_Up
	0.739	0.032	0.744	0.739	0.741	0.709	0.975	0.799	Descending_down
	0.917	0.001	0.957	0.917	0.936	0.935	1.000	0.991	Relaxing_sitting_inhali
	0.944	0.000	1.000	0.944	0.971	0.971	1.000	0.997	Relaxing_standing_exhal
Weighted Avg.	0.923	0.018	0.923	0.923	0.923	0.906	0.992	0.956	

Fig 3: Random Forest classifier with Attribute selection

A difference is assessed as significant if the outcome of a one-tailed binomial sign test is less than 0.05. The annotation v or * indicates that a specific result is statistically better (v) or worse (*) than the baseline scheme (in this case, ZeroR) at the

significance level specified (currently 0.05). At the bottom of each column after the first column is the count of how many times other algorithms are better, equal or statistically worse than baseline schemes. The result shows that random forest

was equal to baseline scheme once, trees are worse once, Bagging was once equal to baseline scheme, and logistic performs the same as Bagging for the baseline. Figure 3 and Figure 4 respectively show the classification accuracy is obtained using Random Forest with and without ReliefF attribute selection. The Relief-F subset does not provide eliminated list of features for all populations when comparing these six activities. We take the features that are assessed to rank well for each single acceleration axis such as the cumulative sum of y linear acceleration, the skewness of z-acceleration, kurtosis of x-gravity, etc.). The activities augur well for body orientation, and the, therefore, a combination of generic features give an indication of the person's posture. Correspondingly, it is observed from the Table 1, the features selected by Relief-F were not similar to the ones chosen as FCBF as this is the ranked list of features, not the curtailed one. The observation from the figure indicates that feature elimination from the feature space during feature selection discovered redundant features and did not show the potential contribution to the classification accuracy. It reveals that the reduced computational overhead associated with the classifier performance results from the suitable feature selection and reduced feature subset. Table 2 shows the accuracy with 5418 examples in the dataset. It shows the much larger gap of accuracy with and without attribute selection with FCBF for Random Forest, however, yet the difference is not significant. Figure 3 and Figure 4 also show the Weka-generated confusion matrix for the Random Forest classifier. Moreover, those include Precision, Recall, and F-measure values. Precision measures the number of correctly identified activities among those classified as the activity. Recall denotes the proportion of instances classified as activity x, among all instances that are activity x, and the true positive value it relates. The F-measure which equals to

$$\frac{2 * Precision * Recall}{Precision + Recall}$$

combines precision and recall values in a hybrid measure of a test's accuracy.

Table 2. Performance Comparison of Three Classifiers

Classifier	Accuracy	Kappa
DT Classifier	89.4611%	.8559
RF with Attribute Section	92.691%	.9011
RF without Attribute Selection	88.3%	.8011
Bagging	93.3011%	.9086
AdaBoost	73.3011%	.7214

The F-measure shown for each class shows that how each activity can be classified with high precision and recall. In particular, activity with the best performance is strolling and walking from the confusion matrix as they have distinct characteristics in producing signal variations along three axes in which running evokes significant accelerations at a certain frequency. Bagging and Random Forest are the best classifiers that give the best performances for each class. Surprisingly enough from the observations is that the similar activities like ascending and descending stairs have some confusions between those and far apart in the value from strolling and running. Relaxing sitting and relaxing standing have similar values far apart from the running and strolling. Both of these

activities resemble very non-periodic acceleration signals. The F-measure does not present significant difference between the classes that means that six activities can be recognized with high confidence.

Kappa Statistics which is an analog of correlation coefficient serves as an excellent choice for showing the classification for the distribution of the data in an unbalanced dataset and Accuracy does not show the actual classification in an unbalanced dataset. [19]. Its value starts from zero for no correlation and approaches to one for the very distinct statistical correlation between the class label and attributes of instances, i.e., among the categories of strolling or running and the values of their descriptors. The area under the curve (AUC) for Roc (Receiver Operating Characteristics) [19] values in both figures indicate the performance of this classifier; the larger the area, the better is the Algorithm. Ranking based evaluation metrics is used increasingly in machine learning and data mining community when dealing with imbalanced data [17]. When the data are imbalanced, cost-sensitive methods must be considered as well.

5. CONCLUSIONS

Toward the end, in this paper, the purpose of the study is to determine the effect of feature construction in different dimensions and feature selection in the reduced subset for the classifier performance on identifying activity. With this aim, we have aggregated the raw time accelerometer data into examples, in which WEKA software tools labels each example with the activities that took place. A comparison was made with accuracy and other evaluation measures regarding feature selection and without feature selection. This comparison shows us our hypothesis on the effectiveness of feature subset space over classification accuracy. Random Forest appeared to be the best classifier. Pursuant to the earlier discussion, the experimentation eventually leads to finding the accuracy to meet our hypothesis that transformed data may result in equally well or better accuracy. Consequently, we may conclude that, all the parameters that define the classifier performance experience the equivalence with reduced feature set. Thus, it effectively minimizes the computation burden on the HAR. In future, the system must recognize interleaving activities, and deployment of intelligent agents could unobtrusively prompt the user for new activity labels and use these labels to build new activity models. We believe that multitude of classification techniques can be employed to raw sensor data Furthermore, we intend to leverage our Android-based data collection/data mining platform to the fullest extent possible.

6. REFERENCES

- [1] Anderson, I., Maitland, J., Sherwood, S., Barkhuus, L., Chalmers, M., Hall, M., Brown, B., and Muller, H.Shakra .2007. Tracking and sharing daily activity levels with unaugment mobile phones. Mobile Networks and Applications, vol. 12 (Nov. 2007), 2-3.
- [2] Parisa R, Diane JC, Lawrence BH, Maureen S., Discovering activities to recognize track in a smart environment. IEEE Trans Knowl Data Eng, 23(4), 527–539.
- [3] L. Bao, S.S. Intille, G. 2004. Activity recognition from user-annotated acceleration data. In Proceedings of the Pervasive, April 2004, 1-17.
- [4] Mathie, M.J., Coster, A.C.F., Lovell, N.H., Celler, B.G. 2004. A pilot study of long-term monitoring of human

- movements in the home using accelerometry. *J. Telemed. Telecare* 10, 144-151 (2004).
- [5] Yu H.; Spenko, M., DuBowsky, S. 2003. An adaptive shared control system for an intelligent mobility aid for the elderly. *Auton, Rob*, 53-66.
- [6] Cho, Y., Nam, Y., Choi, Y-J., and Cho, W-D. 2008. SmartBuckle: Human activity recognition using a 3-axis accelerometer and a wearable camera. *HealthNet*.
- [7] U. Maurer. 2006. Location and Activity Recognition Using eWatch: A Wearable Sensor Platform. *Ambient Intelligence in Every Day Life*, LNCS 3864, Springer, 2006, 86-100
- [8] Gyorbiró, N., Fabian, A., and Homanyi, G. 2008. An activity recognition system for mobile phones. *Mobile Networks and Applications*. 14(1). 82-91.
- [9] T. Huynh, B. Schiele. 2006. Unsupervised discovery of structure in activity data using multiple eigenspaces. *Proc. of the Second International Workshop on Location and Context-Awareness*. Dublin, Ireland, (May 2006), 151-167.
- [10] Krishnan, N., Colbry, D., Juillard, C., and Panchanathan, S. 2008. Real time human activity recognition using tri-Axial accelerometers. *Sensors, Signals and Information Processing Workshop*.
- [11] Liu H, Motoda H. 2007. *Computational Methods of Feature Selection*. CRC Press; 2007
- [12] Mathie, M.J.; Celler, B.G.; Lovell, N.H.F., Coster, A.C. 2004. Classification of basic daily movements using a triaxial accelerometer. *Med. Biol. Eng. Comput.*, 2004, 42, 679–687.
- [13] J. Lester, T. Choudhury and G. Borriello. 2006. "A Practical Approach to Recognizing Physical Activity," *Proc. 4th Int'l Conf. Pervasive Computing (Pervasive 06)*, LNCS 3968, Springer, 2006, 1-16.
- [14] D. Wyatt, M. Philipose, T. Choudhury. 2005. Unsupervised activity recognition using automatically mined common sense. In *Proceedings AAAI*, 05, (July 2005), Pittsburgh, PA, USA, 21-27.
- [15] Ravi, N., Dandekar, N. 2005. Activity recognition from accelerometer data. In *Proceedings the Seventeenth Conference on Innovative Applications of Artificial Intelligence*, University of Washington.
- [16] Tapia, E.M., Intille, S.S. Real-Time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. *Proceedings of the 2007 11th IEEE International Symposium on Wearable Computers*. 1-4, 2007.
- [17] Witten, I. H. and Frank, E. 2005 *Data Mining: Practical Machine Learning Tools and Techniques*. 2nd ed., Morgan Kaufmann.
- [18] Actitracker. 2012. <https://www.actitracker.com/>
- [19] V. N. Vapnik. 1998. *Statistical Learning Theory*. John Wiley & Sons.