HUMAN ACTIVITY RECOGNITION IN WEARABLES USING RANDOM FOREST FEATURE SELECTION TO IMPROVE MODEL PERFORMANCE

Table of Contents

Introduction of the study scope

1.1 Introduction

There is no denying the rise of miniature electronic devices that can be embedded in clothing, or attached to a user's body in the form of ***"Wearables"***. With over 1 Billion wearable connected devices in 2022, the wearable technology market is projected to reach US$ 380.5 Bn by 2028 (*Global Wearable Technology Market Report*, 2022)

The massive growth in the Wearable technology market is due to its wild range of applications including Healthcare. Healthcare Researchers working on applications of Wearables, have identified some key areas of focus: The design and implementation of sensors that unobtrusively and reliably record movement or physiological signals; The design and implementation of algorithms to extract clinically relevant information from data recorded. (Bonato, 2005)

This study focuses on the latter and implements a hardware memory-efficient model which is able to accurately classify human activities (standing, sitting, laying, e.t.c); which could be embedded into a wearable. The objective of this study is to present results obtained by using *Random Forest* classifier for selecting features and to compare its performance without it in terms of classification accuracy and runtime.

This is an active field of research, therefore some of the related work done uses principal component analysis to reduce the dimension of the available datasets and applied some supervised learning algorithms such as ***Logistic Regression*** (Zaki et al., 2020), ***Support Vector Machine(SVM)*** (Anguita et al., 2012).

This work focuses on selecting important features (Lingjun et al., 2018) from the UCI datasets (*Human Activity Recognition Using Smartphones Data Set*, 2012) using ***Random Forest*** because it is a supervised learning algorithm, which has a unique capability of selecting the features that best classify the targets (Breiman, 1999). In this work, important features would be selected and trained on three supervised learning algorithms and a comparison made with a benchmark prediction.

1.2 Project Scope

This study would cover the following aspect:

- Data Visualizations
- Data Analysis
- Machine Learning

It does not cover app development and deployment of machine learning models, although this can be considered for real-time use as all the architecture for making predictions is carried out in this course work.

1.3 Related Work

Recognizing human activity by computer is becoming increasingly important as it is useful in healthcare to monitor patients, for sick and disabled assistance, elderly care in homes, health surveillance, and involuntary actions. It is also used in gaming and entertainment applications.

Many works have been done to improve the ability of computers to recognize activity performed by humans some of the most relevant works leverage devices that are hardware friendly and also sensors attached to the body (Zaki et al., 2020) although this increases the accuracy of recognizing human activity but utilizing this in real-time is almost practically impossible due to the number of sensors that will be attached to the body.

This work focuses on introducing a low computational approach for multi-class classification. This method utilizes the ability of Random Forest Classifier to extract the important features from the datasets, therefore, reducing the size of the feature set and then using a suitable multi-class algorithm to classify the six different activities.

Some of the machine learning methods that have been used to tackle this problem include ***Naïve Bayes***, ***SVM*** (Anguita et al., 2012), **Logistic Regression**, and Ensemble Learning (Hsu et al., 2022). (Zaki et al., 2020) uses the mentioned classifier to classify the different activities after using principal component analysis for dimensionality reduction and the result shows that logistic regression performs better with an accuracy of 94%. (Anguita et al., 2012) suggest a hardware-friendly approach that adapts the support vector classifier and exploits fixed points arithmetic for computational cost reduction. Although this seeks to reduce the complexity of the model by reducing the floating-point precision.

Implementation

2.1 Introduction

The machine intelligent software used to implement this work includes *sci-kit learn* (Buitinc et al., 2013) library, p*andas, NumPy, and matplotlib*. All codes are written in python and jupyter notebook.

This study implements a machine learning model capable of recognizing six different human activity recognition. The following steps were taken to implement the work:

Data preprocessing - The data is divided into 70% training and 30% testing sets. The pandas library is used to load the datasets into a data frame, the data frame object is converted into an array using the *NumPy* library which is also used to perform some relevant computations such as mean, and max. The *sci-kit learn* minmaxscaler class is used to normalize the training data.

Feature Extraction - The feature extraction step is done using the ensemble random forest module and the *SelectFromModel* from *sci-kit learn* Library.

Training of the model - To train the model, three multi-class algorithms are fitted on the training data.

Testing of the model - The model is then used to make predictions and on the test sets.

Analysis of the results -The accuracy of the model is calculated and also the time to make predictions.

2.1 Learning Algorithms

K-Nearest Neighbor

This is a non-parametric supervised learning algorithm that relies on the distance between data points to make its prediction. Hence, the learning and prediction are performed based on the given problem or dataset (Theerthagiri et al., 2020).

It is used for both classification and regression tasks. Predictions are made by searching the similar k-neighbor instances in the entire training data. The *k* represents the number of datasets closest to the prediction instance. The distance can also be calculated based on the Euclidean distance which is given as;

$$Euclidean_{i,j} = \sqrt{\sum_{k=1}^{n} (x_{ik} - y_{jk})^2}$$

Logistic Regression

The Logistic Regression uses the sigmoid (Pratiwi et al., 2020) function for predictive modeling of the given problem. It models the dataset and maps them into a value between 0 and 1. The logistic Regression performs the predictive analysis based on the relationship between the binary dependent variable and the other one or more independent variables from the given dataset. In order to predict the output value (Y), the input values (X_1, X_2,… X_n) are linearly combined using the coefficient values.

Support Vector Machines

SVMs are a set of related supervised learning methods used for classification and regression (Srivastava & Bhambhu, 2010). They belong to a family of generalized linear classification. A special property of SVM is, SVM simultaneously minimizes the empirical classification error and maximizes the geometric margin. So SVM is also called Maximum Margin Classifiers. SVM is based on Structural Risk Minimization (SRM). SVM maps input vectors to a higher dimensional space where a maximal separating hyperplane is constructed. Two parallel hyperplanes are constructed on each side of the hyperplane that separates the data. The separating hyperplane is the hyperplane that maximizes the distance between the two parallel hyperplanes. An assumption is made that the larger the margin or distance between these parallel hyperplanes the better the generalization error of the classifier will be.

Feature Extraction with Random Forest

For a single CART model, (Breiman, 1999) proposed a variable importance measure by using surrogate splits intended to mitigate the risk that the variable importance of a single variable is masked. For an aggregated model such as bagging or boosting the variable importance, the measure is not as limited by the overall size of the tree, and the number of splitting opportunities is vastly increased which implies that masking is less of a problem. Despite the averaging effect, there is still a possibility that a single variable X_2 is not included in the model due to a slightly higher performance obtained when the data instead is partitioned by variable X_1 (say that X_2 is highly correlated with X_1). This in hand implies a tiny importance measure for variable X_2. The random forest model further reduces the likelihood of masking caused by the latter scenario by only allowing a random subset of features available at each split. Thus, for all feature sets not including X_1 the likelihood that the correlated variable X_2 is used as a partitioning variable is increased. Since the individual trees contrary to bagging are grown deep, the contribution in variable importance due to interaction effects is increased. The boosting procedure may on the other hand choose to ignore some variables completely. The permutation importance measure, introduced by (Breiman, 1999), is one of the two most common variable importance measures. To measure the importance of a variable X_i; The idea is to permute all values of this variable, and the variable importance measure is defined as the difference in prediction accuracy caused by the permutation. If the variable consists of purely random noise the prediction accuracy will likely not be affected by permuting the values of this variable. Formally the variable importance is computed as follows. Let β^t denote the OOB samples for a tree t and let $L(T_t\left(x_i\right), y_i)$ denote the prediction accuracy at the ith training example. The importance for the variable X_j in tree t is defined as

$$VI^{(t)}\left(X_J\right) = \sum_{i\epsilon\beta^t} L\left(T_t\left(x_i\right), y_i\right) - L(T_t\left(x_{i\pi}\right), y_i)$$

where $x_{i\pi} = (x_{i1,} ...., x_{\pi j(i),j,} x_{ij+1} ...., x_{ip})$, and where π_j is a random permutation of n integers. In classification settings the prediction accuracy $L(T_t\left(x_i\right), y_i)$ is defined

as

$$L\left(T_t\left(x_i\right), y_i\right) = \frac{\sum_{i\epsilon\beta^t} I(\hat{y}_i^t = y_i)}{\left|\beta^t\right|}$$

where $\hat{y}_i^t = T_t\left(x_i\right)$ denotes the prediction at point xi by tree t, and $I(\cdot)$ denotes the indicator function. Whereas in regression settings the prediction accuracy $L(\hat{y}, y)$ is defined as the RMS error. The variable importance measure for variable X_j is computed as the sum of the importances over all trees in the forest,

$$VI\left(X_J\right) = \frac{\sum_{t\epsilon\beta} VI^{(t)}(X_j)}{ntree}$$

Another commonly used importance measure is the Mean Decrease Impurity (MDI). The importance measure for variable X_J is computed by the sum of all decreases in node impurities where variable X_J is used to partition the data. If we let t_L and t_R denote the two resulting children nodes when partitioning the data at node t, and we let N_t denote the number of examples reaching node t, the decrease in impurity is defined as

$$\Delta_i(s, t) = i(t) - pLi\left(t_{L}\right) - p_R i(t_R)$$

where $i(\cdot)$ is some impurity measure, and $p_L = \frac{N_{t_L}}{N_t}$ $and\, p_R = \frac{N_{t_R}}{N_R}$ The resulting children nodes are obtained when the data is partitioned by the parent node at $s = (X_j < c)$. Lastly the MDI measure is defined by averaging over all trees T and all nodes t,

$$VI\left(X_J\right) = \frac{1}{ntree}\sum_{T} \sum_{t \epsilon T : v(st) = xm} p(t)\Delta_i(s, t)$$

where p(t) denotes the proportion $\frac{N_t}{N}$ of samples reaching node t and v(st) denotes the variable used to split node t.

A common choice of node impurity measure is the ***Gini-coefficient***, a combination which commonly is denoted as the ***Gini Importance***. Whereas in regression settings a common impurity measure is the ***MSE***. The permutation and the Gini importance both capture non-linear relationships, as indicated by the noisy-circle data for the permutation importance.

Furthermore, the two measures capture importance for variables that are correlated with informative predictors. Both the permutation importance and MDI measure with the RME and Gini impurity measures are freely available in the package Random Forest by Liaw and Wiener for the R system for statistical computing. In settings with correlated predictors, the permutation variable importance measure has been observed to put increased importance onto variables that are correlated, likely due to the fact that correlated variables are preferred as splitting candidates earlier in the tree (Strobl et al., 2008). The authors suggest a modified variable importance measure following the logic of permutation tests. The idea is to form a null hypothesis designed to investigate whether predictor variables are informative. If the null hypothesis H0 is specified as a global null hypothesis, i.e., all predictor variables are independent of the target variable $(Y \perp X_{1,} ..., X_P)$. The null hypothesis implies that the joint distribution then factorizes as

$$P\left(Y,\ X_{1,} ..., X_P\right) = P(Y) \cdot P(X_{1,} ..., X_P)$$

If the data is truly generated under the null hypothesis, a permutation of the target variable will not affect the joint distribution due to the factorization. On the contrary, if the null is false and the target variable is permuted, an observed deviation of the joint distribution or some reasonable test statistic computed from it is to be expected.

Under the global null hypothesis, it is expected that the permutation importance measures are distributed as a zero-mean random variable. A deviation from the null hypothesis is expected to imply a change in prediction accuracy which in hand implies a deviation to the permutation importance, the deviation in importance measures is chosen as a test statistic to indicate deviations to the independence assumption. The null hypothesis which corresponds to the original permutation importance assumes that variable X_j is independent to both Y and to

$$Z = X_1, ..., X_{j-1}, X_{j+1}, ..., X_P \ .$$

To mitigate importance deviations arising due to dependency between variables X and Z [8]. suggests a modified permutation scheme where variable X_j is permuted only within groups of observations with $Z = z$.

The conditional permutation importance, given by [9], for variable X_j is computed by the 4 steps,

1. First the OOB-prediction accuracy is computed as in the equation.

$$\frac{\sum_{i\epsilon\beta^t} I(\hat{y}= y_j)}{\left|\beta^t\right|}$$

2. For each variable Z_j used for conditioning. The cut-points that partition this variable are bisected into a grid to form a permutation grid.

3. Within the permutation grid the values of X_j are permuted and the associated permutation accuracy is computed

$$\frac{\sum_{i\epsilon\beta^t} I(\hat{y}= y_j)}{\left|\beta^t\right|}$$

The contribution to the permutation importance measure for variable X_i by tree t is computed as the difference between the non-permuted prediction accuracy and the permuted prediction accuracy.

The permutation importance measure for variable X_i is obtained by summing all importance contributions over all trees, exactly as for the original permutation importance measure.

2.2 Datasets

The dataset used for this course work is the ***UCI*** (Human Activity Recognition Using Smartphones Data Set, 2012) dataset. The dataset is a result of an experiment carried out, which consists of 30 subjects between the age of 19 and 48 years. Each person performed a daily activity which consisted of ***standing, sitting, laying, walking, walking downstairs, walking upstairs,*** with a smartphone attached to their waist. The smartphone has embedded sensors which are capable of reading the acceleration and angular velocity in the ***x-axis, y-axis, and z-axis***, at a constant frequency of 50 Hz of the subjects as they perform the activity. A video record is taken to label the data and the obtained datasets were separated into 70 and 30 percent for training and testing purposes respectively. Further preprocessing steps were carried out to find the time and the frequency component of the features.

S/n	**Variables for features**	**Description**
1	Mean	Mean Value
2	Std	Standard Deviation
3	Mad	Median Absolute Deviation
4	Max	Maximum Value
5	Min	Minimum Value
6	SMA	Signal Magnitude Area
7	Energy	Energy Measure
8	IGR	Interquartile Range
9	entropy	Signal Entropy
10	arCoeff	Autoregressive Coefficient
11	Correlation	Correlation Coefficient Between Signals
12	MaxInds	With largest magnitude, Index of Frequency Components

13	meanFreq	Weighted Average of Frequency Component For Obtaining Mean Frequency
14	BandEnergy	Energy of Frequency Intervals
15	Skewness	Skewness of Frequency Domain
16	Kurtosis	Kurtosis of Frequency Domain
17	Angle	Angles Between the Vectors

Table 2.1: Values calculated from sensor data

2.3 Data Analysis & Visualization

The datasets consist of 563 features in which the activity column contains the different activities and is being used as the label for this task and the other 561 features as the samples. Different visualization and statistical techniques were carried out to better understand the feature and how their relationship with the label.

Correlation Plot

To see the correlation between the differences, the mean features are extracted from the datasets. That is all the calculated mean from the sensor data are extracted and then I compute the correlation between the samples. The figure below shows the correlation between the time computed mean in the time domain. From the figure below it is seen that the correlation between the time domain computed means is mostly below +0.5 and higher than -0.5 this shows that they are not linear dependent on each other. Also, the correlation between the features; *tBodyAccMag-Mean, tGravityAccMag-Mean, tBodyAccJerkMag-Mean, tBodyGyroMag-Mean, and tBodyGyroJerkMag-Mean* is between 0.9 and 1 which shows a strong dependency.

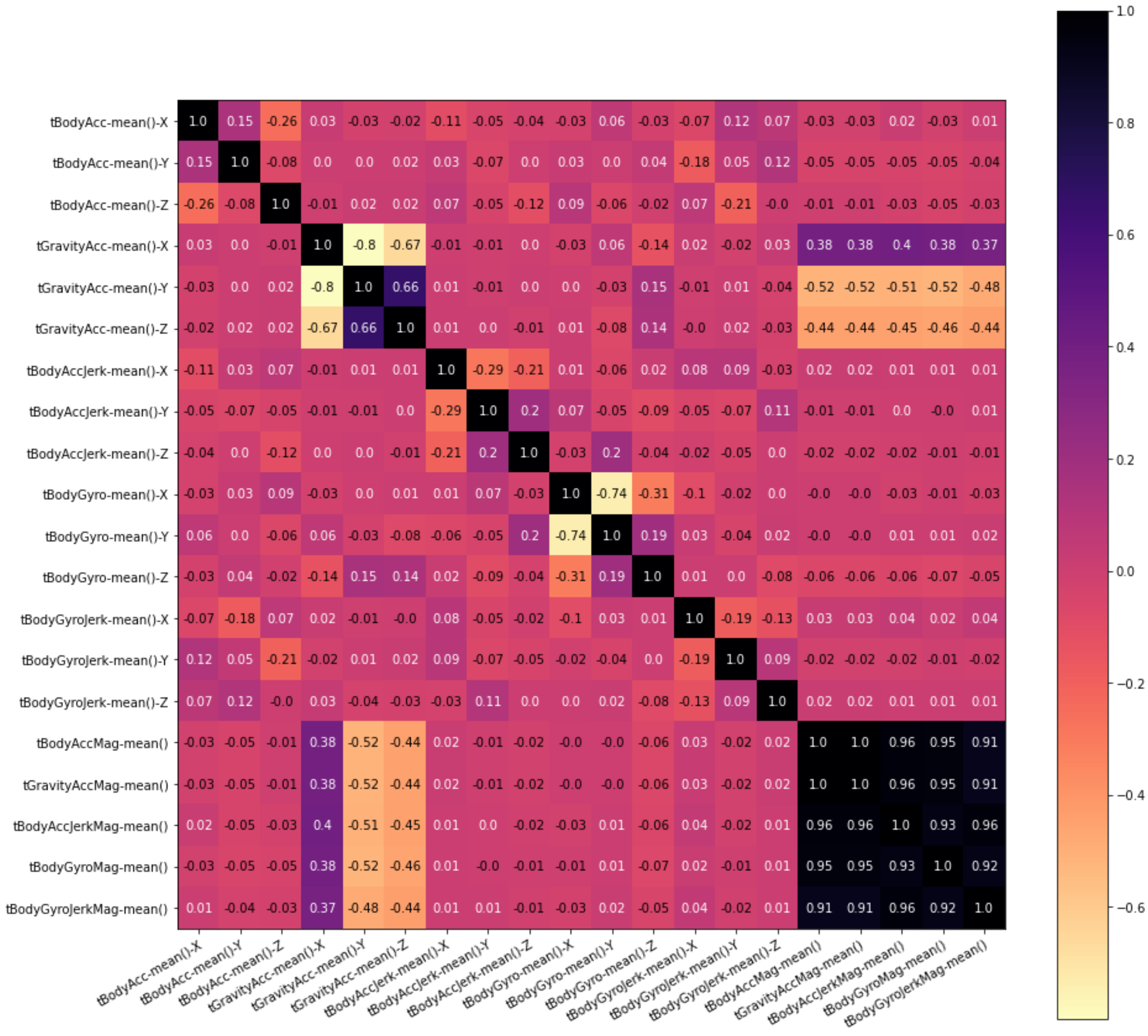

Fig 2.1: Time-domain correlation plots between calculated means

From the figure below it is seen that the correlation between the frequency domain computed means is mostly above +0.5 and below -0.5 this shows that they are linearly dependent on each other. For supervised learning tasks, it is important to drop features that have a strong correlation.

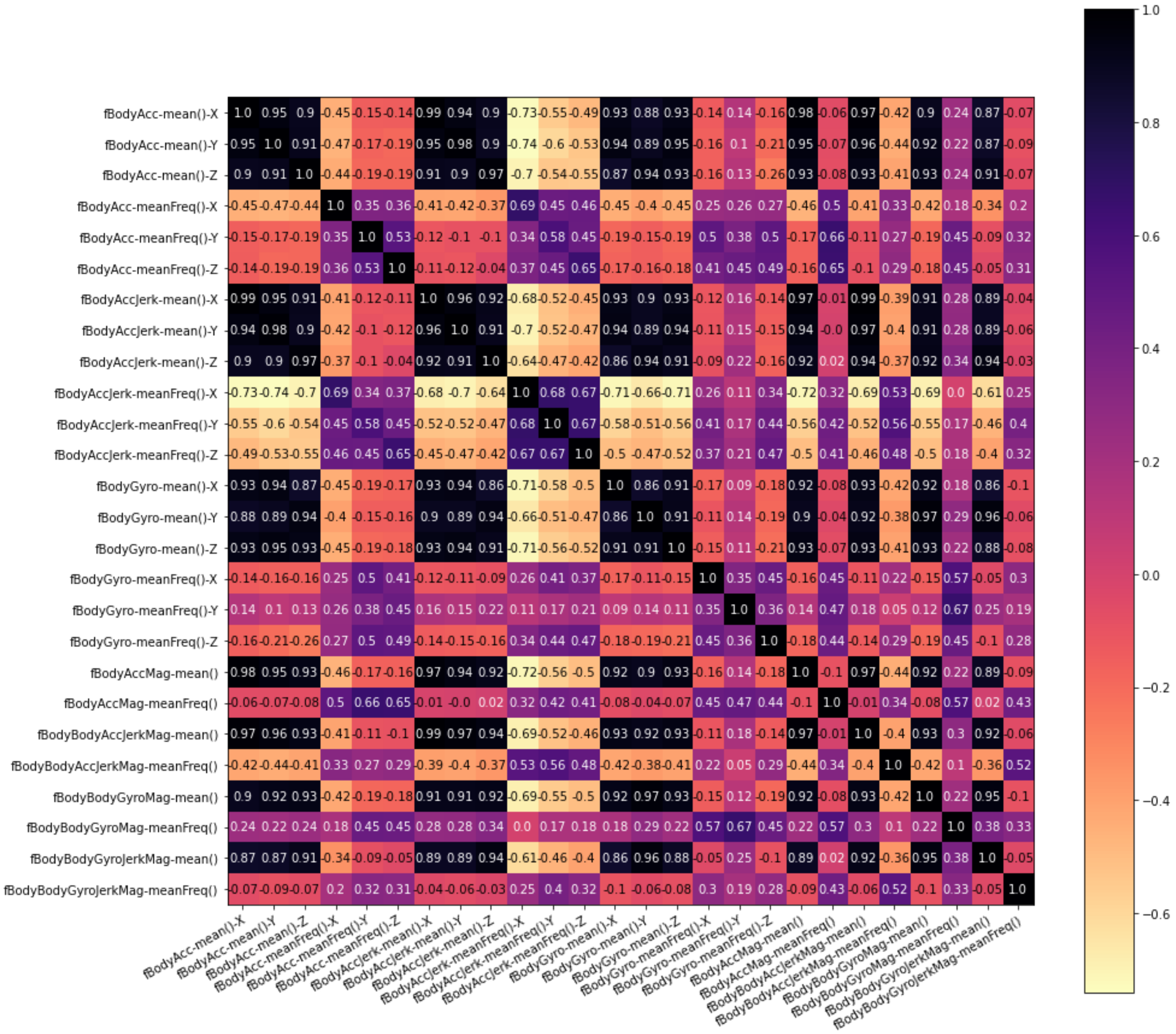

Figure 2.2: Frequency domain correlation plots between calculated mean features

The Line Plots

The mean of the 561 features for each of the classes is taken which is then plotted as a line plot. The figure below shows the variation of the different features of the activities. The line plot for standing, sitting, and laying are very identical as it depicts that the subject is in a static position. While the plots for walking, walking up, and walking down are also similar as the body is commonly in motion.

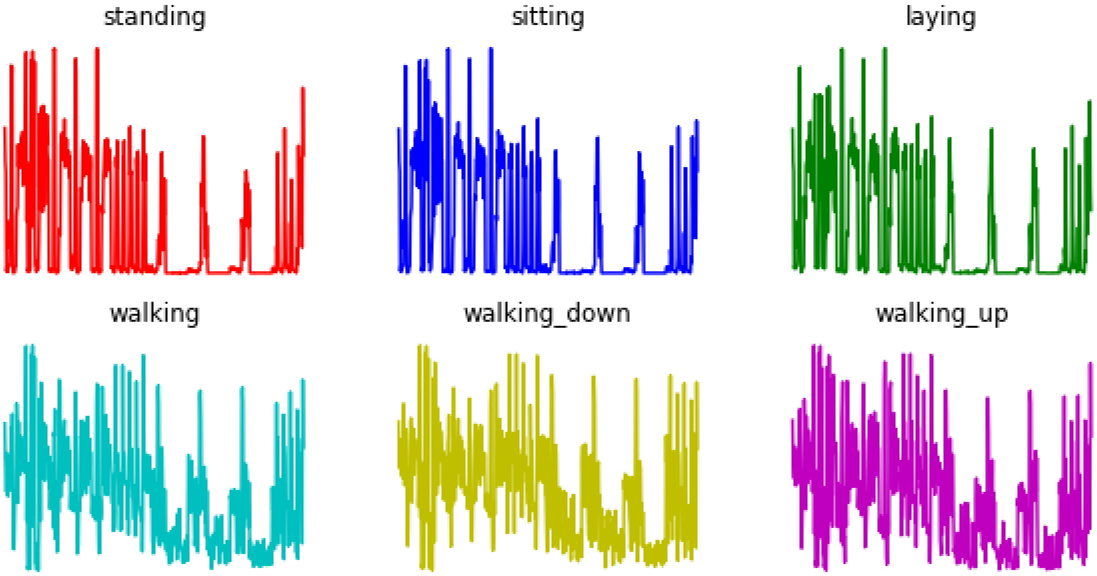

Figure 2.3: Line plots for the mean of the 6 activities

Principal component analysis

This is a linear dimensionality reduction technique which uses singular value decomposition of the feature to project it to a lower dimensional space. The principal components analysis is used to transform the data into 3 feature spaces. That is three principal components; afterwards the total variance is approximately 75%.

Scatter Plots

Principal components scatter plot for the variability between the activities

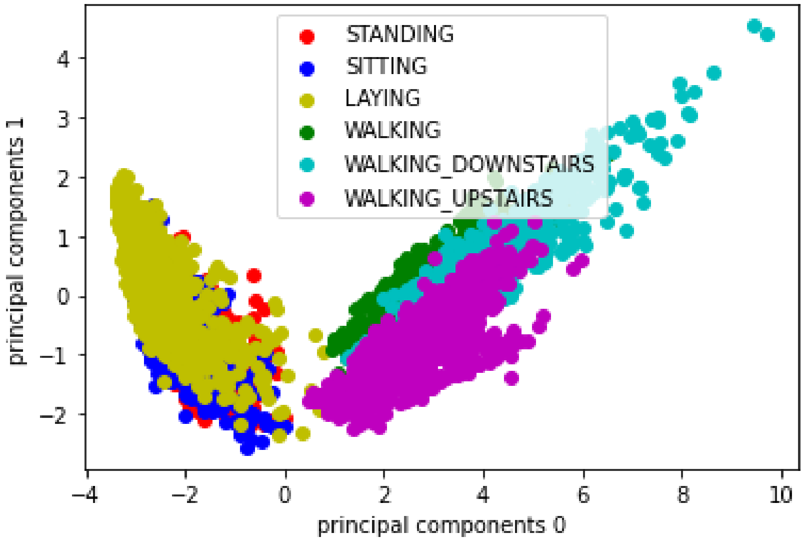

Figure 2.4: Scatter plot for *the mean of the 6 activities*

Pair Plots

The pair plot below gives the marginal distribution between the 6 principal components.

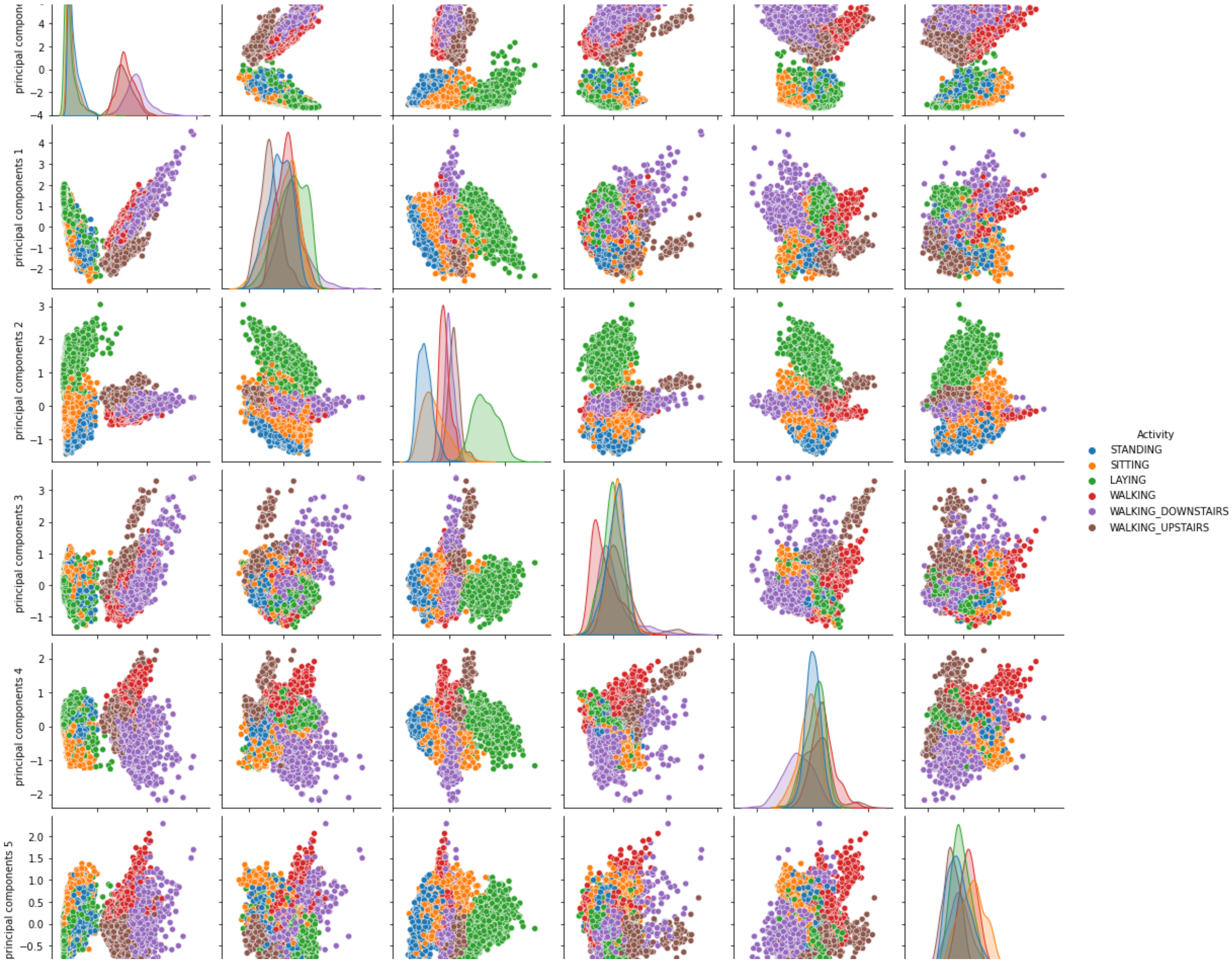

Figure 2.5: Six Principal Components Visualization

5d-plot

The figure below gives the relationship between the 4 principal components and the 6 different activities.

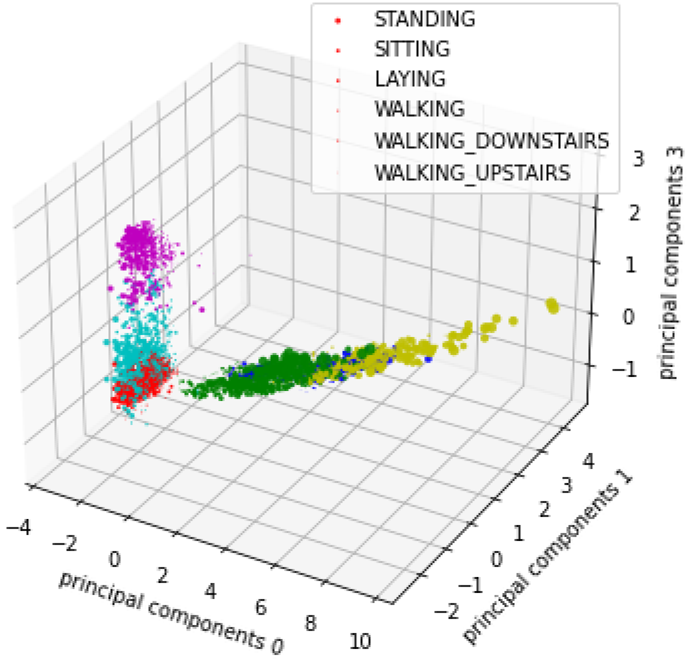

Fig 2.6: Visualizing 5-Dimensional principal components

Experiment

This course work seeks to provide a more computational-friendly model while maintaining accuracy. It also gives an insight into the features which are most important in classifying the activities.

The three classifiers used are logistic regression, support vector machine, and ***k*** nearest neighbor. Since these are distance-based and gradient descent-based algorithms (Theerthagiri et al., 2020) it is important that the features are normalized in other to achieve a great result

3.1 Pre-processing:

The data samples are normalized to be within the range of 0 and 1. Normalization is given as

$$x' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Where: $X = sample$

$X_{min} = minimum\ value\ of\ sample$

$X_{max} = maximum\ value\ of\ sample$

3.2 Feature selection

Random Forest is chosen for the feature extraction as its supervised learning method which selects features that are best for classifying the activities in contrast to principal component analysis which is an unsupervised learning method to select the features with the most variability. The random forest classifier is initialized with the following parameters:

trees = 500
max_feat = 10
max_depth = 50
min_sample = 2

The random forest classifier's feature importance attribute returns the ratio of the contribution of each feature to classifying each activity. The select from model class is used to select features with importances equal to or greater than *0.001*. These new features are used in training the algorithms

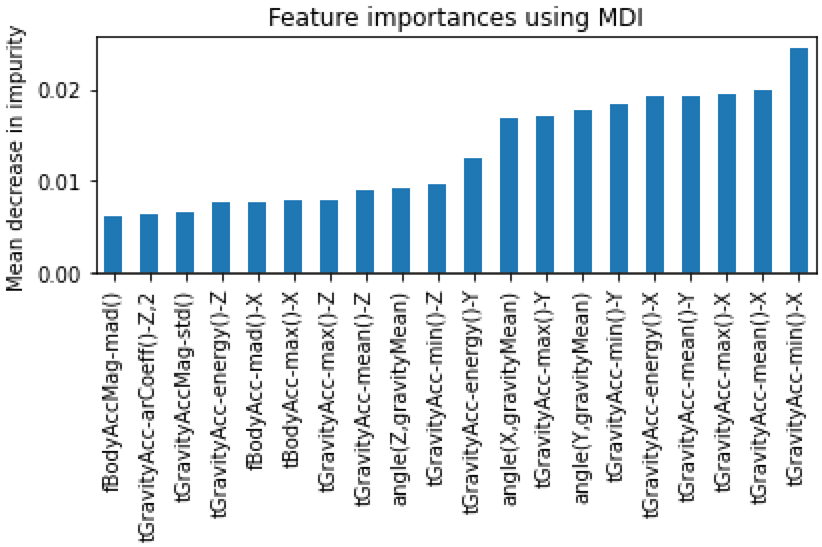

Fig 4.1:

3.3 Results

Three multi-class algorithms; support vector classifier, logistic regression and k-nearest neighbor are trained on all the feature sets and then on the selected features. The following results were obtained:

The size of the training samples and training label is 561 by 7352, and 1 by 7352 respectively

Models	**Accuracy**	**Runtime**
KNN-model	89.99%	0.50s

LR-model	92.79%	0.25s
SVC-model	93.47%	1.13s

Table: 4.1: Summary of accuracy and runtime for making predictions on all features.

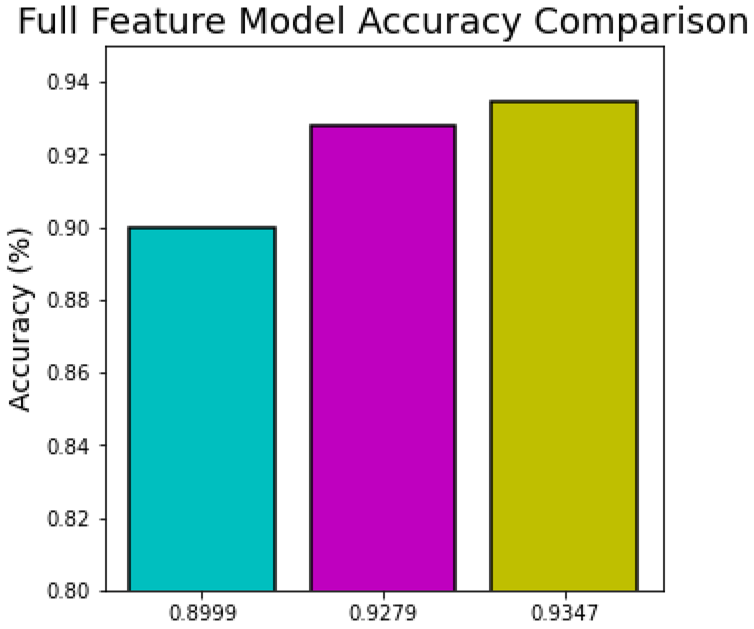

Fig 4.2: Bar Chart of prediction accuracy on all features.

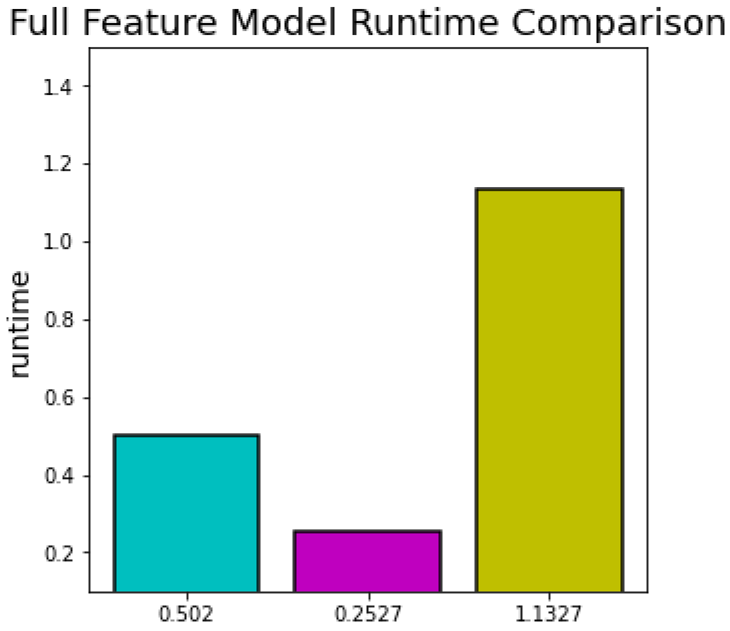

Fig 4.3: Bar Chart of prediction runtime on all features.

The size of the training samples and training label is 265 by 7352, and 1 by 7352 respectively

Models	**Accuracy**	**Runtime**
ME-KNN-model	89.85%	0.42s
ME-LR-model	92.22%	0.21s
ME-SVC-model	92.76%	0.54s

Table 4.2: Summary of accuracy and runtime for making a prediction on the selected features

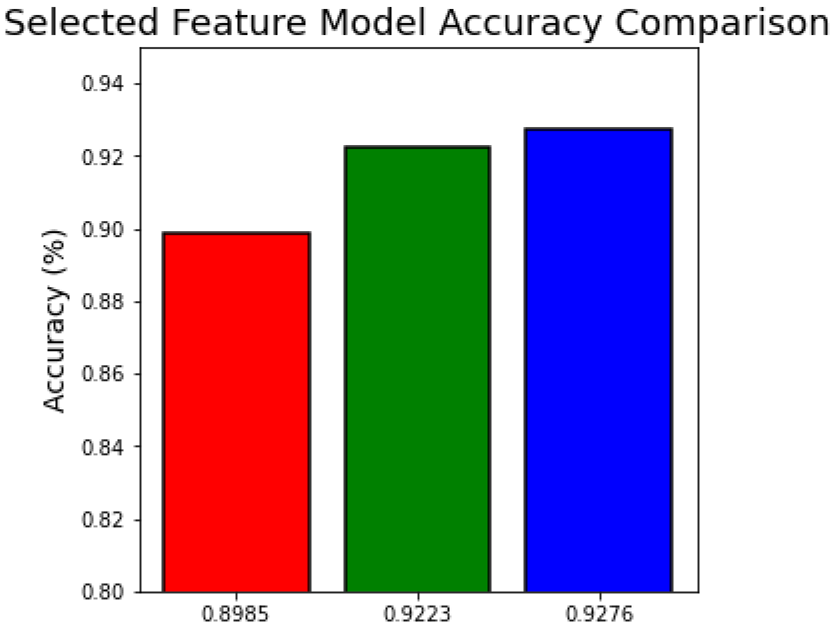

Fig 4.4: Bar Chart of prediction accuracy on selected features

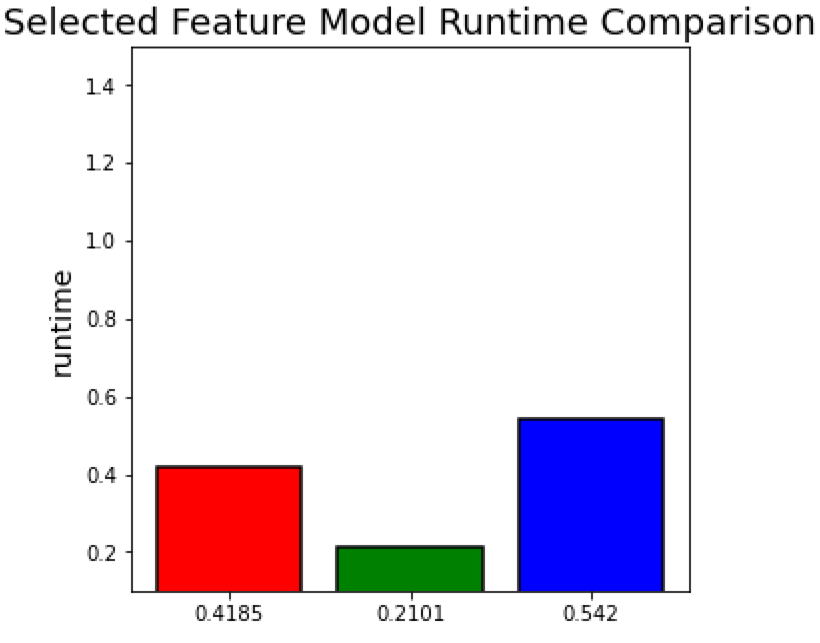

Fig 4.5: Bar Chart of prediction runtime on selected features

Conclusion

This section summarizes the conclusions of this project. It discusses the findings of the project. The limitations are reviewed. It also identifies some suggestions for future work.

This project improves the overall performance of multi-class algorithms for recognizing human activities which includes, standing, sitting, laying, walking, walking down, walking up. The datasets used for training are divided into two; the full feature set, and the selected feature set from the random forest classifier. After training the model the performance metrics which are the accuracy of and runtime for making prediction were calculated, the percentage decrease in accuracy is only 0.16, 0.58, 0.75 percent for the k-nearest neighbor, logistic regression and svm models respectively while the runtime for the three model decreases significantly by 81, 61, 60.72 percent.

4.1 Findings

It was observed from the project that the model runtime decreased significantly when trained on the selected features and the accuracy only decreased by a very small amount. Therefore, random forest classifiers are a good method for improving the memory efficiency of machine learning models.

4.2 Limitations

After evaluating the project, several limitations were discovered, and a more obvious one was noted:

During data virtualization, it was observed that plot charts were a bit complicated due to the enormous amount of features the dataset has - about 561 attributes.

4.3 Future Work

There are several suggestions for future work on this project, some, in particular, stand out:

- More work can be done in the future to tune the parameters of the three multi-class algorithms as this could improve the accuracy of the models.
- This study can be extended to include the actual implementation of the wearable hardware to perform real-time Human Activity Recognition using the built model.

4.4 In Summary

The entire study investigates how the selection of important features for training multi-class algorithms could affect the overall performance which includes the prediction accuracy and prediction runtime. The algorithms considered are support vector machine, logistic regression, and k-nearest neighbor. The work shows that selecting important features can decrease the runtime of the model which makes it memory efficient when deployed as a mobile application.

The accuracy of the memory-efficient models using Random Forest which decreased only by a very small percent is a good trade-off when compared without using it. The model which achieves an accuracy of 92.7% using SVC shows that *Random Forest* is a good method to select important features for multi-class algorithms. The accuracy of the three models could be improved by further tuning the parameters of the multi-class algorithms.

References

Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2012, Dec 3). *Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine*. UPCommons. Retrieved May 13, 2022, from https://upcommons.upc.edu/bitstream/handle/2117/101769/IWAAL2012.pdf

Bonato, P. (2005). *Advances in wearable technology and applications in physical medicine and rehabilitation*.

Breiman, L. (1999). *Random Forests-Random Features*. CiteSeerX. Retrieved May 13, 2022, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.367.9714

Buitinc, L., Loupp, G., & Blondel, M. (2013, September 1). *API design for machine learning software: experiences from the scikit-learn project*. Retrieved May 13, 2022, from https://arxiv.org/abs/1309.0238

Global Wearable Technology Market Report. (2022, April 13). Facts and Factors. Retrieved May 13, 2022, from https://www.fnfresearch.com/wearable-technology-market

Hsu, T. -., Wu, J., & Liu, H. (2022). *Human Activity Recognition Using an Ensemble Learning Algorithm with Smartphone Sensor Data*. MDPI. Retrieved May 13, 2022, from https://www.mdpi.com/2079-9292/11/3/322

Human Activity Recognition Using Smartphones Data Set. (2012, December 10). UCI Machine Learning Repository. Retrieved May 13, 2022, from https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

K, T. (2013). *Wearable Technology and Wearable Devices: Everything You Need to Know. Wearable Devices Magazine*.

Lingjun, H., Levine, R. A., Fan, J., Beemer, J., & Stronach, J. (2018). *Random Forest as a Predictive Analytics Alternative to Regression in Institutional Research*. ScholarWorks@UMass Amherst. Retrieved May 13, 2022, from https://scholarworks.umass.edu/pare/vol23/iss1/1/

Park, S., & Jayaraman, S. (2003). *Enhancing the quality of life through wearable technology*. https://ieeexplore.ieee.org/abstract/document/1213625

Pratiwi, H., Windartogus, A. P., Susliansyah, S., Aria, R. R., Susilowati, S., Rahayu, L. K., Fitriani, Y., Merdekawati, A., & Rahadjeng, I. R. (2020). *Sigmoid Activation Function in Selecting the Best Model of Artificial Neural NetworksSigmoid Activation Function in Selecting the Best Model of Artificial Neural Networks*.

Srivastava, D. K., & Bhambhu, L. (2010, Febuary). *Data Classification Using Support Vector Machine*.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., & Zeileis, A. (2008). *Conditional Variable Importance For Random Forests*.

Theerthagiri, P., Jacob, I. J., Ruby, A. U., & Vamsidha, Y. (2020, November 3). *Prediction of COVID-19 Possibilities using KNN Classification Algorithm*.

Zaki, Z., Shah, M. A., & Wakil, K. (2020). *Logistic Regression Based Human Activities Recognition*.