# Physical Activity Recognition using Inertial Measurement Units (IMUs)

## ECE-5424 FA22 Course Project Report

Shahwar Atiq Khursheed
Subham Swastik Samal
Venkata Adithya Chekuri

**Abstract**

This work involves development and evaluation of a bi-layer classification model to detect physical activities from a multivariate time series dataset namely PAMAP2 [4]. The dataset contains data acquired by recording 9 subjects wearing 3 inertial measurement units (on their hand, chest and ankle) and a heart rate monitor, and performing 18 different activities. The first layer of the bi-layer classification aims to detect the intensity of the activity performed i.e. rest, low, medium and high, and the second layer uses the results of the first layer to perform the final classification. For the second classification layer, the time series was divided to smaller time windows of fixed intervals, from which various statistical, energy, frequency and correlation-based features were calculated and classifiers using these features were tested. Each classifier was tested using two evaluation methods: within-subject cross validation and cross-subject cross validation. Either Random Forests or K- Nearest Neighbor for intensity classification, followed by Support vector machines for activity classification worked best for within-subject evaluation, with a classification accuracy of 98.7%, while the best results for between subject was observed when both the layers were Random forests, with an accuracy of 75.93%.

# Contribution Breakdown

### Source Code

| Code module | Contributing Team Member | Code File |
|---|---|---|
| Data preprocessing – Intensity Based | Subham | preprocess1.py |
| Feature Selection | Shahwar | main.py |
| Classifiers built | Venkata | main.py, main_ts_l2.py |
| Intensity based classification | Shahwar | main.py |
| Within subject and cross subject evaluation –Intensity Based | Shahwar | main.py |
| Time series preprocessing – Activity Based | Subham | time_series_feat.py, ts_prep.py |
| Activity based classification | Subham | main_ts_l2.py |
| Within subject and cross subject evaluation – Activity Based | Subham | ts_eval.py |
| Extended Aim: Heart Rate Prediction | Shahwar | main.py |

### Code Compilation/Results Collected

| Module Run and Results Compiled | Contributing Team Member |
|---|---|
| IB: Data preprocessing | Subham |
| IB: Feature Selection | Shahwar |
| IB: Classification using Random Forest Classifier | Subham |
| IB: Classification using Support Vector Machines | Shahwar |
| IB: Classification using K Nearest Neighbors | Venkata |
| AB: Data preprocessing | Subham |
| AB: Classification using Support Vector Machines | Shahwar, Subham, Venkata (each for their respective IB classified data) |
| AB: Classification using Random Forest Classifier | Shahwar, Subham, Venkata (each for their respective IB classified data) |
| Extended Aim: Heart Rate Prediction | Shahwar |

IB: Intensity Based
AB: Activity Based

### Report Content

| Report Portion | Contributing Team Member |
|---|---|
| Abstract | Subham |
| Introduction | Shahwar |
| Project Aims | Shahwar |
| Classifiers | Venkata |
| Intensity Based Classification | Shahwar and Subham |
| Activity Based Classification | Subham |
| Results and Conclusion | Subham and Shahwar |
| Future Work | Venkata and Shahwar |

# Contents

## Introduction

Human activity recognition and monitoring has become an important topic in recent times due to its increasing role in many sectors like healthcare, human-computer interactions, smart homes etc. A good representation of its efficacy is the study that shows that health condition of a subject can be specified with about 80-90 % of accuracy [3].

A wide variety of machine learning algorithms have been used for human activity recognition. Many studies compare these algorithms for a particular application, e.g. classifying activities based on the data from individuals' smartphones, focusing on the inherent imbalance of data in an individual's daily routine [5]. It shows Support Vector Machines (SVMs) performed the best with multiple data split proportions. But in another study [6] Random Tree Classifiers worked much better than SVMs for an activity recognition. Thus, there is no algorithm out there proved to work best for activity recognition. A model's performance is highly specific to the kind of data available for training.

Most of the research done for activity recognition have used windowing methods to divide the sensor signal into smaller time segments (windows) [7]. Various features were extracted for each window, and finally, classification algorithms are applied separately to each window. For this study, before activity recognition using windows, a classification layer to classify the intensity was proposed, the results which then would be used for activity recognition. The rationale behind this is that the intensity classification layer would act as a filter for the activity recognition, which would avoid activities of different intensities being mislabeled with each other

For this project a publicly available benchmark dataset titled PAMAP2 (Physical Activity Monitoring in the Ageing Population) is being used. It has also been used for other research studies as well. According to [4], supervised learning algorithms like kNN, bagging, and boosting have better accuracy on this dataset. Some researchers have used deep-learning methodologies as well; [8] and [9] have both devised innovative convolutional neural network based models; obtaining better accuracy than standard deep learning models like vLSTM, sLSTM and CNN.

This dataset includes the readings from tree IMUs installed on a subject's hand, chest and ankle, and a heart rate monitor. Each of the IMUs records accelerations and gyroscope readings in the x, y, and z axis. Time stamps are also included for each sample collected. The underlying correlation amongst the pool of activities the subjects perform and readings from each of these sensors can be used to extract a prediction model. This project is an attempt at that using multiple classifiers as an effort to find the best one.

## Project Aims

This section briefly restates the objectives put forward in the project proposal and briefly follows up with these, discussing the specifics of each of these project aims.

<u>Primary Aims:</u>

The project has primarily met the following aims:

1. **Classified activities based on their intensities:** Each activity out of the activity pool was assigned an intensity, based on the nature of the activity. In this first classification, the activity identifications were disregarded and classifiers were made to predict the intensities of the activity being performed. *Table 1* shows the intensity assignment to the pool of pre decided activities.
2. **Classified activities into specific activities:** Using preprocessed time series data and the intensity predictions made by the first stage of classification (intensity based), classifiers were

developed to predict a specific activity from the pool of activities given in *Table 1*. As second stage classification, each classifier was used on 4 sets of data, which was divided based on the intensities assigned to them in the first stage.

3. **Developed and deployed five classifiers in total:** The following classifiers were built and tested on the dataset being used:
    1. SVM for intensity classification
    2. K Nearest Neighbor for intensity classification
    3. Random Forest Classifier for intensity classification
    4. SVM for activity classification
    5. Random Forest Classifier for activity classification
4. **Evaluated the performance of the classifiers used:** All classifiers included were evaluated using the R2 and Classification Accuracy scores. Since all classifiers are multi-class classifiers, ROC curves could not be used directly, and there wasn't enough value to build multiple one-vs-rest ROC curves for this specific application, based on the confusion matrices seen for each evaluation.
5. **Analyzed and compared algorithms:** An analysis was carried out, based on the evaluations, and conclusions were drawn regarding the most useful algorithms.

<u>Extended Aims:</u>

An attempt was also made at classifying the heart rate from the data set instead of the intensities or the activities. For this, a linear regression model was developed. Activity IDs were not considered as a feature but intensities were. Interestingly, feature selection did not drop any of the features so the regression model was built around all 48 features in the dataset. Like the classifiers, the regression model was also evaluated using both within-subject and cross-subject cross validation methods to get the mean square errors and R2 scores for the model. Results of the attempt are discussed in later in the 'Results and Conclusion' section.

# Methodology

## Classifiers

For activity based and intensity based classifications, the algorithms used are:

### Support Vector Machine (SVM)

Support Vector Machine is a supervised classification technique that uses distance between class boundaries, in form of a hyperplane, and data points to choose the best classification model. The hyperplane dimension needs to be shifted from one dimension to the Nth dimension in many situations when differentiation is not as straightforward. The shape of the hyper plane is known as a kernel. It is the functional link between the two observations, to put it more simply. It will give the data additional dimensions so we can more readily distinguish between them. The different kernel types available are.

1. Linear Kernels
2. Polynomial Kernels
3. Radial Basis Function Kernel
4. Sigmoid Kernel

**Random Forest Classification:**

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

**K Nearest Neighbors (KNN):**

The KNN technique calculates the chance that a data point will belong to one group or another based on which group the data points closest to it do. Because it doesn't make any assumptions about the distribution of the underlying data, it is regarded as a non-parametric approach. KNN, in short, seeks to identify the group to which a data point belongs by examining the data points around it.

## Intensity Based Classification

The first step of the bi-layer classification problem classifies data into intensities. This classifier can stand on its own as an activity intensity classifier and can be used further for a more specific classification, the activity itself in this case. This bi-layer strategy makes it easier for the second layer to classify activities since it already has labeled the input with an intensity. However, the disadvantage is that any inaccurate intensity classifications are carried forward to the activity classifications. Thus, the model performance very important for the final classification, since the activity based model performance cannot exceed the intensity based model performance.

### Data Preprocessing

The original dataset consisted of separate .dat files for each subject, which were brought together to form a single dataset. The dataset also consisted of transient activities, basically time of rest between 2 different activities. Since these data were irrelevant for current use, they were dropped. The 'Orientation' columns were also dropped, as they are considered invalid for use by the authors [4]. Since the heartbeat monitor had a sampling frequency of 9 Hz, while the sampling frequency of IMUs was 100 Hz, this resulted in missing data in the file for heartbeat. For this, 'bfill' imputation method was done, which basically imputes the missing values with the data from the previous row. The thought behind this is that the heartbeat would not change much in 0.11 seconds (1/9 Hz). After this, the intensity column was added to the data frame, where the intensities of all the 18 activities were manually assigned based on *Table 1*.

| Intensity | Class assigned in code | Activities in the intensity level |
|---|---|---|
| Rest | 0 | Lying, Sitting, Watching TV |
| Low Intensity | 1 | Standing, Walking, Computer Work, Car driving, Ironing, Folding Laundry |
| Medium Intensity | 2 | Nordic walking, Ascending stairs, Descending stairs, Vacuum Cleaning, House Cleaning |
| High Intensity | 3 | Running, Cycling, Playing Soccer, Rope Jumping |

***Table 1:*** *Activity pool based on intensities assigned*

Once this was done, there were still a few missing values in the dataset. But since they were very few (<0.1% for each column), the imputation was done by using median values of the feature.
The resulting data frame consisted of ~2.7 million rows of data, with 50 columns (46 columns of sensor data, and 4 more columns of timestamps, activity id, subject id and intensity). Implementing the developed intensity classification algorithms on this dataset turned out to be very time consuming (> 5

hours for Random Forests and K-Nearest Neighbor, >8 hours for Support Vector Machine, without results). To improve computational efficiency, for every consecutive 3 rows in the dataset, the mean of the sensor signals was taken to create a shortened dataset of ~0.9 million rows. This dataset was then used for intensity classification, which considerably reduced the time taken (~4 hours).

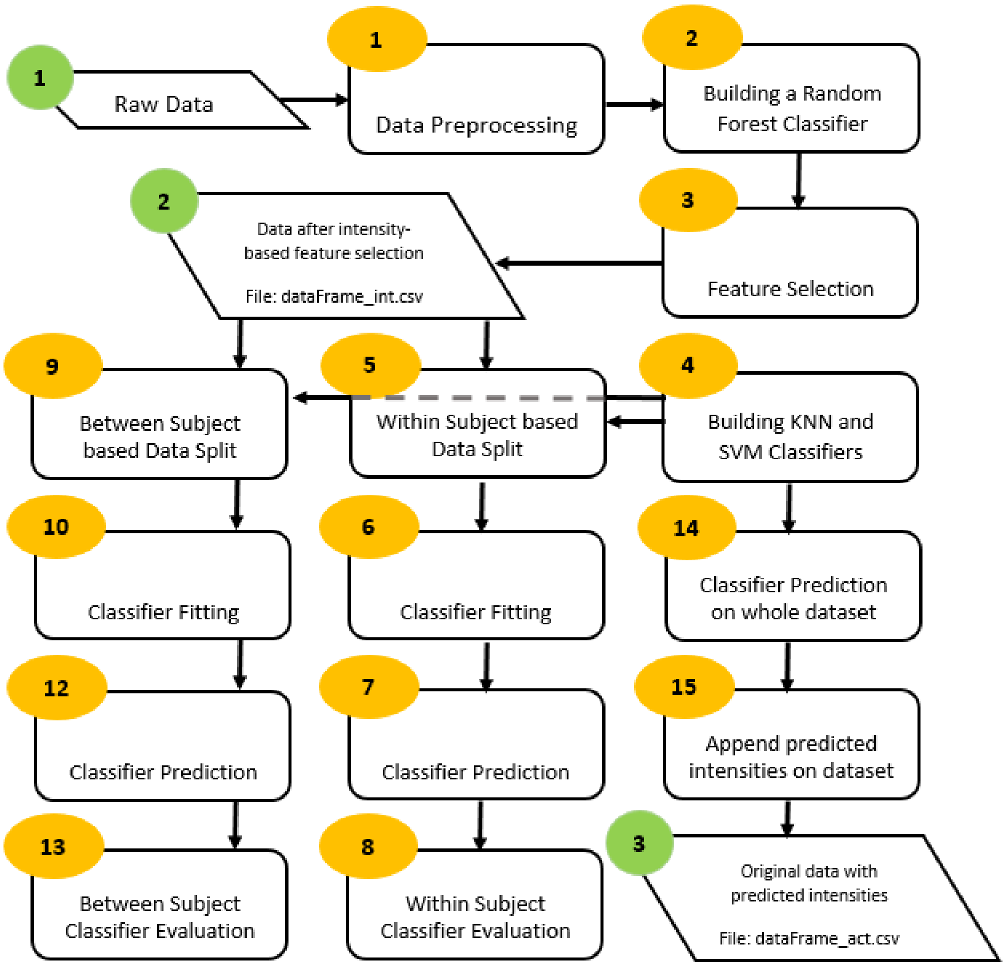


***Figure 1:*** *Flowchart for intensity based classification*

**Feature Selection**

From the preprocessed dataset, the activity ids are singled out and saved to be later added to the final data frame again. Next, the subject id and intensity columns are also dropped because these are identifying information and the target variable, respectively. After a random data split, RFECV (Recursive Feature Elimination with Cross-Validation) feature selection method was used to extract the optimal features in the dataset given. It is a backward feature selection method that eliminates features that don't contribute in optimizing the classifier's performance. The feature selector is built upon a Random Forest Classifier once, with the number of estimators kept at 10 and the criterion being entropy. A Random Forest Classifier is used because of its efficacy for variable importance analysis [2]. The features selected from this are used moving ahead in the program. This skips the feature selection that would have to be conducted every time a new model was to be fit, which in this case would have been 9 times for each within subject and between subject evaluations. This would be computation and time intensive.

To minimize and randomize data leakage, the feature selector was based on a random split of the data. This split was not used in the future but the test data of the all future splits (for within and between subject evaluations) would have seen some of the data through the selector. This is a judgment call made where less computation time is preferred over the potential for data leakage. The selector selects 16 features from the given set of 47 features, excluding the IDs assigned to each subject. This feature-selected dataset was extracted to be used further in the program as shown in *Figure 1*.

**Classification**

Once the initial raw data was preprocessed and trimmed down using feature selection, it was used for the first classification: intensity based classification. To avoid leakage data was split first and then normalization was carried out. The feature-selected dataset was concatenated with the ids column, which was dropped in the feature selection step. This is because the subject ids are required for the iterative evaluation methods being used for each classifier. For this classification, the following classifiers were tested out:

1. SVM with C = 10.0, kernel = linear, decision_function_shape = 'ovr'
2. Random Forest Classifier with criterion="log_loss", bootstrap="False", n_estimators=20, min_samples_leaf=3
3. KNN with n_neighbors=5, metric='minkowski', p=2

Then the predicted intensities are used as an initial classification for the activity based classification (discussed later). Thus, the accuracy here is translated into the model accuracy for the activity based classification. As shown in *Figure 1*, another data Frame in extracted after the classification to be used further in the program.

**Evaluation**

All of the classifiers used were evaluated using two cross validation techniques: within subject and cross subject. These methods are usually used when application or data can be divided into subjects, which in this are the participants. For within subject (WS) evaluation, one subject (participant) is considered. Train and test data are extracted from the data for the subject, based on which a classifier is developed and evaluated. The same is done for all subjects in the data and the performance metrics are then averaged to get the final performance metrics. Within-subject is good at picking up heterogeneous trends within a subject that may be lost with multiple subjects in the data set.

For cross subject evaluation, one subject's data is used as the test subject and the rest of the data is used for training. A model is fit around this split and evaluated. This is done iteratively, changing the test data subject each time until all subjects have been part of the test, also popularly known as Leave-one-subject-out (LOSO) cross-validation. This evaluation makes it sure that the model doesn't have subject bias, i.e. there is no relationship picked up between subject identity and the target variable. This is also how the program will actually be used; it will receive a new subject's data that it has not seen before, giving a better representation of how the program will perform in implementation. Like within subject evaluation, the final performance metrics are a mean of the performance metrics from each iteration. *Figure 2* is a representation of these two methods.

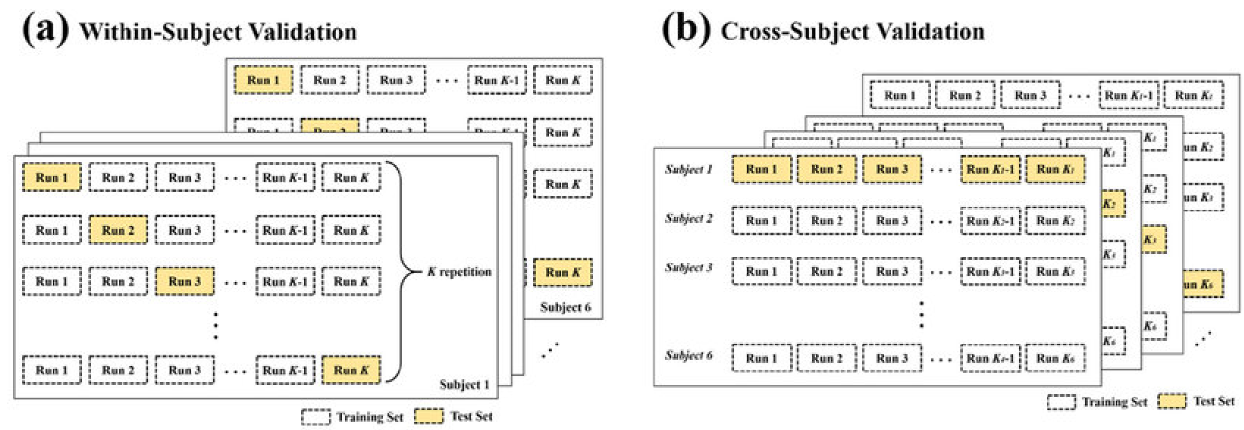


*Figure 2: Data split explained for within and cross subject evaluations. Image taken from [1]. Partial image of the original used to only include relevant material.*

The measure of evaluation was the classification accuracy, which is:

$$Classification\ Accuracy = \frac{Number\ of\ correct\ classifications}{Total\ number\ of\ samples}.$$

For the heart rate regression model, R2 and Mean Square Error were used for evaluation. Where,

$$R2 = 1 - \frac{Sum\ of\ square\ of\ residuals}{Total\ sum\ of\ squares}.$$

$$Mean\ Square\ Error = \frac{\sum_{i=0}^{n} Observed\ values_i - Predicted\ values_i}{Total\ number\ of\ samples(n)}.$$

These two metrics were not used for the classifiers, since these are better representatives of regression models' performance. Results of all evaluations are given in the 'Results and Conclusion' section.

## Activity Based Classification

Once the predicted intensity labels were generated, these results were used for time-series based activity classification. The dataset was first divided into 4 subsets for each classified intensity, and these 4 subsets were divided into smaller time periods of fixed intervals. Then for each of these small time periods, or windows, various statistical, energy, frequency and correlation-based features were generated. These feature vectors for each window were stacked to create a new time-series processed dataset.
Once this dataset was generated, Random Forests and SVM algorithms were implemented to get the activity label for each window.

The model was evaluated by 2 cross validation methods, leave-one-subject-out (LOSO) and within subject (WS) evaluation. Feature selection was also done for each iteration of both the evaluations using correlation and RFECV (Recursive Feature Elimination with Cross-Validation) methods.

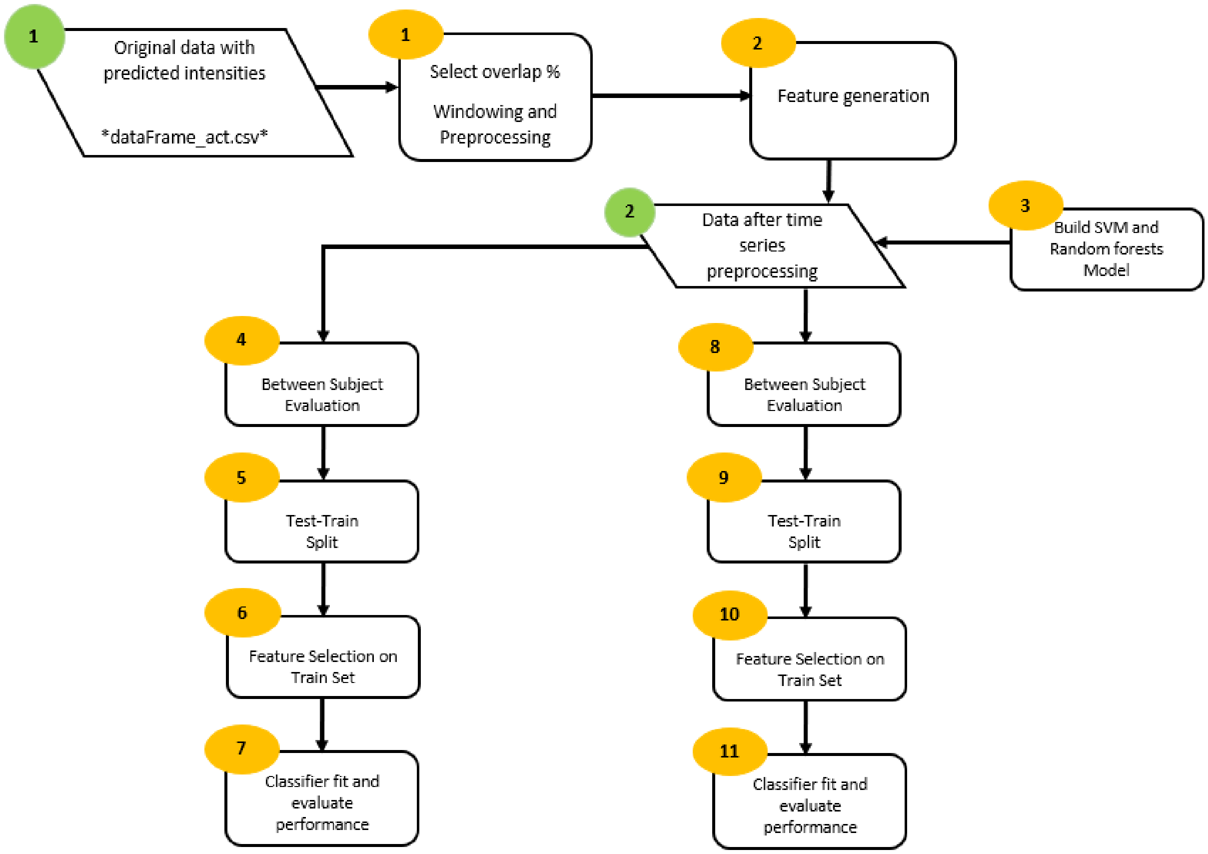


***Figure 3:** Flowchart for activity classification using results of intensity classification*

### Windowing and Preprocessing

A sliding window method of fixed length was used to divide the dataset into smaller subsets. This is advantageous because of the following two reasons: first, this allows us to capture how acceleration and position of hand, ankle and chest change with time. Since these acceleration and position profiles are different for different activities, capturing features of these profiles can enhance activity classification. Secondly, this reduces the dataset size by a considerable amount, leading to quicker runtime. While activity-defined windows and event-defined windows have been used in previous research, sliding window is ideally suited for real-time applications because of its implemental simplicity, and hence was chosen for this model.  A range of window sizes have been used in previous studies from 0.25 s[10]  to 6.7 s[11]. The window size should be sufficient to capture multiple cycles of most activities on this list, motivated by this fact and looking at previous research done, a window of 5 seconds was chosen. Feature extraction on sliding windows with 50% overlap has demonstrated success in past works [12,13], hence this was selected. Overlap of 30% and 70% was also considered to test the performance. The windowing was done for each subject separately, i.e. any particular window would have 5 seconds of signal data of only 1 individual. Once the window subsets were generated, features for each window were generated.

### Feature Generation

Previous activity classification studies have used a wide range of approaches to generate features which characterize windows of body-fixed sensor data, which were then used as inputs to classification schemes.

For this study, time-domain, frequency domain, energy and correlation based features were generated. Time-domain features are simply statistical measures of the window of the sensor data. Based on previous studies in activity recognition [14], the features developed in this study includes mean, range, max, min, standard deviation, median absolute deviation, interquartile range, skewness, kurtosis and harmonic mean of the sensor signals. In order to derive frequency-domain features, the window of sensor data was first transformed into the frequency domain using a fast Fourier transform (FFT) method. Mean frequency, max frequency and wavelengths were calculated along with other statistical attributes. Mean and absolute energy were also calculated using the FFT data. A study has shown that features that measure acceleration correlations between axes can improve recognition of activities involving movements of multiple body parts[15], and hence the acceleration correlations were also calculated. Using all the above feature generation techniques, a total of 1300 features were generated for each window.

**Classification**

Once the features were generated for all the windows, they were stacked to form a new time-series processed dataset, which was then normalized to a 0-1 scale. This dataset was used as an input to the classification algorithms to generate the activity labels. The following two classifiers were used:

1. Random Forest Classifier with criterion="log_loss", bootstrap="False", n_estimators=20, min_samples_leaf=30
2. SVM with C = 1, kernel = linear, decision_function_shape = 'ovr'

Previous studies on activity recognition using acceleration data have shown great results for decision-tree and random forest based classifiers [7], with classification accuracies above 85%, hence random forest classifier was chosen. While Random forest classifiers have been widely used and studied, there haven't been many studies involving SVMs. Hence, SVM was decided as the second algorithm to test its effectiveness in activity recognition.

**Feature Selection and Evaluation**

Based on the evaluation method, the time-series preprocessed dataset was then divided into test and train subsets, out of which the train set was used for feature selection and classifier fitting, and finally evaluated on the test set. Two types of evaluation methods were used, the in-between subject (LOSO) and within subject evaluation, as described in the intensity-based evaluation. The final evaluation metrics were calculated by taking the weighted average of the metrics for all the iterations of the evaluation with the number of test set samples. For effective classification, it is important to identify a set of features which have high discriminative ability. One of the time-efficient methods of feature selection is correlation-based feature selection [16], especially when the number of features is very high, which is the case in hand. Hence, features with high correlation (>0.8) were removed except 1 i.e. if feature 'a' has high correlation with features 'b' and 'c', then only one of features 'a', 'b' and 'c' was kept. Further feature selection was done using the RFECV (Recursive Feature Elimination with Cross-Validation) selector, which was built upon a Random Forest Classifier.

# Results and Conclusion

For the intensity-based classification, Random forests and K-Nearest Neighbor algorithms gave similar results, across LOSO and within subject evaluation as shown in *Table 2*. It was observed that the models worked extremely well when applied on a single subject, therefore the within subject accuracy exceeds 99.8% for both the models. The SVM model, on the other hand, had a significantly higher accuracy for LOSO evaluation, but slightly lower accuracy for within subject (WS) evaluation. The predictions on the overall dataset were considered as the base for the next step, i.e., activity classification. On the overall dataset, the KNN and Random Forests models had high classification accuracy (97.3% and 98.1% respectively).

| Classifier | Classification Accuracy | | |
|---|---|---|---|
| | Within-Subject | Cross-Subject (LOSO) | On Full Dataset |
| SVM | 0.98358 | 0.76403 | 0.81845 |
| Random Forest | 0.99995 | 0.59484 | 0.98091 |
| KNN | 0.99867 | 0.61894 | 0.97303 |

***Table 2:*** *Intensity based classification scores*

For the activity classification, the SVM model consistently performed better than the Random Forests model for within-subject, while the LOSO evaluation scores were much better for the Random Forests model as shown in *Table 3*. Also, since the developed SVM model was the worst performing for intensity classification, the resultant models (L1: SVM, L2: SVM and L1: SVM, L2: RF) have lower accuracies compared to others. Overall, it was observed that Random Forests or KNN for intensity classification, followed by SVM for activity classification (L1: KNN/RF, L2: SVM) have best Within Subject scores (~98.6%). In general, none of the models were up to the mark for LOSO evaluation with the highest accuracy recorded at 75.93% for when both intensity and activity classification was done using Random Forests (L1: RF, L2: RF). Considering accuracies of both evaluations, K Nearest Neighbors model for intensity, followed by Random Forests for activity classification turns out to be the best (Highest average for LOSO and WS).

| Intensity Classifier | Activity Classifier | Evaluation type | Classification Accuracy | | |
|---|---|---|---|---|---|
| | | | Overlap % | | |
| | | | 30 | 50 | 70 |
| Random Forests | Random Forests | WS | 0.8524 | 0.9263 | 0.9621 |
| | | CS/LOSO | 0.7593 | 0.7514 | 0.7445 |
| | SVM | WS | 0.9614 | 0.9763 | 0.9858 |
| | | CS/LOSO | 0.7148 | 0.6970 | 0.6964 |
| KNN | Random Forests | WS | 0.8597 | 0.9338 | 0.9688 |
| | | CS/LOSO | 0.7486 | 0.7371 | 0.7380 |
| | SVM | WS | 0.9555 | 0.9761 | 0.9869 |
| | | CS/LOSO | 0.6882 | 0.6818 | 0.6824 |
| SVM | Random Forests | WS | 0.7825 | 0.8883 | 0.9234 |
| | | CS/LOSO | 0.7049 | 0.6989 | 0.7157 |
| | SVM | WS | 0.9450 | 0.9665 | 0.9768 |
| | | CS/LOSO | 0.6191 | 0.6024 | 0.5970 |

***Table 3:*** *Activity based classification scores, where WS: Within Subject and CS/LOSO: Cross Subject/Leave One Subject Out*

It was also observed that the WS evaluation scores increased as the window overlap percentage was increased from 30% to 70%, but there was no such trend for LOSO evaluation. In WS evaluation, since only observations from a single subject are split into test-train sets and then evaluated, the dataset for each iteration is considerably smaller than the case of LOSO evaluation, where every iteration's train set is all the observations of 8 subjects, and the test set is 1 subject. When the overlap percentage is low, this dataset for WSE doesn't have sufficient observations to fit the model properly, and as the overlap is increased, this problem diminishes. An example of this is shown in *Figure 4*, the number of samples in the test set increases from 50 to 115 as we increase the overlap percentage from 30% to 70%, leading to increase in accuracy from 86% to 98.2%. The dataset for LOSO evaluation is already big enough to begin with, and thus, increasing the observations doesn't improve the model's accuracy.

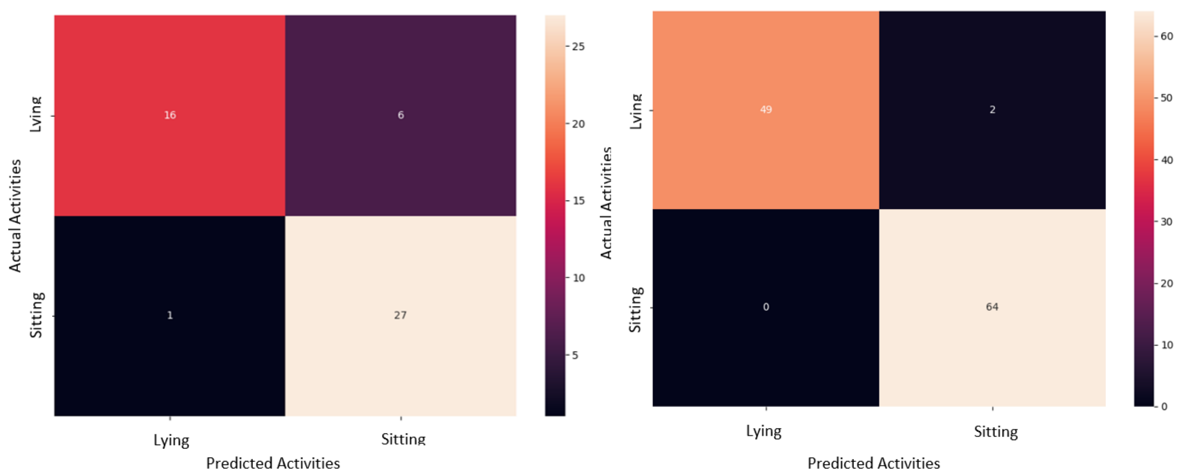


***Figure 4:*** *Increasing overlap from 30% (left) to 70% (right) leads to increase in accuracy*

While the designed models haven't worked sufficiently well for LOSO evaluation, a few factors from the dataset contribute to this. Two of the activities, 'Watching TV' and 'Car Driving' are performed by only 1 subject, hence when that subject was taken as test during LOSO, the model can't predict these activities as they are not present in the training set at all. *Figure 5* shows the confusion matrix for this case. A few other activities like 'Computer Work' and 'Folding Laundry' were performed by only 3-4 subjects, and when one of those subjects are put into the test set, there isn't sufficient observations in the training, leading to decrease in accuracy.

Another interesting result obtained, as shown in *Figure 6* is that the LOSO accuracy score drops for most models for 'low' intensity activities, while the WS score is highest for these low intensity activities. This implies that for these activities, the subject is extremely important for accurate recognition.

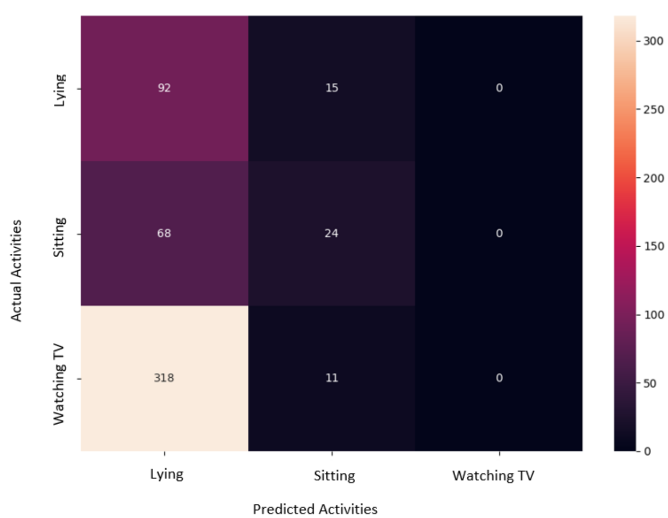


***Figure 5****: An iteration of LOSO evaluation where subject 109 is the test subject. Of the 9 subjects, only subject 109 performs 'Watching TV' activity. Since there are no cases of 'Watching TV' in training set (Sub- 101 o 108), all instances of watching TV are misclassified*

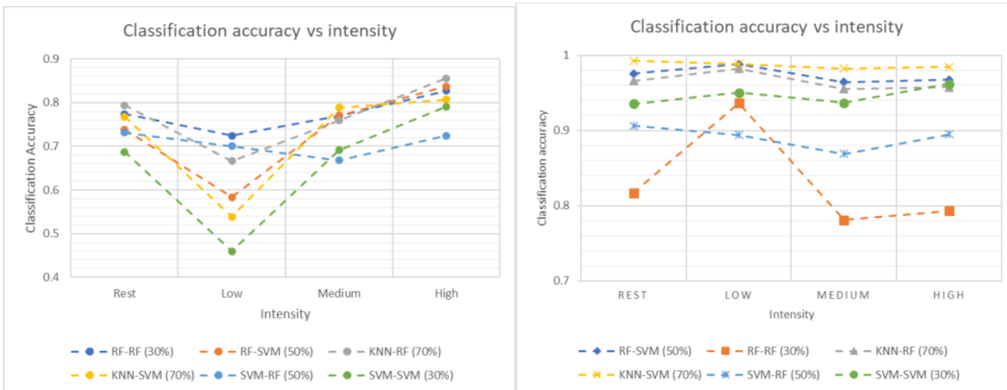


***Figure 6:*** *Accuracy vs intensity for LOSO (Left) and WS (Right)*

The results show that SVMs have given a great accuracy reading. These accuracy measurements should be great for this specific application as whenever the model is put into application, there will be a stream of time series data. Thus even if the classifier gets the activity wrong for one sample it will get it right for samples after and before that sample. Leading to an accurate result overall.

Also, as long as the classifier is not used for sensitive applications, such as medical diagnosis, the level of inaccuracy this best model shows should not be harmful.

The heart rate regression model's performance is given in *Table 4*. The R2 score is good for both within subject and cross subject evaluation. This indicates that the predictions made are not as far from the observed values of the heart rate. This is an indication that the data used from the IMUs is also capable of predicting heart rates when a subject performs different activities.

The interesting bit here Mean Square Error that has a very high value for within subject which indicates very poor performance. While in Cross-Subject evaluation, the means square error shows great performance. A conclusion that can be drawn here is that the smaller data set size in within subject analysis introduces a bias the classifier built upon it. There might be some cases that the classifier was not able to train well on but it saw during testing.

| Evaluation Type | R2 Score | Mean Square Error |
|---|---|---|
| Within-Subject | 0.828550 | 114.696028 |
| Cross-Subject (LOSO) | 0.818173 | 0.111876 |

***Table 4:*** *Heart rate regression model performance*

## Future Work

Human activity recognition and monitoring will be a remarkable concept which will have many uses in the field of health, robotics, and many other futuristic anomalies. Considering in robotics the analysis of physical features of humans can be used to design the movements and actions for robots indicating the type of action it is performing. And coming to the clinical field this procedure can be used to monitor the conditions of old age people and also this kind of approach we used here with Machine learning can be used for clinical studies of human actions.

The classification algorithm model developed in this study can be modified to implement real-time activity recognition. The catch being, a few minutes of activities need to be performed first by the specific person to develop the train set, since the algorithm is not up to the mark when different subject is tested

(LOSO evaluation). Other methods should also be looked on for better activity recognition for different individuals.

The heart rate classifier can used as a healthy heart rate prediction mechanism. Given the readings from IMUs on a subject, the classifier can be built to predict what the heart would be if the person is considered "healthy". The difference between the predicted heart rate and observed heart rate can be used to detect any health abnormalities.

## References


[1] Jeong, Dong-Hwa & Jeong, Jaeseung. (2020). In-Ear EEG Based Attention State Classification Using Echo State Network. Brain Sciences. 10. 321. 10.3390/brainsci10060321.

[2] Chen, RC., Dewi, C., Huang, SW. et al. Selecting critical features for data classification based on machine learning methods. J Big Data 7, 52 (2020). https://doi.org/10.1186/s40537-020-00327-4

[3] Che-Chang Yang and Yeh-Liang Hsu. A review of accelerometry-based wearable motion detectors for physical activity monitoring. Sensors, 10(8):7772–7788, 2010.

[4] Attila Reiss and Didier Stricker. Creating and benchmarking a new dataset for physical activity monitoring. In Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments, pages 1–8, 2012.

[5] Saeed Ali Alsareii, Muhammad Awais, Abdulrahman Manaa Alamri, Mansour Yousef AlAsmari, Muhammad Irfan, Nauman Aslam, and Mohsin Raza. Physical activity monitoring and classification using machine learning techniques. Life, 12(8):1103, 2022.

[6] Joana Chong, Petra Tjurin, Maisa Niemel¨a, Timo J¨ams¨a, and Vahid Farrahi. Machinelearning models for activity class prediction: A comparative study of feature selection and classification algorithms. Gait & Posture, 89:45–53, 2021.

[7] Stephen J Preece, John Y Goulermas, Laurence PJ Kenney, Dave Howard, Kenneth Meijer, and Robin Crompton. Activity identification using body-mounted sensors—a review of classification techniques. Physiological measurement, 30(4):R1, 2009.

[8] Manuel Gil-Mart´ın, Ruben San-Segundo, Fernando Fernandez-Martinez, and Javier Ferreiros-L´opez. Improving physical activity recognition using a new deep learning architecture and post-processing techniques. Engineering Applications of Artificial Intelligence, 92:103679, 2020.

[9] Mutegeki Ronald, Alwin Poulose, and Dong Seog Han. isplinception: an inception-resnet deep learning architecture for human activity recognition. IEEE Access, 9:68985–69001, 2021.

[10] Huynh T and Schiele B 2005 Analyzing features for activity recognition Proc. Conf. Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies (Grenoble) pp 159–64

[11] Ling Bao and Stephen S Intille. Activity recognition from user-annotated acceleration data. In International conference on pervasive computing, pages 1–17. Springer, 2004.

[12] R.W. DeVaul and S. Dunn. Real-Time Motion Classification for Wearable Computing Applications. Technical report, MIT Media Laboratory, 2001.

[13] K. Van Laerhoven and O. Cakmakci. What shall we teach our pants? In The Fourth International Symposium on Wearable Computers, pages 77–83. IEEE Press, 2000.

[14] Baek J, LeeG, ParkWandYun B J 2004 Accelerometer signal processing for user activity detection Knowledge-Based Intelligent Information and Engineering Systems, Pt 3, Proceedings pp 610–7

[15] R. Herren, A. Sparti, K. Aminian, and Y. Schutz. The prediction of speed and incline in outdoor running in humans using accelerometry. Medicine & Science in Sports & Exercise, 31(7):1053–9, 1999.

[16] Maurer U, Rowe A, Smailagic A and Siewiorek D 2006 Location and activity recognition using eWatch: a wearable sensor platform Ambient Intelligence in Everday Life (Lecture Notes in Computer Science vol 3864) (Berlin: Springer) pp 86–102