Report by: Faiyaz Rahman
Student ID: 45303738

The Human Movement Recognition Project (HARP)

Introduction: The term IoT or Internet of Things, is common term that can be heard everywhere. But what does it mean? IoT is the keyword that covering a wide spectrum that related the digital world wide web of things to our physical realm through the widespread of data sensing technologies. In recent years, the advancement in sensors and machine learning capabilities sparked the interest of many organisations. The applications of such IoT innovations gave rise to the possibility to analyse datasets to track human movement behaviour. This research can be used in many fields such as: medical, military and security. This report consists of data extracted from the Human Activity Recognition Project and provides a detailed Engineering report of the SDLC processes involved in collecting, analysing and processing sample datasets and a summary of relevant interesting findings on human movement behaviour and how machine learning models were used to accomplish these tasks. Human movement recognition is a very challenging time series classification task that involves extracting data from 19 individuals performing 13 daily activities. The HARP utilises the concept of distributed devices collaborating with each other to fulfil common goals of recognising human movements. In accomplishing this challenging task of human movement recognition the agile practices of an SDLC was implemented to ensure the project was conducted in an efficient deadline driven environment.

SCRUM Design and Sprints:
In order to ensure the project was timely delivered a SCRUM and Sprint system was used. The project was run over a period of 4 weeks and utilised sprints to keep track of project. In an agile software tool Trello a series of user stories with backlog, user sores, sprints and releases ensured the project was on track. A burn-down chart of the sprints allowed the members working on the project to have a comparison of daily and weekly burn-down and expected burn-down.

The DaLiAc (Daily Life Activities) database contains information about 19 healthy subjects (8 female and 11 male participants). The Activities include: sitting, Lying, Standing, Washing Dishes, Vacuuming, Sweeping, Walking, Ascending Stairs, Descending Stairs, Treadmill Running, Bicycle on ergometer (50w), Bicycle on ergometer (100W) and Rope Jumping. Four sensors were placed on the right hip, chest, right wrist, and left ankle. Each sensor nodes consisted of three accelerometer axis: A1, A2, A3 and three gyroscope axis: G1, G2, G3. For the purpose of this project, the main focus was the chest sensors, Accelerometer data found in columns: 6,7,8 and Gyroscope data found in 9,10,11 which monitored the movement of the chest. At the initial stage of the project, all user stories were backlogged as seen in figure 1.1 and then were completed over 4 sprints.

Note: Trello burn-down charts do not consider factors such as not adding a task in time or adding a task at a later stage. Thus in order to observe work progress, burn-down charts of the sprints were calculated and portrayed on excel for the purpose of this project.

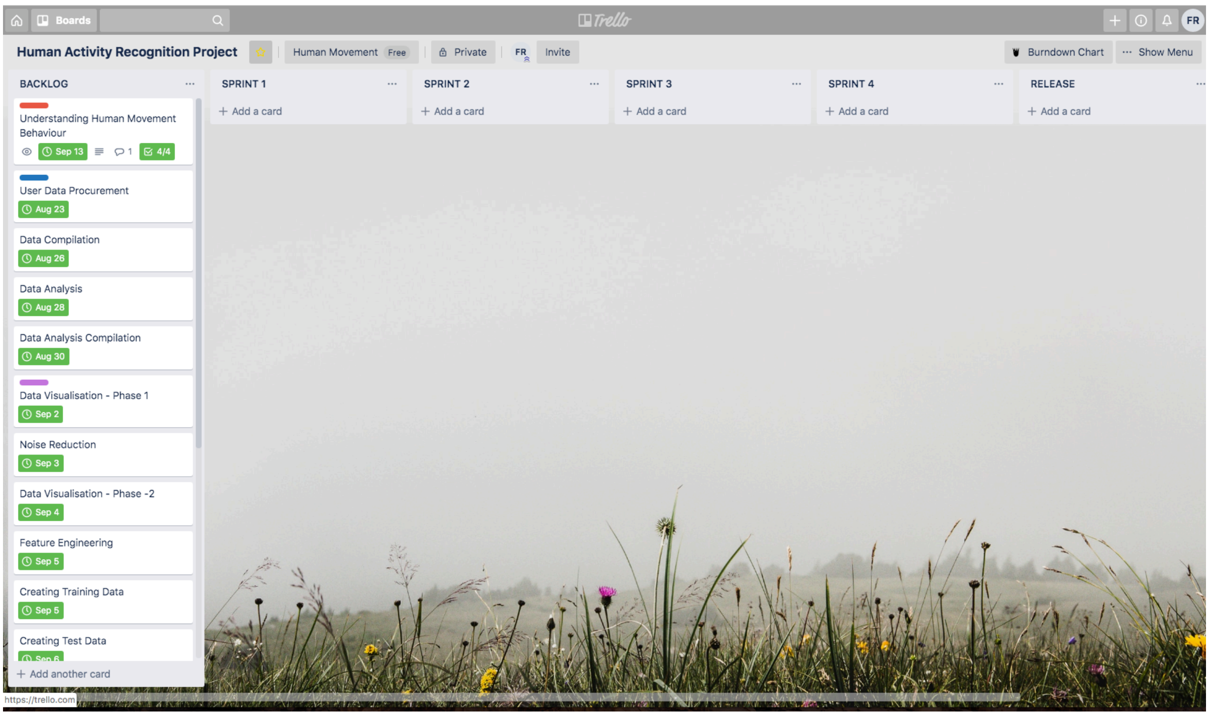

Figure 1.1

Report by: Faiyaz Rahman
Student ID: 45303738

Managing an IoT project can be difficult task as one to many tasks are being worked on at the same time. This a strict agile protocol can be seen in the example user story Figure 1.2 to ensure that the main goal of the task is being completed and to avoid any confusions.

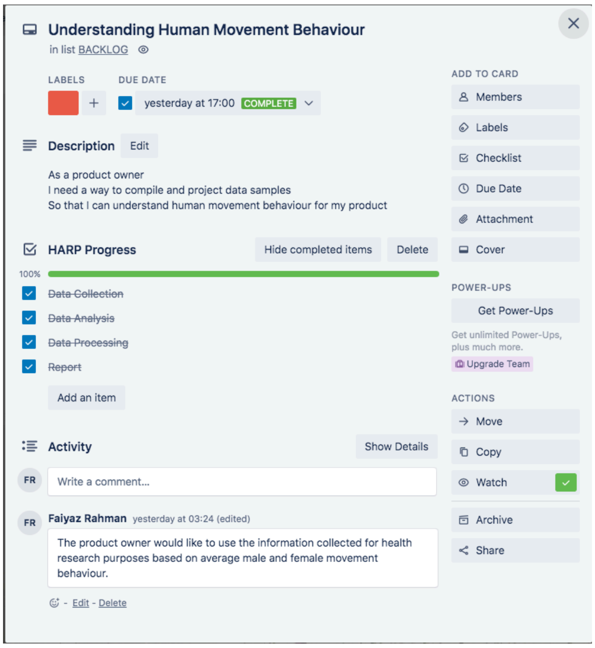

Figure 1.2

Sprint 1

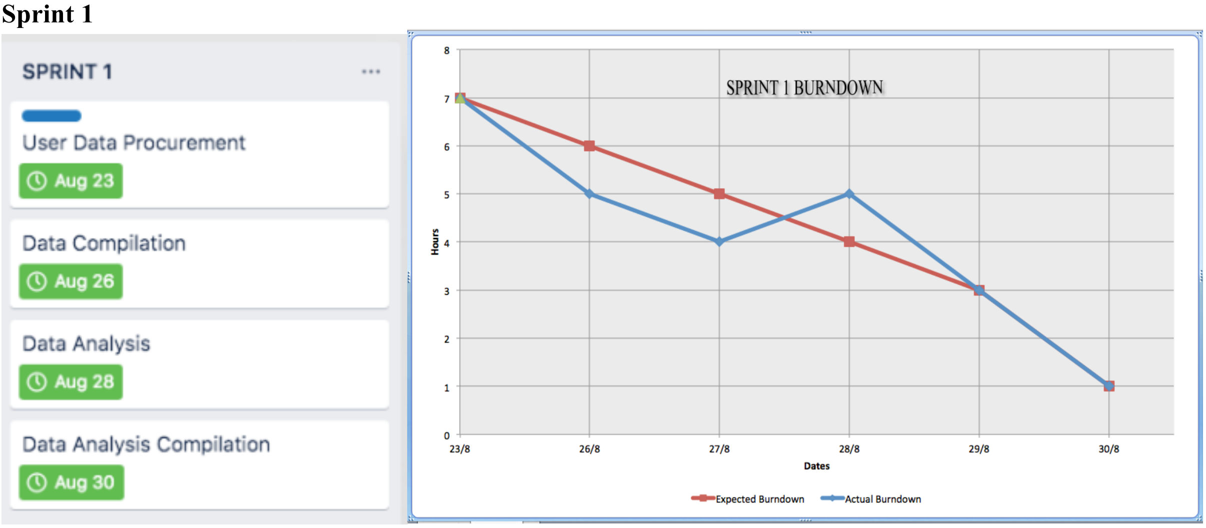

Figure 2.1

This Sprint consisted of 4 tasks over a period of 8 days and a total of 7 working hours. During this stage it was critical to understand the dataset in order to extract information from the DaLiAc database. User data procurement: Collecting 19 datasets and understanding the columns and labels associated with each activity. Data compilation: The 19 datasets needed to be loaded as one file in order to create feature extraction from all the data. Data analysis: The data was required to be analysed to ensure all the sensor data was recorded in and presented in the desired format. Data analysis compilation: After the desired data was analysed a new version of all the compiled data was stored in a desired format. An observation can be seen in the burndown chart (Figure 2.1)that in the early stages of the first week of the project, expected amount of hours were not worked. In order to ensure the sprint was completed on time, the extra hours were required and as a result it is shown in the graph that the expected burndown caught up to the actual burndown by the end of the estimated project deadline.

Sprint 2

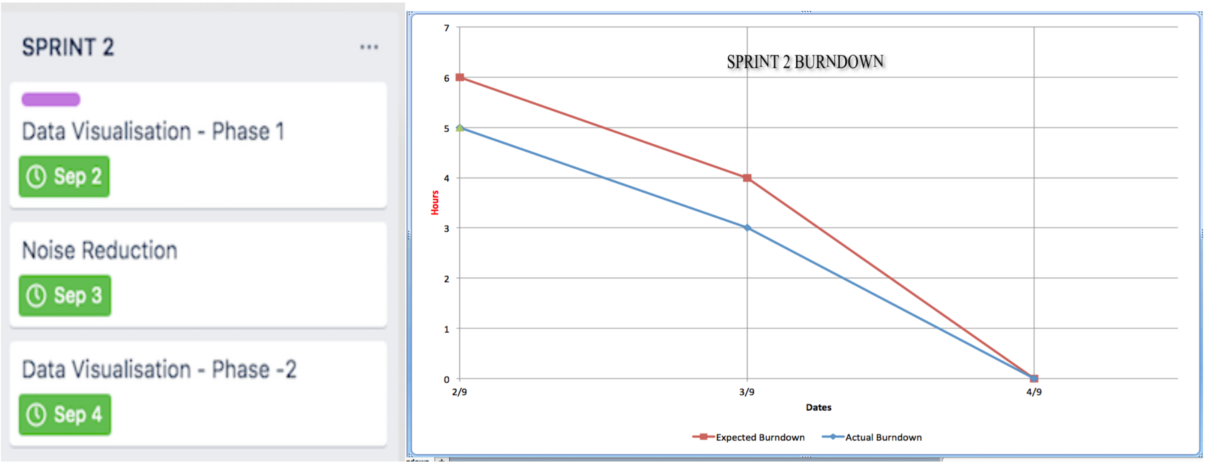

Figure 2.2

Sprint 2 was a short sprint however a very critical sprint. This print was allocated a total for 3 days. This Sprint tasks were allocated 6 hours estimated burndown chart Figure 2.2 however, the tasks required less time. In the burn down chart it is observed that the tasks required a lot less time than expected. For the purpose of this project a pair sensor movement data was visualised from the X,Y and Z axis of a Accelerometer and Gyroscope. This data contained a significant amount of noise so in order to achieve better results, the data was treated with noise reduction and visualised again.

Sprint 3

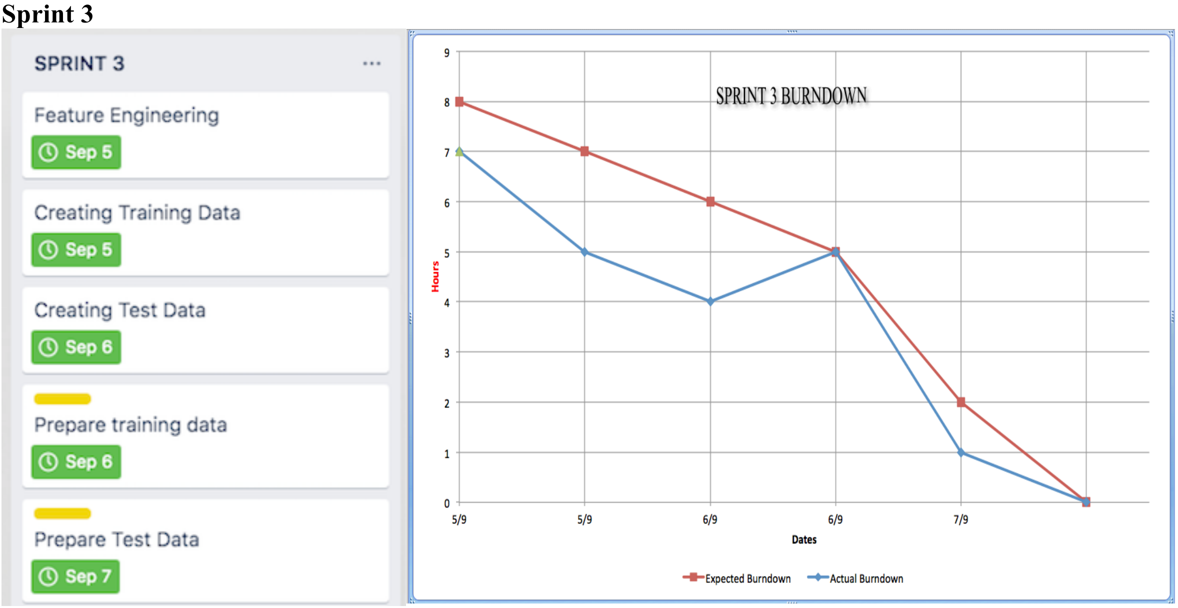

Figure 2.3

Sprint 3 contained the most tasks need to be completed and also consisted of the most critical information such which is feature engineering. A total of 8 hours were allocated over a period of 3 days and as it shown in the burndown chart Figure 2.3. The tasks seeming too less time than expected and produced the desired results by the end of the sprint. This sprint consists of creating and preparing test and training datasets for out machine learning models. This was a tedious process the 19 datasets for chest accelerometer sensors and gyroscope sensor data was used to create testing and training data files.

Sprint 4

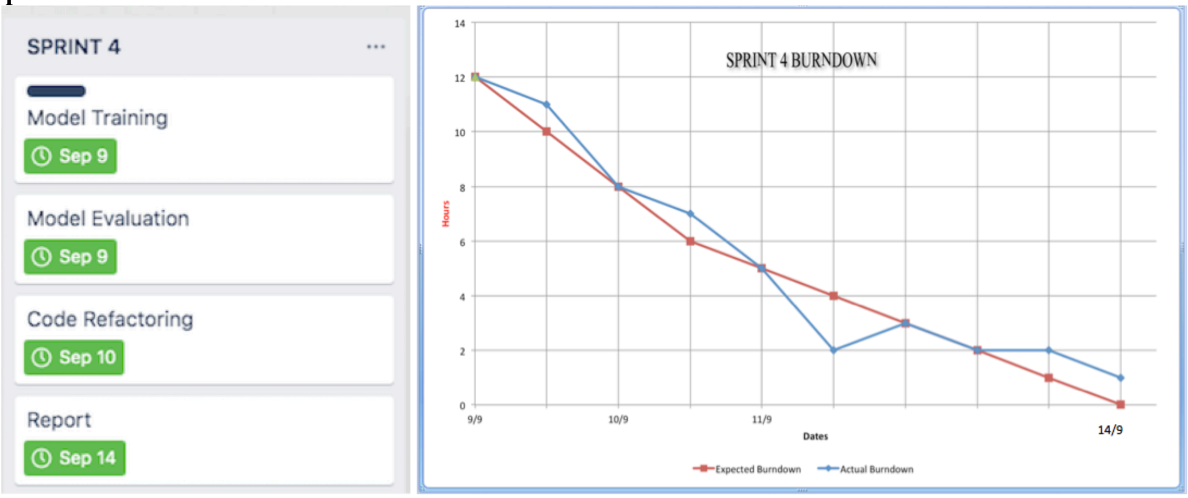

Figure 2.4

The last sprint of the project was conducted over a period of 6 days and 12 hours. It is observed that this Sprint experienced the worst burndown times compared to other sprints. This was caused by an extensive amount of time required to create the ML models using KNN and SVC classifications. The unexpected delays are reflected in the burndown chart Figure 2.4, which demonstrates how the delay in the first task caused a chain reaction for the following tasks. Regardless of any significant delays, the project was released on estimated time as shown in Figure. 3.1

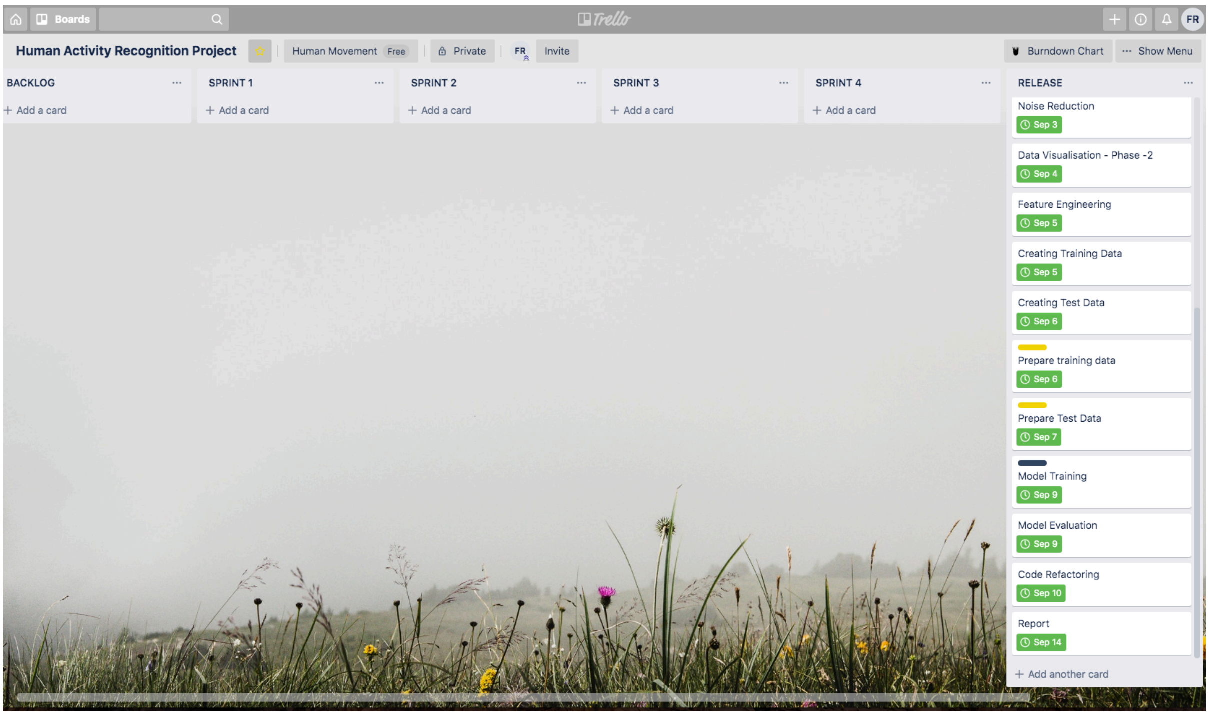

Figure 3.1

Implementation & Evaluation:
Implementation of Sprint 1 – Our target was to understand how certain activities preformed by our 19 participants cause their chest sensors to behave. This will allow us to distinguish a common trend or pattern amongst the participants. Our end goal involves using data visualisation and creating machine-learning model to observe these movement recognition activities. In order to manipulate and view the datasets, python code was used in jupyter notebook in order to compile all the datasets, analyse the data to check if all the datasets merged correctly and compile with our desired datasets.

Sprint 2 – Consists of visualising the data from all the 13 activities and 19 datasets. By reading the instructions from DaLiAc it is known that the chest sensor data for accelerometer and the gyroscope are contained in A1,A2,A3 → 6,7,8 and G1,G2,G3 → 9,10,11 respectively. Sample in figure 4.1

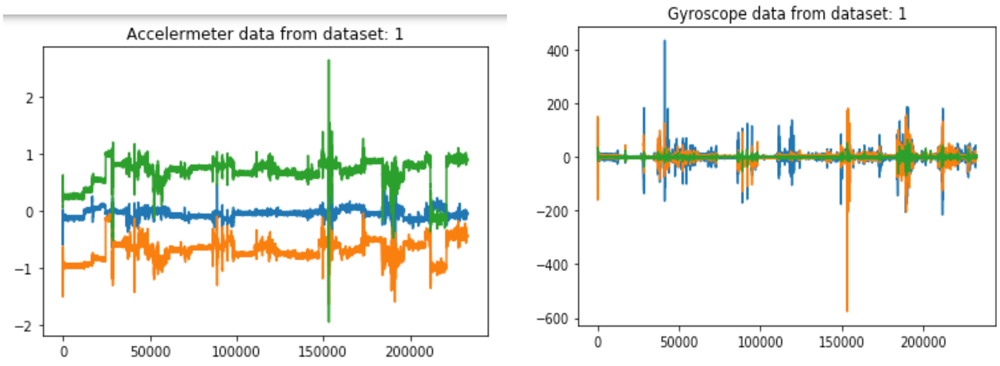

Figure 4.1

It is observed that the signal contains a lot of noise. Thus the Butterworth Low Pass filter was applied in order to set a cut-off limit and harness the data within a limit.

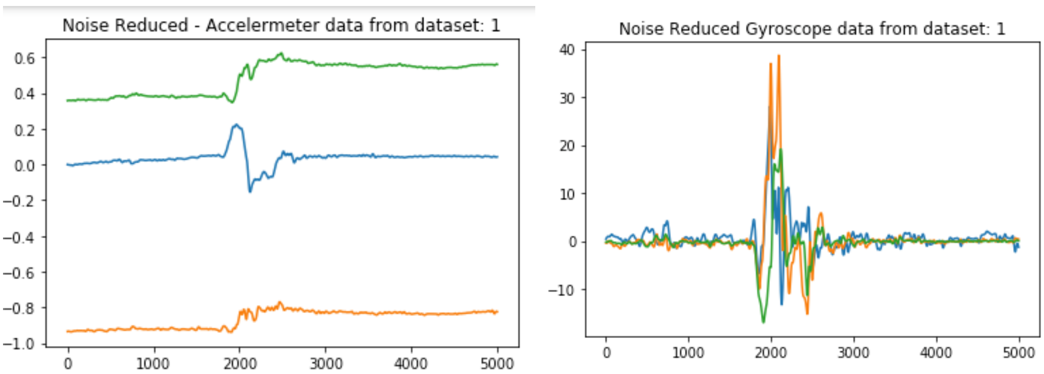

Figure 4.2

The Butterworth filter is designed to have the frequency response as flat as possible in the pass band. This is displayed in figure 4.2. The noise has been significantly reduced and the X and Y-axis values are much lower than the original values.

Sprint 3 – In the is sprint feature extraction took place. A training dataset and test dataset was created using a pre-processing scalar. The Robustscaler was used to pre-process the data, which removes the median and uses the interquartile range and follows the following formula for each feature:

$$\frac{x_i - median(\boldsymbol{x})}{Q_3(\boldsymbol{x}) - Q_1(\boldsymbol{x})}$$

This was followed by a data segmentation for time series-data and chest sensor used for min, max and mean values.

Report by: Faiyaz Rahman
Student ID: 45303738

Sprint 4 – After the test and train datasets were generated, the KNN algorithm and the SVC algorithm was used to generate a confusion matrix and make calculations for precision, recall, accuracy and F1.
KNN (K's Nearest Neighbour) is a learning algorithm that will utilise the train and test files and use data points to predict the classification of a new sample point. KNN results shown in figure 5.1.

On the other hand, a SVC (support vector machine for support vector classification) uses an algorithm to utilise the given data and output an optimised plane with categorises new samples. A different StandardScaler was used for this instance to observe a different output. SVC results shown in figure 5.2.

In order to understand these results, a classification report is generated that calculates the accuracy of the results, the precision that calculates the number of labels identified correctly, Recall – the percentage of labels that were correctly from all the correctly identified labels, The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. and finally Support – diagnoses the evaluation process and calculates the number of actual occurrences of the class the dataset.

From the confusion matrix below (using KNN figure 5.1) we can observe that the confusion matrix wasn't able to correctly classify any of the first activity, which was for sitting (indicated by 0). However it is observed that the second activity was recognised much better than the first activity (positively identified 22 activities from the chest sensors). The precision, recall and f1-score is also quite low indicated from 0.0 to 12.0 referring to the 13 activities respectively: sitting, Lying, Standing, Washing Dishes, Vacuuming, Sweeping, Walking, Ascending Stairs, Descending Stairs, Treadmill Running, Bicycle on ergometer (50w), Bicycle on ergometer (100W) and Rope Jumping.

```
Accuracy:  0.0710172744721689
[[  0  21   0   0   0  12   0   0   0  20   0   0   4]
 [  0  22   0   0   0  14   1   0   0  18   0   0   2]
 [  0  15   0   0   0   9   0   0   0  27   0   0   6]
 [  0   3   0   0   0   1   0   0   0  67   0   0  28]
 [  0   9   0   0   0   1   0   0   0  25   0   0  22]
 [  0   6   0   0   0   0   0   0   0  73   0   0   6]
 [  0   7   0   0   0   1   0   0   0 177   0   0  34]
 [  0   0   0   0   0   1   0   0   0  13   0   0  25]
 [  0  28   0   0   0   0   0   0   0   8   0   0   2]
 [  0   1   0   0   0   0   0   0   0  52   0   0  43]
 [  0   0   0   0   0   2   0   0   0  71   0   0  27]
 [  0   0   0   0   0   1   0   0   0  75   0   0  24]
 [  0   0   0   0   0   1   0   0   0  37   0   0   0]]
```

```
             precision    recall  f1-score   support

        0.0       0.00      0.00      0.00        57
        1.0       0.20      0.39      0.26        57
        2.0       0.00      0.00      0.00        57
        3.0       0.00      0.00      0.00        99
        4.0       0.00      0.00      0.00        57
        5.0       0.00      0.00      0.00        85
        6.0       0.00      0.00      0.00       219
        7.0       0.00      0.00      0.00        39
        8.0       0.00      0.00      0.00        38
        9.0       0.08      0.54      0.14        96
       10.0       0.00      0.00      0.00       100
       11.0       0.00      0.00      0.00       100
       12.0       0.00      0.00      0.00        38

avg / total       0.02      0.07      0.03      1042
```

Figure 5.1

The confusion matrix in figure 5.2 was created using Support Vector Classification, here it is observed that accuracy is much lower to the KNN accuracy. Secondly, the activities labelled from 0-12 (for activities 1-13) in the classification report is also much lower than the KNN results. Here also the SVC failed to classify majority of the activities correctly.

```
Accuracy:  0.036468330134357005

[[  0   1   0   0   0   6   0   0   0   0   0   0  50]
 [  0   0   0   0   0  17   0   0   0   0   0   0  40]
 [  0   1   0   0   0   6   0   0   0   0   0   0  50]
 [  0   0   0   0   0   0   0   0   0   0   0   0  99]
 [  0   0   0   0   0   0   0   0   0   0   0   0  57]
 [  0   0   0   0   0   0   0   0   0   0   0   0  85]
 [  0   0   0   0   0   0   0   0   0   0   0   0 219]
 [  0   0   0   0   0   0   0   0   0   0   0   0  39]
 [  0   0   0   0   0   0   0   0   0   0   0   0  38]
 [  0   0   0   0   0   0   0   0   0   0   0   0  96]
 [  0   0   0   0   0   0   0   0   0   0   0   0 100]
 [  0   0   0   0   0   0   0   0   0   0   0   0 100]
 [  0   0   0   0   0   0   0   0   0   0   0   0  38]]
```

```
             precision    recall  f1-score   support

          0       0.00      0.00      0.00        57
          1       0.00      0.00      0.00        57
          2       0.00      0.00      0.00        57
          3       0.00      0.00      0.00        99
          4       0.00      0.00      0.00        57
          5       0.00      0.00      0.00        85
          6       0.00      0.00      0.00       219
          7       0.00      0.00      0.00        39
          8       0.00      0.00      0.00        38
          9       0.00      0.00      0.00        96
         10       0.00      0.00      0.00       100
         11       0.00      0.00      0.00       100
         12       0.04      1.00      0.07        38

avg / total       0.00      0.04      0.00      1042
```

Figure 5.2

Version Control:

The successful run of the project required a series of version control methods such as: BitBucket, Github and Trello. These tools allow different releases and versions of the changes that took place throughout the project. Both github and Bitbucket provides version tracking through a history of commits. Additionally Bitbucket provides a diff check tool that allows for tracking exact changes in a comparison view of before and after an update

Github Repository

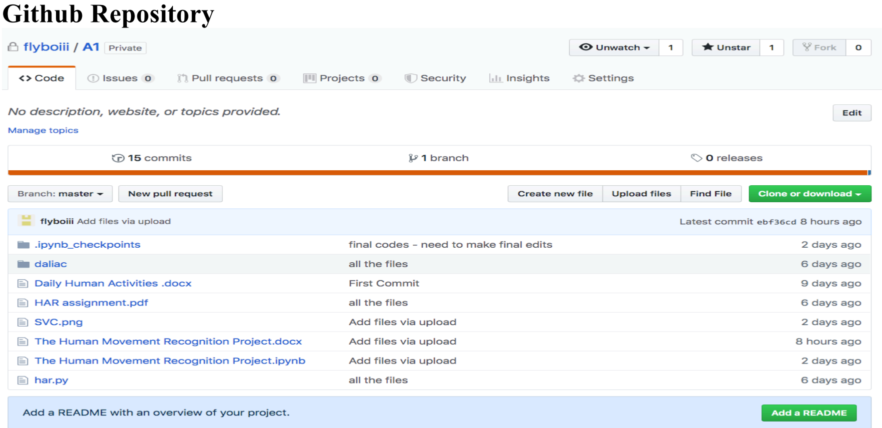

Figure 6.1

BitBucket Repository

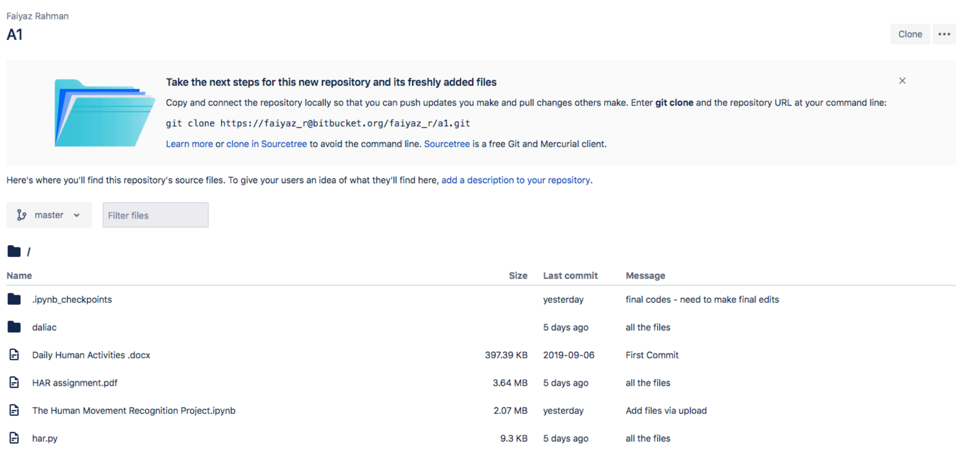

Figure 6.2

Bitbucket - Difference Checker to Track Updates/Changes

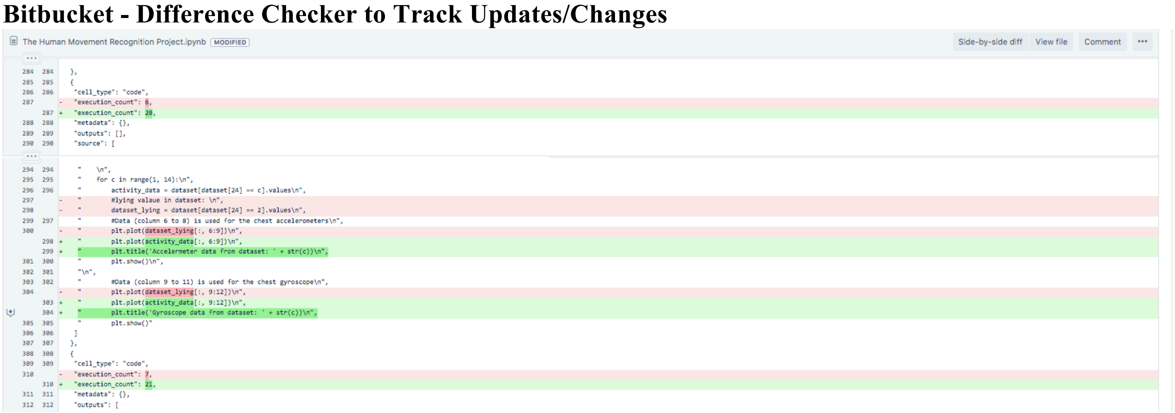

Figure 6.3

Report by: Faiyaz Rahman
Student ID: 45303738

Summary/Conclusion:

This project was a very challenging process of dealing with several datasets and required a lot of patience however by approaching this project with an agile project management techniques. The machine learning approach to recognise human movement behaviour comes with challenges of its own however provides an ample amount of things to learn. An agile approach enforces time constraints for a software development team and challenges them to accomplish these tasks in a deadline driven environment. Although this allows many processes to be expedited, there occurred situations during this project that the unexpected delay caused a chain reaction amongst the other tasks. The compilation of data allowed us to see how the tables were formed and what data we need to extract. Therefore allowing us to manipulate the data with the help of Python code to visualise the chest accelerometer and gyroscope sensors. However, the sensors were very sensitive as a result, the datasets corresponding to the sensors contained a lot of noise. Applying a low pass noise reduction filter known as the Butterworth filter easily solved this problem by prescribing an expected range. The most challenging part of a machine-learning algorithm is to analyse training and test data from a given dataset and test its ability to classify labels correctly. This correct procedure was followed when re-factoring python code to produce training set and test set though the classification report and the confusion matrix allows us to see that the scores were very low. This could be due to the following error:

```
/anaconda3/lib/python3.6/site-packages/sklearn/metrics/classification.py:1135: UndefinedMetricWarning: Precision and
F-score are ill-defined and being set to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
```

This indicates that there were no values to be recorded for certain column as observed in figures 5.1 and 5.2. While using the KNN and SVC classifiers the confusion matrices indicated the number of correctly identified labels for each activity. For the Human Activity Recognition Project it was not sufficient to recognise all a large set of activities. In future releases in order to train the model better, larger datasets can be experimented with including different parameters in order to achieve better results. This project was successfully completed on time by following strict agile methodology such as backlogs, sprints and burdown charts. Although the desired outcome of correctly recognising activities through machine learning was not achieved, a more valuable insight was gained of a deeper understanding of the machine learning approach of recognising human movement recognition.

References: A detailed list of all the resources and files used for this project can be found in the following repositories.

Bitbucket: https://bitbucket.org/faiyaz_r/a1/src/master/

Github: https://github.com/flyboiii/A1

Trello: https://trello.com/b/DKpNQeCU/human-activity-recognition-project

• **BROWNLEE, J.**

Deep Learning Models for Human Activity Recognition

In-text: (Brownlee, 2019)

Your Bibliography: Brownlee, J. (2019). *Deep Learning Models for Human Activity Recognition*. [online] Available at: https://machinelearningmastery.com/deep-learning-models-for-human-activity-recognition/ [Accessed 28 Aug. 2019].

• Website

DALIAC – DAILY LIFE ACTIVITIES

In-text: (https://www.mad.tf.fau.de/research/activitynet/daliac-daily-life-activities/, 2019)

Your Bibliography: https://www.mad.tf.fau.de/research/activitynet/daliac-daily-life-activities/. (2019). *DaLiAc – Daily Life Activities*.

• Website

FATMASARI RAHMAN, L.

Understanding IoT Systems: A Life Cycle Approach

In-text: (Fatmasari Rahman, 2018)

Your Bibliography: Fatmasari Rahman, L. (2018). *Understanding IoT Systems: A Life Cycle Approach*. [online] Elsevier. Available at: https://reader.elsevier.com/reader/sd/pii/S1877050918305106?token=FE11846CE87F900A19C34CB67DE5FBD0DC6128EF720190432B2DA1110423FEB9D7982E3280CA9C1097226EAD465D5DDE [Accessed 5 Sep. 2019].

• Website

MEIDAN, Y.

ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis

In-text: (Meidan, 2017)

Your Bibliography: Meidan, Y. (2017). *ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis*. http://delivery.acm.org/10.1145/3020000/3019878/p506meidan.pdf?ip=101.191.6.164&id=3019878&acc=OA&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E96ED85FD01F1F3FE&__acm__=1568462381_be2cbb081f22ccaa6494325f9763d0a9.

• Website

SKLEARN METRICS

In-text: (sklearn, 2011)

Your Bibliography: sklearn. (2011). *Sklearn Metrics*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html [Accessed 3 Sep. 2019].