Human Activity Recognition with Smartphones

University of Malta

Faculty of Information & Communication Technology

Bachelor of Science in Information Technology (Honours) (Artificial Intelligence) 2nd Year

ICS2000 – Group Assigned Practical Task

Dr Kristian Guillaumier

Sean Farrugia	Owen Agius	Francesca Maria Mizzi
258801(L)	496701(L)	118201(L)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Declaration

Plagiarism is defined as "the unacknowledged use, as one's own, of work of another person, whether or not such work has been published, and as may be further elaborated in Faculty or University guidelines" (University Assessment Regulations, 2009, Regulation 39 (b)(i), University of Malta).

We, the undersigned, declare that the Assigned Practical Task report submitted is our work, except where acknowledged and referenced.

We understand that the penalties for committing a breach of the regulations include loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected, and will be given zero marks.

Sean Farrugia	[illegible]
Student Name	Signature
Owen Agius	[illegible]
Student Name	Signature
Francesca Maria Mizzi	[illegible]
Student Name	Signature

UBSCITHARIFT	Human Activity Recognition with Smartphones
Course Code	Title of work submitted

24/05/2021
Date

Table of Contents

1 - Introduction 3
1.1 - Distribution of Work - Documentation 4
1.2 - Distribution of Work – Implementation 5
2 - Research and Literature Reviews 6
2.1 - A Public Domain Dataset for Human Activity Recognition Using Smartphones 6
2.2 - Training Computationally Efficient Smartphone–Based Human Activity Recognition Models 7
2.3 - Pros and Cons 8
3 - Implementation and Testing 9
3.1 - Support Vector Machines 9
3.1.1 - Radial Kernel 9
3.1.2 - Polynomial Kernel 9
3.2 – K-Nearest Neighbours 10
3.2.1 - Distance Weighted k-NN 10
3.3 - Decision Trees 10
3.4 - Logistic Regression 11
3.5 - Testing 12
4 - Evaluation and Critical Analysis 14
4.1 - Dataset 2: Six New Activities 15
5 - Conclusions 17
6 - References 17
6.1 - Table of Figures 19

1 - Introduction

The main scope of this Assigned Practical Task (APT) is to implement a Human Activity Recognition System (HAR System). The purpose of this system is to identify the action which the user would be doing, solely based off of the changes in motion of the user's body during the performance of specific actions.

Most HAR systems implemented would use specialised motion sensors that would be secured to the users' body, including, but not limited to, the waist, chest, arms and legs. However, the main problem with this type of system is the complex setup the user would be required to wear during the activity, in addition to the added expenses when purchasing these sensors. Considering the simplicity of the application, many users are more likely to get discouraged in using such a complex, albeit excessive, set up. As a result of the rapid advancements in the technological field as well as the efforts of many researchers, this setup has been reduced to needing only a smartphone. This initial set-up made use of a bulkier mounting system through the use of a belt, an aspect of the set-up which can be improved upon, with the users' comfort being the main priority.

Therefore, for our APT, we were aiming to develop a simple Human Activity Recognition prototype which only uses the built-in sensors found in an average smartphone and eliminating the use of a belt mount, allowing the user to carry their phone in their pockets. While this may result in less accurate predictions, it allows users to retain their usual habits; keeping their phone in their pockets.

Moreover, two separate datasets were gathered by the three members working on this APT. The first dataset was done to mimic that made in the paper [1] with six total actions: Walking, Walking Downstairs, Walking Upstairs, Sitting, Standing, and Laying. This dataset was created in order to compare the difference in results gathered and processed by Anguita et al. and ourselves. However, we also collected a second dataset in which we chose physical activities which were not included in the existing data. These also required body movement from the user and were recorded through the accelerometer and gyroscope sensors found in the smartphone. The final new activities are Cycling, Football, Swimming, Tennis, Jump Rope and Push-ups. In summary, the main aim of this project was not only to interpret the original six activities that most Human Activity Recognition papers tend to focus on, but also to recognise another six unique physical activities.

The process of classifying the data with a high accuracy can be divided into two steps: data collection and modelling. In order to collect a sufficient amount of data, the free app 'AndreoSensor' was used. Using this app allowed for the collection of data using the four main inertia sensors: gyroscope, gravity, accelerometer and linear acceleration. The data collected consists of roughly one hour worth of data for each of the 12 activities mentioned above, allowing for the model to be developed with an even distribution of data across all the categories.

After the data is collected, it is pre-processed. The pre-processing entails the removal of “NaN” and duplicated values while also generating statistical readings from the ‘csv’ file produced by ‘AndreoSensor’. After being processed, the data is analysed via a t-SNE algorithm which aids the visualisation of data clusters. Finally, the data is modelled and classified using four different supervised machine learning algorithms: Logistic Regression, Support Vector Machines, Decision Trees and K-Nearest Neighbours.

1.1 - Distribution of Work - Documentation

1 – Introduction	Sean Farrugia, Owen Agius, Francesca Mizzi
2 – Research and Literature Review	
A Public Domain Dataset for Human Activity Using Smartphones	Owen Agius
Training Computationally Efficient Smartphone–Based Human Activity Recognition Models	Francesca Mizzi
3 – Implementation and Testing	
Support Vector Machines	Owen Agius
Radial Kernel	Sean Farrugia
Polynomial Kernel	Owen Agius
K-Nearest Neighbours	Owen Agius
Decision Trees	Francesca Mizzi
Logistic Regression	Sean Farrugia
Testing	Sean Farrugia
4 – Evaluation and Critical Analysis	Owen Agius and Sean Farrugia
5 – Conclusion	Francesca Mizzi

1.2 - Distribution of Work – Implementation

Functions for Statistical Calculations	Sean Farrugia, Owen Agius, Francesca Mizzi
Functions for reading the datasets	Sean Farrugia
Function for processing raw data	Sean Farrugia
Function for splitting up data	Sean Farrugia, Owen Agius, Francesca Mizzi
Data Analysis	Sean Farrugia
Modelling Data	
Support Vector Machines	Sean Farrugia, Owen Agius, Francesca Mizzi
Logistic Regression	Sean Farrugia
K – Nearest Neighbours	Owen Agius
Decision Trees	Francesca Mizzi

2 - Research and Literature Reviews

2.1 - A Public Domain Dataset for Human Activity Recognition Using Smartphones

'A Public Domain Dataset for Human Activity Recognition Using Smartphones' [1] moves away from the traditional method of obtaining body readings and inertia sensor readings. Instead of setting up a variety of body worn sensors to collect the data, the researchers opted for a more comfortable and convenient approach.

Modern smartphones have built in gyroscopes and accelerometers. The use of smartphones with inertia sensors is an alternative approach for the Human Activity Recognition problem, as it makes the data collection procedure much more affordable, accessible and comfortable for the user. This approach also acts as a long-term solution for activity monitoring. An example of this would be a two-hour run. This activity, done using a large number of sensors attached to the body, would be very uncomfortable and impractical. Therefore, the use of smartphones nullifies this problem.

While this is an improvement when compared to the prior method of data collection, this new approach, brought forward by Anguita, Ghio, Oneto, Parra and Reyes-Ortiz, poses the issue of noisier data sets.

The researchers adapted a well-rounded approach for data collection which employs 30 volunteers ranging from ages 19 to 48. The volunteers were selected to perform 6 specific daily life tasks: standing, sitting, laying down, walking, walking upstairs and downstairs. This approach provides a good diversity of physical capabilities which allow a better-rounded classification of the activity.

The components collected via the inertia sensors range between acceleration, gravity, angular speed and magnitude, along with their respective frequencies. The readings are also computed to create a large number of statistics to assist in the classification by generalizing the readings into statistical readings.

Using a Support Vector Machine, Anguita et al. achieved a satisfying 96% accuracy in the classification of the test data. The researchers encountered challenges when classifying the stationary events: sitting, standing and laying down, resulting in a comparatively low 88% accuracy. This is due to a noticeable misclassification between the activities because of the very similar gyroscopic readings.

2.2 - Training Computationally Efficient Smartphone–Based Human Activity Recognition Models

Following the increasing popularity in using wearable systems in Human Activity Recognition systems (HAR Systems), Anguita et al. [2] wanted to take advantage of the rapid development of powerful processors already existing within smartphones. Companies had recently started adding sensors within smartphones such as the hybrid accelerometer and gyroscope in the latest iPhone 4. This showed researchers the benefits of adding gyroscope readings into HAR systems, observing an improvement in results, up to a 13.4% increase accuracy in classifying data.

An issue regarding implementing a HAR system using smartphones was discovered when using the K-NN classifier since it would not work in this application due to large dataset as well as the challenging computations. Thus, Anguita et al. chose to tackle two issues stemming from smartphone-based HAR systems: 1) The lack of a large gyroscope-based dataset and 2) A proper selection of useful features and effective models. The first issue was tackled through the creation of the HAR dataset, developed by the same researchers in the separate paper [1] described above, which contains a large amount of gyroscope-based data. The second issue was solved using two feature selection mechanisms as well as Support Vector Machine (SVM) models.

Since the creation of the HAR dataset was mainly described in another paper [1], not much detail is given other than that the data was collected from 30 volunteers carrying an Android smartphone around their waist performing a series of activities.

The target for the paper was to design a model which can effectively and efficiently run on the limited battery and processing power of a smartphone, which they finally chose to be a linear SVM model.

The experiments performed aimed to compare the performance of the SVM models based on linear and Gaussian kernels, with the result being that both models were substantially identical. However, they chose the linear approach since it was faster than the kernel approach.

2.3 - Pros and Cons

For our task, the greatest advantage is that there is a simple way to collect data and train the chosen supervised machine learning model. The cost isn't considered since the only hardware needed is a smartphone. Moreover, the prototype will be trained assuming that the phone will be placed in the side pocket of the user, allowing them to not be restricted by belt mount.

Should our prototype successfully become an app, many advantages can be seen. The main aim has been already established, that the app would be able to identify the action being done by the user. However, the user themselves would already know what they are doing, so a good usage of this application could be to inform other people what someone is currently doing while they are not in the same room.

It could be used to aid nurses in elderly homes to know what their patients are doing. For example, if they know that it is time for them to be asleep, but the system is still saying that the patient is currently walking, the nurse would be able to go check on the person to see what they are doing. This would be much more efficient than having to check the cameras of all the patients found inside the home.

Another great implementation to use this application for is to record your workout sessions. This can be done by our version due to the newly added actions in our model, including the activities that are more energetic. Thanks to the easy method of gathering data, more models can be easily be made to identify new activities. This could help fitness trainers make sure that their students are doing the workout that they gave them.

Although, no project is perfect it is to be expected that there are some negative aspects during this experiment, the most obvious being the accuracy of the prototype itself. Since the phone will only be capturing the data coming from the changes in accelerometer and gyroscope readings from the side pocket of the user, there is a lack of data from what the entire body is doing. For example, if an activity is being done where the upper body is mostly being used, it might be a bit difficult for the phone's sensors to detect such an activity.

Furthermore, to properly train the chosen model with the already collected and labelled data, a lot of data must be collected. If an activity is only performed a few times while collecting supervised data for the model, it might end up being undertrained due to the lack of data fed to the model. In Anguita[1], they had to hire 30 different people to collect data for them, where in our case we only had 3. On the other hand, it is important not to overfit the data in one activity compared to the rest. If an activity has hours of data collected while another only has a few minutes, it might confuse the model with unorganised data.

3 - Implementation and Testing

To tackle the classification problem at hand, we attempted to classify data using four different supervised machine learning algorithms. The algorithms which were adopted are:

- Support Vector Machines
- K Nearest Neighbours
- Decision Trees
- Logistic Regression

The above choices all approach classification differently and therefore adopting them all increases the probability for correct labelling.

3.1 - Support Vector Machines

The objective of the support vector machine algorithm is to find a hyperplane in the Nth dimensional space which distinctly classifies the data points. The hyperplane can, for example, be a straight line in 2-dimensional space and a plane in 3-dimensional space. Anything using greater dimensions would render the hyperplane difficult to visualise. To separate the two or more classes of data points, there are many possible hyper-planes that can be used. The objective of the support vector machine algorithm is to find a plane that has the maximum margin, which, simply put, is the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

3.1.1 - Radial Kernel

The Radial Kernel is one of the most popular kernels used for Support Vector Machines. This kernel uses the Radial Basis Function and is ideal when used for classification models with a large number of features, as these would be impossible to draw in Euclidean Space. In summary, the radial basis function converts the represented data into an infinite coordinate by using the Taylor Series in the equation $e^{-\gamma(a-b)^2}$ [11].

3.1.2 - Polynomial Kernel

Although the Radial Kernel is more widespread in Support Vector Machine classification, the Polynomial Kernel provides a different approach and procedure in the attempt to classify data. This kernel considers the given features of the data to determine their similarity, while also combining the data itself. Such combinations are known as interaction features [4]. An advantage of implementing this approach is the consideration and full expansion of the kernel prior to evaluation. Contrastingly, the polynomial kernel can suffer numerical instability in cases where the definition is usually either 0 or infinity with an increasing degree [5].

3.2 – K-Nearest Neighbours

The K-Nearest Neighbours assumes that similar things exist in close proximity. The aim of the k-NN algorithm is to classify a new data entry given a predetermined clustered set. The new data entry is classified by comparing the distances from the other labelled variables. The distance is generally measured using Euclidean distance, but it can also be measured using either Hamming, Manhattan or Minkowski distance.

For instance, if k is equal to two, the algorithm would classify the new data entry to be in the same cluster as that of the two nearest neighbours or data points.

For the implementation of k-NN on this project, the number of neighbours was set to be seventeen. Although most of the time the value of k is found by trial and error, since most implementations are given an even value of classes, an odd value of k is chosen. Moreover, an odd value is generally chosen to avoid ties when classifying the new data point. If the value set of k is set to be a relatively smaller value, such as five or seven, the result would be heavily influenced by noise and would also result in a computationally expensive implementation [6].

Albeit not being particularly common, a k-NN can be brought down into two variations, an eager learner or a lazy learner. An eagerly learning k-NN will construct a generalized model before performing a prediction on new given data entries [6]. On the other hand, a lazy learning k-NN waits until the last possible moment before classifying any data point which implies that there is no requirement for learning or training of the model as all of the data points are used upon runtime [6].

3.2.1 - Distance Weighted k-NN

The implementation of the k-NN consists of the parameter which sets the weight to be “distance”. When weighting on the nearest neighbours’ algorithm is used, the algorithm becomes a global instance. This is because all of the training set is used. It is achieved according to their distance, while setting a greater weight to the closer neighbours. Although the algorithm runs slower with the weighting, it was observed that the supervised learning algorithm yields better accuracy when this weighting is implemented. [7]

3.3 - Decision Trees

Decision trees are used to classify data by posing a number of questions related to the features within the items needed to be classified. They are built through the analysis of a training set with examples where the label/classification is already known.

Each question used to classify the data is stored within nodes and, unless it is a leaf node, each internal node has two child nodes; “yes” child and “no” child. Each of these nodes form the decision tree itself. The data which needs to be classified is filtered down each node of the tree, answering the questions as they go, until they finally reach a leaf node. That data is then given the class of the leaf node it has finished in [8].

The leaf nodes within the decision tree can either be identified by a class name or in the below structure:

$$\begin{array}{ll} C_1: & D_1 \\ C_2: & D_2 \\ .. & .. \\ C_n: & D_n \end{array}$$

In the above structure, the C_i's are the logical conditions and the D_i's are the decision trees. Each logical condition has only attribute being $A < T \ or \ A > T$ for an attribute A at a threshold T, or $A = V \ or \ A \ in \ \{V_i\}$ for an attribute A as a value V or as a value in the subset V_i [9].

3.4 - Logistic Regression

Being a very useful algorithm, Logistic Regression can be used to categorise a large amount of data into a number of different labelled classes. Despite its misleading name, Logistic Regression is actually a type of classification method rather than a regression model.

A model like Linear Regression would not be valid for this task since it is not able to classify the given data between different classes. This is why Logistic Regression was considered and used. The primary difference between the two is that in Linear Regression, the model will try to fit the given data through a straight line using a method called "least squares", whereas in Logistic Regression the model will fit the data in an S-shaped graph, which has a range between 0 and 1, both included. The equation of such a graph would be as below:

$$P_{LR}(y = \pm 1 | x, w) = \frac{1}{\left(1 + e^{(-yw^T x)}\right)}$$

The above equation represents the final form that the S-shaped graph would take, a form very similar to the Sigmoid function $\left(S(x) = \frac{1}{1+e^{(-x)}}\right)$ with some additional parameters. In this case, the equation does not refer to the output of the function but rather to the class of the given data point, seeing that Logistic Regression works with supervised data. Another important aspect of this equation would be the different shape; without it, every logistic regression model would have the same shape. The parameter is the actual variable that best fits the model with the training dataset. It is found through the use of the "maximum likelihood" technique, which is calculated by using the below equation:

$$\prod_{i=1}^{2} \Pr(y_i | x_i w) = \prod_{i=1}^{2} \frac{1}{1 + e^{-y_i w^T x_i}}$$

The likelihood is usually determined with the help of the log of odds, where it converts the given S-shaped curve into a straight line going from $-\infty$, where all the data of the bottom class are found, to $+\infty$, where all the data of the top class is found. The log of odds can be found with the equation $\log \frac{p}{1-p}$ in which the probability p is the given y-value of the S-shaped curve, which will in turn be the new y-value of the log of odds graph. The x-values would remain the same so that, ultimately, the graph is a straight line, similar to what would result from Linear Regression. Using this line, the y-intercept and the gradient can be found. Moreover, the maximum likelihood is then found through the testing and comparison of a number of different intercepts and gradients.

The given data would also be labelled between two different categories. Where $f(x) = 1$ represents all the data in the first category and $f(x) = 0$ represents the remaining data of the other class. Since the output of this S-shaped curve is also going to be linear, but in the range of 0 and 1, how would the classification be calculated? This ultimately comes down to the output of the graph being a probability of the input of the model representing specific classes, rather than being a real number.

3.5 - Testing

After doing some research about how each of the chosen supervised machine learning algorithms work, we were ready to start testing them to see which of them would have the best results and possibly consider the reason behind these good outcomes. However, before being able to start training these models, the training data had to be gathered and processed to try and replicate them, as done by Anguita et al. [1], who had over 500 different features from just accelerometer and gyroscope readings.

As already stated, the application AndroSensor was used to obtain the readings of the sensors of our Android smartphones. The Accelerometer, Gravity, Linear Acceleration, and Gyroscope sensors were used and each sensor split into three values representing the X, Y and Z axes. As a rule, we believe that the more sensors used, the more accurate the phone readings were, allowing us to keep our phones in our side pockets.

The settings of the AndroSensor application were set to output a row of data every 0.5 seconds, permitting us to average the readings every 5 consecutive rows, giving us results every 2.5 seconds, as done in the papers. For each of these 5 rows, statistical methods were performed to withdraw more information. Therefore, rather than just having a static reading at a particular time, one would have statistical values experienced during the 2.5 second time window. These statistical functions include average, median absolute deviation, standard deviation, maximum value, minimum value, signal magnitude area, energy (which is the sum of squares divided by the total number of values), interquartile range, entropy and correlation.
All the statistical functions mentioned will be performed on all three separate axes, except for the signal magnitude area and the correlation. This is because the signal magnitude area will be representing the normalized integral of the original values from the 3 axes at once so only one signal magnitude area will be calculated for each 3 axes. With regards to correlation, some connection between two different types of data is compulsory, therefore the correlation of each pair between the axes X, Y and Z will be calculated. Then, after the addition of these features, another column was added at the end of the row adding the label. Finally, the processed data is split into two separate datasets, where one is used to train the model and the other is used to check whether the dataset was properly trained or not.

Before initiating the training of the different models chosen, we visualised the processed data in order to see whether clustering was possible between them. Since there are different features representing each row in the processed datasets, it is difficult to simply plot the data. Therefore, t-Distributed Stochastic Neighbouring Entities were used in to be able to deduct the high-dimensional data set and subsequently plot it. This allows us to see whether the data is clustered in any way before trying to train the models.

Finally, the models were trained with the gathered and processed data as well as with the given labels. The python packages for support vector machines, decision trees, k-nearest neighbours and logistic regression from the scikit-learn python package were used when training the models. For each model, a Grid Search was used to determine the best parameters to use for these models. After this, the accuracy of each model and a confusion matrix of the predictions were displayed.

4 - Evaluation and Critical Analysis

The climax of this project boils down to the successful development of the prototype. As previously mentioned, two separate datasets were tested for this experiment. The first of which included the usual six classes that previous research papers included in their work, which were Standing, Sitting, Laying, Walking Upstairs and Walking Downstairs. Before the models' predictions, we were able to observe the data points through the visualisation done by the t-SNE performed on the dataset.

From the clustering seen in Figure 1, a conclusion was easily drawn that all the static activities performed in the training for these models have a distinct space and are easily clustered. This is expected due to the position of the phone being static and in unique positions during the activities, resulting in different readings. An example of this would be the difference in phone orientation being vertical when standing versus horizontal when laying. The main differences between these three static actions originate in the gyroscope readings of the phone.

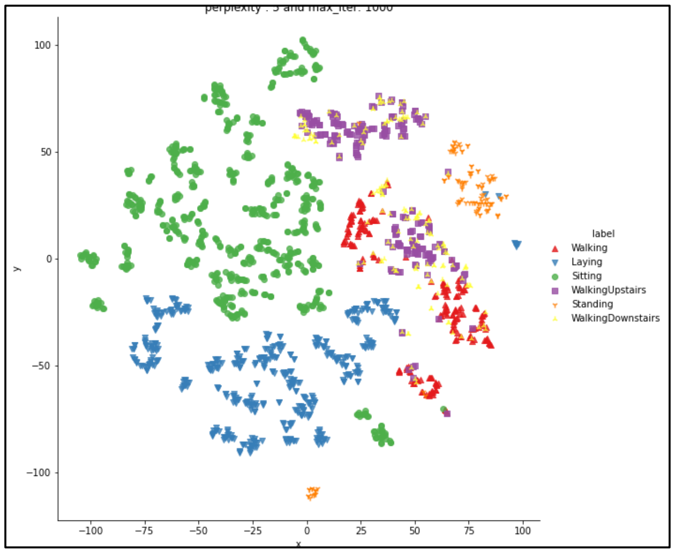

Figure 1 - t-SNE visualisation of the original dataset

On the other hand, there seems to be some difficulty when attempting to cluster the dynamic activities. When walking, albeit close to the others, a clear cluster can be drawn making it possible to distinguish itself from the rest of the visualisation. However, when walking up or down the stairs, the model found difficulty distinguishing between the two activities, mainly due to both actions require a very similar motion. Unfortunately, our assumption that the difference in gravity readings from the phone's sensor would be enough to distinguish between the two was incorrect and insufficient.

Finally, the models were all trained and tested. The best model resulted to be the Logistic Regression with an accuracy of 96%, with L2 regularisation and multinomial multi class option, the confusion matrix of which can be found in figure 2. The k-Nearest Neighbours got an accuracy of 91% with a k value of 7, using the ball tree algorithm. The Radial Basis function got an accuracy of 95.5% with C value = 90 and gamma set to scale. Decision Trees got an accuracy of 92.4% accuracy with a max depth of 7. The Polynomial Kernel resulted in an accuracy of 95.4% with parameters C = 16 and a degree of 1.

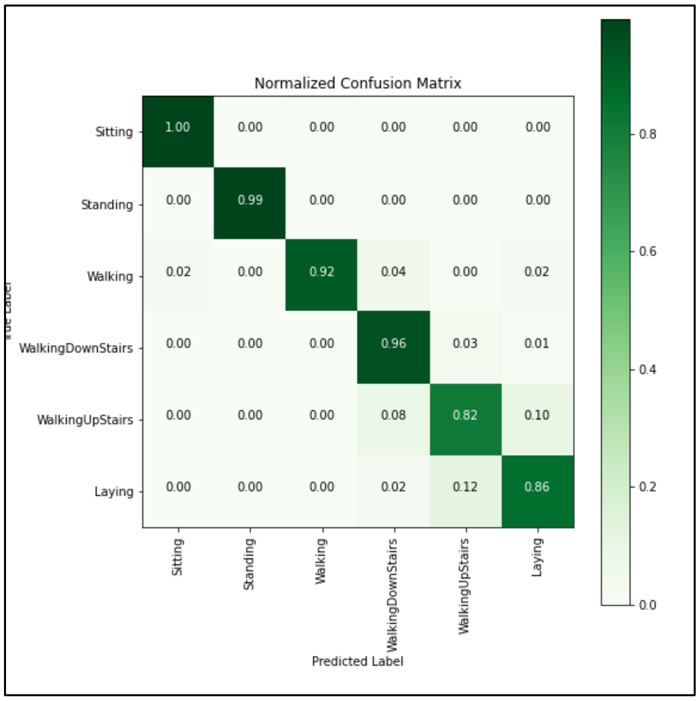

Figure 2 - Logistic Regression Confusion Matrix of the original dataset

It seems that for all the models, the only class that had difficulty in predicting was the walking upstairs classifier. However, an unforeseen misclassification of data was the walking downstairs activity getting interpreted as laying down instead. Other than that, the rest of the classifiers were easily predicted as they are quite distinguishable between each other.

4.1 - Dataset 2: Six New Activities

This dataset includes six unique activities different than those of the activities used in the research papers mentioned. These activities include Cycling, Swimming, Football, Jump Rope, Push Ups and Jogging. Additionally, a Tennis dataset was also generated but was left out from this selection due to a very poor classification following experimentation.

Considering that all these activities entail a large emphasis on movement, it would be expected that the t-SNE would find it difficult to cluster very distinctly. This can be seen in Figure 3 where although most of the Cycling data entries are segregated, a lot of outliers can be found near the Swimming cluster. Similarly, the Football data and Jump Rope data is clustered to be very similar to each other. This might have happened due to very similar movements each of these activities entail, primarily the general jogging during football and an up-down movement during push-ups and jump-rope.

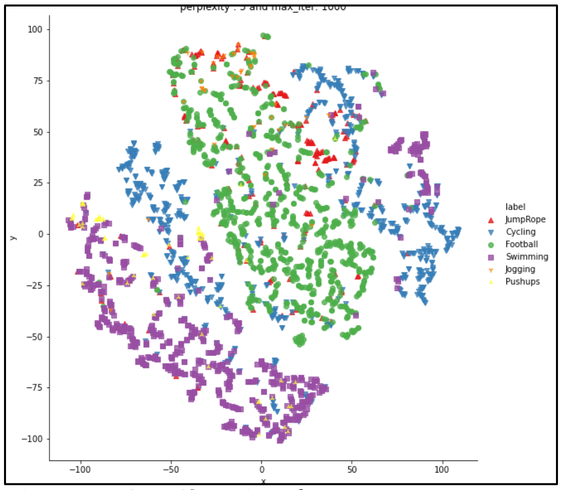

Figure 3 - t-SNE Clustering of New Dataset

The data was collected via the AndroSensor application mentioned earlier using the inertia sensors Gravity, Linear Acceleration, Gyroscope and Accelerometer.

Finally, the activities were all trained and tested. The best classification algorithm was discovered to be the Support Vector Machine using the Radial Basis Function kernel with an accuracy of 87.3% [Fig 4] with the Logistic Regression being a close second with 87.2% [Fig 5] with L2 regularisation and One vs Rest multi class option. The other results of classification for this dataset are unfortunately not as promising as the other dataset created where the k-NN algorithm returned an 83% accuracy with a k value of 7 alongside the ball tree algorithm. Additionally to the SVM algorithm, the Polynomial kernel returned an 84% accuracy with the same parameters of the RBF being a c value of 90 and a scale gamma variable. Finally, the Decision Trees algorithm classified with 83.5% accuracy considering a maximum depth of 7.

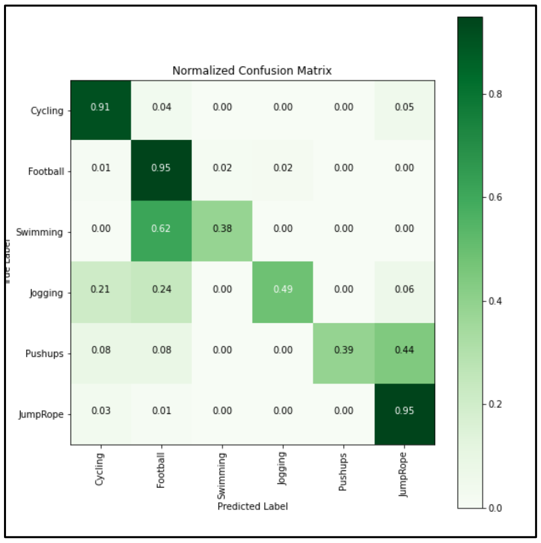

Figure 4 - RBF Confusion Matrix of New Dataset

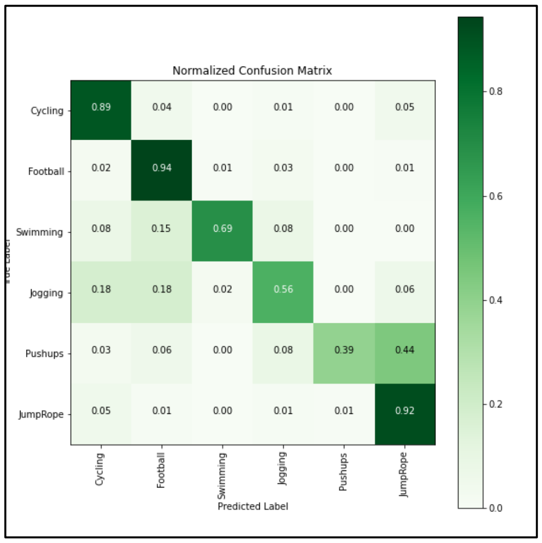

Figure 5 - Logistic Regression Confusion Matrix of New Dataset

5 – Conclusions

Our goal for this assigned practical task was to create an application which would be able to recognise human activity through sensors found only in a smartphone as well as collecting a new dataset to compliment the existing HAR dataset. After numerous tests and experiments, we successfully executed our goals.

The prediction of human activity was generated through four different models: Support Vector Machines (Radial Kernel and Polynomial Kernel), K-Nearest Neighbours and Decision Trees. All four of the models used generated predications with an accuracy of over 83%, evening achieving accuracies as high as 96% with the existing dataset and 87.3% with our new dataset.

The second part of our goal, to add new data and new activities to the existing data, was accomplished through the use of the application "AndroSensor". Overall, we collected data for 7 new activities, choosing the implement only 6 in our model. This data was gathered through the accelerometer and gyroscope sensors found in the average smartphone, which was carried in the users pocket.

A possible future improvement we can foresee to our assigned task is the creation of a mobile application which would be able to recognize the activities in real time. Another possible improvement would be a system which would use all the prediction models to generate one, accurate, prediction for an activity.

6 - References

[1] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," *European Symposium on Artificial Neural Networks*, vol. 3, Apr. 2013, [Online]. Available: https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2013-84.pdf.

[2] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Training Computationally Efficient Smartphone–Based Human Activity Recognition Models," in *International Conference on Artificial Neural Networks*, Sep. 2013, pp. 426–433, [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-40728-4_54.

[3] E. Bulbul, A. Cetin, and I. A. Dogru, "Human Activity Recognition Using Smartphones," *2018 2nd international symposium on multidisciplinary studies and innovative technologies (ismsit)*, pp. 1–6, Oct. 2018, [Online]. Available: https://www.researchgate.net/profile/Erhan-Bulbul-2/publication/329559026_Human_Activity_Recognition_Using_Smartphones/links/5c80c87592851c69505c8ff5/Human-Activity-Recognition-Using-Smartphones.pdf.

[4] Y. Goldberg and M. Elhadad, "splitSVM: Fast, Space-Efficient, non-Heuristic, Polynomial Kernel Computation for NLP Applications," *Proceedings of ACL-08: HLT, Short Papers*, pp. 237–240, Jun. 2008.

[5] C.-J. Lin, "Machine learning software: design and practical use," *Machine Learning Summer School*, 2012. https://www.csie.ntu.edu.tw/~cjlin/talks/mlss_kyoto.pdf.

[6] Avinash Navlani, "KNN Classification using Scikit-learn," *DataCamp Community*, 2018.
https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

[7]"Distance weighted k-NN algorithm," *Data-machine.net*, 2021. http://www.data-machine.net/nmtutorial/distanceweightedknnalgorithm.htm

[8] C. Kingsford and S. L. Salzberg, "What are decision trees?," *Nature Biotechnology*, vol. 26, no. 9, pp. 1011–1013, Sep. 2008, doi: 10.1038/nbt0908-1011.

[9] J. R. Quinlan, "Simplifying Decision Trees," *International journal of man-machine studies*, vol. 27, no. 3, pp. 221–234, Sep. 1987, [Online]. Available: https://dspace.mit.edu/bitstream/handle/1721.1/6453/aim-930.pdf?sequence=2.

[10] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Energy Efficient Smartphone-Based Activity Recognition Using Fixed-Point Arithmetic," *Journal of Universal*

Computer Science, vol. 19, no. 9, pp. 1295–1314, May 2013, [Online]. Available: https://upcommons.upc.edu/bitstream/handle/2117/20437/jucs_19_09_1295_1314_anguita.pdf;sequence=1.

[11] S. K. Lodha and R. Franke, "Scattered Data Interpolation: Radial Basis and Other Methods," Handbook of Computer Aided Geometric Design, pp. 389–404, 2002, doi: 10.1016/b978-044451104-1/50017-4.

[12] M. Kumar and S. K. Rath, "Feature Selection and Classification of Microarray Data Using Machine Learning Techniques," *Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology*, pp. 213–242, 2016, doi: 10.1016/b978-0-12-804203-8.00015-8.

[13] A. Subasi, "Machine learning techniques," *Practical Machine Learning for Data Analysis Using Python*, pp. 91–202, 2020, doi: 10.1016/b978-0-12-821379-7.00003-5.

[14] T. W. Edgar and D. O. Manz, "Exploratory Study," *Research Methods for Cyber Security*, pp. 95–130, 2017, doi: 10.1016/b978-0-12-805349-2.00004-2.

[15] UCI Machine Learning, "Human Activity Recognition with Smartphones," *Kaggle.com*, 2012. https://www.kaggle.com/uciml/human-activity-recognition-with-smartphones?select=train.csv

6.1 - Table of Figures

Figure 1: t-SNE visualisation of the original dataset 14

Figure 2: Logistic Regression Confusion Matrix of the original dataset 15

Figure 3: t-SNE Clustering of New Dataset 16

Figure 4: RBF Confusion Matrix of New Dataset 16

Figure 5: Logistic Regression Confusion Matrix of New Dataset 16