CHuman Activity and Attribute Recognition from iPhone Motion Data

CSC8635 – Machine Learning Assignment Report

Issy Middleton C1000051

Introduction

Ofcom's figures indicate that in 2020 45 percent of households in the UK have two mobile phones and published figures indicate there were over 53 million smart phones alone in the UK in May 2021 and over 2,700 million globally (O'Dea, 2021). It is therefore important to understand how the data they are capable of gathering about our physical activities could be used. The question I am interested in answering is - Can activity data recorded on a smartphone be used to infer additional details about us, and if so, to what degree of accuracy?

Related Work

Researchers from University of California, Irving undertook experiments with a group of 30 volunteers within an age bracket of 19-48 years. The dataset is publicly available and has been used a data source for many research papers (Reyes-Ortiz, 2012). Each person performed six activities wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, linear acceleration and angular velocity was captured at a constant rate of 50Hz.

Support vector classifier and convolutional neural network machine learning models achieved almost 100 percent accuracy on test data (Sun, 2019). Analysis established that within a short amount of time (1-1.5 min) the smartphone has enough data to determine what its user is doing (95%: 6 activities) or who the user is (Walking 94%: 30 participants) and even the basics of a person's specific walking style (Liftoff, 2019).

Broader context

Cardiac abnormalities are one of the leading causes of deaths all over the globe. Smartphone-based systems may offer opportunities for real-time cardiac monitoring and early abnormality detection (Shabaan et. al, 2020). It is important to test the accuracy of predictive models utilising smartphone data.

A paper on wearable devices identified 423 unique devices from 132 different brands. With recent advances in mobile sensor technology, privately collected physical activity data can be used as an addition to existing methods for health data collection in research (Henriksen et. al, 2018). Given the popularity of wearable mobile sensors, it is important to determine whether smartphone activity data could provide additional personal attribute data about the user.

The analysis showing that users can be personally identified by smartphone motion data. This means that sensor data is biographically significant legally, under the General Data Protection Regulations (Liftoff, 2019). Therefore data privacy concerns need to be addressed when data 'relates to or is 'obviously about' a particular individual. It is important that smartphone users understand the data that can be captured. This will allow informed choices to be made and privacy trade-offs to be transparent. Users may find monitoring useful for improving the quality health interventions.

Hypothesis

The hypothesis being assessed is whether personal characteristics of weight, height, age and gender can be predicted from motion sensor captured by a smartphone. The analysis will compare model accuracy.

Experiment

An experiment at Queen Mary University of London generated the dataset being reviewed. Time-series data was generated by accelerometer and gyroscope sensors. It was collected from an iPhone 6s kept in the participant's front pocket using SensingKit which collects information from Core Motion framework on iOS devices. All data collected in 50Hz sample rate. A total of 24 participants in a range of gender, age,

weight, and height performed 6 activities in 15 trials, of varying length, in the same environment and conditions: downstairs, upstairs, walking, jogging, sitting, and standing. Full study conditions are provided by the researchers to allow repetition (Malekzadeh, 2019).

Summary of analysis

There were 24 data subjects in the trial, 10 were females and 14 males, with an average age of 29. Females had a lower average age, height and weight than males. The correlation matrix and scatter plot indicated a correlation between height and weight, also between gender and height. The scatterplot and histograms show distribution by age, weight, height and gender.

The raw data from the 15 activities conducted by the 24 participants was loaded and grouped into the 6 activities. Identifiers for the experiment number, participant number and activity type were added.

The multivariate time-series has 12 features: attitude.roll, attitude.pitch, attitude.yaw, gravity.x, gravity.y, gravity.z, rotationRate.x, rotationRate.y, rotationRate.z, userAcceleration.x, userAcceleration.y, userAcceleration.z.

The size of the data set by activity type differed due to the varying length of the trails. The datasets for the downstairs and jog were the smallest data sets, upstairs and standing next in order, with sitting and walking datasets the largest.

Based on hierarchical clustering, downstairs, upstairs and walk are similar to each other, sitting and standing are also similar, with jogging distinctively different from the other five.

Detecting activity

Variable importance

Detecting subject

Detecting gender

Detecting above average weight

Detecting above average height

Detecting above average age

Supervised learning

mapping data and labels, response variable, use one or more explanatory variables to build models to predict some response, understand how changes in the values of explanatory variables affect the values of a response variable.

Logistic regression - measure the accuracy of a regression model is by calculating the root mean square error (RMSE), a metric that tells us how far apart our predicted values are from our observed values in a model, on average

classification model - response variable is categorical, e.g. on or off. The most common way to measure the accuracy of a classification model is by simply calculating the percentage of correct classifications the model makes.

Unsupervised learning

(no labels), feature compression, dimension reduction, clustering

If there exists a hyperplane that perfectly separates the two classes, then we call the two classes linearly separable.

Clustering
Hard Clustering: Every point belongs to exactly a certain cluster
Soft Clustering: Every point belongs to several clusters with certain degrees

	K-Means	EM
Objective Function	Minimise sum of squared Euclidean distance	Maximise log-likelihood
Type	Distance-based	Density-based
Parameter Updating	Repeat until convergence 1. Assign points to clusters (**hard assignment**) 2. Updating parameters of clusters	Repeat until convergence 1. Compute responsibility (posterior probability of membership--- **soft assignment**) 2. Updating parameters of clusters

48

Regression is an algorithm in supervised machine learning that can be trained to predict real number outputs. Classification is an algorithm in supervised machine learning that is trained to identify categories and predict in which category they fall for new values.

Few examples of linear classifiers are Logistic Regression, Perceptron, Naive Bayes, Support Vector Machines, etc.

Examples of non-linear classifiers are Decision Trees, K-Nearest Neighbour, Random Forest, non-linear-kernel SVM etc.

Classifiers

KNN – K-Nearest Neighbour, Euclidean most popular distance, better for small data
Logistic Regression – linear classifier, linearly separable data
Linear Support Vector Machine – SVM uses margin
RBF SVM
Decision Tree
Random Forest
Neutral Net (one layer)
Naïve Bayes
Multi Layer Perceptron (more than one layer) – non-linear separable data, difficult to see relationship between in put and output, doesn't scale well

CNN – good for high dimensional data (image), parameter numbers reduced, also good for timeseries

Recurrent Neutral Network, LTSM same with with hidden unit
ConvLSTM – training time longer

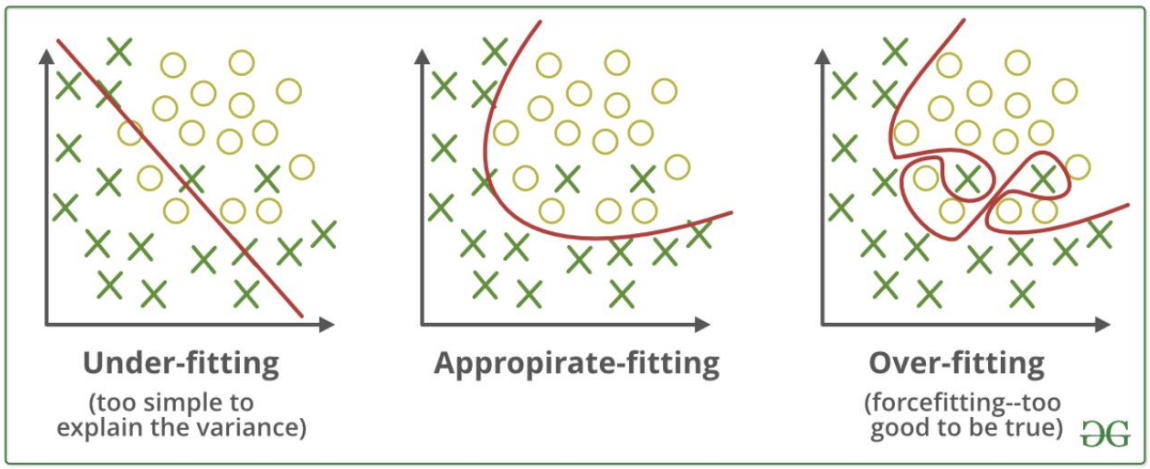
Dimension Reduction
feature de-correlation, preserving features with the largest variance, maximising variance
Principle Component Analysis (PCA) – eigenvalues/eigenvectors, transformation matrix to find value
Linear Discriminant Analysis (LDA) – needs label, minimising intra-class distances while maximising inter-class distances, maximising the ratio of between-class scatter matrix to within-class scatter matrix

Autoencoder – unsupervised because no label needed

Clustering
K-means (hard)
EM (soft)
Example of overfitting and know how to reduce this problem by increasing representative training samples, reducing model complexity by model selection, regularisation– penalty against complexity or ensemble such as decision tree to random forest.

Overfitting occurs when your model learns too much from training data and isn't able to generalize the underlying information. When this happens, the model is able to describe training data very accurately but loses precision on every dataset it has not been trained on. The lower the number of the parameters, the higher the simplicity and, reasonably, the lower the risk of overfitting.

Avoid by cross-validation: An example of overfitting and how to avoid it | by Gianluca Malato | Towards Data Science



Model Fitting Scenarios

Regularisation – lambda is the penalty parameter against complexity, reduces the effect of overfitting, to generalise unseen data.

As lambda grows larger the less the wrongly classified examples are allowed (or the highest the price the pay in the loss function). Then when lambda tends to infinite the solution tends to the hard-margin (allow no miss-classification). When lambda tends to 0 (without being 0) the more the miss-classifications are allowed.

[Regularization in Machine Learning and Deep Learning | by Amod Kolwalkar | Analytics Vidhya | Medium]

Scaling - know at least two feature scaling techniques, and the motivation, i.e. comparing multiple features in the same range

Feature scaling techniques	Scaled data
Rescaling /min-max normalisation	$\hat{\mathbf{x}} = \frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}$
Mean normalisation	$\hat{\mathbf{x}} = \frac{\mathbf{x} - \text{mean}(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}$
Standardisation	$\hat{\mathbf{x}} = \frac{\mathbf{x} - \text{mean}(\mathbf{x})}{\text{std}(\mathbf{x})}$

Demonstrate multi-class LR (softmax), know how many parameters there are. LR is shallow learning, Multi-Layer Perceptron is deep.

	LR	MLP
type	Linear classifier	Non-linear classifier
Interpretability	High	Low
#parameters	Relatively low	Relatively high
training	Simple gradient calculation	Backpropagation

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

If a multilayer perceptron has a linear activation function in all neurons, that is, a linear function that maps the weighted inputs to the output of each neuron, then linear algebra shows that any number of layers can be reduced to a two-layer input-output model.
[Multi-Layer Perceptron (MLP). What is MLP? | by Z² Little | Medium]

Know the motivation of using CNN for high-dimensional input data (e.g., images), role of pooling

MLP is now deemed insufficient for modern advanced computer vision tasks. It has the characteristic of fully connected layers, where each perceptron is connected with every other perceptron. The disadvantage is that the number of total parameters can grow to very high (number of perceptron in layer 1 multiplied by # of p in layer 2 multiplied by # of p in layer 3…). This is inefficient because

there is redundancy in such high dimensions. Another disadvantage is that it disregards spatial information. It takes flattened vectors as inputs.

Convolutional Neural Network (CNN)- Layers are sparsely connected rather than fully connected. It takes matrices as well as vectors as inputs. The layers are sparsely connected or partially connected rather than fully connected. Every node does not connect to every other node.

Multilayer Perceptron model vs CNN | by Saumyadeepta Sen | The Owl | Medium

A common CNN model architecture is to have a number of convolution and pooling layers stacked one after the other.

Why to use Pooling Layers?

- Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network.
- The pooling layer summarises the features present in a region of the feature map generated by a convolution layer. So, further operations are performed on summarised features instead of precisely positioned features generated by the convolution layer. This makes the model more robust to variations in the position of the features in the input image.

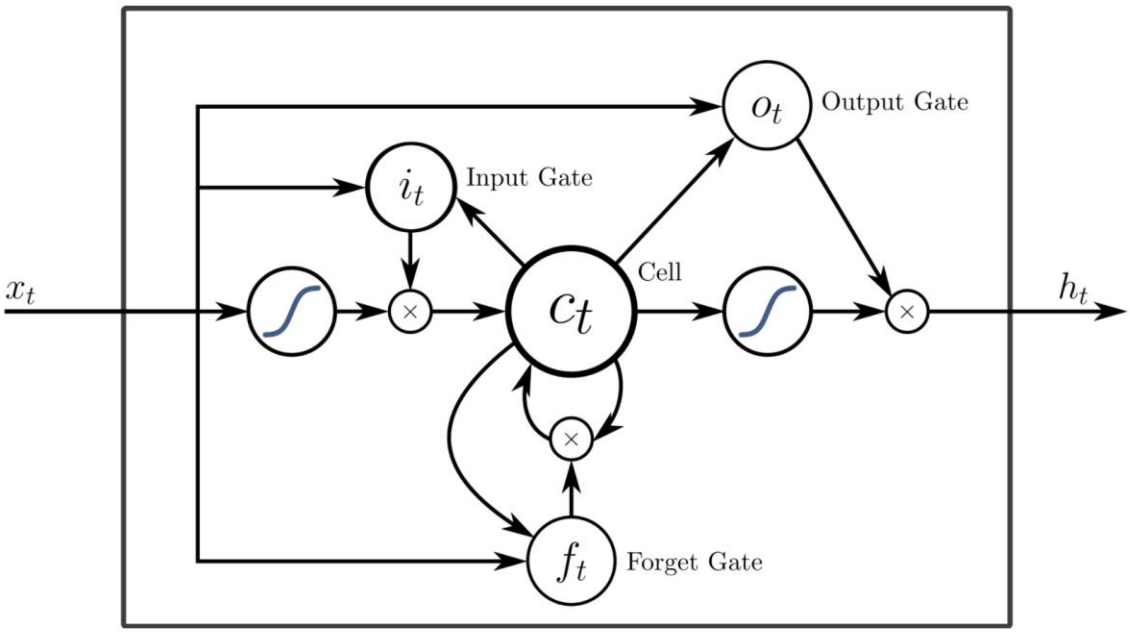
CNN | Introduction to Pooling Layer - GeeksforGeeks

You should be able to describe how to extend MLP to RNN

You should be able to calculate the number of parameters for RNN

Long short-term memory neural network (LSTM) is a type of recurrent neural network (RNN). Similar to feedforward MLP networks, RNNs have two stages, a forward and a backward stage. Each works together during the training of the network.

An LSTM unit is a recurrent unit, that is, a unit (or neuron) that contains cyclic connections, so an LSTM neural network is a recurrent neural network (RNN). The main difference between an LSTM unit and a standard RNN unit is that the LSTM unit is more sophisticated. More precisely, it is composed of the so-called gates that supposedly regulate better the flow of information through the unit - 1) Forget gate, 2) input gate, 3) output gate, 4) memory cell.



neural networks - What's the difference between LSTM and RNN? - Artificial Intelligence Stack Exchange

What is the difference between CNN and ConvLSTM for Human Activity Recognition modelling?

ConvLSTM replaces matrix multiplication with convolution operation at each gate in the LSTM cell. By doing so, it captures underlying spatial features by convolution operations in multiple-dimensional data. ConvLSTM is designed for 3-D data as its input.

ConvLSTM is when you have the matrix multiplication calculation of the input with the LSTM cell replaced by the convolution operation. In contrast, CNN-LSTM are two different modules which are combined together. The CNN is a regular CNN which acts as a spatial feature extractor. The output of the CNN is multiplied by the LSTM cell to learn the temporal features.

Simply said, ConvLSTM and CNN LSTM behave the same functionally where ConvLSTM have the convolution embedded in the architecture while CNN-LSTM have just concatenates the types of networks together externally.
What is the difference between ConvLSTM and CNN LSTM? - Quora

Is AE supervised or unsupervised model?

Semi-supervised learning falls between supervised and unsupervised learning where large amount of unlabeled data along with small amount of labeled data is available. Various conventional machine learning techniques can be used to define and solve this problem as a supervised learning problem. The complexity of the problem may also depend upon the number of classes which may be present in the unlabeled data. Autoencoders can be used to solve such problems. An autoencoder neural network is an unsupervised learning algorithm that applies back propagation, setting the target values to the inputs.
Autoencoders for Semi-Supervised Learning | by Romit Singhai | Medium

Able to find out the best combination of hyper-parameters for RBF SVM

Understand how to construct decision trees for a random forest

Differences between decision tree and random forest

-	Decision tree	Random Forest
Overfitting	Easily	Generalising well
Interpretability	High	Moderate
High-dimensionality data	Poor	Good
Decision Boundary	Block effect	Smooth

Do we need to train K-Nearest Neighbour?, what is the limitation of KNN given big training data

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.

- Step-1: Select the number K of the neighbours
- Step-2: Calculate the Euclidean distance of K number of neighbours
- Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
- Step-4: Among these k neighbours, count the number of the data points in each category.
- Step-5: Assign the new data points to that category for which the number of the neighbour is maximum.
- Step-6: Our model is ready.

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

[K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint]

Probability Theory

- The rule of probability
  - Sum rule $$p(X) = \sum_Y p(X, Y)$$
  - Product rule $$p(X, Y) = p(Y|X)p(X) = p(X|Y)p(Y)$$
- Bayesian theorem: from the product rule $$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

  where $$p(X) = \sum_Y p(X, Y) = \sum_Y p(X|Y)p(Y)$$

Key assumption of Naïve Bayes Classifier is that there is conditional independence of the features/attributes/variables.

What are the two parameters of Gaussian distribution.

The normal distribution N(μ, σ) has two parameters associated with it: 1 The mean μ 2 The standard deviation σ.

You should know the motivation of using Gaussian mixture models (GMM)

Unsupervised data can consist of multiple sets of data points that can follow the Gaussian distribution which directly means in the data there can be multiple peaks presented in any data and extracting such data from a huge data set can be done by the Gaussian mixture model.

For example in unsupervised data there are three sets of data points that follow the Gaussian distribution which means we can build three bell curves on, there will be three mean or peak points. In this type of data set, the gaussian mixture model defines the probability for data points to belong to any of the distributions.

GMM is a clustering method using a probability distribution. K-means clustering is also a clustering method but uses euclidean distance to calculate the difference between data points as closer data can be segregated in one cluster, this is a big difference between K-mean and GMM.

The probability distribution function of GMM can be defined as

$$N(\mu, \sum) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\sum|}} exp(-\frac{1}{2}(x-\mu)^T \sum^{-1}(x-\mu))$$

Where μ= Mean
∑= Covariance Matrix of the Gaussian
d= The numbers of features in our dataset
x=the number of datapoints

In the above expression, we are using the covariance matrix which holds the magnitude of changes in a variable of the data set in the effect of another variable. Basically, it shows the relationship between two variables. Using a covariance matrix in place of the standard deviation gives more accurate results.

All You Need to Know About Gaussian Mixture Models (analyticsindiamag.com)

model parameters of multi-dimensional GMM

Gaussian mixture models are extensively utilized in mining data, recognition of patterns, machine learning, and statistical analysis. In several applications, their parameters are detected using maximal likelihood and EM algorithm and are modeled as latent variables.

EM is a method for estimating the maximum likelihood and is employed to evaluate closed-form expressions to update the model parameters. EM is an iterative technique that possess suitable property in which the maximal likelihood of the data maximization is certified to come near a local maximum.

EM for mixture models contains two steps: the first step is termed as expectation step or E-step which contains computation of the expectation of the component assignments for each data point provided the model parameters. The second step is termed as maximization step or M-step which contains maximization of expectations computed in E-step based on model parameters. Each step contains the updation of parameter values. The complete iterative process continues until the algorithm converges providing a maximal likelihood estimate. Instinctively, the algorithm works as knowing the component assigning for each data point and makes parameter solving in an easier manner. The expectation step is based on the latter case, while the maximization step is linked to the former case. Thus, by considering consecutive values, the nonfixed values are computed in an effective manner.

Gaussian Mixture Model - an overview | ScienceDirect Topics

You should know the parameters to be estimated for GMM (Chicken and Egg picture, need to have a sensible place to start).

Results & Conclusion

The sample was small with less than 30. If the sample size is too small it may be difficult to detect what was intended (Kar & Ramalingam, 2013). The average age of the participants was 29. The information gathered may not be reflective of the population, i.e. the speed of movements within younger individuals may not be reflective of an older population. Therefore any predictions on age given the small sample may not be an accurate prediction of test error.

Evaluate success

Future implications

Reflections
personal and professional development,

process, technologies and methodologies
Scientific Method

cookiecutter Data Science template in GitHub, with Git for version control, Python, Google Colab.
Present findings

Detailed findings of the exploratory analysis

There were 24 data subjects in the trial, with average weight of 72kg (11 stones 4lb), average age of 29 and average height of 174cm (5ft 8.5'), 10 were females and 14 males. Females had a lower average age of 27 compared to the men in the study whose average age was almost 30. Female average weight was 65kg (10 stones 3lb), compared with male average at 77kg (12 stones 2lb). Average height also varies by gender, 166cm (5ft 5') for the women and 179cm (5ft 10') for the men. There was a correlation between height and weight.

Histogram
Correlation Matrix
Scatterplot

The size of the data set by activity type differed due to the varying length of the trails. The datasets for the downstairs and jog were the smallest data sets, upstairs and standing next in order, with sitting and walking datasets the largest.

Using the methods provided by library "dtaidistance.", 400 datapoints (userAcceleration.z) of each activity type are segmented and similarity between each segment is calculated by dynamic time wrapping. Based on hierachical clustering, downstairs, upstairs and walk are similar to each other, sitting and standing are also similar, with jogging distinctively different from the other five.

	Linear Regression	Logistic Regression
Response Variable	Continuous (e.g. price, age, height, distance)	Categorical (yes/no, male/female, win/not win)
Equation Used	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots$	$p(Y) = e^{(\beta 0 + \beta 1 X 1 + \beta 2 X 2 + \ldots)} / (1 + e^{(\beta 0 + \beta 1 X 1 + \beta 2 X 2 + \ldots)})$
Method Used to Fit Equation	Ordinary Least Squares	Maximum Likelihood Estimation
Output to Predict	Continuous value ($150, 40 years, 10 feet, etc.)	Probability (0.741, 0.122, 0.345, etc.)

Gender and average weight prediction using this dataset (Huang, 2019).
Exploratory analysis of data subjects (Shaar, 2019).

References:

A. Henriksen, M.H. Mikalsen, A.Z. Woldaregay, M. Muzny, G. Hartvigsen, L.A. Hopstock, S. Grimsgaard, 2018. *Using Fitness Trackers and Smartwatches to Measure Physical Activity in Research: Analysis of Consumer Wrist-Worn Wearables (Journal of Medical Internet Research)* [ONLINE] Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5887043/ [Accessed 16 December 2021].

Y.C. Huang, 2019. *My analysis on motion sensor data (Kaggle)* [ONLINE] Available at: https://www.kaggle.com/teaprint/my-analysis-on-motion-sensor-data [Accessed 16 December 2021].

D. Liftoff, 2019. *What Does Your Smartphone Know About You? (Kaggle)* [ONLINE] Available at: https://www.kaggle.com/morrisb/what-does-your-smartphone-know-about-you [Accessed 16 December 2021].

Kar, S.S. & Ramalingam, A., 2013. Is 30 the magic number? Issues in sample size estimation. *National Journal of Community Medicine,* 4(1), p. 175.

M. Malekzadeh, 2019. *MotionSense Dataset: Smartphone Sensor Data (Kaggle)* [ONLINE] Available at: https://www.kaggle.com/malekzadeh/motionsense-dataset [Accessed 16 December 2021].

S. O'Dea, 2021. *Number of mobile phones per household in the United Kingdom (UK) in 2020 (Statista.com)* [ONLINE] Available at: https://www.statista.com/statistics/387184/number-of-mobile-phones-per-household-in-the-uk/ [Accessed 16 December 2021].

J.L. Reyes-Ortiz, D. Anguita, A. Ghio, L.Oneto, X. Parra, 2012. *Human Activity Recognition Using Smartphones Data Set (UCI Machine Learning Repository)* [ONLINE] Available at: https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones [Accessed 16 December 2021].

S.T.A. Shaar, 2019. *Starter: MotionSense Dataset : 8c09b08d-5 (Kaggle*) [ONLINE] Available at: https://www.kaggle.com/salahuddinemr/starter-motionsense-dataset-8c09b08d-5 [Accessed 16 December 2021].

M. Shabaan, K. Arshad, M. Yaqub, F. Jinchao, M.S. Zia, G.R. Boja, M. Iftikhar, U. Ghani, L.S. Ambati, R.Munir, 2020. *Survey: smartphone-based assessment of cardiovascular diseases using ECG and PPG analysis (BMC Medical Informatics and Decision Making)* [ONLINE] Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7392662/ [Accessed 16 December 2021].

X. Sun, 2019. *Human Activity Recognition Using Smartphones Sensor Data (medium.com)* [ONLINE] Available at: https://medium.com/@xiaoshansun/human-activity-recognition-using-smartphones-sensor-data-fd1af142cc81 [Accessed 16 December 2021].

G.Wang, Q. Li, L. Wang, W. Wang, M. Wu, T. Liu, 2018. *Impact of Sliding Window Length in Indoor Human Motion Modes and Pose Pattern Recognition Based on Smartphone Sensors (Sensors – Basel, Switzerland)* [ONLINE] Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6021910/ [Accessed 16 December 2021].