Dobbins, C, Rawassizadeh, R and Momeni, E

Detecting Physical Activity within Lifelogs towards Preventing Obesity and Aid Ambient Assisted Living

Chelsea Dobbins, Reza Rawassizadeh, and Elaheh Momeni, "Detecting Physical Activity within Lifelogs towards Preventing Obesity and Aiding Ambient Assisted Living," Neurocomputing, Dec. 2016, http://dx.doi.org/10.1016/j.neucom.2016.02.088

Detecting Physical Activity within Lifelogs towards Preventing Obesity and Aiding Ambient Assisted Living

Chelsea Dobbins [1]*, Reza Rawassizadeh [2] and Elaheh Momeni [3]

[1] Department of Computer Science
Liverpool John Moores University
Byrom Street
Liverpool L3 3AF
United Kingdom
C.M.Dobbins@ljmu.ac.uk
*Corresponding author

[2] Department of Computer Science,
Dartmouth College,
Hanover, NH 03755,
USA
rrawassizadeh@acm.org

[3] Faculty of Computer Science
University of Vienna
Universitätsring 1
1010 Vienna
Austria
elaheh.momeni.roochi@univie.ac.at

Chelsea Dobbins, Reza Rawassizadeh, and Elaheh Momeni, "Detecting Physical Activity within Lifelogs towards Preventing Obesity and Aiding Ambient Assisted Living," Neurocomputing, Dec. 2016, http://dx.doi.org/10.1016/j.neucom.2016.02.088

Detecting Physical Activity within Lifelogs towards Preventing Obesity and Aiding Ambient Assisted Living

Abstract. Obesity is a global health issue that affects 2.1 billion people worldwide and has an economic impact of approximately $2 trillion. It is a disease that can make the aging process worse by impairing physical function, which can lead to people becoming more frail and immobile. Nevertheless, it is envisioned that technology can be used to aid in motivating behavioural changes to combat this preventable condition. The ubiquitous presence of wearable and mobile devices has enabled a continual stream of quantifiable data (e.g. physiological signals) to be collected about ourselves. This data can then be used to monitor physical activity to aid in self-reflection and motivation to alter behaviour. However, such information is susceptible to noise interference, which makes processing and extracting knowledge from such data challenging. This paper posits our approach that collects and processes physiological data that has been collected from tri-axial accelerometers and a heart-rate monitor, to detect physical activity. Furthermore, an end-user use case application has also been proposed that integrates these findings into a smartwatch visualisation. This provides a method of visualising the results to the user so that they are able to gain an overview of their activity. The goal of the paper has been to evaluate the performance of supervised machine learning in distinguishing physical activity. This has been achieved by (i) focusing on wearable sensors to collect data and using our methodology to process this raw lifelogging data so that features can be extracted/selected. (ii) Undertaking an evaluation between ten supervised learning classifiers to determine their accuracy in detecting human activity. To demonstrate the effectiveness of our method, this evaluation has been performed across a baseline method and two other methods. (iii) Undertaking an evaluation of the processing time of the approach and the smartwatch battery and network cost analysis between transferring data from the smartwatch to the phone. The results of the classifier evaluations indicate that our approach shows an improvement on existing studies, with accuracies of up to 99% and sensitivities of 100%.

Keywords: Physical Activity Recognition; Signal Processing; Classification; Lifelogging; Assisted Living; Obesity

1 Introduction

There is undisputable evidence that indicates that engaging in regular physical activity is essential for healthy ageing and plays a key role in preventing premature death and several chronic non-communicable diseases (NCDs), including cardiovascular disease, coronary heart disease, stroke, diabetes, cancer, hypertension, obesity, depression and osteoporosis [1–2]. Globally, the number of older people is increasing and by 2050 is expected to double from 841 million to 2 billion [3]. As a result, addressing the costs of such demographic changes is vital for any nation. Being physically inactive is a global issue that affects 1 in 4 adults and is the fourth leading risk factor for mortality around the world [2], [4–5]. This has serious consequences not only for our health but also on the economy. Currently, 2.1 billion people worldwide are either overweight or obese, with the economic impact of this affliction costing approximately $2.0 trillion [6]. The challenge is therefore to ensure that healthy life expectancy (HLE), i.e. the average amount of years that we live without disease/injury, increases at the same rate as life expectancy, and allows people to work for longer [7].

Obesity is a preventable disease and with proper interventions can be tackled. Most importantly, education, behaviour change and personal responsibility are critical elements of any program to reduce the onset of this disease; however, these are complex, difficult, resource-intensive and time-consuming processes for individuals to achieve [6], [8]. Factors, including the availability and affordability of food, changes in diet, psychological triggers (e.g. stress), increasingly sedentary nature of many jobs (e.g. office work), transportation, and increasing urbanization all contribute to the rise of this epidemic [6], [9]. As we age, our mean body weight and body mass index (BMI) increases, and the effects of years of reduced physical activity and a poor diet becomes more apparent and dangerous in later life, which can lead to premature physical deterioration and cognitive decline [10–11]. As a result, research into the effects of physical inactivity is growing rapidly, with initial results indicating that there are important negative health outcomes for various markers of this type of behaviour [12]. Therefore, in order for us to age healthily and without debilitating illnesses, awareness and modifications towards our lifestyle choices are essential.

In today's society, computing devices are now capable of capturing and storing a phenomenal amount of personal information and are increasingly being used to support lifestyle choices. For instance, smartphones and wearable technologies have enabled us to collect a wide range of personalized data that can be used for self-reflection and thus influence behavioural change. A consequence of such technology innovation, improved connectivity and low-cost sensors is the new era of the Internet of Everything (IoE) [13], which builds on the Internet of Things (IoT) paradigm to a landscape where internet-enabled devices permit people, processes, data, and things to make networked connections that are more relevant and valuable than ever before. This shift is being driven by smaller and more powerful wearable devices that allow items such as smart watches, glasses, health and fitness trackers, etc. to be worn on the body to collect data and transmit this information, over the network, to provide real-time sensing [13]. This market has seen a surge in such items and by 2018 it is predicated that globally, there will be 177 million wearable devices [13]. Furthermore, by 2020 there will be 50 billion internet-connected devices, which will surpass the projected world population of 7.6 billion; thus equating to 6.58 connected devices per person [14]. Additionally, advances in the areas of wireless communication, home automation and medical monitoring systems are also revolutionising the healthcare industry so that healthcare can be transferred from the hospital into the home [15].

One area where this surge in information capture, storage, retrieval and distribution is prevalent is within the area of lifelogging [16]. Lifelogging is a subset of pervasive computing and refers to the unified and continual digital recording of an individual's experiences that have been captured using a variety of sensors and that are stored permanently in a personal multimedia archive [17]. In other words, it is a platform that can be used to gather a continuous flow of personalized information, over a sustained period of time. Such records can then be used for a variety of activities and studies, including self-reflection, health monitoring and other social and health-related studies. For instance, people can examine patterns of their behaviours to reflect on their levels of activity and use this information to improve the quality of life [18]. One important type of information that is required for this activity is physiological data. The prevalent use of wearable devices is a popular method to capture such bodily information as these devices offer low battery utilization and often house a multitude of sensors that are capable of detecting physiological signals. As a result, the reflection of such personalized information provides precise and clear feedback of the user's psychological state in real-time, which may reinforce or contradict the users' self-appraisal [19]. For example, we may think that we are quite active but reflecting on our lifelogging data can either confirm or deny this belief and may result in altering behaviour, such as by walking more. Consequently, collecting data over a sustained period of time yields a phenomenal amount of information [20]. Lifelogs are complex and are composed of a variety of media and information. As such, this heterogeneous nature means that uses simple queries and ranking search results is unlikely to support many of the user's information retrieval tasks in this domain [20–21]. Instead, when creating lifelogs, data needs to be intelligently analysed with efficient indexing and data analysis tools so that such systems can learn and recognize different activities. In order to address this challenge, supervised learning can be used to apply predictive modelling to lifelogging data to address these requirements.

Activity recognition is one of the major focuses of sensing technologies and assisted living [22–24]. However, due to the nature of sensor data, existing approaches are limited to transport and movement mode detection. For instance, body-worn accelerometers are sensitive to placement but do achieve a higher level of accuracy compared to those that are housed in smartphones [22–23]. For example, wearing such a device on the hip has been shown to achieve a higher recognition of activity compared to smartphones, which are often in pockets or in bags [23]. Additionally, the use of visual systems is another avenue that provides an outlet for activity recognition [15], [24–26]. Such systems are mainly focused on assisted living to monitor the activities and behaviours of older people in a smart home environment [24], [26]. Utilising devices such as cameras, microphones, presence sensors, temperature sensors and smart appliances daily activities can be recognised [15], [24–26]. This in turn enables older people to live more independently as their environment is constantly being monitored for their safety. For instance, a change in someone's daily routine could signal an underlying health issue [24]. Nevertheless, continually collecting data produces a phenomenal amount of information. As the accumulation of data increases, the need to provide efficient and intelligent methods of analysing this information is evident. As these datasets grow in size, this provides us with more information that can be used to influence our behaviour. This is an interesting approach, which will be explored further so that dynamic and detailed memories can be created that have a real impact over our lifestyle decisions.

In recognition of these issues, this paper posits our approach that has been designed to 1) collect streams of lifelogging data from wearable accelerometer and heart rate devices, 2) process this information using a combination of signal processing and feature extraction/selection techniques, and 3) utilise supervised machine learning algorithms to detect levels of human activity. As such, the goal of the paper has been to evaluate the performance of the classifiers, which are able to distinguish physical activity from personal lifelogs. Furthermore, we also evaluate the performance of the system in terms of 1) the total processing time required for pre-processing and extracting features from 30 seconds worth of raw data. 2) The total time that it takes, per classifier to classify the data. 3) The differences between the costs of transferring data from a smartwatch to a smartphone, from two different perspectives: battery and transfer time. Accelerometer and heart rate data has been gathered from two publically available datasets – PAMAP2 Physical Activity Monitoring Data Set [27] and Casale et al.'s activity recognition dataset [28]. Furthermore, a smartwatch interface has also been presented that is able to display the results to the user. This outlet can be used for reflection and to influence positive behavioural changes, such as promoting increased physical activity to prevent obesity. It should be noted that, as the main goal of the paper has been to evaluate the classification techniques and the performance of the system, the smartwatch interface has been utilized as a use case to demonstrate our approach and how it could be used in the future to influence behaviour. Therefore, an evaluation of this application, in relation to obesity or usability evaluation of the smartwatch interface, is out of the scope of this paper.

The reminder of this paper is constructed as follows. Section 2 describes related work in the areas of mobile sensing and supervised learning. Section 3 details the approach that we have used to detect physical activity from a collection of lifelogs that have been generated from a study in which multiple users have worn body-mounted sensors to collect physiological data. The results of our supervised learning approach and an evaluation of the system's performance have been presented in section 4, whilst section 5 presents a discussion of these results. Finally, the paper is concluded and the future directions of the research are presented in section 6.

2 Background

One of the main factors that fuels inactivity is lack of awareness and the fallibility of human memory [29]. Our biological memory system is highly selective, is susceptible to forgetting and misremembering events and will fade over time [30]. However, collecting and presenting personal data can be used to gain a greater insight into ourselves and can be used to help people become more aware of their physical activity levels to motivate behavioural change [29]. In aiding this change, technology has enabled human experiences to be enriched in more ways than ever before. This has led to a new paradigm of computing for human experience (CHE) [31] that aims to use a convergence of technology to serve, assist, and cooperate with people to unobtrusively complement and enrich their lives, with minimal human interaction. A key contributor of this approach is the abundance of information that is available through activity recognition platforms, which allow us to capture, store, upload and share almost every moment of daily life. Utilizing data analysis methods, such as machine learning and signal processing, this data can then be explored to enable the creation of intelligent systems that are able to start learning about our lives. As such, this section provides an overview of previous work in the areas of mobile sensing and supervised learning.

2.1 Mobile Sensing

As technology advances, smaller and more powerful sensing devices are being developed that provide us with the ability to constantly monitor ourselves. As such, a new movement of continuously logging our lives has emerged that has enabled us to bring the areas of lifelogging and wearable computing to a new level of sophistication. One of the ultimate goals of the Quantified Self movement (a group of people who diligently track many kinds of data about themselves) and health-related technologies is to a shift the focus from patient care to preventive care, i.e. through monitoring vital biological signs and human activities [32]. Existing mobile and wearable apparatuses are able to track users' activities and monitor some vital signs such as heart-rate. The resource efficient integration of accelerometer sensors into wearable devices, enables these devices to count steps and physical activities. Single sensor tracking, such as counting steps, could be performed on these devices, with more complex activity recognition requiring sensor fusion [33]. For instance, while a user's hand is moving (accelerometer), the audio sensor can be used to detect if the user is inside a vehicle or not. However, the downside of such sensor fusion activities is battery consumption, which is a burden for small and ubiquitous devices such as smartphones or wearables [34]. Therefore, existing commercial implementations of activity recognition, such as Google Play Services[1] or Intel Context Sensing Library[2], perform this process on the cloud. However, handling such processes in the cloud does pose network and privacy issues, but it has the advantage of preserving battery.

[1] https://developer.android.com/google/play-services/index.html
[2] https://software.intel.com/en-us/context-sensing-sdk

As a result of this limitation, existing fitness trackers or smartwatches are usually restricted to step counting, heart-rate monitoring or manual sleep monitoring. Therefore, their reflection mechanisms, such as visualizations, are univariate.

One such approach, by Rawassizadeh et al. [35] is UbiqLog, an open source, lightweight and extendable lifelogging approach for smartphones. The approach collects user-centric data from smartphones, including application usage, other devices in proximity of the user via Bluetooth, SMS messages, call logs, pictures, location and high level activities of the user (e.g. walking, being in a vehicle, running or being on a bicycle). It has been used in a variety of lifelogging studies, where a fundamental requirement that was uncovered was the need to provide a method that identifies correlations between different behaviours. As a result, it has been established that visualizing activities is a very useful outlet aids in assisting users in quantifying their behaviours. However, depicting only one activity (univariate visualization) does not provide an appropriate outlet and is unsuitable for illustrating such correlations between human activities. Therefore, there is a need for multivariate visualizations that are capable of providing such a method of activity-related visualizations. In other works, the Lifestreams approach [36], obtains raw contextual data from the explicit feedback of users (offline surveys), as well as continuous streams of data that have been collected from sensors or applications on-board the mobile device. A change-point detection algorithm has then been proposed to identify correlations between human activity and feedback. However, a drawback of this work is the need for interaction from the user. This is in contrast to UbiqLog [35], which collects sensors data without the need for manual interactions from the user.

Furthermore, a subset of approaches focuses on energy-efficient inferring of users' activities, while collecting contextual data. For instance, the ACE system (Acquisitional Context Engine) [37] mines co-occurrence patterns among context events in an energy-efficient manner. Similarly, MobileMiner [38] and SeeMon [39] focus on efficient and continuous context monitoring. MobileMiner uses association rule mining for predicting contextual activities on the phone, unlike ACE, which pushes all context data to a remote server where the pattern mining takes several hours to complete and therefore has privacy, latency, and data cost issues. SeeMon uses three optimizations: discovering change in context at earlier stages of the processes, exploiting temporal continuity of contexts, and choosing a small subset of sensors sufficient to answer a query. Similar to these approaches, Balan et al. [40] take into consideration each sensor's effect on energy consumption in different sensing policies, whilst Cui et al. [41] propose an approach to reduce the usage of the battery by increasing the sampling rate when the user is moving (detected from accelerometers) and reduce the sampling rate whilst the user is stationary.

As smaller and more powerful devices are developed, this presents us with unique opportunities to harness their power for the purpose of multimodal sensing. These platforms provide an unobtrusive method of collecting statistically relevant data that can be processed and visualized to prompt behavioural change. With such visualizations, a user might be able to identify factors that affect his or her desired behaviour. For instance, driving to work may prevent the user from completing their daily walking plan.

2.2 Supervised Learning in Mobile Sensing

Supervised learning is an area that is well suited to analysing behavioural data, as patterns can be found relatively quickly. For example, a pattern could relate to particular action or mood, which can then be learned over time by the system to detect and predict behaviour. Currently, such methods have been used in photo analysis to discern activity. In one such approach, Byrne et al. [42] have used supervised learning to analyse images to detect semantic concepts. In this work, images have been analysed to detect the user's movements, e.g. reading, giving a presentation or being inside a vehicle. The system has been 75% accurate in classifying the images. As well as analysing images, activity recognition and classifying behaviour can also be achieved using machine learning algorithms [43–45]. In one such approach, Lee and Cho [43] have developed a multi-modal context-aware system, which collects data from a number of wearable sensors, including accelerometers, gyroscopes, physiological sensors, and data gloves. The system is able to recognize the user's activities using probabilistic models, which are built by using an evolutionary algorithm. The optimal probabilistic models, one Context Versus-All (OVA) dynamic Bayesian networks (DBNs), deal with the uncertain, incomplete, and temporal observations from the sensors [43]. The results indicate that different activities can be recognized, using this method. However, whilst activities can be recognized other supporting data is missing, such as photos or location.

In similar work, Qiu et al. [44] use accelerometer data and a Support Vector Machine (SVM) to classify accelerometer features into user activities (sitting or standing, driving, walking or lying down) [44]. The accuracy of each activity was around 90%. This work illustrates how machine learning, and a wearable accelerometer can be used to identify the activities of a wearer to a very high accuracy [44]. Nevertheless, whilst these results are encouraging, the location of these movements is unknown and the devices used are expensive and proprietary in nature. The next section describes our approach for detecting physical activity in lifelogs. In this way, we describe the collection, processing and visualisation of raw sensor data that has been obtained from the wearable accelerometers and heart-rate monitor.

3 Physical Activity Detection in Lifelogs

The availability of smaller body sensing devices presents us with unique opportunities to analyse this data so that lifelogging technologies can move forward into intelligent systems that can learn about their user. As such, our approach, depicted in Fig. 1, focuses on collecting instances of lifelogging data, processing this information and returning these results to the user via their mobile device (e.g. smartwatch or smartphone). Using our approach, we are able to:

1) Collect raw data, including accelerometer and heart rate information, using a number of wearable sensors
2) Process and analyse this data, using various signal processing, feature extraction/selection techniques and classification algorithms, to ascertain the types of activities that are being undertaken
3) Transform this information into a visual illustration of the user's levels of physical activity

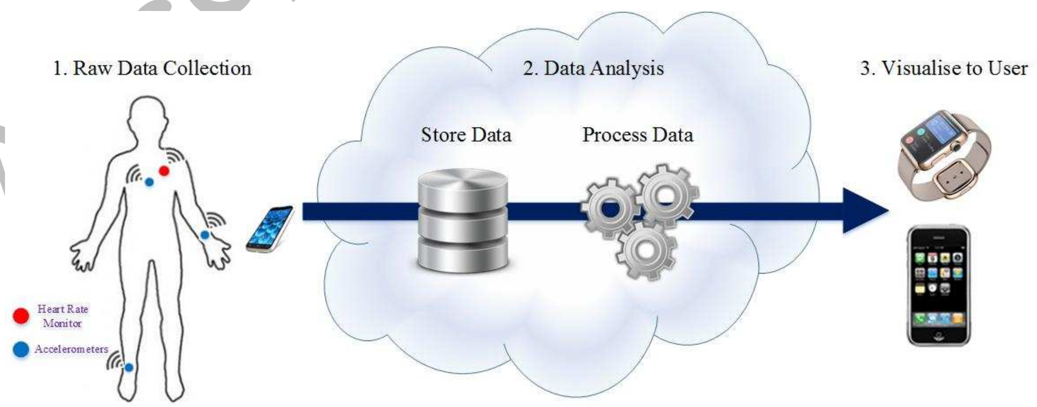


Fig. 1. High-level overview of our approach

In this way, a variety of information can be collected, examined and brought together, to form a snapshot of our levels of activity, which can be used for self-reflection. In realising this idea, and due to the configuration of the datasets, our study focuses on evaluating the performance of ten classifiers abilities to categorise up to nine activities, across a group of 22 subjects. However, prior to classifying the data, a data pre-processing flow is required in order to prepare the collected raw data. This process has been depicted in Fig. 2, with the remainder of this section detailing these steps.

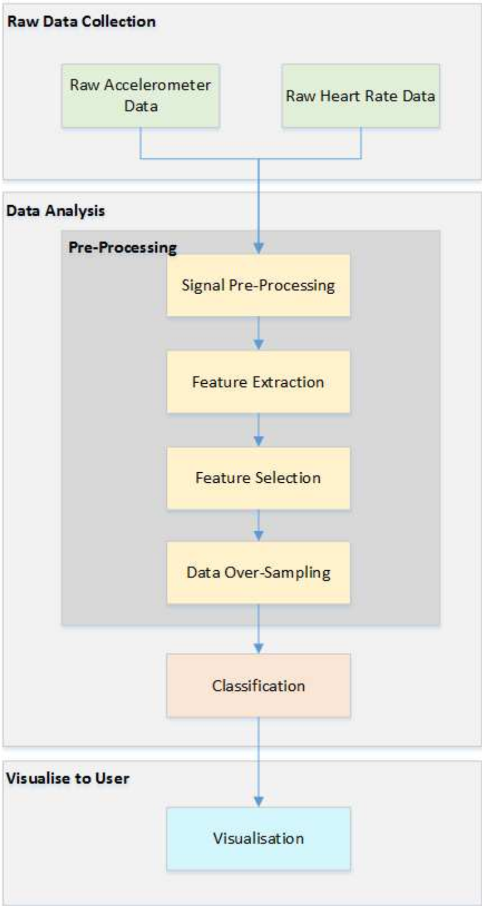


Fig. 2. Data processing steps of our approach

In order to test the validity of our approach one baseline method and two other methods have been employed (see Fig. 3).

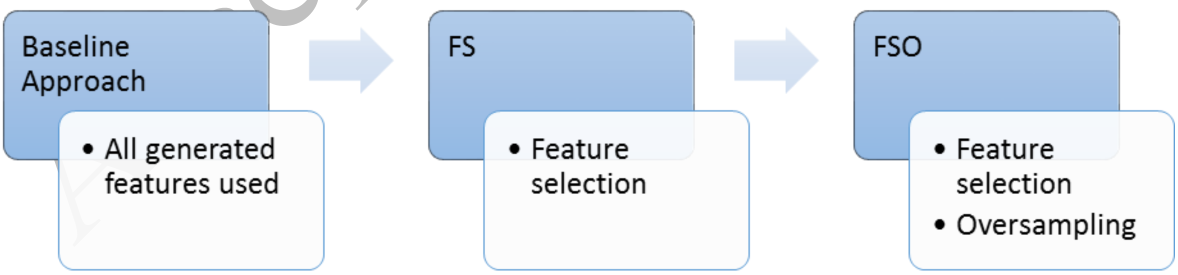


Fig. 3. Methods that we have used to illustrate the effectiveness of our method

The baseline approach utilises all of the generated features. The next method, FS (Feature Selection), will then apply our feature selection approach. This will enable us to remove redundant features and to select a subset of the most important features to determine if the baseline results can be improved upon. The data will not be oversampled within the Baseline or FS approaches. The next method, FSO (Feature Selection and over Sampling), will then oversample the reduced feature set from the FS approach. The results from each method will then be compared within the evaluation.

3.1 Raw Data Collection

Our approach uses two publically available datasets – PAMAP2 Physical Activity Monitoring Data Set [27] and Casale et al.'s activity recognition dataset [28]. The PAMAP2 dataset will here on out be referred to as dataset 1. It contains 13,524,350 instances of raw multivariate, time-series data that has been recorded from three tri-axial accelerometers and a heart-rate monitor. Meanwhile, Casale et al.'s activity recognition dataset will here on out be referred to as dataset 2. This dataset contains 1,926,896 instances of raw univariate, sequential time-series data that has been recorded from a single chest-mounted accelerometer.

In order to collect data, subjects within dataset 1 undertook a series of nine activities that are a blend of inactive states (e.g. sitting) and highly activity (e.g. running). Each subject adhered to the data collection protocol, which included performing each activity for up to three minutes, with one-minute breaks. Subjects within dataset 2 undertook a similar data collection protocol and performed seven activities that ranged from working at a computer to walking. In this instance, data has been collected from 15 participants. Table 1 details the activity of each dataset, the associated Metabolic Equivalent Task (MET) rate, which can be used as an indication of the intensity of a physical activity, and level of activity. The MET rate and level of activity in Table 1 have been provided by using the Compendium of Physical Activity [46], which is a common reference point in the area of estimating energy expenditure of physical activity [47].

Table 1. Summary of Activities Performed [27]–[28]

Activity ID	Activity	Dataset	MET Rate	Activity Level
1	Lying	1	1.0	Light
2	Sitting	1	1.8	Light
3	Standing	1	1.8	Light
4	Ironing	1	2.3	Light
5	Descending Stairs	1	3.0	Moderate
6	Vacuum Cleaning	1	3.5	Moderate
7	Normal Walking	1	3.3 – 3.8	Moderate
8	Running	1	7.0 – 8.0	Highly Energetic
9	Ascending Stairs	1	8.0	Highly Energetic
10	Working at Computer	2	1.5	Light
11	Talking while Standing	2	1.8	Light
12	Standing	2	1.3	Light
13	Standing Up, Walking and Going up\down stairs	2	6.5	Moderate
14	Walking	2	3.3 – 3.8	Moderate
15	Walking and Talking with Someone	2	4.0	Moderate
16	Going Up\Down Stairs	2	3.0 - 8.0	Moderate – Highly Energetic

As it can be seen, the dataset is composed of a suitable mixture of high and low energy actions that will test the classification algorithms' abilities to separate these times.

3.2 Data Analysis

As detailed in Fig. 2, the data analysis stage requires the execution of a number of steps in order to prepare the data before it can be classified. The analysis that has occurred within this stage has been carried out using Matlab v2015a. The below section explains the approach that has been used to undertake this activity.

3.2.1 Signal Pre-processing

Collecting raw physiological data, in general, produces a phenomenal amount of information, which is susceptible to noise. In particular accelerometer data, is sensitive to noise interference, position and movement [22], [48]. Therefore, pre-processing this data essential, in order to characterize the physical activity of the user, within a certain time frame [49]. Within dataset 1, the sampling frequency of the accelerometers was 100 Hz, whilst the heart-rate monitor was 9 Hz. These accelerometers were situated on the dominant side's ankle, around the chest and on the wrist of the dominant arm. Within dataset 2, the sampling frequency of the accelerometer was 52 Hz and this was mounted on the subjects' chest. In order to remove noise and interference from the accelerometer data, a number of filters have been applied (see Fig. 4).

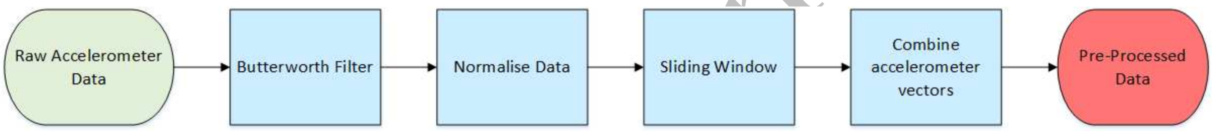


Fig. 4. Model to filter and prepare the accelerometer data

Using the Matlab Filter Design and Analysis tool (FDA)[3], a second-order forward-backward digital low-pass Butterworth filter, with a cut-off frequency of 3 Hz, has been designed and applied to the data. As demonstrated in previous work [50–52], this cut-off frequency is appropriate to filter the data, without losing any information. The signals have then been normalized and a sliding window of 512 samples, with an overlap of 50%, has been applied to the data. The sliding window size corresponds to 5.12 seconds, which is a reasonable amount of time to obtain a suitable period of movement [27], [53–57]. Each accelerometer provides three acceleration vectors of A_X, A_y and A_Z signals. Pythagorean Theorem, as shown in Equation 1, has been used to combine these axes together into a single acceleration vector (a), per accelerometer [58–59].

$$a = \sqrt{Ax^2 + Ay^2 + Az^2} \tag{1}$$

As acceleration data is recorded as minus numbers, squaring the axis values (A_X, A_y and A_Z) ensures that a valid value is returned. The data is now ready for features to be extracted.

3.2.2 Feature Extraction

Features are required so that the machine learning algorithms can classify the data. When these signals are processed, they can be analysed in two modes, time and frequency. In the time domain, simple mathematical and statistical metrics can be used to extract basic signal information from raw sensor data and depicts how a signal has changed over time [49]. In contrast, frequency domain analysis illustrates how the signal's energy is distributed over a range of frequencies [60]. As such, frequency domain techniques have been extensively used to capture the

[3] http://uk.mathworks.com/help/signal/examples/introduction-to-the-filter-design-and-analysis-tool-fdatool.html

repetitive nature of a sensor signal. This repetition often correlates to the periodic nature of a specific activity such as walking or running [49]. The advantage of frequency-related parameters is that they are less susceptible to signal quality variations [61]. Taking a similar approach to [27], [53–57], [62–66], feature vectors have been generated from both domains.

Time Domain Features

From the time domain, within dataset 1, the features that have been extracted include, the mean from the heart-rate monitor, and from the three accelerometers the mean, median, standard deviation, root mean square, variance, and correlation between the ankle to chest, chest to hand and hand to ankle, respectively. Within dataset 2, the features that have been extracted from the chest accelerometer include the mean, median, standard deviation, root mean square, and variance. Each feature has been calculated, per activity and subject (see Fig. 5).

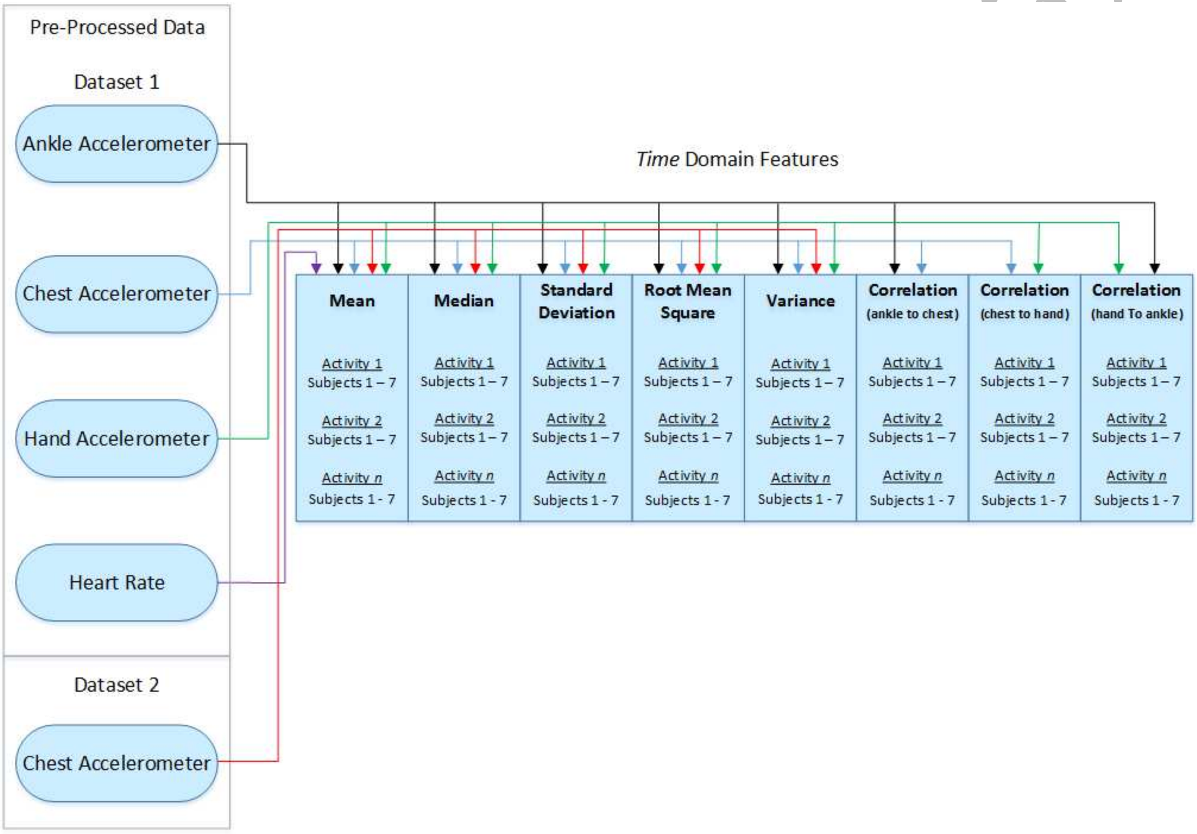


Fig. 5. Feature extraction of the pre-processed data from the time domain

Following the approach in [67], Correlation has been calculated using Pearson's Correlation as shown in Equation 2.

$$\text{Correlation} = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sqrt{\left(\sum x^2 - \frac{(\sum x)^2}{n}\right)\left(\sum y^2 - \frac{(\sum y)^2}{n}\right)}} \quad (2)$$

In this way, given two distinct signals, Pearson's Correlation calculates the correlation between each pair of accelerometer vectors. In this instance, this has been applied between the ankle–chest, chest–hand and hand–ankle accelerometers within dataset 1.

Frequency Domain Features

As previously stated, when analysing signals, translation into multiple domains (time and frequency) is often required. As such, a signal can be converted between these domains using a mathematical operator called a transform (see Fig. 6). One such approach is a Fourier Transform FFT, which decomposes a function into the sum of a (potentially infinite) number of sine wave frequency components. The 'spectrum' of frequency components is the frequency domain representation of the signal [60]. Using Fast Fourier Transform (FFT) and Power Spectral Density (PSD) the signal has been converted from the time to the frequency domain. However, prior to calculating the FFT, the Direct Current (DC) component of the signal needs to be determined and removed (see Fig. 6), as is the case in several studies [53], [65–66]. This element is the mean acceleration of the signal and is often much larger than the remaining coefficients, which results in the signal being distorted [49], [66].

Once the signal has been transformed into the frequency domain, the extraction of frequency-related parameters can occur. From the frequency domain, energy, entropy, peak frequency and median frequency have been calculated. Again, each feature has been calculated, per activity and subject (see Fig. 6).

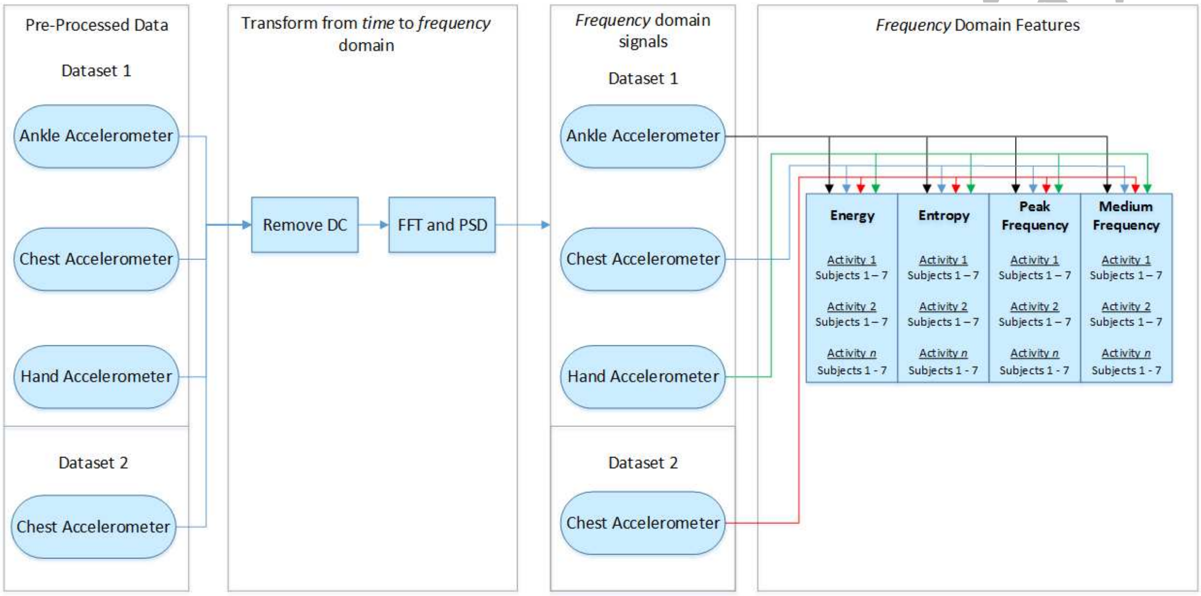


Fig. 6. Feature extraction of the pre-processed data from the frequency domain

Energy has been calculated using Equation 3, whilst Equation 4 calculates the entropy [54]. When calculating the energy of the signal, the sum of the squared discrete PSD components (x) of the signal are utilised. The sum was then divided by the window length for normalization [53].

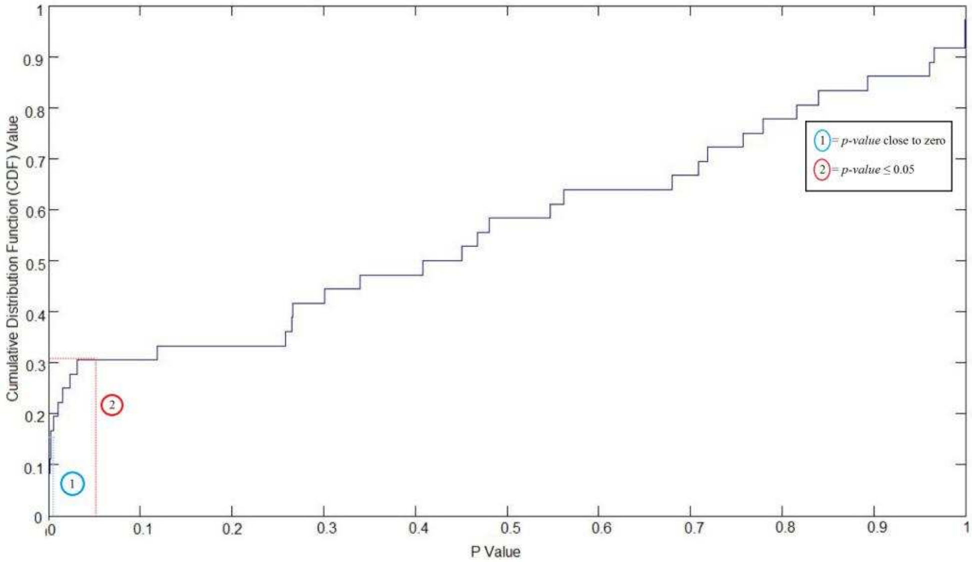
$$Energy = \frac{\sum[x]^2}{\text{length(x)}} \tag{3}$$

$$Entropy = \frac{-\sum[x]\log[x]}{\text{length(x)}} \tag{4}$$

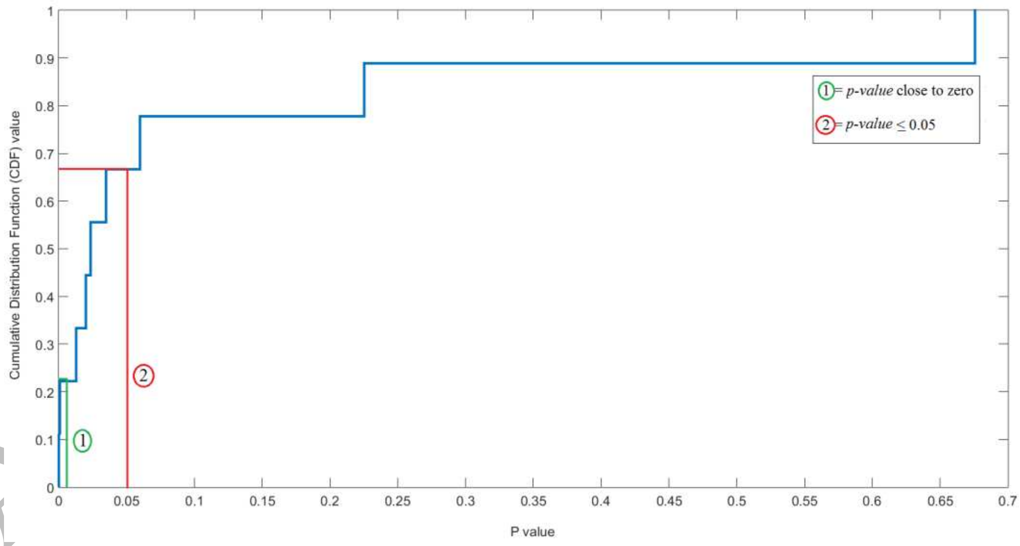
After analysing the literature [27], [53–57], [62–66], these features have been chosen because they represent a range of information about the signal. This method provides an approach that enables data to be condensed into a smaller amount of more useful information.

3.2.3 Feature Selection

Feature selection, or dimensionality reduction, has then been performed in order to find a subset of the most important features. This is necessary, as some of the features might be redundant [68]. To calculate this, the discriminant capabilities of each feature, per dataset, have then been determined using statistical significance. In this way, the datasets have been separated into two groups of training data and the t-test of each feature has then been measured, and the p-values are compared and plotted. This measurement is used to get a general idea of how well separated the two groups are [69–70]. Fig. 7 a) illustrates the resulting p-value plot for dataset 1. As represented by the blue line, 15% of the features have a p-value close to zero. Additionally, as depicted by the red line, approximately 30% of the features have a p-value ≤ 0.05 (when p > 0.05 non-significance is frequently reported [71]). This illustrates that, within dataset 1, ten features are significant. Fig. 7 b) illustrates the resulting p-value plot for dataset 2. As it can be seen, as represented by the green line, approximately 23% of the features have a p-value close to zero. Additionally, as depicted by the red line, approximately 66% of the features have a p-value ≤ 0.05. This illustrates that, within dataset 2, six features are significant.



a)



b)

Fig. 7. (a) Plot of the associated p-values of the features for dataset 1 and (b) p-values of the features in dataset 2

Forward Sequential Feature Selection (FSFS) has then been applied to the datasets to establish exactly which features are important. This feature selection algorithm, is one of the most widely used technique for feature selection and is particularly advantageous and robust against over-fitting [68–69]. The algorithm selects its features by sequentially adding features into the model until the fit is improved [69], [72]. Fig. 8 illustrates the FSFS approach by plotting the misclassification error (MCE) on the test sets as a function of the number of features. In this instance, .the MCE is the number of misclassified observations divided by the number of observations in the dataset.

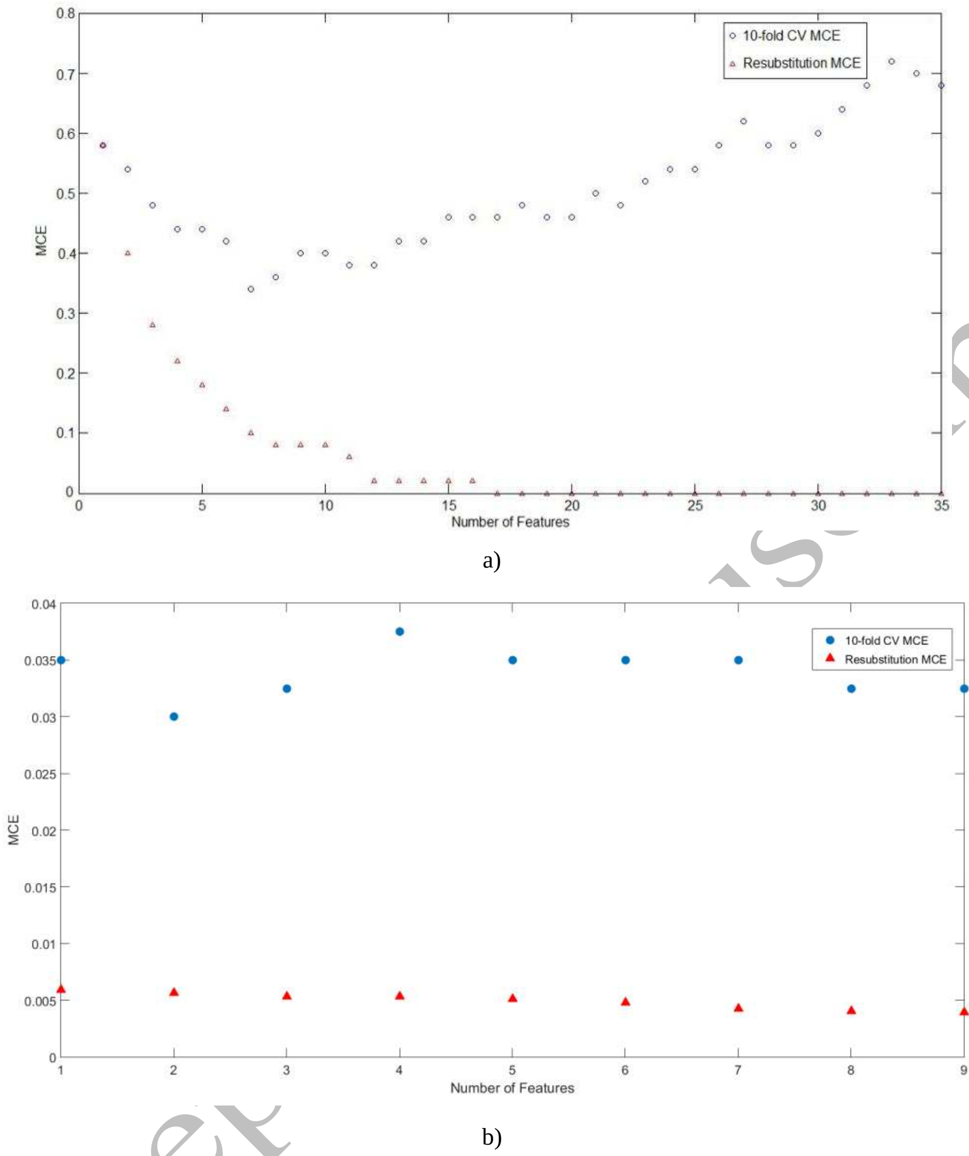


Fig. 8. (a) Forward Sequential Feature Selection (FSFS) method for dataset 1 and (b) FSFS for dataset 2

Fig. 8 a) illustrates the resulting MCE plot for dataset 1. As it can be seen, over-fitting occurs after ten features, as the test error values increase and the re-substitution rate decreases; thus ten is deemed an adequate amount of features to select from the feature set. This also validates the p-value calculation that has been depicted in Fig. 7 a). Through our process of feature extraction, the features that have been selected from dataset 1 include heart rate_mean, hand_mean, chest_mediumFrequency, ankle_peakFrequency, ankle_correlation, hand_variance, chest_variance, ankle_median, ankle_standardDeviation and chest_standardDeviation.

Fig. 8 b) illustrates the resulting MCE plot for dataset 2. As it can be seen, over-fitting occurs after six features, as the test error values increase and the re-substitution rate decreases; thus six is deemed an adequate amount of features to select from the feature set. This also validates the p-value calculation that has been depicted in Fig. 7 b). Through our process of feature extraction, the features that have been selected from dataset 2 include mean, standard deviation, variance, energy, entropy and peak frequency.

3.2.4 Data Over-sampling

A result of pre-processing and reducing the feature set has resulted in a condensed sample of data and in terms of learning behaviour, the minority of true positive results (Sensitivity) is lower. As a result, learning from data sets that contain very few instances of the minority class usually produces biased classifiers that have a higher predictive accuracy over the majority class, but poorer predictive accuracy over the minority class [73]. For example, in the case of recognising the activity of running, given a random sample taken from the dataset, the probability of a classifier classifying an activity as "running" is quite low. This is because the algorithms have a limited number of instances to learn about the features of this activity. The algorithms then have merely one chance to correctly identify running from the test data. The probability of accurately identifying running out of the majority of incorrect activities is quite low. However, in the case of oversampling the data, the algorithms would have more information to learn off and to test against, thus increasing the probability of correctly identifying the activity.

In order to address this problem, the FSO method will employ oversampling to resample the datasets and create synthetic records. Various resampling techniques are available, including under-sampling and over-sampling [74]. Under-sampling reduces the number of records from the majority class to make it equal to the minor class. Meanwhile, whilst over-sampling augments the minority class by exactly duplicating the examples of the minority class and thus enlarges the dataset [75]. In this instance, the Synthetic Minority Over-Sampling Technique (SMOTE) is used rather than reducing the dataset further [74]. Using SMOTE, both datasets have been oversampled using each activity label, in order to generate new synthetic records. This approach is an accepted technique for solving the problems related to unbalanced datasets [74].

3.2.5 Classification Algorithms

This study utilizes a number of powerful statistical classification algorithms that are often implemented for the task of activity recognition [76]. The classifiers considered include the linear discriminant classifier (LDC), quadratic discriminant classifier (QDC), uncorrelated normal density based classifier (UDC), polynomial classifier (POLYC), logistic classifier (LOGLC), k-nearest neighbour (KNNC), decision tree (TREEC), parzen classifier (PARZENC), support vector classifier (SVC) and Naive Bayes classifier (NAIVEBC) [77].

In this approach to data analysis, the classification algorithms are concerned with predictive modelling, i.e. using a sample of labelled training data, we want to test the algorithms ability to learn about this data to predict the behaviour of the unseen test data [78]. As such, the labels that have been used in our approach are the related activity IDs that have been discussed in Table 1.

Validation Methods

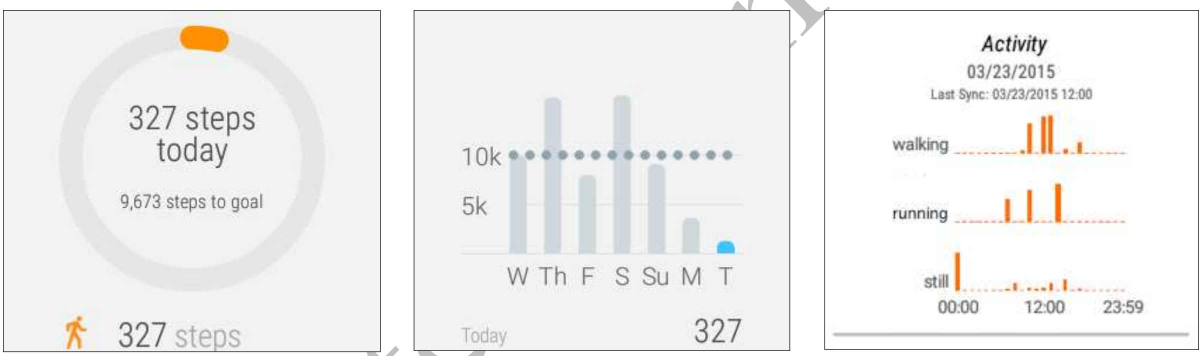
K-fold cross-validation and the holdout method techniques, as well as a Receiver Operating Characteristics (ROC) graph, have been used to determine the overall accuracy and validity of the algorithms. The holdout method is used to partition the dataset into two independent sets, a training set and a test set [79]. The test set is used to estimate only one parameter, i.e. the error rate, and the training set is used to train all other parameters of the classifier, therefore the training set must be larger than the test set [80]. Using a common approach, and to avoid overfitting, the dataset has been separated such that 80% of the whole dataset is designated for training and the remaining 20% for testing [80–82]. In order to maintain generalization, the learning and testing stages have been repeated. In this work, the average performance obtained from 100 cycle simulations has also been utilized. This number is considered, by statisticians, to be an adequate number of iterations to obtain an average [83].

The k-fold cross-validation technique has been used to estimate the accuracy of the classifiers. In this instance, the dataset is randomly partitioned into k mutually exclusive subsets. Training and testing is performed k times [79]. During this evaluation, k has been set to five, using 1 and 100 repetitions, respectively. Setting k = 5 is a typical choice as there is a lower variance in the data [72].

A ROC graph has also been used to summarize the classifier's performance and for visualization and organization. In order to directly compare the classifiers, their ROC performances have been reduced to a single scalar value (representing expected performance) by calculating the Area Under the Curve (AUC) [84]. In terms of the KNNC algorithm, k has been determined by using the leave-one-out error of the dataset.

3.3 Physical Activity Visualization

Using the classification methods described above, enables physical activity/inactivity to be identified from the user's physiological data. After this stage of analysis, the results need to be communicated to the user utilizing devices such as smartphones and smartwatches, this information can then be communicated, via multivariate visualizations, to the user so that they can see how often they are active/inactive (see Fig. 9). This platform has been implemented using Android wear on a Sony S3 smartwatch; with four core ARM Cortex-A7 1.2GHz CPU and 512 MB Ram.



a) b) c)

Fig. 9. (a) Daily smartwatch-based step-counter visualization (b) Weekly step-counter visualizations from GoogleFIT and (c) Our daily and hourly multiple-activity visualizations

Fig. 9 a) and b) depicts the default visualizations for Android Wear[4] implementations to illustrate a daily and weekly log of step count and the remaining number of steps to achieve a specified goal. However, the user is unaware of how this information maps to their activity levels and the types of activities that have been undertaken. Additionally, correlation between different activities is also not supported in existing visualizations. Taking this idea further, a prototype has been developed (see Fig. 9 c)) that visualizes different user activities. Users are now able to identify if there is any correlation between their behaviours. This implementation has been developed as a smartwatch application that aids users in reflecting upon their behaviours, through their collected data.

The next section describes the evaluation results that have been generated using each of the classifiers described in section 3.3. These results have been used to determine each classifier's overall performance, and accuracy, in separating physical activity in lifelogging datasets. An evaluation of the system's performance has also been undertaken. These results have been used to determine the battery consumption of collecting data, the

[4] http://www.android.com/wear/

total execution time that it takes to process the data and the differences, in terms of battery and transfer time, between the costs of transferring data from the smartwatch to the phone, has also been undertaken.

4 Evaluation

This section presents our results that have been obtained for classifying physical activity from lifelogs, using a number of supervised machine learning algorithms. The created datasets have been considered, using an 80% holdout technique and 5-fold cross-validation. The primary focus is to demonstrate how accurate the algorithms are at recognising activities. In this way, we describe the features of an activity and use probabilistic reasoning to filter human digital memory data that contain similar features to those described. The evaluation has been divided into three experiments that tests the classifiers accuracy against 1) the baseline approach, 2) the FS method, which utilises feature selection and 3) the FSO method that utilises feature selection and oversampling. The purpose of the evaluation is to determine if the results of the baseline approach can be improved upon using feature selection and oversampling. This evaluation has been carried out using Matlab and PRTools[5] (a Matlab pattern recognition toolbox). This section has then been concluded with an investigation into the issues of computation time and battery consumption.

The metrics used in this evaluation include the average sensitivity, accuracy (AUC), mean error, standard deviation, false positives (FPs) and false negatives (FNs). The results have been obtained over a 100 simulations that have utilised randomly selected training and testing sets for each iteration. Each of the metrics below has been extended for the multi-class problem and so the measures for multi-class classification have been based on a calculation of the average (μ) of each metric [85]. Sensitivity, also known as recall or true positive rate, has been calculated using the following expression:

$$S\mu = \frac{TP}{TP+FN} \tag{5}$$

In this case, the average sensitivity ($S\mu$) refers to the algorithms ability to correctly identify an activity. In this expression, TP refers to the number of true positives (e.g. running has been correctly classified as running) and FN refers to the number of false negatives (e.g. running has been classified but the user was walking). Additionally, false negatives (FNs) and false positives (FPs) have also been used as a measure within the ROC graph in order to summarize the classifier's performance. The false positive (FP) rate refers to an activity that has been incorrectly identified (e.g. walking has been incorrectly identified as another activity, such as running).

Accuracy refers to the algorithms precision in correctly identifying an activity and has been calculated by using the Area Under the Curve (AUC) function, testauc, in PRTools[6] and then calculating the average. Mean error refers to the average error rate of the classifiers. This has been calculated using the following expression:

$$MER\mu = \frac{1}{N}\sum_{i=1}^{N} a_{i.} \tag{6}$$

In this case, the mean error rate ($MER\mu$), has been calculated against the error rate (a) of the classifiers over 100 simulations (N).The error rate is the number of incorrectly classified instances that the classifier has calculated. Meanwhile, the standard deviation of the error rates has also been calculated using the following expression:

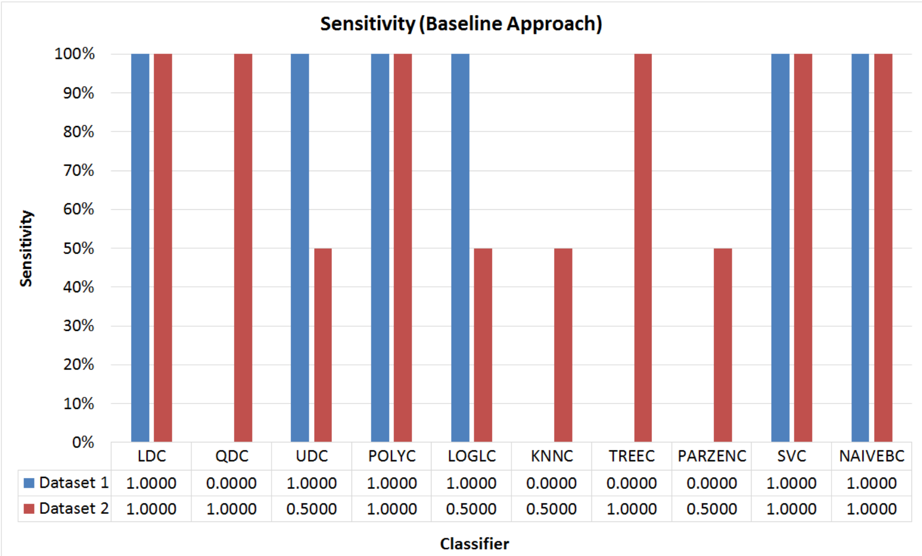
[5] http://prtools.org/
[6] http://www.37steps.com/prhtml/prtools/testauc.html

$$SD = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n}} \quad (7)$$

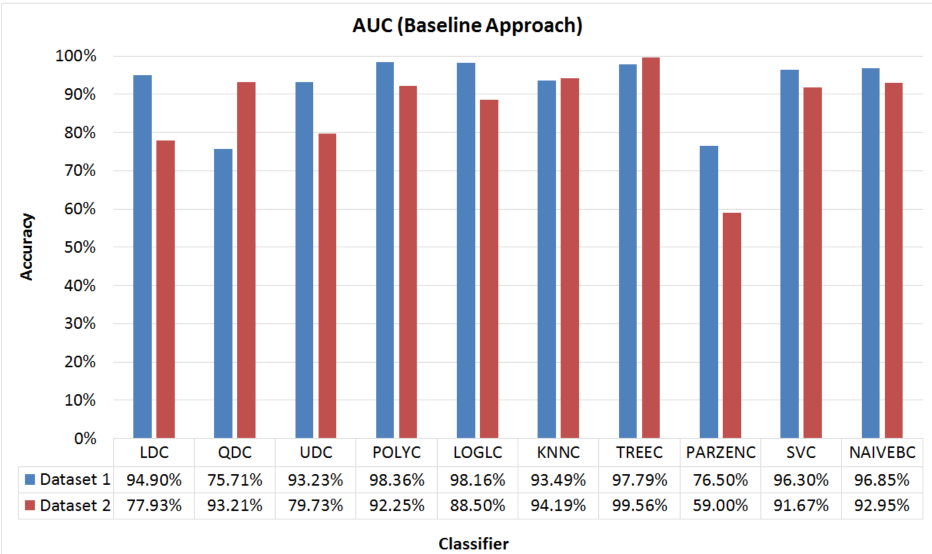
In this case, the standard deviation (SD) has been calculated using the mean (μ) of the error rate (x) of the classifiers over 100 simulations (n).

4.1 Baseline Method – Classifier Performance

This experiment uses datasets 1 and 2 of all the generated features, which have not been reduced using feature selection, nor have been oversampled. Fig. 10 illustrates the mean averages obtained over 100 simulations for the (a) sensitivity, and (b) accuracy for dataset 1 and 2.



a)



b)

Fig. 10. Classifier performance of datasets 1 and 2 for the baseline approach. These figures compare (a) sensitivity, and (b) accuracy between the two datasets

In this initial test, as illustrated in Fig. 10, both dataset did not perform particularly well. Although dataset 2 had a more stable sensitivity rate, these results were still quite low for a few of the classifiers. Dataset 1 did not perform well for a number of classifiers. Nevertheless, both did achieve 100% sensitivity with the LDC, POLYC,

Chelsea Dobbins, Reza Rawassizadeh, and Elaheh Momeni, “Detecting Physical Activity within Lifelogs towards Preventing Obesity and Aiding Ambient Assisted Living,” Neurocomputing, Dec. 2016, http://dx.doi.org/10.1016/j.neucom.2016.02.088

SVC and NAIVEBC classifiers. Dataset 2 outperformed dataset 1 in regards to QDC, KNNC, TREEC and PARZENC. In order to determine the accuracy of the classifiers, the k-fold cross-validation technique has also been used and the results are shown in Table 2.

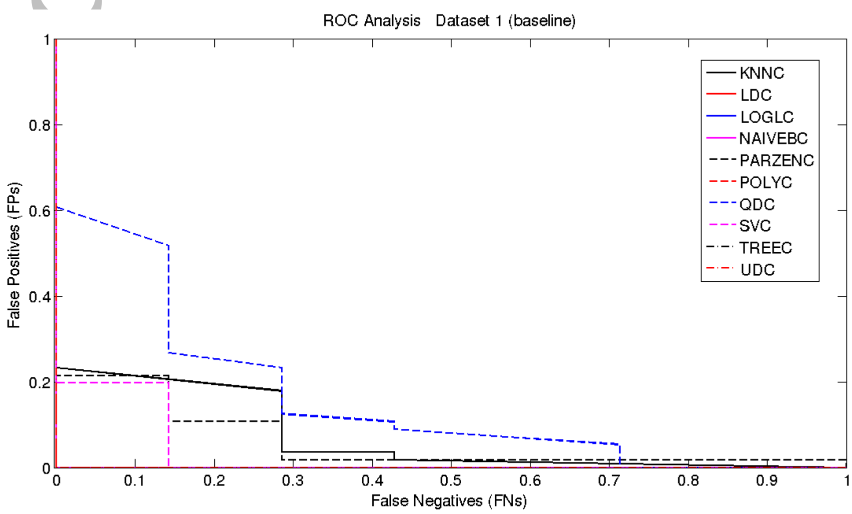
Table 2. Cross Validation results of datasets 1 and 2 (baseline approach)

	Dataset 1					**Dataset 2**				
Classifier	**80% Holdout: 100 Repetitions**		**Cross Val, 5 Folds, 1 Repetition**	**Cross Val, 5 Folds, 100 Repetitions**		**80% Holdout: 100 Repetitions**		**Cross Val, 5 Folds, 1 Repetition**	**Cross Val, 5 Folds, 100 Repetitions**	
	Mean Error	SD	Mean Error	Mean Error	SD	Mean Error	SD	Mean Error	Mean Error	SD
LDC	0.5556	1.12E-15	0.6508	0.6321	0.0490	0.5556	1.12E-15	0.6508	0.6321	0.0490
QDC	0.8889	1.79E-15	0.8889	0.8462	0.0292	0.8889	1.79E-15	0.8889	0.8462	0.0292
UDC	0.6667	7.81E-16	0.6984	0.6184	0.0356	0.6667	7.81E-16	0.6984	0.6184	0.0356
POLYC	0.4444	8.93E-16	0.4603	0.5563	0.0485	0.4444	8.93E-16	0.4603	0.5563	0.0485
LOGLC	0.6667	7.81E-16	0.6349	0.6503	0.0549	0.6667	7.81E-16	0.6349	0.6503	0.0549
KNNC	0.7778	6.69E-16	0.6032	0.6446	0.0409	0.7778	6.69E-16	0.6032	0.6446	0.0409
TREEC	0.2222	4.46E-16	0.2540	0.2529	0.0573	0.2222	4.46E-16	0.2540	0.2529	0.0573
PARZENC	0.7778	6.69E-16	0.6667	0.6624	0.0248	0.7778	6.69E-16	0.6667	0.6624	0.0248
SVC	0.5556	1.12E-15	0.6190	0.5794	0.0423	0.5556	1.12E-15	0.6190	0.5794	0.0423
NAIVEBC	0.4444	8.93E-16	0.6508	0.5222	0.0413	0.4444	8.93E-16	0.6508	0.5222	0.0413

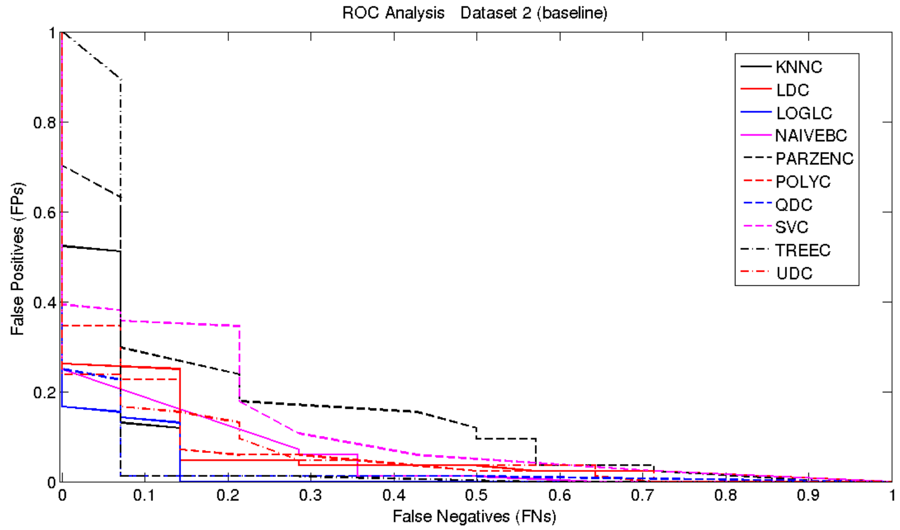
The k-fold cross-validation results, using five folds and one and one hundred repetitions illustrates that, across both datasets, the error rates have slightly improved, for some of the classifiers. However, the error rates are still slightly high. This could be attributed to the size of the dataset.

4.1.1 Model Selection

To evaluate the performance of each classifier, the roc function, within PRTools™, has been used. This function plots the false positives (FPs) against the false negatives (FNs). Therefore, the optimal point of the classifiers is as close to the axis as possible. As a result, the ROC curve depicted in Fig. 11 illustrates the cut-off values for the false negative and false positive rates, for each of the classifiers used in the baseline approach for datasets 1 and 2.



a)



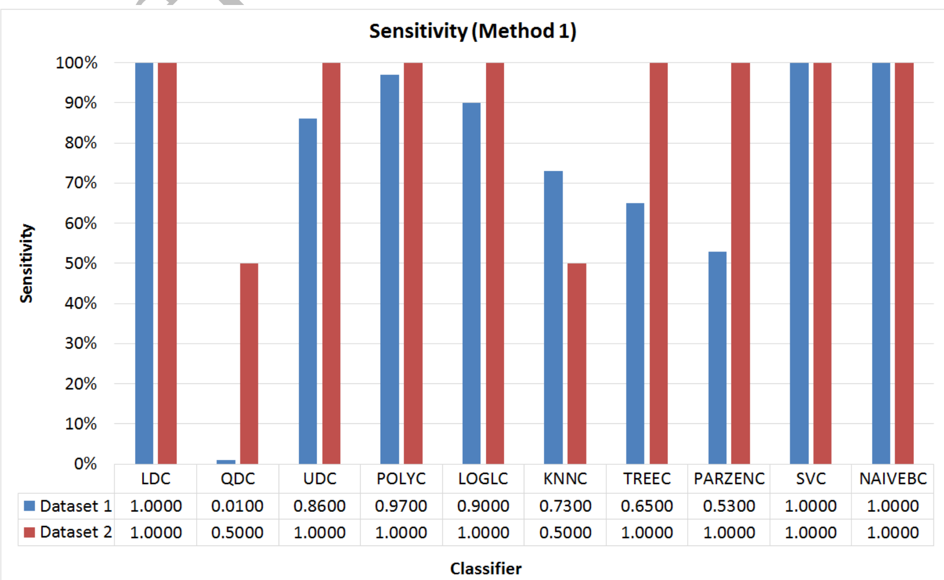
b)

Fig. 11. (a) Receiver Operator Curve (ROC) for all classifiers in dataset 1 and (b) ROC Curve for dataset 2 (baseline approach)

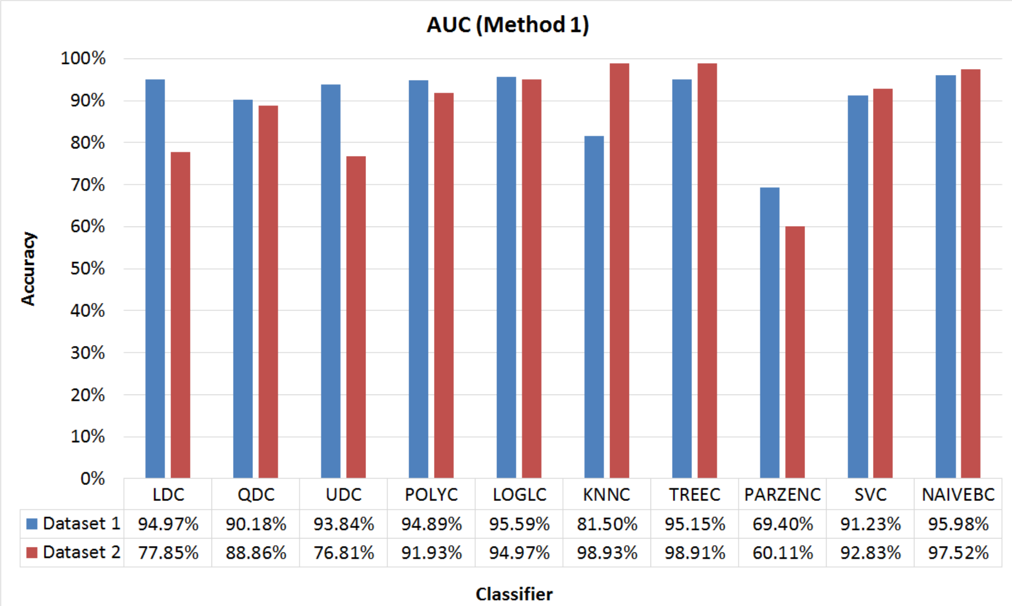
Regarding dataset 1 (see Fig. 11 a)), in terms of accuracy, several of the classifiers used performed well, such as the LDC and NAIVEBC, as they are at point 0, 0 on the graph. The high AUC and sensitivity values in Fig. 10 support these findings; NAIVEBC has an accuracy of 96.85%, and of sensitivity of 100%, whilst LDC's accuracy is 94.90%, and sensitivity was 100%. Regarding dataset 2 (see Fig. 11 b)), in terms of accuracy, several of the classifiers used performed well, such as the QDC and NAIVEBC. The high AUC, and sensitivity values in Fig. 10 support these findings; QDC has an accuracy of 93.21% and sensitivity of 100%, whilst NAIVBC's accuracy is 92.95%, and sensitivity was 100%.

4.2 FS Method – Classifier Performance Using Feature Selection

This experiment uses datasets 1 and 2 of generated features that have been reduced using feature selection but have not been oversampled. Similarly, to the baseline method, the performance for each classifier has been evaluated in terms of sensitivity (true positive rate), mean error, standard deviation and AUC (accuracy). Fig. 12 illustrates the mean averages obtained over 100 simulations for the sensitivity and AUC for dataset 1 and 2.



a)



b)

Fig. 12. Classifier performance of datasets 1 and 2 from the FS method. These figures compare (a) sensitivity and (b) accuracy between the two datasets

In this test, as illustrated in Fig. 12, dataset 2 had a better sensitivity rate than dataset 1 and for some of the classifiers higher accuracy as well. Nevertheless, dataset 1 still performed well, with high sensitivity and accuracy results for LDC, SVC and NAIVEBC, where 100% sensitivity was achieved. Dataset 2 outperformed dataset 1 in regards to QDC. In order to determine the accuracy of the classifiers, the k-fold cross-validation technique has also been used and the results are shown in Table 3.

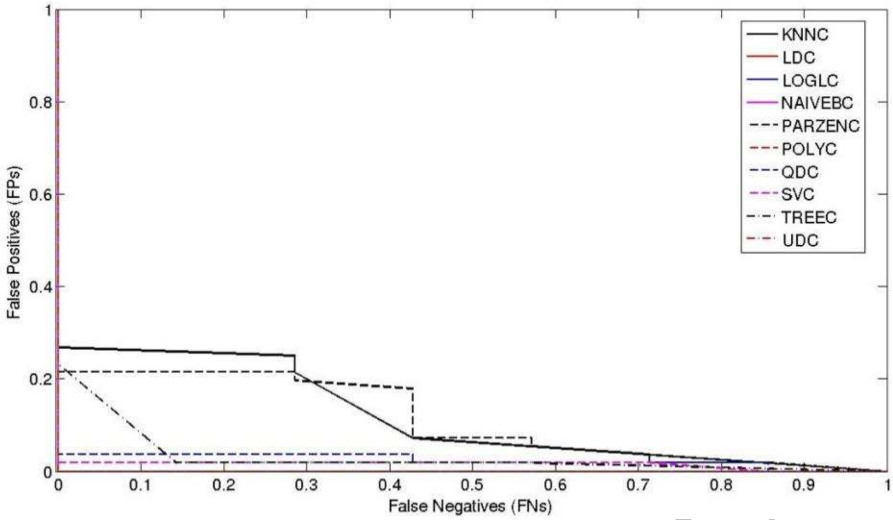
Table 3. Cross Validation results of datasets 1 and 2 (FS method)

	Dataset 1					Dataset 2				
Classifier	**80% Holdout: 100 Repetitions**		**Cross Val, 5 Folds, 1 Repetition**	**Cross Val, 5 Folds, 100 Repetitions**		**80% Holdout: 100 Repetitions**		**Cross Val, 5 Folds, 1 Repetition**	**Cross Val, 5 Folds, 100 Repetitions**	
	Mean Error	SD	Mean Error	Mean Error	SD	Mean Error	SD	Mean Error	Mean Error	SD
LDC	0.4044	0.1519	0.4762	0.4398	0.0409	0.5714	5.58E-16	0.5918	0.5631	0.0249
QDC	0.8767	0.0630	0.8571	0.8578	0.0267	0.5714	5.58E-16	0.5204	0.5227	0.0341
UDC	0.5056	0.1344	0.5556	0.5167	0.0372	0.6429	1.34E-15	0.5816	0.6288	0.0284
POLYC	0.4878	0.1385	0.5714	0.4843	0.0370	0.2143	2.79E-17	0.2653	0.2559	0.0254
LOGLC	0.5511	0.1455	0.6349	0.5721	0.0465	0.2143	2.79E-17	0.3061	0.2757	0.0555
KNNC	0.8133	0.1022	0.7937	0.7984	0.0280	0.2857	2.79E-16	0.4796	0.4455	0.0297
TREEC	0.6333	0.1399	0.6984	0.6606	0.0472	0.2143	2.79E-17	0.1531	0.1404	0.0376
PARZENC	0.8033	0.0919	0.7937	0.7978	0.0272	0.6429	1.34E-15	0.6939	0.7002	0.0237
SVC	0.5811	0.0984	0.5556	0.5935	0.0299	0.4286	5.58E-17	0.3878	0.3817	0.0306
NAIVEBC	0.6044	0.1153	0.5556	0.6051	0.0396	0.1429	1.39E-16	0.1122	0.0982	0.0181

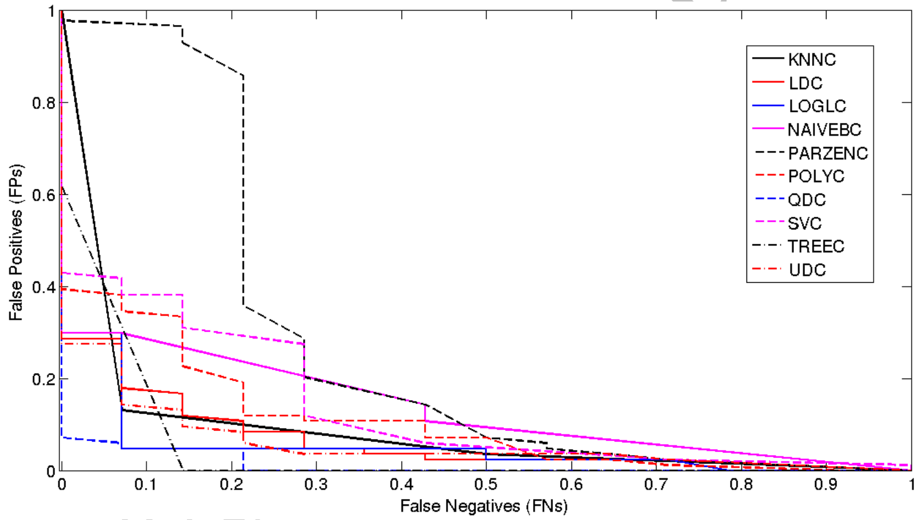
The k-fold cross-validation results, using five folds and one and one hundred repetitions illustrates that, across both datasets, the error rates have slightly improved, for some of the classifiers. However, the error rates are still slightly high. Again, this could be attributed to the size of the dataset.

4.2.1 Model Selection

Fig. 13 illustrates the cut-off values for the false negative and false positive rates, for each of the classifiers used in datasets 1 and 2 for the FS method.



a)



b)

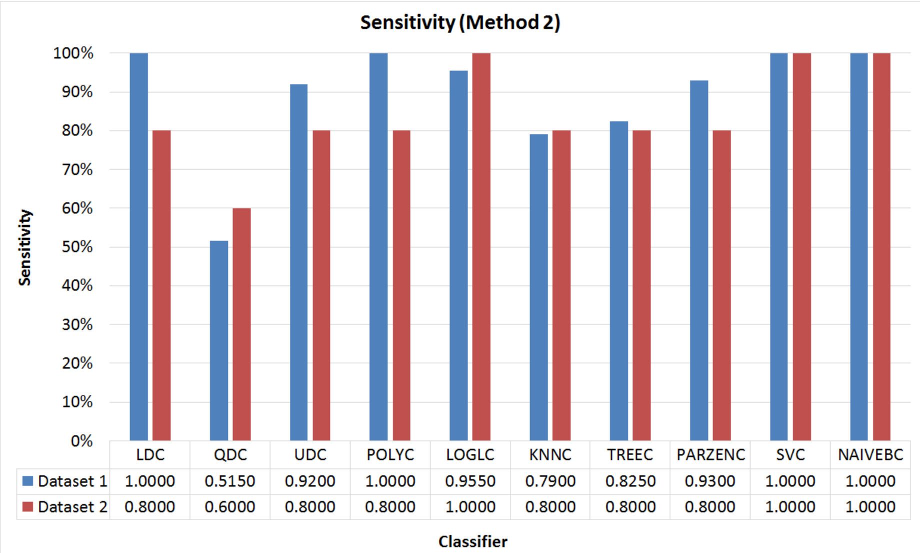
Fig. 13. (a) Receiver Operator Curve (ROC) for all classifiers in dataset 1 and (b) ROC Curve for dataset 2 (FS method)

Regarding dataset 1 (see Fig. 13 a)), in terms of accuracy, several of the classifiers used performed well, such as the LDC and NAIVEBC, as they are at point 0, 0 on the graph. The high AUC and sensitivity values in Fig. 12 support these findings; NAIVEBC has an accuracy of 95.98%, and of sensitivity of 100%, whilst LDC's accuracy is 94.97%, and sensitivity was 100%. Regarding dataset 2 (see Fig. 13 b)), in terms of accuracy, several of the classifiers used performed well, such as the TREEC and POLYC, as they are at point 0, 0 on the graph. The high AUC and sensitivity values in Fig. 12 support these findings; TREEC has an accuracy of 98.91% and sensitivity of 100%, whilst POLYC's accuracy is 91.93%, and sensitivity was 100%.

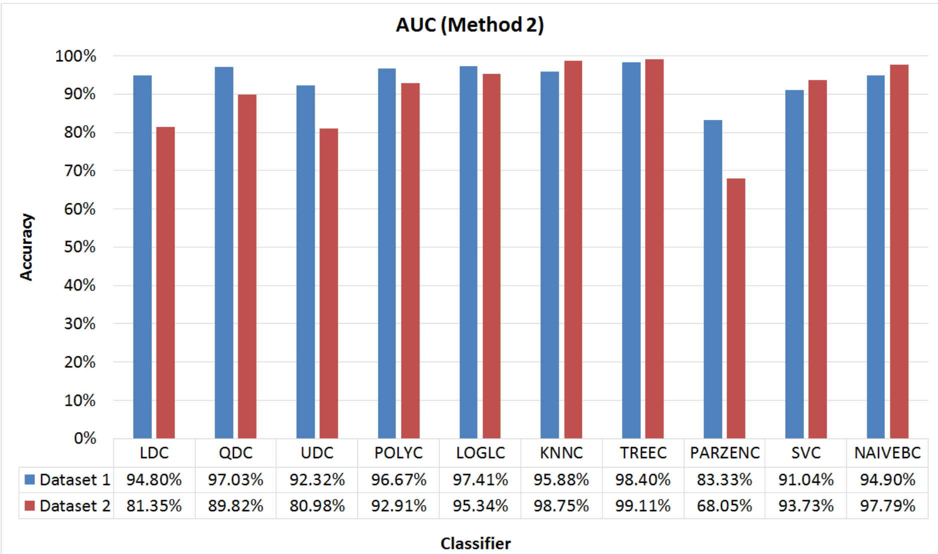
As it can be seen, reducing the number of features has improved the results, and for the majority of the classifiers, sensitivities are exceptionally high. However, an issue with the data is that the classification algorithms have a limited number of records to learn from and test against. As stated in section 3.2.4, a conventional method to rectify this issue is by over-sampling the dataset [86]. Therefore, to establish if the results can be further improved, the datasets have been oversampled and the experiments have been repeated.

4.3 FSO Method – Classifier Performance Using The Oversampled Datasets

This experiment uses datasets 1 and 2 of generated features that have been reduced using feature selection and have been oversampled. Fig. 14 illustrates the mean averages obtained over 100 simulations for the sensitivity and AUC for both datasets.



a)



b)

Fig. 14. Classifier performance for the oversampled datasets 1 and 2. These figures compare (a) sensitivity and (b) accuracy between the two oversampled datasets

In this test, as illustrated in Fig. 14, dataset 1 had a better sensitivity rate than dataset 2 and for most of the classifiers higher accuracy as well. Nevertheless, dataset 2 still performed well, with high sensitivity and accuracy results. Both achieved 100% sensitivity with the SVC and NAIVEBC classifiers. However, the sensitivity rate of LDC has decreased significantly, compared to the initial results. In order to determine the accuracy of the classifiers, the k-fold cross-validation technique has also been repeated and the results are shown in Table 4.

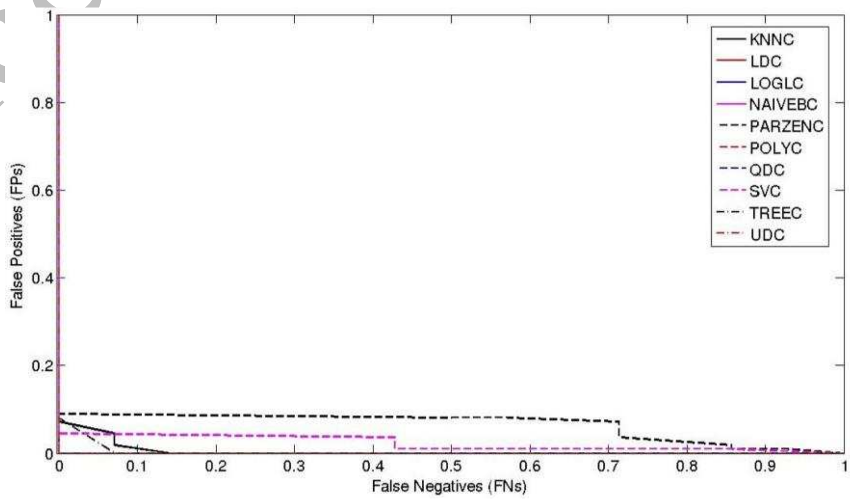
Table 4. Cross Validation results for the oversampled datasets 1 and 2

	Dataset 1					Dataset 2				
	80% Holdout: 100 Repetitions		Cross Val, 5 Folds, 1 Repetition	Cross Val, 5 Folds, 100 Repetitions		80% Holdout: 100 Repetitions		Cross Val, 5 Folds, 1 Repetition	Cross Val, 5 Folds, 100 Repetitions	
Classifier	Mean Error	SD	Mean Error	Mean Error	SD	Mean Error	SD	Mean Error	Mean Error	SD
LDC	0.2133	0.0899	0.2143	0.2217	0.0142	0.4286	5.58E-17	0.4847	0.4757	0.0140
QDC	0.5883	0.1273	0.5952	0.6334	0.0461	0.4000	7.81E-16	0.3929	0.3564	0.0174
UDC	0.3933	0.0988	0.4206	0.4000	0.0300	0.5143	1.23E-15	0.5255	0.5316	0.0197
POLYC	0.2506	0.0949	0.2460	0.2498	0.0137	0.1714	2.51E-16	0.1990	0.2064	0.0139
LOGLC	0.2822	0.1325	0.3016	0.2929	0.0491	0.1714	2.51E-16	0.2551	0.1915	0.0399
KNNC	0.3906	0.1170	0.4127	0.4046	0.0157	0.1714	2.51E-16	0.2194	0.1978	0.0143
TREEC	0.3039	0.0991	0.3254	0.3127	0.0313	0.1429	1.39E-16	0.0561	0.0842	0.0207
PARZENC	0.5689	0.0693	0.5556	0.5668	0.0204	0.5429	1.23E-15	0.5510	0.5497	0.0230
SVC	0.5217	0.0683	0.5635	0.5248	0.0233	0.3143	4.46E-16	0.3418	0.3318	0.0146
NAIVEBC	0.2489	0.0991	0.2778	0.2524	0.0170	0.0000	0	0.0204	0.0260	0.0065

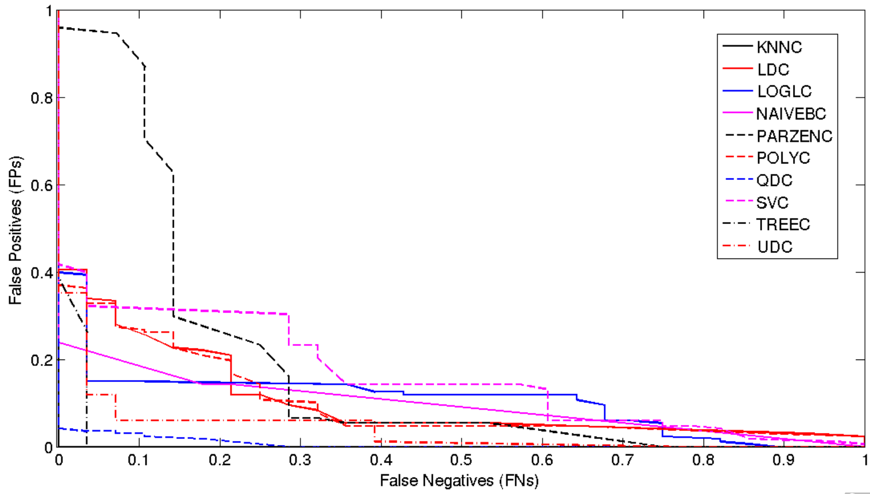
As it can be seen, repeating the experiment with the oversampled datasets, illustrates that the error rates have improved, for most of the classifiers.

4.3.1 Model Selection

As in section 4.1, the ROC curve has been calculated and graphed for both oversampled datasets. This illustrates the cut-off values for the false negative and false positive rates each of the classifiers used in datasets 1 and 2. Regarding dataset 1 (see Fig. 15 a)), in terms of accuracy, several of the classifiers used performed well, such as the LDC and NAIVEBC, as they are at point 0, 0 on the graph. The high AUC and sensitivity values in Fig. 14 support these findings; NAIVEBC has an accuracy of 94.90%, and of sensitivity of 100%, whilst LDC's accuracy is 94.80%, and sensitivity was 100%. Regarding dataset 2 (see Fig. 15 b)), in terms of accuracy, several of the classifiers used performed well, such as the KNNC and LOGLC. The high AUC and sensitivity values in Fig. 14 support these findings; KNNC has an accuracy of 98.75% and sensitivity of 80%, whilst LOGLC's accuracy is 95.34%, and sensitivity was 100%. As it can be seen in Fig. 15, PARZENC performed the worst within both dataset and achieved 83.33% accuracy in dataset 1 and 68.05% accuracy in dataset 2.



a)



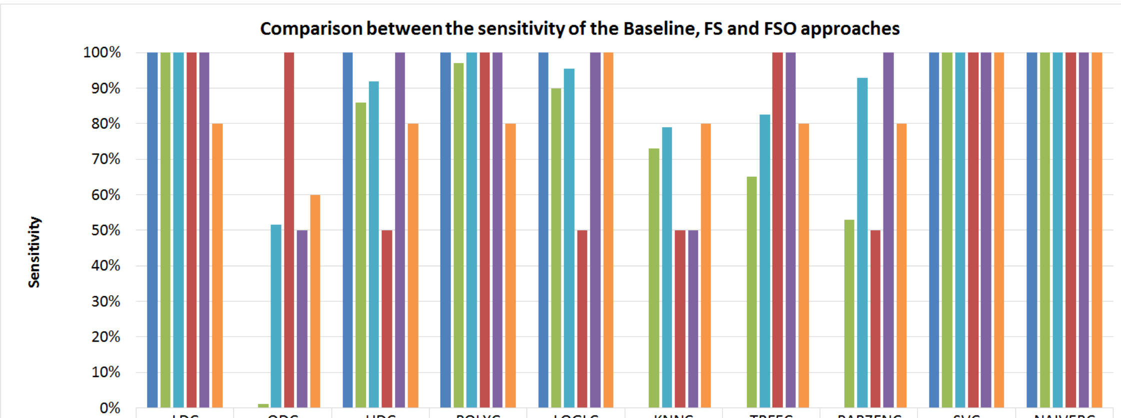
b)

Fig. 15. a) Receiver Operator Curve (ROC) for the oversampled data in dataset 1 and (b) ROC Curve for the oversampled dataset 2

However, compared to Fig. 13, Fig. 15 illustrates a significant improvement, for most of the classifiers. In comparison to dataset 1, KNNC's accuracy has improved dramatically by 14.38%, whilst PARZENC's has improved by 13.93%. QDC's sensitivity has improved by 50%, whilst PARZENC's has improved by 40%. Within dataset 2, PARZENC's accuracy has improved by 7.95%, whilst UDC's has improved marginally by 4.17%. The results indicate that the use of supervised machine learning techniques is encouraging. As demonstrated, these machine learning algorithms are able to learn about our activities, with a high degree of accuracy.

4.4 Comparison of Classification Results

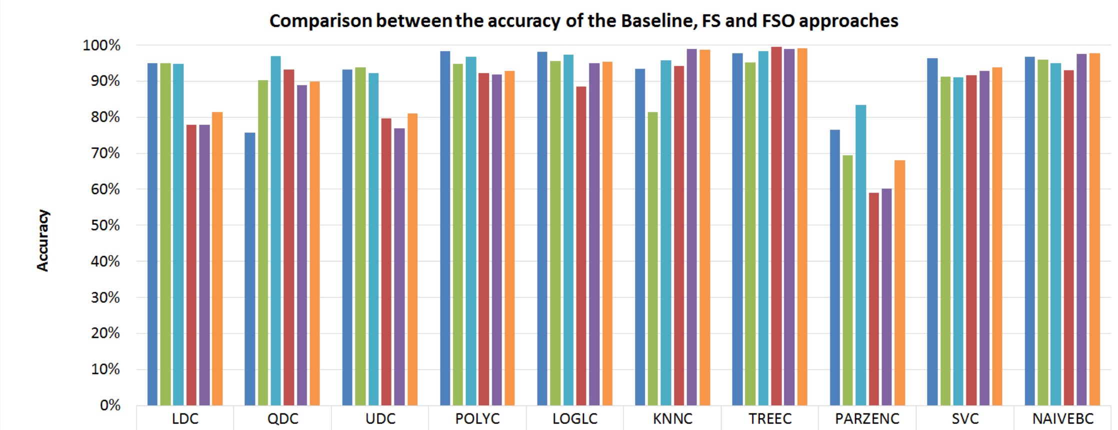
A comparison between the baseline approach, FS method and FSO methods has also been conducted. These results are encouraging and illustrate that selecting the most relevant features and oversampling the dataset has resulted in an improvement in the results for most of the algorithms. For instance, as depicted in Fig. 16 KNNC, TREEC and PARZENC's sensitivity has shown a significant improvement for the FSO method, as opposed to the other approaches.



	LDC	QDC	UDC	POLYC	LOGLC	KNNC	TREEC	PARZENC	SVC	NAIVEBC
Baseline (DS1)	1.0000	0.0000	1.0000	1.0000	1.0000	0.0000	0.0000	0.0000	1.0000	1.0000
FS (DS 1)	1.0000	0.0100	0.8600	0.9700	0.9000	0.7300	0.6500	0.5300	1.0000	1.0000
FSO (DS 2)	1.0000	0.5150	0.9200	1.0000	0.9550	0.7900	0.8250	0.9300	1.0000	1.0000
Baseline (DS2)	1.0000	1.0000	0.5000	1.0000	0.5000	0.5000	1.0000	0.5000	1.0000	1.0000
FS (DS 2)	1.0000	0.5000	1.0000	1.0000	1.0000	0.5000	1.0000	1.0000	1.0000	1.0000
FSO (DS 2)	0.8000	0.6000	0.8000	0.8000	1.0000	0.8000	0.8000	0.8000	1.0000	1.0000

Fig. 16. Comparison between the sensitivity for the Baseline, FS method and FSO method approaches

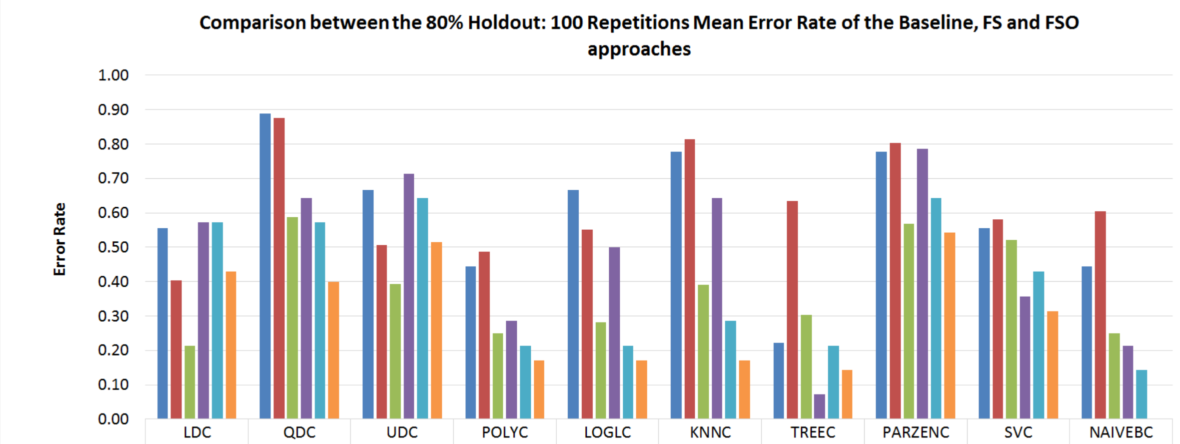
Fig. 17 illustrates a comparison of the accuracy (AUC) results between the baseline, FS method and FSO approaches. As it can be seen, oversampling the dataset has improved these results over the majority of classifiers across both dataset. For instance, as depicted in Fig. 17, QDC, KNNC, TREEC and PARZENC have shown great improvement compared to the FS method and the baseline approach for dataset 1. Similarly for dataset 2, LDC, UDC, POLYC and PARZENC have produced improved results with the FSO method.



	LDC	QDC	UDC	POLYC	LOGLC	KNNC	TREEC	PARZENC	SVC	NAIVEBC
Baseline (DS1)	94.90%	75.71%	93.23%	98.36%	98.16%	93.49%	97.79%	76.50%	96.30%	96.85%
FS (DS 1)	94.97%	90.18%	93.84%	94.89%	95.59%	81.50%	95.15%	69.40%	91.23%	95.98%
FSO (DS 2)	94.80%	97.03%	92.32%	96.67%	97.41%	95.88%	98.40%	83.33%	91.04%	94.90%
Baseline (DS2)	77.93%	93.21%	79.73%	92.25%	88.50%	94.19%	99.56%	59.00%	91.67%	92.95%
FS (DS 2)	77.85%	88.86%	76.81%	91.93%	94.97%	98.93%	98.91%	60.11%	92.83%	97.52%
FSO (DS 2)	81.35%	89.82%	80.98%	92.91%	95.34%	98.75%	99.11%	68.05%	93.73%	97.79%

Fig. 17. Comparison between the accuracy (AUC) for the Baseline, FS method and FSO approaches

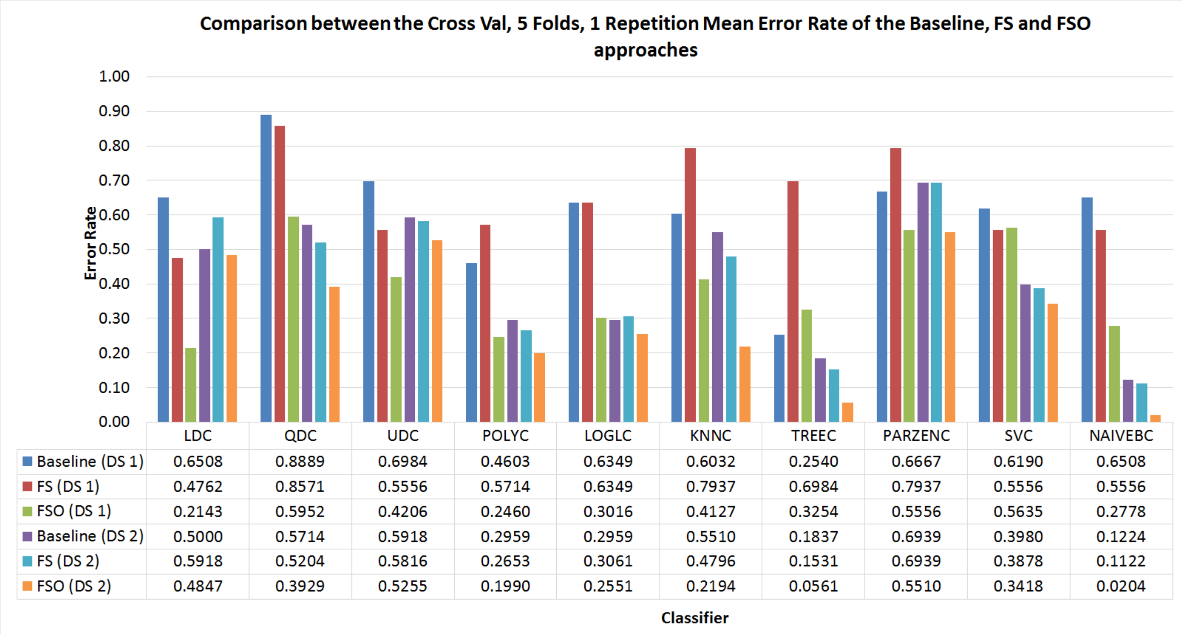
Fig. 18 illustrates a comparison between the mean error rates of the 80% Holdout: 100 Repetition experiment between the baseline, FS method and FSO approaches. As it can be seen, oversampling the dataset (FSO method) has improved the mean error rates across both datasets for the majority of classifiers. In particular, KNNC, QDC and NAIVBC have shown a significant reduction in the errors across both datasets.



	LDC	QDC	UDC	POLYC	LOGLC	KNNC	TREEC	PARZENC	SVC	NAIVEBC
Baseline (DS 1)	0.5556	0.8889	0.6667	0.4444	0.6667	0.7778	0.2222	0.7778	0.5556	0.4444
FS (DS 1)	0.4044	0.8767	0.5056	0.4878	0.5511	0.8133	0.6333	0.8033	0.5811	0.6044
FSO (DS 1)	0.2133	0.5883	0.3933	0.2506	0.2822	0.3906	0.3039	0.5689	0.5217	0.2489
Baseline (DS 2)	0.5714	0.6429	0.7143	0.2857	0.5000	0.6429	0.0714	0.7857	0.3571	0.2143
FS (DS 2)	0.5714	0.5714	0.6429	0.2143	0.2143	0.2857	0.2143	0.6429	0.4286	0.1429
FSO (DS 2)	0.4286	0.4000	0.5143	0.1714	0.1714	0.1714	0.1429	0.5429	0.3143	0.0000

Fig. 18. Comparison between the 80% Holdout: 100 Repetitions Mean Error Rate for the baseline, FS method and FSO approaches

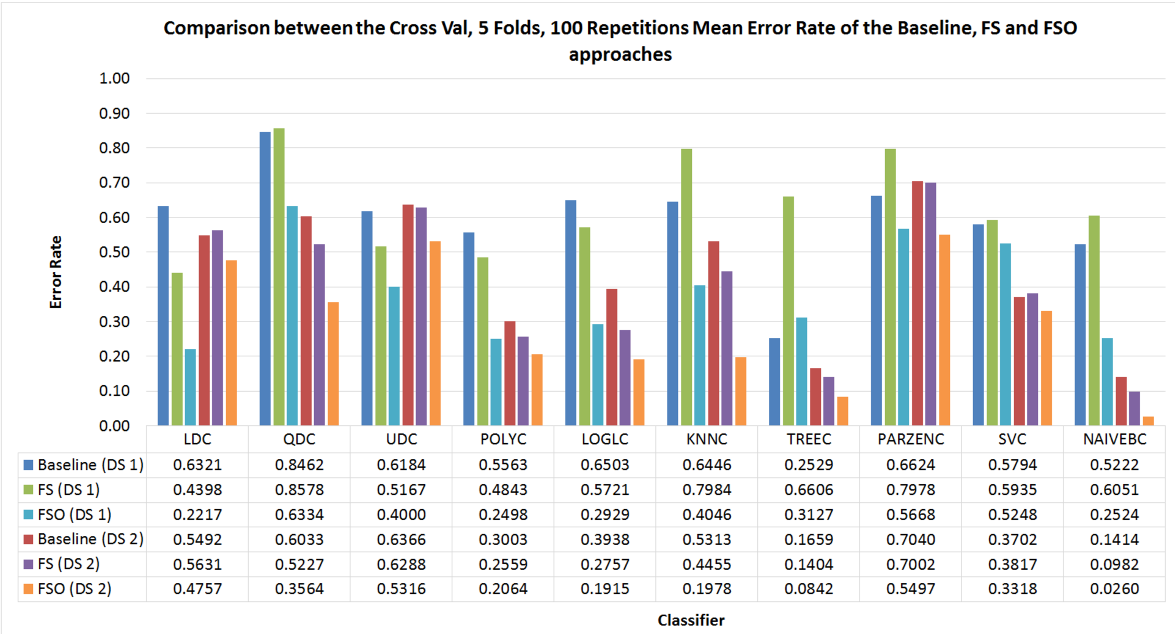
Fig. 19 illustrates a comparison between the mean error rates of the Cross Val, 5 Folds, 1 Repetition experiment of the original and oversampled datasets. As it can be seen, oversampling the dataset has also improved the mean error rates of this experiment, as these have now decreased amongst most of the classifiers. In particular, KNNC, LOGLC, TREEC and PARZENC have shown a significant reduction in the errors.



	LDC	QDC	UDC	POLYC	LOGLC	KNNC	TREEC	PARZENC	SVC	NAIVEBC
Baseline (DS 1)	0.6508	0.8889	0.6984	0.4603	0.6349	0.6032	0.2540	0.6667	0.6190	0.6508
FS (DS 1)	0.4762	0.8571	0.5556	0.5714	0.6349	0.7937	0.6984	0.7937	0.5556	0.5556
FSO (DS 1)	0.2143	0.5952	0.4206	0.2460	0.3016	0.4127	0.3254	0.5556	0.5635	0.2778
Baseline (DS 2)	0.5000	0.5714	0.5918	0.2959	0.2959	0.5510	0.1837	0.6939	0.3980	0.1224
FS (DS 2)	0.5918	0.5204	0.5816	0.2653	0.3061	0.4796	0.1531	0.6939	0.3878	0.1122
FSO (DS 2)	0.4847	0.3929	0.5255	0.1990	0.2551	0.2194	0.0561	0.5510	0.3418	0.0204

Fig. 19. Comparison between the Cross Val, 5 Folds, 1 Repetition Mean Error Rate for the baseline, FS method and FSO approaches

Fig. 20 illustrates a comparison between the mean error rates of the Cross Val, 5 Folds, 100 Repetition experiment of the original and oversampled datasets. As it can be seen, the results have once again improved, as the error rates have decreased amongst all the classifiers. In particular, POLYC, LOGLC, TREEC and KNNC have shown a significant reduction in the errors.



	LDC	QDC	UDC	POLYC	LOGLC	KNNC	TREEC	PARZENC	SVC	NAIVEBC
Baseline (DS 1)	0.6321	0.8462	0.6184	0.5563	0.6503	0.6446	0.2529	0.6624	0.5794	0.5222
FS (DS 1)	0.4398	0.8578	0.5167	0.4843	0.5721	0.7984	0.6606	0.7978	0.5935	0.6051
FSO (DS 1)	0.2217	0.6334	0.4000	0.2498	0.2929	0.4046	0.3127	0.5668	0.5248	0.2524
Baseline (DS 2)	0.5492	0.6033	0.6366	0.3003	0.3938	0.5313	0.1659	0.7040	0.3702	0.1414
FS (DS 2)	0.5631	0.5227	0.6288	0.2559	0.2757	0.4455	0.1404	0.7002	0.3817	0.0982
FSO (DS 2)	0.4757	0.3564	0.5316	0.2064	0.1915	0.1978	0.0842	0.5497	0.3318	0.0260

Fig. 20. Comparison between the Cross Val, 5 Folds, 100 Repetitions Mean Error Rate for the baseline, FS method and FSO approaches

The results are encouraging and indicate that reducing the dataset via feature selection and oversampling has improved the algorithms ability to classifying activities. However, this has been expected, as there is more data for the algorithms to learn from and test against.

4.5 Network Cost and Smartwatch Battery Analysis

Table 5 illustrates the total elapsed time that it takes to pre-process and extract features from 30 seconds worth of raw accelerometer data.

Table 5. Total time of pre-processing 30 seconds of raw accelerometer data

	Data pre-processing steps of our approach				
	Apply Butterworth filter & normalise data	**Sliding window**	**Combine axis**	**Feature extraction**	**Total time**
Time/30 secs raw data	0.106094	0.123297	0.019552	0.050056	**0.298999 sec**

Additionally, Fig. 21 illustrates the total time that it takes, per classifier to classify the data, over one repetition. As it can be seen, NAIVEBC is the fastest performing algorithm, across all the classifiers. This algorithm has also performed well within the experiments; achieving accuracies of 92.95% – 97.79% across the three methods. For the majority of classifiers the baseline approach was the slowest. However, this has been expected as there were a number of redundant features in the dataset. The FS method has reduced the classification time for most of the classifiers, whilst the FSO approach has taken longer than the FS method but is still faster than the Baseline method.

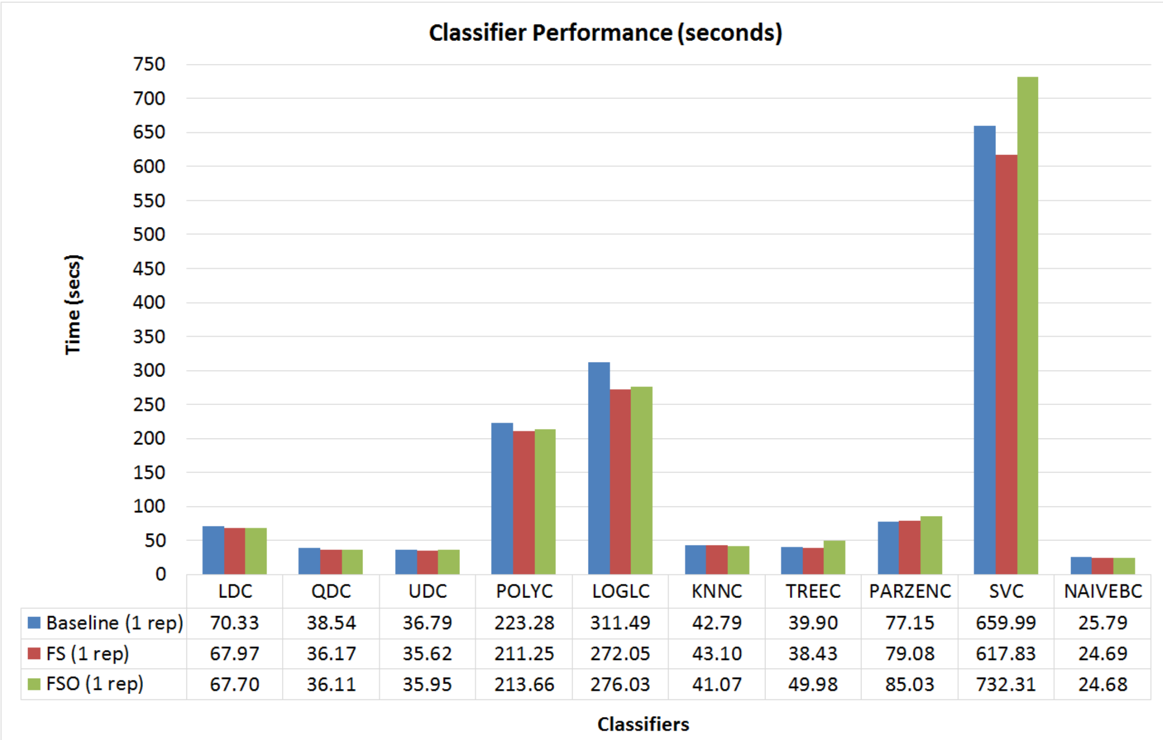
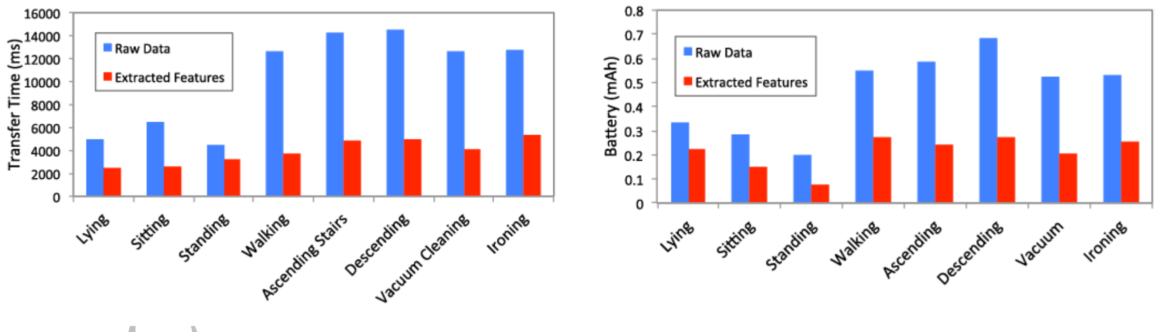


Fig. 21. Total processing time of classifying the data

From this analysis, we can extrapolate that the total time for classifying behaviour should be performed in the cloud. This is due to the fact that small devices, such as wearables, suffer from resource limitations (e.g. limited battery capacity and processing capabilities), which presents major challenges for wearable applications [34]. Consequently, in-line with previous work, complex computational processing, such as classification, has often been performed in the cloud [87]. As a result of these issues, we believe that the process of data analysis should be distributed between both the device(s) and the cloud.

To further analyse this approach, we have used a smartwatch as a sample implementation scenario. As smartwatches continue to be introduced into the market, their computing capabilities have increased in comparison to traditional single use-case wearables, such as body-mounted accelerometers. However, currently the availability of smartwatches and smartphones are dependent. Therefore, to transfer data into the cloud it first needs to be transferred from 1) the watch to the phone, and 2) from the phone to the cloud. Due to this limitation, we believe the cost of transferring data is an important issue for smartwatch applications. This challenge will be highlighted when the system plans to transfer the raw accelerometer data to the cloud. Accelerometer logs are continuously increasing and can grow to become very large datasets. For instance, the accelerometer data that has been analysed in this paper has been collected in three-minute cycles, i.e. approximately 20 MB for 42 minutes of activity. Nevertheless, an ideal system should be able to collect data at all times, which means 24-hour data collection, thus resulting in 685 MB of data per day. Transferring such a huge amount of data is too expensive for a smartwatch, which has a limited battery capability. On the other hand, features that have been identified in section 3.2.3 could be easily computed on the smartwatch, and thus can be used as a substitute for the raw accelerometer data. This is due to the fact that the features that have been extracted are based on simple mathematical methods, such as calculating the average, signal entropy, peak frequency, etc. Although there will be an overhead of calculating these features on the smartwatch, the computational cost is minimal, and so is not worthy for further investigation. Here we report about the differences between the costs of transferring data from the smartwatch to the phone, from two different perspectives: battery and transfer time. Fig. 22 visualizes the daily cost comparison for transferring both types of data, i.e. raw accelerometer data and the features. To achieve the 24 hour costs, we have multiplied 42 minutes to 34.28, as 1440(=minutes of a day) ÷ 42 = 34.28.



a) b)

Fig. 22. Smartwatch to smartphone transfer costs. Figure (a) illustrates the transfer cost in millisecond, whilst figure (b) illustrates the transfer cost in mAh.

As it can be seen in Fig. 22, transferring the extracted features requires less battery power and less time. Therefore, the processing power needed to transfer the features from the smartwatch to the phone is less resource intensive than transferring the raw accelerometer data. However, in-line with similar works, the classification process, which is not resource efficient, will still need to be completed in the cloud [87]. As a result, there is no need to consider the battery depletion on the device for this task. Nevertheless, in-line with this work, computational resources can be conserved by extracting the aforementioned features on the smartwatch and transferring these to the phone, which can then be uploaded to the cloud for further processing.

Such a system has the potential to provide a motivational outlet that can be used to help the user engage in more physical activity. In fully realising this idea, the following hypothetical scenario has been used to illustrate how the system could be utilized for real-life data logging and analysis:

> "Mary is overweight and has never been a particularly physically active person. She leads a predominantly sedentary lifestyle. Her routine consists of sitting at her desk all day at work and then sitting watching TV all night, with no physical exercise. Her weight has been steadily increasing over the past few years and she has started to develop health issues. Although she is not technically obese yet, she is worried that if she continues on this path that she may develop this condition in the future. This has led her to become worried about her weight. Her doctor has also become concerned about her condition and has advised her to become more physically active. She has recently heard of a new technology called lifelogging and thinks that this may help her to become more motivated to engage in physical activity. As a result, she has purchased some wearable sensor node devices so that she can continually record her daily activities. Mary records her daily activities over a period of 16 hours a day, from 7 am to 11 pm, 7 days a week (essentially, from when she wakes up to when she goes to sleep). During this time, when Mary is sitting down for long periods of time, her smart device (e.g. phone/watch) displays a notification that tells her that she has been sitting for x hours and suggests going for a walk/run. When she has been active, another notification congratulates her on this achievement. After continually logging her activities, she has noticed that the notifications have made her aware of her physical inactivity and she begins to move around more and continues to log her life, in this way. After continually logging herself for a prolonged period, Mary notices that she has begun to lose weight and has become more physically active. Her doctor also confirms that her health issues have also began to subside and that soon she will no longer be overweight (if she continues on this path). Mary's lifelogging system has helped her to transform her physical activity habits, lose weight and become healthier all without her having to manually enter data into a system."

From this scenario, we can see how the quantification of physical activities can occur ubiquitously and, with our system, intelligent applications can begin to emerge that uses real-time feedback to alter behaviour. We believe that such outlets are important for providing an accurate measure of physical activity and for inciting behavioural changes.

5 Discussion

This paper demonstrates the initial results that have been obtained by applying signal processing and machine learning algorithms to personal physical activity lifelogs to classify behaviour, with the aim of motivating the user to engage in more physical activity, through multivariate visualizations. As it can be seen from the classification results, this approach yields positive and interesting results. Overall, several of the classifiers performed well across both datasets. These include LDC and NAIVEBC with 100% sensitivity across both datasets. Furthermore, LOGLC, TREEC and NAIVEBC have achieved an accuracy of 95.59%, 95.15% and 95.98% respectively within dataset 1 and KNNC, TREEC and NAIVBC achieving accuracies of 98.93%, 98.91% and 97.52% respectively within dataset 2. Additionally, the SMOTE [74] technique has been used to oversample the dataset, which has enabled the classification results to be improved. This is due to the availability of more data as the accuracy is dependent on "the number of instances, attributes, and classes to be predicted" [76]. Using this techniques has significantly improved the accuracies of most of the classifiers. In particular, within dataset 1, QDC, LOGLC and TREEC have improved with accuracies of 97.03%, 97.41% and 98.40%, respectively.

As indicated in Table 6, our work shows considerable improvements over the current state-of-the-art. The results that we have produced are promising and provide a valid method of classifying behaviour. The novelties that we have provided include:

1) Providing a methodology for processing raw lifelogging data so that features can be extracted/selected

2) Providing a comparison between ten supervised learning classifiers and three approaches to determine their accuracy in detecting human activity

3) Exploring system performance issues, in terms of battery and transfer time, between the costs of transferring data from the smartwatch to the phone

4) An integration that enables the methodology to be employed on a smartwatch.

Table 6. Comparison between our work and previous studies on activity recognition

Reference	Devices used	Number of activities classified	Feature selection	Data analysis technique	Accuracy achieved
Our work	Four accelerometers and a heart rate monitor	16	Yes	1. Linear Discriminant Classifier (LDC) 2. Quadratic Discriminant Classifier (QDC) 3. Uncorrelated Normal Density Based Classifier (UDC) 4. Polynomial Classifier (POLYC) 5. Logistic Classifier (LOGLC) 6. K-Nearest Neighbour (KNNC) 7. Decision Tree (TREEC) 8. Parzen Classifier (PARZENC) 9. Support Vector Classifier (SVC) 10. Naive Bayes Classifier (NAIVEBC)	up to 99%
Lee et al. [18]	Single accelerometer	6	No	1. Artificial neural networks (ANNs)	94.43% – 96.61%
Reiss and Stricker [27]	Three accelerometers and a heart rate monitor	18	No	1. Decision tree (C4.5) 2. Boosted C4.5 decision tree 3. Bagging C4.5 decision tree 4. Naive Bayes 5. kNN	~90%
Casale et al. [28]	Single accelerometer	5	No	1. AdaBoost 2. Bagging	72% – 97%.
Reiss and Stricker [88]	Three accelerometers and a heart rate monitor	14	No	1. Decision trees 2. K- nearest neighbours 3. SVM 4. Naive Bayes 5. Bagging 6. Boosting	90.65% – 94.37%

Our classification results are an improvement over Lee et al. [18], who reported modest accuracies of 94.43% and 96.61%. However, our work has achieved accuracies of up to 99%. In their work, a single tri-axial accelerometer has been used to recognize human activity, whilst artificial neural networks (ANNs) have been trained on all features to recognise the activities [18]. This is in contrast to our work, where more data is obtained with the use of three accelerometers and a heart-rate sensor. By collecting data from multiple devices ensures that a more accurate depiction of the activity occurs, as there are multiple reference points. Furthermore, dataset 1 uses existing data from the publically available PAMAP2 dataset [27]. However, our work in this paper differs from this existing work in [27] and that of Lee et al. [18], in that we have used feature selection algorithms to reduce the feature set so that only the features with the most discriminative capabilities are fed into the classifiers. This is an improvement as redundant features, which could negatively influence the results, are discarded. The work in [27] does not use feature selection and has achieved accuracies of ~90%. This comparable to our work, which has achieved accuracies of up to 99%. This is also an improvement on the state-of-the-art results of similar activity recognition problems, e.g. [18], [27], [88]. Dataset 2 also uses a publically available activity recognition dataset [28]. In this work, Casale et al. [28], do not use feature selection and they have achieved accuracies of 72% – 97%. However, our approach has achieved accuracies of up to 99%. Additionally, the use of our statistical classifiers, in particular SVC, have been shown to produce the best accuracy results within the field of activity recognition [59], [76]. This is in contrast to the ANN approach of [18], which is primarily used within fall detection [76]. These results are promising and provide a valid method of classifying behaviour.

In terms of the visualizations, this aspect demonstrates how intelligent applications can now begin to be developed that are able to utilize machine-learning algorithms to learn about our data and then visualize this information to the user through smartwatches. By correlating and displaying this data to the user, it is more accurate than recalling this information and the user can visually reflect on their levels of physical activity. For instance, using self-reporting methods such as diaries and memory recall, we may think that we are quite physically active. However, when measuring physical activity, these approaches are subject to bias and the fallibility of human memory [89]. These methods are usually unreliable as it is unclear if self-reporting has recorded accurately the events, a distorted memory of the events, an approximation of typical behaviour or a perception of what is considered ideal; i.e. what the user wanted to happen [89]. Nevertheless, the use of technology and such visualizations eliminates this element of uncertainty. Utilizing our approach, a user would be able to reflect upon their levels of activity and if their data indicates that they are physically inactive then this could be the catalyst to alter their behaviour.

These results are very encouraging, demonstrate the validity of our approach and support the use of machine learning in analysing personal lifelogging data. Classification is useful for analysing this information so that such systems can learn about our lives, which could then be used to predict behaviour. This application has the potential to influence the behaviour of individuals by providing an approach that learns about the user through their collected data to predict periods of physical inactivity. This information has then been visually depicted through smartwatch interfaces to provide sufficient motivation to potentially alter behaviour. As it can be seen, logging information about ourselves can be done quite easily with the use of wearable devices and sensors. Over time, such systems are able to learn about our behaviour patterns and could be used to predict when an upcoming period of inactivity will be occurring so that countermeasures can be put in place to alter this behaviour before it occurs and gets unmanageable. In this instance, a prompt can be used to remind/indict our activity levels. Without user intervention, we are able to see our patterns of behaviour. This can be used to reduce obesity levels and to encourage users to leader a healthier lifestyle.

6 Conclusions and Future Work

Undertaking regular physical activity and adopting a healthier lifestyle has been proven to aid in the prevention of obesity and is particularly important for staying active, which would allow us to live longer and more independently into old age. In tackling this, strong evidence is emerging that focuses on the use of technology for implementing positive behavioural changes [29]. As technology develops, wearable devices are becoming smaller and cheaper sensors are being integrated into everyday items. As a result, this widens the opportunities to collect more human-centric information and presents us with new and innovative ways of using this information to increase activity.

This paper has posited our method of classifying and visualising lifelogging data. As a result, the approach enables the system to learn about our behaviours so that the activity classification results can be visualized to the user. By presenting the user with reliable evidence of their activity levels serves as a motivational tool to encourage positive changes. In this sense, the system is able to learn about the user. In achieving this, our approach has been used to collect, process and visualise raw lifelogging data. Using statistically significant methods, features have then been extracted and analysed. The machine learning algorithms that have been chosen for the evaluation have also yielded positive results with great accuracy in detecting physical activity and our initial implementation of a smartwatch interface to visualize this data has also been discussed. Our results have demonstrated the validity of the approach and have produced excellent results in demonstrating the system's accuracy and ability to recognize activity. Overall, we have provided a flexible solution to collect, process and visualise lifelogging data that can be used towards influencing positive behavioural changes. However, further work is required. Future work would consider implementing the system across a focus group of users for user acceptance testing and examine unsupervised methods, whereby untrained data can be fed into the system. In this way, we can ensure the practicality of the framework in the real world.

Acknowledgements

The authors would like to thank the editors and anonymous reviewers for their time, effort and very helpful suggestions that have contributed to improving the quality of this paper.

References

[1] D. E. R. Warburton, C. W. Nicol, and S. S. D. Bredin, "Health benefits of physical activity: the evidence," Can. Med. Assoc. J., vol. 174, no. 6, pp. 801–809, Mar. 2006.

[2] World Health Organization, "Global recommendations on physical activity for health," World Health Organization, 2010.

[3] Department of Economic and Social Affairs Population Division (United Nations), "World Population Ageing 2013," 2013.

[4] W. H. Organization, "World Health Statistics 2014," 2014.

[5] World Health Organization, "Physical activity," Physical activity: Fact sheet N°385, 2015. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs385/en/. [Accessed: 23-Mar-2015].

[6] The McKinsey Global Institute (MGI), "Overcoming obesity : An initial economic analysis," 2014.

[7] The International Longevity Centre UK, "The Economic Value of Healthy Ageing and Working Longer Notes based on the ILC-UK and Actuarial Profession joint debates Supported by Prudential," 2010.

[8] B. Butland, S. Jebb, P. Kopelman, K. McPherson, S. Thomas, J. Mardell, and V. Parry, "Foresight: Tackling obesities—Future Choices," 2007.

[9] World Health Organization, "Obesity and overweight: Fact sheet N°311," 2015. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs311/en/. [Accessed: 24-Mar-2015].

[10] M. S. Tremblay, R. C. Colley, T. J. Saunders, G. N. Healy, and N. Owen, "Physiological and health implications of a sedentary lifestyle.," Appl. Physiol. Nutr. Metab., vol. 35, no. 6, pp. 725–740, Dec. 2010.

[11] D. T. Villareal, C. M. Apovian, R. F. Kushner, and S. Klein, "Obesity in older adults: technical review and position statement of the American Society for Nutrition and NAASO, The Obesity Society," Am. J. Clin. Nutr., vol. 82, no. 5, pp. 923–934, Nov. 2005.

[12] S. J. H. Biddle, N. Pearson, G. M. Ross, and R. Braithwaite, "Tracking of Sedentary Behaviours of Young People: A Systematic Review," Prev. Med. (Baltim)., vol. 51, no. 5, pp. 345–51, Nov. 2010.

[13] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018." pp. 1–40, 2014.

[14] D. Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," 2011.

[15] D. Vergados, A. Alevizos, A. Mariolis, and M. Caragiozidis, "Intelligent Services for Assisting Independent Living of Elderly People at Home," in Proceedings of the 1st international conference on *PErvasive Technologies Related to Assistive Environments (PETRA'08)*, 2008.

[16] S. Whittaker, O. Bergman, and P. Clough, "Easy On That Trigger Dad: A Study of Long Term Family Photo Retrieval," Pers. Ubiquitous Comput., vol. 14, no. 1, pp. 31–43, Mar. 2009.

[17] M. Dodge and R. Kitchin, "'Outlines of a world coming into existence': pervasive computing and the ethics of forgetting," Environ. Plan. B Plan. Des., vol. 34, no. 3, pp. 431–445, 2007.

[18] M. W. Lee, A. M. Khan, and T.-S. Kim, "A single tri-axial accelerometer-based real-time personal life log system capable of human activity recognition and exercise information generation," Pers. Ubiquitous Comput., vol. 15, no. 8, pp. 887–898, Jun. 2011.

[19] S. H. Fairclough, "Fundamentals of physiological computing," Interact. Comput., vol. 21, no. 1–2, pp. 133–145, Jan. 2009.

[20] P. Wang and A. F. Smeaton, "Using visual lifelogs to automatically characterize everyday activities," Inf. Sci. (Ny)., vol. 230, pp. 147–161, May 2013.

[21] L. Kelly and G. J. F. Jones, "Venturing into the labyrinth: the information retrieval challenge of human digital memories," in Workshop on Supporting Human Memory with Interactive Systems, Lancaster, UK, 2007, pp. 37–40.

[22] A. BakhshandehAbkenar, S. W. Loke, and W. Rahayu, "A Framework for Continuous Group Activity Recognition Using Mobile Devices: Concept and Experimentation," in 2014 IEEE 15th International Conference on Mobile Data Management (MDM), 2014, pp. 23–26.

[23] D. Choujaa and N. Dulay, "Activity Recognition from Mobile Phone Data: State of the Art, Prospects and Open Problems," Imp. Coll. London, pp. 1–32, 2009.

[24] A. Fleury, M. Vacher, and N. Noury, "SVM-Based Multimodal Classification of Activities of Daily Living In Health Smart Homes: Sensors, Algorithms, and First Experimental Results," IEEE Trans. Inf. Technol. Biomed., vol. 14, no. 2, pp. 274–283, Mar. 2010.

[25] Z. Lin, A. R. Hanson, L. J. Osterweil, and A. Wise, “Precise Process Definitions for Activities of Daily Living: A Basis for Real-Time Monitoring and Hazard Detection,” in Proceeding of the 3rd Workshop on *Software Engineering in Health Care (SEHC ’11)*, 2011, pp. 13–16.

[26] A. Iosifidis, E. Marami, A. Tefas, I. K. Pitas, and K. Lyroudia, “The MOBISERV-AIIA Eating and Drinking multi-view database for vision-based assisted living,” J. Inf. Hiding Multimed. Signal Process., vol. 6, no. 2, pp. 254–273, 2015.

[27] A. Reiss and D. Stricker, “Creating and Benchmarking a New Dataset for Physical Activity Monitoring,” in The 5th Workshop on Affect and Behaviour Related Assistance (ABRA), 2012.

[28] P. Casale, O. Pujol, and P. Radeva, “Personalization and user verification in wearable systems using biometric walking patterns,” Pers. Ubiquitous Comput., vol. 16, no. 5, pp. 563–580, Jul. 2011.

[29] I. Li, A. K. Dey, and J. Forlizzi, “Using Context to Reveal Factors that Affect Physical Activity,” ACM Trans. Comput. Interact., vol. 19, no. 1, pp. 1–21, Mar. 2012.

[30] A. L. Allen, “Dredging Up the Past: Lifelogging, Memory and Surveillance,” Univ. Chicago Law Rev., vol. 75, pp. 47–74, Sep. 2008.

[31] A. Sheth, “Computing for human experience: Semantics-empowered sensors, services, and social computing on the ubiquitous Web,” IEEE Internet Comput., vol. 14, no. 1, pp. 88–91, Jan. 2010.

[32] E. K. Choe, N. B. Lee, B. Lee, W. Pratt, and J. A. Kientz, “Understanding Quantified-Selfers’ Practices in Collecting and Exploring Personal Data,” in Proceedings of the 32nd Annual ACM Conference on *Human Factors in Computing Systems (CHI ’14)*, 2014, pp. 1143–1152.

[33] H. Banaee, M. U. Ahmed, and A. Loutfi, “Data Mining for Wearable Sensors in Health Monitoring Systems: A Review of Recent Trends and Challenges,” Sensors, vol. 13, no. 12, pp. 17472–17500, Jan. 2013.

[34] R. Rawassizadeh, B. A. Price, and M. Petre, “Wearables: Has the Age of Smartwatches Finally Arrived?,” Commun. ACM, vol. 58, no. 1, pp. 45–47, 2014.

[35] R. Rawassizadeh, M. Tomitsch, K. Wac, and A. M. Tjoa, “UbiqLog: a generic mobile phone-based life-log framework,” Pers. Ubiquitous Comput., vol. 17, no. 4, pp. 621–637, Apr. 2012.

[36] C.-K. Hsieh, H. Tangmunarunkit, F. Alquaddoomi, J. Jenkins, J. Kang, C. Ketcham, B. Longstaff, J. Selsky, B. Dawson, D. Swendeman, D. Estrin, and N. Ramanathan, “Lifestreams: a modular sense-making toolset for identifying important patterns from everyday life,” in Proceedings of the 11th ACM Conference *on Embedded Networked Sensor Systems (SenSys ’13)*, 2013, pp. 1–13.

[37] S. Nath, “Ace: Exploiting Correlation for Energy-Efficient and Continuous Context Sensing,” in Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services *(MobiSys ’12)*, 2012, pp. 29–42.

[38] V. Srinivasan, S. Moghaddam, A. Mukherji, K. K. Rachuri, C. Xu, and E. M. Tapia, “MobileMiner: Mining Your Frequent Patterns on Your Phone,” in Proceedings of the 2014 ACM International Joint *Conference on Pervasive and Ubiquitous Computing (UbiComp ’14)*, 2014, pp. 389–400.

[39] S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, S. Park, T. Park, and J. Song, “Seemon: Scalable and Energy-Efficient Context Monitoring Framework for Sensor-Rich Mobile Environments,” in Proceeding of the *6th International Conference on Mobile Systems, Applications, and Services (MobiSys ’08)*, 2008, pp. 267–280.

[40] R. K. Balan, Y. Lee, T. K. Wee, and A. Misra, "The Challenge of Continuous Mobile Context Sensing," in 2014 Sixth International Conference on Communication Systems and Networks (COMSNETS), 2014, pp. 1–8.

[41] J. Cui, G. Sun, and B. Xu, "Demo: AD-Sense: Activity-driven Sensing for Mobile Devices," in Proceedings of the 9th ACM Conference on Embedded Net*worked Sensor Systems (SenSys '11)*, 2011, pp. 399–400.

[42] D. Byrne, A. R. Doherty, C. G. M. Snoek, G. G. F. Jones, and A. F. Smeaton, "Validating the detection of everyday concepts in visual lifelogs," in Semantic Multimedia, 2008, pp. 15–30.

[43] Y.-S. Lee and S.-B. Cho, "Recognizing multi-modal sensor signals using evolutionary learning of dynamic Bayesian networks," Pattern Anal. Appl., Sep. 2012.

[44] Z. Qiu, A. R. Doherty, C. Gurrin, and A. F. Smeaton, "Mining User Activity as a Context Source for Search and Retrieval," in 2011 International Conference on Semantic Technology and Information Retrieval (STAIR), 2011, pp. 162–166.

[45] T. Phan, "Generating Natural-Language Narratives from Activity Recognition with Spurious Classification Pruning," in Proceedings of the Third International Workshop on Sensing Applications on Mobile Phones - *PhoneSense '12*, 2012, pp. 1–5.

[46] B. E. Ainsworth, W. L. Haskell, M. C. Whitt, M. L. Irwin, A. M. Swartz, S. J. Strath, W. L. O'brien, D. R. Bassett, K. H. Schmitz, P. 0. Emplaincourt, D. R. Jacobs, and A. S. Leon, "Compendium of Physical Activities: an update of activity codes and MET intensities," Med. Sci. Sports Exerc., vol. 32, no. 9, pp. S498–S504, 2000.

[47] A. Reiss and D. Stricker, "Towards Global Aerobic Activity Monitoring," in Proceedings of the 4th *International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '11)*, 2011, p. 12.

[48] R. Liu, M. Liu, X. Sun, and Y. Wei, "Signal Processing and Accelerometer-based Design for Portable Small Displacement Measurement Device," in 2008 International Conference on Embedded Software and Systems, 2008, pp. 575–579.

[49] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. P. Cardoso, "Preprocessing Techniques for Context Recognition from Accelerometer Data," Pers. Ubiquitous Comput., vol. 14, no. 7, pp. 645–662, Mar. 2010.

[50] R. E. Mayagoitia, A. V. Nene, and P. H. Veltink, "Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems," J. Biomech., vol. 35, no. 4, pp. 537–542, Apr. 2002.

[51] G. M. Lyons, K. M. Culhane, D. Hilton, P. A. Grace, and D. Lyons, "A description of an accelerometer-based mobility monitoring technique," Med. Eng. Phys., vol. 27, no. 6, pp. 497–504, Jul. 2005.

[52] Y. Zhang, I. Sapir, S. Markovic, R. Wagenaar, and T. Little, "Continuous Functional Activity Monitoring Based on Wearable Tri-axial Accelerometer and Gyroscope," in Proceedings of the 2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011, pp. 370–373.

[53] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," Pervasive Comput., vol. 3001, pp. 1–17, 2004.

[54] W. Song, C. Ade, R. Broxterman, T. Barstow, T. Nelson, and S. Warren, "Activity Recognition in

Planetary Navigation Field Tests Using Classification Algorithms Applied to Accelerometer Data," in 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2012, vol. 2012, pp. 1586–1589.

[55] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity Recognition from Accelerometer Data," Proc. Natl. Conf. Artif. Intell., vol. 20, no. 3, pp. 1541–1546, 2005.

[56] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers.," Sensors, vol. 10, no. 2, pp. 1154–1175, Jan. 2010.

[57] N. C. Krishnan and S. Panchanathan, "Analysis of low resolution accelerometer data for continuous human activity recognition," in 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 3337–3340.

[58] N. Mamizuka, M. Sakane, K. Kaneoka, N. Hori, and N. Ochiai, "Kinematic quantitation of the patellar tendon reflex using a tri-axial accelerometer.," J. Biomech., vol. 40, no. 9, pp. 2107–2111, Jan. 2007.

[59] M.-C. Yu, T. Yu, S.-C. Wang, C.-J. Lin, and E. Y. Chang, "Big Data Small Footprint: The Design of A Low-Power Classifier for Detecting Transportation Modes," in Proceedings of the 40th International Conference on Very Large Data Bases (VLDB) Endowment 7, 2014, pp. 1429–1440.

[60] MathWorks, "Practical Introduction to Frequency-Domain Analysis," 2014. [Online]. Available: http://www.mathworks.co.uk/help/signal/examples/practical-introduction-to-frequency-domain-analysis.html. [Accessed: 03-Sep-2014].

[61] W. L. Maner, R. E. Garfield, H. Maul, G. Olson, and G. Saade, "Predicting term and preterm delivery with transabdominal uterine electromyography.," Obstet. Gynecol., vol. 101, no. 6, pp. 1254–1260, Jun. 2003.

[62] A. G. Bonomi, A. H. C. Goris, B. Yin, and K. R. Westerterp, "Detection of Type, Duration, and Intensity of Physical Activity Using an Accelerometer," Med. Sci. Sports Exerc., vol. 41, no. 9, pp. 1770–7, Sep. 2009.

[63] P. Srivastava and W.-C. Wong, "Hierarchical Human Activity Recognition Using GMM," in AMBIENT 2012: The Second International Conference on Ambient Computing, Applications, Services and Technologies, 2012, no. c, pp. 32–37.

[64] F. Mokaya, B. Nguyen, C. Kuo, Q. Jacobson, A. Rowe, and P. Zhang, "MARS: A Muscle Activity Recognition System Enabling Self-configuring Musculoskeletal Sensor Networks," in The 12th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN), 2013, pp. 191–202.

[65] X. Long, B. Yin, and R. M. Aarts, "Single-Accelerometer-Based Daily Physical Activity Classification," in Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2009, vol. 2009, pp. 6107–6110.

[66] S. Abdullah, N. D. Lane, and T. Choudhury, "Towards Population Scale Activity Recognition: A Framework for Handling Data Diversity," in Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012, pp. 851–857.

[67] M. Kim and D. Kotz, "Identifying Unusual Days," J. Comput. Sci. Eng., vol. 5, no. 1, pp. 71–84, Mar. 2011.

[68] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," J. Mach. Learn. Res., vol. 3, pp. 1157–1182, 2003.

[69] Matlab, "Selecting Features for Classifying High-dimensional Data," 2013. [Online]. Available: http://www.mathworks.co.uk/products/statistics/examples.html?file=/products/demos/shipping/stats/cvsequentialfsdemo.html#1. [Accessed: 11-Oct-2013].

[70] A.-C. Haury, P. Gestraud, and J.-P. Vert, "The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures," PLoS One, vol. 6, no. 12, p. e28210, Jan. 2011.

[71] W. R. Rice, "Analyzing Tables of Statistical Tests," Evolution (N. Y)., vol. 43, no. 1, pp. 223–225, 1989.

[72] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Editio. Springer Press, 2009.

[73] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. Bowyer, "SMOTEBoost: Improving Prediction of the Minority Class in Boosting," Proc. Princ. Knowl. Discov. Databases, PKDD-2003, pp. 107–119, 2003.

[74] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, no. 1, pp. 321–357, 2002.

[75] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," Adv. Intell. Comput., vol. 3644, no. 1, pp. 878–887, 2005.

[76] M. Habib ur Rehman, C. S. Liew, T. Y. Wah, J. Shuja, and B. Daghighi, "Mining Personal Data Using Smartphones and Wearable Devices: A Survey," Sensors, vol. 15, no. 2, pp. 4430–4469, Jan. 2015.

[77] F. van der Heijden, R. P. W. Duin, D. de Ridder, and D. M. J. Tax, Classification, Parameter Estimation and State Estimation. Wiley, 2005.

[78] A. K. Jain, "Data Clustering: 50 years beyond K-means," Pattern Recognit. Lett., vol. 31, no. 8, pp. 651–666, Jun. 2010.

[79] P. S. Adhvaryn and M. Panchal, "A Review on Diverse Ensemble Methods for Classification," IOSR J. Comput. Eng., vol. 1, no. 4, pp. 27–32, 2012.

[80] F. van der Heijden, R. P. W. Duin, D. de Ridder, and D. M. J. Tax, Classification, Parameter Estimation and State Estimation: An Engineering Approach using Matlab. Wiley, 2004.

[81] P. Adamopoulos and A. Tuzhilin, "Recommendation Opportunities: Improving Item Prediction UsingWeighted Percentile Methods in Collaborative Filtering Systems," in Proceedings of the 7th ACM conference on Recommender systems - *RecSys '13*, 2013, pp. 351–354.

[82] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Third. Pearson Education, 2010.

[83] N. J. Salkind, Statistics for people who (think they) hate statistics, 3rd ed. Sage Publications, 2008.

[84] T. Fawcett, "An introduction to ROC analysis," Pattern Recognit. Lett., vol. 27, no. 8, pp. 861–874, Jun. 2006.

[85] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Inf. Process. Manag., vol. 45, no. 4, pp. 427–437, Jul. 2009.

[86] L.-I. Tong, Y.-C. Chang, and S.-H. Lin, "Determining the optimal re-sampling strategy for a classification model with imbalanced data using design of experiments and response surface methodologies," Expert Syst. Appl., vol. 38, no. 4, pp. 4222–4227, Apr. 2011.

[87] R. Rawassizadeh, M. Tomitsch, M. Nourizadeh, E. Momeni, A. Peery, L. Ulanova, and M. Pazzani, "Energy-Efficient Integration of Continuous Context Sensing and Prediction into Smartwatches," Sensors, vol. 15, no. 9, pp. 22616–22645, Sep. 2015.

[88] A. Reiss and D. Stricker, "Introducing a Modular Activity Monitoring System," in Annual International

Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2011, vol. 2011, pp. 5621–4.

[89] P. Kelly, A. Doherty, E. Berry, S. Hodges, A. M. Batterham, and C. Foster, “Can We Use Digital Life-Log Images To Investigate Active And Sedentary Travel Behaviour? Results From A Pilot Study,” Int. J. Behav. Nutr. Phys. Act., vol. 8, no. 1, p. 44, Jan. 2011.