Machine Learning for Automated Exercise Monitoring from a Wearable Network of Inertial Sensors

Philippe Miranthis

189181770

MEng (Hons) Integrated Mechanical and Electrical Engineering with year-long work placement

'MEng Final Project Report'

2022/2023

Supervised by: Dr Alan Hunter

Assessed by: Prof. Manuch Soleimani

Number of words: 11860

Abstract/executive summary

Several models were developed to attempt to classify barbell exercises, segment individual reps and recognize the quality of the reps from data collected using bespoke wireless inertial measurement units measuring accelerometer and gyroscopic data attached to various locations on the body. These sensors were provided by Shield Performance Products an external company who commissioned this project. Data was collected from participants wearing the sensors in set configurations on the body performing squats, cleans, or deadlifts all using a barbell. An app on a smartphone was used to record the data and upload it to a Google Firebase server. The collected data was then pre-processed, labelled with the start of each rep and fed into convolutional neural networks (CNNs) for segmentation and classification using the sliding window technique across the each of the recordings in the dataset. This method allowed classification and segmentation to be achieved using only a single CNN. Initially a squat classifier using a specific configuration of sensors was developed with a binary output at the CNN achieving a balanced accuracy of 79% but was trained and tested on a small dataset with missing data, using leave-one-out cross validation. This was extended to two CNNs of different architectures developed to classify and segment all the exercises for a specific sensor configuration. These performed poorly achieving 69% and 62% balanced accuracy but were trained and tested on a small dataset with missing data, using leave-one-out cross validation. An investigation of the effect of sensor placement was carried out. This investigation varied the inputs to the multi exercise CNNs by using data from combination of sensors located at different parts of the body. This produced better results as more data was available for each model then for the specific sensor configuration and did not contain data with missing sensors. The investigation achieved a balanced accuracy of 77% on the best performing sensor combination of the upper arm, chest, and upper leg; these locations were noted as the best for sensor placement. t was found that the models struggled with classification due to varying techniques. Issues with the methods and models used were highlighted as well as differences between results of the project and that of the literature. Improvements to the models and methods as well as avenues for future work were suggested.

Acknowledgements

Thanks is given to Dr Alan Hunter, the academic supervisor for this project, and to Jacob McCarthy the owner of shield performance products who commissioned this project and provided the resources to complete it. Thanks, is also given to the staff members of Fitness First Bath who allowed me to use their facility to conduct my experiment and to Ella Noble, head coach of the University of Bath powerlifting society, who was incredibly helpful in getting participants from within her society and helping me run sessions.

Table of Contents

Table of Abbreviations/Acronyms

Term	Abbreviation/Acronym
Personal Trainer	PT
Repetition(s)	Rep(s)
Shield Performance Products	SPP
Machine Learning	ML
Inertial Measurement Unit	IMU
Human Activity Recognition	HAR
Convolutional Neural Network	CNN
Time Series Classification	TSC
Multivariate Time Series Classification	MTSC
Deep Neural Networks	DNN
Leave-One-Out Cross Validation	LOOCV

1 Introduction

1.1 Project Background and Context

It is no secret that the Fitness industry has experienced a period of rapid growth over the last 10 years [1]. The ever-growing obsession of becoming the fittest version of oneself has driven people to explore new ways to exercise, from workout classes such as CrossFit through to technology-based means such as smartwatches and virtual spinning classes like Peloton.

The market for fitness wearable is primarily focused on activity tracking for running and other similar exercises. These devices primarily track heart rate, distance travelled (by GPS) and steps. Although sufficient for many, these devices lack the ability to accurately track gym exercises such as squats and deadlifts. There have been attempts to produce a wearable capable of this, namely the Moov fitness coach. However, these products have been unsuccessful, resulting in a gap in the market for a product with these capabilities.

With an increasing number of people getting into fitness, there has also been an associated uptick in the number of people becoming personal trainers (PTs) [2]. PTs provide a range of services, from workout and nutrition planning, to exercise technique coaching. Unfortunately, PTs can be expensive, and they are not always around. This means that for beginners there can be a steep learning curve for learning new movements that if done incorrectly can lead to injury.

To address this problem and fill the gap in the market, a previous final year project tried to develop a new fitness wearable that could track the user's workout. It aimed to tell users what exercises they did, how many reps they did and, crucially, how well they performed these reps. The project was incomplete with more work needed in the development of the algorithms underpinning the technology as well as more research into how they can be used. The work was spun off into a start-up called Shield Performance Products (SPP) with funding from the UKRI, acceler8 bath and various other sources. SPP is still in a development and research stage and so is working with the University of Bath to develop the software for these devices. This final year project (FYP) aims to work with the devices produced by SPP to help develop the machine learning (ML) algorithms that achieve the desired capabilities and to provide useful information that will aid in the development of the product.

The devices that have been provided consist of an Arduino microcontroller with an inertial measurement unit (IMU) sensor, a battery, and a power switch. The devices are approx. 60x30x25mm in size and an example is shown in Figure 1.1.

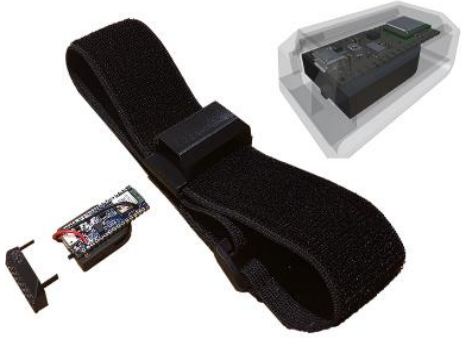


Figure 1.1 Example of one of the sensor modules from SPP including the strap used to attach the module to the body. A 3D model of the device is seen in the top right of the image. This image has been taken from the SPP website.

Each IMU sensor records 3-axis accelerometer and 3-axis gyroscopic data at a sample rate of 35Hz, the sample rate can be changed by reprogramming the sensors. These sensors are attached to the body using Velcro straps that can be adjusted to fit to different parts of the body including forearms, upper arms, chest, abdomen, thighs, lower legs, and the head.

Up to four of these devices can be attached to the user and then connected to a mobile device over a Bluetooth connection, giving a maximum of 24 time series data steams to use for ML development. The mobile device uses an app created by SPP called MyPT that allows users to easily collect exercise data. The underlying principle is that these sensors can detect the motion of a given limb. This motion data can then be used to deduce what exercise was performed and the quality at which it was performed.

The user collects data by recording their exercise, like how one would record a video, by pressing a start button before the exercise and a stop button at the end. The user only performs one exercise from a list of given exercises but can perform as many reps as they want.

1.2 Aims and Research Questions

This project aims to collect exercise data, using the SPP devices provided, from a wide range of participants and develop ML algorithms to classify what exercise a user performed, how many reps they performed and how well they performed it. As well as investigate how varying the usage of these devices can affect the performance of ML algorithms, to provide information on how best to use these devices in the future and uncover useful trends in the data that may help with product development.

To achieve this aim and provide a methodical approach to this problem four key research questions were proposed. Each question builds on one another and helps to ensure that the project produces a presentable piece of work even if not all tiers are achieved. The questions have been structured by considering the critical underlying capabilities of the algorithms and addressing them first. Research questions were formulated after a review of the relevant literature, a consideration of the project aims, as well as a consideration of the time frame for the project. It should be noted, this project does not aim to evaluate the performance of different ML algorithms, rather it looks to apply the most relevant algorithm and adapt its structure to achieve the stated aims. This approach was taken as reviewing the performance of multiple algorithms is time consuming and inhibits the ability to fully analyse the dataset to understand how the model is behaving. The research questions are defined as follows:

1) **Can CNNs be used to segment and classify squat reps for a fixed sensor configuration?**
2) **Can CNNs be used to segment and classify reps of squats, deadlifts, and cleans? Is it better to train a single multi-output network or multiple binary output ones?**
3) **What are the best possible sensor locations for classifying exercises and segmenting reps for squats, deadlifts, and cleans from the data collected?**
4) **Can CNNs be used to recognize the quality of technique for squats, deadlifts, and cleans and what is the best sensor locations for this task?**

As outlined in the research questions, the project will focus solely on back squats, regular deadlifts and power cleans. The future product by SPP will look to classify more exercises as well as different variations of similar exercise (i.e., front squats, overhead squats). However, to ensure sufficient data could be collected for each movement, over the limited project timeframe, only movements that are easily accessible were chosen. More complicated movements such as snatches are often not performed in commercial gyms and so were expected to put people off participating in the study. These movements were also chosen as they are symmetrical about the centre vertical of the body and so reduce some of the complexity that may be introduced with unsymmetrical movements such as lunges.

1.3 Relevant Literature

Inertial sensors like that used by SPP have been widely used in the fields of human activity recognition (HAR) and in sports exercise recognition applications. HAR includes task such as activity classification as well as repetition counting. Much of the work has been centred around using either a smartphone or smartwatch to act as the sensor taking the readings [3] [4] [5] [6] [7] [8] [9], as opposed to having to use additional wearables [10] [11] [12] [13]. There has also been work into using alternative measurement techniques including measuring doppler signals using a smartphone microphone [14], placing RFID tags on dumbbells [15], using radar sensors to measure human movement [16] and wearing textile sensors to measure movement [17].

Much of the research focuses on using ML algorithms to classify the activity being performed. More recently, research has focused on applying deep learning algorithms, particularly convolution neural networks (CNN), to these problems [18] [19] [20] [21]. This move towards CNNs is due to increased data availability, computing power, and the ease of implementation. In addition, there are several packages built around these more recent models that aid to further improve development times. However, a disadvantage of CNNs is their high computational cost which makes them often unsuitable for embedded online low power devices. However, this project does not aim to develop algorithms that will be embedded onto future SPP products. For an overview of deep learning algorithms applied to HAR, see [22]. This project is focused on exercise recognition and so focuses on studies related to this topic area.

Soro et al. [7] used CNN to classify 10 different CrossFit exercises using two smartwatch sensors collecting inertial data (3-axis accelerometer, 3-axis gyroscope), achieving an exercise classification accuracy of 99.96% and a rep counting accuracy of 91% within an error ±1 repetition. The data was collected in a gym using 54 participants, but they artificially remove data that has errors in it due to errors in the smartwatches during the exercise reducing the realness of the data. This paper provides a good amount of evidence to show that even complicated movements can be correctly identified but does not address whether the quality of the exercise can be recognized. In addition, the method in which they achieve their incredibly high classification accuracy is questionable as it is achieved on very controlled data. O'Reilly et al. [23] mention how these technologies may be used to replace the personal trainer to improve form but only go so far to also classify 5 bodyweight exercises (squat, lunge, deadlift, single leg squat and tuck jump). They do however use 5 separate IMU sensors (3-axis accelerometer, 3-axis gyroscope) located on the outside shanks and thighs of the legs as well as the lower lumbar of the spine. This is quite different from many other studies which attempt to limit the number of sensors required to classify the exercise performed. They found that 5 IMUs achieved a 99% classification accuracy and that similar performance of 98% accuracy can be achieved with just a single IMU on the shank. However, they used a decision tree classifier with considerable feature engineering and a somewhat primitive rep segmentation algorithm with a questionable ability to consistently segment the reps in more realistic environments, as opposed to the very controlled lab environment they used. Morris et al. [12] used a SVM classifier to classify periods of up to 13 different types of exercises and then applied a counting algorithm to the segmented exercise regions. The counting algorithm transforms the regions based on exercise type and looks for peaks in the signal to count reps. They used an IMU containing a 3-axis accelerometer and 3-axis gyroscope worn on the right forearm. They achieve good accuracy on their test data with a classification accuracy of 96% on their largest set of exercises and 97% accuracy for counting reps within an error of ±1 rep. This method is not as suitable for the aims of this project as individual reps need to be classified to classify the quality of any given rep.

1.4 Background Theory

Time series classification (TSC) is a very common problem in machine learning and has numerous applications, from power system fault detection [24] to monitoring seismic activity [25]. It consists of identifying patterns in time series data, such as from a temperature sensor taking readings throughout the day. Often information comes from multiple signals needing to be classified simultaneously, leading to the problem multivariate time series classification (MTSC).

Natural language processing and speech recognition are two machine learning fields with a very similar problem nature to that of MTSC due to their temporal/sequential nature. Recent advances of deep neural networks (DNNs) in these two fields have led to an increased use of DNNs in TSC problems [26] [27]. A general deep learning framework for TSC can be seen in Figure 1.2 [26].

DNNs have the advantage over classical ML techniques of requiring very little to no feature engineering to achieve performance on par or better than their competitor. Feature engineering is the process by which new information is extracted from the data by hand by performing transformations on the raw data. This can range from applying a Fourier transform to the raw data to something as simple as calculating the mean of the dataset.

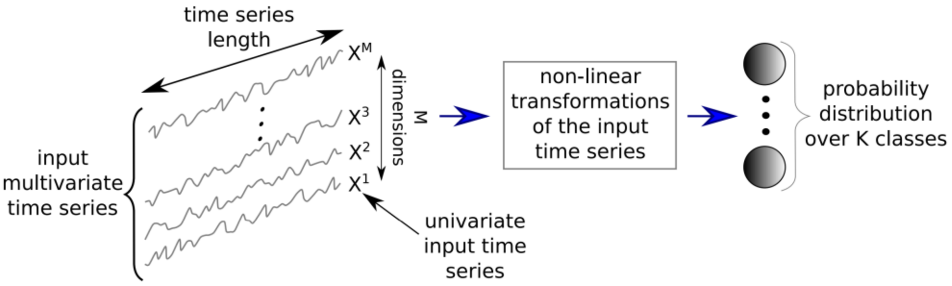

time series length
input multivariate time series
X^M
X^3
X^2
X^1
M dimensions
univariate input time series
non-linear transformations of the input time series
probability distribution over K classes

Figure 1.2 A unified deep learning framework for time series classification, taken from Deep learning for time series classification: a review [26]

One DNN architecture that has seen success in many different applications are CNNs. These work for TSC by convolving a filter over a time series. This is primarily done in 1-D but can be achieved in 2-D for multivariate time series. The values of the filters are normally learned automatically. Applying several filters on a time series will produce a multivariate time series from the initial time series, the dimensions of which are equal to the number of filters used. The convolutional layer is followed by a pooling layer which acts to reduce the length of the input time series by aggregating over a sliding window of the time series. This layer helps to reduce overfitting and reduces the number of trainable parameters in the model. Often multiple layers of convolutions and poolings are used in CNNs. These layers are then followed by a fully connected discriminative classifier. This type of classifier directly learns the mapping from the input time series and outputs the probability distribution of the classes in the dataset. For a CNN this consists of a Neural Network, typically a multi-layer perceptron MLP. Figure 1.3 depicts an architecture for a three convolutional layer CNN.

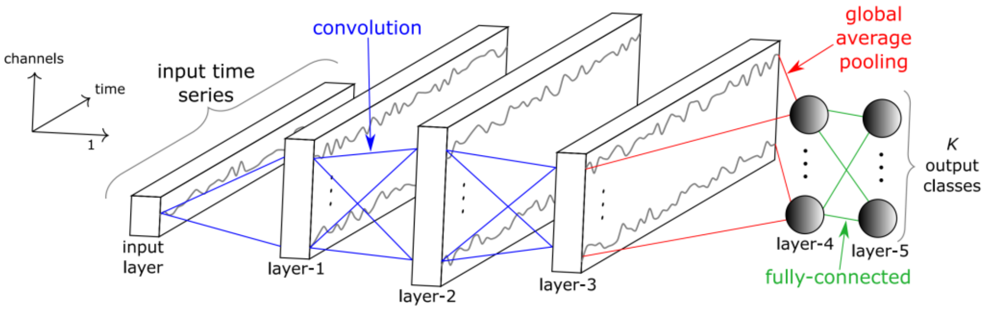


Figure 1.3 A general architecture for a fully convolutional neural network, taken from deep learning for time series classification [26].

CNNs are trained in the same way as MLPs, a feed-forward pass followed by backpropagation. This process is repeated for each training sample, and often re-run over multiple training iterations so called epochs. To help improve the computational efficiency of these models, multiple training samples are fed into the model at once. These groups of training data are called batches and varying the size of these can affect the performance of the model.

An important technique commonly used in time series ML is the sliding window technique, which essentially splits the time series up into smaller 'windows' of data. Windows can be taken at each sample or at a chosen interval for less overlap.

Another important aspect of ML is the analysis of performance of the models and the approach to training. The primary methods are called cross-validation techniques which are techniques for validating the model by splitting the total dataset into training and testing data. There are several cross-validation methods, namely: Holdout, K-fold, Stratified k-fold, Rolling, Monte Carlo, Leave-p-out, and Leave-one-out method. The holdout method is the most common method due to its simplicity only requiring the data to be splitting the data once into testing and training data. However, this method is often unsuitable for small data sets that are prone to significant variation in performance when the test dataset is changed. The other methods attempt to address this problem.

Leave-one-out cross validation (LOOCV) is often used on small datasets. It works by extracting one of samples as the test dataset testing and training on the remaining samples. It then repeats this for each of the samples in the entire dataset. It is important to ensure that the test data and training data do not have values that are correlated.

K-fold cross validation involves splitting the data in *k* groups. One of the groups is then used for the test data and the remainder for the training data. The process is then repeated for each of the groups until each group is used as the test dataset.

There are many metrics used to classify the performance of ML models. The receiver operator characteristic ROC curve is commonly used for binary datasets to measure the performance of the model by varying the model output probability threshold. This allows an insight into how the changing the sensitivity of the model can affect performance. The ROC area under curve (ROC AUC) is a metric used to quantify the ROC curve, a value of 1 is perfect score and indicates that all labels are detected with no false positive labels. Another metric commonly used on unbalanced datasets is the balanced accuracy. This metric is the arithmetic mean of sensitivity and specificity and is commonly used on imbalanced datasets. Confusion matrices are commonly used for models that have multiple classes to identify. These matrices provide a visual representation of how each of the classes is performing by comparing the predicted labels with the actual labels.

2 Methods and Results

2.1 Data Collection Method

Part of the project required exercise data to be collected from a variety of participants. This needed to be carefully planned out to ensure the data collected was in such a way that it helped to achieve the project goals.

As explained in section 1 there are a maximum of four SPP devices that can be connected to the app. A picture of one of the SPP devices and a participant with the devices attached to them can be seen in Figure 2.1.

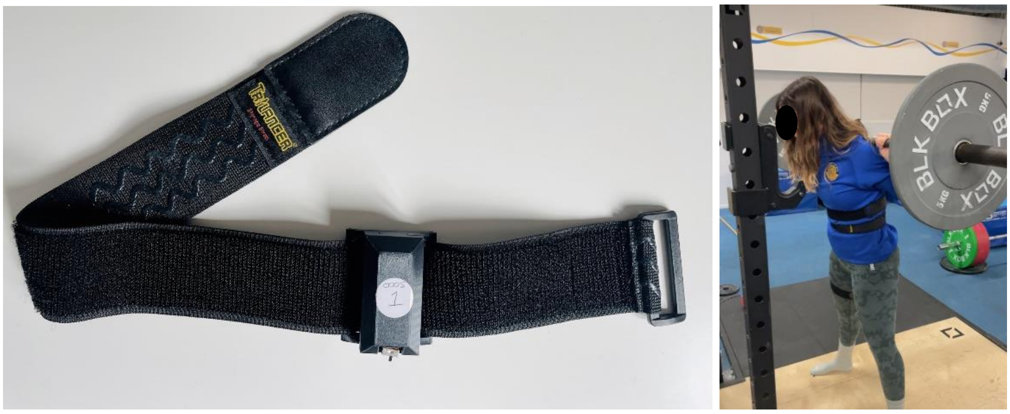
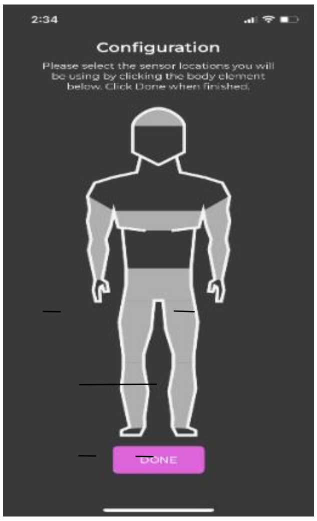


Figure 2.1 One of the SPP devices with strap used to collect data and participant performing a squat with devices attached to them.

When the app is downloaded, you are asked to sign in or create an account with a username and asked to provide some basic personal information such as height, weight, age, and sex. As previously explained, the app works by configuring the sensors to the locations where they have been placed on the body; locations are show in Figure 2.2. The user only performs one exercise from a list of given exercises but can perform as many reps as they want. The user records his set, and the data is then uploaded to a Google Firebase database. The data for each recording is stored on the database in a folder with the username provided. Each recording stores meta data of when the recording was taken, the exercise performed, and the sensor locations used. Each sensor has a group number as well as an individual sensor ID number. This sensor ID number is used to identify which device was placed on the body.



2:34

Configuration

Please select the sensor locations you will be using by clicking the body element below. Click Done when finished.

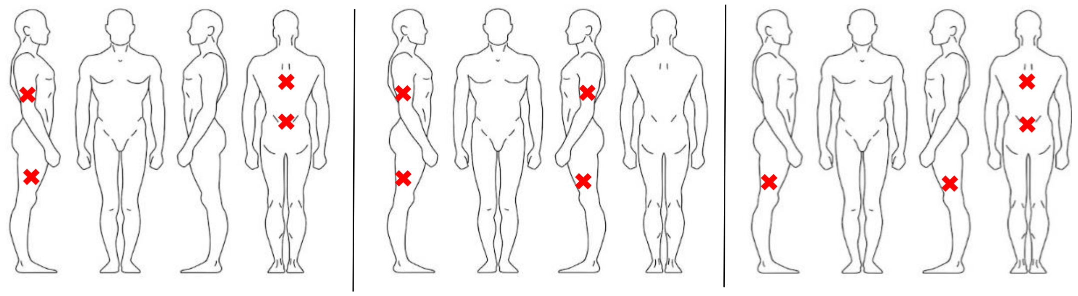
Figure 2.2 Screenshot from the SPP MyPT app showing possible sensor locations on body.

As previously outlined only three different exercises were selected for the project, the back squat, the conventional deadlift, and the power clean. Three exercises were chosen to increase the amount of data collected for each exercise as CNNs are often highly reliant of having large amount of data to produce accurate results. As squats and deadlifts are so commonly performed, participants are also more willing to participate in the study as they are already familiar with the exercise.

It was identified that there is a trade-off between the number of participants used and the amount of data each participant collects. To ensure high participant participation the study needs to take as little time as possible, while also collecting a meaningful amount of data for analysis. For example, a study that has participants perform 10 sets of all 3 exercises for 10 reps, will generate a significant amount of data (100 training reps for each exercise) but will take a significant amount of time to carry out and so participants may be less willing to participate. Less participants reduces variation causing significant biases in the data. In the other extreme, if participants are only asked to perform one set of 10 reps of only one of the exercises, then participants will be more willing as it takes up less of their time which may mean more participants in the study. However, as each participant is producing less data, for this example, 30 more participants would be required to produce the same amount of data as for the previously explained method. Which causes the problem of finding the larger number of participants.

There is also the problem of where to place the sensors. Research questions 3 and 4, require the placement of sensors to be explored so that the effects of having sensors on different parts of the body can be analysed. To ensure this, three separate sensor configurations were chosen, as shown in Figure 2.3, for participants to use to collect data. The configurations were chosen in such a way that key locations are used across multiple configurations. These key locations were suspected of being the most important for classifying exercise quality based on consideration of the biomechanics involved with the three exercises. Having the configurations in such a way allows for experimentation into how each of the sensors effects results while not hindering the amount of data available for any given configuration. For example, if only the right leg and chest sensor want to be evaluated for performance then there is the same amount of data available for it as that of the chest and right arm sensor, assuming all data has been evenly collected across the configurations. Having three different configurations allows for a larger exploration of the optimal sensor position as only four devices can be connected at any given time.

The exact placement of the devices on the body for each configuration was chosen so not to have the devices interfere with the user during the exercise. For example, the chest and abdomen devices are specifically moved to the back as opposed to the front of the body as during a clean the device may get hit by the barbell, possibly damaging the device, and causing malfunctions. The devices were also always placed in the same orientation with the power switch facing to the ground when the user is stood upright with arms by their side.



Configuration 1

Configuration 2

Configuration 3

Figure 2.3 Three different sensor configurations on the body with the SPP device locations marked by the red crosses.

After reviewing the literature on HAR for exercise classification [7] [12] [23] it was decided that the study would aim to collect data from 30 different participants. To ensure that this target is met and that each participant produces a good amount of data, participants were asked to only perform 6 sets of up to 2 different exercises, resulting 6 separate recordings. Only two exercises were chosen as it was found that there was significant time spent switching between exercises.

The participants were asked to perform between 5-12 reps for each set, but this was not strictly controlled as participants differed in ability. It was also done as the weight that participants lifted was not controlled and so they were told they could lift as much as they like up to an amount that they could safely perform about 5 reps with. This meant that for some participants it would be very difficult to perform 12 reps of the weight they chose. By not controlling the weight used, participant skill level and strength can be better accounted for and provides more variation in the dataset. If all the participants were told to lift a very light weight, then the model would never see data for someone lifting heavy introducing biases in the data. The time taken for participants to perform the exercises, for example taking 2 seconds to do a back squat, was also not controlled. This was done as the models need to be able to classify exercises performed at varying rates. To further reduce biases, the exercises that the users performed were randomized but evenly distributed so that each exercise was performed the same number of times as the other ones.

Each participant was given a user ID number to anonymize the data collection and to ensure that the data is organized to allow for partitioning of the different users during training and testing. Each participant only completed the study once using one sensor configuration. These were changed for every three participants, for example three participants would use configuration 1 and the next three would use configuration 2. Although it would be better to have each participant record data for each configuration it was found, during preliminary testing, that there was significant time wasted changing the configurations around. Additionally, due to the devices being in an early development stage, there were issues when configuring and connecting the devices to the phone further increasing the time wasted. During preliminary testing it was found that only using one sensor configuration rather than using all three could save up to 20mins. As stated earlier ensuring that the study is as quick as possible while also collecting a useful amount of data was very important.

To ensure that this method was followed, each user ID had the exercises to be performed and configurations to be used generated before any of the data was collected. This meant that when a new participant was recruited, they could be immediately assigned to an unused ID, and it would already be determined what they would be doing. Each user ID has an SPP account with a fake email address and false fields for the personal information. This was done for data ethics purposes and because this project will not use this meta data. It should be noted that all participants sign consent forms to partake in the study.

To recognize technique quality, participants were asked to rate their lifting experience for the movements they were being asked to perform between: beginner, intermediate and advanced. These could then be used as labels for the machine learning model to train on allowing it to output what experience level it thinks the user is. This method is somewhat simplistic but is suitable considering the limited time frame of the project.

To summarize the process, each participant performs 6 sets of exercises with at most two different exercises being performed. Each participant is assigned a unique user ID number which has a SPP account associated with it. The user ID has the single sensor configuration and the allocated exercises that they will perform associated with it. The participant is asked to use a weight that they can safely lift and is asked to try to perform between 5 to12 reps. The weight that they use is not strictly controlled to allow for a greater variation in perceived effort in the dataset. Each participant is asked what their experience level is for exercises they perform from beginner, intermediate or advanced. This level is then recorded with the user ID for that participant.

Participants were recruited from a local gym in Bath and from the weightlifting and powerlifting society at the University. To try to increase the diversity of the dataset, all ages of people were approached in the local gym and asked if they wanted to participate. The participants from society were all students however there was a large range of abilities and body types. As stated, age, weight and height were not recorded so summary statistics cannot be calculated.

2.2 General Data Pre-processing

The data collected is stored on a Google Firebase which uses a NoSQL structure; each participants data can be accessed by querying the username used for the account the participant was recorded under. The recordings can then be filtered by the exercise that has been performed. However, they cannot easily be filtered based on configuration without the prior knowledge of which configuration is associated with each user. This is due to the way the database is set-up and is another reason why each user was only asked to use one sensor configuration as accessing the data for a single configuration becomes a matter of querying all the data for the associated user IDs.

The data was extracted into folders based on what was trying to be achieved. For example, the first task, given by research question 1 in section 1.2, only requires squat data for configuration 1 and so only that data was extracted for that question. But for research question 2 all the configuration 1 data is required so is stored in a separate folder. Although not the most efficient way for data storage, it simplifies the process for model development as only data that is in the immediate folder needs to be handled.

Each recording from a given participant is saved with a filename containing their user ID, sensor configuration, exercise performed, and the exercise recording number given by the order in which they were recorded. Each file contains the data from each of the four sensors stored as separate columns. Columns are grouped by sensor position and ordered. The order of the groupings in the file is the same between like configurations. The time series data was zeroed and converted to seconds.

In some instances, the data from the sensors were not aligned due to sampling errors. Determining which sensors were sampling incorrectly was difficult as the actual time of the recording was not known as each sensor records independently. It was decided that the data would not be altered or adjusted. Instead, the recording times of each of the sensors was measured and compared with one another. If the data was out of sync by more than 1 second, then the data was discarded. This value was chosen after an analysis of the length of each exercise during the data segmentation stage. The time chosen needed to be such that two separate reps could not occur within the same window. The analysis found that the minimum time between reps was about 2 seconds. This value was halved giving 1 second so that the start of each rep could occur within the same window for each of the sensors.

The exercise recordings needed to be labelled to segment the individual reps for the algorithm to train on. A graphical user interface (GUI) was developed in MATLAB to view the data of each recording and segment the reps in the data. MATLAB was chosen because it has several built in-functions that allow this type of GUI to be easily developed when compared with python. Each recording consisted of the 4 sensors and their 6 time series, these were plotted as shown in **Error! Reference source not found.**in the appendix. The period for each rep and the number of reps was selected based on visual analysis of how long each rep appeared. The start of each rep was then selected using the cursor and the rep windows plotted for verification. The timestamps of the start of the windows along with the size of the windows was then saved into a data structure with the name of the recording. This process was repeated for all the recordings. Additionally, during the segmentation process a note of the sensors used to visually segment the reps was made. This was done to provide qualitative information to assist in determining the optimal sensor locations.

As expected, reps within a given recording will have different time periods, however it was found that this was marginal and so a fixed period for each recording was used to save time. However, the time periods of reps in different recordings were very different between the three exercises.

The periods of the reps determine the window size used in the algorithm. Having a varying window size is difficult to implement due to the algorithm having to figure out what the correct window size should be on unseen data. A fixed window size was chosen based on the dataset being used. For example, the first model only used squat data for one configuration and so only that data was used to inform the window size for that model. The window size was determined by producing a histogram of all the time periods for each recording, calculating the average and analysing what the most suitable value would be based on the results. An example of the histogram of the squat configuration 1 dataset can be seen in in the appendix. The average calculated window size for the reps is then used for the size of the window in the model. It was found that all the models had a similar average window size of about 2.4s.

The timestamps for the start of each rep are used to train the CNN. A new time series of labels for each recording was created to train the classifier, with a length equal to the length of the recording minus the length of the window. Each exercise was assigned a specific value to allow for multiclass classification while a value of 0 was used when there was no exercise start present. To account for error in the labelling process each exercise label was extended to cover a range of about 0.3 seconds or 5 samples either side of the initial true label. This means that for each rep there are 11 labels in the time series. Doing this also helps with training as it increases the amount of data the model is trained. The value used was chosen based on visual analysis of the range over which there was uncertainty of values for the start of reps.

Some of the sensors stopped recording during the experiment resulting in missing sensor data for some of the recordings. It was decided that recordings with missing data would be used for the configuration 1 classifiers due to the limited amount of data performed in this configuration. However, the classifiers used during the investigation of sensor placement required all sensors to be working for the investigation to be conducted.

Each model used either a LOOCV or a k-fold cross validation method. This was done by user as user data is highly correlated and should only appear in one of dataset splits. These methods were chosen because of the limited size of the whole dataset (30 participants). If the model was only tested on one user, then the results become highly dependent on that one user. These methods give a better representation of how the classifier is performing. LOOCV was used for all the configuration 1 classifiers as there was very limited data available for these. A k-fold approach was taken for the classifiers with different sensor combinations as there was more data available.

A scaler was trained on each training dataset and applied to each corresponding test dataset. The scaler chosen was a standard scaler from sklearn which standardizes features by removing the mean and scaling to the unit variance. Standardization is very common practice in ML estimators and ensures that features are scaled so that the model does not bias towards a particular feature due to differences in the inherit size of the features.

The user recordings for each of the training, validation and test datasets are combined to give one continuous multivariate time series for each. This was done as when the sliding window technique is applied there is a loss of data equal to the length of the window size for each dataset the technique is applied to. By combining recordings into a continuous recording for each dataset, the data lost is reduced.

2.3 General Model Architecture

The model algorithms were all implemented in python using Tensorflow 2.11.0 and packages from the scikit-learn library.

The sliding window technique was applied to each of the datasets using the window size determined during the analysis of the rep time periods. No step interval was applied so that windows were overlapping resulting in a window for each sample. Each of these segmented windows is associated with a label. These labels are used for training in the test datasets and used for evaluation of model performance in the validation and test datasets. The window size gives the inputs to the CNN. The CNN produces an output probability for whether an exercise or no exercise is present (probability of 1) or not (probability of 0). Figure 2.4 illustrates this process of how the model produces predictions using a univariate time series and a single output. The process is easily extended to a multivariate time series by using windows for each time series and running them in parallel. For example, configuration 1 consists of 4 IMU sensors, each sensor takes 6 readings. For a window size of 2.4 seconds, at 35Hz there are 72 samples per window. Thus, there would be 24 times series of 72 samples, these time series are kept separate and fed directly into the model.

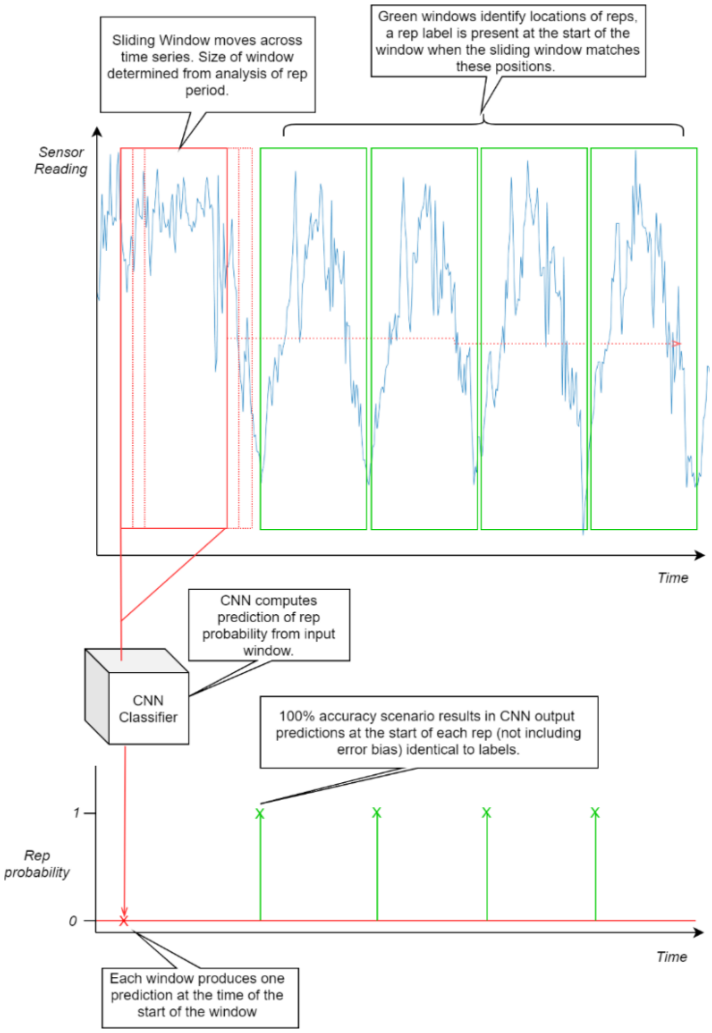

Sliding Window moves across time series. Size of window determined from analysis of rep period.
Green windows identify locations of reps, a rep label is present at the start of the window when the sliding window matches these positions.
Sensor Reading
Time
CNN computes prediction of rep probability from input window.
CNN Classifier
100% accuracy scenario results in CNN output predictions at the start of each rep (not including error bias) identical to labels.
1
Rep probability
0
Time
Each window produces one prediction at the time of the start of the window

Figure 2.4 Diagram of how the sliding window technique is used to feed data through the CNN to predict start of rep probabilities, demonstrated with a univariate time series input example.

The output of binary classifiers is a single value probability for whether the start of that exercise was detected. The binary models are trained for each exercise and the positive labels correspond to when the of a rep is present. The multiclass classifiers give a probability for each exercise where the sum of all the probabilities is equal to 1. The multiclass models are trained on a label that has been encoded to each exercise. It is important to note that this method of identifying locations of reps fulfils both the classification and segmentation requirements.

The model consists of 3 layers of a 1D convolution and max pooling, and 1 fully connected layer. The number of layers and parameters were not highly tuned as the size so to ensure that the model was not overfit to the data. A single architecture was chosen after a review of a similar model used by Soro et al. [7]. The same model structure was used for all classifiers with only the inputs and the outputs of the model being changed depending on the number sensors being used and the classification type. The details of the model can be found in Figure 7.3 Sample plot of results of Binary Squat Classifier for Configuration 1 including input time series sensor data and labels. in the appendix. A batch size of 30 was used and each model was trained for 10 epochs. The model is trained using stochastic gradient descent and a learning rate of 0.001.

All the models used a fixed random seed. The seed chosen for each was selected from analysis of the performance of each of the models after being run on 10 different seeds. This was done to ensure consistency between re-runs of the model and to ensure that no model was using a particularly bad seed to allow for a fairer comparison of the models.

It was found early on that the model was producing unreasonably low accuracy scores despite appearing to perform well at identifying and segmenting reps. There was often a marginal temporal error of less than 0.5s in the time series between when the rep was predicted and where the label was placed. It was decided that this slight error was not significant enough to warrant a false detection and that it should be classed as correct detection. Thus, a prediction error margin method as illustrated in Figure 2.5 Illustration of how the prediction error margin adjusts output predictions to better evaluate the performance of the model. The error adjusted prediction signal is the result of altering each of the prediction based on whether they lie within the margin of error for the label. This method is used for evaluation purposes only.was introduced, to better quantify the performance of the model. This method works by looking at every positive prediction in the prediction after a threshold or max has been applied to the raw probability outputs of the model. Note the use of bars as opposed to impulses are due to the additional labels that have been applied to either side of the initial label to account for errors in the labelling process.

If any label does not lie within the prediction error margin of a given prediction sample, then that sample is not corrected and is retained in the output. If at least one prediction sample has an error margin containing at least one positive label, then all the positive labels in that rep are present in the error adjusted position output. This is because the model is detecting a rep at a given point in time, so all the labels associated with that rep need to be in the output. An example of this method applied to an actual signal can be seen in Figure 7.4 in the appendix. The size of the error margin was set to 0.63s, this equates to 10 samples either side of the prediction. This value, was chosen in line with the observation of most predictions having a temporal error of less than 0.5s

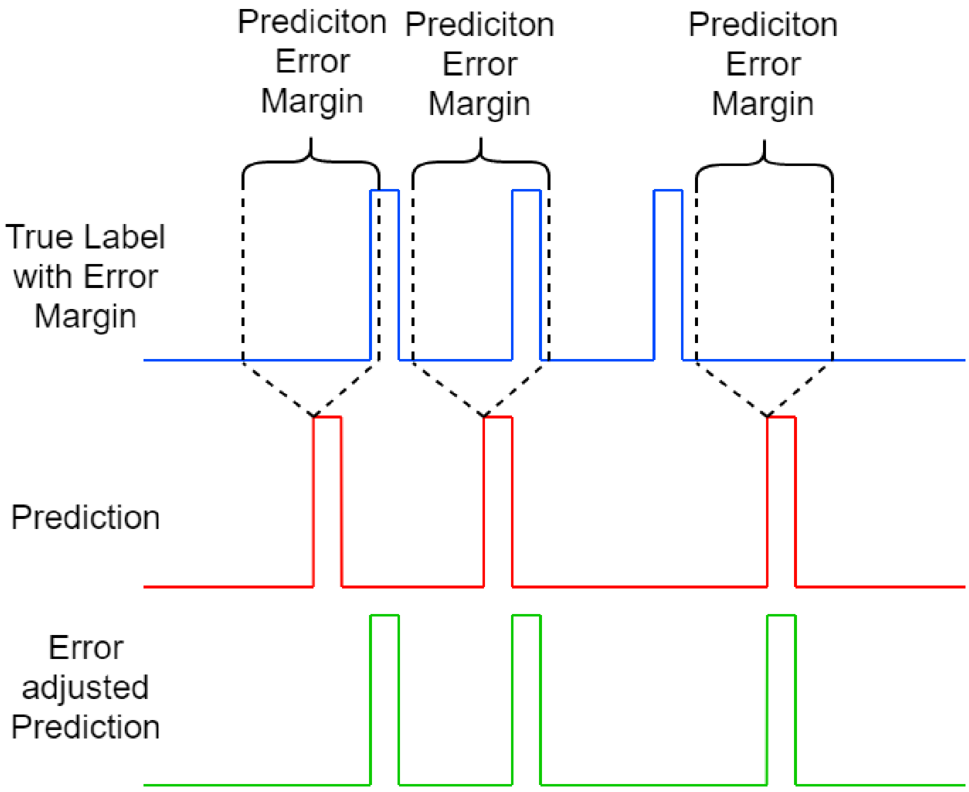


Figure 2.5 Illustration of how the prediction error margin adjusts output predictions to better evaluate the performance of the model. The error adjusted prediction signal is the result of altering each of the prediction based on whether they lie within the margin of error for the label. This method is used for evaluation purposes only.

2.4 Squat Classifier for Sensor Configuration 1

2.4.1 Methodology

To answer research question one of whether squat reps can be segmented from exercise data, from Aims and Research Questions, a single output CNN (binary network) was used and trained on the labelled squat data collected using only sensor configuration 1. Sensor configuration 1 was chosen as the initial test configuration as it covers the most unique locations on the body, as opposed to the other configurations which have sensors located on opposite sides of the body.

2.4.2 Binary Model Results

The results in Figure 2.6 and Table 2.1 have been provided to show how the model responds when the error margin method is not used. The model achieves an average balanced accuracy of 0.66 using a threshold of 0.3. The threshold value is the value above which probabilities are taken as positive predictions.

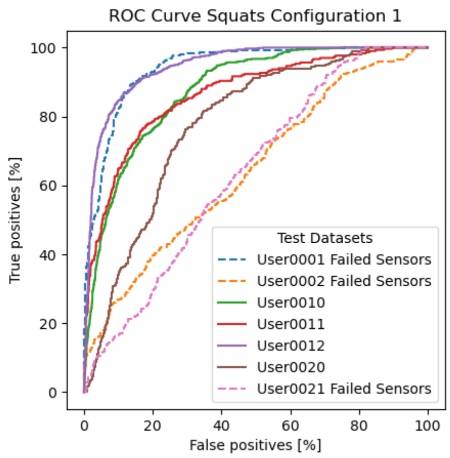


Figure 2.6 Receiver-Operator-Characteristic (ROC) Curve of all test datasets for the initial binary squat classifier for sensor configuration 1.

Table 2.1 Summary statistics table of all test datasets including ROC Area Under Curve values and Balanced Accuracy Scores with 0.3 threshold.

User ID	ROC AUC Score	Balanced Accuracy Score (0.3 threshold)
User0001	0.94	0.70
User0002	0.64	0.50
User0010	0.88	0.72
User0011	0.87	0.78
User0012	0.95	0.85
User0020	0.78	0.57
User0021	0.64	0.50
Average	**0.81**	**0.66**

As explained, the failed sensor data has been included for this model. The model generally performs worse on this data when compared with the data that contains all the sensors. However, the model achieves a very high ROC AUC score for User0001. It is later found during the investigation of the sensor placement that the abdomen sensor missing for this data is not as important as the other sensors in this configuration, hence its better performance.

Figure 2.7 and Table 2.2 demonstrate the performance of the model with the error margin included. The method to calculate the ROC curve and ROC AUC values were adjusted to allow for this new method. As expected, this method improves perceived performance by accounting for slight temporal errors in the output. When not considering the failed sensor datasets the model performs very well achieving nearly perfect accuracy for the User0011 and User0012 datasets.

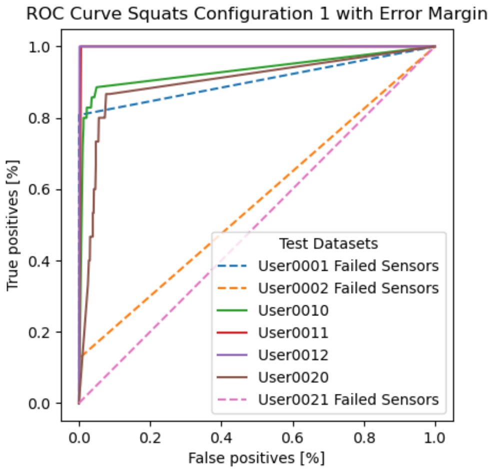


Figure 2.7 Receiver-Operator-Characteristic (ROC) Curve of all test datasets for the binary squat classifier with the prediction error margin for sensor configuration 1.

Table 2.2 Summary statistics table of all test datasets including ROC Area Under Curve values and Balanced Accuracy Scores with 0.3 threshold.

User ID	ROC AUC Score	Balanced Accuracy Score (0.3 threshold)
User0001	0.90	0.85
User0002	0.56	0.50
User0010	0.93	0.90
User0011	1.00	0.98
User0012	1.00	0.99
User0020	0.90	0.84
User0021	0.50	0.50
Average	**0.96**	**0.79**

2.5 Squat, Clean and Deadlift Classifier for Sensor Configuration 1

2.5.1 Methodology

To answer research question two of whether squats, cleans and deadlifts can be segmented and classified from exercise data, two versions of CNNs were tested. The models were run on data with missing sensors due to the limited amount of data available.

The first model consisted of a multiclass output model that is fed with all the recordings for configuration 1. Each of the exercises along with the no exercise (none) class was encoded with a label that the model would learn and outputs the related probabilities of each exercise to the four output nodes. The max probability of these nodes was taken as the predicted label. The ROC values were calculated by treating each output node of the model as binary output and testing against labels only associated with that output. For the none output this needed to be inverted to be compatible with the prediction error margin method before being re-verted.

The second model consisted of 4 single output CNNs trained to only segment and classify 1 of the 4 possible classes. The probability outputs of each of these models were then combined and normalized to sum to 1. The max value after normalization was taken as the predicted label. The ROC values were calculated by taking the output probability for each class of binary CNN before combination. As before, the no exercise class output was inverted for the prediction error margin method before being reverted for combination.

2.5.2 Multiclass Output Model Results

As seen in Table 2.3 and Figure 2.8 the multiclass model can classify and segment different exercise, but with generally poor accuracy. It appears the model generally performs better when the full sensor configuration is available. However, there appear to be exceptions even when all the sensors are available. The missing values are because users only performing 1 or 2 exercises in total.

Table 2.3 Performance metrics of the multiclass output classifier for sensor configuration 1: including ROC AUC values and balanced accuracy scores for each test dataset with summary averages.

User ID	ROC AUC Values				Balanced Accuracy Score	Sensor Failure (Yes/No)
	None	Squat	Clean	Deadlift		
User0001	0.99	0.98		0.99	0.88	Yes
User0002	0.50	0.50		0.50	0.33	Yes
User0003	0.73		0.50	0.72	0.55	Yes
User0010	1.00	1.00		1.00	0.96	No
User0011	0.99	1.00	0.93		0.94	No
User0012	0.85	1.00		0.78	0.75	Yes
User0019	0.85		0.75	0.87	0.59	No
User0020	0.84	0.86		0.84	0.65	No
User0021	0.85	0.90		0.88	0.56	Yes
Average	**0.84**	**0.89**	**0.73**	**0.82**	**0.69**	

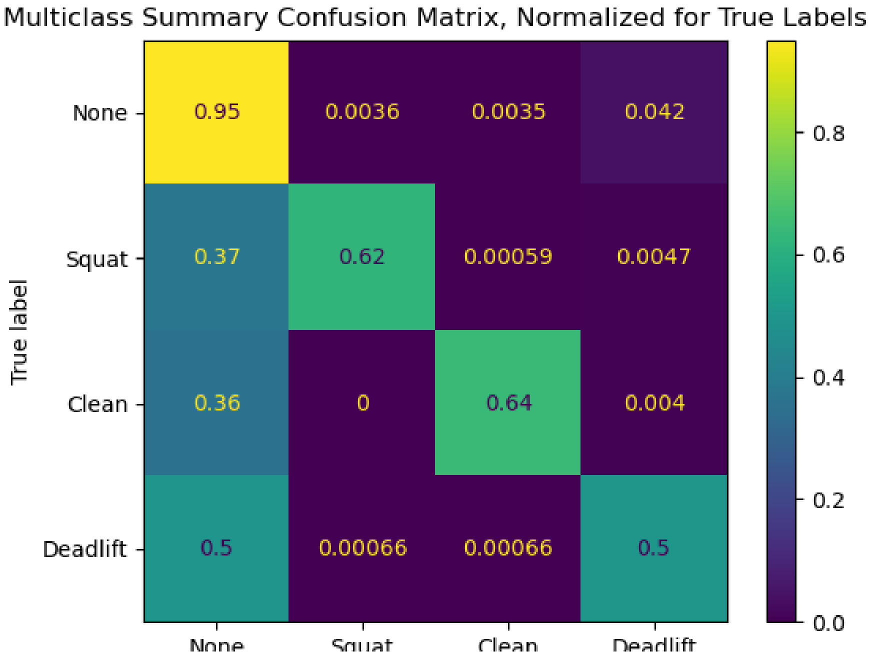


Figure 2.8 Summary confusion matrix for multiclass output classifier trained on all exercises for only configuration 1. True labels have been normalized to 1 to provide better comparison between exercises. Note that colour grading is normalized to the max value in the matrix.

2.5.3 Combined Binary Output for Each Exercise

The results for the combined binary output can be seen in Table 2.4 and Figure 2.9. The multiclass performs better across most of user datasets when compared to this model. This model also takes significantly longer to train then the multiclass model as four separate CNN networks need to be trained and tested. It is hard to know the exact reason for the underperformance of the binary model.

Table 2.4 Performance metrics of the combined binary output classifier for sensor configuration 1: including ROC AUC values and balanced accuracy scores for each test dataset with summary averages.

User ID	ROC AUC Values				Balanced Accuracy Score	Sensor Failure (Yes/No)
	None	Squat	Clean	Deadlift		
User0001	0.96	1.00		0.57	0.68	Yes
User0002	0.55	0.88		0.50	0.33	Yes
User0003	0.79		0.50	0.83	0.63	Yes
User0010	0.98	1.00		0.86	0.86	No
User0011	1.00	1.00	0.96		0.88	No
User0012	0.81	1.00		0.72	0.66	Yes
User0019	0.84		0.50	0.85	0.52	No
User0020	0.76	0.60		0.92	0.50	No
User0021	0.70	0.50		0.95	0.52	Yes
Average	**0.82**	**0.85**	**0.65**	**0.78**	**0.62**	

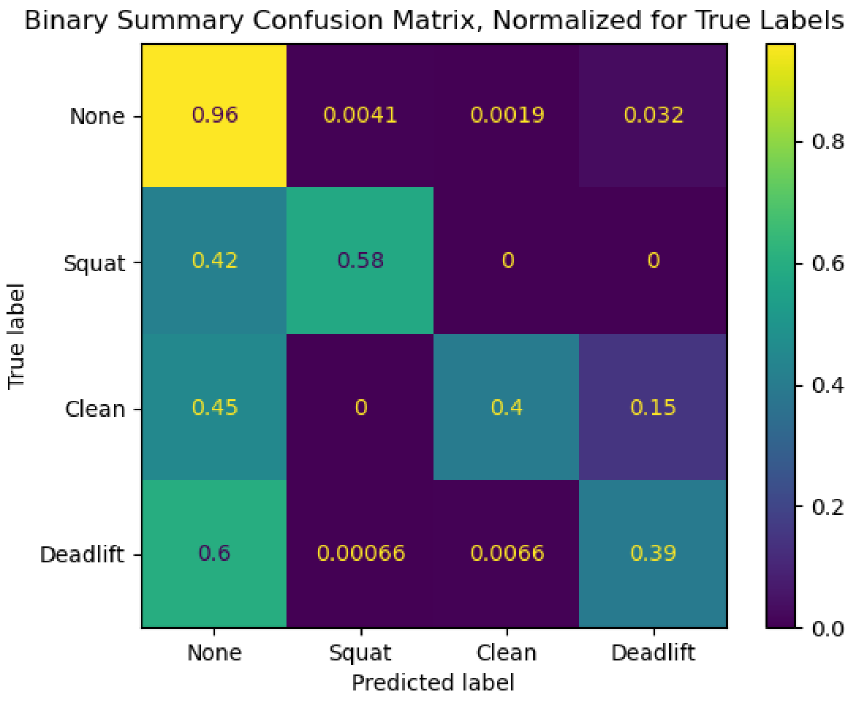


Figure 2.9 Summary confusion matrix for combined binary output classifiers trained on all exercises for only configuration 1. True labels have been normalized to 1 to provide better comparison between exercises. Note that colour grading is normalized to the max value in the matrix.

2.6 Investigation of the Effect of Sensor Placement on the Multiclass Classifier

2.6.1 Methodology

To address research question two of where the ideal sensor placement is, a study on the effect of using combinations of sensors at different locations was carried out. Each possible sensor combination was listed using the sensor location available from the sensor configurations. The number of recordings associated with each possible combination was calculated for each exercise. This was biased towards sensors that appear in more of the combinations such as the right leg sensors. This process inherently omits failed sensors as recordings that do not contain that sensor are not included in the dataset.

A histogram was produced using the qualitative analysis collected during the segmentation process of the sensors used for segmentation. As each recording had sensor locations associated with it corresponding to the sensors used during visual segmentation, the number of recordings associated with each sensor location could be calculated. The histogram was used to gain a better understanding of which of the sensor combinations should be analysed first. Additionally, a threshold was applied to the number of recordings for each exercise of each possible sensor combination that ensure combination with low amounts of data are not used. This threshold was chosen to be 22 based on analysis of bar charts of the number of recordings for each possible sensor combination for each of the three exercises. Using these two forms of analysis the chest and abdomen, chest and right leg, and abdomen and right leg sensor combinations were chosen to be evaluated first.

The multiclass model used in section 2.5 was used for the investigation as it achieves better performance and has a shorter training time. The model inputs were adjusted to allow for different number of sensors to be fed into the model. The seeds of the each of the models were also re-calculated for each sensor combinations.

As previously explained, LOOCV became unsuitable due to the increased amount of available data and so a K-Fold approach was taken, where k is the number of test datasets. The number of folds was chosen to allow for a data split of approximately 20% test data to 80% training data for each fold.

2.6.2 Chest and Right Leg Sensor Combination

Of the initial combinations tested the chest and right leg sensor combination performed the best. The results from the other two initial combinations are presented in the appendix in Figures Figure 7.5 and Figure 7.6, and Tables Table 5.2 and Table 5.3. This model was trained and tested on 33 squat recordings, 26 clean recordings and 33 deadlift recordings.

The results for this model, shown in Table 2.5 and Figure 2.10, show an improvement in performance when compared with the multiclass configuration 1 model in section 2.5.2. The increase in performance is only slight and may be caused by the increase in the amount of data this model is training on as well as that there is no failed sensor data in the dataset. The results do show that there is significant underperformance for classifying and segmenting cleans. It was suspected that the sensors placed on the arm may improve performance for this exercise due to the significantly different motion of the arm during a clean when compared with squats and deadlifts.

Table 2.5 Performance metrics of the multiclass output classifier for the chest and right leg sensor combination: including ROC AUC values and balanced accuracy scores for each 5-fold test dataset with summary averages.

6-Fold Dataset Number	ROC AUC Values				Balanced Accuracy Score
	None	Squat	Clean	Deadlift	
1	0.95	0.97	0.78	0.99	0.72
2	1.00	1.00	1.00	1.00	0.99
3	0.91	0.87	0.47	0.86	0.71
4	0.94	0.93	0.99	0.91	0.82
5	0.95	0.48	0.81	0.95	0.81
6	0.85	0.94	0.59	0.46	0.63
Average	**0.93**	**0.86**	**0.77**	**0.86**	**0.78**

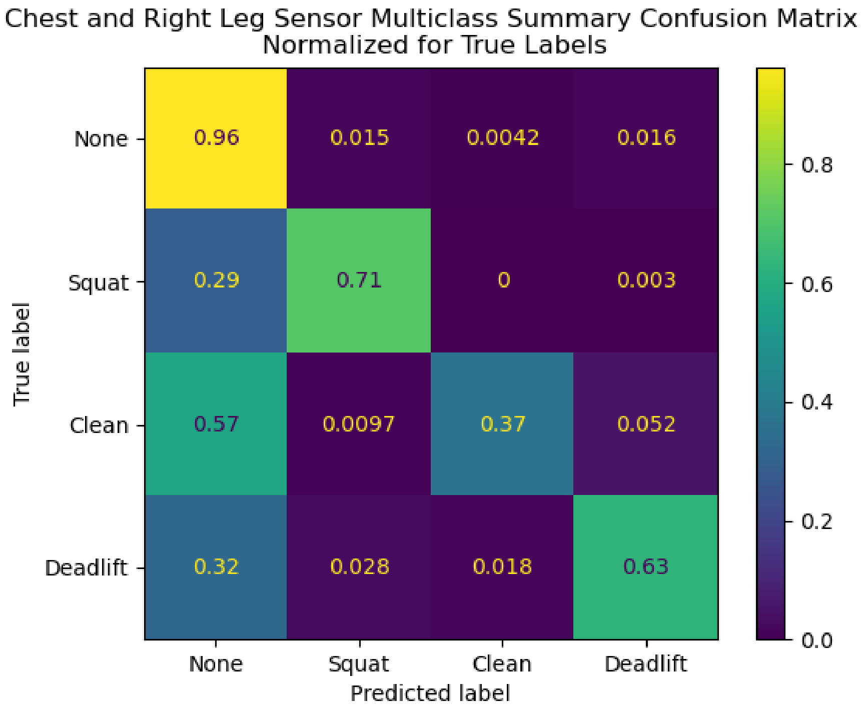


Figure 2.10 Summary confusion matrix for multiclass output classifier trained on all exercises for the chest and right leg sensor combination. True labels have been normalized to 1 to provide better comparison between exercises. Note that colour grading is normalized to the max value in the matrix.

2.6.3 Right Arm and Right Leg Sensor Combination

Using the hypothesis that the right arm may provide additional information to improve clean classification, a model using right arm and right leg sensor combination was tested. This combination was originally omitted as the right arm sensor was not used very often for the visual segmentation of reps, leading to the initial hypothesis that it was not as important as the more used sensor namely: the chest, abdomen, and right leg sensors. This model was trained and tested on 24 squat recordings, 30 clean recordings and 37 deadlift recordings.

Table 2.6 and Figure 2.11 demonstrate the improved performance of using the right arm sensor combined with the right leg when compared with the chest and right leg sensor combination. This combination performs far better on the clean dataset as predicted. However, the balance accuracy score suggests that it performs worse overall then the chest and right leg.

Table 2.6 Performance metrics of the multiclass output classifier for the right arm and right leg sensor combination: including ROC AUC values and balanced accuracy scores for each 4-fold test dataset with summary averages.

5-Fold Dataset Number	ROC AUC Values				Balanced Accuracy Score
	None	Squat	Clean	Deadlift	
1	0.93	1.00	0.85	0.94	0.76
2	0.93	1.00	0.99	0.89	0.87
3	0.90	0.49	0.96	0.98	0.65
4	0.86	0.75	0.94	0.73	0.67
5	0.87	0.79	0.93	0.87	0.69
Average	**0.90**	**0.81**	**0.94**	**0.88**	**0.73**

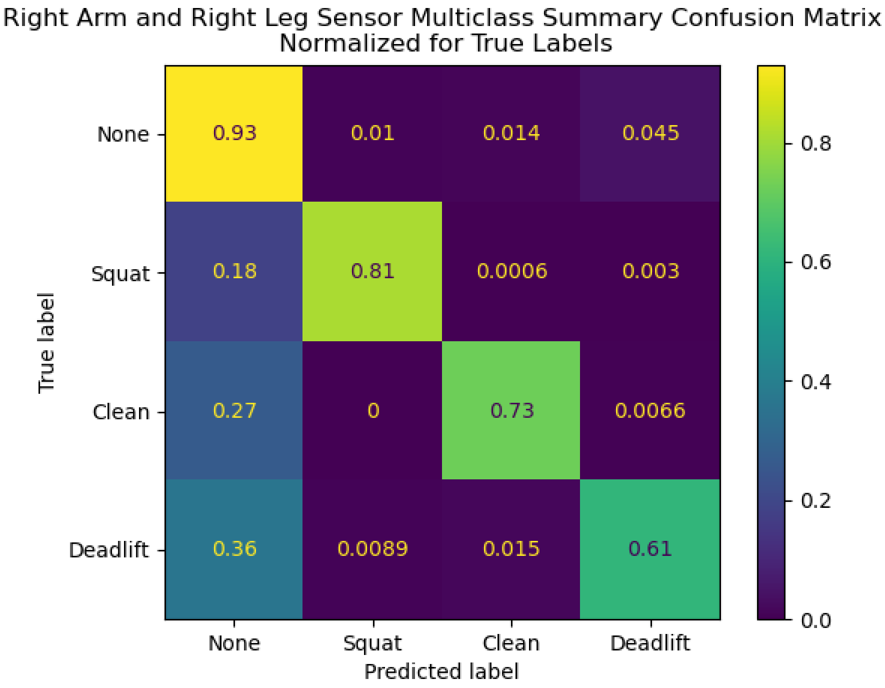


Figure 2.11 Summary confusion matrix for multiclass output classifier trained on all exercises for the right arm and right leg sensor combination. True labels have been normalized to 1 to provide better comparison between exercises. Note that colour grading is normalized to the max value in the matrix.

2.6.4 *Right Arm, Chest, and Right Leg Sensor Combination*

From the results in 2.6.2 and 2.6.3, it was hypothesized that combining the right arm, chest and right leg would give the optimal sensor placements for the locations experimented with. A model with these three locations was trained and tested on 24 squat recordings, 22 clean recordings and 12 deadlift recordings.

As seen by the results in Table 2.7 and Figure 2.12 this model performs well with an improvement in detecting deadlift recordings when compared with the right arm and right leg sensor combination model. However, as there was less data to train on for this model due it is difficult to fully conclude that this sensor combination is better than the of just the right arm and right leg sensors.

Table 2.7 Performance metrics of the multiclass output classifier for the right arm, chest, and right leg sensor combination: including ROC AUC values and balanced accuracy scores for each 4-fold test dataset with summary averages.

5-Fold Dataset Number	ROC AUC Values				Balanced Accuracy Score
	None	Squat	Clean	Deadlift	
1	0.95	0.98	0.42	0.92	0.71
2	0.99	1.00	0.93	0.98	0.89
3	0.95	0.99	0.78	0.87	0.76
4	0.92	0.92	0.46	0.93	0.77
5	0.96	0.90	0.84	0.44	0.72
Average	**0.95**	**0.96**	**0.69**	**0.83**	**0.77**

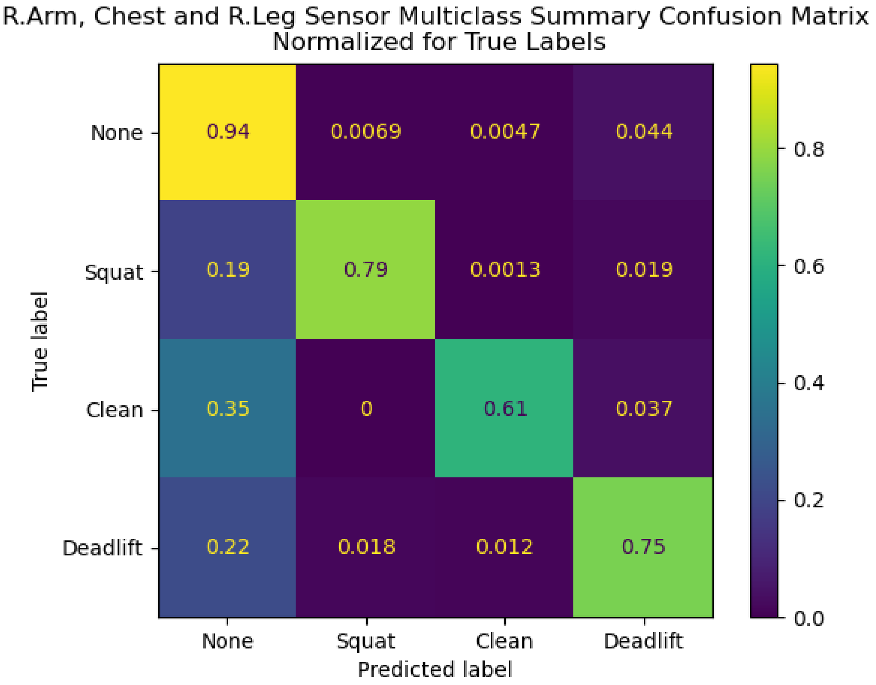


Figure 2.12 Summary confusion matrix for multiclass output classifier trained on all exercises for the right arm, chest, and right leg sensor combination. True labels have been normalized to 1 to provide better comparison between exercises. Note that colour grading is normalized to the max value in the matrix.

3 Discussion of Results

3.1 Data

Most of the collected data was collected from young people, primarily students. This was because this demographic was far more willing to participate as part of a student led project as opposed to older populations and the general population. This would likely have an impact on how the model is able to classify data from older participants. However, of the participants that were collected there was wide range of abilities and body types seen.

Regarding the exercise technique, using a metric that allows participants to only rate their ability from three performance levels is not an accurate way to rank exercise technique. This is because participants were highly likely to either underestimate or overestimate their lifting technique resulting in labels that did not necessarily accurately reflect the ability of the participants. In some cases, the ranking had to be adjusted as participants severely overrated or underrated themselves.

Participants were also very reluctant to perform cleans, particularly having to perform six sets of cleans. As a results, the user IDs with the 6 cleans were skipped over with the intention of finding participants willing to do them. However, these no willing participants could be found before the cut-off for the data collection resulting in less data for cleans being collected as only 26 participants were recruited in total.

3.2 Research Question 1

It appears from the results in section 2.4 that the squat classifier with configuration 1 was able to correctly segment squats from no exercise. It was clear that including data with failed sensors influenced the overall performance of the model. But with this considered and analysing dataset where there were no sensor failures, it appears that the model can segment the squat reps. However, due to the nature of this experiment it is hard to know whether the model is learning what squats are or whether it is simply detecting movement. This is important for multi exercise detectors as learning only movement would result in poor model performance.

3.3 Research Question 2

As highlighted by the results for research question 1 the models used to answer research question 2 needed to demonstrate that they were able to classify exercises and not just movement. This was the motivation for using binary classifiers and a multiclass one. By training binary classifiers on data that includes the desired exercise to be classified as well as all the other exercises, it can be determined via the performance of the model whether CNN models are able to learn the exercises and not just movement. This is important for future work as it informs us that the model can learn a particular exercise as opposed to just movement. As seen from the results the models can learn the different exercises just not particularly well.

The multiclass outperforms the combined binary model. The reason for this is hard to determine but may be associated with how the neurons interact with one another between exercises as opposed to the binary ones where the output of each model is only concerned with the exercise it is given to analyse.

The overall results are underwhelming for both models and again as for the squat classifier are influenced by the quality and amount of data that has been fed into these models. Past work has highlighted that levels of accuracy as high as 99% for exercise classification is possible [7] [13] [23]. However, these works separate classification and segmentation, making it difficult to accurately compare these works with the performance of the models presented in this project. These projects also use highly controlled data that is produced in a lab with different exercises and sensors placed in different locations. Further highlighting the differences between these projects and this one.

The ROC AUC metric for these models is misleading. The method in which this value was calculated had to be coded specifically for this project, as opposed to using an external library. This resulted in a possibly inflated or inaccurate measure of performance as seen by the difference in the curves for the model in Figures Figure 2.6 and Figure 2.7.

These models do however demonstrate that these models are behaving as desired by classifying exercises and not just movement thus answering research question 2, allowing for further investigation into the effects of sensor placement on performance.

3.4 Research Question 3

The method of only investigating certain sensor combinations ensures that only combinations with similar amounts of data are investigated. It also ensures that only recordings are used which contain all the sensors of the specified combination, so the data is more consistent between recordings. This is important, as seen by the performance of the models used in research questions 2 and 3 which vary greatly between users.

The left arm and left leg sensor positions were left out from this investigation. It was decided that symmetry was likely to be less important for classification and segmentation and would be more important for determining technique quality as it would highlight imbalances in the users' form, resulting in poor technique.

Of the initial sensor combinations analysed (chest and abdomen; abdomen and right leg; and chest and right leg) it was found that the chest and right leg was the best performing combination. It is likely that the chest and abdomen sensors were ineffective as these sensors often end up taking very similar readings because of the positioning on the participant's body. It was found that for people with shorter torsos these sensors became far closer together then with taller individuals likely further exaggerating this result. The abdomen likely performed worse than the chest sensor as it is less prone to pick up large upper body movements associated with the biomechanics of the chosen exercises.

The increase in performance introduced using the right arm sensor is not surprising. The motion of the arm is particularly different between the different exercises so despite not being able to visually distinguish where reps occur using only the arm sensor, clearly it contains information that assists the classifier to classify the exercises correctly. However, it is unclear why the overall performance suggested by the balanced accuracy is worse than that of the chest and right leg combination. It may be that the formulation of the balanced accuracy metric is not suitable for measuring model performance and that the confusion matrix should be prioritized in this case as the most reliable metric for model performance.

The right arm, chest and right leg sensor combination does not perform as well as expected. As mentioned, this is likely due to the limited data available for this combination. It does not pass the original threshold of 22 recordings that was applied to the combinations of sensor recordings, as it only contains 12 deadlift recordings. So, it is somewhat inappropriate to compare its performance to the combination models with only two sensors. But from the findings it is likely that the right arm, chest, and right leg would perform best.

Comparing the findings of this project with other works is again difficult due to the dissimilarity between the models used. However, O'Reilly et al. [23] found that a single IMU located on the shank of the leg can achieve 98% model classification accuracy. While their results for 5 IMUs located on the legs could achieve a classification accuracy of 99% for squats, lunges, deadlifts, single leg squats and tuck jumps. Soro et al. [7] had similar results that showed that increasing the number of IMU sensors on the body, one on wrist one on ankle, increased classification accuracy but only marginally, with an increase in performance from 98% to 99%. The results in this paper are thus somewhat in line with the results in the literature in that the sensor placement is very important for exercise classification and that certain sensors are more important than others.

3.5 Research Question 4 and General Findings

Due to time constraints of the project this research question was not fully addressed. However, a visual comparison was made of the data collected of participants with good technique and poor technique. It was found that there were considerable differences between participants, that in some cases resulted in a misclassification of the exercise as a different one.

One example was for User0028 whose squats were mislabelled as deadlifts for the right arm, chest, and right leg sensor combination model. This was particularly interesting as User0028 was noted as having excellent form during their squats. The likely reason for this is, is that this user was able to control the weight that they were lifting more easily than less experienced lifters. This may have resulted in features that are more commonly seen in deadlifts, where controlling the weight throughout the movement is often easier even for those with poor technique. A sample plot of User0028 data can be found in Figure 7.7 along with examples for comparison of correctly labelled squat and deadlift recordings in Figures Figure 7.8 and Figure 7.9 respectively in the appendix

Another situation where exercises were often mislabelled, occurred with cleans. Many participants who were very inexperienced at performing cleans had to perform the movement very slowly, often resembling more of a muscle clean which lacks the power that is used in a full clean. This resulted in these recordings being labelled as deadlifts rather than cleans. This was especially the case for the chest and abdomen sensor combination model as the lack of the information provided by the arm sensors made certain recordings look very similar between deadlifts and cleans. Examples of the data collected by chest and abdomen sensors during a clean performed slowly and a regular deadlift are displayed in Figures Figure 7.10 and Figure 7.11 in the appendix.

What these examples highlight is that the model can identify differences in technique but struggles as differences can often appear to resemble different exercises. This is important as it helps to inform possible avenues for future work.

4 Project Review, Further Work, and Improvements

Despite not achieving high levels of performance, this project does provide valuable information as to how further work should be carried out going forward. It also provides valuable information as to where sensors should be placed and the importance of collecting large amounts of consistent data.

Work done by other researchers indicates that exercise classification is a task that can achieve high accuracy for a wide range of exercises. However, almost all the work is not concerned with counting reps and only achieves classification of a set of exercises, as opposed to individual reps. The few papers that do attempt the counting of reps are not concerned with the segmentation of the reps which is more difficult as it requires the start and end of the rep to be known.

The knowledge that high levels of performance can be achieved for exercise classification, and that a single CNN can struggle with segmenting and identifying reps of participants with varying levels of lifting experience, should be used for future work. It is suggested that an approach that takes these two aspects into account is used when building a model capable of identifying a large range of exercises that also gives feedback as to the quality of technique of each repetition.

A possible solution is a combination of CNNs in 3-stage model format. The first stage tasked with the classification of exercises, the second for segmentation of the reps of the identified exercise, and the third for recognizing technique of the identified exercise. Thus, there would be two models for each exercise (one for segmentation, one for technique) and one model for exercise classification. This allows the models for rep segmentation and technique recognition to be more refined for their tasks. This can possibly mitigate the issues found with the misclassification of certain exercises. It also allows the models developed in the relevant literature that achieve very high levels of accuracy at classification of exercises to be used in the initial stage. Separating the tasks into stages also allows for different ML methods to applied to each of the tasks. Such as, using unsupervised methods on the exercise technique model that could give a score of how well it thinks the rep was performed based on looking for patterns in the data and associating them with better and worse performance. This would remove the need for the rudimentary method of having participants rate their technique used for this project.

In addition, the method of data collection can be improved to allow for more accurate labels and remove the need for manual labelling. There are a few ways to achieve this, the method suggested by Soro [7] of using a vibration to inform the user of when to do a rep is beneficial as the labels for the start of the are then immediately known. Unfortunately, this has limitations as a participant may fail to perform a rep when the vibration is made. An alternative would be to have an assistant record the moments when a rep starts and stops on the same device that is taking the readings from the sensors, giving labels for the start and stop of each exercise. However, this requires an assistant which is not always available. This assistant could be replaced by using voice commands to signal the start and end of the rep; but would require some significant additional coding to implement.

Alternatively, if a project is primarily concerned with the recognition of exercise technique, then only the data from the reps needs to be recorded without any of the unwanted data that occurs when the exercise is not performed but would be required to train the segmentation models and classifier. This method could work by getting the participant to stand as still as possible between reps and space reps out by a couple of seconds. The model would then be able to immediately determine what is a rep by only looking for periods of high energy activity in the time series. This could be achieved using a visual cue to the participant that tells them whether activity is detected to ensure that they are standing still enough that the device is not recording movement. This method is unsuitable for exercise classification as it forces participants to stand perfectly still between reps which is unrealistic in a gym environment.

Additional work could also be done to explore the effects of varying the location of the sensors on these limbs such as placing the sensor on the forearm or wrist. In addition, it was found that sensors located on the upper arm would cause discomfort for participants due to rubbing, particularly during cleans when the arm would be fully contracted. So additional work should be done to ensure that the sensors are comfortable to use.

The sensors used in the project were still in a development stage which explains the issues with missing sensor data. This project has shown the importance of ensuing all the sensors are available and so future work will need to implement strategies to help mitigate this issue.

5 Conclusions

The method for data collection was methodical and well planned but did not result in as much data as was needed to properly develop models. The data collected, although biased, was sufficient to provide evidence for capability. Alternative techniques to gather data have also been suggested by this project and methods for recruiting participants will also need to be improved. The project suggests that significant planning is done on exactly how the data is gathered and that a large pool of willing participants is found before the project is initiated.

The project was able to demonstrate that exercises can be classified and segmented using CNNs but with relatively poor accuracy when compared to the classification abilities of other works [7] [12] [23]. Additionally, the results have shown the importance of sensor placement on the model performance, showing that placing sensors on an arm, the chest and a leg is likely to achieve best performance for classification and segmentation tasks. The identification of the limitations of the models developed on this project, as well as the knowledge of how alternative works could be employed for future use, provide a strong basis for future projects that aim to achieve similar objectives to this project.

This project does not meet all its objectives but did provide evidence of potential success for future projects. The success of this project lied in its ability to generate useful information for future work and for the product development for SPP, which this project has achieved.

6 References

[1] IBISWorld, "Number of gyms and fitness centres in the United Kingdom (UK) from 2011 to 2022 [Graph]," 10 January 2022. [Online]. Available: https://www.statista.com/statistics/1194837/gyms-fitness-centres-uk/?locale=en.

[2] IBISWorld, "Number of personal trainers in the United Kingdom (UK) from 2011 to 2022 [Graph]," 28 April 2022. [Online]. Available: https://www.statista.com/statistics/1194844/number-personal-trainers-uk/?locale=en.

[3] E. Gorman, H. Hanson, P. Yang and et al., "Accelerometry analysis of physical activity and sedentary behavior in older adults: a systematic review and data analysis," *European Review of Aging and Physical Activity,* vol. 11, p. 35–49, 2013.

[4] C. Torres-Huitzil and Alvarez-Landero, "Accelerometer-Based Human Activity Recognition in Smartphones for Healthcare Services," *Neuroinformatics,* no. 5, pp. 147-169, 2015.

[5] A. Davoudi, A. Wanigatunga, M. Kheirkhahan and et al., "Accuracy of Samsung Gear S Smartwatch for Activity Recognition: Validation Study," *JMIR Mhealth Uhealth,* vol. 7, no. 2, 2019.

[6] M. Muehlbauer, G. Bahle and P. Lukowicz, "What Can an Arm Holster Worn Smart Phone Do for Activity Recognition?," in *15th Annual International Symposium on Wearable Computers,* San Francisco, 2011.

[7] A. Soro, G. Brunner and et al., "Recognition and repetition counting for complex physical exercises with deep learning," *Sensors (Switzerland),* vol. 19, no. 3, 2019.

[8] C. Shen and et al., "MiLift: Efficient Smartwatch-based Workout," *IEEE Transactions on Mobile Computing,* vol. 17, no. 7, pp. 1609 - 1622, 2018.

[9] C. Seeger and et al., "MyHealthAssistant: A Phone-based body sensor network that captures the wearer's exercises throughout the day," in *BODYNETS 2011 - 6th International ICST Conference on Body Area Networks,* Beijing, 2011.

[10] R. P. L. B. G. Mason and et al., "Wearables for Running Gait Analysis: A Systematic Review," *Sports Med,* vol. 53, p. 241–268, 2023.

[11] M. Shoaib, S. Bosch, O. D. Incel and et al., "Complex human activity recognition using smartphone and wrist-worn motion sensors," *Sensors (Switzerland),* vol. 16, no. 4.

[12] D. Morris, T. S. Saponas and et al., "RecoFit: Using a wearable sensor to find, recognize, and count repetitive exercises," in *Conference on Human Factors in Computing Systems,* Toronto, 2014.

[13] K.-H. Chang and et al., "Tracking free-weight exercises," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* Innsbruck, 2007.

[14] B. Fu, F. Kirchbuchner, A. Kuijper, A. Braun and D. V. Gangatharan, "Fitness activity recognition on smartphones using doppler measurements," *Informatics,* vol. 5, no. 2, 2018.

[15] H. Ding and et al., "FEMO: A platform for free-weight exercise monitoring with RFIDs," in *13th ACM Conference on Embedded Networked Sensor Systems,* Seoul, 2015.

[16] S. Ahmed and et al., "FMCW Radar Sensor Based Human Activity Recognition using Deep Learning," in *International Conference on Electronics, Information, and Communication,* Jeju, 2022.

[17] U. Ayvaz and et al., "Real-time human activity recognition using textile-based sensors," in *15th International Conference on Body Area Networks, BodyNets,* Tallinn, 2020.

[18] J. e. a. Patalas-Maliszewska, “Inertial Sensor-Based Sport Activity Advisory System Using,” *Sensors,* vol. 23, no. 3, 2023.

[19] J. Yang and et al., “Deep convolutional nerual networks on multichannel time series for human activity recognition,” in *Twenty-Fourth International Joint Conference on Artificial Intellifence,* Buenos Aires, 2015.

[20] N. Y. Hammerla and et al., “Deep, convolutional, and recurrent models for human activity recognition using wearables,” in *IJCAI International Joint Conference on Artificial Intelligence,* New York, 2016.

[21] W. Jiang and et al., “Human activity recognition using wearable sensors by deep convolutional neural networks,” in *MM 2015 - Proceedings of the 2015 ACM Multimedia Conference*, Brisbane, 2015.

[22] H. F. Nweke and et al., “Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges,” *Expert Systems with Applications,* vol. 105, pp. 233-261, 2018.

[23] M. A. O'Reilly and et al., “Technology in strength and conditioning tracking lower-limb exercises with wearable sensors,” *Journal of Strength and Conditioning Research,* vol. 31, no. 6, pp. 1726 - 1736, 2017.

[24] R. Huiying and et al., “Power System Event Classification and Localization Using a Convolutional Neural Network,” *Frontiers in Energy Research,* vol. 8, 2020.

[25] M. Wenner and et al., “Near-real-time automated classification of seismic signals of slope failures with continuous random forests,” *Natural Hazards and Earth System Sciences,* vol. 21, no. 1, p. 339–361, 2021.

[26] H. Ismail Fawaz and et al., “Deep learning for time series classification: a review,” *Data Mining and Knowledge Discovery,* vol. 33, p. pages917–963, 2019.

[27] J. Faouzi, “Deep learning for time series classification: a review,” *Ketan Kotecha. Machine Learning (Emerging Trends and Applications),* 2022.

[28] F. J. Ordóñez and et al., “Deep convolutional and lSTM recurrent neural networks for multimodal wearable activity recognition,” *Sensors (Basel),* vol. 16, no. 1, 2016.

7 Appendix

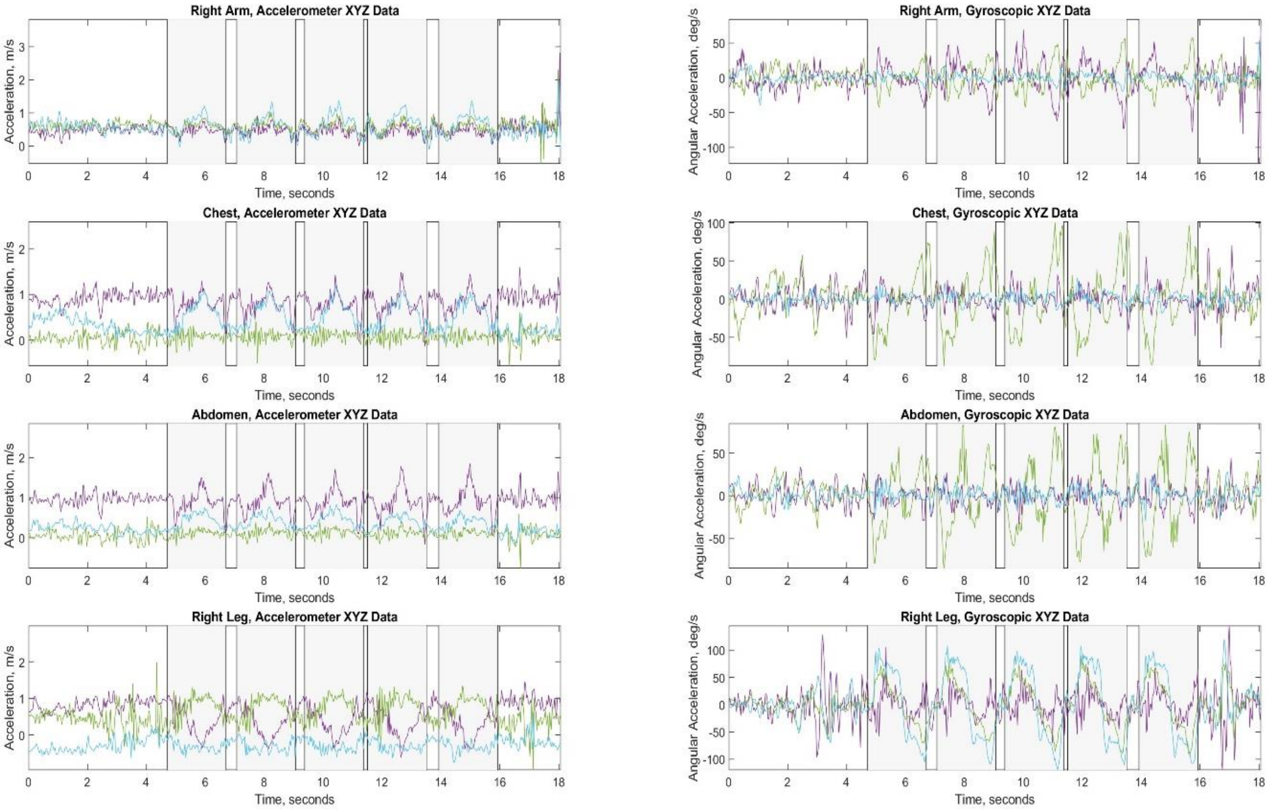


Figure 7.2 Example of sensor data segmentation GUI with reps segmented into their respective windows given by grey boxes.

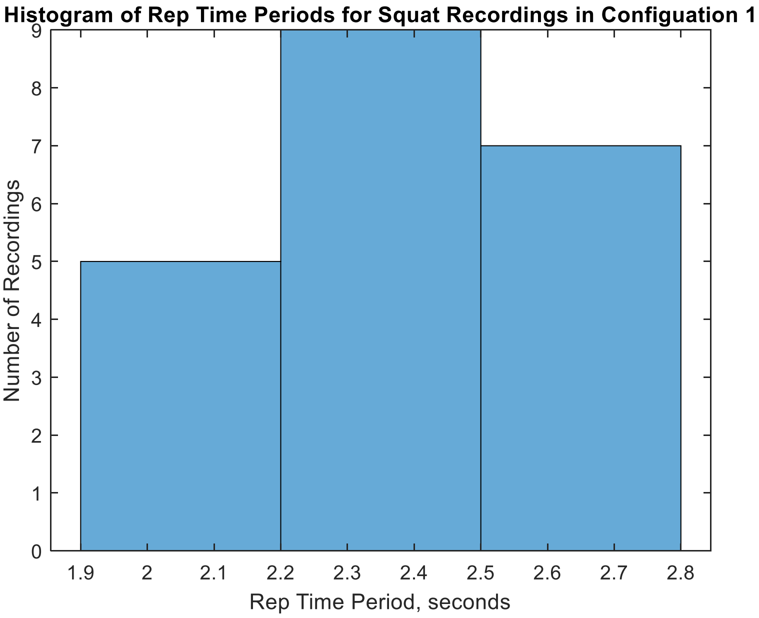


Figure 7.1 Histogram of rep time periods for squats in configuration 1

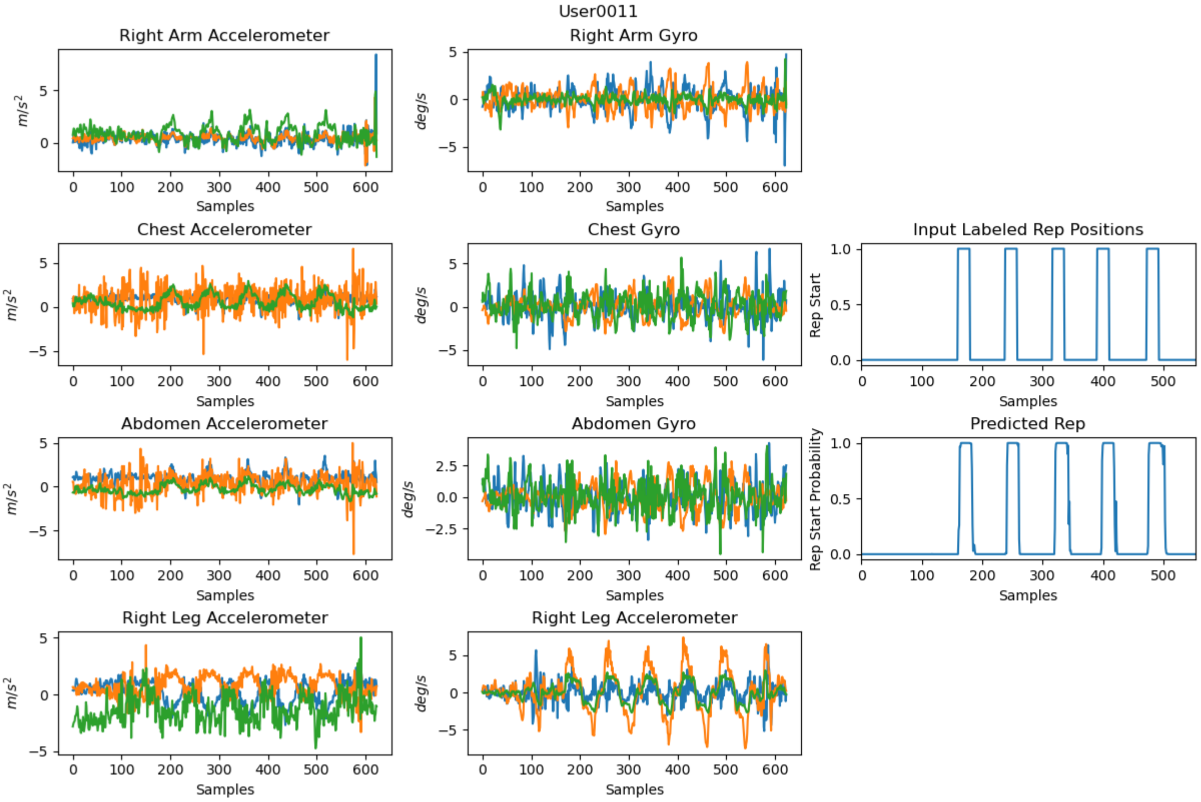


Figure 7.3 Sample plot of results of Binary Squat Classifier for Configuration 1 including input time series sensor data and labels.

Table 7.1 Description of parameters for the CNN classifier models.

CNN Layer	Parameter Values
Convolutional layer 1	Number of filters = 32 Kernal size = 7 Activation function = ReLU
Max pooling layer	Pool size = 2
Convolutional layer 2	Number of filters = 32 Kernal size = 7 Activation function = ReLU
Max pooling layer	Pool size = 2
Convolutional layer 3	Number of filters = 32 Kernal size = 7 Activation function = ReLU
Max pooling layer	Pool size = 2
Dense fully connected layer	Number of neurons = 32 Activation function = ReLU
Output layer	(binary/multiclass) Number of outputs = (1 / 4) Activation function = (sigmoid / softmax)

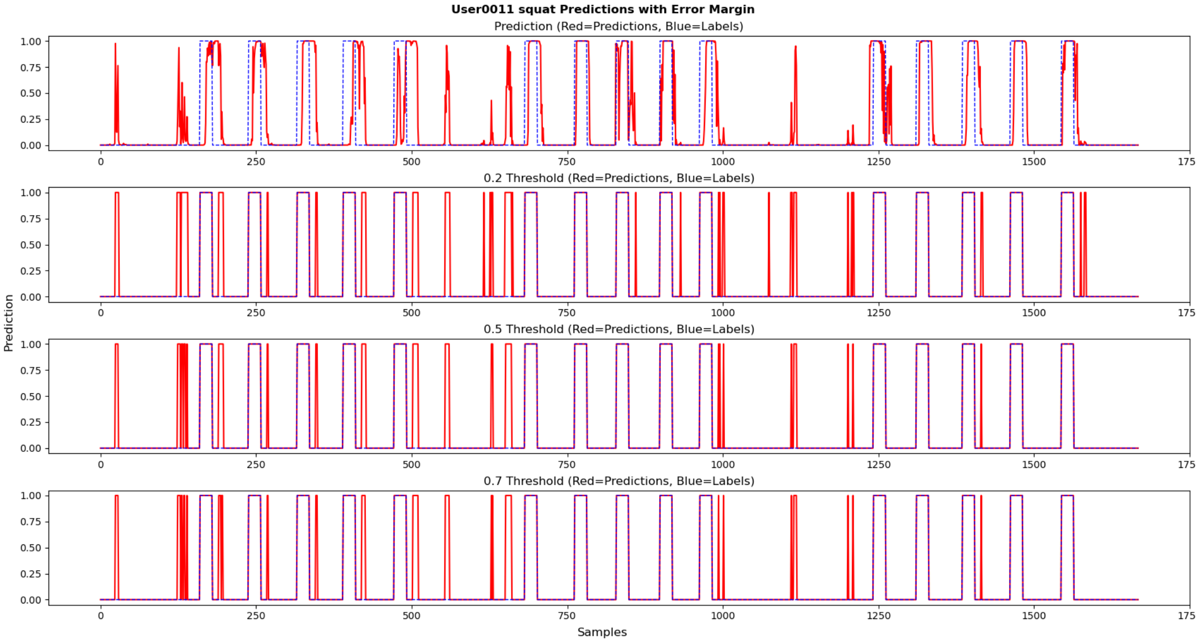


Figure 7.4 Example of the error margin being applied to the prediction signal with varying thresholds applied to the raw signal.

Table 7.2 Performance metrics of the multiclass output classifier for the chest and abdomen sensor combination: including ROC AUC values and balanced accuracy scores for each 5-fold test dataset with summary averages.

5-Fold Dataset Number	ROC AUC Values				Balanced Accuracy Score
	None	Squat	Clean	Deadlift	
1	0.86	0.79	0.55	0.44	0.38
2	0.96	0.95	0.96	0.96	0.86
3	0.93	0.95	0.46	0.92	0.79
4	0.86	0.96	0.73	0.63	0.34
5	0.89	0.48	0.85	0.91	0.72
6	0.85	0.76	0.72	0.49	0.49
Average	**0.89**	**0.82**	**0.71**	**0.72**	**0.60**

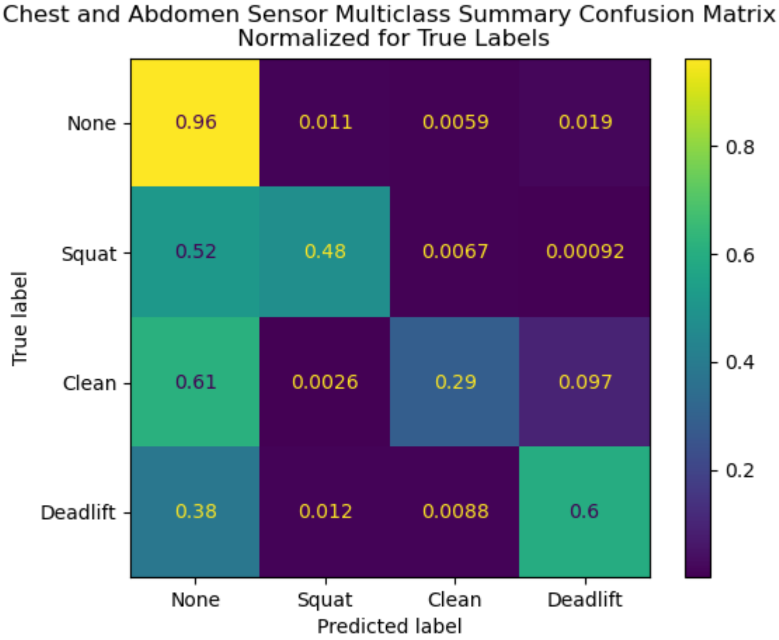


Figure 7.5 Summary confusion matrix for multiclass output classifier trained on all exercises for the chest and abdomen sensor combination. True labels have been normalized to 1 to provide better comparison between exercises. Note that colour grading is normalized to the max value in the matrix.

Table 7.3 Performance metrics of the multiclass output classifier for the abdomen and right leg sensor combination: including ROC AUC values and balanced accuracy scores for each 5-fold test dataset with summary averages.

5-Fold Dataset Number	ROC AUC Values				Balanced Accuracy Score
	None	Squat	Clean	Deadlift	
1	0.76	0.78	0.62	0.68	0.37
2	0.98	1.00	0.99	0.95	0.87
3	0.81	0.62	0.50	0.91	0.58
4		0.82	0.79		0.54
5	0.84	0.48	0.76	0.88	0.64
6	0.92	0.91	0.79	0.45	0.70
Average	**0.86**	**0.77**	**0.74**	**0.77**	**0.62**

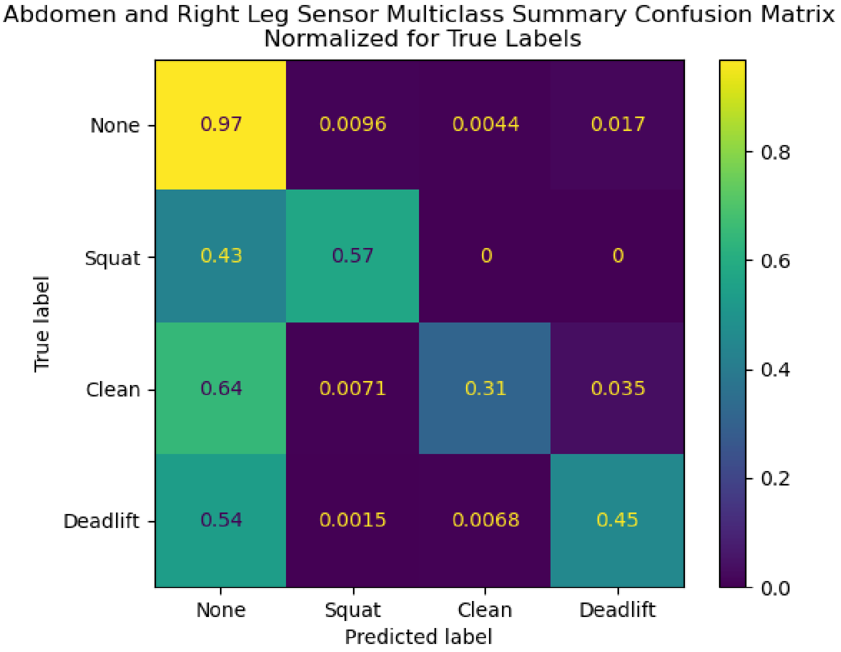


Figure 7.6 Summary confusion matrix for multiclass output classifier trained on all exercises for the abdomen and right leg sensor combination. True labels have been normalized to 1 to provide better comparison between exercises. Note that colour grading is normalized.

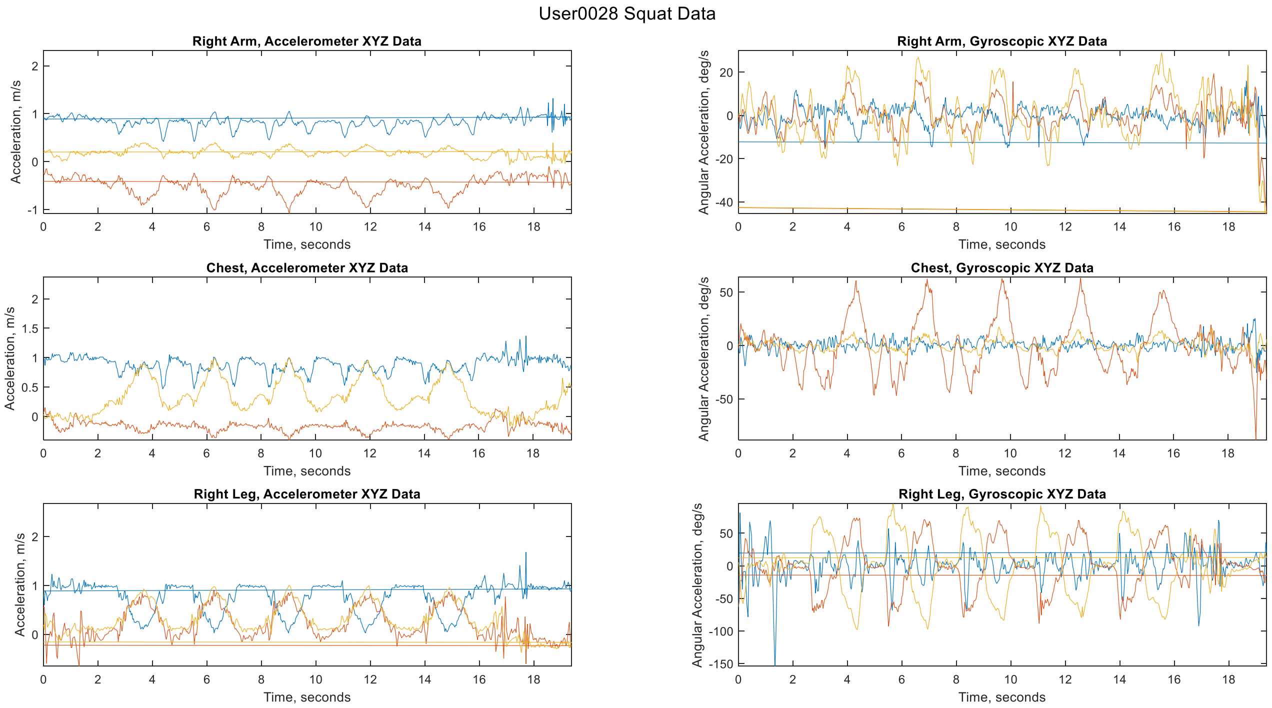


Figure 7.7 User0028 squat recording mislabelled as deadlift by right arm, chest, and right leg sensor combination multiclass classifier.

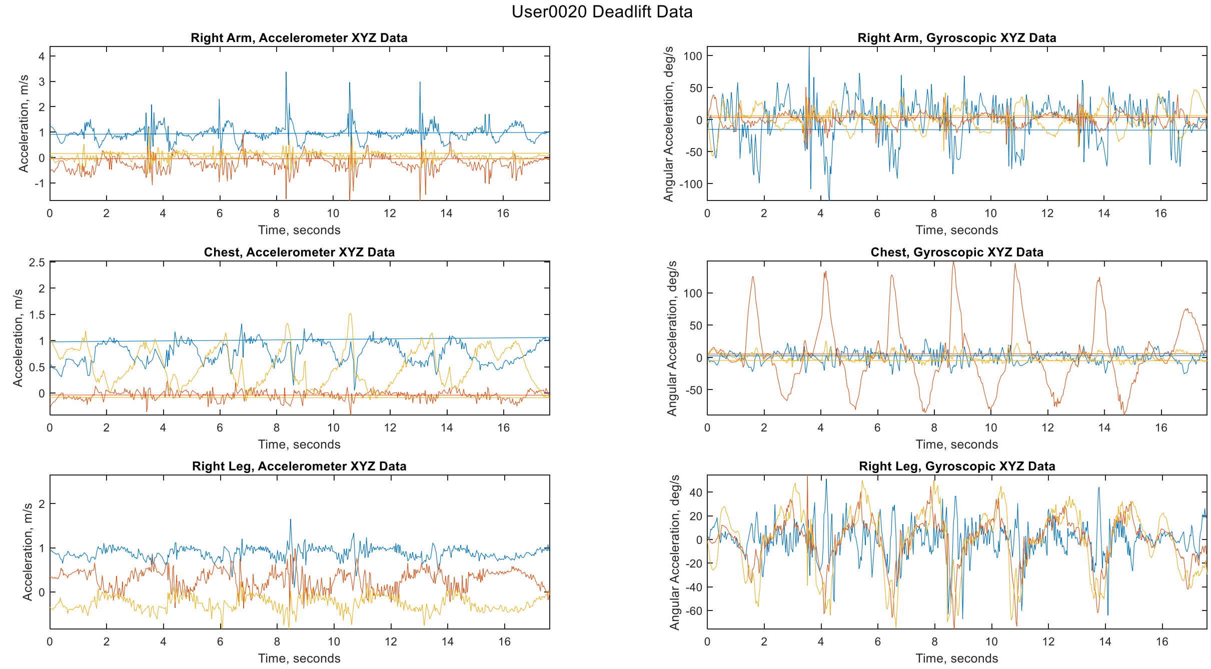


Figure 7.8 User0020 deadlift recording for comparison with figure 7.7.

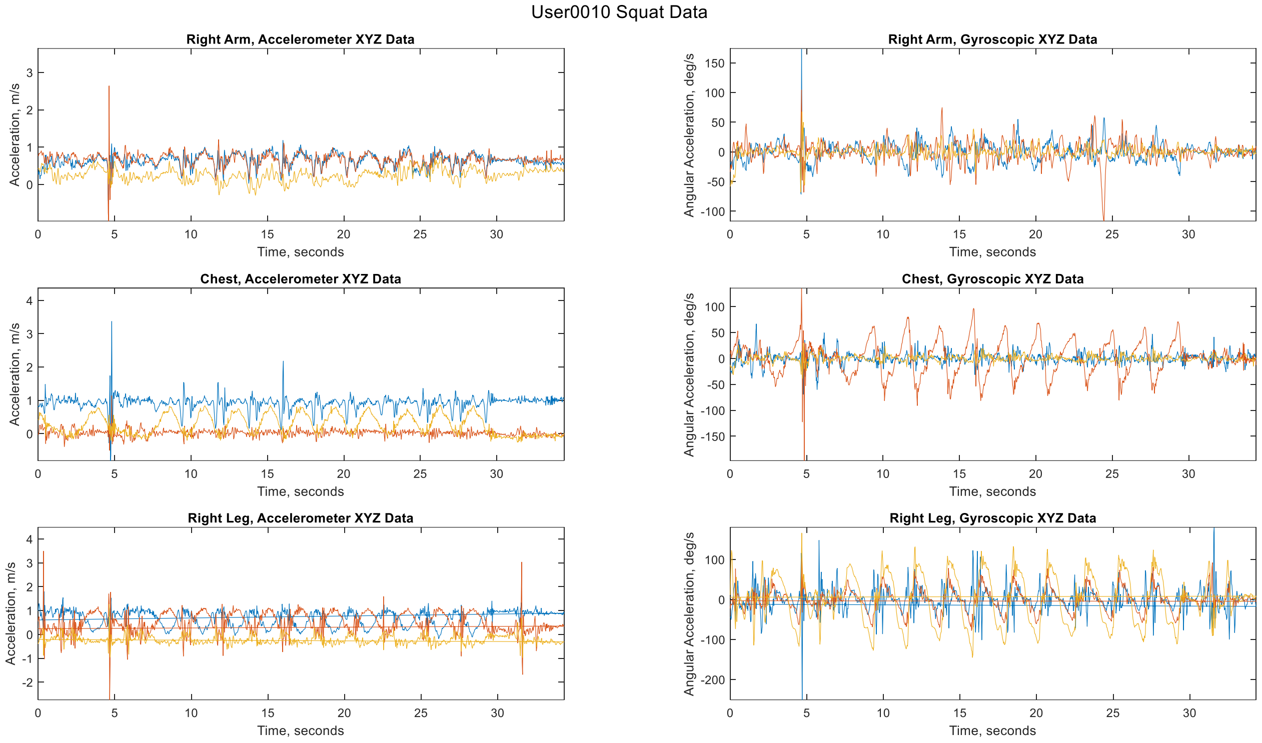


Figure 7.9 User0010 squat recording for comparison with figure 7.7.

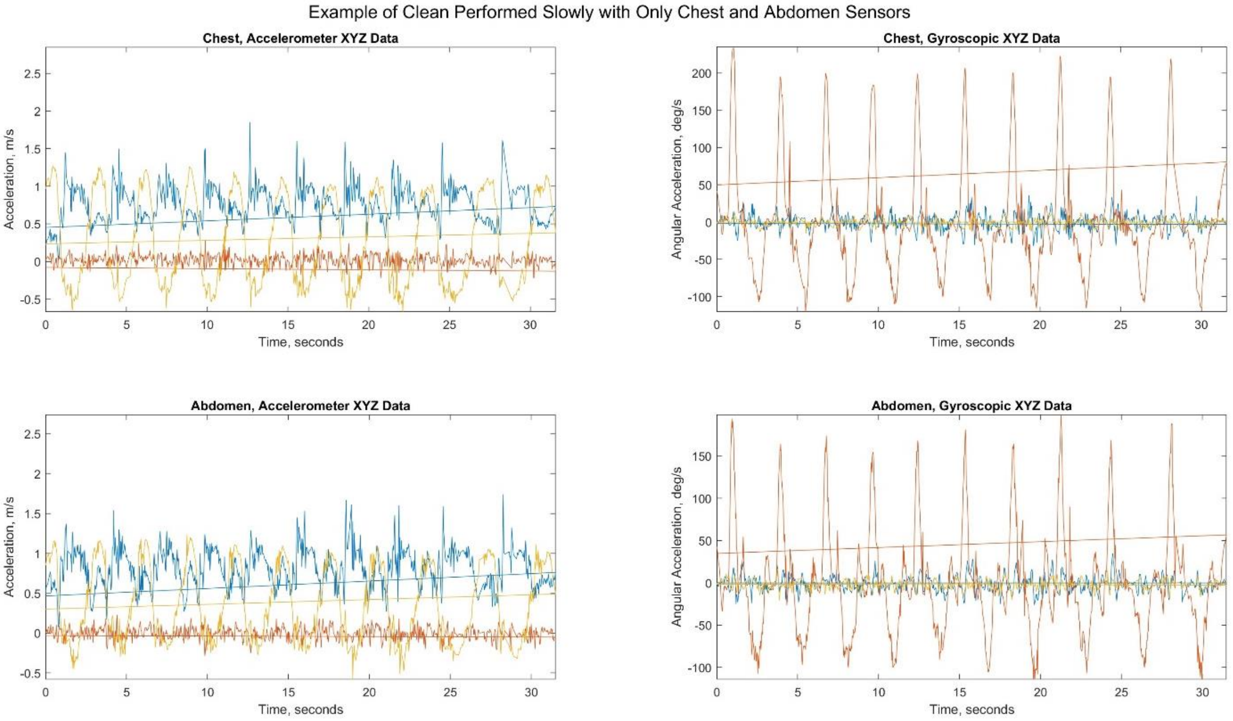


Figure 7.10 Clean performed slowly by an inexperienced participant resulting in mislabelling as deadlift for chest and abdomen sensor combination model.

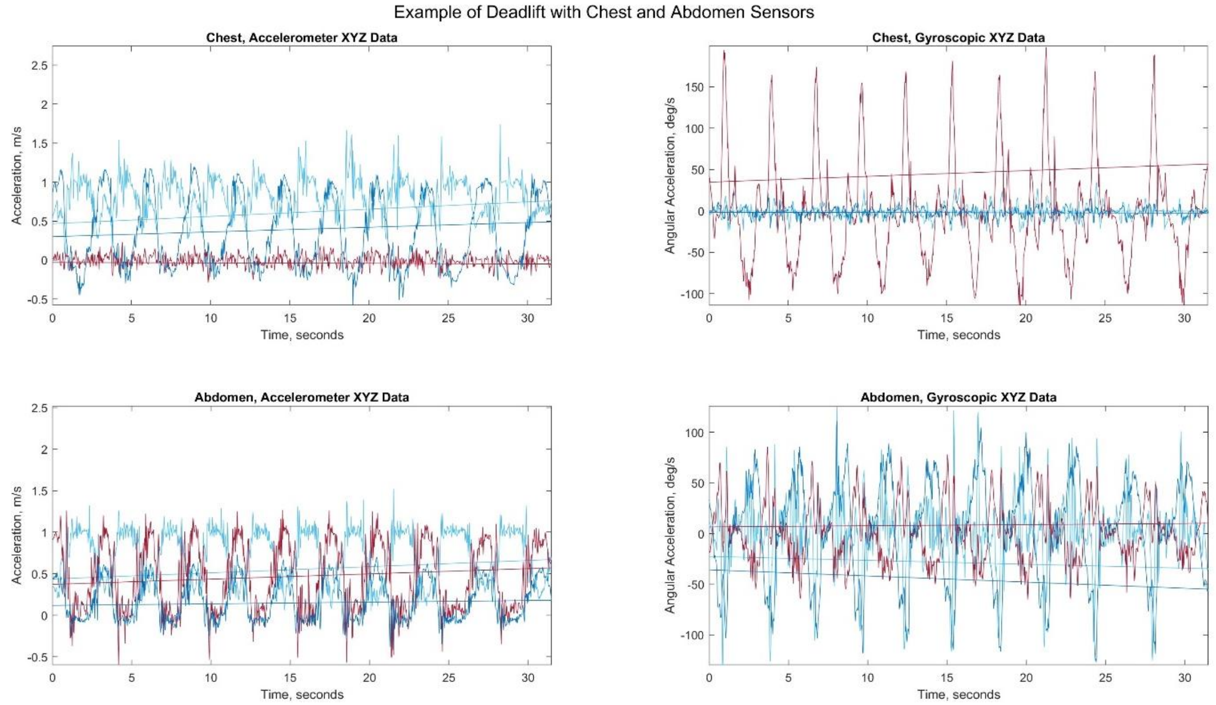


Figure 7.11 Example of deadlift recording for comparison with Figure 7.10