A Comparison of Machine Learning Techniques: Classifying Surfer Motion Using Smartfin IMU Data

Purisa Jasmine Simmons

Advisor: Prof. Ryan Kastner

Undergraduate Honors Thesis
Computer Science and Engineering
University of California San Diego

March 2019

Acknowledgements

I would like to thank my advisors, Professor Ryan Kastner and Professor Curt Schugers, for guiding me in this project and for mentoring the Engineers for Exploration group at UC San Diego. They constantly provide students with interdisciplinary research opportunities that span many different scientific disciplines, always made themselves available and accessible to me when I had inquiries, and always encouraged me to pursue research and a higher education. I am incredibly grateful for having had the opportunity to meet and work with both of them.

I would also like to thank a few more people who have been instrumental in my path to producing this work. Dr. Phil Bresnahan, the lead research engineer of the Smartfin project, taught me that science is not done from behind a desk; this was the first project in which I went out into the ocean every week to collect data on my surfboard, and I will never forget this experience. Professor Christine Alvarado, who gave me my first introduction to research in the CSE department under the Early Research Scholars Program and constantly inspires me with her inclusive approach to teaching the CSE curriculum. Professor Tajana Rosing, who had lunch with me, gave me another research experience, and maybe unknowingly convinced me to pursue a Master's degree. Eric Lo, our resident research and development engineer, answered all of my technical and otherwise computer-related questions.

Finally, I would like to acknowledge my family, especially my parents, for supporting me and giving me the opportunity to pursue a college education. It is a privilege that I have reflected on greatly, and continue to treasure dearly.

Abstract

It is challenging for scientists to collect oceanographic data in nearshore environments because the high amount of wave energy present in these areas makes it difficult to deploy autonomous sensors. To address this issue, scientists at Scripps Institute of Oceanography have developed the Smartfin--a surfboard fin with wireless embedded sensors--since the surf zone encompasses the same coastal regions that these scientists are interested in studying. However, collecting oceanic data with the Smartfin poses new problems, as surfer movement may now potentially bias the information being collected. Therefore, this research explores the use of robust machine learning algorithms to determine how a surfer is moving on his or her board. This research is based on surf sessions recorded around the University of California San Diego's campus, and includes the following locations: La Jolla Shores (La Jolla, CA), Scripps Pier (La Jolla, CA), and K-38 (Baja California, Mexico). The machine learning models experimented with in this thesis include: logistic regression, multilayer perceptron, and support vector machines with linear and non-linear kernels. Through the use of feature engineering on a dataset taken from a single surf session, we were able to predict the following classes: floating, paddling, and surfing with 86% accuracy. However, our mean classification accuracy drops to 70% when training and testing subsets are taken from the combined dataset of all surf sessions, where oceanic conditions within the dataset become more variable. Future work will need to be done to determine whether this level of classification accuracy is adequate for oceanographers who are using the Smartfin to collect data for scientific purposes.

Table of Contents

Chapter 1: Introduction

1.1 Motivation

Understanding our oceans is incredibly important as our oceans produce most of the world's oxygen, store more carbon dioxide than our atmosphere, and regulate earth's climate. However, our oceans are severely undersampled as we rely on scientific measurements from infrequently spaced buoys, piers, and sub-surface moorings. Nearshore environments, like the surf zone, are particularly difficult to collect data in because high energy wave dynamics often break expensive scientific equipment. In order to increase the spatial density of oceanographic measurements, researchers at Scripps Institute of Oceanography have begun developing the Smartfin, a surfboard fin with embedded sensors capable of measuring multiple ocean parameters in nearshore environments. There are multiple sensors contained within the Smartfin, including a temperature sensor, a GPS, and a 9-axis Inertial Measurement Unit (IMU). The IMU alone consists of an accelerometer, gyroscope, and magnetometer. This project has the potential to vastly increase the spatial density of oceanographic measurements by making ocean data collected by the Smartfin's sensors available to the worldwide scientific community. While the Smartfin's hardware has been thoroughly tested by scientists at Scripps, very little data processing has been done. My proposed thesis involves the creation of a processing framework for meaningful analysis of data collected by the Smartfin.

1.2 Purpose

Specifically, my research project focuses on using machine learning techniques on the Smartfin's IMU data to classify surfer motion. We would like to be able to label sections of time-series surfer data as, for example, floating, paddling, or surfing. This is important because we only want to capture scientific information about ocean conditions (such as wave height, period, frequency, and ocean temperature) when the surfer is floating relatively still and thus acting as a buoy. When the surfer is floating relatively still, the IMU can capture more accurate information related to how current wave conditions are affecting the surfer, which will allow us to analyze current wave conditions and create wave climate models in the future. Additionally, it will allow us to record accurate temperature and dissolved oxygen readings in a particular location, compared to when the surfer is moving. When a surfer is not stationary, the IMU will capture noisy information that we do not necessarily want to include in our wave or ocean models, thus we would like to be able to distinguish these movements from each other from our

IMU data. Essentially, this will allow us to collect more accurate data about the ocean in such a way that surfing does not bias the information that we are collecting.

Another reason why we are interested in classifying surfer motion from IMU data is because we would like to encourage surfers to upload their data to the Smartfin website after surfing with the Smartfin. If we provide surfers with surf statistics that they care about, for example: how many waves were caught during a particular session or the distance paddled during a particular session, they may be more likely to upload their Smartfin sensor data. In this way, we are attempting to ensure that surfers continue to make Smartfin data available online to the worldwide scientific community.

1.3 Research Questions

- Can we accurately perform multiclass classification on simple surfer movements using the Smartfin's IMU data?
- Can we accurately perform multiclass classification on more nuanced, or complex, surfer movements using the Smartfin's IMU data?
- Does the inclusion of additional input features via the process of feature engineering on our raw input features produce more accurate classification results for our chosen models?
- Overall, which of the chosen machine learning models produces the most accurate results in classifying surfer motion, and what pre-processing needs to be done to produce that accuracy?

1.4 Related Work

There is already a large body of research related to using machine learning techniques to perform accurate classification on IMU data [1-5]. There is also a growing field of research devoted to performing time-motion analysis of athletes in different sports to assess specific physiological demands of athletes in their given sport [6-9]. Most of this research concerns itself with analyzing training data in order to most closely model a team's training sessions after the competition-style of their given sport, however, there are very few reports that concern themselves with analyzing an individual athlete's personalized data using machine learning techniques [10], [11]. In part, this research could be a valuable step in collecting more data related to surfer motion and individual surfer technique, and could help develop a framework for processing and accurately classifying static and dynamic surfer movement with little human supervision.

1.5 Scope and Limitations

Since it is not feasible to cover every digital filter and every machine learning model in this thesis, the scope of this project is limited to the choice of two digital filters and four supervised machine learning models. This allows us to explore each implementation in detail, and fully discuss the results of the thesis. Additionally, the models will be trained on two distinct sets of labelled data in order to compare their performance. These sets will be denoted the "simple" set and the "complex" set. In the simple set, there are three possible labelled motions that can occur. These labels are: {surfing, paddling, floating}. In the "complex" set, there are more specific labels attributed to certain surfer motions. These labels are discussed later in more detail in Chapter 3.2 (Class Label Descriptions). Again, a limitation of this work is that it is not possible to capture every possible surfer motion that can occur, and thus a subset of all possible surfer motions is chosen to explore in depth. Only complex surfer motions that were contained within this dataset with a large enough number of samples will be included in the "complex motion" dataset to be used for prediction. Another main limitation is that this analysis can only be performed on the surf sessions contained on the Smartfin website which were recorded with a camera during the time of surfing. Therefore, this analysis will only be performed on eight surf sessions, taken from two surfers.

1.6 Thesis Overview

In this thesis, we will explore supervised machine learning techniques to determine the best approach for classifying surfer motion based on data received from the IMU sensor within the Smartfin. The thesis proceeds as follows: a brief theoretical overview of the filtering algorithms and machine learning models will be given in Chapter 2. Chapter 3 will describe the methodology, including the data collection process and all data pre-processing steps taken. Chapter 4 will present the results of the machine learning models, and Chapter 5 will discuss the results. Finally, Chapter 6 will conclude with a summary of what was found, in addition to describing possible future research work that can expand on this topic.

Chapter 2: Theoretical Background

2.1 Supervised Machine Learning

The main goal of this research project is to explore different machine learning models and compare them against one another in order to determine which most accurately classifies the data in a given dataset. Machine learning is a method of data analysis based on the principle that a system can learn to recognize patterns from data. Two main types of machine learning models include supervised learning and unsupervised learning. In supervised learning, a large set of input data and a ground truth label is fed to the model, and in this manner the supervised model is trained to find a pattern in the data to correctly map input data to its corresponding output label. On the other hand, unsupervised learning attempts to learn the intrinsic structure of the dataset without using provided labels. In this thesis, we will focus on the use of supervised machine learning techniques since we are trying to optimize the amount of correct predictions for surfer motion, given the corresponding motion's observed IMU signals.

2.2 Chosen Machine Learning Models

The supervised machine learning models used in this experiment include: Logistic Regression (LR), Multi-Layer Perceptron (MLP), a Support Vector Machines (SVM) with a linear kernel and an SVM with a non-linear kernel. We focused on these four models with the aim of determining which model yields the highest classification accuracy when trained and tested on our Smartfin IMU data. A high level overview of each model is explained below.

The logistic regression model is seen as a simple machine learning model which uses the sigmoid function to linearly separate input data into binary classes. In order to use this model, we must turn our classification problem into a series of simpler binary classifications for each surfer motion that we have labelled and are trying to predict. Probability estimates for each surfer movement will be computed under each of these binary models, and then the class with the largest probability estimate will be chosen as the prediction. This process is known as maximum likelihood estimation. In this way, a multinomial logistic regression model can be created from a series of binary logistic regression models.

The multi-layer perceptron model is an artificial neural network which transforms input data using a series of learned non-linear transformations. The purpose of these transformations is to project the input data onto a space where it then becomes linearly separable. These non-linear transformations are performed in intermediate layers known as hidden layers. The sigmoid function (also known as the logistic function) is a common activation function used by MLP hidden layers; therefore some MLP models can be viewed as logistic regression classifiers after performing non-linear transformations.

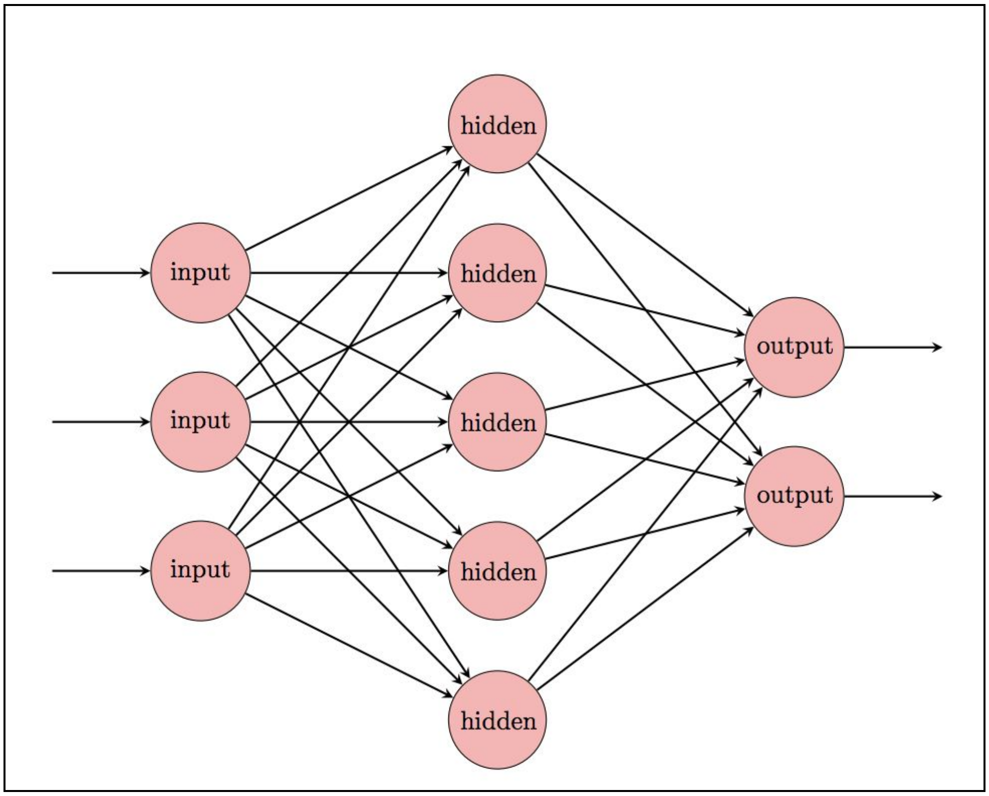

Figure 1. *Illustration of a multi-layer perceptron model, taken from [12].*

Support vector machines attempt to find the optimal hyperplane that best separates the labelled classes in an n-dimensional space, where n is the number of features. The hyperplane is computed from the data point classes in the training set, and is used to determine the class of each instance in the test set. While the logistic regression model optimizes the log likelihood function, SVM's optimal hyperplane tries to find the maximal separation between classes. An SVM model with a linear kernel generally performs comparably to the logistic regression model, but is less susceptible to outliers. Therefore, we will use both an SVM model with a non-linear kernel in addition to an SVM with a linear kernel. The difference between the linear kernel and the non-linear kernel is that a non-linear kernel, like a Radial Basis Function (RBF) performs better when the data is not linearly separable.

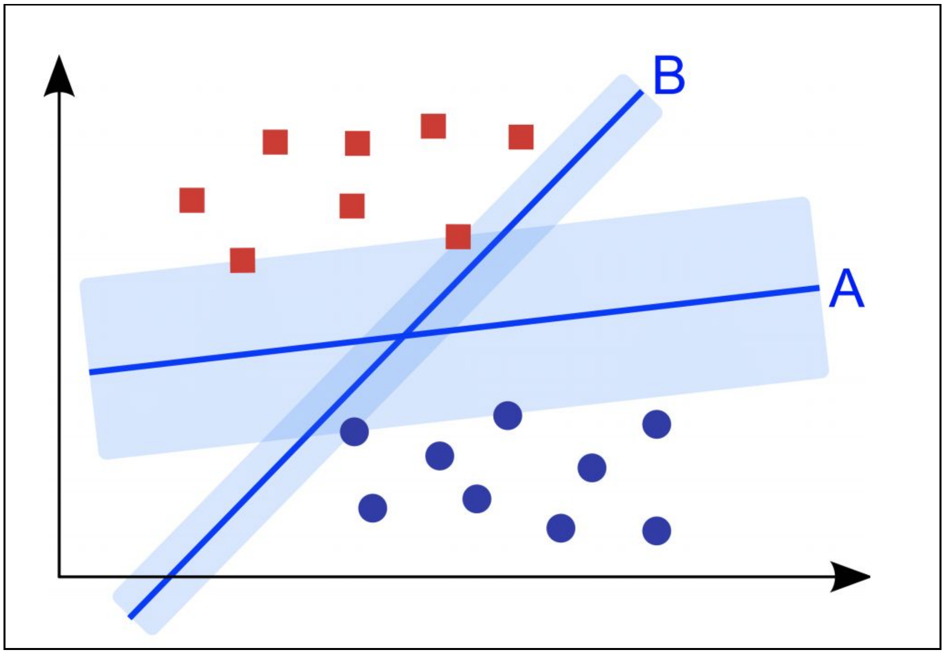

Figure 2. *Example of a SVM model linearly separating a dataset into two classes, with their separation margins highlighted, taken from [12].*

2.3 Evaluation of Models

Each machine learning model that is implemented will be evaluated based on its prediction performance. One method of determining prediction performance is to compute the classification accuracy of the model. Classification accuracy is defined as the number of correct predictions divided by the total number of predictions made. However, if 90% of the labels are "floating" and 10% of the labels are "not floating" in our dataset, our classifier could simply predict "floating" every time and be 90% correct. Therefore, this metric gives us a false sense of accuracy in the case of imbalanced datasets. In order to combat this, our datasets are balanced using both oversampling and downsampling techniques, which are discussed in Chapter 3.3.

The simplest way to compute classification accuracy is to split the entire dataset into two subsets: training and testing subsets. The training set is used to compute the parameters of the classifier, while the testing set is used to determine how well the classifier performs on unseen data. Another technique, known as k-fold cross validation, can be used to generalize the estimated classification accuracy of the model in practice. K-fold cross validation is easy to implement, and generally has lower bias than other ways of determining a model's prediction performance. This technique partitions the dataset into k sets, and uses k-1 of those sets for training the model and the remaining set for validating the model. This process is performed k times, such that the model is evaluated on each partition of the data. The resulting validation scores are then averaged in order to give a better estimate of the model's overall performance. This provides a more accurate estimation of the model's prediction performance than the method described earlier, and will be used to evaluate each of our models in the subsequent experiments.

Additionally, Confusion matrices can be used to visually evaluate how well each of the models predicts a specific class, with respect to other classes. Specifically, they give us a sense of which classes the model is confusing for other classes, and the extent to which that confusion occurs. Confusion matrices for each of the models will be included in the appendix section for some of the experiments.

Chapter 3: Methodology

3.1 Data Collection

Experimental data is collected in the same way that it would be collected by users of the Smartfin; the Smartfin captures information related to both the ocean environment and surfer motion while surfing. The surfer in this experiment is using a 9’ longboard. In order to create ground-truth labels for the Smartfin’s IMU signals, a waterproof sports camera is attached to the nose of the surfboard, facing the surfer. The Smartfin used in this experiment records data from the ocean environment at a rate of 5 Hz. After the surf session, the data is downloaded from the Smartfin via bluetooth and synced with the labelled video footage. Portions of the video footage are labelled in one second increments and the labels from the video footage are then applied to the corresponding IMU data from the Smartfin.

ride_id	UTC	IMU A1	IMU A2	IMU A3	IMU G1	IMU G2	IMU G3	IMU M1	IMU M2	IMU M3	IMU V1	IMU V2	IMU V3	simple_label
15629	**2018-10-31 19:59:02.739**	-0.899627	0.066756	0.861345	5.121951	2.682927	6.829268	-11.0	-155.0	-17.0	-0.271164	-2.567850	0.208956	FLOATING
	2018-10-31 19:59:02.937	-0.918768	0.066756	0.746499	7.073171	1.463415	-0.853659	-17.0	-161.0	-1.0	-0.245643	-2.519998	0.202576	FLOATING
	2018-10-31 19:59:03.135	-1.052755	-0.373487	0.842204	-5.121951	-0.121951	-7.439024	-6.0	-154.0	-8.0	-0.239262	-2.519998	0.204171	FLOATING
	2018-10-31 19:59:03.333	-1.129319	-0.985999	0.880486	-9.024390	-0.365854	-4.146341	-4.0	-148.0	-8.0	-0.258404	-2.607728	0.205766	FLOATING
	2018-10-31 19:59:03.564	-1.225024	-0.641461	1.186742	-5.975610	-1.951220	3.780488	-4.0	-154.0	-28.0	-0.283925	-2.666746	0.242453	FLOATING
	2018-10-31 19:59:03.762	-0.899627	-0.430910	1.339870	-8.292683	-2.317073	7.804878	-7.0	-151.0	-33.0	-0.271164	-2.671531	0.283925	FLOATING

Figure 3. *Dataframe produced by the data collection process, the surfer is floating in the first six instances of this dataframe.*

3.2 Class Label Descriptions

We are experimenting with two different sets of labels. The first set of labels we will denote as the “simple” labels; they include the following surfer motions: {surfing, paddling, floating}. The second set of labels will be denoted as the “complex” labels. They include the following surfer motions, listed in the table below. For the “complex” labels, we decided to label basic surfer motions that we could easily determine from our surf session footage. A description of each class label is included below to familiarize the reader with our class label choices.

Table 1. *Class label descriptions.*

Class Label	**Description**
Floating	Surfer is sitting still on surfboard, floating in water.
Paddling into waves	Surfer is paddling directly into the waves, away from shore, towards the line-up.
Paddling for a wave	Surfer is paddling for a wave, attempting to catch it.
Paddling for position	Surfer is paddling to change their position in the line-up (but not paddling for a wave or into waves).
Surfing	Surfer has caught a wave and is standing up on their surfboard.
Turning Left	While the surfer is floating and turning left.
Turning Right	While the surfer is floating and turning right.
Wipe-out	Denotes that a wave unexpectedly knocked the surfer off of their surfboard while they were surfing.
Pop-up	Surfer motion that occurs between paddling for a wave and actually surfing. The surfer moves quickly from the paddling position, to a push-up position, to standing on their surfboard.
Push-off	Denotes that the surfer was walking in the water alongside their surfboard, and then pushed off the ocean floor before beginning to paddle into waves.
Pull-back leash	The surfer pulls back on the surfboard leash to retrieve their surfboard.
Walking in water	The surfer is walking in the water with their surfboard.
Sit-up	Surfer motion that occurs between paddling and floating. The surfer moves from a position where they are laying on the surfboard to one where they are sitting up on the surfboard.
Lay-down	Surfer motion that occurs between floating and paddling. The surfer moves from a position where they are sitting on the surfboard to one where they are laying down on the surfboard.
Step-off	Denotes that the surfer was surfing, then stepped off of their surfboard to safely dismount from the wave that was being caught.
Sit-back	Denotes that the surfer was paddling for a wave, decided they were not in a good position to catch the wave, and sat back on their surfboard so that the wave would not carry them forwards.
Off-board	Denotes that the surfer is no longer on their board.

3.3 Pre-processing and Balancing Datasets

The data is pre-processed so that any one time only one label is true. For example, when a surfer is both floating and turning, the surfer motion is labelled as “turning” until the surfer stops their turn. Once the turn stops, the “floating” label resumes. Data from the very beginning of the surf session, before the Smartfin and camera are synced, is deleted from the dataset. Similarly, data from the end of the surf session, when the surfer is exiting the water and after the surfer has exited the water, is deleted from the dataset. For some models (such as our SVM and MLP models) the input features are scaled from raw units to standard normally distributed data. This is performed by removing the mean value from each feature, then scaling by dividing non-constant features by their standard deviation. This is a common requirement for many machine learning estimators.

When the class labels in our datasets are not evenly distributed, it becomes necessary to balance our dataset. If the dataset was left imbalanced, than our machine learning models would likely be biased towards overwhelmingly predicting the majority class. As we noted before, in a regular surf session it is much more likely that a surfer is floating than paddling, and more likely paddling than surfing. This same trend occurs in our experimental surf sessions, where instances of the floating class occur overwhelmingly more than the paddling class, and instances of the surfing class occur very infrequently. A distribution of the data for each of the experiments is included in the appendix section.

In order to avoid biased models, we use a synthetic minority over-sampling technique known as SMOTE to balance our training set. A trivial way of balancing our dataset would be to duplicate all instances of our surfing and paddling data, so that there are the same number of instances in each class. However, this is problematic because the machine learning models’ testing data may become contaminated with duplicated data from the training dataset. Using SMOTE allows us to synthetically create new data samples from real ones. This allows us to equally distribute our data samples across the labelled classes, without biasing or contaminating our testing set. SMOTE oversampling is only performed on the training sets, while the testing sets are kept clean and untampered with. In some cases, we will need to downsample to the minority class, rather than upsample to the majority class, in order to ensure that our models are still computationally efficient and do not take too long to run. Details on the specific downsampling method used are included in the appendix sections.

3.4 Additional Features

The raw signal obtained by the Smartfin's IMU only contains accelerometer, gyroscope, and magnetometer data. Therefore, without feature engineering, our models are only given information related to acceleration, angular velocity, and compass heading. However, we can integrate the acceleration data in order to produce an estimate for the instantaneous velocity at a given time. We believe that the inclusion of velocity as an additional feature will improve our models' prediction performance because it will help each model separate classes. For example, we can generally expect to have a higher instantaneous velocity when surfing, compared to paddling or floating. The same argument cannot be made for the acceleration feature.

Since there is no perfect technique for capturing information about the world around us, there will naturally be noise present in addition to the true underlying signal in the IMU data that we have collected. The two main sources of this noise are: errors introduced by measurement tools, and random errors introduced by the data collection process. We would like to mitigate as much random noise as possible in our data so that when we later perform inference on our dataset, it mainly consists of the true signal that we are trying to classify. The Butterworth Filter is commonly used for this purpose; hence, we would also like to experiment with its use on our accelerometer data as an additional feature in our dataset. In a similar fashion, we will perform a moving window average filter in order to reduce noise present in the gyroscope data.

3.5 Parameter Tuning

Parameter tuning is the process of optimizing model parameters and hyperparameters in order to improve the model's prediction performance. While the LR model will not need to undergo a parameter tuning process, the MLP and SVM models will generally benefit from having their parameters tuned. Parameter tuning is thus performed on the MLP and SVM models in our experiments when there is sufficient computational resources to handle the parameter tuning process on the given experimental dataset. More information regarding the specific parameter tuning process used on a given model in a given experiment will be included in the appendix.

3.6 Overview of Experiments

The following experiments will allow us to determine an accurate way of using machine learning techniques to determine surfer motion from the Smartfin's IMU data. The Smartfin IMU data for these experiments is collected in a realistic way that mimics how surfers will actually be using the Smartfin in real-world conditions. Eight surf sessions are conducted over the months of

October-March at two beach breaks in San Diego: Scripps and La Jolla Shores, and one point break in Mexico: K-38. The average wave heights reported for these surf sessions range from 2-5 ft. tall.

In the first experiment (Results Section 4.1), we are interested in how accurately a simple machine learning model predicts each of the individual classes that we have labelled. Therefore, multiple binary logistic regression models will be used to determine whether we believe that the individual classes that we have initially chosen to label would be good classes to continue with in further experiments. In the case of these binary models, we are interested in how accurately each machine learning model predicts an individual feature. Hence the model will be given a positive value for the class that is trying to predict, and a negative value for all other classes. This will give us an idea of how much the features for that individual class differs from the features of all other classes, i.e. how unique the features of that class is. This is often described as a one-vs-all type of machine learning classification task. If the binary machine learning model is able to predict each of the labels in the one-vs-all sense accurately well, we can be more confident that it will accurately predict those labels in the multinomial case.

Our next experiments compare the results of our four different multinomial machine learning models to determine which model performs best in a given scenario. In the first scenario (Results Section 4.2) we are testing how well each model performs when given only the raw IMU signal as input, and is tested on a single surf session using the "simple" classes consisting of {floating, paddling, surfing}. Using only the raw IMU signal gives us a baseline which we will try to improve in subsequent experiments via a process of feature engineering. Using a single surf session allows us to examine how well we can predict the motions from the "simple" dataset when oceanic conditions are held somewhat constant (i.e. surfing in the same spot while tide/wind/wave conditions do not change drastically).

As previously mentioned, feature engineering is performed (Results Section 4.3) to determine how the inclusion of additional input features, which are derived from our raw input features, may improve our classification accuracy. In particular, we examine how the inclusion of the following six features: instantaneous velocity in the x, y, and z directions and average change in angular velocity over a window of past data points in the x, y, and z directions affects the classification accuracy of each model. The same experimental set up from the previous experiment is used: a single surf session using the {floating, paddling, surfing} classes to be predicted.

The next three experiments (Results Section 4.4, 4.5, and 4.6) similarly use additional features present the previous experiment (instantaneous velocity and change in angular velocity), but now uses multiple surf sessions to learn from, rather than one. This allows us to test the robustness of our models; given multiple surf sessions which were taken from multiple locations at different times throughout the year, where oceanic conditions are now much more variable. The main difference between the experiments in Sections 4.4 and 4.5 is the way that the dataset is balanced; in Section 4.4 it is balanced using an upsampling technique (as before) while in Section 4.5 it is downsampled (as it is in Section 4.6). Additionally, the experiment in Results Section 4.4 is performed with and without the use of a Butterworth Filter[1].

The results from the experiments in Sections 4.4 and 4.5 will illuminate how well our models perform when predicting the "simple" motion classes (Results Section 4.5) versus the "complex" motion classes (Results Section 4.6) with the aforementioned experimental set ups.

---

[1] For consistency (and to avoid confusion) in the experimental setups the results of the Butterworth Filter are not included in the table in Results Section 4.4 but are included in the Appendix.

Chapter 4: Results

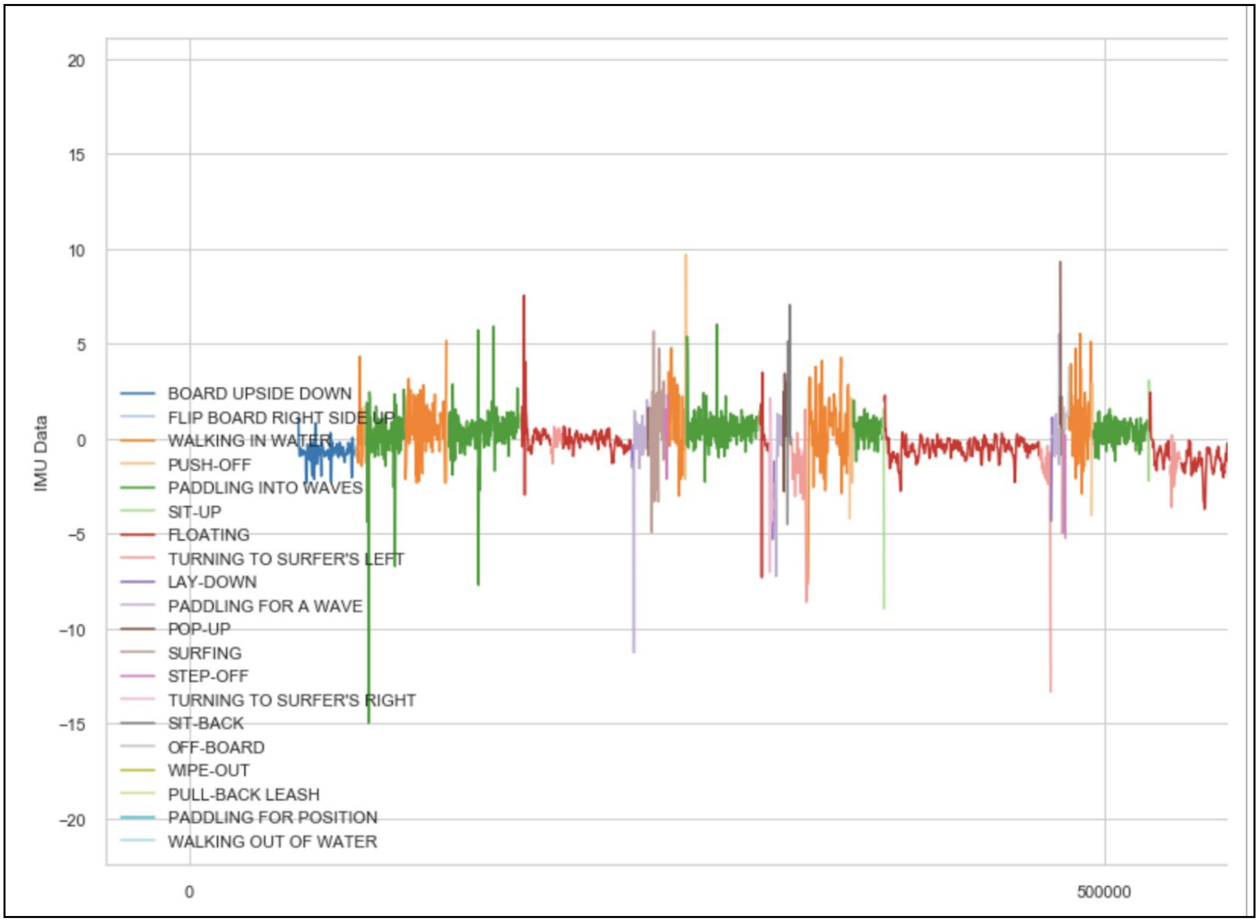

Figure 4. *Acceleration in the forward/backward direction is projected onto the y-axis with m/s^2 units, time is projected on the x-axis in milliseconds, and class labels are color coded.*

4.1 Results: Binary Logistic Regression

Table 2. One-vs-All Binary Logistic Regression Classification Accuracies[2]

Class Label	Classification Accuracy
Floating	83.97%
Paddling into waves	69.27%
Paddling for a wave	69.65%
Paddling for position	81.20%
Surfing	62.54%
Turning Left	69.64%
Turning Right	70.27%
Wipe-out	68.03%
Pop-up	68.93%
Push-off	68.22%
Pull-back leash	78.67%
Walking in water	67.01%
Sit-up	68.86%
Lay-down	65.81%
Step-off	66.03%
Sit-back	81.00%
Off-board	62.99%

The above experimental results only included raw input features from the raw 9-axis IMU data.

[2] The distribution of each class label in the single surf session used in this experiment (and the subsequent single surf session experiments) can be found in the appendix. Each class was balanced using the SMOTE technique to equal the majority class, which consisted of the number of labels not in that class. Classes that were labelled (and may be present in Figure 1) but which did not have a representative sample were removed from the table and were not used in subsequent experiments.

4.2 Results: Raw Signal on "Simple" Classes from Single Surf Session

The following experimental results only included raw input features from the raw 9-axis IMU data, and attempted to predict the classes: {floating, paddling, surfing}.

Table 3. *A summary of the mean classification accuracy scores produced by performing a 10-Fold Cross Validation on each model in experiment 2.*

Model	Mean Classification Accuracy Percentage	Mean Accuracy Score ± Standard Deviation
Multinomial LR	70%	0.708 ± 0.030
Multinomial MLP	70%	0.698 ± 0.15
Multinomial Linear SVM	70%	0.697 ± 0.136
Multinomial Non-Linear SVM	68%	0.682 ± 0.138

4.3 Results: Additional Features on "Simple" Classes from Single Surf Session

The following experimental results included six additional input features calculated from the raw 9-axis IMU data (in addition to the raw input features), which were: instantaneous velocity in the x, y, and z directions and average change in angular velocity over a window of past data points in the x, y, and z directions. This experiment attempted to predict the classes: {floating, paddling, surfing}. This experiment has the same class distribution as the experiment in Results Section 4.2 (Figure XX).

Table 4. *A summary of the mean classification accuracy scores produced by performing a 10-Fold Cross Validation on each model in experiment 3.*

Model	Mean Classification Accuracy Percentage	Mean Accuracy Score ± Standard Deviation
Multinomial LR	86%	0.857 ± 0.031
Multinomial MLP	82%	0.818 ± 0.098
Multinomial Linear SVM	82%	0.817 ± 0.155
Multinomial Non-Linear SVM	80%	0.799 ± 0.120

4.4 Results:
Upsampled Additional Features on "Simple" Classes from Multiple Surf Sessions

The main difference between this experiment and the previous experiment (in section 4.3) is that we are now using multiple surf sessions, rather than a single surf session. This experiment attempted to predict the classes: {floating, paddling, surfing}.

Table 5. *A summary of the mean classification accuracy scores produced by performing a 10-Fold Cross Validation on each model in experiment 4.*

Model	Mean Accuracy Percentage	Mean Accuracy Score ± Standard Deviation
Multinomial LR	70%	0.705 ± 0.009
Multinomial MLP	63%	0.629 ± 0.070
Multinomial Linear SVM	58%[3]	0.581 ± 0.062
Multinomial Non-Linear SVM	N/A[4]	N/A

[3] Calculated by taking the mean of the 3-fold cross validation scores rather than a 10-fold cross validation scores.
[4] We were unable to calculate the multinomial non-linear SVM for this dataset due to limitations in time and computing resources that it took for this algorithm to run.

4.5 Results:
Downsampled Additional Features on “Simple” Classes from Multiple Surf Sessions

This is the same experiment as the previous experiment (in section 4.4), except the data is randomly downsampled to the minority class, rather than upsampled to the majority class. We also experimented with the use of a Butterworth Filter as an additional feature in Experiment 4.4.2, but it did not significantly affect our results (<1% absolute change in mean accuracy percentage for each model).

Table 6. *A summary of the mean classification accuracy scores produced by performing a 10-Fold Cross Validation on each model in experiment 5.*

Model	Mean Accuracy Percentage	Mean Accuracy Score ± Standard Deviation
Multinomial LR	59%	0.585 ± 0.095
Multinomial MLP	69%	0.687 ± 0.024
Multinomial Linear SVM	59%	0.589 ± 0.038
Multinomial Non-Linear SVM	57%	0.572 ± 0.012

4.6 Results:
Downsampled Additional Features on "Complex" Classes from Multiple Surf Sessions

This experiment is similar to the previous experiment in section 4.4.2 in that it utilizes multiple surf sessions and uses a downsampling method. The main difference between this experiment and the previous experiment is that we are predicting the "complex" class labels rather than just the "simple" class labels. This experiment attempted to predict the following classes: {push-off, paddling into waves, sit-up, floating, turning to surfer's left, lay-down, paddling for a wave, sit-back, pop-up, surfing, wipe-out, turning to surfer's right, pull-back leash, paddling for position, step-off}.

Table 7. *A summary of the mean classification accuracy scores produced by performing a 10-Fold Cross Validation on each model in experiment 6.*

Model	Mean Accuracy Percentage	Mean Accuracy Score ± Standard Deviation
Multinomial LR	19%	0.192 ± 0.040
Multinomial MLP	32%	0.323 ± 0.026
Multinomial Linear SVM	25%[5]	0.254 ± 0.007
Multinomial Non-Linear SVM	45%[6]	0.455 ± 0.027

[5] Calculated by taking the mean of the 3-fold cross validation scores rather than a 10-fold cross validation scores.
[6] Same as above.

Chapter 5: Evaluation and Discussion

The first experiment (Results Section 4.1) gives us an indication of the prediction performance of a one-vs-all logistic regression model on each of the classes in a completely labelled dataset, given a single surf session. The results of this experiment show that when given all labelled classes, the model is best able to predict the floating class but is least able to predict the surfing class. This means that it was most able to distinguish instances of the floating class from all other labelled classes in the dataset, but was unable to perform this task well for the surfing class.

The second and third experiments (Results Sections 4.2 and 4.3) demonstrate how well each multinomial model (LR, MLP, SVMs) performs on the "simple" dataset when given a single surf session to learn from. The third experiment shows us that the models perform significantly better, achieving improvements of 12-16% in their mean classification accuracy scores, when additional input features (computed from the raw input features) are included in the dataset. While each model performed comparably in the second experiment, the LR model performed notably better (86% mean classification accuracy) than the SVM model (80% mean classification accuracy). Therefore, it is likely that classes within the dataset became more linearly separable when given the additional input features, and that the instantaneous velocity and average rate of change of angular velocity were good indicators of when a surfer was floating, paddling, or surfing.

The fourth experiment (Results Section 4.4) depicts a reduction in prediction performances in each of the tested models[7], when compared to the third experiment. The difference between the third and fourth experiments was that the third experiment tested a dataset taken from a single surf session, while the fourth experiment tested a dataset taken from multiple surf sessions. This leads us to believe that the variability of oceanic conditions between different surf sessions has an overall effect on each model's prediction performance. Overall, mean classification accuracy scores for each model dropped by 16-24% for each model in the fourth experiment, when compared to the results of the third experiment.

The only difference between the fifth experiment (Results Section 4.5) and the fourth experiment is that the dataset in the fifth experiment is downsampled rather than upsampled as it was in the fourth experiment. This difference in the way the dataset was balanced significantly altered the results of the fourth and fifth experiments. While the LR model performed best in the

[7] The results of the SVM model with a non-linear kernel were unable to be calculated due to constraints in computational resources but was assumed to be poorer than in the third experiment based on the other models' results.

fourth experiment (70% mean classification accuracy), the MLP model performed best in the fifth experiment (69% mean classification accuracy). A possible explanation for this is that the MLP and SVM models improve slightly when dealing with less data because they do not overfit to the training data, while the LR model is not at risk of overfitting, and therefore its prediction performance decreases slightly when given less data to learn from.

The sixth and final experiment (Results Section 4.6) only differs from the fifth experiment in that it is trained and tested on the “complex” dataset, rather than the “simple” dataset. There are a total of 15 labels in this dataset. Here, the SVM model with a non-linear kernel performs overwhelmingly better (45% mean classification accuracy) than all other models that were tested. The LR model which had performed the best in most of the other experiments, for example, performed the worst in this experiment (19% mean classification accuracy). These results support the idea that the “complex” dataset is not linearly separated, which is why the SVM model with a non-linear kernel performs best.

Chapter 6: Conclusion and Future Work

In this thesis, we have analyzed the prediction performance of various machine learning models (logistic regression, multilayer perceptron, and support vector machines with linear and non-linear kernels) in predicting different classes of surfer motion from the Smartfin's IMU data. We have found through experimentation that our models' prediction performance is optimal when our training and validation datasets are taken from a single surf session. Our results were improved when feature engineering was performed to include instantaneous velocity as well as a moving window average of the gyroscope data on data taken from the single surf session. The fact that the logistic regression model performed best in this case leads us to believe that our data becomes more linearly separated after performing feature engineering and that the MLP and SVM models are potentially overfitting to the training set. We believe that our accuracy decreases when training and testing our models on a dataset taken from a combination of eight recorded surf sessions because oceanic conditions are highly variable when looking at multiple surf sessions which were taken from multiple surfers and locations over various months.

Future work will need to be done to determine whether this level of classification accuracy is adequate for oceanographers who are using the Smartfin to collect data for scientific purposes. If not, additional work will need to be done to determine whether it is possible to achieve a higher prediction performance with other additional input features, filters, or machine learning models. Further investigation could also reveal the quantitative effects that different surfers (surfing styles), locations, or times of year have on our models' prediction performances.

References

[1] Adu, Paa & Stefan Bran-Melendez. Optimizing IMU Based Gesture Classification. Stanford University.

[2] Mannini, Andrea & Angelo Maria Sabatini. (2010). Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers. *Sensors*, 10(2), 1154-1175.

[3] Susi, M., V. Renaudin & G. Lachapelle. (2013). Motion mode recognition and step detection algorithms for mobile phone users. *Sensors (Basel, Switzerland)*, *13*(2), 1539-62. doi:10.3390/s130201539

[4] J. Huang, Z. Huang & K. Chen. (2017). Combining low-cost Inertial Measurement Unit (IMU) and deep learning algorithm for predicting vehicle attitude. *IEEE Conference on Dependable and Secure Computing*, pp. 237-239. doi: 10.1109/DESEC.2017.8073847

[5] Zhang R. & C. Li. (2015) Motion Sequence Recognition with Multi-sensors Using Deep Convolutional Neural Network. *Intelligent Data Analysis and Applications. Advances in Intelligent Systems and Computing*, vol 370. Springer, Cham.

[6] Secomb, Josh L., Jeremy M. Sheppard & Ben J. Dascombe. (2015). Time-Motion Analysis of a 2-Hour Surfing Training Session. *International Journal of Sports Physiology and Performance*, 10(1), 17-22.

[7] Redd, Michael J. & David H. Fukuda. (2016). Utilization of Time Motion Analysis in the Development of Training Programs for Suring Athletes. *Strength and Conditioning Journal*, 38(4), 1-8. doi: 101.1519/SSC.0000000000000243

[8] Mendez-Villanueva, Alberto, David Bishop, & Peter Hamer. (2006). Activity profile of world-class professional surfers during competition: A case study. *Journal of Strength and Conditioning Research*, 20(3), 477-482.

[9] Barlow, Matthew J, Karen Gresty, Malcolm Findlay, Carlton B. Cooke, & Mark A. Davidson. (2014). The Effect of Wave Conditions and Surfer Ability on Performance and the Physiological Response of Recreational Surfers. *Journal of Strength and Conditioning Research*, 28(10), 2946-2953. doi: 10.1519/JSC.0000000000000491

[10] Lees, Adrian. (2002). Technique Analysis in Sports: A Critical Review. *Journal of Sports Sciences*, 20(10), 813-828.

[11] Mahfuz, Rehana, Zeinab Mourad, & Aly El Gamal. (2016). Analyzing Sports Training Data
with Machine Learning Techniques. Purdue University.

[12] Luckert, Michael. (2015). Using Machine Learning Methods for Evaluation the Quality of Technical Documents (master's thesis). Linnaeus University, Sweden.

Appendix A
Additional Experiment 4.1 Notes and Results

Table 8: *Distribution of all labelled classes used in Experiment 4.1*[8]

Class Label	**Distribution in Dataset**
Floating	46.89%
Paddling for position	0.52%
Sit-back	0.30%
Pull-back leash	1.54%
Turning to surfer's right	3.95%
Paddling for a wave	4.20%
Turning to surfer's left	4.38%
Paddling into waves	18.86%
Pop-up	0.90%
Sit-up	0.93%
Push-off	0.89%
Wipe-out	0.14%
Walking in water	7.16%
Step-off	0.52%
Lay-down	1.34%
Off-board	2.49%
Surfing	1.95%

[8] Classes that were labelled in this experiment, but which were not used in any of the experiments (ex: "board upside down") do not appear in the table. Classes were not used in experiments if they were not consistently labelled across all surf sessions.

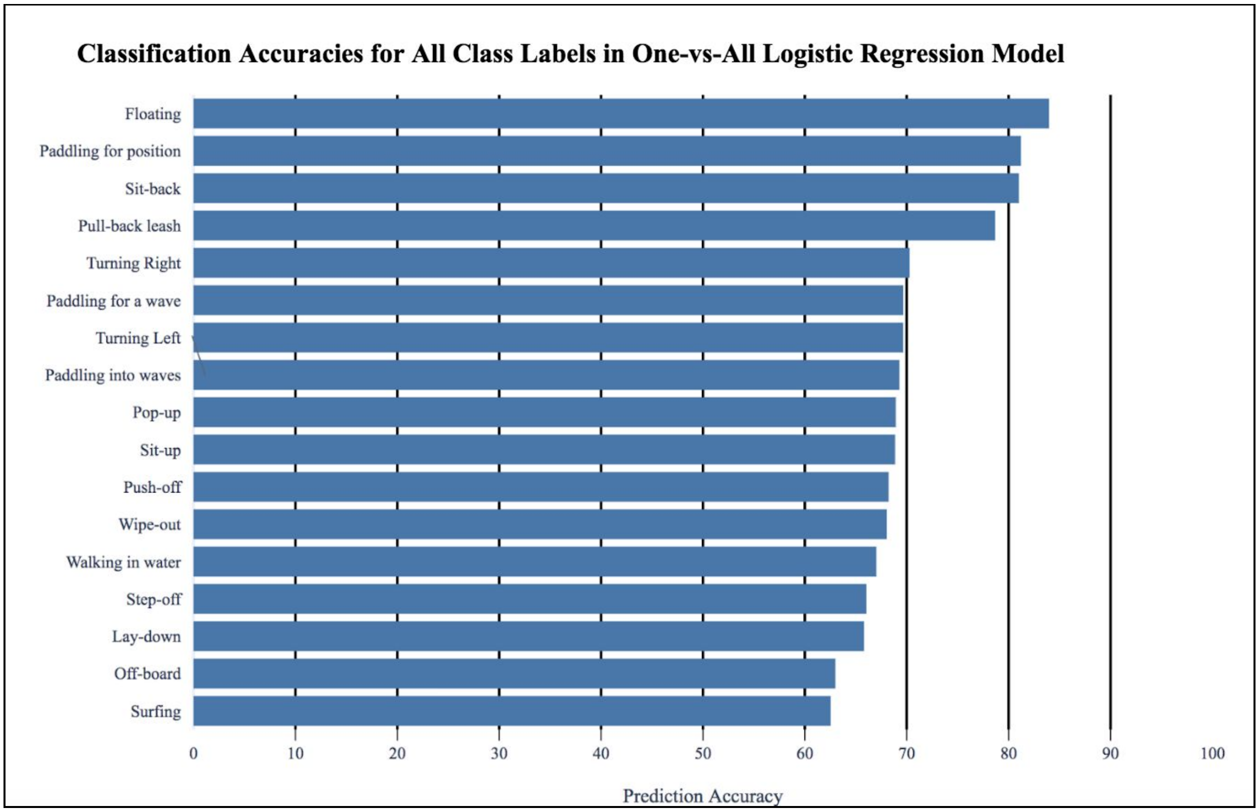

Figure 5. *Results of one-vs-all binary logistic regression experiment, sorted in decreasing order by the model's prediction (classification) accuracy.*

Appendix B
Additional Experiment 4.2 Notes and Results

Table 9: *Distribution of "Simple" labelled classes used in Experiments 4.2 and 4.3*

Class Label	Distribution in Dataset	Total # of Samples
Floating	71.88%	3517
Paddling	26.57%	1300
Surfing	1.55%	76

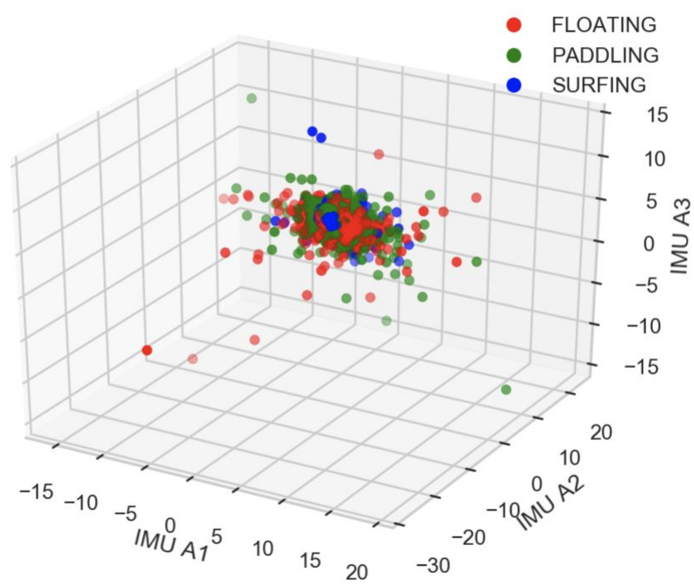

Figure 6. *Visualization of the "simple" dataset in 3 dimensions, with respect to the Smartfin's raw accelerometer values in the x, y, and z directions.*

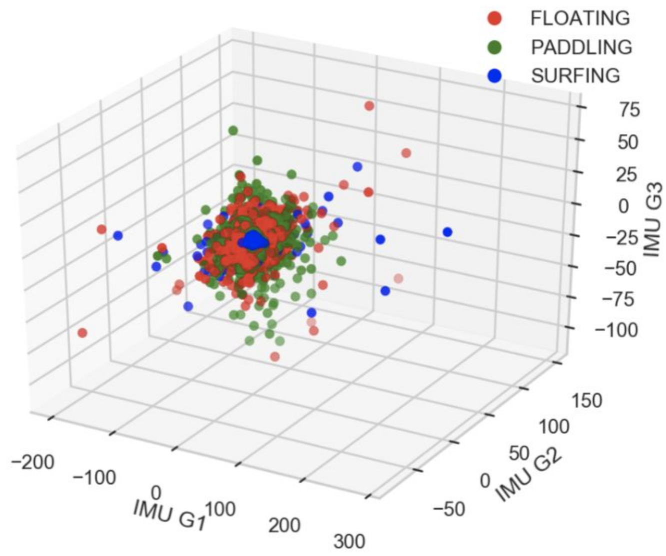

Figure 7. *Visualization of the "simple" dataset in 3 dimensions, with respect to the Smartfin's raw gyroscope values in the x, y, and z directions.*

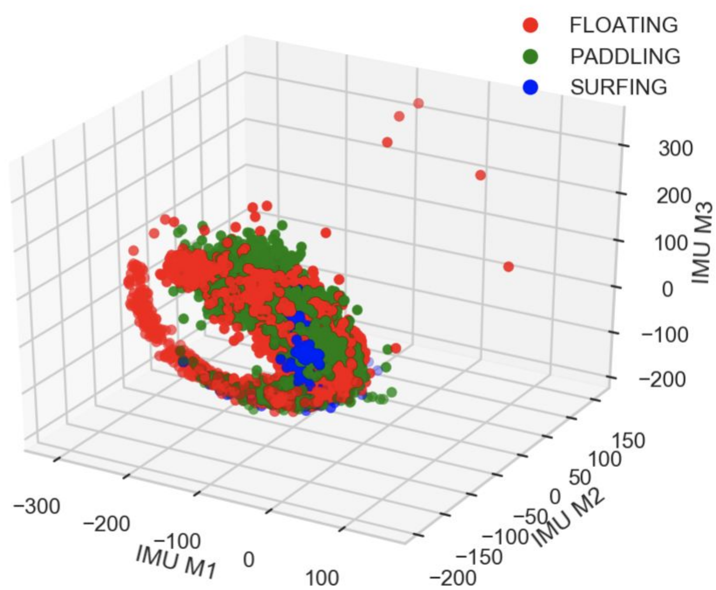

Figure 8. *Visualization of the "simple" dataset in 3 dimensions, with respect to the Smartfin's raw magnetometer values in the x, y, and z directions.*

Multinomial Logistic Regression Model in Experiment 4.2:

The logistic regression model being used in this experiment is imported from the sklearn linear_model library, with the parameters set up as: multi_class='multinomial' and solver='newton-cg'). Here, we focus on an analysis of the model's prediction performance on the "simple" dataset of {floating, paddling, surfing}. The following confusion matrices depict the model's classification accuracy for each label, with the true label on the y-axis and the predicted label on the x-axis. The figure on the left gives a better idea of the number of instances being labelled for each class, while the figure on the right uses normalization to portray classification accuracy more clearly.

Table 10. *The 10-Fold Cross Validation report from the LR model produced the following array of accuracy scores in experiment 4.2.*

Iteration #	1	2	3	4	5	6	7	8	9	10
Score:	0.70588	0.67647	0.67647	0.67647	0.71642	0.76119	0.74242	0.68182	0.73846	0.70769

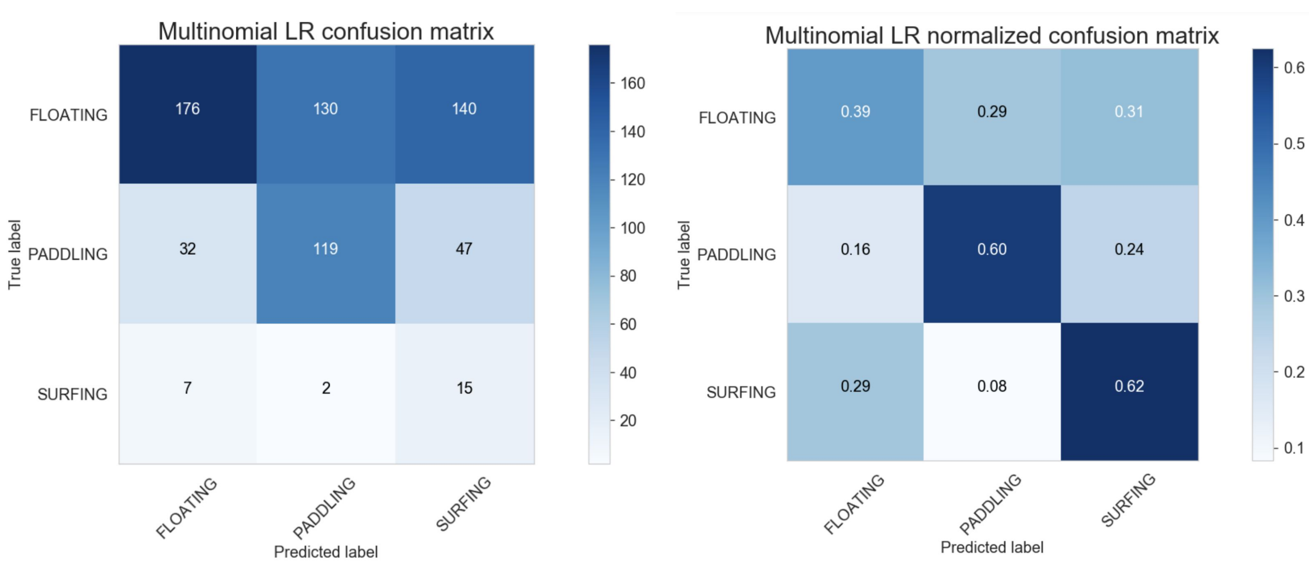

Figure 9. *The confusion matrices, original (left) and normalized (right) produced by the LR model in experiment 4.2.*

Multinomial Multilayer Perceptron Model in experiment 4.2:

The MLP model being used in this experiment is the MLPClassifier imported from the Python sklearn library. Unlike the logistic regression model, the MLP model must undergo a parameter tuning process in order to determine the best parameters to achieve the highest mean accuracy on this dataset. This was done using the GridSearchCV function from Python's sklearn "model_selection" library. The grid space that was searched consisted of the following parameters: {'solver': ['lbfgs'], 'max_iter': [100,200,1000,1500,2000], 'alpha': [1e-5, 0.001, 0.1, 10.0, 1000.0], 'hidden_layer_sizes':[(10,10), (15,15), (100,100), (5,5), (15, 3,15), (10,15)], 'random_state':[0,1,2]}. The GridSearchCV function found that the best parameters for this grid space were: {'alpha': 10.0, 'hidden_layer_sizes': (100, 100), 'max_iter': 100, 'random_state': 2, 'solver': 'lbfgs'}.

Table 11. *The 10-Fold Cross Validation report from the LR model produced the following array of accuracy scores in experiment 4.2.*

Iteration #	1	2	3	4	5	6	7	8	9	10
Score:	0.64925	0.90149	0.68311	0.78326	0.79970	0.75486	0.78593	0.76347	0.40868	0.45359

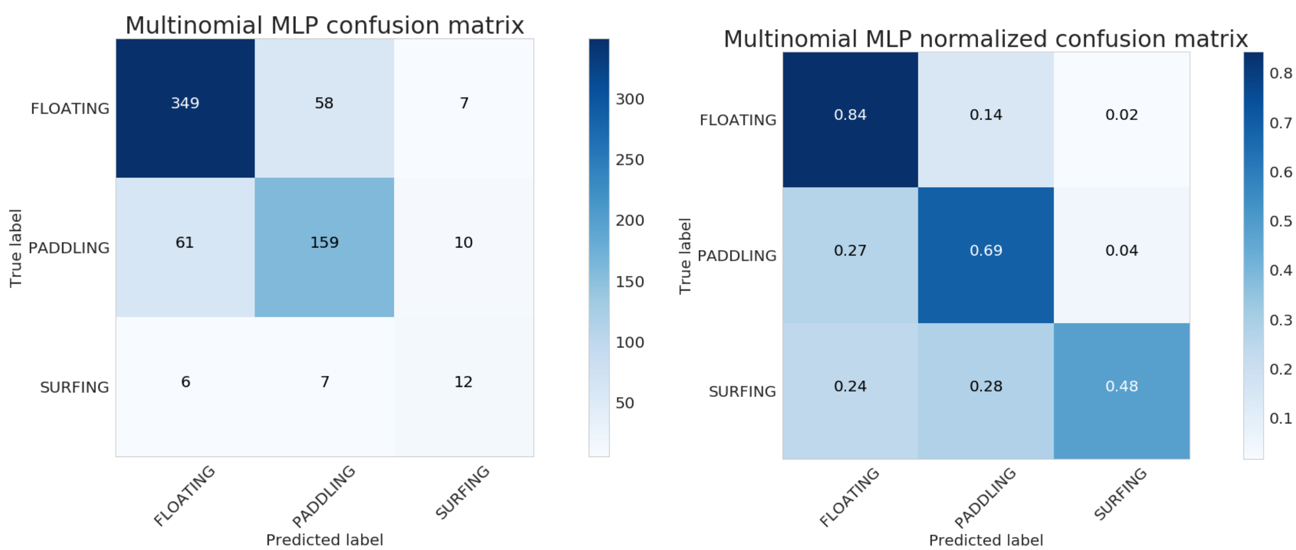

Figure 10. *The confusion matrices, original (left) and normalized (right) produced by the MLP model in experiment 4.2.*

Multinomial Support Vector Machines Models in experiment 4.2:

We perform the same experiment again with the multinomial SVM model. The SVM model used here is imported from the sklearn.svm SVC library. Similar to the MLP model, the SVM model needs to undergo a parameter tuning process to determine the best parameters to be used. Additionally, the SVM model was tested using two different kernels, the 'linear' kernel and the non-linear 'rbf' kernel, to determine which type of kernel would lead to better performance. Again, the tuning parameters were determined by the GridSearchCV function, with the following grid search space: {'C': [0.001, 0.01, 0.1, 1, 10], 'gamma':[0.001, 0.01, 0.1, 1]}. This produced the following 'linear' kernel parameters: {'C': 10, 'gamma': 0.001} and produced the following 'rbf' kernel parameters: {'C': 0.1, 'gamma': 0.001}.

Table 12. *The 10-Fold Cross Validation report from the SVM model with a linear kernel produced the following array of accuracy scores in experiment 4.2.*

Iteration #	1	2	3	4	5	6	7	8	9	10
Score:	0.67761	0.83731	0.72347	0.75486	0.80717	0.76981	0.74401	0.77246	0.50749	0.38323

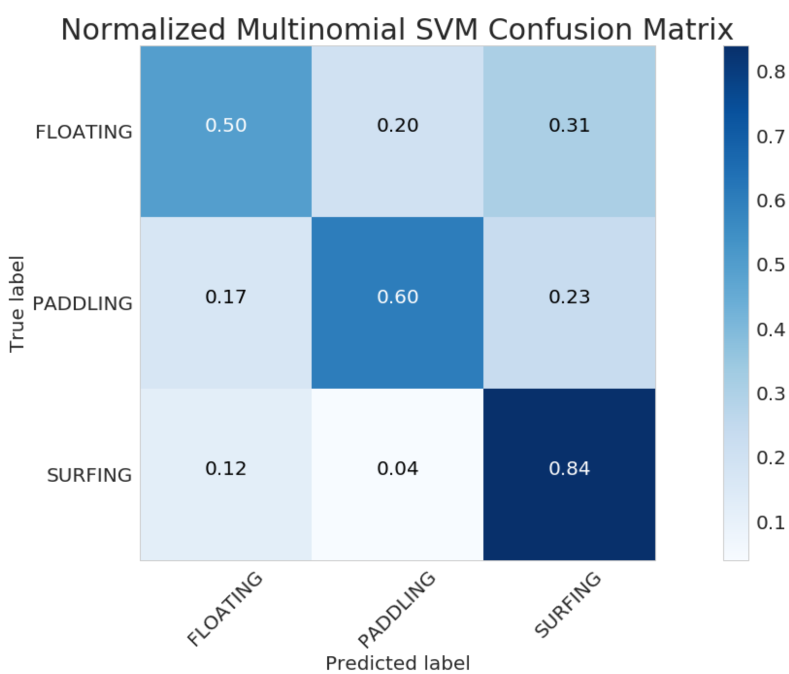

Figure 11. *The normalized confusion matrix produced by the linear SVM model in experiment 4.2.*

Table 13. *The 10-Fold Cross Validation report from the SVM model with a non-linear 'rbf' kernel produced the following array of accuracy scores in experiment 4.2.*

Iteration #	1	2	3	4	5	6	7	8	9	10
Score:	0.60149	0.88507	0.63528	0.67713	0.80568	0.72347	0.78144	0.79192	0.42964	0.48802

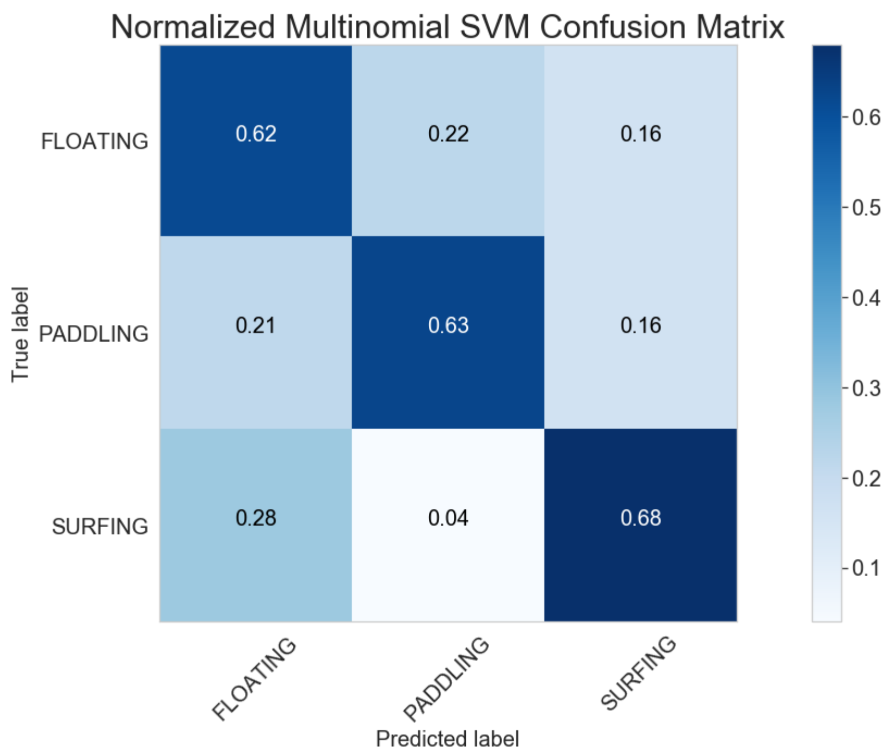

Figure 12. *The normalized confusion matrix produced by the non-linear SVM model in experiment 4.2.*

Appendix C
Additional Experiment 4.3 Notes and Results

The distribution of this dataset was the same as in experiment 4.2 (Figure XX).

Multinomial Logistic Regression Model in experiment 4.3:

We use a multiclass Logistic Regression model from the sklearn linear_model library with the following parameters specified: (multi_class='multinomial', solver='newton-cg') and balance the dataset with an upsampling technique known as SMOTE.

Table 14. *The 10-Fold Cross Validation report from the LR model produced the following array of accuracy scores in experiment 4.3.*

Iteration #	1	2	3	4	5	6	7	8	9	10
Score:	0.86486	0.91216	0.80405	0.87075	0.82993	0.86986	0.88356	0.81507	0.87671	0.84932

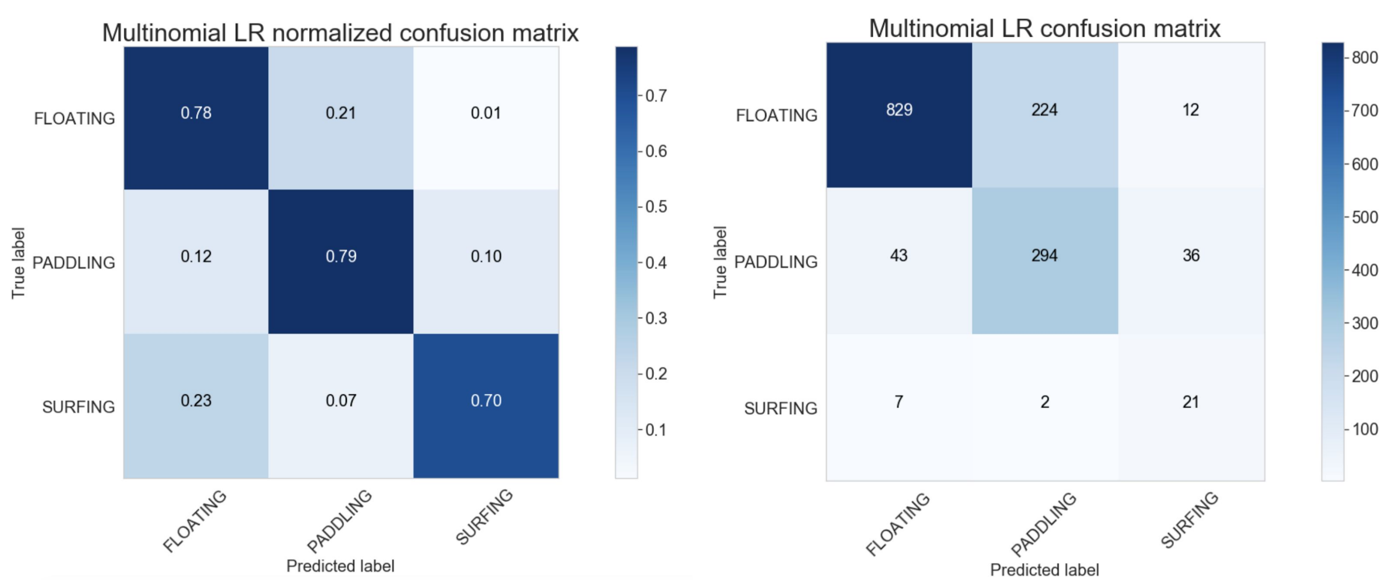

Figure 13. *The confusion matrices, original (left) and normalized (right) produced by the LR model in experiment 4.3.*

Multinomial Multilayer Perceptron Model in experiment 4.3:

We performed the same tuning parameter process as was done in Experiment 2 (Appendix B). The GridSearchCV method produced the following parameters: {'alpha': 0.001, 'hidden_layer_sizes': (100, 100), 'max_iter': 2000, 'random_state': 6, 'solver': 'lbfgs'}.

Table 15. *The 10-Fold Cross Validation report from the MLP model produced the following array of accuracy scores in experiment 4.3.*

Iteration #	1	2	3	4	5	6	7	8	9	10
Score:	0.59388	0.91633	0.90408	0.77143	0.69796	0.89184	0.83436	0.87705	0.86680	0.82787

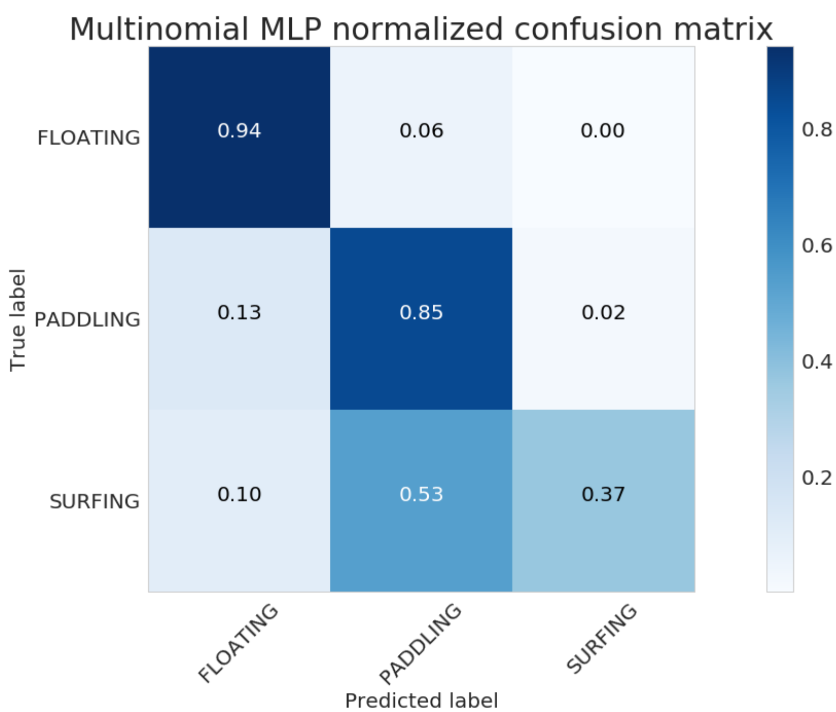

Figure 14. *The normalized confusion matrix produced by the MLP model in experiment 4.3.*

Multinomial Support Vector Machine Models in experiment 4.3:

We performed the same tuning parameter process as was done in Experiment 2 (Appendix B). The tuning parameters produced by this process for the linear SVM model were: {'C': 1, 'gamma': 0.001} and the tuning parameters for the non-linear SVM model were also: {'C': 1, 'gamma': 0.001}.

Table 16. *The 10-Fold Cross Validation report from the SVM model with a linear kernel produced the following array of accuracy scores in experiment 4.3.*

Iteration #	1	2	3	4	5	6	7	8	9	10
Score:	0.38979	0.91837	0.89388	0.91020	0.74898	0.84694	0.76483	0.89959	0.94262	0.85246

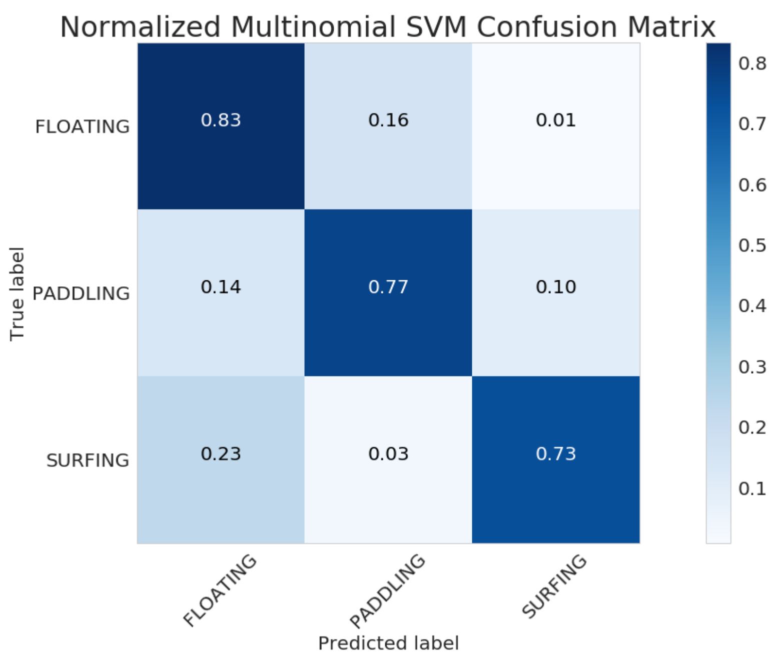

Figure 15. *The normalized confusion matrix produced by the linear SVM model in experiment 4.3.*

Table 17. *The 10-Fold Cross Validation report from the SVM model with a non-linear 'rbf' kernel produced the following array of accuracy scores in experiment 4.3.*

Iteration #	1	2	3	4	5	6	7	8	9	10
Score:	0.54286	0.91837	0.81224	0.75918	0.62857	0.91633	0.84867	0.92418	0.82788	0.80943

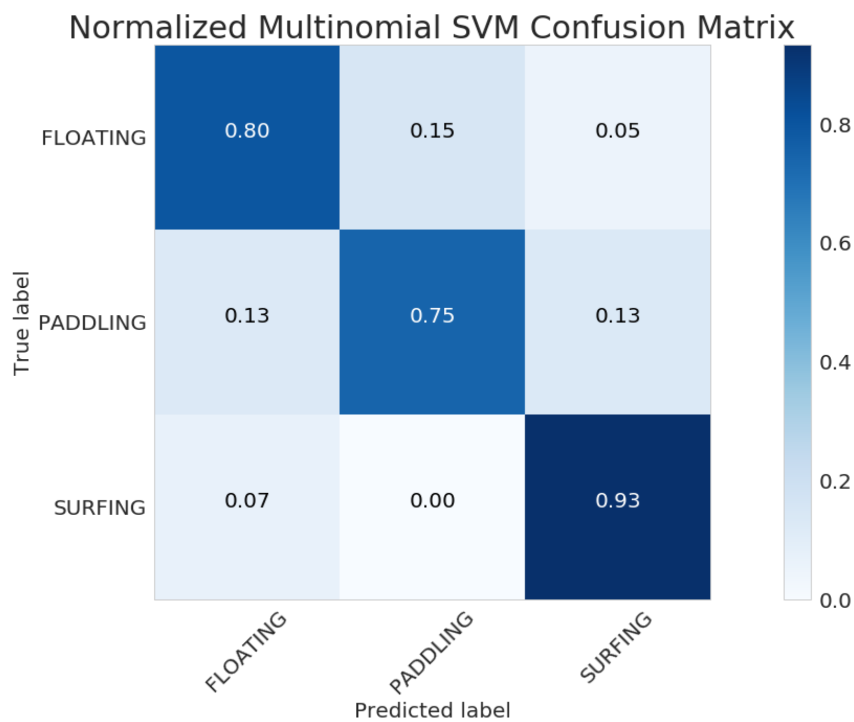

Figure 16. *The normalized confusion matrix produced by the non-linear SVM model in experiment 4.3.*

Appendix D
Additional Experiment 4.4 Notes and Results

Table 18: *Distribution of "Simple" labelled classes used in Experiments 4.4 and 4.5*

Class Label	Distribution in Dataset	Total # of Samples
Floating	66.29%	41456
Paddling	32.96%	20614
Surfing	0.75%	472

Table 19. *The 10-Fold Cross Validation report from the LR model produced the following array of accuracy scores in experiment 4.4.*

Iteration #	1	2	3	4	5	6	7	8	9	10
Score:	0.69382	0.70128	0.70964	0.70165	0.72189	0.71002	0.71467	0.69920	0.69440	0.70720

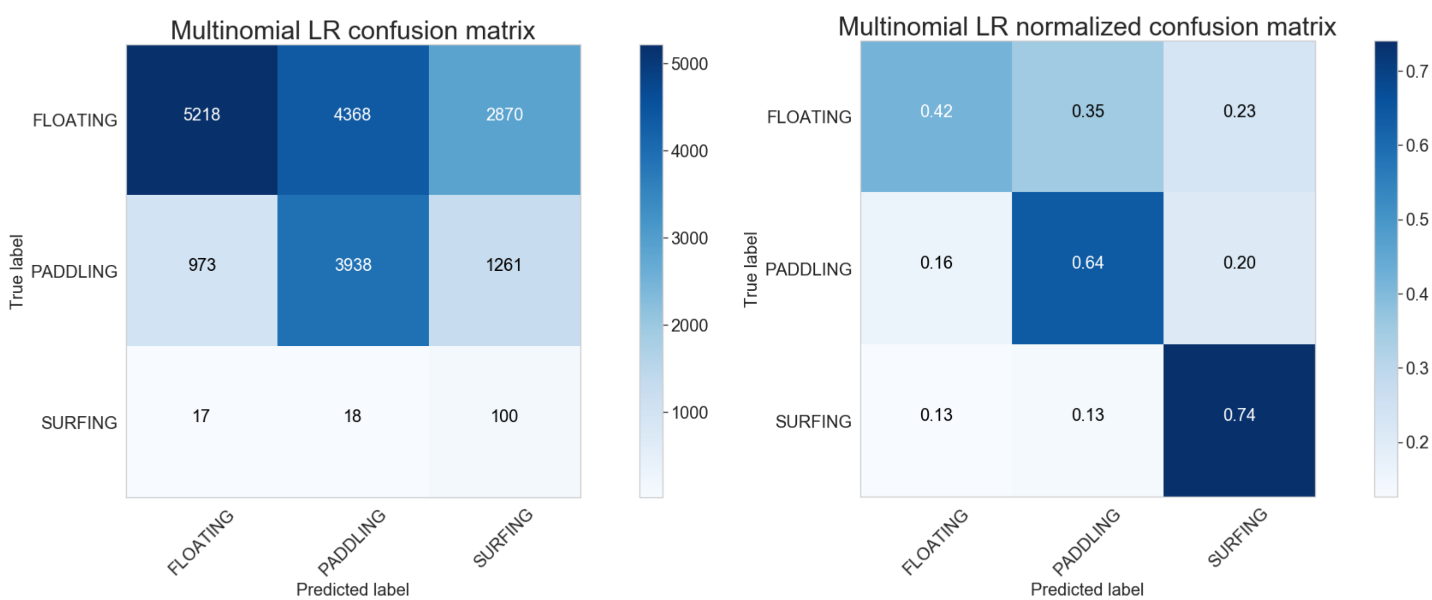

Figure 17. *The confusion matrices, original (left) and normalized (right) produced by the LR model in experiment 4.4.*

Table 20. *The 10-Fold Cross Validation report from the MLP model produced the following array of accuracy scores in experiment 4.4.*

Iteration #	1	2	3	4	5	6	7	8	9	10
Score:	0.66624	0.66273	0.64860	0.70488	0.55580	0.66293	0.45402	0.63169	0.68447	0.61842

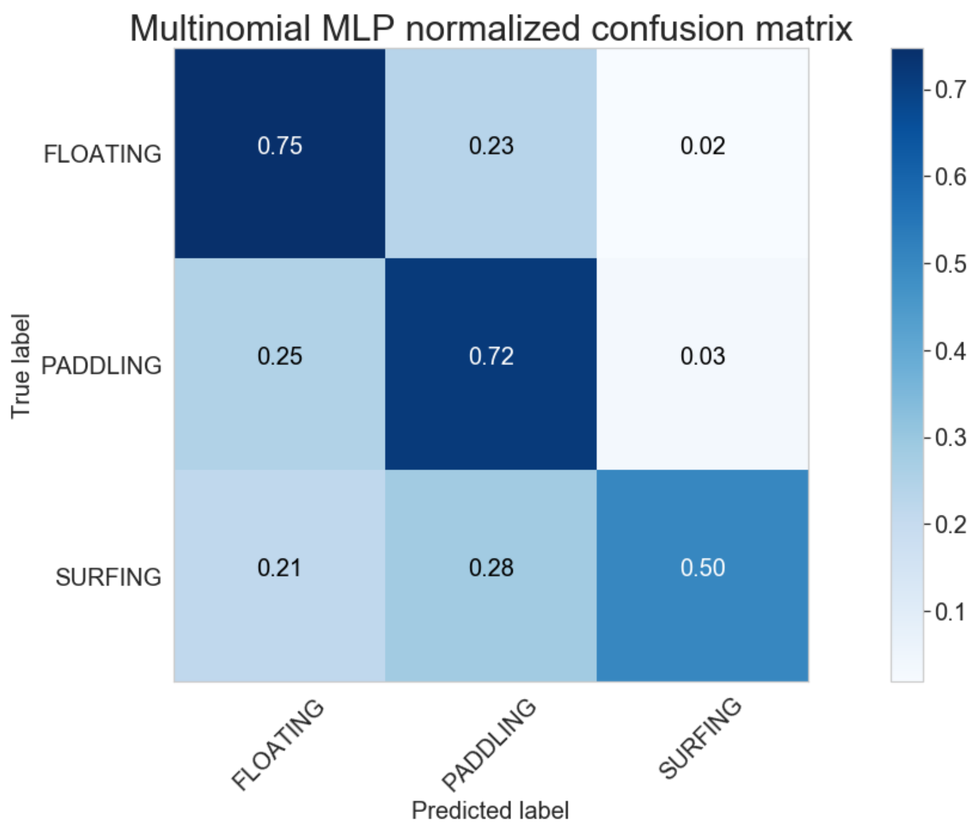

Figure 18. *The normalized confusion matrix produced by the MLP model in experiment 4.4.*

Appendix E
Additional Experiment 4.5 Notes and Results

For brevity, and because the only difference between experiment 4.4 and experiment 4.5 was the way the dataset was balanced (downsampled rather than upsampled in the latter case), the only table included in this section is the summary of each model's mean accuracy score from a 10-Fold Cross Validation the inclusion when Butterworth Filtered data is included as an additional feature. The Butterworth Filter was performed on the accelerometer and gyroscope data, and the original raw accelerometer and gyroscope data was kept as input features.

Table 21. *A summary of the mean classification accuracy scores produced by performing a 10-Fold Cross Validation on each model in Experiment 4.4, with the Butterworth Filtered accelerometer and gyroscope values included as additional features.*

Model	**Mean Accuracy Percentage**	**Mean Accuracy Score ± Standard Deviation**
Multinomial LR	58%	0.580 ± 0.101
Multinomial MLP	69%	0.689 ± 0.032
Multinomial Linear SVM	60%	0.595 ± 0.029
Multinomial Non-Linear SVM	57%	0.570 ± 0.009

Appendix F

Additional Experiment 4.6 Notes and Results

Table 22: *Distribution of "Complex" labelled classes used in Experiment 4.6*

Class Label	Distribution in Dataset	Total # of Samples
Push-off	0.41%	279
Paddling into waves	28.60%	19294
Sit-up	1.26%	854
Floating	61.47%	41456
Turning to surfer's left	1.77%	1191
Lay-down	0.75%	507
Paddling for a wave	1.76%	1187
Sit-back	0.30%	201
Pop-up	0.21%	139
Surfing	0.70%	472
Wipe-out	0.42%	284
Turning to surfer's right	1.62%	1094
Pull-back leash	0.22%	147
Paddling for position	0.20%	133
Step-off	0.30%	203