Classification of Shark Behavior from Accelerometry Data using Machine Learning

Sainesh Karan
Dept. of Electrical Engineering
California State University, Long Beach
Long Beach, USA

Emily Messe
Dept. of Biological Sciences
California State University, Long Beach
Long Beach, USA

Dr. Yu Yang
Dept. of Chemical Engineering
California State University, Long Beach
Long Beach, USA

Dr. Hen Guel Yeh
Dept. of Electrical Engineering
California State University, Long Beach
Long Beach, USA

Dr. Chris Lowe
Dept. of Biological Sciences
California State University, Long Beach
Long Beach, USA

Dr. Wenlu Zhang
Dept. of Computer Science
California State University, Long Beach
Long Beach, USA

Abstract—In this paper, we aim to ascertain the behavior of California Horn sharks (Heterodontus francisci) based on accelerometer data using a machine learning algorithm called Nearest Neighbors. We use digital signal processing techniques such as Fast Fourier Transform (FFT) and Discrete Cosine Transform (DCT) to represent the accelerometer signal in the frequency domain which reduces the data size required for training the classifier, the computation time and the memory resources needed, in addition to improving model accuracy. The shark behavior is classified into four classes namely Resting, Swimming, Feeding and Non-Directed Motion. We compare different combinations of time and frequency domain data on the performance of the algorithm. It is shown that the transform domain data considerably improved the accuracy of the classifier.

Index Terms—Activity Recognition, Shark Behavior, KNN, Digital Signal Processing, Machine Learning

I. INTRODUCTION

Activity recognition pertains to the task of classifying physical activities undertaken by an individual from the collected data. Differentiating behaviors of free-ranging animals provides an insight into their feeding patterns that help us understand their ecology and their impact on the environment. Recently, accelerometers have been employed to unveil the abstruse lives of marine animals. It is extremely arduous to observe such species in their natural environments over long periods due to poor visibility, deeper depths, and adverse environmental conditions [1]. Over the years bio-telemetry devices have become popular amongst researchers to overcome these obstacles and are widely available. An Acceleration Data Logger (ADL) is one such device that measures acceleration caused by body movements that can be used to determine body orientation and kinematics for behavioral classification [1]. ADLs allow data to be recorded and stored at high frequencies, and thus providing high-resolution data with useful information to reveal the animal's behavior. However, the increased volume of high-resolution data is non-trivial to process, and the useful information hidden behind may not be easy to extract. This challenge motivates the research of automatic behavioral classification by using machine learning techniques.

The past few years have seen tremendous development in the fields of machine learning and artificial intelligence, which has spurred a widespread adoption of the technology for several applications. Machine learning algorithms can be broadly categorized into supervised and unsupervised methods. In supervised methods, a training data-set, which consists of the data and the associated outcome/labels, is used to train a model to accurately map the input variables to the outcome/labels. Once the model has been built based on the training data, it can be used to make predictions on new input data. Examples of such algorithms include Decision Trees, Random Forest (RF), K-Nearest Neighbors (KNN) and Neural Networks [1]. Several studies have been conducted in the past to study the behavior of free-ranging animals using data loggers and machine learning techniques. In [2] and [3], the authors demonstrate the utility of using data loggers and machine learning algorithms to classify the behavior of Griffon vultures and Cheetahs, respectively. In [4] authors used machine learning techniques to identify the hunting and feeding pattern of penguins. Apart from behavior classification, authors of [5] and [6] have used data loggers to determine the mortality and energy expenditure of sharks, respectively.

In this paper, we apply the KNN algorithm to classify data collected from the ADL into four behavior classes namely Resting, Swimming, Feeding, and Non-directed motion. KNN is a simple machine learning algorithm in which, a new data point is classified based on the majority of its k nearest neighbors, k being a user defined hyper-parameter. The algorithm does not use any model to fit the new data sample but rather computes the distance of the data point form its neighbors [7]. We also explore the effect of signal transformation techniques such as Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT) on the performance of the classification algorithm. We expect that the transformation techniques will enhance the differentiating features of each behavior class thereby aiding the algorithm in the process while reducing

the computation complexity and time. All data pre-processing, simulations and machine learning techniques used in this paper were implemented in Matlab R2018b® using the Statistics and Machine Learning Tool Box Ver.11.4.

II. DATA AQUISITION AND PRE-PROCESSING

A. Data Aquisition

Laboratory trials were conducted to quantify acceleration signatures of California horn sharks (Heterodontus francisci) for different behaviors including resting, swimming, and feeding (prey capture and handling). Horn sharks were chosen as a model species to represent a demersal, resting shark that are capable of eating hard-shelled invertebrates [8]–[10]. Sharks were collected from Santa Catalina Island, and transported back to the CSULB Shark Lab. Trials occurred with a single shark in a 340 L tank with two GoPro video cameras, one mounted on the bottom of the tank, and one above the tank. For each trial, a shark was fitted with an accelerometer data logger (ADL; Cefas G6a+ or Technosmart AxyDepth) via a custom tag package. Accelerometers recorded 3D acceleration data at 25 Hz. All acceleration data was processed in IgorPro (vers. 6.2, WaveMetrics) using Ethographer [11]. Static acceleration represents the 3D body position and orientation due to gravity (i.e., pitch, X static; roll, Y static; yaw, Z static), whereas dynamic acceleration represents the 3D body movements (i.e., surge, X dynamic; heave, Y dynamic; sway, Z dynamic) as shown in Figure 1. Static acceleration was separated from dynamic acceleration using a 2 second box smoother [12]. The plots for the raw data are shown in Figures 2 and 3.Sharks were tagged approximately 6 hours before the start of the trial to allow the shark to acclimate.

To characterize motion patterns associated with prey manipulation and handling, instrumented sharks were then fed three different prey types including a live purple urchin (< 4 cm test diameter), a frozen and thawed squid (whole, with pen), and a frozen and thawed shrimp. Prey was buried or placed away from the shark in order to characterize natural feeding behaviors expected in the field. A trial concluded when the shark had consumed all trial prey. Trials were approximately 2 hours long, and a total of seven trials were done on four different individuals.

The data generated by the ADL was labelled manually by viewing the GoPro video recording of each trial and identifying the shark behaviour with the corresponding time-stamps. All data samples that were unlabelled were removed from the dataset. An example of the recorded data from dataset number 7 is shown in Figures 2 and 3.

B. Data Pre-processing

A total of seven datasets were formed from the trials. Each dataset was then separated into its four constituent behaviour classes based on the labels. The distribution of different classes in each dataset is presented in table 1. Because the datasets had very few feeding samples in comparison to other classes, carefull consideration had to be given to how the data is split between training and testing. In our scheme, Dataset 5 was

(a) Static acceleration → body orientation of shark

(b) Dynamic acceleration → movements of shark
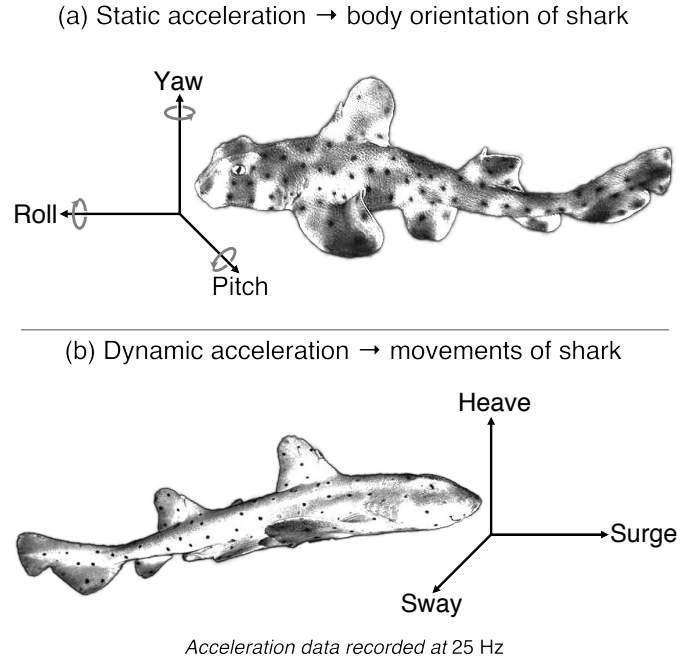

Acceleration data recorded at 25 Hz

Fig. 1. Breakdown of acceleration data recorded at 25 Hz. (a) Static acceleration represents the body position and orientation of the shark due to gravity. Dynamic acceleration represents the body movements of the shark.
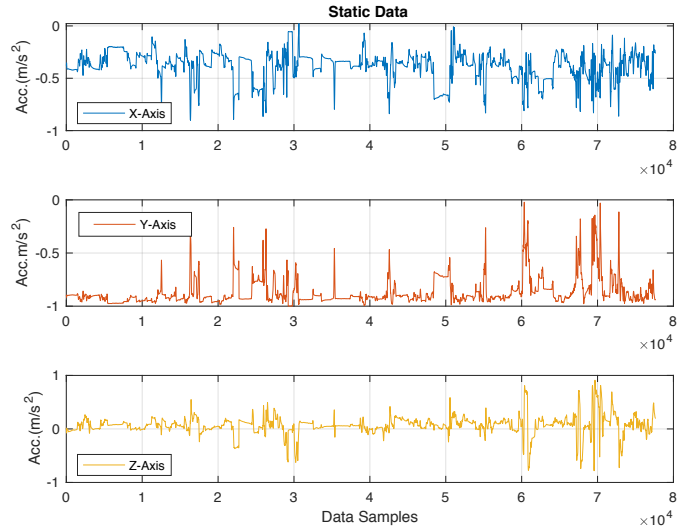

Fig. 2. Example of Static acceleration data collected from the ADL

chosen as the test data in order to maximize the number of feeding samples in the training data while still having enough feeding samples to test the model. In addition, Dataset 5 came from a different individual shark as compared to other datasets, which allows us to examine the generalization capability of the classifier. Next we combined all the datasets and subtracted the mean from all the data samples in the combined dataset so as to remove the DC-bias and noise, and centering the signal around zero. As can be seen from Table 1, the trials were of varying lengths with varying distribution of classes.
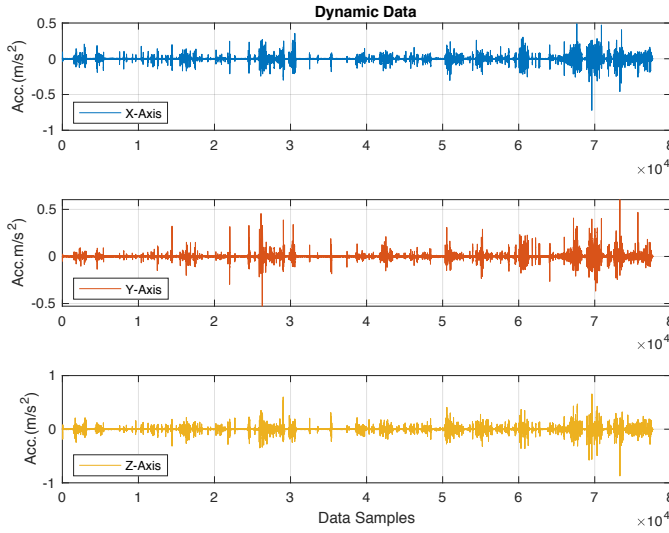
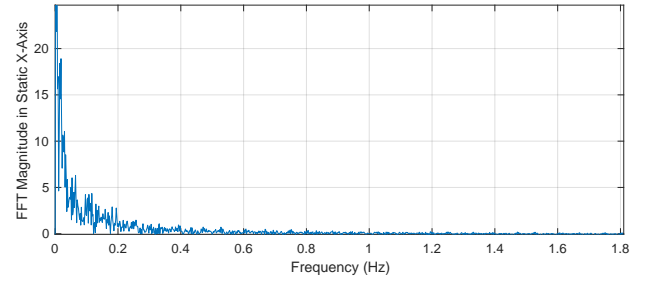Fig. 3. Example of dynamic acceleration data collected from the ADL

Fig. 4. Example of static data with FFT
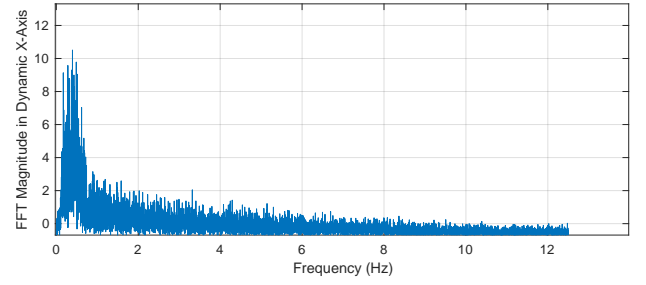

Fig. 5. Example of dynamic data with FFT

Fig. 6. Example of static data with DCT

Fig. 7. Example of dynamic data with DCT

As a result, the datasets were found to be biased with respect to the feeding behavior with the total Feeding samples being an order of magnitude lower than that of the other classes. In order to overcome this bias, we chose a subset of data samples from each of the remaining classes that was most consistent and free of noise. A total of 100,000 samples were chosen for Resting behavior with similarly equal samples taken from each dataset; 90,000 samples were chosen for swimming and 83,900 samples for Non-directed motion, respectively. The distribution of the chosen subsets is presented in Table II.

To improve the classifier efficiency, we convert the input signals into different domains of representation. Transforming signals into frequency domain allows us to capture the repetitive nature of the signals [13]. This repetition often correlates to the periodic nature of a specific activity such as the tail beat of the shark while swimming or movement of the head while feeding. Among the most popularly used transformation techniques is the Fourier Transform, which allows one to represent the most important characteristics of a time-based signal such as its DC and dominant frequency components in the frequency domain as shown in Figures 4 and 5 [13]. Fast Fourier Transform (FFT) decomposes an input signal into its constituent sinusoids. In addition to the FFT, other frequency-based representations have been used such as Discrete Cosine Transform (DCT) which in contrast, decomposes a given signal into its constituent cosine waves. DCT returns an ordered sequence of coefficients such that the most significant information is concentrated at the lower indices of the sequence as can be seen from Figures 6 and 7. This means that higher DCT coefficients can be discarded without losing information, which makes DCT suitable for data compression [14]. We compute the frequency analysis for the time signal of a specific length or window using the Fast Fourier Transform (FFT) algorithm. In this case we use a window of 25 data samples or 1 second so as to match the sampling frequency of the ADL and to avoid the transitioning
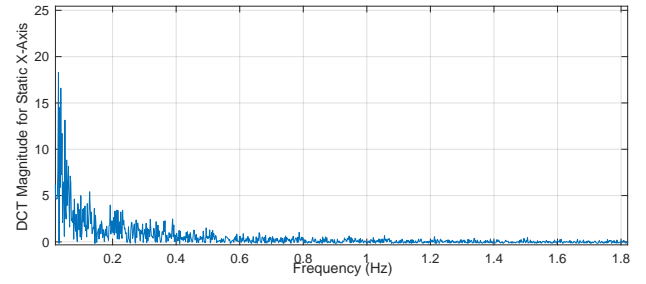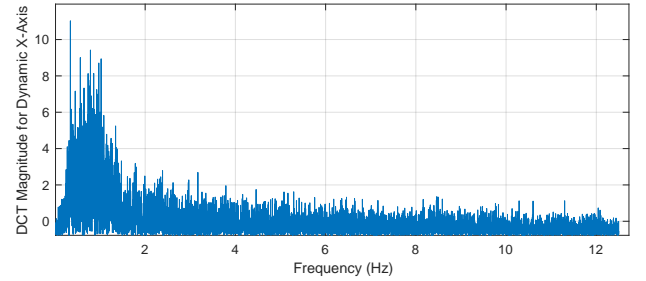
of behaviour classes within the window. In order to overcome the bias in the training dataset against feeding class. We oversampled the feeding data by sliding the window with a 50% overlap [15] so as to increase the number of feeding samples in the training dataset. Due to the symmetric property of Fourier Transform, we use only the 50% of the data in the frequency domain. Similarly, we use only 50% of the data of the DCT and discard the data in the higher frequency bins. However, it should be noted that the data reduction take place

	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Total
Resting	157,750	565,580	379,850	10,250	6,150	27,975	77,374	1,224,929
Swimming	6,200	209,550	5,100	81,525	7,975	19,750	61,475	391575
Feeding	5,700	350	200	1,375	875	2,900	2,100	13,500
NDM	51,950	7,400	15,700	1,750	1,775	27,025	5,400	11,1000
Total	22,1600	782,880	40,0850	94,900	16,775	77,650	146,349	–

TABLE II
CLASS DISTRIBUTION OF CHOSEN SUBSETS USED FOR TRAINING

	Dataset 1	Dataset 2	Dataset 3	Dataset 5	Dataset 6	Dataset 7	Total
Resting	20,000	20,000	20,000	10,000	15,000	15,000	100,000
Swimming	6,000	20,000	5,000	20,000	19,000	20,000	90,000
Feeding	5,700	350	200	1,375	2,900	2,100	12,625
NDM	30,000	7,000	15,000	1,500	25,000	5,400	83,900
Total	61,700	47,350	40,200	32,875	62,900	42,500	–

in when we use the transform domain data by itself. In the event where we combine the time and frequency domain data, the complete FFT and DCT are used so as to keep the input matrix to the model consistent. In such cases, no data reduction takes place. The total data samples in the transform domain are shown in Table III. Thus by considering only the half of the Fourier transform i.e. ignoring the mirror image and the high frequency components of the DCT, we can reduce the total size of the data to be classified by half.

TABLE III
TOTAL DATA SAMPLES IN THE TRANSFORM DOMAIN USED FOR TRAINING
THE MODEL.

Behavior class	Data samples, time domain	Data samples, transform domain	Data reduction representation
Resting	100,000	48,000	2.083
Swimming	90,000	43,200	2.083
Feeding	12,625	6,060	2.083
NDM	83,900	40,270	2.083

III. BEHAVIOR CLASSIFICATION USING K-NEAREST NEIGHBORS

K-Nearest Neighbor is a supervised learning algorithm where the result of new test sample is classified based on the majority of its k nearest neighbors, where k is a user defined hyper-parameter. The neighbors are determined by measuring the distance between the test sample and all the training samples. The goal of this algorithm is to classify a new input based on attributes of the training samples where the training samples are vectors in a multidimensional feature space, each with a class label [16]. Due to its simplicity and low complexity, it has been widely applied in several studies [17] [16] [18]. KNN classifier is model free method meaning it does not learn a model that maps the relation between the inputs and outputs; it instead stores the stores the training samples that are used to classify each new instance of the test data. [19] There are different metrics to determine the distance between the input query and training samples: Euclidean, City Block, Chebychev are some of the popular distance metrics.

The value of k completely depend upon the nature of the data. A higher value of k reduces the effect of noise but reduces the distinction between the boundaries of the classes [16]. In cases where the total number of classes are even, k is selected to be odd to prevent ties during classification. The classifier runs through the training dataset and computes the distance between the test sample x from each training sample and forms a set of the k nearest neighbors. It then computes the conditional probability of x belonging to each class as shown in equation (1) and classifies the test observation to the class with the highest probability.

$$P(i|x) = \frac{k_i}{k} \tag{1}$$

Here k_i is the number of nearest neighbors belonging to class i and k is the number of nearest neighbours. In our study, in order to determine the optimum value of k we ran multiple iterations of the program with different distance metrics and values of k. It was observed that as we increased the value of k the accuracy of Resting and Swimming behaviors also increased but that of Feeding and Non-Directed Motion decreased. This may be since Swimming and Resting classes have the most training samples and as the value of k is increased, the locality of the input query is destroyed and the algorithm starts to look at samples that are not neighbors. We were able to get the best accuracy of Feeding and Non-Directed Motion at $k = 15$ with the city block distance metric while limiting the loss of classification accuracy for resting and Swimming classes.

Another parameter introduced to improve the performance of classifier is the Cost Matrix. It is a square matrix, denoted by $cost(i, j)$, which represents the penalty of mis-classifying a data belonging to class i to class j. Naturally all diagonal elements of the Cost Matrix are zero since there no penalty for correct classification.

The algorithm classifies the new data sample by minimizing the expected cost given by equation 2.

$$y = \arg \min_{y=1...n} \sum_{j=1}^{n} P(i|x)C(i|j) \tag{2}$$

where y is the predicted classification, n is the number of classes and $P(i|x)$ is the condiional probability of class i for observation x and $C(i|j)$ is the user defined cost of classifying an observation as i when its true class is j.

In our study, we developed the cost matrix by trial and error method using the time domain data. We ran multiple iterations of the program and updated the elements of the cost matrix in increments of 0.25 until we were able to strike a balance between accuracies for all the behavior classes. The final cost matrix developed is shown below:

$$\text{cost}(i,j) = \begin{bmatrix} 0 & 9.5 & 2 & 6 \\ 2 & 0 & 3.5 & 4 \\ 1.25 & 5 & 0 & 1.5 \\ 2.25 & 3 & 1.5 & 0 \end{bmatrix}$$

We use the same cost matrix while comparing different pre-processing techniques to keep the comparison consistent. In the next section we compare the effect of the various pre-processing techniques discussed in Section II on the performance of the classifier with and without the cost matrix.

IV. RESULTS

The results for the experiment without using the cost matrix are shown in Table IV:

The KNN classifier was tested on Dataset 5 which was pre-processed in the same manner as the training data using a window of 1 second without any overlap. The results for the experiment without using the cost matrix are shown in Table IV. Overall, this approach exhibits a wide range of accuracy results. Starting with only Time domain data it is observed that the technique surprisingly had the best accuracy for feeding behavior even though its overall accuracy was the lowest. The overall accuracy increases when the transform domain data is used for classification. It increases from 59.05% to 66.89% when using only FFT and to 67.39% after combining FFT and DCT. It however reduces back to 59.95% when time domain data is combined with transform domain. It can also be observed that DCT helps to increase feeding accuracy. The feeding accuracy jumps from 1.40% to 2.90% when FFT is combined with DCT. Similarly, the feeding accuracy jumps from 6.60% to 8.10% when time domain data is combined with DCT. The reason why DCT aids the improvement in feeding accuracy needs to be further investigated. For most cases the accuracy for Resting behavior remained around 50-55% only increasing up to 81% when using a combination of FFT and DCT. The swimming accuracy remained consistent in most cases only dropping to 63% when using only DCT data. This was rather surprising as the maximum training data samples belonged to this class. It was difficult to come to a conclusion on how the accuracy of the NDM behavior was impacted. It can however be observed that NDM accuracy is higher in cases where time domain data is used as opposed to transform domain data. Overall, the performance of the classifier was rather dismal as it struggled to classify feeding data accurately. This challenge was overcome by using a cost

matrix to improve the feeding accuracy. The results for the classifier with the cost matrix are shown in Table V.

From Table V, It can be seen that while the overall accuracy is lower slightly lower than the previous case, but using a cost matrix vastly improves the accuracy of the Feeding behavior. Here the FFT and DCT combination had the highest accuracy for Resting and Feeding as well as the overall accuracy at 59.69% even though it showed a low accuracy of 13.60% for non-directed motion. Similar to the previous table we can observe that DCT aides in improving the Feeding accuracy. However, DCT when used by itself, gave the lowest overall accuracy of 53.74%.

V. CONCLUSION AND FUTURE WORK

In this paper we applied the KNN algorithm to classify shark behavior based on acceleration data collected by an ADL. We used various frequency transformation techniques such as the FFT and DCT to study their impact on the performance of the algorithm. We also compared the effect of different combination of data in time and frequency domain on the algorithm's performance. We can conclude that the transforming time domain data into frequency domain significantly improves the performance of the algorithm and that DCT was instrumental in improving the accuracy of Feeding behavior. We also successfully show that the signal transformation techniques can be used to reduce the overall data required for training the algorithm which results in a reduction in computational cost, memory requirement and computational time.

Future work may include developing a more robust methodology to optimize the parameters such as the value of k and the distance metric. and the cost matrix. A study can also be performed to compare the performance of various other machine learning algorithms such as Support Vector Machines (SVM), Neural Networks and Decision Trees. Also feature extraction can be used to extract statistical features from the raw as well as frequency domain data to help the algorithm differentiate between the classes. The selection of features and the optimum number required could be a topic of further research. In addition to this, future work may also include the use of wavelet analysis to derive time-frequency features from the data. Finally, the ultimate goal of this research is applying the classification models on data collected from the shark's natural environment to unravel feeding patterns of California horn sharks. This would allow researchers to gain an insight into horn shark's energetic demands and activity budgets to better understand their ecology.

VI. ACKNOWLEDGMENT

REFERENCES

[1] L. R. Brewster, J. J. Dale, T. L. Guttridge, S. H. Gruber, A. C. Hansell, M. Elliott, I. G. Cowx, N. M. Whitney, and A. C. Gleiss, "Development and application of a machine learning algorithm for classification of elasmobranch behaviour from accelerometry data," *Marine Biology*, vol. 165, no. 4, Aug 2018.

[2] R. Nathan, O. Spiegel, S. Fortmann-Roe, R. Harel, M. Wikelski, and W. M. Getz, "Using tri-axial acceleration data to identify behavioral modes of free-ranging animals: general concepts and tools illustrated for griffon vultures," *Journal of Experimental Biology*, vol. 215, no. 6, pp. 986–996, 2012.

TABLE IV
CLASSIFICATION RESULTS FOR VARIOUS COMBINATION OF PRE-PROCESSING TECHNIQUES USING KNN CLASSIFIER WITHOUT COST MATRIX IN PERCENT.

	Resting (%)	Swimming (%)	Feeding (%)	Non-directed Motion (%)	Overall (%)
Time domain	50.04	72.0	12.5	53.70	59.05
FFT	59.6	82.8	1.4	52.7	66.89
DCT	81.2	63.4	2.1	24.5	62.64
FFT + DCT	81.9	70.2	2.9	36.0	67.39
Time domain + FFT	52.4	71.1	6.6	62.3	59.95
Time domain + DCT	53.6	70.1	8.1	61.4	60.22
Time domain + FFT + DCT	55.9	70.3	4.5	62.4	60.76

TABLE V
RESULTS FOR VARIOUS COMBINATION OF PREPROCESSING TECHNIQUES USING KNN CLASSIFIER WITH COST MATRIX IN PERECENT.

	Resting (%)	Swimming (%)	Feeding (%)	Non-directed Motion (%)	Overall (%)
Time domain	49.6	66.7	21.0	51.0	56.39
FFT	54.3	72.3	39.8	31.1	59.66
DCT	76.8	48.5	31.9	8.2	53.74
FFT + DCT	77.6	57.6	46.	13.6	59.69
Time Domain + FFT	50.5	66.3	19.9	55.2	56.89
Time Domain + DCT	51.8	66.0	21.1	56.7	57.48
Time Domain + FFT + DCT	53.3	65.2	21.1	56.3	57.57

[3] S. Gr˛new%ꞔlder, F. Broekhuis, D. W. Macdonald, A. M. Wilson, J. W. Mcnutt, J. Shawe-Taylor, and S. Hailes, "Movement activity based classification of animal behaviour with an application to data from cheetah (acinonyx jubatus)," *PLoS ONE*, vol. 7, no. 11, 2012.

[4] S. Chessa, A. Micheli, R. Pucci, J. Hunter, G. Carroll, and R. Harcourt, "A comparative analysis of svm and idnn for identifying penguin activities," *Applied Artificial Intelligence*, vol. 31, no. 5-6, pp. 453–471, Mar 2017.

[5] N. M. Whitney, C. F. White, A. C. Gleiss, G. D. Schwieterman, P. Anderson, R. E. Hueter, and G. B. Skomal, "A novel method for determining post-release mortality, behavior, and recovery period using acceleration data loggers," *Fisheries Research*, vol. 183, pp. 210–221, 2016.

[6] A. C. Gleiss, R. P. Wilson, and E. L. C. Shepard, "Making overall dynamic body acceleration work: on the theory of acceleration as a proxy for energy expenditure," *Methods in Ecology and Evolution*, vol. 2, no. 1, pp. 23–33, Feb 2010.

[7] S. Kaghyan and H. Sarukhanyan, "Activity recognition using k-nearest neighbor algorithm on smartphone with tri-axial accelerometer," *International Journal of Informatics Models and Analysis (IJIMA), ITHEA International Scientific Society, Bulgaria*, vol. 1, pp. 146–156, 2012.

[8] W. Strong, "Behavioral ecology of horn sharks, heterodontus francisci, at santa catalina island, california, with emphasis on patterns of space utilization (sharks)." 1990.

[9] M. A. Edmonds, P. J. Motta, and R. E. Hueter, "Food capture kinematics of the suction feeding horn shark, heterodontus francisci," *Environmental Biology of Fishes*, vol. 62, no. 4, pp. 415–427, 2001.

[10] A. P. Summers, R. A. Ketcham, and T. Rowe, "Structure and function of the horn shark (heterodontus francisci) cranium through ontogeny: Development of a hard prey specialist," *Journal of Morphology*, vol. 260, no. 1, pp. 1–12, 2004.

[11] K. Q. Sakamoto, K. Sato, M. Ishizuka, Y. Watanuki, A. Takahashi, F. Daunt, and S. Wanless, "Can ethograms be automatically generated using body acceleration data from free-ranging birds?" *PLoS ONE*, vol. 4, no. 4, 2009.

[12] E. Shepard, R. Wilson, F. Quintana, A. G. Laich, N. Liebsch, D. Albareda, L. Halsey, A. Gleiss, D. Morgan, A. Myers, and et al., "Identification of animal movement patterns using tri-axial accelerometry," *Endangered Species Research*, vol. 10, pp. 47–60, 2008.

[13] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. P. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Personal and Ubiquitous Computing*, vol. 14, no. 7, pp. 645–662, 2010.

[14] S. Sani, N. Wiratunga, and S. Massie, "Learning deep features for knn-based human activity recognition." 2017.

[15] S. Preece, J. Goulermas, L. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 3, pp. 871–879, 2009.

[16] M. Janidarmian, A. R. Fekr, K. Radecka, and Z. Zilic, "A comprehensive analysis on wearable acceleration sensors in human activity recognition," *Sensors*, vol. 17, no. 3, p. 529, Jul 2017.

[17] S. Rosati, G. Balestra, and M. Knaflitz, "Comparison of different sets of features for human activity recognition by wearable sensors," *Sensors*, vol. 18, no. 12, p. 4189, 2018.

[18] M. Janidarmian, A. R. Fekr, K. Radecka, Z. Zilic, and L. Ross, "Analysis of motion patterns for recognition of human activities," *Proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare - Transforming healthcare through innovations in mobile and wireless technologies*, 2015.

[19] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient knn classification with different numbers of nearest neighbors," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1774–1785, 2017.