Automatic Classification of Biological Targets in a Tidal Channel Using a Multibeam Sonar

EMMA COTTER AND BRIAN POLAGYE

University of Washington, Seattle, Washington

(Manuscript received 26 December 2019, in final form 4 May 2020)

ABSTRACT

Multibeam sonars are widely used for environmental monitoring of fauna at marine renewable energy sites. However, they can rapidly accrue vast volumes of data, which poses a challenge for data processing. Here, using data from a deployment in a tidal channel with peak currents of 1–2 m s^{-1}, we demonstrate the data-reduction benefits of real-time automatic classification of targets detected and tracked in multibeam sonar data. First, we evaluate classification capabilities for three machine learning algorithms: random forests, support vector machines, and *k*-nearest neighbors. For each algorithm, a hill-climbing search optimizes a set of hand-engineered attributes that describe tracked targets. The random forest algorithm is found to be most effective—in postprocessing, discriminating between biological and nonbiological targets with a recall rate of 0.97 and a precision of 0.60. In addition, 89% of biological targets are correctly classified as either seals, diving birds, fish schools, or small targets. Model dependence on the volume of training data is evaluated. Second, a real-time implementation of the model is shown to distinguish between biological targets and nonbiological targets with nearly the same performance as in postprocessing. From this, we make general recommendations for implementing real-time classification of biological targets in multibeam sonar data and the transferability of trained models.

1. Introduction

A major barrier to the development of marine renewable energy is the lack of information surrounding the interactions between marine energy converters and the environment (Copping et al. 2016). Such information needs to include animal behavioral patterns, predator–prey dynamics, and observations of any low-probability–high-consequence interactions such as collision between a marine fauna and a marine energy converter (Copping et al. 2016). Multibeam sonars are particularly attractive to collect observational data due to their range (tens to hundreds of meters, depending on frequency), independence from ambient light, and ability to track individual animals (Melvin and Cochrane 2015). Consequently, they have been used to monitor animal behavior around marine energy converters in the United States and Europe (Viehman and Zydlewski 2014; Lieber et al. 2017; Williamson et al. 2017). However, these observations have faced a perennial paradox: animal presence is sparse in time, such that duty-cycle sampling may not be effective (Cotter et al. 2017), but continuous data acquisition from a single sonar on interannual time scales can generate petabyte-scale data that are unreasonable to store or postprocess. As a result, data collection efforts to date have often been "DRIPy," data rich, yet information poor (Wilding et al. 2017). While putting a human "in the loop" to review data in real time and archive segments of interest can be effective (Hastie et al. 2014), this is not a scalable solution. If the human observer could be replaced by real-time target classification algorithms, it would be possible to ensure that archived data primarily consist of events of interest.

To date, processing of multibeam sonar data at marine renewable energy sites has relied on a combination of user-adjusted thresholds and human review. In Williamson et al. (2017), a multibeam sonar was used to monitor a tidal turbine structure. Data were continually recorded for a 2-week period and targets were automatically tracked in postprocessing using a nearest-neighbor algorithm. These target tracks were parameterized by morphometric measurements (e.g., size, intensity) and behavior (e.g., velocity, direction of travel) to aid in manual classification. Similarly, in Jepp (2017), targets were detected and tracked using a nearest-neighbors approach, then classified based on manually tuned target size thresholds. In addition, targets moving with a constant

Corresponding author: Emma Cotter, ecotter@whoi.edu

DOI: 10.1175/JTECH-D-19-0222.1

velocity were ignored under the assumption that they were passive objects advected by tidal currents. In Lieber et al. (2017), targets were automatically tracked in postprocessing using a Kalman filter, and these target tracks were used to limit human review to periods when targets were present. Francisco and Sundberg (2019) proposed a simple method for classifying targets in multibeam sonar imagery collected at a wave energy test site. Target length was the sole parameter used to distinguish between marine mammal species (e.g., gray seals and harbor seals), but the accuracy of the method was not evaluated (e.g., true or false positive rates). Additionally, all targets were assumed to be fish or marine mammals, and the presence of nonbiological targets that might have similar appearance in multibeam sonar data (e.g., kelp) (Cotter et al. 2017) was not explicitly considered.

Machine learning has the potential to reduce human review effort and multiple datasets collected at marine energy sites have been proposed as training data for future machine learning models (Williamson et al. 2017; Lieber et al. 2017). For example, the support vector machine algorithm has been shown to distinguish between seal and nonseal targets in multibeam sonar data with 95% accuracy (Hastie et al. 2019). Given this potential, the methodology to classify marine fauna in multibeam sonar data can benefit from developments in analogous classification problems. For example, in Dobeck et al. (1997), a two-layer neural network was used to classify naval mines in multibeam sonar data using hand-engineered features that described the size and intensity of detected targets. The neural network was trained using the subset of features that resulted in an optimal classification rate and was able to classify mines with the same probability as an expert human reviewer. Machine learning has also been used for automatic classification of birds in radar data collected at wind energy sites (Rosa et al. 2016). Radar imagery is similar to multibeam sonar imagery in that it can be used to detect targets at relatively long ranges, but the images are relatively low resolution. Rosa et al. (2016) evaluated six machine learning algorithms for classification of target tracks in radar data. A random forest algorithm performed best, achieving recall rates over 90% for discrimination between bird and nonbird targets, and recall rates between 81% and 83% for discrimination between specific groups of birds (e.g., herons and swallows).

Here, we evaluate the performance of three machine learning algorithms for automatically classifying marine fauna in multibeam sonar data. Training data consist of manually reviewed target tracks collected from a tidal channel. As track attributes are hand engineered, two feature selection methods are evaluated. Following this, classification performance is evaluated with varying volumes of training data to understand the requirements for adding new classes of target tracks to an existing model. Finally, classification is implemented in real time at second location, approximately 100 m away. Real-time performance is evaluated, classification models are refined, and recommendations are given for implementing machine learning classification at new marine energy sites.

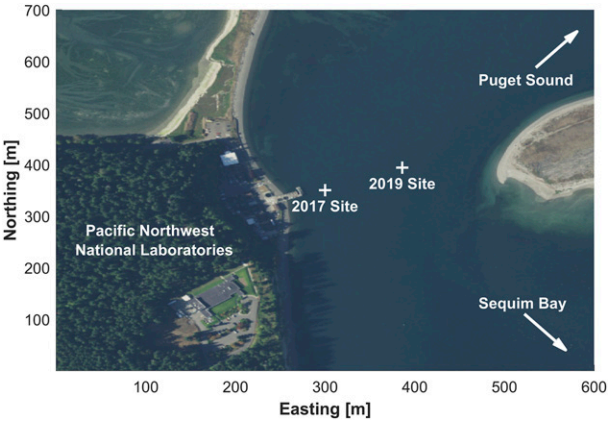

FIG. 1. Sequim Bay test site. The deployment sites where the 2017 and 2019 datasets were collected are indicated.

2. Methods

a. Data

1) TEST SITE

We collected data during two deployments in Sequim Bay, Washington, in 2017 and 2019, at the locations shown in Fig. 1. The site is a tidal channel at the mouth of Sequim Bay that is approximately 250 m wide and up to 10 m deep. In 2017, instrumentation were deployed at a location where the water was ~8 m deep (MLLW) with peak currents of ~1.5 m s^{-1}. The 2019 site was approximately 110 m northeast of the 2017 site, slightly shallower (~7 m MLLW), and experienced stronger currents (peak currents of ~2 m s^{-1}). The seabed at both locations consisted of cobbles, gravel, sand, and shell hash, with slightly more scour of finer grained sediments at the 2019 site due to stronger currents.

2) INSTRUMENTATION

Data were collected by an integrated instrumentation package (an ''adaptable monitoring package''). The multibeam sonar was a Teledyne BlueView M900-2250 (BlueView) with a nominal operating frequency of 2250 kHz and a ping rate of 5 Hz. Beamforming was performed by the manufacturer's propriety software,

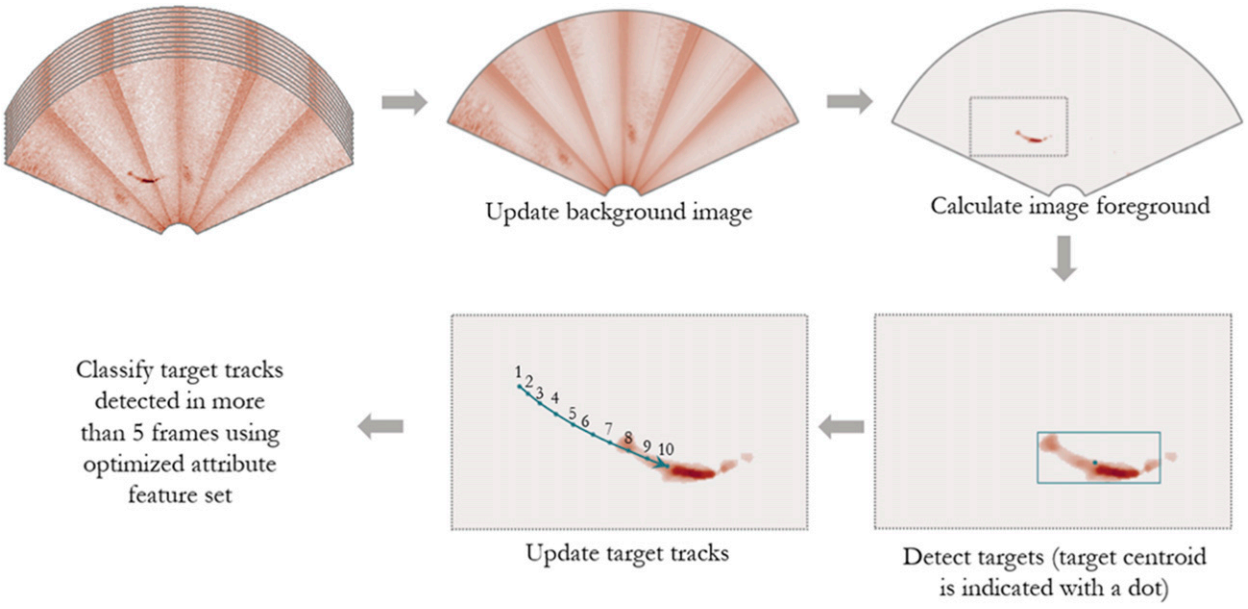

FIG. 2. Overview of target detection and tracking methodology.

ProViewer, which applies a gain correction to each beam based on manufacturer calibration. ProViewer has a user adjustable time-varying gain (TVG) setting to account for attenuation, which was set to $1\,\mathrm{dB\,m^{-1}}$. During data collection, we disabled automatic gain adjustment so that intensity values were comparable throughout the dataset. Following beamforming and gain correction, the resulting backscatter intensities were mapped to an 808×444 eight-bit Cartesian image in ProViewer.

The multibeam data were complemented by a collocated acoustic Doppler current profiler (ADCP) (Nortek Signature 500) and an optical camera system (Allied Vision Manta G201b), which is described in detail in Cotter et al. (2017). The ADCP collected data at 1 Hz with 0.5 m range bins. A 1-min average velocity at a range of 1.5 m was taken as representative of local currents. The ADCP and BlueView were synchronized using a hardware trigger to avoid active acoustic cross talk. The optical camera collected black and white imagery at 5 Hz, and their effective range varied with ambient light and water clarity (0 to 5+ m).

3) DATA COLLECTION AND PROCESSING

During both deployments, candidate targets in the multibeam sonar data were used to trigger data acquisition from the multibeam sonar and optical camera. The sequence for target detection, tracking, and classification is outlined in Fig. 2. This scheme is described in Cotter et al. (2017) and salient details are summarized here. Data acquisition and real-time processing used LabVIEW (National Instruments) and MATLAB (MathWorks). In LabVIEW, multibeam sonar data were stored in a 60-s ring buffer as they were acquired (newest data overwriting the oldest). The ring buffer served three purposes: 1) when an event was detected, data acquired immediately before the event were still stored in memory, and could be archived to provide contextual information; 2) data processing did not need to truly operate in "real time," as long as it operated at the same rate as data acquisition; and 3) data stored in memory could be used to establish an image background. The foreground of each sonar image was calculated in LabVIEW by subtracting a rolling background using the median of the previous 10 s of data and then applying a 5×5 pixel median filter. The foreground of each image was then stored in a temporary directory accessible by MATLAB. In 2017, the background image was updated at the same rate as data acquisition, but the update rate was reduced to 1 Hz in 2019 to reduce the computational load.

Target tracking and processing of the resulting target tracks each operated on a separate worker on a parallel pool in MATLAB. The target tracking worker continually read in new foreground images produced by LabVIEW. An intensity threshold was applied to each foreground image before detecting any targets with an apparent size that exceeded an area threshold in the image foreground and tracking them using a linear Kalman filter (Blackman 1986) with an initial velocity assumption of zero. Targets with centroid positions within 0.25 m of the predicted position of a track were associated with that track. If multiple targets were within this range,

the closest target was used. After each image was processed, any tracks that were updated in that image and had been detected in at least 4 previous images were sent to the processing worker. If a target track was detected in more than five images, it was sent to the processor again every time a new target was associated with the track. Conversely, shorter tracks that did not contain at least five target instances were never sent to the processing worker. Any target tracks that had not been updated for over 1.5 s were removed from memory. When the target tracking worker lagged data acquisition by more than 10 s, data older than 10 s were ignored until target tracking was able to "catch up" to real-time acquisition. This "pressure relief" feature was required to ensure stability in the processing pipeline, but did not have a significant impact on performance, as discussed in section 3d.

Each target track was described by a set of 29 hand-engineered features. As enumerated in appendix A, these included descriptions of target shape, target motion (including relative speed as compared to the observed currents), target intensity, and environmental covariates (e.g., time of day). We note that the apparent size and intensity of a target in the multibeam sonar imagery varied with the position of the target relative to the transducer, such that the features describing a target varied along the track. Further, as the multibeam sonar was uncalibrated, intensity was relative, not absolute, and the foreground intensity was used to describe each target. While multibeam sonars can be calibrated (Foote et al. 2005), calibration was not necessary for this study and is not typically used for general biological studies (e.g., Melvin 2016).

The processing worker was designed such that a user could configure the criteria for a target track to trigger data acquisition. Trigger criteria were user configurable: decisions could be made based on target track features (e.g., only archive data when a tracked target exceeds an area threshold) or environmental covariates (e.g., only archive data when current speeds exceed 1 m s^{-1}), or any combination of these criteria. Following the 2017 deployment, the processing worker was modified to enable decision-making based on the predicted class of a target (e.g., archive data when a target is predicted to belong to a specific class), as described in section 2d(1). Triggers were passed to the LabVIEW data acquisition code via UDP. Upon receiving a trigger, the LabVIEW software enabled artificial illumination for collocated optical cameras for 3 s if ambient light was insufficient to illuminate the image and waited 30 s to center the event in the 60-s ring buffer before archiving the entire buffer to disk.

b. Training and validation data

The target tracks in the archived multibeam sonar data were manually reviewed and used as training data for classification algorithms. The human reviewer (corresponding author of this study) classified each target track as a nonbiological target (N), seal (S), diving bird (B), fish school (F), or "small target" (Sm) on the basis of target appearance, evolution in time, and known species occurrence at the site. When possible, these annotations were verified with concurrent optical camera imagery, but such opportunities were limited by the functional range of that instrument (0–5 m). The "small target" class consisted of drifting kelp/eelgrass or individual fish that could not be reliably differentiated due to sonar resolution. The nonbiological target class included all target tracks that were not associated with flora or fauna: sonar artifacts (e.g., from side lobes of the transducer; Urban et al. 2017) and bubbles from waves, boat wakes, or diving fauna.

A representative example of each target class in the multibeam sonar is presented in Fig. 3. Radial sonar artifacts are apparent around the example "small target" and were classified as nonbiological targets. The fish school is relatively low intensity compared to larger, individual targets and is detected as multiple targets (multiple bounding boxes). The example of seal detection shows concurrent detection of two seals. The nonbiological target is a relatively low-intensity bubble cloud, likely from the wake of a passing boat. While a number of the targets have relatively similar representations in still imagery, their evolution in time is often distinctive. For example, while the diving bird and small target have similar acoustic representations, the bird was inferred to move rapidly in the vertical direction (time-varying apparent intensity without variation in horizontal position), while the small target was advected through the swath in the direction of the tidal current. The features of a target track, not the individual image representations, form the basis for human or machine classification.

The full dataset ($\mathbf{X}$) consisted of an $n \times m$ matrix where each column was a target track (m is the total number of target tracks in the dataset) and each row was a feature ($n = 32$, because one of the 29 features results in 4 separate values). The reviewer-assigned class for each target track was stored in a $1 \times m$ vector $\mathbf{y}$. Each feature vector $\mathbf{x}_n$ was normalized such that the 10th and 90th percentiles ranged between 0 and 1, respectively, as

$$\mathbf{x}_{n,\text{norm}} = \frac{\mathbf{x}_n - P_{10}(\mathbf{x}_n)}{P_{90}(\mathbf{x}_n) - P_{10}(\mathbf{x}_n)}, \tag{1}$$

where P_i represents the ith percentile of $\mathbf{x}_n$, and $\mathbf{x}_{n,\text{norm}}$ is the vector containing the normalized data. This approach was selected over normalizing by the maximum and minimum to prevent outliers from skewing the dataset.

The training data from 2017 were used to evaluate classification models in postprocessing, while the training

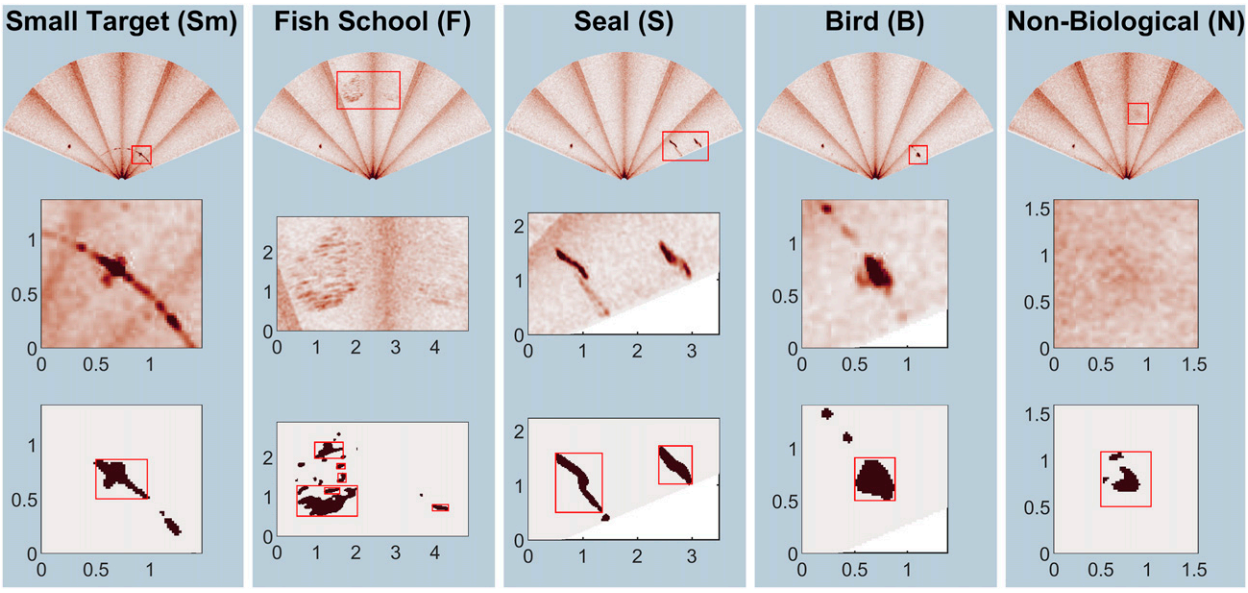

FIG. 3. Examples of each target track class. (top) Sonar image in which the target was detected, (middle) zoomed-in region around the target, and (bottom) binary image used for target detection. The red boxes in the bottom row indicate detected targets that were candidates for target tracking. All units are in meters, and the ''amp'' color map from the cmocean package (Thyng et al. 2016) is used.

data from 2019 were used to evaluate real-time classification model performance. The 2019 data are divided into two subsets—2019a and 2019b. Table 1 lists the dates of acquisition and the total number of target tracks available for each class in each dataset.

c. *Classification model evaluation*

We used the 2017 training dataset to evaluate the feasibility of automatically classifying targets in multibeam sonar data and to compare the performance of the *k*-nearest neighbor, support vector machine, and random forest algorithms for target track classification.

1) CLASSIFICATION ALGORITHMS

There are a variety of machine learning algorithms potentially suitable for target classification. For our application of classifying fauna at marine energy sites, an algorithm must satisfy three requirements. First, the algorithm must be able to discriminate between more than two types of classes (multiclass). Second, the algorithm should perform well with a relatively small training dataset, because some targets of interest (e.g., marine mammals) are relatively rare and training data annotation requires time-intensive expert review. Third, the algorithm must be able to predict the classification of a new target track relatively quickly to be suitable for real-time classification. Based on these requirements and results from prior work, three supervised machine learning algorithms were selected for evaluation: *k*-nearest neighbors (KNN), support vector machines (SVM), and random forests (RF).

In the nearest-neighbor algorithm, an unknown sample is assigned the class of the closest sample in the training dataset: its ''nearest neighbor.'' KNN

TABLE 1. Number of target tracks belonging to each class in each of the three datasets used to evaluate classification performance. The 2017 dataset was used to evaluate classification models in postprocessing, and the 2019 datasets were used to evaluate real-time performance.

Class	Abbreviation	Number of tracks		
		2017 (17–27 Mar 2017)	2019a (1–10 Mar 2019)	2019b (22–28 Mar 2019)
Nonbiological	N	2631	947	1988
Seals	S	147	46	11
Small targets	Sm	302	59	135
Birds	B	81	6	35
Fish schools	F	77	57	85

TABLE 2. Parameters tested for each machine learning algorithm. MATLAB defaults were used for all other parameters.

Algorithm	KNN (ensemble)	KNN (multiclass)	SVM	RF
MATLAB function	*fitcknn*	*fitcknn*	*fitcsvm*	*TreeBagger*
Parameter	*k*	*K*	Kernel function	Number of trees
Values	3, 5, 7	3, 5, 7	Radial basis function (RBF), linear, third-degree polynomial	40, 70, 100, 130, 160, 190

expands on this approach by taking a ''vote'' of the k-nearest neighbors, where k is a user-specified integer. These ''votes'' can be weighted by the distance of each neighbor from the unknown sample, making the algorithm relatively robust to outliers in the training data (Dudani 1976). This algorithm can be sensitive to the value of the parameter k (Zhang and Zhou 2005), and the computational requirement for classification of a new sample increases with the size of the training data and the value of k. The KNN algorithm works for any number of classes (multiclass).

The SVM algorithm finds the hyperplane that best divides a dataset into two classes (Burges 1998). A hyperplane is an n-dimensional division of the dataset, where n is the number of features describing each point in the dataset. The SVM algorithm is a kernel method, meaning that it uses a kernel function to map data into a higher-dimension feature space to find the optimal hyperplane (Burges 1998). For example, when $n = 2$, and a linear kernel function is used, the hyperplane is the line that best divides the data into two classes (each class is on one side of the line). While they have moderate computational requirements for training, SVMs can classify a new data point with low computational cost. SVMs are inherently binary classification models, though there are several methods in the literature for achieving multiclass classification (Duan and Keerthi 2005).

An RF is composed of multiple decision trees, a classification method that divides data into a set of ''nodes,'' or decision points (Breiman 2001). Single decision trees can perform well for classification, but are prone to overfitting (i.e., producing a model that is too specific to the training data and performs poorly for data that were not contained in the training dataset). However, when many decision trees are combined in an RF, the method is more robust. RFs have been shown to perform well with small training datasets (Baccini et al. 2004) and are a multiclass algorithm.

2) ALGORITHM IMPLEMENTATION

All three classification models were implemented in MATLAB (MathWorks). For each algorithm, we evaluated sensitivity to a core parameter, as described in Table 2, while MATLAB default values were used for all other parameters. Consequently, we make no claim that these are optimal implementations for the specific site and dataset; we are simply evaluating sensitivity to parameters with clear importance. Two model configurations were tested: multiclass models, where a single model was trained for all classes, and ensemble models, where a binary classification model was trained for each class, and each target track was assigned the class of the model with the highest probability of classification. For each set of KNN parameters, both multiclass and ensemble models were tested. The probability of classification for KNN models was calculated using the posterior probability. Because SVM is a binary classification algorithm, only ensembles of binary models were tested. The probability of classification for SVM models was calculated using Platt's method (Platt 1999). RF is inherently an ensemble approach, so only multiclass configurations were tested.

3) MODEL VALIDATION

The holdout method was used for cross validation of each classification model. The training data were randomly divided into two subsets: training data (70% of the data) and validation data (30% of the data). The training and validation subsets were stratified (e.g., equal proportions of data from each class). To address the disparity in the number of tracks belonging to each class, each data subset was randomly subsampled (Japkowicz 2000; Wallace et al. 2011) to provide an equal number of training data points from each class. We repeated this cross-validation process 100 times for each classification model to evaluate variability introduced by subsampling.

Classification model effectiveness can be described in many ways. Here, we present a set of metrics selected to be informative for environmental monitoring at marine energy sites, where true positive classifications (identification of animal presence) are of greater relative importance than false positive classifications (archiving or reviewing data that do not contain a target of interest). The recall rate, also called sensitivity or true positive rate, for each class R_c was calculated using the predicted classes of the validation data as

$$R_c = \frac{\mathrm{TP}_c}{N_c}, \tag{2}$$

where TP_c is the number of correct classifications belonging the class, c (e.g., true positives), and N_c is the total number of target tracks in the validation dataset belonging to the class c. The recall rate for all biological targets (flora and fauna) R_{bio} was calculated as

$$R_{bio} = \frac{TP_S + TP_{Sm} + TP_B + TP_F}{N_S + N_{Sm} + N_B + N_F}. \quad (3)$$

The R_{bio} metric is important in our application because it is maximized when targets of biological interest are correctly classified. The correct classification of nonbiological targets is of less significance for monitoring at marine energy sites because these are neither flora nor fauna. The recall rate for binary classification of biological targets R_{bin} was calculated as

$$R_{bin} = \frac{TP_{bio}}{N_{bio}}, \quad (4)$$

where TP_{bio} is the number of validation target tracks belonging to biological targets (i.e., seals, fish schools, small targets, or birds) that were not predicted to be nonbiological targets, but were not necessarily predicted to belong to the correct biological class, and N_{bio} is the total number of biological targets. This metric provides insight into the rate at which targets of interest would be rejected as nonbiological targets. This is relevant because one application of a real-time classification model would be to restrict archived data to sequences containing biological targets. Finally, the precision for binary classification of biological targets P_{bin} was calculated as

$$P_{bin} = \frac{TP_{bio}}{NP_{bio}}, \quad (5)$$

where TP_{bio} is the number of correctly classified biological targets, and NP_{bio} is the total number of targets predicted to be biological. This metric provides insight into the volume of data that does not contain targets of interest, but would be archived if the classification model were used to filter sequences containing biological and nonbiological targets. Other metrics, such as accuracy (Fawcett 2006), that aggregate true positives and negatives into a single metric are more relevant in applications where true positives and true negatives are of equal value.

4) FEATURE SELECTION

While hand-engineered features have been employed in prior ecological classification using machine learning (e.g., Hastie et al. 2019; Rosa et al. 2016), a challenge inherent to their use is that the feature set might contain redundant or weak features that reduce classification performance. To address this, we tested two feature selection methods. There are two general categories of feature selection methods: filter methods and wrapper methods (Kohavi and John 1997). Filter methods take only the training data into account, and select features based on some measure of the information that they contain. Wrapper methods identify the subset of features that provides optimal performance for a particular classification model. To evaluate the consequence of the hand-engineered feature set on classification performance, each classification model was tested with three sets of features: 1) the entire set of 32 features, 2) a subset of features selected using a filter method, and 3) a subset of features selected using a wrapper method. While there are numerous filter and wrapper methods in the literature (e.g., Kohavi and John 1997; Song et al. 2010), we chose two representative approaches. The filter method selected the 16 features whose feature vectors x_n had the highest linear correlation coefficients with the class vector $\mathbf{y}$. The wrapper method used a hill-climbing algorithm with the objective of optimizing R_{bio}, adapted from Dobeck et al. (1997) and detailed in appendix B.

TABLE 3. Data volumes used to evaluate the effect of reduced training data volumes. Note that data were trimmed to have equal representation from all classes before classification (77 tracks from each class). Thirty percent of the available training data (23 target tracks from each class) were used for validation in each case.

Percentage of training data retained	70	60	50	40	30	20	10
Number of tracks per class	54	46	38	31	23	15	8

5) MODEL SENSITIVITY TO TRAINING DATA

The most time-intensive step in developing a classification model is the annotation of training data. Consequently, we repeated the analysis in section 2c for the top-performing classification models (based on R_{bio}) with sequentially reduced volumes of training data (Table 3). The same fraction of training data (30% of the undersampled training dataset) was used for validation in each case to avoid biasing results.

d. *Real-time evaluation*

Classification was implemented in real time during the 2019 deployment. The primary objective of this test was to demonstrate the ability to classify target tracks in real time. While real-time classification may not be necessary for all applications, as previously discussed, it can be used to limit data volumes more selectively than the detection thresholds currently employed. For example, at a site where one type of target (e.g., fish schools) is observed frequently, real-time classification could restrict data acquisition to only record full-bandwidth

data for other target classes while logging only the features of fish schools. In addition, the 2019 deployment provided unexpected insight into the portability of the classification models between locations.

1) Software architecture

The data collection software was modified to enable real-time classification using a model trained with existing training data. Classification occurred on the processing worker, described in section 2a(3). For each target track received by the processor, all features listed in appendix A were calculated. These features were then normalized by Eq. (1) and a trained model was used to predict the class of the target track. Track features were logged so that tracks could be reclassified with a different model in postprocessing. The time lag between data acquisition and target track classification t_{lag} is used to assess real-time implementation feasibility. Time lag t_{lag} was calculated as the difference between the time stamp of the sonar image for the last target associated with a track and the time when that track was assigned a class.

2) Implementation

Based on the performance results presented in section 3a, a random forest classification algorithm using 100 trees and features selected by the hill-climbing algorithm was used for real-time classification. Initially, the model was trained using the 2017 dataset and the processing code was configured to generate a trigger for any target track predicted to be a target of interest (e.g., a seal) or any target of potential interest that was detected for more than one second and had a 75th-percentile area greater than 0.15 m^2 and mean intensity greater than 60 (the maximum intensity of 8-bit images is 255). These thresholds were manually tuned to allow the maximum number of feasibly reviewable false positive detections. By the nature of this threshold method, the detection rate likely varies by target class, and reported classification results only pertain to targets that were successfully detected and tracked for more than one second. Because prior evaluation using similar thresholds (Cotter et al. 2017) demonstrated a recall rate for *detections* of 99%, we have high certainty that detectable target classes are well represented in these data.

A total of 110 one-minute sequences collected using this scheme were initially reviewed, and the 1115 target tracks in these sequences formed the 2019a dataset (Table 1). These data revealed that the classification model trained with 2017 data performed relatively poorly in real time at the new deployment site (see section 3d). To address this, the 2019a dataset was pooled with the 2017 dataset, and the combined dataset was used to retrain the classification model. The updated classification model was then implemented in real time and an additional 326 one-minute sequences containing 2254 target tracks were manually reviewed. These target tracks formed the 2019b dataset (Table 1), and were used to evaluate the updated classification model. Classification using only the 2019a dataset for training was not attempted due to the relatively small volume of training data for some classes (e.g., seals).

3) Evaluation

When calculating the metrics presented in section 2c to assess the performance of real-time classification, tracks were considered to be correctly classified if they were predicted to belong to the reviewer-assigned class for the majority of the track. This distinction is necessary because a track could be reclassified multiple times as the target was sequentially detected. For example, if a target track associated with a seal was sent to the processor 10 times, it would be considered correctly classified (a true positive) if it were classified as a seal at least five times.

We note that the results presented in section 3d(2) are for targets that were detected and tracked in real time. However, while a prediction of the class of each target was made in real time, the results shown were recalculated in postprocessing because classification methods were refined over the course of the deployment, and the logs generated by the real-time processing code enabled reclassification of target tracks in postprocessing by similar models. The models tested in postprocessing should achieve the same results when implemented in real time, because target detection, tracking, and calculation of the features used for classification occurred in real time. Additionally, the same algorithm (RF) was used in real time and postprocessing (albeit with varying training data and feature lists), and tracks of interest were reliably assigned a predicted class with $t_{lag} < 10$ s (see section 3d).

3. Results and discussion

a. Classification model performance

Figure 4 shows the median value of R_{bio} for all classification models tested. In general, models were relatively insensitive to algorithm parameters but highly sensitive to feature selection. Figure 5 shows median and interquartile range for all classification metrics for the models with the highest value of R_{bio} using each algorithm (KNN, SVM, and RF). The full set of metrics for all tested classification models is given in appendix C.

Relatively high performance was achieved for each algorithm, and all three algorithms had binary classification

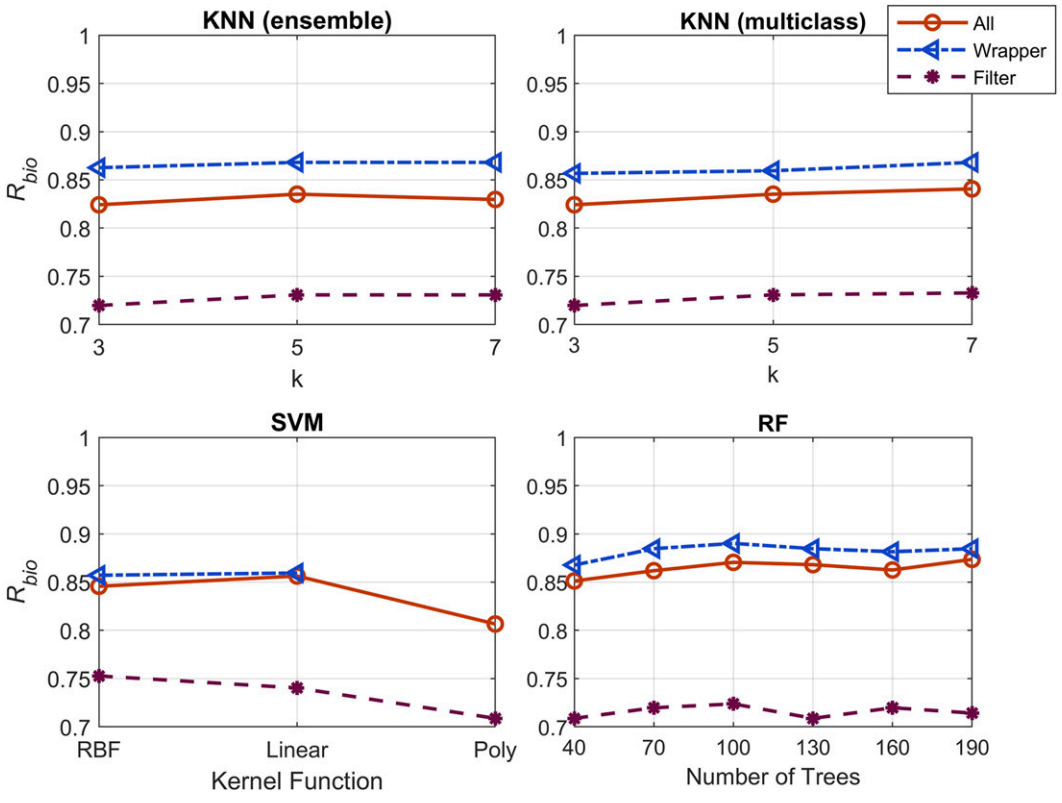

FIG. 4. Median R_{bio} for each classification model tested, separated by algorithm and feature selection method. Note that the y-axis minimum is 0.7 (70% recall rate for all biological targets).

recall rates R_{bin} greater than 0.95, indicating that targets of biological interest were infrequently classified as nonbiological targets. This metric also showed the least variability with the subset of data selected for training (i.e., smallest interquartile range for performance metrics). As shown in Fig. 5, the RF model demonstrated superior performance for most metrics, though each algorithm performed ''best'' by at least one metric.

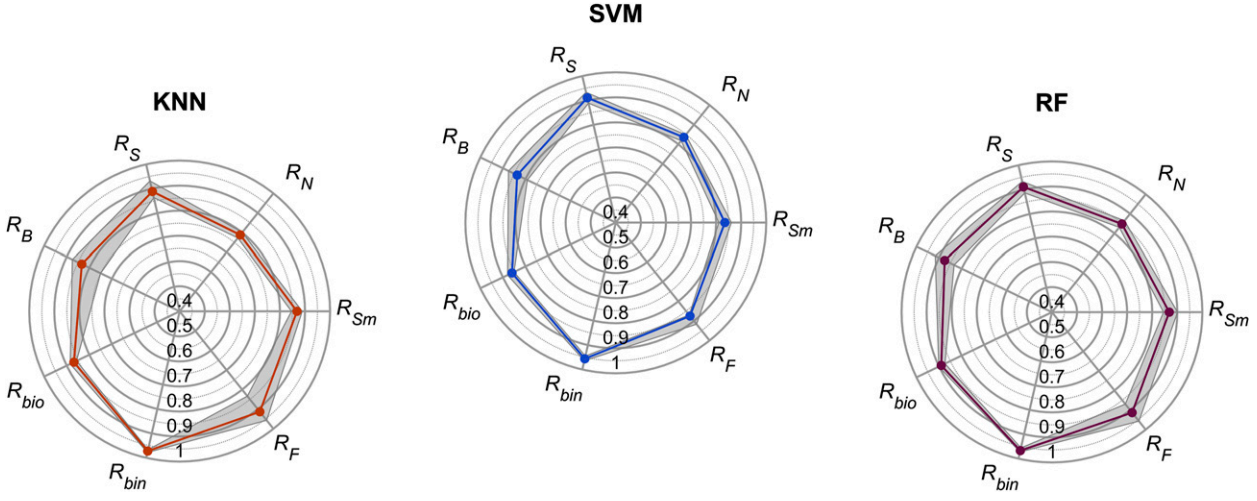

FIG. 5. All recall rates for the top-performing classification model using each machine learning algorithm. The colored line indicates the median value for 100 iterations of classification model validation, and the interquartile range is indicated by the gray shaded region. Note that the center of the plot is 0.4 (i.e., 40% classification rate).

The RF model was best at classifying diving birds and nonbiological targets ($R_B = 0.88$ and $R_N = 0.84$), while the SVM model and RF model achieved similar performance for classification of seals ($R_S = 0.91$), and the KNN model and RF model achieved similar performance for classification of schools of fish ($R_F = 0.91$). The RF model had the highest values of both R_{bio} (0.89) and P_{bin} (0.60, compared to 0.51 and 0.57 for KNN and SVM, respectively). Superior performance of the random forest algorithm for this class of problem is consistent with prior research on radar target track classification (Rosa et al. 2016).

b. *Feature selection*

As shown in Fig. 4, the wrapper method (hill-climbing algorithm) improved classification performance for all models, with an average change in R_{bio} of +0.02 relative to classification using the full set of hand-engineered features. Conversely, the filter method (correlation) reduced performance for all models relative to classification using the full feature set, with an average change in R_{bio} of −0.12. This disparity is a consequence of the redundant hand-engineered features (e.g., multiple statistical descriptions of target size), which are removed by the hill-climbing algorithm but are reinforced by the correlation method.

Figure 6 shows the percentage of the optimized feature lists that each feature appeared in, the individual features selected for each model, and the features selected during the first and second iteration of the hill-climbing algorithm. The features selected by the hill-climbing algorithm varied with classification model. The lengths of feature lists ranged from 13 to 19 features, with a median length of 14.5, and all but one candidate feature appeared in at least one optimized feature list. The hill-climbing algorithm always selected the time of day (T) feature first or second. This is unsurprising, because, in the 2017 training data, seals and fish schools were almost exclusively present at night, while diving birds were almost exclusively present during daylight hours. This feature selection is consistent with time of day being often included as an explanatory variable in ecological models, such as generalized additive models (Rothery and Roy 2001). Similarly, birds and seals were only present around slack water in the training data, making current speed (currSpd) an important feature for several models. The major axis length (Maj) was the only shape feature selected by all models, and was the first or second feature selected for several models. Measures of target area (A, A_{std}, and A_{max}) were selected infrequently, despite being perceived as an important feature for human review. This is likely because the apparent target area changes with target orientation relative to the transducer.

The superior performance of classification models with feature lists optimized by the hill-climbing algorithm reinforces Dobeck et al.'s (1997) conclusion that wrapper feature selection methods that explicitly account for algorithm strengths and weaknesses are preferable to filter-based approaches based purely on the feature set. This approach is robust for hand-engineered feature sets, because it can remove poorly defined or redundant features and features that may match human intuition (e.g., target area), but do not aid in automatic classification. While the hill-climbing approach requires moderate computation time (up to 12 h on a computer with an i7 processor and 16 GB of RAM), this is not expected to be a hindrance to its application to real-time classification. Calculation of an optimal feature set only needs to occur at the time that the model is trained, separate from real-time implementation.

c. *Model sensitivity to training data*

Figure 7 shows R_{bio} for the three classification models in Fig. 5 with varying volumes of training data. As expected, performance generally improves with increasing volumes of training data, though with decreasing marginal benefit. The RF shows superior performance with nearly all volumes of training data, and reaches 99% of the value of R_{bin} with the full training dataset with 38 samples from each class. Overall, this indicates an advantage of the RF model: for automatic classification of infrequently appearing marine animals, high recall rates can be achieved with relatively limited training data.

d. *Real-time implementation*

1) SOFTWARE PERFORMANCE

The real-time target tracking, classification, and triggering software was stable over the course of the deployment, with no software crashes that were not the result of software upgrades (e.g., debugging after implementation of new features) or power outages. The tracking code was able to keep up with data acquisition reliably—tracking lagged data acquisition by more than 10 s on only 63 occasions over a 15-day period (an average of 4.2 times per day). On average, it took the target tracking software 9.5 s to recover and resume normal operation.

Target classification lagged target tracking due to the sequential nature of the real-time architecture. However, with the ring-buffer architecture, a target only needed to be classified within 30 s of detection for associated data to be recorded. When more targets were present, the time

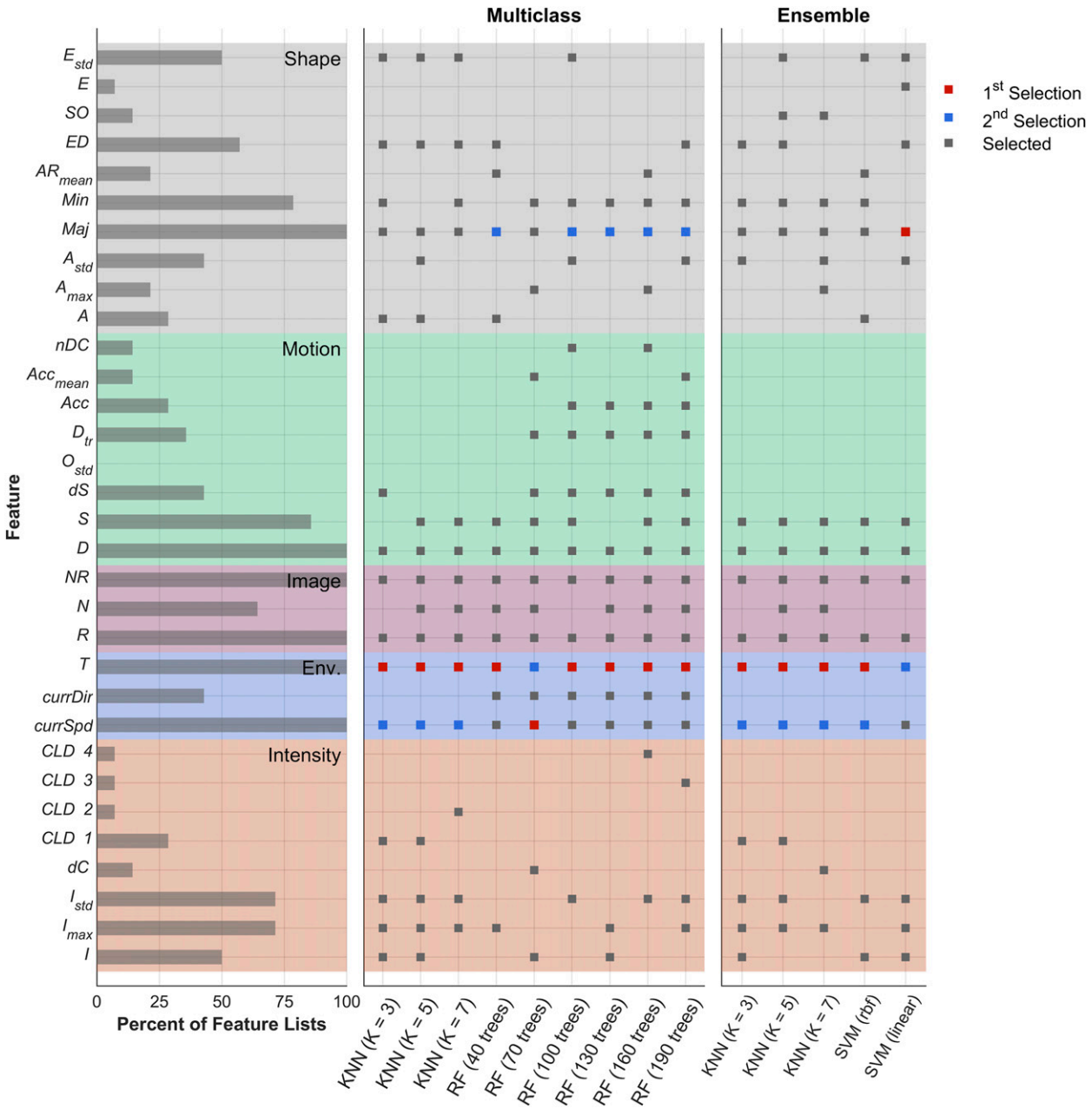

FIG. 6. Features selected by the hill-climbing algorithm. (left) Percentage of all optimized feature lists in which each feature is included. (center),(right) Features selected by each individual model. For all feature lists, the first feature selected by the hill-climbing algorithm is indicated in red, and the second feature selected is indicated in blue. The shaded regions denote the general category of feature (i.e., shape, motion, image, environmental, and intensity descriptors). Features are defined in appendix A.

required to track and classify each target track increased. Figure 8 shows the cumulative density functions of t_{lag} for each target class. The shortest t_{lag} was approximately 0.3 s, 100% of biological targets were classified within 10 s, and 100% of seals were classified within 0.5 s. Ninety percent of nonbiological target tracks were classified within 30 s. Seals were consistently classified fastest because they typically were the only target in the field of view at the time of detection. Conversely, nonbiological targets took longer to classify because they were frequently detected concurrently (i.e., many simultaneous targets), which slowed down the tracking and feature calculation step in the processing pipeline. We note that t_{lag} would vary with the computational power of the control computer, but expect consistency in the relative trends.

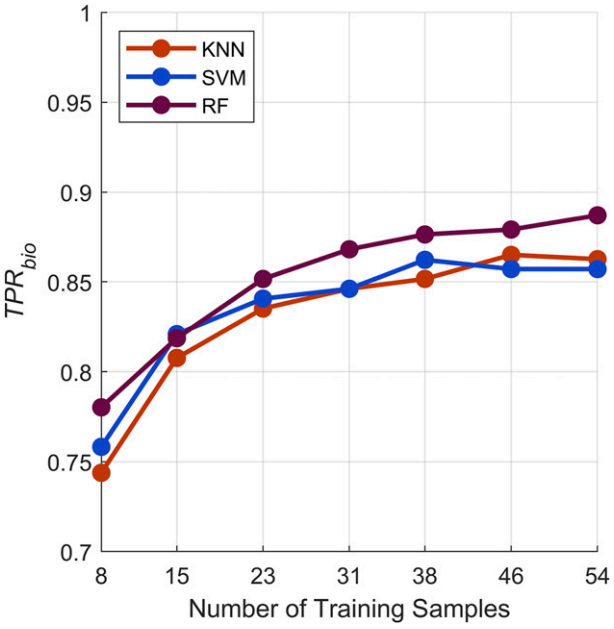

FIG. 7. R_{bio} for the top-performing classification model using each algorithm with varying volumes of training data. For each model, the same number of samples from each target class was used (number of training samples). The median value for 100 iterations of model cross validation is shown. The slight deviation from Fig. 5 is a consequence of the randomized training data selection for each iteration.

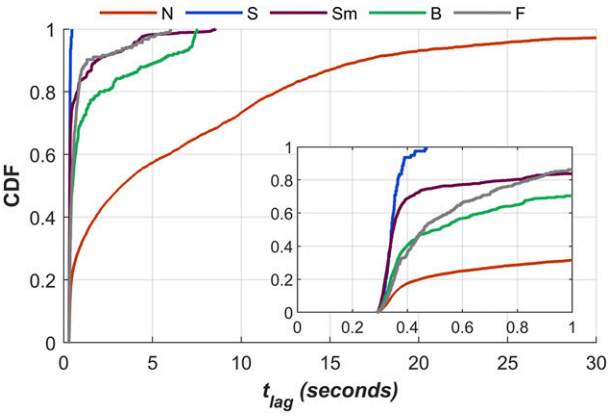

FIG. 8. Cumulative density functions of the time to detect, track, and classify a target (t_{lag}) for each target class [nonbiological (N), seals (S), small targets (Sm), diving birds (B), and fish schools (F)]. The inset highlights the region of the plot where the time to track classification was less than 1 s.

2) CLASSIFICATION PERFORMANCE

Figure 9 compares real-time classification performance for the original classification model (trained using only the 2017 data) and the updated classification model (trained using a blend of the 2017 and 2019a data). Performance using the original model is quite poor when compared to the results using the 2017 data for training and validation in postprocessing (Fig. 5). However, the updated classification model that included site-specific training data improved by nearly every metric. This was an unexpected outcome and a consequence of several key differences between the 2019 datasets and the 2017 dataset. First, fish schools were observed much more frequently in 2019. The 77 fish-school target tracks in the original training dataset were observed in only 11 sequences (e.g., 11 fish schools were tracked, but they were detected as multiple targets). It is likely that this small sample size resulted in overfitting of the original classification model for this class (e.g., classification of fish schools relied on trends that were not be present in a larger training dataset) and resulted in an extremely low classification rate for 2019 fish schools using the original model ($R_F < 0.3$). However, after retraining, fish schools are correctly classified 84% of the time ($R_F = 0.84$). Second, nonbiological targets in 2019 had different characteristics—bubbles were more frequently observed at longer ranges, likely due to the stronger currents. This resulted in a relatively low binary recall rate ($R_{bin} = 0.72$) using the original model, because these targets were frequently misclassified as biological targets. In contrast, the updated classification model achieved a R_{bin} value of 0.95, a value nearly equal to that achieved in postprocessing using the 2017 dataset. Third, seal activity differed in two important ways between datasets. In the 2017 data, seals were observed only during the night, possibly because artificial light from the dock provided an advantage to hunting fish. In the 2019 data, seals were observed day and night. As discussed in section 3e, because the time-of-day environmental covariate is an important parameter (Fig. 6), this degraded classification performance and highlights a trade-off inherent to the inclusion of environmental covariates in classification models. Additionally, seals were more commonly observed at close range in 2019. Specifically, 63% of seals in the 2019a and 2019b datasets were observed at a range less than 2 m, while <10% of seals in the 2017 training data were observed inside this range (Cotter et al. 2017). As shown in Fig. 10, at this range, a seal produces high-intensity sonar artifacts, and accurate manual classification is only possible with concurrent optical camera data. This resulted in relatively low classification rates for seals using either model—R_S for both the original and updated classification models was below 0.6.

With the original model trained using the 2017 dataset, 19% of target tracks classified as biological targets were also annotated as biological targets ($P_{bin} = 0.19$). In other words, if a human reviewed all target tracks classified as biological targets, only 19% would be

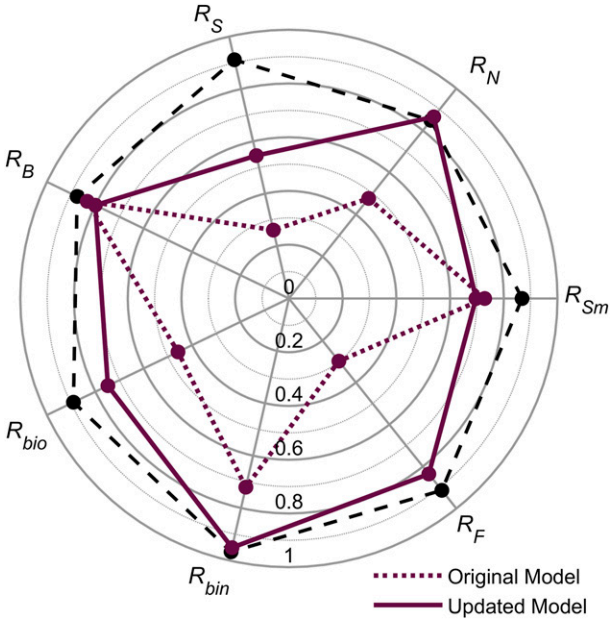

FIG. 9. Real-time classification results, using the original classification model trained with only the 2017 dataset and the updated classification model trained with the 2017 and 2019a datasets. The 2019a dataset was used to validate the original model, and the 2019b dataset was used to validate the updated model. The black dashed line shows the postprocessing performance of the RF model trained and validated using only the 2017 dataset (from Fig. 5).

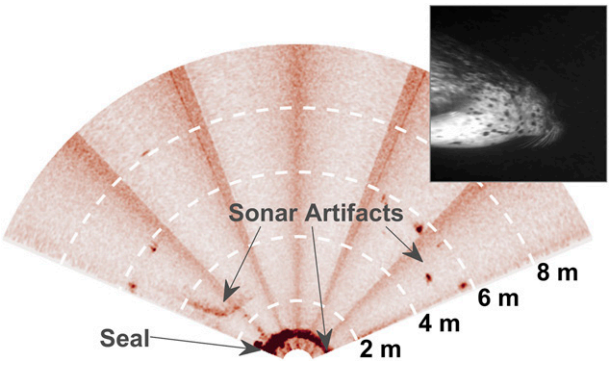

FIG. 10. Seal detected at <1 m range from the transducer. The seal is easily classified by a human reviewer using (top right) concurrent optical data but difficult to distinguish in the sonar data due to high-intensity sonar artifacts.

potential targets of interest. This metric improved significantly for the updated model, increasing to 47%. This represents a significant reduction in human review effort compared to review of all sonar data containing target tracks.

Finally, because of the undersampling used to balance the number of target representations in the training data, real-time classifier performance has a moderate sensitivity to the specific tracks used for training. For example, in five replications, R_{bin} ranged from 0.94 to 0.96. This is in line with the performance ranges observed in postprocess evaluation of 2017 data (e.g., Fig. 5). While uncertainty could be reduced by running an ensemble of classifiers in real time, this would incur a significant computational cost for limited benefit.

e. *Use of environmental covariates*

Harbor seal presence and behavior is often correlated with environmental covariates, such as time of day, stage of the tide, and weather (Schneider and Payne 1983; Calambokidis et al. 1987; Grellier et al. 1996; Hamilton et al. 2014). Consequently, including such factors improved performance of our classification models. As shown in Fig. 11, when environmental covariates are excluded from the feature set, recall rates for birds and seals substantially decrease. This is an advantage of the machine learning approaches tested here relative to image-based deep learning, which cannot easily incorporate environmental covariates. However, such correlation is often location dependent, such that the inclusion of environmental covariates would be expected to reduce classifier portability between sites. This was the case for the present study, even though the distance between locations was small in absolute terms (~100 m) and the two locations had similar physical characteristics. As such, there is a fundamental performance-portability trade-off associated with environmental covariates that should be considered in model formulation and application.

f. *Recommendations for implementation at a marine energy site*

We have demonstrated that machine learning classification of targets in multibeam sonar data can achieve relatively high accuracy, with the exception of classification of large targets detected at close range (e.g., seals ≤2 m from the transducer). Such targets are better treated by optical observations. Classification model effectiveness appears to be strongly site specific, which is consistent with target classification using analogous radar imagery (Rosa et al. 2016). More substantial differences were observed for the same classification model across two locations than for a range of classification models at the same location. Site-specific variation is exacerbated at marine energy sites, particularly tidal energy sites, where high-energy conditions create dynamic environments with different characteristics at spatial scales on the order of 100 m or less (Polagye and Thomson 2013).

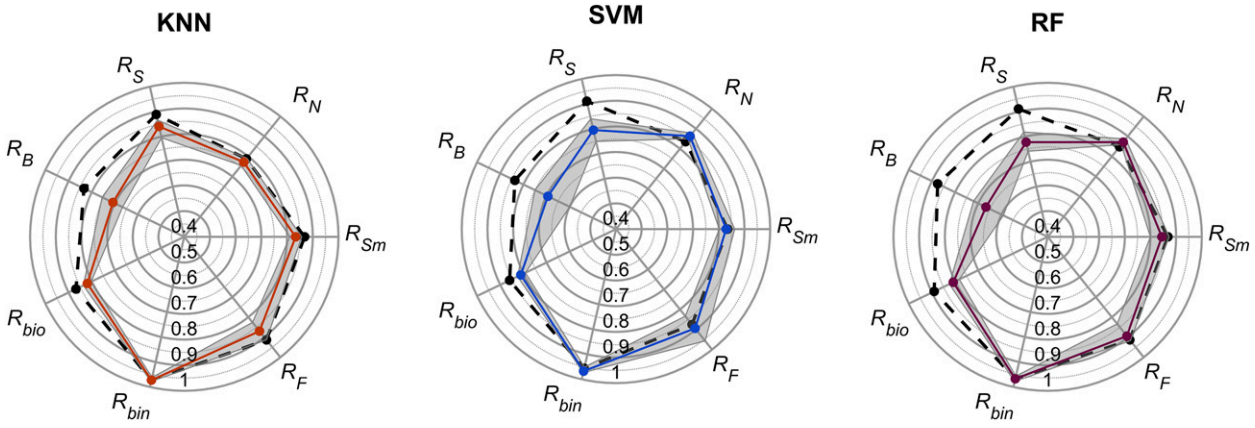

FIG. 11. Results for the top-performing classification model using each machine learning algorithm trained without environmental features. The colored line indicates the median recall value for 100 iterations of classification model validation, and the interquartile range is indicated by the gray shaded region. The black dashed line indicates the results shown in Fig. 5 that included environmental features. Note that the center of the plot is 0.4 (40% recall rate).

We have also demonstrated that binary classification (i.e., biological vs nonbiological) is more effective than taxonomic classification (i.e., target belongs to ''seal'' class). This is also consistent with the findings of Rosa et al. (2016) for classification of birds tracked in radar data. Based on this, we recommend that real-time binary classification of biological targets be used to control data acquisition at marine energy sites, because the ''cost'' to store additional data is relatively low compared to the risk of missing rare fauna. Using this approach, environmental interactions of potential interest can be archived with a relatively high recall rate (over 95%). This would also produce a manageable volume of archived data compared to continuous acquisition, which would allow efficient human review. Because real-time taxonomic classification showed lower recall rates, but still performed relatively well, these results could be used to control automatic adaptive action (e.g., enabling mitigation measures, such as fish deterrents), where the ''cost'' of an incorrect classification is lower.

We recommend the following procedure to train a classification model at a new marine energy site:

- Implement real-time target tracking, using manually tuned thresholds to trigger data acquisition (e.g., apparent size and intensity in sonar imagery). These thresholds should be set based on the smallest/least intense target of interest (e.g., area thresholds should be higher if marine mammals are the only targets of interest, and lower if detection of fish is important). Tuning of thresholds may require iterative human review if either too many or too few targets of interest are being recorded.
- Have a human reviewer annotate recorded target tracks to the finest taxonomic level possible and use this information to build a binary classification model that distinguishes between biological and nonbiological targets. We recommend using a random forest algorithm and a wrapper feature selection method, as this method was relatively insensitive to algorithm parameters and required the least training data to achieve a high recall rate for all classes of interest. Regardless of the algorithm employed, wrapper feature optimization is recommended.
- Evaluate the model in postprocessing using the methods outlined in section 2c. If the model achieves satisfactory performance for the application, implement in real time to limit data acquisition to periods when a biological target is likely present, thereby minimizing the volume of data requiring curation. If the model does not achieve satisfactory performance, continue to acquire additional training data to increase the volume of available training data.
- When finer taxonomic classification is desired, continue acquiring data and reviewing target tracks until approximately 40 target tracks are identified in each class of interest, then retrain the model with these classes specified. We recommend that this model be used to control automatic adaptive action in real time or to guide human review in postprocessing.

Because intensity values are on a logarithmic scale, we recommend using the raw intensity values (e.g., before background subtraction) in a more dynamic environment to describe each target (i.e., an environment in which background intensity changes significantly in time or space). This approach can also be applied to existing datasets in postprocessing. Specifically, the model can be trained and validated using a subset of the available data and then used to automatically process the remaining

data. Such a method could be used to rapidly evaluate temporal and spatial trends in class presence/absence. Additionally, the same target-tracking approach to classification could be implemented with other multibeam sonars, though sonar-specific training data would be required because of variations in apparent target size and intensity. Multibeam sonar calibration (Foote et al. 2005) could streamline this process and provide additional information for human or machine target classification (Trenkel et al. 2008). Translation of classification models between sonars is a potential topic for future research, as is using the predicted probability of classification to identify aberrations that may be indicative of a new class in the sonar imagery.

4. Conclusions

Automatic classification of seals, diving birds, small targets, fish schools, and nonbiological targets in multibeam sonar data using an optimal set of hand-engineered features has been demonstrated in postprocessing and in real time. In postprocessing, using a random forest algorithm, we achieved a recall rate of 0.89 for biological targets. In real time, after limited retraining with site-specific data, we were able to distinguish between biological and nonbiological targets with a recall rate of 0.95 and a precision of 0.60. Development of a classification model at a new test site is expected to be possible with relatively small training datasets (<40 samples from each class). In aggregate, this method appears to be broadly applicable within the marine energy sector to gather information needed to contextualize or mitigate environmental risks.

Acknowledgments. The authors wish to acknowledge Paul Murphy for his work developing data acquisition software, James Joslin, Paul Gibbs, and Harlin Wood their support of the deployments described in this paper and hardware development that made these deployments possible, and Mitchell Scott for productive conversations about machine learning algorithm selection. The authors also wish to acknowledge John Vavrinec, Garrett Staines, Kate Hall, Genvra Harker-Klimes, Kailan Mackereth, and Sue Southard for their support of field operations at Pacific Northwest National Laboratory in Sequim, Washington. Portions of this work were funded by the U.S. Department of Energy under DE-EE0007827. Emma Cotter was supported by a National Science Foundation Graduate Research Fellowship under Grant DGE-1762114.

Data availability statement. The 2017, 2019a, and 2019b training datasets are publicly available in the University of Washington institutional data repository: http://hdl.handle.net/1773/43749.

APPENDIX A

Track Features

The hand-engineered features used to describe each target track are listed in Table A1 and are chosen to represent aspects of target shape, intensity, motion, environmental covariates, and image covariates. These are statistical quantities derived from all targets associated with a track.

Features associated with shape were calculated using the MATLAB *regionprops* function, and are derived from the following:

- Area: The sum of the area of all pixels associated with the target (i.e., pixels above the background intensity threshold). Pixel area is calculated from the Cartesian sonar image using the known minimum and maximum range of the sonar swath as a scale.
- Major-axis length: The length of the major axis of an ellipse with the same second moment as the target.
- Minor-axis length: The length of the minor axis of an ellipse with the same second moment as the target.
- Axis ratio: Ratio of the *major axis length* to the *minor axis length.*
- Equivalent diameter: The diameter of a circle with the same *area* as the target.
- Extent: The ratio of the total number of pixels in a target to the area of the target bounding box.
- Solidity: The ratio of area of the target to the area of the smallest polygon that can contain the target.
- Orientation: Angle between the *major axis* and x axis of the sonar image.
- Position: Location of the weighted centroid of the target.
- Range: Distance from sonar to target position.

Features associated with the pseudocolor appearance of the target in the sonar imagery are derived from the following:

- Intensity: The set of color values for pixels associated with a target.
- Color layout descriptor: 2×2 color layout descriptor (Kasutani and Yamada 2001). This produces four features for each target.
- Weighted centroid offset: Distance between the geometric centroid and intensity-weighted centroid for a target.

Features associated with target motion are derived from target *position*:

TABLE A1. Features used to describe each target track. Features are separated into descriptions of target shape (shape), target motion (motion), target intensity (intensity), the image covariates (image), and the environmental covariates (env.). The minimum and maximum values observed for each feature are listed.

Feature	Abbreviation	Type	Description	Units	Min	Max
Extent	E	Shape	75th percentile of the *extent* of all targets associated with the track	—	0.21	0.98
Extent standard deviation	E_{std}	Shape	Standard deviation of the *extent* of each target associated with the track	—	0	0.33
Solidity	SO	Shape	75th percentile of the *solidity* of all targets associated with the track	—	0.48	1
Equivalent diameter	ED	Shape	75th percentile of the *equivalent diameters* of all targets associated with the track	Pixels	7.2	142.6
Axis ratio	AR	Shape	Mean of the *axis ratio* of all targets associated with the track	—	0.14	0.97
Minor axis length	Min	Shape	75th percentile of the *minor axis lengths* of all targets associated with the track	Pixels	4.9	361.6
Major axis length	Maj	Shape	75th percentile of the *major axis lengths* of all targets associated with the track	m	8.3	361.6
Area	A	Shape	75th percentile of the *areas* of all targets associated with the track	m^2	0.02	8.2
Area standard deviation	A_{std}	Shape	Standard deviation of the *area* of each target associated with the track	m^2	8.4×10^{-4}	4.4
Maximum area	A_{max}	Shape	Maximum *area* of any target associated with the track	m^2	0.02	9.1
Direction changes	nDC	Motion	Number of *direction changes* for a track	Count	0	3
Acceleration	Acc	Motion	75th percentile of the magnitude of the *acceleration* of the tracked target	$m\,s^{-2}$	−4.2	5.4
Mean acceleration	Acc_{mean}	Motion	Mean of the magnitude of the *acceleration* of the tracked target	$m\,s^{-2}$	−4.2	5.4
Distance traveled	D_{tr}	Motion	Point-to-point distance traveled by the target over the duration of the target track	m	8×10^{-4}	10.1
Change in orientation	O_{std}	Motion	Standard deviation of the *orientation* of each target associated with the track	°	0.6	95.0
Relative speed	dS	Motion/env.	Average of the magnitude of the velocity of the target relative to the *current speed*	$m\,s^{-1}$	−0.9	2.2
Speed	S	Motion	Mean target *speed*	$m\,s^{-1}$	0.007	3.3
Duration	D	Motion	Duration of the target track	s	0.7	29.6
Proximal targets	NR	Image	75th percentile of the number of *proximal targets* of each target associated with a track	Count	0	5
Concurrent targets	N	Image	75th percentile of the number of *concurrent targets* of each target associated with a track	Count	0	13
Range	R	Image	Mean target *range*	m	0.2	9.9
Time	T	Env.	*Time* of d when the target track was detected	h	0	12
Current direction	currDir	Env.	*Current direction* when the target track was detected	°	0	359
Current speed	currSpd	Env.	*Current speed* when the target track was detected	$m\,s^{-1}$	0.07	1.0
Color layout descriptor	CLD	Intensity	75th percentile of the *color layout descriptor* for each target associated with the track	—	0.2	187.1
Weighted centroid offset	dC	Intensity	75th percentile of the *weighted centroid offset*	Pixels	0.05	84.1
Intensity standard deviation	I_{std}	Intensity	Standard deviation of *intensity* of all targets associated with a track	Arbitrary	0.9	68.4
Intensity maximum	I_{max}	Intensity	Maximum *intensity* of any pixel in any target associated with a track	Arbitrary	34	235
75th-percentile intensity	I	Intensity	75th percentile of the *intensity* of all pixels contained in all targets associated with a track	Arbitrary	32	203

- Velocity: Velocity of the target, calculated using the distance between every fifth target associated with the track, using the weighted centroid of each target associated with the track and the elapsed time between detections.
- Acceleration: Acceleration of the target, calculated using the first difference of the target *velocity*.
- Direction change: Zero crossing of *acceleration.*

Environmental covariates are as follows:

- Time: Time of day when target was detected, calculated as the number of hours from midnight in local time (i.e., both 1100 and 0100 LT have a value of 1).
- Current direction: Direction of the tidal current measured by the ADCP.
- Current speed: Magnitude of the horizontal velocity of the tidal current measured by the ADCP.

These covariates are effectively constant for the duration of a track (<60 s).

The presence of other targets in the field of view can also be relevant for classification. These are quantified by the following:

- Proximal targets: Number of other targets detected within a 1 m radius of the *position* of a target of interest.
- Concurrent targets: Number of other targets concurrently detected at any range from the target of interest.

APPENDIX B

Feature Selection Algorithm

The hill-climbing feature selection algorithm was implemented as follows:

1) Initialize an empty feature list.
2) Evaluate performance of the classification model (section 2c) with each individual feature added to the existing feature list. On the first iteration, classification performance is tested for each candidate feature in isolation.
3) Add the candidate feature that produced the highest value of R_{bio} in the test cases to the feature list.
4) Return to step 2, and repeat steps 2 and 3 until all features have been added to the list in a ranked order.
5) Select the subset of the feature list that yields the highest value of R_{bio} (i.e., remove features at the end of the sorted feature list if performance begins to degrade as more features are added).
6) Test removal of each feature from the selected feature list to determine if higher performance can be achieved. If removal of any single feature results in a higher value of R_{bio}, remove that feature from the feature list.
7) Repeat step 5 until no features can be removed from the feature list without reducing R_{bio}.

The hill-climbing algorithm was not implemented for the SVM model with a polynomial kernel function, because this kernel function requires more than one feature to operate.

APPENDIX C

Extended Classification Results

Tables C1–C3 show the full set of classification results for all classification models that were evaluated using the 2017 dataset.

TABLE C1. Classification metrics for each KNN model that was evaluated.

k	Feature method	R_{bio}	R_{bin}	P_{bin}	R_N	R_S	R_{Sm}	R_B	R_F
				Multiclass models					
3	All	0.8	0.97	0.47	0.78	0.83	0.91	0.7	0.83
3	Wrapper	0.86	0.98	0.49	0.78	0.91	0.83	0.87	0.83
3	Filter	0.67	0.96	0.53	0.78	0.57	0.78	0.65	0.74
5	All	0.8	0.96	0.48	0.74	0.83	0.87	0.67	0.83
5	Wrapper	0.83	0.97	0.51	0.61	0.87	0.83	0.83	0.83
5	Filter	0.71	0.97	0.54	0.83	0.65	0.83	0.61	0.74
7	All	0.78	0.95	0.49	0.78	0.83	0.87	0.65	0.83
7	Wrapper	0.86	0.98	0.56	0.74	0.87	0.83	0.87	0.87
7	Filter	0.71	0.97	0.47	0.83	0.65	0.87	0.61	0.74
				Ensemble models					
3	All	0.8	0.95	0.48	0.76	0.85	0.78	0.7	0.83
3	Wrapper	0.89	0.97	0.48	0.83	0.91	0.87	0.78	0.96
3	Filter	0.77	0.96	0.53	0.78	0.74	0.83	0.7	0.83
5	All	0.77	0.96	0.49	0.74	0.78	0.87	0.61	0.78
5	Wrapper	0.88	0.97	0.50	0.83	0.87	0.89	0.78	0.96
5	Filter	0.79	0.96	0.55	0.78	0.78	0.78	0.74	0.83
7	All	0.79	0.97	0.49	0.7	0.83	0.89	0.65	0.83
7	Wrapper	0.89	0.98	0.51	0.74	0.87	0.87	0.83	1
7	Filter	0.76	0.97	0.56	0.83	0.7	0.87	0.7	0.83

TABLE C2. Classification metrics for each SVM model that was evaluated.

Kernel function	Feature method	R_{bio}	R_{bin}	P_{bin}	R_N	R_S	R_{Sm}	R_B	R_F
Linear	All	0.87	0.96	0.59	0.78	0.93	0.87	0.83	0.87
Linear	Wrapper	0.76	0.96	0.57	0.87	0.78	0.83	0.8	0.65
Linear	Filter	0.86	0.96	0.58	0.87	0.91	0.87	0.83	0.83
Polynomial	All	0.78	0.96	0.60	0.74	0.87	0.83	0.7	0.78
Polynomial	Filter	0.79	0.97	0.47	0.78	0.83	0.87	0.65	0.83
RBF	All	0.83	0.97	0.47	0.74	0.87	0.83	0.78	0.91
RBF	Wrapper	0.85	0.96	0.48	0.83	0.91	0.78	0.78	0.91
RBF	Filter	0.84	0.96	0.53	0.83	0.83	0.83	0.78	0.91

TABLE C3. Classification metrics for each RF model that was evaluated.

NumTrees	Feature method	R_{bio}	R_{bin}	P_{bin}	R_N	R_S	R_{Sm}	R_B	R_F
40	All	0.87	0.97	0.59	0.91	0.91	0.83	0.85	0.91
40	Wrapper	0.86	0.96	0.58	0.83	0.91	0.83	0.83	0.87
40	Filter	0.71	0.96	0.61	0.87	0.7	0.78	0.61	0.78
70	All	0.88	0.96	0.60	0.87	0.91	0.87	0.87	0.91
70	Wrapper	0.88	0.97	0.60	0.87	0.91	0.83	0.87	0.91
70	Filter	0.73	0.97	0.62	0.83	0.7	0.74	0.65	0.83
100	All	0.88	0.97	0.60	0.91	0.91	0.87	0.87	0.91
100	Wrapper	0.87	0.97	0.60	0.78	0.91	0.83	0.87	0.91
100	Filter	0.72	0.96	0.63	0.91	0.7	0.78	0.65	0.78
130	All	0.86	0.96	0.61	0.83	0.87	0.83	0.83	0.91
130	Wrapper	0.9	0.98	0.62	0.87	0.87	0.91	0.91	0.91
130	Filter	0.7	0.96	0.62	0.87	0.65	0.7	0.65	0.78
160	All	0.89	0.97	0.61	0.87	0.91	0.87	0.87	0.91
160	Wrapper	0.87	0.96	0.61	0.87	0.87	0.8	0.87	0.96
160	Filter	0.71	0.96	0.62	0.91	0.65	0.74	0.65	0.83
190	All	0.88	0.97	0.60	0.87	0.91	0.87	0.87	0.87
190	Wrapper	0.89	0.97	0.62	0.83	0.93	0.83	0.87	0.91
190	Filter	0.72	0.96	0.62	0.87	0.65	0.83	0.65	0.78

REFERENCES

Baccini, A., M. A. Friedl, C. E. Woodcock, and R. Warbington, 2004: Forest biomass estimation over regional scales using multisource data. *Geophys. Res. Lett.*, **31**, L10501, https://doi.org/10.1029/2004GL019782.

Blackman, S., 1986: *Multiple-Target Tracking with Radar Applications.* Artech House, 449 pp.

Breiman, L., 2001: Random forests. *Mach. Learn.*, **45**, 5–32, https://doi.org/10.1023/A:1010933404324.

Burges, C. J. C., 1998: A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discovery*, **2**, 121–167, https://doi.org/10.1023/A:1009715923555.

Calambokidis, J., B. L. Taylor, S. D. Carter, G. H. Steiger, P. K. Dawson, and L. D. Antrim, 1987: Distribution and haul-out behavior of harbor seals in Glacier Bay, Alaska. *Can. J. Zool.*, **65**, 1391–1396, https://doi.org/10.1139/z87-219.

Copping, A., and Coauthors, 2016: Annex IV 2016 state of the science report: Environmental effects of marine renewable energy development around the world. Pacific Northwest National Laboratory Tech. Rep., 224 pp., https://tethys.pnnl.gov/sites/default/files/publications/Annex-IV-2016-State-of-the-Science-Report_MR.pdf.

Cotter, E., P. Murphy, and B. Polagye, 2017: Benchmarking sensor fusion capabilities of an integrated instrumentation package. *Int. J. Mar. Energy*, **20**, 64–79, https://doi.org/10.1016/j.ijome.2017.09.003.

Dobeck, G. J., J. C. Hyland, and L. Smedley, 1997: Automated detection/classification of sea mines in sonar imagery. *Proc. SPIE*, **3079**, 90–110, https://doi.org/10.1117/12.280846.

Duan, K., and S. S. Keerthi, 2005: Which is the best multiclass SVM method? An empirical study. *Multiple Classifier Systems*, N. C. Oza et al., Eds., Springer, 278–285.

Dudani, S. A., 1976: The distance-weighted k-nearest-neighbor rule. *IEEE Trans. Syst. Man Cybern.*, **SMC-6**, 325–327, https://doi.org/10.1109/TSMC.1976.5408784.

Fawcett, T., 2006: An introduction to ROC analysis. *Pattern Recognit. Lett.*, **27**, 861–874, https://doi.org/10.1016/j.patrec.2005.10.010.

Foote, K. G., D. Chu, T. R. Hammar, K. C. Baldwin, L. A. Mayer, L. C. Hufnagle, and J. M. Jech, 2005: Protocols for calibrating multibeam sonar. *J. Acoust. Soc. Amer.*, **117**, 2013–2027, https://doi.org/10.1121/1.1869073.

Francisco, F., and J. Sundberg, 2019: Detection of visual signatures of marine mammals and fish within marine renewable energy farms using multibeam imaging sonar. *J. Mar. Sci. Eng.*, **7**, 22, https://doi.org/10.3390/JMSE7020022.

Grellier, K., P. M. Thompson, and H. M. Corpe, 1996: The effect of weather conditions on harbour seal (*Phoca vitulina*) haulout behaviour in the Moray Firth, northeast Scotland. *Can. J. Zool.*, **74**, 1806–1811, https://doi.org/10.1139/z96-201.

Hamilton, C. D., C. Lydersen, R. A. Ims, and K. M. Kovacs, 2014: Haul-out behaviour of the world's northernmost population of harbour seals (*Phoca vitulina*) throughout the year. *PLOS ONE*, **9**, e86055, https://doi.org/10.1371/journal.pone.0086055.

Hastie, G. D., D. Gillespie, J. Gordon, J. Macaulay, B. McConnel, and C. Sparling, 2014: Tracking technologies for quantifying marine mammal interactions with tidal turbines: Pitfalls and possibilities. *Marine Renewable Energy Technology and Environmental Interactions*, M. A. Shields and A. I. L. Payne, Eds., Springer, 127–139.

——, and Coauthors, 2019: Automated detection and tracking of marine mammals: A novel sonar tool for monitoring effects of marine industry. *Aquat. Conserv.*, **29**, 119–130, https://doi.org/10.1002/aqc.3103.

Japkowicz, N., 2000: The class imbalance problem: Significance and strategies. *Proc. 2000 Int. Conf. on Artificial Intelligence*, Las Vegas, Nevada, ICAI, 111–117, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.1693&rep=rep1&type=pdf.

Jepp, P., 2017: Target tracking using sonars for marine life monitoring around tidal turbines. *Proc. 12th European Wave and Tidal Energy Conf.*, Cork, Ireland, University of Southampton, https://tethys.pnnl.gov/publications/target-tracking-using-sonars-marine-life-monitoring-around-tidal-turbines.

Kasutani, E., and A. Yamada, 2001: The MPEG-7 color layout descriptor: A compact image feature description for high-speed image/video segment retrieval. *Proc. 2001 Int. Conf. on Image Processing*, Thessaloniki, Greece, IEEE, 674–677, https://doi.org/10.1109/ICIP.2001.959135.

Kohavi, R., and G. H. John, 1997: Wrappers for feature subset selection. *Artif. Intell.*, **97**, 273–324, https://doi.org/10.1016/S0004-3702(97)00043-X.

Lieber, L., T. Nilsen, C. Zambrano, and L. Kregting, 2017: Optimising multiple multibeam sonars to assess marine life interactions with an underwater kite. *Proc. 12th European Wave and Tidal Energy Conf.*, Cork, Ireland, University of Southampton, https://tethys.pnnl.gov/publications/optimising-multiple-multibeam-sonars-assess-marine-life-interactions-underwater-kite.

Melvin, G. D., 2016: Observations of in situ Atlantic bluefin tuna (*Thunnus thynnus*) with 500-khz multibeam sonar. *ICES J. Mar. Sci.*, **73**, 1975–1986, https://doi.org/10.1093/icesjms/fsw077.

——, and N. A. Cochrane, 2015: Multibeam acoustic detection of fish and water column targets at high-flow sites. *Estuaries Coasts*, **38**, 227–240, https://doi.org/10.1007/s12237-014-9828-z.

Platt, J. C., 1999: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, A. J. Smola et al., Eds., MIT Press, 61–74.

Polagye, B., and J. Thomson, 2013: Tidal energy resource characterization: Methodology and field study in Admiralty Inlet, Puget Sound, WA (USA). *Proc. Inst. Mech. Eng.*, **227A**, 352–367, https://doi.org/10.1177/0957650912470081.

Rosa, I. M. D., A. T. Marques, G. Palminha, H. Costa, M. Mascarenhas, C. Fonseca, and J. Bernardino, 2016: Classification success of six machine learning algorithms in radar ornithology. *Int. J. Avian Sci.*, **158**, 28–42, https://doi.org/10.1111/ibi.12333.

Rothery, P., and D. B. Roy, 2001: Application of generalized additive models to butterfly transect count data. *J. Appl. Stat.*, **28**, 897–909, https://doi.org/10.1080/02664760120074979.

Schneider, D. C., and P. M. Payne, 1983: Factors affecting haul-out of harbor seals at a site in southeastern Massachusetts. *J. Mammal.*, **64**, 518–520, https://doi.org/10.2307/1380370.

Song, F., Z. Guo, and D. Mei, 2010: Feature selection using principal component analysis. *Int. Conf. on System Science, Engineering Design and Manufacturing Informatization*, Yichang, China, IEEE, 27–30, https://doi.org/10.1109/ICSEM.2010.14.

Thyng, K. M., C. A. Greene, R. D. Hetland, H. M. Zimmerle, and S. F. DiMarco, 2016: True colors of oceanography: Guidelines for effective and accurate colormap selection. *Oceanography*, **29** (3), 9–13, https://doi.org/10.5670/oceanog.2016.66.

Trenkel, V. M., V. Mazauric, and L. Berger, 2008: The new fisheries multibeam echosounder ME70: Description and expected contribution to fisheries research. *ICES J. Mar. Sci.*, **65**, 645–655, https://doi.org/10.1093/icesjms/fsn051.

Urban, P., K. Köser, and J. Greinert, 2017: Processing of multibeam water column image data for automated bubble/seep detection and repeated mapping. *Limnol. Oceanogr. Methods*, **15**, 1–21, https://doi.org/10.1002/lom3.10138.

Viehman, H. A., and G. B. Zydlewski, 2014: Fish interactions with a commercial-scale tidal energy device in the natural environment. *Estuaries Coasts*, **38**, 241–252, https://doi.org/10.1007/S12237-014-9767-8.

Wallace, B. C., K. Small, C. E. Brodley, and T. A. Trikalinos, 2011: Class imbalance, redux. *2011 IEEE 11th Int. Conf. on Data Mining*, Vancouver, BC, Canada, IEEE, 754–763, https://doi.org/10.1109/ICDM.2011.33.

Wilding, T. A., and Coauthors, 2017: Turning off the DRIP ('data-rich, information-poor')—Rationalising monitoring with a focus on marine renewable energy developments and the benthos. *Renewable Sustainable Energy Rev.*, **74**, 848–859, https://doi.org/10.1016/j.rser.2017.03.013.

Williamson, B., S. Fraser, P. Blondel, P. S. Bell, J. J. Waggit, and B. E. Scott, 2017: Multisensor acoustic tracking of fish and seabird behavior around tidal turbine structures in Scotland. *IEEE J. Ocean Eng.*, **42**, 948–965, https://doi.org/10.1109/joe.2016.2637179.

Zhang, M., and Z. Zhou, 2005: A k-nearest neighbor based algorithm for multi-label classification. *Int. Conf. on Granular Computing*, Beijing, China, IEEE, 718–721, https://doi.org/10.1109/GRC.2005.1547385.