

Statistical learning mitigation of false positives from template-detected data in automated acoustic wildlife monitoring

Cathleen M. Balantic & Therese M. Donovan

To cite this article: Cathleen M. Balantic & Therese M. Donovan (2019): Statistical learning mitigation of false positives from template-detected data in automated acoustic wildlife monitoring, *Bioacoustics*, DOI: [10.1080/09524622.2019.1605309](https://doi.org/10.1080/09524622.2019.1605309)

To link to this article: <https://doi.org/10.1080/09524622.2019.1605309>



This work was authored as part of the Contributor's official duties as an Employee of the United States Government and is therefore a work of the United States Government. In accordance with 17 U.S.C. 105, no copyright protection is available for such works under U.S. Law.



Published online: 06 May 2019.



Submit your article to this journal [↗](#)



Article views: 285



View related articles [↗](#)



View Crossmark data [↗](#)

Statistical learning mitigation of false positives from template-detected data in automated acoustic wildlife monitoring

Cathleen M. Balantic ^a and Therese M. Donovan^b

^aVermont Cooperative Fish and Wildlife Research Unit, University of Vermont, Burlington, VT, USA;

^bU.S. Geological Survey, Vermont Cooperative Fish and Wildlife Research Unit, Rubenstein School of Environment and Natural Resources, University of Vermont, Burlington, VT, USA

ABSTRACT

Audio sampling of the environment can provide long-term, landscape-scale presence-absence data to model populations of sound-producing wildlife. Automated detection systems allow researchers to avoid manually searching through large volumes of recordings, but often produce unacceptable false positive rates. We developed methods that allow researchers to improve template-based automated detection using a suite of statistical learning algorithms when false positive rates are problematic. To test our method, we acquired 668 hours of recordings in the Sonoran Desert, California USA between March 2016 and May 2017, and created spectrogram cross-correlation templates for three target avian species. We trained and tested five classification algorithms and four performance-weighted ensemble classifier methods on target signals and false alarms from March 2016, and then selected high-performing ensemble classifiers from the train/test phase to predict the class of new detections thereafter. For three target species, our ensemble classifiers were able to identify 98%, 81%, and 100% of false alarms compared with the baseline template detection system, and comparative positive predictive values improved from 6% to 69%, 87% to 95%, and 2% to 77%. We show that statistical learning approaches can be implemented to mitigate false detections acquired via template-based automated detection in automated acoustic wildlife monitoring.

ARTICLE HISTORY

Received 1 October 2018

Accepted 2 April 2019

KEYWORDS

Automated acoustic monitoring; bioacoustics; false positives; machine learning; species identification; statistical learning

Introduction

Tracking wildlife population dynamics at regional scales requires methods that efficiently accumulate data on species of interest (Pollock et al. 2002). Automated acoustic monitoring of sound-producing wildlife offers one path for characterizing baseline species status and trends across vast landscapes, important within the context of climate change and rapidly shifting land uses. Because obtaining species abundance data is often inefficient, costly, and impractical, research at large spatial scales may instead collect species presence-absence data for use in occupancy models; remote acoustic monitoring is well-positioned to support such data collection because it affords the

CONTACT Cathleen M. Balantic  cathleen.balantic@uvm.edu

This work was authored as part of the Contributor's official duties as an Employee of the United States Government and is therefore a work of the United States Government. In accordance with 17 U.S.C. 105, no copyright protection is available for such works under U.S. Law.

This is an Open Access article that has been identified as being free of known restrictions under copyright law, including all related and neighboring rights (<https://creativecommons.org/publicdomain/mark/1.0/>). You can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission.

opportunity to identify presence or absence of species based on sounds captured on audio recordings (Furnas and Callas 2014; Cerqueira and Aide 2016).

Recent efforts have yielded tremendous growth in large-scale, long-term bioacoustic monitoring programs that accumulate vast amounts of acoustic data well beyond human capacity for efficient examination (Shonfield and Bayne 2017). Such large-scale data acquisition is accompanied by methodologies and software that enable semi-automated detection of sound-producing wildlife species from audio recordings. No approach for automated detection is perfect, and detection methods can vary based on research goals, soundscape characteristics, and acoustic features of a target species sound (Towsey et al. 2012; Stowell et al. 2016). Hidden Markov models (Agranat 2009; Aide et al. 2013; Potamitis et al. 2014; Wildlife Acoustics 2016; Ranjard et al. 2017), spectrogram cross correlation (Mellinger and Clark 2000; Avisoft Bioacoustics 2016; Hafner and Katz 2018), binary point matching (Towsey et al. 2012; Hafner and Katz 2018), band-limited energy detection (Figueroa 2012; Bioacoustics Research Program 2015) and convolutional neural networks (Knight et al. 2017) are common approaches for automatically detecting species by their sounds. Probabilistic classification methods also show promise (Ovaskainen et al. 2018).

Advantages of a template-based automated detection system are the low barrier to entry and ease of use. Template-based detection strategies such as spectrogram cross-correlation and binary point matching are well established in the literature and have a freely available software platform (Hafner and Katz 2018), compared with proprietary commercial software that costs \$399 USD for a single user yearlong subscription (Wildlife Acoustics 2016). Template matching systems require no signal processing expertise; a researcher needs only to be familiar with sounds produced by their focal species. Moreover, template-based acoustic monitoring can be initiated with a single example of a sound issued by a target species – helpful when not many reference calls are available. By contrast, state-of-the-art approaches like convolutional neural networks (CNNs) can demand large amounts of labelled training data beyond the capacity of an individual acoustic monitoring project (Gibb et al. 2018), in addition to the expertise required for hyperparameter tuning and construction of the network architecture. Though there are CNN pipelines available for public use (e.g. for detection of echolocating bats as in Mac Aodha et al. 2018), the programming expertise and set-up required to build a customized CNN for a specific monitoring need may be prohibitive. Furthermore, in a side-by-side evaluation, template matching approaches have compared favourably with CNNs (Knight et al. 2017).

Regardless of the automated detection approach, a detected audio signal is either a true positive detection, which is a sound produced by the target species, or a false positive detection, which is not a signal from a target species. Throughout this paper, we will refer to true positive detections as ‘target signals’ and false positive detections as ‘false alarms.’ Regardless of the automated detection method employed, when acting without human assistance, computer-automated methods often produce an unacceptable number of false alarms, wherein non-target noise is detected and incorrectly assigned to a target species (Acevedo et al. 2009). False alarm rates from computer-automated methods may vary widely from project to project based on the prevalence of similar sounds from non-target sources in the soundscape, acoustic characteristics of sounds made by the target species (Towsey et al. 2012), the type of automated detection

routine used (Corrada-Bravo et al. 2017), the available number of target sound examples upon which automated methods may be trained (Stowell et al. 2016), the quality of training data in terms of how well it represents the data that will be subject to the automated detection method (Knight and Bayne 2018), and selection of score thresholds above which detections may occur (Knight et al. 2017).

We illustrate the process of acquiring both true target signals and false alarms using a spectrogram cross-correlation template as a screening mechanism to accumulate detections for a North American desert songbird, the Verdin (*Auriparus flaviceps*). First, we render a spectrogram of an audio recording (Figure 1(a)), in which a Verdin vocalized three times, each with a characteristic three note whistle at about 4 kHz on the y-axis. We set time and frequency limits that define a cross correlation-based detection template for the song occurring at ~24 seconds within the example recording (Figure 1(b)). This template thus provides an acoustic pattern issued by a known target species, and can be used to scan many recordings in pursuit of Verdin vocalizations. The template is compared to an audio recording in a moving window analysis, in which a correlation between the template and audio file is obtained for each window (Figure 2). We then select a correlation detection threshold for the template, which is a user-specified detection threshold ranging from 0 (no correlation) to 1 (full correlation). Only peaks with scores above the chosen threshold are considered detections. This process facilitates rapid screening of acoustic data to acquire a set of detections, which are either true target

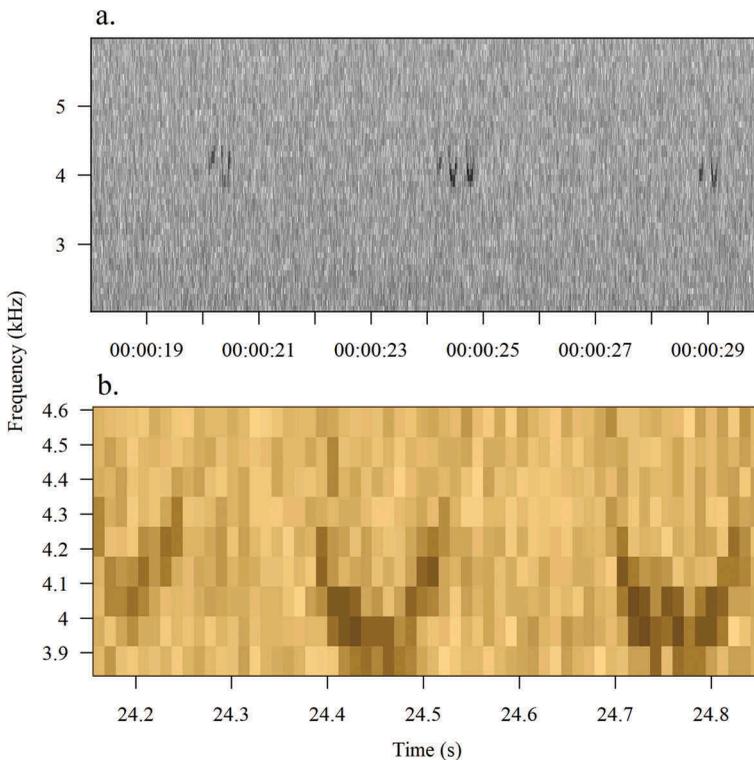


Figure 1. (a) Verdin songbird vocalization within a recording. (b) example template created from Verdin vocalization occurring at ~24 seconds.

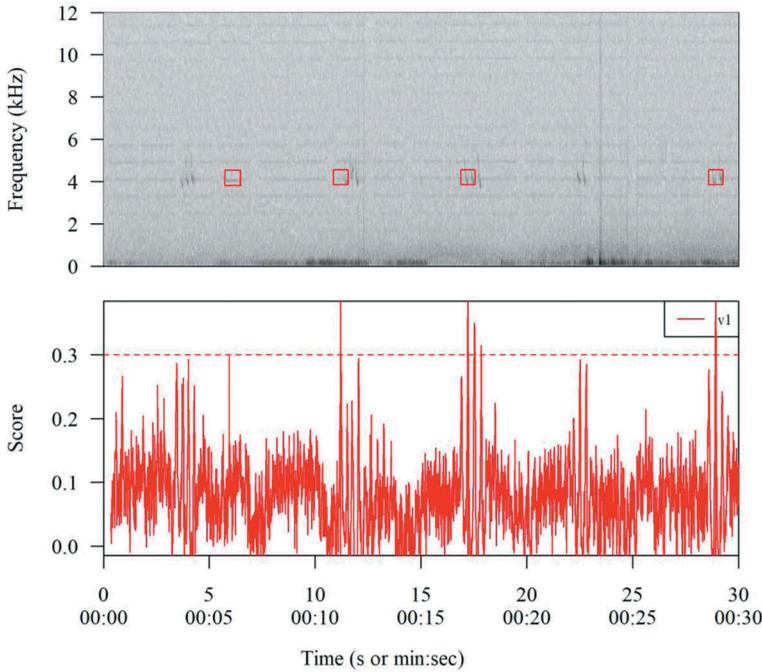


Figure 2. Illustration of event detection via template matching paired with a score threshold. Red boxes in the top panel denote detections, while the red line in the bottom panel indicates a selected threshold (0.3). The first red box is a false alarm produced as a result of electromagnetic interference. The last three red boxes are all target signals wherein the Verdin is actually vocalizing. Note also two occurrence-level false negatives (~4 seconds and 23 seconds), in which the species is vocalizing but no detection occurred.

Table 1. Confusion Matrix Examples to distinguish between true and false positives at the vocalization occurrence level vs. the detection level.

	Actual Class: Vocalization	Actual Class: No Vocalization	
Predicted Class: Vocalization	TP = 3	FP = 1	4
Predicted Class: No Vocalization	FN = 2	TN = 38	40
	5	39	44

Following from [Figure 2](#), a confusion matrix at the ‘vocalization occurrence level’ summarizes all vocalizations issued by the target species and captured within the recording. Three vocalizations are correctly detected and referred to as ‘true positives’ (TP = 3), while one non-vocalization is flagged as signal from the target species, which is a ‘false positive’ (FP = 1). Meanwhile, two vocalizations are missed by the system and are ‘false negatives’ (FN = 2). Lastly, approximately 38 time bands within the recording are appropriately ignored since they contain no vocalizations from the target and are ‘true negatives’ (TN = 38). We highlight the top row to indicate that only the true and false positive detections from [Table 1](#) are considered within this paper.

signals or false alarms ([Table 1](#)). Distinguishing target signals from false alarms is the focus of this paper. Tangentially, this process also results in false negatives at the level of actual vocalization occurrence, in which a species produced a sound but was not detected

by the template matching process (Table 1; see Brauer et al. 2016 and Katz et al. 2016 for assessment of false negatives using a template-matching system).

Although humans may distinguish between true target signals and false alarms by visually examining the spectrogram or listening to the audio file, this approach is inefficient against the sheer volume of data collected in an acoustic monitoring program. Alternatively, after the template screening step has been performed, users may manually verify a small subset of detections as target signals and false alarms and use these to train a variety of classification algorithms that can predict whether template detections are true or false positives. Such an approach describes a form of statistical learning called supervised learning, in which a human labels a subset of data for the algorithm so that it can map existing data to known output classes (Bishop 2006). Once an algorithm has been trained on known data, it can be tested to predict the class – target signal or false alarm – of unknown data.

Two key components must be addressed to undertake supervised statistical learning. First, one must decide which acoustic features (predictive variables) of a detection can be used by the algorithm to predict the outcome (target signal or false alarm). The best predictive features may vary based on sounds produced by any given target species, as well as soundscape circumstances such as wind, rain, anthropogenic noise, or non-target species vocalizing within the same frequency range. Figure 1(b) shows an example of a set of acoustic features: the coloured shading in each pixel of the spectrogram represents the amplitude (or sound intensity) at that point, and each of the amplitude values in this spectrogram serves as an acoustic feature. Other potential predictive features include binned zero-crossing rates, time and frequency contours of the amplitude probability mass function, and summary statistics of the frequency spectrum (Sueur et al. 2008) (Appendix 1). All features for a detected event are defined based on the extent of the template that produced the event.

Once predictive features have been obtained, they are fed into a classification algorithm that will map predictive feature inputs to known output labels (target signal or false alarm). For example, the k -Nearest Neighbours classifier predicts the class of a new observation based on its feature similarity to some ' k ' number of observations within the training set (Cover and Hart 1967). Support Vector Machines seek an equation that optimally separates classes based on a high number of feature dimensions in geometric space (Boser et al. 1992), while Random Forests average a number of feature-based decision trees in order to make predictions (Breiman 2001). Regularized Logistic Regression uses penalized maximum likelihood to shrink the values of predictive feature coefficients, reducing variance so that the resulting model is better equipped to predict outside the range of data upon which it was trained (Zou and Hastie 2005). To improve classification, multiple algorithms may be combined into an 'ensemble' method to predict the class of a new detection.

Classification methods for discriminating between target signals and false alarms thus provide an opportunity in large-scale automated acoustic wildlife monitoring. Climate and land use change are forces that shift the occurrence of species across vast spatial scales, and monitoring these shifts at large scales is paramount for natural resource practitioners tasked with maintaining and sustaining species, populations, and ecosystems. Acoustic monitoring can produce vast amounts of data for this purpose, but existing automated detection algorithms often deliver high rates of false positives

(Acevedo et al. 2009; Buxton and Jones 2012; Marques et al. 2013; Duan et al. 2013; Shonfield and Bayne 2017). Without accessible, straightforward, and generalizable methodologies for the mitigation of false positives in long-term data sets, occurrence-based bioacoustics research will continue to suffer the complications imposed by prohibitive numbers of detection errors, which often pre-empt poor model inference, ill-informed management decisions, and undesirable conservation outcomes (Royle and Link 2006; Miller et al. 2011; Ruiz-Gutierrez et al. 2016).

Objectives

The aim of this study was to explore methods for the semi-automatic removal of false positives to increase the quality of monitoring data. These approaches have been implemented in the R package *AMMonitor* (Balantic et al. unpublished results). Our objectives were to 1) use spectrogram cross-correlation templates as an initial screening step to accumulate detections for focal species in a pilot acoustic monitoring program, 2) train and test statistical learning classification algorithms to distinguish between target signals and false alarms acquired during the template screening phase, 3) use a trained and tested classifier on new detections and compute overall classification performance in comparison to the template screening system.

Materials and methods

Objective 1: use templates as a screening step to acquire focal species detections from field data

Acquire acoustic recordings

We piloted an acoustic monitoring program in the Colorado-Sonoran Desert on public land managed under the auspices of the U.S. Bureau of Land Management (BLM). Autonomous recording units were installed at 16 sites within the BLM-managed Riverside East Solar Energy Zone, a 599 square-kilometre patch allocated as a utility-scale solar renewable energy hub. Because this work is a proof of concept with a focus on methodology rather than on ecological inference, study sites were selected non randomly near microphyll woodland habitat to record songbirds, and historic breeding pond locations with the intention of recording Couch's Spadefoot Toad (*Scaphiopus couchii*). Acoustic monitoring units were located at least 800 metres away from one another to maximize independence of acoustic events.

Each audio recording unit was a modified Android cellular phone (2015 2nd Generation Motorola Moto E model XT1527 with 5.0.2 Lollipop Android Operating System) contained within a weatherproof case and attached to an external 10-watt solar panel for power. Each unit was outfitted with an external omnidirectional electret condenser microphone (JLI-61A, JLI Electronics). All units were secured to U-posts elevated 1.83 metres aboveground. Units recorded in WAV format at a sampling rate of 44.1 kHz. The data collection period ran from March 2016 to May 2017. Units located in microphyll woodland habitats recorded every day for one minute at 6:00, 6:30, 7:00, 7:30, 8:00, 16:00, 16:30, 17:00, and 17:30 PST. Two units located next to historic toad breeding ponds also

recorded for one minute each day at 5:30, 6:00, 6:30, 7:00, 21:00, 21:30, 22:00, 22:30, and 23:00 PST ($n = 9$ surveys per phone per day). We used the CinixSoft Remote Schedule Voice Recorder App (CinixSoft 2014) and Easy Voice Recorder Pro (Digipom 2016) to schedule recordings and remotely send them to our server using the cellular network. All units were in airplane mode while recording to prevent electromagnetic interference that occurs while in cellular data mode.

Create templates for target species

As monitoring targets for this environment, we chose three avian species common to the region: Eurasian Collared-Dove (*Streptopelia decaocto*), Gambel's Quail (*Callipepla gambelii*), and Verdin (*Auriparus flaviceps*). The canonical call from Eurasian Collared-Dove is a three note 'advertising coo' used for mate attraction and territory defense (Romagosa 2012), with calls occurring at frequencies around 0.5 kHz (Figure 3(a)). The Gambel's Quail *kaa* or *cow* call emitted by males, whose principal function is to announce mating availability, is a single upside-down u-shaped note typically occurring

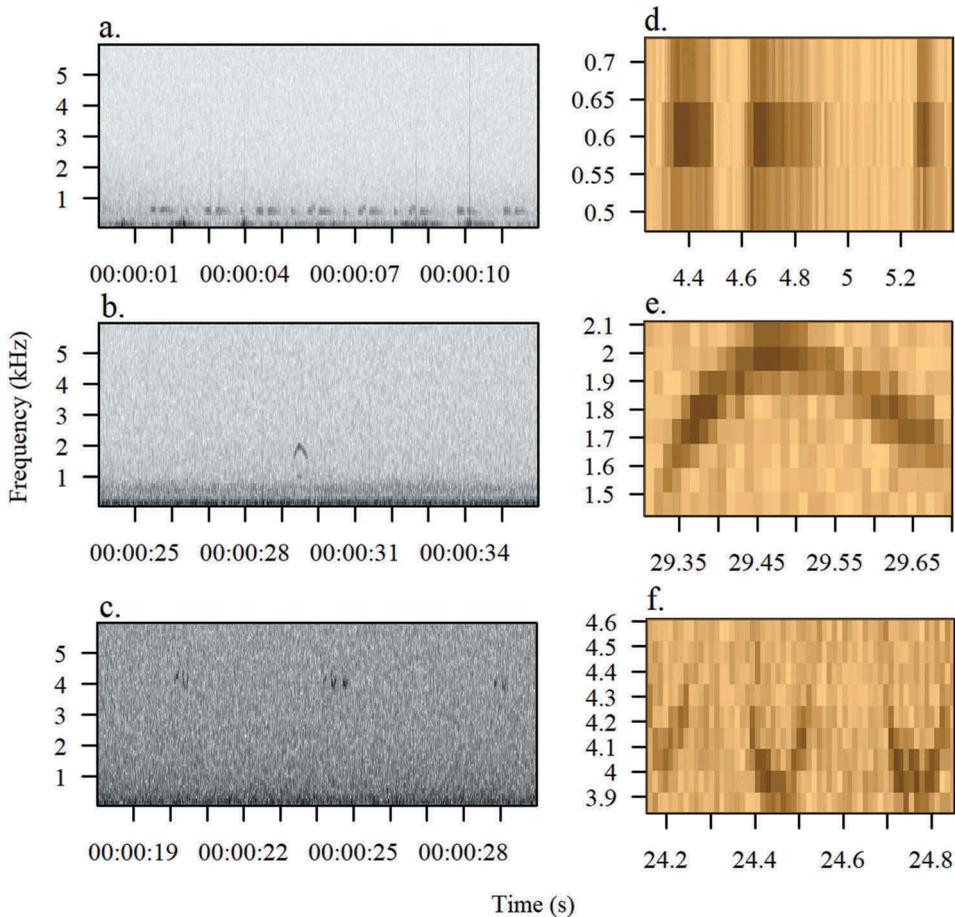


Figure 3. Vocalization examples (a-c) and templates (d-f) for Eurasian Collared-Dove, Gambel's Quail, and Verdin, respectively.

between 1–3 kHz (Gee et al. 2013) (Figure 3(b)). The male Verdin’s ‘whistle song’ is a two to four note whistle occurring around 4–6 kHz; little documentation exists with regard to individual or geographic variation (Webster 1999) (Figure 3(c)).

We created one template for each species from song events chosen out of the recordings acquired in Objective 1 (Figure 3(d-f)), using the *monitoR* R package function *makeCorTemplate()* with a window length of 512, zero overlap, and the Hanning window function. As suggested by Katz et al. (2016), we developed the templates and their accompanying score thresholds iteratively, testing them on recording data outside the recording from which the template was constructed before settling on finalized versions. We conducted this process manually. Recommendations for optimal template creation are sparse, and there is minimal consistency across detection methodologies for selection of the template detection threshold (Shonfield and Bayne 2017). The difficulty in creating and assessing templates, as well as the lack of best practices available for template creation, in fact provided substantial motivation for developing the methods in this paper.

Accumulate detections and obtain associated predictive features

Using the templates and accompanying score thresholds, we ran the *AMMonitor* function *scoresDetect()* to accumulate detections for all recordings. The *scoresDetect()* function employs Pearson’s correlation coefficient to score amplitude values of a moving frame against those in the template, and then isolates local maxima in the score vector to identify detection events (Katz et al. 2016). As in Figure 2, peaks with scores exceeding the score threshold were considered detections, which were either true target signals or false alarms.

Concomitant with the accumulation of detections, we used the *scoresDetect()* function to extract the raw amplitude matrix values associated with each detection, the correlation score, and a number of acoustic summary features acquired via the R package *seewave* (Sueur et al. 2008). These features included binned zero-crossing rates, time and frequency contours of the empirical amplitude probability mass function for each time and frequency bin, quantiles calculated from the empirical cumulative distribution functions of the empirical time and frequency probability mass functions, and statistical properties of the frequency spectrum such as the spectral mean, standard deviation, standard error, median frequency, dominant (mode) frequency, frequency quartiles, centroid, skewness, kurtosis, flatness, and entropy (Appendix 1).

Objective 2: train and test classification algorithms to distinguish between target signals and false alarms for each species

Manually label a subset of detections; split into training and testing data

Once detections had been acquired via the template screening step in Objective 1, we manually verified all detections within the first month of field sampling (March 2016) (Figure 4(a,b)). We used the *AMMonitor* function *scoresVerify()* to assist with manual labelling of all detections as true and false positives for each target species. Each detection was labelled by the lead author primarily by visual identification on the spectrogram. We made additional effort to listen to visually ambiguous detections to confirm their class labels. For Eurasian Collared-Dove (hereafter ECDO), we labelled

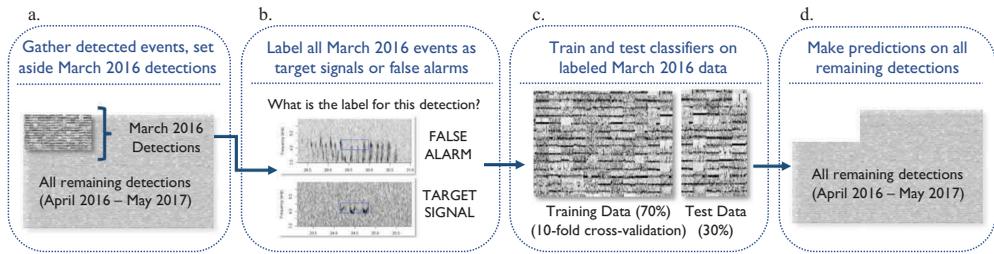


Figure 4. We illustrate the classifier training, testing, and application process using events detected by the Verdin template. (a) After creating the templates in Figure 3 and running the template matching algorithm as shown in Figure 2, we collected all detected events from March 2016 – May 2017 for a single template. Here, all detected events are shown side by side. For model training and testing, we focused on all detected events from March 2016. (b) We used the *AMMonitor scoresVerify()* function to label all March 2016 detections as target signals or false alarms. (c) We split the labeled March 2016 detections into training (70%) and testing (30%) data, using the *caret* function *createDataPartition()* to preserve any existing class imbalances of target signals and false alarms. We performed 10-fold cross-validation on the training data. We evaluated classifier performance on the test data, which was withheld from the classifier algorithms during cross-validation. (d) After the classifier models were trained, tested, and evaluated, we applied a performance-weighted average ‘ensemble’ classifier to make predictions on all remaining detections (April 2016 – May 2017).

detections as target signals if at least two notes were contained within the detection frame. For Gambel’s Quail (hereafter GAQU), we labelled detections as target signals if they contained one frequency-modulated call signal. For Verdin (hereafter VERD), we labelled detections as target signals if at least one frequency-modulated whistle note was contained within the detection frame. Any other detections were labelled as false alarms. After verification, we split the labelled datasets into training (70%) and testing (30%) data, which is a common heuristic data split in statistical learning problems (Weinberger et al. 2006), and used the probability-based *createDataPartition()* function in the R package *caret* (Kuhn 2016) to obtain a balanced split of features, target signals, and false alarms (Figure 4(c)).

Train and test statistical learning classifiers

To construct models for each species and train them on the training data sets, we invoked the *AMMonitor* function *classifierModels()*, which utilizes functions from the machine learning R package *caret* (Kuhn 2016). We trained our classifiers on raw data with no preprocessing (i.e. no scaling or transformation of the acoustic feature data). We used the method ‘knn’ for kernelized k -nearest neighbours which tunes to select an optimal k , ‘svmLinear’ for linear support vector machines, which entails the optimization of a cost parameter, ‘svmRadial’ for radial support vector machines, which optimizes both a cost parameter and the σ value of the radial basis function kernel, ‘rf’ for random forests, which involves tuning a parameter for the number of randomly selected predictor variables, and ‘glmnet’ for regularized logistic regression, which requires tuning a regularization parameter (λ) and a mixing parameter (α).

Because a prohibitive entry point to the use of statistical learning methods involves meticulously tuning algorithms to produce acceptable models, and because our aim was to generate extensible methodology accessible to researchers with little or no statistical learning experience, we used the default *caret* package tuning grids for all five models. Though default values should always be used with care, we opted for the defaults here to keep the methodology as simple as possible (in practice, users are free to use customized values rather than defaults). We used all acoustic features listed in [Appendix 1](#) to train each classifier; we did not select or exclude any features *a priori*, and did not incorporate any higher order terms. After the classifiers were trained and tested, we used the *caret* function *varImp()* to investigate which acoustic features had been most important for distinguishing between target signals and false alarms, but did not use this information to select or exclude features manually. Lastly, within the *AMMonitor* function *classifierModels()*, we used the *caret* function *trainControl()* to apply 10-fold cross-validation during the training phase to reduce model overfitting ([Figure 4\(c\)](#)).

After the training phase, we tested the trained classifiers on the 30% of unseen data (also using the *AMMonitor* function *classifierModels()*) ([Figure 4\(c\)](#)). Although we used cross-validation during the training phase to reduce overfitting, this technique can produce overly optimistic estimates of the training error (Hastie et al. 2009). Therefore, to get a more unbiased understanding of the system’s overall performance, we applied the trained model to completely unseen data during the testing phase to avoid optimistic performance estimates. For every detection, each of the five classifiers yielded a probability that the detection was of the target signal class. While the logistic regression and random forest models output actual probabilities, the support vector machines and k-nearest neighbours fit a sigmoid function on their outputs to return probability-like values between 0 and 1 (Kuhn 2016). For simplicity in evaluation, detections with values of 0.5 or above were classified as target signals; those below were classified as false alarms.

Assess classifier performance on the test set

Since labels for the test data were already known (target signal or false alarm), the training and testing procedure resulted in a confusion matrix summarizing the true classes of each detection and the classes to which they were assigned by each classifier (e.g. [Table 2](#)). We used the *AMMonitor* function *classifierPerformance()* to calculate several measures of classifier performance ([Table 2](#)). The literature contains rich debate over measures of classifier performance (Powers 2007), but the most useful evaluation measures depend on the research motivation behind using classification, as well as on the total number of observations and the balance of classes, which is why we sought a range of evaluation measurements.

For our study, we gave special merit to four performance metrics, all of which range in value from 0 to 1, with scores closest to 1 being most desirable (highlighted in [Table 2](#)). First, sensitivity (a.k.a. recall or true positive rate) is of particular interest because it denotes the proportion of target signals correctly identified by the classifier $[TP/(TP + FN)]$ ([Table 2](#)). Second, specificity (or true negative rate) denotes the proportion of false alarms correctly identified as such, making them true negatives within the confusion matrix $[TN/(TN + FP)]$. Third, positive predictive

Table 2. Confusion Matrix for Detections Only.

	Actual Class Label: Target Signal	Actual Class Label: False Alarm	Row Sum
Predicted Class: Target Signal	TP = 3 [3]	FP = 1 [0]	4 [3] Pos. Pred. Value = $3/4 = 0.75$ [Pos. Pred. Value 3/3 = 1]
Predicted Class: False Alarm	FN = 0 [0]	TN = 0 [1]	0 [1]
Column Sum	3 [3] Sensitivity = $3/3 = 1$ [Sensitivity = 3/3 = 1]	1 [1] Specificity = $0/1 = 0$ [Specificity = 1/1 = 1]	4 $F1 = 2*(0.75*1)/(0.75 + 1) = 0.86$ [F1 = 2*(1*1)/(1 + 1) = 1]

The four detections highlighted in the top row of Table 1 are now subject to ‘detection level’ classification, in which an algorithm is used to reclassify events in an effort to minimize false alarms. The goal of reclassification is to maximize sensitivity, specificity, positive predictive value and F1 score. Ideal conditions are bolded and given in brackets next to the actual condition.

value (a.k.a. precision) expresses the proportion of *predicted* positive detections that are *actually* target signals $[TP/(TP + FP)]$. Finally, the F1 score represents a weighted average of positive predictive value and sensitivity, quantifying the trade-off between a desire for high positive predictive value and high sensitivity. The F1 score is calculated as $2 * \text{Positive Predictive Value} * \text{Sensitivity}/(\text{Positive Predictive Value} + \text{Sensitivity})$. Maximizing all four of these metric scores was a primary goal in classifier evaluation.

We also constructed Receiver-Operating Characteristic (ROC) curves, which plot the true positive rate (sensitivity) against the false positive rate ($1 - \text{specificity}$). Many classification problems involve imbalanced datasets, in which the number of false positive cases greatly outweighs the number of true positive cases or vice versa. Class imbalances undermine performance metrics like accuracy and area under the ROC curve (AUC): a classifier may predict the majority class for most or all observations in the test set and still attain a high accuracy score, which is why measures beyond accuracy are necessary (Zhu and Davidson 2007). To account for this, we also constructed Precision-Recall Curves, which plot positive predictive value (a.k.a. precision) against sensitivity (a.k.a. recall) (Davis and Goadrich 2006). For both ROC and Precision-Recall curves, we defined AUC values matching or exceeding 0.80 as acceptable, and values matching or exceeding 0.90 as high performance, with values of 1 indicating perfect performance.

Create and assess performance-weighted class probability ensemble methods

After performance metrics were computed for each of the five classifiers individually, we used the *classifierEnsemble()* function to aggregate the results of the five classifiers using a weighted average with respect to a selected performance metric. In statistical learning, methods combining predictions across multiple classifiers are known as ‘ensembles,’ in which classification occurs as a consequence of aggregation or integration of multiple distinct algorithms to improve predictive performance. In our simple implementation of an ensemble, the *classifierEnsemble()* function established four performance-weighted ensemble methods, weighting each classifier’s probability that a detection was of class ‘target signal’ by the classifier’s test phase sensitivity, specificity, positive predictive value, or F1 score (Appendix 2). Thus, contributions of lower-

scoring classifiers are diminished, while higher-scoring classifiers have stronger impact on ensemble class predictions for any given detection. We then computed the sensitivity, specificity, positive predictive value, and F1 of the ensemble results.

Objective 3: assess the performance of a trained and tested classifier on new detections

In releasing our trained and tested classifiers ‘into the wild’ on new, incoming template detections, our goal was to use classification to eliminate as many false alarms as possible, while still retaining true target signals needed for meaningful estimations of species occurrence. For this reason, we chose to proceed using the ensemble method weighted by the F1 score as our predictive classifier on new data for all three species.

Using *AMMonitor’s classifierPredict()* function, we invoked the ensemble method weighted by the F1 score to predict the class of all detections across the entire recording dataset that were *not* seen during the training and testing phase. Thus, the training and testing phase occurred on all data from March 2016, and the prediction phase occurred on all data spanning the 14-month period from April 2016 to May 2017 (Figure 4(d)). We then manually verified all detections in the prediction set, and computed metrics to evaluate whether our classification method improved upon the initial template screening step. We calculated the positive predictive value and F1 score for the template screening step, and calculated sensitivity, specificity, positive predictive value, and F1 score for the classifiers to compare performances of the two systems. We assumed that the template screening method had a sensitivity of 1 and specificity of 0 with regard to distinguishing target signals from false alarms.

Results

Objective 1: use templates as a screening step to acquire focal species detections from field data

We collected a total of 40,094 one-minute recordings from March 2016 to May 2017 across 16 smartphone-based audio recorders. An unknown number of recordings contained electromagnetic interference for reasons unknown, all of which were retained in the dataset. We created spectrogram cross-correlation templates for ECDO, GAQU, and VERD (Figure 3(d-f)), and identified score thresholds of 0.43, 0.33, and 0.23, respectively, to use during the screening phase. At these score thresholds, we collected a total of 4,427 detections for ECDO, 1,464 detections for GAQU, and 4,241 detections for VERD, resulting in a total of 10,132 detections.

Objective 2: train and test classification algorithms to distinguish between target signals and false alarms for each species

Manually label a subset of detections; split into training and testing data

There were 631 detections acquired from 54.3 hours of recordings from March 2016 at the selected score thresholds: 323 ECDO, 62 GAQU, and 246 VERD (Table 3). It took approximately one hour to manually verify all March 2016 detections using our chosen

Table 3. Number of manually verified detections from March 2016 used as classifier training and testing data for three focal species: Eurasian Collared-Dove (ECDO), Gambel's Quail (GAQU), and Verdin (VERD).

Template	Total N	Total True		Total False	
ECDO	323	135		188	
		Training 95	Testing 40	Training 132	Testing 56
GAQU	62	34		28	
		Training 24	Testing 10	Training 20	Testing 8
VERD	246	49		197	
		Training 35	Testing 14	Training 138	Testing 59

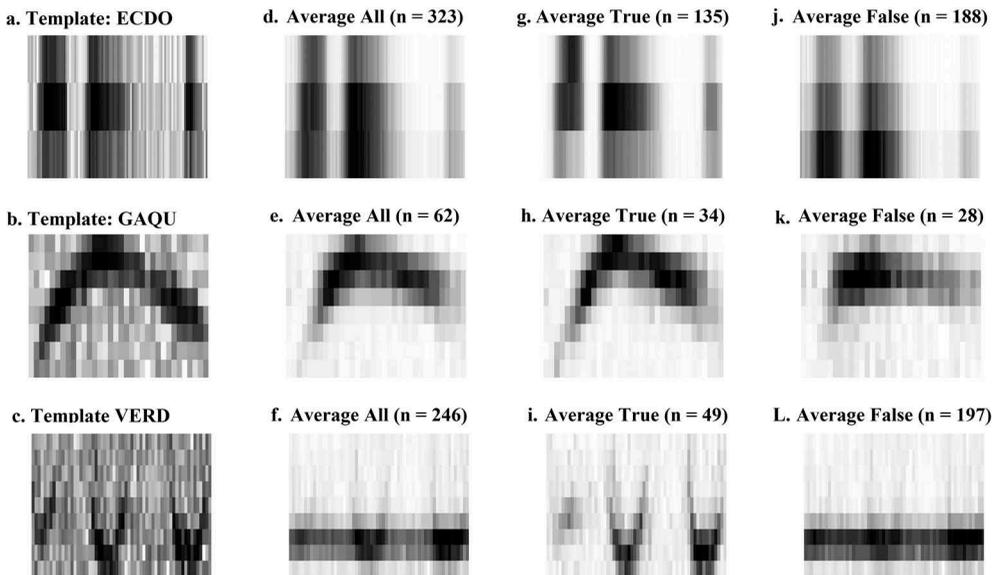


Figure 5. Visual summary of all manually verified detections used as training and testing data. Templates used to collect detections (a-c) are juxtaposed against average spectrograms for all verifications (d-f), all target signal verifications (g-i), and all false alarm verifications (j-l).

verification standards. The ECDO and GAQU datasets were adequately balanced, with 135 true and 188 false for ECDO, and 34 true and 28 false for GAQU. The VERD dataset had a class imbalance with 49 true and 197 false (Table 3). A visual summary of verifications is contained in Figure 5, wherein spectrograms for verified detections were averaged across the amplitude values to show the mean target signal and mean false alarm.

Train and test statistical learning classifiers

Despite the low total number of GAQU detections and considerable class imbalance for VERD, all classification models converged during the training phase and were

Table 4. Assessment of classifier performance on the training data after 10-fold cross-validation. The classifier performance metrics in this table can take on values between 0 (worst) and 1 (best), with NAs occurring where metrics are not possible to calculate. Rows indicate classifiers, and columns indicate performance metrics. Recall that these performance metrics only evaluate the subset of template-detected data, in support of the binary classification problem of distinguishing target signals from false alarms (described in Tables 1 and 2).

Classifier	Sensitivity	Specificity	Pos. Pred. Value	F1
a. ECDO models				
Regularized Logistic Regression	0.99	0.98	0.98	0.99
Linear Support Vector Machine	0.97	0.97	0.96	0.97
Radial Support Vector Machine	0.96	0.96	0.95	0.96
Random Forests	0.98	0.97	0.96	0.97
Kernelized k-Nearest Neighbours	0.98	0.93	0.92	0.95
b. GAQU models				
Regularized Logistic Regression	0.96	0.98	0.98	0.97
Linear Support Vector Machine	0.95	0.96	0.96	0.96
Radial Support Vector Machine	0.83	0.13	0.51	0.63
Random Forests	0.90	0.94	0.94	0.92
Kernelized k-Nearest Neighbours	0.96	1.00	1.00	0.98
c. VERD models				
Regularized Logistic Regression	0.43	0.95	0.69	0.53
Linear Support Vector Machine	0.52	0.95	0.71	0.60
Radial Support Vector Machine	0.00	1.00	NA	NA
Random Forests	0.44	0.98	0.87	0.58
Kernelized k-Nearest Neighbours	0.65	0.91	0.64	0.64

functional for testing and assessment. It took a total of 16 minutes to train and test the models for all three species. Results from the 10-fold cross-validation training phase are contained in Table 4. Performance varied based on the classifier and species of interest, with classifier evaluation metrics ranging from 0.00 to 1.00. For ECDO, all classifiers performed very well, with scores ranging from 0.92 to 0.99 depending on the classifier and evaluation metric. For the GAQU models, performance metrics were very good for all classifiers except for the radial support vector machine; the other GAQU classifiers had scores ranging from 0.90 to 1.00 on all metrics. The VERD classifiers displayed much greater performance variation. Though all VERD classifiers scored well on specificity (values ranged from 0.91 to 1.00), remaining metrics turned out poor to fair performances. A sensitivity score of 0.00 for the radial support vector machine resulted in NA values for that classifier's positive predictive value and F1 score. However, the random forests classifier scored 0.87 on the positive predictive value metric.

Because the k -nearest neighbours and support vector machines algorithms do not provide readily interpretable output with regard to predictive importance of acoustic features, here, we only report feature selection results from the regularized logistic regression and random forest models. Using *caret*'s *varImp()* function, for regularized logistic regression, variable importance is calculated based on the absolute values of the coefficients, with highest absolute values indicating the most important features for prediction. For random forests, variable importance is based on the 'out-of-bag' error, which, essentially, observes the impact of a variable on model accuracy by comparing it against the performance of trees that do not include this variable (see Liaw and Wiener (2002) for details).

Features summarizing statistical properties of the frequency spectrum served as the strongest predictors for distinguishing between target signals and false alarms. For ECDO, the regularized logistic regression classifier identified the standard error of the mean of the amplitude matrix as the top predictor, with spectral standard deviation, mean, and centroid also providing importance. The random forest model identified spectral mode as the top predictor, with spectral centroid and mean also providing some predictive value. For GAQU, the regularized logistic regression classifier identified spectral kurtosis and skewness as the top predictors, with several time-binned zero-crossing rates also providing importance. The random forest classifier identified a number of individual amplitude values as the best predictors, with correlation score also supplying predictive capacity. For VERD, the regularized logistic regression model identified spectral entropy as the top predictor, followed by spectral flatness and spectral kurtosis. The random forest model identified spectral skewness as the most important predictor, with spectral kurtosis and correlation score supplying some predictive impact. Several time and frequency contours and binned zero crossing rates also offered minor predictive value.

Assess classifier performance on the test set; create and assess performance-weighted class probability ensemble methods

Performances on the test data across the various metrics, classification approaches, and templates varied (Table 5). The random forests and kernelized k -nearest neighbours

Table 5. Assessment of classifier performance on the test data. The classifier performance metrics in this table can take on values between 0 (worst) and 1 (best). Rows indicate classifiers, and columns indicate performance metrics. Recall that these performance metrics only evaluate the subset of template-detected data, in support of the binary classification problem of distinguishing target signals from false alarms (described in Tables 1 and 2).

Classifier	Sensitivity	Specificity	Pos. Pred. Value	F1
a. ECDO Models:				
Regularized Logistic Regression	1.00	0.00	0.36	0.53
Linear Support Vector Machine	0.20	1.00	1.00	0.33
Radial Support Vector Machine	0.11	1.00	1.00	0.21
Random Forests	1.00	0.98	0.97	0.99
Kernelized k -Nearest Neighbours	1.00	0.97	0.95	0.97
Ensemble weighted by Sensitivity	1.00	0.95	0.92	0.96
Ensemble weighted by Specificity	1.00	1.00	1.00	1.00
Ensemble weighted by Pos. Pred. Value	1.00	1.00	1.00	1.00
Ensemble weighted by F1	1.00	0.98	0.97	0.99
b. GAQU Models:				
Regularized Logistic Regression	0.64	0.86	0.88	0.74
Linear Support Vector Machine	0.73	0.71	0.80	0.76
Radial Support Vector Machine	1.00	0.00	0.61	0.76
Random Forests	0.82	0.71	0.82	0.82
Kernelized k -Nearest Neighbours	0.73	0.86	0.89	0.80
Ensemble weighted by Sensitivity	0.73	0.86	0.89	0.80
Ensemble weighted by Specificity	0.73	0.86	0.89	0.80
Ensemble weighted by Pos. Pred. Value	0.73	0.86	0.89	0.80
Ensemble weighted by F1	0.73	0.86	0.89	0.80
c. VERD Models:				
Regularized Logistic Regression	0.00	1.00	NA	NA
Linear Support Vector Machine	0.07	0.93	0.20	0.10
Radial Support Vector Machine	0.00	1.00	NA	NA
Random Forests	0.80	1.00	1.00	0.89
Kernelized k -Nearest Neighbours	0.47	0.97	0.78	0.58
Ensemble weighted by Sensitivity	0.67	1.00	1.00	0.80
Ensemble weighted by Specificity	0.00	1.00	NA	NA
Ensemble weighted by Pos. Pred. Value	0.47	1.00	1.00	0.64
Ensemble weighted by F1	0.53	1.00	1.00	0.70

classifiers performed well on the ECDO test data, with performance evaluation metrics ranging from 0.95 to 1.00. Although the other three classifiers performed erratically on the test data, the good performances from the random forests and kernelized k -nearest neighbours classifiers caused all four weighted average ensemble classifiers to perform well on the test data, with scores ranging from 0.92 to 1.00 across metrics. The GAQU classifiers produced midrange performances, with positive predictive values and F1 scores that ranged from 0.61 to 0.89. All four weighted average ensemble classifiers produced the same results, with scores ranging from 0.73 on sensitivity, to 0.89 on positive predictive value. The VERD classifiers had much greater variation in performance. As during the training phase, the random forests classifier displayed the most promise in the testing phase, achieving scores that ranged from 0.80 (for sensitivity) to 1.00 (for specificity and positive predictive value). Meanwhile, the large class imbalance in the VERD data, with many false alarms and few target signals, resulted in regularized logistic regression and radial support vector machine models adept at identifying false alarms (specificity = 1.00) but incapable of identifying target signals (sensitivity = 0.00), consequently producing NA results for positive predictive value and F1 score. Indeed, for VERD, all five classifiers were effective at identifying false alarms, as indicated by specificities ranging from 0.93 to 1.00, but weaker at identifying target signals, with sensitivities ranging from 0 to 0.80.

The weighted ensemble approaches all performed similarly across performance metrics for both ECDO and GAQU, and displayed greater performance variation for VERD (Table 5). The ensemble classifier weighted by F1 score, upon which we chose to focus in advance, was a top-performing model for ECDO and GAQU on all metrics, producing scores ranging between 0.98 and 1.00 (ECDO), and from 0.73 to 0.89 (GAQU). For VERD, the ensemble classifier weighted by F1 score had a specificity and positive predictive value scores of 1.00, but was outperformed by the random forests classifier on sensitivity and F1 score.

ROC curves (Figure 6) of the training data generated acceptable areas under the curve (AUC) in most cases, aside from the radial support vector machine's performance for GAQU and VERD, which was indistinguishable from that of a random guess. ROC curves of the test data varied widely across species: the F1-weighted average 'ensemble' classifier produced the best ROC AUC results for all three species, with AUC values of 1 (ECDO), 0.82 (GAQU), and 0.99 (VERD). Precision-Recall curves (Figure 7) exhibited high performance for ECDO on the training data (all AUC \geq 0.97), high performance for GAQU despite the low amount of training data (aside from the radial support vector machine, all training set AUC \geq 0.98), and variable performance for VERD. On the test data, all three species had at least three classifiers that met or exceeded precision-recall AUC values of 0.80.

Objective 3: assess the performance of a trained and tested classifier on new detections

From April 2016 to May 2017, the template screening phase resulted in 9,485 new detections: 4,104 ECDO, 1,402 GAQU, and 3,979 VERD. Applying the trained and tested ensemble classifiers to these data yielded classifier sensitivities of 0.85 (ECDO), 0.59 (GAQU) and 0.54 (VERD) (Figure 8), compared to sensitivities of 1 in the

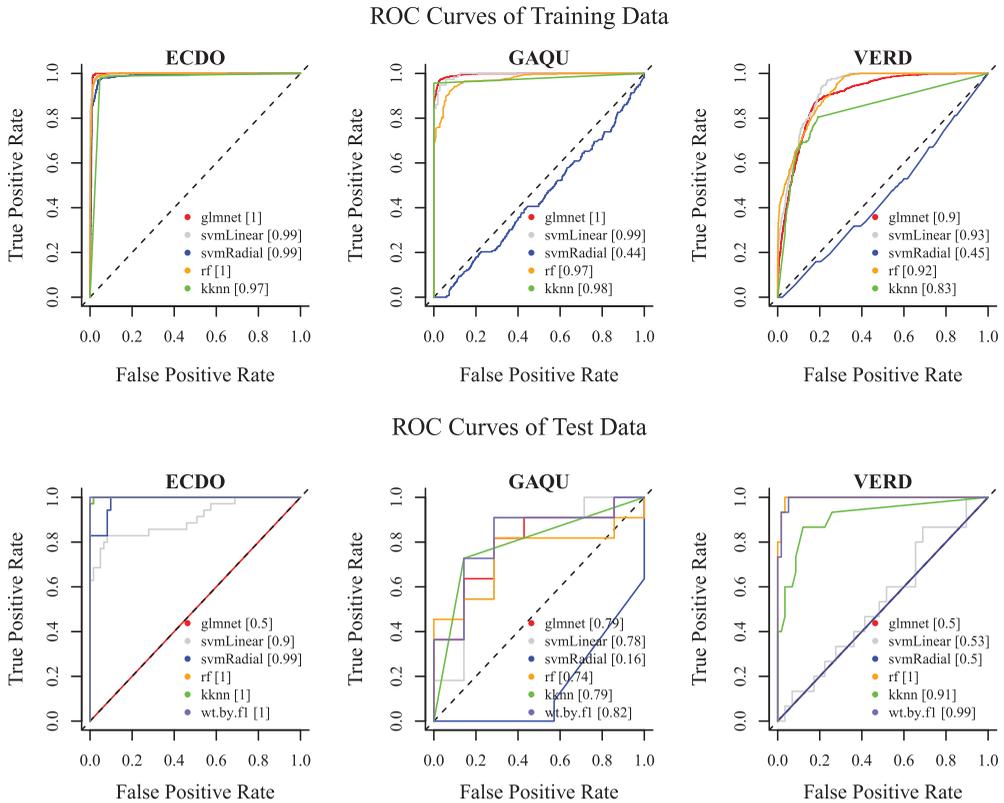


Figure 6. Receiver-Operator Characteristic (ROC) curves describing classifier performance during the training and testing phases. The upper panel shows ROC curves on the 10-fold cross-validated training data for the five classifiers. The bottom panel shows ROC curves on the test data. The ensemble classifiers only make predictions in the test phase, so the bottom panel also demonstrates the ensemble classifier weighted by F1 score. Area under the curve (AUC) is denoted next to each model's name in square brackets. Curves that reach into the upper left corner, with AUC values close to 1, show the best classification performance.

template screening phase. Classifier specificities were 0.98 (ECDO), 0.81 (GAQU), and 1.00 (VERD), compared to specificities of 0 for the template screening phase.

Overall positive predictive values from the classification application phase were 0.69 (ECDO), 0.95 (GAQU), and 0.77 (VERD), compared to positive predictive values of 0.06 (ECDO), 0.87 (GAQU), and 0.02 (VERD) for the template screening phase (Figure 8). F1 scores improved from 0.12 to 0.76 (ECDO) and from 0.04 to 0.63 (VERD) with the classifier system, but declined from 0.93 to 0.73 in the GAQU model (Figure 8).

The majority of false alarms for ECDO stemmed from wind and anthroponic sources such as faraway highway traffic noise, though several false cases were prompted by vocalizations from Greater Roadrunner (*Geococcyx californianus*), White-Winged Dove (*Zenaida asiatica*), and Mourning Dove (*Zenaida macroura*). Most GAQU false alarms resulted from electromagnetic interference, with a few due to Common Raven (*Corvus corax*) and Phainopepla (*Phainopepla nitens*). VERD false alarms occurred

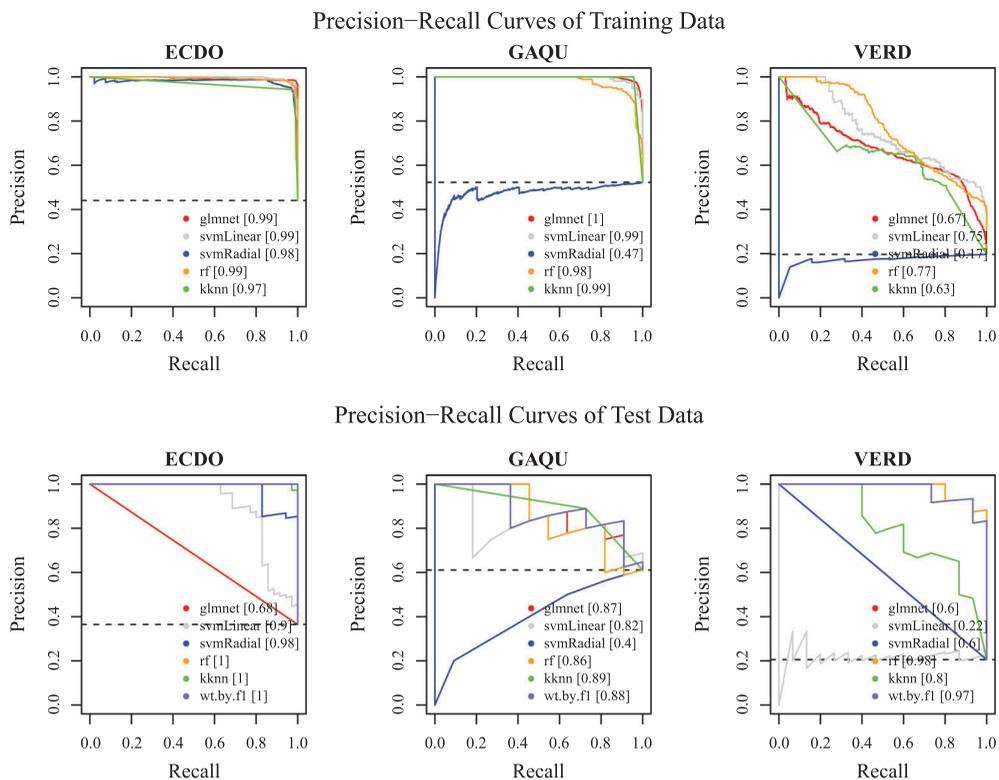


Figure 7. Precision-Recall Curves describing classifier performance during the training and testing phases. The upper panel shows PR curves on the 10-fold cross-validated training data for the five original classifiers. The bottom panel shows PR curves on the test data. The ensemble classifiers only make predictions in the test phase, so the bottom panel also demonstrates performance of the ensemble classifier weighted by F1 score. Area under the curve (AUC) is denoted next to each model’s name in square brackets. Curves that reach into the upper right corner, with AUC values close to 1, show the best classification performance.

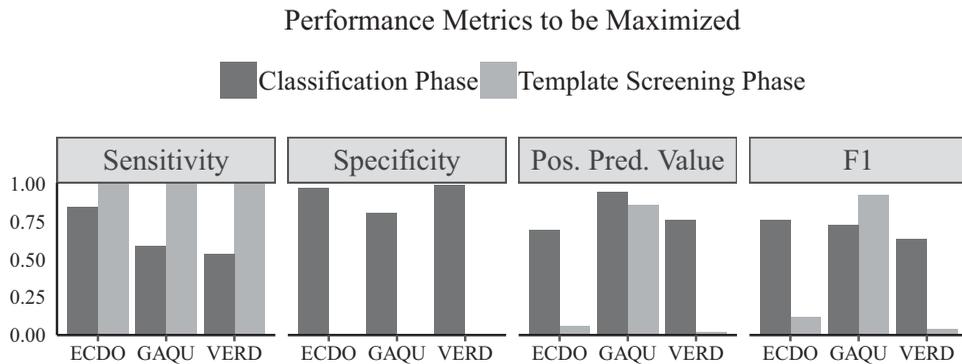


Figure 8. Comparison of performance metrics for the classification and template screening phases. Scores closest to 1 are desired for all metrics.

overwhelmingly as a consequence of electromagnetic interference, though some were caused by crickets and other songbirds.

Discussion

We demonstrated that statistical learning approaches can be used to mitigate false detections acquired within an automated acoustic wildlife monitoring dataset while retaining sufficient true detections for inference about species occurrence status. Compared to a basic template-matching system, the ability to identify false alarms improved, and positive predictive values increased in all cases demonstrated here, though there was a trade-off in capacity to identify all target signals: we observed a decrease in the F1 score for GAQU, though F1 scores for ECDO and VERD increased markedly. Since GAQU is known to be a highly gregarious and vocally available species (Gee et al. 2013), the observed increase in positive predictive value to 0.95 at the expense of sensitivity is likely a desirable trade-off. For a rare or acoustically cryptic species, this trade-off in comparative sensitivity with respect to detected events would not be advantageous.

One interesting outcome is that classifiers occasionally performed better on the test data than on the training data. The Verdin random forests ('rf') model provides an extreme example, performing markedly better during the testing phase than the training phase (compare training vs. testing phase performances in Figures 6 and 7; Tables 4 and 5). Schutten and Wiering (2016) showed that in a classification problem with a low amount of training data, when a case is very close to a classifier's decision boundary, it may be more often classified correctly during the test phase than during the training phase. This is because, during the cross-validation phase, model parameter settings may sacrifice performance during training so that the model will be able to generalize better to unseen cases during the testing phase, particularly when total N is low during training. We note the low number of target signal examples available for Verdin and cite this as a potential explanation for the unexpected result (see Table 3; Verdin training phase N = 35 target signals, testing phase N = 14 target signals).

In this work, we trained and tested our models on all detected events from March 2016, and used the trained models for prediction on all remaining data from April 2016-May 2017. Because there were only 631 detected events to manually label from March 2016, we were able to label all events. In practice, however, researchers may collect thousands of template-detected events over much longer periods of time before choosing to train and test a classification model on that data, which is useful if target species calls are likely to vary seasonally throughout the year. Additionally, the *AMMonitor* framework allows a classification model to be continually updated as new data are labelled. Here, researchers may not have time to label all available detections when training or updating a model. In such cases, to capture the most variation in target species sounds, we recommend taking a spatially and temporally-stratified sample of available detections to choose which events to label.

Several main concepts emerge from this work: first, although other auspicious classification methods implicitly strive to minimize false positives (e.g. Heinicke et al. 2015; Bas et al. 2017; Corrada-Bravo et al. 2017; Ranjard et al. 2017), none that we know of explicitly address false positive mitigation within the context of template-based or

threshold-based detection. In addition to making binary predictions about each detection's class, this method also has the advantage of producing probability values for each detection, which may be aggregated in dynamic occupancy models to predict the overall probability of species occurrence (Balantic and Donovan 2019).

Second, an advantage of this method is the opportunity to create ensemble classifiers that overtly capture a research program's monitoring needs with regard to vocalization characteristics of focal species. For example, researchers might opt for a positive predictive value-weighted ensemble classifier for gregarious species, or a sensitivity-weighted ensemble for rare or cryptic species. Research groups could make a variety of decisions about which classification method(s) to employ in production based on research objectives, characteristics of focal species, and classifier performance during the training and testing phase. Systematic decision tools do not presently exist in this arena, and the interpretation of classifier assessment metrics persists as an under-appreciated challenge when applying statistical learning approaches to real-world problems.

Third, template creation, including selection of the score threshold, is a highly influential component of the detection and classification process. The balance of target signals and false alarms occurring in a dataset is a function of the quality of data from which a template is constructed (Katz et al. 2016; Knight and Bayne 2018), the score threshold selected (Brauer et al. 2016; Katz et al. 2016; Knight et al. 2017), verification standards for manual labelling of target signal and false alarm training data, soundscape features such as non-target noise sources that contribute to detections, individual variation in sounds produced by the target species, and overall vocal availability of the target species, much of which is difficult to know in advance. Low template score thresholds may be necessary for research programs pursuing rare or vocally elusive species, or for circumstances where there is considerable uncertainty around how the template will perform in practice; it follows that large numbers of false alarms are possible, though there is little consistency across detection methodologies for detection threshold selection (Shonfield and Bayne 2017). Large numbers of false negatives, which we did not consider in this paper, are also possible. However, the template-matching approach upon which this work was built performed well in a comparison of different software methods (Knight et al. 2017), and the classification-based framework posed here may offer further improvements to template matching.

Fourth, though we are focused here on methods for eliminating false positives at the event detection level, other avenues for managing false positives may be appropriate depending on the ultimate objective of the study. For example, if detection data are to be used in occupancy models, Miller et al. (2011, 2013) have developed model-based methods for accommodating false positives. We have used the concept of target signal probability values (as described in this work) in dynamic occupancy models that apply the Miller et al. (2013) approach to acoustic monitoring data (Balantic and Donovan 2019). Chambert et al. (2015, 2018) also describe approaches applicable to acoustic monitoring, which are based on detection counts rather than on target signal probability values for any given detection. Banner et al. (2018) extended and applied the Chambert et al. (2018) model in the context of bat monitoring, with software available in the R package *OCacoustic* (accessible via USGS BitBucket). Such occupancy model-based alternatives to the mitigation of event-level false positives may be preferable

depending on study objectives and circumstances. Although we did not focus on event-level false negatives in this paper, many automated acoustic monitoring programs may have an ultimate objective of using automated detection data in occupancy models; it is worth emphasizing that occupancy model-based approaches do accommodate false negatives at the level of the site and species.

Increasing use of automated methods for detecting target species signals from audio recordings demonstrates the growing importance of accessible automated detection methods for acoustic monitoring programs. Template-based software methods like spectrogram cross-correlation and binary point matching present an accessible approach with a low barrier to entry for researchers (Hafner and Katz 2018), but factors like inappropriate score detection thresholds, an unwittingly poor template choice, noisy soundscapes, and acoustic features of the target signal may conspire to generate unacceptably high numbers of false alarms. Here, we investigated statistical learning methods that allow researchers to semi-automatically eliminate large numbers of false alarms, and showed that these methods may improve the monitoring quality of automated detection data from template-based detection systems.

Acknowledgements

This project was supported by the U.S. Bureau of Land Management. The lead author was supported by NSF IGERT grant 1144388. We thank Mark Massar (BLM) for providing the opportunity to work in the Riverside East Solar Energy Zone, Jonathan Katz for assistance with R programming, and reviewers for their helpful comments. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government. The Vermont Cooperative Fish and Wildlife Research Unit is jointly supported by the U.S. Geological Survey, University of Vermont, Vermont Department of Fish and Wildlife, and Wildlife Management Institute.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the U.S. Bureau of Land Management [31409]; U.S. National Science Foundation IGERT Program [1144388].

Data availability statement

The paper will be accompanied by a Github repository (<http://github.com/cbalantic/false-positive-mitigation>) that contains the script for this analysis, the trained/tested classification models, the database of detections and templates, and functions necessary to produce the results.

ORCID

Cathleen M. Balantic  <http://orcid.org/0000-0003-2043-0975>

References

- Acevedo MA, Corrada-Bravo CJ, Corrada-Bravo H, Villanueva-Rivera LJ, Aide TM. 2009. Automated classification of bird and amphibian calls using machine learning: A comparison of methods. *Ecol Inform.* 4:206–214. doi:10.1016/j.ecoinf.2009.06.005
- Agranat ID. 2009. Automatically identifying animal species from their vocalizations. Concord (MA): Wildlife Acoustics, Inc.
- Aide TM, Corrada-Bravo C, Campos-Cerqueira M, Milan C, Vega G, Alvarez R. 2013. Real-time bioacoustics monitoring and automated species identification. *PeerJ.* 1:e103. doi:10.7717/peerj.103
- Avisoft Bioacoustics. 2016. Avisoft-SASLab Pro version 5. 2.10[Computer Software]. [accessed 2019 April 22]. <http://www.avisoft.com/>.
- Balantic CM, Donovan TM. 2019. Dynamic wildlife occupancy models using automated acoustic monitoring data. *Ecol Appl.* e01854.
- Banner KM, Irvine KM, Rodhouse TJ, Wright WJ, Rodriguez RM, Litt A. 2018. Improving geographically-extensive acoustic survey designs for modeling species occurrence with imperfect detection and misidentification. *Ecol Evol.* 8(i. 12):6144–6156.
- Bas Y, Bas D, Julien JF. 2017. Tadarida: A toolbox for animal detection on acoustic recordings. *J Open Res Software.* 5(1):6. doi:10.5334/jors.154.
- Bioacoustics Research Program. 2015. Raven Pro 1.5: interactive sound analysis software [Computer Software]. [accessed 2019 April 22]. <http://www.birds.cornell.edu/raven>.
- Bishop CM. 2006. Pattern recognition and machine learning. Springer. ISBN. 0387310738:9780387310732.
- Boser BE, Guyon IM, Vapnik VN. 1992. A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory - COLT '92. New York, USA: ACM Press. pp. 144–152.
- Brauer C, Donovan T, Mickey R, Katz J, Mitchell B. 2016. A comparison of acoustic monitoring methods for common anurans of the northeastern United States. *Wildl Soc Bull.* 40:140–149. doi:10.1002/wsb.619
- Breiman L. 2001. Random forests. *Mach Learn.* 45:5–32.
- Buxton RT, Jones IL. 2012. Measuring nocturnal seabird activity and status using acoustic recording devices: applications for island restoration. *J Field Ornith.* 83:47–60. doi:10.1111/j.1557-9263.2011.00355.x
- Cerqueira MC, Aide MT. 2016. Improving distribution data of threatened species by combining acoustic monitoring and occupancy modeling. *Methods Ecol and Evol.* 7(11):1340–1348. doi:10.1111/2041-210X.12599.
- Chambert T, Miller DAW, Nichols JD. 2015. Modeling false positive detections in species occurrence data under different study designs. *Ecology.* 96(2):332–339. doi:10.1890/14-1507.1.
- Chambert T, Waddle JH, Miller DAW, Walls SC, Nichols JD. 2018. A new framework for analyzing automated acoustic species detection data: occupancy estimation and optimization of recordings post-processing. *Methods Ecol and Evol.* 9:560–570. doi:10.1111/2041-210X.12910
- CinixSoft. 2014. CinixSoft remote schedule voice recorder v4.2.0. [Android App]. [accessed 2019 April 22]. <http://www.cinixsoft.com/>.
- Corrada-Bravo CJC, Berrios RA, Aide TM. 2017. Species-specific audio detection: a comparison of three template-based detection algorithms using random forests. *PeerJ Comput Sci.* 3:e113. doi:10.7717/peerj-cs.113
- Cover T, Hart P. 1967. Nearest neighbor pattern classification. *IEEE Trans Inf Theory.* 13:21–27. doi:10.1109/TIT.1967.1053964
- Davis J, Goadrich M. 2006. The relationship between Precision-Recall and ROC curves. Proceedings of the 23rd international conference on Machine Learning, Pittsburg, PA.
- Digipom. 2016. Easy voice recorder Pro [Android App]. [accessed 2019 April 22]. <http://www.digipom.com/portfolio-items/easy-voice-recorder/>.

- Duan S, Zhang J, Roe P, Wimmer J, Dong X, Truskinger A, Towsey M. 2013. Timed probabilistic automaton: a bridge between Raven and Song Scope for automatic species recognition. In Munoz-Avila H, Stracuzzi DJ (Eds). Proceedings of the Twenty-Fifth Innovative Applications of Artificial Intelligence Conference. Washington, USA: AAAI, Bellevue. pp. 1519–1524.
- Figueroa H. 2012. XBAT [Computer Software]. Bioacoustics research program. <http://www.xbat.org>.
- Furnas BJ, Callas RL. 2014. Using automated recorders and occupancy models to monitor common forest birds across a large geographic region. *J Wildl Manage.* 79(2):325–337. doi:10.1002/jwmg.821.
- Gee J, Brown DE, Hagelin JC, Taylor M, Galloway J. 2013. Gambel's Quail (*Callipepla gambelii*). *The Birds of North America*. Rodewald PGEd. Ithaca: Cornell Lab of Ornithology. doi: 10.2173/bna.321.
- Gibb R, Browning E, Glover-Kapfer P, Jones KE. 2018. Emerging opportunities and challenges for passive acoustics in ecological assessment and monitoring. *Methods Ecol and Evol.* 10(2):169–185.
- Hafner S, Katz J. 2018. monitoR: acoustic template detection in R. R package version 1.0.7. [accessed 2019 April 22]. <http://www.uvm.edu/rsenr/vtcfwru/R/?Page=monitoR/monitoR.htm>.
- Hastie T, Tibshirani R, Friedman JH. 2009. The elements of statistical learning: data mining, inference, and prediction. 2nd ed. New York: Springer Series in Statistics.
- Heinicke S, Kalan AK, Wanger OJ, Mundry R, Lukashevich H, Kuhl HS. 2015. Assessing the performance of a semi-automated acoustic monitoring system for primates. *Methods Ecol and Evol.* 6(7):753–763. doi:10.1111/2041-210X.12384.
- Katz J, Hafner SD, Donovan T. 2016. Assessment of error rates in acoustic monitoring with the R package monitoR. *Bioacoustics.* 25:177–196. doi:10.1080/09524622.2015.1133320
- Knight EC, Bayne EM. 2018. Classification threshold and training data affect the quality and utility of focal species data processed with automated audio-recognition software. *Bioacoustics.* doi:10.1080/09524622.2018.1503971
- Knight EC, Hannah KC, Foley G, Scott C, Mark Brigham R, Bayne E. 2017. Recommendations for acoustic recognizer performance assessment with application to five common automated signal recognition programs. *Avian Conserv Ecol.* 12(2):14. doi:10.5751/ACE-01114-120214.
- Kuhn M. 2016. caret: classification and regression training. R package version 6.0-71. [accessed 2019 April 22]. <http://CRAN.R-project.org/package=caret>.
- Liaw A, Wiener M. 2002. Classification and regression by randomForest. *R News.* 2(3):18–22. URL: <https://cran.r-project.org/web/packages/randomForest/randomForest.pdf>
- Mac Aodha O, Gibb R, Barlow KE, Browning E, Firman M, Freeman R, Jones KE. 2018. Bat detective—deep learning tools for bat acoustic signal detection. *PLoS Comput Biol.* 14:1–19. doi:10.1371/journal.pcbi.1005995
- Marques TA, Thomas L, Martin SW, Mellinger DK, Ward JA, Moretti DJ, Harris D, Tyack PL. 2013. Estimating animal population density using passive acoustics. *Biological Reviews.* 88:287–309. doi:10.1111/brv.2013.88.issue-2
- Mellinger DK, Clark CW. 2000. Recognizing transient low-frequency whale sounds by spectrogram correlation. *J Acoust Soc Am.* 107(6):3518–3529.
- Miller DAW, Nichols JD, Gude JA, Rich LN, Podruzny KM, Hines JE, Mitchell MS. 2013. Determining occurrence dynamics when false positives occur: estimating the range dynamics of wolves from public survey data. *PLoS One.* 8:e65808. doi:10.1371/journal.pone.0065808
- Miller DAW, Nichols JD, McClintock BT, Grant EHC, Bailey LL, Weir LA. 2011. Improving occupancy estimation when two types of observational error occur: non-detection and species misidentification. *Ecology.* 92:1422–1428. doi:10.1890/10-1396.1
- Ovaskainen O, Moliterno de Camargo U, Somervuo P. 2018. Animal Sound Identifier (ASI): software for automated identification of vocal animals. *Ecol Lett.* 21(8):1244–1254. doi:10.1111/ele.13092.
- Pollock KH, Nichols JD, Simons TR, Farnsworth GL, Bailey LL, Sauer JR. 2002. Large scale wildlife monitoring studies: statistical methods for design and analysis. *Environmetrics.* 13(2):105–119. doi:10.1002/env.514.

- Potamitis I, Ntalampiras S, Jahn O, Riede K. 2014. Automatic bird sound detection in long real-field recordings: applications and tools. *Appl Acoust.* 80:1–9. doi:[10.1016/j.apacoust.2014.01.001](https://doi.org/10.1016/j.apacoust.2014.01.001)
- Powers DMW. 2007. Evaluation: from precision, recall, and F-Factor to ROC, informedness, markedness & correlation. School of Informatics and Engineering, Flinders University. Adelaide (Australia). Technical Report SIE-07-001.
- Ranjard L, Reed BS, Landers TJ, Raynar MJ, Friesen MR, Sagar RL, Dunphy BJ. 2017. MatlabHTK: a simple interface for bioacoustics analyses using hidden Markov models. *Methods Ecol and Evol.* 8(5):615–621. doi:[10.1111/2041-210X.12688](https://doi.org/10.1111/2041-210X.12688).
- Romagosa CM. 2012. Eurasian Collared-Dove (*Streptopelia decaocto*), The Birds of North America. Rodewald PGE. Ithaca: Cornell Lab of Ornithology. Retrieved from the Birds of North America: <https://birdsna.org/Species-Account/bna/species/eucdov>. doi: [10.2173/bna.630](https://doi.org/10.2173/bna.630).
- Royle JA, Link WA. 2006. Generalized site occupancy models allowing for false positive and false negative errors. *Ecology.* 87:835–841. doi:[10.1890/0012-9658\(2006\)87\[835:GSOMAF\]2.0.CO;2](https://doi.org/10.1890/0012-9658(2006)87[835:GSOMAF]2.0.CO;2)
- Ruiz-Gutierrez V, Hooten MB, Campbell Grant EH. 2016. Uncertainty in biological monitoring: a framework for data collection and analysis to account for multiple sources of sampling bias. *Yoccoz NEd. Methods Ecol and Evol.* 7:900–909. doi: [10.1111/2041-210X.12542](https://doi.org/10.1111/2041-210X.12542).
- Schutten M, Wiering MA. 2016. An analysis on better testing than training performances on the Iris dataset. In Proceedings of Belgian Dutch Artificial Intelligence Conference; Nov 2016. Amsterdam (The Netherlands). pp. 10–11.
- Shonfield J, Bayne EM. 2017. Autonomous recording units in avian ecological research: current use and future applications. *Avian Conserv Ecol.* 12(1):14. doi:[10.5751/ACE-00974-120114](https://doi.org/10.5751/ACE-00974-120114).
- Stowell D, Wood M, Stylianou Y, Glotin H. 2016. Bird detection in audio: a survey and a challenge. *IEEE Intl Workshop Mac Learn for Signal Proces, Salerno, Italy.* doi:[10.1109/MLSP.2016.7738875](https://doi.org/10.1109/MLSP.2016.7738875)
- Sueur J, Aubin T, Simonis C. 2008. Seewave: a free modular tool for sound analysis and synthesis. *Bioacoustics.* 18:213–226. doi:<http://dx.doi.org/10.1080/09524622.2008.9753600>
- Towsey M, Planitz B, Nantes A, Wimmer J, Roe P. 2012. A toolbox for animal call recognition. *Bioacoustics.* 21:107–125. doi:[10.1080/09524622.2011.648753](https://doi.org/10.1080/09524622.2011.648753)
- Webster MD. 1999. Verdin (*Auriparus flaviceps*), The Birds of North America (P. G. Rodewald, Ed.). Ithaca: cornell lab of ornithology; retrieved from the Birds of North America. <https://birdsna.org/Species-Account/bna/species/verdin>. doi: [10.2173/bna.470](https://doi.org/10.2173/bna.470).
- Weinberger KQ, Blitzer J, Saul LK. 2006. Distance metric learning for large margin nearest neighbor classification. In *Adv in Neural Inf Process Syst.* 18:1473–1480.
- Wildlife Acoustics. 2016. Kaleidescope [Computer Software]. [accessed 2019 April 22]. <http://www.wildlifeacoustics.com>.
- Zhu X, Davidson I, Eds. 2007. Knowledge discovery and data mining. New York: IGI Global.
- Zou H, Hastie T. 2005. Regularization and variable selection via the elastic net. *J R Stat Soc: Ser B (Stat Method).* 67:301–320. doi:[10.1111/j.1467-9868.2005.00503.x](https://doi.org/10.1111/j.1467-9868.2005.00503.x)

Appendix 1. Summary of acoustic features used as inputs to classification models that predict whether a detection is a true or false positive

Feature	Description
Raw amplitude values	Acquired by way of Fourier Transform. Every single raw amplitude value (in dB) in the matrix of a detected event. Each amplitude value is a measure of signal intensity at that point, and is rendered in coloured shading on the spectrogram.
Correlation Score	Correlation score produced by moving window analysis during template matching.
Zero-Crossing Rate for each time bin	$zcr = 0.5 * \text{mean}(\text{abs}(\text{sgn}(x(t + 1)) - \text{sgn}(x(t))))$ with: N the length of the signal x , and where: $\text{sgn}(x(t)) = 1$ if $x(t) \geq 0$ and $\text{sgn}(x(t)) = -1$ if $x(t) < 0$.
Time Contours for each time bin	Amplitude probability mass function for each time bin
Frequency Contours for each frequency bin	Amplitude probability mass function for each frequency bin
Time.P1	Time initial percentile based on cumulative distribution function generated from time probability mass function
Time.M	Time median based on cumulative distribution function generated from time probability mass function
Time.P2	Time terminal percentile based on cumulative distribution function generated from time probability mass function
Time.IPR	Time interpercentile range based on cumulative distribution function generated from time probability mass function
Freq.P1	Frequency initial percentile based on cumulative distribution function generated from frequency probability mass function
Freq.M	Frequency median based on cumulative distribution function generated from frequency probability mass function
Freq.P2	Frequency terminal percentile based on cumulative distribution function generated from frequency probability mass function
Freq.IPR	Frequency interpercentile range based on cumulative distribution function generated from frequency probability mass function
Spectral Mean	Sum of the product of the spectrogram intensity (in dB) and the frequency, divided by the total sum of spectrogram intensity.
Spectral Standard Deviation	Standard deviation of the mean frequency
Spectral SEM	Standard error of the mean of the amplitude matrix
Spectral Median	The value of the halfway point in ordered frequency values in the data set
Spectral Mode	Dominant frequency of the amplitude matrix
Q1: First quartile (0.25 quantile)	The first quartile; a measure of statistical dispersion. Value that divides the lowest 25% of data from the highest 75%.
Q3: Third quartile (0.75 quantile)	The third quartile; a measure of statistical dispersion. Value that divides the highest 25% of data from the lowest 75%.
Interquartile range (IQR)	$IQR = Q3 - Q1$. A statistical dispersion (variability) measure based on dividing the detected event into quartiles.
Spectral Centroid	$C = \frac{\sum(x*y)}{N}$ with x = frequencies, y = relative amplitude of the i frequency, and N = number of frequencies.
Spectral Skewness	A measure of signal asymmetry. $S = \frac{\sum((x-\text{mean}(x))^3)/(N-1)}{sd^3}$ Spectrum asymmetry increases with $ S $.
Spectral Kurtosis	A measure of signal peakedness. $K = \frac{\sum((x-\text{mean}(x))^4)/(N-1)}{sd^4}$
Spectral Flatness	$F = \frac{N * (\prod(y_i)^{1/N})}{\sum(y_i)}$ With y = relative amplitude of the i frequency, and N = number of frequencies. Ratio between geometric mean and arithmetic mean. Flatness of noisy signals are closer to 1; flatness of pure tone signal is closer to 0.
Spectral Entropy (Shannon's)	$S = -\sum(y \log y) / \log(N)$. Noisy signals have S closer to one, while pure tone signals have S closer to 0.

Appendix 2. Example of weighted average ensemble probability computation

Each performance-weighted ‘ensemble’ method produced a single class probability of true detection ($Target\ Signal_e$), calculated as

$$P(Target\ Signal)_e = [\theta] \cdot [S]$$

where $[\Theta]$ is a vector of length five consisting of the individual probability of a target signal for each of the five classifiers, and $[S]$ is a length five vector of normalized performance scores that sums to 1.0. The $[S]$ vector is computed by dividing each classifier’s score on the metric of interest by the maximum score within the vector, resulting in a vector that represents how proportionally close each score is to the top score for that metric, which is then normalized to sum to 1

1. For a single detection, take the vector of true positive class probabilities for all five classifiers, $[P]$:
 $[P] = [p_1, p_2, p_3, p_4, p_5]$
 $[P] = [0.02, 0.29, 0.20, 0.29, 0.09]$
2. Gather each classifier’s score on the metric of interest (e.g. Sensitivity) in vector $[S]$
 $[S] = [0.86, 0.77, 0.00, 0.86, 0.73]$
3. Compute a vector representing how proportionally close each score is to the highest score:
 $[D] = [S]/\max[S]$
 $[D] = [0.86, 0.77, 0.00, 0.86, 0.73]/0.86 = [1.00, 0.895, 0.00, 1.00, 0.849]$
4. Compute a vector of weights normalized to add to 1, $[N]$:
 $[N] = [D]/\sum([D])$
 $[N] = [1.00, 0.895, 0.00, 1.00, 0.849]/3.74 = [0.27, 0.24, 0.00, 0.27, 0.23]$
5. Compute dot-product of the vector of probabilities $[P]$ times the vector of normalized weights $[N]$ to get a single weighted-average value for the class probability, p_w .
 $p_w = [P] \cdot [N] = [0.02, 0.29, 0.20, 0.29, 0.09] \cdot [0.27, 0.24, 0.00, 0.27, 0.23] = (0.02 \cdot 0.27) + (0.29 \cdot 0.24) + (0.20 \cdot 0.00) + (0.29 \cdot 0.27) + (0.09 \cdot 0.23) = \mathbf{0.17}$
6. For $p_w < 0.5$, class = false alarm. For $p_w > 0.5$, class = target signal. If ties, coinflip for class.
 $p_w = \mathbf{0.17} = \mathbf{false\ alarm\ class}$