Data-Centric Methods for Environmental Sound Classification With Limited Labels

Ali Raza Syed, *Student Member, IEEE*, Enis Berk Çoban, *Student Member, IEEE*, Dara Pir, *Member, IEEE*, and Michael Mandel, *Member, IEEE*

***Abstract*—Arctic boreal forests are warming at a rate 2-3 times faster than the global average. It is important to understand the effects of this warming on activities of animals that migrate to and within these environments annually to reproduce. Acoustic sensors can monitor a wide area relatively cheaply, producing large amounts of data. Yet, only a small proportion of the recorded data can be labeled by hand making it challenging to train high performing sound classifiers for ecoacoustic research. In this work, we explore data-centric methods for improving model performance by utilizing labels more efficiently. We show that indeed data augmentation for a DNN-based multi-label sound classifier yields a relative improvement (37%) in AUC performance. We are able to boost this further by 56% with a novel data valuation method. Our method estimates Shapley values for a multi-label DNN classifier enabling curation of a high quality training set and identification of data quality issues. We demonstrate that with our novel method, we can achieve these gains using as little as 40% of the labeled training data.**

***Index Terms*—Ecoacoustics, environmental sound classification, data-centric machine learning, data augmentation, data valuation, Shapley values, limited labels, data curation.**

I. Introduction

ARCTIC boreal forests, which are crucial for the breeding success of various species such as caribou, waterfowl, and songbirds, have experienced warming at a rate that is two to three times the global average, as well as an intensification of human development [1]. Thus, it is critical to monitor these ecosystems to measure the impact of climate change on these species, and to devise effective interventions and mitigation strategies. One promising approach is the use of acoustic sensors, which offer several advantages over visual sensors.

Manuscript received 14 May 2023; revised 15 December 2023 and 26 February 2024; accepted 27 February 2024. Date of publication 13 June 2024; date of current version 2 October 2024. This work was supported by the National Science Foundation (NSF) under Grant OPP-1839185. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Emmanouil Benetos. *(Ali Raza Syed and Enis Berk Çoban contributed equally to this work.) (Corresponding author: Ali Raza Syed.)*

Ali Raza Syed and Enis Berk Çoban are with the Department of Computer Science, The Graduate Center, City University of New York, New York, NY 10016 USA (e-mail: asyed2@gradcenter.cuny.edu; ecoban@gradcenter.cuny.edu).

Dara Pir is with the Information Technology Program, Guttman Community College, City University of New York, New York, NY 10018 USA (e-mail: dpir@gradcenter.cuny.edu).

Michael Mandel is with the Department of Computer and Information Science, Brooklyn College, Brooklyn, NY 11210 USA, and also with the Department of Computer Science, The Graduate Center, New York, NY 10016 USA (e-mail: mim@sci.brooklyn.cuny.edu).

Digital Object Identifier 10.1109/TASLP.2024.3414332

Compared to visual sensors, acoustic sensors consume less power and data bandwidth, and have a wider field of "view" and longer range. Moreover, the audio modality, when used in conjunction with other modalities, can enhance recognition performance. Once deployed, acoustic sensors can collect vast amounts of data at a rate that requires automated procedures for classification and categorization [2].

In recent years, deep learning has been increasingly used for sound event classification [2], [3], [4]. The performance of these models depends on the quantity and quality of the labeled data used to train them. Yet, obtaining high quality labeled data is costly and sometimes prohibitive [5]. Especially when the quantity of labeled data is limited, its quality can significantly impact performance [6]. Ecoacoustic data can contain multiple overlapping sounds in short clips and annotating these requires engaging experts in a costly, laborious, and error-prone process [4]. For this reason, the amount of labeled data is often limited and subject to quality concerns [2]. In this work, we explore data-centric methods for making maximal use of limited labeled data to train and produce high performing sound classifiers for aiding ecoacoustic researchers.

In machine learning, the usual approach is model-centric. This involves starting with a fixed dataset and iteratively developing the model to improve system performance. In contrast, data-centric methods assume a fixed training procedure for the model and iterate over the data. This approach is driven by the observation that performance can depend not only on the amount of training data but also on the quality of the data, and how representative those are of conditions where the model will be deployed. When the amount of data is limited, data-centric approaches can produce significant gains, particularly for deep learning [6]. While there are methods for utilizing unlabeled data (see Section II-C), in this work, we explore ways to efficiently utilize existing *labeled* data. We expand on early experiments, where we found that model- and data-centric methods for classifying limited labeled data can be complementary [7]. Specifically, we investigate performance gains in environmental sound classification from two data-centric methods: *data augmentation* and *data valuation*.

Data augmentation increases the amount of available training data by simulating new and realistic examples based on existing ones [8]. We include data augmentation in our investigation because it has become a standard part of the pipeline for training modern neural networks and must be included when fixing the model training procedure to study any other data-centric

technique. However, we do not perform an extensive comparison of augmentation techniques and limit ourselves to two state of the art techniques: Mixup [9] and SpecAugment [10]. They generate different perspectives of the same data, helping deep neural networks to generalize. In recent years, both techniques have gained popularity in the field of audio classification.

Data valuation for machine learning is concerned with quantifying the importance of a training example for a given model. Ranking the data based on their value allows for identification of high and low quality data for curating training datasets. While there are several heuristics for ranking data, only a few principled methods exist for valuing their importance for any given model. Shapley valuation [11] is a recent and popular method [12]. Efficient estimation of Shapley values for deep neural networks (DNN) has been limited to classification tasks using accuracy as the metric (see Section II-B3). In this work, we introduce a novel method to estimate Shapley values for a DNN performing a multi-classification task where the metric of interest is AUC (area under the ROC curve [13]).

Our contributions: We present a novel method for estimating Shapley values of training examples for a DNN-based multi-label sound classifier. In early experiments, we presented results [7] demonstrating the method's effectiveness only for a proxy model. In this work, we validate its effectiveness for the original CNN model. To our knowledge, this is the first demonstration of valuing data for a DNN-based multi-label classifier (that uses AUC as a performance metric). We show that the method of valuing and curating limited labeled data can yield a sound classifier with significant performance gains. Moreover, this approach complements any gains due to data augmentation techniques and a model-centric method (global temporal pooling). We also present methods for assessing data quality issues, particularly annotation errors, through analysis based on Shapley values. Specifically, we show how to drill down in multi-label data and identify data quality issues within classes. Finally, we re-annotated our entire dataset for this work. While there are techniques for dealing with noisy annotations in training data, it has been shown that annotation errors are pervasive in common validation and test sets, and can cast doubt on benchmark results [14]. We use the the re-annotated validation and test sets for scoring our methods, thus increasing confidence in our method. Further, using the re-annotated training set as "ground truth" provides a rare opportunity to assess the effectiveness of our data valuation approach. It offers a sharp contrast to data valuation experiments that artificially introduce label noise or examine randomly sampled points for quality issues. We present an empirical distribution of Shapley values for a multi-label classifier, and their correspondence to annotation errors for a real-world dataset of environmental sound.

II. Related Work

A. Data Augmentation

We investigate two data augmentation methods: Mixup and SpecAugment and mixup, the most popular techniques in DCASE 2022 for audio scene classification and bioacoustic event detection [15], [16]. Mixup [9], originally developed for computer vision, creates new samples to balance a dataset [17], [18]. It has proven effective for sound event detection. For instance, it was utilized by top-performing teams in the DCASE 2018 audio tagging challenge [19] and enabled a convolutional recurrent neural network based system to outperform baselines [20]. SpecAugment [10] was introduced for end-to-end speech recognition and achieved state-of-the-art performance on the LibriSpeech 960 and and Switchboard 300 h tasks. A Conformer-based system using a combination of Mixup, SpecAugment, time-shifting, and noise augmentation demonstrated superior performance in sound event detection [21].

B. Data Valuation

Data Valuation in machine learning is concerned with quantifying the value of training examples for a model based on their relative contribution to that model's performance. The primary impetus for data valuation has been to compensate data vendors (e.g., [22]). However, it can also be used for data curation and selection by ranking and selecting a subset of the data for high model performance with faster training. In our context, audio data [23] is relatively cheap to acquire but laborious and expensive to annotate, especially since multiple overlapping sounds are present in short clips [4]. Thus, we explore the feasibility of data valuation for curating our limited labeled acoustic data and identifying examples that are particularly useful or misleading for our model. We avoid the use of heuristic-based methods for measuring the importance of data and limit ourselves to approaches that directly measure the utility of an example for a model. Two primary methods have been employed in data valuation for machine learning: influence functions [24] and Data Shapley values [12].

Influence functions, arising from robust statistics [25], determine the value of an example by measuring the change in parameters of a model when an example is given a little more weight than the other examples in a training set. They have been shown to asymptotically approximate the Leave-One-Out (LOO) value [26]. The LOO value measures the contribution of an example to a specific training set, the "left in" values. Although influence functions have been applied successfully for data valuation [24], they can be difficult or expensive to compute since their formulation involves inverting the model's Hessian matrix (second order derivative of the loss function).

Shapley valuation, originating from economics and cooperative game theory [11], provides a fair method for allocating rewards to individual players based on their relative contributions [11]. In machine learning, Shapley valuation has been used to determine the value of examples based on their importance for a model [12], [27], [28], [29]. Data valuation experiments have shown that Shapley values are better at quantifying the utility of data for complex models like deep neural network classifiers [12]. For these reasons, we employ Shapley values in our work.

1) Shapley Value: In the supervised learning setting, we view training examples as participants in a game. The learning algorithm uses these players to achieve a reward: the performance measured on a held-out set. The Shapley value determines the relative value of the examples for the model by allocating the performance metric among the training examples based on their

utility for the learning algorithm. The Shapley value of an example is defined as follows. Suppose a dataset $\mathcal{D}$ with N examples available for training, and a dataset $\mathcal{D}_{\text{eval}}$ with N_{eval} examples held-out for evaluation. If learning algorithm $\mathcal{A}$ trains on a subset of examples, $S \subseteq \mathcal{D}$, its performance on $\mathcal{D}_{\text{eval}}$ is given by $\text{F}(S)$ (e.g.,F may compute the AUC score). The Shapley value $\sigma(x_i)$ of training example x_i is the expected marginal contribution of x_i to *any* subset of the remaining training examples $S \subseteq D \setminus \{x_i\}$:

$$\sigma(x_i) = \frac{1}{N} \sum_{S \subseteq \mathcal{D} \setminus \{x_i\}} \frac{1}{\binom{N-1}{|S|}} \left[\text{F}(S \cup \{x_i\}) - \text{F}(S)\right] \quad (1)$$

Shapley valuation is the unique allocation scheme satisfying fairness axioms for equitable distribution of rewards in cooperative games [11]. It makes no assumptions about the learning algorithm, the distribution of the training data, or whether the examples are independent or identically distributed.

2) Interpreting Shapley Values: As defined, the Shapley value of an example is the average effect on model performance when that example is added to any random subset of the training data. Thus, an example with a positive Shapley value implies that including that example in a training set would result in improving the performance of the model. Similarly, an example with a negative Shapley value implies the opposite: including that example in training would result in degrading the performance of the model. Examples with Shapley values close to zero are expected to have little effect on model performance when included in a training set. This interpretation is in contrast to the that of Shapley values of *features*, used for model interpretability [30]. The crucial difference is that those Shapley values are calculated based on the model *response* and not its performance metric.

3) Estimating Shapley Values: In general, exact calculation of Shapley values is intractable for realistic dataset sizes. Thus, Shapley values are usually estimated using Monte Carlo algorithms (MC) that can run in polynomial time [31]. The Truncated Monte-Carlo (TMC) algorithm can estimate Shapley values for any model [12]. However, TMC requires re-training the model in each iteration, which is prohibitive for a deep CNN classifier. The KNN-Shapley algorithm [27] computes exact Shapley values for k-nearest neighbors (KNN) models in quasi-linear time without any re-training. For deep neural network (DNN) classifiers, it is possible to take advantage of this faster algorithm by learning a proxy KNN model for the DNN, then computing the Shapley values for the proxy KNN [32]. Empirical results show that these values yield a coarse grained ranking of example importance, thus approximating utility for the original DNN model [32]. However, KNN-Shapley relies on accuracy as the metric of interest. We use a CNN-based multi-label sound classifier that is evaluated using the AUC score; thus, we introduce and evaluate a novel estimation method in Section III-B-1.

C. Environmental Sound Classification With Limited Labels

Despite limited labeled data, prior efforts have been able to utilize deep learning for the classification of environmental sounds, e.g.,to predict broad soundscape components, such as human noise, wildlife vocalizations, and weather phenomena [33] or to measure audible biotic and anthropogenic acoustic activity [34]. Our classification task is akin to these, but with a specific emphasis on natural and less urbanized environments. We also delve into a more fine-grained classification of ecoacoustic events, encompassing songbirds, waterfowl, grouse, insects, and aircraft.

Ecoacoustics researchers address label limitations by incorporating *unlabeled* data through semi-supervised or active learning. Semi-supervised approaches typically employ transfer learning with: pseudo-labeling e.g.,to classify calls of birds and amphibian species [35], pre-trained representations e.g.,to classify orca call types [36], and mean teacher based learning e.g.,to classify sound events from domestic environments [37]. All these methods operate under the shared assumption that high-quality initial labeled data is available. In reality, mislabeled data can skew the initial model and impact performance [38]. Our focus on identifying misleading low-quality data and curating high-quality data complements existing efforts. It enhances label quality with minimal human intervention, serving as a crucial pre-processing step for semi-supervised learning. Our method is also complementary to active learning efforts employed for learning with minimal supervision [39], classification of rare events [40], [41], recognition of novel classes [42] and bird species [43], and reducing annotation efforts e.g.,in low-resource speech recognition [44] and sound classification [45]. These approaches use heuristics for ranking *unlabeled* data based on their expected utility for the model. In contrast, data valuation methods rank the *labeled* data based on their actual contribution to the model's performance on a held-out set. These valuations allow for curating better training sets by identifying high quality data, thus complementing active learning methods.

Another closely related task to our work is *instance selection* [46]. It seeks a high quality subset of the training data with algorithms falling into two categories: filters and wrappers. Filters employ heuristics for selecting representative data, often by removing outliers or noisy examples, while wrappers use the model to identify thresholds for retaining high quality data with little reduction in performance. In general, wrapper methods for neural networks employ rules based on the loss incurred per example [47] and have been limited to smaller data than our scenario. Our method may be viewed as wrapper-based instance selection. Since it is based on Shapley values, it offers a principled approach to measure the relative contribution of examples to a model's performance. This has potential for broader applications than dataset reduction: a complete ranking of the data allows for addressing data quality issues in the collection and annotation pipeline, in addition to curating training data for high performance models.

III. Methods

Environmental sound classification is a machine listening task which requires identifying which sound classes are present in an audio clip. This is a multi-label classification task since environmental sound recordings can contain multiple, potentially overlapping, sounds. Our audio clips are from the EDANSA-2019 dataset [23] and each clip may be labeled with sound events from

TABLE I
NUMBER OF CLIPS PER LABEL IN EACH DIVISION OF THE DATASET

Label	Train	Validation	Test	Total
Biophony	1729	472	490	2691
Bird	1461	420	470	2351
Songbird	492	98	84	674
Waterfowl	158	48	78	284
Grouse	93	29	6	128
Insect	222	25	6	253
Anthrophony	162	64	58	284
Aircraft	44	54	22	120
Silence	17	22	14	53
Total	4378	854	1228	6460

up to 9 sound classes (Table I). Our investigations employ a convolutional neural network (CNN) to perform multi-label sound classification. The performance within each class is scored using AUC, the area under the ROC curve [13]. Since the classifier will be used for identifying and tracking multiple events of interest, we report performance of the multi-label classifier using the macro-averaged AUC score, the mean of per class AUC scores. This ensures that the classifier's evaluation is not skewed by class imbalance (micro-averaging, or weighting the per class scores by class size, would give prominence to performance within majority classes and neglect performance within minority classes) [48].

We address the limited amount of labeled data for our audio classification task by employing data augmentation methods. We evaluate the quality of our multi-labeled data, which may indicate the presence of multiple events, with Shapley values.

A. Data Augmentation for Audio Classification

We use Mixup and SpecAugment for data augmentation. The original Mixup approach combines two randomly selected samples (x_i, y_i) and (x_j, y_j) from training data linearly. We modify this slightly, such that the data points are still combined linearly, but their labels are logically OR-ed:

$$\tilde{x} = \gamma x_i + (1-\gamma)x_j$$
$$\tilde{y} = \max(y_i, y_j). \quad (2)$$

$\gamma \in [0,1]$ is typically chosen by sampling from a beta distribution $Beta(\alpha, \alpha)$ for $\alpha \in (0, \infty)$, but we fix it at 0.5 for simplicity. Our modification to the label combination captures the fact that a linear combination of two sounds contains all of the sounds in either mixture, unlike linear combinations of images, in which partially transparent objects may not fully represent their original class. We select the samples randomly in a manner that ensures an equal probability of selection from all classes. This strategy is instrumental for a balanced dataset, thereby reducing the potential for bias in our results.

SpecAugment works by masking a set of consecutive frequency channels and/or time frames of the log-mel spectrogram, with the option to apply time warping as well.

B. Data Valuation

1) Hybrid MC-Proxy Method: For our model and task, we use a novel hybrid approach: learn a proxy KNN model for the deep CNN classifier, estimate the Shapley values for that proxy model using a Monte-Carlo (MC) algorithm, and use these "proxy" values as estimated Shapley values for the CNN model. In contrast with the proxy method employing KNN-Shapley algorithm [32], our method proposes to estimate the Shapley values for the proxy KNN itself. To our knowledge, our initial experiments [7] were the first application of this method. A priori, it is not evident that these estimated values will serve as good measures of data importance for the original DNN model. Thus, we validate this with empirical results.

To estimate the Shapley values for the multi-label proxy KNN, in each iteration of the MC algorithm, we measure the proxy model's AUC score for a random training subset and its AUC score when an example is added to that subset. The difference in scores is the marginal contribution of that example to a random subset. This process can be repeated several times until all the Shapley values have converged. In practice, we use an equivalent and computationally efficient sampling method, by sampling a random permutation of the training data, then scanning the permutation to determine the random example and subsets [31]. For an example being scanned, the preceding examples in the list constitute a random subset, and the change in performance with and without the example is a marginal contribution statistic for that example. Each iteration of the procedure yields one marginal contribution statistic per training example. The TMC algorithm [12] uses an additional heuristic for early stopping: as a training subset becomes larger, the changes in performance diminish. We stop scanning the permutation once the change falls below a tolerance level.

Since the MC approach is agnostic to both the model and performance metric, it can be used with a multi-label KNN classifier scored using AUC. Although MC methods require retraining the model in every iteration, this approach is still feasible for our scenario because retraining a KNN is much faster than retraining a CNN. To our knowledge, we are the first to use the MC approach with proxy KNN model to estimate Shapley values for a CNN. The inherent limitation of the MC-based approach is that while it is feasible for our data ($\mathcal{O}(10^3)$ examples), it will not scale to much larger data (e.g.,$\mathcal{O}(10^6)$ examples). We show that the resulting Shapley values are good estimates for the proxy model. We conduct experiments to determine if the values are also indicative of example importance for the actual CNN model.

IV. DATA AND EXPERIMENTS

A. Data

The data utilized in this study originates from the EDANSA-2019 dataset, a comprehensive ecoacoustic collection we gathered and published in a separate scholarly work [23]. The dataset encompasses sound recordings from the North Slope of Alaska and neighboring areas over the summer of 2019. Our partners placed recording devices at 100 sites throughout an area of

9000 square miles in the Prudhoe Bay region, the 10-02 area of the Arctic National Wildlife Refuge, and the Ivvavik National Park along with two 400-mile latitudinal transects along the Dalton and Dempster roads. From May to August, each recorder collected data in 150-minute segments, separated by 120-minute gaps, totaling 50,000 hours of recordings.

We selected 34 sites from these locations seeking a diverse range of acoustic sources based on domain knowledge of their acoustic characteristics. From each site, one 75-minute excerpt was randomly chosen across the recording season from those uncontaminated with an undue amount of audio clipping. An expert analyst inspected the spectrograms of these excerpts to identify all non-background sound events, which were then labeled based on listening. Here, background sounds are those not generated by humans or any type of animal, e.g.,wind and rain. The annotated segments ranged from a few seconds to a few minutes. The labeled segments were split into non-overlapping 10-second clips. A total of 97 samples that were shorter than 2 seconds were discarded, based on the consideration that shorter samples may not provide sufficient information for reliable classification of multiple overlapping sounds. To split samples into segments, we assigned all labels for the duration of the original annotated clips to the fixed-length segments used for model inputs.

All recordings were sampled at 48 kHz and collected in stereo. The audio includes noise due to wind and rain and some data is lost due to clipping when the sound becomes louder than the recording device's dynamic range. Rather than averaging both channels of the audio, we select the channel with less clipping for each 10 s clip. We take clipped samples to be those with the maximum or minimum integer value. We calculate the clipping rate by dividing the number of clipped samples in a clip by the total number of samples.

Our annotator created a taxonomy of the sounds present in the recordings, shown in Table I. We refer to the three taxonomic ranks in the dataset as coarse, medium, and fine. The coarse level consists of biophony, anthrophony, and silence; the medium level consists of bird, aircraft, and insect; and the fine level consists of songbird, waterfowl (which includes ducks, geese, and swans), and grouse (which includes different species of grouse and ptarmigan). All clips annotated with a child label in the hierarchy are also labeled with the parent label, although some clips are only annotated with coarse or medium labels. The dataset used for our experiments include samples with the following labels: biophony, bird, songbird, waterfowl, grouse, insect, anthrophony, aircraft, and silence.

The parent classes in our taxonomy encompass a wide variety of sound types. For instance, the anthrophony class includes sounds such as cars and flares from oil rigs, among others. However, due to the limited number of samples for some of these, they were not included as separate labels for training the model. It is important to note that the total number of labels (6,460) is larger than the total number of clips (3,083) because each clip can contain sounds from multiple sources, and thus, can be associated with multiple labels.

For better generalization, we ensure that data from each recording site occur in only one of the training, validation, or test sets. We formulate a multiple knapsack problem where sites are items, weights are the number of samples per site, and knapsacks are the training, validation, and test sets. Using Google OR-Tools [49], we determine optimal solutions per class, picking the solution with the lowest total cost over all classes. Validation and test knapsacks are constrained to be identically sized at 10-20%. The solution score is found by summing the Jensen-Shannon divergence between set distributions and the 60%-20%-20% target distribution per label.

It is important to note that due to the unique distribution of certain classes, such as aircraft and silence, the splitting scheme does not strictly adheres to the 60%-20%-20% target distribution. These classes are predominantly found in specific locations, such as near airports for the aircraft class. As we ensure that data from each recording site only occurs in one of the training, validation, or test sets, it is not always possible to maintain the target distribution for all classes. This is reflected in Table I, where the membership of these classes in the validation set makes up 40-50% of the class samples. This approach was chosen to prioritize the diversity of locations in each set, which is crucial for the generalization of our model.

B. CNN Baseline

For these experiments, we chose not to conduct a parameter or architecture search. Instead we employed a CNN based architecture [50]. The inputs to our model consist of mel spectrograms derived from the 10 s clips. Specifically, we extracted log-scale mel frequency spectrograms utilizing a window size of 42 ms, a hop size of 23 ms, and 128 mel frequency bins. Our model, which is trained entirely from scratch, is composed of 4 convolutional layers with a kernel size of 5×5, succeeded by 2 fully connected layers, following other successful architectures in sound event detection [51].

Moreover, subsequent to the final convolutional layer we conducted a comparative analysis between the use of a global max pooling operation over the time dimension to the averaging of the predictions over time after the softmax [3]. We refer to this global pooling in our experiments as GPool (cf. Table III). The CNN was trained over 1500 epochs, utilizing a learning rate of 0.001 and a batch size of 32. The model from the epoch that yielded the highest minimum AUC across labels on the validation set was chosen for evaluation on the test set. This approach was adopted to optimize the performance of the worst label, thereby ensuring a consistent level of performance across all labels.

C. Data Valuation

We represent examples by extracting 512-dimensional neural features from the penultimate fully connected layer, i.e.,before the softmax output layer of the CNN with the best validation performance and use the 9 labels described above as targets. We use the validation set to tune a multi-label KNN model using scikit-learn [52] and determine $k = 29$. The final proxy KNN model achieves AUC scores of 0.812 and 0.676 on the validation and test sets, respectively. We use the TMC algorithm [12] to estimate Shapley values of training examples for the KNN proxy model on the validation set. We sample one permutation per iteration to determine value updates, and perform 1,000 iterations

TABLE II
VALIDATION SET PERFORMANCE WITH DIFFERENT DATA AUGMENTATION METHODS (WITH AND WITHOUT GLOBAL POOLING)

GPool	Model Augmentation	Biophony	Bird	Songbird	Waterfowl	Anthrophony	Insect	Grouse	Aircraft	Silence	AUC
OFF	-	0.75	0.83	0.80	0.82	0.9	**0.97**	0.9	0.92	0.88	0.86
	mixup	0.84	0.82	0.74	**0.92**	0.89	0.97	0.91	0.9	0.92	0.88
	mixup+SpecAugment	0.84	0.82	0.78	0.88	0.9	0.97	**0.97**	**0.96**	0.81	0.88
	SpecAugment	0.77	0.82	0.80	0.87	0.89	0.92	0.93	0.82	0.82	0.85
ON	-	0.91	0.89	**0.84**	0.87	0.86	0.92	0.89	0.89	0.98	0.89
	mixup	0.86	0.87	0.78	0.89	0.91	0.95	0.94	0.9	0.93	0.89
	mixup+SpecAugment	0.89	0.85	0.76	0.9	**0.96**	0.83	0.79	0.92	0.95	0.87
	SpecAugment	**0.92**	**0.9**	0.80	0.85	0.91	0.92	0.95	0.93	**0.99**	**0.91**

The class columns report the per class AUC score. The last column reports the macro-averaged AUC score. Best scores are indicated in bold.

TABLE III
TEST SET PERFORMANCE WITH DIFFERENT DATA AUGMENTATION METHODS

GPool	Model Augmentation	Biophony	Bird	Songbird	Waterfowl	Anthrophony	Insect	Grouse	Aircraft	Silence	AUC
OFF	-	0.66	0.76	0.66	0.70	0.77	*0.74*	0.77	0.78	0.79	0.73
	mixup	0.74	0.77	0.62	*0.87*	0.77	0.78	0.80	0.85	0.88	0.78
	mixup+SpecAugment	0.72	0.77	0.56	0.84	0.83	0.92	*0.67*	*0.86*	0.63	0.76
	SpecAugment	0.67	0.73	0.60	0.76	0.82	0.79	0.88	0.77	0.73	0.75
ON	-	0.84	0.79	*0.59*	0.81	0.86	0.81	0.65	0.88	0.92	0.79
	mixup	0.80	0.77	0.64	0.85	0.91	0.95	0.89	0.95	0.91	0.85
	mixup+SpecAugment	0.77	0.76	0.61	0.89	*0.88*	0.72	0.80	0.95	0.94	0.81
	SpecAugment	*0.82*	*0.80*	0.77	0.83	0.84	0.83	0.76	0.84	*0.94*	*0.82*

The class columns report the per class AUC score. The last column reports the macro-averaged AUC score. Corresponding values to best results in validation set are indicated in italic.

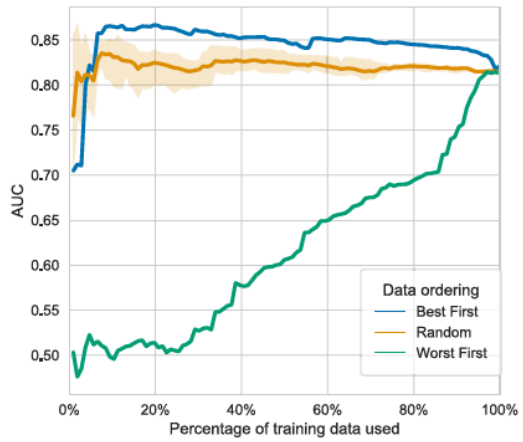

Fig. 1. Test set performance of multi-label KNN, with macro-averaged AUC over all classes, as training data is added in order determined by Shapley values. The Random curve depicts mean performance from three runs with a 95% bootstrapped confidence interval.

per "round." We repeat the procedure until convergence with a tolerance of 0.05.

We evaluate the approximated Shapley values by examining the KNN model's performance on held out test examples by incrementally adding the lowest- or highest-valued training examples to the KNN's training set, as is typical in the data valuation literature [12]. We also measure the model's performance when adding examples at random and record results from three random runs for comparison with the ordering determined by Shapley values. The results are shown in Fig. 1 and discussed in Section V.

Since our Shapley values are estimates based on a proxy KNN model, we investigate their ultimate utility for the original CNN model. We order the data by Shapley values in descending order ("best first") and gradually add these examples to a training subset for a CNN. In each evaluation step, the CNN is trained in the same way as the CNN used for the complete training data set. The CNN model settings are selected based on the model with the highest AUC score on the validation set from the previous experiments (see Table II). The final Shapley evaluation results are then reported by measuring the CNN's performance on the test set. For computational reasons, we evaluate the CNN using 7 training subsets based on the top 7%, 10%, 15%, 20%, 40%, 60%, and 80% fractions of the training data based on their Shapley values. For a baseline, we perform the same procedure but with randomly ordered data. We randomly sample batches with 10%, 20%, 40%, and 80% of the training data and measure the performance scores. This is repeated three times at each point and we use the average as the baseline score. Fig. 2 shows the evaluation results for the CNN.

V. RESULTS AND DISCUSSION

A. Data Augmentation Techniques

Tables II and III show the average AUC across classes on the validation and test sets. We report test set results only to assess generalization performance and avoid their use for any decision making in our experiments. We also show the performance with different model settings. The results on the validation set show that systems using mel spectrogram features and global temporal pooling consistently outperform the alternatives. We hypothesize that even though the information per frame is condensed before reaching the FC layer through global pooling, the FC

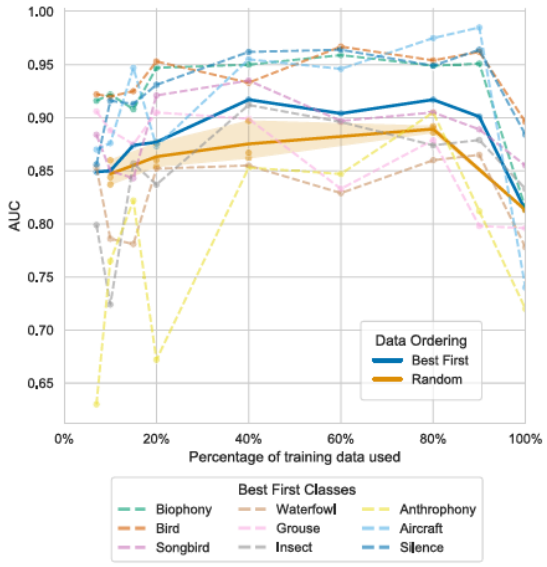

Fig. 2. Test set performance of the multi-label CNN classifier when examples are added to training set in a specific order. The Best First curve depicts performance as examples are ordered from highest to lowest Shapley values. The dotted lines show the corresponding per-class scores. The Random curve depicts mean performance from three runs when examples are added in random order, with a 95% bootstrapped confidence interval.

TABLE IV
AUC PER LABEL OF THE BEST MODEL ON THE VALIDATION AND TEST SETS ALONG WITH THE GENERALIZATION GAP BETWEEN THEM (Δ)

Label	Val.	Test	Δ
Biophony	0.92	0.82	0.10
Bird	0.90	0.80	0.10
Songbird	0.80	0.77	0.03
Waterfowl	0.85	0.83	0.02
Grouse	0.95	0.76	0.19
Insect	0.92	0.83	0.09
Anthrophony	0.91	0.84	0.07
Aircraft	0.93	0.84	0.09
Silence	0.99	0.94	0.05
Average	0.91	0.82	0.09

layer can still process data over time. However, without global pooling, averaging is applied after the softmax function, and the FC layer's perspective is limited to shorter time frames. In terms of augmentations, results are less consistent, although the best validation performance is achieved by SpecAugment alone. This best system achieves a relative improvement (in 1 − AUC) of 37% on the validation set and 33% on the test set over the baseline system using no global pooling or augmentation. Considering class-wise performance, we find that SpecAugment tends to improve performance in classes with more examples, while Mixup tends to improve performance more in classes with fewer examples. Given SpecAugment's superior performance on the validation set, we selected it for further investigations.

Table IV shows the AUC per label of the single best model, using mel spectrogram features, global temporal pooling, and SpecAugment only. For this model, the "songbird" label has the lowest AUC on the validation set and one of two lowest on the test set. However, the small generalization gap between the validation and test sets for "songbird" suggests that it is a difficult class to learn. This motivates our Shapley value analysis of this class below. Except for the biophony label, the use of data augmentation results in higher scores. The Mixup data augmentation method results in more best performing labels overall than the other data augmentation methods.

B. Data Valuation Techniques

1) Evaluation of Shapley Values With the Proxy KNN Model: We evaluate the estimated Shapley values using the method described in Section IV-C. Fig. 1 shows test set scores of average AUC across labels for the multi-label KNN classifier as we add training data in batches of size 32 ordered by the Shapley values computed from the validation set. We observe that the performance using randomly ordered data, averaged over three runs, tends to stay relatively constant, with minor fluctuations. The random ordering of data produces models that underperform the best-first ordering and overperform the worst-first ordering of data. Thus, the Shapley values determine a ranking of the data which is indicative of their utility for the model's performance.

The worst-first curve shows performance degradation as the lowest valued examples are used for training. As more examples are added, the resulting models show little to slight improvement up to the 20% point. Subsequently, the performance begins to improve until about 85% of the examples are added. On adding the last 15% of examples, i.e.,the highest valued examples, there is a sharp rise in performance.

The best-first curve shows a steep increase in performance as the highest valued examples are used for training. A model trained with about 20% of the training data, using the highest valued examples, achieves the highest AUC of about 0.867. Using the entire training data, we achieve an AUC of 0.816; thus, we can obtain a 28% relative improvement (in 1 − AUC) by excluding 80% of the data. After the 20% point, the performance declines very gradually. On adding the lowest valued examples (about 5% of the training data), there is a steeper decline in performance.

The lowest Shapley values are effective for identifying the lowest quality data that are likely misleading the model or especially difficult for learning. These examples can either be discarded or analyzed to identify annotation errors or other data quality issues (see Section V-B-3). In addition, the highest Shapley values effectively identify the highest quality subset of training data for learning the best-performing classifier. These examples are likely conveying the most information required for a high performing classifier. Finally, a large portion of the data may be conveying redundant information. This is suggested by the relatively constant random curve and the gradual performance change in the middle regions of the other two curves. This demonstrates that the estimated Shapley values capture the relative contribution of each example to the classifier's performance. This is especially true for the examples with the lowest and highest values.

2) Evaluation With the CNN Model: While the preceding results are encouraging, we are primarily interested in how well

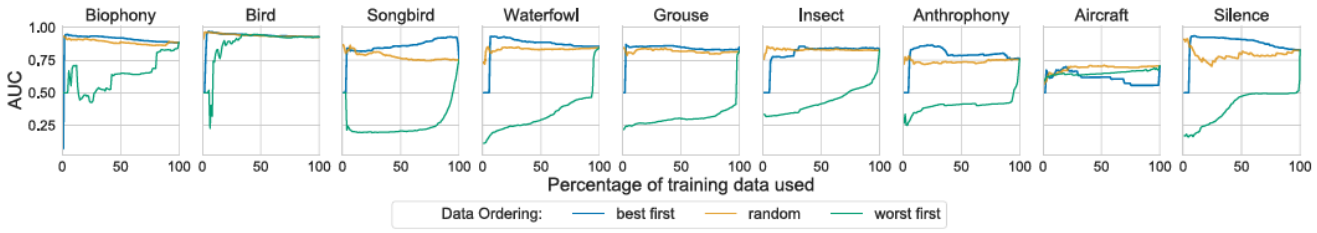

Fig. 3. Test set performance when we evaluate the Shapley values for each proxy binary classifier as training data is added in the order determined by the Shapley values. The random curves depict mean performance from three runs.

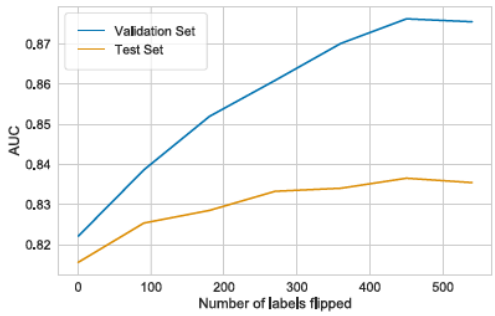

Fig. 4. Performance curve for multi-label KNN classifier when we flip the labels for worst valued examples. We use the Shapley values from a binary KNN classifier to determine which labels to flip.

they apply to the CNN classifier. We repeat the evaluation experiments using the CNN model for the best-first and random cases and obtain the results in Fig. 2. We show the overall model performance using the macro-averaged AUC over all classes ("Mean" curve). In addition, we show performance within classes using the per-class AUC scores. The highest performing model with an AUC of 0.917 is obtained when training on a subset with the top 40% of Shapley values. Using the entire training set, we can achieve an AUC of 0.813. Thus, we can obtain a relative improvement (in $1 - \text{AUC}$) of 56% indicating that we can build the CNN classifier using as little as 40% of the training data.

In the previous evaluation, which used the proxy KNN model, roughly 20% of the training data were sufficient for achieving the highest performance. In comparison, the neural network requires at least twice as much data (40%). We suspect that the main reason is that the neural network implicitly learns a representation space while learning the classification boundary for the multi-label classification task. In contrast, the KNN model begins with the neural features as inputs and learns only the classification boundary. These neural features are the embeddings of the examples in the representation space learned by the neural network. Since our Shapley estimation scheme uses the proxy KNN model, it can only learn the importance of the (embedded) examples for learning the decision boundary, but not their importance for learning the representation itself. This suggests that, for our task, the neural network may require at least twice as much data for representation learning alongside classifier learning.

The Shapley values estimated through the proxy KNN model are particularly effective for identifying lowest-valued examples. This is seen by the steep drop in performance at the tail end of the evaluation curve when the bottom 20% of the data are included for training. We suspect these examples are either misleading or difficult to learn, potentially due to incorrect annotations. Simply excluding the bottom 20% of the examples yields the best classifier with an AUC of 0.917.

While the neural net takes at least 10 hours (and up to 68 hours) to train, the KNN takes 208 ms, on average, for training. The Shapley values estimated from the proxy KNN can also measure example utility for the original CNN model (as shown in Fig. 2). This justifies our use of a proxy nearest neighbors model for efficient and effective estimation of Shapley values for a multi-label neural network classifier.

3) Analysis of Low Shapley Values: In our dataset, 76% of the examples are labeled as birds and 26% are labeled as songbirds. The proportion of songbirds appears quite low and we suspect that a number of songbird examples were missed during annotation. We believe that the classifier is underperforming partly due to missing annotations. The Shapley values estimated from the proxy multi-label classifier capture the utility of data examples for the original CNN model. Since each example has 9 labels, it is difficult to examine exactly why points might have low Shapley values. To drill into class-specific issues and perform further analysis, we train a proxy binary KNN classifier per class and determine the Shapley values of the training examples for the binary classifiers on the validation set. Then, we evaluate the utility of the Shapley values on the test set by adding training data in the order determined by the Shapley values for each binary classifier.

Fig. 3 shows the performance evaluation curves per class. In particular, we see that the songbirds evaluation curve follows a similar trend as the multi-label evaluation curve. The best-first curve improves steadily until performance degrades sharply given roughly the bottom 5% of the data. Upon listening to several of the worst-valued examples, we note a number of clips have songbirds present, but are missing the songbird label. This confirms that some examples are actively misleading for the songbird classifier. We compare the labels with the ground truth labels (i.e.,from the re-annotated training set) and find that 30% (328) of the examples with negative Shapley values have annotation errors. In comparison, only 4% (124) of the positive valued examples have annotation errors. This significant portion of examples with annotation errors can partly explain why the model's performance improves when these examples are excluded from the training set in Fig. 3.

We further investigate using Shapley values to automatically adjust labels to see if model performance can be improved without inspecting individual examples. We want to see if simply flipping labels (i.e.,changing 0 s to 1 s, and 1 s to 0 s) for

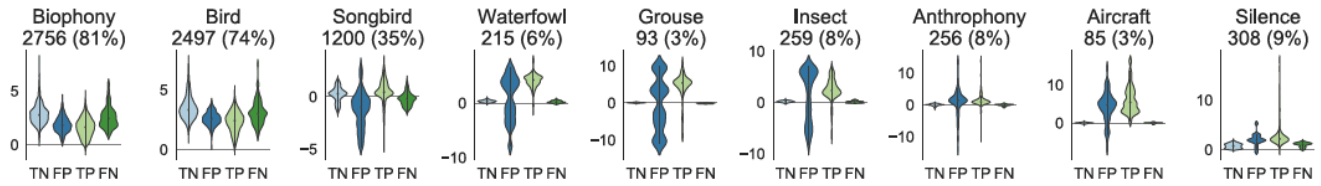

Fig. 5. Distribution of Shapley values, for binary classifiers, grouped by their annotation status as True Negatives (TN), False Positives (FP), True Positives (TP), and False Negatives (FN) between the initial and secondary labeling processes. The FN and FP status means that these examples were initially annotated incorrectly as negative or positive, respectively. The vertical axis shows the standardized Shapley values. Subplot titles also show the number and proportion of clips with this label. To improve the visualization of minority cases, the distributions have been scaled to have the same width.

the worst-valued examples can improve performance. Since Shapley values estimated for the multi-label classifier cannot inform on which label to adjust for an example, we use the Shapley values estimated for the binary classifiers to perform this experiment. However, we measure the effect of adjusting those labels on the original multi-label classification problem. First, we train a proxy multi-label KNN classifier using all of the examples and measure its performance. For each class, we create a list of the training examples indices, ordered by the Shapley values estimated from that class's proxy binary KNN classifier (learned in the earlier experiment). This yields 9 orderings of the training examples, one per class. We perform a number of iterations, flipping 9 labels (one label per class) in each iteration: for each class, we always remove the lowest valued example's index from that class's list, then flip the corresponding class label for that example in the training set. After every 10 iterations, we re-train the proxy multi-class KNN classifier with the training set that contains updated labels and record the performance on a held-out validation set. We continue iterating until we see a drop in the performance for two successive iterations. The results are shown in Fig. 4 along with the corresponding test set performance. From the validation set curve, we see that the performance increases until we have flipped 450 labels (i.e.,50 iterations) and then begins to degrade. The final performance on the test set is measured with AUC of 0.837, an 11.4% relative improvement in (1-AUC) over the performance on the original training data. This shows that we can use Shapley values to determine the likelihood of annotation errors among lowest-valued data using the proxy KNN models. It also suggests that it is possible to use Shapley values to automatically adjust labels and improve model performance using a simple heuristic.

Fig. 5 provides more details on the distribution of Shapley values per sub-components based on their annotation status when compared to the cleaned (re-annotated) training set. Overall, Shapley values for False Positives tend to have larger variances, but also more negative values. This is particularly true for the songbirds class. False Negatives also tend to skew towards lower values for minority classes. For a high majority class like biophony (81% positive examples), very few examples have negative Shapley values and thus, few examples hurt the model's performance within that class. This is likely because the classifier has an abundance of positive examples to learn a good decision boundary and there is little confusion with other classes. For a highly minority class like aircraft (3% positive examples), the lowest and negative-valued examples tend to be false positives (examples that were incorrectly labeled as aircraft). This makes sense: due to the scarcity of positive examples, the classifier struggles to learn the boundary when positive examples are mislabeled.

Considering the more balanced songbirds class, we see that the lowest quality and negatively valued examples tend to be false positives (i.e.,examples mislabeled as songbirds). From these false positive annotations, we consider examples having Shapley values more than two standard deviations below zero, and select ten random examples. After listening to these, we find that two clips contain bird sounds mislabeled as songbirds. We also note from the songbird distribution that a portion of true negative examples (i.e.,correctly labeled as not having songbirds) have negative Shapley values. We select ten random examples for listening and find one example that has songbirds present. This suggests that some examples may have been missed or annotated incorrectly in the second round of annotation. Thus, we are still able to identify annotation errors by concentrating on the negatively valued examples. We also see that the false negatives (examples incorrectly labeled as lacking songbirds) have very low Shapley values, close to zero. These are likely examples that would make higher contributions to the classifier if they were labeled correctly. Our analysis verifies that analyzing Shapley values for binary classifiers can help with analysis of model errors and assessment of data quality issues.

VI. Conclusion and Future Work

We validate a novel Shapley value estimation method for a multi-label classifier using AUC as the performance metric, showcasing its effectiveness for the original CNN classifier. We demonstrate that this approach, of valuing and curating limited labeled data, complements the data-augmentation method that artificially increase the amount of training data. We also present methods for assessing data quality issues, particularly annotation errors, through analysis based on Shapley values. Specifically, we show how to drill down in multi-label data and identify data quality issues within sound classes. Although our method does not scale to very large data, we show that it is efficient and effective for limited labeled scenarios common in ecoacoustic research. Taken together, our methods provide a path for ecoacoustic researchers to make maximal use of scarce labels, both by curation of high quality data, and by identification of data quality issues that may arise from annotation or collection processes.

Acknowledgment

We are grateful to Megan Perra, Scott Leorna, Dr. Todd Brinkman and Dr. Natalie T. Boelman for data collection, and Megan Perra for annotation.

REFERENCES

[1] J. Richter-Menge, M. Jeffries, and E. Osborne, "The Arctic," *Bull. Amer. Meteorological Soc.: State Climate in 2018*, vol. 99, no. 8, pp. 141–168, 2018.

[2] S. Christin, É. Hervet, and N. Lecomte, "Applications for deep learning in ecology," *Methods Ecol. Evol.*, vol. 10, no. 10, pp. 1632–1644, 2019.

[3] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2880–2894, 2020.

[4] M. Cartwright et al., "Seeing sound: Investigating the effects of visualizations and complexity on crowdsourced audio annotations," *Proc. ACM Hum.-Comput. Interaction*, vol. 1, 2017, Art. no. 29.

[5] M. Pichler and F. Hartig, "Machine learning and deep learning—A review for ecologists," *Methods Ecol. Evol.*, vol. 14, no. 4, pp. 994–1016, 2023.

[6] M. Mazumder et al., "Dataperf: Benchmarks for data-centric AI development," *NeurIPS Datasets Benchmarks Track*, 2023.

[7] E. B. Çoban, A. R. Syed, D. Pir, and M. I. Mandel, "Towards large scale ecoacoustic monitoring with small amounts of labeled data," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2021, pp. 181–185.

[8] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015, pp. 3586–3589.

[9] H. Zhang, M. Cisse, Y. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–13.

[10] D. S. Park et al., "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[11] L. S. Shapley, "A value for n-person games," *Contributions Theory Games*, vol. 2, no. 28, pp. 31–40, 1953.

[12] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2242–2251.

[13] J. Fan, S. Upadhye, and A. Worster, "Understanding ROC curves," *Can. J. Emerg. Med.*, vol. 8, no. 1, pp. 19–20, 2006.

[14] C. Northcutt, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," *NeurIPS Datasets Benchmarks Track*, 2021.

[15] I. Martín-Morató et al., "Low-complexity acoustic scene classification in dcase 2022 challenge," in *Proc. Workshop Detection Classification Acoustic Scenes Events*, 2022, pp. 111–115.

[16] I. Nolasco et al., "Few-shot bioacoustic event detection at the dcase 2022 challenge," in *Proc. Workshop Detection Classification Acoustic Scenes Events*, 2022, pp. 136–140.

[17] T. Nguyen and F. Pernkopf, "Acoustic scene classification with mismatched devices using cliquenets and mixup data augmentation," in *Proc. Interspeech*, 2019, pp. 2330–2334.

[18] T. Iqbal, Q. Kong, M. Plumbley, and W. Wang, "Stacked convolutional neural networks for general-purpose audio tagging," in *Proc. Workshop Detection Classification Acoustic Scenes Events*, 2018, pp. 1–4.

[19] I. Jeong and H. Lim, "Audio tagging system for dcase 2018: Focusing on label noise, data augmentation and its efficient learning," in *Proc. Workshop Detection Classification Acoustic Scenes Events*, 2018, pp. 1–4.

[20] S. Wei, K. Xu, D. Wang, F. Liao, H. Wang, and Q. Kong, "Sample mixed-based data augmentation for domestic audio tagging," in *Proc. Workshop Detection Classification Acoustic Scenes Events*, 2018, pp. 93–97.

[21] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Conformer-based sound event detection with semi-supervised learning and data augmentation," in *Proc. Workshop Detection Classification Acoustic Scenes Events*, 2020, pp. 100–104.

[22] R. Raskar, P. Vepakomma, T. Swedish, and A. Sharan, "Data markets to support ai for all," 2019, *arXiv:1905.06462*.

[23] E. B. Çoban, M. Perra, D. Pir, and M. I. Mandel, "EDANSA-2019: The ecoacoustic dataset from Arctic North Slope Alaska," in *Proc. Workshop Detection Classification Acoustic Scenes Events*, 2022, pp. 16–20.

[24] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1885–1894.

[25] F. Hampel, "The influence curve and its role in robust estimation," *J. Amer. Stat. Assoc.*, vol. 69, no. 346, pp. 383–393, 1974.

[26] D. M. Allen, "Mean square error of prediction as a criterion for selecting variables," *Technometrics*, vol. 13, no. 3, pp. 469–475, 1971.

[27] R. Jia et al., "Towards efficient data valuation based on the shapley value," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1167–1176.

[28] J. Kleinberg, C. Papadimitriou, and P. Raghavan, "On the value of private information," in *Proc. 8th Conf. Theor. Aspects Rationality Knowl.*, 2001, pp. 249–257.

[29] A. R. Syed and M. I. Mandel, "Estimating shapley values of training utterances for automatic speech recognition models," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.

[30] C. Molnar, G. Casalicchio, and B. Bischl, "Interpretable machine learning," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discov. Databases*, 2020, pp. 417–431.

[31] J. Castro, D. Gomez, and J. Tejada, "Polynomial calculation of the shapley value based on sampling," *Comput. Operations Res.*, vol. 36, no. 5, pp. 1726–1730, 2009.

[32] R. Jia, X. Sun, J. Xu, C. Zhang, B. Li, and D. Song, "An empirical and comparative analysis of data valuation with scalable algorithms," 2019, *arXiv:1911.07128*.

[33] C. A. Quinn et al., "Soundscape classification with convolutional neural networks reveals temporal and geographic patterns in ecoacoustic data," *Ecological Indicators*, vol. 138, 2022, Art. no. 108831.

[34] A. J. Fairbrass, M. Firman, C. Williams, G. J. Brostow, H. Titheridge, and K. E. Jones, "CityNet–deep learning tools for urban ecoacoustic assessment," *Methods Ecol. Evol.*, vol. 10, no. 2, 2019, pp. 186–197.

[35] M. Zhong et al., "Multispecies bioacoustic classification using transfer learning of deep convolutional neural networks with pseudo-labeling," *Appl. Acoust.*, vol. 166, 2020, Art. no. 107375.

[36] C. Bergler et al., "Deep representation learning for Orca call type classification," in *Proc. Int. Conf. Text Speech Dialogue*, 2019, pp. 274–286.

[37] J. Yan, Y. Song, L.-R. Dai, and I. McLoughlin, "Task-aware mean teacher method for large scale weakly labeled semi-supervised sound event detection," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 326–330.

[38] B. K. d. A. Afonso and L. Berton, "Analysis of label noise in graph-based semi-supervised learning," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, 2020, pp. 1127–1134.

[39] A. Jansen et al., "Coincidence, categorization, and consolidation: Learning to recognize sounds with minimal supervision," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 121–125.

[40] Y. Wang, A. E. M. Mendez, M. Cartwright, and J. P. Bello, "Active learning for efficient audio annotation and classification with a large amount of unlabeled data," in *Proc. Int. Conf. Acoust. Speech Signal Process*, 2019, pp. 880–884.

[41] Z. Shuyang, T. Heittola, and T. Virtanen, "Active learning for sound event detection," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2895–2905, 2020.

[42] Y. Wang, M. Cartwright, and J. P. Bello, "Active few-shot learning for sound event detection," in *Proc. Interspeech*, 2022, pp. 1551–1555.

[43] P. Eichinski, C. Alexander, P. Roe, S. Parsons, and S. Fuller, "A convolutional neural network bird species recognizer built from little data by iteratively training, detecting, and labeling," *Front. Ecol. Evol.*, vol. 10, 2022, Art. no. 810330.

[44] A. R. Syed, A. Rosenberg, and E. Kislal, "Supervised and unsupervised active learning for automatic speech recognition of low-resource languages," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2016, pp. 5320–5324.

[45] W. Ji, R. Wang, and J. Ma, "Dictionary-based active learning for sound event classification," *Multimedia Tools Appl.*, vol. 78, pp. 3831–3842, 2019.

[46] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and J. Kittler, "A review of instance selection methods," *AI Rev.*, vol. 34, pp. 133–143, 2010.

[47] M. Kordos, "Data selection for neural networks," *Schedae Informaticae*, vol. 25, pp. 153–164, 2016.

[48] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to Information Retrieval*, vol. 39. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[49] L. Perron, "Operations research and constraint programming at Google," in *Principles and Practice of Constraint Programming*, Berlin, Germany: Springer, 2011.

[50] K. Piczak, "Environmental sound classification with convolutional neural networks," in *Proc. IEEE 25th Int. Workshop Mach. Learn. Signal Process.*, 2015, pp. 1–6.

[51] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," *Int. Soc. Music Inf. Retrieval*, pp. 805–811, 2018.

[52] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.