TOWARDS LARGE SCALE ECOACOUSTIC MONITORING WITH SMALL AMOUNTS OF LABELED DATA

Enis Berk Çoban,[1,*] *Ali Raza Syed,*[1,*] *Dara Pir,*[2] *Michael I Mandel*[1,3]

[1] The Graduate Center, CUNY, New York, NY, USA {ecoban, asyed2}@gradcenter.cuny.edu
[2] Guttman Community College, CUNY, New York, NY, USA dpir@gradcenter.cuny.edu
[3] Brooklyn College, CUNY, Brooklyn, NY, USA mim@sci.brooklyn.cuny.edu

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 the activities of animals that migrate to these environments annually to reproduce. Acoustic sensors can monitor a wide area relatively cheaply, producing large amounts of data that need to be automatically analyzed. In such scenarios, only a small proportion of the recorded data can be labeled by hand, thus we explore two methods for utilizing labels more efficiently: self-supervised learning using wav2vec 2.0 and data valuation using k-nearest neighbors approximations to compute Shapley values. We confirm that data augmentation and global temporal pooling improve performance by more than 30%, demonstrate for the first time the utility of Shapley data valuation for audio classification, and find that our wav2vec 2.0 model trained from scratch does not improve performance.

Index Terms— Ecoacoustics, data augmentation, data valuation, self-supervised learning

1. INTRODUCTION

Arctic boreal forests have been warming at a rate 2-3 times that of the global average as well as experiencing intensification in human development [1]. These ecosystems are critical for the reproductive success of numerous species, such as songbirds, waterfowl, and caribou. The long-term monitoring of the affected ecosystem is therefore very important to understanding the effect of global climate change on these species and weighing possible interventions and mitigation strategies. The audio approach for autonomous monitoring of the environment to detect events of interest has advantages over other modes, such as lower power, lower data bandwidth, wider field of view, and longer range than visual sensors. The audio mode can also be used alongside others to boost recognition performances.

Classifying and categorizing ecoacoustic data, at rates generated by large sensor networks, requires automated procedures. Developing supervised models for event classification requires labeled data, which is costly and sometimes prohibitive. Furthermore, insufficient training data produces poor performance. This paper explores the use of several techniques to address these problems: self-supervised learning, data augmentation, and data valuation. We build upon previous work [2] by utilizing a much larger unlabeled dataset and several additional methods for mitigating this limitation.

Unlabeled data can be used in semi-supervised learning techniques to help boost classification performances in cases where there is a lack of sufficient labeled data. Wav2vec 2.0 (W2V) [3] is a self-supervised learning method that has been successfully used in speech recognition to achieve state-of-the-art results when trained on only 1% of the labeled data used by fully supervised systems. W2V learns audio representations by processing speech audio first followed by a stage that uses transcribed speech.

Data augmentation techniques are used for artificially, but realistically increasing the amount of training data [4]. Mixup [5] and SpecAugment [6] augmentation methods generate different views of the same data and help deep neural networks to generalize. Mixup was initially designed for computer vision tasks but has proven to be effective in sound event detection and used by top-performing teams in the DCASE 2018 [7]. It enabled a convolutional recurrent neural network based system to outperform baselines [8]. Mixup has also been used for creating new samples to even out the number of samples in an unbalanced dataset [9, 10]. SpecAugment was introduced for end-to-end speech recognition tasks and achieved state-of-the-art performance on the LibriSpeech 960 and and Switchboard 300h tasks. A Conformer-based system using mixup, SpecAugment, time-shifting, and noise augmentation achieved superior performance in sound event detection [11]. SpecAugment and mixup alone achieved the best results in the DCASE 2019 for the audio tagging task [12].

Data valuation quantifies the value of examples used in training a given supervised learning algorithm. The primary motivation to date has been to compensate individuals or vendors who provide examples for data markets (e.g., [13, 14]). Quantifying the value of data for a model can also, however, be used for data curation and selection since it provides a method for ranking and selecting a subset of the data for optimal model performance and faster training. Data valuation may also hold potential for guiding data collection efforts or developing heuristics for Active Learning [15, 16]. Audio data is relatively cheap to acquire but laborious and expensive to annotate, especially for tasks like acoustic detection where multiple sounds may be present in short clips. Thus we are interested in exploring the feasibility of data valuation methods for curating and selecting already labeled acoustic data that are particularly useful or misleading for models. In particular, we use the Shapley value, an idea arising from cooperative game theory [17], and recently proposed as a fair method of determining the utility of data examples for a given model [18, 19]. We report the results of experiments with data augmentation, W2V, and data valuation techniques.

2. METHODS

To address the limited labeled data in our audio classification task we employ self-supervised learning and data augmentation methods described in Subsections 2.1 and 2.2. To assess the quality of our multi-labeled data, potentially indicating the presence of multiple events, we use Shapley values described in Subsection 2.3.

[*] These authors contributed equally.

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

Table 1: Number of clips per label in each division of the dataset, of 3083 clips total. Waterfowl includes ducks, geese, and swans.

2.1. Self-supervised learning: wav2vec 2.0

Self-supervised learning methods take advantage of structure in the data, like word order in text. Similarly, W2V is trained by predicting speech units of deformed segments of input data. Raw waveforms of the speech audio are turned into latent audio representations of each 25ms segment using a convolutional neural network. Half of these embeddings are masked and then input into a transformer module. Transformers summarize the information from the sequence of embeddings into another sequence of feature vectors, which are quantized into targets in the self-supervised objective. The objective of W2V training is to predict the correct quantized speech units for masked positions. The self-supervised pre-trained model is fine-tuned on labeled data by removing the quantization module and using a fully connected layer after the context network.

2.2. Data augmentation for audio classification

We use mixup and SpecAugment for data augmentation, which have shown good performance in prior work on sound event detection tasks [12, 11]. 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 datapoints are still combined linearly, but their labels are logically OR-ed:

$$\begin{aligned} \tilde{x} &= \gamma x_i + (1-\gamma) x_j \\ \tilde{y} &= \max(y_i, y_j). \end{aligned} \tag{1}$$

In general, $\gamma \in [0, 1]$ can be chosen by sampling from a beta distribution $Beta(\alpha, \alpha)$ for $\alpha \in (0, \infty)$, but for simplicity we fix it at 0.5. Our modification to the label combination captures the fact that a linear combination of two sounds contains all of the sounds in either mixture. This is in contrast to linear combinations of images, in which it might be argued that partially transparent objects are not fully representative of their original class.

SpecAugment works by masking a set of consecutive frequency channels and/or time frames of the log-mel spectrogram. Time warping may be applied along with the masking deformation.

2.3. Shapley values

In cooperative game theory, a coalition of players collaborates toward a common goal to earn a reward. The Shapley valuation framework provides one method of fairly allocating rewards to individual players based on their relative contributions [17]. In the supervised machine learning setting, we think of 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.

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be the training data and $\mathcal{D}_{\text{eval}} = \{(x_j, y_j)\}_{j=1}^{N_{\text{eval}}}$ be the held-out data used for evaluation, the validation set in our experiments. Suppose a learning algorithm $\mathcal{A}$ trains on a subset $S \subseteq \mathcal{D}$ of training examples. The performance of the algorithm $\mathcal{A}$ using examples S is measured by an evaluation function $U_{\mathcal{A}}(S)$. The Shapley value $\sigma(x_i)$ of training example x_i is defined as the expected marginal contribution of x_i to *any* subset of the remaining training examples $S \subseteq D \setminus \{x_i\}$:

$$\sigma(x_i) = \frac{1}{N} \sum_{S \subseteq \mathcal{D} \setminus \{x_i\}} \frac{1}{\binom{N-1}{|S|}} \left[U_{\mathcal{A}}(S \cup \{x_i\}) - U_{\mathcal{A}}(S) \right] \tag{2}$$

The Shapley value is an allocation scheme that uniquely satisfies some rudimentary fairness axioms [17, 19]. This valuation approach makes no assumptions about the training data distribution $\mathcal{D}$ or whether the examples are independent or identically distributed.

In general, exact calculation of Shapley values is intractable for realistic dataset sizes. The Truncated Monte-Carlo (TMC) algorithm provides a method for estimating Shapley values for any model, $\mathcal{A}$ [18]. However, each iteration requires re-training the model, which can be time-consuming for a CNN. It was recently shown that Shapley values for deep neural network (DNN) classifiers can be approximated using proxy K-Nearest Neighbor (KNN) models [20]. We use the neural features from the CNN as inputs to learn a KNN classifier. Since training and evaluating a KNN is much faster than training a CNN from scratch, the TMC algorithm can be employed to estimate Shapley values of the inputs for the KNN proxy classifier. We use AUC as the performance metric for our valuation, $U_{\mathcal{A}}(S)$.

3. EXPERIMENTS

3.1. Data

Our data is comprised of sound recordings collected from the North Slope of Alaska and neighboring areas in Canada 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. Each recorder collected data from May through August in segments of 150 minutes separated by gaps of 120 minutes, collecting a total of 50,000 hours of recordings.

We selected 34 sites from these locations seeking a diverse set 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 not contaminated 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. The annotated segments ranged from a few seconds to a few minutes. These labeled segments were split into non-overlapping 10-second clips. All labeled segments were at least 2 seconds long and those shorter than 10 seconds were padded with zeros. Since segments can contain sounds from multiple sources of interest, our task is a multi-label classification problem.

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 10s 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 recordings, shown in Table 1. We refer to the three taxonomic ranks in the dataset as coarse, medium, and fine. The coarse level consists of anthrophony, biophony, and silence; the medium level consists of aircraft, bird, and insect; and the fine level consists of songbird, waterfowl, and grouse. 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 final dataset used for our experiments consists of 3083 samples.

For better generalization, we ensure that data from a recording site occurs 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 [21], 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 holding 10-20%. The solution score is found by summing the Jensen-Shannon divergence between set distributions and the 60%-20%-20% target distribution for each label.

3.2. CNN baseline

For these experiments, rather than perform a parameter or architecture search, we chose to use a CNN based architecture [22] similar to AlexNet [23]. The inputs to our model are mel-spectrograms of the 10 second clips. We extract log-scale mel-frequency spectrograms with a window size of 42 ms, a hop size of 23 ms, and 128 mel-frequency bins. Our model has 4 convolutional layers with a kernel size of 5 by 5, followed by two FC layers, following architectures previously found to be successful in sound event detection [24]. Moreover, after the last convolutional layer we compare the use of a global max pooling operation over the time dimension to the averaging of the predictions over time after the softmax [25]. The CNN is trained for 1500 epochs and the model from the epoch with the highest minimum AUC across labels is selected to evaluate on the test set. The performance of the worst label was optimized to ensure a consistent level of performance across labels.

Data augmentation Given the size of our dataset, neural networks will be prone to overfitting. We apply data augmentation and dropout to overcome this problem. We use two augmentation methods: a modified version of mixup [5] and SpecAugment [6]. For mixup, we compared picking samples at random with and without replacement and without replacement worked best on the validation set, so we report these results.

3.3. Self-supervised learning

The labeled clips make up a small part of our entire collection of recordings. A large amount of unlabeled data can be leveraged by using self-supervised learning techniques to generate condensed representations. W2V encodes inputs with convolutional layers in time domain and feeds them into a transformer to build a representation of the entire sequence. These representations are quantized before being used in the contrastive learning task (pretext-task). We use the smaller BASE model from the original paper, which has 12 transformer blocks, feed forward network with inner dimension of 3,072 and embedding dimension of 768, and 8 attention heads. The original experiments train the BASE model on 53k hours of speech recordings with 64 V100 GPUs for 1.6 days. The output of the BASE model for 10 seconds sample is 499 by 768 which is bigger than our log-mel spectogram input.

Input	Pooling	Augmentation	Valid	Test
Spect.	OFF	—	0.86	0.73
		mixup	0.88	0.78
		mixup+SpecAug	0.88	0.76
		SpecAug	0.85	0.75
	ON	—	0.89	0.79
		mixup	0.89	0.85
		mixup+SpecAug	0.87	0.81
		SpecAug	**0.91**	0.82
W2V	OFF	—	0.71	0.59
		mixup	0.79	0.63
		mixup+SpecAug	0.76	0.62
		SpecAug	0.75	0.58
	ON	—	0.77	0.66
		mixup	0.80	0.68
		mixup+SpecAug	0.81	0.67
		SpecAug	0.73	0.59

Table 2: Average AUC across labels for various settings of input features (Spect.: mel spectrogram, W2V: wav2vec 2.0 embeddings), global max pooling [25], and data augmentation.

We train our own version of the BASE model on 300k randomly sampled 10 second clips from our full dataset. We exclude samples that have a clipping rate higher than 1%, and are left with 234k samples, corresponding to 659 hours of recordings in total. Since our models were already overfitting log-mel spectograms, we reduce the feature embedding to 144 dimensions. The new model's parameter count is 16% of that of the original BASE model (96M). After training this model for 205 epochs (21 days), it achieved a 29% validation and 28% training accuracy, which is much lower than the 75% accuracy reported in the original paper.

3.4. 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 [26] 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 [18] 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 per "round." We repeat the procedure until convergence, after 10 rounds, 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. 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. For multi-label classification, we compute AUC using macro-averaged values, i.e., we average per class AUC scores to account for varying class sizes [27]. The results are shown in Figure 1 and discussed in Section 4.

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	0.92	0.84	0.08
Aircraft	0.93	0.84	0.09
Silence	0.99	0.94	0.05
Average	0.91	0.82	0.09

Table 3: AUC per label of the best model on the validation and test sets along with the generalization gap between them (Δ).

4. RESULTS AND DISCUSSION

Table 2 shows the average AUC across classes on the validation and test sets for the various model settings. The results on the validation set show that systems using mel spectrogram features and global temporal pooling consistently outperform the alternatives. 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 on augmentation. While we do not show detailed results per label due to space limitations, we find that mixup tends to improve performance more in classes with fewer examples, while for specAugment it is in classes with more examples.

Table 3 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.

Unfortunately, wav2vec 2.0 did not help us take advantage of this large amount of unlabeled data. This is potentially due to the small size of the subset of our data that we trained it on or the small number of updates with which we trained it, as these models perform best with huge amounts of data and compute.

The first plot in Figure 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. As we include the examples with highest Shapley values in the training set, the best-first curve quickly achieves the highest AUC of about 0.75 with about 20% of the data. This is well above the performance achieved using the same amount of data selected at random, which remains close to the performance of using all of the data, 0.68. As we add more examples, with lower Shapley values, we note that performance degrades down to this level. Similarly, the worst-first curve shows that as we train with the lowest-valued (worst) examples, the classifier performance does not improve much until we begin adding high-valued examples at the end. This demonstrates how well Shapley values capture the relative contribution of examples to the classifier's performance, and that performance can be improved by identifying and discarding certain examples that are actively misleading to the model.

Listening to the worst-valued examples, we note a number of clips have songbirds present, but are missing the "songbird" label. Labels for birds and songbirds comprise 76% and 26% of the examples, respectively. We suspect that our multi-label classifier may not be performing as well due to a substantial number of missed examples for songbirds. To examine this further, we train a binary KNN classifier to detect songbirds and determine the Shapley values of the training examples for the binary classifier as above. The second figure in Figure 1 shows the performance evaluation for Shapley values of examples used for training the songbird classifier. We see a similar trend for the best-first curve, but notice that the worst-first curve performs well then degrades after adding about 15-20% of the data. This confirms that some examples are actively misleading for the songbird classifier. Overall, we find 104 examples (5.4% of the data) with negative Shapley values, implying that these examples hurt model performance. About 69% (72) of these examples are labeled as not containing songbirds. To test our hypothesis on the original CNN model, we flip the songbird labels (0s to 1s, and 1s to 0s) of the 100 lowest valued examples and re-train the CNN. We find that this improves mean-AUC from 0.92 to 0.928 and min-AUC from 0.804 to 0.822. This lends further support for our hypothesis and demonstrates a potential application of Shapley values.

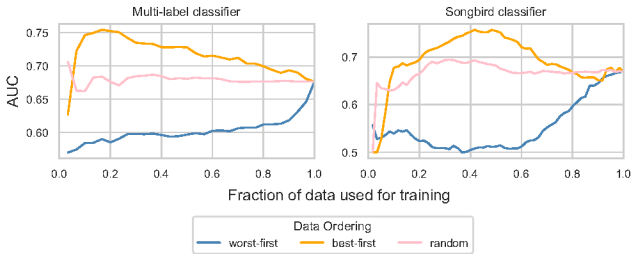

Figure 1: Test set performance of KNN as training data is added in order determined by Shapley values. The random curves depict mean performance from three runs. Left: Average AUC for multi-label classifier. Right: AUC for binary "songbird" classifier.

5. CONCLUSIONS AND FUTURE WORK

This paper has explored several approaches to improving classifier performance on large ecoacoustic datasets with small amounts of labeled data. We confirmed that data augmentation and global pooling can lead to significant improvements in performance and has demonstrated for the first time the utility of Shapley values in data valuation for audio classification. These techniques will be useful in managing our labeling efforts as well as analyzing large amounts of unlabeled data still remaining. Unfortunately, we did not observe any utility of wav2vec 2.0 for this task, but in the future we will explore the utility of training on more data and for longer. We also plan to explore improvements to the main classifier's performance once Shapley values have been computed using the KNN proxy as well as investigating the use of Shapley values to identify high quality *unlabeled* data for active learning.

6. ACKNOWLEDGMENTS

We are grateful to Megan Perra, Scott Leorna, Dr. Todd Brinkman and Dr. Natalie T. Boelman for data collection, and Megan Perra for annotation. This work is supported by the National Science Foundation (NSF) grant OPP-1839185. Any opinions, findings, and conclusions or recommendations are those of the author(s) and do not necessarily reflect the views of the NSF.

7. REFERENCES

[1] J. Richter-Menge, M. Jeffries, and E. Osborne, "The arctic," in *Bulletin of the American Meteorological Society: State of the Climate in 2018*, J. Blunden and D. S. Arndt, Eds., 2018, vol. 99, no. 8, ch. 5, pp. S143–S168.

[2] E. B. Çoban, D. Pir, R. So, and M. I. Mandel, "Transfer learning from youtube soundtracks to tag arctic ecoacoustic recordings," in *ICASSP*, 2020, pp. 726–730.

[3] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460.

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

[5] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[6] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[7] I.-Y. Jeong and H. Lim, "Audio tagging system for dcase 2018: focusing on label noise, data augmentation and its efficient learning," *DCASE Challenge*, 2018.

[8] S. Wei, K. Xu, D. Wang, F. Liao, H. Wang, and Q. Kong, "Sample mixed-based data augmentation for domestic audio tagging," in *DCASE*, November 2018, pp. 93–97.

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

[10] T. Iqbal, Q. Kong, M. Plumbley, and W. Wang, "Stacked convolutional neural networks for general-purpose audio tagging," *DCASE Challenge*, 2018.

[11] 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 *DCASE*, Tokyo, Japan, November 2020, pp. 100–104.

[12] O. Akiyama and J. Sato, "Dcase 2019 task 2: Multitask learning, semi-supervised learning and model ensemble with noisy data for audio tagging," *DCASE Challenge*, 2019.

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

[14] A. Agarwal, M. Dahleh, and T. Sarkar, "A marketplace for data: An algorithmic solution," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019.

[15] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, no. 2, pp. 201–221, 1994.

[16] L. Atlas, D. Cohn, R. Ladner, M. A. El-Sharkawi, and R. J. Marks, II, "Training connectionist networks with queries and selective sampling," in *NeurIPS*, 1990, pp. 566–573.

[17] L. S. Shapley, "A value for n-person games," *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307–317, 1953.

[18] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," in *ICML*, 2019, pp. 2242–2251.

[19] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. Song, and C. J. Spanos, "Towards efficient data valuation based on the shapley value," in *AISTATS*, 2019, pp. 1167–1176.

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

[21] L. Perron and V. Furnon, "Or-tools," Google. [Online]. Available: https://developers.google.com/optimization/

[22] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *MLSP*. IEEE, 2015, pp. 1–6.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *NeurIPS*, vol. 25, pp. 1097–1105, 2012.

[24] K. Choi, G. Fazekas, and M. B. Sandler, "Automatic tagging using deep convolutional neural networks," *CoRR*, vol. abs/1606.00298, 2016.

[25] 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," *Tr. ASLP*, vol. 28, pp. 2880–2894, 2020.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *JMLR*, vol. 12, pp. 2825–2830, 2011.

[27] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008, vol. 39.