Self-Supervised Dataset Pruning for Efficient Training in Audio Anti-spoofing

Abdul Hameed Azeemi, Ihsan Ayyub Qazi, Agha Ali Raza

Lahore University of Management Sciences

21030027@lums.edu.pk, ihsan.qazi@lums.edu.pk, agha.ali.raza@lums.edu.pk

Abstract

The computational cost for training neural anti-spoofing models has rapidly increased due to larger network architectures. Several dataset-pruning metrics have been proposed to increase the training efficiency of these models. However, these metrics require example labels and an initial training step to compute example scores which is computationally intensive. We propose a novel self-supervised pruning metric for efficient dataset pruning in neural anti-spoofing models. Our method identifies important examples and prunes the dataset in an efficient, self-supervised manner using the clustered embedding representation of audios. We demonstrate that our method exceeds the performance of four other pruning metrics on the ASVSpoof 2019 dataset across two anti-spoofing models while being 91% computationally more efficient. We also find differences in the distribution of certain attacks, which helps explain the better performance of self-supervised pruning over other metrics.

Index Terms: Anti-spoofing, fake audio detection, self-supervised, data pruning, automatic speaker verification, ASVspoof.

1. Introduction

Speech synthesis and voice conversion (VC) algorithms have improved rapidly in recent years, allowing for the generation of high-fidelity, natural-sounding audio. Although these algorithms enable important applications within human-computer interaction and assistive technologies [1], they allow the creation of fake audio, potentially leading to identity theft, spread of misinformation, and defeating Automatic Speaker Verification (ASV) systems. ASV systems aim to ascertain whether a particular utterance belongs to the claimed speaker. Thus voice-conversion algorithms can degrade the reliability of ASV systems considerably [2].

Numerous research efforts have been made under the ASVSpoof community to address this challenge to develop anti-spoofing systems that can reliably distinguish between synthetic and bonafide audio. Neural anti-spoofing models have shown encouraging performance in detecting spoofed audio from various attacks. Recent models can directly operate on raw speech and identify the spoofing artifacts effectively. However, training these models is a resource-intensive process, requiring a significant amount of compute, which also prevents their usage in resource-constrained environments (e.g., cheaper GPUs and on-device computing). This has motivated the development of light-weight anti-spoofing systems with limited parameters [3] and knowledge-driven models that utilize micro-features [1, 4]. Another recent work proposes a dataset-pruning method for efficient spoofed audio detection [5], which identifies a subset of examples important for generalization through an example-scoring metric. This subset is suitable for training the anti-spoofing models in resource-constrained environments. Although this approach is effective at significantly reducing the training data requirements, it requires example labels beforehand and needs an initial step to compute example scores for the complete training set, which is computationally intensive.

To address these limitations, we propose a novel self-supervised dataset pruning method for anti-spoofing models, inspired by the recent success of self-supervised data pruning in vision tasks [6]. Our method identifies important examples in an efficient self-supervised manner *without* requiring any initial training to compute example scores. We first obtain the embedding representation of audios in the ASVspoof dataset through the `wav2vec2` model and cluster them using k-means algorithm. We then construct a data subset by identifying the important examples based on their distance from the cluster centers and pruning the prototypical examples. We demonstrate that our method outperforms several other scoring metrics on the ASVSpoof dataset across two anti-spoofing models.

1.1. Our contributions

- To our knowledge, this is the first approach that presents a self-supervised dataset pruning method for efficient training of anti-spoofing models.
- We demonstrate that our method outperforms multiple existing dataset pruning metrics (EL2N [7], forgetting score [8], forgetting norm [5] and random pruning) across two models (AASIST-L [3] and RawNet2 [9]) on the ASVSpoof 2019 dataset [2], in addition to being 91% computationally more efficient than other metrics.
- To explain the qualitative differences between pruning metrics, we analyze the pruned subsets and find differences in the relative position and distribution of certain attacks through t-SNE plots.

2. Related Work

2.1. Dataset pruning

Dataset pruning has recently emerged as a promising tool for enabling efficient training of deep neural networks (DNNs) [7, 6]. Existing methods leverage different pruning metrics for identifying *informative* training examples, which include forgetting score [8], EL2N score, gradient norm score [7], forgetting norm [5], RHO-loss [10] and others [11, 12, 13, 14, 15, 16, 17, 18, 19, 20]. These metrics score individual training examples and select the *informative* training examples while discarding the prototypical examples. This procedure reduces the overall size of the dataset without significantly affecting the

generalization performance. Pruned datasets constructed in this manner enable efficient training of DNNs, thus making them suitable for resource-constrained environments. Several pruning metrics have been proposed for various speech tasks as well [21, 22, 23, 24, 25, 26, 27, 28, 29]. A key property of pruning metrics for DNNs is that they require an initial training run to compute the scores for all the training examples, which makes the pruning procedure costly and does not scale well to large datasets [6]. To address this, recent work proposes a self-supervised dataset pruning algorithm for vision tasks [6], which does not require example labels during pruning and is computationally cheaper. This approach has been shown to match the performance of the best supervised-pruning metric for vision tasks.

2.2. Resource-constrained anti-spoofing

The inefficiency of recent anti-spoofing models has encouraged the development of alternate lightweight spoof detection methods. Knowledge-driven models have been developed to distinguish between spoofed and bonafide audios using only the acoustic microfeatures, including Voicing Onset Time (VOT) and Coarticulation [1, 4]. Alternatively, limited parameter variants of neural anti-spoofing models have been proposed, e.g., AASIST-L [3], that enable efficient inference and match the EER performance of the larger models. Recent work proposes a dataset-pruning algorithm for resource-constrained training of anti-spoofing models [5]. The algorithm leverages *forgetting norm* for scoring individual training examples. This pruning metric combines the granularity of EL2N with the stability of the forgetting scores, allowing more deterministic pruning of ASVSpoof 2019 dataset. However, this metric still requires an initial computation step, making it computationally intensive and limiting scalability to larger datasets.

3. Preliminaries

Let $\mathcal{X}_l$ be a dataset consisting of pairs $(\mathbf{x}, y) \sim \mathcal{P}_{data}$, where $\mathbf{x}$ is the audio instance, $y \in \mathcal{Y}$ is the label (bonafide or spoof) and $\mathcal{P}_{data}$ is the probability distribution over the data. We use $\mathcal{S}$ to denote the data pruning strategy and π to represent the pruning metric. The goal of $\mathcal{S}$ is to leverage π to select *informative* examples from $\mathcal{X}_l$ and construct a smaller dataset $\mathcal{X}_s$. This dataset is then used to train a neural anti-spoofing model $\mathcal{N}_\theta(x)$ having parameters θ such that there is a minimal drop in the generalization performance compared to a model trained on full dataset $\mathcal{X}_l$. The performance of an anti-spoofing system is computed through equal error rate (EER) or the ASV-specific tandem decision cost function (t-DCF) [30].

Recently, there has been a consideration regarding the growing computational cost ($\mathcal{C}$) of the data pruning metrics, which at the minimum scales linearly with the size of the original dataset. The primary contributor to $\mathcal{C}$ is the initial training run for computing scores of each training example. Our goal is to develop a self-supervised pruning metric that minimizes this cost *while* matching or preferably beating the performance of the existing pruning metrics for neural anti-spoofing models.

4. Methodology

4.1. Existing pruning metrics

EL2N score [7]. It is the normed error of a training example $(\mathbf{x}_i, y_i)$ and is calculated as the difference between predicted

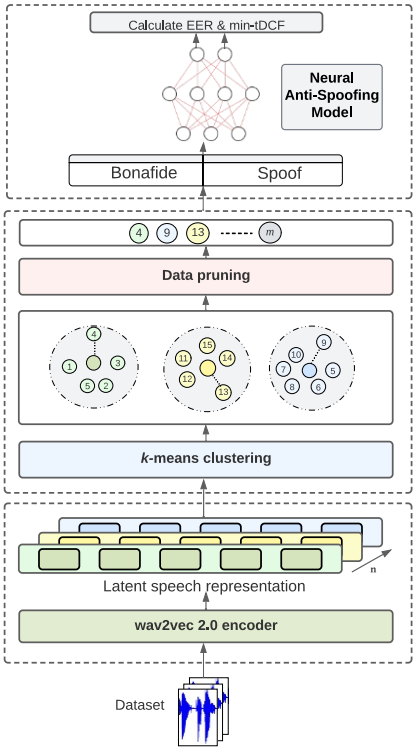

Figure 1: *The workflow for self-supervised dataset pruning for efficient training in audio anti-spoofing.*

probabilities and the ground-truth label,

$$\mathbb{E} \left\| \mathcal{N}_\theta \left(x_i \right) - y_i \right\|_2 \tag{1}$$

It captures the local information about the difficulty of a training example in early training epochs. *Harder-to-learn* examples have a high EL2N score while the *easier-to-learn* examples have a low score. Thus, a pruning strategy S on the EL2N score retains the higher-scoring examples while discarding the rest.

Forgetting Score [8]. It represents the number of times an example is *forgotten* during training, i.e., when the example is incorrectly classified at the epoch t after being classified correctly at $t-1$.

$$\sum_{t=1}^{N} \mathbb{1}_{\mathcal{Z}_i^t < \mathcal{Z}_i^{t-1}} \tag{2}$$

where $\mathcal{Z}_i^t$ is 1 if the example is classified correctly at epoch t and 0 otherwise. Examples with a higher forgetting norm are *harder-to-learn* and are thus preferred by the strategy S for inclusion in the pruned subset.

Forgetting Norm [5]. It combines the stability of forgetting scores with the granularity of EL2N. It achieves this by summing the difference between EL2N scores throughout training only in the epochs where this score increases. This metric has been shown to perform better on ASVSpoof 2019 dataset compared to EL2N and forgetting score.

$$\sum_{t=1}^{N} n_i^t * (\mathbb{E} \| \left| \mathcal{N}_{\theta^t} \left(x_i \right) - y_i \right\|_2 - \mathbb{E} \left\| \mathcal{N}_{\theta^{t-1}} \left(x_i \right) - y_i \right\|_2) \tag{3}$$

Table 1: *Pooled min-tDCF and EER (%) for different pruning metrics evaluated over [0, 0.1, 0.3, 0.5, 0.7, 0.9] dataset pruning fractions on AASIST-L and RawNet2. We do three runs for each combination and report the mean EER and min-tDCF.*

Dataset	Strategy	EER						min-tDCF					
		0	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**	**0**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
RawNet2	Random	5.24	5.68	6.52	6.97	8.32	15.29	0.14	0.15	0.20	0.21	0.24	0.41
	EL2N	5.23	5.84	6.24	7.74	10.24	40.14	0.14	0.15	0.18	0.22	0.32	1.00
	Forgetting Score	5.23	6.02	6.80	7.14	9.01	19.09	0.14	0.16	0.19	0.19	0.25	0.56
	Forgetting Norm	5.23	5.64	6.02	6.76	**8.11**	24.83	0.14	0.15	0.18	0.18	**0.22**	0.66
	Self-Supervised	**5.23**	**5.58**	**5.82**	**6.54**	8.58	**12.84**	**0.14**	**0.15**	**0.16**	**0.17**	0.23	**0.34**
AASIST-L	Random	2.10	2.18	2.67	3.54	4.90	13.74	0.06	0.06	0.09	0.10	0.14	0.34
	EL2N	2.10	2.14	2.99	3.46	5.54	31.48	0.06	0.06	0.09	0.11	0.15	0.69
	Forgetting Score	2.10	2.33	2.79	3.53	4.30	31.67	0.06	0.07	0.08	0.11	0.12	0.79
	Forgetting Norm	2.10	2.13	2.39	2.75	**3.65**	32.87	0.06	0.06	0.08	0.10	**0.11**	0.80
	Self-Supervised	**2.10**	**2.02**	**2.29**	**2.52**	4.51	**10.30**	**0.06**	**0.06**	**0.07**	**0.09**	0.13	**0.27**

Table 2: *EER (%) breakdown on the 13 attacks in the ASVspoof 2019 LA test set using the RawNet2 model on 0.9 pruning fraction.*

Score	**A07**	**A08**	**A09**	**A10**	**A11**	**A12**	**A13**	**A14**	**A15**	**A16**	**A17**	**A18**	**A19**	**t-DCF**	**EER**
Random	11.43	9.56	**2.29**	10.25	10.68	15.11	5.06	11.35	12.04	11.50	23.53	39.18	18.11	15.29	0.41
EL2N	42.08	44.42	34.78	43.59	38.22	42.88	38.40	35.65	41.05	40.88	45.95	**35.08**	37.75	40.14	1.00
Forgetting-Score	18.52	15.69	6.55	16.73	16.34	17.80	11.62	17.78	16.04	13.33	24.62	45.00	23.22	19.09	0.56
Forgetting-Norm	27.74	20.69	7.99	23.91	22.35	26.17	12.13	23.73	24.44	23.24	30.70	38.71	28.49	24.83	0.66
Self-Supervised	**7.62**	**8.59**	2.99	**6.59**	**6.68**	**12.25**	**3.13**	**9.57**	**9.11**	**7.72**	**19.07**	42.65	**16.20**	**12.84**	**0.34**

4.2. Self-Supervised Pruning

We now present a self-supervised pruning method for neural anti-spoofing models. We first utilize a pre-trained `wav2vec2-base` model (with 95M parameters) to obtain the contextualized speech representation $\mathcal{C}$ ($\mathbf{c}_1, \ldots, \mathbf{c}_T$ with T timesteps) for raw audios in the training dataset. `wav2vec2-base` is a self-supervised model consisting of a convolutional feature encoder that obtains latent representation for a raw audio ($\mathcal{X} \mapsto \mathcal{Z}$) and a transformer that learns contextualized representation from the latent representation ($\mathcal{Z} \mapsto \mathcal{C}$). For single audio, $\mathcal{C}$ is a multi-dimensional tensor ($\mathcal{C} \in \mathbb{R}^{l \times h}$), with l as the sequence length, and h as the hidden size. We flatten this and pad it to a length of $50,000$. We repeat this process for each audio in the training set and output the contextual representation for n training instances: $\mathcal{A} \in \mathbb{R}^{n \times 50000}$.

We then run k-means clustering on $\mathcal{A}$ and obtain k clusters. We compute the cosine distance for each training instance to its nearest cluster centroid. This distance theoretically represents the difficulty of each training example [6]. Thus, the *easier/prototypical* examples are closer to their cluster centroid, while the *harder* examples are farther away. The pruning strategy (S) then constructs a ranked list of distances in descending order and discards the bottom p examples while retaining the top $(1 - p)$ examples (for a pruning fraction p). The resulting dataset $\mathcal{X}_s$ is then used for training the neural anti-spoofing model. The complete workflow is summarized in Fig. 1.

5. Experiments and Results

5.1. Dataset and Metrics

We use the ASVSpoof 2019 dataset, specifically its logical access (LA) portion, for our experiments. The LA portion has been split into train, development, and test splits (Table 3). The training and dev splits contain audios generated through six diverse spoofing systems (A01 to A06), including two voice conversion (VC) and six text-to-speech (TTS) systems. The evaluation split comprises audios generated through thirteen separate systems (A07-A19), with two known and eleven unknown systems.

Table 3: *Bonafide and spoof audios in the training, development and test splits of the ASVSpoof 2019 LA portion.*

	Training	**Development**	**Evaluation**
Bonafide	2580	2548	7355
Spoof	22800	22296	63882

We use the tandem decision cost function (t-DCF) [30] and equal error rate (EER) for our evaluation. t-DCF measures the performance of an ASV system in tandem with the anti-spoofing system whereas `EER` only reflects the anti-spoofing performance.

5.2. Models

We use the RawNet2 [9] and AASIST-L [3] model for our experiments. RawNet2 is an end-to-end neural anti-spoofing model with convolutional neural network architecture. Its key components include residual layers, GRU layer, and filter-wise feature map scaling to derive discriminative representations. AASIST-L is another end-to-end anti-spoofing model having 85K parameters and is the light-weight variant of AASIST. A major component of AASIST-L is the heterogenous attention mechanism that effectively models spoofing artifacts. For running our experiments, we use a single 23GB NVIDIA V100 GPU. The implementation of pruning metrics is done in PyTorch. We use a batch size of 32 and Adam optimizer, with 1e-4 initial learning rate and 1e-4 weight decay for training AASIST-L and RawNet2. The models are trained for 50 epochs, and we use the LA development set to select the best checkpoint for final evaluation on the test set.

5.3. Results

Table 1 summarizes the `EER` and `min-tDCF` for five pruning metrics across six pruning fractions on RawNet2 and AASIST-L models, averaged across three independent runs. Overall, Self-Supervised pruning demonstrates improvement in performance over other pruning metrics. For the 0.9 pruning fraction

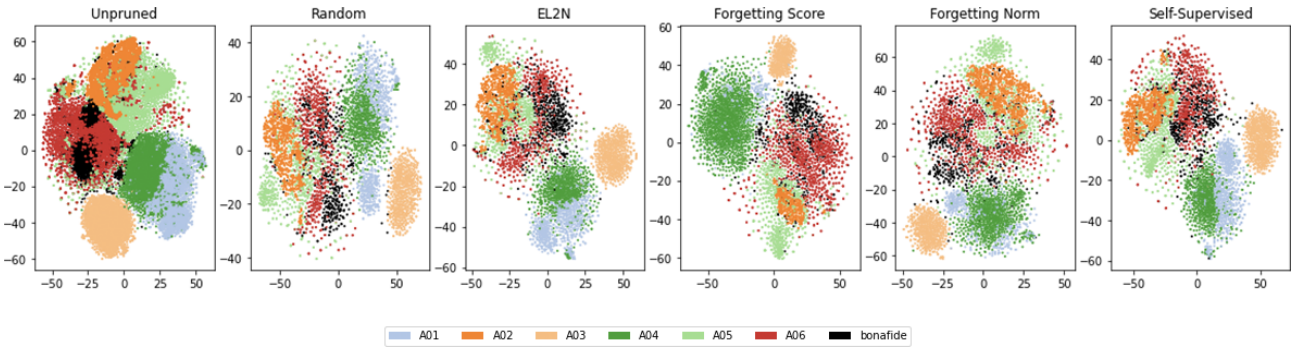

Figure 2: *t-SNE visualizations of ASVSpoof 2019 subsets pruned through different metrics including Random, EL2N scores, Forgetting Score, Forgetting Norm, and Self-Supervised Pruning. The pruning percentage is set to 80%. The colors represent different attacks.*

on RawNet2, we observe a 32.7% lower EER (19.09 vs. 12.84) and a 39.2% lower min-tDCF (0.56 vs. 0.34) compared to the second-best metric (forgetting score). This suggests that self-supervised metric is also a suitable choice for extreme pruning settings. We also compare the performance of different metrics across individual attacks (A07-A19) on the LA evaluation set on 0.9 pruning fraction. The results in Table 2 demonstrate the better performance of self-supervised pruning on each of the individual attacks compared to other pruning metrics.

5.4. Choosing the number of clusters

An important parameter in the self-supervised pruning process is the number of clusters (k) chosen for k-means clustering algorithm. We conduct an experiment to evaluate the impact of various values of k on the performance of the anti-spoofing model. We run clustering with different k and create multiple pruned subsets which are used to train separate anti-spoofing models. As we observe in Fig. 3, the value of k does lead to minor deviations in the final EER; however, the effect is inconsistent across different pruning fractions. For extreme pruning fractions (> 0.8), the changes in EER are more pronounced. This suggests that for practical pruning percentages, the selection of k has little influence over the final performance. If the number of classes is known, selecting that as k is a suitable choice. Interestingly, our observation is consistent with earlier finding in self-supervised pruning on ImageNet in vision tasks where the value of k can deviate an order of magnitude without significantly affecting the accuracy [6].

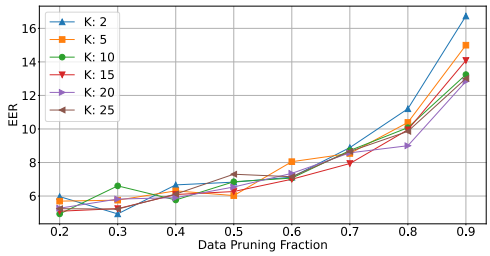

Figure 3: *Test EER of RawNet2 model on different values of k in k-means for self-supervised pruning.*

5.5. Comparison of pruned subsets

To understand the performance differences in pruning metrics, it is important to analyze the changes in the composition of the pruned subsets. We visualize these subsets by first reducing the dimensions of wav2vec2 embeddings through the tree-based t-SNE algorithm — with a perplexity value of 40 as suggested in [2] – and plotting the reduced dimensions. Figure 2 shows the scatter plot of the processed embedding vectors for a pruning percentage of 80%. The colors represent different attacks (A01-A06) in the pruned subsets. We notice differences in the position and distribution of certain attacks across different pruning metrics. The relative position of A01 and A04 in self-supervised is more similar to the unpruned dataset compared to other metrics. Given these observations and the comparison in Table 1 and 2, we hypothesize that the generalization performance of an anti-spoofing model on unknown attacks in the test set (e.g., A07-A19) is influenced by the training attack distribution (A01-A06, bonafide) *and* the quality of individual training instances within subsets created via a particular pruning metric. Further experiments that probe the nature of these individual instances would serve to validate these findings.

5.6. Efficiency of self-supervised pruning

We now compare the efficiency of self-supervised pruning with other pruning metrics, especially forgetting norm. We find that the complete self-supervised dataset pruning procedure primarily involving embedding computation through wav2vec2 and k-means clustering only requires 8.51% of the total pruning time that is required for computation of examples scores in other metrics, specifically forgetting norm (in RawNet2) on a V100 GPU. Thus, self-supervised pruning is $\sim 91\%$ more efficient than other pruning metrics, primarily since there are no expensive computations of example scores in self-supervised pruning.

6. Limitations and Conclusion

We propose a self-supervised dataset pruning method for efficient training in audio anti-spoofing models. Our approach outperforms other supervised metrics and significantly reduces the computational cost associated with a complete training run for determining the example scores. While we evaluated our approach on the widely used, standard spoof dataset (ASVSpoof 2019), further experiments are needed to verify the generalization of self-supervised pruning to other spoofed audio datasets. Additionally, a more granular analysis of the pruned subsets at the level of individual training instances would be helpful to understand the performance differences between different pruning metrics.

7. References

[1] H. Dhamyal, A. Ali, I. A. Qazi, and A. A. Raza, "Fake audio detection in resource-constrained settings using microfeatures," *Proc. Interspeech 2021*, pp. 4149–4153, 2021.

[2] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee *et al.*, "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, p. 101114, 2020.

[3] J.-W. Jung, H.-S. Heo, H. Tak, H.-J. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks," *ICASSP 2022*, pp. 6367–6371, 05 2022.

[4] Y. Gao, J. Lian, B. Raj, and R. Singh, "Detection and evaluation of human and machine generated speech in spoofing attacks on automatic speaker verification systems," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 544–551.

[5] A. H. Azeemi, I. A. Qazi, and A. A. Raza, "Dataset pruning for resource-constrained spoofed audio detection," *Proc. Interspeech 2022*, pp. 416–420, 2022.

[6] B. Sorscher, R. Geirhos, S. Shekhar, S. Ganguli, and A. S. Morcos, "Beyond neural scaling laws: beating power law scaling via data pruning," in *Advances in Neural Information Processing Systems*, 2022.

[7] M. Paul, S. Ganguli, and G. K. Dziugaite, "Deep learning on a data diet: Finding important examples early in training," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[8] M. Toneva, A. Sordoni, R. T. d. Combes, A. Trischler, Y. Bengio, and G. J. Gordon, "An empirical study of example forgetting during deep neural network learning," *ICLR 2019*, 2019.

[9] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with rawnet2," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6369–6373.

[10] S. Mindermann, J. M. Brauner, M. T. Razzak, M. Sharma, A. Kirsch, W. Xu, B. Höltgen, A. N. Gomez, A. Morisot, S. Farquhar *et al.*, "Prioritized training on points that are learnable, worth learning, and not yet learnt," in *International Conference on Machine Learning*. PMLR, 2022, pp. 15 630–15 649.

[11] V. Kaushal, R. Iyer, S. Kothawade, R. Mahadev, K. Doctor, and G. Ramakrishnan, "Learning from less data: A unified data subset selection and active learning framework for computer vision," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1289–1299.

[12] H. Saadatfar, S. Khosravi, J. H. Joloudari, A. Mosavi, and S. Shamshirband, "A new k-nearest neighbors classifier for big data based on efficient data pruning," *Mathematics*, vol. 8, no. 2, p. 286, 2020.

[13] S. Durga, R. Iyer, G. Ramakrishnan, and A. De, "Training data subset selection for regression with controlled generalization error," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9202–9212.

[14] S. Kothawade, N. Beck, K. Killamsetty, and R. Iyer, "Similar: Submodular information measures based active learning in realistic scenarios," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[15] K. Killamsetty, D. Sivasubramanian, B. Mirzasoleiman, G. Ramakrishnan, A. De, and R. Iyer, "Grad-match: A gradient matching based data subset selection for efficient learning," *PMLR 2021*, 2021.

[16] O. Ahia, J. Kreutzer, and S. Hooker, "The low-resource double bind: An empirical study of pruning for low-resource machine translation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3316–3333. [Online]. Available: https://aclanthology.org/2021.findings-emnlp.282

[17] L. Huang, K. Sudhir, and N. Vishnoi, "Coresets for time series clustering," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[18] S. Jiang, R. Krauthgamer, X. Wu *et al.*, "Coresets for clustering with missing values," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[19] I. Jubran, E. E. Sanches Shayda, I. Newman, and D. Feldman, "Coresets for decision trees of signals," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[20] B. Mirzasoleiman, J. Bilmes, and J. Leskovec, "Coresets for data-efficient training of machine learning models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6950–6960.

[21] Y. Wu, R. Zhang, and A. Rudnicky, "Data selection for speech recognition," in *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*. IEEE, 2007, pp. 562–565.

[22] D. Yu, B. Varadarajan, L. Deng, and A. Acero, "Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion," *Computer Speech & Language*, vol. 24, no. 3, pp. 433–444, 2010.

[23] Y. Hamanaka, K. Shinoda, S. Furui, T. Emori, and T. Koshinaka, "Speech modeling based on committee-based active learning," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 4350–4353.

[24] U. Nallasamy, F. Metze, and T. Schultz, "Active learning for accent adaptation in automatic speech recognition," in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 360–365.

[25] K. Wei, Y. Liu, K. Kirchhoff, C. Bartels, and J. Bilmes, "Submodular subset selection for large-scale speech training data," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3311–3315.

[26] T. Fraga-Silva, J.-L. Gauvain, L. Lamel, A. Laurent, V.-B. Le, and A. Messaoudi, "Active learning based data selection for limited resource stt and kws," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[27] C. Ni, C.-C. Leung, L. Wang, N. F. Chen, and B. Ma, "Unsupervised data selection and word-morph mixed language model for tamil low-resource keyword search," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4714–4718.

[28] C. Ni, C.-C. Leung, L. Wang, H. Liu, F. Rao, L. Lu, N. F. Chen, B. Ma, and H. Li, "Cross-lingual deep neural network based submodular unbiased data selection for low-resource keyword search," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6015–6019.

[29] A. Awasthi, A. Kansal, S. Sarawagi, and P. Jyothi, "Error-driven fixed-budget asr personalization for accented speakers," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7033–7037.

[30] T. Kinnunen, K. A. Lee, H. Delgado, N. Evans, M. Todisco, M. Sahidullah, J. Yamagishi, and D. A. Reynolds, "t-dcf: a detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification," *arXiv preprint arXiv:1804.09618*, 2018.