Interspeech 2022
18-22 September 2022, Incheon, Korea

Dataset Pruning for Resource-constrained Spoofed Audio Detection

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 performance of neural anti-spoofing models has rapidly improved in recent years due to larger network architectures and better training methodologies. However, these systems require considerable training data for achieving high performance, which makes it challenging to train them in compute-restricted environments. To make these systems accessible in resource-constrained environments, we consider the task of training neural anti-spoofing models with limited training data. We apply multiple dataset pruning techniques to the ASVspoof 2019 dataset for selecting the most informative training examples and pruning a significant chunk of the data with minimal decrease in performance. We find that the existing pruning metrics are not simultaneously granular and stable. To address this problem and further improve the performance of anti-spoofing models on pruned data, we propose a new metric, Forgetting Norm, to score individual training examples with higher granularity. Extensive experiments on two anti-spoofing models, AASIST-L and RawNet2, and several pruning settings demonstrate up to 23% relative improvement with forgetting norm over other baseline pruning heuristics. We also demonstrate the desirable properties of the proposed metric by analyzing the training landscape of the neural anti-spoofing models.

Index Terms: Spoofing, data pruning, subset selection, automatic speaker verification, ASVspoof, fake audio.

1. Introduction

There has been significant progress recently towards better voice conversion and speech synthesis systems. These systems have numerous applications within assistive technologies, gaming and human-computer interaction [1]. However, they can also be used to generate natural-sounding audios which can defeat biometric identification methods and automatic speaker verification (ASV) systems. This can lead to the potential misuse of these systems, e.g., voice-based security systems can be fooled via fake audios thereby granting access to protected information and restricted areas. Moreover, such techniques can also be used for spreading misinformation. Given these negative impacts of spoofed audios, it is important to design systems that can accurately distinguish between bonafide and synthetic audios generated using a wide range of techniques.

The ASVspoof community has devised a series of challenges and datasets to facilitate the research on spoofing detection systems [2, 3, 4, 5]. The two primary scenarios considered in these challenges are logical access and physical access. Logical access (LA) consists of the attacks through voice conversion and speech synthesis systems including text-to-speech models, whereas the physical access pertains to the attacks via replayed audios. Several deep learning based models have been proposed to tackle the LA attacks which have demonstrated high performance through specialized architectures that can automatically detect the spoofing artefacts present within spectral and temporal domains [6, 7]. However, these models require a significant amount of training data to achieve high performance and a low *equal error rate* (EER) on the test set, which presents a significant challenge when training these models within resource-constrained settings and compute-restricted environments.

The strategies proposed for resource-constrained spoofing detection include the construction of lightweight, robust deep learning based models with limited parameters [7] or the use of acoustic microfeatures in knowledge-driven models, which can distinguish between spoofed and bonafide audios [1, 8, 9, 10]. However, the methods for reducing the size of the training data (or data pruning approaches) have not yet been explored in the context of spoofed audio detection. In this work, we consider the task of automatically pruning the ASVspoof 2019 dataset to obtain a representative subset that can be used for training deep learning based anti-spoofing models in resource-constrained settings (Figure 1). We consider several heuristics and scoring metrics to select the most informative training examples and prune the rest of the data.

1.1. Contributions

The presented research makes the following contributions:

- We apply dataset pruning to neural anti-spoofing models by scoring individual training examples using different metrics (normed error, forgetting score) and selecting a subset of the examples for training, using these scores.
- We propose a new scoring metric, the *forgetting norm*, for better dataset pruning.
- Our empirical evaluation on the ASVspoof dataset [11] and two deep learning-based anti-spoofing models shows that training the model on the subset selected by *forgetting norm* performs better than the existing metrics (random pruning, normed error, and forgetting scores).

2. Background Work

A number of methods have been proposed to quantify the importance of individual training examples in deep learning models for standard classification tasks [12, 13, 14, 15, 16, 17, 18]. These approaches emphasize the identification of *informative* training examples via different heuristics and removing non-representative samples from the dataset. Toneva et al. [19] consider the 'forgetfulness' of a training example by measuring the number of times it is misclassified after being classified correctly during training. The number of such incorrect classifications represents the forgetting score of a particular example. The examples that are repeatedly forgotten are selected to construct a smaller training subset without significantly affecting the generalization performance. Paul et al. [17] propose to score the examples by the norm of the gradient (GraNd) and the norm of the error vector (EL2N). Both these measures are found to be correlated and any of these can be used for removing a large number of less informative examples while retaining the test

10.21437/Interspeech.2022-514

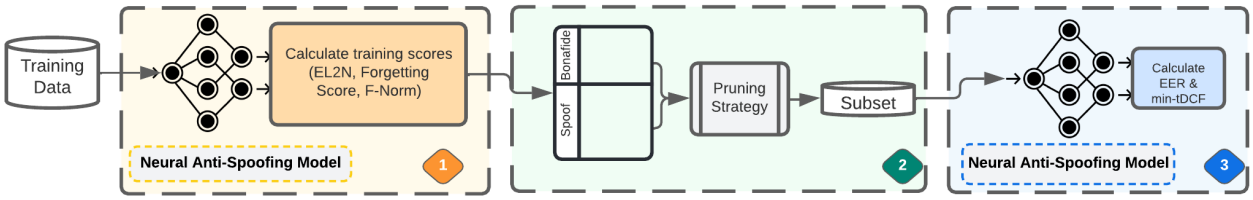

Figure 1: *The overall framework of data pruning for training anti-spoofing models. In step (1), we train a neural anti-spoofing on the complete dataset (ASVspoof in our case) and record multiple scores for each training example: normed error (EL2N score), forgetting score, and forgetting norm. In step (2), we create a data subset by ranking the examples according to the computed score and selecting the most informative examples. We then use this subset in step (3) to train the complete model and then evaluate on the test dataset.*

accuracy on multiple standard vision datasets (CIFAR-10 and CIFAR-100) and convolutional neural networks (ResNet). Another interesting finding is that the loss landscape in the early training epochs contains sufficient information which can be leveraged to construct a smaller subset of the training data without training the neural network for all the epochs.

Coreset selection is another approach of reducing the dataset size by extracting representative samples [20, 21, 22, 23]. To construct the theoretical error boundaries, many coreset selection algorithms require the problem to demonstrate a special structure like convexity. Hence, the application of coreset algorithms is limited in deep learning tasks and optimal solution is seldom guaranteed for the downstream tasks. In speech tasks, especially automatic speech recognition (ASR), subset selection algorithms focus on the generation of phonetically rich and diverse subsets through phoneme level error models [24, 25, 26, 27, 28, 29, 30, 31, 32]. Awasthi et al. [32] propose an objective function to select a subset of sentences for training an ASR personalization model. The created subset contains more challenging and informative sentences than a random selection approach. These error models for ASR and other speech tasks can provide good insights on the properties of representative subsets but are not directly applicable to anti-spoofing due to the different nature of the anti-spoofing problem.

3. Preliminaries

Consider a neural anti-spoofing model $f(x;\theta)$ ($\theta \in \mathcal{R}^d$) that is trained on a labelled dataset $x \in \mathcal{D}_l$ (e.g., ASVspoof). $\mathcal{D}_l$ consists of pairs of audio and the corresponding label (bonafide or spoof) and θ represents the neural network parameters. Our goal is to prune $\mathcal{D}_l$ to obtain a subset B_l such that the performance of the anti-spoofing model $f(x;\theta)$ after training on B_l is better than random pruning. The performance of anti-spoofing is commonly evaluated via equal error rate (EER) or the ASV-centric tandem decision cost function (t-DCF) [33].

Let σ be the softmax function given by $\sigma\left(z_1,\ldots,z_K\right)_k = \exp\left\{z_k\right\} / \sum_{k'=1}^{K} \exp\left\{z_{k'}\right\}$. The output of the anti-spoofing model is then a probability vector $p(x,\theta) = \sigma(f(x,\theta))$, where $f(x;\theta)$ are the logit outputs of the neural anti-spoofing model.

4. Method

We now describe three metrics for scoring individual training which can facilitate dataset pruning. The scores calculated via these metrics are used by the pruning algorithm to construct a subset of the training data by selecting the highest error (most informative) examples.

4.1. EL2N score [17]

The normed error (or the EL2N score) [17] of a training example (x_i, y_i) at epoch t is defined as,

$$\mathbb{E}\left\|f\left(x_i;\theta^t\right) - y_i\right\|_2 \tag{1}$$

It is essentially the norm of the difference between the predicted class probabilities (for spoof and bonafide classes) and the ground-truth label encoded as one-hot vector. An example that is more difficult to classify will have a high normed error as compared to an easier example. A desirable property of the EL2N score is that it captures the difficulty of a training example with greater granularity as compared to the other metrics that are discrete, e.g., forgetting scores. Hence, the error normed scores computed after a few training epochs can be used for data pruning. We refer to the EL2N score computed at epoch t as EL2N_t.

4.2. Forgetting score [19]

Let the binary variable $\text{acc}_i^t = \mathbb{1}_{\hat{y}_i^t = y_i}$ indicate whether the training example i is correctly classified at epoch t. A *forgetting event* is a point in training when the neural network misclassifies a training example after classifying it correctly, i.e., $\text{acc}_i^t < \text{acc}_i^{t-1}$ [19]. The *forgetting score* of a particular training example is the number of times a training example undergoes a *forgetting event*.

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

Toneva et al. [19] demonstrate that the examples with a higher forgetting score are more informative examples and can be used to create a smaller subset for training the neural network. As forgetting events occur throughout the training, the forgetting scores are calculated towards the end of the training. Hence, they are more stable than the other metrics which are calculated on a particular training epoch early in training, e.g., EL2N score.

4.3. Forgetting norm

We introduce a new scoring metric, *forgetting norm*, which is defined as the increase in normed error across two successive epochs. Let $n_i^t = \mathbb{1}_{\text{EL2N}_i^t > \text{EL2N}_i^{t-1}}$ be a binary variable indicating whether the EL2N score at epoch *t* is greater than the EL2N score at epoch *t-1*. *Forgetting norm* is then defined to be,

$$\sum_{t=1}^{N} n_i^t * \left(\text{EL2N}_i^t - \text{EL2N}_i^{t-1}\right) \tag{3}$$

Table 1: *Pooled EER for the four strategies of pruning the training set evaluated over multiple pruning fractions and different neural anti-spoofing models. Each score is calculated by averaging over 10 runs and is then used for a particular pruning strategy. We do three independent runs for each result for 5 epochs and report the mean test EER. The forgetting norm consistently demonstrates the lowest EER at various pruning fractions and architectures. In the last column, we report the EER on the original dataset without any pruning.*

Architecture	Pruning Percentage	Pruning Strategy				No Pruning
		Random	**EL2N**	**Forgetting Score**	**Forgetting Norm**	
AASIST-L	30%	9.27	7.29	7.15	**6.62**	6.17
	60%	15.14	13.01	12.69	**11.63**	
	90%	18.14	17.74	17.95	**16.30**	
RawNet2	30%	14.68	13.39	11.09	**9.85**	8.82
	60%	16.55	15.65	14.01	**12.90**	
	90%	18.38	18.01	17.23	**16.02**	

Table 2: *Breakdown EER (%) performance of all 13 attacks that exist in the ASVspoof 2019 LA evaluation set, pooled min t-DCF, and pooled EER on a pruning fraction of 60% and AASIST-L model. The scores for four pruning strategies are reported. F-Score: Forgetting Score, F-Norm: Forgetting Norm*

Score	**A07**	**A08**	**A09**	**A10**	**A11**	**A12**	**A13**	**A14**	**A15**	**A16**	**A17**	**A18**	**A19**	**t-DCF**	**EER**
EL2N	4.64	7.35	1.22	6.16	2.29	11.3	1.23	10.92	5.88	15.55	19.27	36.16	19.50	0.33	13.01
F-Score	2.40	5.66	1.20	4.31	1.87	8.75	1.12	7.91	4.52	15.40	20.98	33.60	19.50	0.32	12.69
F-Norm	**2.32**	**4.70**	**0.41**	**3.60**	**1.22**	**7.26**	**0.95**	**6.39**	**4.35**	**12.40**	**18.26**	**33.13**	**18.44**	**0.29**	**11.63**

which expands to,

$$\sum_{t=1}^{N} n_i^t * (\mathbb{E} \left\| f\left(x_i; \theta_t\right) - y_i \right\|_2 - \mathbb{E} \left\| f\left(x_i; \theta_{t-1}\right) - y_i \right\|_2) \quad (4)$$

In other words, the forgetting norm is the cumulative sum of the difference of EL2N scores over n training epochs, computed in the epochs where $\text{EL2N}_t > \text{EL2N}_{t-1}$ (and not only the epochs where a forgetting event occurs). Compared to the forgetting score that only gets incremented in case of an actual misclassification (as normed error crosses a threshold), forgetting norm captures the more subtle error increase that may result in a misclassification in later epochs. Hence, *forgetting norm* combines the granularity of *normed* error (EL2N) with the stability of *forgetting* scores.

4.4. Dataset Pruning Algorithm

We now present the dataset pruning algorithm for neural anti-spoofing models (Algorithm 1). We train the model on the ASVspoof dataset and calculate the score for each example according to the selected metric. We then sort the examples in a descending order of the calculated scores and select the highest scoring examples.

Algorithm 1: Dataset Pruning for Neurals Anti-Spoofing Models

Input: Anti-Spoofing Model f, Dataset D_l, Pruning Fraction f, Pruning Strategy s, Training Epoch e
$S \leftarrow$ Train f on D_l and compute scores for each training example on epoch e according to strategy s
$subsetSize \leftarrow (1 - f) * len(D_l)$
$S \leftarrow sort(S, descending = true)$
$B_l \leftarrow S[0 : subsetSize]$

5. Experiments

5.1. Models

We use two neural anti-spoofing models: AASIST-L [7] and RawNet2 [6] for our data pruning experiments.

AASIST-L: AASIST-L [7] is a lightweight end-to-end audio anti-spoofing model based on graph neural networks. The graph modules and heterogeneous stacking graph attention layer can efficiently model spoofing artefacts present in temporal and spectral domains. The *max graph operation* detects various spoofing artefacts in parallel and combines them.

RawNet2: RawNet2 [6] is also an end-to-end audio anti-spoofing model. It is based on a convolutional neural network architecture that ingests raw speech and outputs the prediction: bonafide or spoof. An important part of the RawNet2 architecture is a GRU layer containing 1024 hidden nodes which can produce a single utterance-level representation by aggregating frame-level representations.

5.2. Dataset

The experiments are carried out on the logical access (LA) part of the ASVspoof 2019 dataset [11]. This dataset is split into three parts: train, development, and evaluation set. The attacks present in the train and development sets were created from six spoofing algorithms (A01-A06). The attacks in the evaluation set were created from thirteen algorithms (A7-A19). The pruning is done on the train set only, and the evaluation is done on the original (complete) evaluation set. We consider three pruning percentages: 30%, 60%, and 90% to mirror the low, moderate and extreme resource-constrained settings. The pruning is performed separately on the spoof and bonafide portions in order to maintain the ratio defined in the ASVspoof dataset.

5.3. Metrics

To evaluate the performance of the anti-spoofing model on the pruned dataset, we use equal error rate (EER) and the minimum tandem detection cost function (t-DCF) [33]. EER measures the performance of standalone anti-spoofing system whereas min t-DCF evaluates the combined performance of the automatic speaker verification system and the anti-spoofing system.

5.4. Implementation Details

The scoring metrics are implemented using the PyTorch library in Python. The official implementation of AASIST-L and RawNet is used. We use a single 40GB NVIDIA A100 GPU for running all the experiments. For calculating the scores for each training example, we initiate a computation step after each epoch and record the score (EL2N, forgetting Score, forgetting norm). The scores in each epoch are averaged over 10 runs and then used for a particular pruning strategy. We then train the model for 5 epochs on the pruned dataset. For each test EER and min t-DCF reported, we do three independent runs (with independent model initialization) and calculate the average.

5.5. Results

Table 1 shows the results of pruning experiments via different pruning strategies across multiple neural anti-spoofing models. For each architecture and the scoring metric, we consider multiple pruning percentages and report the pooled EER. We find that forgetting norm consistently performs better than the pruning performed on the basis of random, EL2N, and forgetting scores for the majority of pruning percentages. For AASIST-L and 60% pruning percentage, we notice a 23% relative drop in the EER as compared to the random pruning (15.14% vs 11.63%) and an 8% relative drop as compared to the forgetting score (12.69% vs 11.63%).

Comparison of individual attacks. Table 2 shows a performance comparison for each of the individual attacks present in the ASVspoof LA evaluation subset. We again observe a consistent improvement in pooled EER and pooled min t-DCF for forgetting norm. For multiple attacks, substantial improvements are observed for forgetting norm as compared to the forgetting score, e.g., for the A09 attack, forgetting norm shows a 65% relative improvement (1.20% vs 0.41%).

Properties of Forgetting Norm. To understand the underlying properties of the forgetting norm which contribute to its better performance, we analyze the training landscape by computing the scores (forgetting score and forgetting norm) for each example in every epoch (Figure 2). We find that the forgetting norm demonstrates a continuous behavior as compared to the forgetting score which only has discrete increments. This allows the forgetting norm in the early training epochs to be used for pruning effectively, a property that is not present in the forgetting scores. We also study the correlation between forgetting scores and forgetting norm for each training example in ASVspoof (Figure 3). We find that multiple examples with the same forgetting score can have different values of the forgetting norm, e.g., if an example is misclassified once, the forgetting norm will assign a continuous score to that event whereas the forgetting score will have the discrete value of one. Additionally, the forgetting norm can be non-zero even in the cases where the forgetting score is zero, which can facilitate the early prediction of a misclassification event when it has not yet occurred, e.g. if the normed error of a correctly classified example consistently increases across multiple epochs, there is a high probability that it will be misclassified in a later epoch.

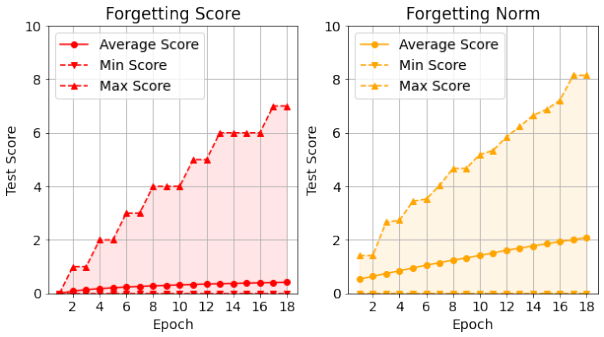

Figure 2: *Comparison of forgetting score and the forgetting norm of the examples over multiple training epochs. Note that the forgetting scores increase in discrete steps after each epoch whereas the forgetting norm adopts a continuous trajectory.*

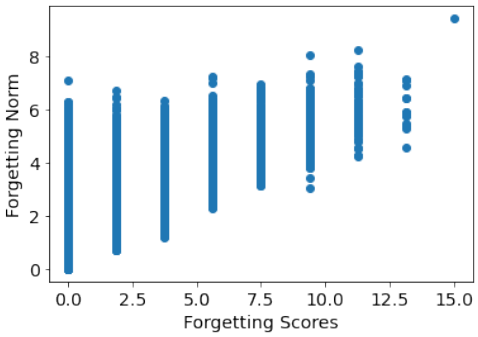

Figure 3: *The forgetting score and the forgetting norm of the examples in the ASVspoof training set. For each forgetting score, there are multiple examples that have a varying forgetting norm which aids in differentiating those examples based on their importance and hence allows for more deterministic pruning.*

6. Limitations and Conclusion

We now discuss some limitations of our approach and the potential future directions for resource-constrained spoofed audio detection. Our pruning experiments were performed on two end-to-end (E2E) neural anti-spoofing models. It needs to be investigated if data pruning is applicable to other types of anti-spoofing architectures that do not operate directly on raw speech waveform. While we investigated the pruning approaches for anti-spoofing on the standard ASVspoof dataset, it needs to be examined if the techniques are applicable to any other datasets that are used in training spoofing countermeasure models. Additionally, it should be explored if the ASVspoof subsets created through the presented pruning approach demonstrate reasonable performance on other spoofing countermeasures too, e.g., GMM-based classifiers for separating spoofed audio from bonafide speech. For future work, it will be interesting to explore the properties of the subsets of ASVspoof created via data pruning approaches and analyze the composition of various types of attacks in those subsets. It will also be useful to leverage data pruning for other related tasks like partially spoofed audio detection, replay attack identification, and spoofing aware speaker verification.

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] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth annual conference of the international speech communication association*, 2015.

[3] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," 2017.

[4] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee, "Asvspoof 2019: Future horizons in spoofed and fake audio detection," *arXiv preprint arXiv:1904.05441*, 2019.

[5] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans *et al.*, "Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection," *arXiv preprint arXiv:2109.00537*, 2021.

[6] 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.

[7] 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," *arXiv preprint arXiv:2110.01200*, 2021.

[8] 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.

[9] D. Paul, M. Pal, and G. Saha, "Spectral features for synthetic speech detection," *IEEE journal of selected topics in signal processing*, vol. 11, no. 4, pp. 605–617, 2017.

[10] X. Xiao, X. Tian, S. Du, H. Xu, E. Chng, and H. Li, "Spoofing speech detection using high dimensional magnitude and phase features: the ntu approach for asvspoof 2015 challenge." in *Interspeech*, 2015, pp. 2052–2056.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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," *arXiv preprint arXiv:2103.00123*, 2021.

[17] 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.

[18] O. Ahia, J. Kreutzer, and S. Hooker, "The low-resource double bind: An empirical study of pruning for low-resource machine translation," *arXiv preprint arXiv:2110.03036*, 2021.

[19] 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," *arXiv preprint arXiv:1812.05159*, 2018.

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

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

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.