Self-Incremental Training for Personalized Voice Command Recognition in a Wireless Audio Sensor Network

Manuele Rusci
ESAT, *KU Leuven*, Belgium
manuele.rusci@esat.kuleuven.be

Hugo Van hamme
ESAT, *KU Leuven*, Belgium
hugo.vanhamme@kuleuven.be

Tinne Tuytelaars
ESAT, *KU Leuven*, Belgium
tinne.tuytelaars@kuleuven.be

***Abstract*—This paper studies self-incremental training in the context of personalized Deep Neural Networks (DNNs) for voice command recognition tailored for resource-constrained sensor nodes. The learning task runs when new unsupervised data becomes available within a Wireless Audio Sensor Network (WASN). After collecting a new multi-sensor dataset of voice commands, we experimentally investigate network-level policies to assign pseudo-labels to the new data. Our baseline analysis shows an accuracy improvement of up to +15% with respect to models pretrained on a large keyword corpus dataset. The multi-sensor labeling strategy closely approximates the performance achieved in a single-sensor scenario providing a clean signal, while we observe +4.7% compared to other sensors with degraded signal quality.**

***Index Terms*—KWS, multi-sensor, wireless audio sensor network, self-training, incremental learning, personalization.**

I. Introduction

Voice command recognition, also referred as Keyword Spotting (KWS), is nowadays a common functionality for several home appliances, e.g., turning on the lights in a room with a simple "Lights On" spoken command. To avoid the transmission of the recorded audio to a remote server for the interpretation of the spoken commands, recent studies have explored lightweight Deep Neural Networks (DNNs) for voice command classification that can fit the limited resources of small battery-power sensor nodes of a WASN [1]. The DNNs are trained end-to-end on a set of predefined keywords, but unfortunately lack mechanisms to learn new keywords or improve the detection accuracy over time with respect to the users.

To address this weakness, a personalized voice command model can define the target keywords when the algorithm is already deployed on the sensor device, by recording a few examples from a specific user [2]. Precedent works used a pretrained DNN-based audio encoder to extract a feature vector representation, namely the prototype, from a set of user-defined keywords, and then used a similarity score between audio data and the prototypes to carry out the detection [3]–[6]. Given the frozen nature of the encoder, the final accuracy of these approaches on a new testset is highly impacted by the distribution shift with respect to the training data [7].

This work is partly supported by the Horizon Europe program under grant agreement 101067475.

This work investigates self-incremental training for personalized voice command recognition within a network of battery-powered audio nodes. To learn the distribution of the target domain data, recent self-training methods assign pseudo-labels to new unsupervised data to retrain or fine-tune the DNN audio model [8]–[11]. Our study generalizes a precedent single-sensor self-learning approach for personalized KWS tasks [12] to a multi-sensor scenario by analyzing two network-level policies for labeling the new data. More in detail, the label assignment functions rely on a global similarity score between the captured audio and the keyword prototype, which is computed by, respectively, minimizing or averaging the sensors' scores at every time stamp. Our study is conducted on a novel dataset of voice commands collected with a network of 4 audio sensor nodes, each one including an ultra-low power microphone and a microcontroller.

This work makes two important contributions:

- we present and publicly release the first-of-its-kind dataset of voice commands collected with a WASN composed by 4 ultra-low power sensors[1], and
- we provide a baseline analysis of self incremental-training on the multi-sensor dataset, and we compare the results vs. an ideal oracle or single-sensor solutions.

Compared with frozen DNNs trained on a large keyword corpus [7], the self-incremental training leads to accuracy improvements of up to +19% and +15%, when using, respectively, a single-sensor or a multi-sensor global score for labeling. In the latter case, the final accuracy closely approximates the best-case scenario that uses a single clear signal, and we observe an increase of up to +4.7% compared to using a sensor distant from the speaker. Hence, the multi-sensor self-training can recover from potential pitfalls due to individual sensor positioning. The code can be found at: https://github.com/mrusci/multi-sensor_voice_command_recognition

II. Related Work

DNNs for personalized voice command recognition typically compare the embedding extracted from a spoken utterance and a set of (few-shot) examples. A frequent choice for a DNN-based audio encoder is the Prototypical Network

[1]The dataset is available at: https://doi.org/10.48804/IEKKVZ

TABLE I
DATASETS FOR VOICE COMMAND AND MULTI-SENSOR SPEECH RECOGNITION.

Dataset	# sensors	Speakers	hours	Task	#keywords
Hey Snips [16]	1 mic	2.2k	96	Keyword	1
Open KWS [17]	1 mic	65	n/a	Spotting	16*
SINS [18]	13 mic arrays	no	n/a	Acoustic Scene Class.	–
VoiceHome [19]	1 mic arrays	3	2.5	Automatic	–
Chime-6 [20]	6 mic arrays	40	49	Speech	–
DipCO [21]	5 mic arrays	16	5	Recognition	–
This Work	**4 mics**	**20**	**23.7**	**KWS**	**5**

* more than 1 keyword may appear in a single utterance.

(ProtoNet) [13], which is trained with a supervised loss that minimizes the distance between the embeddings of samples belonging to the same class [3]–[6]. To assess the keyword detection following a "Query-by-Example" principle, *Chen et al.* [14] compared the embedding extracted from a 2-layer LSTM with a reference vector computed from the reference utterance. Other solutions relied instead on phoneme sequences extracted from an LSTM encoder [2], [15]. Our study builds on top of the precedent set of approaches by using new unsupervised data to incrementally train the audio feature encoder, which is kept frozen by the previous works. Moreover, we focus on a multi-sensor scenario, while the precedent works only focused on single-sensor data.

A few recent studies have employed self-training for Automatic Speech Recognition (ASR), in which the audio DNN models are refined using audio transcriptions predicted by the models themselves [8]–[10]. Following this principle, [11] incrementally fine-tuned a KWS classifier that was initially pretrained on 2.5 million "Ok Google" utterances. The self-training logic has been recently adapted to a personalized KWS scenario [12] by producing pseudo-labels based on the distance between new audio embedding and the prototype of new keywords. As a notable experimental result, after self-learning the accuracy increased by up to +16% when using lightweight DNN models with a number of parameters up to 0.5 M. In this paper, we apply the latest method to a new multi-sensor dataset after defining a logic to aggregate the scores of the sensors.

The work on voice command recognition using a multi-sensor setup is extremely limited, also because of the absence of datasets. Voice command recognition is typically limited to single-sensor benchmarks [2], [16], [17], as reported in Tab. I. Multi-sensor datasets mainly include transcribed audio to study (far-field) speech recognition tasks [19]–[22]. The work [23] described a distributed processing logic to classify the acoustic scenes of the SINS dataset [18]. The authors proposed to combine the node predictions by averaging the posterior probabilities. To target a robust KWS detection, *Ahmed et al.* [24] applied a majority voting scheme over the outputs of the models running on multiple devices. Conversely, we collected a new multi-sensor dataset of voice commands to study model fine-tuning with respect to new user-defined keywords, which is not addressed by precedent works.

III. METHOD

A. Multi-Sensor Personalized Voice Command Model

In a personalized voice command recognition model, the target command is defined at runtime by recording a few spoken examples from a specific user. Let us denote $\{r_i\}, i = 1, ..K$ as the audio recordings of the command, where K is the number of utterances ($K = 3$ in our work). To predict the presence of the target keyword, a similarity score is computed between an audio sample x and the provided examples:

$$f(x) = d(h(x), c_p) \tag{1}$$

where $h(\cdot)$ is a DNN-based audio encoder, d is the Euclidean distance and c_p is a prototype vector computed as:

$$c_p = \frac{1}{K}\sum_{i=1}^{K} h(r_i) \tag{2}$$

The DNN encoder $h(\cdot)$ is trained on a large keyword dataset, the Multilingual Spoken Words Corpus (MSWC) [25], using the triplet loss [26], which was experimentally observed to lead to higher accuracies than using a prototypical loss in open-set personalized voice command recognition [27].

In the case of a WASN, the master node is in charge of collecting and aggregating the similarity scores computed by the sensors into a network-level score y:

$$y = g(f(x^{s_1}), f(x^{s_2}), .., f(x^{s_T})) \tag{3}$$

where T is the total number of sensors, x^{s_1}, x^{s_2},.., x^{s_T} are the audio recordings from the sensors and $g(\cdot)$ is an aggregation function. Note that, in our formulation, the speech processing function $f(\cdot)$ runs locally on the sensors and, therefore, no raw audio data is transmitted out of the nodes.

We consider two aggregation logics: $g(\cdot) = \min()$, denoted as ***gmin***, forcing the network to rely on the most confident sensor, and $g(\cdot) = \text{mean}()$ (***gavg***) aimed at averaging out DNN miss-predictions. Given a moving-window type of processing (Eq. 1), the aggregation function is applied on the distance scores computed by every node at every time-stamp, i.e., the stride time of the moving window.

B. Self Incremental-Training

We apply self-incremental training to fine-tune the parameters of the DNN model $h(\cdot)$ based on a new set of data made available after deployment, namely the adaptation set. The new utterances correspond to keyword (i.e., the positive) and non-keyword (i.e., the negatives) audio data from multiple speakers without true label information.

Our solution, which is a multi-sensor extension of [12], generates pseudo-labels for the new adaptation data by means of the following thresholding rules:

$$\text{pseudo} - \text{positive}: \quad \min_t g(x^{s_1}, .., x^{s_T}) < Th_L \tag{4}$$

$$\text{pseudo} - \text{negative}: \quad \min_t g(x^{s_1}, .., x^{s_T}) > Th_N \tag{5}$$

The computed pseudo-label is propagated to the data of every sensor ($x^{s_1}, .., x^{s_T}$). In the precedent Eq. 4 and 5, the

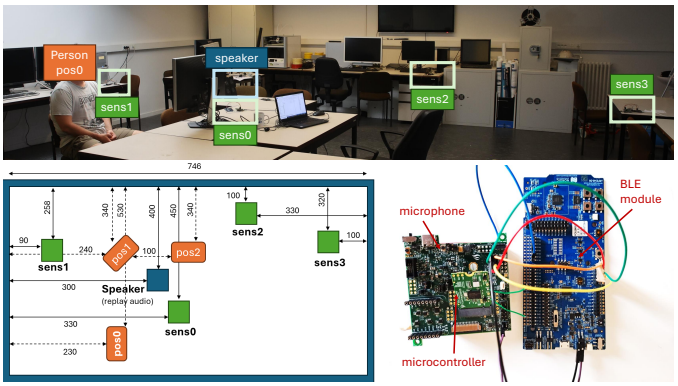


Fig. 1. Recording setup (top) and the map (bottom left), and a sensor prototype (bottom right). The coordinates of the 4 sensor nodes, the speaker and the 3 positions of the volunteers are expressed in cm.

$\min_t$ operator serves to identify the audio window including a keyword or a non-keyword within a speech utterance of variable duration. The threshold values are derived as in [12]:

$$Th(\tau) = dist^p + \tau \cdot (dist^n - dist^p) \tag{6}$$

where $dist^p$ and $dist^n$ are, respectively, the mean distance from the prototype c_p of the few (3 in our work) positive and negative examples. The thresholds used in Eq. 4 and 5 distinguish for the τ parameters. More in detail, we use $Th_L = Th(\tau = 0.3)$ and $Th_N = Th(\tau = 0.9)$, which led to the best accuracy in the reference work.

The final set of pseudo-labeled samples is used to incrementally fine-tune the audio encoder $h(\cdot)$. Likewise the pretraining step, a triple loss is used with a learning rate of 0.0001. The training runs for 4 epochs and an Adam optimizer is used.

C. Multi-Sensor Voice Command Dataset

We collected a multi-sensor dataset of voice commands using the Wireless Audio Sensor Network composed of 4 sensor nodes depicted in Fig. 1. A total of 20 volunteers were instructed to repeat five English voice commands (”Lights On”, ”Lights Off”, ”Music On”, ”Music Stop”, ”Next Song”) from three different positions (indicated in the map), for a total of 15 recordings per keyword. Only in a few cases, the number of samples is reduced to 14 after manual data cleaning. Additionally, we registered 15 per-speaker generic spoken utterances, e.g., ”Set an alarm to *7am*”, which are used as negative examples. Given a duration of 3 seconds per utterance, we collected a total of 1.5 h of audio per sensor. Additionally, the negative data were augmented with 4.4 h of recordings obtained by replaying a set of audio files from the test-clean and dev-clean sets of Librispeech [28]. Every recording was limited to 3 seconds, leading to a total amount of 5.9 h of audio per sensor in our multi-sensor dataset.

Every signal is acquired with an ultra-low power node composed by a GreenWaves Technologies GAP9 development board, which includes the GAP9 microcontroller and a Vesper VM301 microphone, and a Nordic NRF5340 Bluetooth Low Energy (BLE) board (illustrated in the bottom right of Fig. 1). The sampling rate of the sensor is set to 16 kHz. The microcontroller takes the raw audio in chunks of 125 ms, performs PCM-to-PDM demodulation and stores the audio data in a circular buffer in its on-chip memory. Every sensor’s BLE module communicates with an extra *central* node connected to a laptop. The latter node does not contain any microphone.

When a command is spoken, the central node broadcasts a stop signal to all the 4 sensor nodes, forcing the storage of the last 3 seconds of recorded audio. The stored audio is then sent to the laptop via UART. The stop mechanism implementation explains the misalignment between the sensors’ audio tracks: the GAP9 processor samples the ’stop signal’ from the BLE module every 125 ms, and this sampling frequency is not synchronized among sensors. Hence, our dataset is representative of a real-world WASN scenario where data recording and on-device processing tasks are asynchronous among sensors.

D. Experiment Protocol

We run speaker-specific personalized voice command experiments. At every run, we select one of the users and we personalize a speech model for the recognition of a target keyword. As an example, when the command ”Lights On” is chosen, 3 user-specific examples are used to compute the prototype (Eq. 2) while the other 11/12 recordings form the positive testset. The recordings of the other keywords from the same user (i.e., ”Lights Off”, ”Music On”, ”Music Stop”, ”Next Song”) together with the negative commands from all the speakers and the Librispeech recordings compose the negative testset.

On the other side, the adaptation set comprises the unlabeled samples for fine-tuning the DNN model, and it is disjoint from the testset. In our example, the ”Lights On” samples recorded from the other 19 users constitute the positive adaptation samples, while the remaining voice commands from this group of speakers form the negative adaptation set.

Every personalized voice command model is evaluated before and after the self-incremental training by computing the average per-speaker accuracy vs. the true labels of the test samples. The predicted class (”Keyword” and ”Non-Keyword”) is determined by converting the classifier score of Eq. 3 into a binary output. This is done by selecting a threshold value that leads to 0.5 false alarms per hour on the negative test set, which is a common practice for open-set voice command recognition [2], [16].

IV. Experimental Result

We run experiments with multiple lightweight DNN audio models taken from [7], which feature a number of parameters lower than 0.5 M. Every model processes a speech utterance of variable duration with a moving window of length 1 s and stride 0.125 s. The Mel-Frequency Cepstral Coefficients (MFCCs) are computed from the windowed audio using a filter length of 40 ms and a stride of 20 ms. We only feed the first 10 MFCCs to the model, i.e., a 47×10 feature map. Every experiment is repeated 10 times for a total of 10×5 (keywords) $\times$ 20 (speakers) = 1000 self-trainings per model.

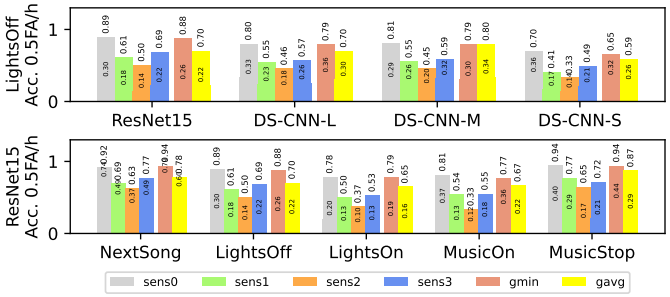


Fig. 2. Accuracy after self-training with oracle labeler for (upper) multiple models to recognize "Lights Off" and (bottom) ResNet15 to recognize multiple keywords. The accuracy before learning is marked with '/'.

TABLE II
AVG. ACCURACY OVER KEYWORDS AFTER SELF-INCREMENTAL TRAINING IN SINGLE- AND MULTI-SENSOR TEST SCENARIOS. WE COMPARE PSEUDO-LABELS GENERATED BY SINGLE OR MULTI-SENSOR AGGREGATED SCORES. THE **BEST** AND **SECOND-BEST** SCORES ARE HIGHLIGHTED.

test scenario	model	pret.	accuracy w/ pseudo-label agent						vs. pretrain		oracle
			sens0	sens1	sens2	sens3	gmin	gavg	min	max	
single-sensor sens0	ResNet15	0.40	**0.56**	**0.54**	0.51	0.52	0.53	0.48	+0.07	+0.16	0.87
	DS-CNN-L	0.46	**0.56**	**0.57**	0.53	0.53	**0.56**	0.50	+0.04	+0.11	0.83
	DS-CNN-M	0.46	**0.56**	0.51	0.48	0.50	**0.53**	0.48	+0.02	+0.10	0.79
	DS-CNN-S	0.42	**0.52**	**0.52**	0.49	0.51	**0.52**	0.49	+0.07	+0.10	0.73
single-sensor sens1	ResNet15	0.24	**0.43**	0.35	0.31	0.34	**0.39**	0.31	+0.06	+0.19	0.62
	DS-CNN-L	0.31	**0.47**	0.43	0.38	0.40	**0.44**	0.36	+0.05	+0.16	0.60
	DS-CNN-M	0.36	**0.47**	0.41	0.35	0.41	**0.44**	0.39	0.00	+0.11	0.53
	DS-CNN-S	0.27	**0.39**	0.37	0.31	0.34	**0.38**	0.33	+0.05	+0.12	0.52
multi-sensor gmin	ResNet15	0.41	**0.57**	**0.53**	0.50	0.51	**0.53**	0.48	+0.08	+0.16	0.86
	DS-CNN-L	0.48	**0.57**	**0.58**	0.53	0.53	**0.57**	0.52	+0.03	+0.10	0.83
	DS-CNN-M	0.49	**0.57**	0.52	0.50	0.52	**0.53**	0.50	+0.01	+0.08	0.77
	DS-CNN-S	0.42	**0.52**	**0.53**	0.49	0.51	**0.52**	0.48	+0.06	+0.11	0.72

First, we analyze the accuracies measured when an ideal labeler, namely an "oracle", is used to assign pseudo-labels to the samples of the adaptation set in single-sensor and multi-sensor scenarios. Every bar in Fig. 2 shows the accuracy after learning, averaged over the total number of speakers. We plot the scores for the "Lights Off" keyword for different models (upper) and the accuracy achieved with ResNet15 over different keywords (lower). In the single-sensor scenarios, indicated by 'sens0', 'sens1', 'sens2' and 'sens3' in the plot, the accuracy is tested only on the test samples captured from one sensor, and the detection score is computed with Eq. 1. On the multi-sensor data, we apply instead the *gmin* and *gavg* aggregation logic (Eq. 3) to obtain the classification output.

Generally, the accuracy increases after self-training, reaching top values for 'sens0' in the single-sensor case, which is the sensor closer to the speaking person providing a clearer signal than the other sensors. The final accuracy scales with respect to the model capacity (Fig. 2 top) and it is keyword-dependent (Fig. 2 bottom). As an example, the voice commands "Lights On" and "Music On" are the hardest to recognize, with a final accuracy $> 10\%$ lower than the other commands (sens0 scenario). Among all the considered keywords, ResNet15 reaches an average score of 88% (sens0 testset), $+46.8\%$ than the pretrained.

In the multi-sensor scenario, the *gmin* logic consistently leads to better results than *gavg*, suggesting that it is most convenient to rely on the most-confident sensor at every time-stamp. This aggregated score leads to a final accuracy of only 1% worse than the sens0 scenario, i.e. the signal with the highest signal-to-noise ratio, but +22.5%, +34.5%, or +21.2% than only using, respectively, sens1, sens2 or sens3.

Tab. II plots the accuracy scores after the self-incremental training when using pseudo-labels. For every reported test scenario and DNN model, we compare pseudo-labeling agents that rely on single- or multi-sensor data. The multi-sensor predictor with *gmin* policy and the sens0 classifier lead to the best accuracies (2% of mean difference), reflecting the trend observed with the oracle. At the same time, the multi-sensor setting leads to up to $+4.7\%$ accuracy than other reported single-sensor labelers.

Overall, the self-incremental training leads to accuracy improvements vs. the pretrained models, with peaks of $+19\%$ and $+15\%$ when relying on, respectively, single-sensor or multi-sensor labeling scores. Still, the process is not able to bridge the gap with respect to the oracle because of the pseudo-labeling errors. For ResNet15 with sens0 or *gmin* labelers, the correct positive samples are, respectively, 65% and 62%, while the correct negative data are 93% and 95%.

Lastly, Fig. 3 illustrates the Top5 negative samples closer to the keyword prototypes. From this analysis, we observe that similar keywords are still close in the feature space after self-training (e.g. 'Lights On' and 'Lights Off'). Therefore, we argue that the process mainly achieves more compact embedding clusters than the pretrained models, leading to improvements in final accuracy.

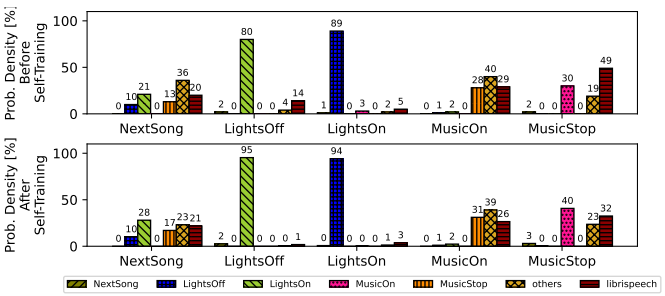


Fig. 3. Distribution density of the Top5 negative samples closer to the *keyword* prototype, before (upper) and after (bottom) self-training. The plot refers to DS-CNN-L with *gmin* labeler and *gmin* test scenario.

V. DISCUSSION AND CONCLUSION

This paper reported a baseline analysis of a self-incremental training process applied to a new multi-sensor dataset of voice commands. By leveraging a global score aggregated among several sensors, we showed an improvement vs. the baseline models of up to $+19\%$ and a final accuracy of up to $+4.7\%$ over single sensor setups. We finally argue that our new dataset and experimental protocol can serve as a benchmark for developing novel self-training methods for resource-constrained devices. A future ambition is to narrow the accuracy gap vs. the ideal oracle baselines or to reach the high-accuracy of KWS models trained offline on pre-collected datasets, i.e. not personalized on-device [1].

REFERENCES

[1] Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra, "Hello edge: Keyword spotting on microcontrollers," *arXiv preprint arXiv:1711.07128*, 2017.

[2] Byeonggeun Kim, Mingu Lee, Jinkyu Lee, Yeonseok Kim, and Kyuwoong Hwang, "Query-by-example on-device keyword spotting," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 532–538.

[3] Yangbin Chen, Tom Ko, and Jianping Wang, "A Meta-Learning Approach for User-Defined Spoken Term Classification with Varying Classes and Examples," in *Proc. Interspeech*, 2021, pp. 4224–4228.

[4] Archit Parnami and Minwoo Lee, "Few-shot keyword spotting with prototypical networks," in *2022 7th International Conference on Machine Learning Technologies (ICMLT)*, New York, NY, USA, 2022, p. 277–283, Association for Computing Machinery.

[5] Jaemin Jung, Youkyum Kim, Jihwan Park, Youshin Lim, Byeong-Yeol Kim, Youngjoon Jang, and Joon Son Chung, "Metric learning for user-defined keyword spotting," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[6] Seunghan Yang, Byeonggeun Kim, Inseop Chung, and Simyung Chang, "Personalized Keyword Spotting through Multi-task Learning," in *Proc. Interspeech*, 2022, pp. 1881–1885.

[7] Manuele Rusci and Tinne Tuytelaars, "On-device customization of tiny deep learning models for keyword spotting with few examples," *IEEE Micro*, 2023.

[8] Daniel S. Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V. Le, "Improved Noisy Student Training for Automatic Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 2817–2821.

[9] Sameer Khurana, Niko Moritz, Takaaki Hori, and Jonathan Le Roux, "Unsupervised domain adaptation for speech recognition via uncertainty driven self-training," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6553–6557.

[10] Long Mai and Julie Carson-Berndsen, "Unsupervised domain adaptation for speech recognition with unsupervised error correction," in *Proc. Interspeech 2022*, 2022, pp. 5120–5124.

[11] Hyun-Jin Park, Pai Zhu, Ignacio Lopez Moreno, and Niranjan Subrahmanya, "Noisy Student-Teacher Training for Robust Keyword Spotting," in *Proc. Interspeech 2021*, 2021, pp. 331–335.

[12] Manuele Rusci, Francesco Paci, Marco Fariselli, Eric Flamand, and Tinne Tuytelaars, "Self-learning for personalized keyword spotting on ultra-low-power audio sensors," *IEEE Internet of Things Journal*, 2024.

[13] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[14] Guoguo Chen, Carolina Parada, and Tara N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5236–5240.

[15] Théodore Bluche, Maël Primet, and Thibault Gisselbrecht, "Small-footprint open-vocabulary keyword spotting with quantized lstm networks," *arXiv preprint arXiv:2002.10851*, 2020.

[16] Alice Coucke, Mohammed Chlieh, Thibault Gisselbrecht, David Leroy, Mathieu Poumeyrol, and Thibaut Lavril, "Efficient keyword spotting using dilated convolutions and gating," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6351–6355.

[17] Théodore Bluche and Thibault Gisselbrecht, "Predicting detection filters for small footprint open-vocabulary keyword spotting," in *Interspeech 2020*, 2020, pp. 2552–2556.

[18] Gert Dekkers, Steven Lauwereins, Bart Thoen, Mulu Weldegebreal Adhana, Henk Brouckxon, Bertold Van den Bergh, Toon Van Waterschoot, Bart Vanrumste, Marian Verhelst, and Peter Karsmakers, "The sins database for detection of daily activities in a home environment using an acoustic sensor network.," in *DCASE*, 2017, pp. 32–36.

[19] Nancy Bertin, Ewen Camberlein, Emmanuel Vincent, Romain Lebarbenchon, Stéphane Peillon, Éric Lamandé, Sunit Sivasankaran, Frédéric Bimbot, Irina Illina, Ariane Tom, et al., "A french corpus for distant-microphone speech processing in real homes," in *Interspeech 2016*, 2016.

[20] Jon Barker, Shinji Watanabe, Emmanuel Vincent, and Jan Trmal, "The Fifth 'CHiME' Speech Separation and Recognition Challenge: Dataset, Task and Baselines," in *Proc. Interspeech 2018*, 2018, pp. 1561–1565.

[21] Maarten Van Segbroeck, Zaid Ahmed, Ksenia Kutsenko, Cirenia Huerta, Tinh Nguyen, Björn Hoffmeister, Jan Trmal, Maurizio Omologo, and Roland Maas, "Dipco - dinner party corpus," in *Interspeech 2020*, 2019.

[22] Shinji Watanabe, Michael Mandel, Jon Barker, Emmanuel Vincent, Ashish Arora, Xuankai Chang, Sanjeev Khudanpur, Vimal Manohar, Daniel Povey, Desh Raj, et al., "Chime-6 challenge: Tackling multi-speaker speech recognition for unsegmented recordings," in *CHiME 2020-6th International Workshop on Speech Processing in Everyday Environments*, 2020.

[23] Gert Dekkers, Fernando Rosas, Toon van Waterschoot, Bart Vanrumste, and Peter Karsmakers, "Dynamic sensor activation and decision-level fusion in wireless acoustic sensor networks for classification of domestic activities," *Information Fusion*, vol. 77, pp. 196–210, 2022.

[24] Shimaa Ahmed, Ilia Shumailov, Nicolas Papernot, and Kassem Fawaz, "Towards more robust keyword spotting for voice assistants," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 2655–2672.

[25] Mark Mazumder, Sharad Chitlangia, Colby Banbury, Yiping Kang, Juan Manuel Ciro, Keith Achorn, Daniel Galvez, Mark Sabini, Peter Mattson, David Kanter, et al., "Multilingual spoken words corpus," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[26] Roman Vygon and Nikolay Mikhaylovskiy, "Learning efficient representations for keyword spotting with triplet loss," in *Speech and Computer: 23rd International Conference, SPECOM 2021, St. Petersburg, Russia, September 27–30, 2021, Proceedings 23*. Springer, 2021, pp. 773–785.

[27] Manuele Rusci and Tinne Tuytelaars, "Few-Shot Open-Set Learning for On-Device Customization of KeyWord Spotting Systems," in *Proc. Interspeech*, 2023.

[28] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.