# Speaker Diarization as a Fully Online Bandit Learning Problem in MiniVox

**Baihan Lin** BAIHAN.LIN@COLUMBIA.EDU
*Columbia University*

**Xinxin Zhang** XZ3149@NYU.EDU
*New York University*

**Editors:** Vineeth N Balasubramanian and Ivor Tsang

## Abstract

We propose a novel machine learning framework to conduct real-time multi-speaker diarization and recognition without prior registration and pretraining in a fully online learning setting. Our contributions are two-fold. First, we propose a new benchmark to evaluate the rarely studied fully online speaker diarization problem. We build upon existing datasets of real world utterances to automatically curate *MiniVox*, an experimental environment which generates infinite configurations of continuous multi-speaker speech stream. Second, we consider the practical problem of online learning with episodically revealed rewards and introduce a solution based on semi-supervised and self-supervised learning methods. Additionally, we provide a workable web-based recognition system which interactively handles the cold start problem of new user's addition by transferring representations of old arms to new ones with an extendable contextual bandit. We demonstrate that our proposed method obtains robust performance in the online MiniVox framework given either cepstrum-based representations or deep neural network embeddings.

**Keywords:** Speaker diarization, online learning, semi-supervised learning, self-supervision, contextual bandit, reinforcement learning

## 1. Introduction

Speaker diarization is a task to label an audio or video recording with the identity of the speaker at each given time stamp. In each time window, speaker recognition is a performed to distinguish the identity of the person who is speaking in a mixed-speaker signal based on voice characteristics with two essential steps: registration and identification (Tirumala et al., 2017). The registration step computes a voiceprint model of each speaker given his or her acoustic samples, while the identification matches existing voiceprint model with real-time audio signal. In laboratory setting, the state-of-the-art approaches usually emphasize the registration step with deep networks (Snyder et al., 2018) trained on large-scale speaker profile dataset (Nagrani et al., 2017). However, in real life, requiring all users to complete voiceprint registration before a multi-speaker teleconference is hardly a preferable way of system deployment. Dealing with this challenge, speaker diarization is the task to partition an audio stream into homogeneous segments according to the speaker identity (Anguera et al., 2012). Recent advancements have enabled (1) contrastive audio embedding extractions such as Mel Frequency Cepstral Coefficients (MFCC, Hasan et al., 2004), i-

vectors (Shum et al., 2013), x-vectors (Snyder et al., 2018) and d-vectors (Wang et al., 2018); (2) effective clustering modules such as Gaussian mixture models (GMM, Zajíc et al., 2017), mean shift (Senoussaoui et al., 2013), Kmeans and spectral clustering (Wang et al., 2018) and supervised Bayesian non-parametric methods (Fox et al., 2011; Zhang et al., 2019); and (3) reasonable resegmentation modules such as Viterbi and factor analysis subspace (Kenny et al., 2010; Sell and Garcia-Romero, 2015). In this work, we propose a new paradigm to consider the speaker diarization as a fully online learning problem of the speaker recognition task: it combines the embedding extraction, clustering and resegmentation into the same problem as an online decision making problem.

**Why is this online learning problem different?** The state-of-the-art speaker diarization systems usually require large datasets to train their audio extraction embeddings and clustering modules, especially the ones with deep neural networks and Bayesian non-parametric models. In many real-world applications in developing countries, however, the training set can be limited and hard to collect. Since these modules are pretrained, applying them to out-of-distribution environments can be problematic. For instance, an intelligent system trained with American elder speaker data might find it hard to generalize to a Japanese children diarization task because both the acoustic and contrastive features are different. To tackle this problem, we want the system to learn continually. To push this problem to the extreme, we are interested in a fully online learning setting, where not only the examples are available one by one, the agent receives no pretraining from any training set before deployment, and learns to detect speaker identity on the fly through reward feedbacks. To the best of our knowledge, this work is the first to consider diarization as a fully online learning problem with bandit feedback. Through this work, we aim to understand the extent to which diarization can be solved as merely an online learning problem and whether traditional online learning algorithms such as the contextual bandits (Langford and Zhang, 2007) (widely used in applications like user modeling (Lin et al., 2020a), phenotyping (Lin et al., 2020b) and epidemic control (Lin and Bouneffouf, 2021)) can be a practical solution.

**What is a preferable online speaker diarization system?** A preferable artificial intelligence (AI) engine for such a realistic speaker recognition and diarization system should (1) not require user registrations before its deployment, (2) allow new user to be registered into the system in real-time, (3) transfer voiceprint information from old users to new ones, and (4) be up running without pretraining on large amount of data in advance. While attractive, assumption (4) introduces an additional caveat that the labeling of the user profiles happens purely on the fly, trading off models pretrained on big data with the user directly interacting with the system by correcting the agent as labels. To tackle these challenges, we formulate this problem into an interactive learning model with cold-start arms and episodically revealed rewards (users can either reveal no feedback, approving by not intervening, or correcting it).

**Why do we need a new benchmark?** Traditional datasets in the speaker diarization task are limited: CALLHOME American English (Canavan et al., 1997) and NIST RT-03 English CTS (Martin and Przybocki, 2000) contain limited number of utterances recorded in controlled conditions. For online learning experiments, a learn-from-scratch agent usually needs a large length of data stream to reach a comparable result. Large scale speaker recognition dataset like VoxCeleb (Nagrani et al., 2017, 2020) and Speakers in the Wild (SITW, McLaren et al., 2016) contain thousands of speaker utterances recorded in various

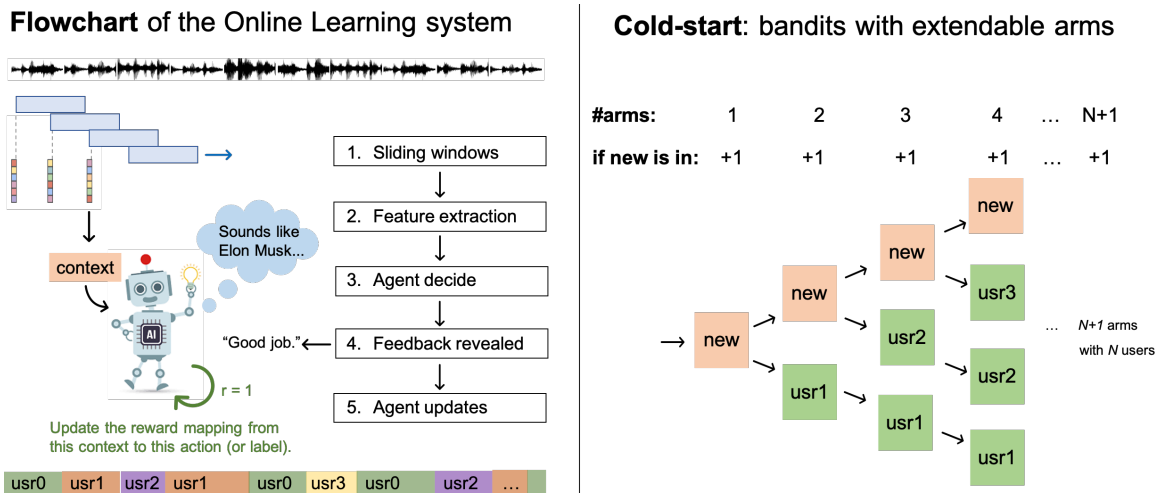


Figure 1: (A) **Flowchart** of the online speaker diarization learning task; (B) Arm expansion process of bandits for cold-start problem.

challenging multi-speaker acoustic environments, but they are usually only used to pretrain diarization embeddings. Recently, DIHARD (Sell et al., 2018; Ryant et al., 2019) is an effective diarization dataset which includes an evaluation set drawn from a diverse set of challenging domains, but it is not clear how to create an infinite long data stream to evaluate a purely online setting. Here we propose a new benchmark called *MiniVox*, which can transform any large scale speaker identification dataset into infinitely long online audio streams with various configurations.

To the best of our knowledge, this is the first approach to apply the *Bandit* problem to the speaker diarization task. We build upon the Linear Upper Confidence Bound algorithm (LinUCB, Li et al., 2010) and propose a semi-supervised learning variant to account for the fact that the rewards are entirely missing in many episodes. For each episode without feedbacks, we apply a self-supervision process to assign a pseudo-action upon which the reward mapping is updated. Finally, we generate new arms by transferring learned arm parameters to similar profiles given user feedback signals. The contributions of our work are summarized as follows:

- We formulate the speaker diarization task as an online bandit learning task.
- We create a semi-supervised solution to the sparsely rewarded online learning problem.
- We create a new benchmark with infinite sequences of multi-speaker utterances.
- In the benchmark, our solution outperforms the standard online learning methods.
- Our system is interactive, register-free, real-time and can be completely web-based.

**Algorithm 1** Online Learning with Episodic Rewards

1: **for** t = 1,2,3,$\cdots$, T **do**
2: $(\mathbf{x}(t), \mathbf{r}(t))$ is drawn according to $\mathbb{P}_{x,r}$
3: Context $\mathbf{x}(t)$ is revealed to the player
4: Player chooses an action $a_t = \pi_t(\mathbf{x}(t))$
5: Feedback $r_{a_t,t}(t)$ for the arm $a_t$ is episodically revealed
6: Player updates its policy $\pi_t$

## 2. Background and Problem Setting

**The Bandit Problem.** In online learning setting, data become available in a sequential order and later used to update the best predictor for future data or reward associated with the data features. In many cases, the reward feedback is the only source where the online learning agent can effectively learn from the sequential past experience. This problem is especially important in the field of sequential decision making where the agent must choose the best possible action to perform at each step to maximize the cumulative reward over time. One key challenge is to obtain an optimal trade-off between the exploration of new actions and the exploitation of the possible reward mapping from known actions. This framework is usually formulated as the *Bandit* problem where each arm of the bandit corresponds to an unknown (but usually fixed) reward probability distribution (Lai and Robbins, 1985), and the agent selects an arm to play at each round, receives a reward feedback and updates accordingly. An especially useful variant of Bandit is the *Contextual Bandit*, where at each step, the agent observes an $N$-dimensional *context*, or *feature* vector before selecting an action. Theoretically, the ultimate goal of Contextual Bandit is to learn the relationship between the rewards and the context vectors so as to make better decisions given the context (Agrawal and Goyal, 2013).

## 3. The Fully Online Learning Setting

### 3.1. Online Learning with Episodic Reward

Algorithm 1 presents at a high-level our problem setting of our interactive learning system for speaker diarization, where $x(t) \in \mathbb{R}^d$ is a vector describing the context $C$ at time $t$, $r_{a,t}(t) \in [0,1]$ is the reward of action $a$ at time $t$, and $r(t) \in [0,1]^N$ denotes a vector of rewards for all arms at time $t$. $\mathbb{P}_{x,r}$ denotes a joint probability distribution over $(x,r)$, and $\pi : C \to A$ denotes a policy. Unlike traditional setting, in step 5 we have the rewards revealed in an episodic fashion (i.e. sometimes there are feedbacks of rewards being 0 or 1, sometimes there are no feedbacks of any kind). We consider our setting online semi-supervised learning (Yver, 2009), where agents continually learn from both labeled and unlabeled data.

## 4. Proposed Online Learning Solution

### 4.1. Contextual Bandits with Extendable Arms

In an ideal online learning scenario without oracle, we start with a single arm, and when new labels arrive new arms are then generated accordingly. This problem is loosely modelled

**Algorithm 2** BerlinUCB

1: **Initialize** $c_t \in \mathbb{R}_+, \mathbf{A}_a \leftarrow \mathbf{I}_d, \mathbf{b}_a \leftarrow \mathbf{0}_{d\times 1} \forall a \in \mathcal{A}_t$
2: **for** t = 1,2,3,$\cdots$, T **do**
3: Observe features $\mathbf{x}_t \in \mathbb{R}^d$
4: **for all** $a \in \mathcal{A}_t$ **do**
5: $\hat{\theta}_a \leftarrow \mathbf{A}_a^{-1}\mathbf{b}_a$
6: $p_{t,a} \leftarrow \hat{\theta}_a^\top \mathbf{x}_t + c_t\sqrt{\mathbf{x}_t^\top \mathbf{A}_a^{-1}\mathbf{x}_t}$
7: Choose arm $a_t = arg\,max_{a\in\mathcal{A}_t} p_{t,a}$
8: **if** the background revealed the feedbacks **then**
9: Observe feedback $r_{a_t,t}$
10: $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_t\mathbf{x}_t^\top$
11: $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_{a_t,t}\mathbf{x}_t$
12: **elif** the background revealed NO feedbacks **then**
13: **if** use self-supervision feedback
14: $r' = [a_t == predict(\mathbf{x}_t)]$ % clustering modules
15: $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r'\mathbf{x}_t$
16: **else** % ignore self-supervision signals
17: $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_t\mathbf{x}_t^\top$

by the bandits with infinitely many arms (Berry et al., 1997). For our specific application of speaker registration process, we apply the arm expansion process outlined in Fig 1b: starting from a single arm (for the "new" action), if a feedback confirms a new addition, a new arm is initialized and stored (more details on the handling of growing arms can be found in later sections).

### 4.2. Episodically Rewarded LinUCB

We build upon the Linear Upper Confidence Bound algorithm (LinUCB, Li et al., 2010) and propose to use the Background Episodically Rewarded LinUCB (BerlinUCB, Lin, 2020), a semi-supervised and self-supervised online contextual bandit which updates the context representations and reward mapping separately given the state of the feedbacks being present or missing, which is shown to be empirically beneficial in many online learning tasks (**?**). As in Algorithm 2, the steps 1 through 12 of BerlinUCB are the same as the standard LinUCB algorithm (Li et al., 2010), and in case of a missing reward, we introduce the steps 13 through 20 as the alternative strategy. We assume that: (1) when there are feedbacks available, the feedbacks are genuine, assigned by the oracle, and (2) when the feedbacks are missing (not revealed by the background), it is either due to the fact that the action is preferred (no intervention required by the oracle, i.e. with an implied default rewards), or that the oracle didn't have a chance to respond or intervene (i.e. with unknown rewards). Especially in the Step 14, when there is no feedbacks, we assign the context $\mathbf{x}_t$ to a class $a'$ (an action arm) with the self-supervision given the previous labelled context history ("*predict*" function). Since we don't have the actual label for this context, we only update the reward mapping parameter $\mathbf{b}_{a'}$ and leave the covariance matrix $\mathbf{A}_{a'}$ untouched. This additional usage of unlabelled data (or unrevealed feedback) is especially important in BerlinUCB.

### 4.3. Self-Supervision and Semi-Supervision Modules

We construct our self-supervision modules given the cluster assumption of the semi-supervision problem: the points within the same cluster are more likely to share a label. As shown in many work in modern speaker diarization, clustering algorithms like Gaussian Mixture Models (GMM, Zajíc et al., 2017) and spectral clustering (Wang et al., 2018) are especially powerful unsupervised modules, especially in their offline versions. Their online variants, however, often perform poorly (Zhang et al., 2019). Nonetheless, we choose three popular clustering algorithms as the self-supervision: GMM, Kmeans and K-nearest neighbors (KNN), all in their online versions.

### 4.4. Complete Engine for Online Speaker Diarization

To adapt our BerlinUCB algorithm to the specific application of speaker recognition, we first define our actions. There are three major classes of actions: an arm "New" to denote that a new speaker is detected, an arm "No Speaker" to denote that no one is speaking, and N different arms "User n" to denote that user n is speaking. Table 1 presents the reward assignment given four types of feedbacks. Note that we assume that when the agent correctly identifies the speaker (or no speaker), the user (as the feedback dispenser) should send no feedbacks to the system by doing nothing. In another word, in an ideal scenario when the agent does a perfect job by correctly identifying the speaker all the time, we are not necessary to be around to correct it anymore (i.e. truly feedback free). As we pointed out earlier, this could be a challenge earlier on, because other than implicitly approving the agent's choice, receiving no feedbacks could also mean the feedbacks are not revealed properly (e.g. the human oracle took a break). Furthermore, we note that when "No Speaker" and "User n" arms are correctly identified, there is no feedback from us the human oracle (meaning that these arms would never have learned from a single positive reward if we don't use the "None" feedback iterations at all!). The semi-supervision by self-supervision step is exactly tailored for a scenario like this, where the lack of revealed positive reward for "No Speaker" and "User n" arms is compensated by additional training of reward mapping $\mathbf{b}_{a_t}$ if context $\mathbf{x}_t$ is assigned to the right arm.

To tackle the cold start problem, the agent grows it arms in the following fashion: the agent starts with two arms, "No Speaker" and "New"; if it is actually a new speaker speaking, we have the following three conditions: (1) if "New" is chosen, the user approves this arm by giving it a positive reward (i.e. clicking on it) and the agent initializes a new arm called "User $N$" and update $N = N + 1$ (where $N$ is the number of registered speakers at the moment); (2) if "No Speaker" is chosen, the user disapproves this arm by giving it a zero reward and clicking on the "New" instead), while the agent initializes a new arm; (3) if one of the user arms is chosen (e.g. "User 5" is chosen while in fact a new person is speaking), the agent copies the wrong user arm's parameters to initialize the new arm, since the voiceprint of the mistaken one might be beneficial to initialize the new profile. Thus, we can transfer what has been learned for a similar context to the new arm. (Potential problems might occur if the number of users grows steadily through misclassifications. In future work, we will investigate possible branch pruning strategies and the processing of very sparse reward feedback.)

| Feedback types | (+, +) | (+, -) | (-, +) | *None* |
|---|---|---|---|---|
| New | $r = 1$ | $r = 0$ | - | |
| No Speaker | - | $r = 0$ | $r = 0$ | Alg. 2 Step 12 |
| User n | - | $r = 0$ | $r = 0$ | |

Table 1: Possible algorithm routes given no feedbacks, or a feedback telling the agent that the correct label is $a*$. (+, +) means that the agent guessed it right by choosing the right arm; (+, -) means that the agent chose this arm incorrectly, since the correct one is another arm; (-, +) means that the agent didn't choose this arm, while it turned out to be the correct one. "-" marks scenarios not applicable.

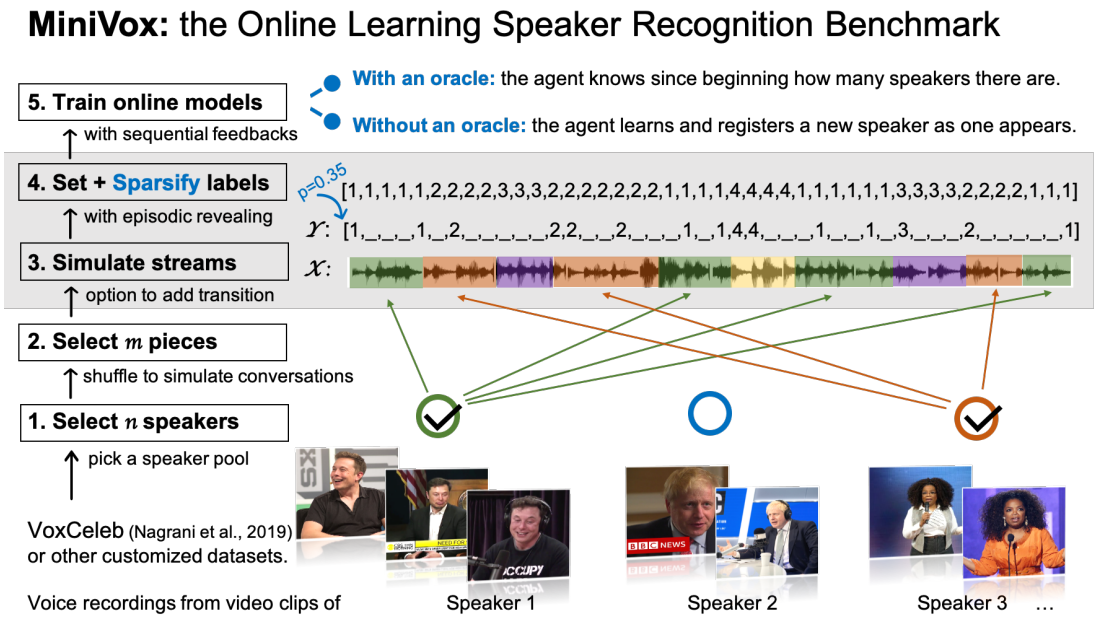


Figure 2: The ***MiniVox* benchmark** and its preprocessing steps

## 5. The *MiniVox* Benchmark

*MiniVox* is an automatic framework to transform any speaker-labelled dataset into continuous speech datastream with episodically revealed label feedbacks. This benchmark is specifically designed to simulate two real-world challenges in interactive systems: (1) the speech data in real life is never truncated into pieces for an intelligent system to classify, and (2) the reward feedbacks (i.e. telling an intelligent system that it is incorrect) is never as timely and complete. In the working example we evaluate in this paper, we apply the MiniVox on the test dataset of VoxCeleb (Nagrani et al., 2020) to create randomly generated multi-speaker "conversations" in continuous streams. Unlike other dataset and evaluation pipelines in speaker recognition (e.g. VoxCeleb (Nagrani et al., 2017), SITW (McLaren et al., 2016)) and speaker diarization (e.g. CALLHOME American English (Canavan et al., 1997), NIST RT-03 English CTS (Martin and Przybocki, 2000)), *MiniVox* environment can

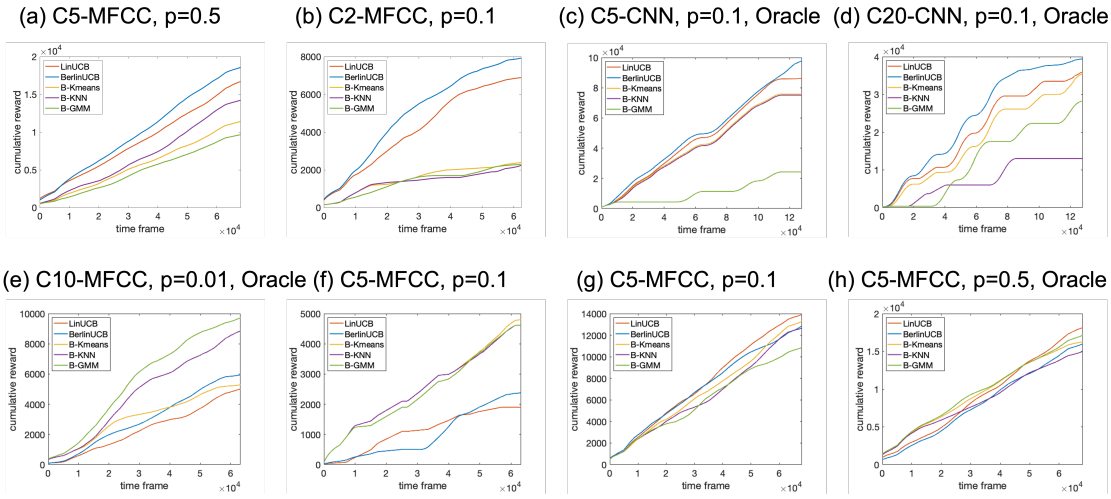


Figure 3: **Example reward curves** where **(a, b, c, d)** BerlinUCB is the best; **(e, f)** the self-supervision is the best; **(g, h)** LinUCB is the best.

generate infinite length of sequences of multi-speaker conversations, and reveal rewards in an episodic fashion.

Since our online learning setting assumes learning voiceprints without any previous training data at all, *MiniVox*'s flexibility in length and configuration is especially important (detailed in Figure 2). *MiniVox* can generate infinite length of multi-user utterances: given a pool of single-speaker-annotated utterances, randomly concatenate multiple pieces with a chosen number of speakers and a desired length. The reward stream is then sparsified with a parameter $p$ as the percentage of time a feedback is revealed. There are two scenarios that we can evaluate in MiniVox: if we assume there is an oracle, the online learning model is given the fixed number of the speakers in the stream; if we assume there is no oracle, the online learning model will start from zero speaker and then gradually discover and register new speakers for future identification and diarization.

## 6. Empirical Evaluation

**Experimental Setup.** We apply MiniVox on VoxCeleb (Nagrani et al., 2017) to generate three data streams with 5, 10 and 20 speakers to simulate real-world conversations. We extract two types of features (more details in later sections) and evaluate them in two scenarios (with or without oracle). The reward streams are sparsified given a revealing probability of 0.5, 0.1 and 0.01. In summary, we evaluate our models in a combinatorial total of 3 speaker numbers $\times$ 3 reward revealing probabilities $\times$ 2 feature types $\times$ 2 test scenarios $=$ 36 online learning environments. The online learning timescale range from $\sim$12000 to $\sim$60000 timeframes, with a frame shift of 10 ms. For notation of a specific MiniVox, we would denote "MiniVox C5-MFCC-60k" as a MiniVox environment with 5 speakers ranging 60k time frames using MFCC as features. We evaluate two scenarios in

*MiniVox*: if we assume there is an oracle, the online learning model is given the fixed number of the speakers in the stream; if we assume there is no oracle, the online learning model will start from zero speaker and then gradually discover and register new speakers for future identification and diarization.

**Metrics.** To evaluate performance in the above MiniVox environments, we report Diarization Error Rates (DER), the standard introduced in the NIST Rich Transcription 2009 (RT-09). In addition, as a common metric in online learning literature, we also record the cumulative reward: at each frame, if the agent correctly predicts a given speaker, the reward is counted as +1 (regardless of the agent's observation).

**Baselines.** We compare 5 agents: The baseline, *LinUCB* is the contextual bandit with extendable arms proposed in section 4. *BerlinUCB* is our standard contextual bandit model designed for sparse feedbacks without the self-supervision modules. To test the effect of self-supervision, we introduce three clustering modules in BerlinUCB (Alg 2 Step 14) denoted *B-Kmeans*, *B-KNN*, and *B-GMM*, whose clustering units are randomly initialized and updated online (with K=5). Lastly, we have a *random* agent. In the oracle-free case, the bandit agent can "explore" by randomly selecting from the "new" arm and the registered user arms, suggesting a possibility of going to infinitely (and incorrectly) many arms of registered users. In the oracle-free case, the bandit agent can "explore" by randomly selecting from the "new" arm and the registered user arms, suggesting a possibility of going to infinitely (and incorrectly) many arms of registered users.

**Feature Embeddings: MFCC and Neural Networks.** We utilize two feature embeddings for our evaluation: MFCC (Hasan et al., 2004) and a Convolutional Neural Network (CNN). We utilize the same CNN architecture as the VGG-M (Chatfield et al., 2014) used in VolCeleb evaluation (Nagrani et al., 2017). It takes the spectrogram of an utterance as the input, and generates a feature vector of 1024 in layer fc8 (table 4 in (Nagrani et al., 2017) for details about this CNN).

For both the MFCC and CNN, we adopted the same input feature extraction procedure. As the standard in speaker recognition and diarization, we converted all audio clips to single-channel 16-bit streams at a 16kHz sampling rate for consistency. In this format, we generate spectrograms in a sliding window fashion using a hamming window of width 25ms and step 10ms, giving a spectrogram of size 512 x 100 for each second of speech. We applied additional normalization on the mean and variance on every frequency bin of the spectrum. At each time point, the MFCC features and CNN features were then individually extracted given a sliding window segment of 500 timeframes from the original utterance stream.

**Why don't we use deep embeddings with pretraining?** Although more complicated embedding extraction modules such as i-vectors (Shum et al., 2013), x-vectors (Snyder et al., 2018), or d-vectors (Wang et al., 2018) are popular diarization models, they require extensive pretraining on big datasets to build a deep embedding (Garcia-Romero et al., 2017) of the voice profile in an offline setting, which is both contradictory to and a much easier task setting to our fully online problem setting, and therefore, out of the scope of this paper. In our online learning problem setting, the LinUCB algorithm is the well-established state-of-the-art in the machine learning community.

**Why do we still include a pretrained CNN embedding?** Indeed, if our end goal is to let the system learn from scratch without pretraining, why do we consider it in our evaluation? The reason why we also included this CNN embedding as part of the evaluation

Table 2: Diarization Error Rate (%) in MiniVox **without** Oracle

| | MiniVox C5-MFCC-60k | | | MiniVox C10-MFCC-60k | | | MiniVox C20-MFCC-60k | | |
|---|---|---|---|---|---|---|---|---|---|
| | $p = 0.5$ | $p = 0.1$ | $p = 0.01$ | $p = 0.5$ | $p = 0.1$ | $p = 0.01$ | $p = 0.5$ | $p = 0.1$ | $p = 0.01$ |
| BerlinUCB | **71.81** | 80.03 | 82.38 | **82.46** | **85.31** | **89.26** | **88.62** | **87.02** | 92.79 |
| LinUCB | 74.74 | **78.71** | 79.30 | 84.36 | 86.73 | 93.36 | 91.35 | 88.94 | **88.46** |
| B-Kmeans | 82.82 | 79.15 | **77.39** | 91.15 | 92.58 | 96.68 | 95.19 | 95.99 | 96.96 |
| B-KNN | 78.71 | 80.62 | **77.39** | 89.73 | 90.05 | 96.68 | 93.43 | 95.99 | 96.79 |
| B-GMM | 85.32 | 83.41 | 87.67 | 90.21 | 94.63 | 98.42 | 92.79 | 96.31 | 97.76 |
| | MiniVox C5-CNN-12k | | | MiniVox C10-CNN-12k | | | MiniVox C20-CNN-12k | | |
| | $p = 0.5$ | $p = 0.1$ | $p = 0.01$ | $p = 0.5$ | $p = 0.1$ | $p = 0.01$ | $p = 0.5$ | $p = 0.1$ | $p = 0.01$ |
| BerlinUCB | **17.42** | **32.03** | 65.16 | **42.77** | **57.41** | **74.02** | **41.72** | **59.06** | 83.28 |
| LinUCB | 17.81 | 32.73 | **58.98** | 49.55 | 68.57 | 81.16 | 51.56 | 83.52 | **74.84** |
| B-Kmeans | 28.83 | 63.67 | 82.58 | 60.89 | 70.89 | 99.55 | 72.03 | 75.31 | 99.53 |
| B-KNN | 28.36 | 82.58 | 82.58 | 60.89 | 82.05 | 99.55 | 72.03 | 74.06 | 99.53 |
| B-GMM | 99.61 | 99.61 | 99.69 | 99.20 | 93.57 | 99.64 | 87.73 | 81.09 | 83.28 |

Table 3: Diarization Error Rate (%) in MiniVox **with** Oracle

| | MiniVox C5-MFCC-60k | | | MiniVox C10-MFCC-60k | | | MiniVox C20-MFCC-60k | | |
|---|---|---|---|---|---|---|---|---|---|
| | $p = 0.5$ | $p = 0.1$ | $p = 0.01$ | $p = 0.5$ | $p = 0.1$ | $p = 0.01$ | $p = 0.5$ | $p = 0.1$ | $p = 0.01$ |
| BerlinUCB | 74.89 | 77.24 | 86.93 | 88.31 | **90.21** | 95.89 | 92.31 | 94.55 | 96.31 |
| LinUCB | **72.83** | 78.12 | **76.80** | **84.99** | 91.63 | 97.00 | **89.10** | 93.43 | **95.67** |
| B-Kmeans | 75.33 | 78.27 | 83.11 | 87.84 | 91.47 | **91.94** | 92.95 | 95.67 | 96.96 |
| B-KNN | 77.39 | 77.97 | 83.99 | 86.73 | 85.78 | 92.58 | 91.83 | 92.47 | 97.44 |
| B-GMM | 74.16 | **76.21** | 77.24 | 88.94 | 84.52 | 92.58 | 95.19 | **91.99** | 97.44 |
| | MiniVox C5-CNN-12k | | | MiniVox C10-CNN-12k | | | MiniVox C20-CNN-12k | | |
| | $p = 0.5$ | $p = 0.1$ | $p = 0.01$ | $p = 0.5$ | $p = 0.1$ | $p = 0.01$ | $p = 0.5$ | $p = 0.1$ | $p = 0.01$ |
| BerlinUCB | **17.27** | **22.19** | 66.02 | **45.18** | **65.27** | 79.38 | 58.75 | **68.98** | 88.83 |
| LinUCB | 17.73 | 32.73 | **58.98** | 50.00 | 72.14 | **65.18** | **53.44** | 70.47 | **83.44** |
| B-Kmeans | 20.55 | 40.70 | **58.98** | 50.27 | 72.50 | 72.32 | 55.16 | 70.86 | 94.06 |
| B-KNN | 20.47 | 41.33 | **58.98** | 49.64 | 72.14 | 77.77 | 54.30 | 89.84 | 96.72 |
| B-GMM | 52.58 | 81.02 | **58.98** | 76.52 | 71.88 | 69.46 | 86.48 | 77.97 | 96.64 |

is that this specific model was trained for speaker verification task on the VoxCeleb dataset (Nagrani et al., 2017), the same pool of utterance data that our MiniVox generated the datastream from. Through the comparison of our purely online learning approach (with MFCC as features) with a more complicated pretrained embedding (with a CNN layer output as features), we aim to investigate whether a learned representation can significantly improve our online learning performance. The CNN model is trained for speaker verification task in VoxCeleb and we are curious about the relationship between a learned representation and our online learning agents. Despite this, we are most interested in the MFCC case, because we aim to push the system to extreme, without any pretraining of any type before deployment: in another word, a fully online learning system.

## 7. Results

Given MFCC features without pretraining, our online agent demonstrated a robust performance (Figure 3a,b,c,d): in most cases, it significantly outperformed the baseline. We wish

to note the overall high diarization error rate in all MFCC benchmarks: first, As many previous pieces of research have shown, the speaker identifiability in MFCC space is too small, especially for short utterances. Second, it is important to keep in mind that the bandit feedback (correct or incorrect classification) makes the *online* speaker diarization problem significantly more challenging, as compared to the standard supervised learning in *offline* speaker diarization, since the true label is never revealed in bandit setting unless the classification is correct. Thus, the diarization error rate in a bandit online setting is expected to be much higher than in the supervised learning setting, which is not due to inferiority of bandit decision making algorithm versus other classifiers, but due to increased problem difficulty, , i.e. the lack of feedback about what the correct decision should have been. Recall that such bandit feedback is often a much more realistic model of agent's interaction with the world, especially in online decision making applications such as online advertisement, clinical trials, and so on, which do not fit into the classical classification framework.

**Learning without Oracle** Table 2 reports the DER in MiniVox without Oracle. In both MFCC and CNN MiniVox environments, we observe that BerlinUCB and its variants outperform the baseline in most cases. The discrepancy of performance of the MFCC and CNN environments can be explained by the innate difficulties of the two tasks: while the CNN embeddings are already well separated because they are pretrained with contrastive loss (Nagrani et al., 2017), in MFCC environments our online learning models need to learn from scratch both how to cluster and how to map reward to features, while balancing the exploitation-exploration tradeoff.

in high-difficulty scenarios (such as C10 and C20), the proposed method outperformed the baseline even when the reward are probability was as low as 0.01. In low-difficulty scenarios, traditional clustering methods like KNN performed the best, while this benefit was inherited by B-KNN and B-Kmeans when feedbacks were sparse (p=0.01). In the CNN environments, we discovered that Kmeans performed the best among all agents. This is expected because the CNN model was trained with the constrastive loss for a high verification accuracy (Nagrani et al., 2017).

**Learning with Oracle** (Table 3). Given a fixed number of speakers, the online clustering modules appear to be more effective. However, the behaviors vary: for instance, we observe that B-GMM performs the poorest in the oracle-free cases, but performs the best in many cases with oracle. We also note that despite the consistent best model in many oracle-free environments, the standard BerlinUCB is surpassed by the baseline and its self-supervised variants in a few MFCC cases with oracle. In certain challenging cases where the reward is sparsely revealed (p=0.01 or 0.1), the self-supervised variants improve the performance of BerlinUCB.

**Is self-supervision useful?** To our surprise, our results suggest that for most cases, the additional self-supervision modules don't improve upon our proposed contextual bandit model. Only in specific conditions (e.g. C20-MFCC p=0.1 with Oracle), the self-supervised contextual bandits outperform both the standard BerlinUCB and the baseline. Further investigation into the reward curve reveals more complicated interactions between the self-supervision modules with the online learning modules (the contextual bandit): B-GMM and B-KNN build upon the effective reward mapping from their BerlinUCB backbone, and benefit from the unlabelled data points to yield a fairly good performance.

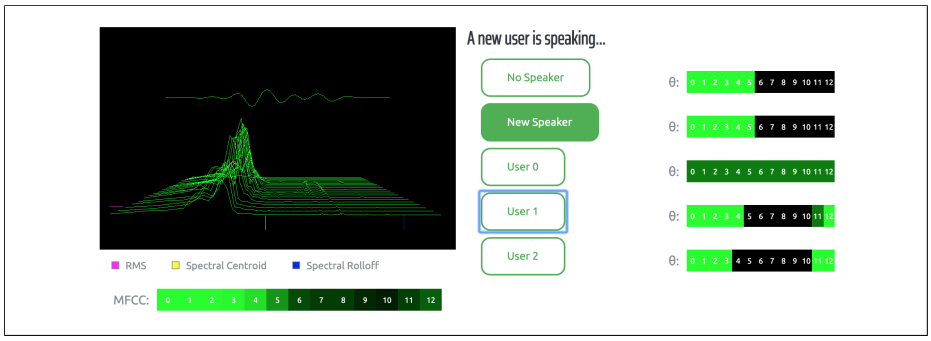


Figure 4: Screenshot of the web system: **VoiceID on the fly**.

## 8. An Interactive System: VoiceID on the fly

In addition to the evaluated benchmark, we also provide a workable web-based system to engage the community into this intriguing online speaker recognition learning problem. "VoiceID on the fly" is an interactive continual learning system (Lin and Zhang, 2020) at https://www.baihan.nyc/viz/VoiceID/ (Figure 4). We compute the MFCCs of the recording in a sliding window fashion given the real-time audio input from microphone, with the MFCC bands color coded in the page. At the start, there are only two buttons available: "No Speaker" and "New Speaker". The agent chooses an arm by setting it be highlighted. If it is correct, we do not have to change it (unless it's "New Speaker", where we need to click on it to confirm creating a new arm). The feature band $\hat{\theta}_a$ of each arm is also color coded real-time to visualize how the agent learns across trials. If it is incorrect, we click on the right arm to give the system a feedback. This demonstration system provides an intriguing example of how an AI agent can learn to recognize speaker identity (1) entirely escaping the necessity of registering user voiceprint beforehands, (2) effortlessly incorporating new users under an optimal exploration-exploitation trade-off, (3) effectively transferring representation of registered user features to new users, and (4) continually learning despite minimal involvement of human corrections (i.e. sparse feedbacks).

## 9. Conclusion and Future Work

To the best of our knowledge, this is the first approach to apply the *Bandit* problem to the speaker diarization task. We formulate the practical task of speaker diarization as an interactive system that episodically receives sparse bandit feedback from users. During unlabelled episodes, we propose to learn from pseudo-feedback generated by self-supervised modules enabled by clustering. For each episode without feedbacks, we apply a self-supervision process to assign a pseudo-action upon which the reward mapping is updated. Finally, we generate new arms by transferring learned arm parameters to similar profiles given user feedbacks. We provide a benchmark to evaluate this task, and demonstrate an empirical merit of the proposed methods over standard online learning algorithm. Lastly we also provide a workable interactive web-based app to engage the general community into this intriguing online speaker diarization task.

Future work include extending the online learning framework in both extraction and clustering modules, developing new ways in branch management (e.g. with adaptive embedding routing like in Lin et al., 2018), considering graph-based self-supervision, and pretraining the reinforcement learning models with offline data (as in Lin, 2021c,b).

The code and data to reproduce the empirical results is implemented with MATLAB and the BanditZoo Python package (Lin, 2021a) and can be found at https://github.com/doerlbh/BanditZoo and https://github.com/doerlbh/MiniVox (specifically for this work).

## References


Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *ICML (3)*, pages 127–135, 2013.

Xavier Anguera, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, 2012.

Donald A Berry, Robert W Chen, Alan Zame, David C Heath, and Larry A Shepp. Bandit problems with infinitely many arms. *The Annals of Statistics*, pages 2103–2116, 1997.

A Canavan, D Graff, and G Zipperlen. Callhome american english speech ldc97s42. web download. *Philadelphia, PA, USA: Linguistic Data Consortium*, 1997.

Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *The British Machine Vision Conference (BMVC)*, 2014.

Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. A sticky hdp-hmm with application to speaker diarization. *The Annals of Applied Statistics*, pages 1020–1056, 2011.

Daniel Garcia-Romero, David Snyder, Gregory Sell, Daniel Povey, and Alan McCree. Speaker diarization using deep neural network embeddings. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4930–4934. IEEE, 2017.

Md Rashidul Hasan, Mustafa Jamil, MGRMS Rahman, et al. Speaker identification using mel frequency cepstral coefficients. *variations*, 1(4), 2004.

Patrick Kenny, Douglas Reynolds, and Fabio Castaldo. Diarization of telephone conversations using factor analysis. *IEEE Journal of Selected Topics in Signal Processing*, 4(6): 1059–1070, 2010.

T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985. URL http://www.cs.utexas.edu/~shivaram.

John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *NIPS*, pages 817–824. Citeseer, 2007.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

Baihan Lin. Online semi-supervised learning in contextual bandits with episodic reward. In *Australasian Joint Conference on Artificial Intelligence*, pages 407–419. Springer, 2020.

Baihan Lin. Banditzoo: a python toolbox for real-world reinforcement learning simulation and evaluation. *arXiv preprint*, 2021a.

Baihan Lin. Model agnostic adversarial attack with offline reinforcement learning. *arXiv preprint*, 2021b.

Baihan Lin. Offline reinforcement learning in bandits. *arXiv preprint*, 2021c.

Baihan Lin and Djallel Bouneffouf. Optimal epidemic control as a contextual combinatorial bandit with budget. *arXiv preprint arXiv:2106.15808*, 2021.

Baihan Lin and Xinxin Zhang. VoiceID on the fly: A speaker recognition system that learns from scratch. In *INTERSPEECH*, 2020.

Baihan Lin, Djallel Bouneffouf, Guillermo Cecchi, and Irina Rish. Contextual bandit with adaptive feature extraction. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2018.

Baihan Lin, Djallel Bouneffouf, and Guillermo Cecchi. Online learning in iterated prisoner's dilemma to mimic human behavior. *arXiv preprint arXiv:2006.06580*, 2020a.

Baihan Lin, Guillermo Cecchi, Djallel Bouneffouf, Jenna Reinen, and Irina Rish. Unified models of human behavioral agents in bandits, contextual bandits and rl. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) Workshop*, 2020b.

Alvin Martin and Mark Przybocki. The nist 1999 speaker recognition evaluation—an overview. *Digital signal processing*, 10(1-3):1–18, 2000.

Mitchell McLaren, Luciana Ferrer, Diego Castan, and Aaron Lawson. The speakers in the wild (sitw) speaker recognition database. In *INTERSPEECH*, 2016.

A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60:101027, 2020.

Neville Ryant, Kenneth Church, Christopher Cieri, Alejandrina Cristia, Jun Du, Sriram Ganapathy, and Mark Liberman. The second dihard diarization challenge: Dataset, task, and baselines. *arXiv preprint arXiv:1906.07839*, 2019.

Gregory Sell and Daniel Garcia-Romero. Diarization resegmentation in the factor analysis subspace. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4794–4798. IEEE, 2015.

Gregory Sell, David Snyder, Alan McCree, Daniel Garcia-Romero, Jesús Villalba, Matthew Maciejewski, Vimal Manohar, Najim Dehak, Daniel Povey, Shinji Watanabe, et al. Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge. In *Interspeech*, pages 2808–2812, 2018.

Mohammed Senoussaoui, Patrick Kenny, Themos Stafylakis, and Pierre Dumouchel. A study of the cosine distance-based mean shift for telephone speech diarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22:217–227, 2013.

Stephen H Shum, Najim Dehak, Réda Dehak, and James R Glass. Unsupervised methods for speaker diarization: An integrated and iterative approach. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(10):2015–2028, 2013.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. IEEE, 2018.

Sreenivas Sremath Tirumala, Seyed Reza Shahamiri, Abhimanyu Singh Garhwal, and Ruili Wang. Speaker identification features extraction methods: A systematic review. *Expert Systems with Applications*, 90:250–271, 2017.

Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopz Moreno. Speaker diarization with lstm. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5239–5243. IEEE, 2018.

B. Yver. Online semi-supervised learning: Application to dynamic learning from radar data. In *RADAR*, 2009.

Z. Zajíc, M. Hrúz, and L. Müller. Speaker diarization using convolutional neural network for statistics accumulation refinement. In *INTERSPEECH*, 2017.

Aonan Zhang, Quan Wang, Zhenyao Zhu, John Paisley, and Chong Wang. Fully supervised speaker diarization. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6301–6305. IEEE, 2019.