Designing a Multi-Factor Quality Evaluation Protocol for Speaker Verification Systems

Ali Aliyev

***Abstract*— This work addresses the challenge posed by the complexity of testing speaker verification models and datasets under real-world conditions. The proposed methodology automatically extracts missing metadata for each utterance, such as codec, language, age, gender, emotion, noise level, duration, and systematically stresses models by simulating bandwidth limits, lossy codecs, noise, volume changes, spectro-temporal masking. Using the Equal Error Rate (EER) as the key metric, we test our methods on the VoxCeleb-1 dataset with ResNet-34 model, which reveals accuracy drops at 8 kHz, in low-SNR scenes and in cross-age trials, while showing robustness to moderate compressions and tempo shifts. The protocol offers an automated standardized and reproducible way to discover a speaker verification model's strengths and weaknesses and can be extended to other speech tasks.**

***Keywords*—speaker verification, speech testing methods, evaluation protocol, voice-biometric.**

I. Introduction

Today, speaker verification is the cornerstone of any voice-biometric system. Voice biometric systems are used for various purposes, such as pre-processing audio for voice assistants, identification and verification of call center and bank customers. The performance of all these systems is directly related to the accuracy of the model in constructing feature vectors of the speaker's voice, but often in real-world conditions, the audio signals also contain noise and other extraneous sounds. Such conditions can degrade the performance of speaker verification models, so it is very important to make our system as robust as possible to noise and extraneous signals.

Unfortunately, there is no standardized set of methods to test such a system, is this field commonly used practice is to use different datasets with varied conditions such as noise, music, codecs, etc. This creates difficulties with understanding accurately under which conditions your system fails and makes you listen to each example yourself to define problematic cases and conditions. This extends the time of the testing procedure after each training.

The goal of this paper is to develop a methodology that will simulate various challenging conditions that are encountered in real life, such as: environmental noise, recording quality artifacts, insufficient speech in the signal, etc. In our methodology, we will use different combinations to get a full picture of the quality of our systems. Using this testing protocol is going to significantly speed up the process of testing and releasing new versions of our model.

II. Related Work

As mentioned before, there are no publicly accepted test protocols for speaker-verification models and systems. Researchers often use different types of datasets containing recordings from multiple environments and quality levels.

One of the most popular and widely accepted datasets for testing is VoxCeleb 1 [1], which is based on interviews of different celebrities from YouTube with a set of over 100K utterances from 1,251 speakers. VoxCeleb1 is split into three parts: VoxCeleb1-Original (Vox1-0), VoxCeleb1-Extended (Vox1-E) and VoxCeleb1-Hard (Vox1-H). Another set of datasets used for speaker-verification task is the NIST Speaker Recognition Evaluations (SRE), it's the oldest running series of benchmark datasets. Conducted by the U.S. National Institute of Standards and Technology (NIST). Despite the large collection of different records in this dataset, it does not solve the key problem – the lack of accurate metadata on quality, noise, codecs, and the presence of other sounds for each recording. This forces us to manually check each audio file. However, it turns out that NIST 1998 SRE [2], contains a study, that describes impact of factors, as gender, duration of speech-segment, input source, microphone type, pitch, speaker health, speaking rate, noise. This work analyzes results the 1998 SRE challenge, but does not offer any methodology for testing such systems, which has led us to this paper.

As we can see there is no publicly accepted testing methodology for voice-biometric systems, all of them rely on different datasets with different recordings qualities and conditions without metadata for each sample. This makes debugging and testing process difficult and time-consuming process and create huge challenges for automating quality control.

III. Methodology

Based on the discussion in previous sections, the main problem of all these datasets is the lack of metadata with all necessary information for testing, but there is also an additional problem, which is the lack of diversity of data in test datasets. it can be a problem when we prepare our system for one type of data and, in production, encounter conditions that are completely different from those in our training dataset. This led us to two tasks. First, creating a set of algorithms to extract correct metadata from existing datasets, it's going to be passive part of the method. S

Second part is going to be active, where we will simulate different conditions on our existing datasets, in order to increase diversity of different hard conditions to properly test our voice biometric systems.

A. Testing metrics

As evaluation metrics in our testing protocol, we decided to use one of the most popular and widely used metrics, such as **Equal Error Rate (EER)**. Using this popular metric will allow us to compare our results more easily with the work of other researchers. EER is the point where false-acceptance rate equals to false-rejection rate, so, basically, it helps us find the sweet spot between these two values.

$$EER = FAR(t_{EER}) = FRR(t_{EER}), where$$

$$t_{EER} = \arg\min |FAR(t) - FRR(t)|$$

- FAR(t) is the rate at which impostor attempts are incorrectly accepted when the score exceeds the threshold t;
- FRR(t) is the rate at which genuine attempts are incorrectly rejected when the score falls below the threshold t.

B. Passive methods

In this section, we are going to define information about the existing condition of our dataset samples, in order to correctly create a connection between metrics and conditions of each sample. For our testing we are going to choose following features of audio samples:

Sampling rate: one of the most important features of audio for voice biometry. With higher sampling rate we have more data for our system, which leads to increased accuracy.

Codec compression: there are a lot of different audio-compression codecs, such as MP3, AAC, OPUS, GSM, AMR etc. Each of them has its own algorithm to compress original audio to decrease size of original sample. So, it is very important to accept samples with different codecs. It is also important to track bitrates and other different parameters of each codec, as they can heavily affect final results.

Language: Even though most biometric systems are text-independent, this does not make them fully language-independent. It's obvious that a system trained for Chinese is not going to perform very well on Russian or any other European language. So, it's important to understand language limitations of speaker verification systems.

Emotions: Different emotions can change vocal tract characteristics, which are going to impact accuracy. Unfortunately, the accuracy of current generation of speaker emotion detection models are limited, so because of this, we will only use three main emotions, such as: angry, neutral and happy. It will help us to minimize the error in our testing results.

Age group: Defining different age groups, such as children, teenagers, adults, seniors is important, because voices naturally change over time due to aging, health conditions, or lifestyle factors.

Gender: Usually, average male voice has an octave lower in pitch, 100Hz vs 200Hz for females, plus men have deeper resonance, it's all due to difference in size of vocal tracts.

Background noise: Estimating level of background noise is crucial in order to detect a weak point of our algorithm. However, predicting SNR can be tricky, so we use the method described in this combined-model [3], where a multi-task VAD-SNR estimation neural network was introduced.

Duration: identifying minimum, optimal, and maximum audio duration for our systems is crucial for production use by customers, because it's a primarily defines the latency of biometric systems. Being able to use the system with very short durations improves user experience and reduces the computational power needed to calculate embeddings, as shown in this research [4].

Defining the above-listed features for our datasets samples is crucial to correctly understand the connection between different conditions and our final results.

C. Augmented methods

In this section we are going further than the passive methodology which was described in the previous section, even though features that was described there are comprehensive, but usually each testing dataset for the speaker-verification task is collected in one domain, which is narrows the diversity of conditions. Collecting a dataset for a speaker- verification task is very hard and can only be done automatically, because manually labeling is not an option in this field. So, it's important to use the data that we have to the maximum and artificially simulate different conditions that can occur in production.

We are going to use a different methods of increasing diversity as much as possible for each previously described feature.

Downsampling: Even though current generative neural networks so-called super resolution networks [5] allow us to increase sampling rate of audio and it is quality they are not accurate enough for us to measure the differences between original and upsampled audio in speaker-verification results. Because of this reason we are going to only decrease the sampling rate – that is, perform downsampling. For downsampling we are going to use the classic method, which involves using low-pass filter to remove frequencies above the Nyquist frequency of the target sample rate, then performing decimation (*keeping every n-th sample, where n = original sample rate / target sample rate*). For our testing protocol we choose the following sample rates:

- 8 kHz – a classic sampling rate for phone calls in GSM/CDMA. It's usually the most important sample rate that should be tested and considered as the primary sample rate for such any speech processing system. It's most also problematic, because we have less information for our speaker verification systems generation and as we mentioned earlier, sampling rate has a huge impact on final accuracy.
- 16 kHz – a new standard for voice calls for phones as part of VoLTE and for VoIP. It's also used in some communication applications.
- 22.05 kHz – half the sampling rate of audio CDs quality. Usually used in computer applications.
- 44.1 kHz and 48 kHz – used in Audio CD and DVD and can be recognized as high-quality audio standard for most cases.

It's worth to mentioning that we only downsample to sample rates below our original sample rate, otherwise we would just be upsampling and adding zeros instead of useful data to the audio samples. Note that when we apply any codec and downsampling, we use the downsampling method that is included in FFmpeg codec-specific libraries.

Simulating codecs: in order to properly apply different codecs, we are going to use FFmpeg. It's containing the

correct and stable version of the audio codecs that we need to apply to our samples. It should be noted that, FFmpeg usually has a few different implementations for each type of codec, so it's crucial to use the correct one in order to get similar samples as they would be in production. One should be aware that almost all codecs have two modes, CBR (Constant Bit Rate) and VBR (Variable Bit Rate), we decided to use CBR, because it improves the repeatability of results and is also more widely used in voice applications, but nothing prevents changing CBR to VBR if it's more applicable for your production conditions.

- **MP3:** For the MP3 codec we have decided to use the ***libmp3lame*** encoder wrapper instead of ***libshine***. It's widely used as the de facto standard encoder and provides us with better audio quality and overall control over bitrate.
- **AAC:** There are also two implementations of AAC, the native implementation and the Fraunhofer FDK based libfdk_aac. It would be fair to say that, in this list AAC is the most challenging in terms of implementation. A lot of companies have their own implementations, which makes our task even harder, but libfdk_aac offers many more options and help us make samples closer to production results.
- **OPUS:** As in other codecs, ffmpeg has its own native implementation and an additional library called libopus, we are going to use the libopus implementation because it offers more options for compression. Also, it offers an application type, where audio is used, such as VoIP, audio and lowdelay, of course we decided to choose VoIP, because of our field of research.
- **GSM:** The three previous codecs we described a very widely used in different applications such as VoIP, streaming services, audiobooks etc, but based on name of codec, we can see that the GSM codec only used in second generation cellular networks for mobile phones, which creates a certain difficulty with its implementation, because it's not used anywhere else, except cellular networks. FFmpeg includes the libgsm library, which correctly implement the GSM codec based on accepted standard. Also, compared with other codecs, here we don't need check any options to choose bitrate or sample rate. Sample rate should be set at 8 kHz and bitrate should be 13 kbps exactly, otherwise it's just not going to work. It has actually made our work much easier, since we don't need to check all different combinations of sample rate and bitrates, and also having one standard implementation give us a guarantee that results of our research and at the production will be the same.
- **AMR:** While GSM codec quality is usually good enough to understand speech of your conversation partner, but it has a fixed bitrate value. From one side, it's made implementation across devices easier, but in very heavily populated areas it creates problem for cellular networks. So, AMR was created to solve this problem. The AMR codec is divided into two parts: **AMR-NarrowBand [6]** and **AMR-WideBand** [7]. AMR-NBhas a fixed 8 kHz sample rate and dynamic bitrates, ranging from 4.75 kbps to 12.2 kbps. Bitrates on cellular networks are change dynamically based on how busy radio channel is, in our implementation based on libopencore-amrnb. Based on the name, we can see that that AMR-WB uses 16 kHz sample rate and also higher bitrate, ranging from 6.6 kbps to 23.85 kbps. AMR-WB is the standard codec for Voice over LTE (VoLTE) and significantly improves quality.

Adding Noise: We decided to use Gaussian noise, widely used in audio field, with such SNR values as: 5 dB, 10dB, 15dB, 20 dB, 30 dB.

Signal modification: We are going to apply different modifiers to our signal:

- Sound volume: increasing or decreasing volume from -30 dB to 30 dB.
- Applying masks to the time and frequency planes of spectrograms. For time mask, masked area will range from 0.1s to 0.5s, for the frequency mask it will range from 250Hz to 1500Hz.
- Time stretching: speeding up or slowing down the original audio by 0.5x, 1.5x and 2x [8].

IV. -TESTING PROTOCOL RESULTS

In this section we will run a series of EER-metric tests by augmentation of the original dataset with methods described, and we will also measure the metrics with the passive methods that were described in previous section. For repeatability of experiments, we will use the VoxCeleb-1 speaker verification datasets. As a base model, we will use ResNet-34 [9] with TSTP trained on VoxCeleb-2 [10] dataset with usage of Kaldi-style [11] F-Banks features. In Figure 1, we demonstrated overall statistics for VoxCeleb-1 dataset, below we have described each feature in more details. First, we tested different age groups combinations. We split our speakers ages into three groups:

- young-adult: 18-35 yr
- middle-aged: 36-55 yr
- senior: ≥ 56 yr

Table 1. EER for age-pair conditions.

Age conditions	**Trials**	**EER (%)**
senior - senior	6049	2.04
middle-aged + senior	5177	1.00
middle-aged + middle-aged	6576	0.70
middle-aged + young-adult	7996	1.21
senior + young-adult	4140	6.90
young-adult + young-adult	7735	1.00

As shown in Table 1, the highest errors were with senior people because of the capabilities of the human body, therefore speech capabilities, begin to decline. Having two seniors instead of one in comparison, obviously increase error, as it shown in comparison with middle-aged and senior pairs. The highest error rate occurs with young-adults and seniors: degradation of senior's voice features and not fully-developed voice features of young people can boost this error significantly, in this dataset and model almost seven times, up to 6.9 %. Because there was not enough, language diversity in the dataset, main language was

English, we decided to only calculate metrics for the English with English pairs and all other languages with English.

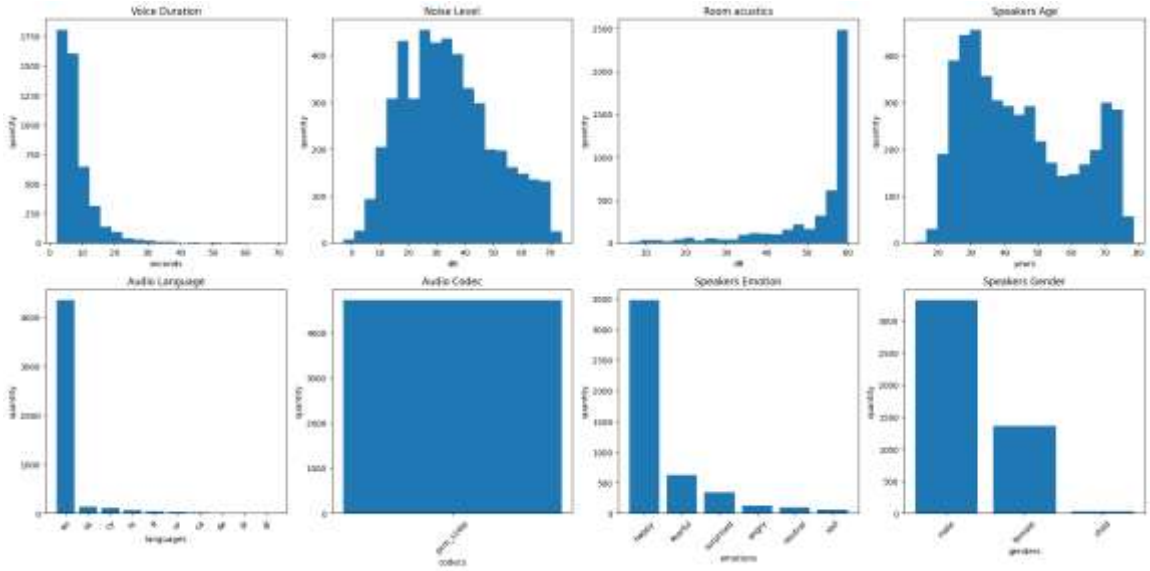

Figure 1. Overall dataset statistics.

Table 2. EER for language-pair conditions.

Language conditions	**Trials**	**EER (%)**
same-language only	32 940	1.07
cross-language only	4780	0.67

As shown in the Table 2, it's easier for the model to separate speakers from different languages, probably it's related to the fact that even though the system is text-independent, that doesn't necessarily mean it would be language dependent. Unfortunately, as is shows in Pic #, imbalance between languages is large, and there is not enough data to draw clearer conclusion about other languages and how different combinations impact our system.

Table 3. EER for gender-pair and age-pair conditions.

Gender and Age conditions	**Trials**	**EER (%)**
same-emotion	22 602	1.08
cross-emotion	15 118	0.93
same-gender	29155	1.14
cross-gender	8565	0.56

It is noticeable in the Table 3, that if a pair of speakers has different emotions and genders, it noticeably decreases the error rate. This is especially can be seen in case, where a male and a female speaker were compared.

Table 4. EER for SNR conditions.

SNR level conditions	**Trials**	**EER (%)**
≤ 10 dB	190	7.65
10 – 20 dB	2050	3.10
20 – 30 dB	10820	1.10
30 – 40 dB	15000	0.78
> 40 dB	9660	0.70

As we can see in the Table 4, in our dataset there is a small number of examples, where the SNR is lower than 10 dB, which can be compared with sound of a subway, this leads to an almost eight-fold increase in the error rate. Overall, the test shows that office-like conditions (20-30 dB), match almost our average metric for the dataset, and that better SNR values can improve our results. We can see that our model has room for improvement in handling noisy signals.

Table 5. EER for speech duration conditions.

Speech duration conditions	**Trials**	**EER (%)**
2-3 s	31	0
3-4 s	3014	1.29
4-5 s	15238	1.04
5-10 s	17440	0.89
> 10 s	1997	0.94

Unfortunately, this dataset contains not a statistically significant amount of 2-3 second audio files to get correct metrics, so we obtained only 31 trials, which are not enough. Overall, we can see in the Table 5, that increasing speech length up to 10 s decreases the error rate, probably because the statistical pooling that was used in our model, which averages features over timeline and makes our model more accurate. However, increasing the audio to more than 10 s increases the error rate compared with 5-10s samples, it maybe because these examples contain more useless or harmful signals than the previous ones. It may also be due to the fact that we have only about 2000 samples longer than 10 s. Because of this, more detailed analyze on another dataset is required.

Overall, our implemented passive and active testing methods including metadata extraction, noise simulation, codec simulation, and signal modifications enabled us to pinpoint specific weaknesses of ResNet-34 VoxCeleb-trained model, such as significant error increase in low-SNR (<10 dB) conditions, cross-age trials involving seniors and young adults, and frequency-masked spectrograms. We also confirmed strengths, including stable performance across most codec types, resilience to moderate volume changes, and minimal degradation under typical telephony sampling rates (16 kHz). These insights provide concrete targets for

improving noise robustness, cross-age generalization, and frequency-information preservation in future model versions. However, it should be noted that, the values that we provided cannot be 100% accurate, because detecting, age, emotion, language, snr and duration relied on different kinds of neural-network models for each task, each of which has its own error rate. But even with this inaccuracy in the models used, they help us understand overall trends for each feature.

Table 6. Active methods EER statistics

Condition	**EER (%)**
Downsampling:	
- 16 kHz	1.01
- 8 kHz	1.97
Codecs simulating:	
- MP3	1.01
- AAC	1.00
- OPUS	1.02
- GSM	1.21
- AMR-NB	1.23
- AMR-WB	1.04
Gaussian noise (SNR):	
- 5 dB	1.45
- 10 dB	1.21
- 15 dB	1.04
- 20 dB	1.03
- 25 dB	1.02
Volume modification	
- -30 dB	1.02
- -20 dB	1.01
- -10 dB	1.01
- +10 dB	1.01
- +20 dB	1.01
- +30 dB	1.01
Speed perturbation:	
- 0.5x	1.06
- 1.5x	1.06
- 2x	1.17
Time masking:	
- 0.1s	1.01
- 0.25s	1.08
- 0.5s	1.14
Frequency masking:	
- 10 bins	1.28
- 15 bins	1.41
- 20 bins	1.60

We can easily spot in the Table 6, that downsampling the original audio from 16 kHz to 8 kHz almost doubles the error rate. But at the same time, using 8 kHz codecs, such as GSM and AMR-NB, does not increase the error rate at the same extent because these codecs are adopted to compress human voice much better, than simple downsampling. Overall, the other codecs behave close to the original version, which shows that original model was very well trained with different codecs. The original training pipeline of the model used Gaussian dithering, so it was trained to ignore even significant SNR levels of this kind of noise. Changing the volume in either direction did not significantly change our results.

Altering the playback speed to 0.5x or 1.5x changed the error rate by only for 0.05%, but 2x increased it by 0.16%. This is also related to our pipeline, which already uses speed perturbations of 0.9x and 1.1x to the original samples. We also applied time and frequency masking, while time masking, even 0.5s didn't affect to much our error rate, it probably because overall our dataset examples speech length starts at least from 3 seconds, so, loosing a random 0.5s segment is not too harmful. However, frequency masking (mel-bin masking) is different, even the smallest masking of a random 10-bin range caused the EER rise to 1.28 %. And this was a random 10-bin range masking, masking the 10 first bins would increase the error rate, because the first 255 Hz contain more fundamental voice information than other frequencies. This is probably because the original training pipeline did not use SpecAugment [12], which masks random parts of input spectrogram, so our model is less resistant to losing certain parts of the information.

V. Conclusion

The proposed methodology consists of two complementary components designed to cover both real-world and synthetic stress testing of speaker verification systems. Passive analysis - automatic extraction of comprehensive metadata for every test sample, including codec type and parameters, sampling rate, spoken language, speaker age group, gender, emotion category, background noise level (SNR), and utterance duration. This analysis allows us to correlate model performance with specific acoustic and demographic conditions without manually inspecting each recording, enabling large-scale, objective evaluation. Active simulation - systematic application of controlled signal degradations to replicate challenging real-world scenarios in a reproducible way. These include downsampling to telephony and sub-telephony bandwidths, applying a variety of common speech codecs (MP3, AAC, OPUS, GSM, AMR-NB/WB) at fixed bitrates, injecting Gaussian noise at defined SNR levels, altering playback volume, applying spectro-temporal masking, and performing speed perturbations. This controlled augmentation ensures that all conditions are tested consistently, independent of dataset limitations. Together, these components form a standardized, fully automated testing protocol that not only evaluates how a given model performs under diverse acoustic and demographic conditions, but also reveals its exact points of failure. The approach is dataset-agnostic, scalable, and can be extended to related speech-processing tasks such as anti-spoofing, diarization, and automatic speech recognition.It will help us not only compare different models, but also gain a deeper understanding of our current dataset and model for further improvements, which is not possible during manual testing's or by averaging overall datasets samples scores. The implementation of the proposed passive and active testing protocol is publicly available [13].

In future work, the author will explore adding additional methods that can be useful for other types of models and will also work on increasing the accuracy of the methods currently used.

References

[1] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in Proc. Interspeech 2017, Stockholm, Sweden, Aug. 2017, pp. 2616–2620

[2] G. R. Doddington, M. A. Przybocki, A. F. Martin, and D. A. Reynolds, "The NIST speaker recognition evaluation—Overview, methodology, systems, results, perspective," Speech Communication, vol. 31, nos. 2–3, pp. 225–254, 2000.
[3] M. Lavechin et al., "Brouhaha: Multi-Task Training for Voice Activity Detection, Speech-to-Noise Ratio, and C50 Room Acoustics Estimation," in Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Taipei, Taiwan, Dec. 2023, pp. 1–7.
[4] H. Zeinali, K. A. Lee, J. Alam, and L. Burget, "SdSV Challenge 2020: Large-scale evaluation of short-duration speaker verification," in Proc. Interspeech 2020, Shanghai, China, Oct. 2020, pp. 731–735.
[5] H. Yamamoto, K. A. Lee, K. Okabe, and T. Koshinaka, "Speaker augmentation and bandwidth extension for deep speaker embedding," in Proc. Interspeech 2019, Graz, Austria, Sept. 2019, pp. 406–410.
[6] ITU-T Recommendation G.711, Pulse Code Modulation (PCM) of Voice Frequencies, Int. Telecommun. Union, Geneva, Switzerland, Nov. 1988.
[7] ITU-T Recommendation G.722.2, Wideband Coding of Speech at Around 16 kbit/s Using Adaptive Multi-Rate Wideband (AMR-WB), Int. Telecommun. Union, Geneva, Switzerland, Jul. 2003.
[8] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in Proc. Interspeech 2015, Dresden, Germany, Sept. 2015, pp. 3586–3589.
[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[10] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in Proc. Interspeech 2018, Hyderabad, India, Sept. 2018, pp. 1086–1090.
[11] D. Povey, A. Ghoshal, G. Boulianne et al., "The Kaldi speech recognition toolkit," in Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Waikoloa, HI, USA, Dec. 2011, pp. 1–4.
[12] D. S. Park et al., "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in Proc. Interspeech 2019, Graz, Austria, Sept. 2019, pp. 2613–2617.
[13] A. Aliyev, "Passive and Active Speaker Verification Testing Protocol – Implementation," GitHub repository, 2025. [Online]. Available: https://github.com/Spectra456/passive-active-sv-testing-protocl/tree/master

A. Aliyev is with the Peter the Great St.Petersburg Polytechnic University, Polytechnicheskaya, 29 B, Saint-Petersburg, Russia (e-mail: aliev.aa@edu.spbstu.ru).