Integrated System for Speaker Diarization and Intruder Detection using Speaker Embeddings

Illia Zaiets[1], Vitalii Brydinskyi[1], Dmytro Sabodashko[1], Yurii Khoma[1], and Khrystyna Ruda[1]

[1] *Lviv Polytechnic National University, Bandery 12, Lviv, 79013, Ukraine*

Abstract

This paper explores the use of diarization systems which employ advanced machine learning algorithms for the precise detection and separation of different speakers in audio recordings for the implementation of an intruder detection system. Several state-of-the-art diarization models including Nvidia's NeMo, Pyannote, and SpeechBrain are compared. The performance of these models is evaluated using typical metrics used for the diarization systems, such as Diarization Error Rate (DER) and Jaccard Error Rate (JER). The diarization system was tested on various audio conditions, including noisy environment, clean environment, low amount of speakers, and high amount of speakers. The findings reveal that Pyannote delivers superior performance in terms of diarization accuracy, and thus was used for implementation of the intruder detection system. This system was further evaluated on a custom dataset based on Ukrainian podcasts, and it was found that the system performed with 100% recall and 93.75% precision, meaning that the system has not missed any criminal from the dataset, but could sometimes falsely detect a non-criminal as a criminal. This system proves to be effective and flexible in intruder detection tasks in audio files with different file sizes and different amounts of speakers that are present in these audio files.

Keywords

Deep learning, diarization, speaker embeddings, cyber security, intruder detection

1. Introduction

Nowadays digital technologies are changing the world around us at an incredible speed and we are faced with a huge amount of information to process every day. This poses a challenge for many industries, especially cybersecurity and big audio data processing, where accurate and, most importantly, timely data analysis becomes a key success factor. This paper dives into this topic by proposing the development of a speech diarization system based on state-of-the-art machine learning libraries to effectively detect intruders by their voices [1–5].

To ensure the effectiveness and accuracy of the developed diarization system, the VoxConverse dataset was used. This dataset contains a wide range of audio recordings, from single speeches to complex discussions with overlapping voices, allowing the system to be tested in a variety of conditions and is an ideal testing environment.

Particular attention was paid to how the systems handled the most common challenges in modern audio, such as noise, overlapping voices, and varying speaker volumes. It was these challenging recordings that helped us select the best library for the system.

We have developed a methodology for testing and analyzing data to compare diarization libraries, allowing us not only to assess the accuracy of recognition, but also to understand the strengths and challenges of each system and how best to use them.

To evaluate the diarization libraries, we used metrics that help to objectively assess the accuracy and reliability of each library: DER and JER.

CPITS-2024: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2024, Kyiv, Ukraine
EMAIL illia.zaiets.mkbas.2022@lpnu.ua (I. Zaiets); vitalii.a.brydinskyi@lpnu.ua (V. Brydinskyi); dmytro.v.sabodashko@lpnu.ua (D. Sabodashko); yurii.v.khoma@lpnu.ua (Y. Khoma); khrystyna.s.ruda@lpnu.ua (K. Ruda)
ORCID: 0009-0007-0754-0463 (I. Zaiets); 0000-0001-8583-9785 (V. Brydinskyi); 0000-0003-1675-0976 (D. Sabodashko); 0000-0002-4677-5392 (Y. Khoma); 0000-0001-8644-411X (K. Ruda)

The results of the evaluation of the diarization libraries were used to select the best one that would be suitable for building a system capable of accurately and efficiently identifying and separating speakers in audio recordings to detect intruders. Ultimately, the Pyannote library was selected as a key element of the system difications, which is determined by the comprehensive security system of multi-level information technology [6, 7].

2. Materials and Methods

Audio diarization technology detects and distinguishes individual speakers in audio recordings. This technology is based on the complex analysis of voice data, using machine learning and deep learning algorithms to recognize the voice characteristics of each speaker to identify the individuals involved in a conversation. This includes analyzing tone of voice, speaking speed, accents, and other unique features that distinguish one speaker from another. Its main task is to divide the audio stream into separate segments so that each segment represents the moment when one person speaks or when there is a change of speakers.

This process seeks to answer the question: "Who speaks when?" throughout the audio recording [8]. Thanks to this, the analysis of audio materials becomes much easier, especially in situations where there are many participants in a conversation and their voices often overlap or change each other. Thus, audio diarization is becoming an indispensable tool for understanding and analyzing complex audio recordings, particularly in the context of cybersecurity and other areas where speaker identification accuracy is critical [9]. Examples of how diarization is used:

- Identify different speakers in an audio file. For cybersecurity investigations, where it is important to understand who exactly participated in the conversation.
- Analyzing communications, such as intercepted phone calls or meeting notes, can help identify suspicious or malicious activity.
- Detecting fraud attempts in telephone calls, for example, by identifying inconsistencies in voices or attempts at manipulation.
- Detecting fraud attempts in telephone calls, for example, by identifying inconsistencies in voices or attempts at manipulation.
- Automate the process of distributing and analyzing audio files, simplifying the work of analysts.
- Protecting the confidentiality of information by "monitoring" audio communications in large organizations to ensure that confidential information is not disclosed.

Overall, audio diarization plays an important role in cybersecurity, helping to detect and prevent fraud, crime, and other cyber threats.

Such an approach plays an important role in protecting information and identifying potential threats, which is especially relevant in the context of the growing number of cyberattacks and fraudulent activities.

However, the implementation of audio diarization in the context of cybersecurity faces several challenges. One of the main ones is the presence of background noise in audio recordings, which can significantly complicate the process of speaker recognition. To solve this problem, various methods of filtering and cleaning the audio signal are used. Another important aspect is the variability of speech features, such as accents, intonations, and speech speed. This requires audio diarization systems to be highly flexible and able to adapt to a variety of conditions.

Audio diarization involves several critical steps to accurately recognize and separate speakers in audio recordings. These stages include Voice Activity Detection, Overlapped Speech Detection, Speaker Change Detection, Segmentation, Speaker Embedding Extraction, Clustering, and Neural Diarizer.

The diarization pipeline is shown in Fig. 1

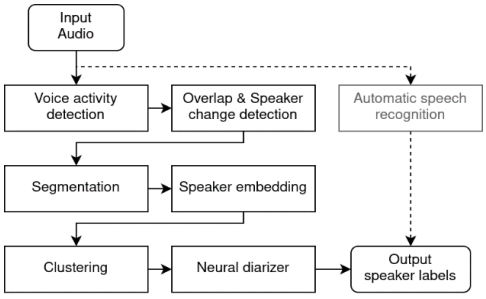

Figure 1: Diarization structure containing its main steps

Automatic Speech Recognition—Fig. 1 demonstrates that the ASR process can be used in parallel with diarization, if necessary [10].

Voice Activity Detection—the first stage is voice activity detection, where the system determines whether a voice signal is present in a certain audio segment. This enables users to filter out quiet areas or noise, focusing solely on segments with voice activity such as speech [11].

Speaker Change Detection & Overlapped Speech Detection—the process includes speaker change detection and detection of moments when several speakers are speaking at the same time. The system analyzes the audio stream to detect the moments when one speaker finishes speaking and another starts. This helps to divide the audio into segments, each of which reflects the speech of a particular speaker. At this stage, the diarization process faces several challenges. First, the quality of the audio recording can vary considerably, and noise, echo, and other audio interference can make it difficult to identify speakers. Secondly, taking into account a variety of speech features, including accents, dialects, and intonations, is an important aspect of ensuring accurate diarization. Third, voice overlap, when multiple people speak at the same time, presents a challenge for accurate speaker segmentation and identification [12].

Segmentation then segmentation takes place, where the audio recording is divided into smaller parts for detailed analysis. Each segment is checked for unique features of the speaker [13].

Speaker Embedding Extraction is one of the most important stages in the extraction of speaker characteristics. The system identifies unique voice attributes, such as timbre, tempo, and intonation, which allows the creation of a unique "fingerprint" of each speaker [14].

Clustering at this stage, clustering takes place, where segments with similar characteristics are organized together. This allows the system to recognize and group segments belonging to the same speaker [15].

Neural Diarizer is the final stage that involves the use of neural diarizers. Neural diarizers are deep learning-based systems that can automatically identify different speakers in complex audio recordings. They use powerful neural networks to analyze audio signals, pick up subtle differences between different voices, and effectively cluster audio segments according to speakers.

Thanks to neural networks, the diarization process becomes more accurate and flexible. Neural diarizers can efficiently process a large amount of audio data and are also better able to cope with challenges such as overlapping voices or changing recording conditions [16].

The key aspect of audio diarization is the use of specific machine learning algorithms. Neural networks, for example, work effectively with the task of classifying audio segments by speakers. Clustering helps in grouping similar speech features, which makes it easier to identify individual speakers. Speaker recognition is important for determining who is speaking at a particular moment in a recording.

Data processing is also an integral part of the process. Extracting features from audio, such as tone, speech rate, and other characteristics, is critical to the accuracy of the algorithms. Data normalization ensures the homogeneity of the input data, which helps in improving the accuracy of machine learning models.

When it comes to optimizing and tuning models, it is important to choose the right hyperparameters to maximize the efficiency of the diarization. Optimization strategies include choosing the model architecture, adjusting the learning rate, and other parameters that can affect the result.

Last but not least, the results are analyzed and validated. This includes evaluating the accuracy of the algorithms on different datasets and analyzing errors, which allows for further improvement of the diarization methods.

The tools used, including PyAnnote [17], NVIDIA NeMo [18], and SpeechBrain [19], have a variety of functionalities for complex speech analysis, speaker identification, and speaker diarization.

These are three of the main and most popular Python libraries used in diarization tasks. PyAnnote is great for automatic audio annotation and speaker identification. NVIDIA NeMo offers powerful tools for working with neural networks, which is ideal for complex diarization tasks. SpeechBrain, with its flexibility and open-source nature, is another great tool for speech processing and diarization. In general, the use of machine learning algorithms in audio diarization tasks is an

important step towards the development of speech processing technologies, offering more accurate and efficient solutions.

3. Aim of Research

The main goal of this article is to implement a system that can accurately recognize and separate the voices of speakers in audio recordings and detect intruders. This is of great importance not only for information security, but also for other areas where it is important to analyze speech accurately. To achieve this goal, we analyzed the capabilities of libraries such as Pyannote, NVIDIA NeMo, and SpeechBrain, and built a diarization system capable of intruder detection based on this analysis.

4. Models Overview

a. PyAnnote

Pyannote represents an important direction in the development of audio diarization algorithms. It is an open-source tool developed for audio data processing, especially focused on diarization tasks.

The main advantage of Pyannote is its flexibility and high accuracy, provided by the use of deep learning algorithms. It uses neural networks to analyze audio recordings, detect speaker identification features or embeddings, and separate and classify them. This allows Pyannote to efficiently separate audio recordings into segments, each corresponding to a specific speaker, even in difficult conditions where voices overlap or there is background noise.

In addition to diarization, Pyannote also provides tools for other audio processing tasks, such as voice activity detection, and gender and age recognition, making it a multifunctional solution.

Another important aspect of Pyannote is its community and open nature. Developers and researchers can contribute their improvements and adaptations, which contributes to the constant updating and improvement of the tool. This also means that the library is constantly adapting to new challenges and technological breakthroughs in audio data processing.

In general, Pyannote is an impressive audio diarization solution that continues to evolve and find new applications in a variety of areas, for both research and commercial use. This library is especially valuable for its ability to perform complex audio processing tasks, providing reliable and accurate results.

b. Nvidia NeMo

NVIDIA NeMo, which stands for Neural Modules, is an innovative approach to machine learning and audio analysis. This library, created by NVIDIA, specializes in applying deep learning to a variety of speech-processing tasks, including audio diarization.

NeMo's special feature is its modular architecture, which allows researchers and developers to easily create, customize, and optimize different components of neural networks for specific tasks. This makes NeMo not only a powerful tool for machine learning experts but also an accessible solution for a wider range of users who may not have deep knowledge in this area.

In the context of audio diarization, NeMo uses advanced deep learning algorithms to efficiently recognize and separate speakers in audio recordings. With its high accuracy and ability to process complex audio data, NeMo is becoming an important tool in tasks that require recognizing different voices, even in the presence of noise or overlapping voices.

NVIDIA NeMo is also continuously updated to include the latest advances in machine learning and speech processing. This ensures that users have access to the most advanced technologies to solve their problems.

In summary, NVIDIA NeMo plays a significant role in today's audio diarization process by offering flexible, scalable, and high-performance solutions for a variety of research and commercial applications in different fields, including cybersecurity.

c. SpeechBrain

SpeechBrain is another important player in the field of machine learning algorithms for audio diarization. This open-source tool was developed as a one-stop solution for a variety of speech-processing tasks, including audio diarization, speech recognition, and speech synthesis.

SpeechBrain is flexible, allowing users to easily customize and adapt the system to their specific needs. The use of deep learning algorithms allows SpeechBrain to efficiently process complex audio recordings and accurately separate the speech of different speakers by returning their embeddings.

One of the advantages of SpeechBrain is its ability to handle large amounts of data, making it an ideal solution for processing audio recordings on the scale required today. Also important is its ability to adapt to different recording conditions, including different languages, accents, and sound quality.

SpeechBrain is also characterized by its openness, which facilitates a community of researchers and developers to work together to improve and adapt the tool. This creates a dynamic environment for innovation and development in the field of speech processing.

Overall, SpeechBrain offers a feature-rich solution for audio diarization tasks, providing high accuracy, flexibility, and scalability, which is important for a wide range of applications, from scientific research to commercial audio processing projects.

5. Experiment Setup

A high-performance NVIDIA RTX 3090 graphics card was chosen to effectively solve the tasks of audio diarization. This choice was made due to its high computing power and optimization for deep learning tasks, which is critical for the efficient processing and analysis of large amounts of audio data.

a. Metrics

Diarization Error Rate (DER)—error of detecting the segment's boundaries and overlaps in the audio recording considering the true or false assignment of speaker identifier to the audio recording segment. This error is the main for diarization and is to be the generally accepted metric in commercial systems.

Diarization error rate can be calculated using the following equation:

$$DER = \frac{T_a + T_m + T_c}{T}, \qquad (1)$$

where T is the total duration of an audio file, T_a is the duration of non-speech falsely detected as speech in an audio file, T_m is the duration of speech falsely detected as non-speech in an audio file, and T_c is the duration of speaker confusion in an audio file.

Jaccard Error Rate (JER) is an error that determines how often the speakers are falsely detected as other speakers. This metric is based on the Jaccard index, which measures the similarity between the sets.

Jaccard error rate can be calculated using the following equation:

$$JER = 1 - \frac{\sum_{i=0}^{N} |S_i \cap D_i|}{\sum_{i=0}^{N} |S_i \cup D_i|}, \qquad (2)$$

where S_i is the set of the segments for a speaker i in the test dataset, D_i is the set of the segments where speaker i was predicted, and N is the total number of speakers.

b. Initial Analysis

Before starting extensive testing on massive data from various Python libraries, we first conducted experiments with the Pyannote algorithm using audio recordings with a wide range of conditions. This included files with different numbers of participants, variations in noise levels, and different degrees of speech overlap. This way, we can better understand Pyannote's performance and reliability in different acoustic scenarios, which is critical for further analysis of larger data.

Initially, a two-minute audio file was selected for analysis, which was a recording of a news broadcast. The peculiarity of this recording was the high level of background noise, although there were no overlapping audio tracks. This choice allowed us to evaluate how efficiently the algorithm can process audio with complex sound environment conditions, not complicated by the simultaneous speech of several speakers.

The audio file with two speakers and background noise is visualized in Fig. 2.

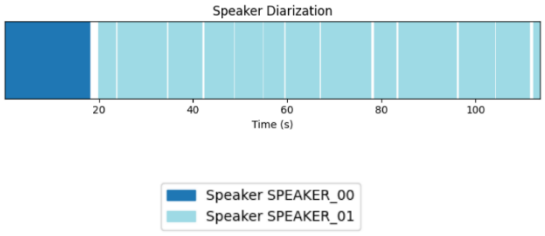

Figure 2: Timeline of an audio file containing two speakers with background noise

The next step in the study was a more complex task for the Pyannote library. A 16-minute

audio recording of a conference with 11 people present at the same time was chosen for analysis. This recording was characterized by a significant level of noise, changes in speech volume, and frequent interruptions between speakers. This made it possible to evaluate Pyannote's ability to effectively cope with the high level of complexity in speech recognition and speaker identification in multi-voice audio.

The audio file with eleven speakers with background noise is visualized in Fig. 3.

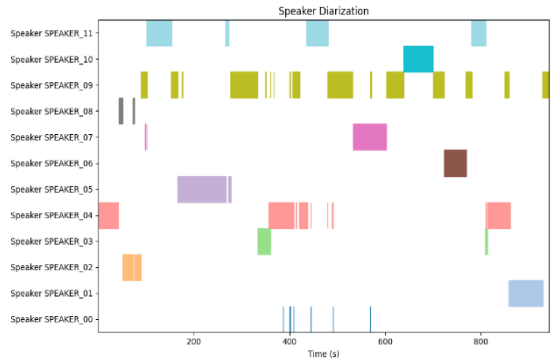

Figure 3: Timeline of an audio file containing eleven speakers

The study was continued by selecting an audio recording of a clean speech with no background noise, overlapping tracks, or interruptions in speech. This recording is ~15 minutes long and represents ideal conditions for analysis, which makes it possible to evaluate the algorithm's performance under optimal conditions. This choice allows you to establish a baseline level of accuracy of the diarization system under ideal conditions, without external interference.

The audio file with two speakers with a clean background is shown in Fig. 4 with background noise.

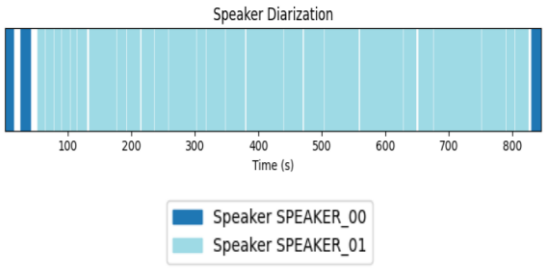

Figure 4: Timeline of an audio file containing two speakers with clear audio

Table 1 contains the results of the initial study of the robustness of the diarization library Pyannote.

Table 1
Initial analysis results

Conditions	DER	JER
Noisy environment; 2 speakers	0.19	0.19
Noisy environment; 2 speakers	0.76	0.75
Clean environment; 2 speakers	0.07	0.07

From the experiment results it can be seen that the diarization system performed well on the smaller amount of speakers, though a bit worse when the noisy environment was present. When it comes to diarization on the bigger number of speakers in the noisy environment, the system did not perform well, so it is not recommended to use this system with data, where there are a lot of speakers with potential overlaps and a noisy environment on top.

c. Test Dataset for Model Selection

The VoxConverse dataset [20] was chosen to evaluate popular Python diarization libraries. VoxConverse is an extensive dataset that was created for speech diarization tasks and is a good resource for researchers and developers in this field. This dataset includes a large number of audio recordings that cover a wide range of scenarios from public speeches and interviews to newscasts and debates. A special feature of VoxConverse is the presence of recordings where speech overlap is observed, which is very typical in real-world settings and is of great interest for research.

The audio recordings in VoxConverse are annotated with detailed labels that include time intervals and speaker identifiers. This information is extremely valuable as it allows us to accurately assess how different diarization algorithms and systems perform in detecting and distributing speech among different speakers. Such annotations are important for comparing the results of diarization systems with the "ideal" and evaluating their effectiveness.

The large amount of data in VoxConverse enables deep and comprehensive analysis. This allows researchers to evaluate dialysis systems in a variety of settings, including scenarios with a variable number of speakers, different noise levels, and different speech styles and accents. This diversity helps to improve the reliability and accuracy of dialysis

systems and contributes to the development of more versatile and adaptive solutions.

Thanks to its openness and accessibility, VoxConverse has become a valuable tool for the community to conduct collaborative research and development in the field of speech diarization. The use of such datasets helps researchers identify new challenges that modern systems may face and develop more efficient algorithms for speech processing.

Out of the entire VoxConverse dataset of 464 records, the first 50 records of the dataset were selected for the tests to reduce the time to perform the diarization and reduce resource usage. These selected records have different lengths, ranging from 3 to 20 minutes, which provides a wide range of conditions to evaluate the performance of my chosen Python machine-learning libraries. Not only does this approach allow for a focus on detail, but it also provides practical relevance by demonstrating how systems adapt to variability in real-world speech scenarios. This helps to gain a deeper understanding of each system's performance in situations that may occur in real life and identify potential areas for further improvement.

d. Model Selection for Intruder Detection System

For each of the selected libraries, we developed the appropriate code, taking into account their unique features. The goal was to ensure that the final result in each of them complied with the generally accepted RTTM standard for diarization timestamps. This methodology provided the ability to equally evaluate and compare the results obtained using different diarization systems.

As part of the experiment, we used models of these libraries trained on the VoxConverse dataset to evaluate their effectiveness in real-world conditions. The main goal of this experiment is to determine which of these libraries is best suited for the final task of detecting an intruder.

The following metrics were chosen to evaluate the performance of each library: average DER and JER. These metrics were calculated based on 50 selected test recordings from VoxConverse. We also took into account the diarization time for each system to evaluate which algorithm is the most efficient in this parameter. This analysis will not only help identify the most accurate voice recognition system but also determine which one provides the best ratio of speed and quality of data processing.

The model selection experiment results are presented in Table 2.

Table 2
Model selection experiment results

Model	Elapsed time	DER	JER
SpeechBrain	3m 53s	0.31	0.31
NVIDIA NeMo	17m 32s	0.41	0.41
Pyannote	20m 7s	0.14	0.14

SpeechBrain, an open-source machine learning library, impressed with its processing speed, taking only 3 minutes and 53 seconds, although accuracy leaves much to be desired and additional tuning is required to achieve optimal results. The average DER of 31% can be considered satisfactory, given the openness of the library and the complexity of the data.

NVIDIA's NeMo took longer to process—17 minutes and 32 seconds—but showed good diarization results, especially given the complexity of the audio data. The average DER of 14% indicates the efficiency of the algorithm.

For the intruder detection system, it was decided to use Pyannote, which showed the best results in diarization of this dataset. With an average DER of 9%, Pyannote effectively handles the challenges of the dataset. Despite the fact that Pyannote's processing time was 20 minutes and 7 seconds, this is compensated by its high accuracy and high-quality documentation, which allows users to quickly get started with the library. Although the processing time is not a decisive factor compared to NeMo, the time to implement and configure Pyannote was significantly shorter.

Thus, given the speed, accuracy, and ease of use, Pyannote was the choice for the final task, demonstrating an excellent balance between processing time and quality of results.

e. Diarization for Intruder Detection Task

Data Preparation

Before developing and analyzing an intruder detection system, it is necessary to collect and prepare data that will be used to train and test the model. This stage involves selecting appropriate audio recordings and processing them to ensure effective training of the system. After creating a reliable and representative training sample, the next step is to develop a method for detecting and identifying potential criminals in the database. The use of various methods of speech diarization will allow us to test the effectiveness of the system and ensure its practical use in real-life scenarios.

For the study, several episodes of a well-known Ukrainian YouTube podcast were selected, where two hosts are constantly participating and different guests come to each episode. In the experiment, some guests were conditionally labeled as "intruders". Five separate three-minute audio recordings were created for each guest to extract their voice embeddings. The total size of the dataset for identifying "intruders" is 56 recordings, each of which is two to three minutes long. Out of this number, 15 recordings include the voices of the identified "intruders", while the remaining 41 do not. This provides a unique opportunity to evaluate how well the developed diarization system performs in recognizing and separating voices in real-life situations, which is key for application in practical scenarios.

The dataset used for this experiment can be found here [21].

The example of a prepared audio file for intruder detection is shown in Fig. 5.

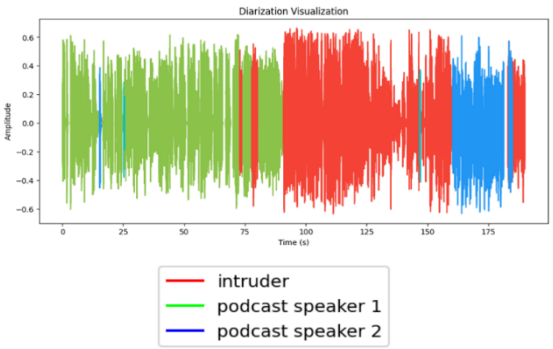

Figure 5: Example of an audio file containing an intruder's voice

Intruder Detection System Implementation

Moving forward, the subsequent phase in our research involves the creation of a bespoke algorithm. This algorithm is tailored specifically to extract embeddings from audio recordings that contain the vocal patterns of individuals labeled as "intruders". The core process of this development entails the transformation of the distinctive vocal characteristics of each speaker into complex, high-dimensional numerical vectors. These vectors are a crucial element as they encapsulate the unique voice features in a quantifiable form.

The strategic utilization of these voice embeddings plays a vital role in our study. It enables a more refined and in-depth comparison and analytical process. This is achieved by measuring the cosine distances between these numerical vectors. By analyzing these distances, we can ascertain with a high degree of precision whether a particular segment of speech can be attributed to a specific "criminal" or another speaker. This methodology is highly effective in distinguishing between different voices in an audio recording. This approach is key to the development of systems used in forensic research and other areas where it is necessary to accurately identify a person by voice.

The next step in the research is to apply an algorithm to collect embeddings from all suspect recordings in the database. This process involves analyzing each audio file and extracting the corresponding embeddings. Once the embeddings are collected, they are clustered. This procedure allows you to group similar voice characteristics, which is key to simplifying the subsequent identification process. Clustering reduces the need to make multiple comparisons between each segment's echo and all of the offender's echoes, thereby increasing the efficiency and accuracy of identification. In addition, clustering helps to identify common characteristics of the voices of the "intruders", which can help to accurately identify potential suspects.

The processing of podcasts includes downloading each audio file, running diarization on this file, and then selecting only the segments larger than 5 seconds. This is done

to ensure detailed analysis and accurate identification of the different voices in the recording. Each segment is then checked against the attacker's speech patterns, which are predefined and stored in a database. This technique allows you to accurately identify the moments of the suspect's presence in the audio material and also provides the ability to identify attackers who speak only in certain parts of the podcast. This is important to ensure high identification accuracy without affecting processing speed.

Before comparing segments, you need to specify a key threshold parameter that can dramatically change the results of the study. The threshold plays a crucial role in determining whether a voice in a podcast segment matches the voice of a known intruder. It serves as a measure for comparing the level of similarity between voice embeddings. The key point is that if the cosine distance between the segment's embedding and the nearest intruder's embedding is less than this threshold, the system recognizes the presence of an intruder in that segment.

The intruder detection system is shown in Fig. 6.

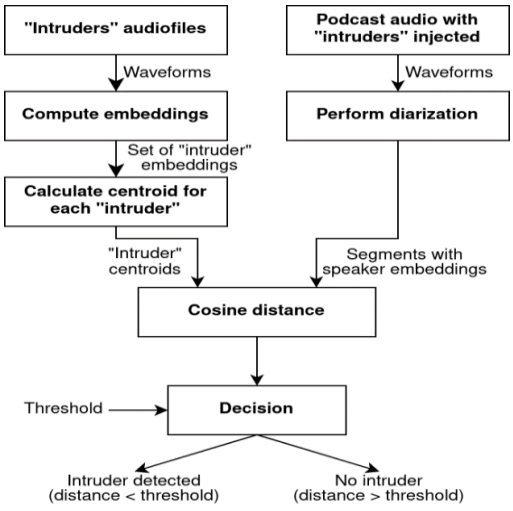

Figure 6: Intruder detection system

Intruder Detection System Experiment

After the development of the main components of the system is completed, the next stage will be its launch and testing on a selected data set. This will allow us to evaluate the functionality and efficiency of the developed system in real conditions. An important part of this process will be the analysis of the results, which will help identify the strengths and weaknesses of the program, as well as possible areas for further improvement. Testing on a dataset will not only confirm the program's ability to effectively recognize intruder voices but will also provide valuable insight into its overall accuracy and reliability in various use cases.

During the meticulous analysis of the dataset, the algorithm demonstrated a remarkable level of accuracy in identification tasks. Among the entirety of the audio files that were processed, it is noteworthy that only a single file was erroneously classified as containing the voice of a criminal. This particular outcome may serve as an indicator of possible limitations inherent within the algorithm itself, or it could alternatively point to specific characteristics of the audio file that might have influenced its recognition capabilities. Significantly, all other files within the dataset were identified with a high degree of accuracy, a fact that robustly affirms the effectiveness and reliability of the system we have developed.

Moreover, this solitary instance of misidentification, while being an outlier, is also of considerable value. It offers critical insights and catalyzes further in-depth analysis and fine-tuning of the algorithm. By closely examining this case, we can gain a deeper understanding of the algorithm's current capabilities and limitations. This understanding is instrumental in guiding subsequent enhancements and optimizations. Our goal is to refine the algorithm's precision in accurately distinguishing between the presence of intruders and non-intruders across a diverse range of audio scenarios. This ongoing process of improvement is pivotal in ensuring that the system remains highly efficient and effective in various real-world applications.

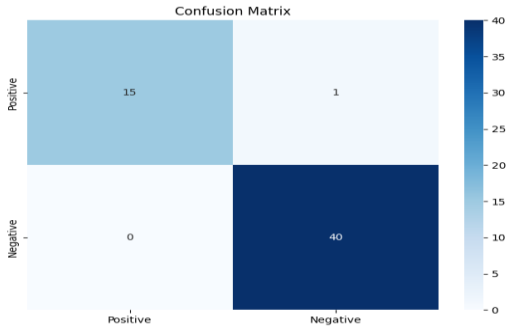

Figure 6: Intruder detection experiment confusion matrix

Drawing upon the data derived from the confusion matrix, as illustrated in Figure 6, we

can compute several crucial algorithm performance metrics, notably Accuracy, Precision, Recall, and the F1-Score. These metrics are indispensable as they furnish insightful details regarding the model's proficiency in precisely detecting intruders. Furthermore, they shed light on the model's dependability in minimizing instances of false positives and false negatives. The attainment of high values in these metrics is a clear indicator that the system we have developed is highly competent in its designated functions. It excels in identifying intruders with remarkable accuracy and is characterized by a minimal occurrence of errors. This aspect of the system's performance is not only a testament to its effectiveness but also highlights its reliability in critical situations where the accurate detection of intruders is paramount.

Table 3
Intruder detection experiment results

Accuracy, %	Recall, %	Precision, %	F1-score, %
98.21	100.00	93.75	96.77

The conclusion of this audio diarization experiment revealed that high accuracy in detecting intruders is achievable, but requires careful tuning of the system to the characteristics of each audio recording. The key factors affecting the success of identification are the "min_segment_duration" and "similarity_threshold" hyperparameters. Setting the minimum segment duration helps to avoid misidentifying intruders, although it may result in missing their short utterances. On the other hand, fine-tuning the similarity threshold for embeddings is important for accurately recognizing the voices of intruders while avoiding false positives. You should also pay attention to the timbre of the voice, as it can significantly improve the results, especially when voices with similar characteristics are present in the audio recording. Thus, an individualized approach to each audio file and its features is the key to effectively detecting criminal activity in different audio contexts.

6. Conclusions

In this paper, we conducted a comprehensive analysis that compares various deep learning models specifically in the sphere of speaker diarization, with a particular focus on their application in detecting intruders. We evaluated the resilience and effectiveness of these diarization systems across a spectrum of environmental conditions. Central to our study is the development of an innovative intruder detection system, which is fundamentally based on the principles and technology of speaker diarization. The results of our investigation reveal a notable compatibility of diarization models within the realm of intruder detection, particularly highlighted by their proficiency in identifying unauthorized individuals within audio recordings or live audio streams. A key outcome of our experimental findings is the discernible superiority of the Pyannote diarization model. This model demonstrated exceptional performance in diarization, evidenced by achieving the lowest DER at 14% and the lowest JER at 14% as well. Despite its relatively slower inference time compared to other models, the accuracy and reliability it brings to intruder detection significantly outweigh this limitation.

In the development of an intruder detection system, we chose to implement the Pyannote diarization model as its core component. The performance of the system was remarkably high, demonstrating an accuracy rate of 98.21%. This high level of accuracy was further complemented by a perfect recall rate of 100.0%, indicating that every single intruder present in the dataset was successfully identified by the system. Additionally, the system exhibited a precision of 93.75%, which, although not flawless, is significantly commendable. The F1-score, which is a balanced measure of the system's precision and recall, stood at an impressive 96.77%, underscoring the system's overall efficacy.

It is noteworthy, however, that a small fraction of the speakers were incorrectly classified as intruders. However, it is overshadowed by the system's paramount accomplishment: its unfailing ability to detect every intruder included in the dataset. This aspect, above all, highlights the system's value as a reliable tool in intruder detection scenarios.

References

[1] O. Romanovskyi, et al., Prototyping Methodology of End-to-End Speech Analytics Software, in: 4th International Workshop on Modern Machine Learning

Technologies and Data Science, vol. 3312 (2022) 76–86.
[2] I. Iosifov, et al., Transferability Evaluation of Speech Emotion Recognition Between Different Languages, Advances in Computer Science for Engineering and Education 134 (2022) 413–426. doi: 10.1007/978-3-031-04812-8_35.
[3] I. Iosifov, O. Iosifova, V. Sokolov, Sentence Segmentation from Unformatted Text using Language Modeling and Sequence Labeling Approaches, in: VII International Scientific and Practical Conference Problems of Infocommunications. Science and Technology (2020) 335–337. doi: 10.1109/PICST51311.2020.9468084.
[4] O. Iosifova, et al., Analysis of Automatic Speech Recognition Methods, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 2923 (2021) 252–257.
[5] O. Iosifova, et al., Techniques Comparison for Natural Language Processing, in: 2nd International Workshop on Modern Machine Learning Technologies and Data Science, vol. 2631, no. I (2020) 57–67.
[6] V. Dudykevych, H. Mykytyn, K. Ruda, The Concept of a Deepfake Detection System of Biometric Image Modifications Based on Neural Networks, IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek) (2022). doi: 10.1109/khpiweek57572.2022.9916378.
[7] Y. Shtefaniuk, I. Opirskyy, Comparative Analysis of the Efficiency of Modern Fake Detection Algorithms in Scope of Information Warfare, 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (2021) 207–211. doi: 10.1109/IDAACS53288.2021.9660924.1
[8] X. Miro, et al., Speaker Diarization: A Review of Recent Research, IEEE Trans. Audio, Speech, Lang. Process. 20(2) (2012) 356–370. doi: 10.1109/tasl.2011.2125954.
[9] V. Khoma, et al., Development of Supervised Speaker Diarization System Based on the PyAnnote Audio Processing Library, Sensors 23(4) (2023) 2082. doi: 10.3390/s23042082.
[10] A. Hannun, et al., Deep Speech: Scaling up end-to-end Speech Recognition, arXiv: preprint (2014).
[11] J. Ball, Voice Activity Detection (VAD) in Noisy Environments, ArXiv (2023).
[12] S. Cornell, et al., Overlapped Speech Detection and Speaker Counting Using Distant Microphone Arrays, Comput. Speech Lang. 72 (2022) 101306. doi: 10.1016/j.csl.2021.101306.
[13] M. Kotti, V. Moschou, C. Kotropoulos, Speaker Segmentation and Clustering, Signal Process. 88(5) (2008) 1091–1124. doi: 10.1016/j.sigpro.2007.11.017.
[14] M. Jakubec, et al., Deep Speaker Embeddings for Speaker Verification: Review and Experimental Comparison, Eng. Appl. Artif. Intell. 127 (2024) 107232. doi: 10.1016/j.engappai.2023.107232.
[15] N. Dawalatabad, et al., ECAPA-TDNN Embeddings for Speaker Diarization, Proc. Interspeech (2021) 3560–3564. doi: 10.21437/Interspeech.2021-941.
[16] D. Garcia-Romero, et al., Speaker Diarization Using Deep Neural Network Embeddings, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2017) 4930–4934. doi: 10.1109/ICASSP.2017.7953094.
[17] H. Bredin, Pyannote.Audio 2.1 Speaker Diarization Pipeline: Principle, Benchmark, and Recipe, INTERSPEECH (2023) 1983–1987. doi: 10.21437/interspeech.2023-105.
[18] E. Harper, et al. NeMo: A Toolkit for Conversational AI and Large Language Models. URL: https://github.com/NVIDIA/NeMo
[19] M. Ravanelli, et al., SpeechBrain: A General-Purpose Speech Toolkit, ArXiv (2021).
[20] J. Chung, et al, Spot the Conversation: Speaker Diarisation in the Wild, INTERSPEECH (2020) 299–303. doi: 10.21437/interspeech.2020-2337.
[21] I. Zaiets, Dataset of Ukrainian Podcasts for Intruder Detection by Voice (2024). doi: 10.57967/hf/0701.