NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Robust Voice Activity Detection using DNN Approaches

Name: Parashar Kshitij

Project ID: SCSE23-0748

Supervisor: A/P Chng Eng Siong

Examiner: Dr. Josephine Chong

Date: 10th May 2024

Presentation Outline

- Introduction
- Literature Review
- Methodology & Results
- Conclusion & Future Recommendations

INTRODUCTION

- Background
- Objective and Scope

LITERATURE REVIEW

- Audio Concepts
- Pyannote, Silero, MarbleNet
- Evaluation Metrics
- ONNX

METHODOLOGY & RESULTS

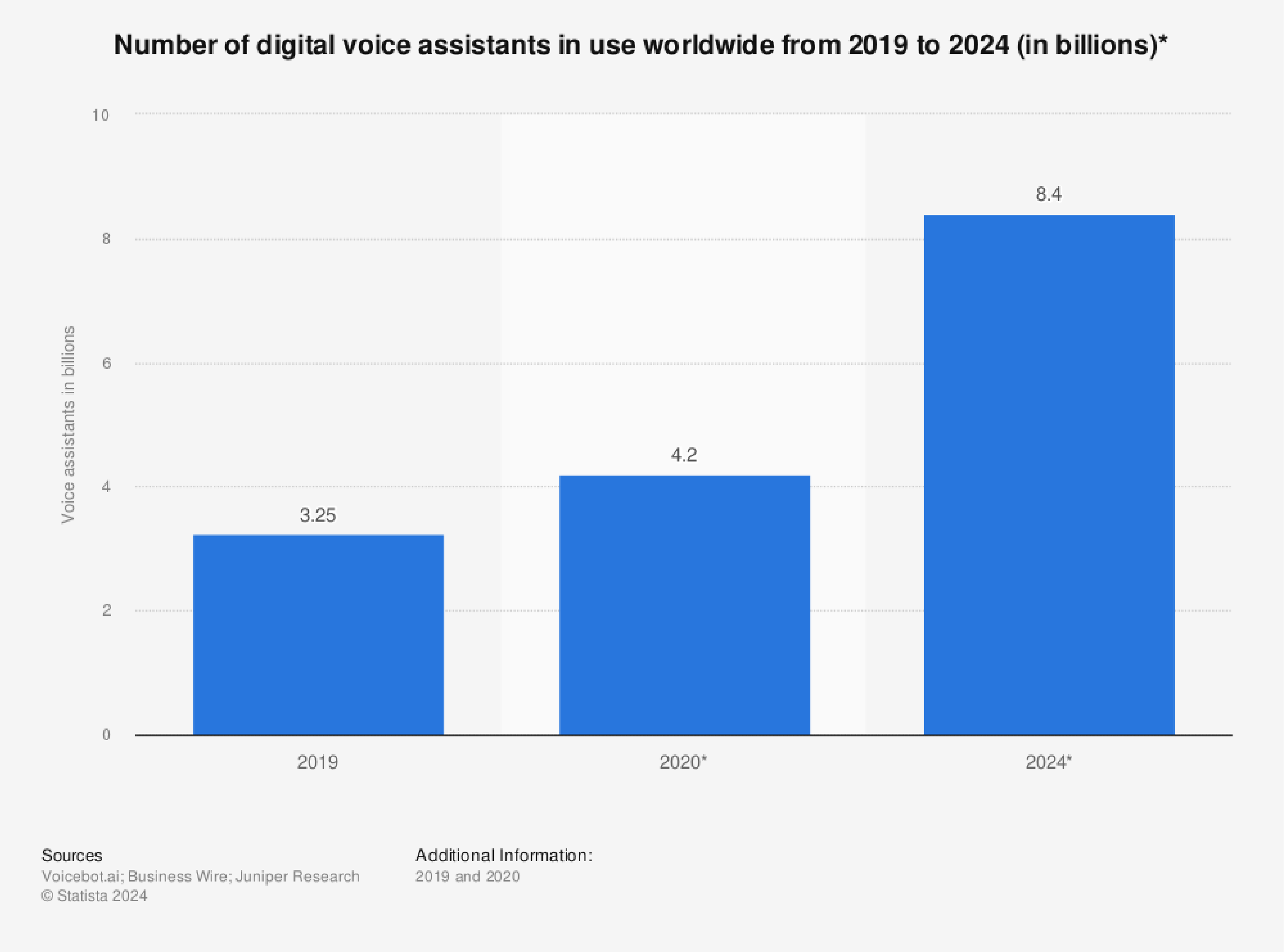
- Experiment 1
- Experiment 2
- Experiment 3
- Experiment 4
- Experiment 5

CONCLUSION

- Contribution
- Future Work

Background

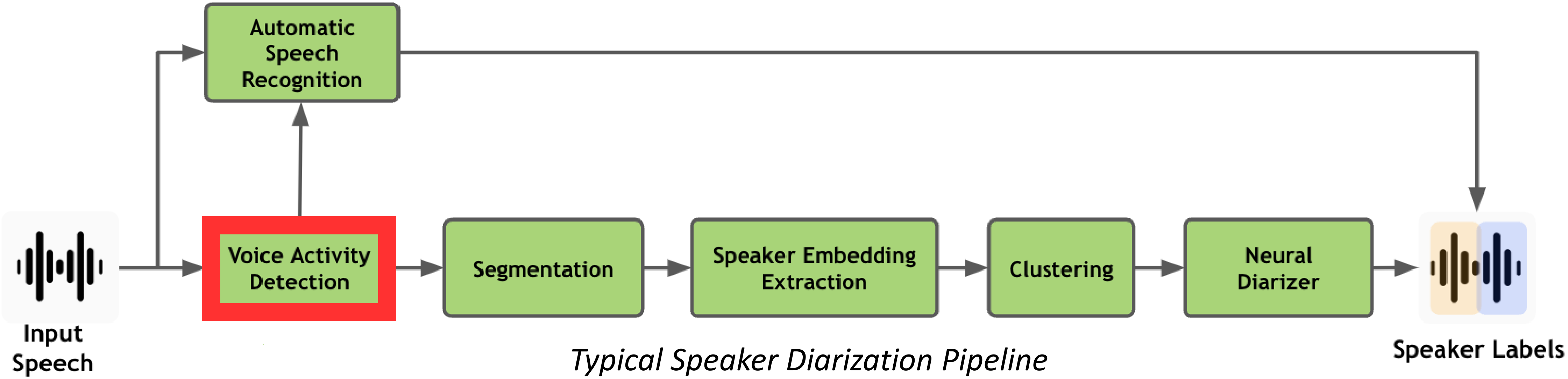
The number of digital voice assistants is projected to grow to **8.4 billion** in 2024



F. Laricchia, Number of voice assistants in use worldwide 2019-2024, Mar. 2022. [Online]. Available: https : / / www . statista . com / statistics / 973815/worldwide-digital-voice-assistant-in-use/.

Background

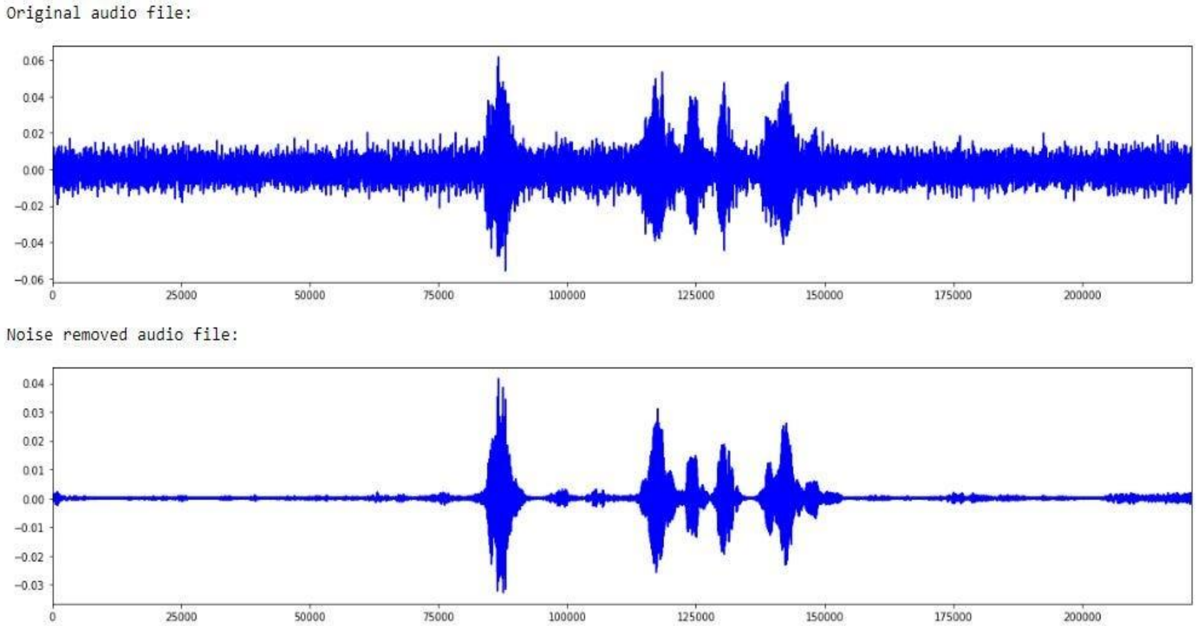
- **Purpose**
  - Distinguish between speech and non-speech segments to improve the accuracy of diarization pipeline
- **Applications**
  - Meeting transcription
  - Speaker identification
  - Voice-controlled devices
- **Approaches**
  - Energy
  - DNN models
  - Hybrid methods



Typical Speaker Diarization Pipeline

Motivation

- **Background Noise**
  - Noise interference challenges accurate speech detection.
- **Domain Adaptation**
  - Adapting to diverse environments and recording conditions is crucial.
- **Real-time processing**
  - Handling longer audio streams demands efficient methods.
- **Evaluation metrics and benchmarks[1]**
  - Standard evaluation metrics and benchmarks are lacking in VAD, hindering method comparison and assessment.



[1] S. Yadav, P. A. D. Legaspi, M. S. O. Alink, A. B. J. Kokkeler, and B. Nauta,“Hardware implementations for voice activity detection: Trends, challenges andoutlook,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 70, no. 3, pp. 1083–1096, 2023

Objectives & Scope

Models:

- Pyannote
- Silero
- MarbleNet

Datasets:

- DIHARD III
- AliMeeting

Evaluate models based on:

- Accuracy
- Missing Detection
- False Alarm
- ROC-AUC score
- Real Time Factor

Develop a robust VAD model

Scope: Single-channel far-field voice activity detection tasks

- Audio Concepts
- Pyannote, Silero, MarbleNet
- Evaluation Metrics
- ONNX

INTRODUCTION

- Background
- Objective and Scope

LITERATURE REVIEW

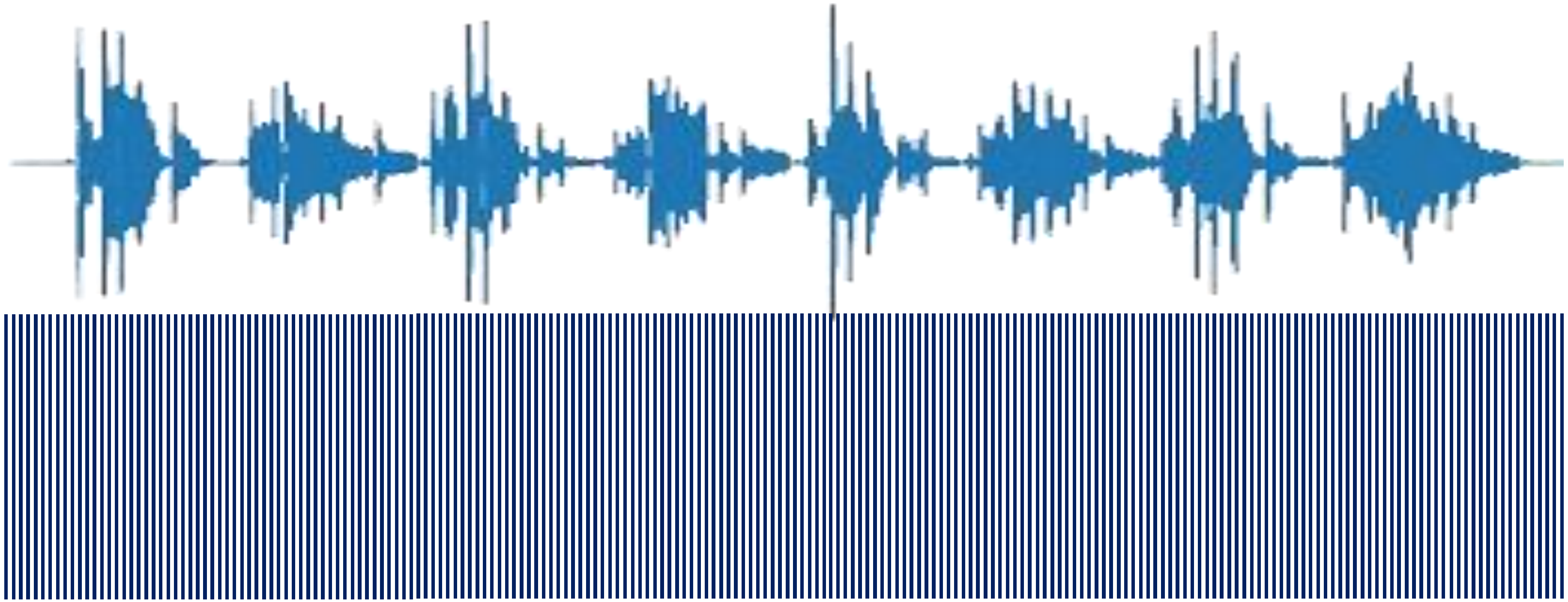
METHODOLOGY & RESULTS

- Experiment 1
- Experiment 2
- Experiment 3
- Experiment 4
- Experiment 5

CONCLUSION

- Summary
- Contribution
- Future Work

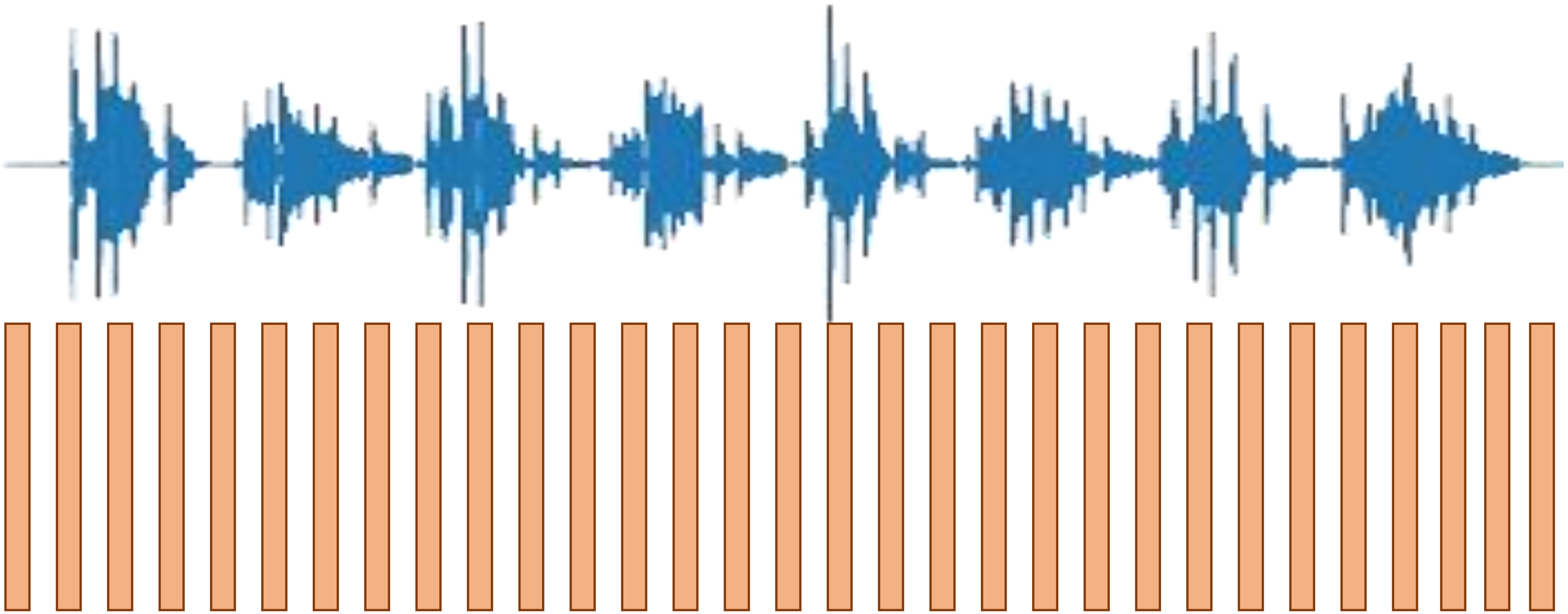
Audio Processing



Audio sampled at 16kHz

Frame

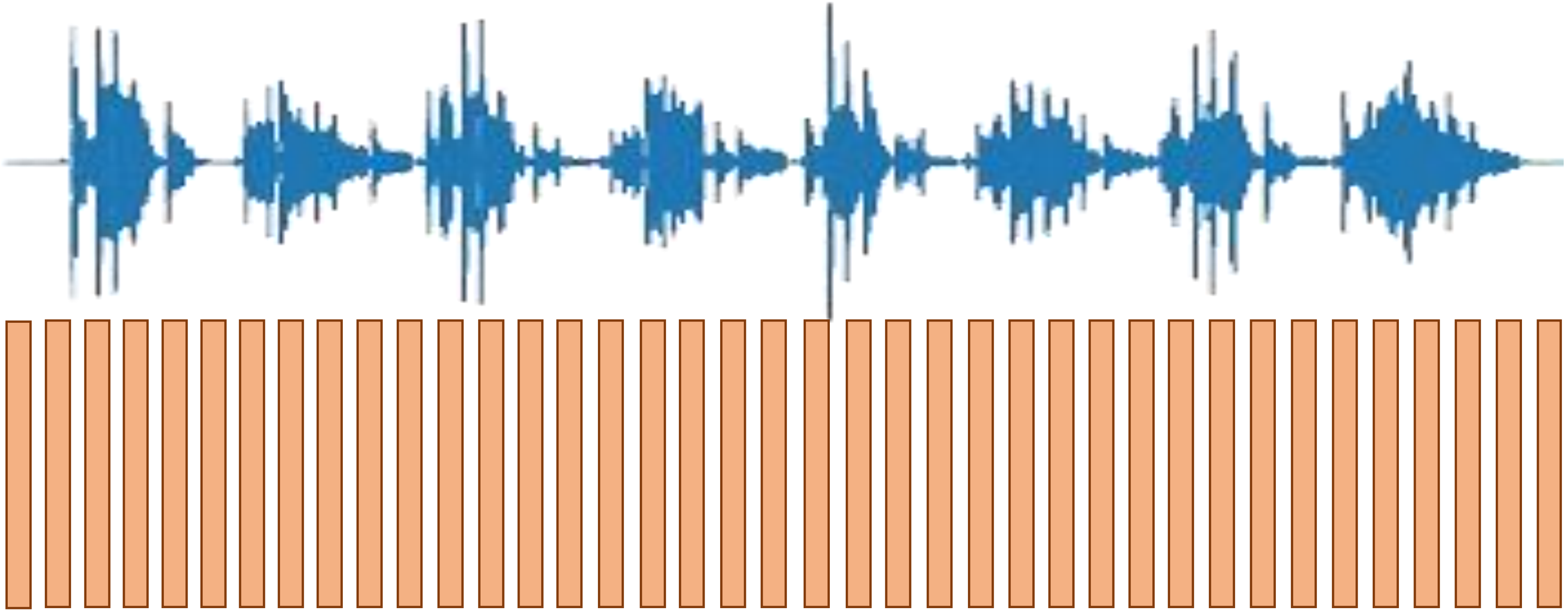
A frame is a single sample of an audio file.



Audio frames

Frame Step

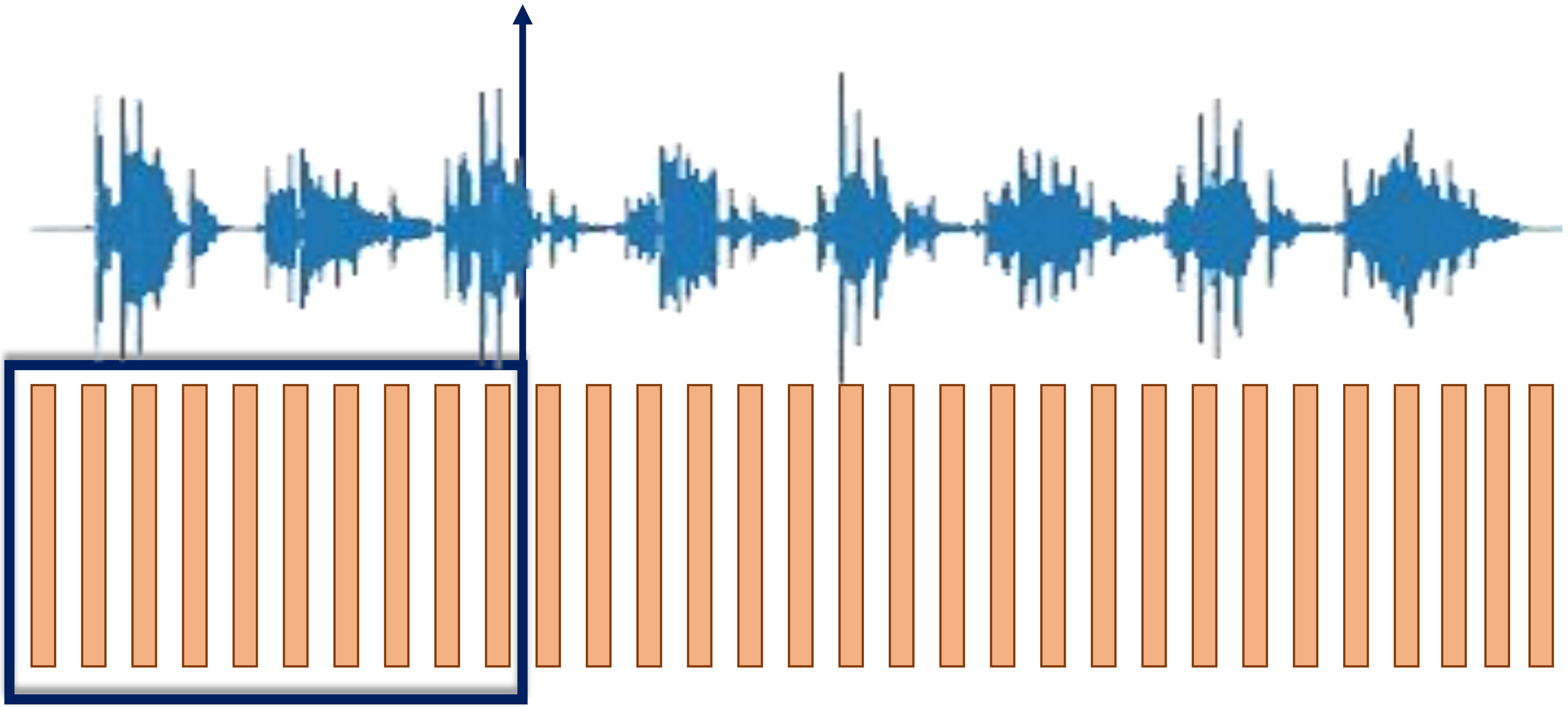
Determines the overlap or spacing between any two consecutive frames



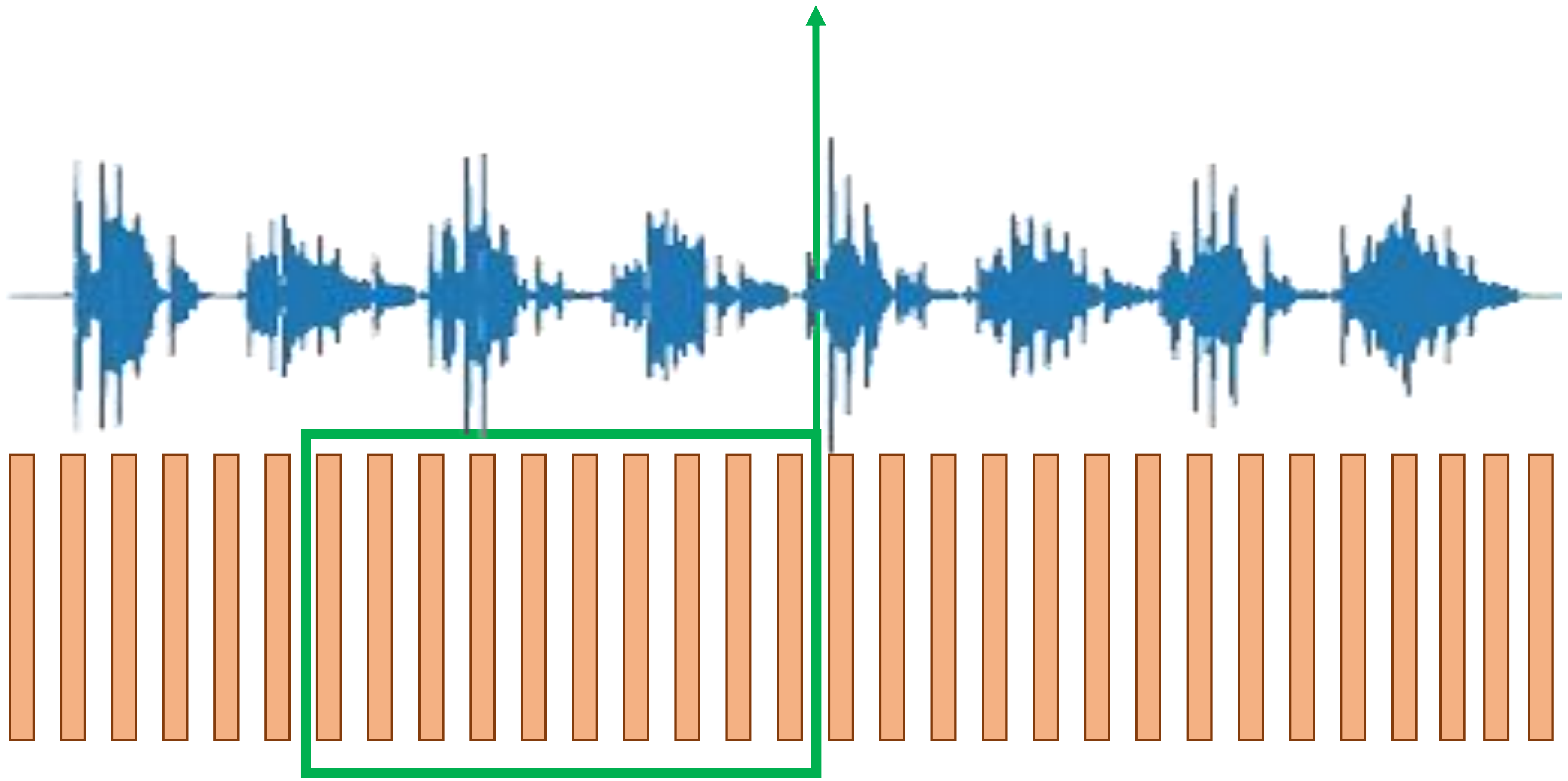
Audio frames with a smaller frame step

Chunk

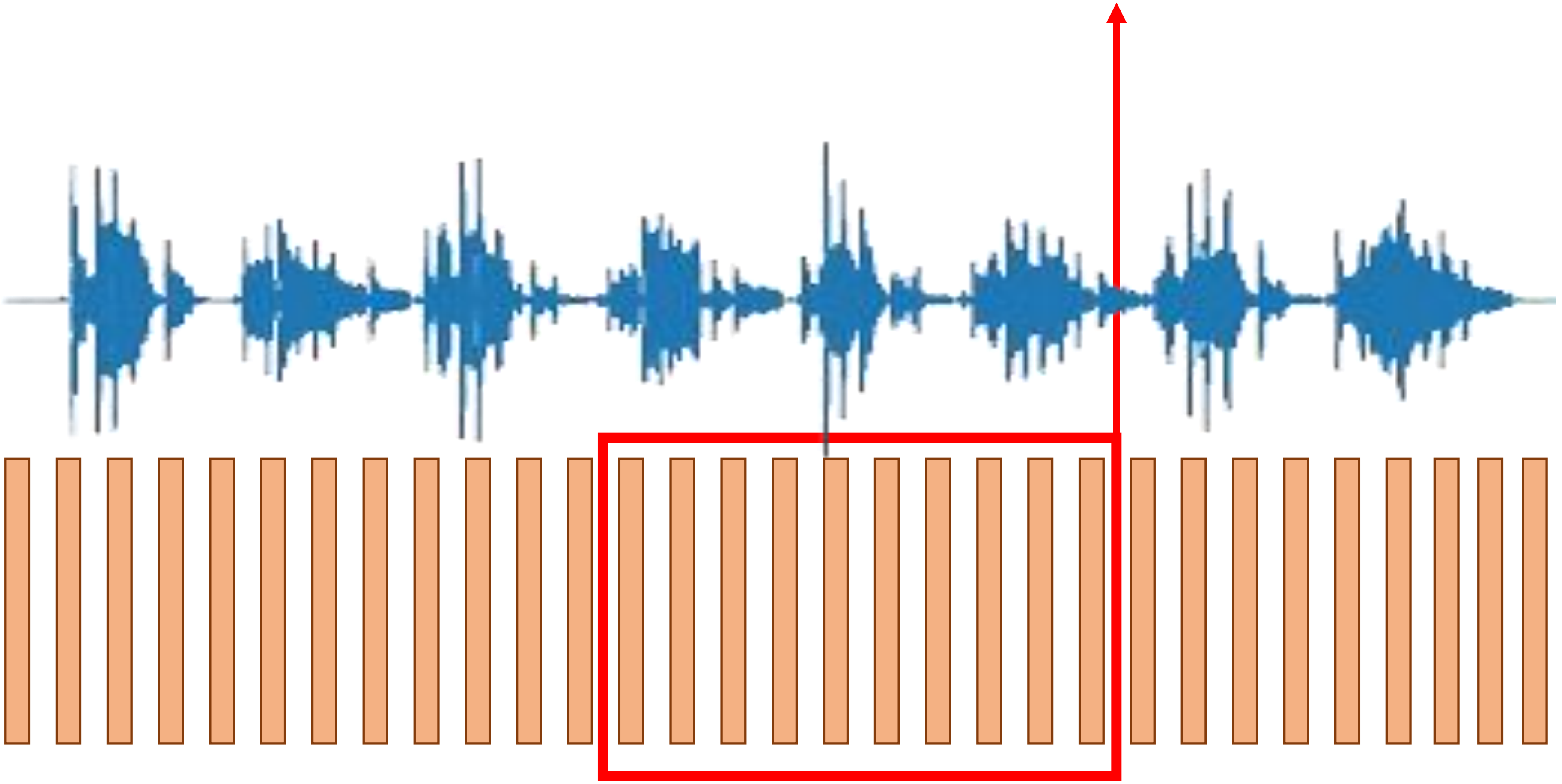
A larger segment of audio data that encompasses multiple consecutive frames



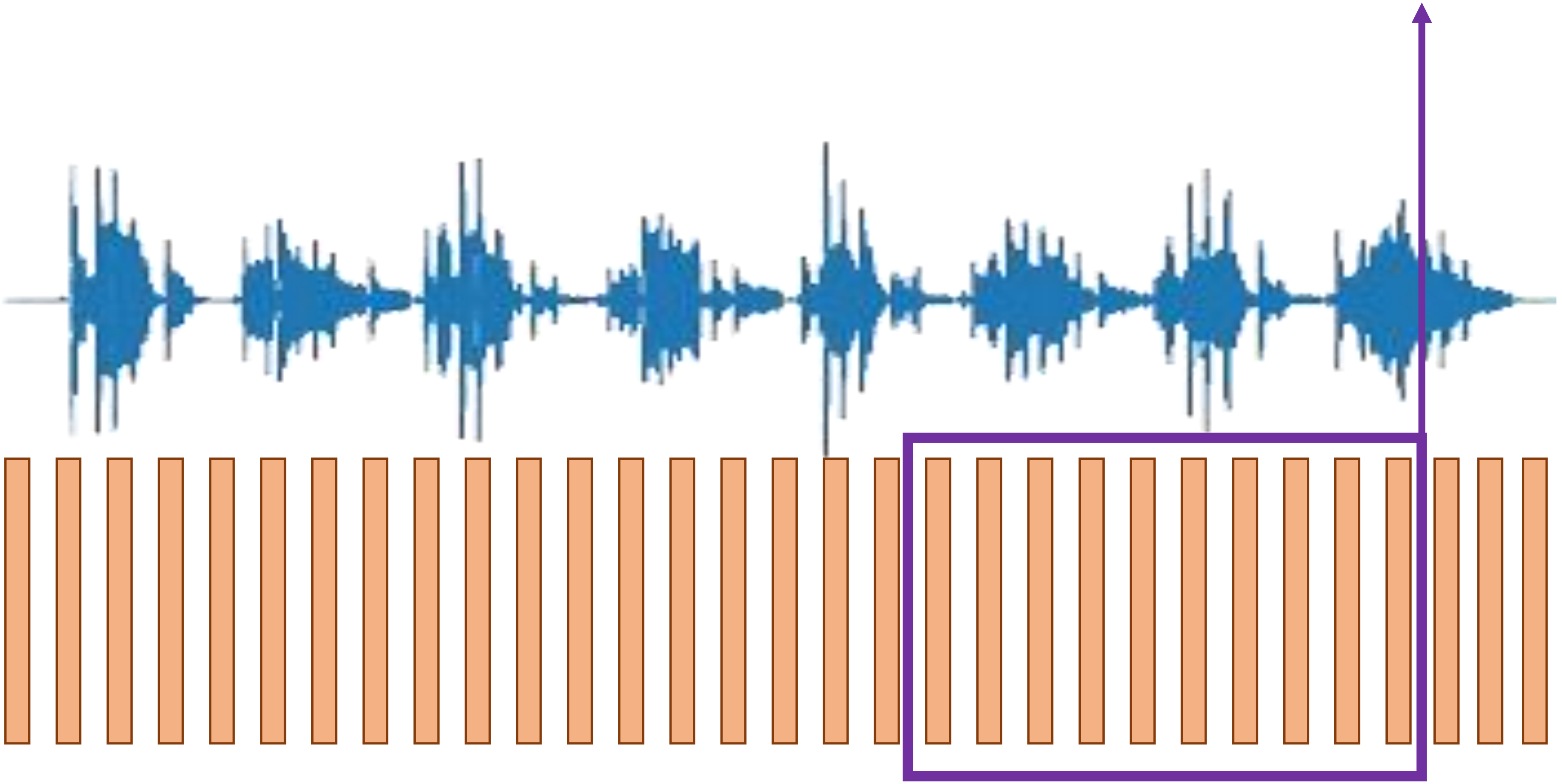
Chunk



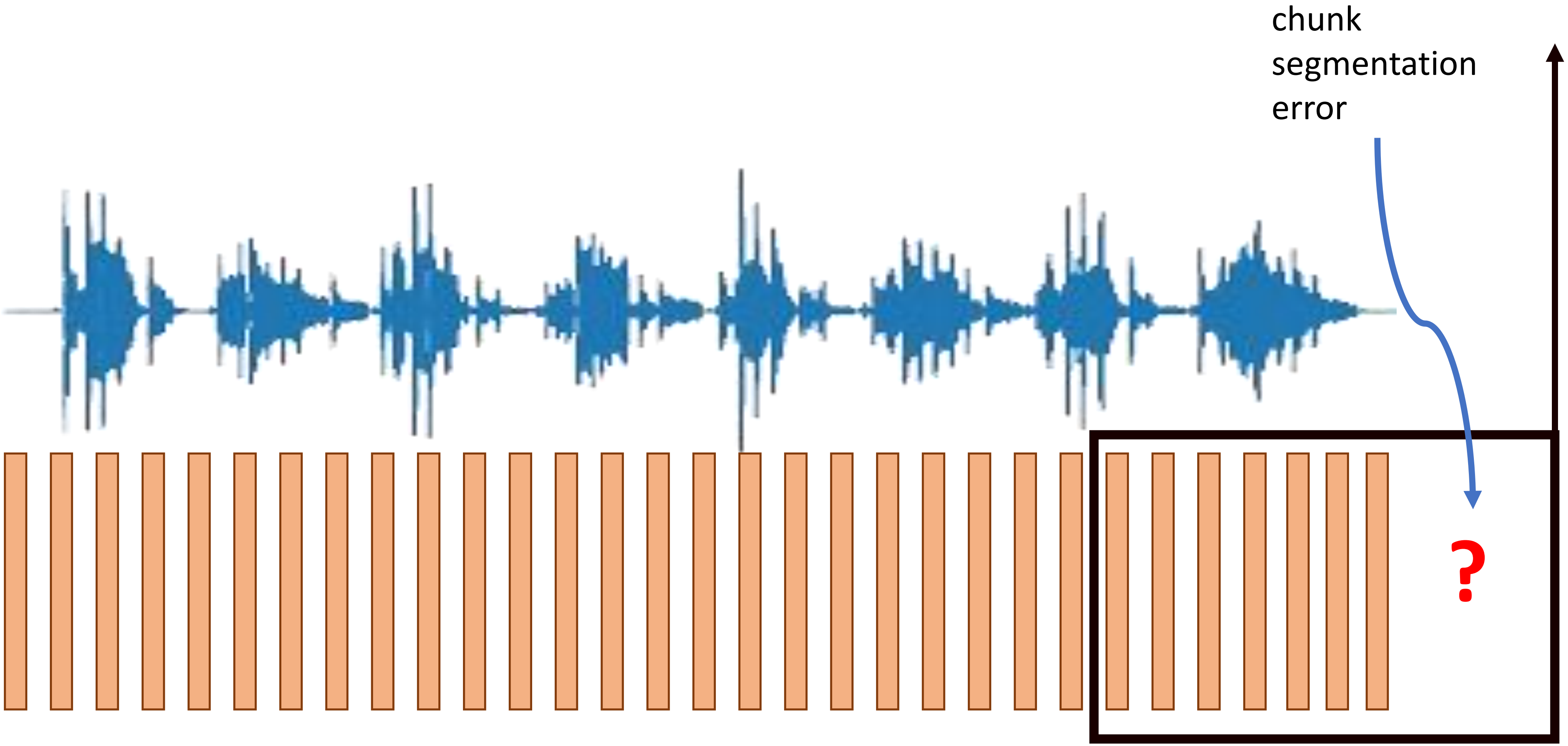
Chunk



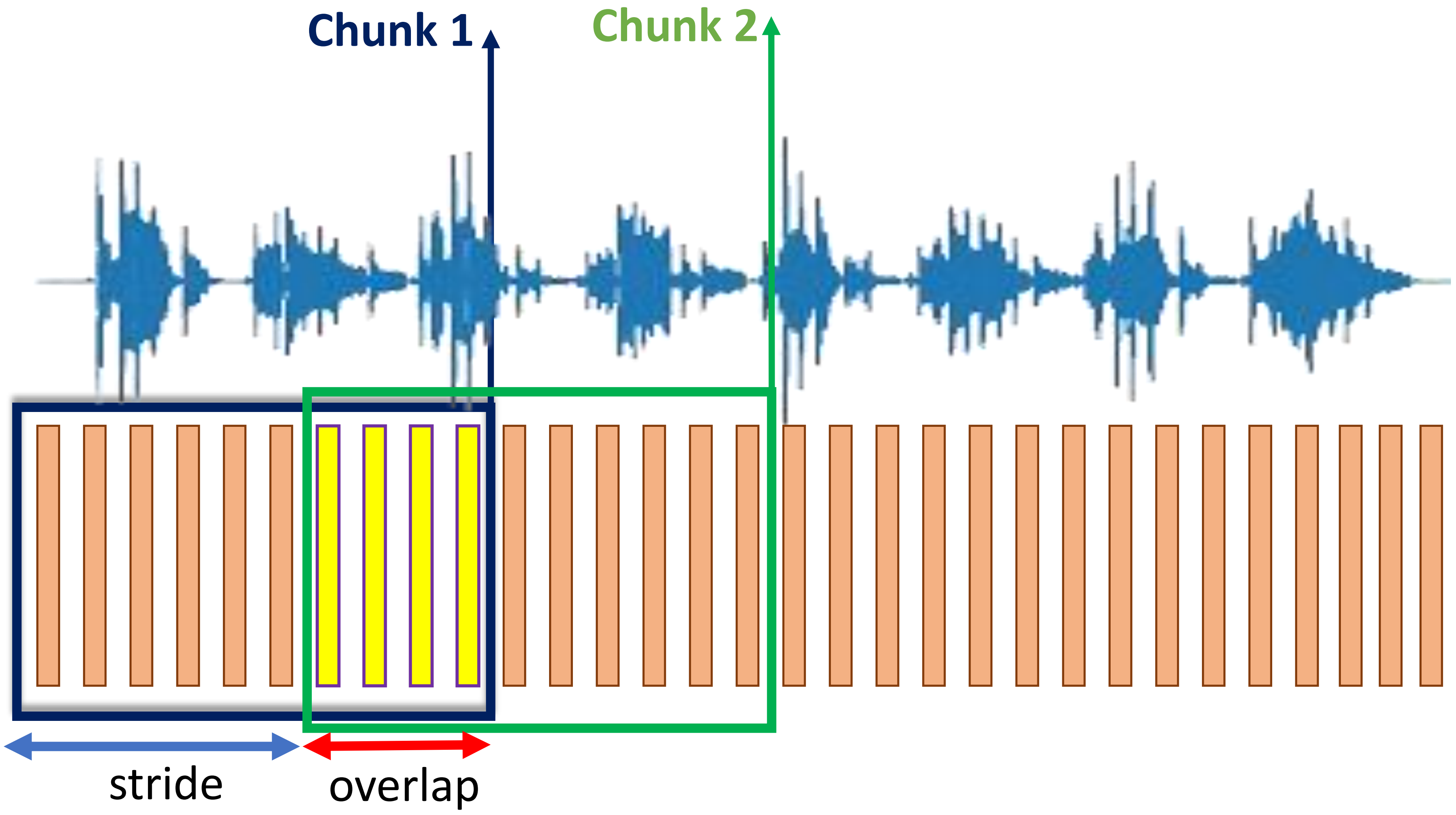
Chunk



Chunk



Stride & Overlap



Evolution of VAD Architectures

Late 1990s – Early 2010s

- Hidden Markov Models
- Gaussian Mixture Models
- Exploration of different features (MFCC, Mel)

2011 - 2017

- Multi Layer Perceptron
- Recurrent Neural Network
- Convolutional Neural Network

2017 onwards

- Long Short Term Memory
- Bidirectional Long Short Term Memory

Pyannote

1. **Feature Extraction:** Employs SincNet to identify the most important features. Implements a bank of band-pass filters where low and high cutoff frequencies are learned
2. **Architecture:** LSTM (type of RNN) layers for modeling temporal dependencies, and a feedforward layer for predicting frame-level speech probabilities.

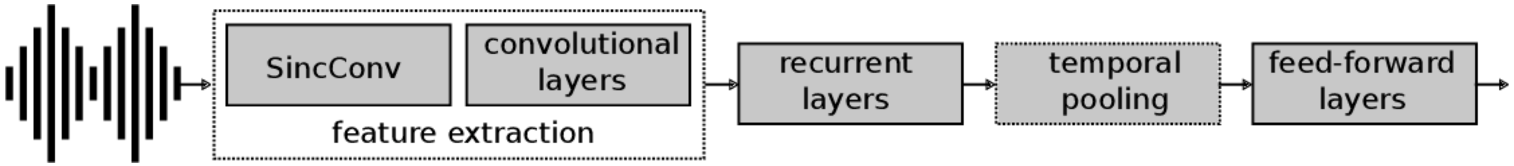


Table 1: *Voice activity detection // FA = false alarm rate (%) / Miss. = missed detection rate (%)*

VAD	AMI [8, 15]			DIHARD 3 [9]			VoxConverse [11]		
	FA	Miss.	FA+Miss.	FA	Miss.	FA+Miss.	FA	Miss.	FA+Miss.
silero_vad	9.4	1.7	11.0	17.0	4.0	21.0	3.0	1.1	4.2
dihard3 [9]	NA	NA	NA	4.0	4.2	8.2	NA	NA	NA
Landini et al. [12]	NA	NA	NA	NA	NA	NA	1.8	1.1	3.0
pyannote 1.1 [16]	6.5	1.7	8.2	4.1	3.8	7.9	4.5	0.3	4.8
Ours – *pyannote 2.0*	3.6	3.2	**6.8**	3.9	3.3	**7.3**	1.8	0.8	**2.5**

H. Bredin and A. Laurent, End-to-end speaker segmentation for overlap-aware resegmentation, 2021.

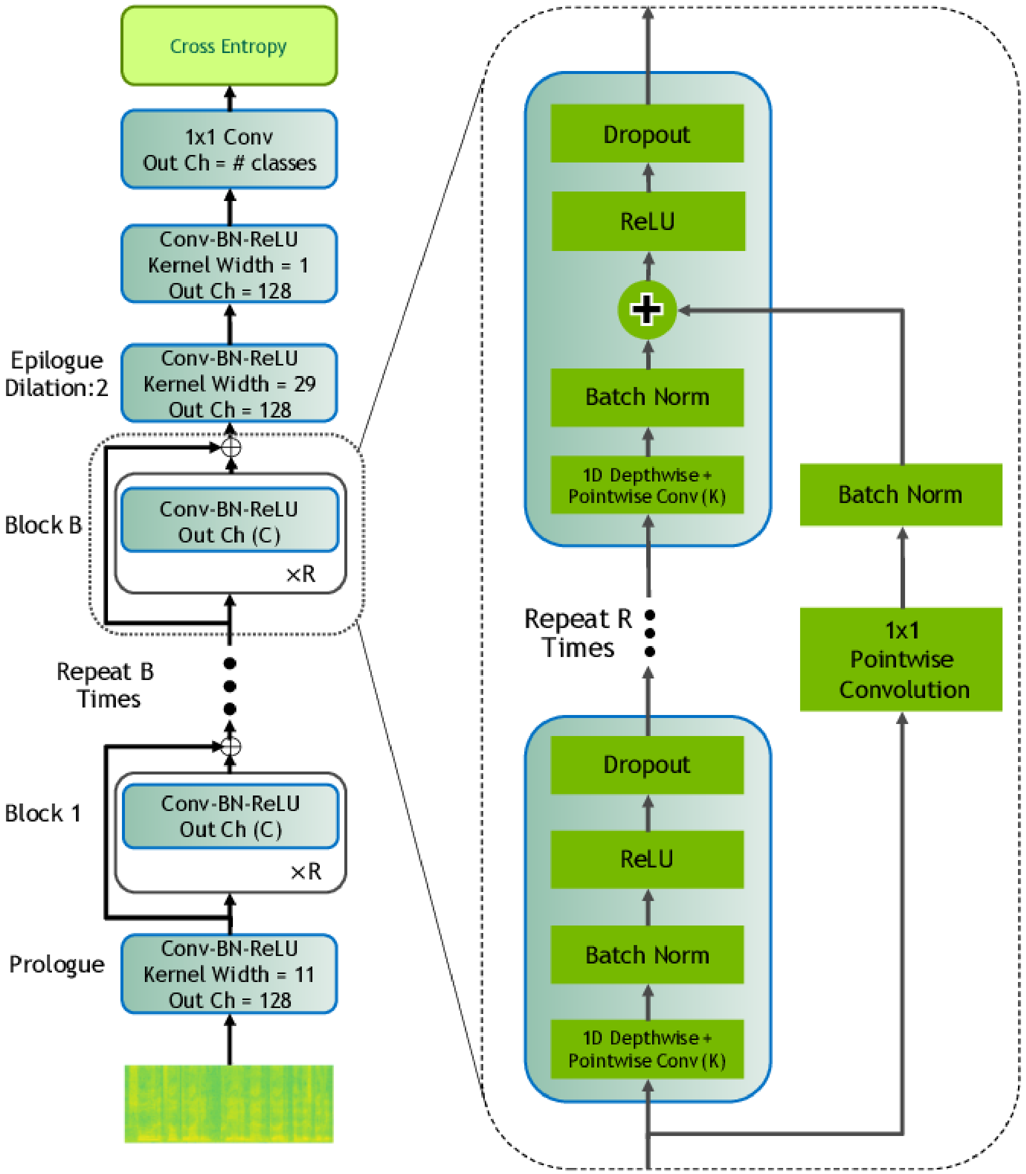
Silero

1. **Feature Extraction:** Employs Short-Time Fourier Transform (STFT) as a feature extraction technique.
2. **Adaptable Sampling Rates**: Works seamlessly with both 8kHz and 16kHz sampling rates, enhancing versatility.
3. **Real-time Detection**: Enables real-time voice activity detection by processing audio in manageable parts.
4. **Architecture:**
    1. Encoder consisting of convolutional layers
    2. Decoder with an LSTM layer
    3. Convolutional and sigmoid layers for output.

S. Team, Silero VAD: Pre-trained enterprise-grade voice activity detector (VAD), number detector and language classifier, https : / / github . com / snakers4/silero-vad, 2021

MarbleNet

1. **Feature Extraction:** uses Mel-Frequency Cepstral Coefficients (MFCC).
2. **Architecture:** Multiple CNN blocks.



F. Jia, S. Majumdar, and B. Ginsburg, Marblenet: Deep 1d time-channel separable convolutional neural network for voice activity detection, 2021.

Datasets

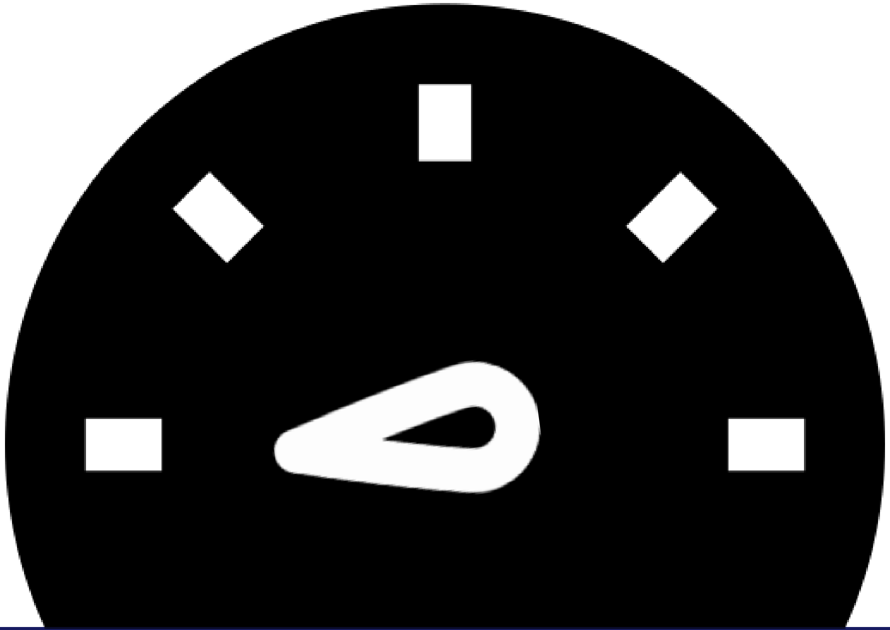
Parameters	AliMeeting[2]		DIHARD III (Full)[3]	
	Train	Eval	Train/Dev	Eval
Duration (in hours)	104.75	4	34.15	33.01
# Recordings	212	8	254	259
Overlap ratio (%)	42.27	24.20	10.70	9.35
Language	Mandarin		English (Not all native speakers)	
Domain	Meeting		Audiobooks, Broadcast Interview, Clinical, Court, Maptask, Meeting, Restaurant, Webvideo, etc. (11 in total)	
Audio File Format	.WAV		.FLAC	
Ground Truth Format	.TextGrid		.RTTM	

[2] F. Yu, S. Zhang, Y. Fu, et al., “M2met: The ICASSP 2022 multi-channel multi-party meeting transcription challenge,” CoRR, vol. abs/2110.07393, 2021.

[3] N. Ryant, P. Singh, V. Krishnamohan, et al., The third DIHARD diarization challenge, 2021.

Evaluation Metrics

Abbreviations:
TP: True Positive
TN: True Negative
FP: False Positive
FN: False Negative



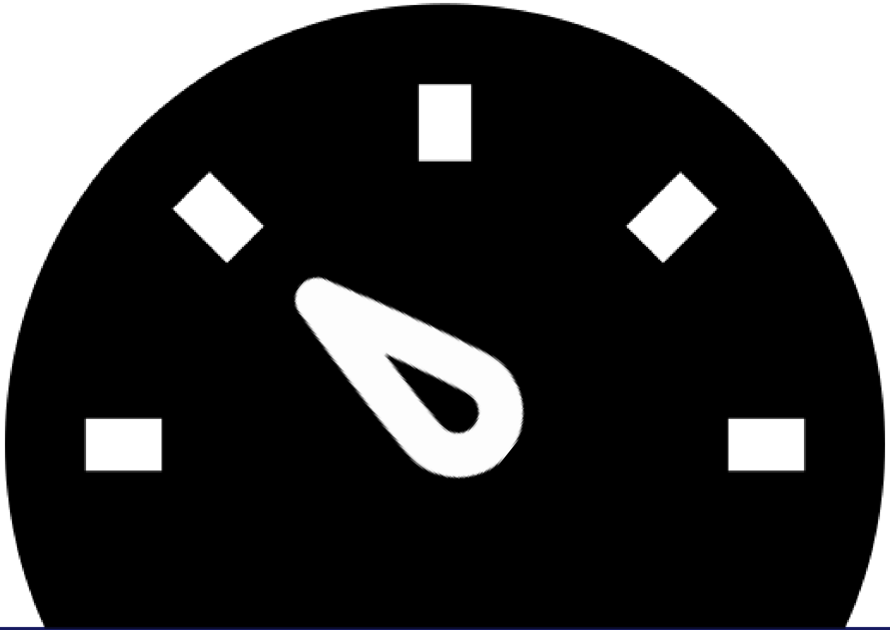
$$Accuracy = \frac{\#\, correct\ predictions}{\#\, total\ predictions} = \frac{TP + TN}{TP + FP + FN + TN}$$

Evaluation Metrics

Abbreviations:
TP: True Positive
TN: True Negative
FP: False Positive
FN: False Negative

$$Missed\ Detection\ Rate = \frac{FN}{FN + TP}$$

$$Accuracy = \frac{\#\ correct\ predictions}{\#\ total\ predictions} = \frac{TP\ +\ TN}{TP + FP + FN + TN}$$



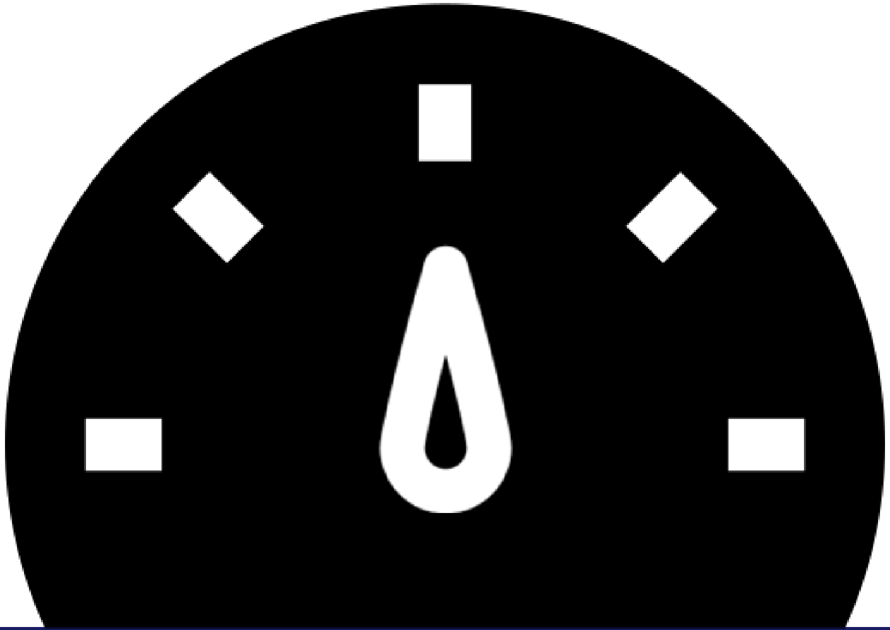
Evaluation Metrics

Abbreviations:
TP: True Positive
TN: True Negative
FP: False Positive
FN: False Negative

$$False\ Alarm\ Rate = \frac{FP}{FP + TN}$$

$$Missed\ Detection\ Rate = \frac{FN}{FN + TP}$$

$$Accuracy = \frac{\#\ correct\ predictions}{\#\ total\ predictions} = \frac{TP\ +\ TN}{TP + FP + FN + TN}$$



Evaluation Metrics

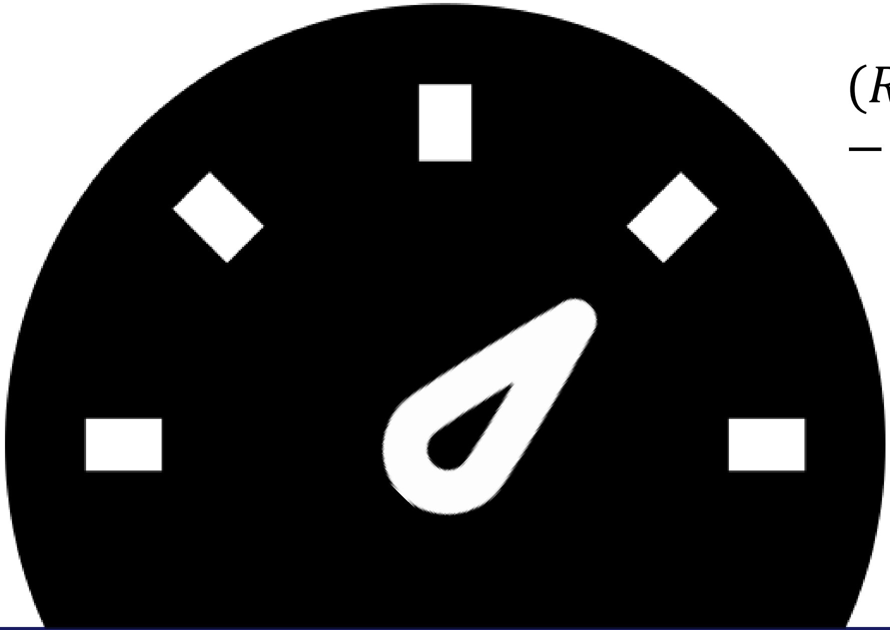
Abbreviations:
TP: True Positive
TN: True Negative
FP: False Positive
FN: False Negative

$$False\ Alarm\ Rate = \frac{FP}{FP + TN}$$

$$Missed\ Detection\ Rate = \frac{FN}{FN + TP}$$

$$ROC - AUC$$
$$(Receiver\ Operating\ Characteristic - Area\ Under\ the\ Curve)$$

$$Accuracy = \frac{\#\ correct\ predictions}{\#\ total\ predictions} = \frac{TP\ +\ TN}{TP + FP + FN + TN}$$



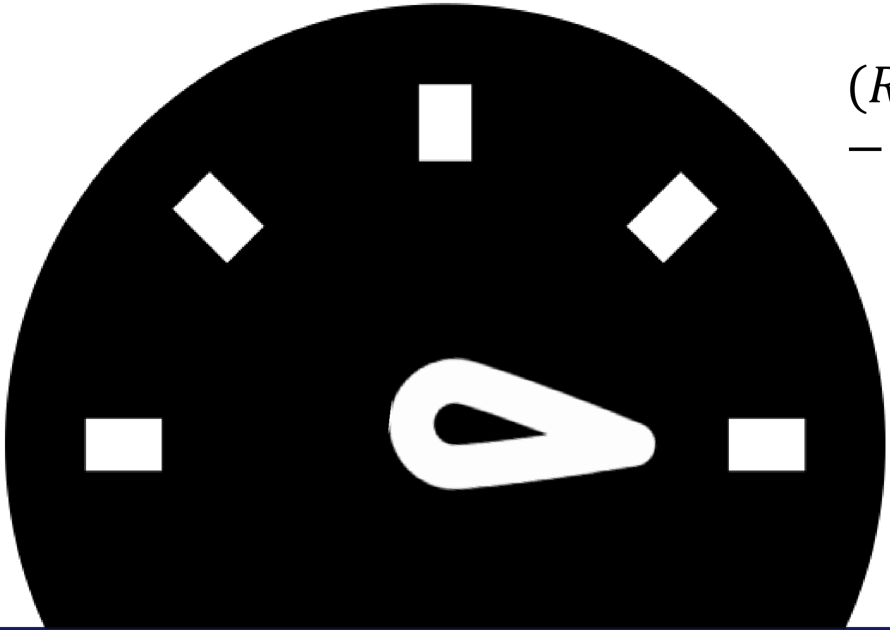
Evaluation Metrics

Abbreviations:
TP: True Positive
TN: True Negative
FP: False Positive
FN: False Negative

$$False\ Alarm\ Rate = \frac{FP}{FP + TN}$$

$$Missed\ Detection\ Rate = \frac{FN}{FN + TP}$$

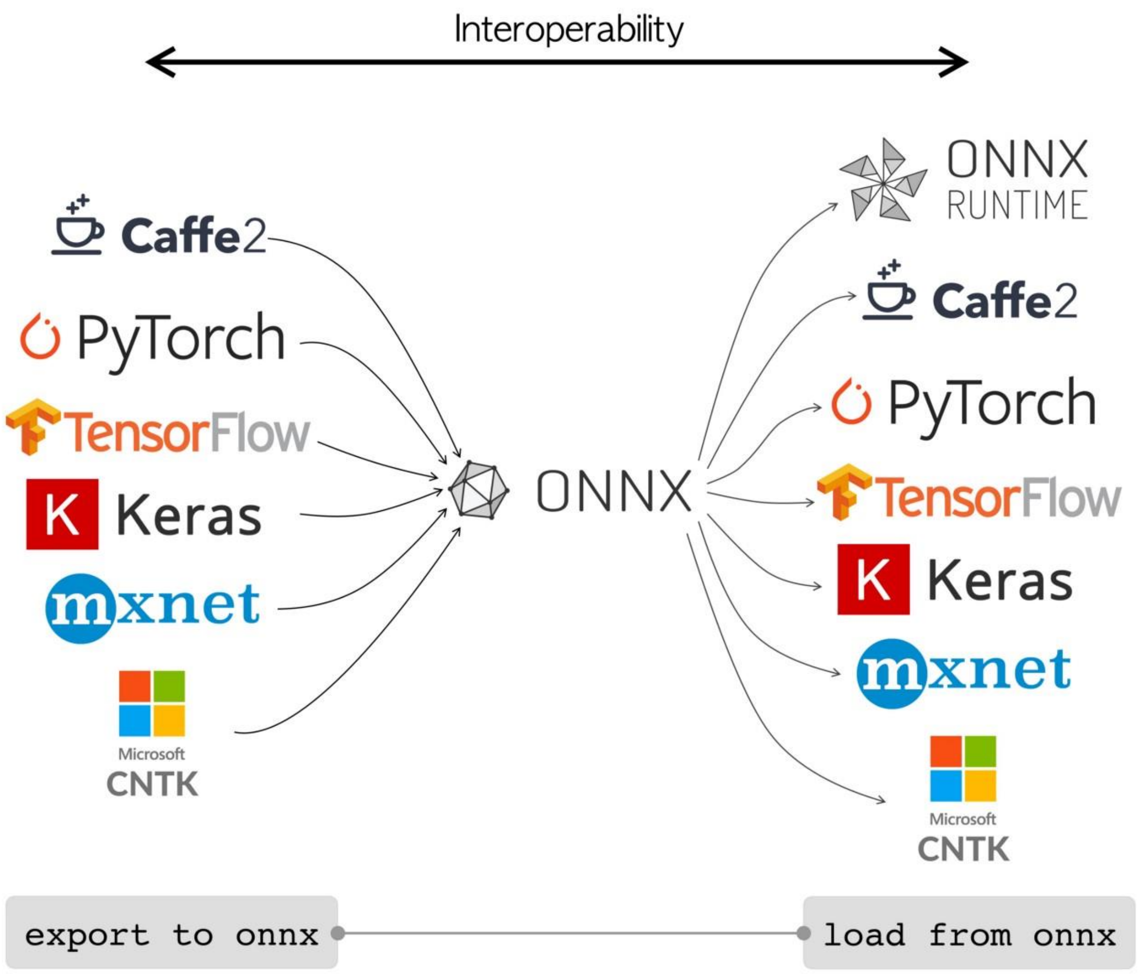
$$ROC - AUC\ (Receiver\ Operating\ Characteristic - Area\ Under\ the\ Curve)$$



$$Accuracy = \frac{\#\ correct\ predictions}{\#\ total\ predictions} = \frac{TP\ +\ TN}{TP + FP + FN + TN}$$

$$Real\ Time\ Factor = \frac{Time\ taken\ to\ process\ a\ file}{Duration\ of\ the\ file}$$

Open Neural Network Exchange



INTRODUCTION

- Background
- Objective and Scope

LITERATURE REVIEW

- Audio Concepts
- Pyannote, Silero, MarbleNet
- Evaluation Metrics
- ONNX

METHODOLOGY & RESULTS

- **Experiment 1**
- **Experiment 2**
- **Experiment 3**
- **Experiment 4**
- **Experiment 5**

CONCLUSION

- Summary
- Contribution
- Future Work

Overall Approach

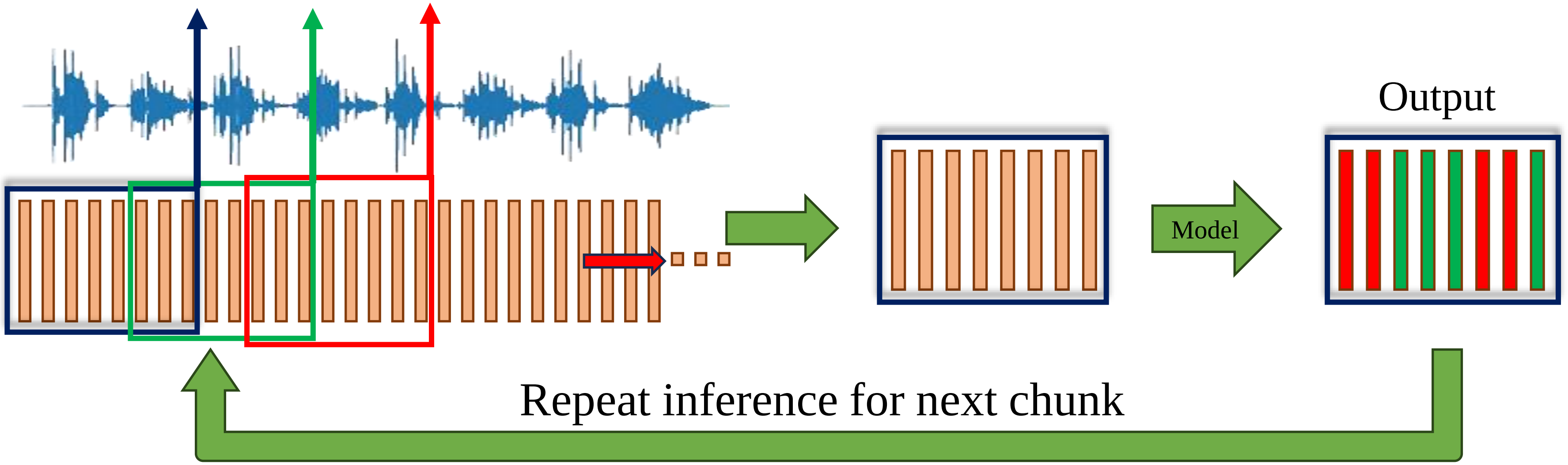
- Survey performance of SOTA models
- Understand the effects varying chunk sizes and strides
- DIHARD III domain-wise analysis
- Retrain best performing model
- ONNX for inference

Operating Environment:

- Aspire 2A GPU cluster provided by National Super Computing Centre (NSCC).
- **CPU:** HPE Cray EX 2x AMD EPYC Millan 7713
- **GPU:** NVIDIA A100-40G SXM

① Initial Inference

- **Pyannote** was obtained by the replication of training procedure as described by Bredin et al. with chunk size 5s and stride as 4s.
- **Silero** was obtained from the official Github repository made by the Silero team.
- **MarbleNet** was obtained through a collaboration with AI Singapore.



① Initial Inference

DIHARD III	Model	Chunk size	Stride	Accuracy	MDR	FAR	ROC-AUC	RTF
	Pyannote	5s	5s	**0.97**	1.00	2.00	**0.941**	0.095
	Silero			0.83	11.1	6.10	0.795	0.022
	MarbleNet			0.77	1.60	21.6	0.536	**0.011**

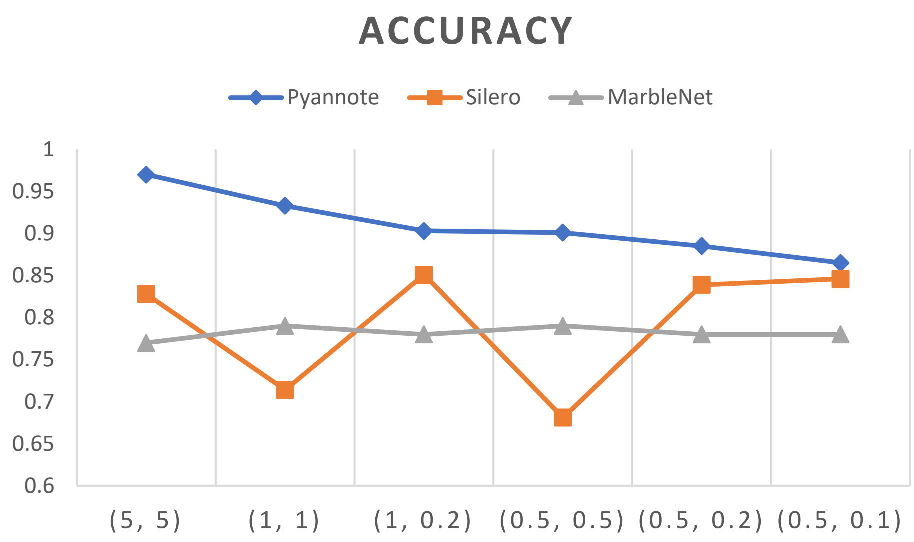
AliMeeting	Model	Chunk size	Stride	Accuracy	MDR	FAR	ROC-AUC	RTF
	Pyannote	5s	5s	**0.95**	**3.00**	**1.7**	**0.777**	**0.011**
	Silero			0.74	21.9	4.0	0.624	0.038
	MarbleNet			0.92	8.7	4.8	0.732	0.047

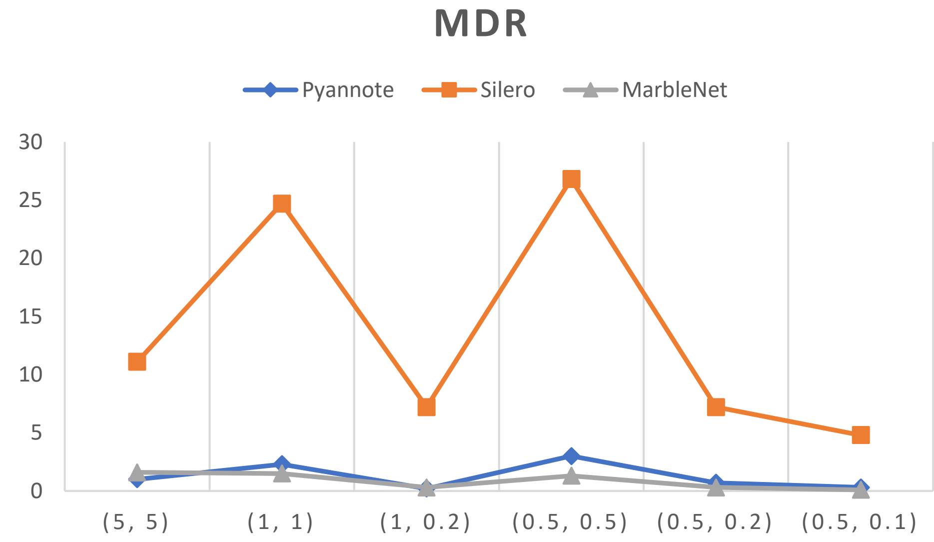
- Pyannote demonstrates superior performance compared to Silero and MarbleNet under the conditions of a 5-second chunk size and stride on both the DIHARD III and AliMeeting datasets.

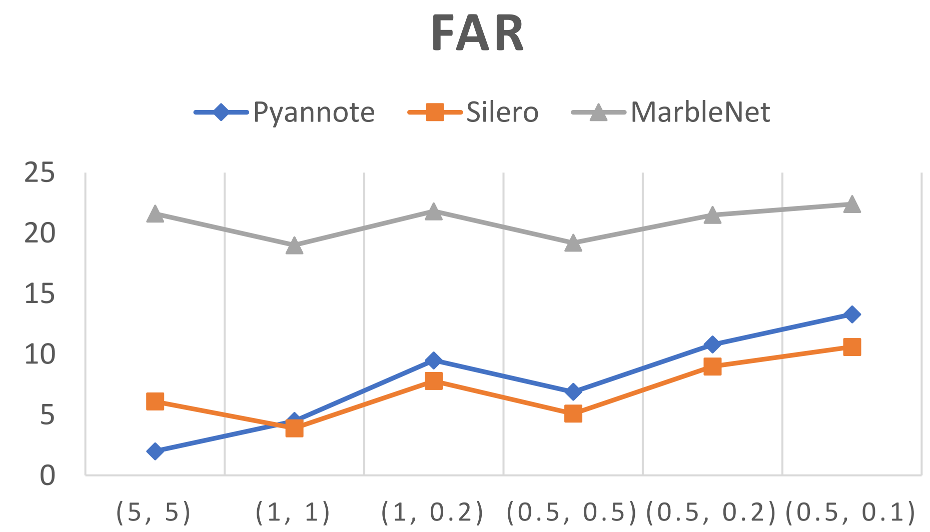
② Effect of Varying Chunk Sizes and Strides

- Aim to investigate the impact of varying chunk sizes and strides on the performance of Pyannote, Silero, and MarbleNet.
- Some important hyperparameters
  - Range of chunk sizes tested: 0.5s – 5s
  - Range of strides tested: 0.1s – 5s
  - Prediction threshold: 0.5
  - Dataset: DIHARD III Eval set

② Effect of Varying Chunk Sizes and Strides





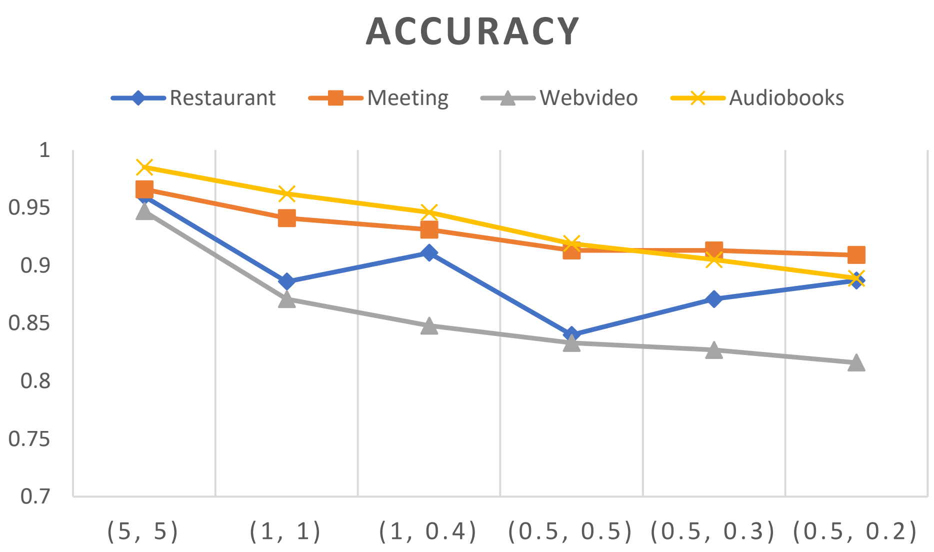


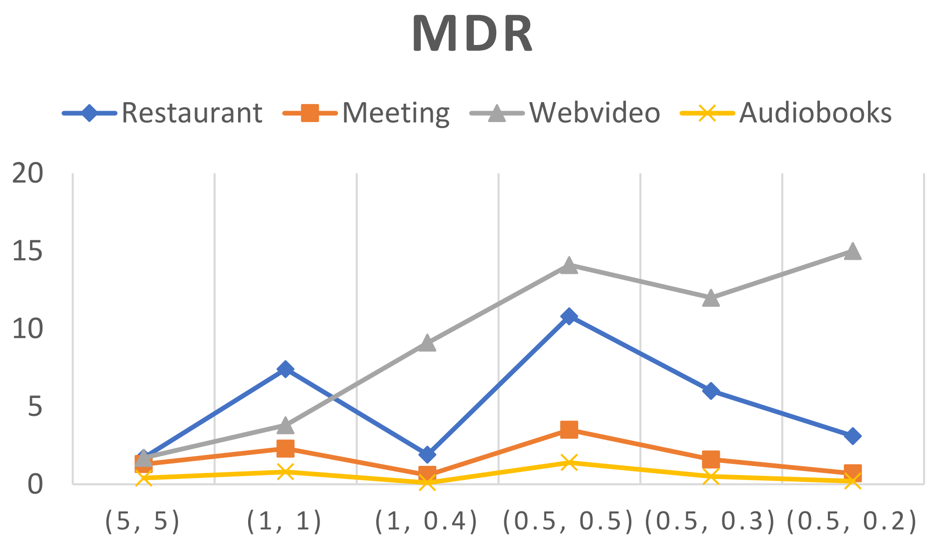
- **X Axis Legend:**
  - **(p, q):** p is chunk size, q is stride
- **Conclusion:** Pyannote exhibits the best performance with higher average evaluation metrics

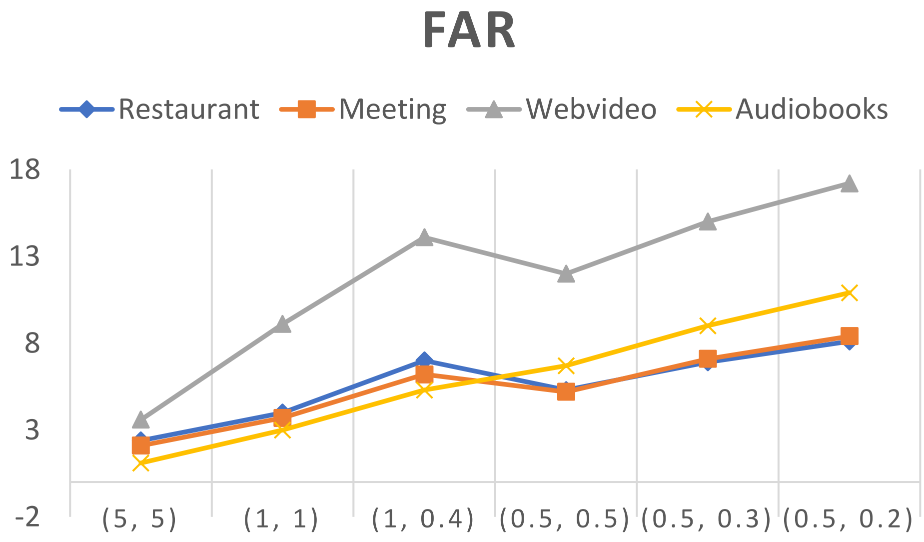
③ DIHARD III domain-wise analysis with Pyannote (5,4)

- From Experiment 1 and 2, Pyannote is the best performing model.
- Aim to understand the performance of Pyannote for different domains of DIHARD III
- Domains of interest: Restaurant, Meeting and Webvideo

③ DIHARD III domain-wise analysis with Pyannote (5,4)







Conclusion

- Larger chunk size ⟹ Higher accuracy & ROC-AUC
- For fixed chunk size, decreasing stride ⟹ decreasing accuracy & ROC-AUC (except restaurant)

③ DIHARD III domain-wise analysis with Pyannote (5,4)

Restaurant

Chunk	Stride	Acc	MDR	FAR	ROC-AUC	RTF
5s	5s	0.960	1.7	2.4	0.888	0.078
1s	1s	0.886	7.4	4.0	0.776	0.076
1s	0.4s	0.911	1.9	7.0	0.680	0.035
0.5s	0.5s	0.840	10.8	5.3	0.704	0.070
0.5s	0.3s	0.871	[illegible]	[illegible]	[illegible]	[illegible]
0.5s	0.2s	0.887	[illegible]	[illegible]	[illegible]	[illegible]

Meeting

Chunk	Stride	Acc	MDR	FAR	ROC-AUC	RTF
5s	5s	0.966	1.3	2.1	0.929	0.088
1s	1s	0.941	2.3	3.7	0.875	0.074
1s	0.4s	0.931	0.6	6.2	0.809	0.034
0.5s	0.5s	0.913	3.5	5.2	0.821	0.070
[illegible]	[illegible]	[illegible]	[illegible]	7.1	0.778	0.048
[illegible]	[illegible]	[illegible]	[illegible]	8.4	0.747	0.032

For small chunk size and stride, decent performance for chunk size 1s and stride 0.4s is observed

Webvideo

Chunk	Stride	Acc	[illegible]	[illegible]	[illegible]	[illegible]
5s	5s	0.947	1.7	3.6	0.917	0.080
1s	1s	0.871	3.8	9.1	0.812	0.087
1s	0.4s	0.848	9.1	14.1	0.747	0.030
0.5s	0.5s	0.833	14.1	12.0	0.759	0.070
0.5s	0.3s	0.827	12.0	15.0	0.721	0.048
0.5s	0.2s	0.816	15.0	17.2	0.693	0.031

Audiobooks

[illegible]	[illegible]	[illegible]	[illegible]	FAR	ROC-AUC	RTF
5s	5s	0.985	0.4	1.1	0.973	0.091
1s	1s	0.962	0.8	3.0	0.929	0.075
1s	0.4s	0.946	0.1	5.3	0.883	0.029
0.5s	0.5s	0.919	1.4	6.7	0.843	0.070
0.5s	0.3s	0.905	0.5	9.0	0.797	0.049
0.5s	0.2s	0.889	0.2	10.9	0.756	0.032

4 Retraining Pyannote

- Aim to train Pyannote on different values of chunk sizes and strides
- Model nomenclature format: **Py (x, y)**
  - **Py:** Pyannote
  - **x:** Training chunk size (s)
  - **y:** Training stride (s)

Layer	Hyperparameter
SincNet	stride
LSTM	hidden_size
	num_layers
	bidirectional
	monolithic
	dropout
Linear	hidden_size
	num_layers

Other Hyperparameters
Max. no. of epochs
Batch size
Chunk size (s)
stride (s)
Frame size (s)
Frame shift (s)
Optimiser
Learning rate
Max. Gradient Norm
Number of channels
Sample rate (Hz)

4 Training Pyannote

Domain	Model	Acc	MDR	FAR	ROC-AUC	RTF
Restaurant	Py (5, 4)	0.911	1.9	7.0	0.680	0.035
	Py (1, 0.4)	0.891	3.0	7.9	0.642	0.034
Webvideo	Py (5, 4)	0.848	1.1	14.1	0.747	0.030
	Py (1, 0.4)	0.831	3.9	13.0	0.761	0.035
Meeting	Py (5, 4)	0.931	0.6	6.2	0.809	0.034
	Py (1, 0.4)	0.848	0.9	14.2	0.596	0.034
Audiobooks	Py (5, 4)	0.946	0.1	5.3	0.883	0.029
	Py (1, 0.4)	0.976	0.7	1..7	0.956	0.035
DIHARD III Eval set	Py (5, 4)	0.920	0.6	7.4	0.821	0.029
	Py (1, 0.4)	0.920	1.8	6.2	0.846	0.034

1. Restaurant Domain: Py (5, 4) achieves higher accuracy (0.911) than Py (1, 0.4) (0.891), showing superior event detection performance.

2. Consistency Across Domains: Py (5, 4) consistently outperforms Py (1, 0.4) in Webvideo and Meeting domains, maintaining lower MDR and FAR values.

3. Chunk Size and Stride Impact: Larger chunk size and stride (Py (5, 4)) generally lead to better performance across domains.

4. Audiobooks Exception: Py (1, 0.4) achieves higher accuracy (0.976) than Py (5, 4) (0.946), indicating differences in generalization capabilities.

4 Training Pyannote

Py (3, 2)							
Chunk	Stride	Domain	Acc	MDR	FAR	ROC-AUC	RTF
3s	3s	Restaurant	0.723	24.8	2.9	0.731	0.035
		Webvideo	0.775	17.3	5.2	0.808	0.036
		Meeting	0.834	15.9	0.8	0.874	0.034
		DIHARD III Eval set	0.897	8.7	1.6	0.908	0.036

Py (1.5, 1)							
Chunk	Stride	Domain	Acc	MDR	FAR	ROC-AUC	RTF
1.5s	1.5s	Restaurant	0.692	29.1	1.7	0.751	0.035
		Webvideo	0.767	19.1	4.2	0.807	0.036
		Meeting	0.831	16.3	0.6	0.878	0.034
		DIHARD III Eval set	0.882	10.5	1.2	0.903	0.036

Py (0.75, 0.5)							
Chunk	Stride	Domain	Acc	MDR	FAR	ROC-AUC	RTF
0.75s	0.75s	Restaurant	0.645	34.3	1.1	0.742	0.035
		Webvideo	0.776	17.8	4.6	0.810	0.036
		Meeting	0.834	15.8	0.7	0.880	0.034
		DIHARD III Eval set	0.887	9.8	1.5	0.902	0.036

- While larger configurations like Py (5, 4) yield superior performance in most challenging domains, smaller configurations like Py (1.5, 1) and Py (0.75, 0.5) may excel in less noisier domains.

4 Training Pyannote

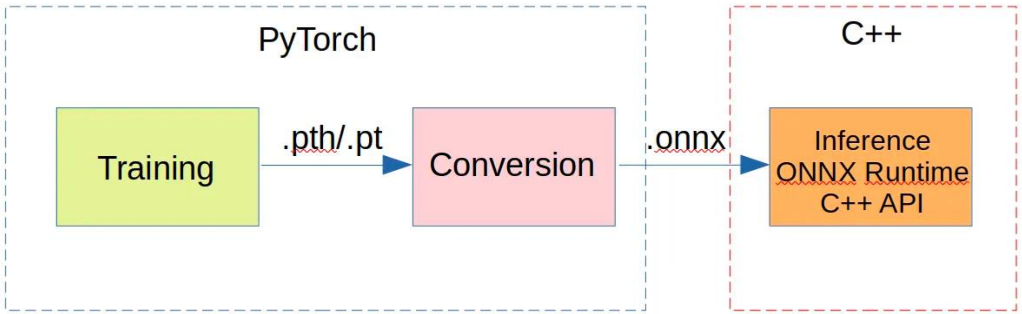
Comparing Py (5, 4) with Model K = Py (0.1875, 0.125)						
Domain	**Model**	**Acc**	**MDR**	**FAR**	**ROC-AUC**	**RTF**
Restaurant	Py (5, 4)	0.727	21.9	5.4	0.640	0.035
	Model K	0.622	36.1	1.8	0.714	0.035
Webvideo	Py (5, 4)	0.772	9.8	13.0	0.702	0.030
	Model K	0.755	18.7	5.8	0.784	0.035
Meeting	Py (5, 4)	0.809	13.7	5.4	0.755	0.034
	Model K	0.801	18.8	1.2	0.848	0.034
DIHARD III Eval set	Py (5, 4)	0.842	7.3	8.5	0.754	0.029
	Model K	0.869	11.2	1.9	0.885	0.034

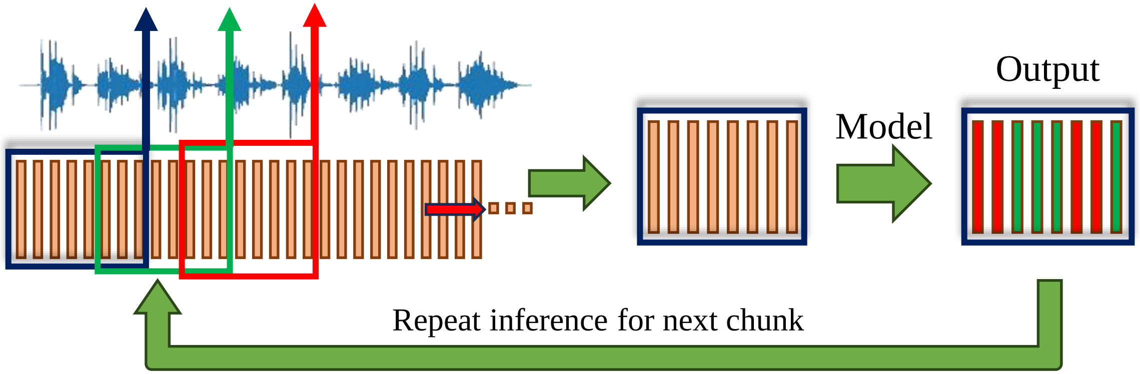
Inference chunk size and stride = 0.1875s

1. Models trained on small chunk sizes and strides may not perform optimally during inference with same configuration.
2. Py (5,4) demonstrates robustness and generalization across various chunk sizes and strides.
3. Larger chunk sizes, like in Py (5,4), provide more context to the BLSTM-based model, resulting in accurate predictions.

5 ONNX for Inference

- Aim to study the difference in inference times using Python and C++ based models
- Model Configuration:
  - Model: Py (5,4)
  - Inference chunk size: 5s
  - Inference stride: 5s
  - Prediction threshold: 0.5





5 ONNX for Inference

S.No.	Length of Audio	Python	C++
1	616.803s	33.08s	2s
2	49s	1.81s	<1s
3	599.952s	26.07s	1.8s
4	368.43s	15.29s	1.2s
5	33hrs 28s	58min 20s	12min 3s

- Significant and consistent reduction in inference time when using the C++ based model.
- This scalability is crucial for real-time VAD systems that may need to process long audio recordings efficiently.

INTRODUCTION

- Background
- Objective and Scope

LITERATURE REVIEW

- Audio Concepts
- Pyannote, Silero, MarbleNet
- Evaluation Metrics
- ONNX

METHODOLOGY & RESULTS

- Experiment 1
- Experiment 2
- Experiment 3
- Experiment 4
- Experiment 5

CONCLUSION

- Contribution
- Future Work

Contribution

01 Comprehensive evaluation of leading VAD models

02 Performed domain-wise analysis of Pyannote

03 Retrained Pyannote model with varying hyperparameters and developed a SOTA model

04 Explored strategies for enhancing scalability and production readiness

Future Work

- **01** Investigate ensemble learning techniques to enhance VAD performance.
- **02** Explore methods for adaptive VAD systems to dynamically adjust to changing environments.
- **03** Expand VAD models to multi-class classification for speaker counting problem
- **04** Incorporate transfer learning to leverage pre-trained models and improve performance.

Thank You!

Questions?