NISA (Non-Intrusive Signal Analysis) Development, Release, Noise Type Classifier

Lucia Eve Berger

M. Sci in Software Engineering

Managers:

DoYeong Kim, Dushyant Sharma

Outline

Objectives

Pipeline

- NISA++
- Feature extractions
- Model Selection

Results

Integration & Release

Objectives

Expand the NISA++ development, using alternative feature extraction and smaller architectures

- Experiment with CNN architectures
- More flexible data provision and feature extraction

High Level

Refactor the NISA framework

- More flexible training and testing {controlled via config}
- Exploited latest CNN technology and feature extraction

NISA++ Release

- Released 16K, 8K model
- Released the SwishNet VAD for general use

Developed Model for Noise Type Classification

- Classification model for the Noise Type/Codec Model

NISA Pipeline

Pipeline

Preprocessing

Selection & corruption

Training

Supervised problem

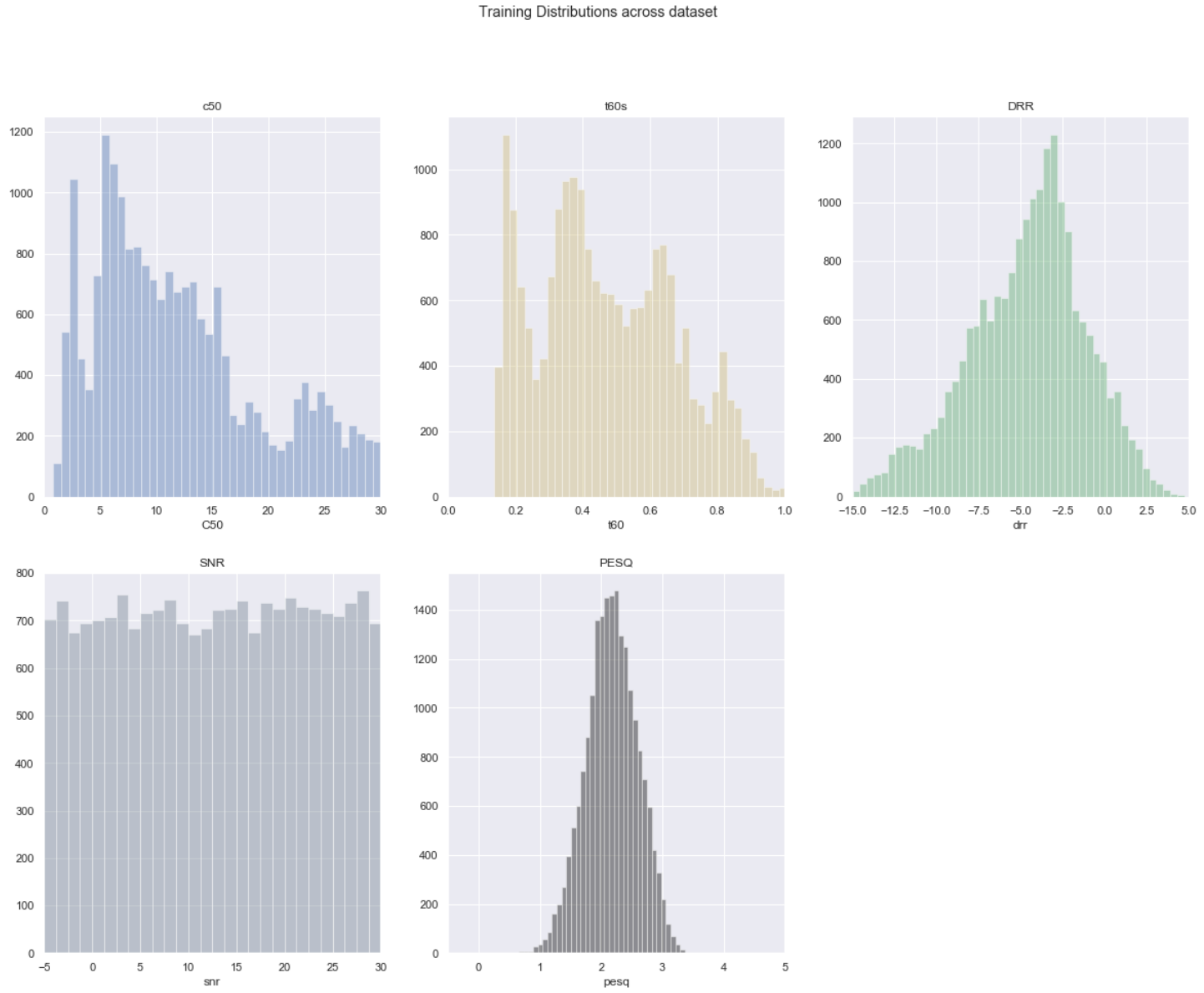
Testing

Comparing against larger baselines

Training NISA

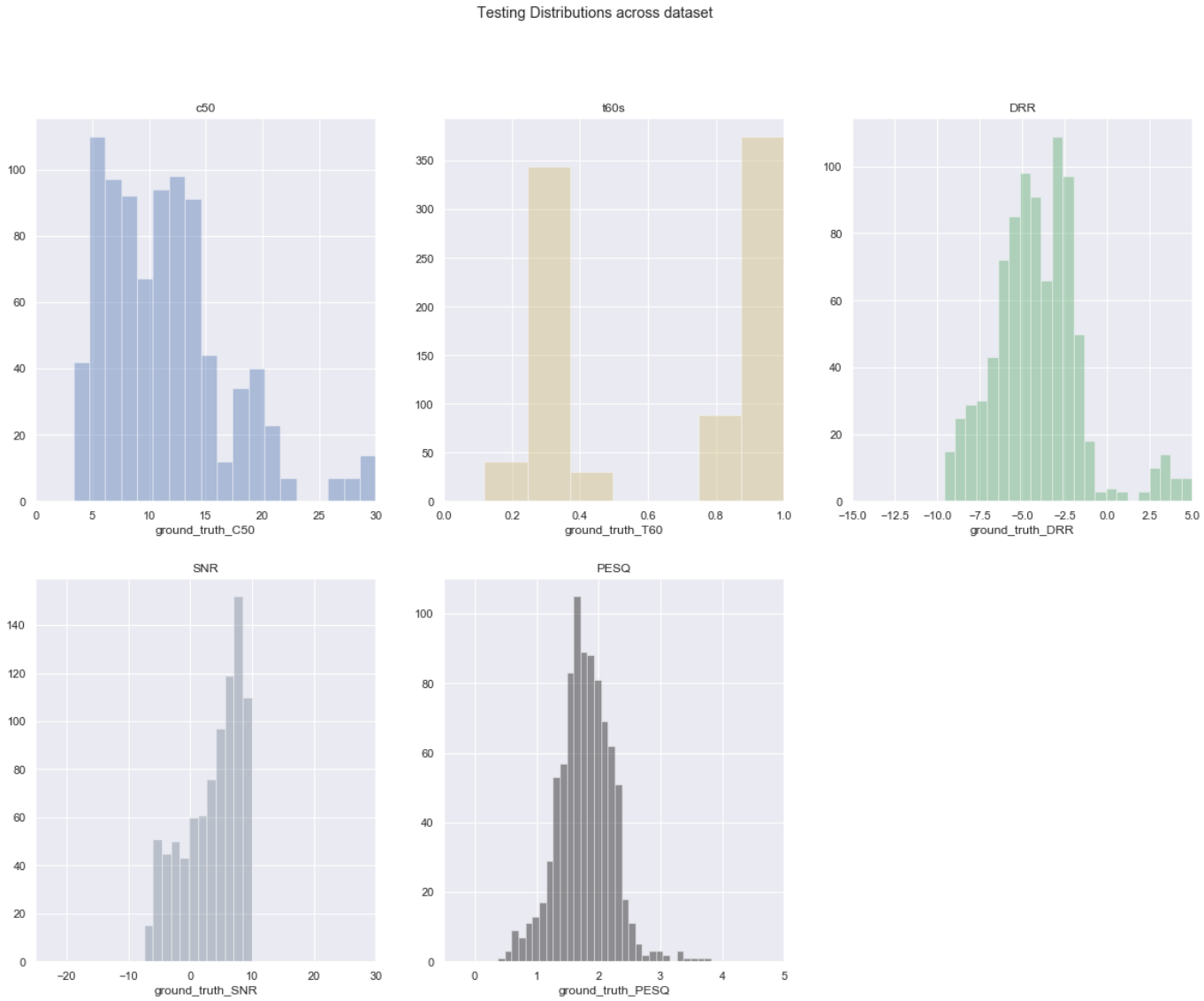
- Wallstreet journal corpus
- Sampled **large range** of metrics
  - Sampled within bins
  - Augmentation
- ~20,038 utterances

Parameter	Definition
C50	Speech clarity
T60	Estimators of room reverberation time
SNR	Signal to noise ratio
DRR	Direct-to-reverberation-ratio
PESQ	Perceptual Evaluation of Speech Quality
VAD	Voice activity detection



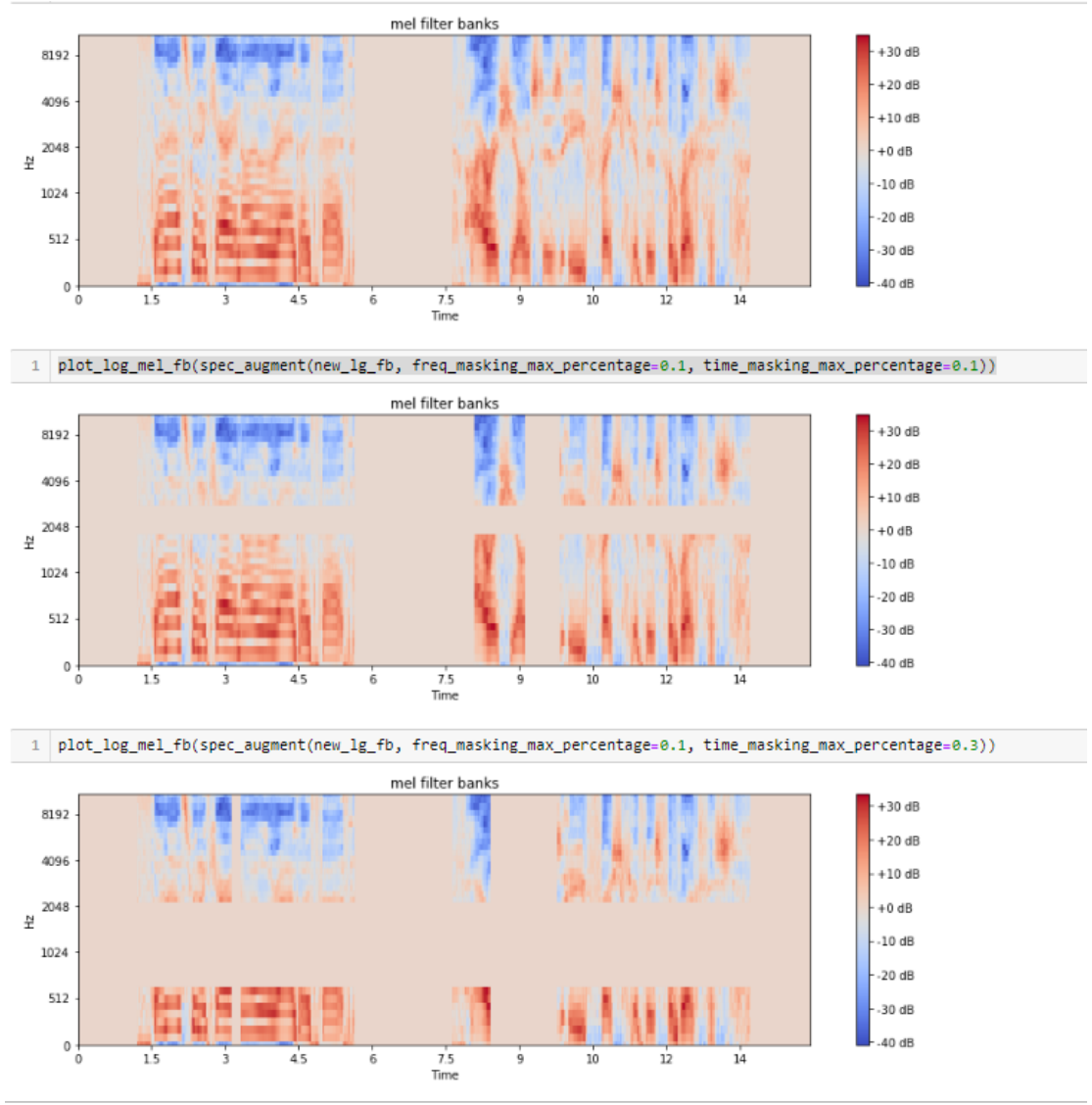
Testing Data {ASYM}

- Corrupted with corruption toolkit
- Highly noisy and reverberant
- Subset of libre-test data



Feature Extractions

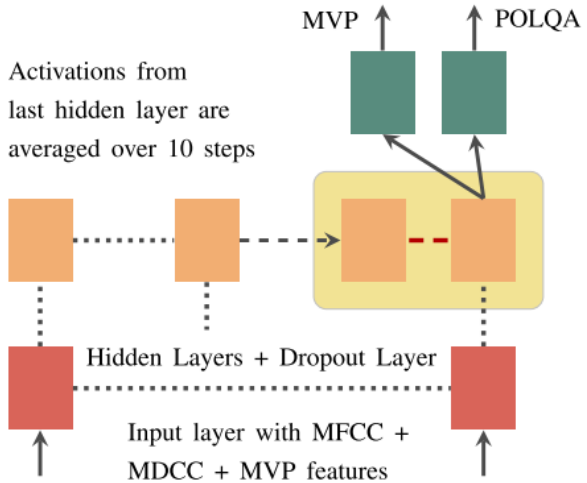
- Log Mel Filter-banks (40/80)
  - Spec Augmentation
    - Different frequency components
- PASE
  - Problem agnostic speech features
  - waveform based with out of box, no re-training
  - “derive useful speech representations by employing a self-supervised encoder-discriminator approach”
  - https://arxiv.org/abs/1904.03416
- MREC features
  - LMFCCs, MDCCs, FFVs



Model Selection 1

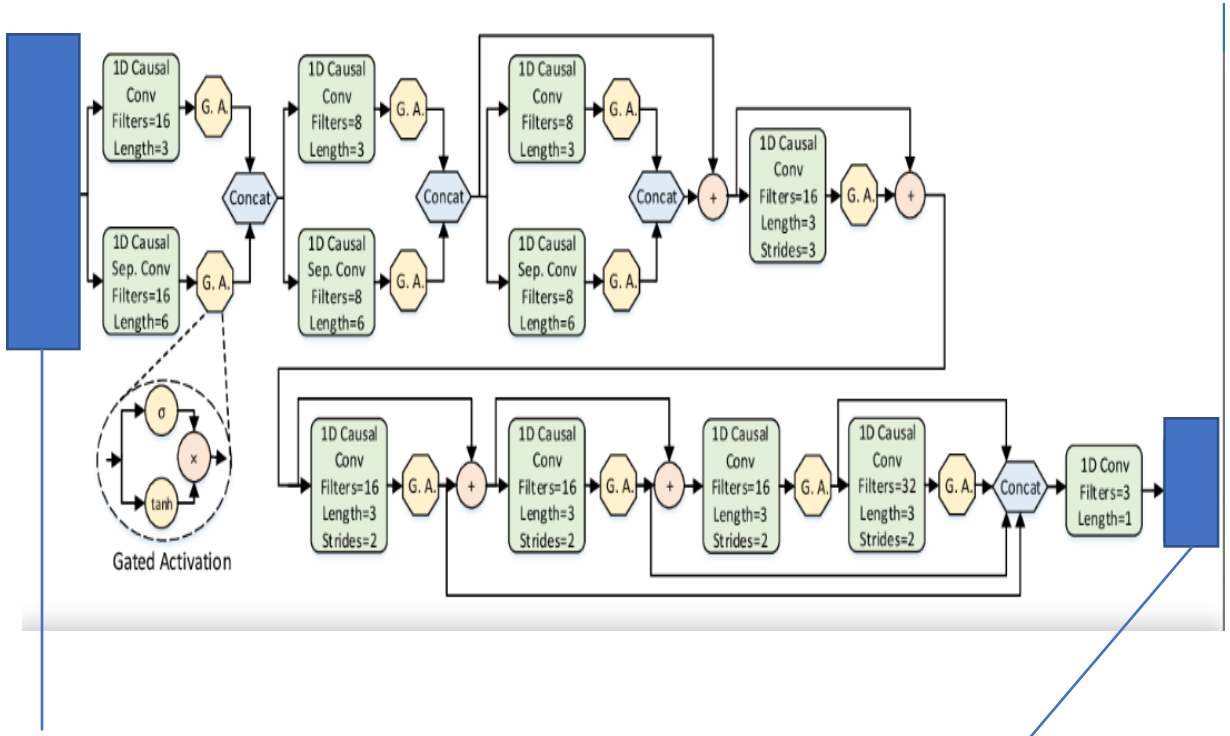
- Robust **frame-wise multi-parameter estimator**
- Experimented several Neural Networks
  - 3-layer Stacked LSTM:
    - n_hidden1=80/100
    - n_hidden2=54
    - n_hidden3=27
    - With output: {averaged over 10 steps}

- **Advantage to this strategy:** Given that LSTMs operate on sequence data, it means that the addition of layers **adds levels of abstraction** of input observations over time. In effect, chunking observations over time or representing the problem at **different time scales.** https://machinelearningmastery.com/stacked-long-short-term-memory-networks/



Model Selection 2

- 1D CNN
  - Model that we use for the VAD binary classification problem
  - Modified
    - Output layer
    - Number of filters
    - Length of kernals
    - Optimization parameters
    - https://arxiv.org/abs/1812.00149



80 log mel filter banks with spec

Linear output with 5 nodes

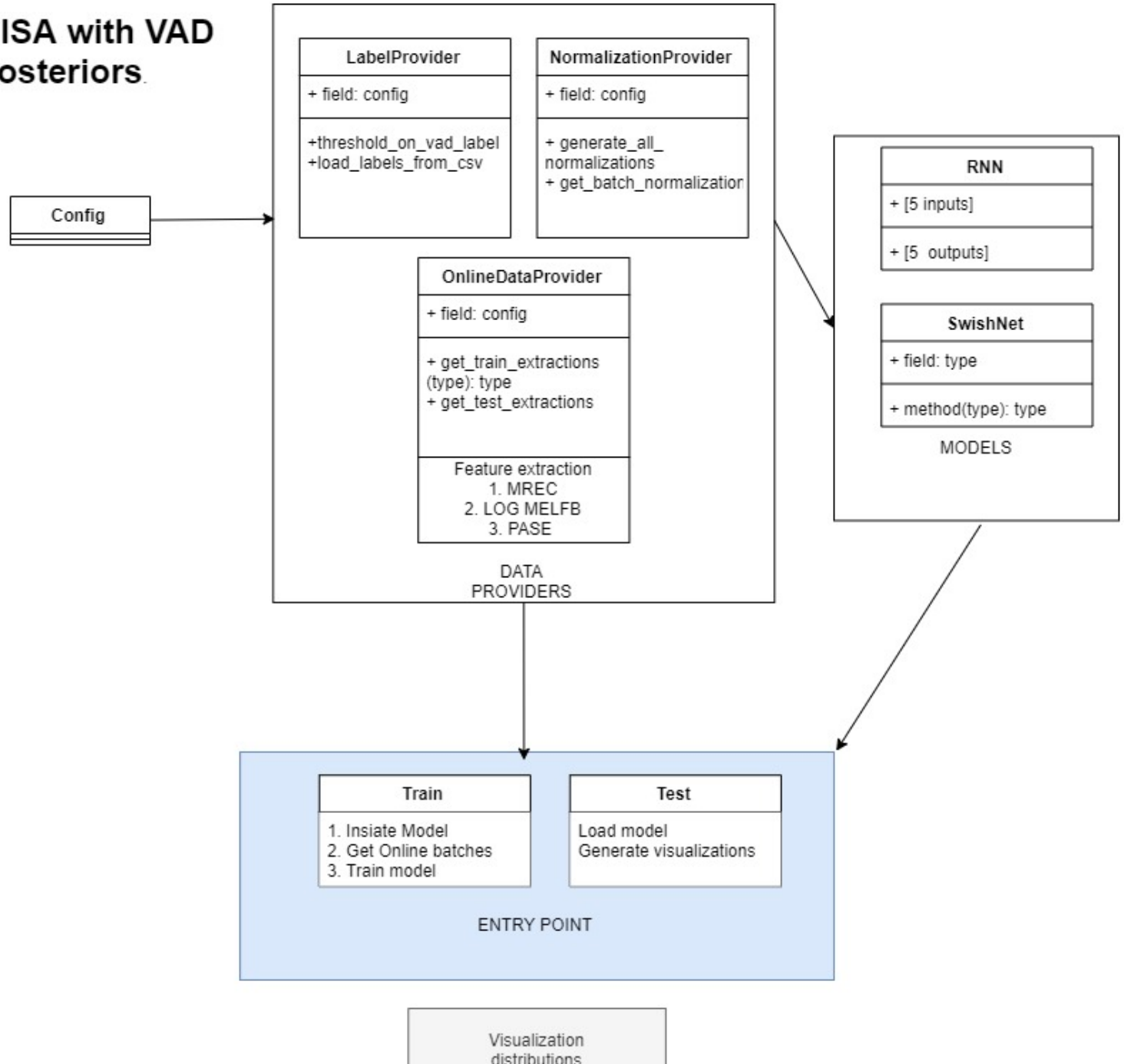
Training and Testing Architecture

- Config controls the Training and testing outside of the entry point
- More flexible
  - Extraction type
  - Model Type {LSTM} or {SWISH}
  - Modularized/Customizable outside of the codebase

```
[INPUT_DATA]
save_dir={directory where the model is saved}
normalization_parameters=normalization file: if this does not exist, that it will be created
logger_file=
train_data=OFFLINE EXTRACTED FEATURE DIRECTORY
train_vad_labels=GROUND_TRUTH_VAD_LABELS/
train_score_labels= csv of train_ground_truth.csv
uses_vad_posteriors=0: FALSE
use_vad_features=0: FALSE
uses_vad_ground_truth=1:
validation_data=not used anymore
extraction_type=mrec * very important
features_input_size=268 *very important: sets the size of the input on the y axis
file_ending=*.txt *feature extraction type{file ending type}
window_size= Size of the chunk
snr_labels= extracted SNR labels
uses_snr_labels= whether the extractor SNR labels are used {Boolean}

[MODEL_PARAMETERS]
model_type=LSTM or SWISH: how the training parses which model to load
train_batch_size= the size of the batches per training
train_num_epochs= epochs per training
number_of_indices= count of chunks that are taken out each time
dropout_keep_prob=1.0
learning_rate= starting learning rate
last_filter_size= {only for the SWISH}
input_nodes=6 = {how many parameters to estimate}
starting_dropout=0.7 {starting dropout}
uses_weighting=1 {whether to use the weighting vectors}
```

NISA with VAD posteriors.



Evaluation

Evaluation Explanation

- **Models:**
- **LSTM:** Stacked 3 layer LSTM that has three hidden layers. Each of these LSTM layers contains multiple memory cells.
- **CNN:** 1D CNN with larger filters than SwishNet Model: *Across temporal domain feature extractions as channels*

- **RMSE: Root Mean Squared Error**
- **MAE: Mean Absolute Error**
- **Correlation: Pearson Correlation between Ground Truth and Predicted**

ASYM Test Results

- Internal Dataset
- Same window size (350 ms)

PARAMETERS	LARGER {FILTERS} CNN	LSTM + LOG MEL FILTERBANKS	BASELINE LSTM +MREC
C50 RMSE	**4.05**	**3.5**	**3.45**
C50 correlation	0.756	0.747	0.84
T60 RMSE	**0.344**	**0.3**	0.35
t60 correlation	0.67	0.63	0.785
DRR RMSE	**2.44**	**2.71**	**5.66**
DRR correlation	0.39	0.369	0.562
Snr RMSE	**3.744**	**3.54**	**4.02**
snr correlation	0.75	0.7	0.9369
PESQ RMSE	**0.28**	**0.28**	**0.26**
PESQ correlation	0.83	0.85	0.855
VAD	**0.932**	**0.929**	
Trainable Parameters:	**~54,000**	**~89,000**	**~124,000**

Narrowband NISA Release ASYM data

- To be used with the NTE/VM2T
  - products
  - Created 8KHz SCT configs from 16K

  setup

  `--sample_rate=8000`

ASYM: Value	RMSE	MAE	Pearson
C50	3.46	2.66	0.79
T60	0.295	0.23	0.75
SNR	3.54	2.4	0.774
DRR	2.65	2.03	0.4
PESQ	0.29	0.23	0.866
VAD	F1score: **0.925**		

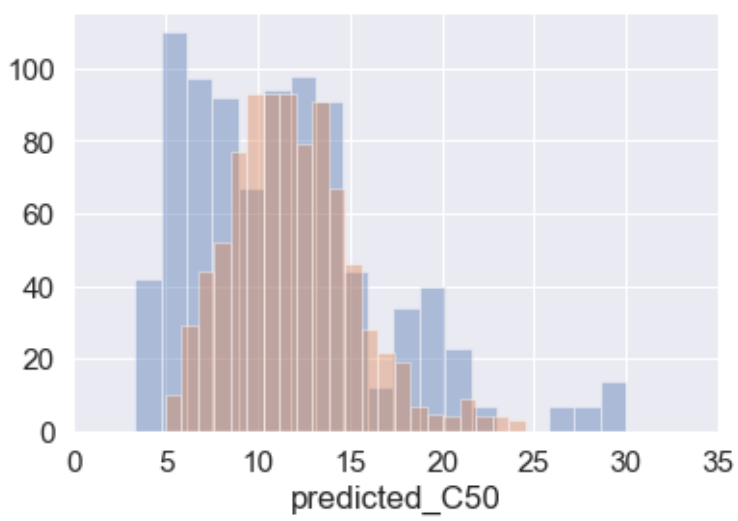
ACE Test Data-Set for the EUSPICO paper

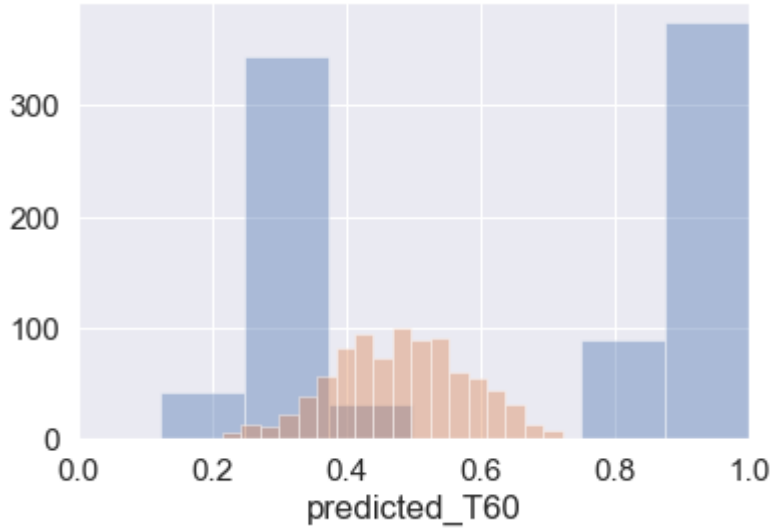
- Published dataset
- Focused on Reverb parameters

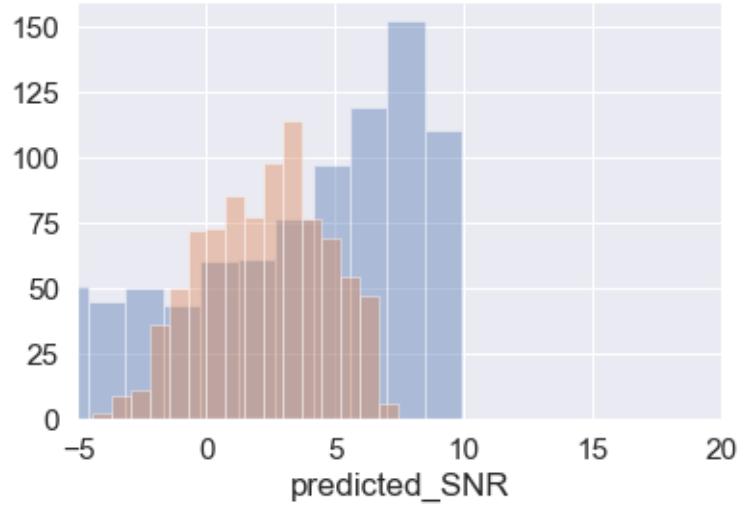
METRICS	MFB+SA+ CNN	MFB+SA+ LSTM	MFB + LSTM	PASE+ LSTM	MREC + LSTM
RMSE C50	3.5	**2.922**	2.99	4.58	3.205
MAE C50	2.9	2.35	2.38	3.71	2.5185
RMSE SNR	3.8	4.3	5.3	3.94	3.75
MAE SNR	3.17	3.5	4.26	3.1	2.97
TRAINABLE PARAMETERS:	16,346	125824	125824	125824	125824

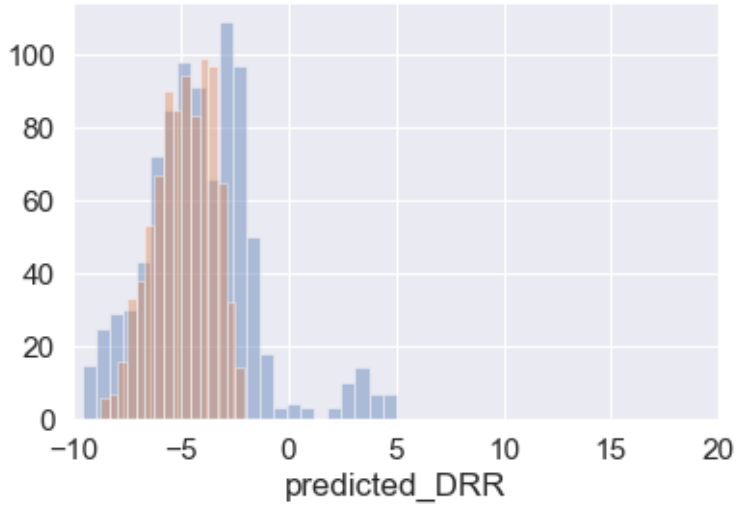
Evaluation Distributions: Missing **outliners?**

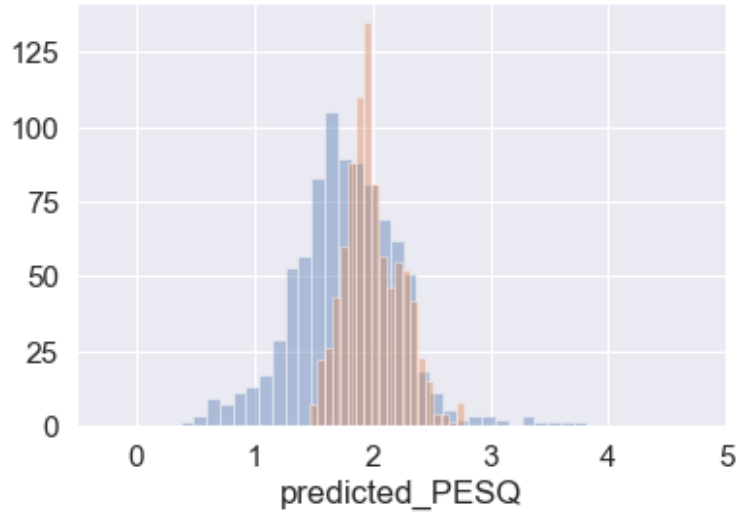
Blue ground truth, Orange predicted, add some weighting strategy to capture full spread!











Real Time Factor Estimate

NISA Parameters Estimate	CNN MELFB + Spec:	LSTM + MELFB + SPEC
10 samples {per utterance}	~0.01 s	~0.076 s

Model Loading	CNN MELFB + Spec:	LSTM + MELFB + SPEC
One Time Cost	~1.31s	4.45

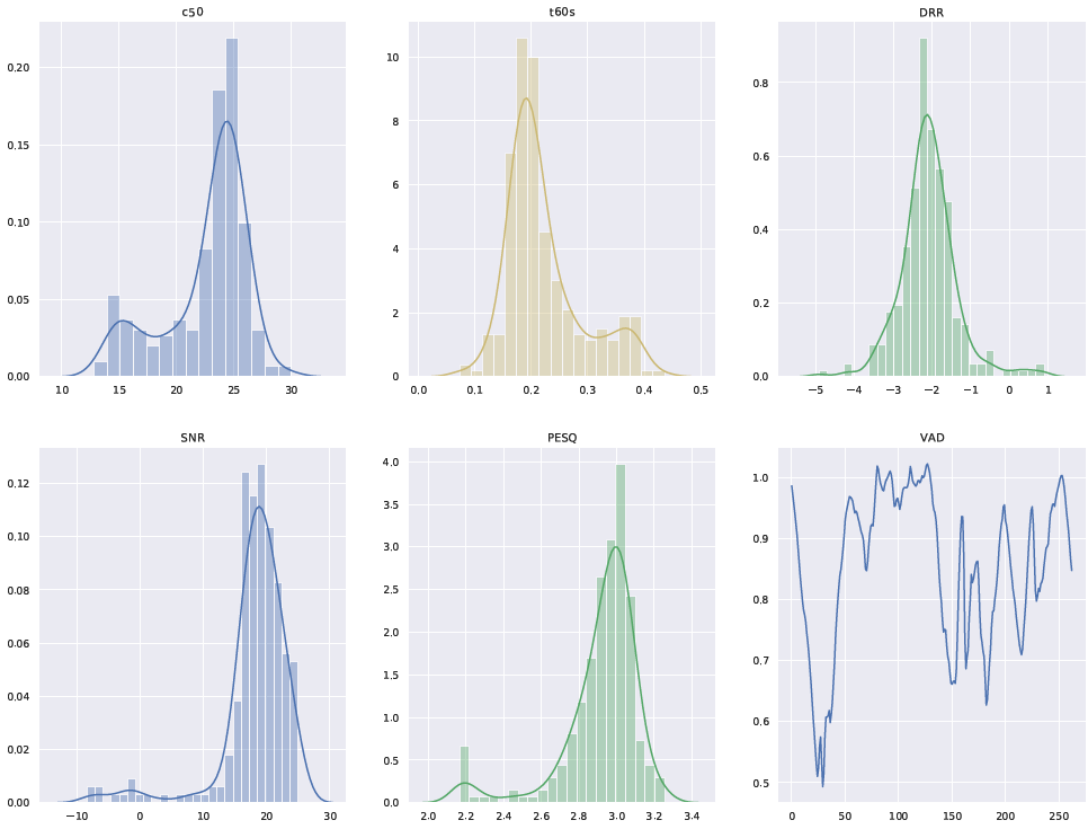
Observations:
Very little change in RMSE with 10 samples or iterating by 1 (increase by x5)

- **Python environment: very slow**
- **Hardware variability:**
- time_estimates.xlsx

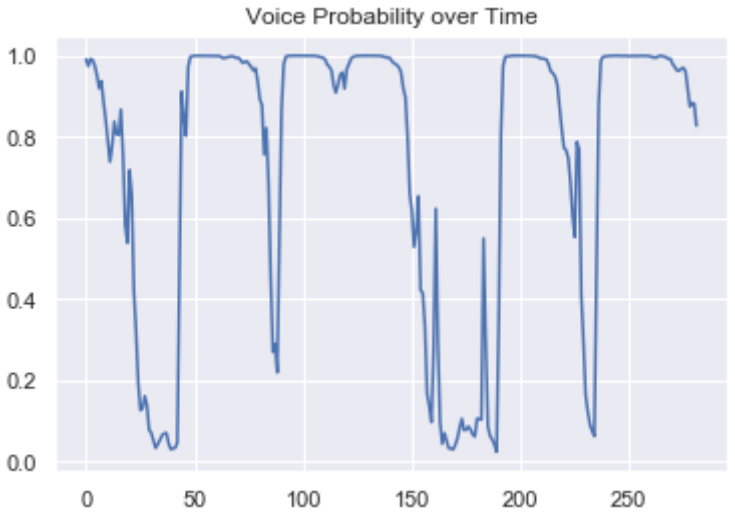
Release of NISA

NISA++ per utterance mode

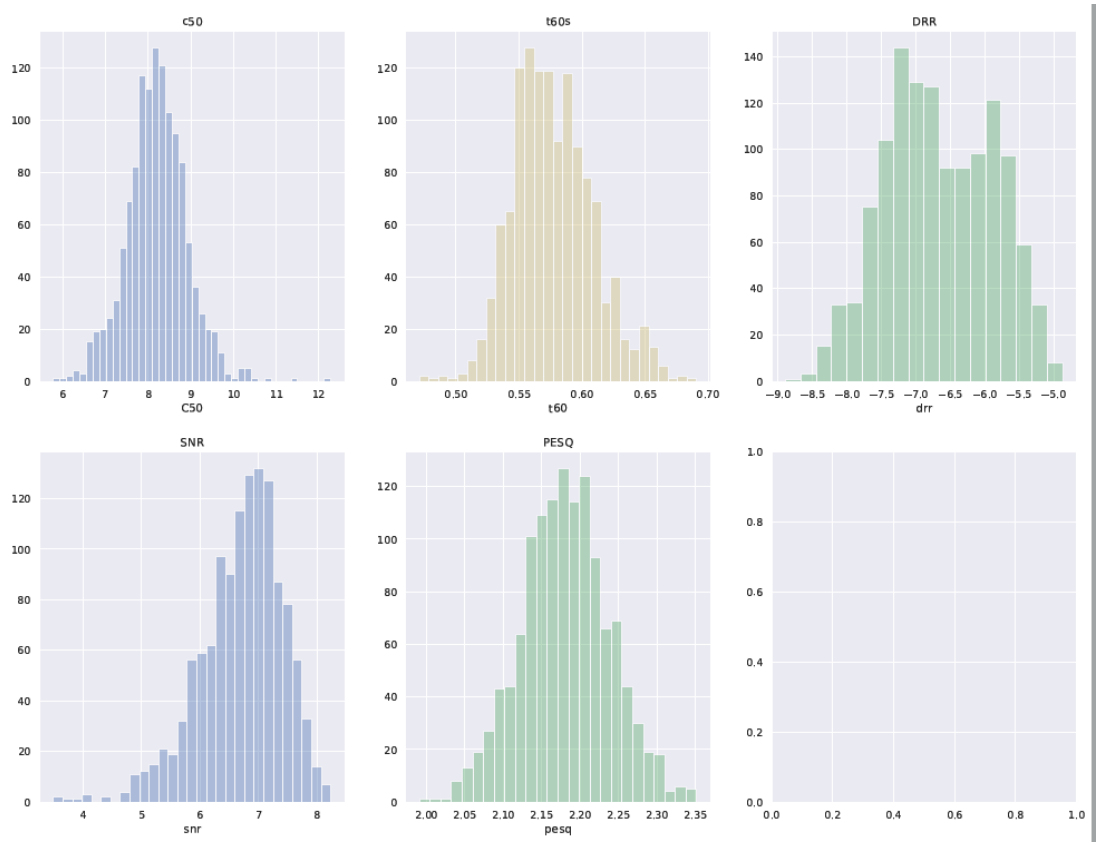
Per utterance mode

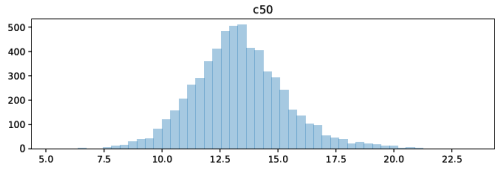


SwishNet VAD estimator



NISA ++ Batch Mode





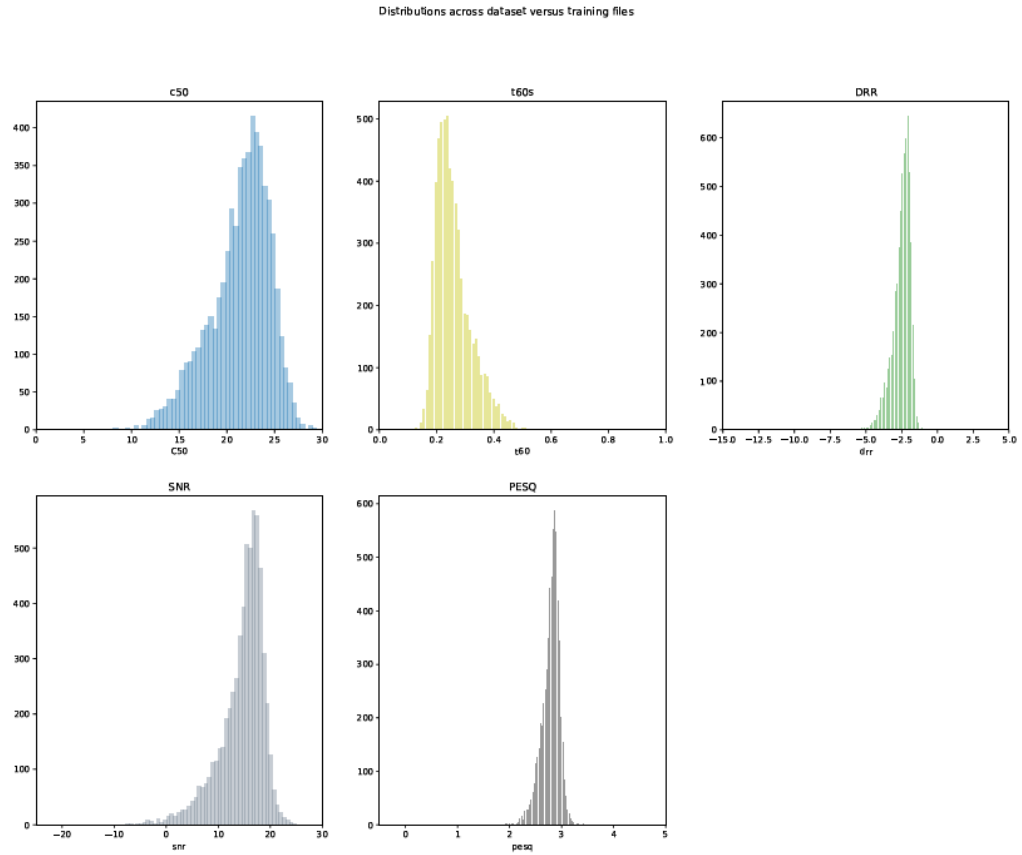
	C50
count	5957.0
mean	13.333
std	2.021
min	5.287
25%	12.052
50%	13.244
75%	14.508
max	23.437

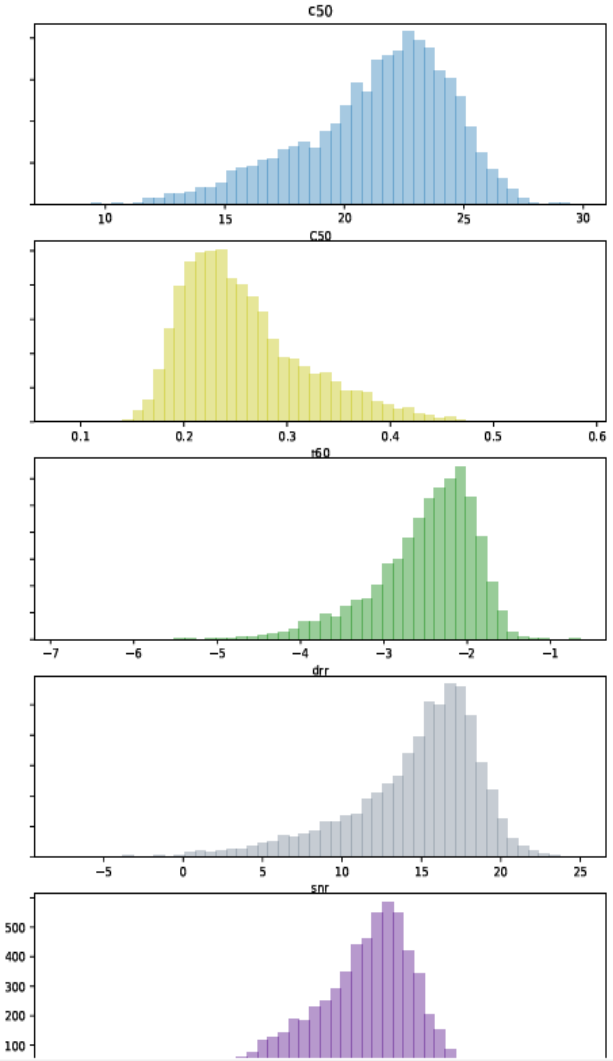
1) High Level Distributions
2) Min/Max, Quartile information per parameter
3) SwishNet VAD posteriors

```
/gpfs/amr_alg/others/lucia/conda/envs/TF_GPU/bin/python run_NISA.py
--is_batch=[0,1] : whether the code is run as a single file or many files
--file_path={}: file or list of files
--output_dir={}: where the output will be written
--is_verbose=1: whether to output pdfs as visualization tools
--vad_threshold=0-1: threshold for the vad posterior
--run_vad_only=0: whether to run only SwishNet


OPTIONAL PARAMS:
--is_verbose_per_utterance=1: whether to output per file pdfs
--sample_rate: desired sample rate {possibilities are 8000,16000}
```

NTE Charter Use-Case





C50	
count	6129.0
mean	21.359
std	3.177
min	8.101
25%	19.587
50%	21.911
75%	23.643
max	29.87

t60	
count	6129.0
mean	0.258
std	0.062
min	0.08
25%	0.214
50%	0.245
75%	0.29
max	0.584

drr	
count	6129.0
mean	-2.513
std	0.636
min	-6.887
25%	-2.829
50%	-2.384
75%	-2.065
max	-0.644

snr	
count	6129.0
mean	14.58
std	4.275
min	-7.762
25%	12.635
50%	15.591
75%	17.442
max	25.0

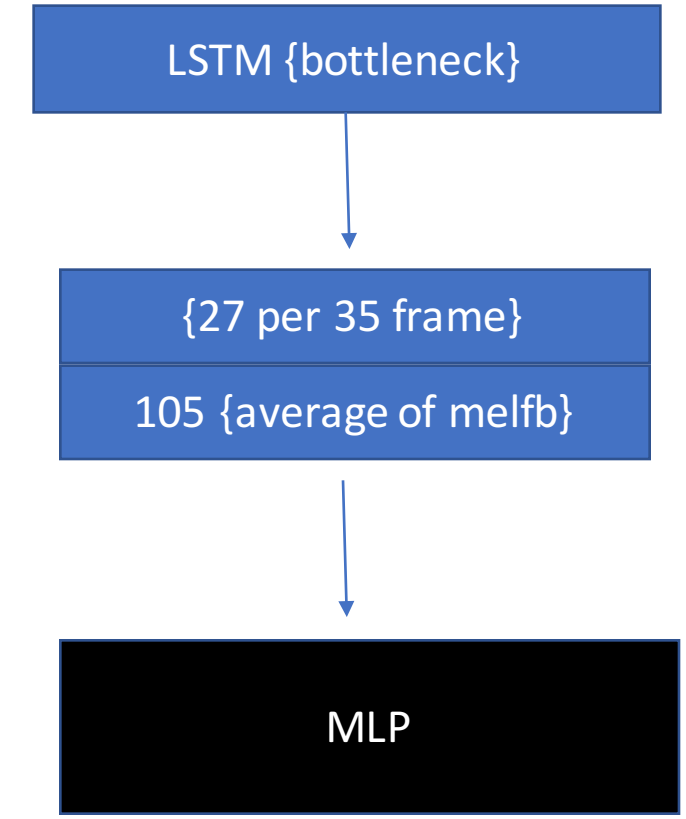
pesq	
count	6129.0
mean	2.79
std	0.169
min	1.912
25%	2.697
50%	2.82
75%	2.903
max	3.432

Noise Type Classifier/CODEC

Ongoing

Noise Type Classifier

- Noise Types:
  - Ambient, Babble, Music, White
- 2 Approaches:
  - Baseline Swishnet (1D CNN)
  - Bottleneck LSTM => MLP



Evaluation

SwishNet Classification:

1 second window size

	Predicted AMBIENT	Predicted BABBLE	Predicted MUSIC	Predicted WHITE
Ground Truth AMBIENT	**0.83**	0.13	0.027	0.001
Ground Truth Babble	0.22	**0.67**	0.13	0.04
Ground Truth MUSIC	0.16	0.10	**0.69**	0.07
GROUND TRUTH WHITE	0.02	0.002	0.01	**0.97**

MLP Bottleneck:

1.05 second window size

	Predicted AMBIENT	Predicted BABBLE	Predicted MUSIC	Predicted WHITE
Ground Truth AMBIENT	**0.46**	0.37	0.1	0.05
Ground Truth Babble	0.16	**0.79**	0.021	0.021
Ground Truth MUSIC	0.44	0.47	0.05	0.03
GROUND TRUTH WHITE	0.202	0.024	0.006	**0.76**

Take-Aways & Next Steps

Hands-on exposure to audio data processing, corruption and modelling

1. *Hand-off to Cheng* ☺
2. *Submitted a Paper to EUSIPCO 2020*
    1. Non-Intrusive Estimation of Speech Signal Parameters using a Frame-based Machine Learning Approach

Questions?

Thank you!

Special thanks to Dushyant & Carl for the mentorship.

Code:

NISA: https://git.labs.nuance.com/dushyant.sharma/NISA

Literature Reviews: https://nuance.jiveon.com/people/lucia.berger/blog

Sources : https://jeddy92.github.io/JEddy92.github.io/ts_seq2seq_conv/