RandomForest-based Voice Activity Detector with Robustness Improvement by Data Augmentation

Chuwei Liu

Submitted in partial fulfillment of the requirements for the
Master of Music in Music Technology
in the Department of Music and Performing Arts Professions
Steinhardt School
New York University

Advisor: Brian McFee

December 2, 2019

Abstract

Voice Activity Detection(VAD) distinguishes the speech segments from the non-speech contents in the audio. Implementing VAD could benefit subsequent complex tasks includes speech transmission, recognition, and enhancement. Previous VAD scheme such as ITU-T G.729B standard relies on heuristic criteria that classification was achieved by considering one or more instantaneous features. These methods are challenged when dealing with complex situations such as high levels of background noise and a variety of sound events. For example, in the urban sound environments, we expect our VAD could not only detect the speech presence but also could distinguish it from other non-speech disturbances such as car horns, siren, and other sound events. In this thesis, a VAD algorithm using Random Forest is proposed for detecting speech occurrence in the urban sound environment.

We start with synthesizing the soundscape data set due to the lack of Off-The-Shelf data set for our speech/non-speech binary classification task. It is achieved by injecting speech data from corpus Mozilla Common Voice into the UrbanSound-SED, which is a 10,000 ten-second soundscapes data set and originally for multi-class, urban sound events classification problems. As a result, the completed data set contains 22.2 hours of audio, in which 16.6 hours of data for training with Speech/Non-Speech ratio of 1:10. For testing the model as a function of SNR, the test set was generated in multiple versions with different event SNR range from 0 to 24, which means all sound events are 0-24 Loudness Units relative to Full Scale(LUFS) above the background Brownian noise that normalized to -30 LUFS.

In order to use the Random Forest classifier with a promising result, we adopt the model selection strategy that consists of metric selection, feature selection, and hyper-parameters selection. Due to the skewed classes in our data set, the unbiased metric balanced accuracy is selected as the primary metric to evaluate model performance. Moreover, we weight the cost of misclassification based on class frequency during the training. For optimizing the hyper-parameters and ensure the model's output is reliable, the Grid Search with Group K-Fold Cross Validation is implemented. The former promises the combination of hyper-parameters with the best performance, and the later guarantee a reliable result by average multiple validation score.

We compare the result of using novel frontend Per-Channel Energy Normalization(PCEN) to baseline Mel frequency Cepstral Coefficients(MFCC) as the learned feature. The results indicate PCEN and MFCC perform closely on balanced accuracy, but PCEN seems more robust in low SNR situations. Also, PCEN slightly outperforms MFCC in other metrics such as Area Under the Receiver Operating Characteristic Curve(AUROC) score and F-

score. As a result, the trained model could obtain a balanced accuracy of 60%-67.5%, 0.85-0.878 AUROC score across SNR range of 0-24.

We further improve the model's robustness by a stage of data augmentation, and this is done by expanding the training set with the help of open-source library Muda. During this stage, Pitch Shift, Time Stretch, Dynamic Range Compression, Additive Colored Noise, and Impulse Response Convolution, totally five deformations are tested in different ways. By making use of Pipeline and Union operator, we compare the result of the model that trained on individual perturbations of train set(Union) and complex perturbations that multiple deformation effects accumulated(Pipeline). The results indicate except IR convolution, all deformations with their custom deformations, are valid for our problem that slightly improve the model performance with 2% increment on balanced accuracy. The Colored Noise deformation outperforms others with 4% increment that results in the balanced accuracy achieve an average 69.2% across all SNR range. In addition, it shows that Colored Noise deformation exposes less to SNR level, which meets our requirement of robustness improvement.

Since there is a strong correlation between consecutive frames for both speech and non-speech contents, we use Hidden Markov Model for modeling the speech occurrences, then perform Viterbi Decoding to find the most likely speech/non-speech states sequence given the observation likelihoods that produced by our model. This scheme delays the transition between states and reduces the misdetection; as a result, our model's predictions are more realistic, and the balanced accuracy could score 83.6% on average across all SNR ranges.

Acknowledgements

I would like to thank the following people:

Dr. Brian McFee, for his support and guidance throughout this work. Justin Salamon, Juan Bello, Tae Hong Park, and other MARL group members, for their research about urban sound and tools that greatly facilitate our experiment, Dr. Schuyler Quackenbush for showing me the world of signal processing, audio coding, and how these problems are done at a low level using the C programming language, and other NYU Music Technology faculty, for their joint effort for offering a memorable academic experience.

And finally, I would like to thank my mom and dad, I couldn't have achieved anything without your support, as well as the rest of family and friends for their constant encouragement.

Contents

List of Figures

1 Introduction

Voice Activity Detection(VAD) is the process of detecting the presence of speech and non-speech segments from the given signal. It's an enabling function that allows devices or applications dynamically change its work status depends on the occurrence of speech. Different VAD algorithms were conducted in various filed: For the audio transmission in Voice over Internet Protocol(VoIP), a standard proposed by ITU Telecommunication Standardization Sector(ITU-T) called G.729 Annex B uses VAD for the silence compression. [Benyassine et al., 1997]; VAD was also used for noise suppression/reduction by estimating and subtracting the noise spectrum from signal. [Woo et al., 2000]; Moreover, it helps with speech enhancement and further contributes to the robustness of speech recognition system, [Ramirez et al., 2007] which is a predominant audio-related subfields of artificial intelligence. [Russell and Norvig, 2016] Especially in the past few years with the trend of AI came back, we could see the AI's subdisciplines are not only developing at a fast pace but also work in concert, and those gadgets rely on the interaction with speech are seemingly everywhere.

One challenge with the use of these gadgets in daily life is they are not robust enough to various environments. For example, waking up your iPhone by saying the magical keywords when you get up at 6 am in your bedroom is different than it was used when walking down the Broadway. Noise complaints are not only harmful to people who live in a metropolitan area like New York City but also interference with the use effect of those gadgets. Several researches and projects that focus on the urban sound environment were conducted by NYU Steinhardt Music Technology department and Center for Urban Science and Progress: For most of the noise complaints in NYC are related to loud music, construction noise and loud talking. [Park et al., 2014] A taxonomy with data set for urban sound classification was proposed due to the lack of common categorization and annotated data. [Salamon et al., 2014] An open-source library Scaper was presented, which facilitates the process of synthesizing soundscape. [Salamon et al., 2017] And the project Sounds of New York City (SONYC) is still in progress that aims to create realistic urban sound dataset by collecting soundscape through the sensors in NYC, and allows volunteers help with identifying and labeling the captured sound events online. [Bello et al., 2018]

Although those previous researches are helpful somehow, there is no project or data address the problem of detecting speech against the urban noise. Hence, in this thesis, a voice activity detector based on one of the state-of-the-art machine learning approaches, the Random Forest algorithm was proposed for this speech/non-speech binary classification. Achieving this goal requires three steps: (1) Due to the lack of Off-the-Shelf dataset for this task, it's necessary to create vast amounts of real-world data of annotated urban sound. (2) Implementing proper feature extraction algorithms to obtain the instantaneous features that could accurately represent speech. (3). Train the Random Forest model properly, make sure the result reliable and maximize its generalization ability as possible. Moreover, we perform a step of data augmentation for improving the model's robustness by expanding the train set with perturbations and evaluating the model with unmodified test data. Within each step, a series of experiments are presented to compare different feature extraction methods, the effect of various deformations for data augmentation, and how a stochastic model helps with making predictions.

The remainder of this thesis arranged as follows: In section.2, the implementations of VAD in different fields are reviewed first and followed by reviewing three realization modes of VAD: heuristic criteria, statistical model, and machine learning. Then we introduces prior works on urban sound classification and soundscape data synthesis that carried out under the direction of Urban Sound Taxonomy [Salamon et al., 2014] [Salamon et al., 2017] In Section.3, the methodology of soundscape generation, feature engineering, Viterbi Decoding, and data augmentation are presented: first, a detailed procedure of extracting Mel-frequency Cepstral Coefficient (MFCC) is performed, which is used as the baseline feature for our system. Then, we review a novel frontend called Per Channel Energy Normalized(PCEN), which is an alternative to log-mel frontend for reducing the dynamic range of mel-filterbank energy [Wang et al., 2017] [Lostanlen et al., 2018]. Following with a subsection focuses on tuning our Random Forest classifier, which is also known as the model selection that includes feature comparison, hyper-parameters optimization, and grid search with group-K-fold cross-validation. Moreover, we introduce the background knowledge of Hidden Markov Model with its solution Viterbi Decoding, and data augmentation for musical audio signals. Section.4 describes the implementation of our experiment. By making use of Mozilla Common Voice and UrbanSound-SED, we form the data set that meets our requirements. In the stage of model selection, we determine to use the balanced accuracy as the proper metric for our imbalanced data set and evaluate our model with different feature selection as a function of SNR. In order to improve the robustness of our model in the end of this section, various deformation methods are implemented as the way of data augmentation: we make use of the open-source library Muda to generate perturbations of the train set for training and tuning the classifier, and evaluate it on the unmodified test set. In Section.5, we conclude the work and address several problems found during the experiment. Moreover, we raise some exciting topics inspired by this work that worthy of study in the future.

2 Prior Work

2.1 Background

In the early days, VAD was mainly used as an enabling technique for speech coding and noise reduction. For encoding the speech during signal transmission, the idea behind is discontinuous transmission(DTX) through Variable Bit Rate(VBR): By detecting the presence of speech, the devices(e.g., encoder, transmitter) are dynamically switched on or off depends on whether the current signal frame is speech or non-speech. The benefits of doing this are various: First, the average power consumption of the device is reduced since VAD avoids the waste by switching the device to a low-power state when no speech detected; this saving seems just a drop in the ocean but means a lot for those devices with limited power capacity such as cell phones. Second, integrating VAD in an audio codec allows VBR applied during encoding, which saves the bandwidth over transmission progress. In terms of noise-related processing, includes various suppression/cancellation techniques for stationary noise, echo, reverberation, and other complex acoustic scenes, the basic idea is estimating noise spectral, then subtract it from the signal which contains both noise and speech. In this case, VAD works on the opposite that focuses on locating the non-speech segments and perform spectral analysis. After learning the acoustic characteristics of the non-speech signal, further spectral subtraction and speech enhancement could be implemented.

In the past few years, the trends of Artificial Intelligence came back with achievements in both theory and hardware. Various types of intelligentized gadgets and products came out, which have a close relationship to the field of Computer Vision(CV), Natural Language Processing(NLP), and Automatic Speech Recognition(ASR). As a critical enabling method, VAD facilitates the use of those technologies considerably. Remarkable techniques such as Keywords Spotting, Speaker Verification, Speech Synthesis and Translation, etc. As you might have noticed, those techniques have gradually become a ubiquitous part of our everyday life nowadays. And implementing those techniques in our daily life requires the device robust enough to face the complex environment in the real world.

For people who live in the metropolitan area such as New York City, noise pollution would be the biggest complaint that not only harmful to our health but also cause disturbing effects on communication. Several researches have been conducted that focus on this topic. In [Ising et al., 2004], authors reviewed past researches and showed how the noise pollution impacts on health effects; the Citygram project with the sonic analysis offered a comprehensive workflow from capturing to analyzing the urban sound environment. [Park et al., 2013] [Park et al., 2014]The varied acoustic environment in an urban area also makes the implementation of VAD somewhat challenging, because instead of facing stationary background noise, the disturbance from various acoustic events is more complicated to analyze. As the result, for those VAD approaches that rely on adapting threshold for measured features on the temporal and spectral domain, they are no longer reliable even troublesome because of misclassification. Therefore, the desired VAD should not only detect the presence of speech but also classify the speech from other acoustic events.

Although deep learning with its superset machine learning is not new fields as the subbranch of AI, they are definitely growing at a fast pace in recent years. It allows the computer to learn from experience, named training data, and obtain the generalization ability to make the prediction on new input. And the learning process is called supervised training if user feed the model with tagged data. Moreover, it could be further grouped into classification and regression, while the former deals with the categorization task and the latter focus on predicting specific value. Implementing different machine learning algorithms for the VAD task also started a long time ago: as early as the 80s, Decision Trees and Bayesian were used for voiced-unvoiced-silence (V-U-S) classification in speech analysis. [Dattatreya and Sarma, 1981] [Siegel and Bessey, 1980]. The K-nearest-neighbor(KNN) was tested for speech and non-speech discrimination [Lu et al., 2002a]; and lots of works utilize Support Vector Machine(SVM) as well. [Enqing et al., 2002] [Ramírez et al., 2006] [Kinnunen et al., 2007]

In this thesis, the presented voice activity detector is based on Random Forest algorithms, one of the state-of-the-art machine learning scheme proposed by Leo Breiman in 2001. [Breiman, 2001]. Although most of the current research deal with VAD task by using neural network, the reason of implementing Random Forest rather than other machine learning and deep learning schemes are as follows: (1) It's commonly accepted and proved that under the premise of input data with good quality, deep learning could only outperform if the amount of data increased to a certain extent. [Moolayil, 2019] A typical machine learning issue is the lack of quality data. Hence, we have to create synthetic data with precise annotations artificially to meet our needs and conditions that unlikely to achieve any time soon. With a limited amount of data and computing ability, it's better to stay with machine learning rather than deep learning. (2) Comparing to other algorithms, Random Forest is less likely to over-fitting because it is a type of ensemble learning method that uses multiple decision trees to make a prediction. On the other side, compared to neural network work as a black box, Random Forest has better interpretability when we want to get an insight into the importance of features. (3) Feature engineering is a fundamental process for machine learning. For audio-related processing, it means extracting the proper instantaneous features as the temporal/spectral representation that interests us. And the process of extracting features could be one of the most critical abilities that should be mastered for being a student in the Music Technology field. Another reason for implementing the feature extraction in detail is because we want to test the novelty feature, PCEN frontend in the VAD task, for which we have to tune and compare the parameters that involved. (4). The last cause of using Random Forest is it's simple to handle the data that not necessary to do any normalization or scaling on it.

The rest of this section will review the prior works and literature that relevant to this thesis. It will starts from basic voice activity detecting approaches based on heuristic criteria, then the statistical modeling approaches and machine learning-based methods. Next, the development of Random Forest will be introduced. As part of this thesis, we will also look back on that produce profound influences on this work.

2.2 VAD

2.2.1 Heuristic Criteria

Heuristic evaluation for VAD is straightforward and easy to implement. Its mechanism assumes some temporal or spectral features could represent speech so that by applying the threshold on one or more heuristic, it should be efficient to identify the presence of speech. One example of this approach is energy thresholding that measuring the signal energy either every single frame or the average of a period and identify the speech frame if it exceeds certain thresholds. Several heuristic criteria are explained in this subsection, even not adopted in this thesis.

Back to 70s, Rabiner and Sambur proposed an algorithm of using short time energy as criteria for locating the endpoint of utterances. [Rabiner and Sambur, 1975] In which the peak energy and silence energy were measured within a 10-ms window under 10kHz sampling rate. Then compute upper and lower threshold according to the given equation so that the start point can be determined as the energy exceeds the lower for the first time, and the endpoint would be the moment that energy falls below the lower before the next rise. Instead of using magnitude directly, the logarithmic energy was used respect to the non-linearities of human perception. [Atal, 1976] There are many variants based on energy proposed in subsequent researches such as the 4Hz modulation energy and low-band energy ratio, which measure the sub-band energy after bandpass filtering. [Scheirer, 1997] Others compute the percentage of frames that lower than half of mean RMS power in a 1-second window, which could effectively separate speech from music. Similarly, an alteration is the energy in 4-sub-bands [Liu et al., 1997] and so on.

Another important feature used in both previous research is Zero Crossing; a temporal criterion indicates high-frequency contents(noisiness). Because the voiced speech is produced when vocal cord vibrating with the periodic pulses of air passing through, and unvoiced speech is random-like sound without cord vibration. Zero Crossing Rate(ZCR) is not only enabled to differentiate speech from noise(silence) but also could indicate the difference between voiced and unvoiced speech. As a result, ZCR is more likely to be used in speech recognition that needs further phonetic analysis. What improved further based on ZCR are linear prediction zero-crossing ratio(LP-ZCR) [El-Maleh et al., 2000] which is the ratio of the count of zero crossings in the input signal to the count of the version processed with the LP analysis filter, namely the LP residual. Another remarkable one called High Zero Crossing Rate Ratio(HZCRR) which proved effective for separating speech and music. [Lu et al., 2002b]

One standardized VAD scheme called ITU-T Recommendation G.729 Annex B was integrated into the original G.729 codec as part of extension. [Benyassine et al., 1997] It uses a set of parameters including spectral distortion, energy difference, low-band energy difference, and zero-crossing difference, to compress the detected silence for DTX. This standard is remarkable that frequently be used as the baseline scheme in relevant researches. Other novel features were proposed in successions such as spectral entropy [Renevey and Drygajlo, 2001] and long-term spectral divergence [Ramırez et al., 2004]; those were proved with promising results even better than G.729B standard.

As we just mentioned, the implementation of LP-ZCR relies on the Linear Prediction analysis, which initially came from a series of speech recognition researches: Itakura and Saiko proposed a maximum likelihood approach for automatic phoneme discrimination. [Itakura, 1968] And later the Linear Prediction Coding was proposed in [Atal and Hanauer, 1971] and tested for speech modeling [Itakura, 1975b] [Itakura, 1975a] and VAD task [Rabiner and Sambur, 1975] [Atal, 1976]. The idea behind is by applying a time-varying filter on the excitation of quasiperiodic pulses train, or a white-noise source depends on whether the modeled word is voiced or unvoiced. Given the n-th order LPC filter, the spectral envelope could be wrapped and represented by a few coefficients, and its residual signal is sufficient for further analysis and transmission. In Rabiner's works, only the first LPC coefficient was used to indicate the cepstrum difference between voiced, unvoiced, and silence signal. In 1990, the Perceptual Linear Predictive(PLP) was proposed as a comparable feature; it takes account of properties of human hearing such as critical-band resolution, the equal-loudness curve, and intensity-loudness power law, proved more consistent with speech analysis. [Hermansky, 1990]

The Mel-Frequency Cepstrum Coefficients(MFCCs), which is the baseline feature in this thesis, would be the dominant feature for speech-related research nowadays. It takes the property of human ear perception into account when separating the spectral envelope: using multiple overlapping triangular windows non-uniformly spaced on the power spectrum according to the mel-scale to construct a filter bank that simulates the non-linearity of human ears. [Stevens et al., 1937] Section.3 presents the detailed procedure of extracting MFCCs. It was proved that MFCCs outperform LPC in speech estimation. [Dave, 2013] However, the MFCC has the deficiencies that not robust to noise. Low signal-to-noise ratio (SNR) or a relatively flat spectrum will degrade the performance significantly. As the result, several improvements were presented such as the Gammatone Cepstral Coefficients(GTCCs), which implemented the gammatone filter bank with equivalent rectangular bandwidth bands and proved more effective in representing the spectral characteristics of non-speech signals. [Valero and Alias, 2012]

A critical step during extracting MFCCs is taking the logarithmic of the mel-filtered power for dynamic range compression. In 2016, Power-Normalized Cepstral Coefficients(PNCCs) was presented in which the log compression was replaced by a power-law nonlinearity compression, which reduces the variances in response to the low-level signal. Plus with asymmetric filtering for suppressing the noise and a temporal masking module in consideration of the onset-sensing property of the auditory system; and this novelty feature was proved helpful for speech recognition in reverberant environments. [Kim and Stern, 2016]Later, the Per Channel Energy Normalization(PCEN) frontend was proposed, which uses an Automatic Gain Control(AGC)-based dynamic range compression to replace the static log compression. The trained model outperformed the log-mel for keyword spotting test in noisy and far-field situations [Wang et al., 2017], and proved computationally efficient even in real-time. [Bello et al., 2018]As the result, the PCEN will be tested in this thesis that we will compare the model performance between PCEN and MFCC as the learned feature.

2.2.2 Statistical Modeling

In 1984, a short-time spectral amplitude(STSA) estimator by utilizing the minimum mean-square error(MMSE) proposed by Ephraim and Malah. This algorithm was proved comparable to Wiener Filtering that initially used for speech enhancement. Furthermore, they introduced one influential method that employing the gaussian random variables to model the speech and noise spectral components. [Ephraim and Malah, 1984] This approach was used in [Sohn et al., 1999] for modeling the Discrete Fourier Transform (DFT) coefficients of noise, speech, and noisy speech respectively, and compute the likelihood ratio for each frequency band by the probability density functions conditioned on speech absent or present. To get the final decision, the geometric mean of the likelihood ratio was computed and the decisions were smoothed by using a hidden Markov model(HMM) to model the speech occurrence. Based on this work, the complex Laplacian and Gamma probability density functions were presented as the alternative to Gaussian Random Variables in the following research with promising results. [Chang et al., 2006] With the development of machine learning, later, the decision rule of the likelihood ratio test mentioned above was replaced by the support vector machine (SVM), which treats the likelihood ratios as a feature vector and obtain the optimal hyperplane for distinguishing the sample which belongs to one kind or another. [Jo et al., 2009]

As early as the 80s, some machine learning algorithms were already applied to voiced-unvoiced-silence classification. In [Dattatreya and Sarma, 1981], Dattatreya and Sarma proposed an algorithm formed from the Bayesian Minimum Cost scheme and Binary Decision Tree; the result showed this approach is computationally complex at that time. Today, with the support from computing ability growing and optimization of algorithms, we could see how these algorithms are coming back in practice. As one of the supervised learning algorithms, the decision tree, which is a branching method that exploring outcomes by continuously take the criterion into consideration. It iteratively splits the problem into smaller subsets for maximizing the information gain(or minimize the entropy) to select a splitting attribute when partitioning a training set. Commonly used decision tree algorithms are Iterative Dichotomiser 3(ID3), C4.5, and Classification and Regression Trees(CART), and each has its improvement than the previous. As the representative work, CART is a binary decision tree algorithm that firstly proposed by L.Breiman in 1984. [Breiman et al., 1984] It can work with either the discrete or continuous data, namely the classification or regression, selecting the features based on the Gini coefficient that maximizing the purity for each child node. And the coefficient, also called Gini impurity is given by

$$Gini_{(T)} = 1 - \sum_{i=1}^{k} p_i^2 \tag{1}$$

where the T represents a training data, i is the index of classes that begins with first class up to k-th class; and p is the probability of T belonging to class i.

For combating the over-fitting, the Pruning in CART is accomplished by cross-validation that checks the impact of removing each split to the tree, then prune those splits considered as 'not helpful.' But still, a single decision tree will easily fit the training data perfectly. In contrast to that, Random Forest adopts the concept of ensemble learning that uses several trees, namely weak learners, and group them to predict by balancing the results of each tree. Moreover, to decrease the variance, Random Forest employs the Bootstrap aggregating(Bagging), which is a meta-algorithm that randomly create subsets from training samples and use each subset to fit different trees in parallel [Breiman, 1996] This state-of-the-art method has the advantage of more accurate, stable, and less susceptible to the 'greedy search.' In this thesis, we use the Random Forest Classifier offered by the off-the-shelf library scikit-learn [Pedregosa et al., 2011], which is free open-source package of standard implementation for almost all the classic machine learning algorithms.

The Random Forest in scikit-learn is based on the perturb-and-combine(PC) technique for making up the unstable performance that due to the small perturbations in the training set. In PC, perturbing is the phase that creating multiple models by manipulating the distributions of data and combine means simply voting (classification) or averaging (regression) the results from the perturbing stage for a single prediction. [Breiman et al., 1998] As mentioned above, Bagging is the subsampling method in the perturbing phase when constructing the Random Forest model: it creates various sub-samples first, and each sub-sample remains the same size as the original dataset but with the different number of occurrences for each case.

Unlike decision tree, for each node in the Random Fores, the best split is determined by computing the Gini impurity on the current dataset as the given formula above. If its value less than the threshold or too few samples could be manipulated, stop splitting node and return it; otherwise, compute the impurity value for each feature, find the lowest (the best) one for splitting the data into two subsets. What's different from the original algorithms proposed by L.Breiman is in the combine phase, the single prediction is made by averaging probabilistic predictions of each tree. With the help of Bagging and randomizing the number of features to consider when splitting, the relevance between each tree is significantly reduced, and the forest estimator obtains lower variance at overall levels.

2.3 Soundscape Generation

Due to the lack of existing data set that addressing the problem of detecting speech in the urban environment, it's necessary to generate the synthetic data that could reflect the situation in the real world. Fortunately, recent research about urban sound classification and soundscape generation make this laborious task easier to implement. In this thesis, we follow the principle of Urban Sound Taxonomy [Salamon et al., 2014] to generate our soundscape data set. The data set UrbanSound8K was presented under the guideline of Urban Sound Taxonomy [1]. It contains 10 low-level classes sound that collected from Freesound[2]. In which each audio snippets were trimmed in 4 seconds duration, arranged with balanced distribution and ready to be used for synthesizing soundscapes. Furthermore, Scaper, an open-source library for generating soundscapes was presented [Salamon et al., 2017]. The idea behind is decomposing the soundscape into the foreground and background sound, and by combining them under probabilistically control for increasing the variability. With the support of this powerful tool and well-annotated sound bank, they presented UrbanSound-SED available online.[3] Which contains 10,000 soundscape clips with 10 seconds duration, each of clips consists of foreground sound events that sampled from 10 low-level classes and mixed with Brownian noise as background sound. Through the well defined probabilistic specification in Scaper, rational distribution of sound events with variability was achieved and could be used to represent the urban soundscape in the real world. In the end, the data set was tested with promising results, in which the MFCC was used as the learned feature in a Convolutional Recurrent Neural Network (CRNN), and a Convolutional Neural Network (CNN), and crowdsourcing experiments.

In UrbanSound 8K, the 10 low-level classes are air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren and street music; For creating the dataset for this thesis, a slightly change based on this taxonomy is replacing children playing class with speech, and the corpus of speech is sampled from open-source project Mozilla Common Voice,[4] which contains tons of speech clips donated by web users.

[1] https://urbansounddataset.weebly.com/urbansound8k.html

[2] https://freesound.org/

[3] https://github.com/justinsalamon/scaper$_waspaa$2017

[4] https://voice.mozilla.org/en

3 Methodology

Before introducing the methodology and the designed realization process, let's summarize the problems that remain to be solved: (1) lacking off-the-shelf datasets for this classification problem of detecting speech in the urban environment. (2)We've decided to use Random Forest algorithms as the classifier for this problem; for improving the robustness of the model, further analysis is worthy, which includes ①If different feature selection with various parameters can help improve the model performance. ②How to find the combination of hyperparameters that promises the best score. ③Whether the data augmentation is valid for our problem, if so, what type of deformations help with improving the robustness the most. (3)How do we accurately assess the generalization ability; in other words, what kind of metrics could give us a convincing result and truly reflect the model performance?

In this methodology section, we introduce the proposed system that includes several stages: first, we start with a subsection of synthetic data generation for addressing problems (1). Followed by introducing the feature engineering to address ①, in which the detailed procedure of extracting MFCC and PCEN is presented. Finally, to address ② and ③, we introduce the Group-K-Fold Cross-Validation based on grid search, to find the optimal model parameters value and avoid the arbitrary and capricious behavior.

3.1 Synthesizing Datasets

The model performance is highly dependent on the quality of training data, which should accurately reflect the informatics in the real world. Therefore, four requirements should be kept in mind before synthesizing the dataset for VAD in the urban environment:(1)All the sound clips should come from real field-recording. (2) It should cover sound events that usually occur in the urban environment. (Variability of non-speech sources) (3)It should consider the difference between the speakers.(Variability of speech source) (4) It should accurately reflect the distribution of sound events in urban environment. The Taxonomy for Urban Sound with its example dataset UrbanSound 8K solves several problems mentioned above, and the dataset has been proved in multi-class classification problem. [Salamon et al., 2014] We generate the synthetic soundscape data under the guideline of Urban Sound Taxonomy, regarding the workflow of UrbanSound-SED generation. With the help of Scaper, the generated soundscape data set satisfied most of the requirements above. In the original taxonomy, authors categorized the urban acoustic environment into four top-level groups with multiple lower-level sound events as the leaves on each branch, the resulting taxonomy as below:

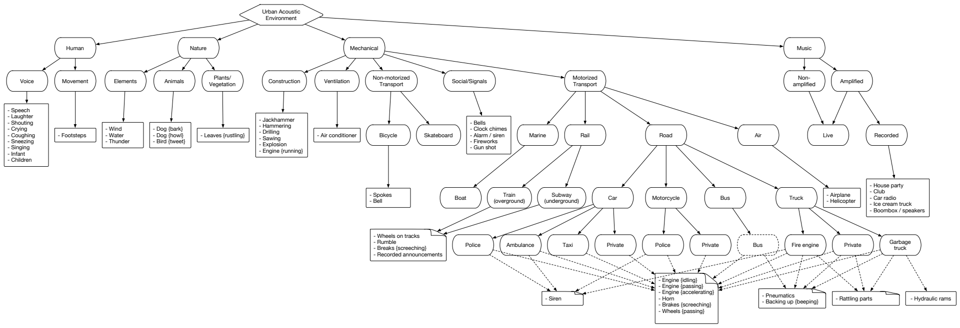


Figure 1: Urban Sound Taxonomy. [Salamon et al., 2014]

Based on this taxonomy, we further change the UrbanSound-SED data set for binary classification. This is done by replacing the original speech class the 'children playing' with new human voice samples collected from Mozilla Common Voice, which is a crowd-sourced speech corpus contributed via the web contribution. This project allows everyone to donates the speech by recording the clips of reading the given sentence and let other users validate its quality by voting. This mechanism of contributions makes the collection meet our requirements in terms of both quantity and quality. As the result, the processed speech data, with other pre-sorted clips that offered in UrbanSound8K, are sampled as foreground sound events and mixed with background noise under given specification to form the synthetic soundscape. The process of soundscape generation remains the same as

UrbanSound-SED [Salamon et al., 2017]

To compare the model performance at different SNR levels, the SNR of foreground events in the train set is sampled uniformly between 0-30 dB to promise the variability that the model could learn while the test set was synthesized in 5 different versions with different SNR in the range 0-6, 6-12, 12-18, 18-24, 24-30. Moreover, to test the novel feature PCEN frontend, all the train/test sets are separately generated with different feature extraction algorithms, which results in two systems trained and tested in parallel.

As the results, for each system, 6000 soundscape samples with uniformed SNR are generated for training and 1888 samples for each SNR range, 9440 in total for evaluating our model as a function of SNR.

3.2 Feature Engineering

3.2.1 Mel-frequency cepstral coefficients (MFCCs)

The baseline feature MFCC is computed through the following steps: first, the short-time Fourier Transform (STFT) is performed over frames, in which each frame processed with discrete Fourier Transform(DFT) as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \cdot n \cdot k/N} \quad \text{for } k = 0, 1, ..., N-1 \tag{2}$$

where the $k = 1, 2, 3, ..., N$ is the coefficient given the N-point DFT.
Among a variety of MFCC implementations, we use the MFCC FB-40 for constructing the mel-scale filter bank. [Slaney, 1998] The MFCC FB-40 from the Auditory Toolbox contains 40 equal area triangular filters that below 1000Hz, first 13 filters are spaced linearly with 66.67Hz interval; the other 27 filters spaced logarithmically above 1KHz with the log step defined as:

$$LogStep = \exp\big(\frac{\ln(\frac{f_{c40}}{1000})}{N_{LogFilter}}\big) \tag{3}$$

Where the f_{c40} is the center frequency of the last filter, $N_{LogFilter}$ is the number of logarithmically spaced filters. And each filter within the filterbank is defined as:

$$H_i(k) = \begin{cases} 0 & \text{for} \quad k < f_{b_{i-1}} \\ \frac{2(k - f_{b_{i-1}})}{(f_{b_i} - f_{b_{i-1}})(f_{b_{i+1}} - f_{b_{i-1}})} & \text{for} \quad f_{b_{i-1}} \leq k_{b_i} \\ \frac{2(f_{b_{i+1}} - k)}{(f_{b_{i+1}} - f_{b_i})(f_{b_{i+1}} - f_{b_{i-1}})} & \text{for} \quad f_{b_i} \leq k_{b_{i+1}} \\ 0 & \text{for} \quad k > f_{b_{i+1}} \end{cases} \tag{4}$$

In which the $i = 1, 2, 3, ..., M$ is the index of each triangular filter, f_{b_i} indicates $M + 2$ boundary points.

Next, compute the log-energy of magnitude spectrum within each passband:

$$X_i = \log_{10} \big(\sum_{k=0}^{N-1} |X(k)| H_i(k) \big), \quad \text{for } i = 1, 2, ..., M \tag{5}$$

and get the cepstral coefficients through the Discrete Cosine Transform (DCT):

$$C_j = \sum_{i=1}^{M} X_i \cdot \cos \big(j \cdot (i - 1/2) \cdot \frac{\pi}{M} \big), \quad \text{with } j = 1, 2, ..., J \tag{6}$$

where the $|_{X(k)}|$ is the magnitude spectrum, $H_i(k)$ is the filter bank, and j is the index of cepstral coefficients. As the result, the MFCC FB40 are extracted as our baseline feature.

3.2.2 PCEN Scaling on Mel Spectra

As the comparison to log amplitude scaling, the Per Channel Energy Normalization frontend will be tested. Previous researches showed it outperformed the MFCC as an acoustic frontend in the far-field and reverberant environment. [Wang et al., 2017] With the advantage of noise suppression and emphasize foreground sound, PCEN seems suitable for detecting voice in the urban environment. It consists of three stages: dynamic range compression(DRC), adaptive gain control(AGC), and temporal integration. Here we represent the mel-frequency spectrogram as:

$$S_{(t,f)} = \sum_{k=0}^{N-1} |X(k)| H_i(k) \quad \text{for } i = 1, 2, ..., M \tag{7}$$

where t and f denote the time step and mel frequency. The first stage is time integration, which essentially smooths the mel-frequency spectrogram by applying a first-order infinite impulse response (IIR) filter as:

$$M_{t,f} = (S * \phi_T)(t, f) = (1 - b)M(t - \tau, f) + bS(t, f) \tag{8}$$

in which ϕ_T, τ, b denote the first-order IIR filter, discretization time step in seconds(default value = 1 in original proposed algorithm [Wang et al., 2017]), and the smoothing coefficient b defined as:

$$b = \frac{\sqrt{1 + 4T^2} - 1}{2T^2} \tag{9}$$

and this weight has the range $0 < b < 1$ that mainly determined by hop size. The given time constant computed as:

$$T = \frac{TimeConstant \cdot SamplingRate}{HopSize} \tag{10}$$

The Time Constant is a transient threshold on both temporal and spectral domains. For example, usually, the background noise is relatively stable within a short time period, while foreground events will modulate faster on amplitude or shift more frequently between different mel-frequency sub-bands. As the result, by adjusting this threshold,the gain level of smoothed $M_{t,f}$ would be controlled dynamically for suppressing the background sound as:

$$G_{(t,f)} = \frac{S(t, f)}{(M(t, f) + \varepsilon)^{\alpha}} \tag{11}$$

This stage is called Adaptive Gain Control, in which the ε is for numerical stability ,α is the gain factor for normalization. Following the Adaptive Gain Control, the last stage, namely Dynamic Range Compression, is performed as:

$$PCEN_{(t,f)} = (G_{t,f} + \delta)^r - \delta^r \tag{12}$$

where the δ is the positive bias introduces the point of nonlinear compression, and r is this compression exponent.

3.3 Model Selection

3.3.1 Metric Selection

There are many metrics for evaluating machine learning algorithms, depends on the type of task(classification or regression) and the distribution of variable that interest us. Choosing proper methods for evaluation makes the result reliable and promises good generalization ability for new input.

As mentioned, the way we generate soundscapes data set follows the workflow of UrbanSound-SED generation [Salamon et al., 2017], which results the dataset for multiclass classification in original research but imbalanced for our task. In the data set we generated, the positive label only takes about 10% of total data that could be considered as the minority class. For traditional machine learning algorithms, the model would be biased towards the majority class [Guo and Viktor, 2004] which results the accuracy paradox; For Random Fores, this leads to few or even none of the speech be sampled during the bootstrapping, which makes the model poorly perform on the minority. Considering this problem, in this thesis we use balanced accuracy as the primary metric, which is simply the arithmetic mean of sensitivity and specificity [Brodersen et al., 2010]. The balanced accuracy is given by:

$$BalancedAccuracy = \frac{(\frac{TP}{P} + \frac{TN}{N})}{2} \tag{13}$$

where TP denotes true positive, TN is true negative. As we can see,it considers proportional correctness of each class individually.In other words, it computes the sensitive and specificity for each class respectively then average over total numbers of class. [Urbanowicz and Moore, 2015] [Mosley, 2013]

For presenting a solid evaluation, other unbiased metrics such as Area Under the Receiver Operating Characteristic Curve(AUROC), F1 score, confusion matrix and evaluation toolbox SED EVAL [Mesaros et al., 2016] are also used as reference.

The Receiver Operating Characteristics (ROC) Curves is based on True Positive Ratio(TPR) and False Positive Ratio(FPR), in which the TPR, also called recall or $Sensitivity$, indicates the amount of actual positive correctly being predicted:

$$Sensitivity(TPR) = \frac{TruePositive}{TruePositive + FalseNegative} = 1 - Specificity \tag{14}$$

Where the $Specificity$, namely the True Negative Ratio(TNR), computed as:

$$Specificity(TPR) = \frac{TrueNegative}{TrueNegative + FalsePositive} \tag{15}$$

We can see for both sensitivity and specificity; the denominators are always the sum of actual positive or actual negative, which results in the evaluation not affected by imbalanced data. Since we want to increase TPR but keep FPR as lower as possible, the sharper a ROC curve, the better it performed on classification.

In other words, it is possible to use the area under the curve to quantify the evaluation; this is known as the Area Under ROC score(AUROC). It was being called a truly nonparametric index of sensitivity so that being suggested as an important accuracy metric. [Green et al., 1966] The value of 1.0 indicates perfect classification. Moreover, Hosmer et al. proposed a qualitative guidance for quantifying the quality of discrimination:

Score	Assessment
0.5	No Discrimination
(0.5,0.7)	Poor Discrimination
[0.7,0.8)	Acceptable Discrimination
[0.8,0.9)	Excellent Discrimination
[0.9,1)	Outstanding Discrimination

What needs to be emphasized that for binary classification, balanced accuracy is equivalent to the area under the receiver operating characteristic (AUROC) curve if we compute the FPR and TPR with the predicted class directly. And the AUROC score would be increased if using an entire range of probabilities to compute FPR and TPR. Taking one prediction as an example that two receiver operating characteristic curves are plotted in fig.2 and fig.3 that results from different FPR and TPR. In which the smooth one are implemented with more linear interpolation with the trapezoidal rule for computing the area. Therefore, in this thesis all the AUROC scores are defined as the area computed by FPR and TPR that calculated from the predicted probability.By doing this, the AUROC score is more accurate and nonconflicting with balanced accuracy.

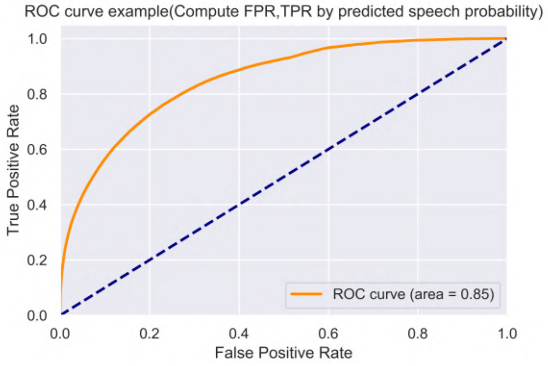


Figure 2: AUC calculated from FPR and TPR of predicted speech probability, which is the AUROC score used in our experiment

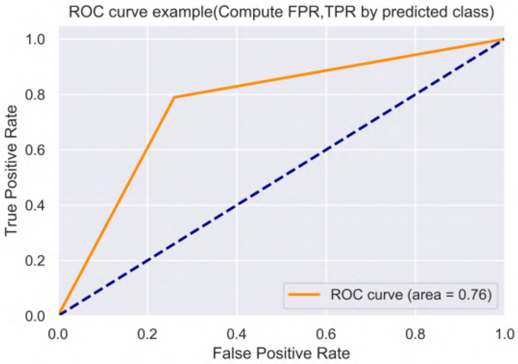


Figure 3: AUC calculated from FPR and TPR of predicted class, which is equivalent to balanced accuracy in binary classification

Sed Eval is an open-source library for evaluating sound event detection systems and acoustic scene classification. [Mesaros et al., 2016] For our VAD task, it provides a transparent way to compare the onset and offset of the speech event between the reference and prediction; hence in the evaluation stage, a segment-based evaluation will be performed by comparing the ground truth and our model's output on the test set at frame-level.

3.3.2 Hyperparameters Optimization

The Random Forest algorithm consists of multiple hyper-parameters. Hence, hyper-parameters optimization is one of the goals in the training progress, and it is achieved by performing the grid search. Grid search is the progress iteratively using different combinations, namely the parameter grid, to find the optimal value by comparing the model performance in each iteration. Implementing grid search is easy but can be very time consuming since it's an exhaustive searching approach, to handle it we could start with a coarse grid then finer the range on new grids.

We start by setting up the number of trees in the forest. One biggest advantage of using Random Forest is, more trees will effectively decrease the variance of the forest with no risk of over-fitting because of bootstrapping. But large scale of trees is computationally expensive, and it was proved that the generalization error of Random Forest would converge to certain values with more trees added. [Breiman, 2001] Even it was suggested that tuning number of trees is not recommendable for classification and more trees are always better. [Probst and Boulesteix, 2017]In our implementation, we will perform a search only on the number of trees first to find its maximum value before the generation ability reaches the performance plateaus, and fixed the amount of estimator as such number for all following experiments.

Once determined a rational amount for the number of trees in the forest, we believe the such number of uncorrelated trees could average out their individual error, stabilize overall performance before the diminishing returns. This gives us an outline of making the predictions as:

$$\hat{p_i} = \frac{1}{T}\sum_{t=1}^{T} I(\hat{y_{it}} = 1) \tag{16}$$

$$\hat{y_i} = \begin{cases} 1 & ,\hat{p_i} > 0,5 \\ 0 & ,Otherwise \end{cases} \tag{17}$$

where $\hat{y_{iy}}$ denotes the prediction by t-th tree (t = 1,2,...,T) for the i-th observation from the dataset; $\hat{p_i}$ is the probability of i-th observation classified as 1(the positive label) by averaging the predictions over all T trees; and the final prediction will be made as positive only if the averaging result greater than 0.5.

Move on other hyper-parameters. Essentially, we are focusing on the bias-variance trade-off between the strength of each tree and the overall stability. It usually achieved by tuning the depth of trees(max depth), the number of feature variables to consider for splitting(max features), the minimum number of observations required at the leaf node(min samples leaf), and the minimal size of observation needed for performing the split(min samples split). Other hyper-parameters such as criterion(Gini impurity or Information Gain), and the limitation of the amount of leaf nodes, act less effective or similar to the former so will not be covered in our grid search.

The limitation of the depth of trees controls how many splits that each tree keep expanding. It could grow without restriction so that the tree will keep expanding until pure leaves, which means the tree will capture more information from data. Consequently, for a single decision tree, it is in the risk of over-fitting. On the contrary, a shallow tree will result in more learning error due to the bias(under-fitting) but won't damage our model since the ensemble method with bagging will handle this problem.

Another way of constraining the depth of trees is by limiting the number of observations in the terminal node, namely, the min sample leaf in scikit-learn. This means that the tree could keep growing until only a single observation left at the leaf node. At the opposite, limit this value would lead to pruning if the splitting will result the end node with few observations, and the results are produced by averaging on those groups of samples, which offers better generalization ability. Another hyper-parameter that related to this is the minimal size of observations that required for splitting a node(min samples split in scikit-learn). Instead of pruning by setting the minimum requirements of observations at leave, this will just stop splitting if the resulting node has less observation than the min samples split.

To avoid over-fitting, the algorithms also randomly select the subset from all the features to consider when splitting a node. Heuristic from [Geurts et al., 2006] suggests the amount to the square root of the size of the global features. It's closely related to the quality of feature extraction; the value should be set high if only a few relevant information captured which ensures that each splitting could include useful variables to be considered. [Goldstein et al., 2011] This randomization prevents the tree fitted too closely and make trees less correlated with each other. Also, it was proved that the time complexity of the computation would decrease linearly when reducing the number of features to consider. [Wright et al., 2017]

3.3.3 Cross Validation

To avoid over-fitting and get better confidence in the process of evaluating, cross-validation is often implemented, which is a method of measuring generalization error by using holdout data. The general description of multifold CV proposed by Geisser [Geisser, 1975]. The K-Fold Cross Validation is an alternative to the computationally exhaustive Leave-One-Out scheme, which successively left out each data point for validation and fit the model with remaining data. The K-Fold Cross Validation improved it by preliminary dividing the train set D into K partitions, the model will be fitted with $k - 1$ folds of data and tested with the remaining one, namely the validation set. And this procedure repeated K times so that each fold will be used as a validation set for once iteratively, and the model performance measured by averaging over iterations. By doing this, the variance of predictions is reduced, and the model's stability could be guaranteed.

Since we take frame as the unit to arrange data, which risks the model of data leakage that a splitting might allocate frames from the same soundscape to both train and validation set, within numerous splitting schemes, we adopt the Group K Folds cross-validation in our experiment, which is a modified version of K-Fold Cross-Validation. It ensures the samples from the same soundscape file will not be separated when splitting. And it will be performed during the grid search for each combination of hyper-parameters.

3.4 Viterbi Decoding: The Smoothing Scheme

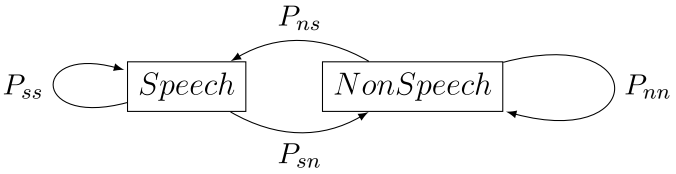
Because the model makes predictions on frame level, there is much erroneous decision, which makes the segmentation discontinuous; hence, one more stage for smoothing the misclassification and ambiguity is needed. The median filter is a common non-linear scheme that still used in recent research, which the final decision on each frame was made by considering the median of the frame with four neighboring bins. [Thambi et al., 2014]In our experiment, the Hidden Markov Model was used for modeling the speech occurrences, then perform the Viterbi

Decoding for rebuilding the corresponding status sequence(the final segmentation) given the observations likelihoods(the initial prediction sequence made by classifier). To explain this process, we have to start with First Order Markov Chain.

The Markov Chain was proposed by Claude Shannon in 1984 for modeling the natural language. [Shannon, 1948] Its embodiment on probability also called Markov Assumption that presented as:

$$P(q_i = a|q_1...q_{i-1}) = P(q_i = a|q_{i-1}) \tag{18}$$

Formally, it says given a sequence of state variables $q_1, q_2, ...q_{i-1}$, for the probability of current state being a is only depend on its previous state q_{i-1}, where a could be either speech or non-speech from the binary state sequence we denoted as Q. Moreover, a transition matrix A is computed from train set, which is a set of $p_{ss}, p_{ns}, p_{sn}, p_{nn}$ that encodes the probabilities of transitioning between state s(speech) and state n(non-speech).



And each of them is computed as the conditional probability of the transition from state i to state j:

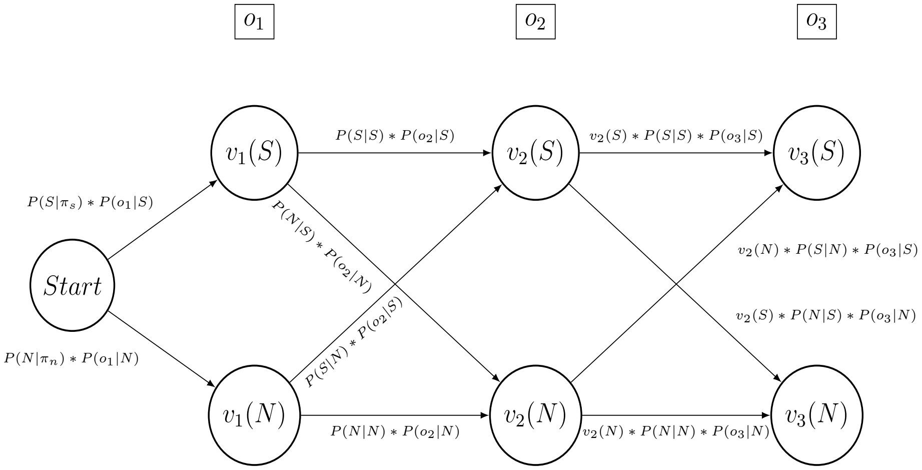
$$p_{ij} = P(q_t = j|q_{t-1} = i) \tag{19}$$

$$\sum_{j=1}^{N} s_{ij} = 1 \tag{20}$$

Where the i, j represent the state of either speech or non-speech, N denotes the number of states with $1 \leq i, j \leq N$ and N=2 in our binary classification problem. One last component that required to build the Markov Chain is the initial probability vector, which consists of the initial distribution over the probability of start states. In this thesis, we assume the initial states for speech and non-speech with uniform distribution. As the result, given the state sequence Q, the transition probability matrix A, and the initial probability vector $\pi = [\pi_n = 0.5, \pi_s = 0.5]$, the Markov Chain is well specified.

Unfortunately, for most cases in our daily life, the states we are interested in can not be observed directly,i.e., they are hidden. Instead, we may only have a sequence of observations O with another sequence of the observation likelihoods E, which also known as the emission probability matrix that encodes the probability of the observation o_t at time t being generated from state i. In our case, the time step t would be the frame index, and through the feature engineering, we obtained the MFCCs, the short time spectral representation of the sound as the sequence of observations. And its likelihoods sequence was generated by our Random Forest Classifier. Such extension for the Markov chain is called Hidden Markov Model.

Next, we have to uncover the hidden layer and find out the most likely states sequence corresponding our observation likelihood sequence. This is related to one of the three basic problems about Hidden Markov Model that given the observations sequence and model, find out the corresponding optimal states explanations [Rabiner, 1989] The solution to this problem is Viterbi Decoding, and the visualized Viterbi path in our problem as shown below:



Where we use our model to process the observations sequence from left to right, and it produces the state likelihoods $E_j(o_t)$ given the current observation j. The value of each cell represented as $v_t(j)$, which stands for the probability that given the first t observations likelihoods and the most probable state sequence $q_1, ...q_{t-1}$; and it is computed as:

$$v_t(j) = \max_{q_0,q_1,...q_{t-1}} P(q_0, q_1, ...q_{t-1}, o_1, o_2, ..., o_t, q_t = j|\lambda) \tag{21}$$

And we could use maximum over all previous states that we've already computed at time $t-1$ so the value of $v_t(j)$ at time t given the state j could be expressed recursively as:

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) a_{ij} E_j(o_t) \tag{22}$$

where $v_{t-1}(i)$ is previous Viterbi path probability at time step $t-1$, a_{ij} denotes the transition probability from state q_i to q_j, and $E_j(o_t)$ is the likelihood of the observation o_t given state j. Moreover, in Viterbi decoding we have to keep track the best $E_j(o_t)$ during each iteration for backtracking the best path from the end to beginning after recursion stage. A fully developed function for implementing Viterbi decoding was released in librosa [McFee et al., 2019] which eases the task and save a lot of work in our experiment.

3.5 Data Augmentation

For machine learning algorithms, one straightforward way to improve the generalization ability is fitting the model with more data as possible. Data Augmentation is the technique that gets around the problem of limited data by synthesizing more samples. It is widely used for object recognition problems in the computer vision field. Usually, several deformations are applied to training data such as rotation, reflection, scaling, etc. to expand the dimension of input. Such kind of deformation on visual features also was implemented in researches of audio and speech. Recently, Google Brain team proposed SpecAugment, an augmentation policy that directly transform the log-mel spectrogram with time wrapping, frequency masking, and time masking rather than the audio itself, to dealing with the problem of over-fitting, and large amount data is needed for Automatic Speech Recognition with Deep Learning. [Park et al., 2019]

Instead of implementing the transformation on visual representation, many other methods could be applied to audio directly. It requires the transformation not only deform the observable features(i.e., the position of annotations) but also have to change the labels and values as well. In Scaper, users could apply pitch shifting and time stretching on sound events while synthesizing the soundscape in order to increase the variability. [Salamon et al., 2017] Another option for quickly generating perturbations of training data is by using the open-source library Muda, in which user could customize their deformer, includes pitch shifting, time stretching, mixing with colored noise, and impulse response convolution; even multiple deformation objects could be combined to form complex transformation [McFee et al., 2015]

4 Implementation

4.1 Synthetic Soundscape Data Generation

4.1.1 Phase 1: Speech Corpus Processing.

In order to obtain the voice data that meet our requirement on variability and quality, we used the largest public data set of human voices contributed by Mozilla Common Voice, which consists of enormous amount of voice data in 29 different languages from more than 42,000 contributors. Currently, the data set has already collected 39,577 clips, 1,087 hours in total data for English.[5]

The form of web donation promises the variability of speakers that covers different ages, genders, and accents, but it also raises a challenge that the quality of the collected data was not uniformed though other users have validated all the data we used. This nonuniformity includes the various sampling rates that contributors used, the differences in duration and contents, and the onset and offset of the content we concern. For example, when users are contributing their voice, they have to start and stop recording by themselves, which might bring unexpected silences and noise (i.e., mouse-clicking) into the clip. Other worse cases, such as a relatively long audio clip only contains one transient cough, the distortion due to the user wrongly operated on the audio interface, or other non-speech contents recorded in the clip. All those conditions may have serious consequences of the wrong annotations. Since we have to synthesize both the train and test set, the misannotation would mislead the model during training and fail the metric when testing at the frame-level. Although those problems could be eased when we have enormous data, for the used clips, we manually cropped out the speech as possible to minimize the chance of leading inaccuracy during training and evaluating.

As the result, we trimmed 10,000 speech files by cropping out the valid voice contents from the silence/noise that occurred at the beginning and the end of the clips. The visualized statistic showed in figure.4, in which we could see our speech

[5] https://voice.mozilla.org/

corpus data set has the mean duration of 2.9s, with the maximum 16.75s and minimum 0.2s.

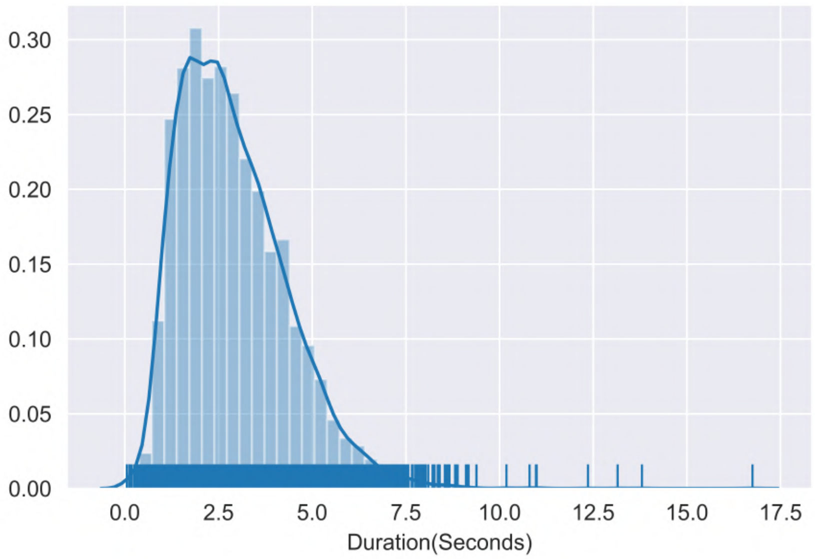


Figure 4: Speech Corpus Statistic

4.1.2 Phase 2: Inject New Human Voice Class

The UrbanSound8K data set groups audio slices into 10 folds. This prevents the bias of the same slice being sampled for both train and test set. In each fold, the amount of clips for each class is limited to avoid the vast difference in class distribution. All the clips collected from online sound repository Freesound.[6] As mentioned in the methodology section, we replaced the audio slices of children playing class by our trimmed human voice data without changing the number of clips, total duration, and distribution in each fold. Then, follow the procedure of generating UrbanSound-SED data set to create ours.

[6] http://www.freesound.org

As the result, the synthesized data set is similar to UrbanSound-SED that consists of 10 classes with various distributions and polyphony. One problem that has to be addressed is this procedure results in a data set for multiclass classification that each of the 10 classes along the distribution is average. This leads to skewed classes for our binary classification task with the 1:9 rate of speech vs. non-speech. Fortunately, it could be compensated by adjusting the weights on the class during training and performing the model evaluation with unbiased metrics.

4.1.3 Phase 3: Specification Tuning in Scaper

To increase the variability of the synthetic soundscape data, we applied various sound events specification, which probabilistically defines the properties of sound events. First, the duration of the clip was set to 10 seconds long, with Brownian noise normalized to -30 LUFS as background sound to simulate the 'hum' tone in the urban environment. Next, taking example by UrbanSound-SED that three distributions were used for defining the occurrence of foreground events:

Uniform distribution: given the number of foreground events generated randomly , the foreground events will spawn at a random time step between 0 to 10 uniformly; normal distribution($\mu = 5$, $\sigma = 3$), which is a Gaussian distribution with mean 5 and standard deviation 3; similarly, the bimodal distribution uses two Gaussians and let the event only follows either-or($\mu_1 = 3$, $\mu_2 = 7$, $\sigma = 2$). And for each event,its duration was chosen randomly uniformly between 0.5-4 seconds. By defining the onset of events in this way, the generated soundscapes are similar as possible to the distribution in the real world that some of the events occur independently, and some appear in polyphony that multiple events overlap with each other. As the result, the statistic of sound events duration and the number of events per class are presented in figure.5 and figure.6.

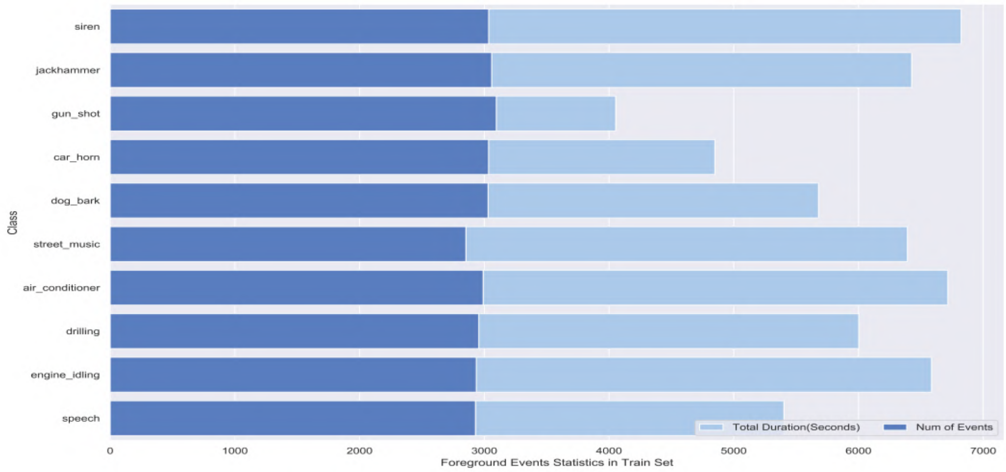


Figure 5: Occurrence duration and number of events per class in train set

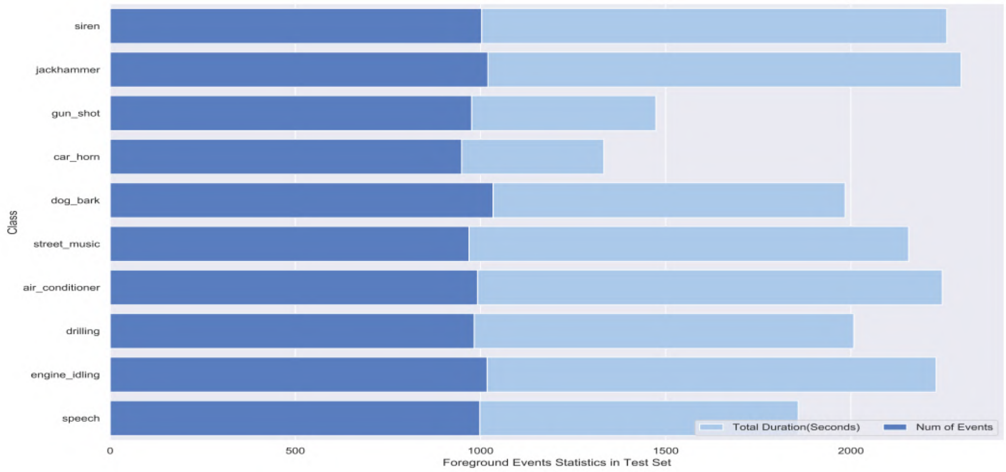


Figure 6: Occurrence duration and number of events per class in test set

4.2 Model Selection and Evaluation

The proposed Random Forest VAD was evaluated in various ways since we have multiple relative comparisons and trend analysis: first we want to examine how different feature extraction algorithms affect the model performance; Second, we want to evaluate our model as a function of SNR, test its robustness against different background noise level; Third, we expect to see the model performance improved through Data Augmentation and Viterbi Smoothing Scheme.

One of the first issues is determining the metric for our problem. As mentioned, we avoid bias from skewed class via two ways: First, we weight the cost of misclassification based on the class frequency that puts more emphasis on the minority class by inversely increase its weight. Second, all the metrics we select are unbiased such as balanced accuracy, which computes the arithmetic mean of sensitivity (true positive rate) and specificity (true negative rate). The former would be used during the training phase while the latter is used for grid search cross-validation and evaluating the model performance on the test set.

In the pre-processing stage, first, all the audio data are resampled at 22050Hz. The frame analysis is performed through a sliding window of 1024 samples with a hop size of 512, namely a 40ms analysis frame with 50% overlapping. Within the analysis frame, different feature extraction algorithms are performed respectively, and the data are arranged in the form of Numpy array with a dimension of [number of frames, number of coefficients] to meet the requirement of scikit-learn's interface.

We start with a basic level that using the Random Forest Classifier with default parameters settings. This forest only consists of 10 unpruned trees with gini impurity as criterion. Each tree in the collection fit with partial data through bootstrapping, at each node, it consider the square root of number of all features during training to calculate the best split. The tree will keep growing until we obtain pure leaves or only one sample left in the internal node. And the prediction is voted out by averaging all the trees' probabilistic prediction.

4.2.1 Hyperparameters Optimization and Cross Validation

As mentioned in the methodology section, there are 5 crucial hyper-parameters usually be tuned for Random Forest. In our implementation, we only turned three of them before data augmentation involved because first, we found the model often reaches performance plateaus when increasing the trees amount to 400 if the dimension of data is unchanged. Further trees would only cause computation intensive with little improvement in performance. Therefore, we fixed the number of estimators in the forest equals to 400, for which we believe the reasonable amount that large as possible in a definite scope. Second, since only 20 cepstral coefficients kept for both MFCC and PCEN features, there is no need to turn the hyper-parameter 'max feature' on such a low-dimension scale. Take our case as an example; typical choices such as square root and log of 2 would lead to the same result that 7 features would be considered when looking for the best split. On the other hand, we want to keep injecting randomness to keep trees less correlated, so it's better not to consider all of the features as well. As a result, the number of 'max feature' is fixed to the square root and remains unchanged through the following grid search. Since the scale of data is quite large, the grid search could easily slow down the computation if the grid has too many parameters. To avoid this, we limit the scale of the parameters grid so that the number of hyper-parameters combinations not exceeding 125(5 for each parameter). And the grid search is performing on a coarse grid first to see its variation trend, then go finer the range for each hyper-parameters, repeat the searching on the new grid until the model reaches the performance plateaus.

During the stage of feature extraction, a vector of file index was also created to indicate the source of each frame sample. This is used as the Group Index when implementing group-K-Cross-Validation to make sure that frame samples from the same file will not appear in different folds. Considering the time, we only set 3 splits for cross-validation. The procedure of hyper-parameters optimization with cross-validation takes about 150 hours on the given grid. In which it searches over 125 hyper-parameters combinations with 375 fittings in total. The grid search cross-validation is repeated once we change the data set, i.e., change the data set that extracted by different feature extraction algorithms or train the model with perturbations when implementing data augmentation. By doing this, we could promise the best hyper-parameters combination and score respect to the different training sets.

Taking one grid search cross-validation as an example, and its visualized process is shown below. In which we could observe the model significantly generalize better as increasing the number of max depth, and this trend of increment terminated once the depth reaches about 35. Second, we could observe that as the tree goes deeper, the hyper-parameter 'min sample leaf' and 'min sample split' have the trend of approaching the default values given by the scikit-learn: Fewer samples to considered for splitting a node into two subsets, and fewer samples on leaf node would benefit the model performance. As a result, each decision tree was fully grown and constructed, then pruning by making use of Gini to measure purity.

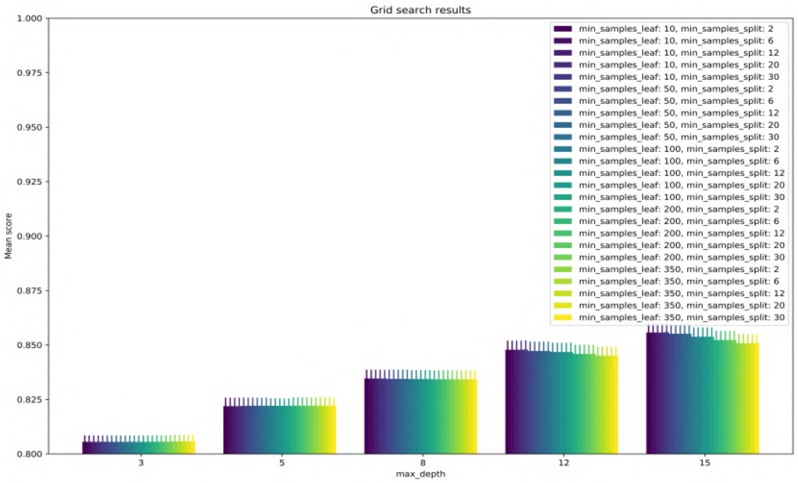


Figure 7: GridSearch CV Example.1

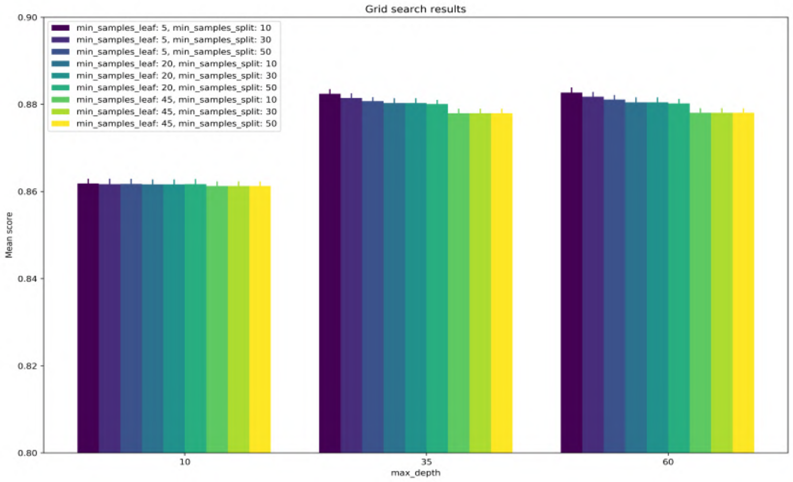


Figure 8: GridSearch CV Example.2

The grid search cross-validation is performed separately on three systems(MFCC, PCEN, PCEN-Lostanlen) to identify the best features with a higher score. The PCEN-Lostanlen features were processed by PCEN algorithms but using parameters that Lostanlen proposed: As he pointed out, the default parameters setting of PCEN in librosa are suitable for indoor application, and he further proposed a set of parameters for bioacoustic event detection that gaussianize and decorrelate the subbands successfully. [Lostanlen et al., 2018]The resulting best hyperparameters combinations with its best validation score are as follows in the table:

Features	Best Score	Num Of Trees	Max Depth	Min Split	Min Leaf
MFCC	0.8827	400	35	10	5
PCEN	0.8700	400	35	20	7
PCEN(Lostanlen)	0.8734	400	35	6	7

4.2.2 Feature Selection: Performance of model as a function of SNR

The stage of feature selection compares the validity of three feature extraction algorithms under the premise of the same scale of grid search cross-validation: the higher score means our algorithm could capture more information that correlated with the class. In order to test the ability to discriminate speech from non-speech at different SNR level respect to features, the test set was generated in 18 versions that each algorithm was used to extract feature separately on test sets with the SNR range 0-6, 6-12, 12-18, 18-24, 24-30, 0-30(Full Range). Through this comparison, we could see how the feature affect model performance in environments with different noise levels.

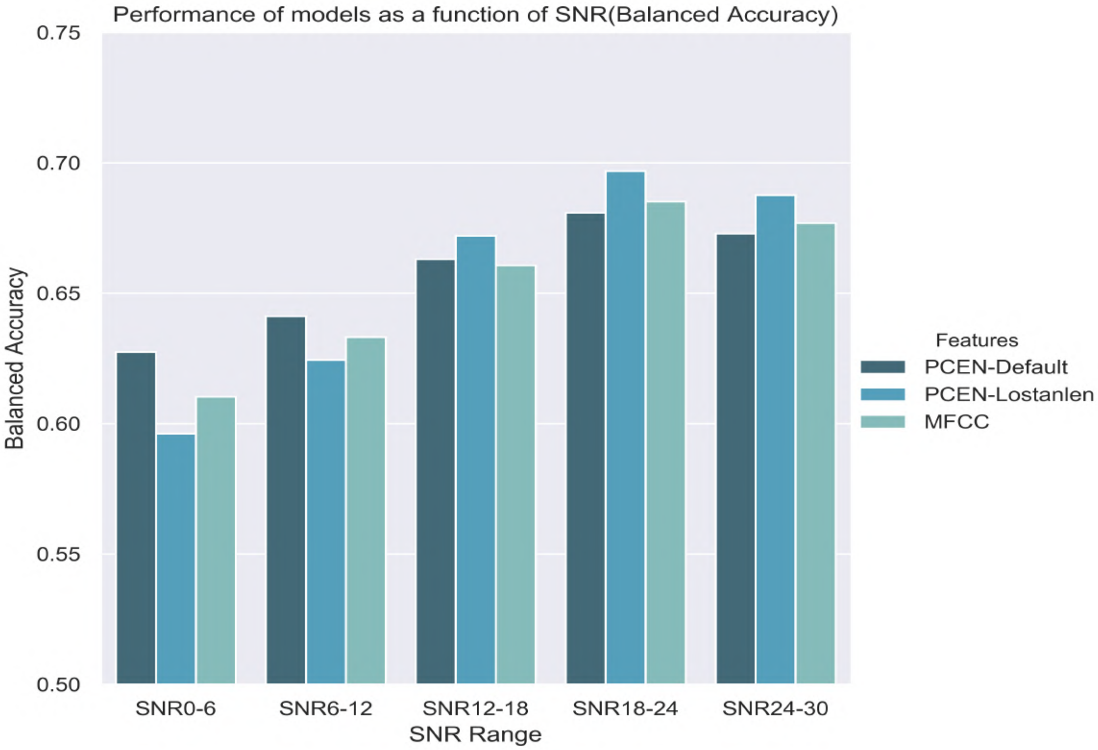


Figure 9: Balanced Accuracy vs. SNR

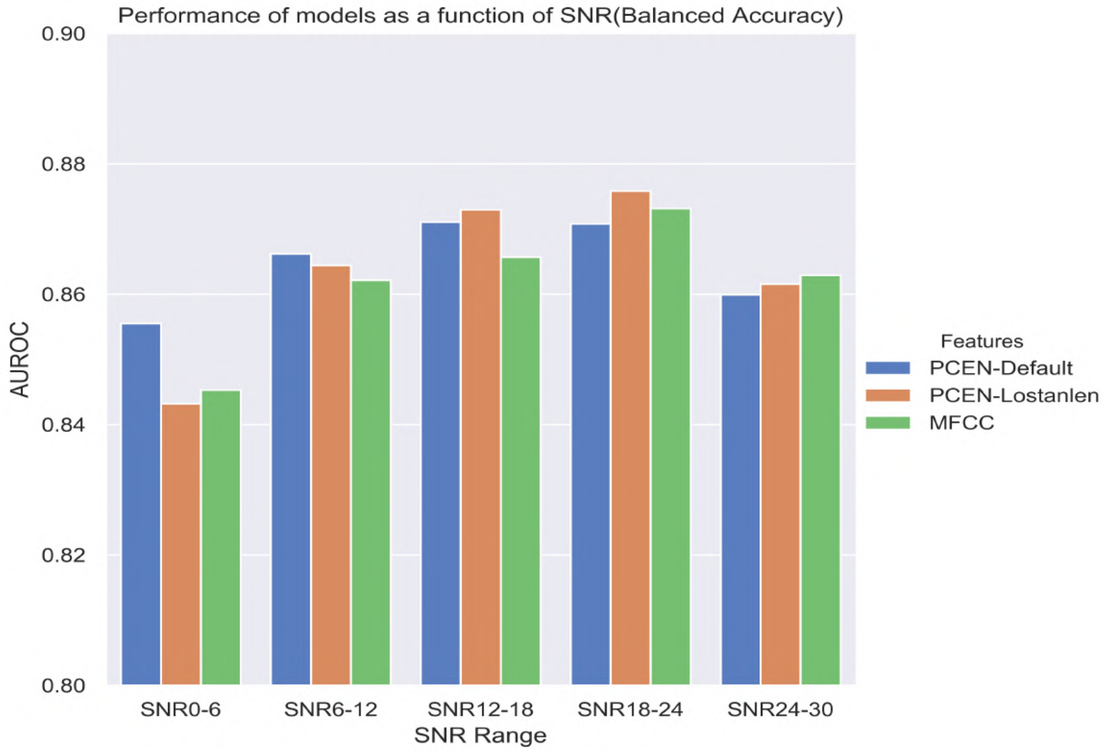


Figure 10: ROCAUC vs. SNR

From the above, we could observe that first for both of those three systems; the model performance is getting better as increasing SNR of foreground events except SNR 24-30. The potential reason is the foreground events would be generated with audible clipping distortion at such a high SNR level. As mentioned in methodology section, Scaper defines the sound events SNR as the LUFS units above the background level. In our experiment, since the background Brownian Noise was normalized to -30 LUFS first, there is a risk that distortion could happen when generating the foreground event with SNR 30. As a result, we discard the test set on SNR 24-30 in later experiments. Second, although the MFCC obtained the best score during the grid search cross-validation, the difference between the three systems is quite small that less than 0.1. Therefore, we have to examine the model performance on the test set and then decide which feature extraction algorithm meets our requirement of robustness.

As a function of SNR, those three features perform closely as well. It is worth noting that the difference between two PCEN-based systems: Lostanlen's setting seems influenced more by SNR. It performs poorly in low SNR range but getting better as SNR increasing, and outperforms the others when SNR higher than 12. By contrast, using the default librosa parameter settings seems more robust when sound events have a lower SNR. This property is important that ensure our model effective at a very low SNR condition. As a result, we determine to proceed with the rest of the experiment with the model that fits with data extracted by default PCEN parameters.

4.3 Augmented Train Set: Five Stages Transformation Pipeline

For data augmentation, the open-source library Muda facilitates the procedure of deforming audio with its annotations. The used deformations include time stretching, pitch shifting, additive colored noise, impulse response convolution, and dynamic compression; some of them only change the observable features, others deforms both the annotations' position and values of the observations. (i.e., time stretching, pitch shifting, and IR convolution) The deformed data could be either used for measuring or improving robustness, which depends on whether to use the perturbations as the train or test set.. [McFee et al., 2015] Since we want to improve the robustness of Random Forest voice activity detector, in the stage of data augmentation, we keep the test data unchanged and fit the model with perturbations to see if the model could generalize better. We use five deformation objects shown in the table below. The pitch shift and time stretch are done by Rubber Band audio[7], which transposes the pitch normally with zero mean and unit variance. The duration of soundscape was stretched with a rate that sampled log-normally with a scale of 0.3. The dynamic range compression was done by Sox[8],

[7] https://breakfastquay.com

[8] http://sox.sourceforge.net/

it applies the dynamic range characteristics given by Dolby-E standard Speech preset. [Dolby, 2002]. The Colored Noise deformer applies additive Brownian noise on the original soundscape with weight than randomly sampled between [0.1,0,9]. And IR Convolution deformer convolves the soundscape with a given impulse response file that offered in the Isophonics project conducted by Centre for Digital Music (C4DM) group at Queen Mary, University of London. [Stewart and Sandler, 2010]

Deformation	Parameters Setting
Random Pitch Shifting	Pitch $\sim \mathcal{N}(\mu = 0, \, \sigma^2 = 1)\,.$
Random Time Stretching	$\ln(Rate) \sim \mathcal{N}(\mu = 0, \, \sigma = 0.3)\,.$
Colored Noise	Brownian Noise $\in [MinWeight = 0.1, MaxWeight = 0.9]$
Dynamic Compression	Dolby E standards: speech
IR Convolution	Isophonics Room Impulse Response Data Set: Great Hall sample IR

Muda offers two building blocks named Union and Pipeline, which allow users to assemble complex transformation stages from single object conveniently. The Pipeline generates the training condition with cumulative effect and results in a sequence of complex transformations. In this thesis, 5 training conditions of increasing complexity were generated through the pipeline as:(N)No Augmentation, (P) Pitch Shift, (PT) Pitch Shift and Time Stretch, (PTD) Pitch Shift, Time Stretch, and Dynamic Range Compression, (PTDC) Pitch Shift, Time Stretch, Dynamic Range Compression, and Colored Noise, (PTDCI) Pitch Shift, Time Stretch, Dynamic Range Compression, and IR Convolution. The test-set score(balanced accuracy) over SNR ranges presented in Fig.11. The result shows all deformation except IR convolution consistently outperforms the baseline(No Augmentation). Although the improvement is relatively small($<3\%$), it proves our data augmentation approach works effectively and motivates more attempts. We further tried Union operator, which only applies one deformation at a time so that 5 transformations result in 5 training conditions with individual effect, and we could compare the result of each deformation.

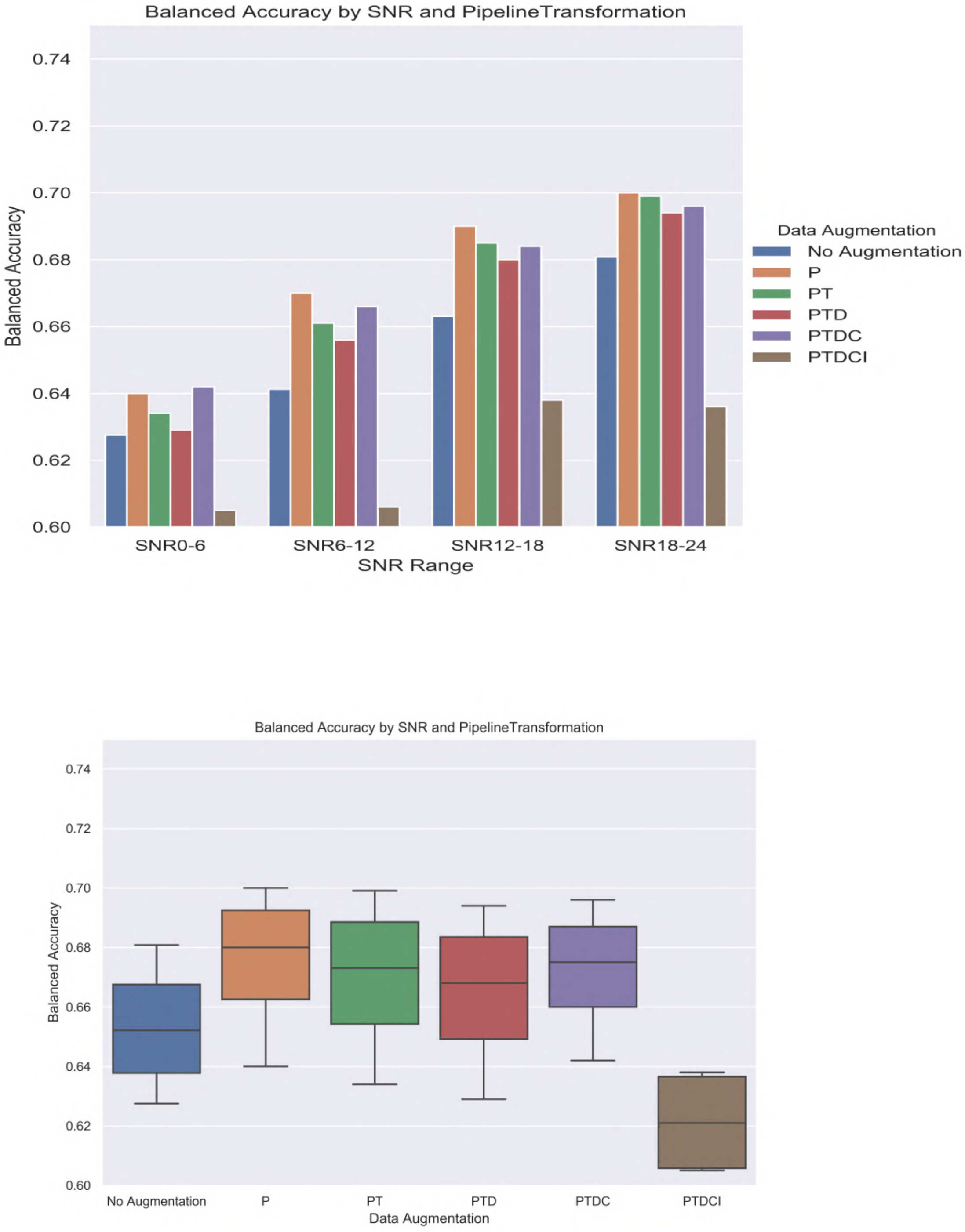


Figure 11: Pipeline test-set score (Balanced Accuracy) distributions over SNR 0-6,6-12,12-18,18-24

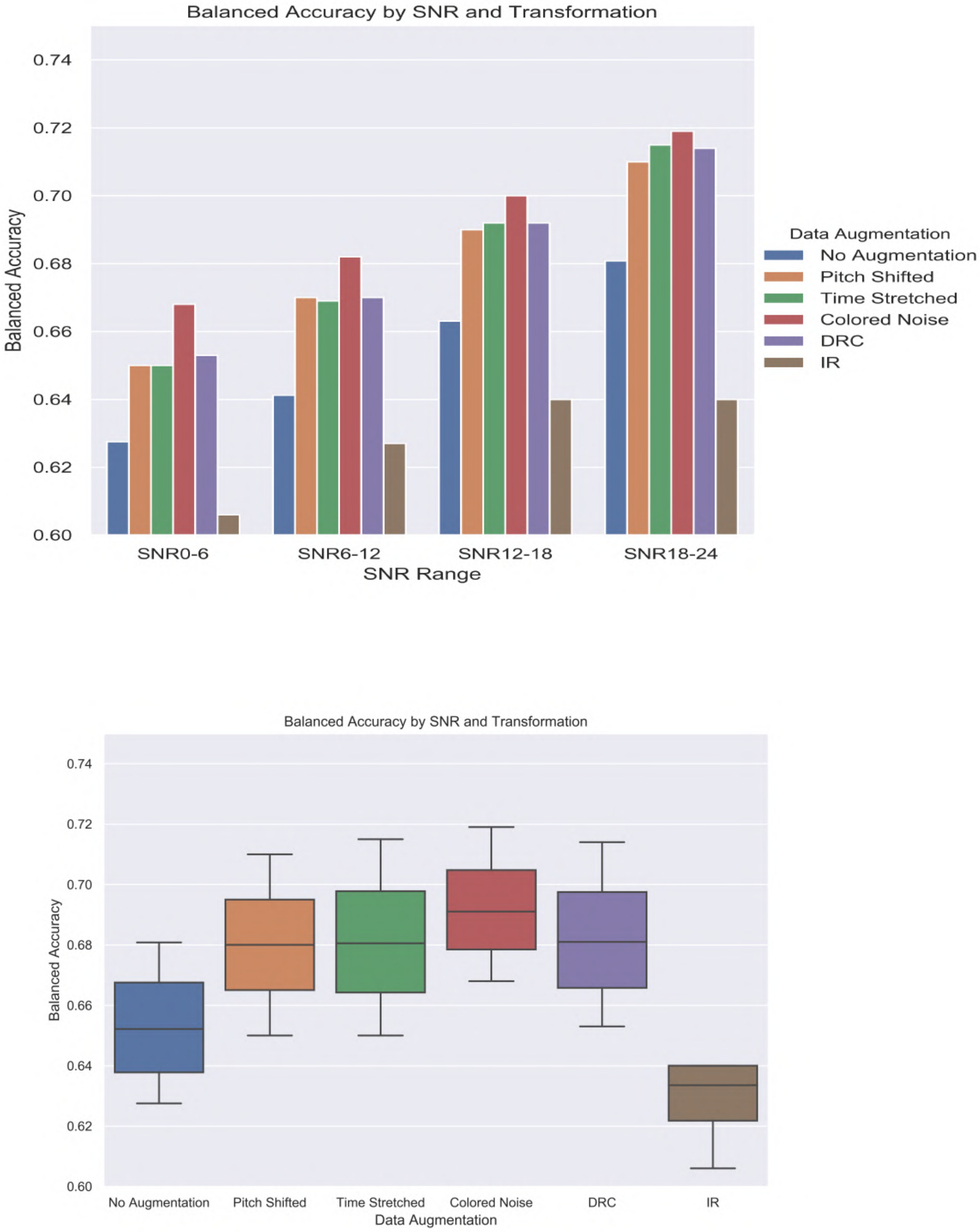


Figure 12: Union test-set score (Balanced Accuracy) distributions over SNR 0-6,6-12,12-18,18-24

Figure.12 shows the distribution of test-set performance across SNR. We could see for all deformations except Impulse Response convolution, at least %2 improvement after applying the transformation. Within five deformations, the colored noise deformation outperform others with %4 average improvement across SNR. And the upper and lower quantiles are also higher than no augmentation and other conditions. In consideration of SNR level, we could conclude that Colored Noise deformation would outperform others significantly when SNR is lower, as the SNR increases, all four deformations' results (Pitch Shift, Time Stretch, Colored Noise, Dynamic Range Compression) are getting close. In other words, the Time Stretch and Dynamic Range Compression deformations are exposed more to the SNR level.

As a result, we believe the Colored Noise deformation would be the most effective and simplest way that helps with improving the model performance for our problem. It has advantages of background noise-robust property, and transform the original data without complex pipeline and redundant outputs. Therefore, we continue the experiment with the model that trained with PCEN default settings and augmented by colored noise deformation. The model's output, the predicted probability sequence will be used as the observation likelihood and ready to be smoothed out by Viterbi Decoding scheme, which ensures we could get the best prediction.

4.4 Final Result With Viterbi Decoding

The implementation of Viterbi Decoding computes the most likely sequence of states given the conditional state predictions probability, which is our model's output. We assume the speech and non-speech states have the same initial probability 0.5, and compute the transition matrix from the training set: For our binary classification problem, the transition matrix approximately would be a diagonal matrix since both speech and non-speech events are tend to be continuous temporally, the current state would highly possible to stay unchanged comparing the previous one. The computed transition matrix is shown in figure.13: By taking the predicted speech class probability as the conditional likelihoods given the observation, which is the cepstral coefficients with PCEN frontend at time t, The Viterbi Decoding scheme computes the most likely sequence of states as mentioned in methodology. Figure.14 shows how the balanced accuracy improved through three stages, in which we could see the Viterbi Decoding effectively raise the score exceed 80% for all SNR ranges.

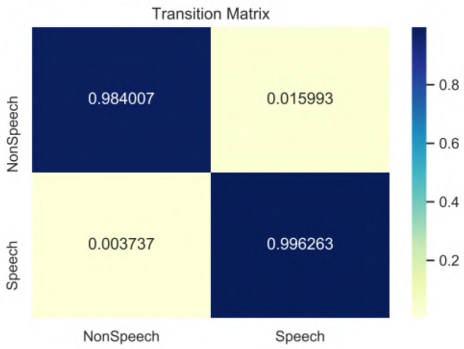


Figure 13: Transition Matrix Computed from Train Set

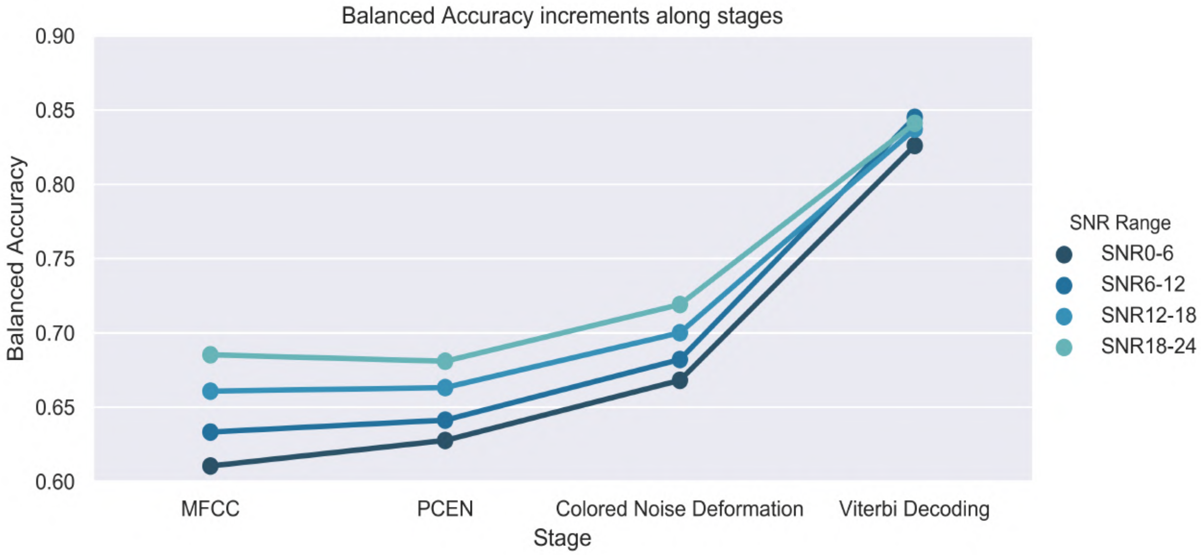


Figure 14: Increment of Balanced Accuracy through Data Augmentation and Viterbi Decoding

To illustrate the effect of Viterbi Decoding, we took one soundscape data from the test set as an example. The visualized annotations for ground truth and the prediction are shown in figure.15. The top two spectrograms depict the difference between MFCC and PCEN. From which we could see how PCEN avoids the disturbing of stationary noise, enhance the transient and onset of events. Sound events that slow changes in loudness such as drilling and engine idling are mostly discarded while others with higher amplitude modulation(e.g., siren and speech) are retained. The green line in the third plot indicates the ground truth of speech, blue and orange lines are the predicted speech class label and its probability. Moreover, we could see the frame-level predictions are intermittent. By modeling speech occurrence with the first-order Markov process, the correlative characteristic between consecutive frames was retained. The black dot line with triangle indicates the final decision of our system, in which the transient between states was delayed. Through the visualized annotations plot at the bottom, we could see the predicted speech label mostly overlaps the ground truth, which is what we look forward to.

Finally, we evaluated the most likely state sequence by using the segment-based metric offered in open source toolbox Sed Eval, which are proposed for evaluating polyphonic sound event detection. We defined the time resolution as 0.1, which means the evaluation would be performed in a 100 milliseconds long segment. Sed Eval provides a quick way to evaluate the system by calling the evaluator, which compares the CSV-formatted reference and estimation in pairs. As a result, for each soundscape file in the test set, we used the model to make the prediction and write a txt file for recording the information of onset, offset, and predicted class. Then, pair it with the reference as the file list argument of the evaluator function. The result is shown in the table below:

Event label	F-1	Precision	Recall	Sensitivity	specificity	Balanced Accuracy
Speech	40.3%	26.7%	82.2%	82.2%	65.0%	73.6%

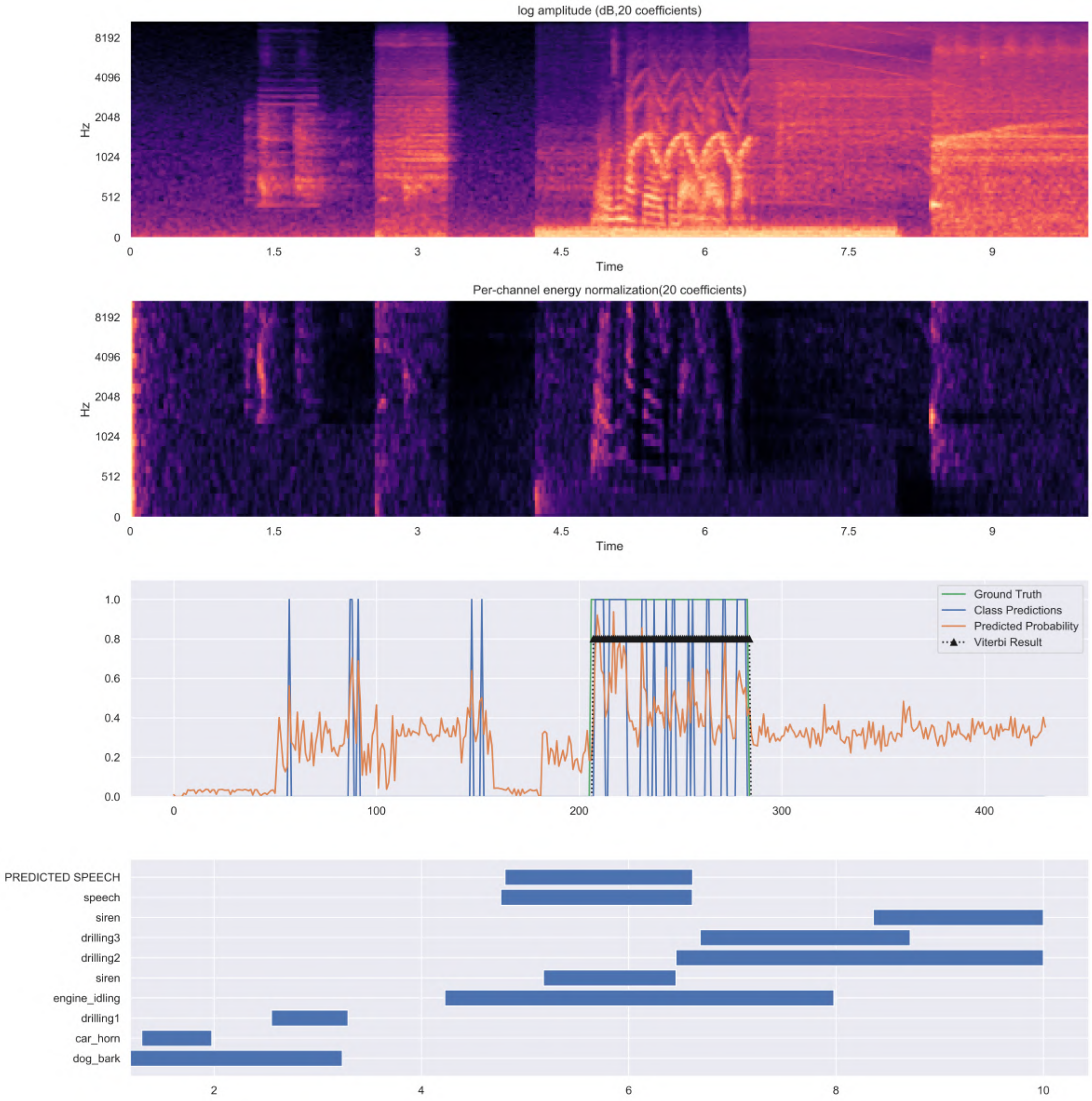


Figure 15: Visualized annotations of example file: soundscape-test-uniform35.wav

5 Conclusions

In this thesis, we made use of the Random Forest classifier to detect speech activity in the urban noise environment. For solving the problem of lacking suitable data collection, we followed the principle of Urban Sound Taxonomy, generate the synthetic soundscape data set by making use of Mozilla Common Voice and UrbanSound8K. The generating process remains the same as generating the UrbanSound-SED data set. To deal with the problem of skewed classes, in the training phase, we adjusted the costs of misclassification by inversely weighting the class proportionally to its frequency in the train set first. Then adopted unbiased metrics such as balanced accuracy and AUROC score through cross-validation and test-set evaluation.

During the stage of model selection, we first compared the influence of different feature extraction algorithms. This comparison was implemented with grid search cross-validation, which promises the best hyper-parameters combination and reliable result. The average result on the test set indicates that novel frontend PCEN performs very closely to our baseline feature, the log-mel frontend-based MFCC FB-40. But PCEN outperforms MFCC when taking the model as a function of SNR: PCEN demonstrated its robustness for dealing with low SNR sound events. Moreover, we compared two PCEN-based algorithms that use different parameter settings. Comparing to the default settings in librosa, the parameters proposed by Lostanlen is two-sided: It significantly outperform others in SNR range 12-24, but also discards the robust property in low SNR situations.

To further improve the robustness of our Random Forest classifier, we performed a step of data augmentation that expand the train set by generating perturbations through the software framework MUDA. We implemented five deformation objects includes pitch shifting, time stretch, dynamic range compression, additive colored noise, and impulse response convolution to transform the set. In order to test different effects of deformations, we arranged those deformations through Union and Pipeline separately. The former generates perturbations with individual effects while the latter kept transforming the train set with cumulative effects. Result indicate that except IR convolution, all other deformations could slightly contribute to the performance as we expected. (<4% increment on balanced accuracy)

We also observed that within the various deformations, the single Colored Noise deformation exposed less to SNR changing: It outperforms others when sound events have low SNR value. As SNR increasing, all deformations except IR convolution result in close performance. As a result, we only adopted Colored Noise deformation during the stage of data augmentation, for which we believe it meets our requirement of improving the robustness and easy to implement that we could effectively improve the model performance without assembling complex deformations and generating too many perturbations.

At the ending phase of our system, the speech occurrences were modeled by Hidden Markov Model. And a Viterbi Decoding hang-over scheme was implemented to find the most likely state sequence as final output. In order to find the most likely state sequence, it makes use of the predicted probability that produced by our Random Forest classifier as the conditional observation likelihoods; and a transition matrix computed from the train set that encodes the conditional probability of moving between speech and non-speech states. As a result, this smoothing scheme significantly boost the balanced accuracy from an average of 69.2% to 83.7%.

5.1 Discussion And Future Work

Our experiment did raise several problems worth to be addressed. First, as the result of evaluating model performance as a function of SNR, the model surprisingly degraded when predicting the test set with max SNR range 24 - 30. The most possible explain is the clipping distortion that generated by Scaper. This is due to the SNR in Scaper are set relative to the background in LUFS; the distortion might be generated if users set the level of background level too high.(e.g., -30 LUFS in our experiment) We contacted and confirmed with Scaper's author, and the warning would be added to the new vision of the library for informing users when events are distorting due to the high SNR values.

Another problem that has been addressed by Scaper's author is: Although Scaper is a powerful tool for generating synthetic soundscape data, it can not meet the requirement of richness and complexity of the realistic. This means Scaper still can't be used as the replacement for the data from real-world recordings. This warning makes us look forward to another remarkable project by NYU Music and Audio Resarch Lab called SONYC URBAN SOUND TAGGING (SONYC-UST), which is in the proceeding and almost complete. The SONYC-UST is a multilabel dataset of real-world recordings that captured through SONYC acoustic sensor network. [Bello et al., 2018] Currently, it is in the phase of human annotating. The annotation campaign is holding on Zooniverse that volunteers are manually annotating the presence of all classes in recordings. (92% completed in the meantime of this thesis). As a result, we believe the SONYC-UST data set should be a good alternation for our task in the future.

One most important property of PCEN is its enhance the transition between the stationary regime and stationary regime, which could be understood as the speech onset enhancement in our problem. This is critical for noise and reverberation robustness, and makes PCEN might be a good choice for more complex problems such as speech enhancement in reverb or far-field situations.The result of our experiment proves PCEN is more reliable when detecting the speech with low SNR value. Therefore, we have reasons to believe that PCEN merit additional study, especially for the problems with a background of far-field/low SNR situations.

Occasionally, we discovered a Scaper's extension project named AmbiScaper[9], which was developed to create realistic reverberant ambisonic soundscape data. With the rapid development of immersive audio, it would be an interesting attempt to implementing PCEN for sound event detection in the synthetic ambisonic soundscape, e.g., sound source localization.

[9] https://github.com/andresperezlopez/ambiscaper

References

[Atal and Hanauer, 1971] Atal, B. S. and Hanauer, S. L. (1971). Speech analysis and synthesis by linear prediction of the speech wave. *The journal of the acoustical society of America*, 50(2B):637–655.

[Atal, 1976] Atal, B. Rabiner, L. (1976). A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(3):201–212.

[Bello et al., 2018] Bello, J. P., Silva, C., Nov, O., DuBois, R. L., Arora, A., Salamon, J., Mydlarz, C., and Doraiswamy, H. (2018). Sonyc: A system for the monitoring, analysis and mitigation of urban noise pollution. *arXiv preprint arXiv:1805.00889*.

[Benyassine et al., 1997] Benyassine, A., Shlomot, E., Su, H.-Y., Massaloux, D., Lamblin, C., and Petit, J.-P. (1997). Itu-t recommendation g. 729 annex b: a silence compression scheme for use with g. 729 optimized for v. 70 digital simultaneous voice and data applications. *IEEE Communications Magazine*, 35(9):64–73.

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[Breiman et al., 1998] Breiman, L. et al. (1998). Arcing classifier (with discussion and a rejoinder by the author). *The annals of statistics*, 26(3):801–849.

[Breiman et al., 1984] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and regression trees.

[Brodersen et al., 2010] Brodersen, K. H., Ong, C. S., Stephan, K. E., and Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121–3124. IEEE.

[Chang et al., 2006] Chang, J.-H., Kim, N. S., and Mitra, S. K. (2006). Voice activity detection based on multiple statistical models. *IEEE Transactions on Signal Processing*, 54(6):1965–1976.

[Dattatreya and Sarma, 1981] Dattatreya, G. and Sarma, V. (1981). Bayesian and decision tree approaches for pattern recognition including feature measurement costs. *IEEE transactions on pattern analysis and machine intelligence*, (3):293–298.

[Dave, 2013] Dave, N. (2013). Feature extraction methods lpc, plp and mfcc in speech recognition. *International journal for advance research in engineering and technology*, 1(6):1–4.

[Dolby, 2002] Dolby, E. (2002). Standards and practices for authoring dolby digital and dolby e bitstreams. *Dolby Labortories, Inc.*

[El-Maleh et al., 2000] El-Maleh, Khaled, Klein, M., Petrucci, G., and Kabal, P. (2000). 2000 ieee international conference on acoustics, speech, and signal processing. In *Speech/music discrimination for multimedia applications*, volume 2, pages 2445–2448.

[Enqing et al., 2002] Enqing, D., Guizhong, L., Yatong, Z., and Xiaodi, Z. (2002). Applying support vector machines to voice activity detection. In *6th International Conference on Signal Processing, 2002.*, volume 2, pages 1124–1127. IEEE.

[Ephraim and Malah, 1984] Ephraim, Y. and Malah, D. (1984). Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on acoustics, speech, and signal processing*, 32(6):1109–1121.

[Geisser, 1975] Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American statistical Association*, 70(350):320–328.

[Geurts et al., 2006] Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

[Goldstein et al., 2011] Goldstein, B. A., Polley, E. C., and Briggs, F. B. (2011). Random forests for genetic association studies. *Statistical applications in genetics and molecular biology*, 10(1).

[Green et al., 1966] Green, D. M., Swets, J. A., et al. (1966). *Signal detection theory and psychophysics*, volume 1. Wiley New York.

[Guo and Viktor, 2004] Guo, H. and Viktor, H. L. (2004). Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *ACM Sigkdd Explorations Newsletter*, 6(1):30–39.

[Hermansky, 1990] Hermansky, H. (1990). Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752.

[Ising et al., 2004] Ising, H., Kruppa, B., et al. (2004). Health effects caused by noise: evidence in the literature from the past 25 years. *Noise and Health*, 6(22):5.

[Itakura, 1968] Itakura, F. (1968). Analysis synthesis telephony based on the maximum likelihood method. In *The 6th international congress on acoustics, 1968*, pages 280–292.

[Itakura, 1975a] Itakura, F. (1975a). Line spectrum representation of linear predictor coefficients of speech signals. *The Journal of the Acoustical Society of America*, 57(S1):S35–S35.

[Itakura, 1975b] Itakura, F. (1975b). Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):67–72.

[Jo et al., 2009] Jo, Q.-H., Chang, J.-H., Shin, J., and Kim, N. (2009). Statistical model-based voice activity detection using support vector machine. *IET Signal Processing*, 3(3):205–210.

[Kim and Stern, 2016] Kim, C. and Stern, R. M. (2016). Power-normalized cepstral coefficients (pncc) for robust speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(7):1315–1329.

[Kinnunen et al., 2007] Kinnunen, T., Chernenko, E., Tuononen, M., Fränti, P., and Li, H. (2007). Voice activity detection using mfcc features and support vector machine. In *Int. Conf. on Speech and Computer (SPECOM07), Moscow, Russia*, volume 2, pages 556–561.

[Liu et al., 1997] Liu, Z., Huang, J., Wang, Y., and Chen, T. (1997). Proceedings of first signal processing society workshop on multimedia signal processing. In *Audio feature extraction and analysis for scene classification*, pages 343–348.

[Lostanlen et al., 2018] Lostanlen, V., Salamon, J., Cartwright, M., McFee, B., Farnsworth, A., Kelling, S., and Bello, J. P. (2018). Per-channel energy normalization: Why and how. *IEEE Signal Processing Letters*, 26(1):39–43.

[Lu et al., 2002a] Lu, L., Zhang, H.-J., and Jiang, H. (2002a). Content analysis for audio classification and segmentation. *IEEE Transactions on speech and audio processing*, 10(7):504–516.

[Lu et al., 2002b] Lu, L., Zhang, H.-J., and Jiang, H. (2002b). Content analysis for audio classification and segmentation. *IEEE Transactions on speech and audio processing*, 10(7):504–516.

[McFee et al., 2015] McFee, B., Humphrey, E. J., and Bello, J. P. (2015). A software framework for musical data augmentation. In *ISMIR*, pages 248–254.

[McFee et al., 2019] McFee, B., Lostanlen, V., McVicar, M., Metsai, A., Balke, S., Thomé, C., Raffel, C., Lee, D., Lee, K., Nieto, O., Mason, J., Zalkow, F., Ellis, D., Battenberg, E., , , Yamamoto, R., Bittner, R., Choi, K., Moore, J., Wei, Z., nullmightybofo, Friesch, P., Stöter, F.-R., Hereñú, D., Thassilo, Kim, T., Vollrath, M., Weiss, A., Carr, C., and ajweiss dd (2019). librosa/librosa: 0.7.0.

[Mesaros et al., 2016] Mesaros, A., Heittola, T., and Virtanen, T. (2016). Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6):162.

[Moolayil, 2019] Moolayil, J. (2019). *An Introduction to Deep Learning and Keras*, pages 1–16. Apress, Berkeley, CA.

[Mosley, 2013] Mosley, L. (2013). A balanced approach to the multi-class imbalance problem.

[Park et al., 2019] Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019). Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*.

[Park et al., 2013] Park, T. H., Turner, J., Jacoby, C., Marse, A., Musick, M., Kapur, A., and He, J. (2013). Locative sonification: Playing the world through citygram. In *ICMC*.

[Park et al., 2014] Park, T. H., Turner, J., Musick, M., Lee, J. H., Jacoby, C., Mydlarz, C., and Salamon, J. (2014). Sensing urban soundscapes. In *EDBT/ICDT Workshops*, pages 375–382. Citeseer.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

[Probst and Boulesteix, 2017] Probst, P. and Boulesteix, A.-L. (2017). To tune or not to tune the number of trees in random forest. *Journal of Machine Learning Research*, 18:181–1.

[Rabiner, 1989] Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[Rabiner and Sambur, 1975] Rabiner, L. R. and Sambur, M. R. (1975). An algorithm for determining the endpoints of isolated utterances. *Bell System Technical Journal*, 54(2):297–315.

[Ramirez et al., 2007] Ramirez, J., Górriz, J. M., and Segura, J. C. (2007). Voice activity detection. fundamentals and speech recognition system robustness. In *Robust speech recognition and understanding*. IntechOpen.

[Ramırez et al., 2004] Ramırez, J., Segura, J. C., Benıtez, C., De La Torre, A., and Rubio, A. (2004). Efficient voice activity detection algorithms using long-term speech information. *Speech communication*, 42(3-4):271–287.

[Ramírez et al., 2006] Ramírez, J., Yélamos, P., Górriz, J., and Segura, J. (2006). Svm-based speech endpoint detection using contextual speech features. *Electronics letters*, 42(7):426–428.

[Renevey and Drygajlo, 2001] Renevey, P. and Drygajlo, A. (2001). Entropy based voice activity detection in very noisy conditions. In *Seventh European Conference on Speech Communication and Technology*.

[Russell and Norvig, 2016] Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

[Salamon et al., 2014] Salamon, J., Jacoby, C., and Bello, J. P. (2014). A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044. ACM.

[Salamon et al., 2017] Salamon, J., MacConnell, D., Cartwright, M., Li, P., and Bello, J. P. (2017). Scaper: A library for soundscape synthesis and augmentation. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 344–348. IEEE.

[Scheirer, 1997] Scheirer, E. Slaney, M. (1997). Construction and evaluation of a robust multifeature speech/music discriminator. *IEEE international conference on acoustics, speech, and signal processing*, 2:1331–1334.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423.

[Siegel and Bessey, 1980] Siegel, L. and Bessey, A. (1980). A decision tree procedure for voiced/unvoiced/mixed excitation classification of speech. In *ICASSP'80. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 53–56. IEEE.

[Slaney, 1998] Slaney, M. (1998). Auditory toolbox. *Interval Research Corporation, Tech. Rep*, 10(1998).

[Sohn et al., 1999] Sohn, J., Kim, N. S., and Sung, W. (1999). A statistical model-based voice activity detection. *IEEE signal processing letters*, 6(1):1–3.

[Stevens et al., 1937] Stevens, S. S., Volkmann, J., and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190.

[Stewart and Sandler, 2010] Stewart, R. and Sandler, M. (2010). Database of omnidirectional and b-format room impulse responses. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 165–168. IEEE.

[Thambi et al., 2014] Thambi, S. V., Sreekumar, K., Kumar, C. S., and Raj, P. R. (2014). Random forest algorithm for improving the performance of speech/non-speech detection. In *2014 First International Conference on Computational Systems and Communications (ICCSC)*, pages 28–32. IEEE.

[Urbanowicz and Moore, 2015] Urbanowicz, R. J. and Moore, J. H. (2015). Exstracs 2.0: description and evaluation of a scalable learning classifier system. *Evolutionary intelligence*, 8(2-3):89–116.

[Valero and Alias, 2012] Valero, X. and Alias, F. (2012). Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification. *IEEE Transactions on Multimedia*, 14(6):1684–1689.

[Wang et al., 2017] Wang, Y., Getreuer, P., Hughes, T., Lyon, R. F., and Saurous, R. A. (2017). Trainable frontend for robust and far-field keyword spotting. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5670–5674. IEEE.

[Woo et al., 2000] Woo, K.-H., Yang, T.-Y., Park, K.-J., and Lee, C. (2000). Robust voice activity detection algorithm for estimating noise spectrum. *Electronics Letters*, 36(2):180–181.

[Wright et al., 2017] Wright, M. N., Dankowski, T., and Ziegler, A. (2017). Unbiased split variable selection for random survival forests using maximally selected rank statistics. *Statistics in medicine*, 36(8):1272–1284.