CLASSIFICATION OF ANOMALOUS MACHINE SOUNDS USING I-VECTORS

A Dissertation
Presented to
The Academic Faculty

By

Maham Tanveer

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2020

CLASSIFICATION OF ANOMALOUS MACHINE SOUNDS USING I-VECTORS

Approved by:

Dr. David V Anderson, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mark Andrew Davenport
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Justin Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: April 24, 2020

If everything seems under control, you're not going fast enough

Mario Andretti

Dedicated to those who think they are in the wrong place.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor Dr. David Anderson for his valuable and consistent support, guidance and feedback. His guidance has helped me to shape the path of my research project and his support has kept me motivated through difficult times.

I would like to thank Chuyao Feng who has taken time out to help me with my work and provide practical guidance. I am grateful for his technical help in getting this project started and keeping it afloat.

I would also like to thank the committee members Dr. Davenport and Dr. Romberg for their time.

In the end, I am grateful for the support of my family and friends. I am grateful for my parents, my sister, my brother and my best friend for their continued encouragement and ignoring my moods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of the proposed work is to analyze and study the use of i-vectors for Anomalous Detection of Sounds (ADS) in Machines. ADS is a very hot research area because of its widespread practical usage in audio surveillance, product inspection and maintenance among many other areas. It has applications in both supervised and unsupervised tasks. Supervised ADS tasks incorporate known structure for anomalous data while unsupervised tasks operate on outlier detection methods. In either scenario, anomalous data is often scarce and restricts the confidence in performance metrics of a model.

ToyADMOS is a recently released database that provides a solution by making a database using toys. Since toys are cheap it is easier to introduce defects and generate sound samples. This provides an opportunity to work with different kinds of models and test their applicability to machine sounds.

I-vectors were released in 2011 by Dehak et. al for speaker recognition tasks. They have since been used in music recognition, accent classification, age regression, and many other acoustic based problems. To the best of our knowledge they have not been studied for ADS tasks in machines, which provides an exciting opportunity for researching their use for a new domain of problems. Our contribution in this research work can be divided into three parts. First, we demonstrate i-vectors' suitability for modelling the acoustic features of machines. Second, we analyze different methods of training the Universal Background Model for i-vectors and discuss the results. Lastly, we show both supervised and outlier-based detection techniques and discuss their results.

CHAPTER 1
INTRODUCTION

Anomalous Detection of Sounds (ADS) has received a lot of attention because of it's diverse and practical applications. ADS has been used for surveillance [1], gun-shot detection [2], product inspection [3] and product maintenance [4]. ADS is used both as an independent measure or in addition to visual/other information. Prompt response to changes observed in equipment sounds can increase reliability and safety with expensive and dangerous machinery.

ADS is divided into two broad categories, supervised ADS and unsupervised ADS. Supervised ADS comprises of tasks where anomalous sounds and their acoustic structures are defined and can then be used to train the models. This includes environment detection, gun shot detection, audio tagging etc. These models are specific to the type of anomalies being studied and may perform badly or unexpectedly in case of unexpected anomaly.

Unsupervised ADS tasks are more common in situations where anomalies are not defined but there is an ample information of the type of normal or expected acoustic structure expected. An anomaly is defined as anything which is significantly outside this normal or expected structure, or an outlier. Therefore, unsupervised ADS problems are popularly dealt with outlier detection techniques. The distance between a model trained on normal sounds and the given anomalous (or test) sound is taken. This difference is known as anomaly score, and it determines whether the test sample is an outlier or not based on a threshold [5]. Figure (1.1) shows the difference between the two techniques.

ToyADMOS [6] is a recently released database in which different toys are used to model machinery sounds. Each toy data is further divided by using different combinations of mechanical components like motors and gears. In each toy, and each mechanical combination, intentional defects and mechanical problems are produced. These sounds are then recorded

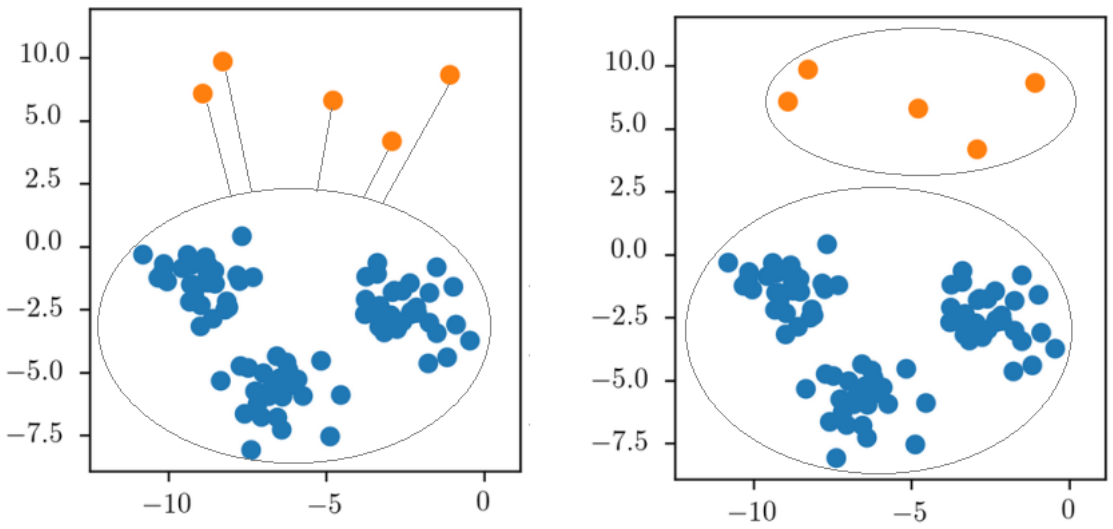

Figure 1.1: Unsupervised vs Supervised Detection of anomalies

in a low-noise environment. With a decent data size of anomalous sounds is possible to study different techniques in both supervised and unsupervised ADS acoustic modeling techniques and classification.

The basic steps to our ADS detection are pre-processing of data, i-vector generation and classification, both supervised and unsupervised. Pre-processing of audio files consists of several steps of extracting relevant information from its structure. This may consist of time domain based features like RMSE or zero crossing rate, frequency domain based features like dominant frequency or spectral centroid, and perceptual features like MFCs (Mel frequency ceptrums).

In 2011, Dehak et. al [7] published I-vectors as an improvement on Joint Factor Analysis for speaker classification. I-Vectors are a compact and low dimensional representation of an audio signal based on a Universal Background Model.

I-vectors have been used for various applications like accent classification [8], age classification [9], infant crying detection [10] and acoustic scene classification [11, 12]. I-vectors have been used to achieve very strong results in capturing acoustic similarities or differences within acoustic or sound data.

To the best of our knowledge I-Vectors have not been studied with Machine based Anomalous Detection of Sounds. ADS systems based on auditory input are highly dependant on the representation, feature extraction and pre-processing of the input. I-vectors which work well in capturing the intra-speaker dissimilarities and provide a low dimensional representation to work with, are a good candidate for researching in ADS. As ADS is researched from both unsupervised and supervised point of view, we are also going to study this from both perspective.

The thesis is structured as follows. Chapter 2 details relevant literature review of both ADS and i-vector. Mathematical techniques and details are discussed for calculation of i-vectors. We also discuss why we opted for i-vectors, how the models will be trained using i-vectors based on ADS methodologies and why these methodologies are employed for our case. In Chapter 3 current methods for ADS and popular techniques of using i-vectors for classification and outlier detection methods are discussed. Chapter 4 details in length our proposed solution, the system setup, and performance metrics used. It includes different techniques of training the UBM, the metrics used in training and generating i-vectors, metrics and techniques used in supervised classification and outlier detection methods. Chapter 5 provides a lengthy detail of all results and a discussion of the outputs. Finally Chapter 6 wraps up the thesis in a concise conclusion of our solution and future proposed work.

CHAPTER 2
BACKGROUND STUDY

In this section we go over technical background and concepts in Anomalous Detection of Sounds, the process of i-vector generation, principal component analysis and supervised/unsupervised classification models.

2.1 Anomalous Detection of Sound

Anomalous Detection of Sound (ADS) is a very hot field because of it's vast implementation in physical and practical scenarios. Various sensors are used to detect sound and vibration input for detection of faults and maintenance requirement in machines [3]. The most basic method is to employ an expert who listens to the sounds and provides an estimate of the machine's condition. However, this is severely limited by a human's physical capacity, prone to over looks and has gaps of no surveillance. [13].

2.1.1 What are Anomalies?

Anomalies are data samples that do not conform to the defined structure of normal data. Anomalies can be introduced in the data through many different reasons. It could be intentional like hacking a system, credit card fraud etc, or through unintentional unwanted sources like break down of a machine, a sick animal etc. Some applications of anomaly detection include surveillance [14], animal husbandry [15, 16] and in air crafts. [17].

Information taken through audio input have several benefits. (1) Audio is much cheaper to record, analyze and store. It is much more feasible to have a high quality audio microphone running 24/7 than, for example, a high quality camera. On the other hand audio input, in some cases, can suffer from low accuracy. Therefore, often a low/medium quality visual input is incorporated with audio input for classification. In [18] a vehicle detection

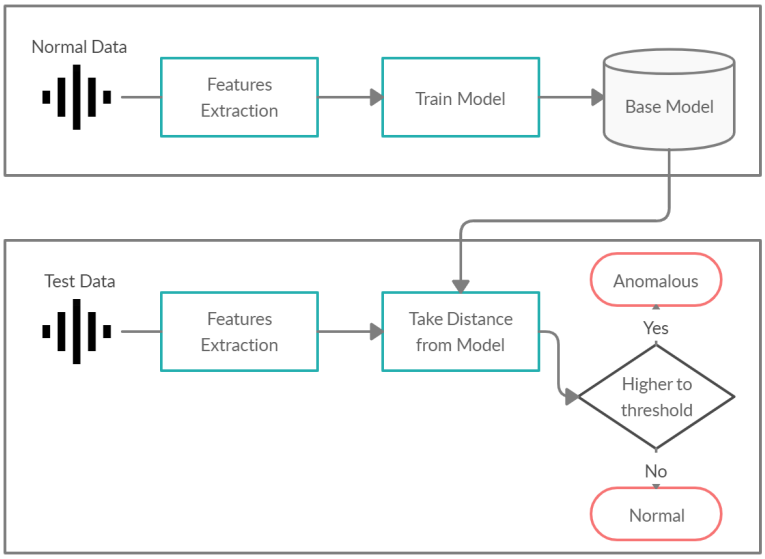

Figure 2.1: Flow of Unsupervised ADS

system is proposed using both audio and visual input. (2) Many scenarios do not have the option of visual information. For example, in environmental detection for security purposes, such as scream or gunshot detection [2], audio is the only input available.

2.1.2 Supervised vs Unsupervised ADS

As mentioned before ADS operates on two fronts. If there is enough information available for the type of anomalous sounds expected, than a model can be trained to detect them specifically. Such systems are able to detect anomalies with high accuracy but require both normal and anomalous data to recognize the properties of anomalous data points [19].

Methods to detect anomalies without utilizing anomalous data have also evolved to cater to situations where anomalous data is either extremely scarce or it has no expected structure. Therefore, they are tackled through unsupervised one-class classification techniques [20]. Such methods have been used in machine health inspection [21, 22], security [23] and non-speech detection [24].

Any test signal that deviates from this developed model or structure is regarded as anomalous or abnormal. Selecting audio features which maximize this difference is very

important in developing a robust model, which is also a very difficult task. This difficulty increases if the differences between normal and anomalous data are present in only a small subset of features. An anomalous "score" is calculated as the distance between the trained model and the test signal. This score is compared to a threshold and a binary decision of "normal or anomalous" is given. Figure (2.1) shows the basic flow of this process.

2.2 Performance Metrics

In a good anomalous event detection system True Positive Rate (TPR), the anomalies correctly identified as outliers, is maximized while False Positive Rate (FPR), normal sounds classified as outliers, is minimized. Figure (2.2) shows an average situation of medium overlap between the distributions of normal and anomalous samples. Let the green peak represent normal samples and the blue peak represent anomalous samples. A boundary defined by the overlay aims at capturing the maximum anomalous samples while trying to minimize the normal samples in its range. Anomalous samples in this range are the "True Positives" while the normal samples in this range are "False Positives". True positive rate and false positive rate are thus defined as

$$TPR = \frac{TruePositive}{TotalPositive} \tag{2.1}$$

$$FPR = \frac{FalsePositive}{TotalNegative} \tag{2.2}$$

We also have True Negative Rate (TNR) and False Negative Rate (FNR)

$$TNR = \frac{TrueNegative}{TotalNegative} \tag{2.3}$$

$$FNR = \frac{FalseNegative}{TotalPositive} \tag{2.4}$$

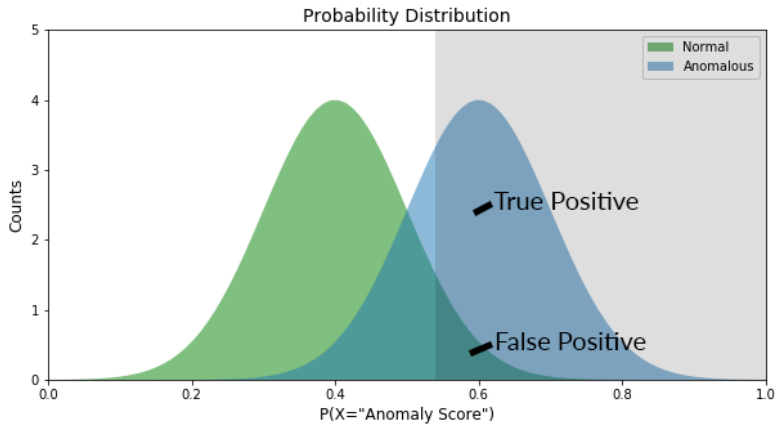

Figure 2.2: TPR vs FPR

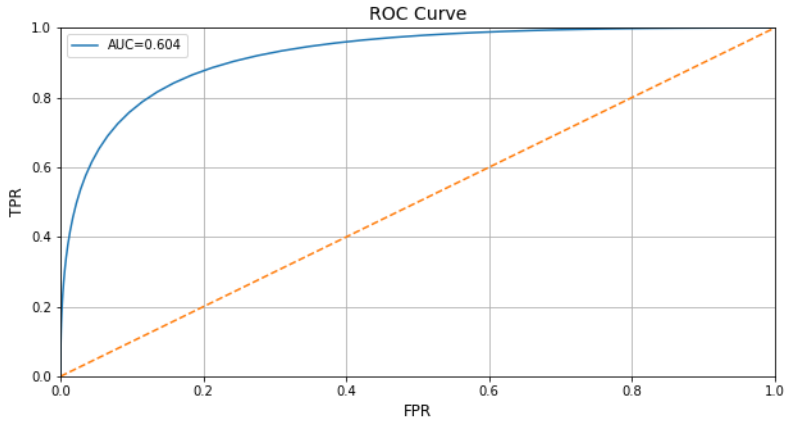

Figure 2.3: Area under ROC Curve

2.2.1 Accuracy

Accuracy is the number of samples that have been correctly identified. In other words it is

$$Accuracy = \frac{TrueNegative + TruePositive}{Total} \tag{2.5}$$

Accuracy is the most primitive and simple to apply performance metric. But if the classes are imbalanced it can be misleading.

2.2.2 Area Under ROC Curve

Area under ROC curve (AUC) is another common performance metric and a representation of the trade-off between FPR and TPR. The higher the AUC the better the model is in differentiating between normal and anomalous samples. An AUC of 1.0 is a perfect model while an AUC of 0.5 represents a model randomly guessing. An AUC of 0.5 is usually considered the worst a model can perform. Anything below 0.5 is often an issue in the way data has been labelled prior to training or testing. Figure (2.3) shows this trade-off between FPR and TPR.

Limitations of Accuracy and AUC

Accuracy and AUC are suitable in situations where the positive and negative samples have comparable number of samples. For ADS cases this is not always the case, in fact in most cases we have far more normal samples as compared to abnormal. AUC also requires that the models give probability to each sample based on a varying threshold of boundary. Some one class classifiers used in ADS unsupervised detection do not assign a probability, but rather give a defined label of 1 or 0. In such a situation AUC is also not a suitable metric to utilize.

2.2.3 F1 Score

In ADS test cases we are often left with very few anomalous samples to deal with. Consider the scenario in Figure (2.4). If we calculate accuracy we get $Acc = \frac{TruePositive+TrueNegative}{Total} = \frac{982}{990+20} = 97\%$

This may lead to a false interpretation of the models capacity. However looking at True Positive Rate we get $TPR = \frac{TrueNegative}{Total} = \frac{2}{20} = 10\%$ and $FPR = \frac{FalsePositive}{Total} = \frac{10}{990} = 1\%$. Very low TPR shows the model failed in recognizing anomalies completely. However with Accuracy we get an inflated sense of accomplishment. AUC which depends on both TPR being high and FPR being low, will be biased due to very low FPR. Which again is

Predicted

Actual		Normal	Anomalous
	Normal	980	10
	Anomalous	18	2

Figure 2.4: Precision and Recall

dependant on the high skew of class samples in both classes and is not a representation of the models capability.

Therefore we need another performance measure to append to our list of performance metrics. F1 score is a good measure in such a case. It depends on Precision and Recall.

Precision and Recall

Precision and Recall are another two metrics which aim to understand the $individual$ performance of both normal and anomalous classification. They are defined as

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{2.6}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2.7}$$

Precision shows how many samples detected as anomalous are actually anomalous, while Recall shows how many anomalous samples were correctly identified. Finally F1 score is defined as

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.8}$$

F1 Score is used to have a balance between Precision and Recall. It is a suitable measure to use in cases of large difference between class sizes, and to give a priority to anomalous samples being identified correctly.

2.3 Speaker Recognition and i-vector

In [7] i-vector were proposed as an improvement over Joint Factor Analysis (JFA) for speaker recognition. Speaker recognition is the task of determining whether an audio sample comes from a specific speaker or not. It is used to determine identity of a speaker, determine a change in speaker, group segments belonging to the same speaker, remove segments belonging to speakers not in question etc.

The process of i-vector generation is lined out in Figure (2.5). It consists of extracting features from audio, training a Universal Background Model (UBM), calculating the Baum-Welch (BW) statistics, training a Total Variability Subspace (TVS) and calculating the i-vector. After i-vector have been calculated, dimension reduction techniques and normalization may be applied if needed.

2.3.1 Feature Extraction

There are three ways of extracting features from an audio file [25]. Frame level, block level, and file level. Frame level divides the audio into small chunks and then extracts features for each chunk. Every chunk is uniquely classified and aggregated results over the entire file are used to provide a final comment or label through scoring [26]. Block level features generation analyzes the audio in a block wise manner. Each block is comprised of multiple frames (as previously discussed). Features extracted from one block are combined over a file to form a complete representation.

File level feature extraction has become a very popular technique. It is suitable for applications where there is not a time based information to be captured. In [12] the authors present a fusion of i-vector generated using file-level cepstral features and a deep convolutional network. A major advantage of file level features is that dimension reduction techniques like Principal Component Analysis, Linear Discriminant Analysis and similar projection techniques can be used.

File based features can be used to train Gaussian Mixture Models which produce a representation of the audio input. These representations can then be used for similarity scoring and classification. For example in [27] a Gaussian model is trained with MFCCs and FP and the similarity between songs is found using Kullback Leibler divergence.

2.3.2 Universal Background Model

Audio features like MFCCs or spectral features like spectral centroid, spectral density etc, are used to train a Universal Background Model (UBM). MFCCs have become a favourite and have been used in many audio applications like music [25], turbine engines [28] and DC machines [29]. MFCCs provide a compact representation of the spectral envelope of the audio. Perceptual Linear Prediction (PLP) is another feature very similar to MFCCs. It is motivated by hearing perception. PLP is shown to have some improved noise robustness, but MFCCs are generally thought of as safer choice specifically for speech tasks.

UBM is a Guassian Mixture Model (GMM) that is composed of hundreds of Guassians which aim to model the feature distribution of all input audio files. In [30] audio evaluation methods are discussed using Gaussian Mixture Models for diagnosis and classification of asthma attacks based on frequency analysis of breathing sounds. [31] also proposes a gender classification system based on GMMs which classifies based on gender. A GMM based UBM is a set of super-vectors of means, standard deviations, and weights. These means, standard deviations, and weights correspond to each Guassian in the model.

2.3.3 Baum-Welch Statistics

File-level features are used alongside UBM to train a set of statistics for each set of features [32]. These statistic are called Baum-Welch (BW) statistics. For an M length feature vector the 0^{th} (N) and 1^{st} (F) order BW stats are calculated as follows

$$N(i) = \sum_{n=1}^{M} \gamma_n(i) \tag{2.9}$$

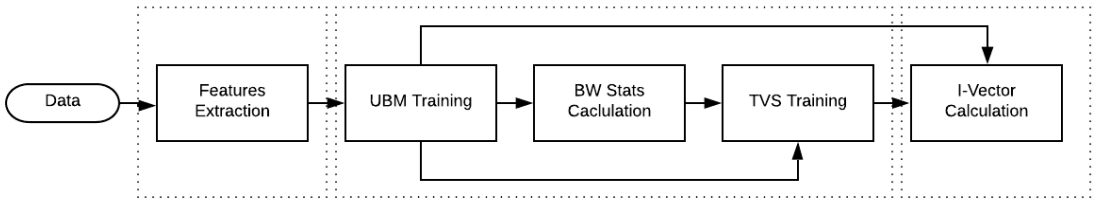

Figure 2.5: i-vector Generation

$$F(i) = \sum_{n=1}^{M} \gamma_n(i) * Y_n \tag{2.10}$$

Where $\gamma(i)$ is the posterior probabilities of Guassian component of GMM, and Y is MFCC feature vector.

2.3.4 MAP Adaptation and i-vector Extraction

I-vector model both speaker and channel variability and are used for speaker verification and classification. Total Variability Space (TVS) is a subspace trained using UBM and BW Stats. TVS assumes that the super-vectors which represent a set of features in GMM can be decomposed as

$$\mu = m + Tw \tag{2.11}$$

Here m is the mean super-vector in GMM corresponding to UBM. T is the low dimensional TVS space. Subspace T is calculated using factor analysis. w is normally distributed $N(0, 1)$ vector, known as the i-vector.

Maximum A-Posteriori (MAP) Adaptation

The process of adapting the super-vector in GMM to a specific feature set is called MAP adaptation. Target training data is aligned to the current GMM model, with sufficient statistics the model learns to adapt the internal model to the current complete training feature

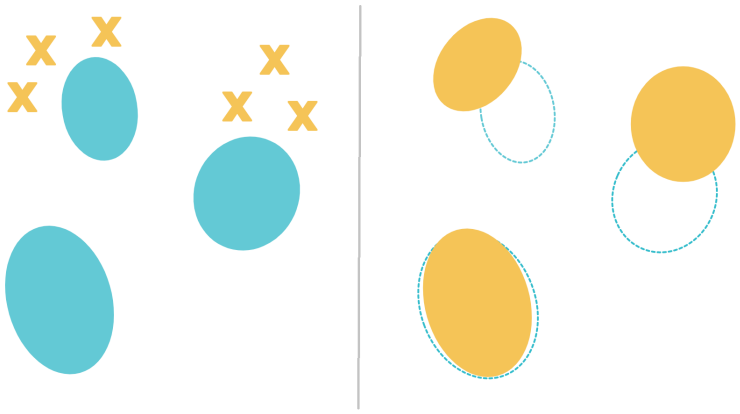

Figure 2.6: MAP Adaptation

set. This adaptation is captured in the subspace TVS. Individual i-vector then represent each file separately as mentioned above. This process can be seen in Figure (2.6). The blue ellipses represent the GMM peaks in a 2D representation. The orange crosses show the training data.

I-vector is a compact representation of the audio input, which is then used in place of the audio file for classification, regression or any other modelling technique. A method for calculating T and w is detailed in [33].

2.4 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) [34] is a supervised linear transformation technique used to remove irrelevant or unnecessary dimensions. LDA is popularly employed in reducing the dimension of i-vector which are often very high. It projects the i-vector into a new sub-space that is tuned to optimize inter-class separability while minimizing intra-class separability. The dimensions of LDA output is one less than the number of classes. It is commonly used for i-vector, however it is more suitable in places where number of speakers is high as otherwise resulting dimension can be too low to be of use.

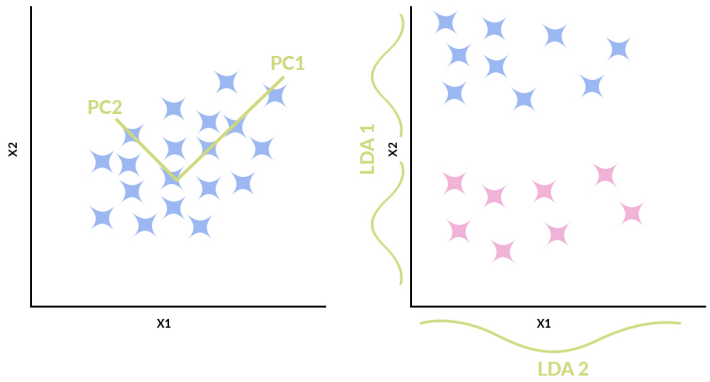

Figure 2.7: PCA vs LDA

2.5 Principal Component Analysis

Principal Component Analysis [35] is a feature extraction and dimension reduction technique. It finds a set of dimensions that are all orthogonal to each other, that is they are all linearly independent to each other. Eigenvectors and eigenvalues corresponding to the covariance matrix of data are calculated. Eigenvalues are then sorted from high to low, and so are their respective eigenvectors. Top k sorted eigenvectors are taken where k is the desired dimension of the output. If the number of samples are n and m is the original dimension, then the resulting transformation can be seen as

$$Output_{k*n} = EigenVectors_{k*m} \cdot Input_{m*n} \tag{2.12}$$

CHAPTER 3
METHODS AND TECHNIQUES

In this chapter we will discuss some current methods of approaching ADS problems. This will provide some insight of the kinds of approaches currently being employed. We will also detail some popular techniques employed in i-vectors based classification and regression. After this section, we will be at a position to make our case of using i-vectors for ADS cases.

3.1 Anomalous Detection of Sound

Our problem consists of detecting anomalous machine sounds, which is traditionally accomplished using unsupervised techniques. As mentioned previously, this is because the structure of anomalous sounds is not defined. ADS, event detection, or anomaly detection is an immense field with many applications in niche domains. In this section we focus more on unsupervised techniques and their challenges.

3.1.1 The Challenge

In theory, anomaly detection is to define a region of interest and declare any outlier as anomaly. However in practise this introduces a lot of challenges and difficulties. [20]

- Normal region boundaries are hard to define, as normal behavior is not completely predictable as well. Anomalous sounds may also be very close to the structure of normal, which makes it even more challenging to define a boundary.
- Availability of labeled data to test systems response to anomalies is usually low.
- If anomalies have been maliciously introduced, they may have masking making it harder to differentiate from normal data.

- Noise can sometimes mask or make it harder to define an anomaly from a noisy normal sample.
- The definition of anomaly changes from task to task. This depends on the expected fluctuation in normal data and the severity of missing anomalous samples. For eg. it is acceptable to reject a finger print for security at a military base over the slightest deviation from the normal. However, for personal phones and tablets, it would not be desirable to have an average user input their finger prints a couple of times before logging in.

Due to these reasons anomaly detection is often narrowed to the task, database, situation and severity.

3.1.2 Traditional Methods for Unsupervised ADS

An anomaly score $A(x, \theta)$ is calculated as the distance of the test sample from a normal model. Here, x is the the test sample and θ is our set of parameters of the normal model. [5]. x is often a processed version of raw audio composed of a set of extracted features.

3.1.3 Guassian Mixture Models

A PDF-based process may use Gaussian Mixture Model (GMM) [36] to model the normal data and separate outliers based on aposterior likelihoods. The GMM is trained on the normal data and anomaly score is calculated as follows

$$A(x, \theta) = -ln\, p(x/\theta, y = 0) \tag{3.1}$$

where y is the label of the data, and y=1 is an outlier. The p or probability measure is calculated using

$$p(x) = \sum_{k=1}^{K} \pi_k G(X/\mu_k, \sigma_k) \tag{3.2}$$

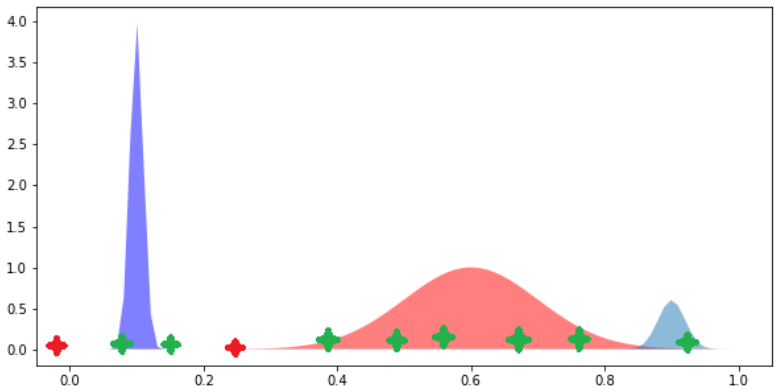

Figure 3.1: Guassian PDF Models

μ_k and σ_k are the mean and standard deviation of the k^{th} Gaussian of the model. Figure (3.1) shows a simplified 2D three peak Gaussian model with some outliers.

If the anomaly score is above a threshold, the test sample is categorized an outlier. Accuracy or performance metrics in this case depend on calculating the true positive rate, outliers correctly identified as outliers; and false positive rate, normal sounds incorrectly identified as outliers. Figure (2.2) shows the trade-off relationship between the two factors. There is a limit to how much a well chosen threshold can increase the model performance. It is essential to have the model parameters θ optimized to reduce the area of overlap between TPR and FPR. [3]

GMMs have also been used for supervised anomaly detection. In [2] the authors use a parallel set of two GMMs independently trained on audio inputs of screams and gunshots. As the structure of the 'anomaly' is determined or expected, a supervised technique works well in detecting it.

3.1.4 Simple Auto Encoder

An Auto Encoder (AE) is a deep learning technique to learn the core structure of a set of data. There has been a lot of work in using an AE to construct a normal model [5]. An AE uses two neural networks termed encoder (E) and decoder (D). The E task is to encode or

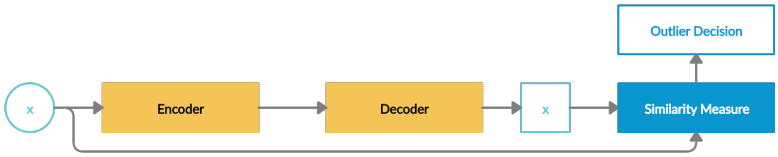

Figure 3.2: Auto Encoder Outlier Detection

convert the training data to a latent vector z. The D attempts to reconstruct the input using this latent representation producing $\widetilde{x}$. In this way a set of latent features representing the input data is trained.

$$z = E(x/\theta_E) \tag{3.3}$$

$$\widetilde{x} = D(z/\theta_D) \tag{3.4}$$

The models are trained to reduce the reconstruction error $||x - \widetilde{x}||^2$. For making a normal model the parameters are trained on normal samples only. Afterwards test samples are fed into the AE and compared to their reconstructed version.

Theoretically if the model is able to reconstruct it well, the sample most likely follows the normal configuration and therefore is classified as normal. If the reconstruction is poor, the sample most likely is an outlier. Figure (3.2) shows an overview of this method of outlier detection. Care needs to be taken in training the AE, if it is too generalized it would not be able to detect the outliers.

3.1.5 Advanced Auto Encoder

There are many variations of the classic Auto Encoder employed for specific tasks. Two versions used for ADS outlier detection are detailed below.

Denoising AE

As mentioned earlier Auto Encoders can suffer from over generalization or over fitting. Denoising Auto Encoders bypass this issue by training the AE on a corrupted version of

the input. This forces the latent feature layer to learn more robust features of the training data. This can be specially useful in case of anomaly detection where anomalies may share many features with normal data, and a simple AE may not be able to understand the difference on its own. The input x is corrupted using an additive Guassian noise

$$x_{noisy} = x + N(x, \sigma * I) \tag{3.5}$$

Afterwards the Auto Encoder is used to train the latent feature layer with x_{noisy} as the input and x as the target. The process discussed in Figure (3.2) is then used to check for outliers.

In [37] the authors use auditory spectral features with a bidirectional Long Short-Term Memory (LSTM) denoising autoencoder. Auditory spectral features are extracted using Short Time Fourier Transform. As both the encoder and decoder are neural networks, a variety of different types of neural networks can be utilized here. Traditional neural networks do not have a concept of context. Recurrent neural networks employ loops to deal with this short coming. LSTMs are a type of recurrent neural network that have shown to work very well with long term dependencies. Bidirectional LSTMs involve making a copy of the first recurrent layer in the network, and providing the input sequence as-is to one layer, and reversing it for the other layer. Using Bidirectional LSTMs give a strong sense of context and completeness to the analysis of audio signals.

Variational AE

Vartional Autoencoder defer from the traditional AE on the fact that they are used to generate new data, without an input to reconstruct. Variational AEs learn the parameters of the probability distribution representing the training data. We can then model from this probability distribution and generate new and unique data samples, similar to Generative adversarial network (GANs).

An anomaly detection method is proposed in [38] using reconstruction probability from

a variational autoencoder. Reconstruction probability includes the probablistic properties of the variational autoencoder, which makes it a better measure than reconstruction error. They also focus on using the variational autoencoder's generative capabilities to analyze the cause of the anomaly. In [39] a variational autoencoder based anomaly detector is presented that is able to incorporate both supervised and unsupervised cases. Thereby improving results with known anomalies through supervised classification but not degrading performance with unknown anomalies through unsupervised detection.

3.1.6 Support Vector Machine

Support Vector Machines (SVMs) are a type of supervised machine learning algorithm, used in binary classification problems. An SVM takes input data for both classes and generates a decision boundary that best separates them. This boundary, known as the hyperplane, is one which maximizes the margins from both classes. That is, the boundary is as far as possible from both classes. This gives higher robustness to the classifier. Kernels are also frequently used with SVMs for non-linear and complicated data sets by effectively mapping the input data into a higher-dimensional space prior to finding a decision boundary. In [1] the authors use an SVM based model for road surveillance, to detect emergency situations of tire skidding and car crashes.

One-Class SVM

One-class SVM is a classification method for detecting anomalies. Unlike the traditional SVM which is trained with samples from both classes One-class SVM is trained with one class samples which is the "normal" class. It makes a models based on the properties of normal cases and from this it can predict which test samples are unlike the normal training samples. This is useful for detecting anomalies which are typically scarce and varied in nature.

3.2 I-vectors for Classification and Regression

i-vectors are a feature-modelling technique that builds upon acoustic features extracted from the audio input. To the best of our knowledge i-vectors have not been used with machine sounds. Therefore, in this section we discuss some other popular applications they have been used in and that were referred to while researching this project.

3.2.1 Speaker Verification

This is a broad section covering all elements of detecting a speaker. This could be based on individual identification, age or gender identification, accent classification or any other aspect that can be used. In [7] i-vectors are presented for use in speaker verification.Two datasets NIST 2006 with 350 males, 461 females, and 51,448 test utterances, and 2008 SRE with 1140 females, 648 males and 37,050 files, are used to train and test the model respectively. Channel compensation techniques, within-class covariance normalization (WCCN), linear discriminate analysis (LDA), and nuisance attribute projection (NAP) were used. Within Class Covariance Normalization (WCCN) uses information from class labelled training data to find orthonormal directions in feature space that maximize task-relevant information. Two classification methods were used to test the model, SVM with cosine kernel, and a cosine similarity measure.

Distinct features of an individual beyond their identity, can be extracted from their voice. This includes age, gender, accent etc. [9] use Least Squares Support Vector Regression to estimate age of speakers. Speaker age regression means estimating age of a speaker from an unknown utterance. They also employ supervised classification with Cosine Distance Scoring which relates the unseen utterance to the model parameters and finds the closest class corresponding to the speakers age. The database “aGender” with seven age-gender classes was used for testing. In [40] an accent recognition model based on i-vectors is presented. They used Support Vector Machine (SVM), the Naive Bayesian Classifier

(NBC) and the Sparse Representation Classifier (SRC). The accents are differentiated by speakers native to Hindi, Russian, Thai, Vietnamese, American English, and Cantonese, speaking English language. SVM and SRC are shown to perform best with i-vectors.

3.2.2 Language Detection

Language Identification (LID) is the task of recognizing the language being spoken in an audio, given that only one language is being spoken. In [41] a language recognition system based on i-vectors is presented. In this paper, the authors trained an SVM for each individual class (or language). The number of boundaries thus match the number of classes of languages. WCCN is used alongside LDA and Neighborhood component analysis, two dimension reduction techniques, to show improvement over baseline methods.

In [42] an i-vector-based out-of-set LID system is presented. LID is usually trained with a set of known languages that are labelled. However, in cases where a language sample was not in the training set, LID systems tend to perform very poorly. The authors of [42] present an out of set (OOS) data detection method using a combination of two models. One model was trained on target data and another model was trained with unlabeled data for OOS detection. The Kolmogorov-Smirnov test is used to select candidates for OOS from unlabelled data set. Every test sample is then fed into both models and per-class outlier score is determined. This outlier score determines the confidence of the current i-vector not belonging to any of the trained data. Three OOS detection methods are used (1) One-Class SVM (2) kNN and (3) Distance to cluster centroid.

3.2.3 Music Classification

Music is an integral part of the human experience. Digital music has now almost completely replaced all other forms of music; and therefore, technology for efficiently retrieving digital music data is desired. Identification of music automatically from either artists or genre is an interesting application of i-vectors. Firstly, modelling the characteristics and features of the

music is essential. Different audio features are employed for this purpose; for example a Fluctuation Pattern which captures the variability in the rhythms. MFCCs show the timbrel aspect of a song and also work well in modelling the human voice. In [25] the authors develop both supervised and unsupervised classification and identification models using i-vectors for music similarity and artist classification.

For music similarity they use a block level similarity measure, which includes a set of block features including spectral pattern, delta spectral pattern, variance delta spectral pattern, correlation pattern, and spectral contrast pattern along with rhythm and timbre information and the first derivative of a cent-scaled spectrum (SD). This set of features is used with MFCCs to generate the i-vectors. A KNN distance classification algorithm is then used for music similarity measures.

For supervised artist classification they employ MFCCs with SD. They use LDA to reduce dimension of the i-vectors from 400 to 19, with 20 distinct artists to classify. Finally probabilistic linear discriminant analysis (PLDA), KNN, and discriminant analysis (DA) classifiers are used to test the system.

3.2.4 Environment Detection

Environment detection is a challenging application of i-vectors. Data is normally noisy, has less structure, is more diverse, and spans a larger frequency range. With an increasingly technological world with everyone owning smart devices an acoustic digital understanding of the environment can have many benefits. These include handicapped assistance, surveillance, and educational purposes. In [43] the authors present the argument that traditional methods like SVM and GMM are not enough to understand the very diverse nature of environment sounds. Therefore, they use deep learning architectures including deep neural networks (DNN), recurrent neural networks (RNN) and convolutional deep neural networks (CNN).

As discussed before i-vectors are generated using utterance level features. In [43]

the authors use a combination of features including Mel-frequency cepstral coefficients (MFCC), log Mel-Spectrum, pitch, energy, zero-crossing rate, mean crossing rate etc. A database from the DCASE challenge with fifteen unique locations is used to test the system. Their findings show that DNN, RNN and CNN perform competitively with i-vectors, though no system among GMM, i-vectors, DNN, RNN or CNN consistently performs better over all fifteen scenarios. A fusion of temporal specialized models (CNNs, RNNs) with resolution specialized models (DNNs,i-vectors) are shown to improve the accuracy significantly.

3.2.5 Concluding Remarks

Anomalous Sound Detection is shown to be primarily outlier detection techniques. This is because ADS databases often have more normal sounds as opposed to anomalous sound. Even with anomalous sounds available the anomalous data is either scarce or difficult to analyze, which is why outlier techniques work well. However supervised classification is also used in some cases and can provide interesting insights into the type of outliers or anomalous data present.

I-vectors are a very powerful tool for use in speaker recognition in both supervised and unsupervised domains. In our case we aim to use i-vectors to differentiate between 'normal' and 'anomalous' sound samples within one toy using both supervised and outlier-detection techniques. Following the research on current methods, we use SVMs, KNNs, Discriminant Analysis and Naive Bayes as the supervised classifiers. We also use One-Class SVM as an outlier detection method. The system is discussed in Chapter 4, and results in Chapter 5.

CHAPTER 4

PROPOSED SYSTEM AND SOLUTION

Our system is made up of five steps. (1) Extraction of PLP features (2) Training the UBM model either by case or all-inclusive (3) Generating i-vectors (4) Normalization and (5) Classification, supervised and unsupervised. PLP features are extracted using rastamat toolbox [44] and UBM model and i-vectors generated using MSR Identity Toolbox [45]. Classification is done in both supervised and unsupervised ways. For supervised classification KNN-Cosine distance, Discriminant Analysis, Naive Bayes and SVM classifiers are used. For unsupervised outlier-detection a one-class SVM is used. An overview of the process is shown in Figure (4.1)

4.1 Database

The database ToyADMOS released by Koizumi et al. [6] is being used to test our system. ToyADMOS provides a decent amount of anomalous samples with diversity and variety. Operational sounds of the toys are recorded in both their normal state and with intentional defects. The database has three toys, ToyCar, ToyConveyor, and ToyTrain. Every toy is then further divided into 'cases' with unique mechanical components like gears, motors and pulleys, etc. Within each case there is a further sub division of normal to anomalous sounds. Anomalous sounds have a variety of defects for each case.

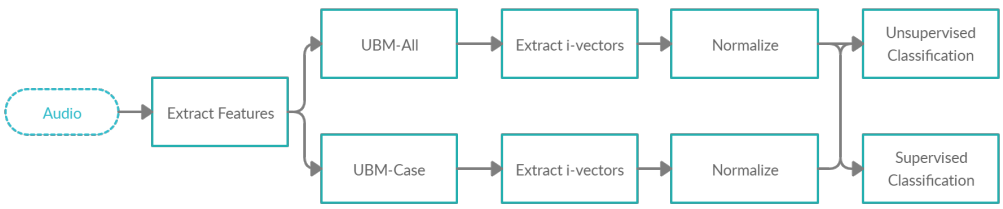

Figure 4.1: Proposed System Flow Chart

Four cases are provided for both ToyCar and ToyTrain, while three cases are given for ToyConveyor. There are two types of files, IND and CNT. We will be using IND files which include the entire duration from starting and stopping a toy, and are each 10-11 seconds long. There are 1350 normal sounds and around 260 anomalous sounds for each case for both ToyCar and ToyTrain. There are 1800 normal sounds and about 400 anomalous sounds for ToyConveyor per case. A more detailed description of the database is provided in Appendix A.

Following the authors experiment in [6], only channel 1 is taken for each ToyCar and ToyConveyor recording. For ToyTrain, since it is moving, all channels have been combined. The audio is then mixed with environment sound provided with the database. A random chunk of 11 seconds of environmental sound is extracted and mixed with the target sound. A 10dB boost is given to noise samples for ToyCar and ToyConveyor, and to target sound in ToyTrain. This is also following the authors experiment to balance the noise-signal ratio. The audio files are then down-sampled to 16kHz before being fed into feature extractor.

4.2 Feature extraction

Perceptual Linear Prediction (PLP) features were used to train UBM and extract i-vectors. It has been used in speaker recognition [31], audio segmentation and clustering [46] and music classification [47]. We take 12th order PLP features, their deltas and double deltas and append them in a super-vector.

A remark on MFCCs

We also tested Mel-frequency cepstral coefficients (MFCCs) for our utterance level features in training the UBM and extracting i-vectors. However we realized that MFCCs while working well for stationary toys, Car and Conveyor, did not work for non-stationary toy, Train. Analyzing models made through Guassian Mixtures there was little to no differ-

ence between structures of abnormal and normal data from Train made through MFCCs, therefore it was discarded.

4.3 UBM training and i-vector generation

Each toy and its subsequent case is treated as a separate speaker. This gives eight speakers each for ToyCar and ToyTrain, and six speakers for ToyConveyor. For each speaker, every 11 second IND file is taken as a separate channel. For each speaker-channel combination there is one PLP representation vector. The UBM is trained on this aggregated PLP super-vector for each toy separately.

We employ two methods of training UBM. All-UBM and Case-UBM. In All-UBM, the UBM is trained on training data comprising of all cases within one toy. Then it is used to extract i-vectors for all cases within that toy. In Case-UBM, the UBM is trained separately on each case and i-vectors are extracted for that case specifically. In All-UBM about twenty hours of data is taken for training the UBM for each toy. ToyCar and ToyTrain have about five hours of data per case, while ToyConveyor have about six hours of data per case. In Case-UBM about five hours of data is used for training the UBM for ToyCar and ToyTrain, and six hours of data is used for ToyConveyor. As only one case is used to train the UBM, only one case is used for training the BW stats, T-space and i-vector generation. A similar amount of data is used to train the BW Stats, T-Space and subsequent steps leading to i-vector generation. For All-UBM, even though all cases are used to extract the i-vectors, for further tests of classification, each case i-vectors are extracted separately for testing. Afterwards they may be fed to the classifiers independently or combined into one all inclusive set. Figure (4.3) shows the process of UBM training and extracting the i-vectors graphically.

Total Variability Space (TVS) is trained with 250 dimensions, which produces 250 dimension i-vectors.

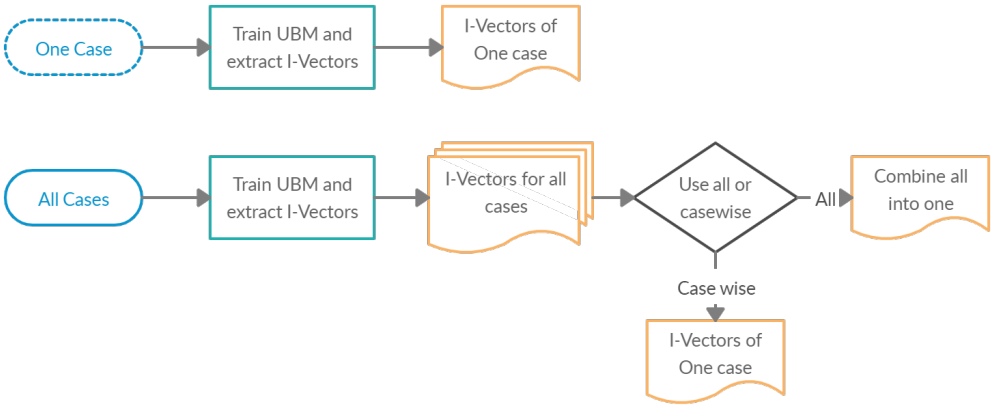

Figure 4.2: UBM Training Methods

Training-Testing Split

An 80/20 split is taken for training the UBM and generating training/testing i-vectors. Random 80 % data is taken for training making sure to maintain ratio of the normal/anomalous samples. This process is repeated 3 times with individual random data and the whole process up to classification is also performed 3 times. In end the classification results are averaged over the 3 trials.

4.4 Supervised classification

Different classifiers are used to classify between normal and abnormal sample points. (1) K-Nearest Neighbor (KNN), (2) Discriminant Analysis (DA), (3) Naive Bayesian Classifier (NBC), and (4) SVM Quadratic. Classification is done on either case wise, that is classifying normal and anomalous within one case only, or with all cases combined. Case wise can be done in two ways, either through a UBM trained on that case specifically or with a UBM trained on all cases.

KNN-Cosine

Cosine distance (equation 4.1) has been successfully used with i-vectors in [7] to calculate the similarity between two vectors. Therefore we use KNN-Cosine for supervised classification purposes for one of our test conditions.

$$k(w_1, w_2) = \frac{w_1^t . w_2}{||w_1||||w_2||} \tag{4.1}$$

Discriminant Analysis

i-vectors are assumed to have a normal distribution of $N(0, 1)$. This makes Discriminant Analysis, which assumes classes have Gaussian distributions,a very suitable method for our purposes. This is the second classifier we tested.

Naive Bayes Classifier

Naive Bayes Classifier are a collection of classifiers based on Bayes Theorem (equation 4.2). Bayes Theorem finds the probability of an event or outcome, given another event or evidence. In equation 4.2 let y be the outcome or label, and X the feature set. $P(X/y)$ gives the probability of a set of features given a specific class label. $P(y)$ is the probability of a specific class and $P(X)$ is probability of a specific feature set.

$$P(y/X) = \frac{P(X/y)P(y)}{P(X)} \tag{4.2}$$

The 'Naive' part of Naive Bayesian Classifier assumes all features are independent of each other. This makes solving the equation 4.2 much easier, and it can be reduced to equation 4.3.

$$P(y/x_1, x_2...x_n) = \frac{P(y) \prod P(x_i/y)}{P(x_1)P(x_2)..P(x_n)} \tag{4.3}$$

Removing the constant denominator and converting to a classifier model, we get

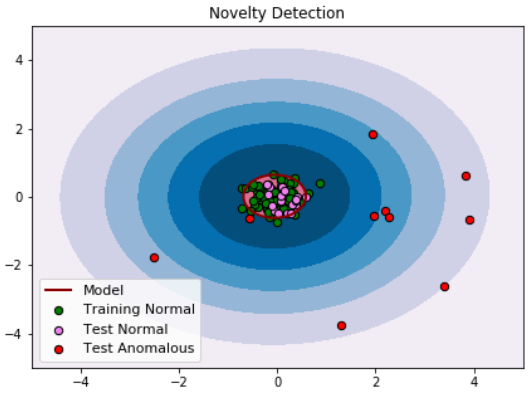

Figure 4.3: One Class SVM

$$y = argmax_y P(y) \prod P(x_i/y) \tag{4.4}$$

Which is the definition of Naive Bayes Classifier. It has also been shown to work well with i-vectors [40]. This is the third classifier we tested.

Support Vector Machine

A Support Vector Machine (SVM) is a binary classifier, which is defined by a separating hyperplane. With labeled training data, the algorithm finds the best hyperplane to separate the two classes. Our two classes are 'normal' and 'anomalous'. An SVM with a quadratic kernel worked best with our data overall so this was used as the fourth classifier. SVMs have been used extensively with both i-vectors e.g. in [41, 48, 49] and ADS cases [1].

4.5 Unsupervised Classification

For unsupervised or outlier detection we have used the one-class SVM. The one-class SVM is an unsupervised algorithm which is tuned to learn novelty or outlier detection. It is trained on the normal data and classifies every test sample as belonging to the trained data or an outlier to the trained data. Figure (4.4) shows a one-class SVM trained on normal data.

Using the training i-vectors, only the normal i-vectors are used to train the model. This process is repeated 3 times using 3 different random training i-vectors generated through UBM training. The testing data normal and anomalous samples are used to test the model. The f1-score is tuned to pick the labels of anomalous as the target.

CHAPTER 5
RESULTS AND DISCUSSION

In this section performance of i-vectors classification for both supervised and unsupervised cases is discussed under both types of UBM training. We also compare results between unbalanced and balanced data in supervised classification.

5.1 Baseline

For our baseline we compare our results to the experiments of the author in [6]. A total of 1000 random samples were picked from normal samples of each toy and each case. The rest of normal samples and anomalous samples were used to test. This gave around 350 normal samples and 260 anomalous test samples for each ToyCar and ToyTrain, and around 800 normal samples and 400 anomalous test samples for ToyConveyor.

An unsupervised auto encoder structure was used, and trained with random 1000 samples of each case. One case was tested at a time. The encoder and decoder had one fully connected neural network (FCN) layer. Four hidden FCN layers with 512 hidden units were used with RelU connection. The encoder output had 128 dimensions. The reconstructed output from the decoder was compared to the input and a reconstruction error was calculated. If this error exceeded a threshold for even one frame, the whole audio input was considered anomalous. In the paper the authors had given result for area under ROC curve, for case 1 of each toy. We have used their provided code to get results for other cases. In Table 5.1 and Table 5.2, the results of the authors auto encoder structure are detailed.

Table 5.1: Baseline Results: AUC

	ToyCar	**ToyConveyor**	**ToyTrain**
Case 1	87.4	98.1	84.3
Case 2	94.6	96.2	84.7
Case 3	86.4	98.2	62.6
Case 4	97.5		66.3

Table 5.2: Baseline Results: F-Measure

	ToyCar	**ToyConveyor**	**ToyTrain**
Case 1	83.8	89.4	75.2
Case 2	90.7	86.5	73.4
Case 3	77.8	89.4	15.7
Case 4	90.7		27.1

5.2 Performance Metric

An F1 score is calculated as the performance metric. This is preferred over the plain accuracy metric or area under curve of the ROC because of the mismatch of normal to abnormal class size. Normal samples are on average 5-6 times more numerous as compared to abnormal samples.

For unsupervised, one-class SVM area under curve is also not a suitable measure so F1 score is used here as well. For unsupervised tests 80% random values from normal data are taken to train the model and the model is tested with the remaining normal samples and all anomalous samples. This process is repeated three times and the output F1 score averaged.

Difference to baseline

The baseline results are provided as area under curve (AUC) of ROC curve and F-measure score at 10% false positive rate. Because of our limitations with difference in sample sizes

of test, and also the classification technique of one-class SVM, we have only provided F1 score (or F-Measure). F1 score is a specific point on the ROC Curve. For F1 score to be high both precision and recall need to be high. On average the F1 score is a more strict parameter of performance as compared to AUC. To maintain comparability we will discuss AUC scores but will directly compare our F1 score to baseline F1 score separately.

5.3 UBM - All

In this section UBM trained on all cases from one toy are noted which makes around 20 hours of data per toy.

5.3.1 Supervised Results

Results of F1 Score from KNN, Bayes, DA and SVM for both unbalanced and balanced i-vectors are noted. Table 5.3, 5.4 and 5.5 give results for ToyCar, ToyConveyor and ToyTrain respectively. For every test we have results from each case independently, their average and also a test for all cases. In 'All' normal values are compiled into one normal data set and all anomalous samples for one anomalous data set.

Table 5.3: Toy Car Supervised Results UBM-All

Classification	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Avg**	**All**
KNN Cosine	0.1	21.3	38.3	65.3	31.2	0.0
Discriminant Analysis	71.2	72.7	88.7	90.6	80.8	0.0
Bayes	96.8	99.3	97.8	100	98.4	0.0
SVM	95.1	98.7	97.3	99.7	97.7	0.0
Baseline FMeasure	83.8	90.7	77.8	90.7	85.6	
Baseline AUC	87.4	94.6	86.4	97.5	91.4	

Table 5.4: Toy Conveyor Supervised Results UBM-All

Classification	Case 1	Case 2	Case 3	Avg	All
KNN Cosine	100	93.4	99.5	97.6	97.4
Discriminant Analysis	98.3	90.7	96.3	95.1	85.9
Bayes	100	98.3	100	99.4	97.2
SVM	100	99.2	99.7	99.6	98.7
Baseline FMeasure	89.4	86.5	89.4	88.4	
Baseline AUC	98.1	96.2	98.2	97.5	

Table 5.5: Toy Train Supervised Results UBM-All

Classification	Case 1	Case 2	Case 3	Case 4	Avg	All
KNN Cosine	51.3	8.8	24.6	16.6	25.3	0.0
Discriminant Analysis	78.9	67.6	41.9	44.9	58.3	6.3
Bayes	95.7	90.9	96.2	95.8	94.4	1.9
SVM	88.8	81.0	90.9	91.7	88.1	1.4
Baseline FMeasure	75.2	73.4	15.7	27.1	47.8	
Baseline AUC	84.3	84.7	62.6	66.3	74.5	

5.3.2 Unsupervised One-class SVM Results

Tables 5.6 - 5.8 show results from one-class SVM for all three toys. Values of parameters gamma and nu are swept over ranges [0.01,10] and [0.01,0.5] and the set with best results is chosen for each test.

Table 5.6: Toy Car One-class SVM Results UBM-All

Method	Case 1	Case 2	Case 3	Case 4	Avg	All
One-Class SVM i-vector	83.3	96.6	97.3	99.7	94.2	32.3
Baseline FMeasure	83.8	90.7	77.8	90.7	85.6	
Baseline AUC	87.4	94.6	86.4	97.5	91.4	

Table 5.7: Toy Conveyor One-class SVM Results UBM-All

Method	Case 1	Case 2	Case 3	Avg	All
One-Class SVM i-vector	99.7	80.6	99.2	93.1	91.6
Baseline FMeasure	89.4	86.5	89.4	88.4	
Baseline AUC	98.1	96.2	98.2	97.5	

Table 5.8: Toy Train One-class SVM Results UBM-All

Method	Case 1	Case 2	Case 3	Case 4	Avg	All
One-Class SVM i-vector	80.4	57.7	59.2	57.26	63.6	25.7
Baseline FMeasure	75.2	73.4	15.7	27.1	47.8	
Baseline AUC	84.3	84.7	62.6	66.3	74.5	

5.3.3 Discussion UBM-All Results

The difference, improvement or depreciation in results, are shown in Tables 5.9 and 5.10. Car and Train show very poor results when all cases have been combined into one. This may be due to high differences between the mechanical components within cases for these two toys. Overall the supervised results are good and consistent for individual cases, with Naive Bayes and SVM showing the best results. In Unsupervised one case of Car and one case of Train do not perform better than baseline. However, over the other cases the improvement is high.

Table 5.9: UBM-All: Improvement of Supervised Best Result relative to Baseline F-Measure

	ToyCar	ToyConveyor	ToyTrain
Case 1	13.0	10.6	20.5
Case 2	8.6	12.7	17.5
Case 3	20.0	10.6	80.5
Case 4	9.3		68.7
Average	**11.3**	**11.6**	**46.8**

Table 5.10: UBM-All: Improvement of Unsupervised One-class SVM relative to Baseline F-Measure

	ToyCar	ToyConveyor	ToyTrain
Case 1	-0.5	10.3	5.2
Case 2	5.9	-5.8	-15.7
Case 3	19.5	9.8	43.5
Case 4	9.0		30.1
Average	**8.5**	**4.7**	**15.8**

5.4 UBM -Case

In this experiment, the UBM is trained on one case at a time, and the same case is used for extracting i-vectors. As before, results for both balanced and unbalanced data are provided for supervised results.

5.4.1 Supervised Results

Results of F1 Score from KNN, Bayes, DA and SVM for both unbalanced and balanced i-vectors for UBM-Case are given in Table 5.11, 5.12 and 5.13. For every test we have results from each case and their average.

Table 5.11: Toy Car Supervised Results UBM-Case

Classification	Case 1	Case 2	Case 3	Case 4	Avg
KNN Cosine	88.9	12.4	97.9	99.3	74.6
Discriminant Analysis	88.7	70.3	98.6	99.3	89.2
Bayes	96.6	94.5	98.9	99.6	97.4
SVM	95.9	92.1	98.7	100	96.7
Baseline FMeasure	83.8	90.7	77.8	90.7	85.6
Baseline AUC	87.4	94.6	86.4	97.5	91.4

Table 5.12: Toy Conveyor Supervised Results UBM-Case

Classification	Case 1	Case 2	Case 3	Avg
KNN Cosine	99.5	95.7	99.7	98.3
Discriminant Analysis	98.5	87.3	98.5	94.7
Bayes	100	97.8	100.0	99.3
SVM	100	99.1	99.7	99.6
Baseline FMeasure	89.4	86.5	89.4	88.4
Baseline AUC	98.1	96.2	98.2	97.5

Table 5.13: Toy Train Supervised Results UBM-Case

Classification	Case 1	Case 2	Case 3	Case 4	Avg
KNN Cosine	29.5	3.9	12.4	7.6	13.4
Discriminant Analysis	84.9	70.7	70.3	66.7	73.1
Bayes	97.7	93.7	94.7	98.9	96.3
SVM	91.4	85.9	92.1	90.0	89.9
Baseline FMeasure	75.2	73.4	15.7	27.1	47.8
Baseline AUC	84.3	84.7	62.6	66.3	74.5

5.4.2 Unsupervised One-class SVM Results

Same as before, a one-class SVM is trained on 80% of normal sounds randomly selected. The remaining normal sounds and anomalous sounds are used for anomaly or novelty detection. Results of this experiment with UBM-Case are listed in Tables 5.14-5.16.

Table 5.14: Toy Car One-class SVM Results UBM-Case

Method	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Avg**
One-Class SVM i-vector	96.3	97.6	98.0	99.2	97.8
Baseline FMeasure	83.8	90.7	77.8	90.7	85.6
Baseline AUC	87.4	94.6	86.4	97.5	91.4

Table 5.15: Toy Conveyor One-class SVM Results UBM-Case

Method	**Case 1**	**Case 2**	**Case 3**	**Avg**
One-Class SVM i-vector	99.7	96.4	99.3	98.5
Baseline FMeasure	89.4	86.5	89.4	88.4
Baseline AUC	98.1	96.2	98.2	97.5

Table 5.16: Toy Train One-class SVM Results UBM-Case

Method	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Avg**
One-Class SVM i-vector	90.1	77.1	62.8	77.2	76.8
Baseline FMeasure	75.2	73.4	15.7	27.1	47.8
Baseline AUC	84.3	84.7	62.6	66.3	74.5

5.4.3 Discussion UBM-Case Results

The comparison to baseline, the improvement or depreciation, is shown in Tables 5.17 - 5.18. Compared the UBM trained on all cases we see a significant improvement through out the results, specifically for the unsupervised scenario.

Table 5.17: UBM-Case: Improvement of Supervised Best Result relative to Baseline F-Measure

	ToyCar	ToyConveyor	ToyTrain
Case 1	12.8	10.6	22.5
Case 2	3.8	11.2	20.2
Case 3	21.1	10.5	79.0
Case 4	8.9		71.7
Average	**11.7**	**10.8**	**48.3**

Table 5.18: UBM-Case: Improvement of Unsupervised One-class SVM relative to Baseline F-Measure

	ToyCar	ToyConveyor	ToyTrain
Case 1	12.5	10.6	14.8
Case 2	6.9	12.5	3.7
Case 3	20.2	10.2	47.1
Case 4	8.5		50.0
Average	**12.0**	**11.1**	**28.9**

5.5 Concluding Remarks

Over all results show that i-vectors are a good method of capturing the features for machine sounds. For supervised results we see good and improved results for all of the toys. With supervised not only do we see a significant improvement over the baseline results, but we also see very high accuracy results in themselves. Unsupervised results have also shown improvement over the baseline results. UBM trained on a case wise basis performs better than UBM trained on all cases.

The experiments point to i-vectors being a viable and dependable method for working with machine sounds anomalous sound detection, in both supervised and unsupervised scenarios.

CHAPTER 6
CONCLUSION

In this research project we investigated the use of i-vectors for classification in ADS use cases, which to the best of our knowledge has not been investigated before. We have extensively researched methods which bridge the Anomalous Detection of Sounds (ADS) in machines and popular classification methods used with i-vectors. We report our results using different training techniques of Universal Background Model (UBM) for an in-depth analysis. We have used the database ToyADMOS for our research work because it provides a generous variety and number of anomalous samples to develop and test models on.

We have shown results for both supervised and unsupervised applications of ADS. For each test in supervised we have used K-Nearest Neighbours, Naive Bayes, Discriminant Analysis and Quadratic SVMs as classifiers. For unsupervised we have used one-class SVM which is trained on normal samples only. All of these tests are done for UBM trained two ways, i.e UBM-All and UBM-Case. UBM-All is a UBM trained on all data of one toy, while UBM-Case is trained on one case of a toy.

Our results show that i-vectors are a good choice for using with Machine sounds. We have achieved on average achieved high, robust and consistent results for both supervised and unsupervised methods of classifying.

Further work on this project can include additional utterance features to be added alongside PLP. We had reviewed a number of features like MFCCs, spectral centroid, zero crossing rate and short time energy, and there is potential to incorporate them to improve the models.

Appendices

APPENDIX A
DESCRIPTION OF DATA BASE

The ToyADMOS dataset [6] consists of three types of datasets for three different tasks, with a different toy. Normal and anomalous sounds are collected for each. Normal sounds are where the toy operates according to specifications, while anomalous sounds are when the target machine is made to operate anomalously by adding extraneous objects or introducing defects.

Toy Car

Intended for product-inspection tasks. A toy car called 'mini 4WD' is used, which is driven by a small motor and gears/shafts. The toy car moves on an inspection device. Sound data is collected with four microphones set close to the inspect device. Each further case of toy car is designed with a combination of two types of motors and bearings, giving a total of four cases. Each IND normal and anomalous sound is 11 seconds long. There are 1350 normal samples and around 250 anomalous samples for each case. Anomalous sounds were produced by damaging the shaft, gears, tires, and voltage.

Toy Conveyor

Intended for fault diagnosis tasks in a stationary machine. A toy conveyor is fixed on a desk, and is used to transport a mini tin toy. Sound is again collected with four microphones, with one attached to body of conveyor and rest on the table. Three different sizes of conveyors produced by same manufacturer are used to create three cases of Toy Conveyor. There are 1800 normal IND and about 350 anomalous samples, each 10 seconds long, per case. Anomalous sounds were produced by damaging the tension pulley, trail pulley, and belt and changing the voltage

Table A.1: Types of Defects in Toys, taken from [6]

ToyCar		ToyConveyor		ToyTrain	
Parts	**Anomaly**	**Parts**	**Anomaly**	**Parts**	**Anomaly**
Shaft	Bent	Tension Pully	Excessive tension	First Carriage	Chipped wheel axle
Gears	Deformed,Melted	Tail Pully	Excessive tension, Removed	Last Carriage	Chipped wheel axle
Tires	Coiled, plastic/steel ribbon	Belt	Attached three metal objects	Straight railway track	Broke, Obstruction, Disjointed
Voltage	Over/Under	Voltage	Over/Under	Curved railway track	Broke, Obstruction, Disjointed

Toy Train

Intended for fault diagnosis tasks in a non-stationary machine. A toy train operates on a railway track. Sound data is collected with four microphones. Each case of toy train is through a combination of two types of trains (commuter and a bullet) and two types of scales (HO-scale and N-scale). Giving a total of four cases. Each case has 1350 normal samples and 270 anomalous samples. Anomalous sounds were produced by damaging the first/last carriage and straight/curved railway track

Environmental Sounds

Environmental noise is provided with the database. It is intended to simulate a factory environment and included noise samples collected at real factory locations. These include collisions, drilling, pumping and airbrushing.

REFERENCES

[1] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Audio surveillance of roads: A system for detecting anomalous sounds," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 1, pp. 279–288, 2015.

[2] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, IEEE, 2007, pp. 21–26.

[3] Y. Koizumi, S. Saito, H. Uematsu, and N. Harada, "Optimizing acoustic feature extractor for anomalous sound detection based on Neyman-Pearson lemma," in *2017 25th European Signal Processing Conference (EUSIPCO)*, IEEE, 2017, pp. 698–702.

[4] Y. Kawaguchi, R. Tanabe, T. Endo, K. Ichige, and K. Hamada, "Anomaly detection based on an ensemble of dereverberation and anomalous sound extraction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 865–869.

[5] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, "Unsupervised detection of anomalous sound based on deep learning and the Neyman–Pearson lemma," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 212–224, 2018.

[6] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, IEEE, 2019, pp. 313–317.

[7] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[8] A. DeMarco and S. J. Cox, "Native accent classification via i-vectors and speaker compensation fusion.," in *Interspeech*, 2013, pp. 1472–1476.

[9] J. Grzybowska and S. Kacprzak, "Speaker Age Classification and Regression using i-vectors.," in *INTERSPEECH*, 2016, pp. 1402–1406.

[10] I.-A. Bănică, H. Cucu, A. Buzo, D. Burileanu, and C. Burileanu, "Automatic methods for infant cry classification," in *2016 International Conference on Communications (COMM)*, IEEE, 2016, pp. 51–54.

[11] M. Dorfer, B. Lehner, H. Eghbal-zadeh, H. Christop, P. Fabian, and W. Gerhard, "Acoustic scene classification with fully convolutional neural networks and I-vectors," *Proceedings of the Detection and Classification of Acoustic Scenes and Events*, 2018.

[12] B. Lehner, H. Eghbal-Zadeh, M. Dorfer, F. Korzeniowski, K. Koutini, and G. Widmer, "Classifying short acoustic scenes with i-vectors and CNNs: Challenges and optimisations for the 2017 DCASE ASC task," *DCASE2017 Challenge*, 2017.

[13] Y. Ono, Y. Onishi, T. Koshinaka, S. Takata, and O. Hoshuyama, "Anomaly detection of motors with feature emphasis using only normal sounds," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 2800–2804.

[14] C. Clavel, T. Ehrette, and G. Richard, "Events detection for an audio-based surveillance system," in *2005 IEEE International Conference on Multimedia and Expo*, IEEE, 2005, pp. 1306–1309.

[15] P Coucke, B. De Ketelaere, and J. De Baerdemaeker, "Experimental analysis of the dynamic, mechanical behaviour of a chicken egg," *Journal of sound and vibration*, vol. 266, no. 3, pp. 711–721, 2003.

[16] W. Gutierrez, S Kim, D. Kim, S. Yeon, and H. Chang, "Classification of porcine wasting diseases using sound analysis," *Asian-Australasian Journal of Animal Sciences*, vol. 23, no. 8, pp. 1096–1104, 2010.

[17] P. Grabill, T. Brotherton, B. Branhof, J. Berry, and L. Grant, "Rotor smoothing and vibration monitoring results for the US army VMEP," INTELLIGENT AUTOMATION CORP POWAY CA, Tech. Rep., 2009.

[18] P Piyush, R. Rajan, L. Mary, and B. I. Koshy, "Vehicle detection and classification using audio-visual cues," in *2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, 2016, pp. 726–730.

[19] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2000.

[20] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[21] D. L. Iverson, "Inductive system health monitoring," 2004.

[22] S. E. Guttormsson, R. Marks, M. El-Sharkawi, and I Kerszenbaum, "Elliptical novelty grouping for on-line short-turn detection of excited running rotors," *IEEE Transactions on Energy Conversion*, vol. 14, no. 1, pp. 16–22, 1999.

[23] A. Rabaoui, M. Davy, S. Rossignol, and N. Ellouze, "Using one-class SVMs and wavelets for audio surveillance," *IEEE Transactions on information forensics and security*, vol. 3, no. 4, pp. 763–775, 2008.

[24] J. Portelo, M. Bugalho, I. Trancoso, J. Neto, A. Abad, and A. Serralheiro, "Non-speech audio event detection," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2009, pp. 1973–1976.

[25] H. Eghbal-Zadeh, B. Lehner, M. Schedl, and G. Widmer, "I-vectors for Timbre-Based Music Similarity and Music Artist Classification.," in *ISMIR*, 2015, pp. 554–560.

[26] Y. E. Kim and B. Whitman, "Singer identification in popular music recordings using voice coding features," in *Proceedings of the 3rd international conference on music information retrieval*, vol. 13, 2002, p. 17.

[27] E. Pampalk, "Audio-based music similarity and retrieval: Combining a spectral similarity model with information extracted from fluctuation patterns," in *Proceedings of the International Symposium on Music Information Retrieval*, 2006.

[28] A. F. Geib, C. C. Kuo, M. Gawecki, E. Tsau, J. W. Kang, and P. R. Scheid, *MFCC and CELP to detect turbine engine faults*, US Patent 8,655,571, 2014.

[29] A Głowacz and Z Głowacz, "Diagnostics of DC machine based on analysis of acoustic signals with application of MFCC and classifier based on words," *Archives of metallurgy and materials*, vol. 57, no. 1, pp. 179–183, 2012.

[30] P Mayorga, C Druzgalski, R. Morelos, O. Gonzalez, and J Vidales, "Acoustics based assessment of respiratory diseases using GMM classification," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, IEEE, 2010, pp. 6312–6316.

[31] Y.-M. Zeng, Z.-Y. Wu, T. Falk, and W.-Y. Chan, "Robust gmm based gender classification using pitch and rasta-plp parameters of speech," in *2006 International Conference on Machine Learning and Cybernetics*, IEEE, 2006, pp. 3376–3379.

[32] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[33] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.

[34] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, "Fisher discriminant analysis with kernels," in *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*, Ieee, 1999, pp. 41–48.

[35] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[36] M. Seck, F. Bimbot, D. Zugaj, and B. Delyon, "Two-class signal segmentation for speech/music detection in audio tracks," in *Sixth European Conference on Speech Communication and Technology*, 1999.

[37] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2015, pp. 1996–2000.

[38] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, no. 1, 2015.

[39] Y. Kawachi, Y. Koizumi, and N. Harada, "Complementary set variational autoencoder for supervised anomaly detection," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 2366–2370.

[40] M. H. Bahari, R. Saeidi, D. Van Leeuwen, *et al.*, "Accent recognition using i-vector, gaussian mean supervector and gaussian posterior probability supervector for spontaneous telephone speech," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 7344–7348.

[41] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Twelfth annual conference of the international speech communication association*, 2011.

[42] H. Behravan, T. Kinnunen, and V. Hautamäki, "Out-of-set i-vector selection for open-set language identification.," in *Odyssey*, vol. 2016, 2016, pp. 303–310.

[43] J. Li, W. Dai, F. Metze, S. Qu, and S. Das, "A comparison of deep learning methods for environmental sound detection," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, pp. 126–130.

[44] D. P. W. Ellis, *PLP and RASTA (and MFCC, and inversion) in Matlab*, online web resource, 2005.

[45] S. O. Sadjadi, M. Slaney, and L. Heck, *MSR identity toolbox*, 2013.

[46] H. Meinedo and J. Neto, "Audio segmentation, classification and clustering in a broadcast news task," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, IEEE, vol. 2, 2003, pp. II–5.

[47] R Thiruvengatanadhan, "Speech/music classification using plp and svm,"

[48] A. Kanagasundaram, R. Vogt, D. B. Dean, S. Sridharan, and M. W. Mason, "I-vector based speaker recognition on short utterances," in *Proceedings of the 12th Annual Conference of the International Speech Communication Association*, International Speech Communication Association (ISCA), 2011, pp. 2341–2344.

[49] M. McLaren and D. Van Leeuwen, "Source-normalized LDA for robust speaker recognition using i-vectors from multiple speech sources," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 755–766, 2011.