ENGN3712 Research Methods

The Relative Binaural Transfer Function as a Spatial Feature of Sound for Improving the Accuracy of Acoustic Emotion Recognition through a KNN, MLP and CNN Model

Punjaya Wickramasinghe
u6310965

Supervisors: Prasanga Samarasinghe & Manish Kumar
Semester 1 2021

Australian
National
University

Table of Contents

Acknowledgements

First and foremost, I would like to thank my supervisors Manish and Prasanga. Without their support, none of this would have been possible. Prasanga for having the faith in me to take this project head-on and for being my supervisor for another semester, I really appreciated it because I thoroughly enjoyed the subject and a newfound passion for acoustic signals processing and machine learning thanks to you. And to Manish, for the many hours spent helping me understand concepts, showing me how to use Qualtrics, and giving me the peace of mind to know I have someone to get immediate help from if I need it - I am so grateful for your help.

And to my family, thank you for your unconditional support. To my friends, thank you for being so understanding about my commitment to the project. And to my girlfriend, thank you for your support and for helping me narrow down the list of emotions, for finding so many sources of audio files for me on YouTube, and for helping me trial the listening tests. I really appreciate every single one of you.

Finally, to the participants of the experiment who completed the surveys, a lot of whom were my friends. From those who spent over an hour completing all 4 surveys, to those who even just completed 1. It would not have been possible without your time and contribution.

Abstract

This study explores the ability of spatial features of sound, namely the Relative Binaural Transfer Function (ReBTF), in potentially improving acoustic emotion recognition (AER). The features' ability to do so will be evaluated with the use of various machine learning algorithms including a single traditional ML algorithm - K-Nearest Neighbours (KNN) - and two deep learning algorithms - the Multi-Layer Perceptron (MLL) and Convolutional Neural Network (CNN). The audio features used for the study included the Zero-Crossing Rate, Root Mean Square Energy, Spectral Centroid, Spectral Rolloff, Spectral Flatness, Tonnetz features, Mel-Frequency Cepstral Coefficients and the Mel Spectrogram. The emotion classes that were chosen to be classified included *happy*, *sad*, *neutral*, *calm*, *curious and stress.* There were 30 participants in the study, sampled through convenience sampling, and the set of audio samples consisted of 200 ten-second audio clips which were manually sampled from YouTube. The KNN model showed an overall increase of 1.36% of model prediction accuracy and also showed an improvement of 3.41%, on average, for the precision of the model in predicting individual emotions, when the ReBTF was included in the set of features. On the contrary, the two deep learning models suggested that the ReBTF reduced the performance of the models by 2.12% on average. The reliability of the deep learning results are questionable, however, due to numerous limitations including the insufficient dataset size for deep learning applications and the lack of hyperparameter tuning. Ultimately, with consideration to the KNN results, the ReBTF and potentially even other spatial audio features, seem like they can contribute meaningfully towards enhancing AER. Limitations of the study were predominantly the dataset size and lack of control over the experimental conditions of listening tests, and it is suggested that future exploration addresses these limitations

Introduction

Motivation of Research

More specifically, human-machine interactions take two dominant forms: audio, and textual Communication (Rathor et al., 2021). Currently, textual communication is the more prominent of the two but is an extremely tedious process when considering the disparity in the speed of human thought, as compared to the speed of human typing. On the contrary, spoken language is our most natural mode of communication and technology that enables such communication between humans and machines is in its infancy. For example, the relatively recent introduction of voice assistants, into the commercial market and their current capabilities is a testament to that (Lanjewar et al., 2015). With the frequency of human-machine interactions increasing rapidly, improving the emotional intelligence of these modern-day voice recognition technologies is vital for more seamless human-machine interactions (Seo et al., 2019). This artificial emotional intelligence is inherently encompassed by a machines ability to recognise emotions through audio. Not only will automating the recognition of emotions through sound allow for an enhanced human-machine experience, but it also enables language and emotions to be understood more comprehensively. The emotional state of humans alters the meaning of the same linguistic content and enhancing the current state of human-machine interaction will involve enabling machines to understand how to detect these alterations (Nanavare et al., 2015).

Applications in Acoustic Emotion Recognition

Beyond voice assistants, there are a plethora of other applications for which acoustic emotion recognition (AER) can be applied. Firstly, the advancement of AER would enable chatbot services to be used more often to provide 24-hour service to customers in a variety of disciplines. For example, an emotionally intelligent chatbot could be used for psychological applications, to conduct an assessment of a patient's emotional state to make preliminary conclusions about potential mental health illnesses (Rathor et al., 2021). Chatbots can be used in a business context, to provide dynamic recommendations depending on the emotion detected in speech from a client. They can also be used to design a smart and secure automated home. Finally, in the context of music, emotion recognition can allow for superior music genre classification which is a technology that companies such as Spotify and Apple Music are in the process of implementing currently (Panda et al., 2018).

The Knowledge Gap

Evidently, the abundance of applications that would benefit from AER technology indicates a substantial demand to improve the current state of said technology. The majority of research in this space circulates around temporal and spectral audio features, and how they can be used to optimise AER and there is lacking consideration of how spatial features of sound can impact contribute to optimistic AER (El Ayadi et al., 2011; Kerkeni et al., 2019; Kim et al., 2019; Yang et al., 2012). Through this study, we propose that spatial features of sound, namely the Relative Binaural Transfer Function (ReBTF), may provide an avenue for further improving the performance of emotion classifiers in accurately recognising emotions. Therefore, the scope of this research is to assess whether the classification accuracy of various machine learning (ML) models - K-Nearest Neighbours, Multi-Layer Perceptron and Convolutional Neural Networks - may be improved with the incorporation of the ReBTF into a standard set of temporal and spectral features.

This study is an extension of a previous, preliminary investigation into the topic and aims to improve the size of the dataset, the quality of features selected, the number of participants sampled, and the ML algorithms used. The research process included data collection, data pre-processing and finally, the analysis. The data collection consisted of expanding the size of the dataset from the previous study, audio pre-processing, and then finally creating and distributing online surveys. The data pre-processing consisted of data manipulation, feature extraction and two feature selection processes - the Add On In and Boxplot Techniques. Finally, the analysis involved hyperparameter tuning, followed by training and testing the data on the KNN, MLP and CNN models.

The following structure will be used for the remainder of the report. Section 1 will consist of a literature review that introduces the relevant background for the research topic and identifies the benchmark for classification accuracy as well as the gap in the literature. Section 2 will present the experimental procedure which will outline three main stages; data collection, data pre-processing and the analysis stages. Section 3 presents a description of the results and a discussion about the key findings with considerations to limitations of the study. Finally, Section 4 outlines various avenues for future work.

1. Literature Review

1.1 Relevant Background

1.1.1 Model of Emotions

Before a computerised emotion classifier may be developed, a fundamental model for defining emotions must first be established. Given the highly ambiguous nature of emotions, there is no universally agreed-upon definition to coin the term; rather, there are a multitude of theories that seek to encapsulate the true conceptualisation of emotions (Sharar et al., 2016). These theories can broadly be categorized into two approaches for understanding emotions; the dimensional and the categorical approach. The dimensional approach defines emotions as an incredibly dense range of experience, which can be described through fundamental attributes of emotion such as the level of arousal, valence and potency (Sharar et al., 2016). On the contrary, the categorical approach suggests that there are only a handful of primary emotions that all other emotions can be sub-categorized under (Yang et al., 2012).

A large portion of literature uses the dimensional approach as this provides a more soft assignment of emotions which is ideal to account for the impact of the subjectivity of emotions; however, the present study was scoped around the categorical approach. In particular, many researchers agree that there are a total of six universal emotions; happiness, anger, sadness, surprise, disgust and fear (Ekman et al., 1969; Ekman and Friesen, 1971, 1976; Ekman, 1992; Tomkins and McCarter, 1964; Ekman and Friesen, 1971; Johnson-Laird and Oatley, 1992 cited in Matsuda et al., 2013). On top of this collection, calmness, curiosity and neutrality were added for this study. These additions were made to allow for a greater scope in the range of emotions for the present study. Since anger, fear and disgust all characterise a strong negative experience, they were placed under a single category known as 'stress'. Therefore, the emotion classes used for this study were: happy, sad, neutral, calm, curiosity and stress.

It is worth noting that using such a small range of emotions introduces the problem of participants experiencing a greater sense of ambiguity when annotating the audio samples. Therefore, the quality of the audio annotations will likely be decreased since having fewer emotions to choose from reduces the likelihood that the emotion felt by the participant is represented in the words presented to them. Nevertheless, the benefit of a small range of emotions is that it provides an inherently increased likelihood that the classifier's predictions will be more accurate.

1.1.2. Traditional Machine Learning vs Deep Learning Algorithms

Traditional Machine Learning

Machine learning (ML) is, at its core, using training data to detect patterns and create rules for making predictions about new, unseen, testing data. Traditional ML involves the extraction of features from raw data. These features are then used by the model for training. Since the algorithm's performance is highly dependent on the choice of features provided to the model, optimising the selection of features is an imperative component of traditional ML. The subsequent issue is that there is not universally agreed upon method of feature selection that can consistently allow for optimal model performance (Lech at el., 2020). Typically, this problem is dealt with through iteratively testing a large variety of distinct feature groups or by implementing particular feature selection techniques. Hence, the major limitation associated with traditional ML is the significant variation in performance that can arise through feature selection (Lech at el., 2020).

Deep learning

The rapid growth of innovation surrounding machine learning has recently given rise to the age of deep learning - a modern approach for ML that helps bypass the issues of feature selection. Deep learning (DL), unlike traditional ML, uses an end-to-end network that takes raw data as input to generate outputs. This eliminates the need for any substantial feature extraction or feature selection process, and greatly simplifies the overall process of ML (Kaur & Kumar, 2021). The cost of such a convenient ML approach is the resource intensity of DL, requiring exceptionally larger datasets than traditional ML approaches to achieve optimal performance. More specifically, DL requires sample sizes on the order of millions, as opposed to the hundreds or thousands needed for traditional ML (Lech at el., 2020). In general, DL is more suitable for complex problems - problems that traditional ML approaches struggle with such as image or audio classification.

For the purpose of the present study, both traditional ML and DL approaches will be implemented to broaden the scope of potential findings. In particular, a single traditional ML algorithm will be used - K-Nearest Neighbours (KNN) - and two DL algorithms will be used - Multi-Layer Perceptrons (MLP) and Convolutional Neural Networks (CNN).

K-Nearest Neighbours

K-Nearest Neighbours (KNN) is a supervised traditional ML algorithm that performs classification by assigning each sample to a cluster based on the clustering of the k-nearest samples (Kaur & Kumar, 2021). For this study, the measurement of distance used was the Euclidean distance. The algorithm operates under one main hyperparameter - the number of nearest neighbours.

Multi-Layer Perceptron

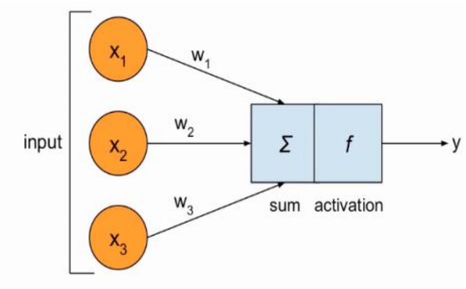


Figure 1.1 Artificial Neuron

To understand the Multi-Layer Perceptron (MLP), a brief discussion of neural networks is necessary. As the term suggests, neural networks are simply a network formed by artificial computational units known as neurons. A neuron has inputs, weights associated with each input, a summation function, an activation function and an output. In this way, the output of a neuron is related to the input through the transformation performed by the summation and activation functions (Botalb et al., 2018). A collection of neurons is referred to as a layer. The weights of each neuron are optimised during the training phase.

The Multi-Layer Perceptron is one of the most basic artificial neural network architectures. It is a forward feeding system made up of a single input layer, a variable number of hidden layers, and a single output layer (Botalb et al., 2018). It requires that the input data is flattened into a single vector.

Convolutional Neural Network

The Convolutional Neural Network (CNN) is a more advanced neural network when compared to the MLP, that is used for processing multidimensional data with a grid pattern (Yamashita et al., 2018). It adaptively learns the hierarchy of features within input data, building from recognising low to high level patterns. The hidden layers of a CNN are composed of convolution, pooling and fully connected layers, each serving its own purpose.

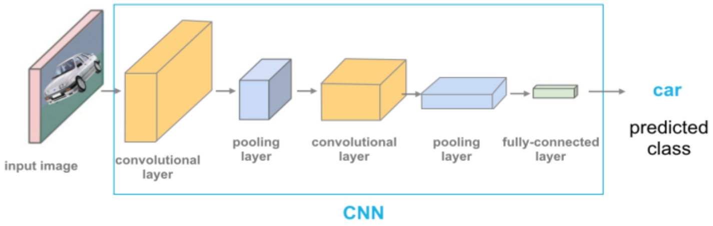


Figure 1.2: General Architecture of Convolutional Neural Network

The purpose of the convolutional layer is to extract features using element-wise multiplication between the input data and a filter (Amrutha et al., 2021). The pooling layer then uses these extracted features and reduces their dimensionality. In turn, this reduces the number of parameters and the amount of computation to be performed by the network (Yamashita et al., 2018). The fully connected layers flatten the multidimensional features from the convolution and pooling layers, and map them to an output. The benefits of CNN is that computation and overfitting is significantly reduced, and the model is more tolerant towards distortions in the data due to pooling. The main hyperparameters for a CNN are the number and size of the filters.

1.1.3. Features of Sound and the Relative Binaural Transfer Function (ReBTF)

The features of sound may be decomposed into 3 primary categories: temporal (time-related), spectral (frequency-related) and spatial features. Each category of features has differing requirements for extraction. Temporal features can be extracted directly from the signal waveform. Spectral features need to be obtained from the signal spectrogram. Spatial features require binaural audio to be extracted.

Spatial features enable humans to localise sound, to infer both the direction and distance of a sound source (Cuadrado et al., 2020). Given that spatial audio is the natural way in which we hear sound, spatial features are likely involved in inducing stronger, more raw emotions from people as compared to hearing monaural audio. As a result, this paper explores the potential contribution of spatial features towards AER (Cuadrado et al., 2020; Fletcher, 2011). In particular, the spatial feature of interest for this study is the Relative Binaural Transfer Function (ReBTF). This feature represents an approximation for the Relative Transfer Function, which describes the ratio of the transfer functions of the two channels of a binaural recording. Extracting the feature requires binaurally recorded audio - that is, audio with two distinct channels.

1.1.4 Related Work

Lanjewar et al. (2015) explored the ability of two ML algorithms - a Gaussian Mixture Model (GMM) and a KNN - to predict emotions. They used MFCCs, Wavelet features and pitch-related features and selected a total of 6 emotion classes for prediction; happy, sad, angry, neutral, fearful and surprised. They used the Berlin Database of Emotional Speech (Emo-DB) for their analysis, and ultimately achieved an overall accuracy of 66% and 51% for the GMM and KNN models respectively. They further concluded that the GMM was extremely strong in predicting anger and sad emotions (92% and 89% precision respectively) while the KNN was strong in detection of happy and angry emotions (90% and 72% respectively). From their results, they recommended that the two methods are combined using a Voting Classifier, to potentially improve the quality of predictions.

Rathor et al. (2021) aimed to maximise the prediction accuracy of distinguishing between emotional states using the following audio features: Chroma, Spectral Contract, Tonnetz and MFCC. A KNN, MLP, Support Vector Machine (SVM), Random Forest and Voting classifier was used in various combinations to classify between happy, sad, fearful and neutral speech. They used the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) database and found that the optimal prediction accuracy to be 88.09% when a combination of MLP, SVM and Random Forest classifiers were used under soft voting conditions. The recency of this study emphasises its reliability.

Soundarya & Arumugaaudio (2020) used a combination of MFCC and Linear Predictive Cepstral Coefficients (LPCC) to perform AER using a Decision Tree, Random Forest and CNN classifiers. The emotion classes they sought to classify were: happy, sad, angry, fear, surprise, neutral and disgust which is similar to Lanjewar et al. (2015). Through their investigation, they achieved an optimal model accuracy of 70.3% when using the CNN with the MFCC features.

Amrutha et al. (2021) explored emotion recognition of speech audio by using MLP and CNN models to classify speech into 8 emotion classes; happy, sad, neutral, angry, fear, calm, surprised and disgusted. Their audio data came from the RAVDESS and Surrey Audio-Visual Expressed Emotion (SAVEE) databases. The primary features used were the MFCC, Chroma vectors and Zero-Crossing Rate (ZCR). Ultimately, they developed a MLP and CNN model with an overall accuracy of 79% and 88% respectively.

Kaur & Kumar (2021) described a number of approaches for detecting emotions in speech audio, through KNN, MLP, CNN and Random Forest classifiers. The CNN was provided the audio spectrograms as inputs, whereas the other three models (KNN, MLP and Random Forest) were provided MFCCs as input. They sourced their audio samples from Emo-DB and used the following emotion classes: happy, sad, angry, neutral, disgusted, bored, and fearful. Their MLP classifier produced the best result, with overall accuracy of 90.36%.

Reference	Features extracted	Classifiers used	Results (accuracy)
Rathor et al. (2021)	MFCC, Tonnetz features, Spectral Contrast, Chroma vectors	KNN, MLP, SVM, Random Forest, Voting Classifier	MLP + SVM + Random Forest: 88%
Lanjewar et al. (2015)	MFCC, Wavelet features, Pitch	GMM, KNN	GMM: 66%, KNN: 51%
Soundarya & Arumugam. (2020)	MFCC, Linear Predictive Cepstral Coefficients (LPCCs)	Decision Tree, Random Forest, CNN	CNN with MFCC: 70.3%
Kaur & Kumar. 2021	Spectrogram, MFCC	KNN, MLP, CNN, Random Forest	MLP: 90%
Amrutha et al. (2021)	MFCC, Chroma vectors, ZCR	CNN, MLP	CNN: 86%, KNN: 79%

Table 1.1: Compassion of literature in AER.

With such a high classification accuracy of 90%, the results obtained by Kaur & Kumar stand as the benchmark for the current study. As such, many of the choices of features and the ML algorithms for the present work, were inspired by this study. That is, MFCC and spectrograms were also used in the present study, and we also implemented a KNN, MLP and CNN model similar to them. Notably, none of the presented literature used any spatial features, which further outlines the gap in the literature that is being addressed through this study.

1.2. My Research

The present work is an extension of a previous study completed in May 2021, which was a preliminary exploration into the potential impact of the ReBTF on improving acoustic emotion recognition. While it was found that the ReBTF may, in fact, play a role in improving AER, the result was acquired by an experiment that had many limitations. There was a relatively small dataset, the reliability of the audio annotations and the validity of the features used was questionable, and there was a poor selection of ML techniques applied. Hence, the primary motivation for the current study is to further validate the findings of the previous study with more reliable and valid results. Four key improvements were made to the previous study which was mostly motivated through the literature.

Firstly, the size of the dataset used in the previous study was 38 samples. When considering that this is an application in ML, 38 samples is not sufficient data to be able to adequately train a model. This sample of 38 was increased to 200 - an increase of 526%.

Secondly, the number of people annotating the audio files was previously only 3 which is an extremely small number of trials to justify accurate annotation of the audio files. Hence, the number of participants was also increased to 30 people - an increase of 1000%.

Thirdly, the temporal and spatial features used in the previous study included the Zero-Crossing Rate (ZCR), Root Mean Square Energy (RMSE), Spectral Centroid and Rolloff and the Mel-Frequency Cepstral Coefficients (MFCC). Beyond the fact that these are commonly used features in AER applications (Ramdinmawaii et al., 2017; Er et al., 2020), their selection was not backed by any further justification. Therefore, two feature selection processes are employed - the Add On In and Boxplot techniques - and based on the literature presented above (Kaur & Kumar, 2021), the Mel Spectrogram, Spectral Flatness and Tonnetz features were added to the previous selection. The final selection of features to be extracted are ZCR, RMSE, Spectral Centroid, Spectral Rolloff, Spectral Flatness, Tonnetz features, Mel Spectrogram and Mel-Frequency Cepstral Coefficients.

Finally, the problem of emotion classification is predominantly discrete in nature. Considering this, the use of linear regression in the previous study was misguided at best, since linear regression is a method of predicting outputs on a continuous scale, as opposed to a discrete one. To cater for this, the present study specifically employed classification algorithms such as KNN, MLP and CNN. The next section will present the experimental procedure of the study.

2. Experimental Procedure

The study consisted of 3 main stages: data collection, pre-processing, and analysis. The overall experimental procedure can be described through the following framework.

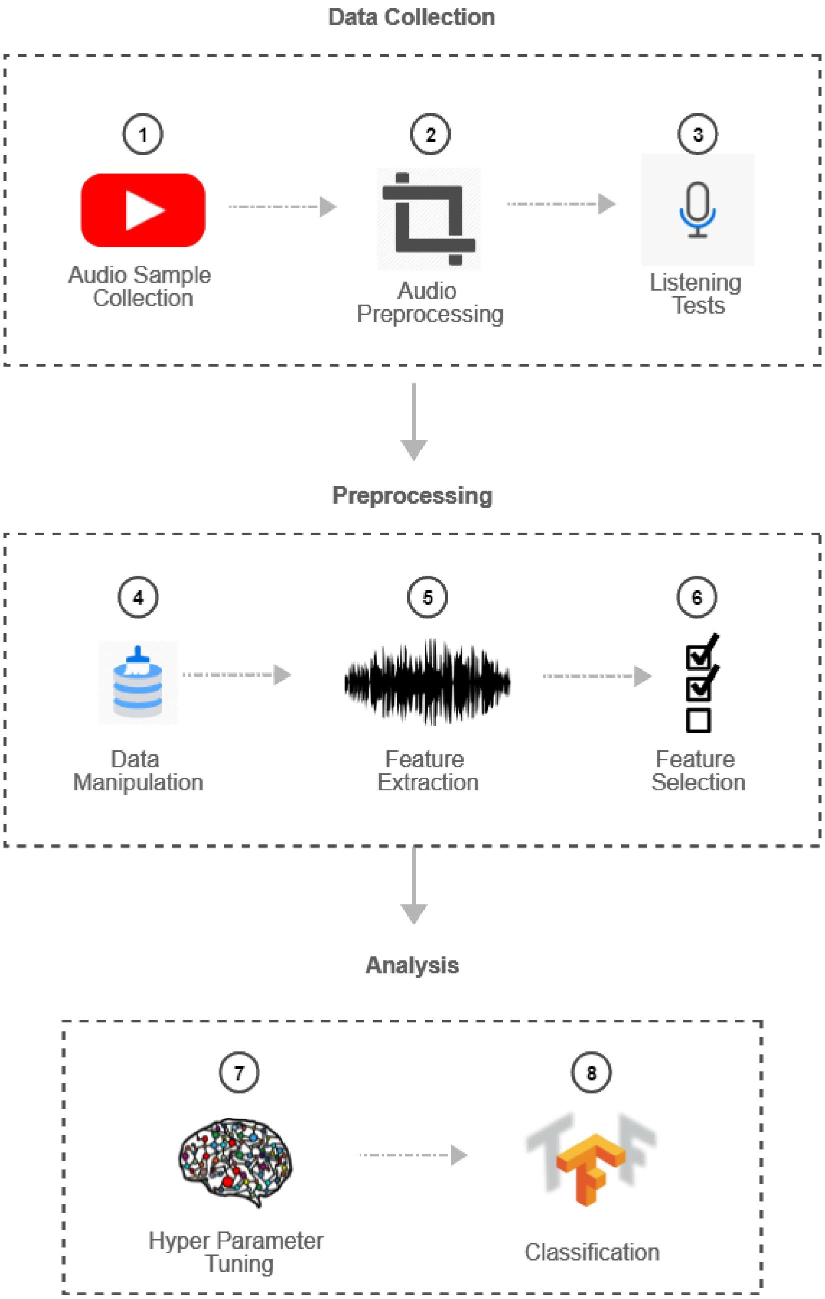


Figure 2.1: Proposed framework for Experimental Procedure

2.1. Dataset Expansion and Listening Tests

2.1.1. Sampling Process and Audio Preparation

Given that binaurally produced audio is not readily available in the majority of the publicly available audio databases, the process of expanding the previous collection of 38 ten-second binaurally recorded audio samples was an extremely expensive process. A set of 162 new binaurally recorded audio samples were sampled, which increased the overall size of the dataset from 38 to 200. A key consideration when sampling audio, was to ensure that the new 162 samples were generated such that each of the emotion classes was equally represented. This ensures the likelihood of having a balanced dataset is increased. Therefore, this process involved finding 27 samples belonging to each of the 6 classes, as perceived by the researcher. The 162 new samples were primarily sourced from YouTube and SoundCloud. Parallel to sampling, each sound was checked as being binaural through MATLAB and cropped to be precisely 10 seconds in length using the open-source digital audio editor, *Audacity*.

2.1.2. Qualtrics Survey

The chosen method for distributing the survey was online, via Qualtrics - a digital survey platform. Participants would use a URL to open the survey digitally, where the audio files along with the six emotions as options would be visible (Appendix A). The set of 200 audio files contained 33 minutes worth of audio approximately. Assuming the entire set of 200 audio files were provided to participants to annotate in a single survey, the completion rate of the survey would have likely been extremely low as it would have taken an estimated 40 - 50 mins to complete. To navigate this obstacle, the set was broken down into 4 equal sets of 50 audio files which ensured that completing a given survey was approximately a 10 to 12-minute task - a much more reasonable survey length. Note, to ensure the 4 surveys had sounds with an approximately even distribution of the emotion classes, the order of the files was randomised using a random list generator on RANDOM.ORG.

It was also decided that certain emotion classes would be categorised with synonyms that closely resemble a similar sensation to said emotion. For example, happiness was presented on the survey as ‘happy/energised’. This grouping of emotions allowed for the likelihood of random responses to be reduced as providing greater depth to the description of the emotions would assist the participant in determining the appropriate choice of emotion felt. Ultimately, this would make the annotation task simpler for the participants which would likely lead to more consistent results.

A trial survey was created using the first set of audio files to mitigate the chance of errors or confusing information occurring after the release of the survey to the participants. This trial survey was distributed and used to make adjustments to improve the survey quality such as improving readability and accessibility to the survey. All 4 final surveys were then created which included a formal participant consent form and debrief to reveal the deception in the experiment. The method used for sampling participants was convenience sampling - the surveys were distributed entirely through word of mouth to people known by the researcher. The goal was to receive at least 30 responses per set, which is equivalent to 30 people responding to all 200 audio samples. This goal was achieved with an average number of responses per survey of 30.3.

2.2 Data Pre-Processing

2.2.1. Data Manipulation

Timing restriction

Another important consideration was that of poor survey responses - in particular, survey responses that took much less time than the minimum expected duration to annotate each audio file with appropriate consideration. It was outlined in the instructions that it was mandatory for all participants to listen to the entire 10-second audio file. This was done as a control. Therefore, given that each of the four surveys had 50, ten-second audio files, the absolute minimum response time would be 500 seconds or 8.3 minutes. Giving consideration to loading times and time needed to contemplate the emotion felt for each sound, a minimum expected response time of 10 minutes was deemed acceptable. All responses that were below a 10-minute duration were removed from the dataset.

Empty values

The problem of participants leaving certain audio files unlabelled was also moderately prevalent in the obtained dataset. There were multiple instances of participants being unable to annotate a file due to technical issues with Qualtrics. This combined with occasional participant laziness is likely the cause of empty values. The ideal approach for minimising the impact of empty values would be to delete an entire response if there were any missing values. This allows for missing values to be dealt with without introducing any biases. However, given that there were only 30 responses per survey - an extremely small sample - it was important not to reduce the number of responses any further by deleting responses with missing values. The only exception to this was that any responses that had more missing values as compared to the filled-in values, were deleted since these responses likely hold little merit. Therefore, a statistical approach was taken to fill the missing entries appropriately. Namely, the mode was used as a measure of central tendency to fill in empty values. The mode of all responses for a given sound represents the emotional class that has most frequency been chosen as representing that sound. Therefore, the mode was the statistical matric used for filling the empty values. This introduces somewhat of a bias however, and may skew the results away from the true emotional label for that sound, but it is preferred to deleting the sample altogether as has been discussed.

Increasing the size of the dataset

As mentioned, the sample size of 200 is extremely small for applications of DL models such as MLP and CNN. To reduce the negative impact of a small dataset on the DL results, the current set of 200, ten-second long exerts were split into 400 five-second long exerts. Given that the audio files were sampled such that the sound throughout each clip was quite regular, the two segments of each sound were assumed to have the same label as the parent sound file. The length of each segment was also made uniform to ensure small inconsistencies between samples were eliminated. That is, the lowest number of samples found in any of the 400 newly formed samples was 433264, and every other audio segment was cropped to match this size to ensure consistency of length between audio files.

2.2.2. Deciding the Success Metric

Based on the processed data, the target labels were deduced which mapped each segmented audio sample, to an emotion class. Table 2.1 displays the distribution of the target labels between the 6 emotion classes for the 400 samples.

Emotion Class	**Percentage of sounds from this class**
Happy (1)	21.5%
Sad (2)	11.5%
Neutral (3)	14.0%
Calm (4)	24.0%
Curious (5)	11.0%
Stress (6)	18.0%

Table 2.1: Distribution of samples between the emotion classes for all 400 samples

The ideal distribution to achieve a perfectly balanced dataset would be approx. 17% of the labels per emotion class. Evidently, the distribution percentages stray from this value, ranging from 11 % - 24%. Since these values are closely distributed around the ideal 17% distribution, the dataset was deemed as being balanced.

For balanced datasets, the most appropriate statistical measure of success for classification tasks is ‘accuracy’ which is defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{FN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}$$

Where: TP is True Positive, TN is True Negative, FP is False Positive, FN is False Negative

A measure of precision would also be used to assess how well each model was able to predict each individual emotion. Precision is defined as follows:

$$\text{Precision} \ = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

2.2.3. Feature Extraction

The majority of the feature extraction was completed using the Python module *Librosa* - an open-source, audio signals processing library. For the temporal and spectral features, the two channels of each binaural recording were averaged to extract said features. On the other hand, the extraction of the ReBTF required both channels of the binaural recording separately, and was performed through custom MATLAB code, which is entirely credited to Manish Kumar.

The input parameters into the librosa functions used to extract features were kept consistent and were as follows:

- Hop length: 512
- Window length for Fast Fourier Transform: 2048
- Overlap Ratio: 0.75
- Number of MFCC’s: 13

Through this process, we extracted two different sets of features for the KNN implementation and then for the MLP and CNN implementations which will now be discussed.

Feature Extraction for the KNN Model

For the KNN model, we focused on the ZCR, RMSE, Spectral Centroid, Spectral Rolloff, Spectral Flatness, 13 MFCC’s, the average of the MFCCs and 6 Tonnetz features along with their average. In total, this accounted for 26 features. For each feature vector, the mean and standard deviation (std) was calculated, which took the total number of features to 52. Repeating this extraction for 400 audio samples led to a total feature set dimensionality of (400 x 52).

Feature Extraction for the MLP and CNN Models

For the MLP and CNN implementations, the raw MFCC and Mel Spectrogram was extracted from each sound file. There were 13 MFCC's extracted at 414 time intervals for each audio file, meaning the feature set for each audio file had a dimensionality of (414 x 13). When this extraction was repeated for all 400 samples, the overall dimensionality of the feature set was (400 x 414 x 13). Similarly for the Mel Spectrogram, a raw (414 x 128) matrix of features was extracted for each audio sample, taking the total dimensionality of this feature set to (400 x 414 x 128).

2.2.4. Feature Selection

Add One In Selection

Feature selection as a pre-processing step for machine learning is highly effective in reducing the dimensionality of data and hence, reducing irrelevant data, increasing learning accuracy, and improving comprehensibility (Rong et al. 2009). This is completed through either feature projection (PCA, LDA, CCA) or feature selection; the latter of which is the focus. Feature selection aims to select a subset of the total features in such a way, that retains maximal relevance to the target (Tang et al., 2014). This approach is preferred to feature projection, as the physical meaning of the features are preserved. For classification applications such as that of this project, one of the proposed methods for feature selection is through the 'wrapper model'. This model follows 3 major steps: subset generation, subset evaluation and result validation.

In the context of this study, the wrapper model was used as inspiration to create the *Add One In* selection process. It was employed as follows. A basic selection of 22 features was chosen to form the control (Appendix B). Every one of the remaining features was then added iteratively to the control selection one at a time. This was the subset generation. Based on a subset evaluation, if the addition of the feature improved the classification accuracy, it was added to a list that collects such features. These features were then added to the control feature group to form a new set of features which formed the final set of features (Tang et al., 2014). Due to the relatively small number of features and size of the dataset, this process was not too expensive to run as it might have been for a larger dataset. Through this process, the final set of *Add One In* features was selected (Appendix B)

Boxplot Selection

Chowdhury et al. (2019) used a boxplot visualisation method to better understand the ability for a given feature to distinguish between emotions. They visualised boxplots to see the spread of each feature against one another for a given emotion, to see which features have distinct ranges of values for a given emotion. This enables a visual understanding to be made as to how well a feature can distinguish between emotions

Based on this approach, a similar feature selection process was executed in this study. Based on the complete set of extracted features, a boxplot was produced for each feature to assess the variation in the spread of that feature as the emotions were varied (Figure _). The main selection criteria for this selection method was that, for a given feature, the interquartile range (IQR) for at least one emotion had to be clearly distinct from all others. For example, it is evident that the RMSE (mean) and Tonnetz 5 (std) are particularly useful features for distinguishing sounds belonging to the emotion classes *Happy* and *Stress* respectively. Note, conclusions drawn from these boxplots aren't definitive and are susceptible to inaccurate representation due to the relatively small dataset. Refer to Appendix B for the full list of features selected through this technique.

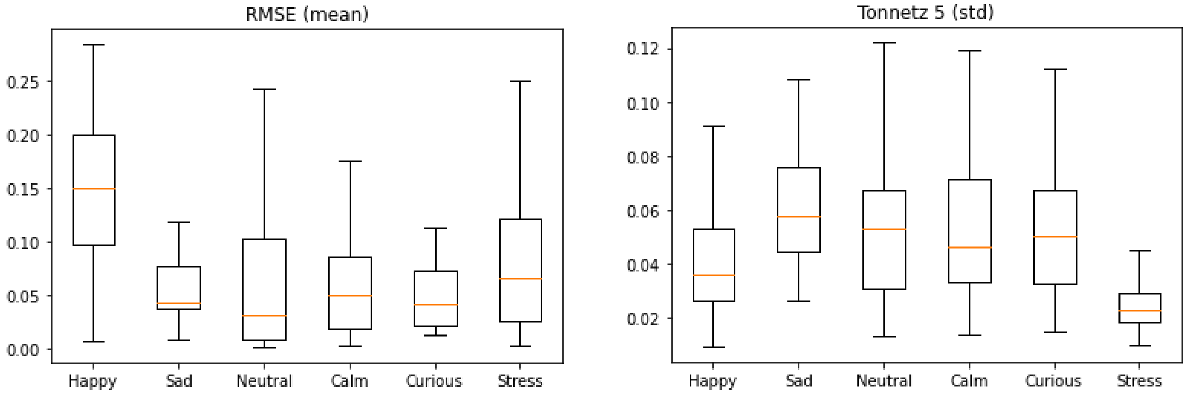


Figure 2.2: Boxplots for RMSE (mean) and Tonnetz 5 (std) of emotions vs, range of feature values

2.3 Analysis

2.3.1. Hyperparameter Tuning through K-Fold Cross Validation

Hyperparameter tuning is an integral part of machine learning. For a given ML model, it involves searching for the combination of hyperparameters that allow for the optimal output to be obtained from the model. The primary approach for hyperparameter tuning used in this study was using a K-Fold Cross-Validation.

Validation in machine learning is the process of tuning a model by first training it and testing the performance of it on a validation set. This validation set is separate from the testing set and exists for the sole purpose of hyperparameter tuning. The validation set is typically chosen as a random subset of the training set. This introduces an element of randomness into the results as there is a chance that the randomly chosen samples in the validation set are dominated by one or a few emotion classes, as opposed to having a more even distribution. K-Fold Cross-Validation is an algorithm that allows for this source of error to be eliminated. It is the process of segmenting the training set into K evenly sized segments, and iteratively choosing 1 set at a time as the validation set. The model is evaluated based on the chosen validation set, and this is repeated K many times. This produces K many model performance results which are then averaged to find the overall validation performance. This is the metric that is used to assess the performance of the model. The set of hyperparameters that produce the best validation result, will be the final set of hyperparameters chosen for evaluating the testing dataset.

The next step in the Analysis phase of the experimental procedure was to prepare and implement the 3 different models. This was done in 3 stages for each model: splitting the dataset into the training, testing and validation sets; creating the different test groups and finally, implementing the model through code.

2.3.2. Preparing and Implementing the Traditional ML Model (KNN)

Splitting the dataset into training, testing and validation sets

The set of 400 samples was split into a training set of 320 samples and a testing set of 80 samples. The validation set is to be created from the 320 samples, such that it contains 40 samples. This ensured the entire dataset was split into training, validation and testing sets according to a 7:1:2 ratio. This splitting of the dataset was performed using the *train_test_split*() method in sklearn's *model_selection* module. Note, the exact same split was used for the MLP and CNN models.

Creating test groups

As mentioned, using a traditional ML algorithm such as KNN comes with the requirement that a certain degree of feature selection is performed. The quality of the features selected determines the quality of the model's performance. Therefore, a total of 8 test groups were formed, each with a unique combination of features, to increase the likelihood of producing a strong model. These test groups are outlined below in Table 2.2. Note, each high-level group (i.e. Group 1, 2, 3, 4) is referred to as different 'feature groups', while each low-level group (Group 1A, Group 1B, Group 1C, Group 2A, etc.) is referred to as different 'test groups'. Further description of each group is provided in Table 2.2 below.

Test Group	Feature Group	Subset	Size of Group	Description of Group
Group 1A	Complete Feature Set	All	(400, 52)	The complete feature set
Group 1B	Complete Feature Set	Mean	(400, 26)	The subset of mean features from the complete feature set.
Group 1C	Complete Feature Set	Standard Deviation (std)	(400, 26)	The subset of standard deviation features from the complete feature set.
Group 2A	MFCC Feature Set	All	(400, 26)	All of the MFCC features
Group 2B	MFCC Feature Set	Mean	(400, 13)	The subset of mean features from the MFCC feature set.
Group 2C	MFCC Feature Set	Standard Deviation (std)	(400, 13)	The subset of standard deviation features from the MFCC feature set.
Group 3	Add One In Feature Set	N/A	(400, 24)	The feature set selected through the Add On In feature selection process
Group 4	Boxplot Selected Set	N/A	(400, 13)	The feature set selected through the Add On In feature selection process

Table 2.2: Description of all test groups for KNN implementation

Implementing the code

The function *KNeighborsClassifier* from the *sklearn* python module was used to build and evaluate the KNN model. The main input parameter of this function was *n_neighbors* - the number of neighbours to be used when clustering points. According to Zhongguo et al. (2017), it was determined that the range of n_neighbor values to try when evaluating the model should be 1 to the square root of the number of data samples. Therefore, the range of *n_neighbors* values used was 1 to 20. Note, this is a hyperparameter and the choice of this range is mostly arbitrary. Given that KNN is a 'distance-based classifier' as explained in Section 1.1.2, it was necessary to standardise the dataset to ensure all variables contributed equally to the distance measures. This standardisation was completed through the *StandardScaler* function within the *sklearn* module, which maps a given dataset onto the standard normal distribution. The KNN model was built by passing in the standardised training dataset. The *GridSearch*CV python module was then used to iteratively assess how the model performs for the entire range of n_neighbours values from 1 to 20. For each of these iterations, an 8-fold cross-validation split was used. The optimal hyperparameters are chosen based on a comparison of validation results. These hyperparameters were finally fed into the model, and used to assess the overall performance of the model using the test data.

2.3.3. Preparing and Implementing the Deep Learning Models (MLP & CNN)

Splitting the dataset into training, testing and validation sets

The training, validating and testing split for the MLP and CNN models were created the same way as described above for the KNN model.

Creating test groups

Unlike traditional ML, deep learning models such as the MLP and CNN do not require any feature selection. Therefore, raw data was used as input into the MLP and CNN. Each of the two models were assigned two identical feature sets - the MFCC and Mel Spectrogram features - resulting in a total of 4 test groups, as outlined in Table 2.3 below.

Test Group	**High-Level Group**	**Model**	**Size of Group**	**Description of Group**
Group 5A	MFCC raw data	MLP	(400, 414, 13)	The raw MFCC data with the MLP model
Group 5B	Mel Spectrogram raw data	MLP	(400, 414, 128)	The raw Mel Spectrogram data with the MLP model
Group 6A	MFCC raw data	CNN	(400, 414, 13)	The raw MFCC data with the CNN model
Group 6B	Mel Spectrogram raw data	CNN	(400, 414, 128)	The raw Mel Spectrogram data with the CNN model

Table 2.3: Description of all test groups for MLP and CNN implementation

Implementing the code

The implementation of the MLP and CNN models both relied heavily on the *TensorFlow* and *Keras* Python libraries. Note, the majority of the credit for code used for MLP and CNN analysis is credited to V. Velardo (2020).

For the MLP, the model was built as a sequential neural network, with a single input layer, 3 hidden layers and a single output layer. The input layer was created using the *keras.layers.Flatten()* method, in order to flatten the multidimensional input data into a column vector. The 3 hidden layers were then all made using the *keras.layers.Dense()* method, with 512, 256 and 64 neurons respectively. The choice of these respective numbers of neurons was entirely arbitrary. Each of these hidden layers used the *Rectified Linear Activation Unit (ReLu)* as the activation function. Finally, the identical *keras.layers.Dense()* method was used with 6 neurons to comprise the output layer, which used the *Softmax* function for activation.

For the CNN on the other hand, there were 3 convolutional layers, 3 max-pooling layers, a single hidden dense layer and a single output layer. Similar to the MLP, it was built as a sequential neural network. The convolutional layers were all created identically using *keras.layers.Conv2D()*, with the number of layers as 32, filter size as 3x3, activation function as *ReLu*, and with zero padding used. The max-pooling layers were created with *keras.layers.MaxPooling2D()*, each had a filter size of 3x3, a stride size of 2x2 and once more, zero padding being used. These convolutional and max-pooling layers were stacked interchangeably. A batch normalisation layer was placed at the end of each stack to normalise the values. This was done with the *BatchNormalization()* method. The output of the 3rd convolutional layer was flattened using *keras.layers.Flatten()* and passed into a dense layer with 64 neurons and *ReLu* as its activation function. Finally, the output layer was constructed as a dense layer with 6 neurons, and *Softmax* activation.

Once both the MLP and CNN models were built, they were both optimised using the 'Adam' optimiser - a variant of stochastic gradient descent. The learning rate used was 0.0001, the loss function used was *sparse_categorical_crossentropy* and the metric assigned to the model was *accuracy*. Both models were run for 500 epochs across all test groups. No hyperparameter tuning was performed for either DL model. The models were then validated and tested to measure performance.

3. Results and Discussion

3.1. Results

For each test group, the model's performance was measured with and without the inclusion of the ReBTF. The results are presented in Table 3.1 and Table 3.2 below, which show the model performance for the traditional ML model (KNN) and the DL model (MLP & CNN) results respectively.

3.1.1 Results for Traditional ML Model (KNN)

Test Group	Accuracy without ReBTF	Accuracy with ReBTF
Test Group 1A (Complete, all)	70.00%	70.00%
Test Group 1B (Complete, mean)	68.75%	70.00%
Test Group 1C (Complete, std)	51.25%	51.25%
Test Group 2A (MFCC, all)	66.25%	68.75%
Test Group 2B (MFCC, mean)	66.25%	66.25%
Test Group 2C (MFCC, std)	31.25%	31.25%
Test Group 3 (Add One In)	71.25%	75.00%
Test Group 4 (Boxplot)	53.75%	53.75%

Table 3.1: Traditional ML (KNN) results for all 8 test groups. Green highlight indicates the best overall performer within each feature group (Group 1, 2, 3 and 4).

When the ReBTF was and wasn't included in the feature set, the range of accuracy values was 31.25% to 75% and 31.25% to 71.25% respectively. The average accuracy when the ReBTF was and wasn't included in the feature set was 60.78% and 59.84% respectively.

When considering the best performing test group within each feature group, the results were as follows. For Group 1, the best performing test was Group 1A which was the entire collection of features. This test had an overall performance of 70% for when the ReBTF was both included and excluded. For Group 2, the best performing test was Group 2A which was the entire collection of MFCC features. This test had an overall performance of 66.25% without the ReBTF, which improved to 68.75% when the ReBTF was included. There was only one test within Group 3 - the Add One In features - and the overall performance was 71.25% without the ReBTF, which improved to 75% with the inclusion of the ReBTF. Similarly for Group 4 - the Boxplot features -, there was only one test within this group and the overall performance was 53.75% with and without the ReBTF

Overall, the best performing model from all variations of test groups (with and without the ReBTF), was Test Group 3 (the Add One In features) with the ReBTF included. This test group resulted in an overall accuracy of 75%. The worst performing model was Group 2C - the standard deviation feature subset of the MFCC feature group. The performance accuracy was 31.25% regardless of whether the ReBTF was included or not.

Confusion Matrices and Precision Tables

Confusion matrices are a performance measurement tool used predominantly in ML classification applications. In this context, they are a matrix representation with the target emotion against the predicted emotions. Figure 3.1 depicts the confusion matrix for Test Group 1A both with and without the ReBTF.

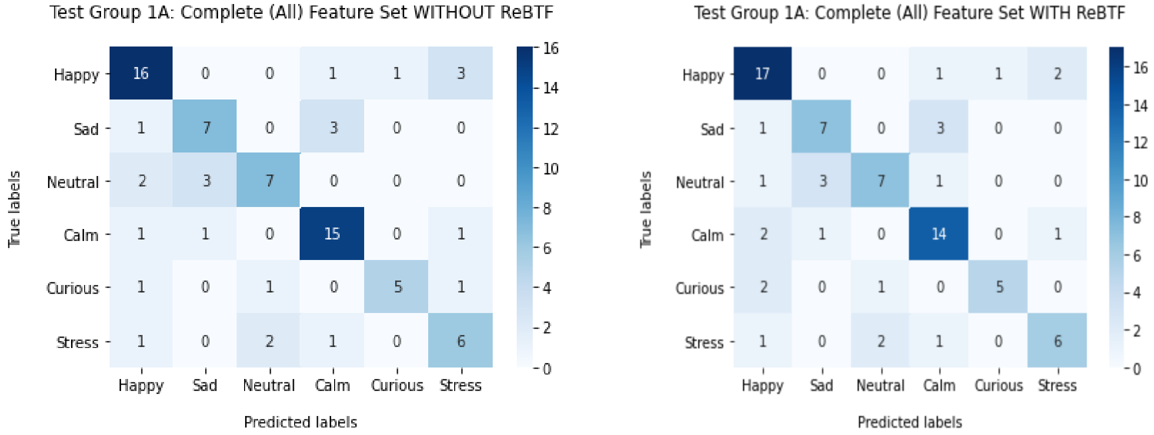


Figure 3.1: Confusion matrices for Test Group 1A with and without the ReBTF

Suppose the matrix has i rows and j columns. The confusion matrix indicates the probability that emotion i is recognised by the model as emotion j. Hence, values that lie on the main diagonal of the matrix represent values where the predicted emotion matched the target emotion. The percentage of the data that lies on the diagonal represents the accuracy of the model. Likewise, the precision of the model in predicting each emotion can be calculated as the percentage of predicted values for a given emotion that were accurately predicted. Note these descriptions of accuracy and precision are consistent with the explanation from Section 2.2.2.

Table 3.2 summarised the precision of the model from Test Group 1A in predicting each emotion class with and without the ReBTF. This information may be directly extracted from the confusion matrices. Since the table is more informative for the interest of this study, the confusion matrices for the rest of the Test Groups will be found in Appendix C. Below are the tables for the 4 best performing models for each feature group (highlighted in green in Table 3.1 above). Note, it is important to discuss the findings for various test groups and not just the best one as this increases the reliability of the findings.

For Test Group 1A, the most precisely predicted emotion was *curiosity* with an overall precision of 0.83, both when the ReBTF was and wasn't included respectively (Table 3.2). The most poorly predicted emotion for the group without the ReBTF was *stress* with a precision of 0.55, while that of the group with the ReBTF was *sad* with a precision of 0.64. Note, the average change in performance when the ReBTF is included will be introduced and elaborated on in Section 3.2 (Discussion).

Emotion	Precision (without ReBTF)	Precision (with ReBTF)
Happy	0.73	0.71
Sad	0.64	0.64
Neutral	0.70	0.70
Calm	0.75	0.70
Curious	0.83	0.83
Stress	0.55	0.67

Table 3.2: Comparison of precision for Test Group 1A (Complete, all) with and without the ReBTF

For this Test Group 2A, the most precisely predicted emotion was *happy* with an overall precision of 0.75, both when the ReBTF was and wasn't included respectively (Table 3.3). The most poorly predicted emotion for when the ReBTF was and wasn't included was *curious*, with a precision of 0.60 and 0.50 respectively.

Emotion	Precision (without ReBTF)	Precision (with ReBTF)
Happy	0.75	0.75
Sad	0.62	0.67
Neutral	0.67	0.73
Calm	0.62	0.65
Curious	0.50	0.60
Stress	0.67	0.62

Table 3.3: Comparison of precision for Test Group 2A (MFCC, all) with and without the ReBTF

For Test Group 3, the most precisely predicted emotion was *happy* with an overall precision of 0.88 and 0.85 respectively, both when the ReBTF was and wasn't included respectively (Table 3.4). The most poorly predicted emotion was *neutral* both for when the ReBTF was and wasn't included with a precision of 0.18 and 0.20 respectively.

Emotion	Precision (without ReBTF)	Precision (with ReBTF)
Happy	0.85	0.88
Sad	0.38	0.46
Neutral	0.20	0.18
Calm	0.47	0.48
Curious	0.57	0.60
Stress	0.55	0.54

Table 3.4: Comparison of precision for Test Group 3 (Add One In) with and without the ReBTF

For Test Group 4, the most precisely predicted emotion was *curious* with an overall precision of 0.86 and 1.00 respectively, both when the ReBTF was and wasn't included (Table 3.5). The most poorly predicted emotion for the variation without the ReBTF was *stress* with a precision of 0.54, while that of the group with the ReBTF was *sad* with a precision of 0.62.

Emotion	Precision (without ReBTF)	Precision (with ReBTF)
Happy	0.80	0.83
Sad	0.67	0.62
Neutral	0.70	0.78
Calm	0.71	0.75
Curious	1.00	0.86
Stress	0.54	0.70

Table 3.5: Comparison of precision for Test Group 4 (Boxplot) with and without the ReBTF

3.1.2 Results for Deep Learning Models (MLP & CNN)

Test Group	Accuracy without ReBTF	Accuracy with ReBTF
Group 5A (MLP, MFCC)	47.5%	42.5%
Group 5B (MLP, Mel Spectrogram)	38.7%	30.0%
Group 6A (CNN, MFCC)	48.8%	52.5%
Group 6B (CNN, Mel Spectrogram)	45.0%	52.5%

Table 3.6: Deep Learning (MLP & CNN) Results for all 4 test groups

From Table 2.6, the accuracy values ranged from 38.7% to 48.8% when the ReBTF was excluded, while it ranged from 30% to 52.5% when the ReBTF was included. The average accuracy when the ReBTF was included was 45%, while that when the ReBTF was excluded decreased slightly to 44.4% respectively. The average accuracy for the MLP and the CNN were 39.7% and 49.7% respectively.

For Group 5, the best performing test was Test Group 5A which was MFCC features combined with the MLP. This test had an overall performance of 47.5% and 42.5% with and without the ReBTF respectively. For Group 6, the best performing test was Test Group 6A which was once again the MFCC features but this time combined with the CNN. This test had an overall performance of 48.8% without the ReBTF, which increased to 52.5% when the ReBTF was included.

In terms of the best performing model from all variations of test groups, there were two tied winners. These were Group 6A and Group 6B with the ReBTF included, which had an accuracy of 52.5% for both test groups. The worst performing model was Group 5B with the ReBTF included. This model had an accuracy of 30.0%.

Confusion Matrices and Precision Tables

Similar to the previous section, information from the confusion matrices for these tests may be translated directly into precision comparison tables. Below are the tables for the 2 best performing test groups for each feature group (highlighted in green in Table 3.6 above). Refer to Appendix D for the confusion matrices.

For Test Group 5A, the most precisely predicted emotion was *curious* with an overall precision of 0.57 when the ReBTF and *happy* with an accuracy of 0.67 when the ReBTF wasn't included. The most poorly predicted emotion was *sad* both for when the ReBTF was and wasn't included with a precision of 0.25 and 0.08 respectively.

Emotion	Precision (without ReBTF)	Precision (with ReBTF)
Happy	0.57	0.29
Sad	0.08	0.25
Neutral	0.42	0.33
Calm	0.45	0.43
Curious	0.50	0.67
Stress	0.22	0.45

Table 3.7: Comparison of precision for Test Group 5A with and without the ReBTF

For Test Group 6A, the most precisely predicted emotion was *happy* for when the ReBTF was included and *calm* when the ReBTF wasn't included, with an overall precision of 0.57 and 0.67 respectively. The most poorly predicted emotion was 'sad' both for when the ReBTF was and wasn't included with a precision of 0.25 and 0.08 respectively.

Emotion	**Precision (without ReBTF)**	**Precision (with ReBTF)**
Happy	0.47	0.50
Sad	0.45	0.36
Neutral	0	0.33
Calm	0.50	0.45
Curious	0.40	0.38
Stress	0.26	0.36

Table 3.8: Comparison of precision for Test Group 6A with and without the ReBTF

3.2. Discussion

3.2.1. Findings

Primary Findings (related to the ReBTF)

Based on the results presented in Section 3.1, the following tables may be deduced which demonstrate the overall change in accuracy and precision once the ReBTF is included in a dataset.

Test Group	**Description of features used**	**Percent change in accuracy when ReBTF included**
Test Group 1A	Complete, all	0%
Test Group 1B	Complete, mean	+ 1.82%
Test Group 1C	Complete, std	0%
Test Group 2A	MFCC, all	+ 3.77%
Test Group 2B	MFCC, mean	0%
Test Group 2B	MFCC, std	0%
Test Group 3	Add On In	0%
Test Group 4	Boxplot	+ 5.27%

Table 3.9: Percent change for traditional ML (KNN) results for all 8 test groups when ReBTF is included. Green indicates an improved result, red indicates a worsened result, and white indicates no change.

Firstly, it can be seen from the KNN results, that the inclusion of the ReBTF only ever improved the overall model accuracy. For 5 out of the 8 tests, it left the model accuracy unchanged whereas, for 3 of the tests, it improved the model accuracy. In fact, on average, the performance increased by **1.36%** across all test groups. Even though this was only a minor overall improvement in performance, this was to be expected seeing as the inclusion of 1 feature into a set of 13 to 26 other features should not be making a substantial improvement to the result,

Test #	Description	Percent change in accuracy when ReBTF included
Test Group 5A	MLP, MFCC	-10.5%
Test Group 5B	MLP, Mel Spectrogram	-22.5%
Test Group 6A	CNN, MFCC	+7.6%
Test Group 6B	CNN, Mel Spectrogram	+ 16.7%

Table 3.10: Percent change for DL (MLP & CNN) results for all 4 test groups when ReBTF is included. Green highlight indicates an improved result, red highlight indicates a worsened result, and a white highlight indicates no change.

For the DL results, the inclusion of the ReBTF improved the model's performance for 2 out of the 4 tests. For the other 2 tests, the ReBTF decreased the overall accuracy of the model. On average, the inclusion of the ReBTF decreased the model's performance by 2.12%. This result suggests that the inclusion of the ReBTF may decrease the overall performance of AER, as opposed to increasing it. Albeit the reliability of the DL results must first be discussed.

The inclusion of a single feature into a set of 13 MFCC features or 128 Mel Spectrogram features, should not induce as big of an improvement as was found for these various trials. For example, it was found for Test 5B and Test 6B that the model's performance was decreased by 22.5% and increased by 16.7% respectively. Such jumps in performance increase or decrease are extremely unlikely given that only one additional feature was added into the set. Additionally, hyperparameter tuning was not performed on the DL models, which further decreased the consistency and accuracy of the outputs. Furthermore, when the tests were repeatedly run over numerous trials, completely different results would arise each time making the results extremely random in nature. The results that were ultimately presented in this report were the results from the final iteration of testing. Furthermore, there is a significant limitation that is inherent to using deep learning for this study. Deep learning requires millions of samples of data to be trained sufficiently, but the dataset available for this study was 3 orders of magnitude lower than this. Deep learning was implemented purely to increase the scope of the study, and it was expected that the results obtained would lack credibility.

Conversely, the KNN model which relies on a traditional ML approach was free of each of the aforementioned limitations. The introduction of the ReBTF only changed the performance of the model by 5.27% at most (Test Group 4), which is an acceptable value. Furthermore, upon running multiple iterations of the same test, identical results were always produced, providing a much more stable understanding of the impact of the ReBTF. Finally, a sample size of 400 is sufficient for training a KNN model, assuming that an appropriate feature selection process is completed. For the reasons mentioned, the DL results will need further validation before any weight can be given to them. As such, the remainder of the discussion will mainly focus on the KNN results.

Emotion	Average precision (without ReBTF)	Average precision (with ReBTF)	Percent change
Happy	0.78	0.79	+ 1.3%
Sad	0.58	0.60	+ 3.5%
Neutral	0.57	0.60	+ 5.3%
Calm	0.64	0.65	+ 1.2%
Curious	0.73	0.72	- 0.3%
Stress	0.58	0.63	+ 9.5%

Table 3.11: Average precision for each emotion with and without the ReBTF, across 4 top performing KNN test groups and the percent change when the ReBTF is included

As seen above in Table 3.11, the inclusion of the ReBTF also improved the precision in the KNN models' ability to correctly predict each of the 6 emotion classes. The precision for the prediction of the *stress* emotion class was the most improved, with an increase in precision of 9.5% when the ReBTF was included. The precision for the prediction of the *curious* was the only class to see a decrease in performance. It should be noted that the decrease was only by 0.3. Refer to Appendix E for the entire table for the percent change in prediction precision for each emotion when ReBTF was included for the 4 best performing test groups.

Secondary Findings:

These findings are more related to AER more generally, as opposed to findings that explore the impact of the ReBTF on AER specifically.

Based on the boxplots, all Tonnetz and MFCC mean features were concluded to be extremely poor in distinguishing between any pair of emotions. On the contrary, the Tonnetz std features tended to have a unique range of values for the *happy* and *stress* emotion classes. Both RMSE mean and std features had outstanding ability to distinguish happy sounds from the rest. No features seemed to be able to distinguish the neutral and the curious sounds too well. Albeit, when performing the Add One In feature selection process, the features that were identified as likely being poor for AER during the Boxplot selection process such as the MFCC 4 (mean), actually seemed to improve the Add One In based model. Likewise with the features that were predicted from the Boxplot selection to increase performance, didn't necessarily do so based on the Add One In process such as the ZCR (std). Therefore, the selection criteria of the Boxplot feature selection process may not be as reliable as was initially thought.

The benchmark accuracy for this study was set by the research conducted by Kaur & Kumar (2021) in which their MLP classification model achieved a prediction accuracy of 90%. The highest accuracy reached through the present study was only 75% by the KNN model with the Add One In feature set including the ReBTF. This suggests there are likely a plethora of limitations that exist with the current study that could be improved to enhance the overall model performance.

3.2.2. Limitations

Participant sampling method and the number of participants

The participant sampling method used - convenience sampling - could be improved. While this method of sampling is likely to result in the largest participant number, a significant bias is introduced through this sampling method. That is, those who chose to participate in the study may have an inherently common trait from those that chose not to. This is known as volunteer bias. For this reason, the sample is also not fully representative of particular characteristics such as cultural background which indicates that the annotations may be culturally biased. Hence, the reliability of the audio annotations, and thus the reliability of the model performance, are put into question. Additionally, while the number of participants was a substantial improvement from the previous study, it was still not large enough. An example of a problem that arose due to this limitation, was that there were multiple audio samples that were bimodal or even trimodal in terms of the emotion class chosen by the majority. For these particular audio samples, the experimenter selected which label out of the two modes would best represent the data. This inherently introduces experimenter bias into these labels, and in future, would have been better dealt with by getting another outside opinion on the audio file

Improvements to the experimental procedure

Given outside circumstances, the experiment needed to be conducted online. This meant that the audio samples had to be formatted in such a way that they were short and were arranged one after the other so that participants could complete the survey quickly before experiencing survey fatigue. As a result of the samples being so short and participants listening to them in quick succession, there is a strong chance that there was insufficient time for the samples to induce true emotions within participants. Upon collecting feedback about the survey from participants, this fact was further confirmed. Ultimately, this suggests that the annotations of the study may be subject to a certain degree of randomness. Furthermore, the online format of the experiment meant that it was almost impossible to control the environmental conditions in which participants completed the survey. This likely resulted in participants rushing the survey, or completing the survey in any one of many emotional states which could have affected the way in which they responded to the audio samples. Had the study been conducted in person, participants could have, for example, been asked to enjoy a short game prior to beginning the listening test, as a way of increasing the likelihood that participants are in a similar mindset prior to the listening tests.

Audio sample size

As mentioned, the relatively small sample set was the most concerning limitation of the study. In particular, the accuracy and the reliability of the DL models suffered greatly as a result. Given that there are currently no open-source databases containing binaurally reproduced audio, this limitation was unavoidable.

Unbalanced dataset

While the dataset was considered to be mostly balanced during the analysis, it was still not perfectly balanced. There was a degree of bias present in the labels, as there were more *happy and calm* sounds than there are any other type of sound. As a result, there is a chance that sounds with more labels were predicted more often. The ideal approach for addressing this limitation may have been to use oversampling, which is the process of synthetically generating new observations based on the collected data.

Using hyperparameter tuning for deep learning to reduce the randomness

While hyperparameter tuning was performed for the KNN model, it was not done so for the MLP and CNN models. This hyperparameter tuning could have been implemented using the *Keras Tuner* module in *TensorFlow*'s library. For the MLP, this would have allowed for the number of layers, the number of neurons in each layer, and the activation and loss function to be optimised. On top of this list, the filter size, stride length and padding type could have been optimised for the CNN model.

4. Further Work

Further work based on limitations:

There are a number of improvements that can be made to the current study based on its limitations. For instance, future work that can repeat the experiment with a substantially larger dataset would be highly insightful for using DL to further validate whether the ReBTF can improve AER performance. Any further work that implements hyperparameter tuning for the DL algorithms, perfectly balances the dataset, ensures greater control over the experimental conditions, and samples participants strategically to account for potential cultural and volunteer bias, are all useful avenues.

Implementing the dimensionality approach

The current study used the categorical theory of emotions to explore the ReBTF. As mentioned, the dimensional approach provides more of a soft assignment of emotions which would be extremely useful for navigating the subjective nature of emotions. Another added benefit of using the dimensional approach would be that participants would find the annotation process more straightforward, since they have more range to choose from when labelling the extent of arousal, valence or potency that they felt.

More sophisticated feature selection techniques

The feature selection and dimensionality reduction processes used in this study were quite basic in comparison to some of the techniques that exist. Potential techniques to be implemented in the future for dimensionality reduction include Principal-Component Analysis (PCA), Curvilinear Component Analysis (CCA) and Linear Discriminant Analysis (LDA). In terms of feature selection techniques, some of the more renowned approaches include Information Gain and the Fisher Score, which may assist in disregarding particular features that have less relevance for AER.

Suggestions from the previous study

Some of the suggestions for future work as recommended from the previous version of this study still apply. That is, using a greater number of spatial features or simply using different ones. This comes with the difficulty of being extremely time-intensive since the audio would likely have to be recorded by the experimenter in order to retain all information about the spatial features of the sound. Nevertheless, it should be explored in the future as it would be extremely insightful for the subject matter. Finally, using a single type of sound as opposed to mixing speech, music and background noise. This would allow for a more concrete understanding of how the ReBTF impacts emotion classification for a particular type of sound.

Conclusion

The present work was an investigation of whether a particular spatial feature of sound - the Relative Binaural Transfer Function (ReBTF) - can improve the accuracy of acoustic emotion detection. This was done with respect to 3 machine learning models - namely K-Nearest Neighbour (KNN), Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN). This was an expansion of a previous, preliminary investigation into the study which involved using a larger sample of audio files, more participants for annotation, more suitable ML algorithms, and introducing feature selection to improve the relevance of the features used. It was found that the inclusion of the ReBTF improved the AER performance of the KNN model by 1.38% on average. Conversely, the inclusion of the ReBTF decreased the performance of the MLP and CNN models by 2.12% on average. Albeit, this finding holds little credibility due to the many limitations present in the implementation of both the MLP and CNN models, particularly the small dataset size. With respect to the KNN model, it was also found that the inclusion of the ReBTF also individually improved the precision of the model's prediction of all individual emotion classes, except for *curious. The* major limitations of the study included the participant sampling method, lack of control over experimental conditions and most importantly, a relatively small dataset. Nevertheless, the results obtained are very encouraging given the size of the dataset, and further investigation into the topic is necessary to further validate and expand on the findings.

Bibliography

Rathor, S., Khandelwal, D., Nigam, H., & Tomar, S. (2021). Modeling of Acoustic emotion recognition using Artificial Intelligence and Machine Learning. IOP Conference Series: Materials Science and Engineering, 1116(1), 012129. https://doi.org/10.1088/1757-899x/1116/1/012129

Lanjewar, R. B., Mathurkar, S., & Patel, N. (2015). Implementation and Comparison of Speech Emotion Recognition System Using Gaussian Mixture Model (GMM) and K- Nearest Neighbor (K-NN) Techniques. Procedia Computer Science, 49, 50-57. https://doi.org/https://doi.org/10.1016/j.procs.2015.04.226

Seo, Y.-S. and J.-H. Huh (2019). "Automatic Emotion-Based Music Classification for Supporting Intelligent IoT Applications." Electronics 8(2).

Nanavare, V. V., & Jagtap, S. K. (2015). Recognition of Human Emotions from Speech Processing. Procedia Computer Science, 49, 24-32. https://doi.org/https://doi.org/10.1016/j.procs.2015.04.223

Panda, Renato & Malheiro, Ricardo & Paiva, Rui Pedro. (2018). Novel Audio Features for Music Emotion Recognition. IEEE Transactions on Affective Computing. 11. 614 - 626. 10.1109/TAFFC.2018.2820691.

El Ayadi, M., et al. (2011). "Survey on speech emotion recognition: Features, classification schemes, and

databases." Pattern Recognition 44(3): 572-587.

Kerkeni, L., et al. (2019). Automatic Speech Emotion Recognition Using Machine Learning: https://www.intechopen.com/online-first/automatic.

Kim, Y., et al. (2010). "Music emotion recognition: A state of the art review." Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010.

Yang, Y.-H. and H. H. Chen (2012). "Machine Recognition of Music Emotion." ACM Transactions on Intelligent Systems and Technology 3(3): 1-30

Sharar, S. R., Alamdari, A., Hoffer, C., Hoffman, H. G., Jensen, M. P., & Patterson, D. R. (2016). Circumplex Model of Affect: A Measure of Pleasure and Arousal During Virtual Reality Distraction Analgesia. Games for health journal, 5(3), 197–202. https://doi.org/10.1089/g4h.2015.0046
Matsuda, Y.-T., Fujimura, T., Katahira, K., Okada, M., Ueno, K., Cheng, K., & Okanoya, K. (2013). The implicit processing of categorical and dimensional strategies: an fMRI study of facial emotion perception [Original Research]. Frontiers in Human Neuroscience, 7(551). https://doi.org/10.3389/fnhum.2013.00551

Lech, M., Stolar, M., Best, C., & Bolia, R. (2020). Real-Time Speech Emotion Recognition Using a Pre-trained Image Classification Network: Effects of Bandwidth Reduction and Companding [Original Research]. Frontiers in Computer Science, 2(14). https://doi.org/10.3389/fcomp.2020.00014

Kaur, J., & Kumar, A. (2021). Speech Emotion Recognition Using CNN, k-NN, MLP and Random Forest. In S. Smys, R. Palanisamy, Á. Rocha, & G. N. Beligiannis, Computer Networks and Inventive Communication Technologies Singapore.

Botalb, Abdelaziz & Moinuddin, Muhammad & Al-Saggaf, Ubaid & Ali, Syed Saad. (2018). Contrasting Convolutional Neural Network (CNN) with Multi-Layer Perceptron (MLP) for Big Data Analysis. 1-5. 10.1109/ICIAS.2018.8540626.

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into Imaging, 9(4), 611-629. https://doi.org/10.1007/s13244-018-0639-9

Amrutha, K., Sunanda, P., Rohit, M., Rama, S. (2021). Speech Emotion Recognition using Acoustic Features. International Research Journal of Engineering and Technology (IRJET),

Cuadrado, F., et al. (2020). "Arousing the Sound: A Field Study on the Emotional Impact on Children of
Arousing Sound Design and 3D Audio Spatialization in an Audio Story." Frontiers in Psychology 11(737).

Fletcher, M., 2011. The Effect Of Spatial Treatment Of Music On Listener's Emotional Arousal. Journal on the Art of Record Production

S Soundarya and N Arumugam. (2020). Speech Emotion Recognition Based on CNN Combined with Decision Tree Classifier, International Journal for Modern Trends in Science and Technology, Vol. 06, Issue 02, pp.:100-104.

Ramdinmawii, E., et al. 2017. Emotion recognition from speech signal, IEEE.

Er, M. B. (2020). "A Novel Approach for Classification of Speech Emotions Based on Deep and Acoustic
Features." IEEE Access 8: 221640-221653.

Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. In Data Classification: Algorithms and Applications (pp. 37-64). CRC Press. https://doi.org/10.1201/b17320

Chowdhury, S., Vall, A., Haunschmid, V., & Widmer, G. (2019). Towards Explainable Music Emotion Recognition: The Route via Mid-level Features.

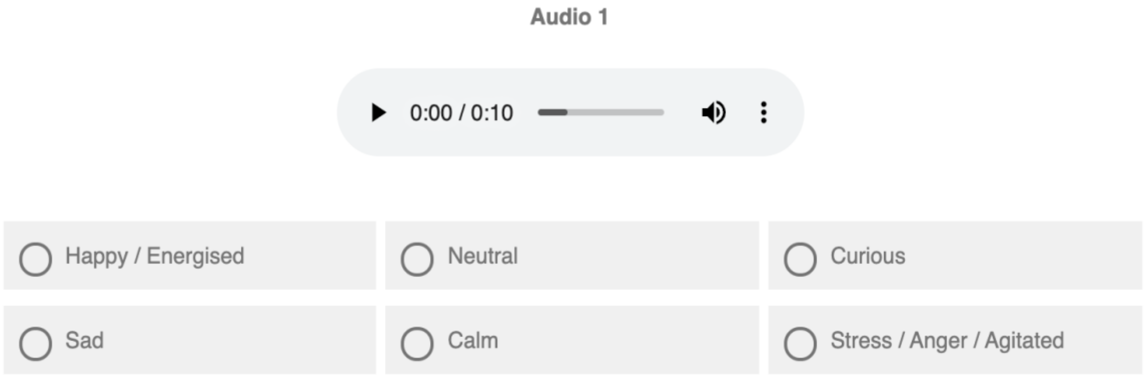
Zhongguo, Yang & Hongqi, Li & Liping, Zhu & Qiang, Liu & Ali, Sikandar. (2017). A case based method to predict optimal k value for k-NN algorithm. Journal of Intelligent & Fuzzy Systems. 33. 1-10. 10.3233/JIFS-161062.

Velardo, V., 2020. 16- How to Implement a CNN for Music Genre Classification. [video] Available at: <https://www.youtube.com/watch?v=dOG-HxpbMSw&t=1596s> [Accessed 1 October 2021].

Abdulraheem, A., Abdullah Arshah, R., & Qin, H. (2015). Evaluating the Effect of Dataset Size on Predictive Model Using Supervised Learning Technique. International Journal of Software Engineering & Computer Sciences (IJSECS), 1, 75-84. https://doi.org/10.15282/ijsecs.1.2015.6.0006

Appendix

Appendix A: Extract of Survey from Qualtrics



Audio 1

○ Happy / Energised	○ Neutral	○ Curious
○ Sad	○ Calm	○ Stress / Anger / Agitated

Appendix B: Features for Add One In and Boxplot Feature Selection

Control selection of features for Add One In	**Selected features from Add One In**	**Features used in Boxplot features**
ZCR (mean)	*ZCR (mean)*	*ZCR (mean)*
RMSE (mean)	*RMSE (mean)*	*RMSE (mean)*
Spectral Centroid (mean)	*Spectral Centroid (mean)*	*Spectral Rolloff (mean)*
Spectral Rolloff (mean)	*Spectral Rolloff (mean)*	*MFCC avg (mean)*
Spectral Flatness (mean)	*Spectral Flatness (mean)*	*MFCC 1 (mean)*
MFCC avg (mean)	*MFCC avg (mean)*	*MFCC 2 (mean)*
MFCC 1 (mean)	*MFCC 1 (mean)*	*ZCR (std)*
MFCC 2 (mean)	*MFCC 2 (mean)*	*RMSE (std)*
MFCC 3 (mean)	*MFCC 3 (mean)*	*MFCC avg (std)*
MFCC 6 (mean)	*MFCC 6 (mean)*	*Tonnetz avg (std)*
MFCC 8 (mean)	*MFCC 8 (mean)*	*Tonnetz 2 (std)*
MFCC 10 (mean)	*MFCC 10 (mean)*	*Tonnetz 4 (std)*
MFCC 13 (mean)	*MFCC 13 (mean)*	*Tonnetz 5 (std)*
RMSE (std)	*RMSE (std)*	
MFCC avg (std)	*MFCC avg (std)*	
MFCC 3 (std)	*MFCC 3 (std)*	
Tonnetz avg (std)	*Tonnetz avg (std)*	
Tonnetz 1 (std)	*Tonnetz 1 (std)*	
Tonnetz 2 (std)	*Tonnetz 2 (std)*	
Tonnetz 3 (std)	*Tonnetz 3 (std)*	
Tonnetz 4 (std)	*Tonnetz 4 (std)*	
Tonnetz 5 (std)	*Tonnetz 5 (std)*	
	MFCC 4 (mean)	
	MFCC 4 (std)	

Appendix C - KNN Confusion Matrices for Other Test Groups

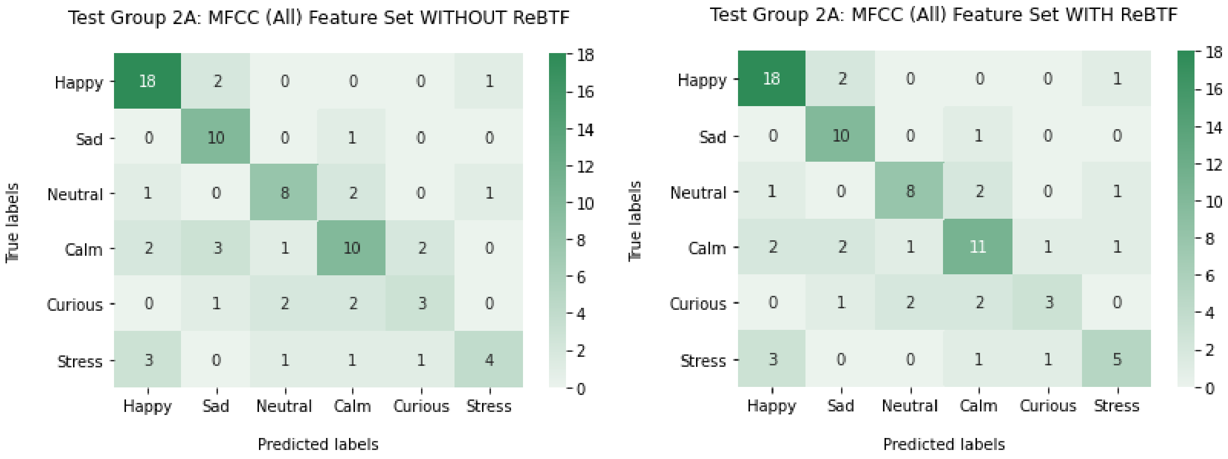


Figure C1: Confusion matrices for Test Group 2A with (left) and without (right) the ReBTF

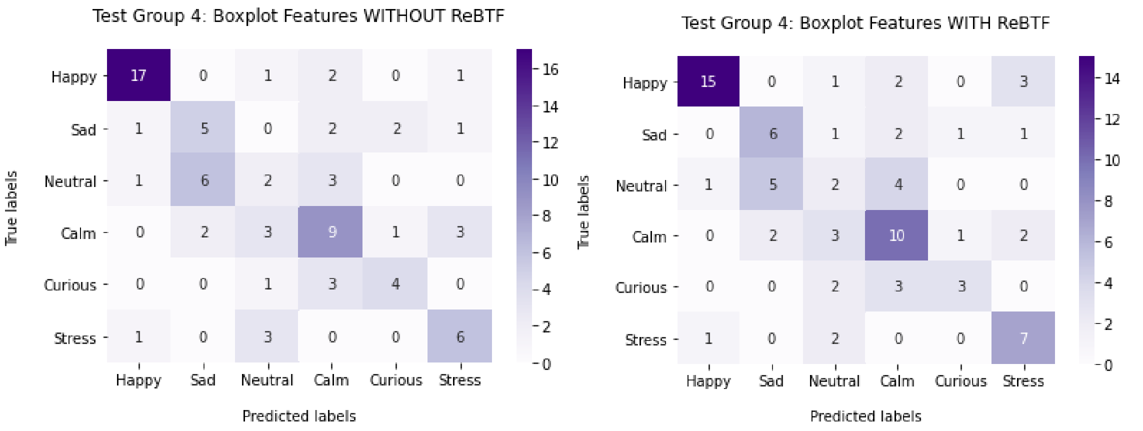


Figure C2: Confusion matrices for Test Group 4 with (left) and without (right) the ReBTF

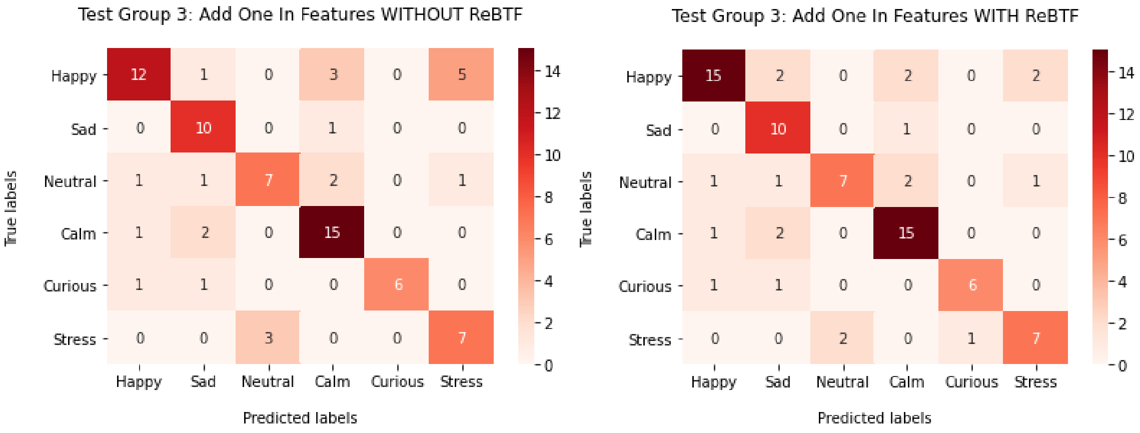


Figure C3: Confusion matrices for Test Group 4 with (left) and without (right) the ReBTF

Appendix D - MLP and CNN Confusion Matrices for Other Test Groups:

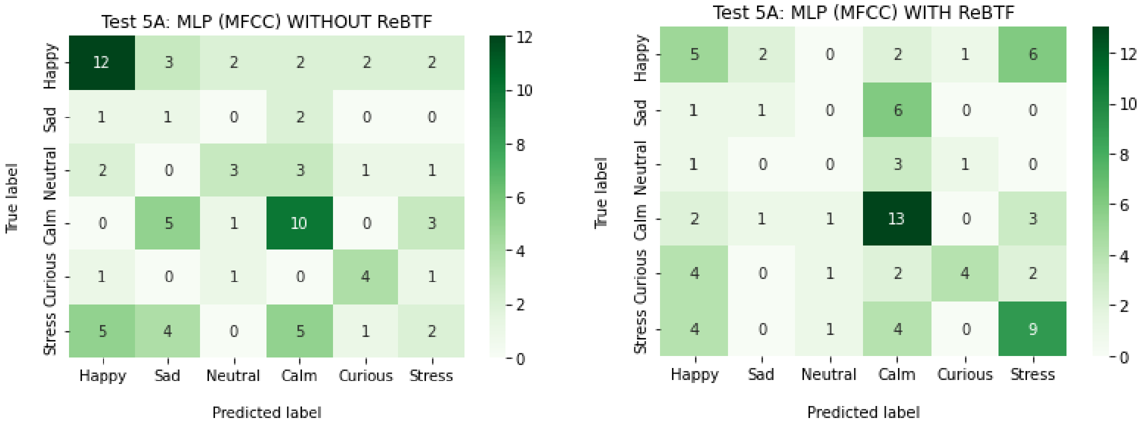


Figure D1: Confusion matrices for Test Group 5A with and without the ReBTF

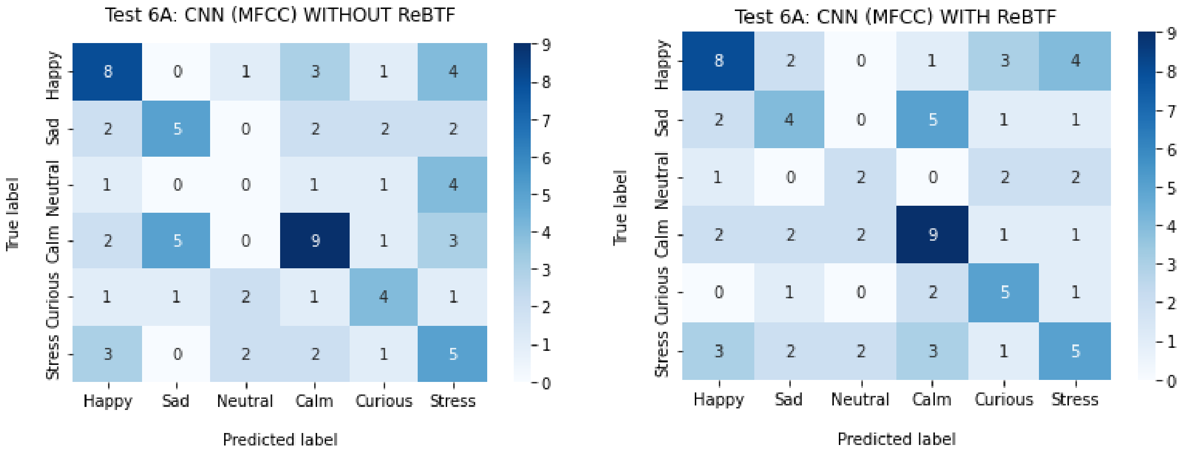


Figure D2: Confusion matrices for Test Group 6A with and without the ReBTF

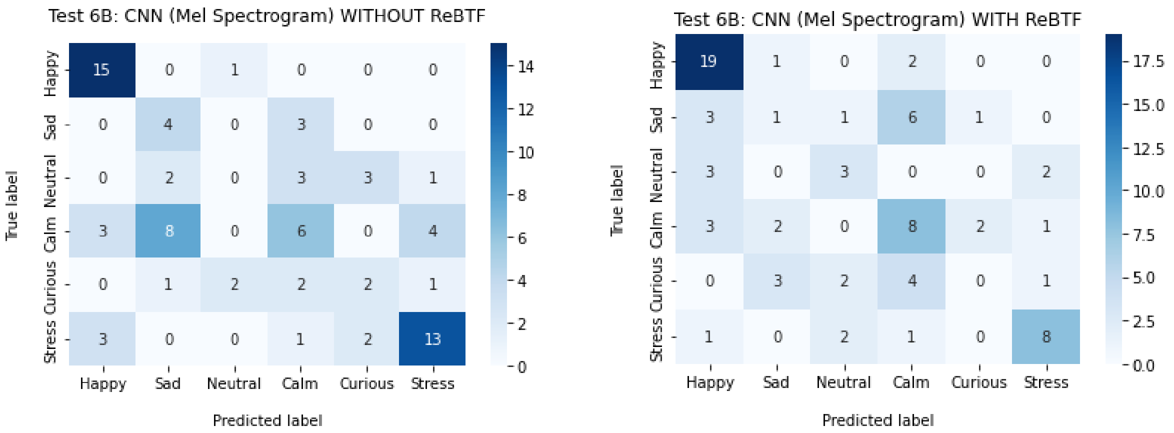


Figure D3: Confusion matrices for Test Group 6B with and without the ReBTF

Appendix E: (KNN) Percent change in precision when ReBTF is included for the 4 best performing test groups.

Emotion	Test Group 1A	Test Group 2A	Test Group 3	Test Group 4
Happy	-2.60%	0%	+ 3.81%	+ 4.17%
Sad	0%	+ 6.67%	+ 20.0%	- 6.25%
Neutral	0%	+ 9.09%	- 9.09%	+ 11.11%
Calm	- 6.67%	+ 3.52%	+ 0.53%	+ 5.00%
Curious	0%	+ 20.0%	+ 5.00%	- 14.3%
Stress	+22%	- 6.25%	- 1.28%	+ 30.0%

Note: Green highlight indicates an improved result, red highlight indicates a worsened result, and a white highlight indicates no change.