Interpretable speaker diarisation for child language environment analysis

F Tembo

orcid.org/0000-0003-1454-3164

Dissertation accepted in fulfilment of the requirements for the degree *Master of Science in Engineering Sciences with Computer and Electronic Engineering* at the North-West University

Supervisor: Prof PA van Vuuren

Graduation: August 2023

Student number: 36688983

DECLARATION

I, Francis Anesu Phineas Tembo hereby declare that the dissertation entitled “Interpretable speaker diarisation for child language environment analysis” is my original work and has not already been submitted to any other university or institution for examination.

_____

Francis Anesu Phineas Tembo

Student Number: 36688983

Signed on the 21st day of November 2022 at Potchefstroom, North West, South Africa.

ABSTRACT

Speaker diarisation is an essential upstream processing step in many speech processing applications. In child natural language environment analysis, it is used to partition long-format recordings into segments with respect to each speaker in the audio stream. After the daylong recording is partitioned into speaker segments, speech recognition algorithms can be used to determine the quality and quantity of speech input the child has experienced from their mother, father, siblings, or other caregivers. This information can be used to track a child's language understanding allowing researchers to investigate various topics related to their development.

Long-format recordings present the least invasive way to capture speech data in a way that does not affect people's behaviour, making them suitable to pick up interactions in all potential language use scenarios. Although the importance of this technology is apparent, accessibility to these tools has been a major hurdle for widespread adoption. The reason behind this is that speech technology involving child speech is quite hard to develop. Infants and children do not follow traditional speaking patterns, and the data is much more difficult and expensive to collect and annotate.

In light of these limitations, this study aims to investigate how interpretability, a technique for explaining machine learning models, can be used to develop a diarisation model. This is accomplished by first exploring the literature on the topic of interpretability, analysing the broad desiderata related to it, and developing a taxonomy that consolidates the categorisation of interpretability methodologies. Since speaker diarisation is a speech processing task, interpretability is explored in the context of speech processing systems. The current literature on this topic suggests that the choice of interpretability techniques is dictated by the class of the underlying machine learning model and the nature of the input speech to which the model is trained.

The study explores gender perception through the classification of speech samples in order to establish a foundation for how interpretability may be investigated for the diarisation model. This is investigated through two classes of deep neural network architectures. One network accepts input speech in the form of spectrograms in the time-frequency domain. The other network receives the input speech as raw waveforms in its time-domain representation.

Interpreting these models requires that the explanations convey that the model has latched on to the idea of gender discriminative information, such as a difference in pitch and formant frequencies. Explanations for the spectrogram-trained model are generated using the expected gradients method. The attributions on the spectrogram highlight that the model focuses on the formants and the distances between them. The interpretation of the raw waveform model is done by determining the cumulative frequency of the filters in the first convolutional layer. This response is shown to favour spectral information relating to discriminative speaker information.

The diarisation problem is investigated in a voice type classification setting, taking into consideration the class of speakers generally expected in the child's natural language environment. Both the proposed VGG16 model trained on spectrograms and the SampleCNN model trained on raw waveforms are competitive compared to the existing state-of-the-art benchmark. The interpretations of these models are generated similarly to the gender perception problem, in which both models give an understanding of information used to differentiate between speakers.

Although the VGG16 model performed better than the SampleCNN model, and its explanations are more human-readable, the interpretations of the VGG16 model via the expected gradients interpretability method did not yield any information as to how the model may be improved. On the other hand, the SampleCNN model has a similar performance to the current benchmark. The filters' cumulative frequency response in the first convolutional layer can be manipulated by varying the filter sizes to favour speaker-related spectral information. With respect to the input space, the expected gradients interpretations on the VGG16 provide local to semi-local interpretability. In contrast, the SampleCNN's cumulative response provides global interpretability and is a direct interpretation of the internal representations learned by the network.

When the diarisation model's decisions and or learnt representations can be explained and confirmed using domain-specific speaker discriminative information, the model can be regarded as interpretable. Depending on the nature of the diarisation model, these interpretations can be used in the same sense to actively influence the model's architecture or give suggestions regarding the nature of the input to improve performance.

KEYWORDS: diarisation, interpretability, deep neural network, convolutional neural network, bidirectional long short-term memory, fundamental frequency, formant

ACKNOWLEDGEMENTS

My most tremendous gratitude to Prof. PA van Vuuren and Niel Swart for their mentorship and support throughout my master's studies.

Most importantly, without my partner Yolanda, none of this would have been possible.

TABLE OF CONTENTS

1. INTRODUCTION 1

1.1. BACKGROUND TO STUDY 1

1.1.1. Nature of daylong recordings 1

1.1.2. LENA™ 2

1.1.3. ACLEW DiViMe 3

1.2. CORPORA OF CHILD-CENTERED DAYLONG RECORDINGS 4

1.3. RECENT FOCUS 6

1.3.1. End-to-End Diarisation Systems 6

1.3.2. Speaker Embeddings for Diarisation Systems 7

1.4. INTERPRETABILITY 7

1.5. PROBLEM STATEMENT 9

1.6. RESEARCH QUESTION AND SUBQUESTIONS 9

1.6.1. Research question 9

1.6.2. Research sub-questions 9

1.7. RESEARCH METHODOLOGY 9

1.8. CRITERIA FOR VALIDATION 10

1.9. DISSERTATION LAYOUT 12

2. LITERATURE REVIEW 14

2.1. INTERPRETABILITY 14

2.2. IMPORTANCE OF INTERPRETABILITY TO THE DEVELOPMENT LIFE CYCLE OF MACHINE LEARNING MODELS 15

2.3. PROPERTIES OF AN INTERPRETABLE MODEL 16

2.3.1. Transparency 16
2.3.2. Post-hoc interpretability 17
2.4. TAXONOMY OF INTERPRETABILITY METHODS 17
2.4.1. White box vs black box 18
2.4.2. Intrinsic vs post-hoc 18
2.4.3. Global vs local 18
2.4.4. Passive vs active 18
2.4.5. Type of explanations 18
2.4.6. Defining an interpretability taxonomy 19
2.5. ATTRIBUTION METHODS 20
2.5.1. Model agnostic attribution methods 20
2.5.2. Model-specific attribution methods 24
2.5.3. Comparing attribution methods 30
2.6. HIDDEN SEMANTICS 31
2.6.1. The human brain and semantic mapping 32
2.6.2. Visualising individual neurons and semantic dictionaries 33
2.7. INTERPRETABILITY METHODS FOR SPEECH PROCESSING APPLICATIONS 34
2.7.1. Explaining models trained on raw waveforms 37
2.7.2. Interpretable architectures for raw waveform modelling. 39
2.7.3. Explaining models trained on time-frequency representations 41
2.8. DISCUSSION 42
3. GENDER PERCEPTION 44
3.1. PROBLEM DEFINITION 44

3.2. ACOUSTIC DIFFERENCES BETWEEN MALE AND FEMALE SPEECH 44
3.3. METHOD 46
3.4. DATASET 47
3.5. WHITE-BOX MODEL 47
3.5.1. Feature extraction for white-box model 47
3.5.2. Training the white-box model 49
3.5.3. Performance of the white-box model 50
3.5.4. Interpretation of the white-box model 52
3.5.5. Assessing the white-box model approach 56
3.6. CLASSIFICATION OF RAW WAVEFORMS 57
3.6.1. Classification performance on raw waveforms 60
3.6.2. SampleCNN model interpretation 62
3.7. CLASSIFICATION THROUGH SPECTROGRAMS 64
3.7.1. Classification performance on spectrograms 66
3.7.2. Spectrogram model interpretation 68
3.8. DISCUSSION 71
4. VOICE TYPE CLASSIFICATION 73
4.1. PROBLEM DESCRIPTION 73
4.2. ACOUSTIC DIFFERENCES BETWEEN ADULT AND CHILD SPEECH 74
4.3. DATASET 75
4.4. BASELINE SETUP 76
4.5. PROPOSED MODELS 77
4.6. EXPERIMENTAL RESULTS 78

4.7. PERFORMANCE ON LENA RECORDED DATA. 80

4.8. MODEL INTERPRETATIONS 81

4.9. IMPROVING THE PERFORMANCE OF THE SAMPLECNN ARCHITECTURE 85

4.10. IMPROVING THE PERFORMANCE OF THE VGG16 ARCHITECTURE 89

4.11. IMPROVING PERFORMANCE ON THE LENA RECORDED DATA 90

4.12. DISCUSSION 96

5. CONCLUSION AND RECOMMENDATIONS 98

5.1. CONCLUSION 98

5.2. RECOMMENDATIONS 99

REFERENCES 100

APPENDIX A 111

A.1. FORMULATIONS OF EXTRACTED SPEECH FEATURES. 111

A.2. DETAILED INFORMATION ON THE CHILD SPEECH DATASET. 114

APPENDIX B 115

B.1 CONVOLUTIONAL NEURAL NETWORKS 115

B.2 BATCH NORMALISATION 117

B.3 RECURRENT NEURAL NETWORKS 118

APPENDIX C 122

C.1. RULE EXTRACTION 122

C.1.1. Decompositional Approaches 122

C.1.2. Pedagogical Approaches 123

C.2. EXPLANATION BY EXAMPLE 123

LIST OF FIGURES

Figure 1.1: How data flows through the different DiViMe tools. 3

Figure 1.3: Producing human-readable explanations from a black box machine learning model through interpretability. 8

Figure 2.1: Loan approval process using traditional scoring function, a human evaluator is included in the decision making. 14

Figure 2.2: Loan approval process via a machine learning model with no human evaluator. 14

Figure 2.3: Lipton's framework of interpretability. 16

Figure 2.4: Organisation of interpretability methods. 19

Figure 2.5: a. Irrelevant segments greyed out. b. Positive/Negative overlayed. c. LIME heatmap. 21

Figure 2.6: Force plot of Shapley values for a single instance. 23

Figure 2.7: SHAP summary plot ranking the features in order of decreasing importance. 23

Figure 2.8: a. Input image. b. Saliency map. The colour bar corresponds to relevance scores 25

Figure 2.9: *a*. Input image. b. Guided backpropagation pixel attribution map. 26

Figure 2.10: Results of Grad-CAM for digits six and seven, respectively. 27

Figure 2.11: LRP heatmaps for the numbers zero and one. 27

Figure 2.12: Integrated Gradients heatmaps for the numbers zero and one. 28

Figure 2.13: Model explanations for MNIST digits. 30

Figure 2.14: SHAP explanations extended for the hidden layer. 32

Figure 2.15: Example of a semantic map of the human brain. 32

Figure 2.16: Visualisation of the activated neurons 33

Figure 2.17: Possible aggregation of neurons as a. Individual neurons. b. Spatial Activations. c. Channel activations. iii. Neuron Groups. 34

Figure 2.18: Raw waveform and amplitude envelope. 35

Figure 2.19: Spectrogram obtained by applying the STFT to a speech sample. 35

Figure 2.20: MFCC visualisation. 36

Figure 2.21: Cumulative frequency responses for convolutional layers with 300 and 30 kernel lengths, respectively. 38

Figure 2.22: Input speech signal and the relevance signal. 38

Figure 2.23: Cumulative frequency responses of CNN and SincNet on speaker identification. 40

Figure 2.24*:* Example of filters learned by the conventional CNN and SincNet 41

Figure 3.1: Upper respiratory system of a human showing the position of the 44

Figure 3.2: Frequency spectra for male and female speech samples for the word zero. 45

Figure 3.3: Raw waveforms and spectrograms for the utterance of the word zero for male and female speakers, respectively. 46

Figure 3.4: PCA and TSNE clustering on the extracted feature dataset. 49

Figure 3.5: Illustration of k-fold cross-validation 49

Figure 3.6: Mean cross-validation accuracy and training accuracy per decision tree on training data 51

Figure 3.7: Confusion matrix for decision tree. 51

Figure 3.8: Truncated visualisation of the decision tree. 52

Figure 3.9: Force plot explanations for a female class prediction. 53

Figure 3.10: Bar plot showing the mean absolute SHAP values for each variable. 53

Figure 3.11: Summary plot showing the magnitude of each feature and its impact on the decision tree's prediction across the whole dataset. 54

Figure 3.12: Explanation for misclassifications. 55

Figure 3.13: Distributions of the male and female mean fundamental frequencies, respectively. 55

Figure 3.14: Confusion matrix for decision tree trained on fewer features. 56

Figure 3.15: Difference between filter shapes of SincNet layers and SampleCNN layers. 57

Figure 3.16: The different layers that make up a unit of a basic convolutional block. 57

Figure 3.17: The different layers that constitute a squeeze excitation block. 58

Figure 3.18: SampleCNN architectures with (a) Basic blocks and (b) Squeeze excitation blocks. 59

Figure 3.19: SampleCNN architectures with BiLSTM layers for sequence learning. 59

Figure 3.20: Box plots for the validation score for each model across all four folds. 60

Figure 3.21: Confusion matrix for the SE block with BiLSTM model. 61

Figure 3.22: Scatter plot representation of one second's worth of male speech. Each of the 16000 data points is colour-coded with Shapley value attributions for the true class label. 62

Figure 3.23: Scatter plot representation of one second's worth of female speech. Each of the 16000 data points is colour-coded with Shapley value attributions for the true class label. 63

Figure 3.24: Cumulative frequency response. 64

Figure 3.25: Bottleneck residual block. 65

Figure 3.26: (a) ResNet like architecture (b) ResNet like architecture with BiLSTMs 65

Figure 3.27: Transformation from raw waveform to spectrogram representation via the FFT. 66

Figure 3.28: Box plots for the validation score for each model across all four folds 67

Figure 3.29: Confusion matrix for the ResNet block with BiLSTM model. 67

Figure 3.30: Expected gradients explanation for a correct prediction by the ResNet block architecture 68

Figure 3.31: Expected gradients explanation for a correct prediction by the ResNet Block with BiLSTM architecture 69

Figure 3.32: Attribution maps for the ResNet block architecture's prediction on the spoken digit zero for female (top) and male (bottom) spectrograms. 70

Figure 4.1: Illustration of the voice type classification process. 73

Figure 4.2: The architecture for the baseline state of the art model. 77

Figure 4.3: The proposed architecture for the voice type classifier. The VGG16 backbone acts as a feature extractor. 77

Figure 4.4: Box plots of the performance for each model on the repeated experiments. 79

Figure 4.5: Confusion matrix for the VGG16 multilabel model. 80

Figure 4.6: Confusion matrix for the VGG16 multiclass model 81

Figure 4.7: Cumulative frequency response for the multilabel SampleCNN model. 82

Figure 4.8: Expected gradients explanations for the key-child target prediction. 82

Figure 4.9: Hidden semantic explanations for the first, third, fifth, eighth and eleventh convolutional layers in the VGG16 model. 83

Figure 4.10: Density distributions showing the ranges of the estimated (a) F0 and (b) the formants. 84

Figure 4.11: (a) Input spectrogram with axes. (b) Input spectrogram with expected gradients attributions 85

Figure 4.12: Cumulative frequency responses of 30 and 300 filter lengths models, respectively. 86

Figure 4.13: Cumulative frequency response of the 30-kernel length architecture 87

Figure 4.14: Cumulative frequency responses of the 300-kernel length architecture 87

Figure 4.15: Extended SE BiLSTM architecture.....88

Figure 4.16: Cumulative frequency responses of 300 and 3 filter lengths, respectively.....89

Figure 4.17: Extended VGG16 architecture. 89

Figure 4.18: Example of the Siamese network architecture. 91

Figure 4.19: Proposed Siamese network architecture for speaker type similarity estimation. 92

Figure 4.20: Confusion matrix for Siamese model.....92

Figure 4.21: Integration of voice type classifier with voice type estimation.....93

Figure 4.22: Confusion matrixes of (a) The combined model (b) the Original single classifier. 94

Figure 4.23: Visualisation of PCA and TSNE clustering on spectrogram images. 95

Figure 4.24: Visualisation of PCA and TSNE clustering on the extracted speech features 95

Figure 4.25: Visualisation of PCA and TSNE clustering on the VGG16 embeddings. 96

LIST OF TABLES

Table 1.1: Tools available in the DiViMe virtual machine and their functional categories. 3

Table 1.2: Corpora hosted on the Databrary and HomeBank repositories. 5

Table 1.3: The confusion matrix. 11

Table 1.4: Description of performance metrics. 11

Table 3.1: Description of extracted speech features. 48

Table 3.2: Precision, recall and f1-scores for the decision tree. 51

Table 3.3: Classification performance of SampleCNN architectures in terms of mean and standard deviations across all four folds. 60

Table 3.4: Precision, recall and f1-scores for the SE block with BiLSTM model. 62

Table 3.6: Classification performance of ResNet-like architectures in terms of mean and standard deviations across all four folds. 66

Table 3.7: Precision, recall, and f1-scores for the ResNet block with BiLSTM model. 68

Table 3.8: Estimated voice source parameters for AudioMNIST samples 70

Table 4.1: Average formant frequency for men, children and women. 74

Table 4.2: Average formant frequency for different age groups of children. 74

Table 4.3: Description of each source included in the dataset. 75

Table 4.4: Training configurations for each model. 78

Table 4.5: Classification performance of voice type classification models 79

Table 4.6: Classification performance of voice type classification models on LENA recorded data. 81

Table 4.7: Estimated voice source parameters for the key-child speaker 84

LIST OF COMMON ABBREVIATIONS

BiLSTM	Bidirectional long short-term memory
CNN	Convolutional neural network
DNN	Deep neural network
KCHI	Key child
LENA	Language ENvironment Analysis
LSTM	Long short-term memory
OCH	Other child
MFCC	Mel-frequency cepstral coefficients
NON	Non speech
ResNet	Residual network
RNN	Recurrent neural network
SE	Squeeze excitation
SHAP	Shapley additive explanations

CHAPTER I

1. INTRODUCTION

A child's natural language environment plays an essential role in developing language and vocabulary understanding. The linguistic skills gained during the early stages of infancy can be used to estimate a child's future cognitive ability [1]. The advent of automated speech processing technology has allowed researchers to analyse hours of naturalistic audio recordings from children's language environments in order to investigate and understand the activity areas that stimulate verbal engagement.

Previously researchers had to rely on rather complex methods to turn their hypotheses about the mechanisms that influence processing linguistic input into reproducible well-grounded knowledge. For instance, the research by Kuhl [2] investigated how social interactions act as gates that initiate language learning. The study shows how through a social context infants at nine months of age can learn a new language after less than five hours of exposure to it. The research used a combination of advanced and non-invasive brain activity measurement tools to track the children's response to language stimulation. In contrast, a much more recent study [3] successfully explored a similar topic leveraging automatic speech processing technology. Demonstrating how reasonably simple but powerful speech processing systems can change the way people conduct research in a positive manner.

For caregivers, this technology provides feedback on how well they are doing in regards to eliciting talk from their child or providing language inputs and cognitive stimulations that improve their child's literacy outcomes. Access to this information is crucial as a caregiver's language input is one of the most reliable predictors of a child's future language ability. Furthermore, most parents are not well versed in this knowledge and, worst of all, overestimate how much they talk to their children [4].

The tools provide information on how to set trajectories for a child's language development or help pick up onset signs of language disorders at an early stage. While the technology has grown in popularity over the years, the primary hurdle for adoption has been the accessibility of these tools to researchers and caregivers.

1.1. BACKGROUND TO STUDY

1.1.1. Nature of daylong recordings

Naturalistic daylong recordings of a child's linguistic environment provide insight into the dynamics of what a child hears in a typical day as they move between different acoustic environments. They document the language experience of a child as well as the associated development. Corpora with such data present other levels of difficulty for speech processing systems as it contains speech that is entirely foreign to them [5]. Several factors contribute to this unfamiliarity, which are [5][6]:

- Paralinguistic vocalisations like laughter or cooing are prominent for children in the early stages of infancy. Even though these are non-speech, they prompt communicative reactions.
- Child speech has a higher variability between speakers than adult speech. This is because children have a smaller vocal system that changes as they grow, making it harder to model their acoustic and linguistic characteristics.
- Infant speech is spontaneous and has a high level of overlap due to the lack of social or communicative experience.

- Adult speech when talking to children differs from adult-to-adult conversations. It consists of shorter utterances with higher pitch excursions.

These hurdles are influential enough to require a different paradigm of thinking when formulating algorithms for dealing with such recordings since the methods that are developed for adult speech fail against child speech [7]. This difficulty in dealing with child speech is one of the significant reasons the application of this technology to children has significantly lagged in comparison to adult speech [8].

1.1.2. LENA™

The Language ENvironment Analysis (LENA™) is a proprietary automatic speech processing system for collecting and analysing daylong recordings from a child's linguistic environment. The system consists of a body-worn audio recorder and an automatic speech processing platform [9]. The children wear specialised clothing that houses the recorder, and as they go through their daily activities, the recorder collects raw audio recordings. For the system's current iteration, the captured data on the recorder is uploaded to a cloud platform that embodies an automatic speech processing pipeline.

On the cloud platform, the uploaded raw audio data goes through numerous steps through the automated processing pipeline [10][11]:

1. The first step includes speech diarisation of the recording into mutually exclusive categories: adult female speech, adult male speech, key-child speech vocalisations, other-child vocalisations, noise, and electronic and overlapping sounds.
2. The child of interest (key-child) vocalisations are further segmented into speech and non-speech sounds. In this pipeline, the speech is not only limited to words but other non-verbal communication such as babbling and cooing. Whilst the non-speech category encompasses vegetative sounds like burping and emotion reactional sounds like crying.
3. A linguistic unit counter estimates how much has been said in each adult speaker segment.
4. The data is further analysed to give an estimation of the conversational turns between the adult and the key-child.

The LENA™ system stands as the go-to gold standard for most researchers conducting studies involving child speech. Due to its ease of use, numerous studies have been conducted using this technology, studies investigating the influence of viewing television on children [12], relationships between socioeconomic status and future language ability [15][16], the nature of autism spectrum disorders [15] and the patterns of typical development [16]. Consequently, because of LENA™ 's popularity, multiple efforts have been made to study the system's reliability [12][19].

Since LENA™ works straight out of the box, it has been quickly adopted by parents and other caregivers. The system gives concise information about how much language input the child is being exposed to, as well as other elements such as TV and electronic sounds in his/her audio environment. Furthermore, LENA provides quantitative and qualitative reports which help assess a child's language development over time.

Despite the popularity of the LENA™ system, it still imposes some limiting factors on research. A major criticism of the system is that its models might have overfitted to being solely trained on American English recordings gathered by their specific recorder [5][18]. This imposes limitations on generalisation for foreign languages and multilingual environments where the language changes across different settings. Furthermore, the asking price for this system is quite considerable at USD

300 per recording unit and USD 12 000 per software license [19][5]. Whilst an argument can be constructed to justify the price, LENA™ is still proprietary and fundamentally a closed system. This makes the LENA™ a black box where there is no way to improve upon their models incrementally, and the user has no information about the design decisions and how they affect performance.

1.1.3. ACLEW DiViMe

ACLEW DiViMe is a collection of open-source tools for analysing recordings from children's language environments. The system is a popular alternative to the LENA™ [19][5]. Here researchers can package their speech processing algorithms for widespread reuse. The tools are bundled up within a virtual machine, and whilst there is a prescribed data flow, as shown in Figure 1.1, these tools can be used independently of one another or combined to tackle a specific problem.

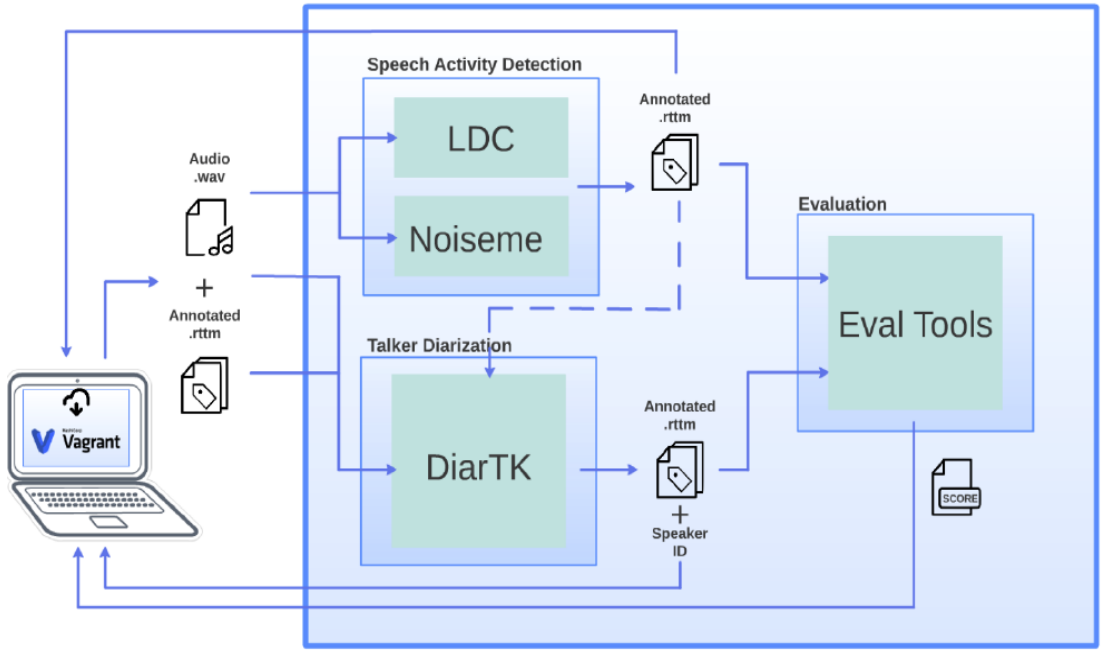

Figure 1.1: How data flows through the different DiViMe tools.
Source: Adapted from [20].

The table below categorises each tool available with the DiViMe virtual machine [21]:

Table 1.1: Tools available in the DiViMe virtual machine and their functional categories.

Tool	Category
Noisemes	speech activity detection
OpenSMILE	speech activity detection
Threshold-Optimized Combo	speech activity detection
DiarTK	speaker diarisation
Yunitate	speaker diarisation
Vocal Maturity Classifier	linguistic unit estimator
Word Count Estimator	linguistic unit estimator

The speech activity detection algorithms detect the presence of human speech for an audio segment. Typically, in a speech processing pipeline a speech activity detector is a preprocessing step used to determine whether or not further speech-related information should be inferred from the audio sample. Linguistic unit estimators are algorithms that extract and quantify linguistic input from speech data such as the number of words, syllables or phonemes.

As indicated in Table 1.1, multiple tools tackle the same category problem, these tools have fundamentally different approaches, and the performance of each tool for a specific task may vary. This gives users the flexibility to pick the right set of tools suitable for their specific application.

The open-source approach of the DiViMe system is a clear positive compared to the closed system nature of the LENA™ system. It has more flexible tools that can be used on recordings gathered by non-LENA™ hardware, and it is open to improvement by other researchers. There is evidence that these systems output roughly the same performance when compared on the same corpora [20]. However, both solutions were proven to fail when applied to non-English recordings of Tsimané children of lowland Bolivia [5].

Even though DiViMe is free, the LENA™ system continues to maintain an edge against it despite the considerable asking price. LENA™ works straight out of the box with its mature, well-built hardware and easy to use graphical user interface. The DiViMe is entirely void of a visual interface and requires basic knowledge of Unix shell commands which may intimidate users [19].

1.2. CORPORA OF CHILD-CENTERED DAYLONG RECORDINGS

The usefulness of child-centred recordings for investigating children's language development has encouraged more attention towards gathering corpora of this nature. In addition to introducing new challenges and pushing the limits of current speech processing and acquisition technology, child-centred data has had an enormous impact in a variety of fields associated with child development. In contrast to adult speech, there are fewer publicly available child speech corpora [22]; this might be attributed to ethical and legal implications associated with collecting and distributing such data. However, there are online scientific repositories where researchers can share and access data in accordance with guidelines that respect and protect the research participants.

The three most common repositories for accessing child-centred daylong recordings are the Open Science Framework (OFS), Databrary and HomeBank [19]. All three repositories have varying levels of access where some data is available publicly, limited to members or requires special permission to access. The HomeBank and Databrary solely specialise in the child language experience, whilst OFS is more of a melting pot for different research efforts. To this effect, OFS allows more flexible ways of structuring data, making it difficult for re-users, especially if the authors did not exercise their due diligence in organising the data properly.

Databrary and HomeBank maintain a boilerplate structure to which researchers adhere when structuring and documenting their data. This ensures the conservation of relevant information about the research participants, such as their socioeconomic demographics. Furthermore, HomeBank requires researchers to use a single universal formatting scheme for their annotations which makes the data more appealing to work with [19][23]. The table below details some of the corpora that are hosted on the Databrary and HomeBank repositories.

Table 1.2: Corpora hosted on the Databrary and HomeBank repositories.

Corpus	Language	Source	Access	LENA recorded?
Bergelson SEEDLingS	Mostly English	HomeBank	Membership	Yes
ACLEW-Starter	English, Spanish, Tzeltal	Databrary	Several Levels	Mostly
Paido	English, Greek, and Japanese.	HomeBank	Public	No
Warlaumont	English, Spanish	HomeBank	Membership	Yes
McDivitt	English	HomeBank	Private	Yes
Winnipeg	English, French	HomeBank	Membership	Yes
Lyon	French	HomeBank	Membership	Yes
Tsay	Mandarin	HomeBank	Public	No

It is important to note that some corpora hosted on the HomeBank repository can be accessed on Databrary by registered members. For instance, the SEEDLingS corpus or the Warlaumont corpus constitutes part of the ACLEW-Starter corpus. Depending on the nature of the study, researchers can either collect and formulate their own corpus or extract from pre-existing corpora that fit their subject of investigation.

A child-centred corpus codenamed 'BabyTrain' [11] has been seemingly gaining attention for its versatility in training and evaluating speech technology, even beyond child-related research [24][25][26][27]. The corpus is an amalgam of 8 different corpora summing up to 260 hours' worth of recordings. It inherits most of its data from the HomeBank and Databrary repositories, namely: the ACLEW-Starter, Paido, Lyon, Tsay and Warlaumont corpus. It also includes three other private corpora of interest; the Tsimané corpus from children of lowland Bolivia[5], the multilingual Vanuatu corpus[28], and the Namibia corpus containing recordings of the Yu'|hoan language.

Due to its diversity, the BabyTrain dataset introduces other dimensions of difficulty for speech processing technology. The corpus spans ten different languages, and each language has different phonetic properties which need to be modelled by the system. The Tsimane corpus, which consists of daylong recordings extracted from children of the Amazon, has already been shown to be challenging to handle for both the DiViMe and LENA systems[5]. BabyTrain also contains recordings of the same language taken from children in different countries. For example, the ACLEW-Starter corpus includes recordings of English-speaking families in the US, Canada, and the United Kingdom. Another desirable attribute of this dataset is that in addition to incorporating popular languages like Mandarin, English and French, lesser-known dialects like the Vanuatu and Yu'|hoan languages are included in it. Perhaps, a system developed on such a dataset will have a better chance of handling under-resourced languages like Zulu or Xhosa.

1.3. RECENT FOCUS

1.3.1. End-to-End Diarisation Systems

Two major speech processing tasks are considered when analysing audio data from child natural language environments: speaker diarisation and speech recognition. First, the speaker diarisation will determine who spoke and when by partitioning the audio signal into homogeneous segments with respect to speaker identity. A typical speaker diarisation system is usually a combination of independent components, which include but are not limited to; an automatic speech detection module that separates speech and non-speech segments, a feature extraction module that produces speaker-discriminative embeddings from smaller audio segments and an unsupervised clustering module which will determine the number of speakers and assign a speaker identity to each embedding segment. After determining who spoke and when the speech recognition component will determine what has been said by each speaker.

Traditional speaker diarisation systems have been mainly developed with the approach of unsupervised clustering of speaker embeddings [29][30]. The clustering approach is associated with several problems such as the missing of rapid conversational turns, difficulties in optimisation since it is trained in an unsupervised manner and a failure to deal with speaker overlap [31][32][33]. Recent approaches have adopted a much more appealing end-to-end system architecture. The acoustic features are fed directly into a single deep neural network (DNN), producing the diarisation results.

As shown in figure 1.2, the end-to-end architecture removes the need for the speech activity detection module, the speaker embedding extraction model and the clustering-based speaker identification module [30][33]. The end-to-end network can be directly trained and optimised to reduce errors, whilst the multiple parts of traditional diarisation systems are developed individually and cannot collectively be directly tuned for errors due to incorrect diarisation.

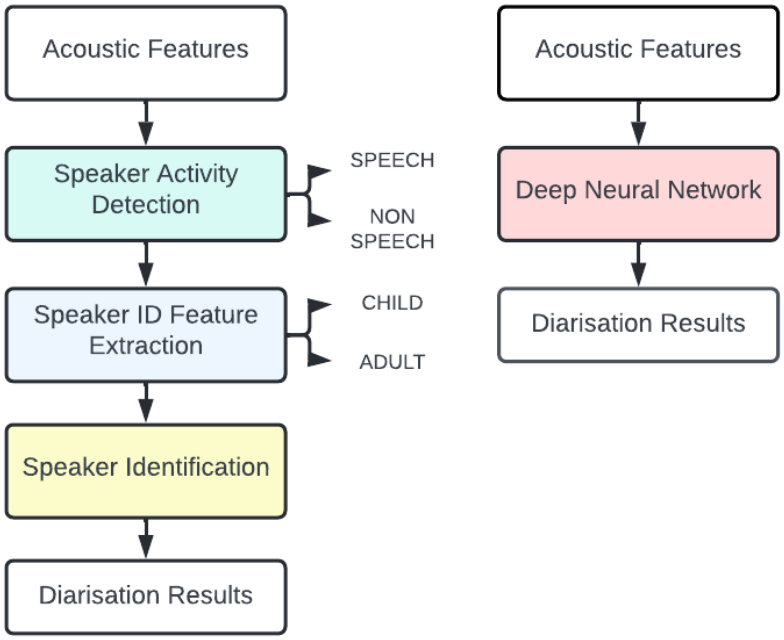

Figure 1.2: Diarization system architectures (left) traditional clustering-based (right) end-to-end deep neural network-based.

End-to-end fully supervised diarisation systems significantly outperform their unsupervised counterparts [30][31][34]. A recent study [11] presented a novel end-to-end voice type classifier that outperformed LENA™ by leveraging specialised convolutional neural networks (CNN) and a stack of bi-directional long-short term memory (LSTM) networks. This system was trained on 260 hours' worth of recordings from the previously mentioned BabyTrain dataset.

The dataset was of particular interest because it contained long-format recordings which were collected using different recorders, including the LENA™ talk pedometer. However, to level the playing field during evaluation, the LENA™ system was only tested on recordings collected by its proprietary recorder. Considering five classes; key-child, other-child, adult female speech, adult male speech and speech, an F-measure between precision and recall was considered for each class. The proposed model significantly outperformed LENA™ for each class. Although the performance differed per class, the average performance across all five classes was still significant compared to that of LENA™.

1.3.2. Speaker Embeddings for Diarisation Systems

Traditional speaker diarisation systems employ identity vectors (i-vectors). These are low-dimensional feature representations of a given utterance [35]. The goal of i-vectors is to reduce large dimensional speaker data into a small vector whilst preserving relevant information. Although one can design a state-of-the-art diarisation system based on i-vectors, they carry a sensitivity to segment duration. This sensitivity imposes limitations when dealing with short segments where it is hard to differentiate between the channel and speaker effects [36].

The current success of deep neural networks has enabled much more effective feature representations to be extracted from audio data. Two common successors to the i-vectors are x-vector and d-vector speaker embeddings. These are higher-level feature representations compared to the i-vectors, and employing such feature representations leads to a significant increase in diarisation performance [34].

X-vectors were developed initially for speaker recognition with the goal of producing a DNN model that can yield low dimensional embeddings that effectively capture speaker characteristics over an entire speech utterance [37]. This is achieved by leveraging time-delayed neural networks as a frame-level feature extractor to capture utterance-level features. On the other hand, D-vectors achieve the same objective by replacing the time-delayed layers with either feed-forward, convolutional, or LSTM layers [38][39]. Both feature representations allow discrimination capability for speaker recognition, verification, and diarisation.

1.4. INTERPRETABILITY

Interpretability represents the extent to which humans can understand the decision logic of a machine learning model. A model is considered to be highly interpretable if humans can fully understand the cause-and-effect relationship between the model's input and the output transformations. The interpretability of a model is usually explored in instances where humans need more information than what is provided by performance metrics. For instance, deep neural networks have been proven to achieve near-human accuracy in several tasks. However, they primarily operate as black boxes where no explanations are given as to why specific features are selected over others during training or inference.

There are situations where this information is required in order to establish accountability for the model's decisions.

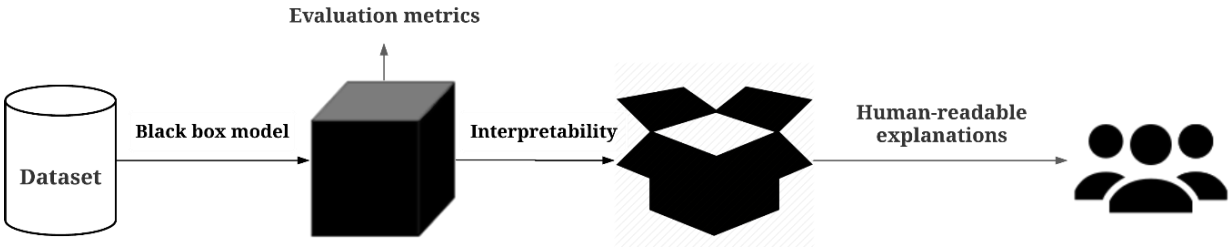

Figure 1.3: Producing human-readable explanations from a black box machine learning model through interpretability.

As to be discussed in the study's literature review, there are various reasons why interpretability can be explored for a particular application. One of the most important reasons is to build confidence that the model has learned the correct representations within the data. The confirmation usually comes in the form of relating the patterns learned by the model with established domain knowledge about the subject matter.

One such example of an interpretability study is that of the *SimpleNet* architecture [40], a deep neural network speech processing architecture whose design is inspired by analysing how convolutional neural networks model raw waveforms from the perspective of audio signal processing. According to this interpretability study, there are two principal contributors to the limitations in performance for speech processing architectures, the aliasing effect due to pooling layers and the unnecessary staking of convolutional layers.

The aliasing effect is an undesirable phenomenon in signal processing that distorts signals during sampling. If aliasing occurs, the original signal is distorted such that it cannot be reconstructed from the resulting samples. Based on the Nyquist sampling theorem, the authors establish how the down-sampling nature of the pooling layer results in aliasing. This means the convolutional layers that proceed the pooling operation are rendered less effective as they receive corrupted frequency components that are not a fair reflection of the original signal.

Contrary to computer vision tasks, the paper suggests that it is unnecessary to stack convolutional layers for raw waveform processing. The basis for this assumption is derived from how the result of stacked convolutional layers (with linear activations) is a linear combination of frequency components of the original signal in the frequency domain. This means that the result of multiple convolution layers is the same as that of one for an audio signal. Increasing the number of layers consequently increases the complexity of optimisation for the network, paving the way for the possibility of exploding or vanishing gradients.

To this effect, the *SimpleNet* architecture has only one convolutional layer and one pooling layer for its front-end layers. This approach outperformed architectures with multiple stacked convolutional and pooling layers in speaker gender and emotion recognition tasks.

Evidently, interpretability mainly played the role of a diagnostic tool in this study. By looking at the response of the filters in the frequency and time domains, the authors were able to explain the spectral

information modelled by the filters in the first convolutional layer. However, the study does not establish a strong relationship between the learned representation and the domain-specific knowledge

1.5. PROBLEM STATEMENT

Most diarisation systems are built on black-box models, from the proprietary LENA™ system, where there is no information at all about the underlying algorithms, to the open-source DiViMe system, where the algorithms are provided but without a proper discussion on how they model data.

Black box models have high predictive power and are able to model hierarchical representations from complex data. However, this comes at the expense of not being able to understand and explain these learned representations. The case can be made for how understanding these learned representations helps design more robust systems.

Typically for improving DNN-based systems, much thought and effort is placed on collecting more training data, training larger networks, or using a new neural network architecture based on much more efficient mathematics. To be fair, these approaches result in significant improvements to the current algorithms. However, it is not always certain that the resources are available for collecting more data or training larger networks.

The *SimpleNet* model discussed in section 1.4 shows how popular conventional neural network designs do not always translate well in other domains. Each layer in the *SimpleNet* architecture is there by design and serves a particular purpose. After establishing the contribution of each layer in the network, the authors could articulate a concise architecture for the model whose structure is purely as a consequence of its function. Granted, the relationship between learnt hierarchical representations and established domain-specific knowledge needs to be established and how that relationship may influence the network's architecture.

Pursuing an interpretable diarisation system for daylong recordings will provide insight into the complexities associated with dealing with such data from the perspective of the model's internal representations and how these representations can be used to design a more robust system.

1.6. RESEARCH QUESTION AND SUBQUESTIONS

1.6.1. Research question

With the rise of utilisation of deep neural networks in processing daylong recordings. To what extent can a diarisation model be interpretable and how does the model's interpretability affect its overall performance and design?

1.6.2. Research sub-questions

1. What are the types of interpretability?
2. What are the principles of interpretability for deep learning models?
3. What are the provide the most intelligible explanations for speech models?
4. How can interpretability be used to improve performance?

1.7. RESEARCH METHODOLOGY

This study consists of empirical research on the interpretability of deep learning-based diarisation models. It will explore how interpretability explanations can be used to understand model decisions and consequently improve their performance. The methods consist of the following elements:

1. Literature review – a literature review to gain an understanding of the landscape of interpretability techniques for deep learning neural networks and their application in speech processing systems.

2. Dataset construction – the curation of the data to be used for investigating the problem. This consists of the preprocessing, creation of the ingestion pipeline and splitting into the training, testing and validation sets.
3. Experimental development:
    a. Model exploration – the exploration of candidate algorithms and the motivation behind their design.
    b. Model development – training, testing and evaluating the candidate models.
    c. Interpretability – interpretability study to produce explanations for known sample labels and the internal propertied of the developed models.
    d. Model refinement – architecture redesign based on interpretability study findings.
4. Discussion and assessment of experimental findings.

Prior to investigating the diarisation of long-format recordings, the problem is first simplified to a gender classification problem. Gender perception is of particular importance because one of the primary objectives of analysing children's natural language environments is to account for the linguistic input provided by the male and female caregivers respectively. The study explores this problem by experimentally developing models that can estimate the gender of a speaker using speech samples and then produce interpretability explanations for the trained models. This stage of the study serves to investigate the feasibility of the requirements formulated by the literature review and refine them for the speaker diarisation problem.

The diarisation of long-format recordings is investigated as a voice-type classification problem. Similar to the gender perception problem, the study experimentally explores different architectures that estimate the class of a speaker using speech samples. This part of the study extends the investigation by using the interpretability explanations of the trained models to suggest how the neural network architectures can be improved. The improvement is meant to push the model towards using the more relevant features of the input speech. These relevant features of speech are most likely the difference in speech acoustics of the target speakers. Finally, the interpretability explanations and performance of the refined models are compared against their predecessors and conclusions are drawn.

1.8. CRITERIA FOR VALIDATION

The success of this study is validated based on the satisfaction of two criteria: 1) the improved performance of the resulting diarisation model compared to the current state-of-the-art benchmark, and 2) the confirmation of the model's learned representations against domain-specific knowledge of speech diarisation.

1.8.1. Performance Metrics

The performances of the resulting model and the current benchmark are evaluated using standard evaluation metrics for supervised machine learning models. The first evaluation method taken into consideration is the confusion matrix. Whilst the matrix is not an evaluation metric itself, it gives a general overview of the performance of the model.

The confusion matrix represents a model's performance in counts of its true-positive (TP), false-positive (FP), true-negative (TN) and false-negative (FN) predictions. As shown in Table 1.3, each column represents a predicted class, whilst the rows represent the actual class. This information is important for evaluating each class's in-domain performance and determining which classes are being confused with one another.

Table 1.3: The confusion matrix.

		Predicted class	
		Positive	Negative
Actual class	Positive	True Positives	False Negatives
	Negative	False Positives	True Negatives

Table 1.4 describes the evaluation metrics and their corresponding formulas in confusion matrix terminology.

Table 1.4: Description of performance metrics.

Metric	**Description**	**Formula**
Accuracy	Overall performance of the model as described by the percentage of correct predictions overall predictions.	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	A measure of the correctness of positive predictions. A lower precision signifies a large number of FP predictions by the model.	$\frac{TP}{TP + FP}$
Recall	The ratio of TP to all positive samples in the dataset.	$\frac{TP}{TP + FN}$
F-measure	A combination of the precision and recall metrics. A desirable F1 score represents the model experiencing low FPs and FNs. The best possible score is one, and the worst possible score is zero.	$\frac{2TP}{2TP + FP + FN}$

1.8.2. Model interpretability

To satisfy the second criterion, an interpretability study is performed to confirm if the patterns learned by the model coincide with the domain knowledge of speech properties. This is primarily achieved by using an interpretability technique to produce 'explanations' of model predictions for samples with known labels. In this context, explanations are a form of description that 'explain' which properties of the input sample that influence the model's decision. If a diarisation model predicts that a speech sample contains speech from a certain class of speaker, an interpretability explanation gives a description of which elements of the speech sample steered the prediction towards that speaker class. This criterion is satisfied if the explanations agree with conclusions that might have been drawn using established knowledge of speech patterns.

The explanations are also produced in the context of introspective descriptions of the internal properties of the resulting diarisation model. The 'internal properties' refer to the inner processing

mechanisms of the model that transform the input to produce the output/decision. For a model based on neural networks, the internal mechanics are the hidden layers between the input layer and the output layer that produces the prediction.

Since interpretability explanations provide insight into what influence the model's decisions. The model can be refined such that its decisions are influenced to a greater extent by the relevant features of the input. This may in turn lead to an improvement in the overall performance of the diarisation model.

1.9. DISSERTATION LAYOUT

The section describes the subsequent structure of this report.

Chapter 2 – Literature Review

This chapter discusses the state of the art in terms of interpretability of deep learning models and how it is implemented in the area of speech processing systems. Consequently, the requirements for interpreting the diarisation model are also established in the chapter.

Chapter 3 – Gender recognition

The third chapter details methods and experiments to solve a gender classification problem. It aims to investigate the gender-specific differences in speech and how this information can be used to explain if a model has learned the correct representations. Furthermore, the fidelity of the interpretability explanations is explored in relation to how well these gender-specific differences can be communicated.

Chapter 4 – Voice type classification

In this chapter, diarisation is investigated as a voice-type classification problem. The approach to gender perception explored in chapter 3 is expanded to consider the speaker discriminative information for each speaker type present in a child's natural language environment. The interpretability explanations are then explored as to how they can provide information to improve upon the diarisation model.

For each model experiment, the results are discussed in relation to the main research objective. The final solution to the research problem is established in this chapter, and it is evaluated using the criteria described in section 1.8.

Chapter 5 – Conclusion and Recommendations

This chapter concludes the study, by giving a summary of the work done and discussing its successes and shortcomings. Furthermore, recommendations will be provided for further development on the subject matter.

Appendix A
This chapter gives complementary information that supports the experimental investigations conducted in this study.

Appendix B
This chapter will detail several key elements of deep learning relevant to this study.

Appendix C

This chapter explores additional interpretability methods**.**

CHAPTER II

2. LITERATURE REVIEW

This chapter explores the landscape of deep learning model interpretability and how it is applied in speech processing applications. The aim is to address the research sub-questions outlined in section 1.6.2. The investigation starts by exploring the general requirements of interpretability and the properties of interpretable models. The findings are used to formulate a categorisation that organises interpretability methods based on the type of explanations they produce and the explicitness of those explanations. The categorisation helps to narrow down the set of interpretability methods to consider for the research problem. Finally, interpretability is explored in the area of speech processing systems. Here the requirements for interpreting the diarisation system are established.

2.1. INTERPRETABILITY

The last decade has seen an unprecedented advance in machine learning research; this has notably been fuelled by breakthroughs in the area of deep learning, from improved architectures for handwritten digit classification to protein folding prediction. Nowadays, it is increasingly common to have these models embedded in systems we interact with daily. It should come as no surprise that as machine learning systems become more widely used in both commercial and academic contexts, there is a growing demand for these models not only to perform well but also to be interpretable.

For instance, let us consider a machine learning model used to recommend whether a client's request for a loan should be approved or denied. In the past, the approval essentially depended on whether the client had a good and lengthy credit history. A scoring algorithm is used to derive a credit score from the client's credit history, and depending on this score, the institution can decide to either accept or deny the request.

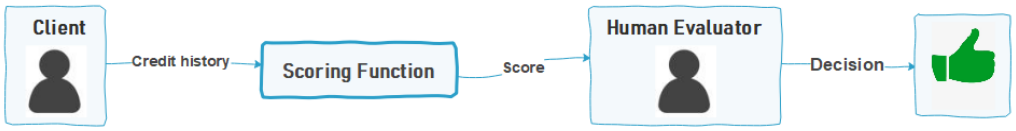

Figure 2.1: Loan approval process using traditional scoring function, a human evaluator is included in the decision making.

More recently, these institutions have developed proprietary machine learning models trained on numerous samples of loan outcomes. The model can estimate with a high level of accuracy and precision whether the client will be able to pay back the loan as per requirements.

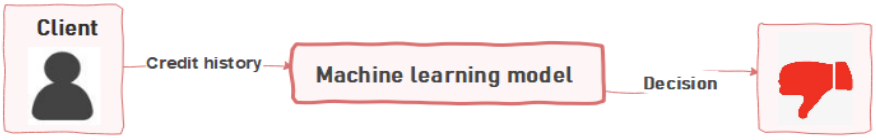

Figure 2.2: Loan approval process via a machine learning model with no human evaluator.

Unlike the credit score, the model's prediction is not simply a guiding score but its final decision. Now it begs the question, how did the model come to this conclusion? How can this prediction be explained to the loan officer or the client? How can the financial institution determine its risk, given that the loan is approved? These are some of the questions that interpretability aims to answer.

An essential aspect of interpretability is that it allows humans to reason about the model's output. Machine learning models are primarily judged on how well they optimise an objective function.

However, because the objective function is usually based solely on performance metrics, it does not capture all the desiderata relevant to the real-world implications of the problem. Desiderata in this context reflects other complementary criteria to which the model should be optimised via interpretability or the fundamental ideas of what one may want from exploring interpretability. Interpretability research for machine learning models has a wide range of requirements, with some of the essential principles being [41][42]:

i. Transferability - investigating potential problems or failures that may arise when there is a change in distribution between the training and testing environments. Currently, humans have a greater capacity to generalise than machine learning models. We can transfer our learned knowledge to unfamiliar situations. However, machine learning models can fail in these situations where the test or deployment environments differ from the conditions in which it was initially trained. This change can result from a natural shift of the data, or the presence of the model itself alters the environment. When the model fails in this way, it invalidates its future predictions, making them not worthwhile. Interpretability can safeguard the model against such potential problems by understanding the conditions that can cause such failures.

ii. Trust - a critical aspect of developing a machine learning model is our confidence when deploying it in real-world scenarios. If one can confirm that the model has learned the correct relationships or, in some cases, that it has discovered new valid associations, then we can trust that the model will perform as intended regarding real scenarios and objectives.

iii. Causality - models learn causal relationships that researchers can test in the real world to formulate or confirm their hypotheses.

iv. Legal and ethical implications - some higher-level machine learning applications require the systems to adhere to ethical and legal standards. Humans must be able to interpret and understand the decision logic of the underlying model to make sure it is not exhibiting any form of bias. For instance, it is crucial for any real-world model that can influence a person's quality of life not to display any form of racial or ethnic discrimination.

2.2. IMPORTANCE OF INTERPRETABILITY TO THE DEVELOPMENT LIFE CYCLE OF MACHINE LEARNING MODELS

An essential aspect of interpretability that is often left out when discussing desiderata is its importance to the development life cycle of machine learning models in general. Most focus is targeted towards the need for interpretability when the predictive performance is not enough to satisfy a problem but not how interpretability can be used to improve it. Deep learning architectures are often developed taking into consideration the available resources. The model can then be scaled up later for better performance if more resources are available. However, it is not always the case that resources are available for collecting more data, training larger networks, or running further experiments with different hyper configurations.

This is especially true for diarisation algorithms targeted for child language environment analysis. It is very costly and time-consuming to collect daylong recordings, and not much data is publicly available. Just as regularisation techniques like dropout and data argumentation are helpful when developing robust architectures in low-resource situations, interpretability can be as instrumental. By providing information about what the model is learning, the influence of parameter changes and the conditions that force the model to make mistakes, one can use this information to actively change the mod-

el's anatomy to improve performance. Furthermore, interpreting the model during development may help uncover imbalances within the training data itself. In cases where the model is known to struggle, a prompt for human intervention can be established.

2.3. PROPERTIES OF AN INTERPRETABLE MODEL

Whilst the area of research for investigating the interpretability of machine learning models has become a very active field over the years, there has not been a consensus on the formal and technical definition of interpretability. As a result, the research on this topic is very diverse and does not always overlap on how to explore the idea of interpretability. Consequently, this has inspired several studies that try to consolidate the idea in frameworks through which we can discuss the properties of interpretable models and the different motivations for pursuing them.

Lipton [41] proposed a very interesting approach on interpretability. This paper proposes a novel framework motivated by the idea that interpretability is not restricted to one single idea but reflects several concepts.

Lipton suggests that interpretability is fragmented into two major categories, transparency and post-hoc explanations. Transparency is concerned with how the model works, and post-hoc interpretations extract knowledge from the trained models. As illustrated in figure 2.3, the two categories comprise multiple sub-dimensions that pertain to distinct aspects of the anatomy of the model.

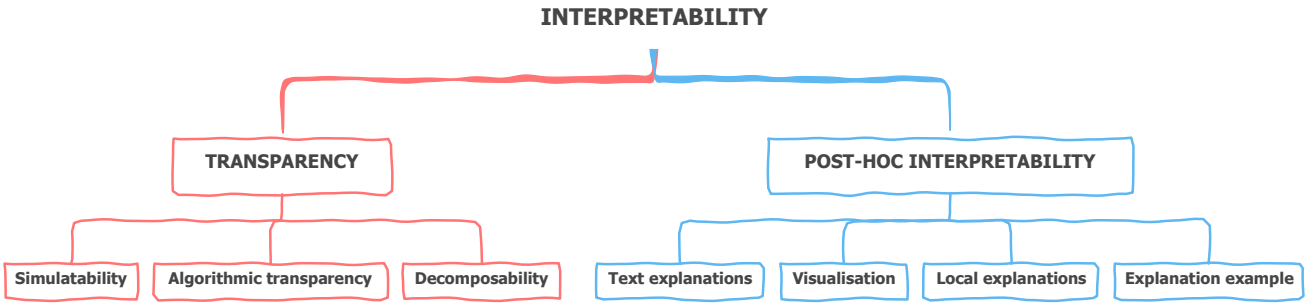

Figure 2.3: Lipton's framework of interpretability.

2.3.1. Transparency

Transparency is made up of three key elements, simulatability, algorithmic transparency, and decomposability. Each of these elements is discussed at different levels of abstraction of the machine learning model. The entire model is considered for simulatability, the training level is considered for algorithmic transparency, and decomposability focuses on the level of the model's parameters.

i. Simulatability - given the model parameters and input data, can a human go through each step of the algorithm's calculations to produce the required prediction? This allows the human counterpart to understand the influence of the training data on the model parameters. Machine learning algorithms such as decision trees and linear models are considered to have a high resolution of simulatability. Therefore, they are generally considered interpretable because their calculations are relatively simple, and it is easy to carry out the necessary steps to produce a prediction. Linear models also have a desirable trait in which the parameters reflect the importance of each input feature. However, whether a machine learning model has this property depends on the model size to a greater extent than the specific type of the model.

ii. Algorithmic Transparency – the ability to explain the learning principle of the algorithm and the guarantees that it gives for convergence. This attribute concerns how the algorithm will

formulate the model, but not the learned model itself or the data it has learned from.

iii. Decomposability – does the model provide an understanding of what is happening at each step of the algorithm or provide an intuitive explanation for other model attributes such as its parameters and input.

If one strictly bases model interpretability in terms of transparency, they are limiting themselves to solutions that have compact and simple models. In most cases, simple and compact models do not perform as well as black-box models (low transparency). Deep neural network architectures are an excellent example; they have limited simulatability, and there is no guarantee that the learning algorithm will converge in some instances. Furthermore, people do not show this form of transparency, nor do we require these types of transparency explanations in order to understand each other thought processes. Just as the exact neural computations in our brains have no implication on how we interpret each other's decisions, we rely on post explanations to explain the logic behind our reasoning. These post explanations are also valid for model interpretability, and there are also referred to as post-hoc interpretability.

2.3.2. Post-hoc interpretability

Post-hoc interpretations are used to provide information on what else the model can confer after it has been trained. There are three distinct dimensions that can be explored:

i. Text explanations – just as we humans verbally justify our reasoning, the model can provide a semantically meaningful justification for its prediction in a natural language format.
ii. Visualisation – a visualisation technique is used to determine what is essential for the model's decision making.
iii. Local explanations – instead of providing a global explanation of the model (entire mapping), local changes are computed for an output class due to a specific input. This is used to derive what influences the model's decision logic locally.
iv. Explanation by example – similar to how people justify their actions via analogies, these explanation types try to investigate if the model can show which examples in the training data are related to input and output.

Whilst post-hoc explanations have the potential to be useful for interpretability research, Lipton [41] urges that they should be accepted with caution. There is no absolute guarantee that they can provide exact mechanistic explanations of how the model works. There is the possibility that resources can be wasted whilst trying to formulate explanations that have a shallow impact on the problem it is trying to solve.

2.4. TAXONOMY OF INTERPRETABILITY METHODS

In an effort to consolidate interpretability techniques under a unified framework, there exist numerous studies that try to provide meaningful categorisation for interpretability methods [41][42][43][44]. A common theme in these studies is the provision of a taxonomy based on some meaningful criterion to which any interpretability method can be classified and discussed. For instance, a simple categorisation is to organise the methods depending on whether they can be applied to any machine learning model, model-agnostic, or limited to a specific class of models, model-specific. This section explores some of the common concepts that emerge when discussing interpretability methods.

2.4.1. White box vs black box

Depending on the opaqueness of a model, it can be either a black box or a white box. A black-box model is one where we do not have any information on its internal mechanics. Only the inputs and outputs are visible but not the transformation between the two. A white box on the other end is the direct opposite of a black box. The model is inherently transparent, following the same sentiment described by Lipton [41] when discussing model transparency. For a white-box model, it is relatively easy to ascertain the important features or how the input is mapped to the output by looking directly at the machine learning process. Examples of white-box models are formulated from concepts such as linear regression and decision trees.

2.4.2. Intrinsic vs post-hoc

Interpretability methods are classified as intrinsic if the interpretability is achieved by imposing limitations on the machine learning model's complexity so that the model is interpretable. This is similar to the transparency property discussed by Lipton [41], which is concerned with how the model works prior to training. Post-hoc interpretability refers to interpretability methods that are applied after training. However, in some cases, post-hoc interpretation methods can be used on intrinsically interpretable models since post-hoc techniques are not a property of the model itself.

2.4.3. Global vs local

With respect to the input space, local interpretability is concerned with interpretations of individual predictions, whilst global interpretability explains the entire decision logic of the model. There also exists an intermediate between the two, semi-local interpretability, where it exhibits a mixture between the two but not in their entirety. To explore these concepts from a more practical perspective, consider the previously mentioned loan predictor system mentioned at the beginning of this chapter. Global interpretability is achieved for this system if we can explain the entire logic behind determining the loan decision. This can be achieved by using something similar to a decision tree or a set of logic rules. Local interpretability, on the other hand, entails explaining individual loan requests. Semi-local interpretability means producing explanations for a group of inputs/loans at a time.

2.4.4. Passive vs active

Active methods introduce some form of intervention prior to training the model to encourage its interpretability. Interventions can be in the form of changes to the training and optimisation algorithms or fine-tunings to the network architecture in the case of deep neural networks. In contrast, passive interpretation methods try to extract knowledge and meaningful patterns from models that have already been trained. By definition, this criterion of passive and active interpretations is similar to Lipton's take on transparency and post-hoc interpretability. Passive interpretability is synonymous with post-hoc interpretability and active interpretability to transparency. However, Lipton's take gives a higher-resolution discussion by further fragmenting the two ideas into subcategories that infer interpretations from different parts of the model.

2.4.5. Type of explanations

Interpretability techniques can be grouped and ranked depending on the type of explanations they produce and how explicit those explanations are. Zhang *et al* [42] introduces four classes of interpretability methods: logic rules, hidden semantics, attribution, and explanation by examples. The

study also looks at how logic rules are the most effective explanations, followed by hidden semantics explanations, attributions, and finally, explanations by example. This section will go over these categories in greater detail.

2.4.6. Defining an interpretability taxonomy

The essential function of any categorisation framework is to help narrow down the search for an applicable interpretability method. The taxonomy relevant to this study is shown in figure 2.4. This is heavily inspired by the categorisation discussed by Zhang *et al* [42]. The focus here is placed on organising the interpretability methods based on the explanatory power of the explanations they produce. Since the primary goal of interpretability is to provide an understanding of the model's predictions. An interpretability method that produces the most expressive explanations is the most desirable. In addition to expressive power, there are other properties of explanations that can be considered, such as [43]:

i. Algorithmic complexity – with respect to computational complexity, how complex is the algorithm producing the explanations?
ii. Portability – this describes the range of models the technique can be used.
iii. Translucency – how much the explanation depends on looking into the mechanistic inner workings of the machine learning model.

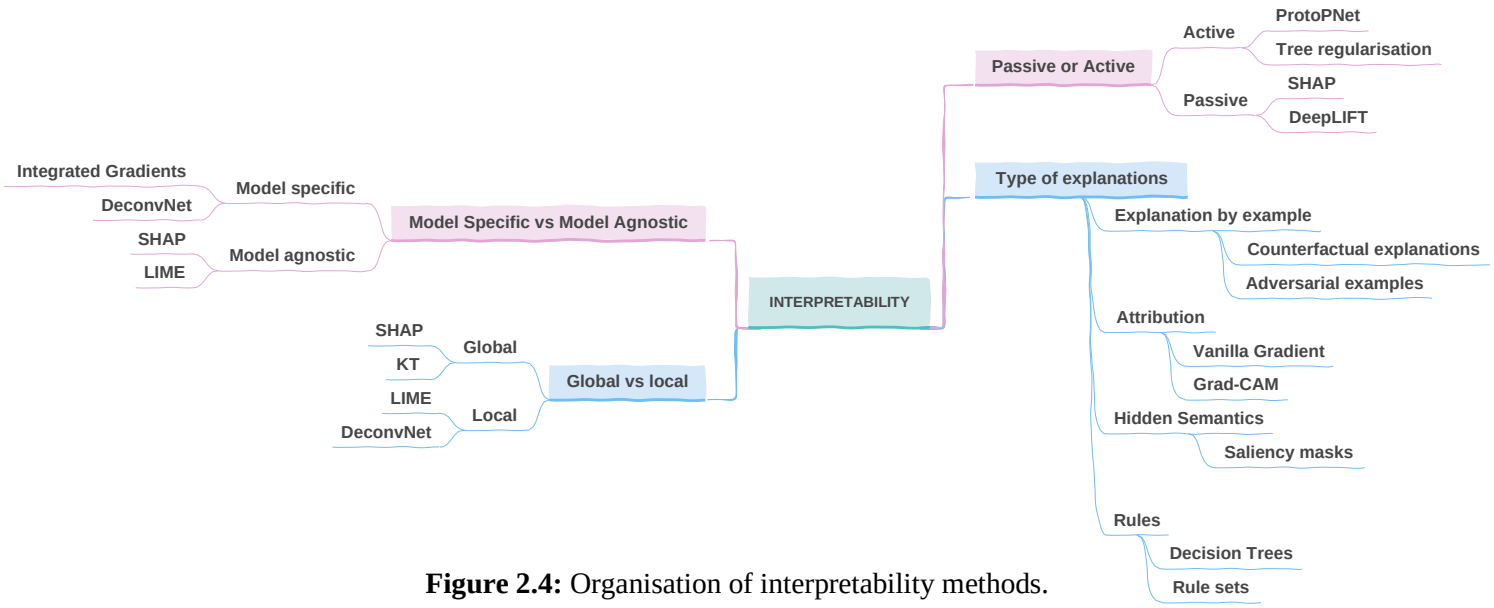

Figure 2.4: Organisation of interpretability methods.

The defined taxonomy embodies the crucial elements that one should consider when defining the scope of an interpretability study. The model-specific and agnostic elements determine the family of methods available for interpreting specific model types. When a model-specific method does not exist, one can simply choose one that is not dependent on the model class (model-agnostic method) or in situations where there are methods on both ends. It might be helpful to use them simultaneously for comparisons. Another important aspect of interpretability to consider is the axiom of global versus local interpretability. One might argue that understanding the overall decision logic of a model via global interpretability might be more beneficial than having a local explanation over a group of data samples.

In essence, because the search space for interpretability methods is so ample, it is important to investigate the landscape via a framework so that one can choose a set of methods that tick the correct boxes. The following section of this chapter will explore the different attribution and hidden semantic techniques. Highlighting the strengths and weaknesses of each class of methods and how techniques within each class compare against each other. The logic rules and explanation by example methods are relegated to appendix A as to the author's knowledge, these methods are less popular with deep neural networks trained on speech.

2.5. ATTRIBUTION METHODS

Attribution methods determine how important each feature of the input is to the overall prediction. In cases where the input features have semantic meaning, it is easy to assign a global score to each feature. Whereas in situations where the features have no semantic meaning, such as pixels in images, it is difficult to attribute that a pixel in a particular position consistently has the same contribution to the prediction. Currently, most attribution methods are geared toward providing visual explanations for computer vision models. In this literature review, attribution methods are explored using two categories, model-specific attribution methods and model-agnostic attribution methods.

2.5.1. Model agnostic attribution methods

2.5.1.1. Local interpretable model-agnostic explanations (LIME)

As the name suggests, LIME is a model-agnostic interpretability method. This means it can be used to explain the predictions of any machine learning model. In the original paper [45], LIME is introduced as a surrogate model, i.e., a simple model trained to approximate a more complex model. In this case, the surrogate model is interpretable, and it is used in a local sense to explain the complex model's predictions.

LIME investigates how a model's predictions are affected by channelling variations of the input data into it. This is accomplished by generating a surrogate dataset containing the perturbed samples whilst the original predictions of the black box are conserved as labels for this surrogate dataset. This new dataset is then used to train an interpretable model. The model can take the form of something as simple as a decision tree or linear regression as long as the resulting model gives a good local approximation of the original model.

To summarise the working principle of LIME in more technical terms. Consider an example *x;* LIME works by trying to fit, in a local sense, an interpretable model that is true to the predictions of the original machine learning model, *f(x),* in the neighbourhood of *x.*

$$g(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g), \tag{2.5.1}$$

where g is the explanation model for the instance x. The loss function L measures the closeness of the explanation g to the prediction of the original model f, whilst the model's complexity $\Omega(g)$ is kept considerably low. The variable G represents the family of possible explanations, for instance, all the possible decision tree models that can be of use. The variable π_x is the proximity measure that defines the size of the neighbourhood around x to which we consider for the explanation. LIME will only optimise for the loss L. The complexity is user-defined by determining the maximum number of features that the chosen surrogate model may use.

LIME can also be adapted to provide pixel attribution explanations for image classifier models. The advantage of using LIME is that it determines explanation weights that indicate which features are negatively or positively correlated with the predicted class. Figure 2.5 shows an instance of an

explanation provided by LIME for a prediction made by a fruit classifier. The fruit classifier model is a simple convolutional neural network architecture trained on the Fruits 360[1] image dataset [46]. The dataset has over 100 classes of vegetables and fruits, but for demonstration purposes, the classes are reduced to only 16 for this example.

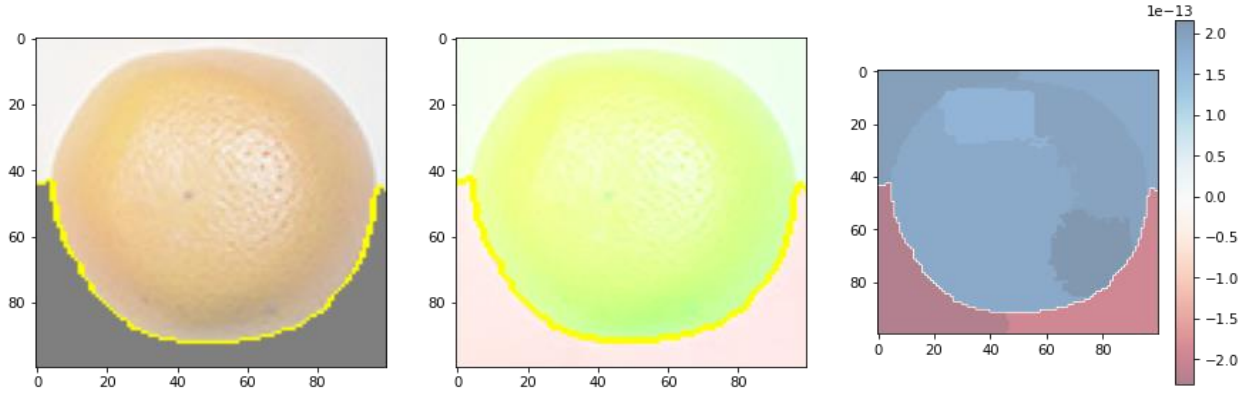

Figure 2.5: a. Irrelevant segments greyed out. b. Positive/Negative overlayed. c. LIME heatmap.

The figure above depicts the different ways one can visualise an explanation generated by LIME. The images are achieved by using the LIME algorithm to determine the relevance of each part of the image to the model's prediction. The relevance is then superimposed on the original image to produce the human-readable explanations shown in figure 2.5.

In the first image, the greyed-out parts show the areas that did not contribute positively to explaining the prediction of orange. The second image shows both the positive and negative regions separated by the yellow boundary. Similar to the first image, the irrelevant segment is the region under the yellow boundary. Whilst most of the orange fruit is highlighted in green to show a positive contribution, some areas are regarded as more important represented by the darker green shade.

Lastly, the final image results from mapping the explanation weights generated by LIME to corresponding superpixels. These superpixels are not necessarily individual pixels themselves but represent a group of pixels or entire image segments. It is important to note that the LIME's perturbation strategy is not executed on individual pixels by these super pixels. This is because not much change is achieved by randomly changing individual pixels.

The magnitude of an explanation weight is represented by the different shades of colour, as shown by the symmetrical colour bar. The dark grey patches correspond to positive coefficients, and on the other end of the spectrum, the red regions represent negative coefficients. Compared to the first image, the light red parts of the image correspond to the irrelevant greyed-out segments. The different shades of red here show that the patch on the left has a much more significant negative contribution.

Whilst LIME has a very intuitive and straightforward approach, the method has several limiting factors. As pointed out by [47], the explanations produced by LIME can be somewhat unstable. In a simulated setting, explanations of two points within very close proximity of each other can vary greatly. Furthermore, correctly defining the neighbourhood is a huge problem when implementing LIME on specific kinds of data.

[1] https://www.kaggle.com/moltean/fruits

2.5.1.2. Shapley Additive exPlanations (SHAP)

Introduced by Lundberg *et al* [48], SHAP aims to explain the contribution of each feature to the overall model prediction using Shapley values [49]. Based on game theory, Shapley values prescribe how to split the reward among cooperative game participants equitably. Assuming all players did not contribute equally whilst playing the game, in order to distribute gains fairly, the distribution must adhere to two fairness principles:

i. Additivity – the sum of rewards received across all players must be equal to the total earnings.

ii. Consistency - a player with more contributions will earn more than a player with fewer contributions.

The Shapley value for a player will define the total amount of earnings they are going to receive. To determine this value, we compute the marginal contribution for the player, which involves running the game with and without the player for every possible subset of players. The difference in total reward for each possible subset, with and without the player, is averaged out for all possible subsets, and this becomes the Shapley value.

In the case of SHAP, the players are replaced with input features or a subset of features, and the reward is the model's prediction. A feature's contribution is defined as a decrease in prediction error for regression or an increase in probability for classification. Assuming one has the full coalition of all present input features and all possible subsets of the features, minus the feature of interest *j*. To determine the Shapley value for the input feature *j* and a particular subset, one first determines the contribution of *j* once it is added to the subset. This contribution is then weighted by the likelihood of randomly picking that subset of features over all possible subsets. Finally, the weighted contributions are summed up across all possible subsets.

The explanation *g* can be defined as:

$$g(z') = \emptyset_0 + \sum_{j=1}^{M} \emptyset_j z'_j, \tag{2.5.2}$$

where $\emptyset_j \in \mathbb{R}$ are the Shapley values (feature attribution for the input feature *j*), *M* is the maximum coalition size and $z' \in \{0,1\}^M$ is the coalition vector (vector of simplified features). The coalition vector describes the absence of a feature with a 0 and its presence with a 1.

SHAP can be considered an extension of LIME [50], the explanations generated by computing Shapely values achieve local additive feature attribution. Local in the sense that an explanation is developed by perturbing feature(s) for a single instance. The critical difference between LIME and SHAP is that because the latter is based on computing Shapley values, it guarantees a fair distribution of the contribution for each feature. LIME will also assume that the local model is linear, which in some cases makes it difficult to define the locality if the decision surface is complicated. SHAP is maybe a bit more desirable because of its simplicity and theoretically solid guarantees rooted in cooperative game theory [49]. However, for large models computing Shapley values is quite resource-expensive, which might require subsampling to avoid. Another critical weakness to acknowledge about SHAP is that since it provides additive attribution explanations for features, if a model is not of an additive nature, then the Shapley values can be misleading.

To demonstrate the type of explanations provided by SHAP, a simple shallow neural network was trained on the Pima Indians diabetes dataset[2]. The dataset contains several diagnostic measurements such as the patient's age, body mass index and diastolic blood pressure. The network is trained to use these features to estimate if the patient is at risk of having diabetes or not. After the model has been trained, the Shapley values for each input feature are calculated for each sample in the holdout test set (dataset samples left out of training data) using the method described earlier in this section.

Figure 2.6 shows the type of explanation SHAP provides for a single prediction. This figure is called a force plot[3]; it follows the basic idea that each input feature contributes a force to push the model towards a certain decision [50]. The forces are the calculated Shapley values, and each prediction is considered starting from the base value (the average for all predictions). The value *f(x)* represents the predicted value, and the forces (Shapley values) are made to balance on the prediction, similar to balancing weights on a seesaw.

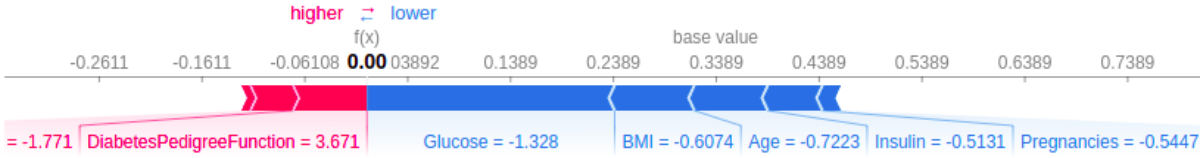

Figure 2.6: Force plot of Shapley values for a single instance.

For this example, the model has correctly estimated that this patient is not at risk of developing diabetes. The features in red push the prediction towards the patient having diabetes (1), and the blue features push the prediction towards the correct prediction (0). According to the force plot, the most significant contributing factors toward the correct prediction are the glucose, body mass index, age, and insulin level.

More ideally, the Shapley values can be aggregated to formulate a global explanation. This gives a much better understanding of feature importance. To obtain a global perspective, the Shapley values for each feature are averaged for all instances in the holdout test set. The figure below shows the result of ranking the features in order of decreasing importance in a SHAP summary plot [50].

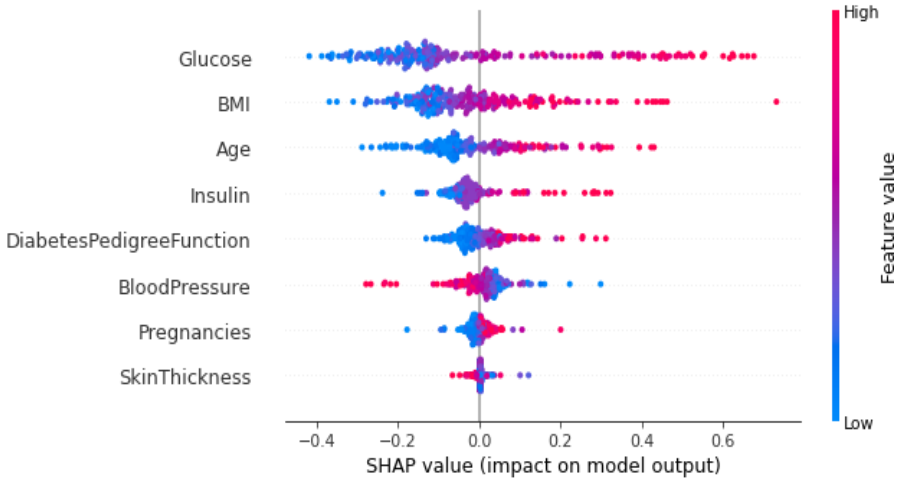

Figure 2.7: SHAP summary plot ranking the features in order of decreasing importance.

[2] https://www.kaggle.com/uciml/pima-indians-diabetes-database
[3] The SHAP illustrations are produced using the *shap* python package (https://pypi.org/project/shap/) built by authors of the original SHAP research.

In this representation, the feature importance is combined with feature effects. The left *y*-axis is a top-to-bottom ranking of the input feature in order of overall impact on the model's output. This ranking is determined by calculating the mean absolute SHAP values over the entire holdout test set. Hence each dot on the summary plot has three properties: the feature importance on the left *y*-axis, a colour showing a feature is high or low, and the SHAP value on the *x*-axis. Taking for instance the age feature, as the age increases the impact on the model output increase, and the patient is more likely at risk of developing diabetes.

Overall, this representation is more favourable because it shows the direct relationship between the magnitude of a feature and its impact on the model's output for each sample in the dataset. Referring to Lipton's framework [41], the summary plot provides interpretability from a global perspective, whilst the force plot only explains one sample providing local interpretability. Furthermore, additional relationships can be derived from the results provided by SHAP values, such as the SHAP dependence plots or the SHAP interaction values. Each of these depicts interactions of features on different levels of abstraction.

Both SHAP and LIME are regarded as perturbation-based methods because they construct model explanations by analysing the effects of altering the input features on the model's decisions. The underlying idea behind methods of this kind hangs on the intuition that if distortions are made to the important features of the input, then the model's decision logic is heavily affected. As opposed to when unimportant features that are perturbed, then the model's decisions are not affected. We can think of this as a proxy for sensitivity analysis, where one can determine how the model's prediction is affected by slight changes in input.

2.5.2. Model-specific attribution methods

The model-specific methods of interest in this category are the ones that exclusively work with neural networks. These methods are called gradient-related methods. As their name suggests, these methods use gradient and or backpropagation to determine the influence of each input feature in the model's predictions. This type of attribution is analogous to the traditional sensitivity analysis method used when designing control systems. Since neural networks are trained using gradient descent or its variants, the idea of gradients is very familiar since they are used to optimise the model's objective function.

Gradient-type methods are prevalent in computer vision, where the generated explanations are represented as masks superimposed on the original image. These masks are commonly referred to as saliency maps or sensitivity maps and are usually the same size as the input image. There are various flavours of saliency map-generating techniques, and these methods mainly differ in how they handle gradient computations.

To demonstrate the saliency masks produced by the different attribution methods, the discussion of each method includes an example of applying the algorithm to explain predictions made by a simple convolutional neural network. The same network is used across each method and it is trained to classify between images of handwritten numbers from the MNIST[4] dataset.

2.5.2.1. Vanilla gradient

Vanilla gradients were the first step toward using gradients to produce pixel-wise attribution explanations [51]. For a given class of interest, the gradient of the loss function is calculated with respect to the pixels in the input image. The result is a pixel-wise map with the exact dimensions as

[4] http://yann.lecun.com/exdb/mnist/

the input image, containing positive and negative values corresponding to the contribution of each pixel. The implementation steps are quite simple. First, the input image is forward propagated through the network. Then for the class interest, the gradient score is computed with respect to the pixels in the input image. This is done whilst setting all the other classes to zero. The calculation step is represented by:

$$E_{grad}(I_0) = \left.\frac{\partial S_C}{\partial I}\right|_{I-I_0}, \tag{2.5.3}$$

where I represents the image and S_c is the model score for class c.

Finally, the gradients can be visualised by either showing the positive and negative contributions independently or just showing the absolute values. Figure 2.8 shows an input image and the corresponding saliency map produced by the vanilla gradients approach. As mentioned earlier the saliency map is produced by first propagating the input image from the input of the trained network then using equation 2.5.3 the gradients are calculated for each pixel in the input image with respect to the network's prediction.

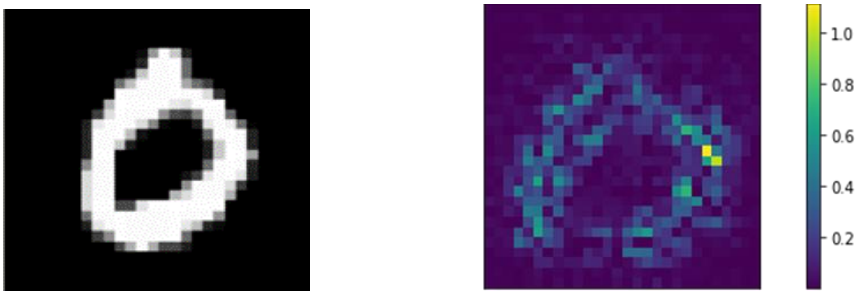

Figure 2.8: a. Input image. b. Saliency map. The colour bar corresponds to relevance scores

The active areas represent the pixels that influenced the network's decision. From a purely qualitative perspective, the map highlights the areas of the image that the model 'looked at' to output the prediction of class zero. The higher the attribution value on the colour bar, the higher relevance of the corresponding pixel to the model's prediction.

One of the main disadvantages of using vanilla gradient indicated by [52] is its proclivity to saturate. For instance, when the magnitude of the activation is negative, the rectified linear unit activation function is used. The activation does not change and is always limited to a zero value. In this case, the activation is said to have saturated. This is undesirable because the gradient will also be zero, and the algorithm will not consider the neuron's contribution to the outputs.

2.5.2.2. DeconvNet

Introduced by Zeiler et al. 2014 [53], DeconvNet reverses the convolutional, pooling and activation operations of a convolution neural network (ConvNet) until it reaches the required input space. The method is computationally similar to the vanilla gradient approach but only differs in how it handles the backpropagation of gradients through the ReLU activation. For DeconvNet, the ReLU layer is reserved, and the network caches information about previous activations for each neuron in a layer. This is done in such a manner that during backpropagation from layer *n* to *n-1*, the activations that were zero in layer *n* during forward propagation are cached and cast to zero in layer *n-1*.

There is also an extension to the DeconvNet method called guided backpropagation [54]. This technique combines both the vanilla gradient and DeconvNet methods. When backpropagating gradients, the ReLU activation (similar to vanilla gradients), guided backpropagation only considers the positive error gradients (similar to DeconvNet).

Similar to figure 2.8, the image below shows a pixel attribution map produced by applying the guided backpropagation to the network's prediction for the handwritten digit seven. The gradients calculation is now extended to only consider positive gradients that flow through the ReLU activation.

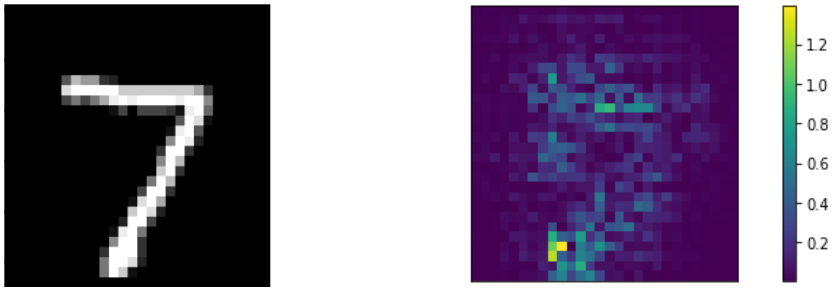

Figure 2.9: *a.* Input image. b. Guided backpropagation pixel attribution map.

2.5.2.3. Gradient-weighted Class Activation Map (Grad-CAM)

Grad-CAM was developed to provide visual interpretations for convolutional neural network architectures [55]. Similar to the previously discussed saliency methods it uses gradients to produce relevance scores for each neuron relative to a decision of interest. However, unlike the other methods, the gradients in Grad-CAM are not backpropagated back to the input image but to the last convolutional layer. The result is a coarse localisation map that indicates areas of an image the model focused on to produce the prediction for a specific class. To give a more technical definition for this method, consider the resulting localisation map:

$$L^c_{Grad-CAM} \in \mathbb{R}^{u \times v} = ReLU\left(\sum_k \alpha^c_k A^k\right), \tag{2.5.4}$$

where *c* is the class of interest, *v* and *u* are the height and width of the explanation, respectively.

First, the image of interest is forward propagated through the CNN, and the activation (score) for the class of interest is obtained before it reaches the softmax layer. After all the other class activations are set to zero, the gradient of the current class activation $\frac{\partial y^c}{\partial A^k}$ can then be backpropagated to the last convolutional layer in the architecture (before any other types of neural network layers). The ReLU activation is defined as:

$$ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}, \tag{2.5.5}$$

where *x* is the input to the neuron.

The gradients can then be applied and averaged as represented by the following equation:

$$a^c_k = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}. \tag{2.5.6}$$

The class of interest is represented by y^c, a^c_k are the weights that indicate how important each feature map A was to the overall model's decision. To ensure that only positive values are considered, a weighted sum between A and a^c_k are fed into a ReLU activation.

Figure 2.10 shows the result of applying the Grad-CAM algorithm to the predictions of input images of the handwritten numbers six and seven respectively.

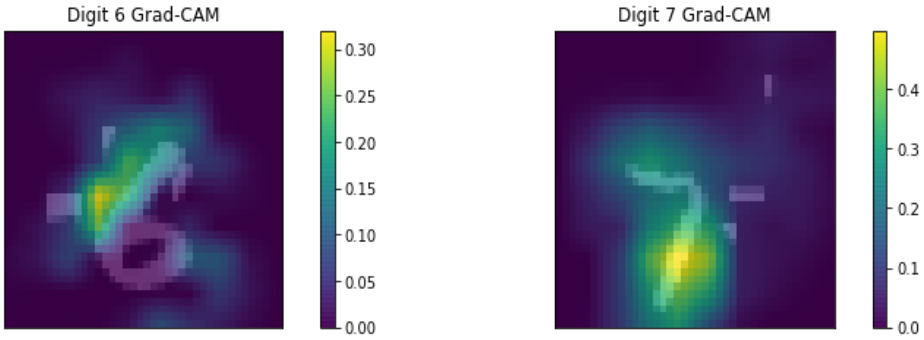

Figure 2.10: Results of Grad-CAM for digits six and seven, respectively.

Grad-CAM can be improved by combining it with guided backpropagation. This will help improve the fidelity of its heatmap since the last convolutional layer produces low-resolution activation maps. However, Guided backpropagation backpropagates to the input pixels; hence, its results have far more resolution and detail. The resulting method is called Guided Grad-CAM.

2.5.2.4. Layer-wise relevance propagation (LRP)

LRP [56] works by conservatively backpropagating the prediction of a network using carefully crafted propagation rules. The propagation is implemented recursively from the output to the input layer, where a heatmap with relevance scores is produced and superimposed over the query image. For the neuron i in the l^{th} layer to the j neuron in the $(l+1)^{\text{th}}$ layer, the relevance score R is computed as:

$$R_{i\leftarrow j}^{l,l+1} = R_j^{l+1} \frac{a_i w_{ij}}{\sum_h a_h w_{hj}}. \tag{2.5.7}$$

LRP is considered conservative because the magnitude of the prediction is preserved whilst back-propagating through the network, making it equal to the sum of the relevance values of the input layers. The conservative nature of the LRP method is inherited from the layer-wise conservation principle, which makes sure whatever is being propagated is preserved between the neurons of adjacent layers. The conservation principle is defined as:

$$R_i^l = \sum_j R_{i\leftarrow j}^{l,l+1}, \tag{2.5.8}$$

where R_i^l is the relevance of the i^{th} neuron of layer l and consequently $R_{i\leftarrow j}^{l,l+1}$ is the relevance of the j^{th} neuron in the next layer. LRP is also classified as a decomposition-based method because the model's prediction is decomposed into pixel-wise relevancies that represent the contribution of each pixel to the overall classification. The image below shows the result of applying the LRP algorithm to the network's prediction of the input images zero and one respectively.

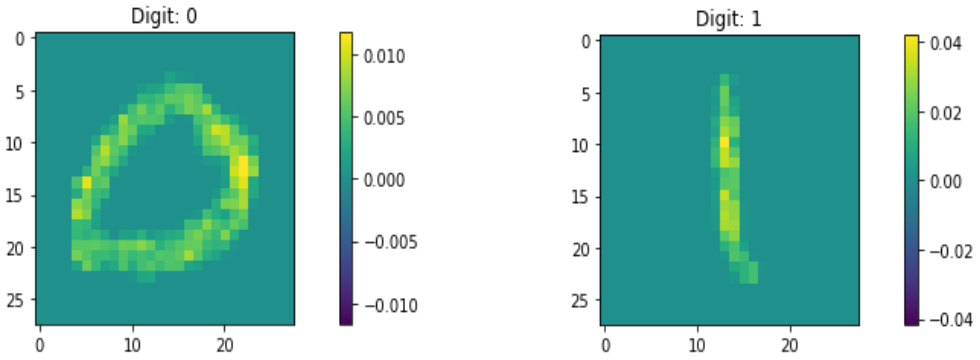

Figure 2.11: LRP heatmaps for the numbers zero and one.

The propagation rules can be formulated specifically for each layer in the network to produce high-level explanations for complex models. However, because LRP uses modified forms of backpropagation, it is limited to specific types of network implementations.

2.5.2.5. Integrated gradients

The integrated gradients method [57] is a perturbation-type attribution method that has found popularity in the field of medical imaging. The method is inspired by the idea of Shapley values from cooperative game theory discussed earlier in section 2.5.1.2. More specifically, the Aumann-Shapley values for infinite player games [58]. Integrated gradients work by generating a set of intermediate images between a baseline image containing all zeros to a reference input image. For each image generated along this path, the gradient of the model's output with respect to each individual pixel in the intermediate image is calculated and then cumulatively summed over the whole path of images. The result is a heatmap highlighting pixels important to the model's prediction as demonstrated in the figure below.

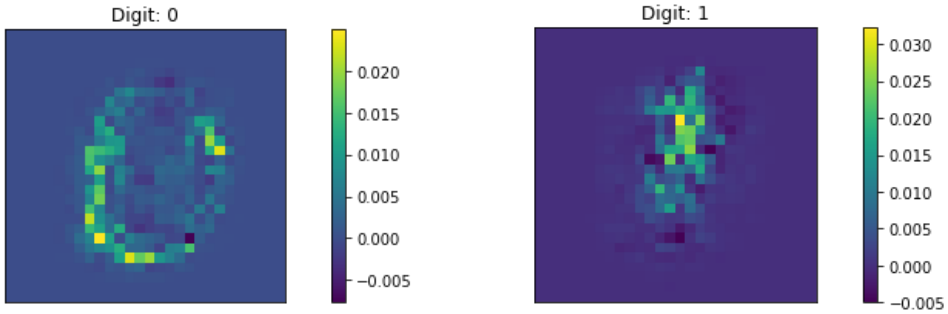

Figure 2.12: Integrated Gradients heatmaps for the numbers zero and one.

The pixelwise heatmap *IG(x)*, produced for the baseline image $x' \in R^n$ and the given input image $x \in R^n$ is defined as:

$$IG_i(x,\ x') := (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha\,(x - x'))}{\partial x_i}\, d\alpha, \tag{2.5.9}$$

the model is represented by $F: R^n \rightarrow [0,1]$ and R^n is a straight-line path between *x'* and *x*.

The integrated gradients approach was developed to satisfy two desirable axioms for attribution methods, sensitivity[5] and implementation invariance[6]. The authors argue that gradients methods such as guided propagation and deconvolutional neural networks do not hold for at least one of the two axioms. Firstly, attribution methods that rely on gradients to produce explanations break the sensitivity rule. For a method to satisfy this rule, it has to assign a non-zero attribution to a feature whenever the predictions for the input and baseline are not the same when only that one feature is different. Gradient-based methods break this rule because, in some cases, it may have a zero gradient at the input (flatten), regardless of the function values being different at the baseline and input.

The idea of implementation variance is motivated by how two models are functionally equivalent if, for all inputs, their output is the same regardless of how they are implemented. Attribution methods should satisfy this quality and produce similar attributions for functionally equivalent models. Typically, gradient-based attribution methods are fine in this regard because, based on the chain rule, gradients are invariant to implementation. However, methods such as layer-wise relevance

[5]Sensitivity - a feature is given zero attribution/importance if the model does not depend on it.
[6]Implementation invariance - two networks that approximate the same function must have the same attributions.

propagation break this rule because they use discrete gradients instead of instantaneous gradients. To compute discrete gradients, layer-wise relevance propagation uses a modified version of backpropagation that breaks the chain rule.

Similar to SHAP, this approach is based on a well-grounded mathematical concept. For integrated gradients, it is the theorem of calculus for line integrals. This theorem ensures that the produced attributions will always add up in a meaningful way, satisfying the condition of completeness. This is a common characteristic of robust interpretation methods. Their effectiveness can be demonstrated by satisfying notable mathematical theorems.

2.5.2.6. Smooth-Grad

Smooth-Grad [59] is not an attribution method by itself but is a technique used to denoise the results produced by gradient-based attribution methods. This is achieved by adding noise to the output and averaging over the noisy gradients. When training neural networks, the goal is to train the model for a specific task instead of keeping the gradients from fluctuating (keeping them smooth in a sense). Smooth-Grad aims to correct this by averaging over maps to even out the fluctuations.

2.5.2.7. Expected Gradients

The expected gradients method is an extension of integrated gradients [60]. It aims to avoid any ambiguity that comes with selecting a meaningful baseline by integrating gradients over multiple references. Integrated gradients provide relevance scores by accumulating gradients of images interpolated between the baseline value and the current input. The baseline is intended to represent some uninformative reference input that symbolises a lack of knowledge about the input characteristics' value. For computer vision tasks, the baseline can be chosen to be an image with all black pixels. This means integrated gradients will deem black pixels as areas of no interest. This can be problematic for situations where the target object has black pixels. Furthermore, there is no rule of thumb for determining this baseline input, making the step of choosing a baseline quite an arbitrary process.

Expected gradients avoid using an arbitrary baseline by integrating over a dataset. Expanding from the integrated gradients equation, to avoid choosing a specific baseline *x',* the expected gradients for feature *i* are defined as:

$$ExpectedGradients_i(x) := \int_{x'} \left((x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha \right) p_D(x') dx', \quad (2.5.10)$$

where *D* represents the underlying data distribution.

The authors note that since integrating directly over the training distribution is intractable, the integrals can instead be reformulated as expectations:

$$ExpectedGradients_i(x) := \underset{x' \sim D, \alpha \sim U(0,1)}{\boldsymbol{E}} \left[(x_i - x'_i) \times \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} \right]. \quad (2.5.11)$$

If a linear function is used to approximate the model between the sample to be interpreted and each sample in the background data, the expected gradients computed will approximate SHAP values. The input features are assumed to be independent.

The figure below shows expected gradients explanations for input images of the handwritten digits four and zero. The red pixels are positive SHAP values that pushed the model's decision towards that

class, whilst the blue pixels are negative SHAP values that push the prediction away from the predicted class. This idea is similar to the force plot shown in figure 2.6.

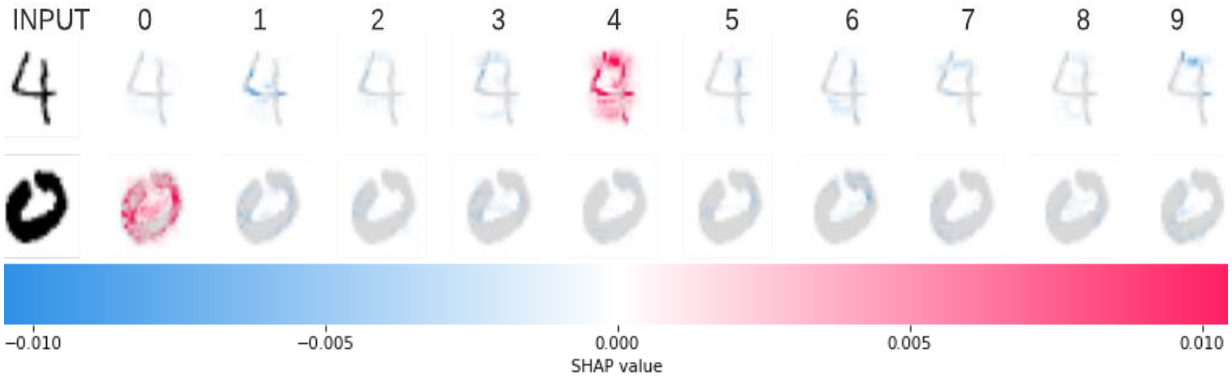

Figure 2.13: Model explanations for MNIST digits.

The network used in this example has a softmax activation function for its output layer, meaning expected gradients will produce explanations equal to the number of classes. In this instance, MNIST has ten target classes from the number zero to nine. Hence each input image will have ten visualisations explaining the elements of the input which influenced the model's decision.

For the digit zero, the model focuses on the central part of the digit. This makes sense because the digit zero is circular in nature and hollow in the middle. The prediction of the correct class has positive attributions on the central part, whilst other classes have negative attributions if that central part is not hollowed. The prediction for the digit four is rather interesting, the model seems to latch on the idea that the top part of the digit has a gap. This can be confirmed by looking at the attributions given for the digits seven and nine, the negative attributions are predominantly given to the top part of the digits because they connect, pushing the prediction away from these classes.

2.5.3. Comparing attribution methods

Attribution methods are the natural form of methods used to interpret the working mechanisms of machine learning models, especially those trained on image data. With the growing popularity of these methods, interest has also peaked to assess the quality and scope of the explanations they produce. Most notably, Adebayo *et al* [72] introduced statistical randomisation tests to evaluate saliency-based methods. The first test establishes the effect of the randomisation of model parameters. The basis of this experiment was to investigate whether there is a difference in explanations between a trained model and an untrained model with the same architecture but has been randomly initialised. If the saliency model truly depends on the learned weights of the model, then the explanations of the two models on the same input should be different. However, if the explanations are similar, the attributions are insensitive to model parameters, and the attribution method fails this test.

The second test focuses on the effect of randomising labels of the training data. For this experiment, two models with the same architecture are trained: one on the original data and the other on the data with randomly permuted labels. Similar to the first experiment, an attribution method must provide different explanations for these two models. If the explanations are the same, however, the attributions are not dependent on the relationship between the instances and labels of the original training data.

These experiments show that some salience maps techniques do not form any relationship with the training data or the network parameters. This is made worse because we are inclined to qualitatively

focus on whether the model focuses on the image's relevant areas when evaluating saliency maps. The authors caution that the reliance on visual explanations provided by saliency maps can be misleading, suggesting that the most suitable saliency map methods to use for formulating explanations are the Gradients and Grad-CAM techniques. The two seem to have a reasonable relationship with the model's trained parameters and the labels in the training data. The experiments were conducted on visual saliency methods for image classification; these ideas are transferable to other domains.

Although integrated gradients fail the sanity checks proposed by Adebayo *et al* [61], it does pass other important desirable properties of attribution methods such as [57][60]:

i. Implementation invariance - two networks that approximate the same function must have the same attributions.
ii. Sensitivity - a feature is given zero attribution/importance if the model does not depend on it.
iii. Performance at saturation - attributions must not be affected by the saturation of gradients. The model is saturated when it perfectly fits that data such that the gradient signal through the model is zero.
iv. Completeness - attributions should add up to the difference between the output of a function at the input (the query to be explained) and the output of that function at a baseline.
v. Linearity - if a model is a linear combination of two sub-models, then the attributions must be a linear combination of both the sub-models 'attributions.
vi. Symmetry-preserving - identical attributions should be given to symmetric variables with identical values.

Whilst Grad-CAM passes sanity checks [61], it does not satisfy other important axioms such as implementation invariance and performance at saturation. An important aspect to consider when choosing an attribution method is its theoretical basis. Expected gradients are particularly attractive because they combine well-grounded ideas from Shapley values, Integrated gradients and Smooth-Grad. Taking a further step from integrated gradients, expected gradients allow the whole dataset to be used as a background distribution as opposed to a single baseline.

2.6. HIDDEN SEMANTICS

Most of the interpretability methods discussed so far focus on the input and output layers to produce explanations. Olah *et al* [62] suggest that this might be because in areas like computer vision, these layers have a clear inherent meaning associated with them. For instance, the output softmax layer comprises of class labels and their corresponding probabilities and the input layer is made up of image pixels.

One of the attractive properties of neural networks is their ability to discover hierarchal representations with different levels of abstraction as we go deeper into the network. Arguably suggesting that the power of neural networks lies within their hidden layers. Zhang *et al* [54] indicated that local attribution methods discussed in the previous section could be adapted to explore hidden semantics interpretability. This shifts the focus from assigning credit or blame to the input features to discovering how important each neuron/channel in the hidden layer was to the overall prediction. The figure below shows the SHAP attribution explanations for the same network interpreted by figure 2.13 extended to the hidden layer. The attributions are mapped to the third convolutional layer in the network with respect to the model's prediction.

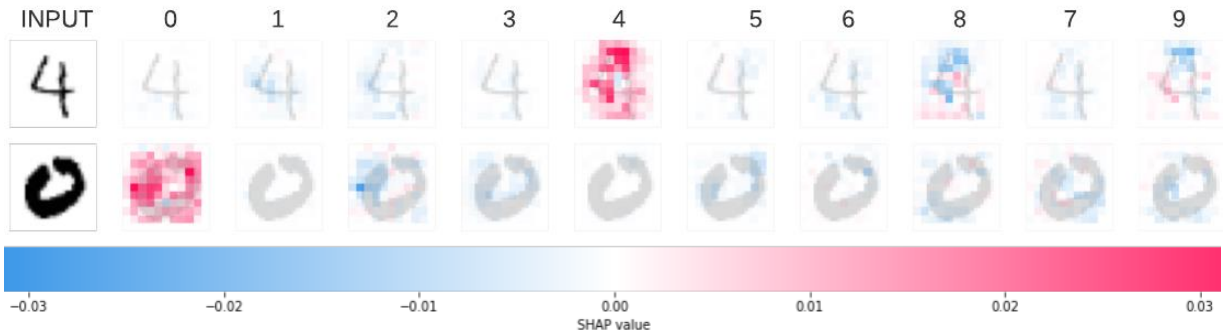

Figure 2.14: SHAP explanations extended for the hidden layer.

2.6.1. The human brain and semantic mapping

In the early days of neural network research, it was common to assume a solid relationship between biological neurons and artificial neurons. As research progressed, the differences between the two have become increasingly apparent. However, there has been no change to the initial goal of building neural network models that generalise as well as our human brains. Whilst we currently do not have a definitive model for the human brain, neuroscience research has made strides towards establishing how the brain responds to different stimuli. One such example relevant to this study is the research by Huth *et al* [63], which formulated a semantic map of the brain indicating the areas that hold information about language understanding. The subjects of this study were made to listen to 2 hours' worth of stories from a radio station whilst their brain activity was measured using functional magnetic resonance imaging (fMRI).

The authors were able to model a relatively symmetrical distributed map across the cerebral hemispheres, indicating how the brain is organised to respond to different semantic concepts. Figure 2.15 shows an example of a semantic atlas of the human brain generated by the Brain interface[7]. The different colours indicate different concepts of the real world as represented by the legend.

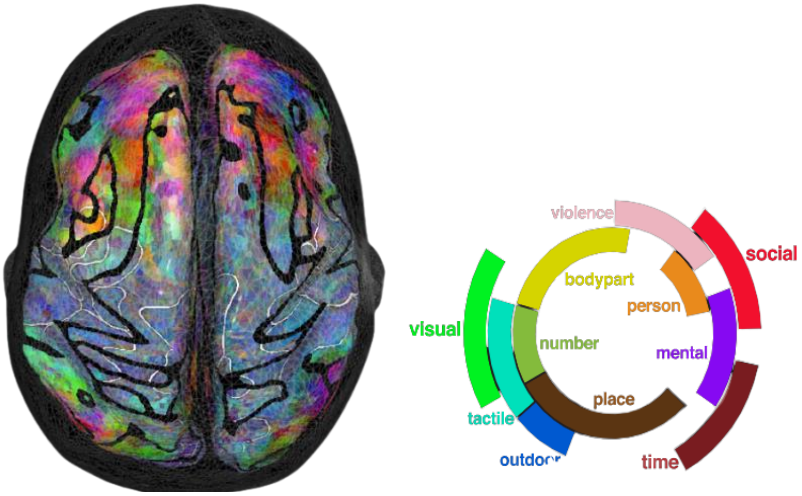

Figure 2.15: Example of a semantic map of the human brain.
Source: *Adopted from* [63].

Creating a semantic map does not place emphasis on understanding how the neurons are connected or how the neurochemical reactions result in the 'firing' or activation of the neurons. What matters most is that one can measure this activity with a certain degree of acceptable accuracy. An earlier study

[7] https://gallantlab.org/huth2016/

[64] showed how brain activity could be used to produce quantitative models that provide insights into cortical representations. These quantitative models or 'encoder' models can then be used to develop corresponding decoder algorithms. A decoder can transform the encoded signal into an approximation of the original representation. Loosely speaking, the decoder model is an inverted encoder model. The authors developed a decoder that could reconstruct from brain activity measurements a set of frames that closely resembled the movie the subjects were watching.

Whilst there is a significant difference between their building blocks, the brain and deep neural network architectures both give evidence that they can discover and hold abstract ideas about the world. The brain is organised into interconnected functional parts, and an idea that exists in the real world has a distributed representation across these multiple parts. We can infer this distribution of ideas through a semantic atlas and produce encoder and decoder models. A semantic map can ultimately be used to confirm if an idea is represented correctly within the human brain. This creation of functional maps of the brain is synonymous with interpreting neural networks through hidden semantics methods. The proceeding section will show how the real-world concepts modelled by a neural network can be distilled into intelligible representations.

2.6.2. Visualising individual neurons and semantic dictionaries

Olah *et al* [62] introduced the concept of a semantic dictionary that enables one to make sense of hidden neuron activations through feature visualisation. A semantic dictionary consists of pairings between neuron activations and their corresponding visualisations sorted in order of magnitude. As a result, a semantic dictionary gives a finer grain look at what each individual neuron is detecting. These can be aggregated to show the activation of a network for a given spatial location and consequently the image as a whole. Initially, these activations are abstract vectors that are less intelligible and less desirable to work with. The adoption of semantic dictionaries not only aids the migration from less meaningful and abstract representations but also gives a way to represent the learned hierarchical representations in more familiar examples.

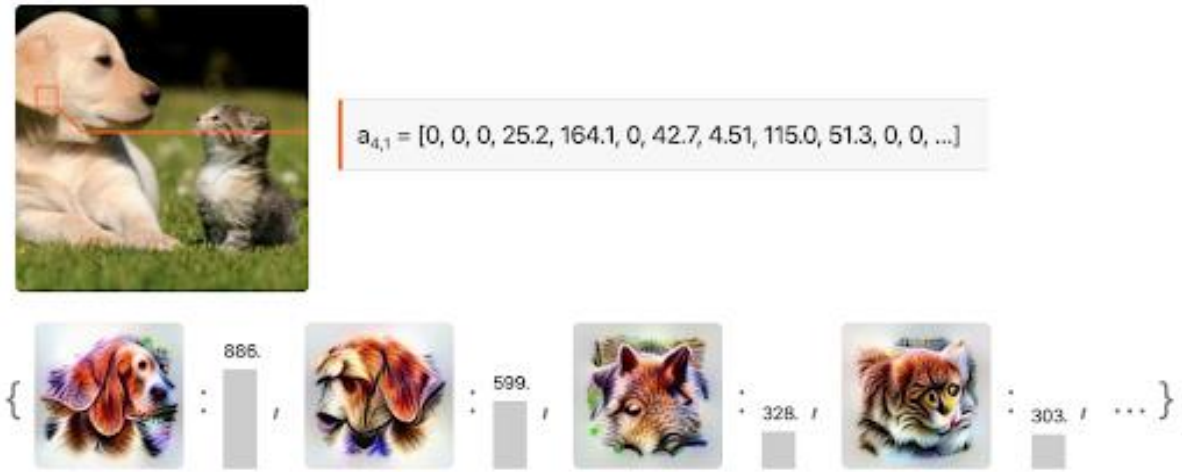

Figure 2.16: Visualisation of the activated neurons.

Source: *Adopted from* [62].

Looking closely at figure 2.16, this representation offers a somewhat intelligible interpretation of what the network has learned. The highlighted activations show that the network has some understanding of ears, the first two show floppy dog ears and the third one is pointed cat ears. It makes some sense that these channels have the highest magnitude of activation since the area of interest of the input image is

the dog's ears. This is all familiar to how humans can discern between animals by looking for distinct body parts. Information like this is likely to have a representation within the human brain to which we can discover through semantic mapping via brain activity measurements.

The authors also discuss how it is attractive but lossy to reduce these learned representations into words. Although words represent concepts that we are familiar with as humans, the network might have learned a more nuanced abstract concept than what we are accustomed to.

There are several ways neurons in hidden layers can be aggregated to give seemingly meaningful representations. Olah *et al* [62] discuss the aggregation of neurons across channels, layers and spatial locations as a way to condense visualisations to a human-intelligible scale.

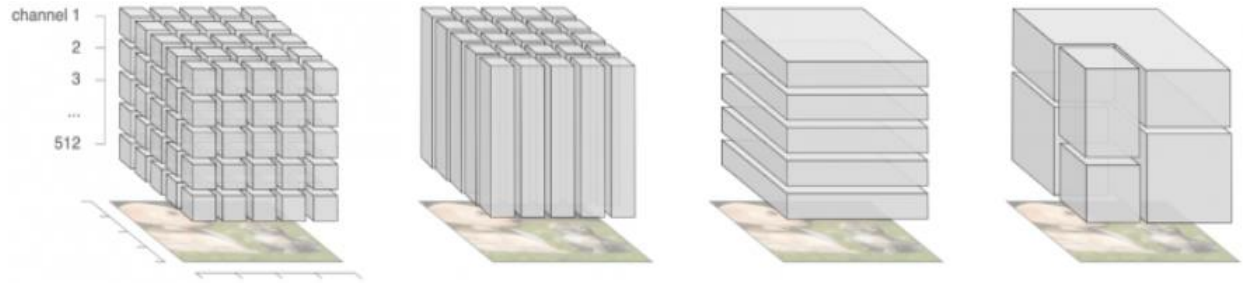

Figure 2.17: Possible aggregation of neurons as a. Individual neurons. b. Spatial Activations. c. Channel activations. iii. Neuron Groups.
Source: *Adapted from* [62].

Each of these aggregations depicted in the figure above works best when applied to specific situations. These mainly depend on the content of the image. More effectively, these groupings can also be combined to produce more informative explanations.

The visualisation and aggregations of neurons have some clear advantages over model explainers like SHAP and LIME, which completely ignore hidden layers. The interfaces produced from visualisations and attribution give researchers information about the internal representations discovered by the model. However, this approach has an inherent weakness associated with it. Since the explanation is generated per input instance, it does not show the overall view of the network in a global sense. Furthermore, taking into consideration the properties of interpretability discussed by Lipton [41]. The visualisation does not satisfy algorithmic transparency or simulatability; its main focus is to explore interpretability as a form of decomposing the black-box model.

2.7. INTERPRETABILITY METHODS FOR SPEECH PROCESSING APPLICATIONS

Interpretations of speech processing systems depend on the nature of speech features to which the underlying machine learning models were trained. The features fall into three broad signal domain categories, the time-domain, frequency-domain and time-frequency representations. Time-domain features extracted from raw waveforms of speech, these include features such as the root-mean-square energy and amplitude envelope. Figure 2.18 depicts a speech signal in the time domain, showing the progression of amplitude with respect to time and the corresponding amplitude envelope outlined in red.

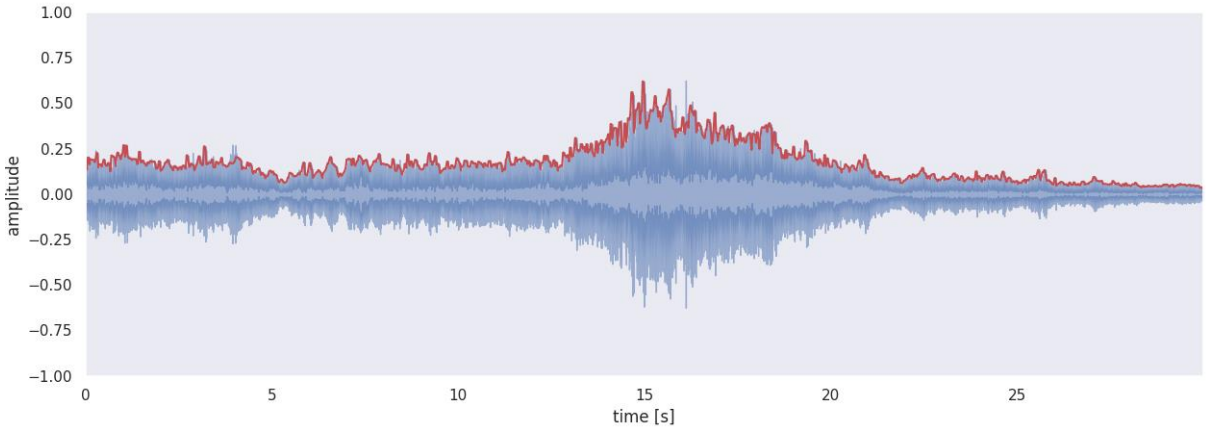

Figure 2.18: Raw waveform and amplitude envelope.

Frequency domain features are extracted from the result of transforming the speech signal from the time domain to the frequency domain via the Fourier transform. This includes features such as spectral centroid and band energy ratio. Time-frequency representations are amalgamations of time-domain and frequency-domain features. These are obtained by applying the Short-Time Fourier Transform[8] (STFT) to the time domain speech waveform. The spectrogram is a popular example of a time-frequency representation; it depicts the spectrum of frequencies of a signal over time. The image below is an example of a spectrogram produced from a speech sample. The y-axis shows the increase in frequency in kHz, the x-axis depicts the increase in time and the colour bar represents the different levels of energy for each frequency.

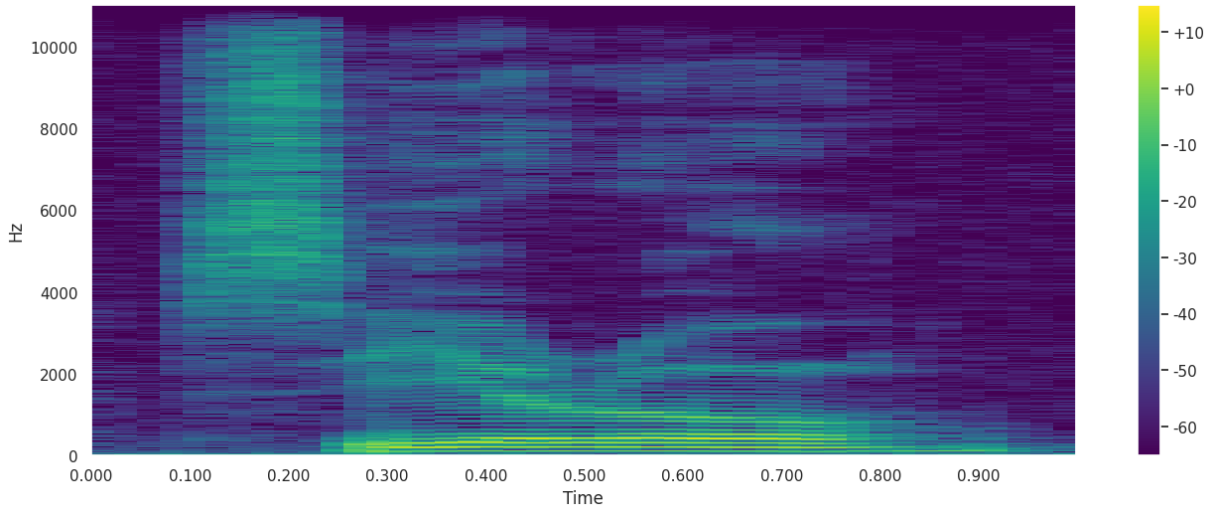

Figure 2.19: Spectrogram obtained by applying the STFT to a speech sample.

Humans are known to perceive sound logarithmically, which means we can distinguish differences in lower frequencies better than higher ones. As a result, the mel-scale was born. This is a logarithmic scale, which assumes that equal distances on the scale correspond to the same perceptual distance. The conversion from the frequency (f) to the mel-scale (m) is given by:

[8] The short-fourier transform of a signal is determined by taking the Fast Fourier Transform on small local time segments of the signal.

$$m = 2595 \cdot \log\left(1 + \frac{f}{500}\right) \quad (2.7.1)$$

The mel-scale can be used to convert a spectrogram to a mel spectrogram. A mel spectrogram reduces the number of features from a spectrogram by preserving information that is relevant to how humans perceive speech. The advantage of the mel spectrogram is immediately apparent when one is developing models that involve speech; you would want your model to learn the relevant information quickly. The image below shows an example of a mel-spectrogram produced from the sample speech sample used for the spectrogram in figure 2.19. A clear difference between the two figures is easily perceptible. Most notably the mel-spectrogram discards the higher frequency components represent in the spectrogram.

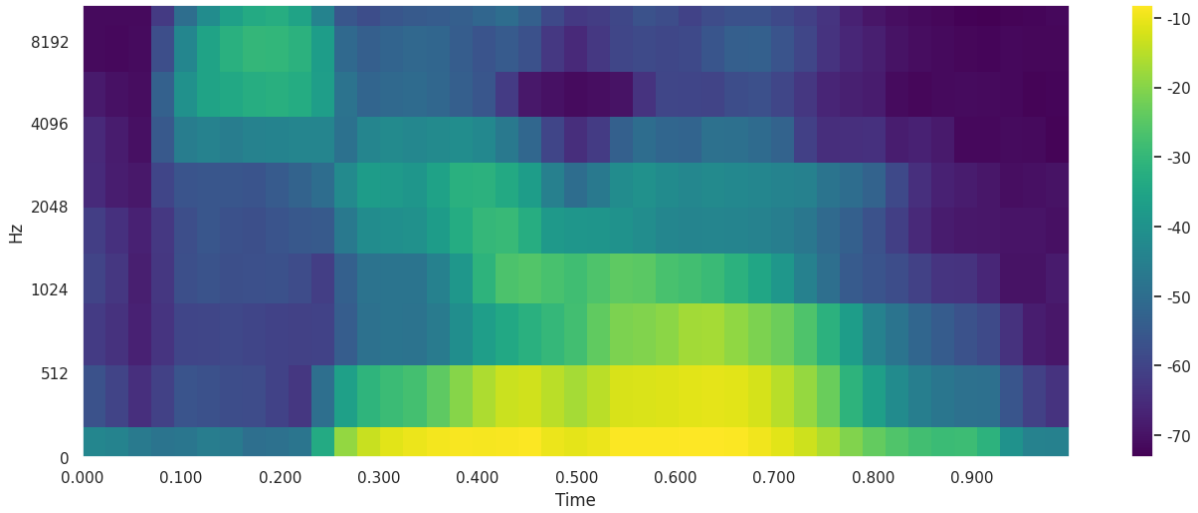

Figure 2.20: Mel-spectrogram.

The mel-spectrogram can be refined to produce what are referred to as mel frequency cepstral coefficients (MFCCs). These contain information about the rate of change in different spectral bands MFCCs are very attractive because they quickly convey the different values that construct formants. These formants are an essential characteristic of speech related to the human vocal tract, which can be used to discern between speakers.

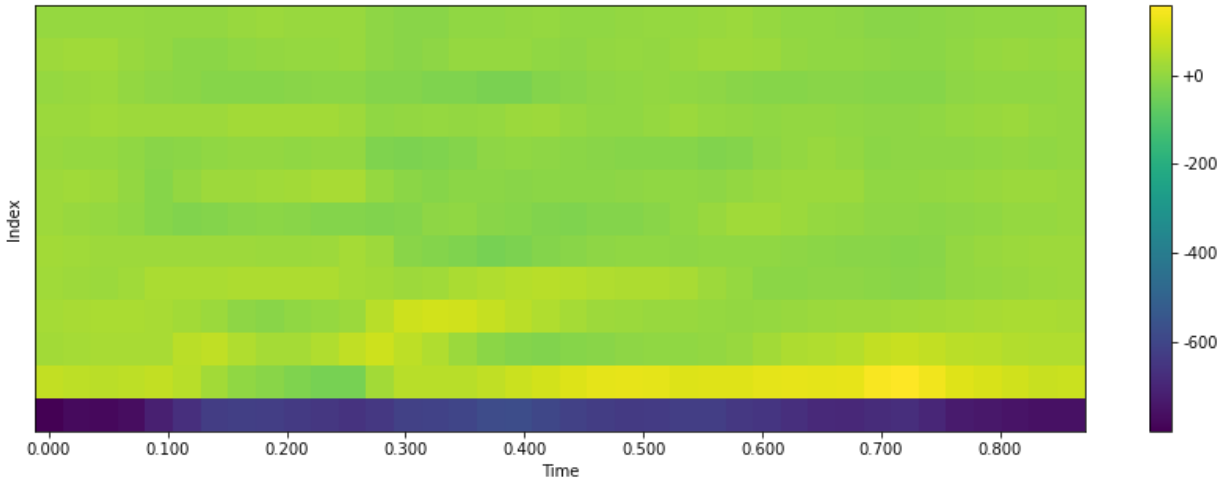

Figure 2.20: MFCC visualisation.

It is important to establish the different speech feature representations because, in retrospect, most of the interpretability methods for neural networks are geared towards explaining networks trained on images or tabular data. Since the spectrograms and MFCCs are basically the image representation of speech, the saliency methods discussed in section 2.5 can be used to interpret neural networks trained on spectrograms. However, because of the fundamental differences between one-dimensional and two-dimensional feature representations, these methods cannot be easily adopted to explain the former [40][65].

2.7.1. Explaining models trained on raw waveforms

Previously research towards developing neural networks for speech data had been centred toward using handcrafted features such as the MFCCs and Mel filter banks. A significant reason for this is because of the high dimensionality of the raw waveforms. For instance, if a speech signal is sampled at 144 kHz, this means one second of speech contains 144 000 sample points. Furthermore, speech in its time-domain representation contains useful information at different time scales. Smaller time scales carry fine structure information which interacts to form global relationships at larger time scales. This means a machine learning model must learn all these interactions at different time scales making modelling raw waveforms a difficult problem.

On the other hand, whilst comprehensive and sound knowledge is used to design these hand-crafted like MFCCs features, some useful information is lost because of high compression ratios and human knowledge bias [40]. This has prompted an increase in investigating the modelling of directly learning from the audio signal within the time domain. This approach has been successful in providing a significant performance boost for several applications, including; speaker identification in noisy and reverberant environments [66], classifying environmental sounds [67], and learning speaker embeddings for speaker diarisation [39].

A major reason for this success is due to a better understanding of how neural networks process raw waveforms using digital signal processing techniques. For instance, a recent study [68] used both frequency and time domain analysis to understand the behaviour of one-dimensional CNNs with respect to carefully crafted time-varying 1-dimensional signals. The learned filters were shown to mimic well-established filters such as the second-order Gaussian and Sav-Gol filters. This approach falls into the category of hidden semantics, the learned filter weights are visualised, and their characteristic responses are investigated via spectral analysis.

A similar approach was used to establish an understanding of how a CNN-based speaker verification system could discern discriminative speaker information relating to the human vocal tract system [69]. The study introduced two levels of analysis; the first level entailed visualising the cumulative frequency response of the learned filters in the first convolutional layer. The cumulative frequency response of the learned filters was calculated using the following formula:

$$F_{cum} = \sum_{k=1}^{n_{f1}} \frac{F_k}{\|F_k\|_2} \tag{2.7.2}$$

Where n_{f1} denotes the number of filters in the first convolutional layer, k is the filter index, and F_k denotes the magnitude spectrum of the filter f_k obtained using a 512-point Discrete Fourier Transform (DFT). The visualisations show how the kernel length influences the filters to learn different spectral information. A shorter kernel length was shown to steer the model towards learning speaker discriminative information related to the formant regions, and the longer kernel lengths helped learn information related to the fundamental frequency.

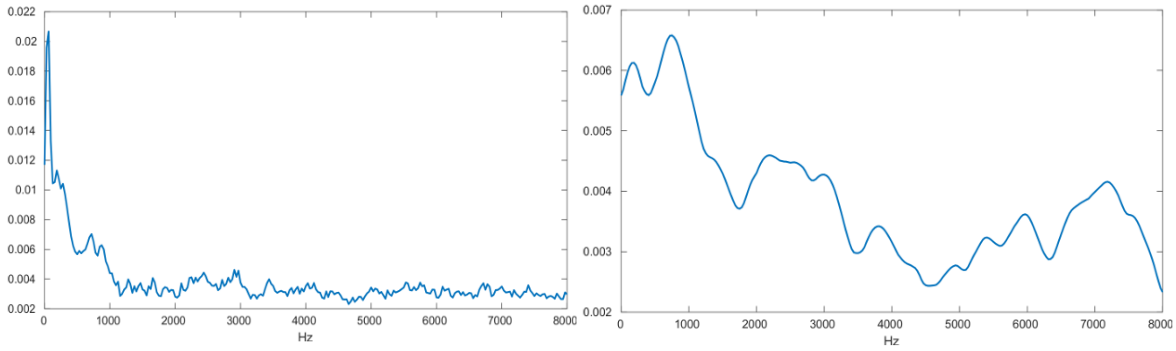

Figure 2.21: Cumulative frequency responses for convolutional layers with 300 and 30 kernel lengths, respectively.
Source: *Adopted from* [69].

The second level of analysis interpreted the filter responses as a spectral dictionary, assuming the magnitude frequency response of the input signal can be estimated as:

$$S_t = \left|\sum_{k=1}^{n_{f1}} \langle s_t, f_k \rangle \mathrm{DFT}\{f_k\}\right| \tag{2.7.3}$$

Where S_t is the magnitude of the frequency response, s_t represents the input speech signal, n_{f1} is the number of filters in the first convolutional layer, k is the filter index, f_k represents the kernel of interest, and DFT is the Discrete Fourier of Transform. By expressing the filters f_k in accordance with Fourier sines and cosines, respectively, then the magnitude of the frequency response S_t is simply the Fourier magnitude spectrum s_t. Making use of the magnitude spectrum the authors were able to show how the spectral peaks conveyed by a linear predictive spectrum coincide with the spectrum of the CNN architecture.

These ideas of spectral analysis were further investigated in a similar study that incorporated a gradient-based technique to produce attribution explanations [65]. Using guided backpropagation, the authors were able to generate a visual explanation in the form of a relevance signal. The signal is computed in the same manner as a pixel attribution map. However, instead of using the pixel values, the gradient for the model's prediction is computed against each data point in the raw waveform. The relevance of a data point corresponds to the contribution/relevance that particular point had to the prediction. The figure below shows the input signal and the relevance signal.

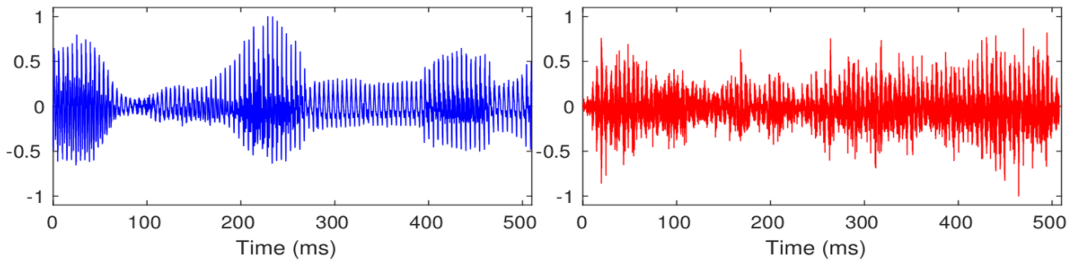

Figure 2.22: Input speech signal and the relevance signal.
Source: *Adopted from* [65].

The study points out how the explanation does not give much insight into what audio signal features influenced the model's decision. This is due to the absence of visual cues similar to those present in image explanations. However, the relevance signal provides some utility in regard to explaining the

model's attention to the periodicity of the input signal. The authors ultimately conclude the need for further explanations through spectral level analysis.

2.7.2. Interpretable architectures for raw waveform modelling.

The interpretation of models trained on raw waveforms has inspired the design of efficient neural network architectures for speech processing. Most notably, the SincNet architecture [70], a novel specialised CNN for speech processing whose first layer exploits parametrised sinc functions to discover more meaningful filters. The authors point out that the significant downside to replacing hand-crafted features with raw waveforms is a lack of interpretability of the learned filter banks. They aimed to develop more interpretable convolutional filters for processing raw waveforms.

The first layer of SincNet performs band-pass filtering of the raw waveform by convolving it using specific parametrised sinc functions. By utilising these functions, they impose constraints on the shape of the learned filters such that only the high and low cut-off frequency properties are modelled from the data. The following set of mathematical operations can describe the operations of the SincNet filters:

$$y[n] = x[n] * h[n] = \sum_{l=0}^{L-1} x[l] \cdot h[n-1] \tag{2.7.4}$$

This equation defines for a standard CNN, the result $y[n]$ of a time-domain convolution between the raw input sign input $x[n]$ and the Finite Impulse Response (FIR) filter $h[n]$ of filter length *L*.

Similarly, SincNet performs the same operation but replaces $h[n]$ with the predefined function g that depends on fewer learnable parameters θ.

$$y[n] = x[n] * g[n, \theta] \tag{2.7.5}$$

Inspired by digital signal processing, the function g is defined such that the resulting filter-bank is made up of rectangular band-pass filters. In the frequency domain, the magnitude of a generic band-pass filter can be described by the equation:

$$G[f_1, f_2] = \prod\left(\frac{f}{2f_2}\right) - \prod\left(\frac{f}{2f_1}\right), \tag{2.7.6}$$

where $rect$ is the rectangular function in the magnitude frequency domain and f_1 and f_2 are the learned low and high cut-off frequencies.

In the time domain, the function g is given as:

$$g[n, f_1, f_2] = 2f_2 sinc(2\pi f_2 n) - 2f_1 sinc(2\pi f_1 n), \tag{2.7.7}$$

where $sinc(x)$ is:

$$sinc(x) = \frac{\sin x}{x} \tag{2.7.8}$$

Limits are imposed on equation 2.7.9. to ensure that $f_2 \geq f_1$ and $f_1 \geq 0$. These limits are such that its input has the following parameters:

$$f_1^{absolute} = |f_1| \tag{2.7.9}$$

$$f_2^{absolute} = f_1 + |f_2 - f_1| \tag{2.7.10}$$

To mitigate the abrupt discontinuities at the end of function g, windowing is applied by multiplying a truncated version of the function g with the window function w. Thus, g becomes:

$$g_w[n, f_1, f_2] = g[n, f_1, f_2] * w[n], \tag{2.7.11}$$

w is the Hamming window which is defined as:

$$w[n] = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{L}\right). \tag{2.7.12}$$

The restrictions imposed on the first convolutional layer of the SincNet architecture give it several advantages over standard CNN architectures. Firstly, regardless of the filter length L the model is limited to modelling two parameters for each filter. This consequently decreases the number of parameters that need to be learned by the overall architecture making it easier to optimise. To the same effect, SincNet converges faster during training because prior knowledge about the filter shapes is already imposed on the first convolutional layer. This makes it easier to learn their characteristics.

Most importantly, the filters learned by the first convolutional layer of SincNet are more interpretable and intelligible to humans compared to that of standard CNNs. This is because direct meaning can be deduced from the feature maps that coincide with a well-grounded knowledge of the physical properties of speech. Figure 2.23 shows a plot of the cumulative frequency response obtained by adding up the filter responses for both the standard CNN architecture and SincNet, respectively. The graph results from training both models on a speaker identification task and is meant to highlight the spectral information learned by the filters.

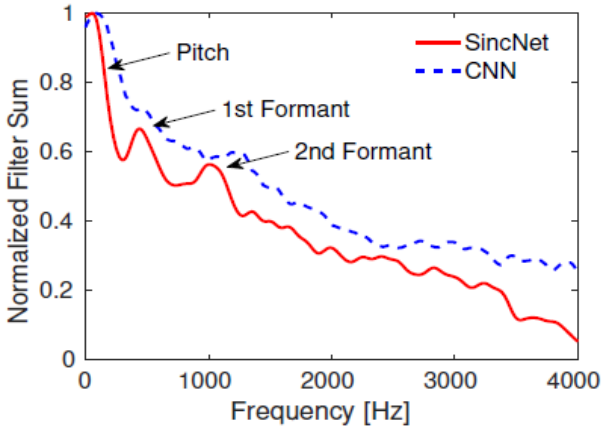

Figure 2.23: Cumulative frequency responses of CNN and SincNet on speaker identification. Source: Adopted from [70].

The first peak on the SincNet plot shows that the filters could discover information related to the pitch region. The second peak at about 500 Hz corresponds to the first formants, and the third peak captures the second formants ranging between 900 and 1400 Hz. Compared to the SincNet plot, the one for the

CNN does not give any indication that the model has learnt spectral information with any clear meaning.

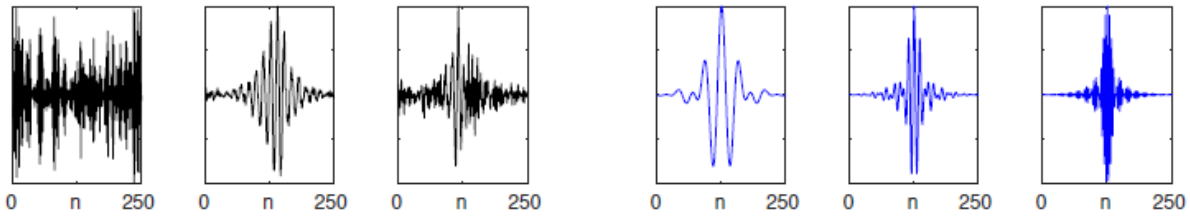

Figure 2.24*:* Example of filters learned by the conventional CNN and SincNet.

Adopted from [70].

The performance of the proposed SincNet and that of the standard CNN architecture were compared on speech and speaker recognition tasks using multiple datasets. Both models were trained and tested for the speech recognition tasks on the TIMIT and DIRHA speech datasets. The proposed model achieved a better average speech recognition performance compared to the conventional CNN model. For the speaker recognition tasks, the TIMIT and Librispeech datasets were used. SincNet performed much better[9] with lower classification error rates (CER%) for speaker identification on both datasets and a lower equal error rate (EER%) on Librispeech for speaker verification[10].

The success of SincNet inspired several efforts to investigate its utility in other speech-related tasks. The open-source child-centred diarisation system [11] aforementioned in section 1.3.1 uses SincNet filters as its feature extractor. However, unlike the original SincNet paper, this approach did not disclose any information about how the system modelled the input data and how filter responses influenced their architecture's design. In the case of diarisation of daylong recordings, such information may be essential to establish an understanding of which features of inputs influence a model to distinguish between the different speakers in a child's environment.

Similar to the aforementioned studies [65][69], by looking at the cumulative filter responses in both the frequency and time domains, the study was able to connect how the spectral information modelled by the layers corresponds to the knowledge of acoustic properties of speech. While popular interpretability techniques such as saliency methods are not common when explaining models trained on raw waveforms, spectral analysis fills this void and produces far more intelligible results.

2.7.3. Explaining models trained on time-frequency representations

Most speech processing models are trained on time-frequency representations of audio signals. These take the form of two-dimensional images of mel-spectrograms and MFCCs. This means that instead of spectral level analysis, these models can be interpreted using the visual saliency methods described in section 2.5. An extensive study explored the viability of several of these gradient-based explanations for audio classifiers [71]. The research covered several explanation methods, including vanilla gradients, integrated gradients, smooth-grad and Grad-CAM. The methods were evaluated using the four axioms initially discussed in section 2.4.6. These include translucency, portability, expressive power and algorithmic complexity. The study's findings indicate that smooth-grad provides

[9] Performance (CER%) of SincNet on TIMIT and Librispeech was 0.85% and 0.96% respectively. Compared to 1.65% and 1.00% for the conventional CNN model.

[10] The equal error rate (EER%) on the Librispeech dataset for SincNet and the conventional CNN was 0.51% and 0.58% respectively.

the optimal trade-off between these two qualities. For Integrated Gradients, the zero-amplitude baseline is more faithful, whereas the data-set-mean baseline is more resilient. Grad-CAM, on the other hand, is the least robust and faithful.

Interpretation of time-frequency representations has also inspired several model-specific explanation methods for speech applications. Most notably, the gradient-adjusted neuron activation profiles (GradNAPs) were developed for automatic speech recognition models trained on spectrograms [72]. GradNAPs were introduced to extend the event-related potential (ERP) technique used to analyse brain activity through electroencephalography (EEG). Similar to how ERP can estimate the distributed response to stimulus across the brain, GradNAPs can estimate and group neuron responses in neural network architecture to different stimuli. This holds a similar idea to the semantic dictionary discussed in section 2.6.2; however, although speech can be represented as images in the form of spectrograms, these networks' introspection (hidden semantics) cannot be approached the same way as semantic dictionaries. The result of aggregating the activations in the form of what is depicted in figure 2.16 will hold less meaning because the speech data is far more complex, and visual cues such as the understanding of the structure of objects are absent.

GradNAPs' approach is to first implement an alignment step where each layer's activations are centred with the corresponding time of spectrogram frames. This is achieved by computing the sensitivity and neuron activations in each layer for every spectrogram frame. The sensitivity, in this case, is the gradient of the prediction with respect to each layer's activations. A strong activation identifies the importance of a prediction with a high absolute sensitivity value. Hence the activations are centred at the time t of maximum $(|gradient| \odot activation)$. Ultimately, the gradients are centred so that they align with activation of the highest importance to the model's prediction.

Several studies have also explored how to adapt interpretability methods designed for computer vision models to work for models trained in time-frequency representations of speech [73][74]. Whist these work fine for the input layer, caution must be taken when adopting these techniques for hidden layers.

2.8. DISCUSSION

The literature study explores how interpretability encompasses several useful ideas that refer to different parts of machine learning systems. Depending on its application, the objectives and scope of interpretability need to be thoroughly defined. After which, one can start to explore interpretability methods that can satisfy the requirements.

To narrow down the body of the review, a concise interpretability taxonomy is established in section 2.4.6. The most important aspect of this taxonomy is the categorisation of interpretability methods based on the explanatory power of the explanations they produce.

Following this notion of the explanatory power of interpretability methods, the literature explores the different types of explanations and how these methods compare against each other. Attribution type methods are the most popular type of explanations, especially for deep neural networks. As a result, this has inspired several studies that explore the viability of attribution methods through experiments that investigate their completeness. As indicated by the comparison of attribution methods in section 2.5.3, the expected gradients methods satisfy multiple desirable axioms. Furthermore, expected gradients combine several well-grounded ideas from Shapley values, integrated gradients and smooth-grad. To the best of the author's knowledge, it is yet to be investigated if expected gradients can be used to produce meaningful explanations for speech models.

For speech processing applications, there are generally two schools of thought for interpreting their underlying models. These depend heavily on the nature of the data to which these models were trained. Attribution and hidden semantics methods can be used to generate explanations for models trained on spectrograms. For models trained on raw waveforms, hidden semantics through spectral-level analysis has been shown to present the most desirable results. Furthermore, spectral analysis of the learned filters is attractive because it provides global interpretability. It is also a direct interpretation of the network's internal learned representations.

Taking into consideration the taxonomy established in section 2.4.6 and the literature discussed on interpreting speech models, the most likely methods of interest for this study are post-hoc methods which produce attribution or hidden semantic type explanations.

Therefore, the objective for interpretability in this study is to provide explanations that explain the performance of the model and how it can be improved. The interpretability explanations must be able to verify that the diarisation model has learned speaker discriminative information. This requires detailed discussion is required to establish the speaker discriminative information as well as the effectiveness with which each interpretability approach may express this information.

CHAPTER III

3. GENDER PERCEPTION

This chapter explores the classification of speech samples into male and female classes using deep neural networks. It investigates if the expected gradients technique can be used to explain predictions for models trained on speech in both raw waveform and spectrogram forms.

3.1. PROBLEM DEFINITION

Ideally, a speaker diarisation model should be both speaker and language-independent. This means it should not experience a drop in performance when applied to different languages and speaker characteristics. Furthermore, it should operate well in both controlled and less desirable acoustic environments.

To better understand the complexity of speaker diarisation in child natural language environments one can first explore the subset problem of gender classification. This subproblem is of particular importance because the downstream processing steps require this information to estimate the amount of linguistic input contributed by the female and male caregivers respectively. Hence a diarisation model must learn features that differentiate between male and female speech.

As discussed in the conclusion of the literature review, to verify that the model has learned the relevant information, interpretability can be used to communicate if the model's decisions are aligned with established domain knowledge. In this case, the domain knowledge refers to gender-specific differences in speech acoustics.

3.2. ACOUSTIC DIFFERENCES BETWEEN MALE AND FEMALE SPEECH

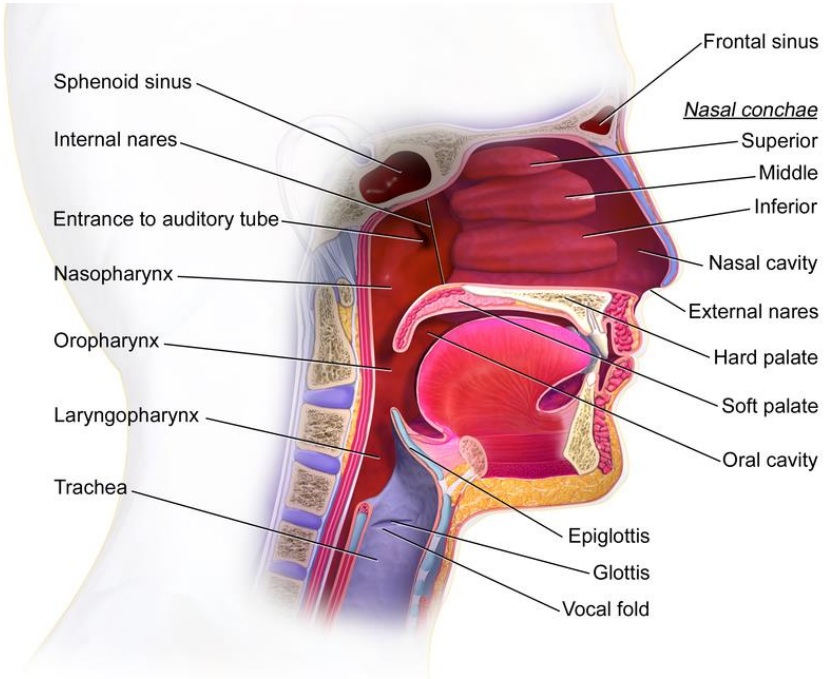

Figure 3.1: Upper respiratory system of a human showing the position of the vocal organs [11].

Source: Adopted from [75].

[11] Image obtained under the Creative Commons Attribution 3.0

The most apparent way to investigate the difference between male and female speakers is to consider the sexual dimorphisms of their vocal organs within the source-filter model of speech production. This model posits that speech results from combining a source of sound energy and a filter that modifies it. The larynx or glottis shown in figure 3.1 is a sound energy source [76][77]. At the larynx, the air expelled from the lungs flows through the vocal folds, which close and open in a quasiperiodic fashion. This opening-closing frequency of the vocal folds determines the source's fundamental frequency (F0) and the resulting pitch.

For the same age group, men are generally considered to be larger than women. One can assume that the vocal folds of women are lighter and smaller than those of men; hence they vibrate at a faster rate. This is immediately perceptible as pitch, and women are on average considered to have a higher pitch than men [78][79][80][81]. Simpson reports that for German and English speakers, males have an average fundamental frequency of 100 to 120 Hz and females 200 to 220 Hz [81].

This difference in pitch can be confirmed by looking at the placement of the harmonic frequencies between men's and women's speech on a frequency spectrum as depicted in figure 3.2. The graphs are the results of taking the fast fourier transform of male and female speech samples in their time-domain representation.

The male harmonics (frequency peaks) are closer to each other than those of a female with a lower fundamental frequency. The difference in fundamental frequency is reliable in measuring the difference between adult males and females. However, it is essential to note that the range of disparity between the fundamental frequencies between women and men differs across cultures.

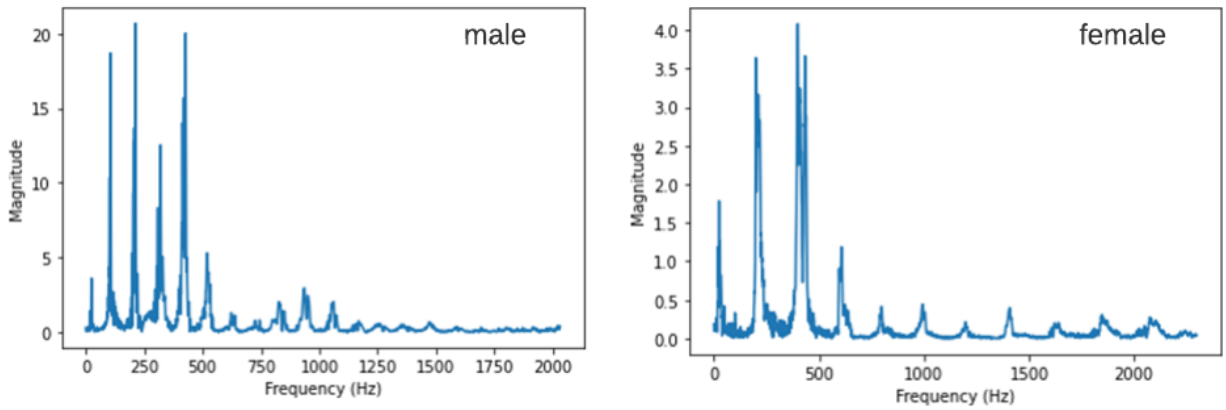

Figure 3.2: Frequency spectra for male and female speech samples for the word zero.

Another aspect to consider at the glottal source is the proportion of time the vocal folds spend in open and closed states. This ratio of opening and closing of the vocal folds is described as the Glottal Open Quotient, and it affects voice quality [82][83]. For modal voicing (the standard form of voicing), this ratio is 50/50, meaning for each pitch period, the vocal folds spend an equal amount of time in both open and closed states. Since women are considered to have thinner vocal folds there are less likely to achieve complete closure when phonating sounds. This results in a significantly different glottal open quotient for each pitch period than men [84].

The filter part of the source-filter dichotomy is the product of the size and shape of the vocal tract [76][77]. The vocal tract acts as a time-varying signal filter that suppresses specific sound energy frequencies whilst allowing others to pass. Generally, the vocal tracts of women tend to be smaller

than those of men of the same age [78][79][80], suggesting that the differences in the formant[12] frequencies scale inversely with the size of the talker [85]. If a woman and a man are made to produce the exact sound, one can see that the men's formant frequencies are scaled down with respect to the woman's formant frequencies. The figure below shows the waveforms and corresponding spectrograms of male and female speech for the utterance of the number zero. The red lines trace the formant frequencies which show that the male formant frequencies have a much lower frequency value than those of the female.

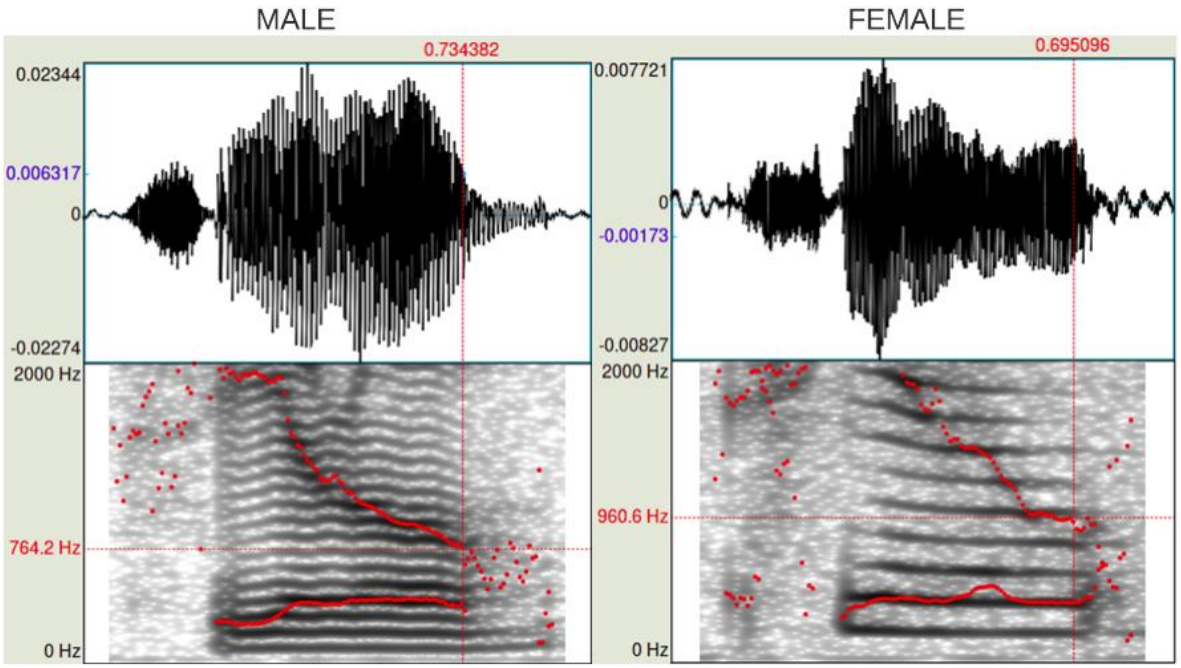

Figure 3.3: Raw waveforms and spectrograms for the utterance of the word zero for male and female speakers, respectively.

The acoustic differences between men and women cannot only be attributed to anatomical differences. For instance, the difference in voice resonance between males and females is disproportionately more significant than the corresponding differences in vocal tract size [86]. This implies that the differences in voice resonances are not entirely systematically correlated with vocal tract size. Part of this is because gender is expressed through voice acoustics that are not fully described with biological reasoning but are instead culturally learned in an archetypal manner or are somewhat arbitrary [86][87][88]. This further increases the complexity of the gender perception problem. However, the fundamental frequency and formant frequencies have been established in studies to be strong cues of speaker gender, with the fundamental frequency being the most significant contributor [86][89].

3.3. METHOD

Considering the discussion from section 2.7 of the literature review, the gender perception problem is investigated using both time-domain and time-frequency domain representations of speech. The time-domain representation is in the form of waveforms of the raw speech signals, whilst the time-frequency domain representation is in the form of spectrograms. The difference between the two

[12]The term "formant" refers to the vocal tract's acoustic resonance. They correspond to the energy-dense frequency peaks in the spectrum. They are named F1, F2, F3 and F4 corresponding to the first, second, third and fourth formants.

approaches will shed light on the limitations imposed by the input representation regarding the model's performance and the quality of interpretability explanations. Furthermore, a white-box model is introduced to serve as a baseline for interpretability. The baseline is trained on voice source parameters extracted from the raw waveforms in the dataset; these voice properties are discussed in table 3.1.

3.4. DATASET

A portion of the LibriSpeech[13] corpus was used to investigate the gender perception problem [90]. In its totality, this corpus contains 1000 hours of English speech derived from audiobooks as part of the LibriVox[14] project. The speech was recorded under controlled conditions and sampled at 16kHz. The subset used in the following experiments is the *dev-clean*[15] subset which contains 5.4 hours of speech with 20 female speakers and 20 male speakers. Each speaker in *dev-clean* contributed eight minutes of speech. No enhancement or preprocessing was applied to the data prior to training. Each audio sample was simply normalised and split into segments of 16000 data points with a 4000-point overlap. Since the audio was sampled at 16 kHz, each sample in the training dataset has 16000 data points representing 1 second of speech. Some speakers were taken from the *test-clean* subset so that the training samples could reach 30000 with 15000 samples from females and the same number for men. Overall, the dataset used for model development contained more than 8 hours of speech.

3.5. WHITE-BOX MODEL

Before looking at gender classification through black-box deep neural networks, it is essential to investigate if the problem can be solved by making use of an intrinsically interpretable white-box model. Rudin [95] pointed out that this approach is important to make sure before moving to a more complex and opaque model that the problem cannot be satisfied by a simple and inherently interpretable machine learning algorithm. In this case, the white-box model of choice is a decision tree classifier. As its name suggests, a decision tree is a machine learning model that takes the form of a prototypical tree-like structure. At each branch, a test is performed and depending on its outcome the branch subdivides into more branches splitting the data into subsets. Depending on the specified depth, the branches stop subdividing to become leaves, and these leaves carry the decision for the prediction.

3.5.1. Feature extraction for white-box model

To make the data more suitable for a decision tree, several high-level voice source estimates are extracted from each raw waveform in the dataset using the algorithms included with the VoiceLab[16] and Praat[17] software packages [91][92]. The descriptions of the features are given in table 3.1; the detailed formulations of these features are given in Appendix A.

One can perform dimensionality reduction to see if the features offer meaningful separation of the data before proceeding to train the model with these features, first by principal component analysis (PCA) [93] and then by t-distributed stochastic neighbour embedding TSNE [94]. PCA and TSNE are dimensionality reduction methods that transform high-dimensional data into a low-dimension space that conserves some meaningful patterns from the original data.

[13] https://www.openslr.org/12
[14] https://librivox.org/
[15] The subsets of the Librispeech dataset labelled as '*clean'* are considered to be less difficult and less noisy than the other subsets.
[16] https://voice-lab.github.io/VoiceLab/
[17] https://www.fon.hum.uva.nl/praat/

Table 3.1: Description of extracted speech features.

Feature	Description
Mean pitch (meanF0Hz)	Estimation of the mean fundamental frequency. The measure is determined by the autocorrelation method described in [95].
Standard deviation of pitch (stdevF0)	The standard deviation of the pitch estimations.
Average formant	The average of the estimated formant frequencies F1, F2, F3 and F4 [96].
Mean formant frequencies	The individual averages of the first to fourth formant frequency estimations.
Median formant frequencies	The individual medians of the first to fourth formant frequency estimations.
MFF	Interaction of the vocal-tract length and glottal-pulse rate [97].
Formant position (pF)	The average standardised formant value for the first four formants. The formants are standardised using means and standard deviations between sexes [98].
Fitch vocal tract length	Estimation of the speaker's vocal tract length [99].
Formant dispersion	Correlation of the formant frequency dispersion and vocal-tract length in relation to body size [99].
Harmonic-to-noise ratio (HNR)	The ratio between periodic components and non-periodic components comprising of a segment of voiced speech [100].
Jitter Measures	Measures that characterise how frequency varies from cycle to cycle [100].
Shimmer Measures	Shimmer is a form of reverberation effect in which the reverberation process incorporates enhanced harmonics from the input signal. These shimmer measures characterise the effects of amplitude variation on the sound wave[100].

Figure 3.4 shows the results of applying the PCA and TSNE algorithms respectively to the dataset of extracted features. The purple data points represent the female class samples and the yellow male class samples. Although TSNE shows a much better separation of the data than PCA, both techniques show a fair clustering of two classes as expected.

Whilst this is not exhaustive exploratory data analysis, it gives some confidence that these features or a combination of these features hold information that can be used to discriminate between male and female speakers.

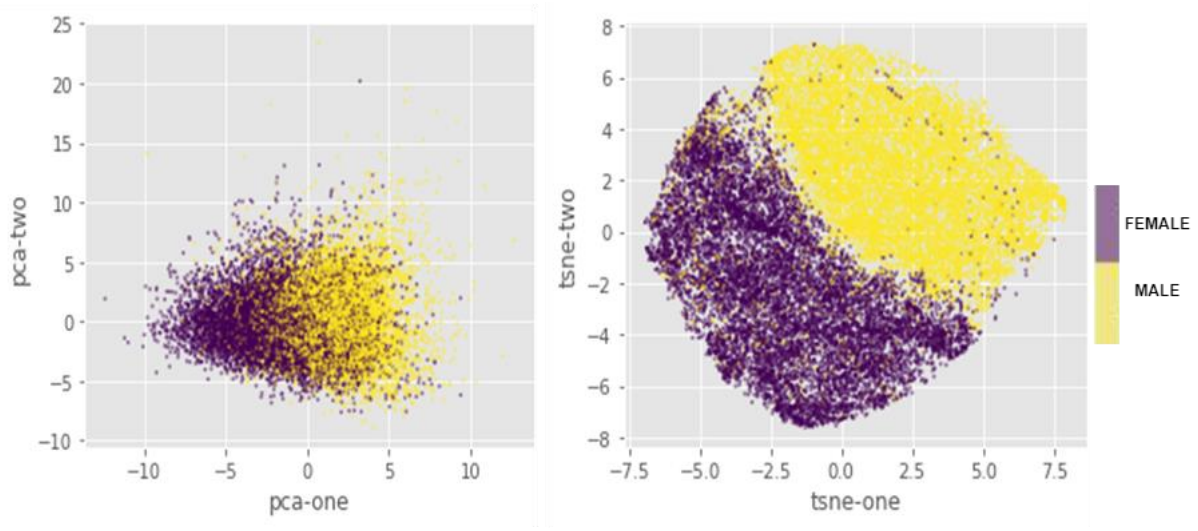

Figure 3.4: PCA and TSNE clustering on the extracted feature dataset.

3.5.2. Training the white-box model

Ideally, the main goal of training a decision tree is to learn the smallest tree that can sufficiently model the training data. The depth of the decision tree determines whether the model overfits or underfits the training data. An overfitted decision tree performs very well on the training samples but relatively worse on new unseen samples in the test set. On the other hand, an underfitted model is not complex enough to model the training data; hence its performance on the training and testing data are both undesirable. The ideal decision tree is of sufficient depth such that it performs well on the training data, and the resulting model is able to generalize well on unseen samples (test data).

To identify the suitable depth to satisfy our problem, a search is done using *k*-fold cross-validation. This is a resampling strategy where the training data is split into *k* equal folds, and the model is trained on *k*-1 folds and evaluated on the remaining fold. This process is repeated *k* times.

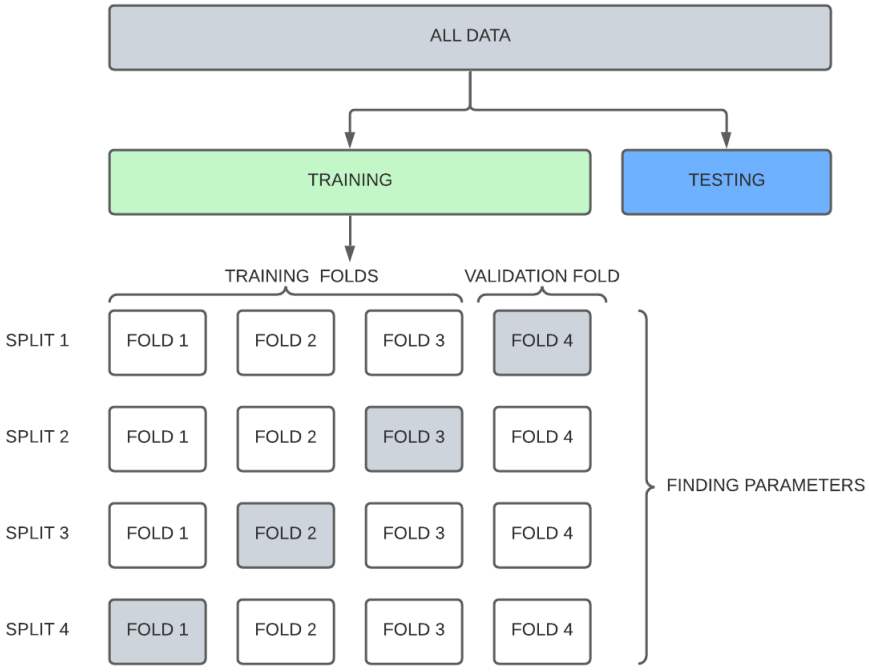

Figure 3.5: Illustration of k-fold cross-validation.

Figure 3.5 depicts an example of 4-fold cross-validation (k=4). The available data is first split into a testing and training set. The training set is now four separate folds to which for four iterations, three of the folds are used for training and the remaining fold is used for validation. The final evaluation of the model is conducted on the testing set.

The resampling nature of *k*-fold cross validation allows one to search for the optimal model parameters by fitting the model in different configurations and evaluating the performance of these model variations across the different folds. In this case, the best decision tree depth will give an acceptable performance on both the training and validation sets without a significant difference in performance on the two sets.

For this experiment, 10-fold cross-validation is conducted to train decision trees of depths 1 to 20. This means starting from a depth of one; each decision tree goes through 10 iterations of resampling and evaluation. Each of the ten folds has 2224 samples, and the testing set has 5560 samples. The steps of the training process are as follows:

iii. Split the available data into training and testing subsets.

iv. Split the training data into ten equal independent folds.

v. Select one of the folds for validation and the remaining nine are used as training samples.

vi. Train a decision tree of the desired depth on the nine training folds and validate the model on the validation fold taking note of the performance.

vii. Steps three and four are repeated until each of the ten folds is used for validation.

viii. The model's mean and standard deviation are derived by adding all of the model scores calculated in step five for each of the ten folds.

ix. Steps three to six are repeated for decision tree depths from 1 to 20.

x. A tree depth is then chosen for the model that produces the most optimal mean and standard value of the model scores.

xi. The optimal tree depth is then trained using the entire training set and evaluated using the testing set created in step one.

Another aspect to consider about a decision tree's depth is the resulting model's interpretability. Increasing a decision tree's depth consequently increases the model's complexity making it less interpretable. Using Lipton's framework [41], the model loses interpretability transparency in the two axioms described in section 2.3.1, simulatability and decomposability. It is much more difficult for a human to comprehend and follow the steps that produce the model's estimations.

3.5.3. Performance of the white-box model

Figure 3.6 shows a graph of the mean cross-validation accuracy and the training accuracy as the decision tree depth increases. From tree depths two to eight, as the depth of the decision tree increases, the corresponding mean cross-validation and training accuracies increase. From the tree depth of nine, the mean cross-validation accuracy starts to plateau whilst the training accuracy increases until the tree depth of 12. Here the decision tree has a perfect accuracy on the training data whilst there is no change in the mean cross-validation accuracy. This signifies that the model has

overfitted on the training data, suggesting that each leaf of the decision tree has precisely one sample, and the sample's class equals the prediction.

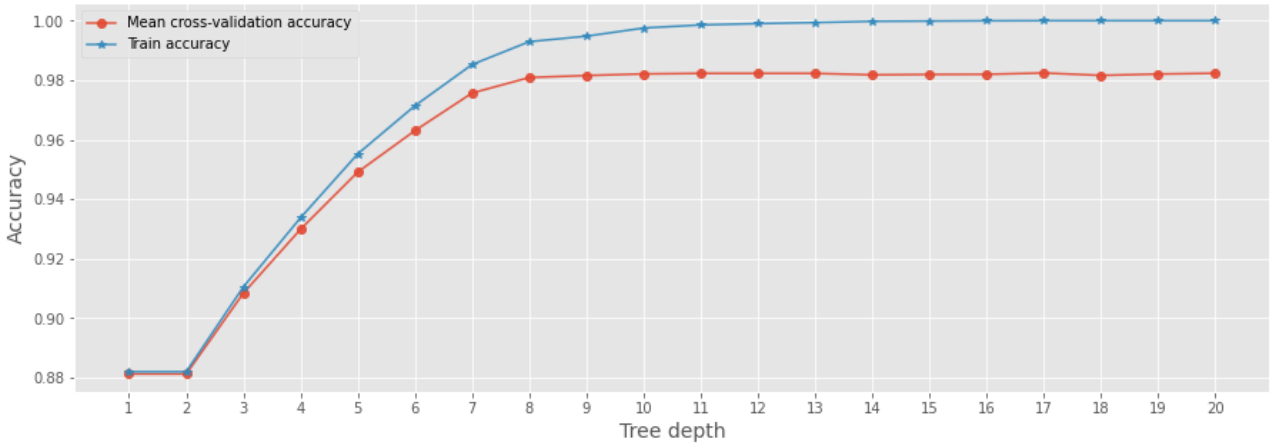

Figure 3.6: Mean cross-validation accuracy and training accuracy per decision tree on training data.

The graph also shows that the optimal depth of the decision tree is nine; at this depth, the mean cross-validation stops increasing with the increase in depth, and the model has not yet reached a perfect classification accuracy on the training data. The model achieves 98.5% accuracy on the holdout test set described in steps one and nine of the training process.

The confusion matrix in figure 3.7 depicts the performance of the resulting decision tree on the hold-out test set. Thirty-three of the male samples were misclassified as female, and conversely, fifty-three female samples were incorrectly classified as male. The corresponding Table 3.2 gives the performance metrics derived from the confusion matrix as discussed in section 1.8. The model has a slightly lower precision score for the male class signifying that the model has more false-positive classifications for this class. On the other hand, the female class has an overall lower f1-measure and recall indicating the model has more false negatives collectively for this class.

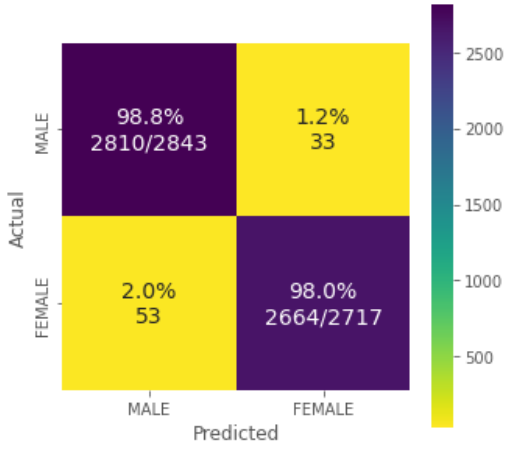

Figure 3.7: Confusion matrix for decision tree.

Table 3.2: Precision, recall and f1-scores for the decision tree.

Class	Precision	Recall	F1-score
male	0.98	0.99	0.99
female	0.99	0.98	0.98

3.5.4. Interpretation of the white-box model

One of the main advantages of using a shallow decision tree is the ease of representation of its entire decision logic. Figure 3.8 shows a truncated visualisation of the decision tree. The first decision is made based on the mean fundamental frequency. This is encouraging because, as discussed in section 3.2, the fundamental frequency (F0) is the most reliable cue for speaker gender, followed by the formats. This is also an indication as to why the decision trees with a relatively low depth achieve acceptable performance; in figure 3.6, the decision trees with a depth of one and two achieve a mean validation accuracy of 88%.

The first node of the decision tree is referred to as the root node (the top of the tree at depth zero). Given a set of speaker features, this node will check whether the mean fundamental frequency (meanF0Hz) is smaller or equal to 156.66Hz. If this is true, the decision flow moves to the root's right child node at depth one. At this node, another check is performed to check whether or not the mean fundamental frequency is smaller or equal to 140.513Hz. Again, if this is true, the class of the speaker is most likely male, and the decision flow moves to the next root node. These checks are performed until a leaf node is reached; this is where a node does not divide into any child nodes, and the decision of the class is given without any further checks.

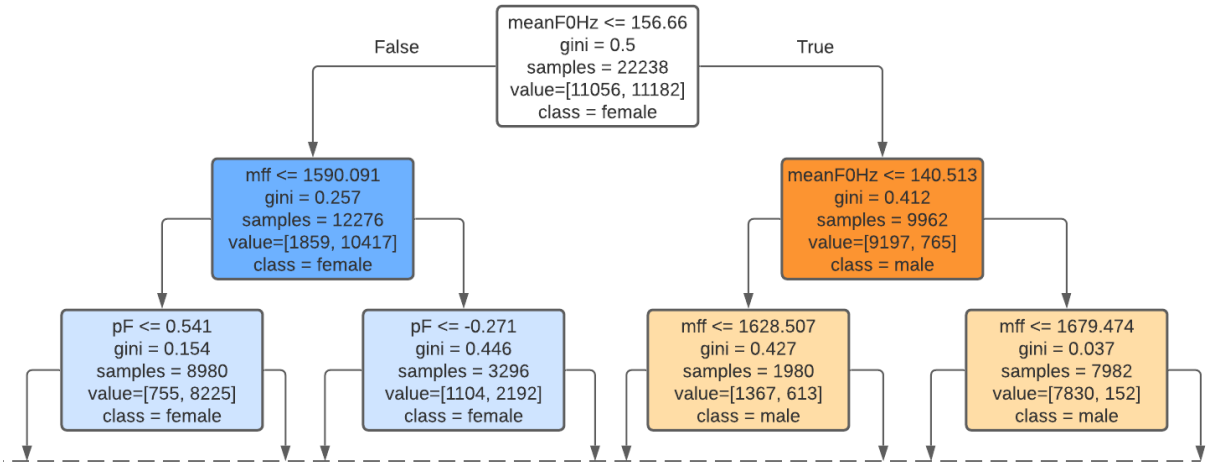

Figure 3.8: Truncated visualisation of the decision tree.

At each node, the *samples* property indicates the number of training instances that are applicable to it. For instance, the number of samples at the root node is equal to 22238. The *value* property at each node refers to the number of samples of each class the node applies. For instance, on the right side at depth one, the node applies to 9197 male samples and 756 female samples. The *gini* attribute refers to the impurity of a node. This is such that a node is referred to as pure if the *gini* is equal to zero; this would mean that all the training samples belong to the same class. The *gini* or *gini* impurity is determined by:

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2, \tag{3.5.1}$$

where $p_{i,k}$ represents the ratio of class *k* instances among the training instances in the *ith* node.

To get a more informative perspective of the model's decision making, the interpretability of the decision tree model is explored using the SHAP values as demonstrated in section 2.5.1.2. Firstly, one would like to visualise the model's decision logic for a single prediction (local interpretability).

The image below shows a force plot prediction on female speech features. The red bars represent how much a feature pushes the prediction towards a particular class, and the blue bars represent how much that feature decreases prediction towards that class. The base value represents the mean prediction and *f(x)* represents the prediction taking into consideration all features. In this case, the model's decision towards a female prediction (1) is mostly influenced by the mean pitch/fundamental frequency (meanF0Hz), the MFF and the average formant frequency (avgFormant). The formant position (pF) prediction swayed towards a male prediction (0).

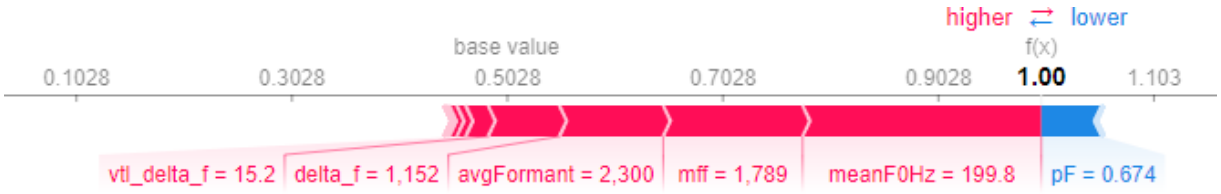

Figure 3.9: Force plot explanations for a female class prediction.

The force plot is primarily useful for understanding the model's prediction for a single or multiple limited instances. This is especially useful for understanding misclassifications which will be demonstrated later in this section. To achieve interpretability from a global perspective, one can use the bar and summary plots introduced in section 2.5.1.2.

The figure below is a global bar plot depicting the mean absolute SHAP values for each feature across the data. This helps illuminate how on average, the magnitude (positive or negative) of each feature's contribution toward the predicted speaker gender. The higher the mean absolute SHAP values for a feature, the more influential that feature is. According to the bar plot, the mean fundamental frequency (meanF0Hz) is the most influential feature, followed by the interaction of the vocal-tract length and glottal-pulse rate (MFF), and the formant position (pF).

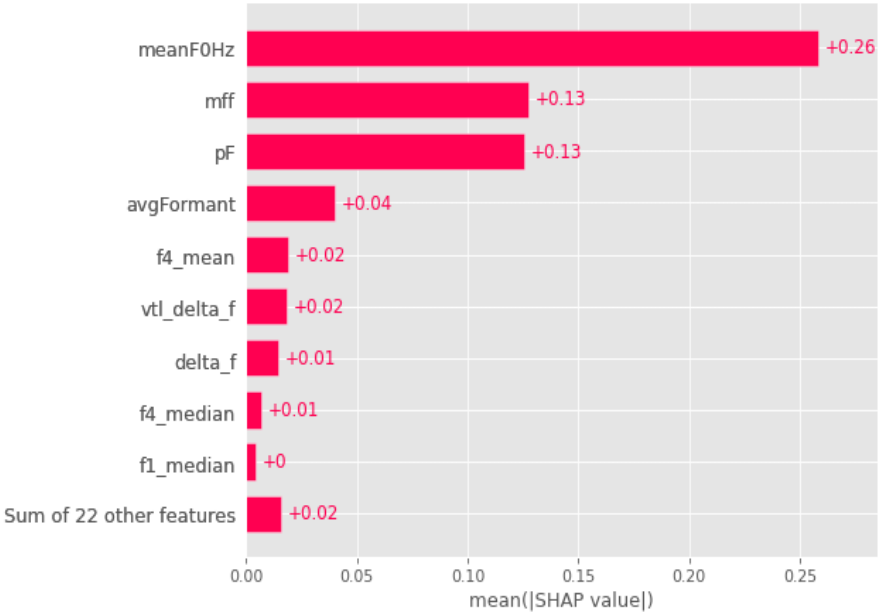

Figure 3.10: Bar plot showing the mean absolute SHAP values for each variable.

The summary plot provides a more informative representation than the global bar plot. It gives the direct relationship between the magnitude of a feature and its impact on the model's output across the whole dataset. The left axis of the summary plot ranks the input features from top to bottom according

to their mean absolute SHAP values, similar to the global bar plot. The x-axis gives the SHAP value, and the colour bar corresponds to the magnitude of the input feature itself.

The summary plot in figure 3.11 highlights that the model is hugely influenced by a relatively small portion of the features, with the mean fundamental frequency (meanF0Hz) being the most important feature. The higher the fundamental frequency (meanF0Hz) values, the higher the impact on the model's output and the most likely speaker class is female. This coincides with the visualisation of the decision tree depicted in figure 3.8, the first branch checks whether the mean fundamental frequency is greater or lower than a certain value.

The second feature of importance is MFF; as described in table 3.1, this represents the interaction between the vocal tract length and the glottal-pulse rate. As discussed in section 3.2, the glottal pulse rate or glottal quotient determines voice quality, and vocal tract length determines the formant frequencies. The relationship between the two characteristics was proved to be robust for inferring sex and age categorisation [101]. The next two important features are the formant position (pF) and the mean of the first four formant frequencies (avgFormant).

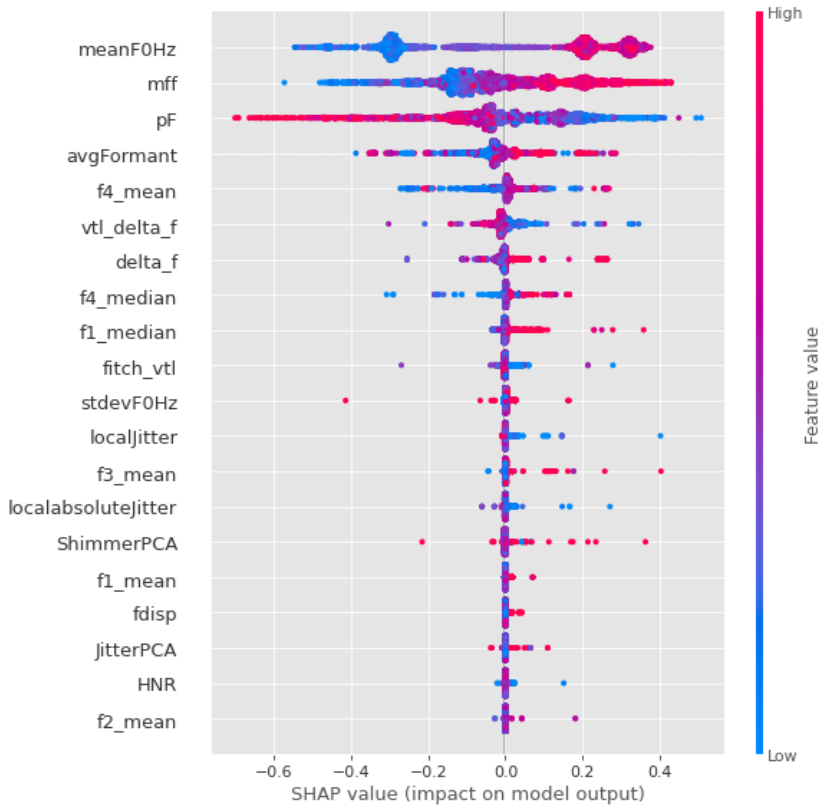

Figure 3.11: Summary plot showing the magnitude of each feature and its impact on the decision tree's prediction across the whole dataset.

The decision tree model performs quite well on these features, and its decisions are based on the domain knowledge established on gender differences in speech. The four most important features highlighted by the global bar and summary plots are all descriptive of the differences in sexual dimorphisms of male and female vocal organs within the source-filter model of speech production.

The next step is to understand the conditions that force the model to fail. This can be investigated by looking at the attribution explanations for misclassifications. The image below shows an explanation for two misclassified instances. For the uppermost force plot, the target class was female (1), but this

instance was estimated as male (0). The second force plot shows a misclassification for a male (0) instance estimated as female (1).

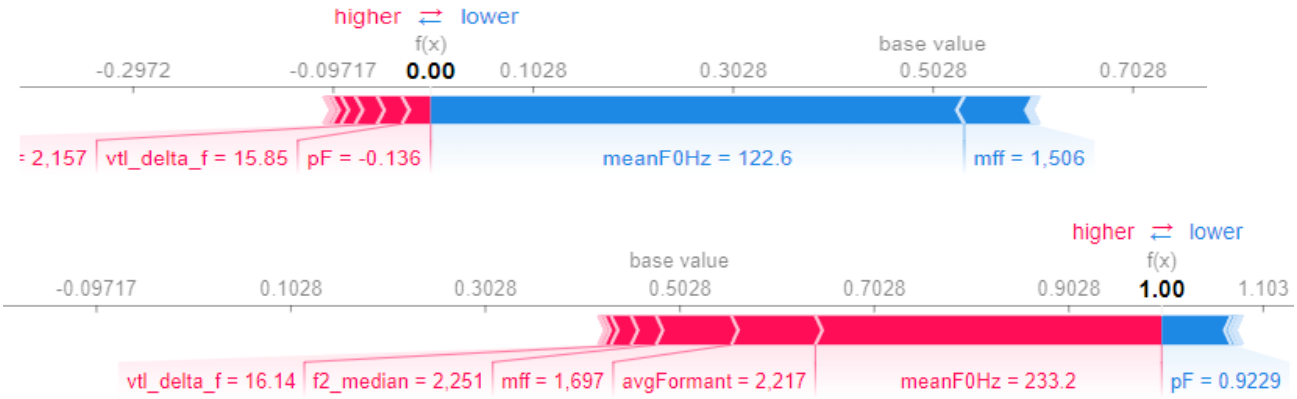

Figure 3.12: Explanation for misclassifications.

For both force plots, the largest contributing feature to the misclassifications is the mean fundamental frequency (meanF0Hz). To try and understand why the model fails for these instances, one can look at the distributions of the mean fundamental frequency for both the male and female speakers in the dataset.

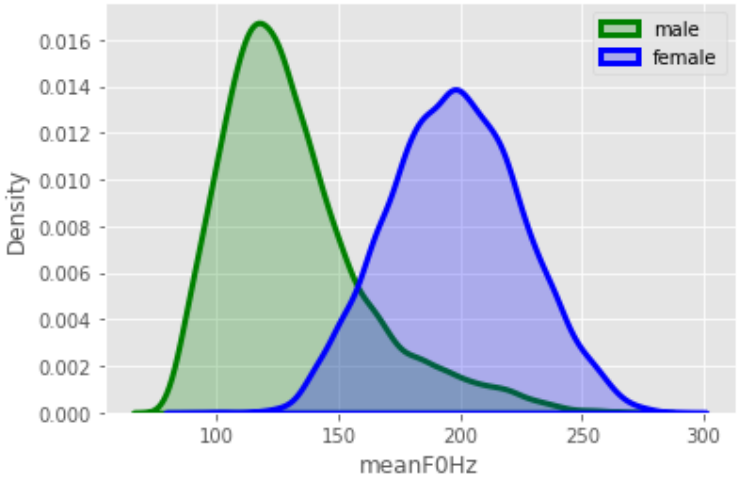

Figure 3.13: Distributions of the male and female mean fundamental frequencies, respectively.

The distributions show that most male instances in the dataset had a mean F0 between 100Hz and 150Hz, while female instances were between 150Hz and 250Hz. The mean fundamental frequency for the female instance misclassified as male is 122.6Hz, which comparing the two mean F0 distributions of male and female speakers, is more likely to be estimated under as male than female. The misclassification makes sense because the estimated mean F0 is too low to be considered female, and the other remaining features had too little influence to push the model towards the correct prediction. The same is true for the male instance misclassified as female. The mean fundamental frequency for this instance is 233.2Hz, which is most likely to be considered under the female distribution. These cases represent outlier samples within the dataset, either the speakers truly do not fall into their perceived gender category, or the feature extraction algorithm failed to estimate the speaker properties correctly.

3.5.5. Assessing the white-box model approach

The white-box model represents the ideal approach to solving the gender perception problem. The input features hold semantic meaning, making the input data inherently interpretable. The model itself is transparent, and its limitations are known.

There are several ways the findings of interpretability can be used to improve upon the decision tree model. The most apparent is the removal of unnecessary speech features. By looking at the summary and bar plots, one can opt to remove speech features with a relatively low impact on the model's output. These features are identified with relatively low mean SHAP values. In this case, it is possible to only retain 8 features out of 32. The selected features have a mean SHAP value greater than 0.01 indicated on the global box plot.

Retraining the decision tree on these retained features produces a model with competitive performance. The confusion matrix for the model is shown in the figure below. Compared to the model trained on all features, this confusion matrix shows that this model experiences less misclassification for the female class but more for the male class. Roughly the same performance is achieved by using significantly fewer features, and this also reduces the number of features that need to be extracted in the feature extraction step.

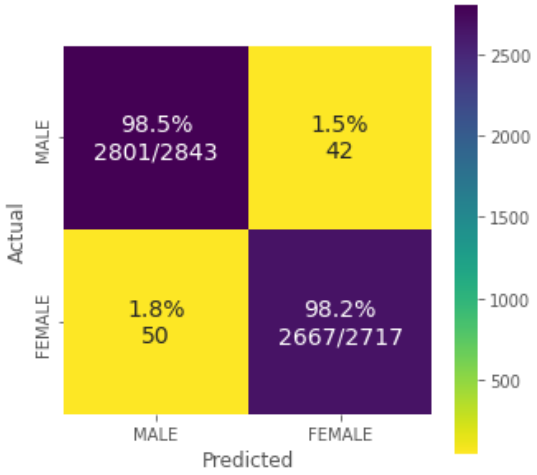

Figure 3.14: Confusion matrix for decision tree trained on fewer features.

Another attractive property of a shallow decision tree is that rule extraction is a relatively easy process. Depending on the decision tree's depth, the full decision logic can be expressed using simple IF-THEN rules that a human can comprehend. This attests to the model's transparency.

The major drawback to the decision tree approach is that the model's performance heavily depends on the performance of the feature extraction algorithms. In some cases, the feature extraction algorithms could not extract speaker features from some samples, reducing the total samples available for model development.

Ideally, one may want to develop a model that directly learns from the input itself; this is the attractive property of learning directly from raw waveforms. As mentioned in section 2.7.1, feature extraction can introduce bias when there are gaps in human knowledge.

3.6. CLASSIFICATION OF RAW WAVEFORMS

The classification of raw waveforms is investigated through four proposed CNN[18] architectures [102]. All four architectures are variations of the SampleCNN architecture first introduced for music classification [103] and extended for ASR acoustic modelling [104]. The main difference between SampleCNN based architectures and most CNN architectures for modelling raw waveforms is the difference in filter sizes implemented in the first convolutional layers. SampleCNN architectures have smaller filter sizes, which take 2 or 3 samples compared to 251 samples used by architectures like SincNet[70]. The smaller filter sizes protect the network from possibly learning the same filter shapes at different time periods. Furthermore, the filter lengths influence the potential bandwidth response for the convolutional layer. Certain filter lengths are known to favour specific frequency information.

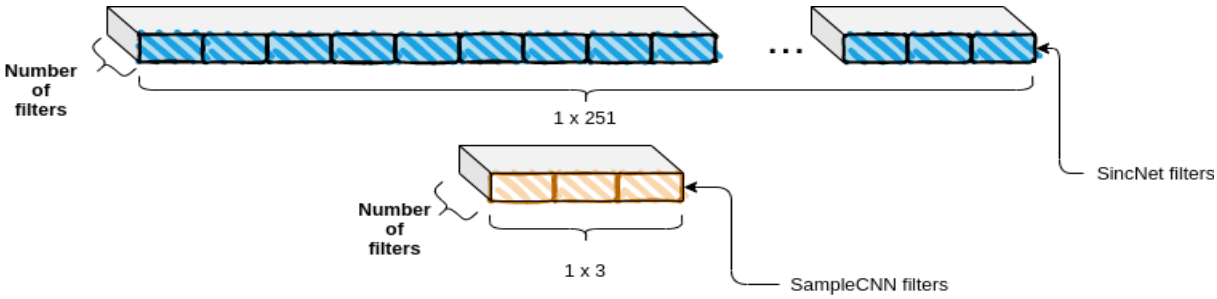

Figure 3.15: Difference between filter shapes of SincNet layers and SampleCNN layers.

All four proposed architectures implement at least one 'distinct' convolutional block, either the basic block or the squeeze and excitation block. These two blocks were introduced for multiple audio-related tasks, including keyword spotting and acoustic scene tagging [105]. The basic block is depicted in the figure below, showing the convolutional layer that accepts the input, followed by a batch normalisation layer, a ReLU activation layer, and a max-pooling layer.

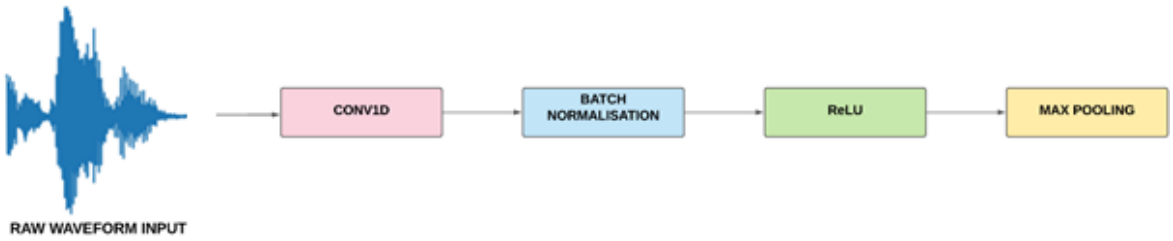

Figure 3.16: The different layers that make up a unit of a basic convolutional block.

Source: Adapted from [105]

The Squeeze-and-Excitation "SE" block extends the basic block by adding the squeeze and excitation operations. The squeeze operation extracts a channel-wise statistic via the global average pooling layer over time [106]. This means the output feature map of the basic-block layer with dimensions $C * T$ is reduced to $C * 1$ channel-wise statistics. T and C represent the dimensionality in time and filter channels. These statistics are averages of the filter activations over time for each channel, and the outputs of the basic blocks usually are greater or equal to zero. The excitation operation then takes these channel-wise statistics and determines the weight of each channel. This is referred to as excitation. These weights are learnt by the two fully connected layers to which the first one has a

[18] https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

ReLU activation and the other a sigmoid. The scale block rescales the channel-wise statistics *C * 1* back to the dimension *C * T*. The SE block allows the network to explicitly model the interdependencies between channels [105][106].

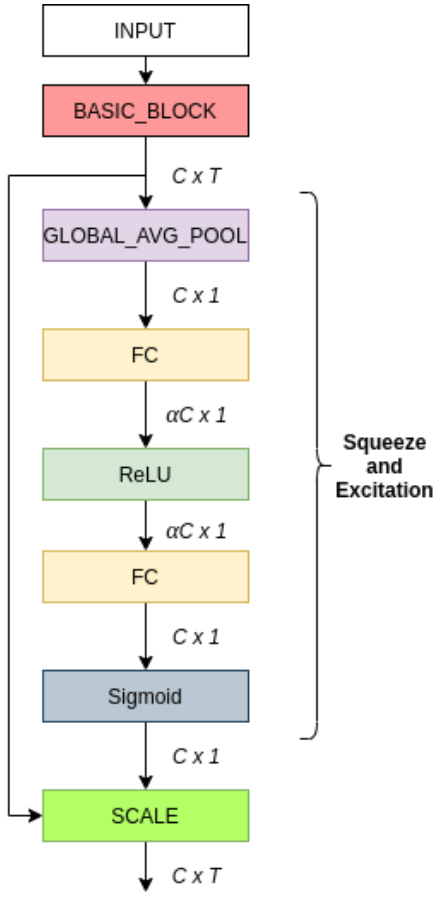

Figure 3.17: The different layers that constitute a squeeze excitation block.
Source: *Adapted from* [105]

The dimensions of the fully connected layers can be adjusted by changing the amplifying ratio *α.* This determines the extent to which the channel dimensionality is expanded or reduced. During this investigation, the hyperparameter *α* is kept the same as in the original implementation at 2^4 [105].

The first two architectures that were investigated are shown in figure 3.18. The only difference between these two is the use of basic blocks within one and SE blocks within the other. Both model types were trained and evaluated in a *k*-fold cross-validation setting. Taking into consideration the small nature of the dataset, the *k* value that was chosen was four. This will ensure that every sample in the dataset will have a chance of appearing in both training and evaluation sets. The resulting models will have a better chance of avoiding overfitting.

Similar to the depiction shown in figure 3.5, the final evaluation of the model is conducted using the holdout test set. The training was performed using the Adadelta[19] [107] optimiser with a learning rate of 0.001, a ρ value of 0.95 and an Ɛ value of 10^{-7}.

[19] https://keras.io/api/optimizers/adadelta/

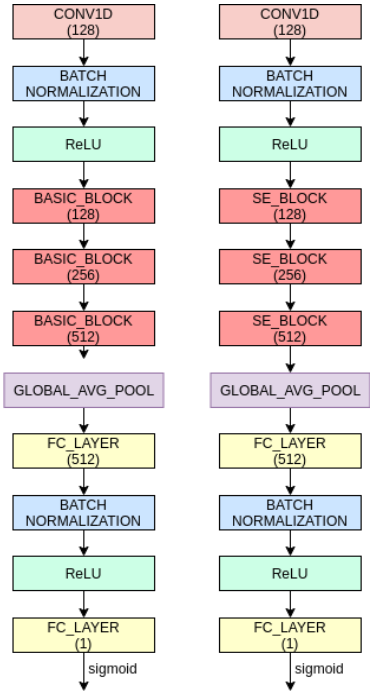

Figure 3.18: SampleCNN architectures with (a) Basic blocks and (b) Squeeze excitation blocks.

The second set of investigated architectures replaces the global average pooling layer of the first two models shown in figure 3.18 with a stack of Bidirectional Long Short-Term Memory (BiLSTM) layers[20] [108][109]. The BiLSTM's ability to capture sequential information allows the network to disentangle temporal information from features extracted by the convolutional layers [110]. In this particular case, the BiLSTM layers receive the rescaled channel-wise statistics $C * T$. To reduce the difficulty during training, only 56 BiLSTM units were employed for both models. The two architectures were trained using the same configuration as the previously mentioned models.

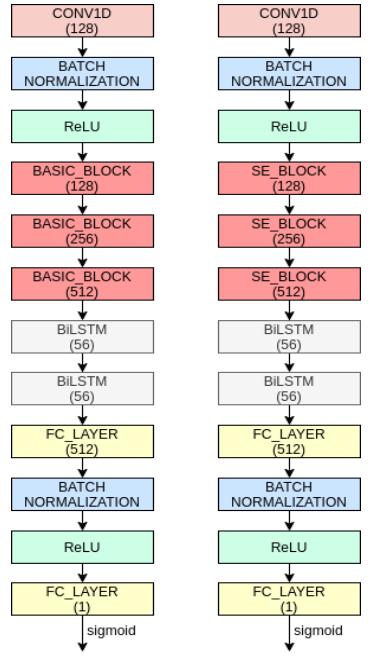

Figure 3.19: SampleCNN architectures with BiLSTM layers for sequence learning.

[20] https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

3.6.1. Classification performance on raw waveforms

Table 3.3 gives a summary of the classification results of all four models in terms of mean and standard deviations across all four folds. It was sufficient to train each of the architectures for 200 epochs; training for longer did not yield any significant increase in performance.

The mean training accuracy is the average performance of the model across all four training folds. Likewise, the mean validation accuracy is the average performance of the model across all four validation folds, and the mean testing accuracy is the average performance of each model on the testing holdout dataset.

Table 3.3: Classification performance of SampleCNN architectures in terms of mean and standard deviations across all four folds.

Model	No. of parameters	Mean training accuracy (%)	Mean validation accuracy (%)	Mean testing accuracy (%)
Basic block	811, 393	96.06 ± 0.00085	94.92 ± 0.002	94.29 ± 0.08
Basic block with BiLSTM	890, 017	96.75 ± 0.007067	96.365 ± 0.0078	95.87 ± 0.0697
SE block	898, 417	96.18 ± 0.003123	95.04 ± 0.0026	94.83 ± 0.29
SE block with BiLSTM	977, 041	97.05 ± 0.0026	96.81 ± 0.002396	96.35 ± 0.39

All four models have a relatively acceptable performance. However, to get a more informed perspective on how the models perform against each other, one can make use of box plots. The figure below shows the box plots for the validation score for each model across all four folds.

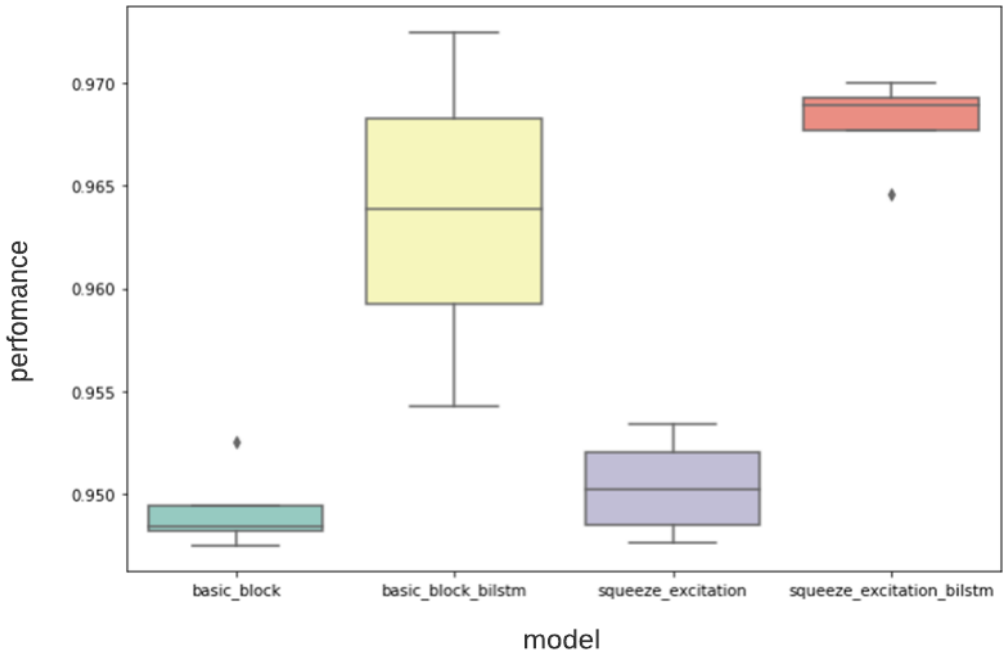

Figure 3.20: Box plots for the validation score for each model across all four folds.

The box plots indicate that the models that have sequential learning via BiLSTMs perform slightly better than their counterparts. This suggests that the employment of sequence learning within the architecture has a more significant impact than swapping out the basic convolution block with a squeeze excitation block.

The performance of the basic block BiLSTM and squeeze excitation BiLSTM models have comparable performance. The mean performance of the squeeze excitation BiLSTM models is slightly higher than that of the basic block BiLSTM models. To remove any ambiguity as to which of the two models is the better model, a paired t-test is conducted between the two classes of models to determine if the difference in the mean validation performance is significant. The setting for the hypothesis testing is as follows:

i. Null hypothesis (H0) - there is no significant difference in the mean validation accuracy of the basic block BiLSTM and squeeze excitation BiLSTM models.

ii. Alternative hypothesis *(*H1*)* – there is a significant difference in the mean validation accuracy of the basic block BiLSTM and squeeze excitation BiLSTM models.

iii. P(D|H0) determines the *p*-value; the lower the *p*-value, the greater confidence in rejecting the null hypothesis (H0).

iv. A *p*-value of less than 0.05 (usually $\leq$ 0.05) is considered statistically significant. It offers strong evidence against H0, as there exists less than a 5% chance H0 is correct (and the results are random). As a result, H0 is rejected in favour of H1.

The results of the t-test give a *t*-value of -1.4180 and a *p*-value of 0.25. The *p*-value gives strong evidence against the alternative hypothesis, indicating that there is no significant difference between the performance of the two classes of models. However, it is important to note that the variance in performance of the squeeze excitation BiLSTM models is smaller than that of the basic block BiLSTM models.

The image below gives the confusion matrix of the squeeze excitation BiLSTM model on the holdout test set. Table 3.4 shows the corresponding classification metrics derived from it. Whilst the f1-measure for the two classes is the same, the model experiences slightly more false-positives for the female class and false-negative for the male class.

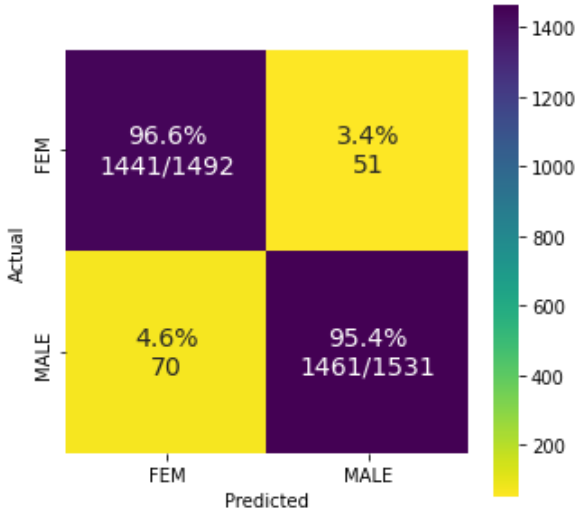

Figure 3.21: Confusion matrix for the SE block with BiLSTM model.

Table 3.4: Precision, recall and f1-scores for the SE block with BiLSTM model.

Class	Precision	Recall	F1-score
female	0.95	0.97	0.96
male	0.97	0.95	0.96

3.6.2. SampleCNN model interpretation

The interpretation of the SampleCNN models takes two forms, first through attribution explanations and then spectral analysis through hidden semantics. To limit the discussion, explanations are only explored for the SE block with BiLSTM model.

The attribution explanations are generated using the expected gradients method discussed in section 2.5.2.7 [60]. This method was chosen because it satisfies implementation invariance and completeness interpretability properties, among other desirable axioms discussed in section 2.5.3.

Figures 3.22 and 3.23 show scatter plots representing male and female speech, respectively. These samples are recordings of the word zero which were first introduced in figure 3.3. The highlighted points show positive and negative attributions that contributed to the model's prediction. The positive attributions represent elements of the raw waveform that push the model towards the predicted class, and conversely, the negative attributions push the model away from the predicted class. For these two examples, the model correctly predicted true labels for the male and female utterances.

As indicated by the explanations, the model 'focuses' on the outermost part of the waveforms. This can be representative of the waveform's envelope (outermost outline), the slow change in the waveform's intensity across time or the overall amplitude contour of the sound [111].

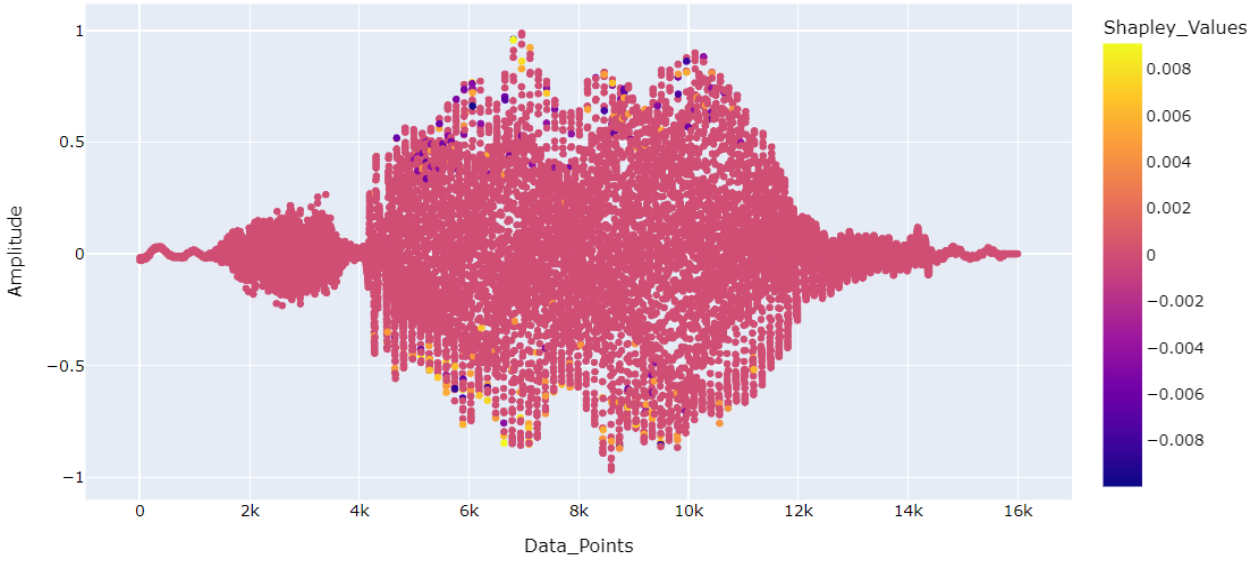

Figure 3.22: Scatter plot representation of one second's worth of male speech. Each of the 16000 data points is colour-coded with Shapley value attributions for the true class label.

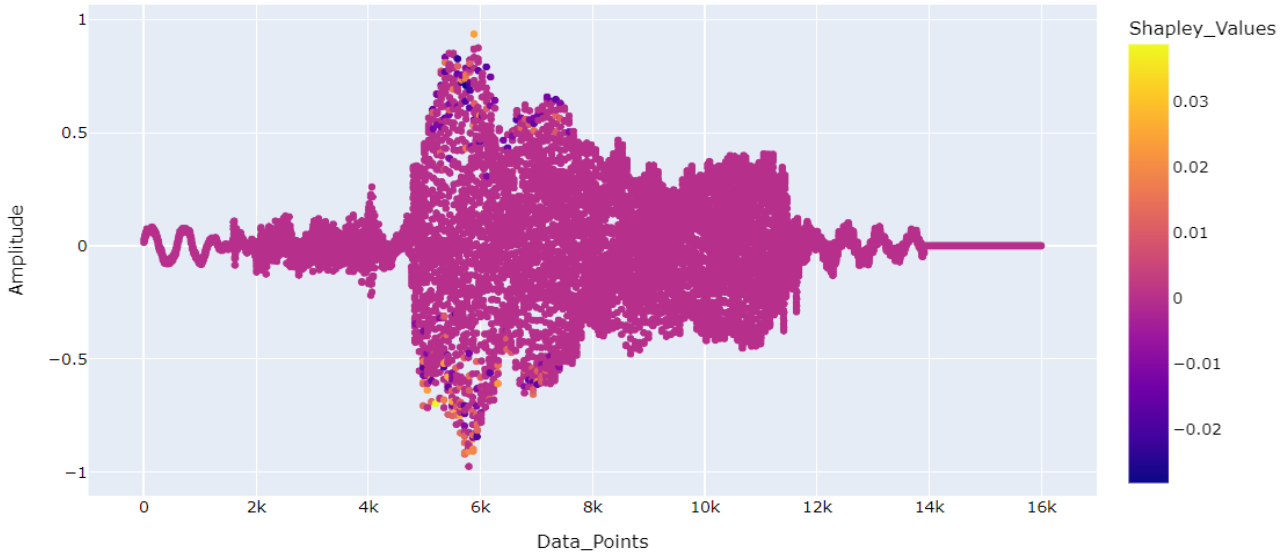

Figure 3.23: Scatter plot representation of one second's worth of female speech. Each of the 16000 data points is colour-coded with Shapley value attributions for the true class label.

Whilst the idea of the envelope appears to be captured by the model, other temporal categories associated with it, such as the fine structure and periodicity, are not immediately represented in the explanation. The envelope, periodicity and fine structure properties make up the framework for describing temporal information in speech [111].

The temporal fine structure describes fast changes, such as details corresponding to high frequencies within the signal. Periodicity represents the presence of voicing (the periodic and aperiodic vibration of the vocal cords in the larynx) and cues for intonation or pitch. Periodicity tracking can be used to compare the fundamental frequency. A slower rate of repetition represents a lower F0 whilst a higher rate of repetition represents a higher F0**.** To be fair to the model, parts of the waveform that are closer to zero or have very little periodicity are given no attribution, but there is no true indication that the model has an understanding of periodicity (presence of voicing).

Similar to findings in previous studies, this type of attribution explanation is unsatisfactory because the visualisation does not give obvious cues as to what the model is focusing on. For convolutional layers trained on images, the visualisations are very attractive because it is easy to ascertain that the model is focusing on the correct ideas. This idea also translates to the aggregation of the hidden layers via semantic dictionaries as discussed in section 2.6.

The spectral level analysis is performed similarly to the research discussed in section 2.7.1. Figure 3.24 shows the cumulative frequency response for all the learned filters in the first convolutional layer. The first convolutional layer is particularly of interest because as the layers go deeper, the corresponding layers learn more complex features. This layer receives the high dimensional input and learns low-level features, which are then modelled by the deeper layers. In previous studies, the first convolution layer was shown to learn spectral level information corresponding to speaker discriminative features [65][69][70].

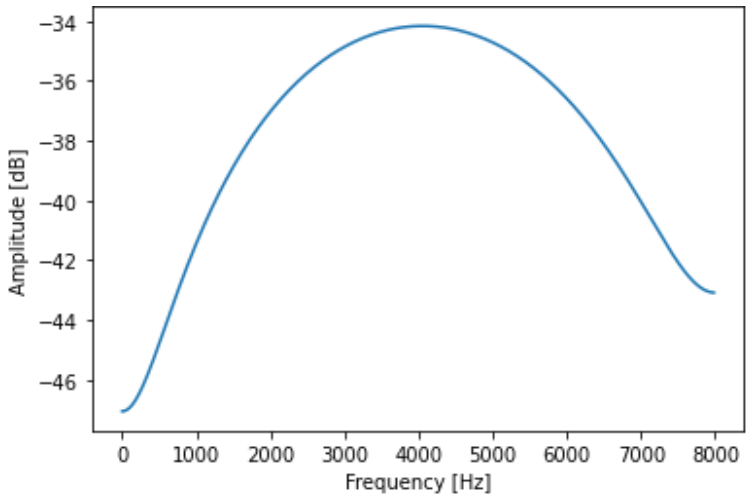

Figure 3.24: Cumulative frequency response.

This response was determined by:

$$F_{cum} = \frac{1}{n_{f1}} \sum_{k=1}^{n_{f1}} F_k, \tag{3.6.1}$$

Where n_{f1} denotes the number of filters in the first convolutional layer, and F_k denotes the frequency response of the filter f_k.

The cumulative frequency response shows which frequency ranges the learnt filters can cover. In this instance, the response looks rather rudimentary compared to the response illustrated for the SincNet filters in figure 2.23. This is due to the very small filter lengths of the SampleCNN architectures. As expected, the shorter kernel lengths emphasise learning speaker discriminative information related to the formants as opposed to the lower frequency range, which contains fundamental frequency information. This is consistent with the findings of previous research [65][69].

3.7. CLASSIFICATION THROUGH SPECTROGRAMS

Gender classification through spectrogram was investigated via two CNN architectures. Both architectures employ the concept of residual blocks[21] [112]. These allow inputs to propagate faster across layers through residual connections, enabling information flow from the first to the last layers [113].

Figure 3.25 shows an example of a residual block with a one-by-one skip connection. The information from the input flows in two different paths. One of the paths is via a conventional straight flow of information in a neural network from layer one to layer two to layer three and so forth. The second path is referred to as a shortcut or skip connection because it 'skips' intermediate layers and is added to a later layer within the network.

The one-by-one convolution carried out by the skip connection minimizes the number of parameters and the number of matrix multiplications because of the bottleneck it creates. The goal is to make residual blocks as thin as feasible in order to maximize depth and reduce the number of variables. Architectures that employ residual blocks can be referred to as 'ResNet like' architectures after the original residual network implementation [112].

[21] http://d2l.ai/chapter_convolutional-modern/resnet.html

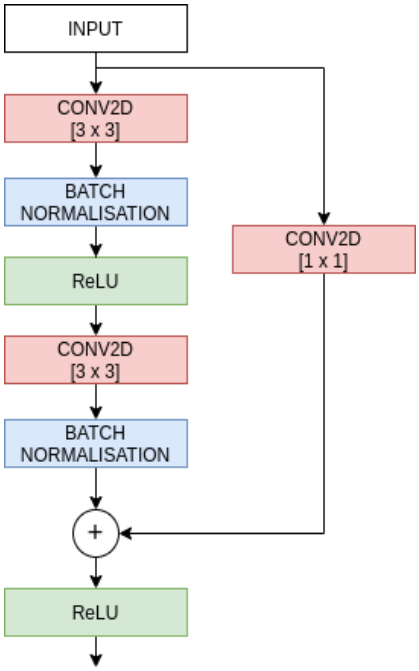

Figure 3.25: Bottleneck residual block.

Similar to the set-up of classification of raw waveforms, one of the architectures employs sequence learning. However, instead of learning sequential information from the features discovered by the ResNet blocks, the input is fed to directly a stack of BiLSTMs. Features discovered by the ResNet and BiLSTM blocks are then concatenated and fed to a dense output layer. The following figures show the two architectures.

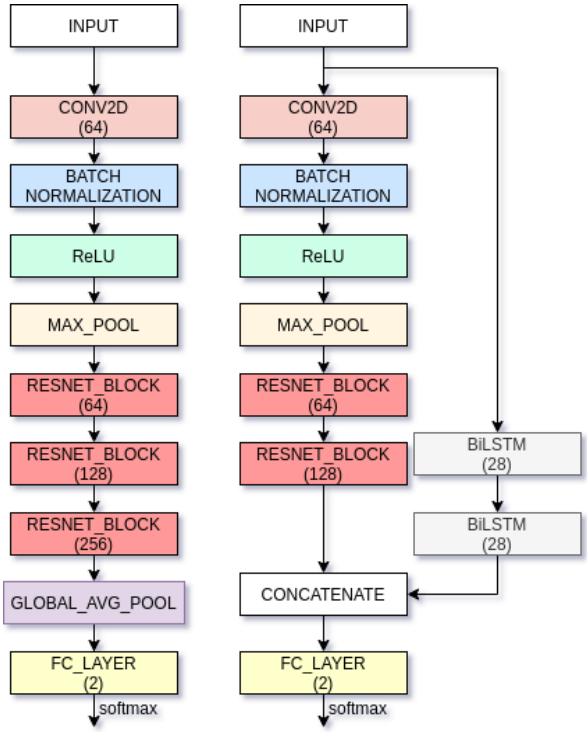

Figure 3.26: (a) ResNet like architecture (b) ResNet like architecture with BiLSTMs.

This combination of convolutional blocks and recurrent neural layers is a very prominent design in architectures for speech processing. The DeepSpeaker [114] and RawNet [115] state of the art architectures for speaker verification include recurrent neural networks after a sequence of convolutional blocks.

The spectrogram for each audio sample was produced by taking its short-time Fourier frequency considering 455 samples per segment, an overlap of 420 points between samples over a Hann window. During training, the spectrogram images are resized to 128 by 128 and converted to grayscale to make them suitable for input by the architectures.

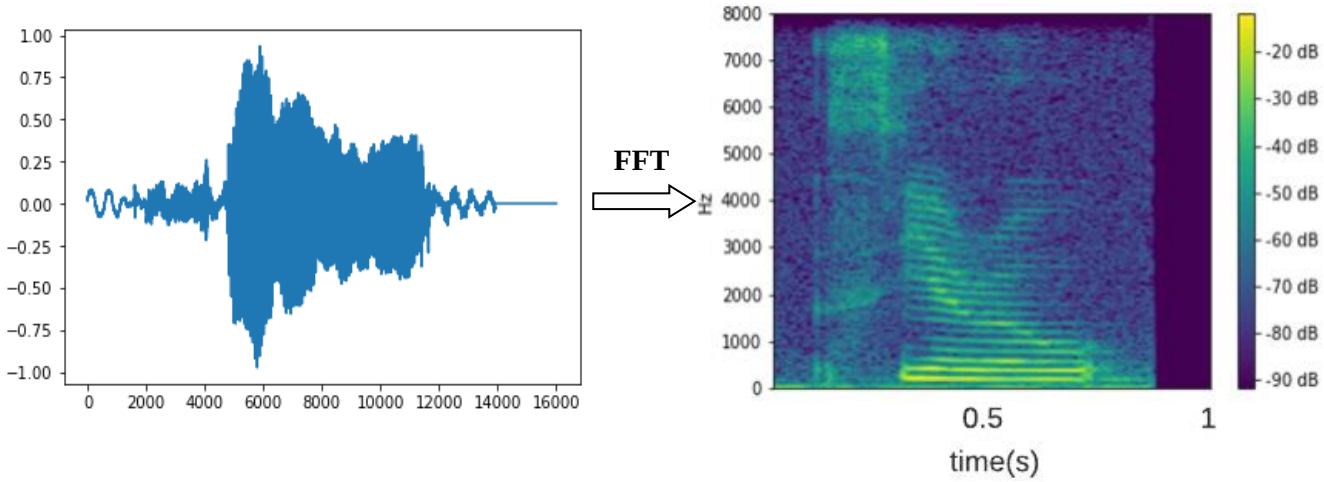

Figure 3.27: Transformation from raw waveform to spectrogram representation via the FFT.

Both models are trained using the same configuration as the architectures for the raw waveform classification. Each model is trained in a four-fold cross-validation setting. The training was performed using the Adadelta [107] optimiser with a learning rate of 0.001, a ρ value of 0.95 and an ε value of 10^{-7}.

3.7.1. Classification performance on spectrograms

Similar to table 3.3, the table below summarises the classification results for both classes of models in terms of mean and standard deviations across all four folds. It was sufficient to train each architecture for 200 epochs; training for more extended periods did not substantially improve performance.

Table 3.6: Classification performance of ResNet-like architectures in terms of mean and standard deviations across all four folds.

Model	No. of Parameters	Mean training accuracy (%)	Mean validation accuracy (%)	Mean testing accuracy (%)
ResNet block	1,233,602	94.64 ± 0.0021	95.08 ± 0.0035	94.57 ± 0.0015
ResNet block with BiLSTM	628,338	98.76 ± 0.0005	97.32 ± 0.0013	96.92 ± 0.0015

At first glance, both models have relatively good performance. The boxplot in figure 3.28 shows that the architecture that employs sequence learning has higher mean validation accuracy. This is reminiscent of the SampleCNN architectures with BiLSTM layers. Performing a standard paired t-test on the distribution of validation accuracies give a *t*-value of -16.95 and a *p*-value of 0.000447. The low *p*-

value signifies strong evidence to accept the alternative hypothesis, indicating that there is a significant difference in performance between the models.

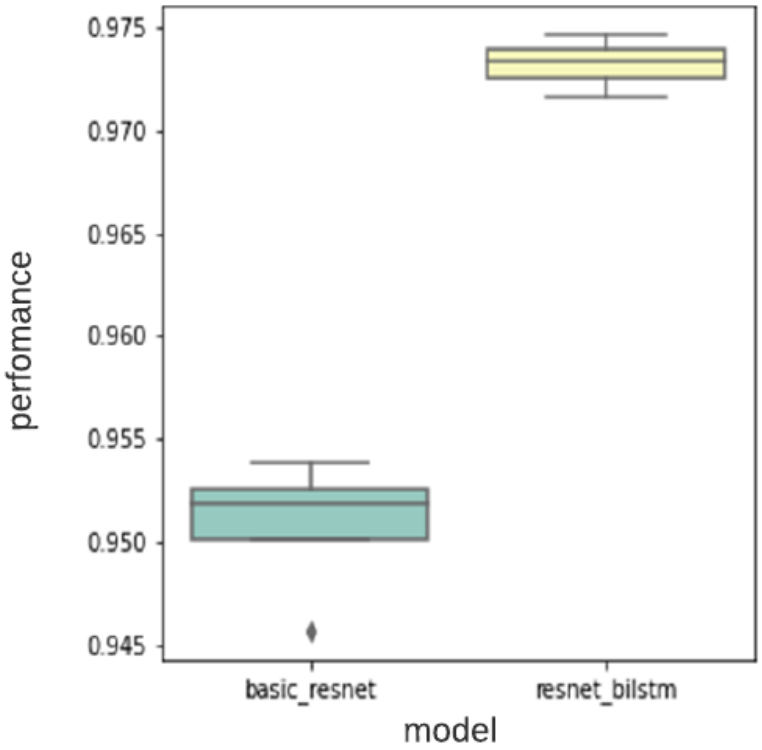

Figure 3.28: Box plots for the validation score for each model across all four folds

Learning sequence information independently from the convolutional blocks allowed the ResNet block with BiLSTM architecture to outperform the other model with significantly fewer model parameters. As described in section 3.2, speech data contains temporal information, implying that sequence learning is an essential aspect of creating speech models.

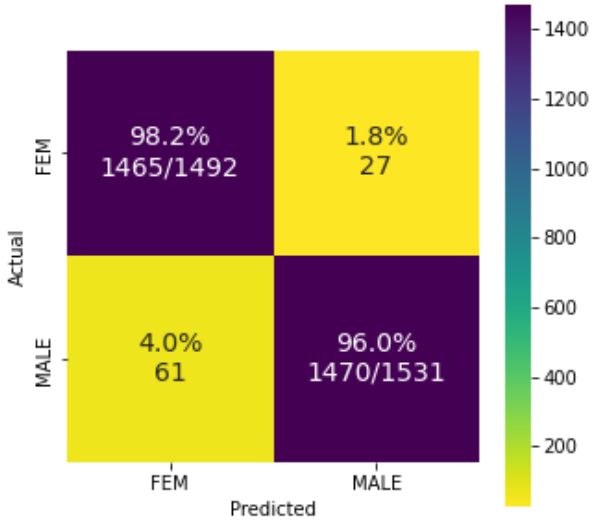

Figure 3.29: Confusion matrix for the ResNet block with BiLSTM model.

The confusion matrix of the squeeze excitation BiLSTM model on the holdout test set. Table 3.7 gives the corresponding classification metrics derived from it. The f1-score for the male and female

classes is the same. However, the model experiences slightly more false positives for the female class and more false negatives for the male class. Overall, the model shows expectable performance.

Table 3.7: Precision, recall, and f1-scores for the ResNet block with BiLSTM model.

Class	Precision	Recall	F1-score
female	0.96	0.98	0.97
male	0.98	0.96	0.97

3.7.2. Spectrogram model interpretation

Similar to the SampleCNN architectures, the predictions for both architectures are explored using expected gradients. Figure 3.30 shows the attribution maps for the ResNet block architecture on a spectrogram instance of male speech. Since the model has a softmax activation function in its output layer, the expected gradients produce two attribution maps showing which parts of the spectrogram pushed the model towards (positive attributions) and away (negative attributions) from each class.

The first spectrogram shows the input; the second shows the explanation for the female class and the last for the male class. According to the explanations, the model is primarily concerned with the progression of lower dark frequency bands and the spaces between them. This is encouraging because dark bands represent the presence of formants. The spacing/positioning of the formants can be used to differentiate between male and female speakers, as discussed earlier in section 3.2.

The activity of the attributions in the lowermost frequency bands is most likely to suggest the model's search for the position of the fundamental frequency (F0**)**, which is, as discussed earlier, a very strong indicator of speaker gender. It is important to note that the formants F1 and F2 are used to determine the formant space configuration. This is used for the planning of articulatory movements for vowel production for a speaker [116].

The annotations a through c on the spectrogram represent the dark bands that most likely represent formants F1 through F4. For this instance, the model has no attributions for the uppermost formant F4, but the other formants receive relatively an equal amount of attention.

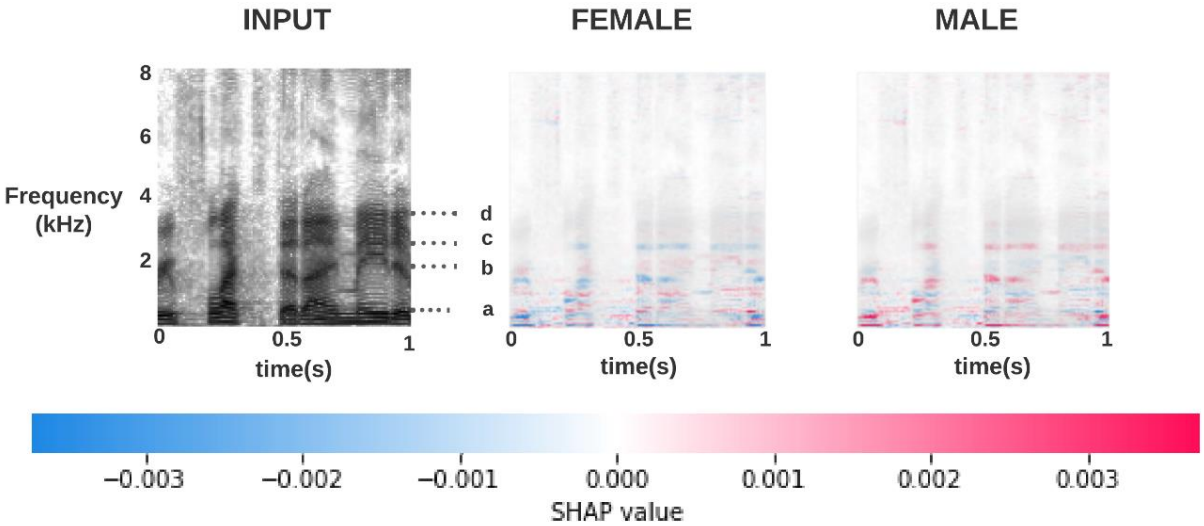

Figure 3.30: Expected gradients explanation for a correct prediction by the ResNet block architecture.

At first glance, the spectrograms are already revealing information familiar to what we know about speech and do so without introducing a new layer of abstraction to make sense of the information presented by the explanation.

Figure 3.31 shows the attribution maps for ResNet block architecture that employs sequence learning via a stack of BiLSTMs. Comparing the model explanations shown in figures 3.31 and 3.32, the model with sequence learning pays more attention to the dark bands compared to the previous model despite these attributions being less defined/concentrated.

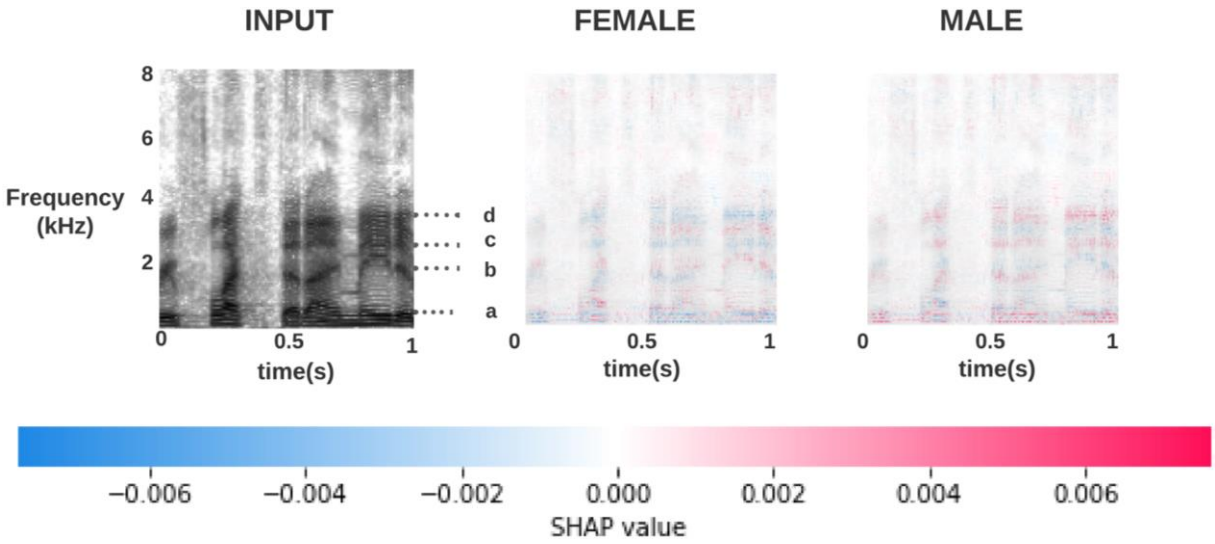

Figure 3.31: Expected gradients explanation for a correct prediction by the ResNet Block with BiLSTM architecture.

The range of Shapley values is higher for the explanations shown in figure 3.30, signifying that separately learning temporal features might help elevate the pressure of making decisions only from the information learned by the residual blocks. Since the input is fed to two separate branches in the network, the higher Shapley values might be because specific parts of the input are regarded as important to both branches. This also protects the model from becoming overly reliant on features that can be discovered by the convolutional layers or, conversely, the recurrent layers [110],

Figure 3.32 show the explanations for speech taken from the AudioMNIST dataset. The spectrograms are female and male speech utterances of the word zero. This example is interesting because the data distribution is different from that of the training and testing data. The AudioMNIST dataset was recorded using a different microphone at a different sampling rate. Furthermore, the speakers are not native English speakers. The model was able to predict the correct speaker class for both instances. Hence, in the first row of spectrograms, the female class has more positive attributions than the male class, and the reverse is true for the second row.

Similar to figures 3.29 and 3.30, the model pays the most attention to the dark bands in the lower quarter of the spectrogram. However, as discussed in section 2.5, the weakness of these explanations is that they are, to an extent, qualitative in nature.

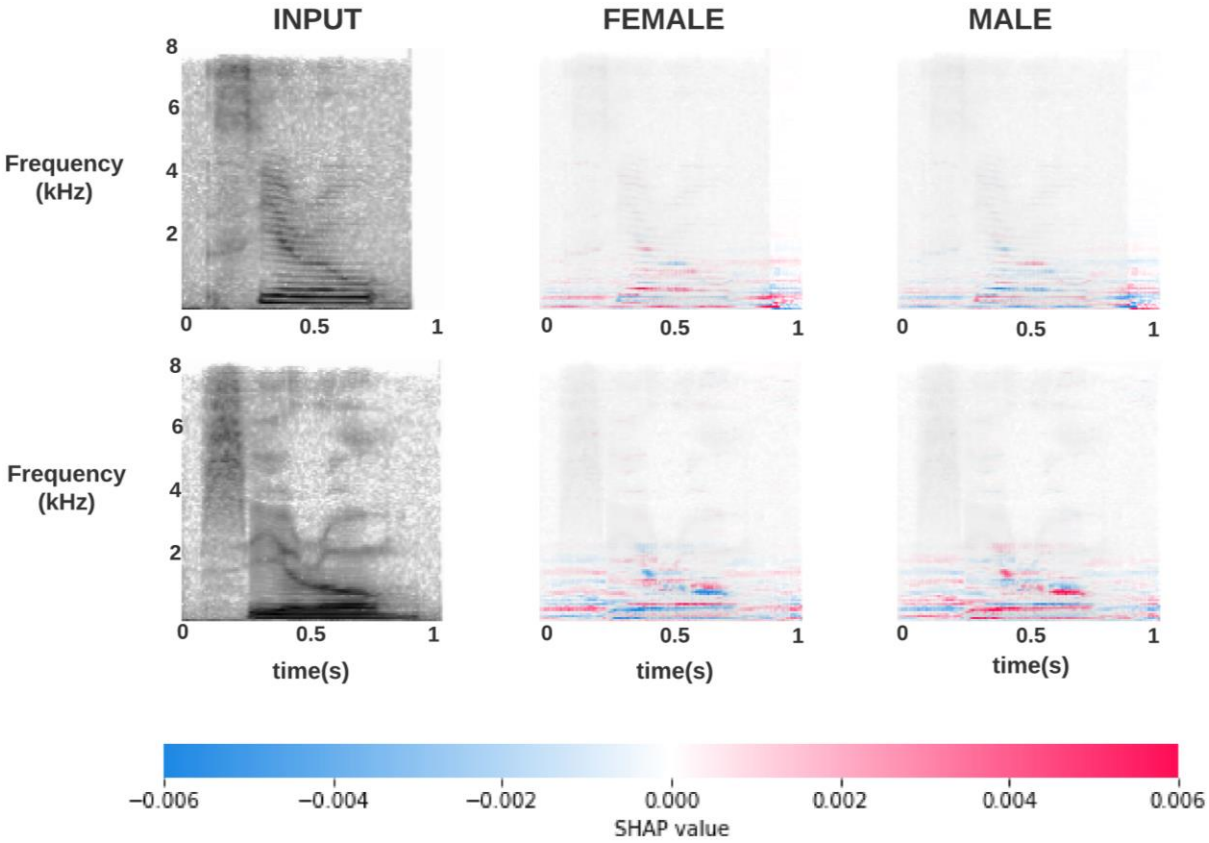

Figure 3.32: Attribution maps for the ResNet block architecture's prediction on the spoken digit zero for female (top) and male (bottom) spectrograms.

In an attempt to validate these assertions, the mean fundamental frequency and first four formants are estimated for the two utterances similar to the feature extraction of the white-box model. The values are given in the table below.

Table 3.8: Estimated voice source parameters for AudioMNIST samples.

Speaker	Mean F0 (Hz)	Mean F1 (Hz)	Mean F2 (Hz)	Mean F3 (Hz)	Mean F4 (Hz)
female	208.05	447.31	1611	2643.96	3673.26
male	106.8	434.42	1453.03	2449.16	3339.16

Since formants are highly dynamic, the distribution by which the formants range should be considered. Figure 3.33 shows the distributions for the fundamental frequency and each of the formant frequencies. In each distribution, the female sample has higher estimated frequency components than its male counterpart. The difference is less prominent for the F1 and F2 but it is much more significant for F0, F3 and F4.

The expected gradients explanations shown in figure 3.32 indicate that the model considers the frequency range which corresponds to **F0**, F1 and F2 for its predictions. The male and female distributions of F0 show a clear difference between the two speakers. This suggests that considering the estimated voice source parameters and their distributions can make the expected gradients explanations less qualitative in some cases.

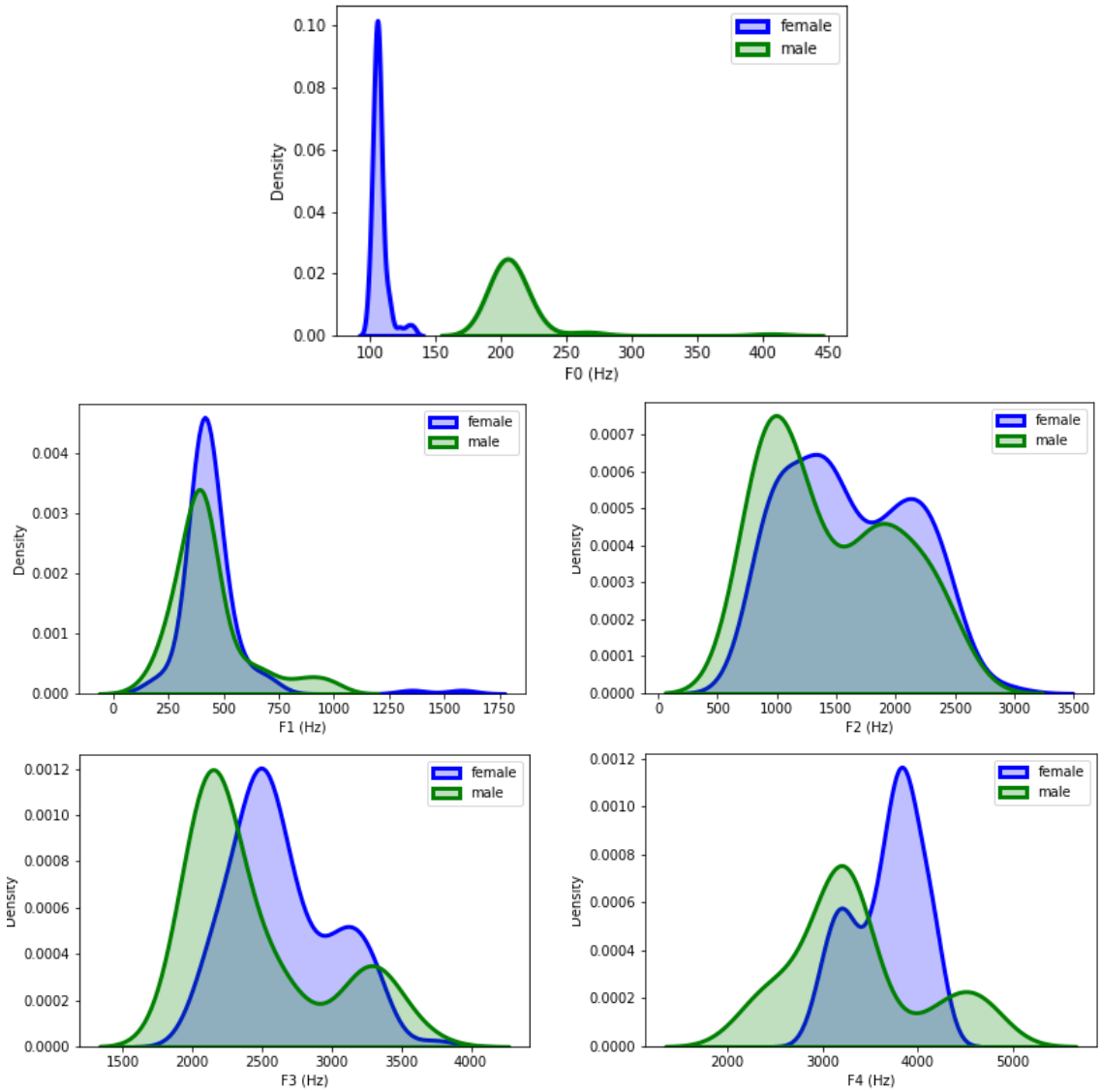

Figure 3.33: Density distributions showing the ranges of the estimated F0, F1 F2 F3 and F4 of the female and male speech samples.

3.8. DISCUSSION

The experiment on gender perception was able to explore several fundamental axioms crucial to solving the research problem:

i. Exploring the gender discriminative properties of speech.
ii. Exploration of the problem through a white-box model.
iii. Evaluating expected gradients attribution explanations for models trained on time-domain (raw waveform) and time-frequency-domain (spectrogram) features.
iv. Explaining the hidden semantics of the raw waveform trained model through spectral analysis.

At first glance, there is no significant difference in performance between the raw waveform and spectrogram models. In this case, the best performing spectrogram model performed slightly better than the best SampleCNN model. The performance of the white-box decision tree model is competitive against the black-box neural networks with a relatively shallow depth. However, the major drawback

to the white-box approach is the model's reliance on the performance of the feature extraction algorithms.

The experiments on the proposed network architectures confirm that the expected gradients are suitable to produce explanations for both raw waveforms and spectrograms. However, the attribution explanations for spectrograms are more human-readable than raw waveforms explanations. It is easier to infer the fundamental frequency and subsequent harmonics from the spectrogram representation of speech. This frequency information is integral to distinguishing between male and female speakers, and it is not immediately apparent when using raw waveforms as input.

On the other hand, modelling through raw waveforms allows one to explore the model's hidden semantics through spectral-level analysis. As discussed in the literature review, hidden semantics explanations have more explanatory power than attribution explanations. The spectral level analysis was able to uncover meaningful explanations. The cumulative frequency response of the filters in the first convolutional layer shows that they place emphasis on the formant frequency regions. This implies that the model has directly learned speaker discriminative information, which is also consistent with the existing literature [65][69][70].

Initially, the spectrograms are easier to explain because the input is already in human-readable form. However, by themselves, these explanations are qualitative; they require to be supported by speaker estimations. The expected gradients explanation on raw waveforms is unsatisfactory primarily because there are no obvious visual cues such as the dark bands that indicate formant position, unlike the spectrograms. It is yet to be investigated if hidden semantic explanations via expected gradients can provide helpful explanations for spectrogram trained models.

In this case, modelling through raw waveforms is preferable because the explanations are a direct interpretation of the information learned by the filters in the network. Furthermore, the SampleCNN interpretations are not only limited to time-domain explanations and provide global interpretability. Once a model is trained on spectrograms, the model interpretations are limited to the information represented by spectrograms.

It is yet to be established how these interpretations can be used to improve the model's performance. As highlighted in section 1.6.2, one of the objectives is to investigate how interpreting a model can give insight into how it can be improved. This means that for the diarisation problem, the interpretability explanations must not only be explicit but also actionable towards improving the model's architecture.

CHAPTER IV

4. VOICE TYPE CLASSIFICATION

This chapter explores the solution to the diarisation problem in a voice type classification context. The difference in speech acoustics between speakers is established based on their physical differences, similar to the previous chapter. The problem is explored via two proposed models, and their performance is compared against the current state-of-the-art benchmark. For each proposed model, the interpretability explanations are used to help improve the model.

4.1. PROBLEM DESCRIPTION

The basic function of a diarisation system is to distinguish between different speakers and use that information to segment an audio stream. Meaning regardless of whether it is an end-to-end or a multi-step system, a diarisation system must solve speaker verification in the case of individual speakers or voice type classification in the case of groups of similar speakers. Similar to the gender perception problem discussed in the previous chapter, the voice type classifier has to learn discriminative speaker information in order to differentiate between speaker vocalisations.

For child language environment analysis, there are generally four classes of interest; adult female speakers represented by the mother (F), adult male speakers represented by the father (M), the key-child of interest (KCHI) and the other children (OCH) that interact with the key-child. This means that on top of the gender-specific differences discussed in section 3.2, the classifier must learn speech differences between adult and child vocalisations. This also comes with a layer of complexity as child vocalisation exhibit properties such as paralinguistics, higher variability between child speakers and spontaneity.

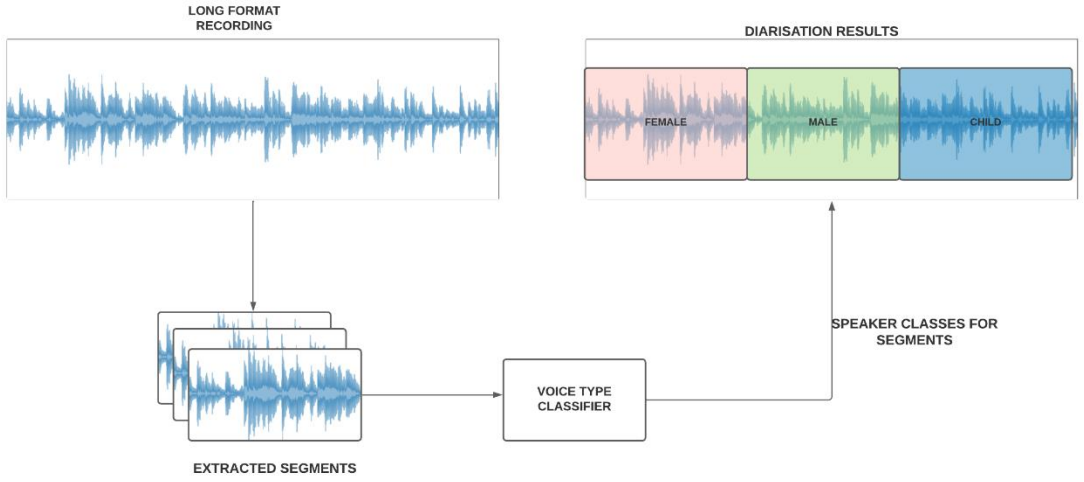

Figure 4.1: Illustration of the voice type classification process.

It is important to note that the voice type classification approach to diarisation does not include the timestamps in the training data as these are expensive to obtain and include during training. Hence, for each audio stream, a voice type classifier will only assign a speaker to fixed-length (1-second) segments. The results are then automatically aggregated after the length of the audio stream has been fully classified. This will help accommodate fast speaker changes. The choice for the fixed-length segment can also be regarded as a hyperparameter. In the study that formulated the baseline [11], the model takes in two-second chunks of audio data. There is no motivation given as to why the design

choice was made, especially since the original SincNet[70] implementation was trained on speech utterances 200ms long. The choice to take in one second worth of audio data is motivated by the study on SampleCNN architectures [105]; this duration for the input is used for the acoustic scene tagging and keyword spotting tasks.

4.2. ACOUSTIC DIFFERENCES BETWEEN ADULT AND CHILD SPEECH

Similar to gender, age also has an effect on the voice source parameters, such as the fundamental frequency and the formants [117][118]. Since the voice source parameters depend heavily on the structure and size of the articulatory organs, there is a direct relationship between the development of the vocal organs and the resulting speech properties. The table below provides the mean resonance frequencies in Hertz of the formants F1, F2 and F3 for men, children and women, as reported in [119].

Table 4.1: Average formant frequency for men, children and women.

Speaker	Formant	heed	head	had	hod	haw'd	who'd
Men	F1	270	530	660	730	570	300
	F2	2290	1840	1720	1090	840	870
	F3	3010	2480	2410	2440	2410	2240
Children	F1	370	690	1010	1030	680	430
	F2	3200	2610	2320	1370	1060	1170
	F3	3730	3570	3320	3170	3180	3260
Women	F1	310	610	860	850	590	370
	F2	2790	2330	2050	1220	920	950
	F3	3310	2990	2850	2810	2710	2670

The articulatory organs of children are smaller and more underdeveloped compared to that of adults. The vocal tract length of infants is estimated to be around 6 to 8 centimetres compared to 15 cm and 18 cm for the average female and male adults, respectively[120][121]. The data represented in table 4.1 shows that children have higher formant frequencies than adult male and female speakers. These high formant frequencies eventually lower as their articulatory organs grow and develop. Similarly, the vocal folds of children are much smaller and thinner than those of adults; hence children have a higher fundamental frequency/pitch. The F0 rapidly declines within the first year of life but remains relatively stable between three to four years [120]. The table below gives the approximate formant frequencies and mean fundamental frequency for the different age-sex groups in childhood [120].

Table 4.2: Average formant frequency for different age groups of children.

Age-sex Group	F1 range (Hz)	F2 range (Hz)	F3 range (Hz)	Mean F0 (Hz)
2-year-old, male and female	400-1000	1500-3500	No data	300
4-year-old, male and female	480-1200	1700-3600	3400-4300	280
6-year-old, male and female	350-1200	1500-3200	3300-3700	260
12-year-old female	375-950	1600-3000	3000-3500	230
12-year-old males	325-900	1300-2700	2700-3300	230

Similar to the gender perception problem, the interpretations of the model must be able to communicate these differences in voice source parameters for each speaker class.

4.3. DATASET

The dataset consists of a collection of daylong recordings which originate from multiple publicly available child-centred sources. Inspired by the BabyTrain [11] corpus mentioned in section 1.3.1, the dataset was compiled taking into consideration several key factors such as; multiple recording devices, multilingual environments, under-sourced languages, non-English languages and English from multiple sources. These factors are derived from the main criticism of the LENA™ system established in section 1.1.2. The table below summarises the sources included in the dataset. A more detailed description of the number of samples included from each source is given in Appendix A.

Table 4.3: Description of each source included in the dataset.

Corpus	Description	Language
FallsChurch	Naturalistic longitudinal recordings of a male child exposed to English and Japanese from one to two years [122]	Japanese and English
Tsay	Spontaneous conversations between children ages one to five years and their caregivers from Minnan-speaking families in Southern Taiwan [123].	Chinese
Utrecht	Unstructured recordings of two Dutch males between the ages of two to three. Interactions are primarily with their mothers during daily routines and some structured picture book interactions [124].	Dutch
VanDam Public 5-minute	LENA recorded naturalistic child interactions from 53 families. Some of the recordings include children with mild-to-moderate hearing loss. The average child age was 28.9 months [124].	English
Garmann	Longitudinal recordings of three children aged 14 to 23 months exposed to the Norwegian language from Oslo, Norway [125].	Norwegian
CCLAS	Semi-structured recordings of the interactions between a female child and her native Cree-speaking mother. The sessions were captured over two years as the child developed from 24 to 36 months [126].	Cree
MCF	Naturalistic longitudinal recordings of three children exposed to the Swedish and Portuguese languages. The recordings were collected from birth to primary school years [127].	Swedish and Portuguese
Demuth	Naturalistic long-format recordings documenting four children from the Mokhotlong district in Lesotho interacting with their primary caregivers, siblings, friends and extended family. The children were between two to four years [128].	Sesotho
Thomas	Naturalistic longitudinal recordings of a male child exposed to English from two to four years [129].	English-UK

To account for occurrences of non-speech within an audio stream, samples of non-communicative vocalisations and random noises were added to the training data. The aim is to help eliminate the need for a voice activity detector. Traditional diarisation systems require a separate model to detect if each segment contains speech or not before it passes to the next step in the diarisation pipeline. Representing these non-speech occurrences in a separate class means that in addition to classifying between the four-speaker types, the voice type classifier can also determine if a speaker is present or not. The non-speech samples are derived from various sources [130][131]. The non-speech samples include instances of clapping, breathing, coughing, crying, sneezing, footsteps, laughing, snoring, white noise, brown noise, pink noise, and silence.

Overall, the dataset has five classes, the key-child (KCH), other-child (OCH), female (F), male (M) and non-speech (NON). Each class has 12000 instances[22] , bringing the dataset's total instances to 60 000. More than five different speakers represent each speaker class. For the male and female classes specifically, the samples that are included do not only belong to the father and mother. These classes include situations where the child's grandparents or other grownups are present. The aim is to help increase the range of diversity of speakers for each class.

The samples in the dataset were created by making use of the timestamps in the accompanying transcriptions. For each long-format recording, the timestamps are used to extract utterances for the corresponding speaker. The extracted speaker instances shorter than one second are discarded, and instances longer than one second are partitioned into one-second segments with 250-millisecond (25%) overlap.

Since the audio is annotated using human transcriptionists, there is room for introducing errors in the training samples. It is also important to note that the audio is annotated in such a way that does not acknowledge cross-talk, i.e., multiple speakers speaking at the same time.

No speech enhancement was applied to the data prior to training. Since the instances are captured at different sampling rates, all the speech segments were resampled to 8kHz, and stereophonic samples were also converted to monophonic.

4.4. BASELINE SETUP

The voice type classification model introduced by [11] serves as the performance baseline for the proposed model. The baseline classifier is an open-source DNN network that is trained to classify vocalisations produced in a child's natural language environment. The network combines the SincNet architecture [70] with a stack of bidirectional recurrent layers (BiLSTMs) and manages to outperform the state-of-the-art LENA™ system introduced in section 1.1.2. The baseline model treats the voice type classification problem as a multi-label classification problem. This means that for each input audio stream that is divided into L frames $S = \{s_1, s_2, s_3, \dots, s_N\}$ the expected sequence of output labels is $y = \{y_1, y_2, y_3, \dots, y_N\}$ and y_i has the dimension K equal to the number of labels. If the j^{th} class is activated then $y_{i,j} = 1$ and $y_{i,j} = 0$ otherwise. A major difference between the LENA voice type classifier and the baseline model is that the latter is a multi-label classifier allowing multiple classes to be activated at the same time. The LENA™ classifier is a multi-class classifier allowing only one class to be activated at a time, it also has an overlap class that represents an overlap between noise and speech.

[22] An instance represents a one second speech sample with its associated speaker label.

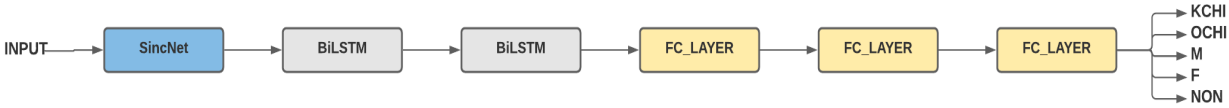

Figure 4.2: The architecture for the baseline state of the art model.

The baseline model's architecture is depicted in figure 4.2. The sigmoid activation function is applied to the last layer, and the network is trained to minimize the binary cross-entropy loss:

$$BCE = -\frac{1}{KM}\sum_{i=1}^{M}\sum_{j=1}^{K} y_{i,j}\cdot \log\hat{y}_{i,j} + \left(1 - y_{i,j}\right)\cdot \log\left(1-\hat{y}_{i,j}\right). \quad (4.4.1)$$

Where $\hat{y}_{i,j}$ is the predicted score (a number between 0 and 1) of dimension K (K = number of classes), $y_{i,j}$ is the target label, and M is the mini-batch size of fixed-length samples drawn from the training set at training time. The baseline system was trained in two phases. First, utilising the entire dataset and then again but withholding the VanDam corpus as the holdout validation set. The purpose of the second phase is to determine the performance of the baseline model on hold-out LENA recorded VanDam Public 5-minute. The training was performed using the Adadelta [107] optimiser with a learning rate of 0.01, $a\ \rho$ value of 0.95 and an ε value of 10^{-7}.

4.5. PROPOSED MODELS

Similar to the gender perception problem, two classes of architectures are investigated to solve the voice type classification problem. First, the SE block with BiLSTM SampleCNN network introduced in section 3.6. This architecture proved competitive for the gender classification problem. The second architecture of interest is the VGG16 architecture. VGG16 is an established CNN architecture known primarily for optimising the depth of a network by using small 3-by-3 convolutional layers with a stride of 1 and a 2-by-2 max pooling layer with a stride of 2 [132]. The architecture was initially introduced for object localisation in images, and the 16 denotes that the model has 16 convolutional layers. For this application, the dense layers or the original VGG16 implementation are replaced by a dropout layer followed by a sigmoid activation layer.

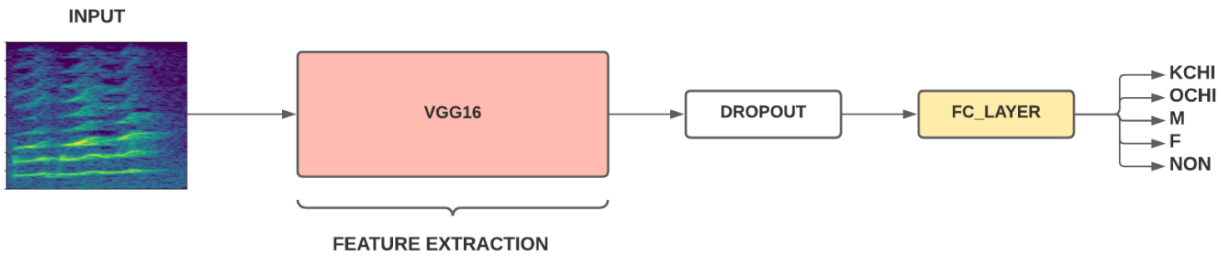

Figure 4.3: The proposed architecture for the voice type classifier. The VGG16 backbone acts as a feature extractor.

Similar to the baseline model, both of the proposed models are trained in a multilabel setting to minimise the binary cross-entropy loss. The models are also trained in two phases, with all the data and then again but withholding the VanDam corpus as the holdout validation set.

Furthermore, these experiments are repeated whilst also swapping out the activation function for the final layer to a softmax activation. In this configuration, the proposed architectures now treat the voice type classification problem as a multiclass classification problem similar to the LENA™ system. This means the networks are trained to minimize the categorical cross-entropy loss:

$$CCE = -\frac{1}{K}\sum_{i=1}^{K}\sum_{j=1}^{J} y_j \cdot \log\left(\hat{y}_j\right) + \left(1 - y_j\right) \cdot \log\left(1 - \hat{y}_j\right). \quad (4.5.1)$$

4.6. EXPERIMENTAL RESULTS

The table below gives a summary of the training configurations for each model. After rigorous hyperparameter tuning experiments, these configurations provided the best performance for each model without resulting in overfitting on the training set.

Table 4.4: Training configurations for each model.

Model	Parameters	Input	Epochs	Learning Rate	Optimizer	Loss
Baseline	1, 075,525	raw waveform	400	0.01	Adadelta	BCE
SE block with BiLSTM	1,749,365	raw waveform	400	0.01	Adadelta	BCE
SE block with BiLSTM	1,749,365	raw waveform	400	0.01	Adadelta	CCE
VGG16	14,717,253	spectrogram	50	0.01	Adadelta	BCE
VGG16	14,717,253	spectrogram	50	0.01	Adadelta	CCE

Table 4.5 gives the performance for each model in both multilabel and multiclass approaches, respectively[23]. The accuracy reported in this table is the test accuracy on the hold-out dataset. As discussed in section 1.8, the f1-score is determined by:

$$f1 = \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}. \quad (4.6.1)$$

Where TP represents the true-positive predictions, FP represents the false-positive predictions, and FN represents the false-negative predictions. This score is a combination of the precision and recall metrics. It can also be defined in terms of precision and recall:

$$f1 = \frac{2*(\text{PRECISION}*\text{RECALL})}{\text{PRECISION} + \text{RECALL}}. \quad (4.6.2)$$

A desirable f1-score represents the model experiencing low FPs and FNs. The best possible score is one, and the worst possible score is zero.

As reported in table 4.5, there is no significant difference in performance between the baseline and the SampleCNN models. The VGG16 models perform significantly better than the baseline and the SampleCNN models with higher f1-scores across all speaker classes. The SampleCNN model

[23] Even though the multilabel models have the ability to activate multiple classes at the same time. The f1-score for these models is determined the same way as that the multiclass models, taking the class with the highest probability score as the prediction.

performs slightly better in the multiclass classification configuration than the multilabel classification configuration, with a higher overall performance on the test holdout dataset and higher f1-score on the female and other child classes. On the other hand, the VGG16 architecture performs slightly better in the multilabel configuration giving higher f1-scores on the female, male and child classes. The high f1 scores for the non-speech class signify that each model has learned to detect speaker activity.

Table 4.5: Classification performance of voice type classification models.

Model	Labels	Accuracy	F1-score					Mean (F1-score)
			KCHI	F	M	OCH	NON	
Baseline	Multilabel	0.68	0.56	0.53	0.71	0.65	0.92	0.674
SE block with BiLSTM	Multilabel	0.70	0.60	0.55	0.74	0.67	0.93	0.698
SE block with BiLSTM	Multiclass	0.71	0.60	0.56	0.74	0.71	0.93	0.708
VGG16	Multilabel	0.79	0.70	0.68	0.83	0.81	0.97	0.798
VGG16	Multiclass	0.79	0.71	0.65	0.82	0.80	0.97	0.790

To get a more informed perspective on how the models perform against each other, the training experiments for each model are repeated four more times to produce the boxplots shown in the figure below. The performance reported in the y-axis is the performance on the holdout test set.

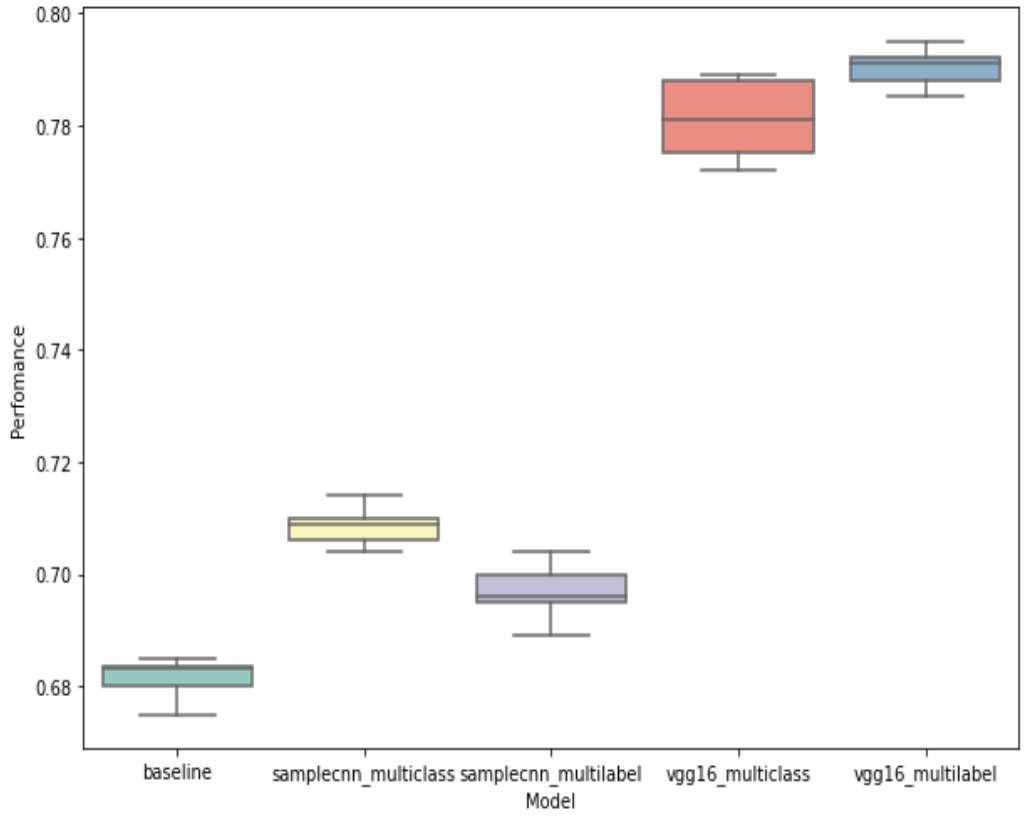

Figure 4.4: Box plots of the performance for each model on the repeated experiments.

The boxplots confirm the initial results given in table 4.5. The proposed models outperform the current state-of-the-art baseline system. This is indicated by the higher mean performance across all the two classes of proposed models and their variations. The VGG16 models have relatively comparable performance; performing a paired t-test on the distributions of the performances of these models gives a *p*-value of 0.11 and a *t*-value of 2.06. As established in section 3.6.1, this indicates that there is no significant difference in performance between these two variations. However, the variance in performances for the VGG16 multilabel configuration is smaller.

The major aspect that stands out from these results is how the performance on the key-child and female classes is the lowest for all the models. Looking at the confusion matrix for the VGG16 multilabel model shown in figure 4.5, most of the confusion is between the female and the key child classes. A significant number of key-child voice samples are misclassified as female, and the same is true for female samples. This trend was found consistent for each model and led to reduced performance for all classifiers.

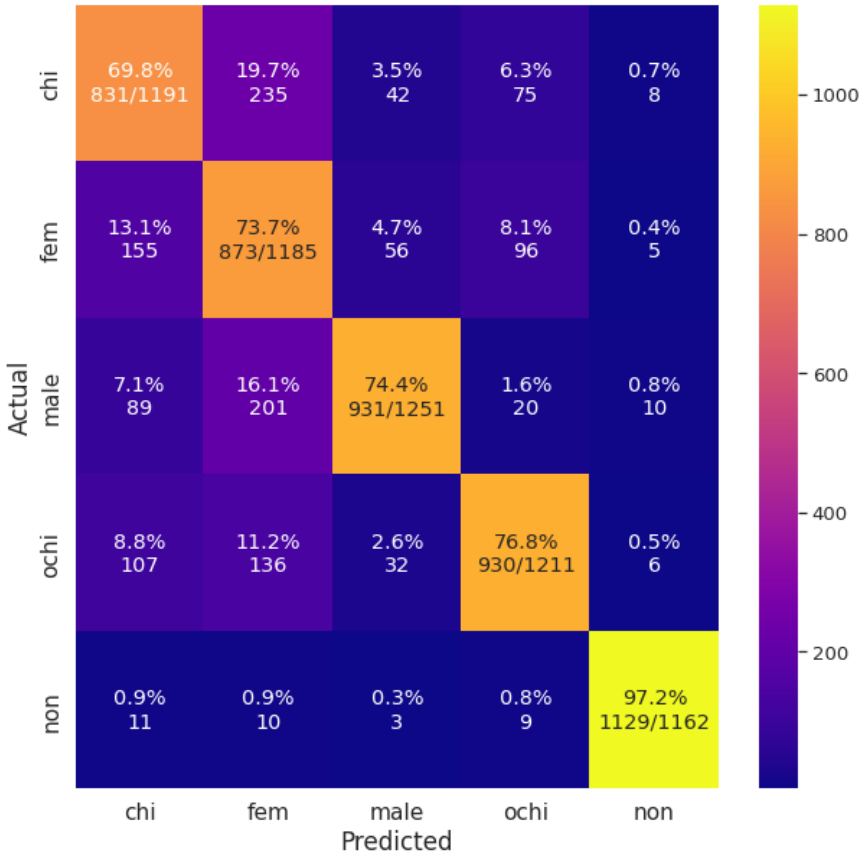

Figure 4.5: Confusion matrix for the VGG16 multilabel model.

4.7. PERFORMANCE ON LENA RECORDED DATA.

In a similar fashion to table 4.5, the table below reports the performance of all the models on the VanDam validation holdout dataset. As mentioned previously, this corpus is of particular interest because it was collected using the LENA talk pedometer and contains American English. Whilst in the training dataset, there are two variants of English, the UK variant in the Thomas corpus and the variant in the bilingual FallsChurch corpus.

Table 4.6: Classification performance of voice type classification models on LENA recorded data.

Model	Labels	Accuracy	F1-score					Mean (F1-score)
			KCHI	F	M	OCH	NON	
Baseline	Multilabel	0.44	0.36	0.27	0.39	0.17	0.86	0.41
SE block with BiLSTM	Multilabel	0.48	0.33	0.29	0.48	0.31	0.83	0.448
SE block with BiLSTM	Multiclass	0.48	0.38	0.27	0.49	0.23	0.86	0.446
VGG16	Multilabel	0.51	0.36	0.31	0.56	0.24	0.94	0.482
VGG16	Multiclass	0.52	0.36	0.32	0.55	0.22	0.93	0.476

According to table 4.46, the baseline and the SampleCNN models performed relatively the same, with the VGG16 models performing slightly better. These models also have relatively low performances on the female, key-child and other child classes. The low performances, in general, show that all the models do not fare well on the VanDam dataset. Granted, the proposed models perform slightly better than the baseline. The image below shows the confusion matrix for the VGG16 multiclass model.

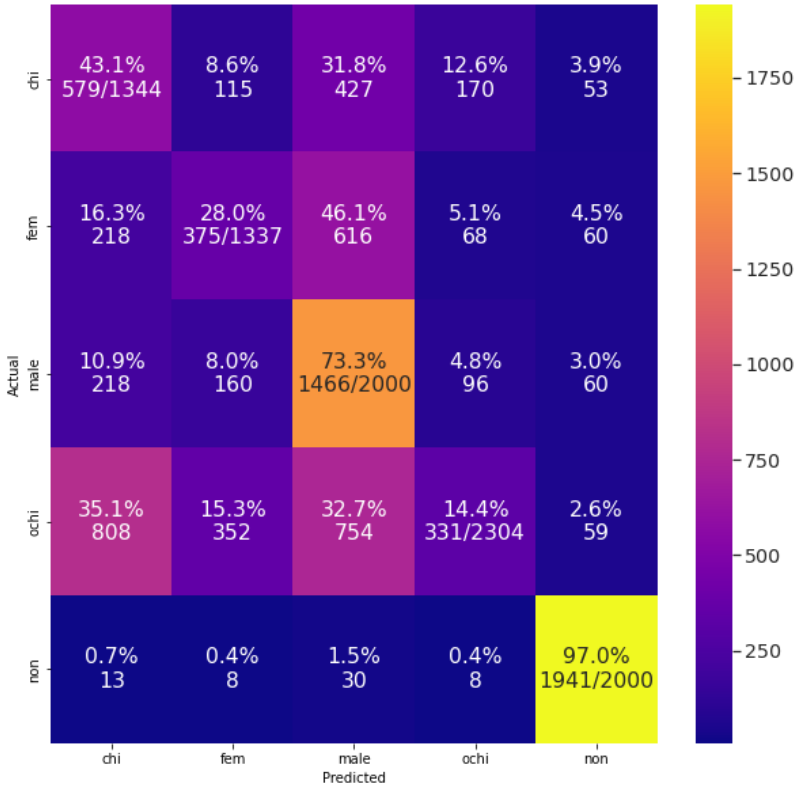

Figure 4.6: Confusion matrix for the VGG16 multiclass model.

4.8. MODEL INTERPRETATIONS

Similar to the gender classification problem, explanations are generated for the proposed models to try and uncover what influenced their decisions. The interpretations of the SampleCNN model are explored only via spectral analysis. This is because, based on the findings of the gender classification experiments, the attribution explanations via expected gradients do not yield any meaningful explanations. Figure 4.7 shows the cumulative frequency response of the filters in the first

convolutional layer of the SampleCNN (SE block with BiLSTM) multilabel model. The response is generated in the same manner as for the gender classifier using equation 3.6.1. The response reveals that the first convolutional layer's filters concentrate on high-frequency information. The strongest cues for speaker discrimination are located within the lower frequency ranges, and the filters do not place much emphasis on this frequency information.

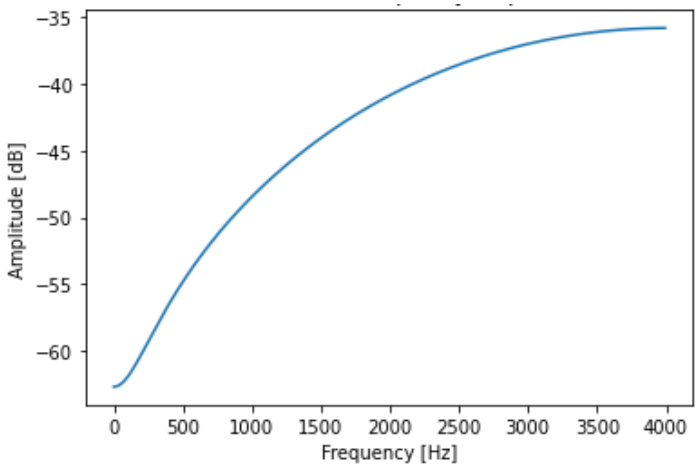

Figure 4.7: Cumulative frequency response for the multilabel SampleCNN model.

Moving on to the VGG16 spectrogram model, the image below shows the attributions for the input layer, the explanation is for the model's correct prediction of the child class. Each column of images represents an explanation for the positive (red) and negative (blue) attributions that pushed the model towards that class. The y-axis for each spectrogram represents the frequency in kilohertz (kHz), and the x-axis is the time given in seconds.

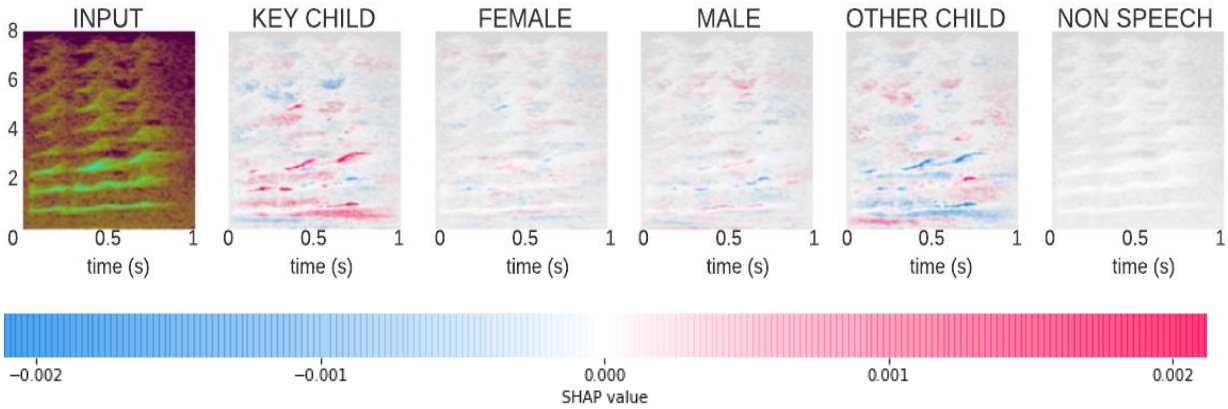

Figure 4.8: Expected gradients explanations for the key-child target prediction.

Interestingly, the most activity for the SHAP values is present for the explanations of the key-child (KCH) and the other-child (OCH) classes. This is most likely because these speaker types are closest in age and have similar speech qualities, making them the most likely candidates. On the other hand, the other-child class has more negative attributions in the spaces between the formants.

To expand on these explanations, the expected gradients method is extended to visualise the response of the hidden layers of the VGG16 architecture to the same child instance as in figure 4.8. Starting with the first convolutional layer and moving on to the third, fifth, eighth and then finally the eleventh

convolutional layer, as shown in figure 4.9. These explanations are produced by mapping the expected gradients produced for the input to the desired hidden layer.

By the first convolutional layer, the model has already recognised to negatively attribute the formant bands and give positive attributions on the lowest and highest parts of the spectrogram for the non-speech class. The latter is the most likely the position for low and high-frequency noise information, and the formants represent the presence of speech. This gives confidence that the model can detect voice activity.

As the model goes deeper, it seems to pay attention to more areas of the spectrogram, giving more attributions to the relative space. This may be because the filter sizes decrease and the receptive field increases as one goes deeper into the architecture.

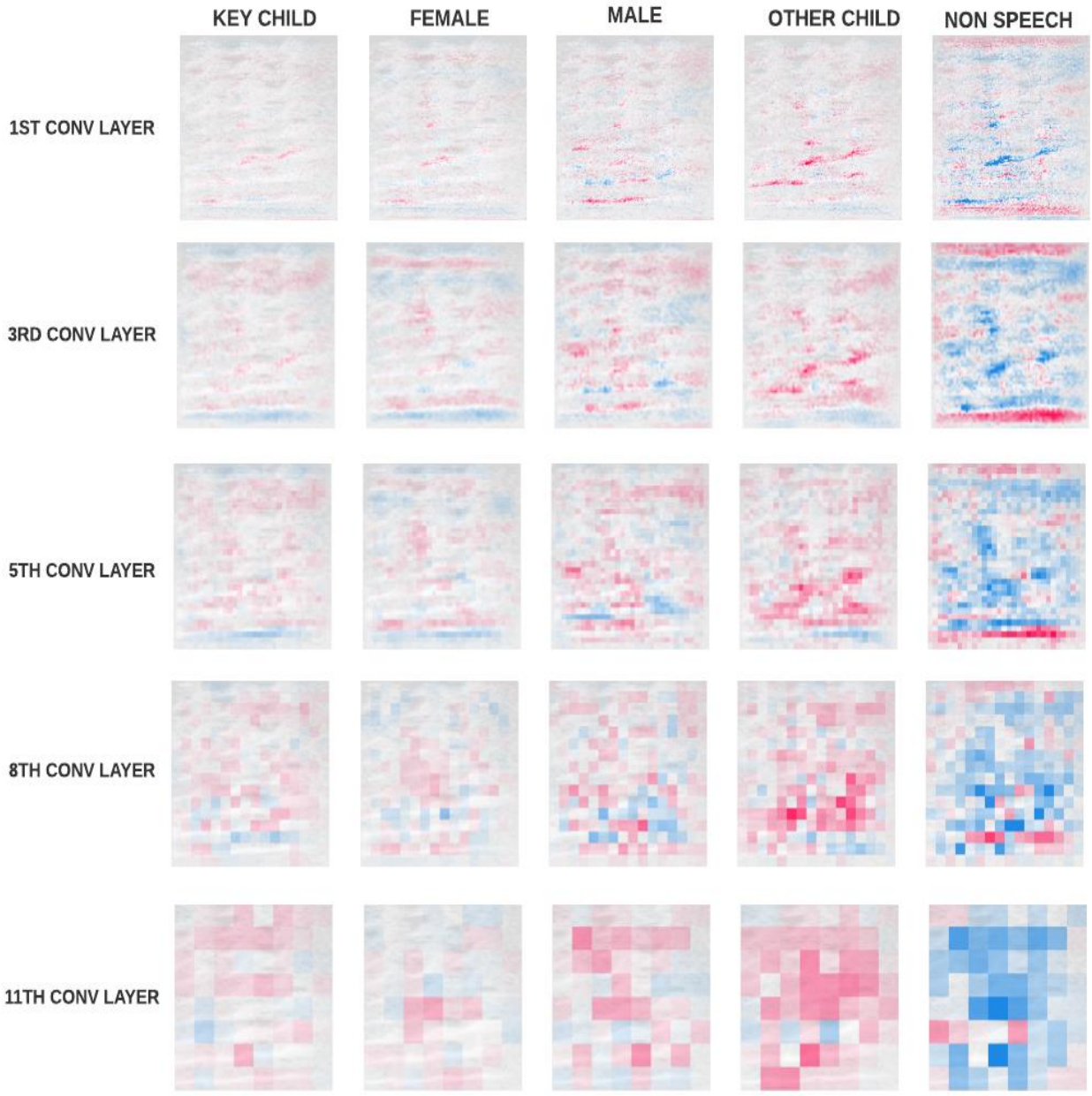

Figure 4.9: Hidden semantic explanations for the first, third, fifth, eighth and eleventh convolutional layers in the VGG16 model.

What is surprising about the hidden semantic explanations is that as the model goes deeper into the network, the attributions seem to hint at the other-child class being the model's prediction. Even though in the last convolutional layer the other-child class has more positive attributions than the target-child class, the ultimate model decision was the correct prediction of the child class. This seems to show how these hidden semantic explanations by themselves can be misleading if not put in the appropriate context.

To avoid making this discussion purely qualitative, the explanations are also explored taking into consideration the estimated voice source properties for the input utterance. The table below reports these estimations of this input utterance extracted using the algorithms discussed in section 3.5.1.

Table 4.7: Estimated voice source parameters for the key-child speaker.

Speaker	Mean F0 (Hz)	Mean F1 (Hz)	Mean F2 (Hz)	Mean F3 (Hz)	Mean F4 (Hz)
key-child	208.05	697.34	1195.17	1949.04	2646.15

Similarly, the figures below show the frequency ranges for the estimated F0 and formants respectively. The aim is to relate the information from the extracted voice source parameters with the attribution explanations on the corresponding spectrogram.

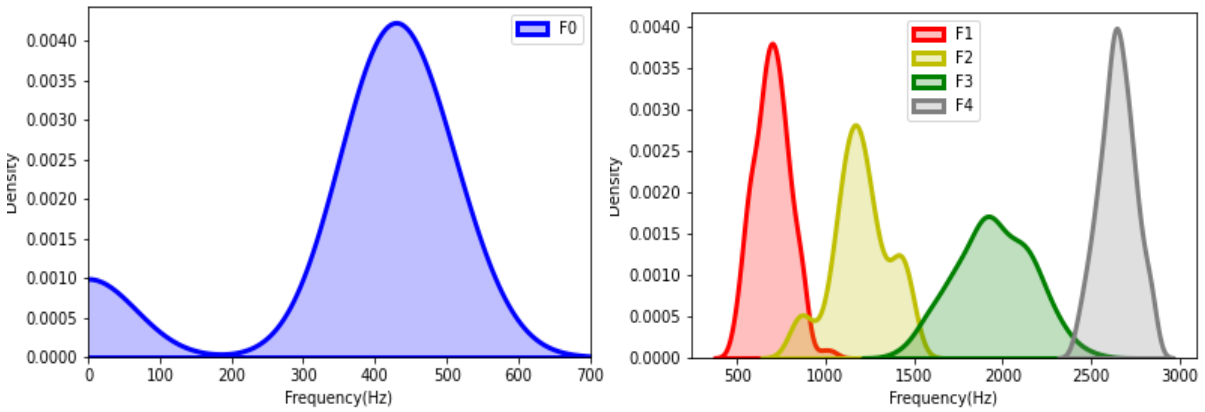

Figure 4.10: Density distributions showing the ranges of the estimated (a) F0 and (b) the formants.

Figure 4.11 shows the spectrogram of the input utterance with its axes and the corresponding attribution explanation for the prediction of the key-child class extracted from the first column of figure 4.8. The dashed lines represent the position of the estimated voice parameters given in table 4.7.

Correlating the estimates in table 4.7, their frequency ranges in figure 4.10 and the spectrogram in figure 4.11, one can deduce three observations:

i. The model does not take into account the position of the fundamental frequency for the child prediction. Relatively looking at figure 4.8, the positive attributions for this frequency range are given to the non-speech class. Typically this frequency range corresponds to low-frequency noise samples in the dataset.

ii. The distribution ranges of the estimated formants correspond to the most positively attributed dark bands on the saliency map explanation.

iii. For the speaker classes, the model does not place much emphasis on the higher frequency bands which do not hold any immediate speaker discriminative information.

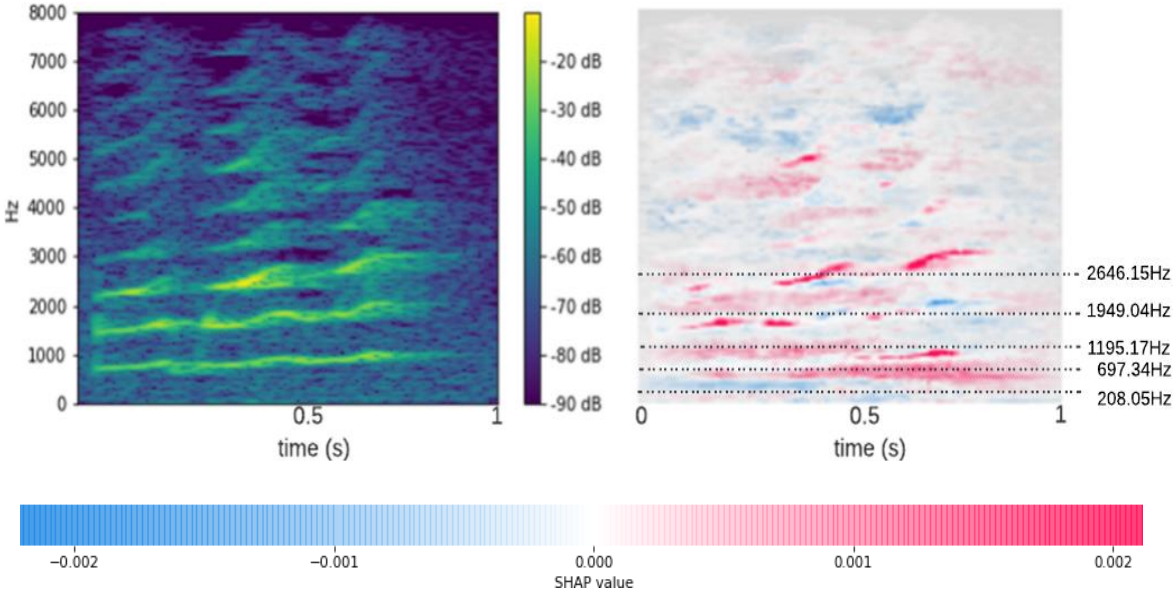

Figure 4.11: (a) Input spectrogram with axes. (b) Input spectrogram with expected gradients attributions. The annotations are the average fundamental frequency and the average formant frequencies respectively.

4.9. IMPROVING THE PERFORMANCE OF THE SAMPLECNN ARCHITECTURE

Taking into consideration the model interpretations, this section will detail efforts to improve the model performances. Starting with the SampleCNN architecture, the frequency response in figure 4.7 shows that the small filter lengths do not sufficiently favour the desired frequency range, which contains the established speaker discriminative information. In previous studies, longer filter lengths have been proven to favour lower-frequency speaker information relating to the fundamental frequency [65][69][70].

It would be interesting to explore how changing the length of the filters in the convolutional layers influences the performance and cumulative frequency response of the SampleCNN model. The multiclass SampleCNN architecture is retrained whilst varying the filter lengths first by a magnitude of 10 and then by a magnitude of 100. These iterations of the SampleCNN architectures are trained using the same configuration as the original SE BiLSTM multiclass architecture described in table 4.4.

The table 4.8 gives the classification performance of the SampleCNN models of varying kernel lengths. It reports that increasing the kernel length increases the model's performance on the holdout test set. The performance of the longer kernel models is the same for the non-speech class and the other child class. However, the model with the 30-filter length slightly outperforms the 300-filter length model on the key-child class. Whilst the latter model has a slight edge on the female and male speaker classes.

Table 4.8: Classification performance of the SampleCNN models with varying kernel lengths.

Model	Filter size	Accuracy	F1-score					Mean (F1-score)
			KCHI	F	M	OCH	NON	
Original SE block with BiLSTM	3	0.7100	0.60	0.56	0.74	0.71	0.93	0.708
30-SE block with BiLSTM	30	0.7373	0.65	0.58	0.76	0.74	0.94	0.734
300-SE block with BiLSTM	300	0.7385	0.64	0.61	0.77	0.74	0.94	0.738

Figure 4.12 shows the cumulative frequency response of the first convolutional layer for the 30-kernel length and 300-kernel length SampleCNN models, respectively. A closer look at the frequency response for the 30-kernel length architecture shows that the most prominent peaks correspond to the ranges of the formant frequencies given in figure 4.10.

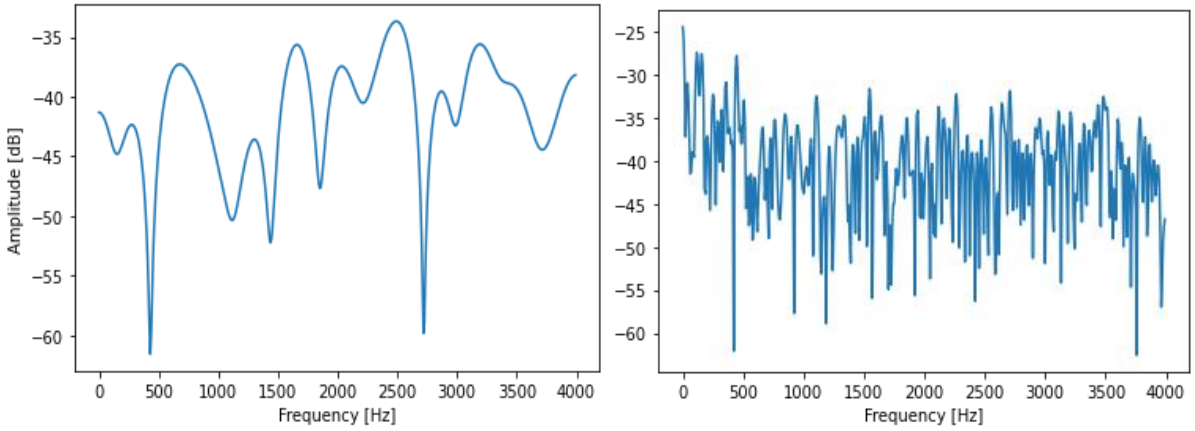

Figure 4.12: Cumulative frequency responses of 30 and 300 filter lengths models, respectively.

The figure below shows the mean of the estimated voice parameters for the key-child speaker given in table 4.7, highlighted with the dashed red lines on the cumulative frequency response for the 30-kernel length architecture. Furthermore, the spectrogram shown in figure 4.11 shows a more pronounced time-dependent variation for the higher formant frequencies than the lower ones. This explains the slight differences between the frequency response peaks shown in figure 4.13 and the time-average extracted formant frequency values.

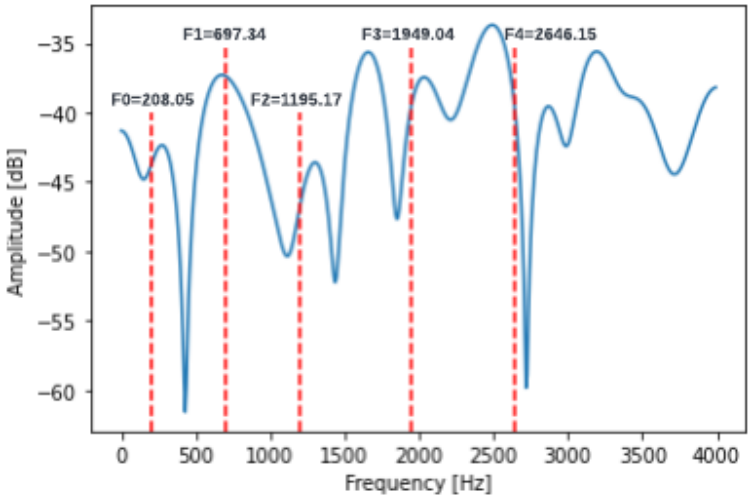

Figure 4.13: Cumulative frequency response of the 30-kernel length architecture.

The cumulative frequency response for the 300-kernel length architecture shows that the filters in the first convolution layer emphasise learning the lower frequency information with the highest peaks corresponding to the range of the estimated fundamental frequency F0 shown in figure 4.10. This is evident in the downward trend of the frequency response peaks, as shown in figure 4.13. The mean of the formant frequencies features from table 4.7 are also highlighted in this graph.

Compared to the frequency response for the original model shown in figure 4.7, both the longer kernel models pay more attention to the lower frequency spectral information. This yields a corresponding increase in performance across all the speaker classes but most significantly on the key-child and female speaker classes.

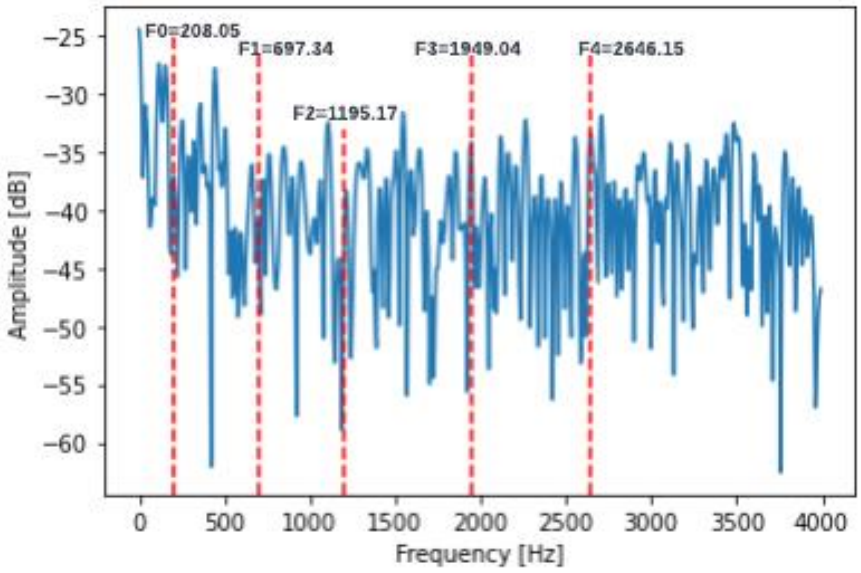

Figure 4.14: Cumulative frequency responses of the 300-kernel length architecture.

Furthermore, instead of simply replacing the small filter lengths with larger ones, a multiview approach can be adopted where input raw waveform is fed to multiple feature extractors, and the learned features are combined in the later layers [110]. This is represented in figure 4.15. The first convolutional layer for each feature extractor branch has a different kernel length. In this case, one of these branches retains the small kernel length of 3 samples whilst the other branch has a much larger

filter length of 300 samples. The resulting architecture was trained for using the same configuration laid out in table 4.4 for the SE BiLSTM multiclass architecture.

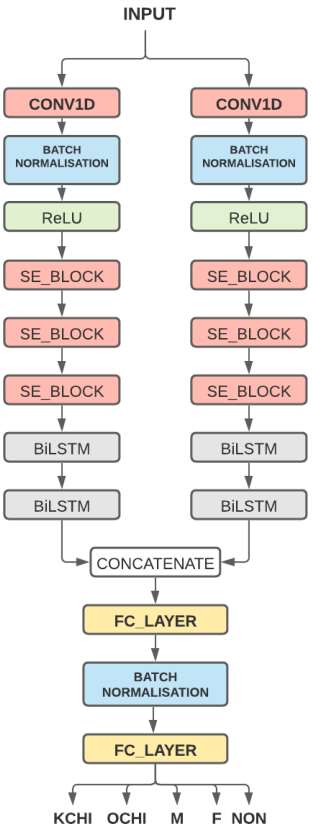

Figure 4.15: Extended SE BiLSTM architecture.

The table below gives the result of the extended architecture and the original SE BiLSTM model. The multiview architecture only yields a slight improvement on the other child (OCH), female (F) and male (M) classes. It is important to note that the multiview model is outperformed by the previously mentioned iterations with longer kernel lengths, whose performances are reported in table 4.8.

Table 4.9: Classification performance of the SampleCNN models with varying kernel lengths

Model	Labels	Accuracy	F1-score					Mean (F1-score)
			KCHI	F	M	OCH	NON	
Original SE block with BiLSTM	Multiclass	0.71	0.60	0.56	0.74	0.71	0.93	0.708
Extended SE block with BiLSTM	Multiclass	0.72	0.60	0.57	0.75	0.72	0.93	0.714

Figure 4.16 shows the cumulative frequency response for the first convolutional layers of each feature extraction branch in the updated SampleCNN architecture. The response for the small filter lengths is similar to that of the original model. This is no surprise since these have the same filter length of three. The response of the larger filter length is similar to that of the 300-kernel length SampleCNN

model kernel shown in figure 4.14 providing better coverage of the relevant frequency information ranges which are most likely to contain speaker discriminative information.

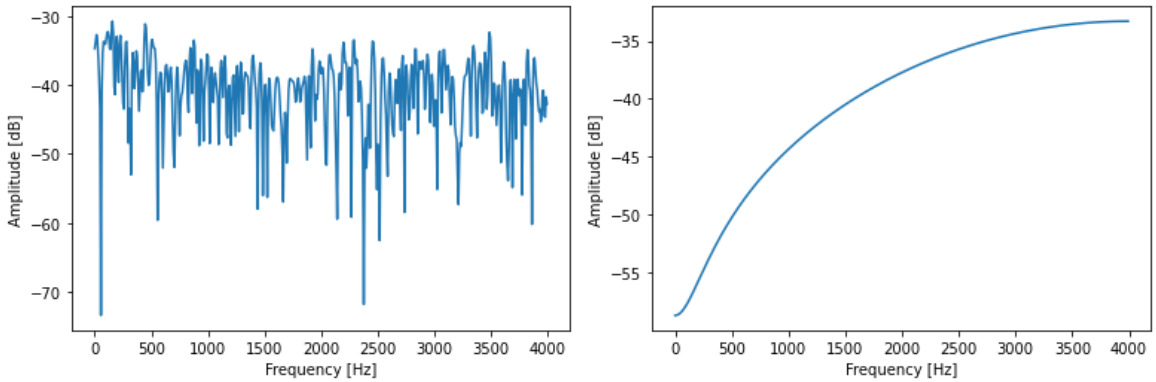

Figure 4.16: Cumulative frequency responses of 300 and 3 filter lengths, respectively.

4.10. IMPROVING THE PERFORMANCE OF THE VGG16 ARCHITECTURE

In an effort to improve the performance of the VGG16 architecture, the network is extended by introducing a stack of bidirectional LSTMs. In the previously discussed gender classification experiments, the ResNet block with BiLSTM architecture performed slightly better than its counterpart because it incorporated sequence learning.

This extended architecture was trained using the same configuration as the previous VGG16 model described in table 4.4. The resulting model did not offer a significant boost in performance. Unlike the explanations provided for the SampleCNN models, it is not quite straightforward how the VGG16 architecture can be improved using interpretability. The interpretations are useful to confirm whether or not the model is latching on to the correct information through visual inspection. The explanations are especially useful if there are artefacts within the input data that can only be observed in its spectrogram form.

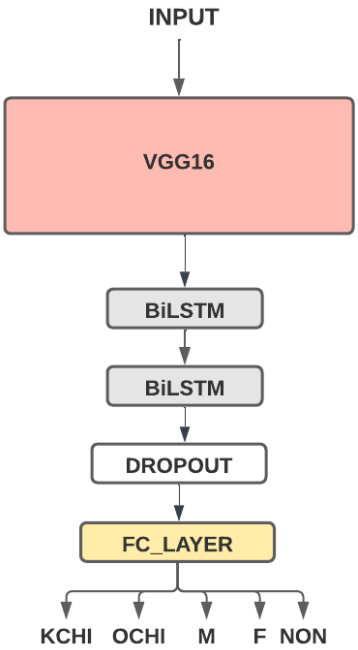

Figure 4.17: Extended VGG16 architecture.

Furthermore, it may be helpful to train different established architectures and interpret their decisions to see which model best captures the relevant information. Referring to the taxonomy established in section 2.4.6, interpretations for the VGG16 are most likely limited to passive interpretability methods. Without a clear motivation as to how to improve the network to learn the speaker's discriminative information, it is not immediately obvious if a form of intervention (active interpretability) can be introduced before training the model to improve its performance.

4.11. IMPROVING PERFORMANCE ON THE LENA RECORDED DATA

Although the proposed models managed to be competitive against the current state-of-the-art voice type classifier, the performance on the VanDam dataset shows that the model does not have good enough transferability for unseen data collected from a different acoustic environment. This is a massive problem since children's natural language environments vary greatly. To this effect, one may consider augmenting the speaker diarisation results with speaker verification. Speaker verification is a classification task in which a person's identity is verified using their voice signature/characteristics. This is similar to biometric verification, where a person's facial features or fingerprint is used to authenticate their identity by doing a similarity search through a database. A voice signature can be an embedding vector such as the d-vector and x-vectors discussed in section 1.3.2, or voice samples in their raw waveform or spectrogram form.

4.11.1. Speaker type similarity estimation.

The speaker verification problem is approached via speaker type similarity estimation for this study. Similarity estimation or similarity learning aims to learn a similarity function that can measure how similar objects are and returns a similarity score. Typically, the higher the similarity score, the more similar the objects are, and a low similarity score signifies that the objects are different. Hence, given two speech utterances, the speaker type similarity model determines if the two given utterances belong to the same speaker class.

The primary distinction between speaker type similarity estimation and voice type classification is that the latter determines whether a given utterance belongs to a male, female, key-child, other-child, or non-speech class of speakers, whereas the former verifies whether the claimed speaker class is correct.

The speaker similarity problem is investigated via the Siamese architecture [133]. As shown in figure 4.18, this is a shared weights architecture that receives two input utterances and produces comparable output vectors. In the original implementation [134], the twin networks are joined by an energy function which would decrease the energy of the two inputs if they are dissimilar and increase if they are similar.

In this work, the Euclidean distance is employed to calculate the distance between the speaker embeddings[24]. This distance between the two speaker embedding vectors *S1* and *S2* is determined by:

$$\|S1 - S2\|_2 = \sqrt{\sum_{i=1}^{n}(S1_i - S2_i)^2} \tag{4.11.1}$$

The calculated distance is fed to a layer with a sigmoid activation function which outputs a value between zero and one. This value determines the similarity of the speaker embeddings, with zero being dissimilar and one being identical.

[24] Speaker embeddings are the results of transformation of the speech samples by the shared weights network.

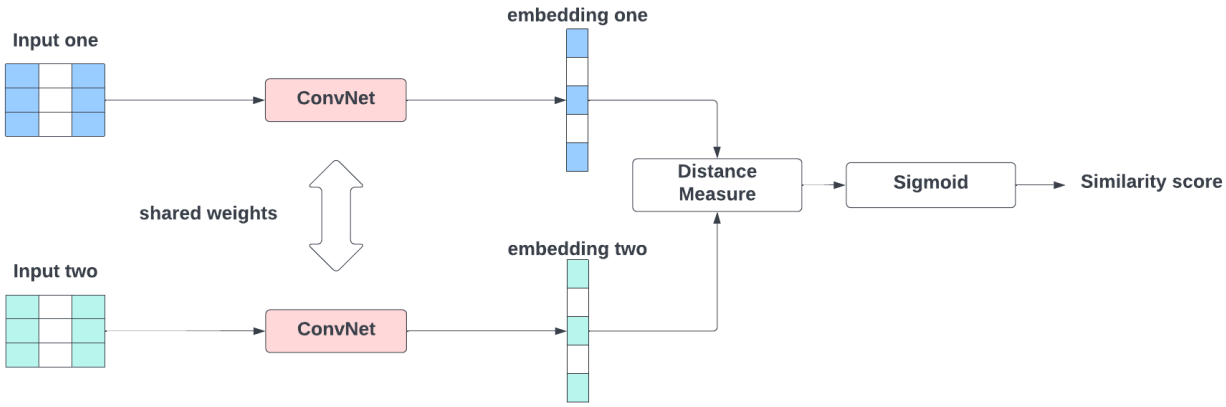

Figure 4.18: Example of the Siamese network architecture.

The siamese approach is fundamentally different to the approach of the previously investigated architectures. The former learns representations based on the difference between speakers instead of formulation representations from ground zero. This enables siamese networks to learn from fewer training instances.

The dataset is reorganised into positive and negative pairs to train the network. Positive pairs are two instances of speech that belong to the same speaker class, the label for positive pairs is one, signifying that these pairs are similar. Negative pairs are two speech instances that belong to different speaker classes, e.g., a pair of a male speech sample and a female speech sample constitutes a negative pair. The label for a negative pair is a zero, signifying that the pair is not similar.

The pairs are generated randomly; for positive pairs, each speech sample in a speaker subset is randomly paired with another sample from the same speaker class. The opposite is true for negative pairs, each sample in a speaker subset is randomly paired with a sample from another speaker class. Due to memory limitations, only 4500 samples are retained for each speaker class after random shuffling. As a result, the total pairs generated amount to 45000; half of these are positive pairs, and the other are negative pairs. 10% of the available data is used as the validation set during training.

To limit the discussion, the task is only pursued via the proposed VGG16 architecture shown in figure 4.3. This architecture class outperformed the baseline, and the SampleCNN models and expected gradients explanations have shown that its predictions can be interpreted in a human-readable form.

This network serves as the backbone feature extractor for the Siamese network. The last fully-connected layer of the proposed VGG16 architecture is replaced with a 64-neuron fully-connected layer. This means the speaker embedding *S1* and *S2* are 64-dimensional vectors. Since the speaker type estimation problem is essentially a binary classification task, the network is trained to minimize the binary cross-entropy loss defined in equation 4.4.1.

It was found effective to train the network for only 100 epochs with a stochastic gradient optimiser. The initial learning rate was set to 0.01 and made to reduce by a factor of 0.2 if the validation accuracy did not improve for ten consecutive epochs.

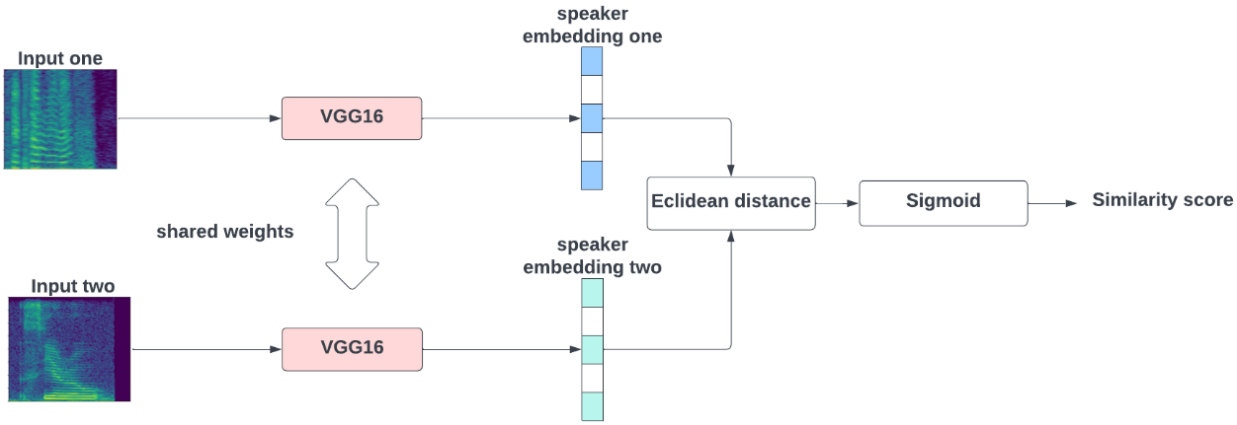

Figure 4.19: Proposed Siamese network architecture for speaker type similarity estimation.

For the testing step, the VanDam dataset is used to generate the positive and negative pairs for the test hold-out dataset. The model achieved a 68% accuracy on the pairs randomly generated from the VanDam dataset. The table below gives the performance of the siamese model in terms of classification metrics. The model has a slightly lower precision score for the negative pairs signifying that the model experiences more false-positive classifications for this class. On the other hand, the positive pairs have an overall lower F1 measure and recall indicating the model has more false negatives collectively for this class. This can also be confirmed by looking at the confusion matrix shown in figure 4.20, and the model has fewer misclassifications for negative pairs than for positive pairs

Table 4.10: Classification metrics for the siamese model.

Class	Precision	Recall	F1-score
Negative pair	0.67	0.71	0.69
Positive pair	0.69	0.66	0.67

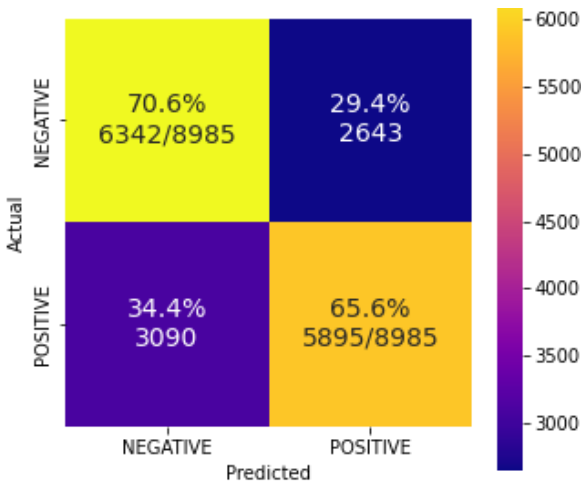

Figure 4.20: Confusion matrix for Siamese model.

4.11.2. **Voice type classification with voice type similarity estimation**.

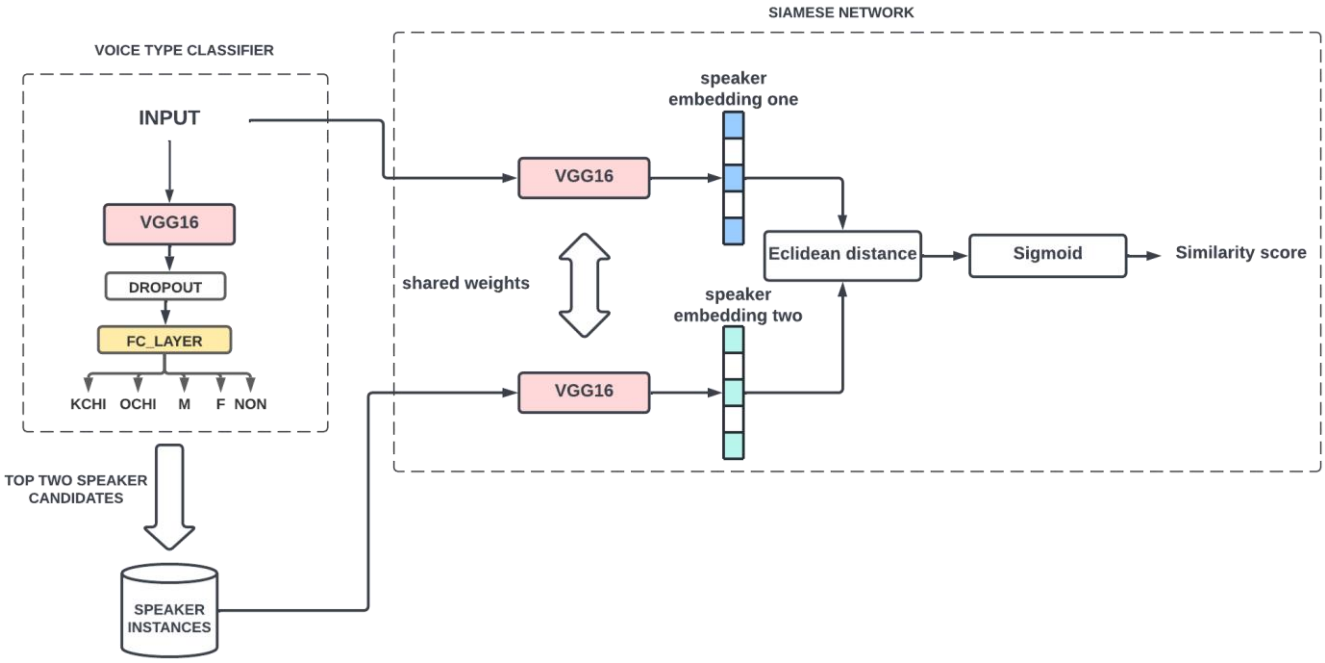

Figure 4.21: Integration of voice type classifier with voice type estimation.

The two classifiers are integrated to produce one diarisation system in such a way that the voice type classifier nominates the possible candidates for a speech segment, and the voice type similarity estimator validates which of the given speaker candidates is the true speaker. The final layer of the VGG16 multiclass voice type classifier shown in figure 4.3 has a softmax activation function. Meaning each prediction is a distribution of probabilities of each target class over all possible target classes. The softmax activation function is defined as:

$$\hat{p}_k = \sigma(\boldsymbol{s}(\boldsymbol{X}))_k = \frac{e^{(s_k(\boldsymbol{X}))}}{\sum_{J=1}^{K} e^{(s_j(\boldsymbol{X}))}}, \tag{4.11.2}$$

Where K represents the total number of classes, $\boldsymbol{s}(\boldsymbol{X})$ is the vector containing the scores of each class for the instance $\boldsymbol{X}$ and $\sigma(\boldsymbol{s}(\boldsymbol{X}))_k$ is the estimated likelihood that instance x belongs to class k given the scores of each class for that instance.

Since the VGG16 multiclass voice type classifier estimates the softmax probabilities for each speaker class, the best two candidates are chosen by picking the classes with the two highest probability scores. The validation is then done by estimating the similarity between the given speech segment and the stored speech samples for each of the two possible candidate speakers. The candidate with the highest mean similarity value is chosen as the most likely speaker type. Ten utterances are randomly chosen and kept for each speaker class. The mean similarity for each speaker is determined by:

$$mean\ similarity = \bar{L} = \frac{1}{n}\sum_{i=1}^{n} L_{i,s}. \tag{4.11.3}$$

Where n is the number of speaker utterances and $L_{i,s}$ is the similarity score estimated by the voice type estimator for speaker utterance i and speech input s.

Figure 4.22: Confusion matrixes of (a) The combined model (b) the Original single classifier.

Figure 4.22 shows the confusion matrixes for the combined voice type classifier with voice type similarity estimation and the vanilla voice type classifier based on the VGG16 multiclass architecture. Both models were trained on the entire dataset, leaving the VanDam corpus as the holdout test set. Hence the confusion matrixes give the performance of the models on the holdout test set.

The confusion matrixes show how the combined classifier has a lower performance compared to the original single classifier. The classes that suffer the most are the male and other child speakers. The combined model misclassifies more male instances under the female and child classes but surprisingly decreases the amount of misclassified male instances in the other child and non-speech classes. The performance on the other-child class falls drastically; as indicated by the confusion matrices, the performance decreases from 14.4 to 1.8 %. More instances of this speaker type are misclassified under the male and female classes.

The reason for the confusion matrices is to help determine which situations require the verification of the predictions of the voice type classifier by voice type similarity estimation. For instance, fewer other child instances are misclassified as male instances, and fewer male instances are misclassified under the other child speaker class. This type of trade-off in performance is not experienced between all of the possible speaker pairs. In some situations, a decrease in misclassifications for one class corresponds to an increase in misclassifications for the other. If the increase in misclassification is less significant than the decrease in misclassification, one could opt to allow the verification to be executed in such a situation. For instance, between the key-child and male speaker classes, the combined classifier has fewer misclassifications of the key-child instances as male, but more male samples are classified under the key-child. The former has a more significant decrease than the increase in the latter situation. The model benefits from using voice type verification in this situation.

It is important to note that there is a possibility of inherent overlap between the class of speakers either due to the actual similarity of the speakers, mislabelling of speaker instances or unlabelled cross-talking. The figure below shows the results of running PCA and TSNE on the spectrogram

images, similar to what is explored in section 3.5.1. However, unlike the gender perception problem, both methods could not form a meaningful separation between the speaker classes.

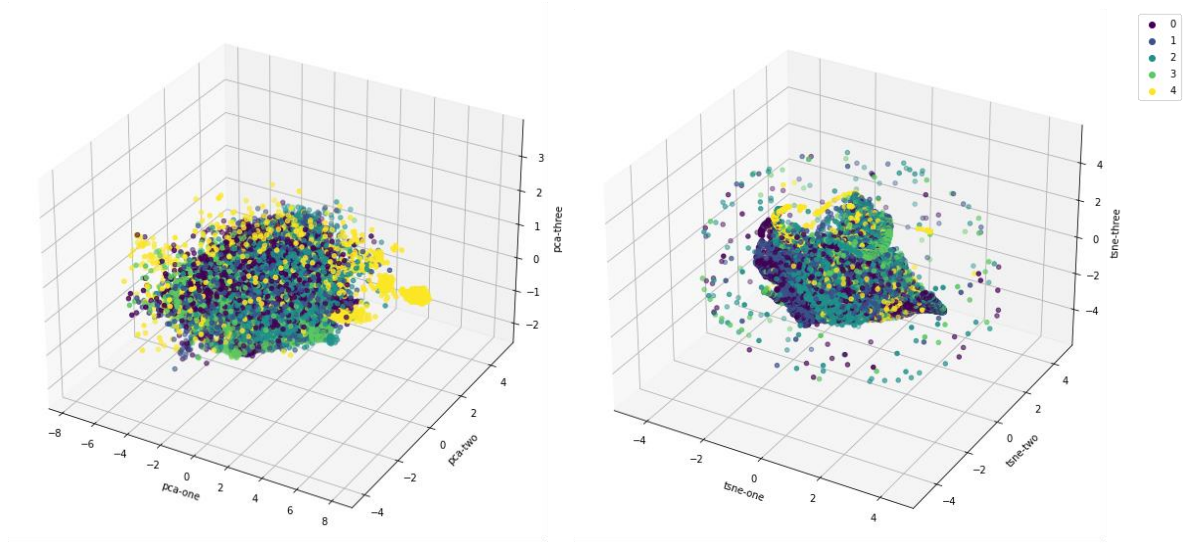

Figure 4.23: Visualisation of PCA and TSNE clustering on spectrogram images.

The same pattern is observed when the instances in the dataset are transformed into the speech estimation space detailed in table 3.1. The non-speech class is not extracted since it contains non-speaker instances.

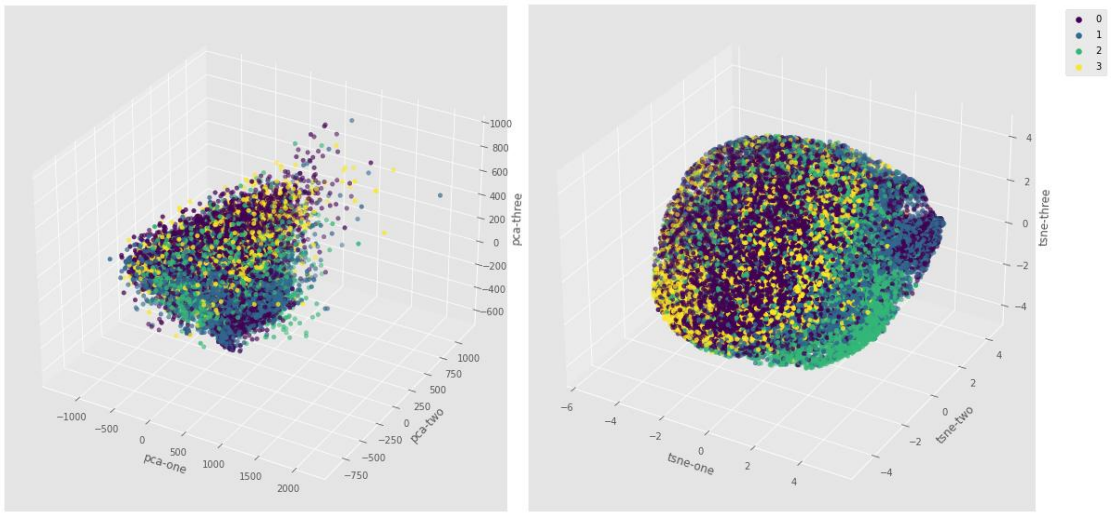

Figure 4.24: Visualisation of PCA and TSNE clustering on the extracted speech features.

To visualise the extent to which the VGG16 network achieves separability of these classes, PCA and TSNE can be performed on the embeddings/activations extracted from the hidden layers. For this particular example, a 64-neuron dense layer is added before the classification layer to reduce the dimension of the features extracted by the neural network. The image below shows the visualisation of PCA and TSNE on the extracted features from the hidden layer of the trained VGG16 architecture

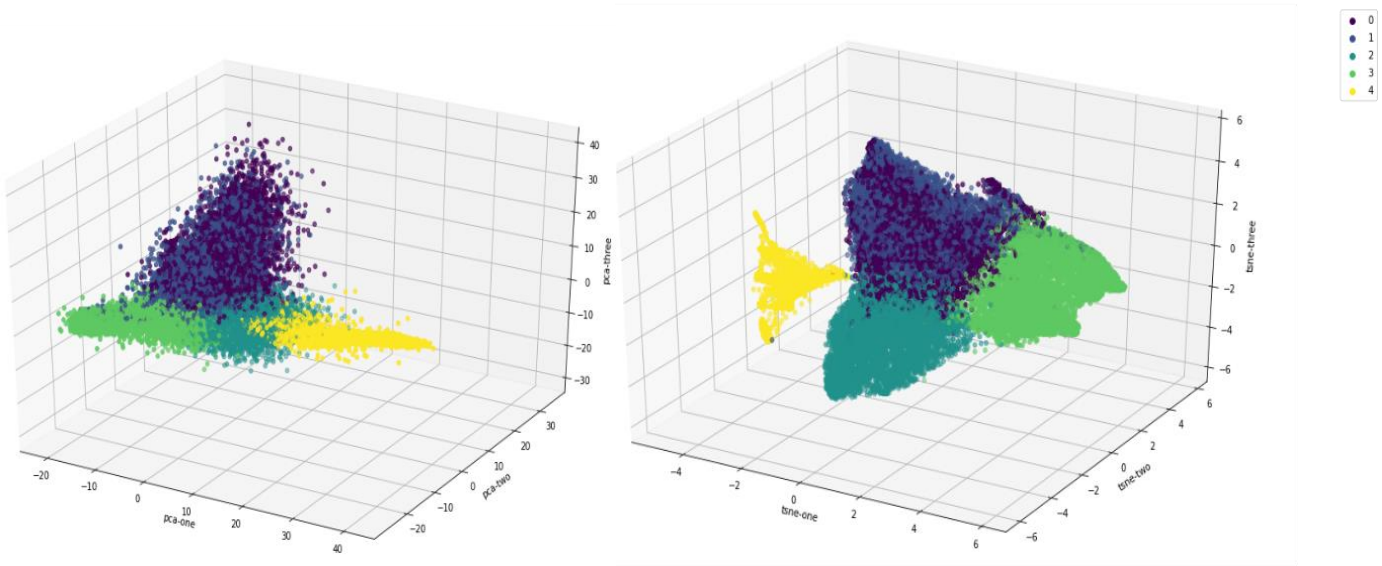

Figure 4.25: Visualisation of PCA and TSNE clustering on the VGG16 embeddings.

The features show a fair clustering of the five classes for both methods. The non-speech class represented in yellow (4) has the furthest cluster from all the other classes signifying why the network performs very well in identifying this class. All the speaker classes have a general overlap, most notably the key-child (0) and female (1) speaker classes. This explains why the model often confuses these two classes. Ideally, one would want to maximise the separation between all the classes. This could be achieved by using loss functions such as the additive margin softmax, which has been found successful in face and speaker recognition applications [135][136].

Ultimately, the performance of the combined classifier shows how such a design should be implemented with caution. The simple implementation described in this section introduces errors to classes in which the model was performing fine before the fact and a compound of errors to situations where the original model was already failing. Furthermore, the choice of the stored instances for each speaker class can be improved by introducing a methodology in which the stored speaker instances are close enough in the distance space to represent a speaker fully but not too close that they are identical.

4.12. DISCUSSION

Following the validation criteria established in section 1.8, the proposed architectures outperform the current state-of-the-art baseline on all classification metrics. Of the two proposed classes of models, the VGG16 significantly outperforms the SampleCNN models in both multilabel and multiclass configurations.

The model interpretations for the VGG16 model are explored using both attribution and hidden semantic approaches using the expected gradients method. The attribution explanations show the VGG16 model focusing on the relevant spectral information related to discriminative speaker properties. This was also confirmed by relating the attribution explanations for the model's decision to voice source properties extracted from the input samples. The hidden semantic explanations extend

the attribution explanations to the hidden layers, showing the Shapley value explanations superimposed on the model's feature maps.

On the other hand, the SampleCNN models are interpreted by analysing the spectral response of the filters in the first convolution layer. The response was shown to frequency information that relates the high formant frequencies and the high-frequency noise comparing the explanations for the proposed models. The interpretations for the SampleCNN architecture are a more faithful representation of the information learned by the model. The cumulative frequency response is a direct interpretation of the response of the filters in the first convolutional layer. The explanations are more actionable in regard to formulating improvements for the model. As explored in section 4.9, the filter responses can be tuned by varying the filter lengths for the first convolutional layers. This was shown to produce a significant improvement in performance for some speaker classes.

The explanations for the VGG16 model are also informative, and in some cases, they can be supported by speaker estimations support them. The major drawback to these explanations is that they are less actionable regarding improving the architecture. However, the explanations are more human-readable because of the visual cues provided by the saliency map. It is much easier on a higher level to qualitatively ascertain that the model focuses on the wrong ideas represented on the spectrograms. Hence the improvement might not come in the form of actionability on the model itself but on the nature of the input, which can be transformed into other time-frequency representations such as the MFCCs, scalograms, and mel spectrograms.

Even though the models outperform the current state-of-the-art voice type classifier, their performance on the VanDam dataset shows that the models do not have a good transferability on data from a different acoustic environment than that contained in the training dataset. The exploration of the siamese network shows one of the ways a model can be protected from invalidating the diarisation system’s future predictions.

CHAPTER V

5. CONCLUSION AND RECOMMENDATIONS

5.1. CONCLUSION

The study explores the formulation of a diarisation model for child natural language environment analysis under a voice type classification context. The main objective was to explore how interpretability applies to the diarisation system and how it can affect its overall performance and design. It was found that the most important part of this problem is to define the interpretability requirements. However, because the discussion on interpretability encompasses multiple ideas, the literature review attempts to establish a consolidated framework to explore the different types of interpretability techniques and the principles of interpreting deep learning models.

The most important aspects of this framework are the categorisation of interpretability methods based on the explanatory power of the explanations they produce and the scope of interpretability with respect to the input space.

Based on explanatory power, logic rules are the most compelling explanations, followed by hidden semantics explanations, attributions, and finally, explanations by example. Attribution-type methods are the most popular type of explanations, especially for deep neural networks. Ultimately, the choice of the interpretability method depends on several factors:

i. The nature of the input - either the speech is in its time-frequency domain representation as spectrograms or time-domain representations as raw waveforms.

ii. The type of machine learning algorithm – there are specific interpretability methods applicable to a specific class of machine learning techniques.

iii. The fidelity of the explanations - the interpretability technique can produce in regards to explanatory power.

iv. The scope of the interpretations with respect to the input space - local interpretability by interpretations of individual predictions or global interpretability through explaining the entire decision logic of the model.

The interaction between these factors is first established via the gender perception problem. First, the discriminative information is explicitly described in relation to how it can be used to differentiate between speakers. The strongest predictor for speaker gender is the frequency of vibration of the vocal cords, which determines the source’s fundamental frequency (F0) and the resulting pitch. This is followed by the vocal tract's acoustic resonances, which relate to formant frequencies.

The gender perception experiments show how the expected gradients technique can be used to produce attribution explanations for deep neural networks trained on spectrograms and raw waveforms respectively. The expected gradients technique was chosen because it satisfies several desirable axioms, such as implementation and performance at saturation. However, these explanations are shown to be more useful for models trained on spectrograms as it is easier to infer the model has latched on to the idea of fundamental frequency or the formants by analysing the attributions on the spectrogram.

To satisfactorily explain a model trained on raw waveforms, spectral analysis is required to derive the cumulative frequency response of the filters in the first convolutional layer. Since the explanations for the raw waveform model are derived through explanations of a hidden layer, they have more explanatory power than the attribution explanations of the spectrogram-trained models.

The same sentiment is applied to finding a solution to the diarisation problem. The proposed voice type classification models outperform the current state-of-the-art benchmark. Moreover, the proposed VGG16 model trained on spectrograms significantly outperforms its SampleCNN counterpart. The VGG16 model is interpreted via the expected gradients technique, and the SampleCNN model via spectral analysis of the filters in the first convolutional layer. Interpretations for both models were able to uncover speaker discriminative information. However, the attribution explanations for the VGG16 did not help provide insights toward model improvement. For the SampleCNN model, the cumulative frequency response of the filters in the first convolutional layer can be adjusted by varying the filter length. This can be done so that the response covers the relevant frequency information, which can improve the model's performance.

In cases where the diarisation model is known to fail, intervention can be introduced in the form of a supporting model. The study proposes a siamese network that learns to estimate the similarity between the pair of speakers. Interpretability can also be used here as a feedback mechanism to ensure that the model learns the differences specific to these speakers.

In conclusion, a diarisation model can be regarded as interpretable if the model's decisions and or learned representations can be explained and confirmed considering speaker discriminative information. The models trained on the spectrogram features perform significantly better than the current state of the art, and the explanations for these models provide local to semi-local interpretability though the explanations are easily human-readable. On the other hand, the SampleCNN architectures have similar performance to the current benchmark, and the explanations provide global interpretability and are more informative regarding model improvement.

5.2. RECOMMENDATIONS

One of the main hurdles in developing child-related speech applications is the lack of availability of child-centred corpora. On the other hand, there is a large amount of publicly available adult speech data, and consequently, this area of research has far exceeded the advancement of child speech applications. It would be interesting to investigate if transfer learning from adult speech to child speech can help improve the overall performance of the diarisation model and what would be the implications of the different transfer learning techniques on the model's interpretability.

The interpretability explanations covered in this study try to speculate which features of the input are relevant to the model's decision. In the future, one may want to speculate how these features are used and the causal relationships learned by the model.

REFERENCES

[1] A. Vouloumanos and S. Curtin, ‘Foundational tuning: How infants’ attention to speech predicts language development’, *Cogn Sci*, vol. 38, no. 8, pp. 1675–1686, 2014, doi: 10.1111/cogs.12128.

[2] P. K. Kuhl, ‘Early Language Learning and Literacy: Neuroscience’, *Mind Brain Education*, vol. 5, no. 3, pp. 128–142, 2011, doi: 10.1111/j.1751-228X.2011.01121.x.Early.

[3] N. F. Ramírez, S. R. Lytle, and P. K. Kuhl, ‘Parent coaching increases conversational turns and advances infant language development’, *Proc Natl Acad Sci U S A*, vol. 117, no. 7, pp. 3484–3491, 2020, doi: 10.1073/pnas.1921653117.

[4] J. A. Richards, J. Gilkerson, D. Xu, and K. Topping, ‘How Much Do Parents Think They Talk to Their Child?’, *J Early Interv*, vol. 39, no. 3, pp. 163–179, 2017, doi: 10.1177/1053815117714567.

[5] J. Karaday, C. Scaff, and A. Cristia, ‘Diarization in Maximally Ecological Recordings: Data from Tsimane Children’, in *The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*, 2018, no. August, pp. 30–35. doi: 10.21437/sltu.2018-7.

[6] J. H. L. Hansen, M. Najafian, R. Lileikyte, D. Irvin, and B. Rous, ‘Speech and language processing for assessing child–adult interaction based on diarization and location’, *Int J Speech Technol*, vol. 22, no. 3, pp. 697–709, 2019, doi: 10.1007/s10772-019-09590-0.

[7] J. Xie, L. P. García-Perera, D. Povey, and S. Khudanpur, ‘Multi-PLDA Diarization on Children’s Speech’, in *Interspeech*, 2019, pp. 376–380. doi: 10.21437/interspeech.2019-2961.

[8] G. Yeung and A. Alwan, ‘On the difficulties of automatic speech recognition for kindergarten-aged children’, *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2018-Septe, no. September, pp. 1661–1665, 2018, doi: 10.21437/Interspeech.2018-2297.

[9] J. Gilkerson *et al.*, ‘Mapping the early language environment using all-day recordings and automated analysis’, *Am J Speech Lang Pathol*, vol. 26, no. 2, pp. 248–265, May 2017, doi: 10.1044/2016_AJSLP-15-0169.

[10] M. VanDam and N. H. Silbert, ‘Fidelity of automatic speech processing for adult and child talker classifications’, *PLoS One*, vol. 11, no. 8, pp. 1–13, 2016, doi: 10.1371/journal.pone.0160588.

[11] M. Lavechin, R. Bousbib, H. Bredin, E. Dupoux, and A. Cristia, ‘An open-source voice type classifier for child-centered daylong recordings’, May 2020, Accessed: Jun. 26, 2020. [Online]. Available: http://arxiv.org/abs/2005.12656

[12] S. E. Ambrose, M. Vandam, and M. P. Moeller, ‘Linguistic input, electronic media, and communication outcomes of toddlers with hearing loss’, *Ear Hear*, vol. 35, no. 2, pp. 139–147, 2014, doi: 10.1097/AUD.0b013e3182a76768.

[13] J. Gilkerson, J. A. Richards, S. F. Warren, D. K. Oller, R. Russo, and B. Vohr, ‘Language experience in the second year of life and language outcomes in late childhood’, *Pediatrics*, vol. 142, no. 4, 2018, doi: 10.1542/peds.2017-4276.

[14] E. C. Merz, E. A. Maskus, S. A. Melvin, X. He, and K. G. Noble, 'Socioeconomic Disparities in Language Input Are Associated With Children's Language-Related Brain Structure and Reading Skills', *Child Dev*, vol. 91, no. 3, pp. 846–860, 2020, doi: 10.1111/cdev.13239.

[15] J. R. Dykstra, M. G. Sabatos-Devito, D. W. Irvin, B. A. Boyd, K. A. Hume, and S. L. Odom, 'Using the Language Environment Analysis (LENA) system in preschool classrooms with children with autism spectrum disorders', *Autism*, vol. 17, no. 5, pp. 582–594, Sep. 2013, doi: 10.1177/1362361312446206.

[16] F. J. Zimmerman *et al.*, 'Teaching by listening: The importance of adult-child conversations to language development', *Pediatrics*, vol. 124, no. 1, pp. 342–349, Jul. 2009, doi: 10.1542/peds.2008-2267.

[17] M. Canault, M. T. le Normand, S. Foudil, N. Loundon, and H. Thai-Van, 'Reliability of the Language ENvironment Analysis system (LENA™) in European French', *Behav Res Methods*, vol. 48, no. 3, pp. 1109–1124, 2016, doi: 10.3758/s13428-015-0634-8.

[18] A. Cristia, S. Ganesh, M. Casillas, and S. Ganapathy, 'Talker diarization in the wild: The case of child-centered daylong audio-recordings', in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2018, vol. 2018-Septe, pp. 2583–2587. doi: 10.21437/Interspeech.2018-2078.

[19] M. Casillas and A. Cristia, 'A step-by-step guide to collecting and analyzing long-format speech environment (LFSE) recordings', *Collabra Psychol*, vol. 5, no. 1, pp. 1–21, 2019, doi: 10.1525/collabra.209.

[20] A. le Franc *et al.*, 'The ACLEW DiViMe: An easy-to-use diarization tool', in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2018, vol. 2018-Septe, pp. 1383–1387. doi: 10.21437/Interspeech.2018-2324.

[21] 'Divime's Documentation'. https://divime-fetze.readthedocs.io/en/latest/#welcome-to-divime-s-documentation (accessed Jun. 26, 2020).

[22] P. G. Shivakumar and P. Georgiou, 'Transfer Learning from Adult to Children for Speech Recognition: Evaluation, Analysis and Recommendations', in *arXIv*, no. 6, 2018. doi: arXiv:1805.03322.

[23] M. VanDam *et al.*, 'HomeBank: An Online Repository of Daylong Child-Centered Audio Recordings', *Semin Speech Lang*, vol. 37, no. 2, pp. 128–142, 2016, doi: 10.1055/s-0036-1580745.

[24] X. Wang, J. Du, A. Cristia, L. Sun, and C.-H. Lee, 'A Study of Child Speech Extraction Using Joint Speech Enhancement and Separation in Realistic Conditions', *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7304–7308, 2020, doi: 10.1109/icassp40776.2020.9053875.

[25] S. Kataria, P. S. Nidadavolu, J. Villalba, N. Chen, P. Garcia-Perera, and N. Dehak, 'Feature Enhancement with Deep Feature Losses for Speaker Verification', in *Institute of Electrical and Electronics Engineers (IEEE)*, Apr. 2020, pp. 7584–7588. doi: 10.1109/icassp40776.2020.9053110.

[26] S. Kataria, P. S. Nidadavolu, J. Villalba, and N. Dehak, ‘Analysis of Deep Feature Loss Based Enhancement for Speaker Verification’, *arXiv preprint arXiv:2002.00139*, pp. 459–466, 2020, doi: 10.21437/odyssey.2020-66.

[27] L. P. Garcia Perera *et al.*, ‘Speaker Detection in the Wild: Lessons Learned from JSALT 2019’, *arXiv preprint arXiv:1912.00938*, pp. 415–422, 2020, doi: 10.21437/odyssey.2020-59.

[28] A. Cristia and H. Colleran, ‘Excerpts from daylong recordings of young children learning many languages in Vanuatu’. 2007.

[29] L. Sun *et al.*, ‘A Novel LSTM-Based Speech Preprocessor for Speaker Diarization in Realistic Mismatch Conditions’, *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2018-April, pp. 5234–5238, 2018, doi: 10.1109/ICASSP.2018.8462311.

[30] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, and K. Nagamatsu, ‘End-to-End Neural Diarization: Reformulating Speaker Diarization as Simple Multi-label Classification’, pp. 1–11, 2020, [Online]. Available: http://arxiv.org/abs/2003.02966

[31] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, ‘End-to-end neural speaker diarization with permutation-free objectives’, in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2019, vol. 2019-Septe, pp. 4300–4304. doi: 10.21437/Interspeech.2019-2899.

[32] M. Najafian and J. H. L. Hansen, ‘Speaker independent diarization for child language environment analysis using deep neural networks’, in *2016 IEEE Workshop on Spoken Language Technology, SLT 2016 - Proceedings*, Dec. 2017, pp. 114–120. doi: 10.1109/SLT.2016.7846253.

[33] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, ‘End-To-End Neural Speaker Diarization with Self-Attention’, *2019 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019 - Proceedings*, pp. 296–303, 2019, doi: 10.1109/ASRU46091.2019.9003959.

[34] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, ‘Fully Supervised Speaker Diarization’, *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2019-May, pp. 6301–6305, 2019, doi: 10.1109/ICASSP.2019.8683892.

[35] E. Fini and A. Brutti, ‘Supervised Online Diarization with Sample Mean Loss for Multi-Domain Data’, in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7134–7138. doi: 10.1109/icassp40776.2020.9053477.

[36] M. Rouvier, P. M. Bousquet, and B. Favre, ‘Speaker diarization through speaker embeddings’, in *2015 23rd European Signal Processing Conference, EUSIPCO 2015*, 2015, pp. 2082–2086. doi: 10.1109/EUSIPCO.2015.7362751.

[37] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, ‘Deep neural network embeddings for text-independent speaker verification’, in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2017, vol. 2017-Augus, pp. 999–1003. doi: 10.21437/Interspeech.2017-620.

[38] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, ‘Generalized end-to-end loss for speaker verification’, *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2018-April, pp. 4879–4883, 2018, doi: 10.1109/ICASSP.2018.8462665.

[39] J. Jung, H. Heo, I. Yang, S. Yoon, H. Shim, and H. Yu, ‘D-vector based speaker verification system using Raw Waveform CNN’, *Advances in Intelligent Systems Research (AISR)*, vol. 150, no. Anit 2017, pp. 126–131, 2018, doi: 10.2991/anit-17.2018.21.

[40] Y. Gong and C. Poellabauer, ‘How do deep convolutional neural networks learn from raw audio waveforms?’, in *ICLR 2018 Conference Paper*, 2018, pp. 1–12.

[41] Z. C. Lipton, ‘The mythos of model interpretability’, *Commun ACM*, vol. 61, no. 10, pp. 35–43, 2018, doi: 10.1145/3233231.

[42] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, ‘A survey on neural network interpretability’, *ArXiv*, Dec. 2020, Accessed: Mar. 17, 2021. [Online]. Available: http://arxiv.org/abs/2012.14261

[43] D. v. Carvalho, E. M. Pereira, and J. S. Cardoso, ‘Machine learning interpretability: A survey on methods and metrics’, *Electronics (Switzerland)*, vol. 8, no. 8, pp. 1–34, 2019, doi: 10.3390/electronics8080832.

[44] N. Burkart and M. F. Huber, ‘A survey on the explainability of supervised machine learning’, *Journal of Artificial Intelligence Research*, vol. 70. pp. 245–317, 2021. doi: 10.1613/JAIR.1.12228.

[45] M. T. Ribeiro, S. Singh, and C. Guestrin, ‘“Why should i trust you?” Explaining the predictions of any classifier’, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-Augu, pp. 1135–1144, 2016, doi: 10.1145/2939672.2939778.

[46] H. Mureşan and M. Oltean, ‘Fruit recognition from images using deep learning’, *ArXiv*, vol. 10, pp. 26–42, 2017, doi: 10.2478/ausi-2018-0002.

[47] D. Alvarez-Melis and T. S. Jaakkola, ‘On the robustness of interpretability methods’, *ArXiv*, vol. abs/1806.0, 2018.

[48] S. Lundberg and S.-I. Lee, ‘A Unified Approach to Interpreting Model Predictions’, *NIPS*, 2017, doi: https://arxiv.org/abs/1705.07874.

[49] W. Thomson and A. E. Roth, *The Shapley Value: Essays in Honor of Lloyd S. Shapley.*, vol. 58, no. 229. 1991. doi: 10.2307/2554979.

[50] C. Molnar, *Interpretable Machine Learning*, 2nd ed. 2022. Accessed: Feb. 28, 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[51] K. Simonyan, A. Vedaldi, and A. Zisserman, ‘Deep inside convolutional networks: Visualising image classification models and saliency maps’, *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*, vol. abs/1312.6, 2014.

[52] A. Shrikumar, P. Greenside, and A. Kundaje, ‘Learning important features through propagating activation differences’, in *International conference on machine learning*, 2017, pp. 3145–3153. [Online]. Available: PMLR

[53] M. D. Zeiler and R. Fergus, ‘Visualizing and understanding convolutional networks’, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8689 LNCS, no. PART 1, pp. 818–833. doi: 10.1007/978-3-319-10590-1_53.

[54] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, ‘Striving for simplicity: The all convolutional net’, *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*, vol. abs/1412.6, 2015.

[55] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, ‘Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization’, *Int J Comput Vis*, vol. 128, no. 2, pp. 336–359, 2020, doi: 10.1007/s11263-019-01228-7.

[56] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. R. Müller, and W. Samek, ‘On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation’, *PLoS One*, vol. 10, no. 7, p. e0130140, 2015, doi: 10.1371/journal.pone.0130140.

[57] M. Sundararajan, A. Taly, and Q. Yan, ‘Axiomatic attribution for deep networks’, *34th International Conference on Machine Learning, ICML 2017*, vol. 7, pp. 5109–5118, 2017.

[58] R. Aumann, L. Shapley, and S. Lloyd, *Values of Non-Atomic Games*. Princeton University Press, 2015.

[59] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, ‘SmoothGrad: Removing noise by adding noise’, *ArXiv*, 2017, doi: 10.48550/ARXIV.1706.03825.

[60] G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, and S. I. Lee, ‘Improving performance of deep learning models with axiomatic attribution priors and expected gradients’, *Nat Mach Intell*, vol. 3, no. 7, pp. 620–631, 2021, doi: 10.1038/s42256-021-00343-w.

[61] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, ‘Sanity checks for saliency maps’, in *Neural Information Processing Systems*, 2018.

[62] C. Olah *et al.*, ‘The Building Blocks of Interpretability’, *Distill*, vol. 3, no. 3, 2018, doi: 10.23915/distill.00010.

[63] A. G. Huth, W. A. de Heer, T. L. Griffiths, F. E. Theunissen, and J. L. Gallant, ‘Natural speech reveals the semantic maps that tile human cerebral cortex’, *Nature*, vol. 532, no. 7600, pp. 453–458, 2016, doi: 10.1038/nature17637.

[64] S. Nishimoto, A. T. Vu, T. Naselaris, Y. Benjamini, B. Yu, and J. L. Gallant, ‘Reconstructing visual experiences from brain activity evoked by natural movies’, *Curr Biol*, vol. 21, no. 19, pp. 1641–1646, Oct. 2011, doi: 10.1016/j.cub.2011.08.031.

[65] H. Muckenhirn, V. Abrol, M. Magimai-Doss, and S. Marcel, ‘Understanding and Visualizing Raw Waveform-Based CNNs’, 2019, pp. 2345–2349. doi: 10.21437/interspeech.2019-2341.

[66] D. Salvati, C. Drioli, and G. L. Foresti, ‘End-to-end speaker identification in noisy and reverberant environments using raw waveform convolutional neural networks’, *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2019-Septe, pp. 4335–4339, 2019, doi: 10.21437/Interspeech.2019-2403.

[67] Y. Tokozume and T. Harada, ‘Learning environmental sounds with end-to-end convolutional neural network’, in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 2721–2725. doi: 10.1109/ICASSP.2017.7952651.

[68] R. Sakrepatna Srinivasamurthy, ‘Understanding 1D Convolutional Neural Networks Using Multiclass Time-Varying Signalss’, *ProQuest Dissertations and Theses*, p. 99, 2018.

[69] H. Muckenhirn, M. Magimai-Doss, and S. Marcel, ‘On learning vocal tract system related speaker discriminative information from raw signal using CNNs’, *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2018-Septe, pp. 1116–1120, 2018, doi: 10.21437/Interspeech.2018-1696.

[70] M. Ravanelli and Y. Bengio, ‘Interpretable Convolutional Filters with SincNet’, *arXiv preprint arXiv:1811.09725*, 2018, [Online]. Available: http://arxiv.org/abs/1811.09725

[71] P. Priums, ‘Gradient-Based Explanations for Audio Classifiers Computer Science’, Master Thesis, Johannes Kepler University Linz, 2019. [Online]. Available: www.jku.at

[72] A. Krug and S. Stober, ‘Gradient-Adjusted Neuron Activation Profiles for Comprehensive Introspection of Convolutional Speech Recognition Models’, no. Dl, 2020, [Online]. Available: http://arxiv.org/abs/2002.08125

[73] B. Soomro, A. Kanervisto, T. N. Trong, and V. Hautamäki, ‘Towards Debugging Deep Neural Networks by Generating Speech Utterances’, 2019, no. 1, pp. 3213–3217. doi: 10.21437/interspeech.2019-2339.

[74] A. Krug and S. Stober, ‘Introspection for convolutional automatic speech recognition’, in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2019, pp. 187–199. doi: 10.18653/v1/w18-5421.

[75] B. Blaus, ‘Medical gallery of Blausen Medical 2014’, *WikiJournal of Medicine*, vol. 1, no. 2, 2014, doi: 10.15347/wjm/2014.010.

[76] K. Stevens, ‘Acoustic Phonetics’, in *Journal of The Acoustical Society of America - J ACOUST SOC AMER*, vol. 109, 2000, p. 607. doi: 10.1121/1.1327577.

[77] G. Fant, ‘Acoustic Theory of Speech Production’, in *Acoustic Theory of Speech Production*, 1971. doi: 10.1515/9783110873429.

[78] W. T. Fitch and J. Giedd, ‘Morphology and development of the human vocal tract: A study using magnetic resonance imaging’, *J Acoust Soc Am*, vol. 106, no. 3, pp. 1511–1522, 1999, doi: 10.1121/1.427148.

[79] G. E. Peterson and H. L. Barney, ‘Control Methods Used in a Study of the Vowels’, *Journal of the Acoustical Society of America*, vol. 24, no. 2, pp. 175–184, 1952, doi: 10.1121/1.1906875.

[80] J. Hillenbrand, L. A. Getty, M. J. Clark, and K. Wheeler, ‘Acoustic characteristics of American English vowels’, *Journal of the Acoustical Society of America*, vol. 97, no. 5, pp. 3099–3111, 1995, doi: 10.1121/1.411872.

[81] A. P. Simpson, ‘Phonetic differences between male and female speech’, *Linguistics and Language Compass*, vol. 3, no. 2, pp. 621–640, 2009, doi: 10.1111/j.1749-818X.2009.00125.x.

[82] J. Kane, S. Scherer, L. P. Morency, and C. Gobl, ‘A comparative study of glottal open quotient estimation techniques’, *BDL*, vol. 178, pp. 0–41, Jan. 2013.

[83] R. Wright, C. Mansfield, and L. Panfili, ‘Voice quality types and uses in North American English’, *Anglophonia*, vol. 27, no. 27, Nov. 2019, doi: 10.4000/anglophonia.1952.

[84] J. Lohscheller, J. G. Švec, and M. Döllinger, ‘Vocal fold vibration amplitude, open quotient, speed quotient and their variability along glottal length: Kymographic data from normal subjects’, *Logoped Phoniatr Vocol*, vol. 38, no. 4, pp. 182–192, Dec. 2013, doi: 10.3109/14015439.2012.731083.

[85] G. Fant, ‘A note on vocal tract size factors and non-uniform F-pattern scalings’, in *Speech Transmission Laboratory Quarterly Progress and Status Report*, 1966, vol. 1, pp. 22–30. [Online]. Available: http://www.speech.kth.se/qpsr

[86] T. J. Neuhaus, ‘Gender Perception Dependent on Fundamental Frequency, Source Spectral Tilt, and Formant Frequencies’, Bowling Green State University, 2020. [Online]. Available: http://rave.ohiolink.edu/etdc/view?acc_num=bgsu1586535654035042

[87] L. Ma and J. D. Woolley, ‘Young Children’s Sensitivity to Speaker Gender When Learning From Others’, *Journal of Cognition and Development*, vol. 14, no. 1, pp. 100–119, Jan. 2013, doi: 10.1080/15248372.2011.638687.

[88] V. Cartei, A. Garnham, J. Oakhill, R. Banerjee, L. Roberts, and D. Reby, ‘Children can control the expression of masculinity and femininity through the voice’, *R Soc Open Sci*, vol. 6, no. 7, p. 190656, Jul. 2019, doi: 10.1098/rsos.190656.

[89] V. G. Skuk and S. R. Schweinberger, ‘Influences of fundamental frequency, formant frequencies, aperiodicity, and spectrum level on the perception of voice gender’, *Journal of Speech, Language, and Hearing Research*, vol. 57, no. 1, pp. 285–296, 2014, doi: 10.1044/1092-4388(2013/12-0314).

[90] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, ‘Librispeech: An ASR corpus based on public domain audio books’, *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2015-Augus, pp. 5206–5210, 2015, doi: 10.1109/ICASSP.2015.7178964.

[91] D. R. Feinberg and O. Cook, ‘VoiceLab: Automated Reproducible Acoustic Analysis’, 2020. doi: 10.31234/osf.io/v5uxfV.

[92] P. Boersma and V. van Heuven, ‘Speak and unSpeak with Praat’, *Glot International*, vol. 5, no. 9–10, pp. 341–347, 2001.

[93] V. Rokhlin, A. Szlam, and M. Tygert, ‘A randomized algorithm for principal component analysis’, *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1100–1124, Sep. 2009, doi: 10.1137/080736417.

[94] L. van der Maaten and G. Hinton, ‘Visualizing data using t-SNE’, *Journal of Machine Learning Research*, vol. 9, no. November 2008, pp. 2579–2625, Nov. 2008.

[95] P. Boersma, ‘Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound’, *Proceedings of the Institute of Phonetic Sciences*, vol. 17, pp. 97–110, 1993.

[96] K. Pisanski and D. Rendall, ‘The prioritization of voice fundamental frequency or formants in listeners’ assessments of speaker size, masculinity, and attractiveness’, *J Acoust Soc Am*, vol. 129, no. 4, pp. 2201–2212, Apr. 2011, doi: 10.1121/1.3552866.

[97] D. R. R. Smith and R. D. Patterson, ‘The interaction of glottal-pulse rate and vocal-tract length in judgements of speaker size, sex, and age’, *J Acoust Soc Am*, vol. 118, no. 5, pp. 3177–3186, Nov. 2005, doi: 10.1121/1.2047107.

[98] D. A. Puts, C. L. Apicella, and R. A. Cárdenas, ‘Masculine voices signal men’s threat potential in forager and industrial societies’, *Proc Biol Sci*, vol. 279, no. 1728, pp. 601–609, Feb. 2012, doi: 10.1098/rspb.2011.0829.

[99] W. T. Fitch, ‘Vocal tract length and formant frequency dispersion correlate with body size in rhesus macaques’, *J Acoust Soc Am*, vol. 102, no. 2, pp. 1213–1222, Aug. 1997, doi: 10.1121/1.421048.

[100] J. P. Teixeira, C. Oliveira, and C. Lopes, ‘Vocal Acoustic Analysis – Jitter, Shimmer and HNR Parameters’, *Procedia Technology*, vol. 9, pp. 1112–1122, 2013, doi: 10.1016/j.protcy.2013.12.124.

[101] D. R. R. Smith and R. D. Patterson, ‘The interaction of glottal-pulse rate and vocal-tract length in judgements of speaker size, sex, and age’, *J Acoust Soc Am*, vol. 118, no. 5, pp. 3177–3186, 2005, doi: 10.1121/1.2047107.

[102] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, ‘A Guide to Convolutional Neural Networks for Computer Vision’, in *Synthesis Lectures on Computer Vision*, vol. 8, no. 1, 2018, pp. 43–99. doi: 10.2200/s00822ed1v01y201712cov015.

[103] J. Lee, J. Park, K. L. Kim, and J. Nam, ‘SampleCNN: End-to-end deep convolutional neural networks using very small filters for music classification’, *Applied Sciences (Switzerland)*, vol. 8, no. 1, p. 150, 2018, doi: 10.3390/APP8010150.

[104] J. Guo, N. Xu, X. Chen, Y. Shi, K. Xu, and A. Alwan, ‘Filter sampling and combination CNN (FSC-CNN): A compact CNN model for small-footprint ASR acoustic modeling using raw waveforms’, in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2018, vol. 2018-Septe, pp. 3713–3717. doi: 10.21437/Interspeech.2018-1370.

[105] T. Kim, J. Lee, and J. Nam, ‘Comparison and Analysis of SampleCNN Architectures for Audio Classification’, *IEEE Journal on Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 285–297, 2019, doi: 10.1109/JSTSP.2019.2909479.

[106] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, ‘Squeeze-and-Excitation Networks’, *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 8, pp. 2011–2023, 2020, doi: 10.1109/TPAMI.2019.2913372.

[107] M. D. Zeiler, ‘ADADELTA: An Adaptive Learning Rate Method’, *arXiv preprint arXiv:1212.5701*, vol. abs/1212.5, 2012, [Online]. Available: http://arxiv.org/abs/1212.5701

[108] P. Zhou *et al.*, ‘Attention-based bidirectional long short-term memory networks for relation classification’, in *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Short Papers*, 2016, pp. 207–212. doi: 10.18653/v1/p16-2034.

[109] A. T. Mohan and D. v. Gaitonde, ‘A Deep Learning based Approach to Reduced Order Modeling for Turbulent Flow Control using LSTM Neural Networks’, *ArXiv*, Apr. 2018, doi: 10.48550/arxiv.1804.09269.

[110] N. A. He and S. Ferguson, ‘Multi-view Neural Networks for Raw Audio-based Music Emotion Recognition’, *Proceedings - 2020 IEEE International Symposium on Multimedia, ISM 2020*, pp. 168–172, 2020, doi: 10.1109/ISM.2020.00037.

[111] S. Rosen, ‘Temporal information in speech: acoustic, auditory and linguistic aspects.’, *Philos Trans R Soc Lond B Biol Sci*, vol. 336, no. 1278, pp. 367–373, Jun. 1992, doi: 10.1098/rstb.1992.0070.

[112] K. He, X. Zhang, S. Ren, and J. Sun, ‘Deep residual learning for image recognition’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, Jun. 2016, doi: 10.1109/CVPR.2016.90.

[113] K. He, X. Zhang, S. Ren, and J. Sun, ‘Identity mappings in deep residual networks’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9908 LNCS, pp. 630–645, 2016, doi: 10.1007/978-3-319-46493-0_38.

[114] C. Li *et al.*, ‘Deep Speaker: an End-to-End Neural Speaker Embedding System’, *ArXiv*, vol. abs/1705.0, 2017, [Online]. Available: http://arxiv.org/abs/1705.02304

[115] J. W. Jung, H. S. Heo, J. H. Kim, H. J. Shim, and H. J. Yu, ‘RawNet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification’, *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2019-Septe, pp. 1268–1272, 2019, doi: 10.21437/Interspeech.2019-1982.

[116] F. H. Guenther and D. Johnson, ‘A computational model using formant space planning of articulator movements for vowel production’, *J Acoust Soc Am*, vol. 97, no. 5, pp. 3402–3402, 1995, doi: 10.1121/1.412518.

[117] J. T. Eichhorn, R. D. Kent, D. Austin, and H. K. Vorperian, ‘Effects of Aging on Vocal Fundamental Frequency and Vowel Formants in Men and Women’, *Journal of Voice*, vol. 32, no. 5, pp. 644.e1-644.e9, Sep. 2018, doi: 10.1016/j.jvoice.2017.08.003.

[118] R. Pribuisiene, V. Uloza, and V. Kardisiene, ‘Voice characteristics of children aged between 6 and 13 years: Impact of age, gender, and vocal training’, *Logoped Phoniatr Vocol*, vol. 36, no. 4, pp. 150–155, 2011, doi: 10.3109/14015439.2011.569756.

[119] F. Geissler, J. H. Appleton, and R. C. Perera, ‘The Development and Practice of Electronic Music’, *Notes*, vol. 32, no. 4, p. 775, 1976, doi: 10.2307/897748.

[120] H. K. Vorperian and R. D. Kent, ‘Vowel acoustic space development in children: a synthesis of acoustic and anatomic data’, *J Speech Lang Hear Res*, vol. 50, no. 6, pp. 1510–1545, Dec. 2007, doi: 10.1044/1092-4388(2007/104).

[121] R. Mugitani and S. Hiroya, ‘Development of vocal tract and acoustic features in children’, *Acoust Sci Technol*, vol. 33, no. 4, pp. 215–220, 2012, doi: 10.1250/ast.33.215.

[122] M. Ota, ‘Minimality constraints and the prosodic structure of child Japanese’, 1998, pp. 331–344.

[123] J. S. Tsay, ‘Construction and Automatization of a Minnan Child Speech Corpus with some Research Findings’, *Computational Linguistics and Chinese Language Processing*, vol. 12, no. 4, pp. 411–442, 2007.

[124] F. Wijnen, ‘The development of sentence planning.’, *J Child Lang*, vol. 17, no. 3, pp. 651–675, Oct. 1990, doi: 10.1017/s030500090001093x.

[125] N. G. Garmann, P. Hansen, H. G. Simonsen, and K. E. Kristoffersen, ‘The Phonology of Children’s Early Words: Trends, Individual Variation, and Parents’ Accommodation in Child-Directed Speech’, *Front Commun (Lausanne)*, vol. 4, p. 10, 2019, doi: 10.3389/fcomm.2019.00010.

[126] Y. Rose, C. Dyck, E. Swain, and J. Brittain, ‘The Acquisition of Metrical Opacity: A Longitudinal Case Study from Northern East Cree’, *34th Annual Boston University Conference on Language Development*, vol. 2, pp. 339–350, 2010.

[127] M. Cruz-Ferreira, ‘Child multilingualism at home and in school A comment on David Deterding’s review of Three is a Crowd?’, *International Journal of Applied Linguistics*, vol. 18, pp. 110–113, Mar. 2008, doi: 10.1111/j.1473-4192.2008.00182.x.

[128] K. Demuth, ‘Acquisition of Sesotho’, in *The Cross-Linguistic Study of Language Acquisition*, vol. 3, D. Slobin, Ed. Hillsdale NJ: Lawrence Erlbaum Associates, 1992, pp. 557–638.

[129] E. Lieven, D. Salomo, and M. Tomasello, ‘Two-year-old children’s production of multiword utterances: A usage-based analysis’, in *Cognitive Linguistics*, 2009, vol. 20, no. 3, pp. 481–507. doi: 10.1515/COGL.2009.022.

[130] K. J. Piczak, ‘ESC: Dataset for environmental sound classification’, *MM 2015 - Proceedings of the 2015 ACM Multimedia Conference*, pp. 1015–1018, Oct. 2015, doi: 10.1145/2733373.2806390.

[131] J. F. Gemmeke *et al.*, ‘Audio Set: An ontology and human-labeled dataset for audio events’, in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2017, pp. 776–780. doi: 10.1109/ICASSP.2017.7952261.

[132] K. Simonyan and A. Zisserman, ‘Very deep convolutional networks for large-scale image recognition’, *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015.

[133] G. Koch, R. Zemel, and R. Salakhutdinov, ‘Siamese neural networks for one-shot image recognition’, in *ICML deep learning workshop*, 2015, vol. 2.

[134] J. BROMLEY *et al.*, ‘Signature Verification Using a “Siamese” Time Delay Neural Network’, *Intern J Pattern Recognit Artif Intell*, vol. 07, no. 04, pp. 669–688, 1993, doi: 10.1142/s0218001493000339.

[135] F. Wang, W. Liu, H. Dai, H. Liu, and J. Cheng, ‘Additive margin softmax for face verification’, *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, vol. 25, pp. 926–930, 2018.

[136] J. A. Chagas Nunes, D. Macêdo, and C. Zanchettin, ‘AM-MobileNet1D: A Portable Model for Speaker Recognition’, *Proceedings of the International Joint Conference on Neural Networks*, 2020, doi: 10.1109/IJCNN48605.2020.9207519.

[137] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, ‘Design of an Image Edge Detection Filter Using the Sobel Operator’, *IEEE J Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, 1988, doi: 10.1109/4.996.

[138] Z. C. Lipton, J. Berkowitz, and C. Elkan, ‘A Critical Review of Recurrent Neural Networks for Sequence Learning’, *arXiv preprint arXiv:1506.00019*, 2015, [Online]. Available: http://arxiv.org/abs/1506.00019

[139] S. Hochreiter and J. Schmidhuber, ‘Long Short-Term Memory’, *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[140] L. Fu, ‘Rule learning by searching on adapted nets’, in *Ninth National Conference on Artificial Intelligence*, 1991, pp. 590–595.

[141] R. Setiono and H. Liu, ‘Neurolinear: From neural networks to oblique decision rules’, *Neurocomputing*, vol. 17, no. 1, pp. 1–24, 1997, doi: 10.1016/S0925-2312(97)00038-6.

[142] S. Biswas, M. Chakraborty, B. Purkayastha, P. Roy, and D. Thounaojam, ‘Rule Extraction from Training Data Using Neural Network’, *International Journal of Artificial Intelligence Tools, World Scientific*, vol. 26, Jul. 2017, doi: 10.1142/S0218213017500063.

[143] Y. Wang, H. Su, B. Zhang, and X. Hu, ‘Interpret Neural Networks by Identifying Critical Data Routing Paths’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8906–8914, 2018, doi: 10.1109/CVPR.2018.00928.

APPENDIX A

A.1. FORMULATIONS OF EXTRACTED SPEECH FEATURES.

A.1.1. Mean pitch

The mean pitch is determined by means of acoustic periodicity detection based on an autocorrelation method. The intuition behind this is based on the idea that the most likely candidates for a sound's acoustic pitch period are determined by the position of the maximum autocorrelation function of the sound [95]. Autocorrelation is defined as:

$$r_x(\tau) \equiv \int x(t)x(t+\tau)dt.$$

Where *x(t)* denotes the time signal, and τ denotes the lag.

The global maximum for $r_x(\tau)$ is located at τ = 0, *and* if there are other global maxima, the signal is considered periodic with a lag of period T_0. Hence the fundamental frequency F0 is determined as:

$$F0 = \frac{1}{T_0},$$

If there are no other global maxima other than at $\tau = 0$, local maxima are considered such that if the highest of the maxima lags at τ_{max} and has a height of $r_x(\tau_{max}$. The signal is considered to have a periodic part with the harmonic strength R_0, which is equal to the normalized autocorrelation of the local maxima $r'_x(\tau_{max}$:

$$r'_x(\tau) = \frac{r_x(\tau)}{r_x(0)}.$$

A.1.2. Formants

The formants are extracted using Praat's Formant Path algorithm based on linear predictive coding (LPC). The mathematical formulations[25] are rather extensive but are covered extensively in these studies [91][92].

A.1.2. Average formant

The average of the estimated formant frequencies **F1**, **F2**, **F3** and **F4** [96].

$$average\ formant = \frac{1}{n}\sum_{i=1}^{n} f_i.$$

Where *n* represents the number of formations and f_i represents the *i*th formant frequency.

A.1.3. MFF

Interaction of the vocal-tract length and glottal-pulse rate [97].

$$mff = \left(\frac{1}{n}\prod_{i=1}^{n} f_i\right)^{\frac{1}{4}}$$

[25] https://www.fon.hum.uva.nl/praat/manual/Sound__To_Formant__burg____.html

A.1.4. Formant position

The average standardised formant value for the first four formants. The formants are standardised using means and standard deviations between sexes [98].

$$formant\ position = \frac{1}{n}\sum_{i=1}^{n} f_i.$$

A.1.5. Formant dispersion

$$formant\ dispersion = \frac{1}{n-1}\sum_{i=1}^{n} f_{i+1} + f_i$$

A.1.6. Vocal tract length

Estimation of the speaker's vocal tract length [99].

$$vtl = \frac{1}{n}\sum_{i=1}^{n}(2n-1)\frac{f_i}{4c}$$

A.1.7. Vocal tract length Δf

$$vtl\Delta f = \frac{c}{2\Delta f},$$

$$\Delta f = \frac{c}{2vtl}.$$

Where c represents the speed of sound.

A.1.8. Harmonic-to-noise ratio

The ratio between periodic components and non-periodic components comprising of a segment of voiced speech [100].

$$HNR = 10log\ \frac{AC_V(T)}{AC_V(0) - AC_V(T)}.$$

$AC_V(T)$ represents the peak as the index position corresponding to the period of the signal.

A.1.9. Jitter measures

Measures that characterise how frequency varies from cycle to cycle [100]. It is induced by instabilities in the vibration of the vocal folds.

$$Jitter(local, absolute) = jitta = \frac{1}{N-1}\sum_{i=1}^{N-1}|T_i - T_{i-1}|,$$

T_i represents the duration for the period, and *N* is the total number of periods. The *jitta* is the mean absolute difference between two consecutive periods.

$$Jitter(local) = jitt = \frac{jitta}{\frac{1}{N}\sum_1^N T_1}100,$$

Jitt is an extension of *jitta* divided by the average period of the signal.

$$Jitter(rap) = rap = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1}\left|T_i - \left(\frac{1}{3}\sum_{n=i-1}^{i+1} T_n\right)\right|}{\frac{1}{N}\sum_{i=1}^{N} T_i}100,$$

The *rap* jitter measure represents the average disturbance.

$$Jitter(ppq5) = ppq5 = \frac{\frac{1}{N-1}\sum_{i=2}^{N-2}\left|T_i - \left(\frac{1}{5}\sum_{n=i-2}^{i+2} T_n\right)\right|}{\frac{1}{N}\sum_{i=1}^{N} T_i}100,$$

The *ppq5* jitter measure represents the average disturbance within five periods.

A.1.10 Shimmer measures

Shimmer is a form of reverberation effect in which the reverberation process incorporates enhanced harmonics from the input signal. These shimmer measures characterise the effects of amplitude variation on the sound wave[100].

The shimmer measures are calculated similarly to the jitter measures, with the exception that the shimmer measures take into account the maximum amplitude of the speech signal.

$$Shimmer(local) = Shim = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1}|A_i - A_{i+1}|}{\frac{1}{N}\sum_{i=1}^{N} A_i}100,,$$

Shim is computed by dividing the mean absolute difference of two successive periods by the mean amplitude.

$$Shimmer(local, dB) = ShdB = \frac{1}{N}\sum_{i=1}^{N-1}\left|20\,log\left(\frac{A_{i+1}}{A_i}\right)\right|,$$

ShdB is the mean absolute difference of the difference between two successive periods expressed as the base ten logarithms.

$$Shimmer(apq3) = apq3 = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1}\left|A_i - \left(\frac{1}{3}\sum_{n=i-1}^{i+1} A_n\right)\right|}{\frac{1}{N}\sum_{i=1}^{N} A_i}100,$$

The *apq3* measurement quantifies the amplitude disturbance's quotient across three periods.

$$Shimmer(apq5) = apq5 = \frac{\frac{1}{N-1}\sum_{i=2}^{N-2}\left|A_i - \left(\frac{1}{5}\sum_{n=i-2}^{i+2} A_n\right)\right|}{\frac{1}{N}\sum_{i=1}^{N} A_i}100,$$

The apq5 parameter specifies the ratio of five-period perturbation amplitudes.

A.2. DETAILED INFORMATION ON THE CHILD SPEECH DATASET.

Table A.1 gives the number of samples extracted for each speaker across each corpus[26]. Some speakers do not have samples. In the studies that collected this data, these speakers were not included in the investigation.

Table A.1: Description of the child-centred corpora used in this study.

Dataset	**Language**	**Female** (s)	**Male** (s)	**Key Child** (s)	**Other Child** (s)	**Total accumulated duration** (hh:mm:ss)
VanDam Public 5-minute	English	1334	2000	1334	2304	01:56:12
FallsChurch	Japanese and English	2371	4000	1500	-	02:11:11
Tsay	Chinese	1209	-	2002	1281	01:14:52
Utrecht	Dutch	1335	-	1341	-	00:44:36
Garmann	Norwegian	1335	10000	1335	218	03:34:48
CCLAS	Cree	1335	856	1335	22	00:59:08
MCF	Swedish and Portuguese	1335	143	1335	-	00:46:53
Demuth	Sesotho	1400	-	1335	8200	03:02:15
Thomas	English-UK	1335	787	1335	-	00:57:37
Total accumulated duration (hh:mm:ss)		03:36:29	04:56:26	03:34:12	03:20:25	**15:27:32**

[26] The accumulated durations are given in the format hours:minutes:seconds i.e. hh:mm:ss.

APPENDIX B

This chapter will detail several key elements of deep learning relevant to this study.

B.1 CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks (CNNs) are a subset of neural networks that typically consist of convolutional, pooling and fully connected layers stacked together in various combinations [102].

The convolutional layer consists of a collection of filters/kernels that perform the convolutional operation over the input to produce a feature map. The convolution operation is done by sliding the kernel over the input with respect to its dimensions. The image below depicts the convolution operation of a 2-by-2 kernel over a 3-by-3 input image.

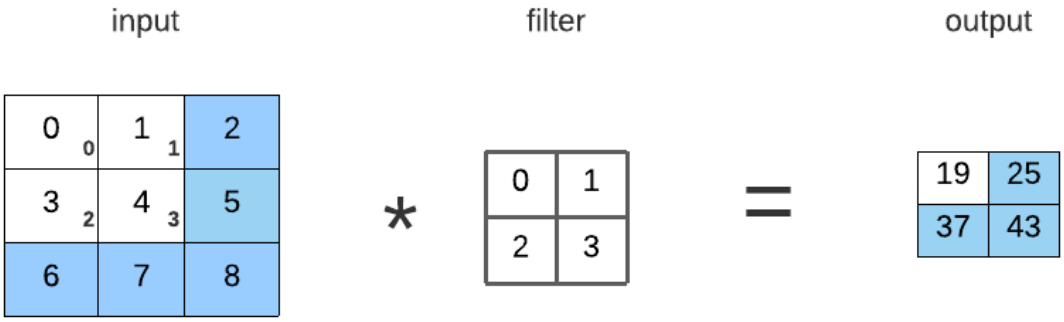

Figure B.1: The convolution operation.

The output dimensions are determined by how the convolution operation was conducted. Considering the dimension of the input to be *n*-by-*n* and the filter's dimension to be *f*-by-*f*, the dimensions of the output feature map are determined by:

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \; by \; \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

The stride represented by *s* denotes the number of pixels the filter traverses after each operation. The padding *p* represents the processing of equally adding zeros to the boundaries of the image. The padding can be specified so as to conserve the original dimensions of the input, and the resulting convolution operation is referred to as to as a same convolution. On the other hand, if padding is not specified, the resulting convolution is referred to as a valid convolution.

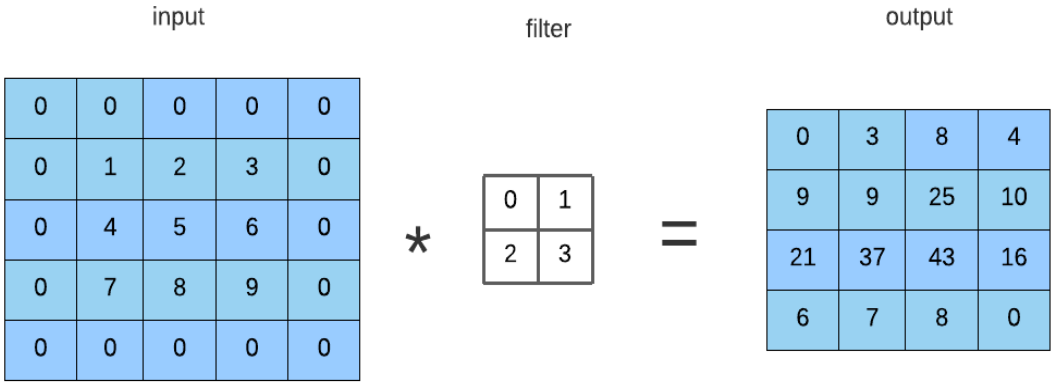

Figure B.2: A valid convolution.

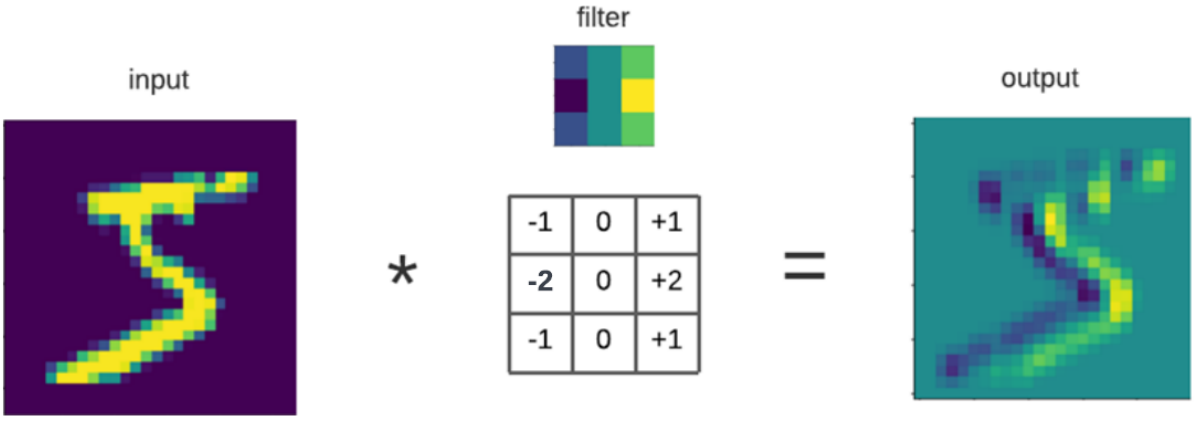

Figure B3: A valid convolution with the Sobel kernel.

The image above shows a valid convolution of a 28-by-28 image of the MNIST digit five by a 3-by-3 filter. The filter is a well-established handcrafted kernel called the Sobel filter [137]; it is designed to detect horizontally running edges within an image[27]. In a convolutional neural network setting, the values of the filters are primarily randomly initialised and then tuned during the learning process.

A pooling layer performs a downsampling operation. This is typically applied to the results of the convolutional layer resulting in a reduction in the number of parameters. There are two pooling forms: max-pooling, where the maximum of the neighbouring values is taken and average pooling, where the average values are taken instead.

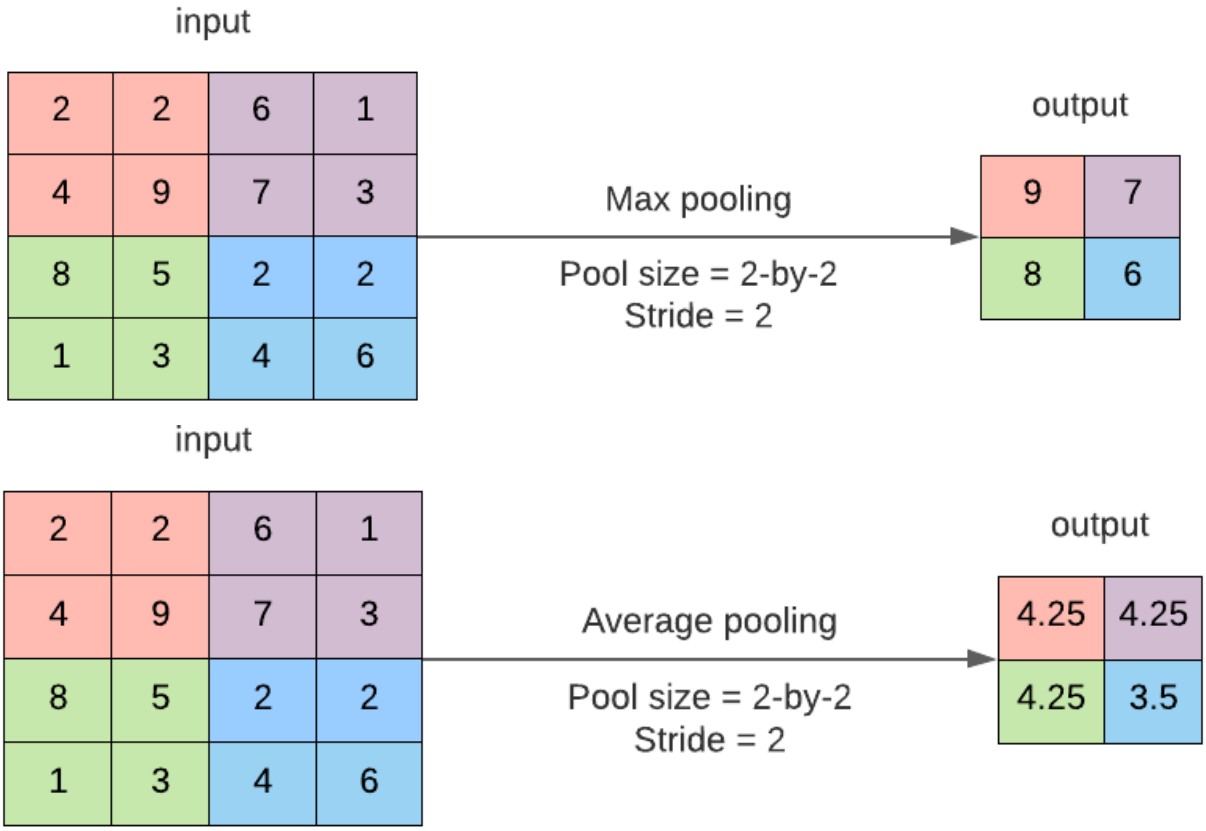

Figure B4: The difference between max pooling and average pooling.

[27] https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm

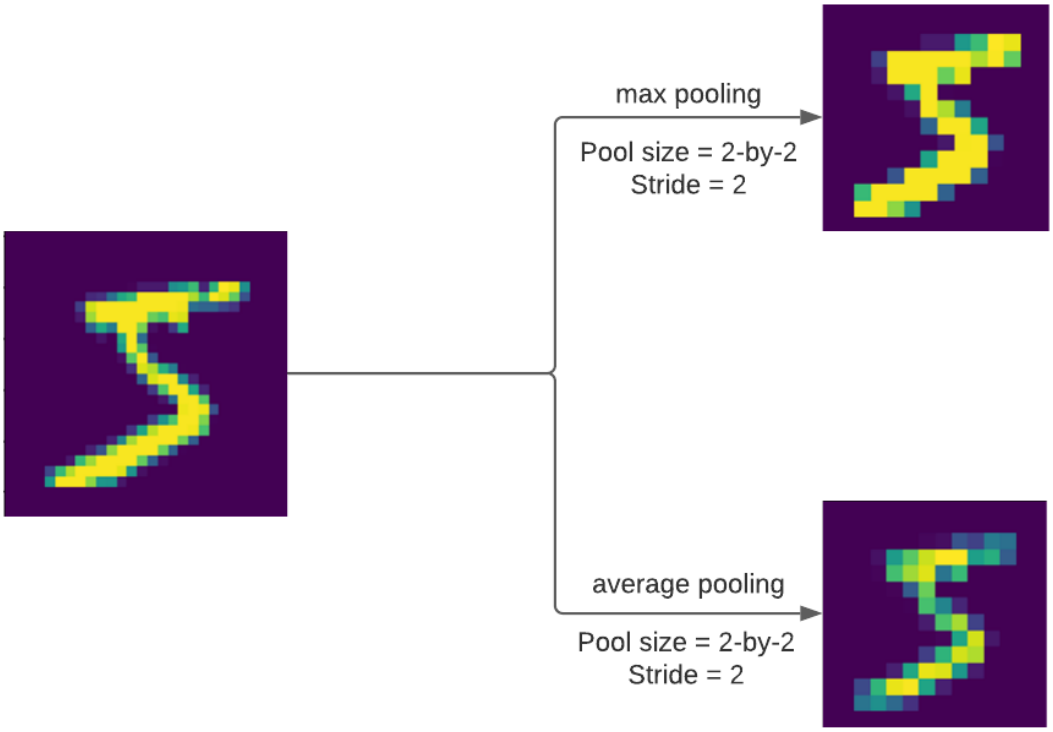

Figure B5: The difference between max and average pooling operations.

The fully connected layers are equivalent to convolutional layers with one-by-one filters. Each node in the fully connected layer is connected to every node in the preceding layer.

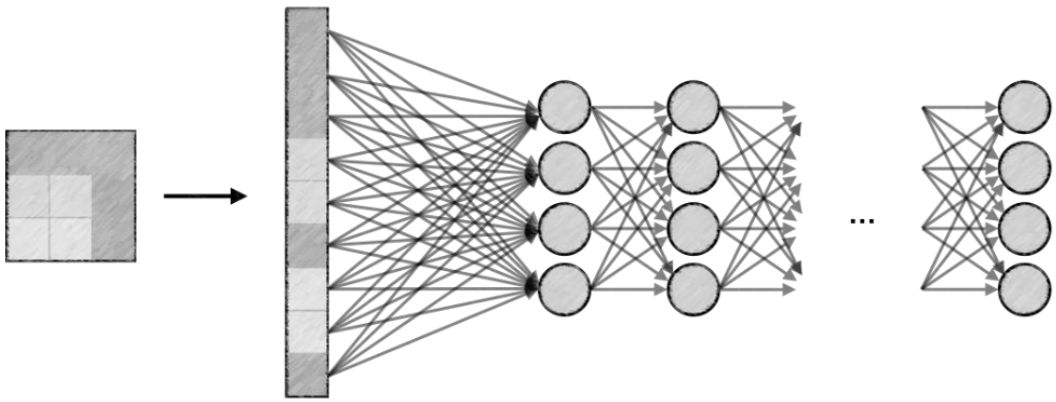

Figure B6: Fully connected layers.

B.2 BATCH NORMALISATION

Batch normalisation is a normalisation algorithm that speeds up and improves the stability of deep neural network training. This is achieved by using the mean and variance of the current mini-batch to normalise the activation vectors from the hidden layers. Normalisation can be applied right after or before the nonlinear activation depending on the application. The first step of the batch normalisation layer is to determine the mean μ and the variance σ^2 of the activation values across the mini-batch:

$$\mu = \frac{1}{n}\sum_i Z^{(i)},$$

$$\sigma^2 = \frac{1}{n}\sum_i (Z^{(i)} - \mu)^2,$$

Where n is the size of the mini-batch and $Z^{(i)}$ is the activation vector. The batch normalisation layer then normalises the activation vector by:

$$Z^{(i)}{}_{norm} = \frac{Z^{(i)} - \mu}{\sqrt{\sigma^2 - \epsilon}}.$$

This way, the output follows a typical normal distribution across the mini-batch for each neuron. The mathematical constant ϵ ensures numerical stability.

In the final step, batch normalisation determines the layer's output by applying the linear transformation:

$$\check{Z} = \gamma * Z^{(i)}{}_{norm} + \beta.$$

Where γ and β are both trainable parameters that are responsible for adjusting the standard deviation and bias, respectively, adjusting these parameters allows one to choose the optimum distribution for each hidden layer. The network computes the μ and σ corresponding to the current batch at each iteration, and the hyper-parameters γ and β are trained through gradient descent.

B.3 RECURRENT NEURAL NETWORKS

RNNs are a category of ANNs designed to process data in a sequential or temporal order. The main principle of RNNs is that they process information by iterating incrementally through it and maintaining an internal state that holds information relative to what has been processed so far [138]. The image below shows the difference between a feed-forward network and an RNN. The different nodes of the feed-forward network are collapsed into a single layer of the recurrent network. The parameters of the networks are represented by A, B and C.

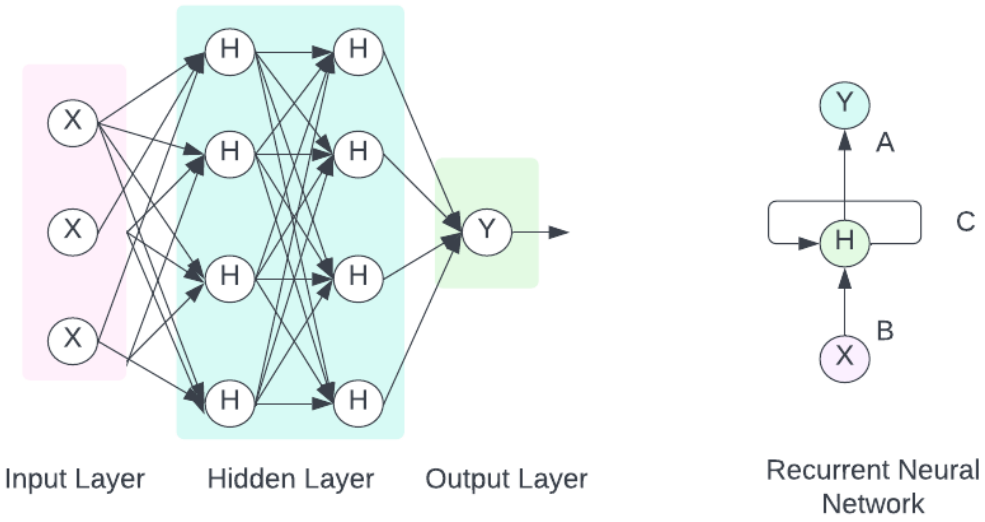

Figure B7: Fully connected network (left) and Recurrent neural network (right).

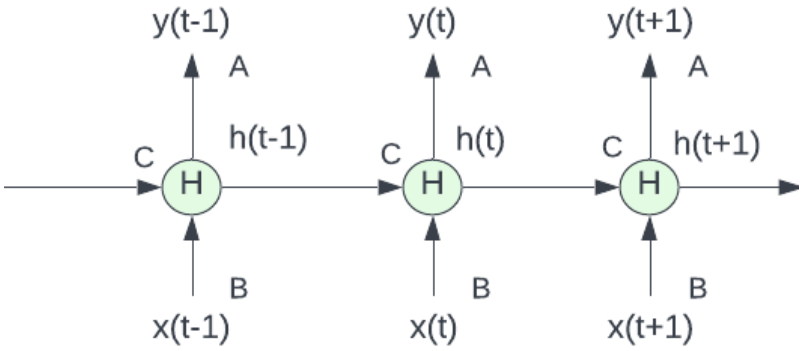

Figure B8: Unrolled recurrent neural network.

In figure B8, the RNN is depicted in its unfolded form. *x(t)* represents the input vector at time *t*, *h(t-1)* is the old state, and the new state *h(t)* is determined by:

$$h(t) = f_c\big(h(t-1), x(t)\big).$$

Where f_c is the function with parameter *c*.

B.4. LONG SHORT-TERM MEMORY NETWORKS

RNNs are susceptible to the vanishing gradients problem, which hinders them from modelling longer sequences. Gradients carry information that is required to update the RNN parameters, and as the gradient continues to decline dramatically, the accompanying parameter updates become negligible because no meaningful learning happens. The solution to the problem was the introduction of LSTM networks [139], a special type of RNN that can handle long-term dependencies. A tanh layer can be used to construct the typical RNN depicted in Figure B8.

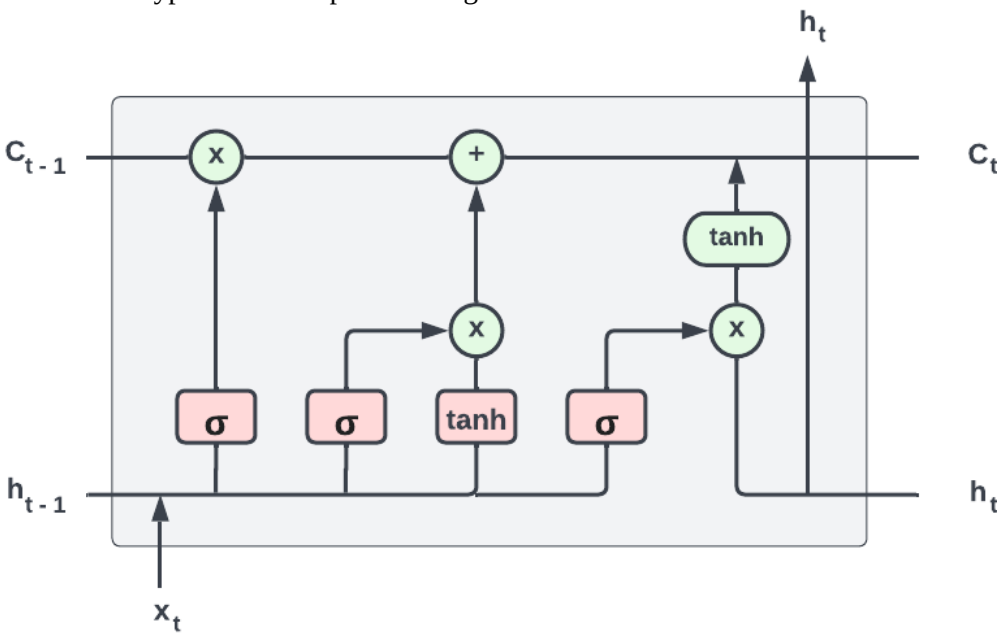

Figure B9: Long short-term memory unit

The figure above gives the depiction of an LSTM unit. An LSTM cell constitutes four interacting neural network layers compared to just one in a standard RNN. In the depiction of the LSTM cells, the rectangular elements represent neural networks, the solid lines represent vectors, and the circular ele-

ments represent pointwise operations. The interaction between the neural networks within the LSTM cells can be described in three steps.

The first step is to determine how much of the past information should be omitted for the current time step. This is determined by the sigmoid function:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big).$$

Where h_{t-1} is the previous state, x_t is the current input W_f and b_f are coefficients specific to the forget gate f_t. The figure below highlights the path that executes this operation.

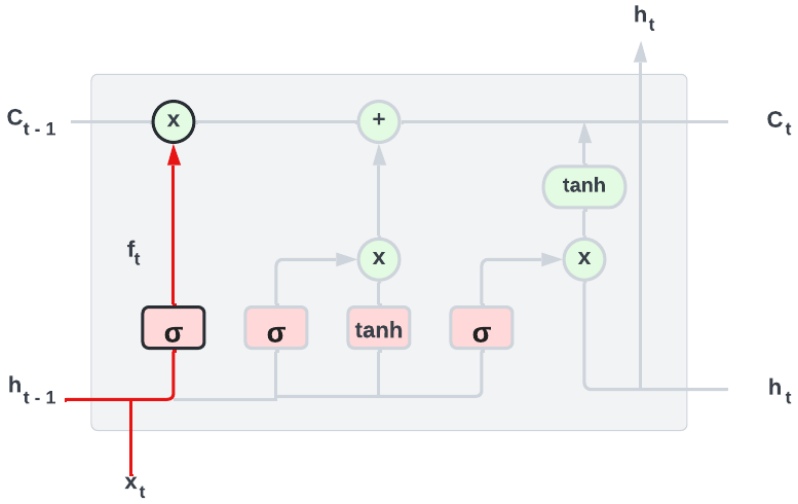

Figure B10: Illustration of the forget gate.

The second step determines how much the unit adds to the current state. Depending on the relevance of the current time step, the input gate i_t can determine which information can be added.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i),$$
$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C).$$

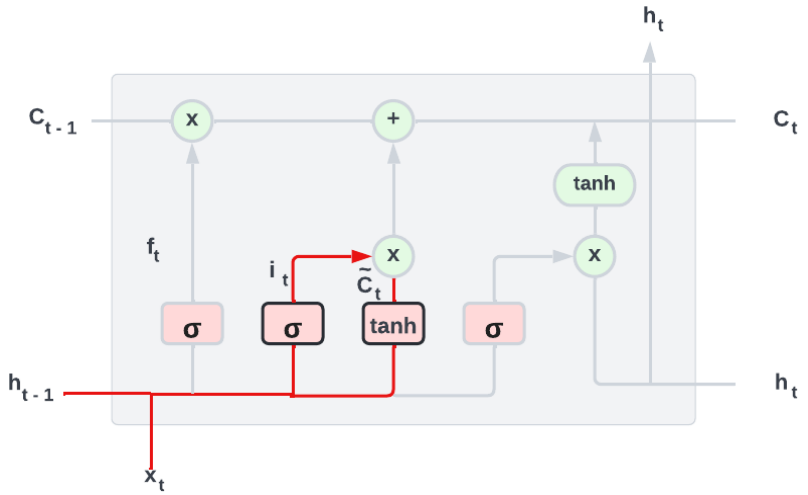

Figure B11: Illustration of the input gate.

In the final step, the output gate determines which information is allowed to impact the output in the current time step.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o),$$
$$\tilde{C}_t = o_t * tanh(C_t).$$

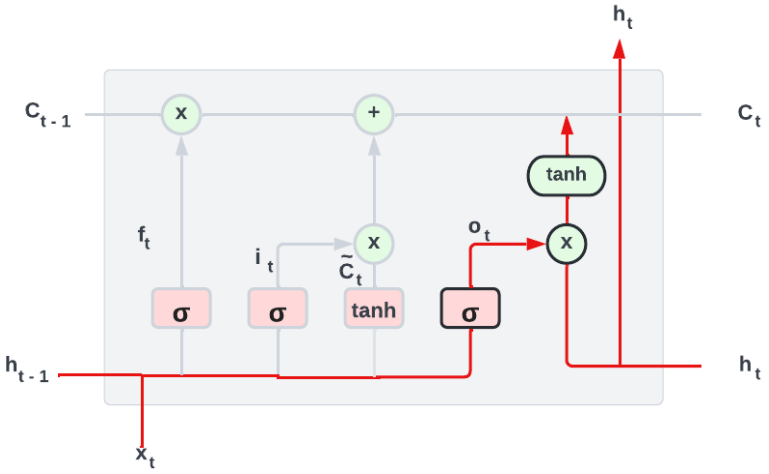

Figure B12: Illustration of the output gate.

B.5 BIDIRECTIONAL LSTM NETWORKS

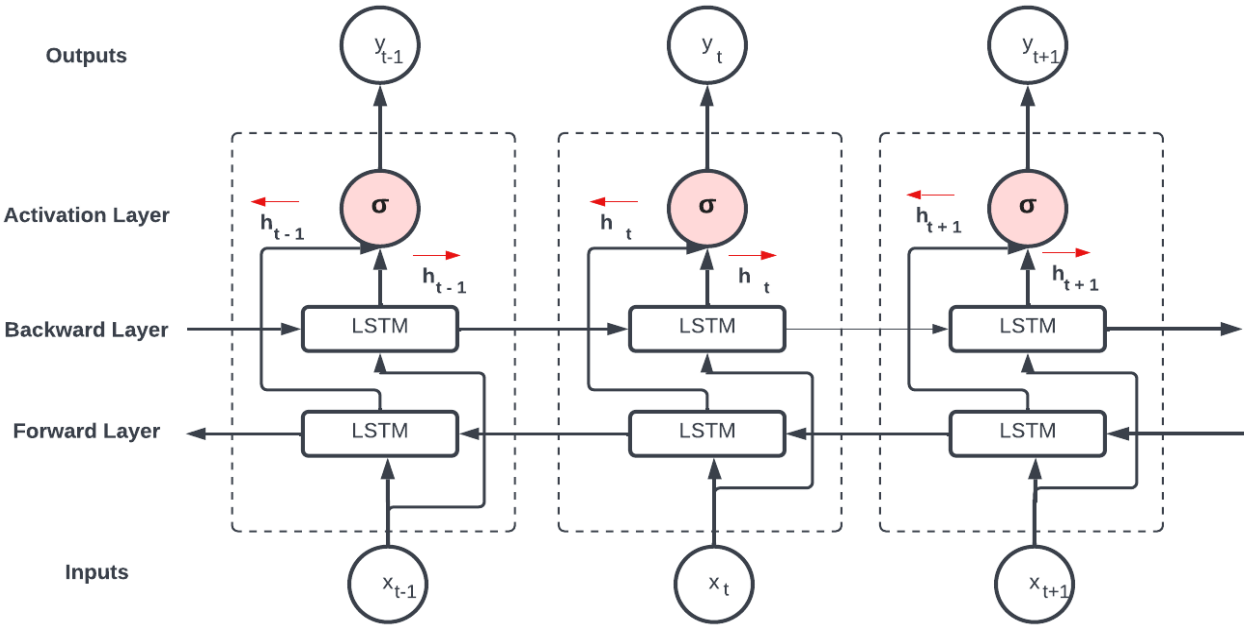

Figure B13: Illustration of the output gate.

Bidirectional LSTM networks extend the LSTM architecture allowing temporal or sequential information to flow from both directions, whereas in conventional LSTMs, the information flow is unidirectional [108]. As the name implies, the biLSTM/BiLSTM architecture is made up of two LSTMs, one of which directs the flow of information in one way and the other of which directs it in the opposite direction. This is directly beneficial to the algorithm's context as the information available to it also increases.

APPENDIX C

C.1. RULE EXTRACTION

Logic rules have since been a very attractive way to represent the functionality and behaviour of algorithms and systems. Rule extraction methods aim to learn the decision rules or a decision tree from the decisions made by a model. Consider a model that has been trained on data samples with input features x_1, x_2, x_3, and x_4 to give an output of y. A clear and straightforward way to explain how the model's decision logic works is to represent it as a set of IF-THEN rules. To say that IF x_1 and x_2 are present THEN the prediction is y_1. Since logic rules are usually explicitly interpretable themselves, rule extraction methods consequently provide explicit explanations making them very attractive to consider when exploring the interpretability of a model.

Rule extraction methods can be divided into two fundamental groups decompositional and pedagogical approaches. Decompositional approaches extract rules using network-specific attributes such as their weights and overall architecture. Pedagogical approaches treat the network as a black box which is then used to formulate the training instances for traditional rule learning methods [42][44].

C.1.1. Decompositional Approaches

Techniques of this kind extract rules by paying attention to the connections in a neural network model. Consider a simple shallow one-layer neural network:

$$y = \sigma(\sum \omega_i x_i + b)$$

Where y is the prediction, σ is the sigmoid activation function, x is the input vector, b and ω are the bias term and trainable weight matrices, respectively.

The rule extraction method will search for a combination of a range of values for the attributes x_i that maximise the prediction y. This approach is sufficient for shallow networks, but it becomes very computationally expensive when considering network architectures with a large number of layers. Instances like these require an optimised way to search for the combination of attributes.

The Knowledgetron (KT) algorithm [140] was one of the first methods that proposed an optimised way to search for the input attributes. This entails dividing the attributes into positive and negative attributes using the signs of their weights. A combination of positive attributes is only considered if it can produce the desired output prediction. Then for each combination of positive attributes, the negative attributes are defined by the combination of input attributes that did not contribute to the prediction. Assuming n attributes of x_i are Boolean in nature, and can take either absent, true or false values. The KT algorithm helps reduce the search space from $O(3^n)$ to $O(2^n)$. This is true because when considering positive attributes, they can either take true or absent values and the same is true when considering negative attributes.

NeuroRule is a predecessor to the KT algorithm[44]. This technique first prunes the network to remove all unnecessary connections between neurons. If continuous data exists, it is discretised, and then the rules are generated by an automated rule extraction technique that employs something similar to the KT algorithm. A limitation of this method is that it is restricted to discrete data. The NeuroLinear algorithm proposed in [141] adopts a technique that slightly changes the NeuroRule technique to work on continuous input directly without discretising.

C.1.2. Pedagogical Approaches

Pedagogical methods learn logic rules directly from the examples generated by a black-box model. Essentially the problem is reduced to learning a decision tree or traditional rule learning. An example of a pedagogical approach is the rule extraction by reverse engineering (RxREN) algorithm[142]. RxREN uses reverse engineering to prune the unimportant input features. The features are considered less important if their omission does not significantly change the prediction. There also exists various flavours to this method, such as the rule extraction from neural networks using classified and miss-classified data (RxNCM) algorithm [142]. This differs from the RxREN algorithm by using an inverted version of the pruning process where the features that significantly impact the accuracy of the prediction are removed instead of the reverse.

The rule extraction methods discussed so far provide global interpretability. There are other methods that provide local and semi-local interpretability. Critical data routing paths (CDRPs) introduced by Wang *et al* [143] is a rule extraction method that provides local interpretability explanations. As the name suggests, the method works by finding important data routing paths for each input through the network. The method was proposed on CNNs where on each layer, every output channel is associated with a non-negative weight that indicates how important (critical) it is. Intermediate filter weights or critical nodes associated with positive gates must first be discovered before the CDRPs can be determined. From these nodes, meaning can now be inferred that utilises the critical paths found to provide local explanations. Zhang *et al* [42] point out that the representation provided by CDRPs may not be human intelligible as it is more of just an activation pattern than an actual explanation of the model's predictions.

C.2. EXPLANATION BY EXAMPLE

Zhang *et al* [42] considered these explanations to be the least implicit among all the previously discussed categories of methods. They produce explanations by demonstrating how the model behaves using a set of instances taken from the training data. There are several different approaches for producing these kinds of explanations [42][50]:

- Adversarial examples – explanations are the results of analysing how attacking the model's input changes its prediction.
- Counterfactual explanations – these explanations are developed by measuring how much an instance from the data must change in order to change the model's prediction.
- Explanation using prototypes – this makes use of a prototype and criticism dichotomy. Where a set of data instances are chosen to represent the prototype, and all the other instances that are not sufficiently represented by the prototype become the criticisms.

These seem to be the least popular type of methods considered when exploring interpretability. This might be because these explanations seem only sensible when they can represent the instance in an understandable way. This works well for data such as images and text, but the explanation becomes less meaningful when it comes to instances with less structured features