Machine learning based approaches to automatic stuttering event detection

Machine Learning based Approaches to Automatic Stuttering Event Detection

A Doctoral Thesis

by

Abedal-kareem M Al-banna

Submitted in partial fulfilment of the requirements for the award of

Doctor of Philosophy

of

Loughborough University

Student number: B933640

Project duration: October 1, 2020 – October 1, 2023

Acknowledgements

First, I would like to thank God for giving me this opportunity after 20 years of getting my master's degree. Through my PhD journey, I was blessed with a supportive family, outstanding supervision and comfort environment at Loughborough University.

I would like to acknowledge this thesis to my late parents; your love, direction, and support throughout your life shaped my identity, and your absence is deeply felt.

My deepest thanks go to my supervisors, Professor Eran A. Edirisinghe and Dr Hui Fang; your encouragement, contributions, and guidance gave me the self-confidence to finalise this work. I remember all of your words which helped me to overcome this research challenge. In addition, I would like to extend my thank to Dr Georgina Cosma for her advice and comments on my R1 and R2 annual reports.

To my beloved wife, Haneen, thank you for everything you provided to me, not only in the PhD journey but also throughout my life. I know how difficult life was for you when I left for the UK, but I know who you are. A special thanks to my daughters Beso and Roro and my son Basel; I want each of you to learn that education is not preparation for life but life itself, so continue learning throughout your life.

My heartiest thanks go to my brother Mahmood Albanna, and sisters, Souna Albanna, Abeer Albanna, and Ola Albanna, for being with my family in Jordan and supporting them in this long journey. My most enormous thanks must be given to my brother Basel Albanna who has supported me throughout my life and taught me how to achieve my goals and missions in this life.

I would also like to thank colleagues who shared with me my best moment in this journey: Mohammad Arafa, Wael Hadi and Mohammad Musa.

Abstract

Stuttering is a speech fluency disorder affecting 1% of the global population. To provide an automatic and objective stuttering assessment tool, the subject area of **Stuttering Event Detection (SED)** is under extensive investigation in advanced speech research and applications. Despite significant progress achieved by various Machine Learning (ML) and Deep Learning (DL) models, SED directly from speech signals requires to be improved due to the heterogeneous and overlapped nature of stuttering speech. With the key focus of enhancing the state-of-the-art of research in SED the **primary goal** of this thesis is to investigate different ML / DL and feature engineering techniques for robust SED based on acoustic features that directly detect stuttering events from speech signals.

The first part of this thesis demonstrates the capabilities of different DL approaches such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) architectures and hybrid approaches, such as ConvLSTM against experimental data. Moreover, this work suggests and evaluates a novel SED model architecture that detects stuttering events directly from speech signals, using a log mel spectrogram as a sole acoustic feature and a 2D atrous convolutional network to learn spectral and temporal feature representations. While the proposed DL approach has shown promising results in SED, improving model generalisation and robustness using cross-datasets and domain adaptation, and evaluating the performance against traditional ML approaches in SED, is vital.

Therefore, the second part of this thesis investigates the impact of stuttering event representation on detection performance. This part starts by rigorously investigating the effective use of eight common ML classifiers on two publicly available large-scale datasets to automatically detect stuttering events using multiple objective metrics (prediction accuracy, recall, precision, and F1 score). In addition, this part evaluates

the performance of SED and observes the impact of applying ASR pre-trained features on each stuttering event. Moreover, different experiments evaluate the impact of three time-domain features, zero crossing rate, the auto-correlation of spectral flux onset, and fundamental frequency features, on SED performance. These experiments prove that using contextual information, pre-trained models, and time-domain features, helps improve SED performance.

Finally, the thesis proposes an attention-based multi-feature DL model for stuttering event detection using a Convolutional Block Attention Module (CBAM). A novel attention-based model is proposed to effectively learn frame-level and temporal representations by considering contextual, pitch, time-domain and auditory-based spectral features. The multi-feature fusion approach, using time-domain features (zero crossing rate, spectral flux onset strength envelope, fundamental frequency), automatic speech recognition embeddings, and auditory-based spectral features, is capable of improving SED performance significantly, outperforming state-of-the-art methods. In addition, a convolutional block with attention maps along two separate dimensions based on CBAM is introduced for SED. This contribution demonstrates the effectiveness of this lightweight module in performing automatic feature selection by assigning shared weights to the intermediate feature map and focusing on the salient features of speech regions, leading to improved performance in SED.

Contents

List of Figures

List of Tables

List of Abbreviation

Abbreviations

Abbreviation	Definition
WHO	World Health Organisation
AI	Artificial Intelligence
SLP	Speech Language Pathologist
PWS	Person Who Stutter
DS	Developmental Stuttering
NS	Neurogenic Stuttering
WM	White Matter
PS	Psychogenic Stuttering
DAF	Delay Auditory Feedback
FAF	Frequency Altered Feedback
CWS	Children Who Stutter
SSI-3	Stuttering Severity Instrument Three
SSI-4	Stuttering Severity Instrument Four
ASR	Automatic Speech Recognition
SED	Stuttering Event Detection

Abbreviation	Definition
DSM	Diagnostic and Statistical Manual of Mental Disorders
DL	Deep Learning
ML	Machine Learning
MFCC	Mel-Frequency Cepstral Coefficient
RF	Random Forest
SVM	Support Vector Machine
MLP	Multi-Layer perceptron
kNN	k-Nearest Neighbour
HMM	Hidden Markov Model
GMM	Gaussian Mixture Model
DT	Decision Tree
QDA	Quadratic Discriminant Analysis
GNB	Gaussian Naïve Bayes
ILP	Integer Linear Programming
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
CBAM	Convolutional Block Attention Module
NSF	National Science Foundation
FL	Focal Loss
TP	True Positive
TN	True Negative
FP	False Positive

Abbreviation	Definition
FN	False Negative
UAR	Unweighted Average Recall
TDNN	Time Delay Neural Network
ACF	Auto-Correlation Function
LDA	Linear Discriminant Analysis
LPCC	Linear Predictive Cepstral Coefficient
ZCR	Zero Crossing Rate
FF	Fundamental Frequency
SFO	Spectral Flux Onset Strength Envelope
ReLU	Rectified Linear Unit
STFT	Short-Time Fourier Transform
FFT	Fast Fourier transform
DSRM	Design Science Research Methodology
CB	Class-Balanced
CE	Cross-Entropy

List of publications

Journal publications and conference proceedings

1. A. -K. Al-Banna, E. Edirisinghe and H. Fang, ”Stuttering Detection Using Atrous Convolutional Neural Networks,” *2022 13^{th} International Conference on Information and Communication Systems (ICICS)*, 2022, pp. 252-256, doi: 10.1109/ICICS55353.2022.9811183.
2. A.-K. Al-Banna, E. Edirisinghe, H. Fang, and W. Hadi, “Stuttering Disfluency Detection Using Machine Learning Approaches,” Journal of Information & Knowledge Management., vol. 21, no. 02, Jun. 2022, doi: 10.1142/S0219649222500204.

Submitted papers

1. A. -K. Al-Banna, E. Edirisinghe and H. Fang, ”Multi-feature based deep attention model for stuttering events detection”, Expert Systems with Applications journal., submitted on 06, April. 2023 (Under review).

Personal profile

1

Introduction

Imagine working as an educator for university-level students and asking each student to answer a trivial and direct question orally, ”What is your name?”. In general, the educator may think that any university-level student will answer this question quickly and easily. In fact, the answer to this question is challenging for some students, e.g., a student with communication problems may need help to answer this question smoothly. Therefore, the adjective trivial that describes this question is debatable. **Communication** involves exchanging meaning, knowledge and concepts between the **transmitter** (educator) and the **recipient** (student) using a clear and well-structured code, called **language**. Any defect of one of these components may interrupt the flow of **human communication**. In human communication, language consists of two basic units phonemes and morphemes. Using these units, humans can generate words and sentences. These units can be expressed in different forms, i.e., written, **spoken** or signed.

Spoken language (speech) is commonly used in human communication to exchange thoughts, concepts and ideas between individuals and groups. The transmitter articulates a programmed mouth movement to form a sound wave sequence representing words, phrases and sentences. These sound waves are represented by acoustic energy, which the recipient will perceive and interpret. Speech production and planning is a complex process involving integration between the nervous, respiration, phonation and articulation systems (Ronald B. Gillam 2022). Any defect of this

collaboration may cause **speech disorders**. Speech disorders are classified based on the speech production process, e.g. **fluency**, articulation, and voice disorders. A fluency disorder is an abnormal interruption of the speech rate, smoothness and effort (Ronald B. Gillam 2022). One of the most common fluency disorders is called **stuttering or stammering**, stuttering is a neurodevelopmental speech disorder that affects 70 million people worldwide. World Health Organisation (WHO) describes stuttering as "speech that is characterised by frequent repetition or prolongation of sounds or syllables or words, or by frequent hesitations or pauses that disrupt the rhythmic flow of speech. It should be classified as a disorder only if its severity is such as to markedly disturb the fluency of speech."(World Health Organisation 2010).

In general, People Who Stutter (PWS) have common symptoms either in speech or behaviour. Repetition, uncontrolled pauses in the natural flow of speech (blockage), and prolongation are the main speech symptoms, while blinking eyes, head movement, foot-tapping, nose flaring, and facial grimaces are common behavioural symptoms (Hampton 2008, Riley et al. 2004); these symptoms are called physical concomitants. A PWS tries to hide disfluency using these concomitants. In speech pathology, the speech symptoms of stuttering are categorised into four groups, as listed in Table 1.1. Speech Language Pathologists (SLPs) commonly track and observe the four groups of symptoms to evaluate stuttering severity. This process is tedious and time-consuming for SLP and PWS. In evaluation sessions, the SLP exerts efforts to manually observe and analyse stuttering events from different speech contexts such as monologues, dialogues or reading. Automatic detection of stuttering events may help the SLP in stuttering evaluation sessions.

Automatic SED is a widely recognised challenge due to diverse determinants such as the **nature of stuttering speech** and the **lack of reliable training data** (Lea et al. 2021, Kourkounakis et al. 2021). Regarding the nature of speech, it is known that any speaker generally has different speech rates and vocal tracts, which may affect speech production, e.g. females have a higher pitch than males because they have a shorter vocal tract. In addition, the speech rate of the elderly is normally lower than children. Therefore different speaker characteristics such as age, gender, and accent

Table 1.1: The main categories (core behaviours) of speech symptoms in stuttering and the sub-categories of repetition, such as word and sound, are listed in the table.

Stuttering event	Example	Type
Repetition	**Th-th-this** Stuttering is not simple	Primary
Interjection	Your voice **um ah** should be heard	Secondary
Prolongation	Your **vvvvvvoice** should be heard	Primary
Block	**k(involuntary stop)** Kareem	Primary

may affect the performance of SED and its generalisation ability. Moreover, people who stutter tend to have fairly different kinds of stuttering events (Ronald B. Gillam 2022). As mentioned above, speech symptoms of stuttering are categorised into four groups, which could be heterogenic and overlapped. In repetition; for instance, PWS may produce three to four repetitions at the beginning of the word or even within words, e.g. (foot,b,b,b, ball) or (ka, ka,kareem). Detection of these variations of stuttering events is challenging and needs massive training data.

Unfortunately, **scarcity** and **reliability** of training data are the main challenges that affect the performance of the existing models. The margin of the accuracy of these models is approximately 45%, as explained in **Chapter 3**. The reliability and scarcity of training data maybe are leading causes of this margin. One of the main factors influencing data reliability is the **agreement** in the data annotation phase. The data annotation process is vital and may affect the reliability of training data. Nevertheless, the annotation of stuttering events is difficult and requires SLPs (Barrett et al. 2022), e.g., the normal annotator will not distinguish between sound repetition and block; also, it is challenging to differentiate between part-word and sound repetition. Despite the scarcity of existing datasets, the only dataset that presented the agreement on the annotation process is SEP-28k (Lea et al. 2021), recently curated by Apple. However, the block and prolongation agreement in the SEP-28k is 11% and 25%, respectively. This agreement on certain stuttering events proves the difficulty of the annotation step. UCLASS (Howell et al. 2009) is a common unlabelled dataset used in stuttering research. Most previously existing models are trained based on manually labelled data

with a limited number of speakers. Therefore, the generalisation and robustness of these models cannot be reliably concluded.

The **primary goal** of this research is to investigate different ML approaches to create a robust **SED** based on acoustic features that directly detect stuttering events from the speech signal. Model robustness means the ability of the model to detect stuttering core behaviours and the ability to generalise. This goal is backed up by evidence from the literature explained in **Chapter 3**.

1.1. Research motivation

In recent years, more efforts have been devoted to advancing speech research and applications for people with motor problems. While previous studies employed DL and ML in speech interaction applications for blind people, the elderly, and limited hand dexterity applications (Li et al. 2019)(Almutairi et al. 2022)(Lea et al. 2022). Only a few previous research attempts have investigated using ML and DL algorithms in SED. Stuttering is a neurodevelopmental speech disorder affecting 1% of the global population. Therefore, leveraging SED in different applications may impact the quality of life for 70 million PWS worldwide in different aspects. Firstly, integrating a robust SED with the current speech assistive technology may improve their understanding of stuttering speech. Secondly, SED plays a significant role in streamlining and easing the stuttering severity evaluation for Children Who Stutter (CWS), which may impact the therapy plan for CWS.

Despite the gradually growing use of voice assistants over the years, these technologies did not generalise yet to understanding stuttering speech (Mitra et al. 2021). Current assistive technologies, such as Google, Alexa, and SIRI, were initially trained on fluent speech data. Therefore, primary stuttering behaviours, such as audible block, repetitions and prolongation, will be pruned with current assistants, which could limit their performance. Thus, integrating existing Automatic Speech Recognition (ASR) with SED may enhance their generalisation ability to understand stuttering events, and PWS will benefit from these technologies. Stuttering evaluation is another crucial

application that needs an efficient SED.

Stuttering evaluation by SLP in early childhood (at preschool age), especially between three and five years old is required. It probably helps diagnose stuttering and may allow 20% of CWS to treat it before being chronic (Villegas et al. 2019). However, stuttering evaluation is not affordable for many, and it is tedious and time-consuming for SLPs and the PWS. In general, SLPs use two methods to evaluate stuttering severity, perceptual scaling and counting procedure, as explained in **Chapter 2**. In the counting procedure, the SLP tracks and manually counts the percentage of frequency of Stuttered Syllables (%SS) or the Stuttered Words (%SW) and the duration of the stuttering event. Accordingly, automatic SED may help SLPs in stuttering evaluation sessions and streamline and ease the severity evaluation process.

1.2. Research scope

Like other detection tasks, SED contains three essential steps: data preparation, feature extraction and modelling. In the **data preparation** phase, a detection model is built using a small manually annotated dataset derived from UCLASS (Howell et al. 2009), FluencyBank (Bernstein Ratner & MacWhinney 2018), SEP-28k (Lea et al. 2021), artificially generated datasets or custom datasets collected by the researchers. Howell & Sackin (1995), Howell et al. (1997), Tan et al. (2007), Hariharan et al. (2012), Chee et al. (2009*b*), Pálfy (2014), Mahesha & Vinod (2016) created detection model, based on selected samples from the UCLASS dataset. Tan et al. (2007), Kourkounakis et al. (2021) performed the detection task using an artificially generated dataset, while Lea et al. (2021), Jouaiti & Dautenhahn (2022) built the detection model using SEP-28k and FluencyBank datasets. Despite promising results of stuttering detection methods in previous research, these methods have several limitations. The first limitation is data transparency; there is no clear benchmark and official split for the datasets. Previous researchers built their detection model based on manual labelling for the UCLASS dataset; however, the number of observations on each stuttering class and the agreement in the annotation process have not been determined (Barrett et al. 2022). The dataset size and the agreement in the annotation process may affect the

performance of the detection model. Moreover, it may change the nature of speech data used in training, which may influence the SED generalisation and robustness.

In addition, annotation of stuttering speech is challenging and needs SLP expertise. The non-expert annotator will not distinguish between sound repetition and block; also, it is challenging to differentiate between part-word and sound repetition. The second limitation is excluding stuttering core behaviours and fluent class from the detection task. For example, Howell & Sackin (1995), Howell et al. (1997), Tan et al. (2007), Hariharan et al. (2012), Chee et al. (2009*b*), Pálfy (2014), Mahesha & Vinod (2016), Kourkounakis et al. (2020, 2021), Jouaiti & Dautenhahn (2022) excluded block or fluent classes, while Sheikh et al. (2021), Chee et al. (2009*b*) excluded interjections. Fluent class is vital in detection since most speech, even with stuttering, is fluent. Excluding stuttering core behaviours may influence the SED generalisation and robustness. Given the data preparation obstacles, this thesis will answer the following question.

RQ-1 *To which level can a robust stuttering event detection model be created, based on perceived acoustic features as a sole model input, given a limited number of reliable stuttering samples and observations?*

Previous research employed different **feature extraction** and acoustic processing techniques in SED. Most of the research works focus on acoustic analysis and parametric and nonparametric feature extraction (Fook et al. 2013). Typically in feature extraction, the waveform signal is converted into a form of parametric representation that is more discriminative and reliable than the actual signal. The parametric representation for each speech signal can be represented by a feature vector used in the detection task. Chee et al. (2009*b*), Ravikumar et al. (2009), Tan et al. (2007), Hariharan et al. (2012), Pálfy (2014), Mahesha & Vinod (2016), Sheikh et al. (2021) used Mel-Frequency Cepstral Coefficient (MFCC) as a sole acoustic feature in SED . Tan et al. (2007), Ravikumar et al. (2009), Tan et al. (2007), Hariharan et al. (2012), Pálfy (2014), Mahesha & Vinod (2016) utilised time domain features such as Auto-Correlation Function (ACF), envelope parameters, duration energy peaks and fundamental frequency to detect specific stuttering events such as prolongation. Mohapatra

et al. (2022) employed a pre-trained ASR, namely WAV2VEC2, to increase the performance of the SED. Limited research exists in the literature that employed multi-feature fusion in the SED. Lea et al. (2021) used frequency domain, time domain features and pitch articulatory features in the detection task. Jouaiti & Dautenhahn (2022) combined MFCC and phoneme classes and probabilities to detect four stuttering events. However, the impact of these features on each stuttering event is not considered. In addition, combining other time-domain features and pre-trained ASR features may enhance the performance of SED. Accordingly, this research will answer the following question.

RQ-2 *What is the impact of stuttering event representation on detection performance?*

Different ML and DL approaches have been proposed in the literature for stuttering detection. Previous research employed ML algorithms such as Support Vector Machine (SVM), Multi-Layer perceptron (MLP), k-Nearest Neighbour (kNN), Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM) to enhance the performance of SED. In addition, DL approaches such as CNN, RNN, and sequence-to-sequence methods are commonly used in stuttering detection models. The authors in the literature provide single model architecture with unique and sole acoustic features such as MFCC, mel spectrogram or other time domain features. However, employing multi-feature such as frequency or time domain features with attention architecture may enhance the performance of the SED. Consequently, the following research question will be answered in this research.

RQ-3 *To which level can an attention-based, multi-feature fusion model architecture enhance the performance of SED?*

1.3. Research contributions

The research work in this thesis has resulted in several original contributions to the stuttering detection domain. The key contributions of this research are summarised in the following subsections.

1.3.1. Stuttering detection using an atrous CNN

This contribution proposes a new model for detecting stuttering events that help SLPs evaluate stuttering severity. The model is based on a log mel spectrogram and a 2D atrous convolutional network designed to learn spectral and temporal features. The model performance was rigorously evaluated on 47 (18 female, 29 male) speakers of monologue speech from the UCLASS dataset, annotated by two SLPs, and the FluencyBank dataset. The experimental results indicate that the model outperforms state-of-the-art Bi-LSTM and ConvLSTM methods in prolongation with an F1 score of 52% and 44.5% on the UCLASS and FluencyBank datasets, respectively. In addition, the model gains 5% and 3% margins on the UCLASS and FluencyBank datasets for the fluent class.*This contribution was published in The IEEE International Conference on Information and Communication Systems (ICICS 2022)* (Al-Banna, Edirisinghe & Fang 2022).

1.3.2. An investigation of ML models for stuttering event detection and of different feature representations

This contribution investigates and evaluates eight well-known ML classifiers on two recently published datasets (FluencyBank and SEP-28k) to automatically detect stuttering events using prediction accuracy, recall, precision, and F1 score. The experimental results show that a Random Forest (RF) classifier achieves the best performance, with an F1 score of 42% and 34% on the FluencyBank and SEP-28k datasets, respectively. In addition, experiments were conducted to evaluate the impact of the frequency domain, time domain and pre-trained ASR features on SED performance. The experiments evaluated the effect of three time-domain features, the zero crossing rate, the auto-correlation of spectral flux onset and the fundamental frequency. Further experiments were executed to measure the impact of a pre-trained ASR model on the performance of SED. The findings of this contribution indicate that the performance of SED increases approximately by 9 % by using these features. *Part of this contribution was published in the Journal of Information & Knowledge Management*(Al-Banna,

Edirisinghe, Fang & Hadi 2022).

1.3.3. An attention-based multi-feature DL for stuttering event detection using CBAM

A novel attention-based model is proposed to effectively learn frame-level and temporal representations by considering contextual, pitch, time-domain and auditory-based spectral features. A multi-feature fusion approach, using time-domain features (ZCR, SFO, FF), ASR embeddings, and auditory-based spectral features, is capable of improving SED performance significantly, which outperforms state-of-the-art methods. In addition, a convolutional block with attention maps along two separate dimensions based on CBAM (Sanghyun Woo, Jongchan Park, Joon-Young Lee 2018) was introduced for SED. This contribution demonstrates the effectiveness of this lightweight module in performing automatic feature selection by assigning shared weights to the intermediate feature map and focusing on the salient features of speech regions, leading to improved performance in SED.

1.4. Outline of the thesis

This section summarises the chapter outline in this thesis.

Chapter 2 starts by providing the reader with an overview of stuttering as a fluency disorder. This overview includes aspects such as the types, symptoms, causes, treatment and evaluation of stuttering. This chapter also focuses on the nature of sound and the bio-acoustic representation of human speech. Moreover, it reviews existing stuttering datasets (UCLASS, FlunecyBank and SEP-28k). Furthermore, this chapter explains the SED related tasks, handling the imbalanced nature of stuttering datasets, experimental setup and evaluation metrics.

Chapter 3 begins with an introduction, highlighting the importance of SED. In addition, it provides a comprehensive literature review and a background study on stuttering event detection using traditional ML, DL and multi-feature approaches. The

conclusion section summarises the main points covered in the literature review and establishes a connection to the subsequent chapters.

Chapte 4 explains the data preparation phase using the time interval annotation technique. The chapter explores the data segmentation approach and presents data preparation and annotation details. In addition, it provides an overview of the annotation tool that was developed and used in the annotation process. Finally, the chapter comprises the annotation procedure and agreement measurement.

Chapter 5 focuses on research question **RQ-1** by demonstrating the capabilities of convolutional, recurrent architectures and hybrid approaches, i.e. ConvLSTM, in detecting stuttering events directly from the speech signal using mel-scale features. Moreover, this chapter suggests and evaluates a novel SED model architecture that detects the stuttering events directly from the speech signal. The model uses a log mel spectrogram as a sole acoustic feature and a 2D atrous convolutional network to learn spectral and temporal features representation.

Chapter 6 investigates the impact of stuttering event representation on detection performance and answers research question **RQ-2**. The chapter starts by investigating and evaluating eight traditional machine learning classifiers against two published large-scale datasets. Moreover, it evaluates the performance of SED and shows the impact of employing ASR pre-trained feature extractions on each stuttering event. Furthermore, the chapter shows different groups of experiments that evaluate the impact of three acoustic features, the zero crossing rate, the auto-correlation of spectral flux onset, and fundamental frequency features, on SED performance.

Chapter 7 proposes to utilise multiple acoustic features extracted based on different pitch, time-domain, frequency domain, and automatic speech recognition feature to detect stuttering core behaviours more accurately and reliably. In addition, both spatial and temporal attention mechanisms are exploited, as well as Bi-LSTM modules, to learn better representations to improve the SED performance. The chapter

suggests a novel attention-based model to effectively learn frame-level and temporal representations by considering contextual, pitch, time-domain and auditory-based spectral features. In addition, this chapter provides an answer to **RQ-3**.

Chapter 8 summarises research findings, makes overall conclusions, and proposes future work.

2

Background of Study

2.1. Introduction

This chapter presents an overview of stuttering as a fluency disorder, including aspects such as the types, symptoms, causes, treatment and evaluation of stuttering. This chapter also focuses on the nature of sound and the bio-acoustic representation of human speech. Moreover, it explores existing stuttering datasets, namely UCLASS, FluencyBank and SEP-28k. Furthermore, this chapter explains SED related tasks, handling the imbalanced nature of stuttering datasets, experimental setup and evaluation metrics.

2.2. Stuttering overview

Stuttering or stammering is a common neurogenic, psychogenic fluency disorder wherein people have uncontrolled blocks on the natural flow of speech (Ronald B. Gillam 2022). It may include physical concomitants, e.g. facial grimaces, head movement, and movement of extremities (Riley et al. 2004). In addition, stuttering usually happens at preschool age, but it may also occur at ages between seven and thirteen in some cases. Like other speech disorders, stuttering is classified into three types, neurogenic, developmental and psychogenic where each of these types has different causes will be explained in this section. Speech pathology research demonstrates different treatment approaches for stuttering; after an extensive assessment session by

an SLP, a decision of the best treatment plan can be decided using these assessment approaches. Therefore, the SLP provides PWS with appropriate techniques to enhance fluency and develop communication skills. This section provides an overview of stuttering and explores the types, symptoms, causes, treatment techniques and stuttering assessment approach.

2.2.1. Stuttering types

Previous studies classify stuttering into three types: developmental, neurogenic and psychogenic, and these types may have the same symptoms and impact on a PWS. Common signs of stuttering are repetitions (word, phrase, syllable, sound), audible blocks and prolongation (Iimura & Miyamoto 2020). Riley et al. (2004) state that severe stuttering may have physical concomitants such as eyes blinking, head movement, nose flaring and grimacing. Developmental Stuttering (DS) is a widespread stuttering type; it is called developmental because it occurs in early childhood, especially between two and eight years old, which is the period of language and speech development for children. This type is a temporary disorder, and 95% of DS cases may cure without treatment (Yairi & Ambrose 2013). Nevertheless, only developmental stuttering can be cured without treatment, especially in the early stage, while the other types are considered chronic disorders and may need treatment. Moreover, speaking in front of people and talking on mobile phones tend to worsen stuttering while singing, reading loudly, and talking alone reduce disfluency. Iimura & Miyamoto (2020) state that the leading cause of this type is brain anatomical and functional deformities.

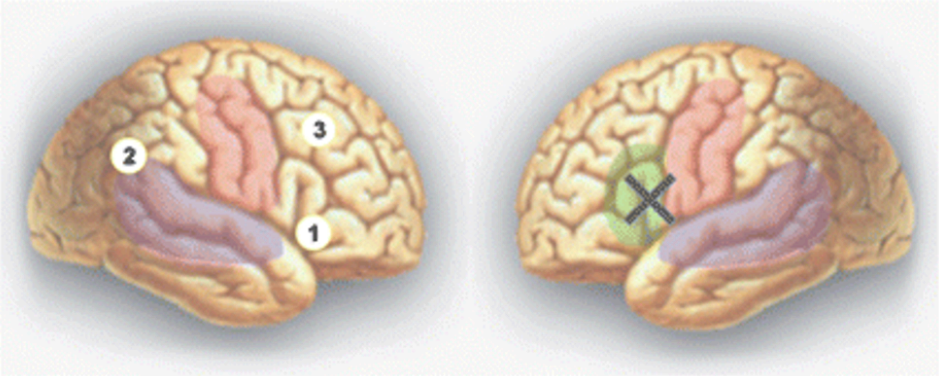
In general, people who have Neurogenic Stuttering (NS) have had a history of normal speech production before the injury or disease that causes NS, and most of these injuries affect the central nervous system. In a few cases, the developmental stuttering may convert to NS if it is not treated in the early stage. NS can be described as a stuttering type connected with functional defects in transferring auditory information into speech motor commands (Cai et al. 2012). On the other hand, Cieslak et al. (2015) demonstrated that NS is caused by genetic determinants that affect brain func-

tions, and their empirical study stated a relationship between NS and speech-motor White Matter (WM). Moreover, NS typically appears following some injury or disease to the central nervous system, e.g. the brain and spinal cord, including the cortex, subcortex, and cerebellar.

Psychogenic Stuttering (PS) is a type of stuttering caused by psychogenic factors (Ronald B. Gillam 2022). Several factors play a significant role in PS; some are related to family, and some are related to school. For example, most parents are anxious about their children's language development and are highly interested in this issue. Thus, this anxiety will affect child psychology and may cause disfluency. Few studies in the literature distinguish between PS and other stuttering kinds, especially NS. In contrast, the main difference between developmental and psychogenic stuttering is that the PS has sudden onset throughout utterances rather than at the initiation of utterances as in DS.

2.2.2. Causes of stuttering

One of the main functions of the brain is speech generation and production. Accordingly, the left inferior frontal cortex of the brain is responsible for speech planning and executive control of speech. This part of the brain, as illustrated in Figure 2.1, may have structural anomalies for PWS (Kell et al. 2009). On the other hand, the bilateral superior temporal cortex and bilateral articulatory motor cortex are responsible for phonology and auditory feedback. Cai et al. (2012) state that the functional defects in transferring auditory feedback to speech motor command may cause stuttering. Iimura & Miyamoto (2020), Hampton (2008) consider that the preeminent cause of DS and NS includes auditory feedback dysfunction. However, Al-Nafjan et al. (2018) state that developmental stuttering has no clear neurological origin. Cieslak et al. (2015) find that genetic determinants may affect neurological functions. The empirical study of Cieslak et al. (2015) showed the relationship between WM and speech fluency. Therefore, any defect in inverse models responsible for motor command may cause stuttering. Consequently, techniques such as Delay Auditory Feedback (DAF) or Frequency Altered Feedback (FAF) might stimulate this brain area and enhance fluency.



(a)

Figure 2.1: Schematic overview of speech production in fluent speakers and stutterers (Kell et al. 2009)

2.2.3. Stuttering treatment

Early stuttering treatment is essential, especially in the preschool stage, because the school environment may increase stuttering for Children Who Stutter (CWS), which may change from developmental to chronic. Despite the role of medication in treating neurogenic and psychogenic disorders, no medicine has yet been proven to solve the stuttering problem. Thus, there is no real cure for NS and PS, and the methods in the literature can enhance fluency but not avoid stuttering completely. Different approaches for stuttering therapy and fluency enhancement have been demonstrated in the literature. These methods include indirect or direct therapy. In addition, some devices (assistive devices) and mobile applications may improve fluency.

Indirect therapy is proposed based on the assumption that there is a correlation between a CWS stuttering and the CWS environment. Therefore, the fluency of CWS can be improved by changing and modifying this environment (Conture et al. 2007). Parents are the key players in this method. The SLP helps parents and provides them guidelines to adjust their communication styles with CWS. These guidelines are related to the diagnostic theory of stuttering, which states that stuttering results from parental attention to the child's disfluencies (Ratner & Tetnowski 2014). For instance, parents should pay attention to the child speaking, and when the parents talk to the child, the conversation should be slow and clear; also, avoid talking to the CWS in a

way that exceeds their ability.

On the other hand, the direct approach focuses on modifying behaviour, social responses, emotions and disfluencies of CWS. Additionally, this modification may enhance the fluency and social behaviour of CWS. Moreover, Cooper & Cooper (1985) state that direct treatment may not be proper for preschool-age children. However, in some cases, the SLP begins with short-term indirect treatment, and if necessary, the treatment approach will proceed to direct treatment (Sidavi & Fabus 2010). Sidavi & Fabus (2010) state that this method can be executed by SLP or parents when the CWS are conscious or disappointed by their stuttering. Furthermore, Van Riper (1973) suggested a direct treatment method suitable for children and adults who stutter. This method consists of six phases: Motivation, identification, desensitization, variation, approximation and stabilization. This method aims to encourage PWS by providing them with reassuring information about their fluency and showing some cases that have been treated. Further, the SLP attempts to build a comfortable environment to make PWS share feelings and emotions about the disorder.

2.2.4. Stuttering severity assessment

An SLP usually uses assessment methods to measure the severity level of stuttering. There are two methods for stuttering evaluation. Sidavi & Fabus (2010) classified the evaluation methods into two main categories: perceptual scaling and counting procedures. In perceptual scaling, the severity is rated based on a predefined scale, such as mild, severe, or low. While in the counting procedure, the frequency and duration of the stuttering event are computed. Frequency is a standard counting method. In this method, the SLP observes the percentage of Stuttered Syllables (%SS) or the percentage of Stuttered Words (%SW). Van Riper (1982) claimed that the %SS is more valid than %SW because the disfluency could happen in several locations in the same word. The duration is essential for some stuttering events, such as prolongation and audible blocks. It is estimated, aside from frequency count, to improve the reliability and validity of the evaluation. For example, in the Stuttering Severity Instrument Three (SSI-3) (Riley et al. 2004), the average of the longest three blocks and the frequency

determines the severity level.

Stuttering severity evaluation is a debatable issue in speech therapy, and there is still disagreement between SLPs on assessment procedures. The main problem in severity assessment methods is variability, wherein the severity of stuttering may increase or decrease according to different situations. For example, stuttering may appear only when a person speaks in front of a group of people, while in other places such as at home, the stuttering will disappear. Additionally, stuttering evaluation is a time and effort-consuming process, and it is a complex and inconsistent issue even if a part of speech is measured (Sidavi & Fabus 2010). Despite the difficulty in severity assessment, this assessment is important in stuttering treatment and may affect the therapy plan. Therefore, the reliability and validity of assessment methods should be considered.

The reliability of severity evaluation plays a significant role in stuttering therapy, and refers to the level of assessment agreement between SLPs for the same stuttering case. Unfortunately, different approaches to stuttering severity evaluation often give conflicting results. One source of contrasting results may be the disagreement on disfluency events, e.g.,Yairi & Ambrose (2013) state that the whole-word repetition is one of the disfluency events, while Cook & Howell (n.d.) find that there is no need for this event. Howell & Davis (2011) conducted a study to create a model that predicts whether stuttering in children will continue to persist or recover by teenage age; the severity measures were performed with the exclusion of whole-word repetitions on the predicted groups.

Despite the disagreement between severity assessment approaches, all methods classify stuttering events into seven categories: sound repetition, part-word repetition, word repetition, phrase repetition, interjection, prolongation, and block. SLPs track and observe these events before or after a treatment session. Moreover, some methods, such as (G Riley 2009), have a scoring system to help the pathologist in the evaluation process.

SSI-3 (Riley et al. 2004) is a standard evaluation of stuttering severity, and it is used in most languages (Bakhtiar et al. 2010). This evaluation is usually used to dif-

Table 2.1: Common stuttering events

Event	Description	Example
Sound Repetition	Duplication of a phoneme	th-th-the
Whole Word Repetition	Duplication of a word	kareem kareem
Part-Word Repetition	Duplication of a word syllable	My sch-school
Phrase Repetition	The duplication of a whole phrase	I need I need your help
Interjection	Adding extra sounds	um, uh
Prolongation	long time word pronunciation	Hoooow are you
Block	Uncontrolled pause and interruptions of speech	I need (stop) you

ferentiate mild stuttering cases from normal ones (Arnold et al. 2005) and track the enhancement of speech fluency for PWS in the treatment journey (Miller & Guitar 2009). Moreover, this evaluation's main parameters are the Syllable Frequency (%SS), the duration of the three most prolonged stuttering events and the physical concomitant's total score (Tahmasebi et al. 2018). This assessment's reliability has been proved in the literature (Hall et al. 1987, Riley 1991, Lewis 1995). However, some reports had some concerns about this evaluation (Lewis 1995). As a result, a new version, SSI-4 (G Riley 2009), has been developed. The naturalness of speech is the main concern about SSI-3 solved in SSI-4 and SSI-4 overcomes the reliability issues of SSI-3. However, both evaluations use the same procedures in computing frequency and the duration of the disfluency. Table 2.2 shows the differences between the two versions.

Table 2.2: SSI evaluation parameters matrix

Evalution parameters	SSI-3	SSI-4
Syllable frequency(%SS)	Yes	Yes
Duration	Yes	Yes
Physical concomitants	Yes	Yes
Naturalness of speech	No	Yes

2.3. Stuttering speech representation

Stuttering speech is similar to normal speech, produced due to the air pressure variation and vocal tract resonance. A human articulates a programmed mouth movement to form a sound wave sequence representing words, phrases and sentences. These sound waves are represented by acoustic energy. Air molecules will be pushed and pressured by this energy. A speech wave can be represented as a waveform (time-domain) by taking samples of the pressured air molecules over time; the number of samples in speech detection is usually between $8khz$ and $16khz$ per second (sampling rate). The time domain signal shows the intensity of the pressure variation (amplitude) over time, where the number of wave cycles in the signal is called frequency, measured in Hertz. As depicted in Figure 2.2, a stuttering event may follow an inconsistent periodic pattern. It may include different frequencies in the same speech signal due to diverse determinants such as speech rate based on gender, speaker and vocal tracts for children and adults. The time-domain signal is insufficient for the detection and recognition tasks because it carries multiple acoustic features. For example, a one-second speech signal may have 44 thousand samples. However, extracting temporal features from the time domain speech signal may be suitable for speech recognition and detection tasks.

Although different frequencies are presented in the signal, the time-domain signal presents only amplitude against time. The frequency domain (spectrum) representation of speech signals may contain more information reflecting the speech's nature.

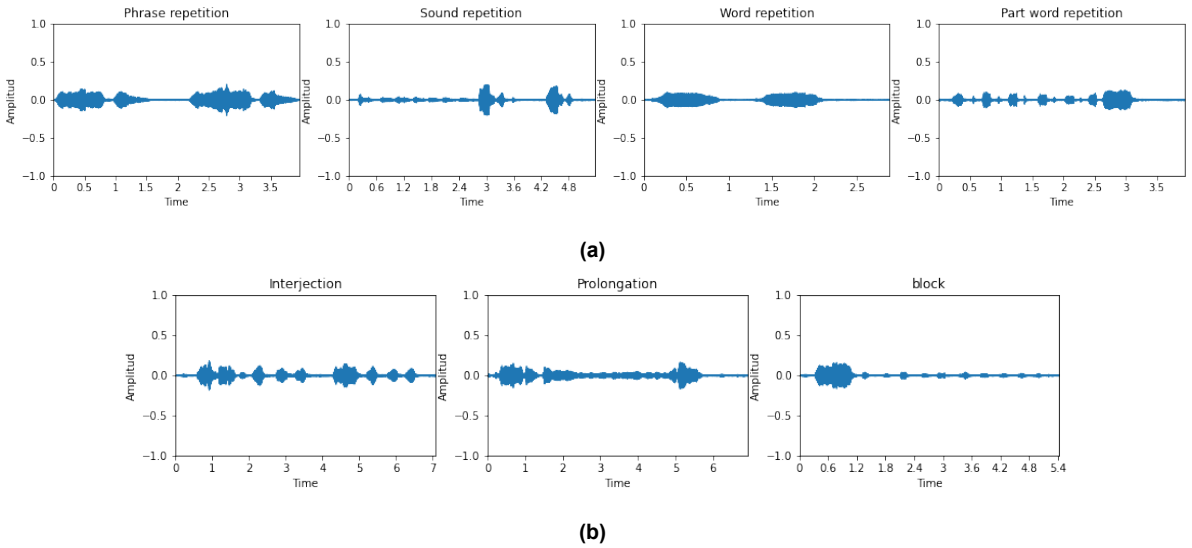


Figure 2.2: Time domain representation

In signal processing, the Fourier transform (Bracewell & Bracewell 1986) is applied to convert the signal from the time domain to the frequency domain, as shown in Figure 2.3. It decomposes the signal into frequency bins and the amplitude corresponding to that frequency. When the time-domain signal is converted to the frequency-domain, the time information is lost, and this information is vital in stuttering event detection models. Thus, another representation that maintains time and frequency information should be considered.

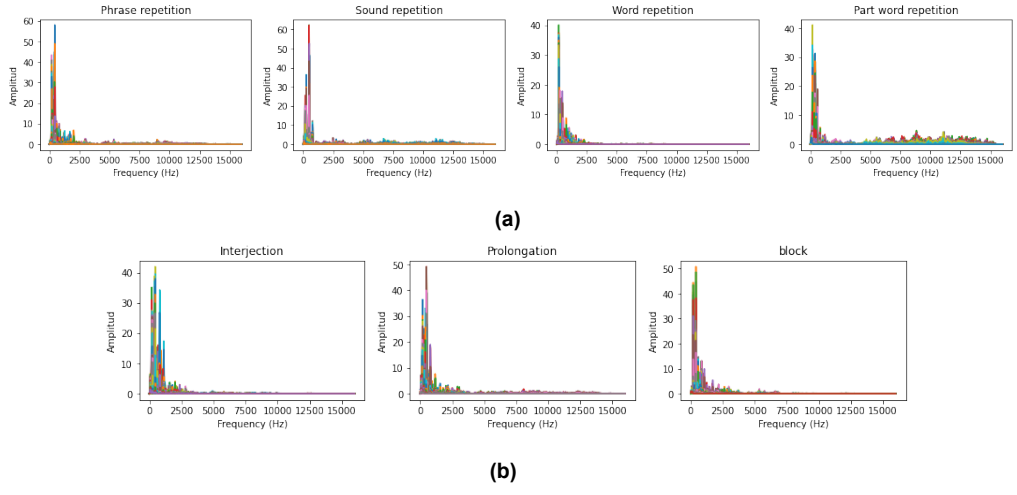


Figure 2.3: Frequency domain representation

2.3.1. Short-Time Fourier Transform (STFT)

A time-frequency method, the Short-Time Fourier Transform (STFT) applies to non-stationary signals to generate a spectrogram. A spectrogram illustrated in Figure 2.5 is a visual representation of the frequency versus the time it maintains, the time and frequency of the signal in a single plot. In a spectrogram, the signal's energy is illustrated by different colours; the darker the colour, the lower the energy. In the STFT, as illustrated in Figure 2.4, the speech signal is separated into several overlapped frames by multiplying the signal with a window function and applying the Fast Fourier transform (FFT) to each frame. The signal $\gamma(t)$ is divided into a short time frame signal with an n ms frame size and an s ms stride. In speech processing, the frame size is usually between $20\ ms$ and $40\ ms$ with 50% ($+/-10\%$) overlap between consecutive frames (Fayek 2016). After framing, window functions such as hamming apply to each frame. Then the STFT is computed as

$$P_s = \frac{|FFT(x_i)|^2}{N}$$

where x_i is the i^th frame of signal x, and N is the length of the windowed signal after padding with zeros usually = 512.

2.3.2. Mel spectrogram

As mentioned, sound waves are represented by acoustic energy, which a human perceives and interprets. The ear perceives pitch (frequency) in a non-linear way. A mel scale is a logarithmic transformation of the signal frequency that emulates how the human ear perceives frequency. In most cases, a human can distinguish lower frequencies, e.g., $200\ Hz$ from $300\ Hz$, but it is not easy to differentiate higher frequencies, e.g.,$2000\ Hz$ from $2100\ Hz$. Stevens et al. (1937) proposed a perceptual scaling illustrated in Figure 2.6 that shows this relation. The figure shows that if the distance between two sound intervals is the same, the perception of that sound is different. The STFT signal can be converted to a mel spectrogram as illustrated in Figure 2.5 by converting the frequency to mel scale as

$$m = 2595\text{log}_{10}(1 + \frac{f}{700})$$

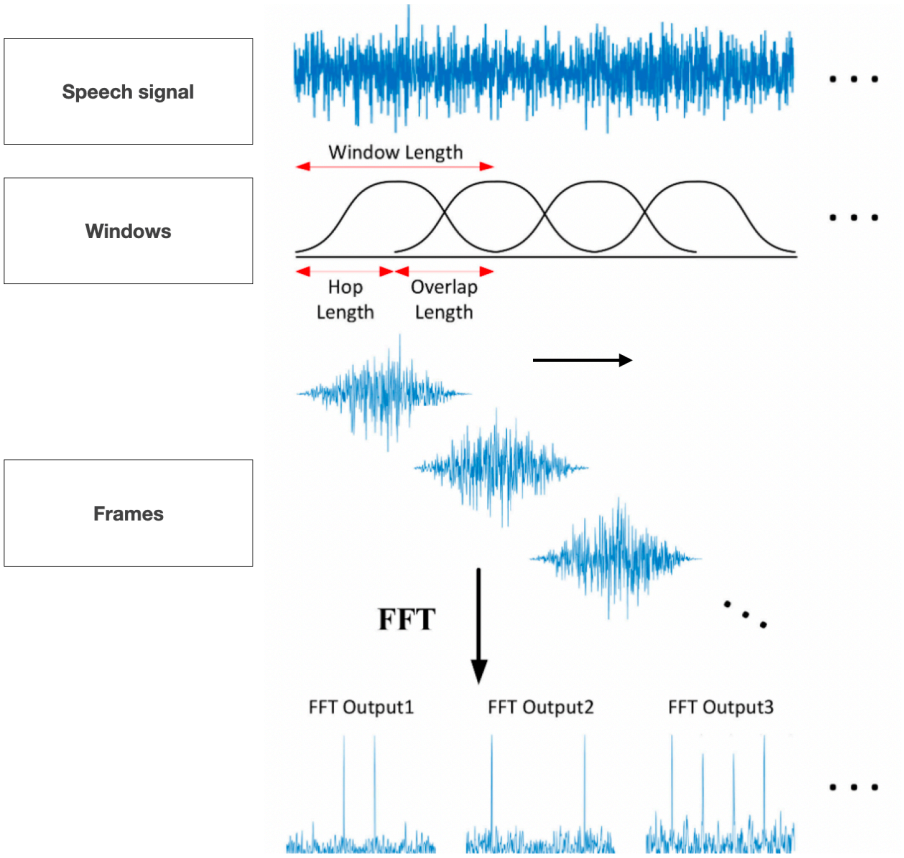


Figure 2.4: Schematic overview of Short-time Fourier transform (STFT) (Jeon et al. 2020)

2.4. Public corpora of stuttering speech

Data play a significant role for ML and DL models. Despite the availability of public corpora for speech recognition and detection tasks such as Libirspeech (Panayotov et al. 2015), TIMIT (Garofolo, John S. et al. 1993), and Arabic Speech Corpus (Mubarak et al. 2021). There are limited datasets in the literature for stuttering speech recognition (Alharbi et al. 2018, Kourkounakis et al. 2021, Lea et al. 2021, Khara et al. 2018). Thus, most stuttering detection and recognition research employ a small manually annotated dataset with limited speakers or artificially generated data. In a traditional ASR approach, disfluency data must be annotated with a full verbatim/time-aligned transcription (Alharbi et al. 2018). Furthermore, getting a public dataset for stuttering disfluency is more challenging than one with normal speech, and the amount of data currently available is small. This section reviews existing and public stuttering event datasets.

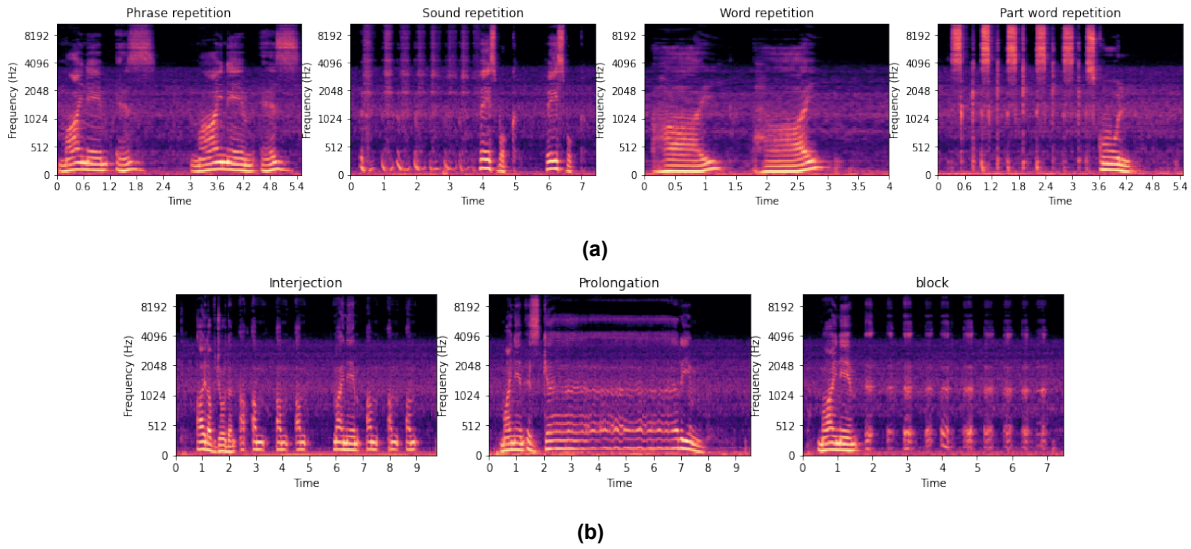


Figure 2.5: Spectrogram representation

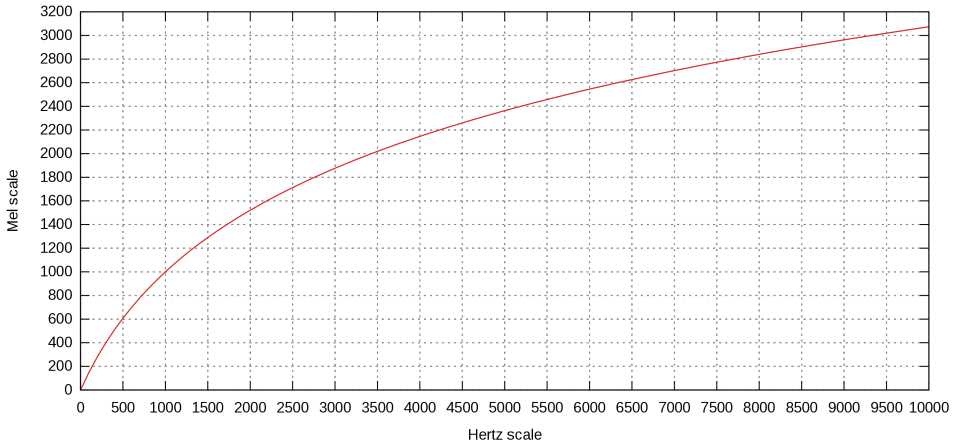


Figure 2.6: Mel scale vs Hertz scale. (Krishnavedala 2013)

2.4.1. UCLASS Dataset

The UCLASS (Howell et al. 2009) corpus created by the Wellcome Trust project at University College London is a common unlabelled dataset used in stuttering research. This corpus provides 456 records for people who stutter between 5 and 47 years old. The first release of this corpus was introduced in 2004 that includes monologue speech records for 138 distinct speakers, 120 male and 18 female. Later in 2008, the project team launched UCLASS release two, which contains 317 monologues, SSI-3 text readings and conversational speech. Table 2.3 shows the details of this corpus. In addition, information related to the records and speaker had been attached to the corpus.

Table 2.3: UCLASS dataset description

Release	Type	No speakers	Age	Age (σ)	Male	Female
One	Monologue	138	$5-47$	6.1	120	18
Two	Monologue	82	$7-20$	2.4	76	6
Two	Reading	108	$7-20$	2.9	93	15
Two	Conversation	128	$5-20$	2.5	110	18

Each speech record in UCLASS contains two pieces of information. Firstly, database fields associated with the speaker include gender, age, handedness, family history, physical problems (language, hearing), and treatment type. Secondly, databases related to the speech recording, such as orthographic, phonetic transcription, time alignment, tape noise, quality of the record and environmental noise. In the ASR disfluency classification approach, orthographic and phonetic transcription and time alignment are vital. However, a limited number of samples in UCLASS have transcription and time alignment. In orthographic transcription, the conventional alphabet and stuttering events labelling will be transcribed to represent the speech sample. In contrast, the phonetic transcription generates based on a phoneme concept. A phoneme is a discrete and distinctive unit of a language that can differentiate words. Most languages contain between 20 to 60 phonemes.

2.4.2. FluencyBank dataset

In 2018, the National Science Foundation (NSF) announced a new dataset called FluencyBank (Bernstein Ratner & MacWhinney 2018). The FluencyBank is a dataset provided for the studies of fluency development, and it is part of a larger open-access repository called TalkBank. This dataset includes monolingual and bilingual speech for Children and Adults Who Stutter (C/AWS), clutter (C/AWC), and second language learners. Speech sample data for this dataset are collected from stuttering assessment sessions of interview and reading tasks. In the interview task, a common set of interview questions were asked by SLP to the PWS, while in the reading task,

the PWS read ”Friuli,” a reading passage from SSI-4. Each data sample is linked with a transcription written in CHAT format (MacWhinney, 2000). However, most of these transcriptions are not annotated with stuttering events. The dataset comes with two separate sub-datasets, Voices-CWS and Voices-AWS. Table 2.4 describes this dataset.

Table 2.4: FluencyBank dataset description

Corpus name	Speaker age (Year)	Number of samples	Media type
Voices-CWS	$9-17$	12	video
Voices-AWS	adults	46	video

2.4.3. SEP-28k dataset

Apple provided a new dataset called SEP-28k (Lea et al. 2021). The SEP-28k dataset contains $28,177$ audio clips curated from 385 episodes of 8 stuttering YouTube podcasts. For each episode, $40-250$ three-second intervals near the speech pause segment had been extracted and annotated. In (Lea et al. 2021) also, $4,144$ (3.5) hours of the audio clip taken from the FluencyBank dataset (Bernstein Ratner & MacWhinney 2018) were annotated and validated using Fleiss Kappa inter-annotator agreement. This dataset consists of six stuttering events. Table 2.5 presents the total number of stuttering events in the FluencyBank and SEP-28k datasets for five different disfluency classes and one fluent class. Stuttering events include prolongation, block, sound repetition, word repetition, and interjection. Overall, we can observe that the non-stuttering event is the most frequent in both datasets. It is also clear that the distribution of data is imbalanced.

Table 2.5: Number of samples in FluencyBank and SEP-28k datasets

Dataset	Prolongation	Block	Sound Rep	Word Rep	Interjection	No Stuttering	Number of samples
SEP-28K	3480	1679	2517	2295	4322	13884	28177
FluencyBank	526	292	421	361	754	1790	4144

2.5. SED related tasks

2.5.1. Detection by Classification Task

The main objective of SED approaches is to recognise stuttering events within the speech signal. Thus, two approaches are cited in the literature to achieve this goal: detection-by-classification and ASR-based approach. In the detection-by-classification approach illustrated in Figure 2.7, a temporal acoustic region within the speech signal is annotated in a binary manner with one or more stuttering events. These annotations indicate if the stuttering events are active or inactive in that region. In addition, these annotations may include extra metadata describing the segment, such as start and end intervals, as well as the speaker id and gender. In SED models, the duration of the temporal region is generally between three and four seconds. e.g., in the SEP-28K dataset and FluencyBank, each segment has a fixed size of three seconds. Using the fixed-sized region makes supervised learning methods suitable for SED task, where the speech segments and the annotations are used to train the model, as presented in Figure 2.8.

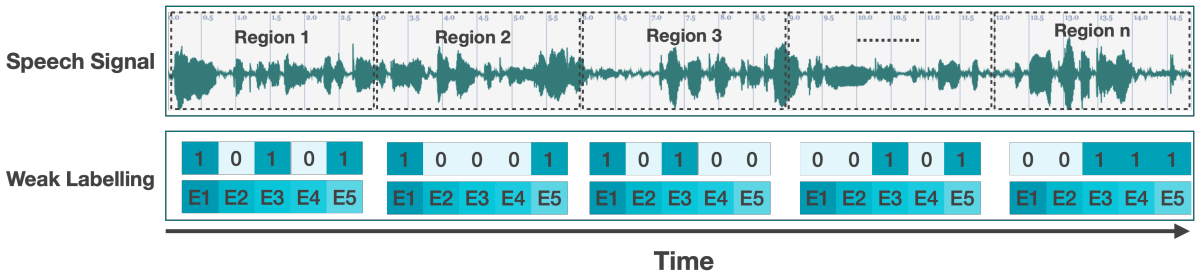


Figure 2.7: The detection-by-classification approach shows that a temporal acoustic region within the speech signal is annotated in a binary manner with one or more stuttering events

In the training phase, the SED model learns from hand-crafted or learned features and the temporal annotation data representing each stuttering segment. Stuttering annotation is frustrating and needs SLP, as mentioned in **Chapter 1**. Even between SLP, there is a disagreement in the annotation process on specific stuttering events. Therefore, this disagreement should be resolved before the training phase. In the testing phase, the extracted features for each speech segment are passed to SED; the output is a binarised vector indicating each consecutive segment's active and inactive

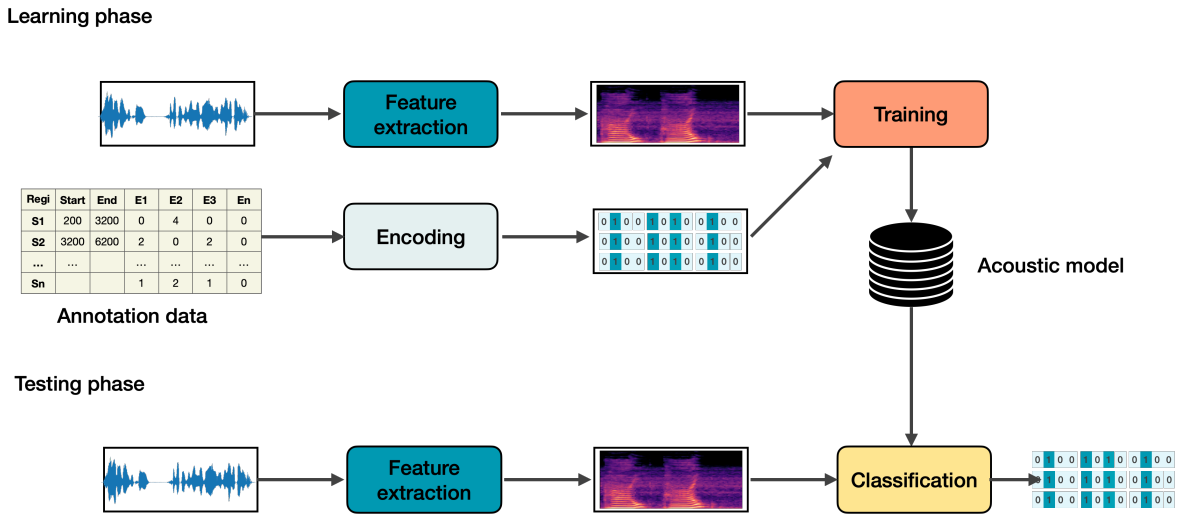


Figure 2.8: Supervised learning methods for SED task, where the speech segments and the annotations, are used to train the model

status for each stuttering event.

2.5.2. Handling class imbalance

As mentioned in the previous section, the data distribution in the existing datasets is imbalanced amongst stuttering events, where the non-stuttering class (majority class) is dominant in both datasets while stuttering primary events are minority classes. Figure 2.9 and Figure 2.10 illustrate this distribution. In fact, most of the speech samples, even in stuttering, are fluent, especially in mild and low severity level cases; this nature of stuttering speech may affect on model training phase. The class imbalance problem may occur in different DL and ML detection tasks, the training of these tasks is based on the assumption that the distribution of data among classes is balanced. The imbalance problem will affect model training because the model will be biased toward the majority classes. Previous research used different approaches to tackle this problem, such as oversampling, downsampling, and reweighing. In this thesis experiments, reweighing in the loss function is used to overcome this problem.

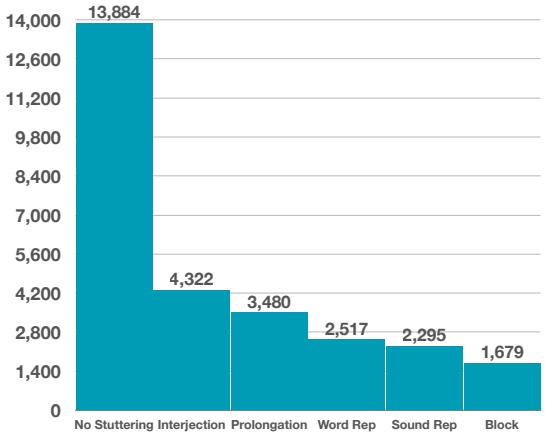


Figure 2.9: The distribution of data amongst events in the SEP-28K dataset

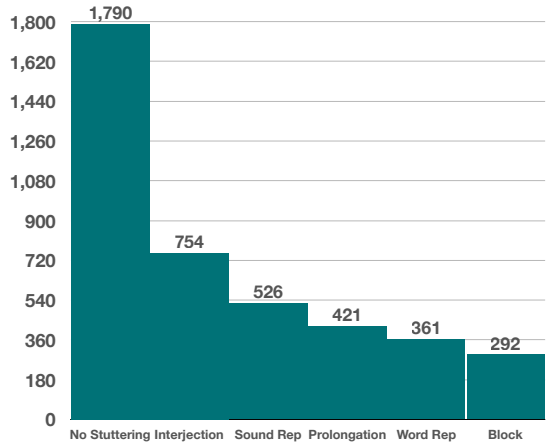


Figure 2.10: The distribution of data amongst events in the FluencyBank dataset

The reweighing technique is a method that re-balances the loss function by selecting an effective number of samples $E_{n_y} = \frac{(1-\beta^n)}{(1-\beta)}$ for minority and majority classes, where n is the number of samples and $\beta \in [0, 1)$ is a hyperparameter. Class-Balanced (CB) loss (Cui et al. 2019) is one reweighing method. For example, suppose x is the input sample associated with label y where $y \in \{1, 2, ..., C\}$ and C is the number of dataset classes. Then the Class-Balanced Loss (CB) is given as

$$CB(\mathbf{p}, y) = \frac{1}{E_{n_y}}\mathcal{L}(\mathbf{p}, y) = \frac{(1-\beta)}{(1-\beta^{n_y})}\mathcal{L}(\mathbf{p}, y)$$

where **p** is a vector of predicted class probabilities where $p_i \in [0, 1] \forall i$, $\mathcal{L}(\mathbf{p}, y)$ is a loss function, and E_{n_y} is the effective number of samples for class i.

Focal Loss (FL) (Lin et al. 2020) is another reweighing technique used to tackle the effect of the imbalance distribution of the data. FL is a hyperparameter modulating factor $(1-p_t)\gamma$ with tunable focusing parameter $\gamma \geq 0$ that attaches to a loss function to increase attention on misclassified and hard examples. It decreases the relative loss for well-classified examples. FL is obtained as

$$\text{FL}\,(p_\text{t}) = -\alpha_\text{t}\,(1-p_\text{t})^\gamma\,\text{log}\,(p_\text{t})$$

In DL, the Cross-Entropy (CE) loss is typically assigned in classification tasks with

sigmoid and softmax activation functions. CE loss is defined as

$$\text{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise} \end{cases}$$

where $p \in [0, 1]$ represents the estimated probability calculated by the model for the class with label $y = 1$ and $p \in \{0, 1\}$ identifies the ground-truth class. In (Lin et al. 2020) p_t is proposed for notational convenience as

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases}$$

The majority classes in imbalanced data dominate the gradient. Therefore, If $CE(p_t) = CE(p, y)$ α_t balances the importance of both good and bad examples.

$$\text{CE}\left(p_\text{t}\right) = -\alpha_\text{t} \log\left(p_\text{t}\right)$$

2.5.3. SED Evaluation

SED performance is evaluated in this thesis using k-fold cross-validation and hold-out test. Moreover, multiple statistical metrics are considered to rigorously evaluate the quality of the SED model.

Performance metrics

- *Accuracy* is an evaluation matrix commonly used in SED model evaluation. It represents the total number of correctly predicted stuttering events to the total number of events. Accuracy is computed from the confusion matrix, which is a breakdown of the prediction scores into TP, TN, FP and FN. Accuracy is defined as

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

- *Precision* represents the ratio of correctly predicted positive stuttering events TP to the total number of positive events $TP + FP$. Precision is computed as

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

- *Recall* represents the model's sensitivity. It measures the ratio of correctly predicted positive events TP to all stuttering events in the actual class $TP + FN$. Recall is calculated as

$$Recall = \frac{TP}{TP + FN}$$

- *F1-score* is the weighted average of $Recall$ and $Precision$. It is used to measure the performance of imbalanced datasets and is computed as

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

- Unweighted Average Recall (UAR) is a combination of Sensitivity $\frac{TP}{TP+FN}$ and Specificity $\frac{TN}{FP+TN}$. In a binary classification task, sensitivity is the ratio of the total predicted positive to the total number of positives, and it is called the recall on the positive class. While specificity is the total correctly predicted negative to the total number of negatives, it is called recall on the negative class. UAR is calculated as

$$UAR = \frac{specificity + sensitivity}{2.0}$$

- ERR (Equal Error Rate) is a statistical metric that measures the misclassified predictions by the model. It is calculated as

$$EER = \frac{FP + FN}{FP + FN + TN + TP}$$

k-fold cross-validation and hold-out test

In a hold-out method, stuttering datasets are split into training and test sets to observe how the trained models perform on unseen stuttering events; typically, random splitting with a 10 % and 90 % test and training sets, respectively, was followed. In addition,

k-fold cross-validation has been used to evaluate the results of the experiments. In k-fold, the training set is randomly divided into k-fold as depicted in Figure 2.11. One fold is employed to validate the model performance, while the remaining is used to train the model. In the thesis experiments, 10-fold cross-validation has been used; in this case, the training data is split into 10 folds, where each fold is considered a test.

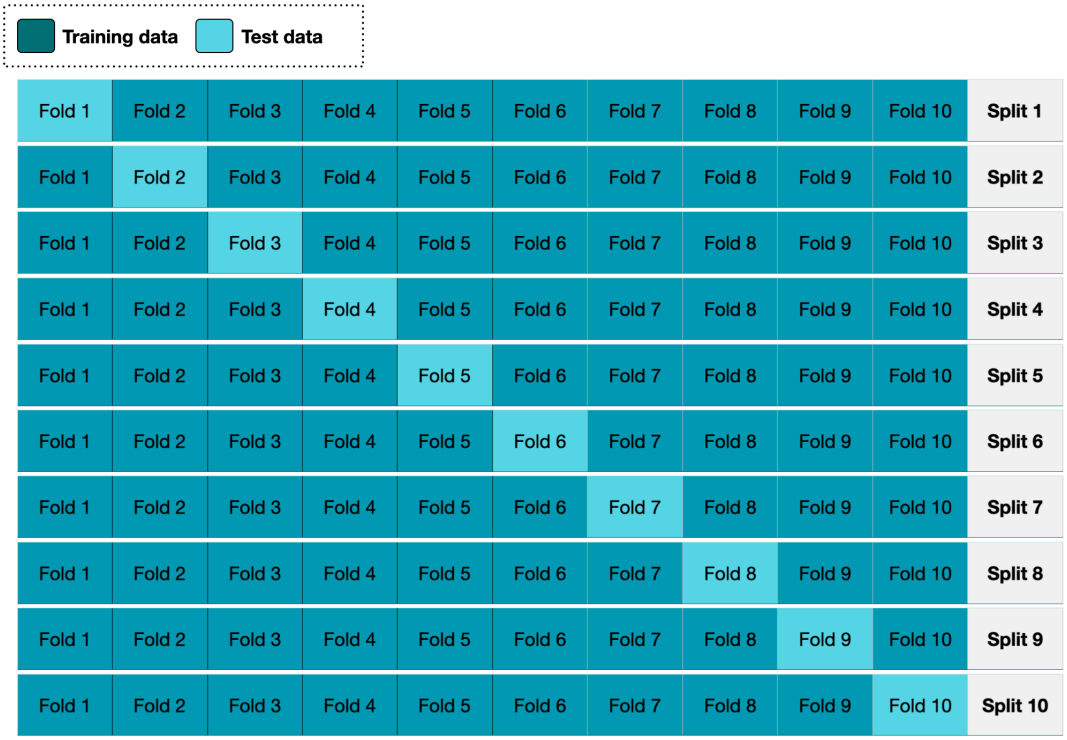


Figure 2.11: In 10-fold cross-validation, the training set is randomly divided into 10 folds, where one fold is employed to validate the model performance, while the remaining is used to train the model

2.6. Chapter summary

This chapter presented an overview of stuttering as a fluency disorder, including topics such as the types, symptoms, causes, treatment and evaluation of stuttering. Like other speech disorders, stuttering is classified into three types neurogenic, developmental and psychogenic each of these types has different causes, as explained in this chapter. Speech pathology research demonstrates different treatment approaches for stuttering. After an extensive assessment session by SLP, the best treatment plan can be decided; however, the treatment plan may not reduce all stuttering events. There-

fore, the SLP helps PWS with appropriate techniques to enhance fluency and develop communication skills. In addition, this chapter focused on the nature of sound and the bio-acoustic representation of human speech.

Moreover, reviewed the existing stuttering datasets, namely UCLASS, FluencyBank and SEP-28k. Furthermore, this chapter explains the SED related tasks, handling the imbalanced nature of stuttering datasets, experimental setup and evaluation metrics. In the next chapter, a comprehensive literature review and a background study in stuttering event detection using DL and ML approaches will be covered.

3

Literature Review

3.1. Introduction

As mentioned in **Chapter 1**, the motivation of this work is to investigate creating a robust SED that may help in stuttering severity evaluation and enhance the current speech assistive technology. Stuttering severity evaluation is time-consuming for SLP and PWS. The SLP manually tracks and observes all stuttering symptoms, such as speech symptoms, physical concomitants, and the naturalness of speech. e.g. the SLP in the counting procedure presented in **Chapter 2** tries to count each stuttering event through or after the assessment sessions, SED is the essential activity in this assessment. To provide an automatic and objective stuttering assessment tool, SED is under extensive investigation for advanced speech research and applications. Despite significant progress achieved by various ML and DL models, SED directly from speech signal is challenging due to the heterogeneous and overlapped nature of stuttering speech. SED contains three essential operations: data preparation, feature extraction, and modelling. In the data preparation phase, a detection model is built using a small manually annotated dataset derived from UCLASS (Howell et al. 2009), FluencyBank (Bernstein Ratner & MacWhinney 2018), SEP-28k (Lea et al. 2021), artificially generated datasets or custom datasets collected by the researchers. In the feature extraction phase, previous research focuses on acoustic analysis, parametric and nonparametric feature extraction. Various traditional ML and DL models have been proposed in for stuttering detection in the modelling phase. Previous research

used machine learning algorithms such as SVM, MLP,kNN, HMM, GMM, TDNN and Artificial Neural Network (ANN) to enhance the performance of SED. In addition, DL approaches such as CNN, RNN, and sequence-to-sequence methods are commonly used in stuttering detection models.

This chapter begins with an introduction highlighting the importance of SED. It then provides a comprehensive literature review and a background study on stuttering event detection using traditional ML, DL and multi-feature approaches. The conclusion section summarises the main points covered in the literature review and establishes a connection to the subsequent chapters.

3.2. Stuttering event detection using ML

There are existing studies on stuttering event detection and how parametric, nonparametric, hybrid features and acoustic processing techniques are leveraged with ML and DL (Barrett et al. 2022). A summary of stuttering detection and classification models is presented in Table 3.1.

Different ML methods have been proposed in the literature for stuttering detection. In, Howell & Sackin (1995), as the first SED method, a fully connected ANN was applied to detect two stuttering events, prolongation and repetition. In addition, ACF with a time-shifted version, spectral features and envelope parameters have been extracted from the audio signal and used as input to the network. The model was trained on two minutes speech samples using a single vector representing the speech signal's envelope parameters or a mix of ACF and spectral coefficients. The model's best accuracy was 82 % in the prolongation event using ACF and spectral coefficients and 79 % in the repetition using envelope parameters alone.

Howell et al. (1997) appropriated an ANN model for fluent, repetition and prolongation stuttering events. The model was trained on speech samples from 12 speakers on the reading task. Moreover, manual segmentation for linguistic units and disfluencies categorisation for each segment are followed before feature extraction. Furthermore, spectral measures, fragmentation measures, duration, and energy features in

an acoustic signal have been utilised as input to the model. The results determined that the most suitable parameters for the stuttering detection task were spectral fragmentation for the whole word event and the supralexical disfluencies of energy in the part word event. The model achieved an average accuracy of 92 %.

Czyzewski et al. (2003) suggested a stuttering detection model based on Rough Sets and ANN. In addition, it reports results for automatic recognition of stuttering speech in standard and DAF speech. The features vector of formant frequencies and its amplitude were used to detect vowel prolongation, syllable repetition, fluent, and stop-gaps. The feature parameters were selected because the contractions of articulation muscles affect and change the articulation system, and these changes are visible in spectral and cepstral analysis. The experiments were conducted on six fluent speakers and six people with stuttering. The best accuracy achieved was 78.10 %.

Several research works (Tan et al. 2007, Smołka et al. 2007) used HMMs in SED. Tan et al. (2007) developed Malay Speech Therapy Assistance Tools that help pathologists diagnose and train children who stutter. The system model trained on 35 stuttering samples on certain words, the model's accuracy was 96 % on normal speech and 90 % on stuttering samples, and a pre-emphasised MFCC feature vector was employed as input to the model. Moreover, Smołka et al. (2007) proposed two techniques on SED of prolonged fricative phonemes using MFCC and HMM classification methods on 38 Samples. The accuracy of the model was 80 %.

Świetlicka et al. (2009) proposed a binary classification model that utilised MLP networks and spectral measure feature to create the SED model. 59 fluent and non-fluent samples from 8 speakers were used to train the proposed model, and the best accuracy was 88.1 %. (Chee et al. 2009*b*, Chee et al. 2009*a*) suggested repetition and prolongation classification model based on kNN and LDA. In addition, MFCC and LPCC feature extraction are implemented to check model effectiveness in recognising stuttering events. The accuracy was 90 % using MFCC and 89 % with Linear Predictive Cepstral Coefficient (LPCC) feature extraction.

Mahesha & Vinod (2016) demonstrated a SED model based on GMM and MFCC

as an acoustic feature, where the GMM parameters were estimated. GMM is used because it can represent a large group sample distribution and it directly integrates with ASR. In addition, the model could recognise what was being said regardless of the stuttering event. The model was trained on 200 selected samples from the UCLASS dataset 50 for each disfluency event, and these events include syllable repetition, word repetition, prolongation, and interjection. Moreover, speech samples in this research were segmented and annotated manually; 80 % of data were used for training and the rest for testing. MFCC features were extracted based on the short-time window analysis with a $20\ ms$ frame size and a $10\ ms$ overlap.

The research concluded that the classification of disfluencies could be a pattern recognition problem. Moreover, the model's accuracy depends on the number of mixture components and the MFCC coefficients because it may allow the best-fit data representation. Nevertheless, the average accuracy was more than 96%.

Villegas et al. (2019) measured respiratory and heart rate biosignals for 68 participants to indicate two-class stutter or not-stutter, the study applied on a reading task. A multilayer perceptron with 40 hidden layers with statistical features such as mean, standard deviation employed to make a binary classification for the data the model's average accuracy reported was 82.6%. Dash et al. (2018) suggested a method to correct and recognise stuttering events with an adequate time. Therefore, amplitude thresholding within neural networks was designed to remove prolongation from speech samples. Further, a string repetition removal algorithm using an existing Text-to-Speech (TTS) system was implemented to remove repetitions. The model achieved an overall stutter classification of 86% on a test set of 50 speech sample.

Sheikh et al. (2021) proposed a lightweight model architecture for stuttering detection based on Time Delay Neural Network (TDNN) with MFCC as a sole acoustic feature. The authors argued that TDNN could capture the contextual pattern of stuttering events. The model was evaluated on 128 unique speakers from the UCLASS dataset, 18 female and 110 male samples. The speech samples are annotated manually. The model outperforms the state-of-the-art method ConvLSTM on block and fluent classes with 46% and 70% F1 scores, respectively. However, the model achieved lower F1

scores on repetition and prolongation with 27% and 16%. The authors optimised the baseline model with different filter bank sizes and dilation rates. The authors argued that a large context window increases the detection of prolongation and repetition. In contrast, the detection of block and fluency decreases, which may be true because the prolongation and repetition last longer than the block and fluency. TDNNs can capture the temporal relation by increasing the receptive field across the frequency domain. However, increasing the receptive field on the time and frequency domains of the speech signal may enhance the detection rate.

In addition to traditional ML, DL approaches such as convolutional, recurrent neural networks, and sequence to sequence methods are commonly used in stuttering detection models.

3.3. Stuttering event detection using DL

Various DL techniques have been suggested for SED. As presented in this section, DL approaches such as CNN, RNN networks, and sequence-to-sequence methods are commonly used in stuttering detection models.

A recurrent network with Bidirectional Long-Short Term Memory (BI-LSTM) followed by Integer Linear Programming (ILP) post-processing employed in (Zayats et al. 2016), this research proposed a different method in disfluency classification for word sequences. Pattern match features were designed to decrease the vocabulary size in training, which improved the word sequence's performance; the miss rate was 19.4% over each repetition. The model accomplishes state-of-the-art performance for disfluency detection and correction detection tasks.

Kourkounakis et al. (2020) proposed a deep neural model based on BI-LSTM and a residual network. In the proposed model, 25 samples of the UCLASS dataset were used, each sample split into 4-second audio clips and labelled manually because most of the UCLASS data are unlabelled. Moreover, these samples were selected from the UCLASS dataset due to the availability of orthographic transcription. The training data were 50 minutes of the speech contains fluent and stuttering events.

After the annotation process, each 4-second of the UCLASS speech signal is framed into $10\ ms$ short-timeframes. After framing, they applied a window function of $25\ ms$, and STFT was used to generate the spectrogram features. The spectrogram was then fed to a residual network with six blocks and 18 convolutional layers, which were employed as a feature embedding layer. A ResNet model was used, and the detection task for each stuttering event was formulated as a binary classification problem. The learned feature embeddings are provided to 2 recurrent layers, each consisting of 512 LSTM units, and to make the model learn both past and future embedding, they employed a BILSTM. The authors used leave-one-subject-out cross-validation. The training model used 24 speakers, and the last speaker was used for testing.

In the evaluation, the authors used accuracy and missing rate. The result shows that the model outperformed state-of-the-art in sound repetition and revisions by 41.90% and 22.14%. The authors stated that despite the other research ignoring the interjection event, the model detects interjection with 81.4% accuracy.

The authors demonstrated a new approach for stuttering events detection, and the model was able to detect most of these disfluencies. Moreover, they employed a ResNet as a feature embedding layer rather than the other methods, which employed only the LSTM and Bi-LSTM layer. However, the UCLASS data are imbalanced, and the classification model will be biased for the majority classes since they used binary classification for each disfluency. Therefore, the accuracy and the missing rate may not be proper matrices to evaluate the performance. Consequently, the precision, recall and F1 score could be more proper for these data. Sheikh et al. (2021) examined the ResNet and LSTM architecture on the UCLASS release one with 128 speakers. The authors stated that the F1 score for the repetition was 22.00%, the prolongation and block respectively were 28.00% and 44.00%, and the fluent class was 52.00%. Therefore, more experiments maybe required to improve model generalisation and robustness using cross datasets and domain adaptation.

Chen et al. (2020) demonstrated a real-time self-attention transformer model that predicts punctuation marks and was able to detect repetition and interjection in real-time. This model helps to suspend part of the output with a controllable time delay to

meet the real-time limitations in partial decoding, which is required in other automatic speech recognition (ASR) solutions. The proposed study achieved 38% miss rate and 70.5% F1 scores. Moreover, the model was trained using the English IWSLT2011 benchmark dataset and an in-house Chinese annotated dataset.

Kourkounakis et al. (2021) proposed the FluentNet architecture, which is an end to end network for disfluency classification. The input signal clipped into a fixed size 4-second audio clip recorded with $16khz$ sample rate then converted to spectrogram using STFT with 256 filters. Each frequency frame within a spectrogram was generated using a $25\ ms$ frame and a stride length of $10\ ms$ (60% overlap). These spectrograms passed through Squeeze-and-Excitation Residual Network for vectorisation purposes. The acoustic vectors then passed to BI-LSTM to find a relationship between different spectrograms in a time-series manner. An attention mechanism had been added to the final recurrent layer to concentrate on the important features required for stuttering classification. FluentNet utilised the ResNet and Squeeze-and-Excitation (SE) to effectively represent acoustic features.

Mohapatra et al. (2022) proposed a deep learning approach based on a convolutional neural network and contextual and data distillation using Wav2vec 2 (Baevski et al. 2020) hidden states. The proposed method evaluated different data sizes, using a few samples of SEP-28k and FluencyBank datasets; the authors argued that the model was generalised across different speakers. In addition, the authors resolved the data reliability problem by utilising the agreement between three annotators. Using more acoustic features rather than Wav2vec embeddings may enhance the robustness of SED. The model achieves average F1 scores of 70.60 and 64.80 on the SEP-28k and FluencyBank datasets, respectively.

3.4. Stuttering events detection using multi-feature DL

Lea et al. (2021) proposed a new dataset SEP-28K with $32k$ audio clips,$28k$ collected from public podcasts, and $4k$ clips from the FluencyBank dataset annotated using the time-interval technique explained in Chapter 2. These data are annotated with six

disfluency events blocks, prolongations, sound repetitions, word Repetitions, interjections, and fluent. In addition, the author proposed a multi-task learning ConvLSTM stuttering detection model based on LSTM and a convolutional neural network. The baseline model consists of a single RNN layer with LSTM. While the enhanced model created based on a single convolutional layer and batch normalisation layer produces weights for the model features followed by an LSTM layer and classification layer with two branches.

Different time domain, articulatory and frequency domain features are used in this model. Mel-filterbank energy with 40 dimensions was used with a cut-off frequency of zero Hz and 8000 Hz, and $100\ Hz$ as a sample rate. In addition to the filter bank, three F0 dimensions, vector size with 41 phoneme probabilities extracted from a pre-trained acoustic model and articulatory vocal-tract features were utilised. To resolve the imbalanced nature of stuttering speech data, the author used weighted cross-entropy with a FL on the fluent/disfluent branch. The concordance correlation coefficient was applied to resolve the inter-rater agreement issue. The model was evaluated on two datasets, FluencyBank and SEP-28k. The F1 scores for this model were 66.2% on SEP-28K and 75.8% on the FluencyBank dataset. The proposed model achieved a good F1 score of 66.2% on SEP-28k. The paper shows that employing multi-feature may improve SED generalisation and robustness. Therefore, applying other time-domain and pre-trained ASR features in the detection task could enhance the detection performance.

Jouaiti & Dautenhahn (2022) suggested a neural network that combines MFCC and phoneme classes and probabilities to detect four stuttering events. The model also contains another branch for disfluency classification. After the 3-second speech signal was downsampled to $8KHz$, a 20×47 vector of MFCC coefficient, phoneme class probability with 18×299 dims and phoneme estimation 1×299 dims were extracted. The extracted features are then fed into three parallel models. Each model contains a Bi-directional LSTM layer followed by a ReLU activation function, time distribution and dense layers. The output of each model is merged and passed via two dense layers, followed by batch normalisation and dropout layers. At the end of the

network, two separate classifiers are employed to detect and classify stuttering events. The author used 10-fold cross-validation to validate the model against three datasets 25 speakers extracted from UCLASS, selected samples from FluencyBank and SEP-28k. The model outperforms the model of Lea et al. (2021) in all aspects except the interjection and block events on the FluencyBank dataset. The authors utilised the undersampling technique to resolve the imbalanced nature of stuttering speech data by randomly deleting speech samples.

3.5. Chapter summary

To provide an automatic and objective stuttering assessment tool, SED is under extensive investigation for advanced speech research and applications. Despite significant progress achieved by various ML and DL models, SED directly from speech signal is challenging due to the heterogeneous and overlapped nature of stuttering speech and the reliability of training datasets. Therefore, this chapter provided a comprehensive literature review and a background study on stuttering event detection using deep learning, machine learning approaches and multi-feature fusion methods. The next chapter will explain the steps to prepare the experimental data, explore the data segmentation approach and present data preparation and annotation details.

Table 3.1: Summary of previous works in stuttering classification and detection

Authors	Year	Classifier	Features	Dataset size	Stuttering events	Results
Howell & Sackin (1995)	1995	ANN	ACF Envelope parameters	Two minutes 6 samples	Prolongation Repetition	Best Acc: 82 % for Prolongation 79 % for Repetition
Howell et al. (1997)	1997	ANN	Duration Energy peaks	12 sample	Repetition Prolongation	Acc: 92 %
Czyzewski et al. (2003)	2003	ANN	Formant	12 speech sample 6 Fluent 6 with stopgaps	Stop-gaps Syllable Repetitions vowel Prolongation	Avg.Acc: 78.1 %
Tan et al. (2007)	2007	HMM	MFCC	20 sample of Fluent speech 15 of artificial shuttering speech	Repetition Prolongation Block	Acc: 96 % for normal speech 90 % for artificial shuttering speech
Ravikumar et al. (2009)	2009	SVM	MFCC	15 samples of speech	Repetition Non-Repetition	Acc: 94.35 %
Hariharan et al. (2012)	2012	kNN LDA	LPC LPCC WLPCC	39 Samples from UCLASS	Prolongation Repetition	WLPCC 97.06% LPCC 95.69% LPC 93.14%
Pálfy (2014)	2014	SVM	MFCC	16 Samples from UCLASS	Word Repetition Prolongation	Acc: 98.00 %
Zayats et al. (2016)	2016	BLSTM	MFCC	Switchboard Corpus	Repetition	F1: 85.9 MR: 19.4
Mahesha & Vinod (2016)	2016	GMM	MFCC	50 selected sample from UCLASS	Syllable Repetition Word Repetition Prolongation Interjection	Avg.acc 96 %
Alharbi et al. (2018)	2018	Finite State Transducer Amplitude Time Thresholding	Word lattice	129 UCLASS	Sound Word Part-Word Phrase Repetition Revision Prolongation	Avg. MR: 37% false positive rate FPR 0.89%
Dash et al. (2018)	2018	STT	STT Amplitude	50 Speech Samples	Repetition Prolongation	Acc: 86%
Kourkounakis et al. (2020)	2020	Residual network BI-LSTM	STFT	25 speech sample UClass	Phrase Repetition Sound Repetition Word Repetition Revision Prolongation Interjection	MR 10.03% Acc :91.15%
Chen et al. (2020)	2020	Transformer	Word Embeddings	English IWSLT2011 benchmark	Fluent None-Fluent	MR: 38% F1: 70.5
Kourkounakis et al. (2021)	2021	SE network BI-LSTM	STFT	25 speech sample UClass,20 hours LibriStutter	Phrase Repetition Sound Repetition Word Repetition Revision Prolongation Interjection	MR: 9.35% Acc: 91.75%
Lea et al. (2021)	2021	ConvLSTM	Filterbank Pitch features articulatory features	SEP-28k 28000 audio clip FluencyBank 4000 audio clip	Interjection Sound Repetition Word Repetition Prolongation Fluent	F1: on SEP-28K 66.2% F1 on FluencyBank 75.8%
Sheikh et al. (2021)	2021	TDNN	MFCC	138 speaker UCLASS	Interjection Repetition Fluent Prolongation	Acc: 50.79%
Jouaiti & Dautenhahn (2022)	2022	BI-LSTM	MFCC Phoneme classes Phoneme probability	UCLASS FluencyBank	Interjection Sound Repetition Word Repetition Fluent Prolongation	F1: 73.38%
Mohapatra et al. (2022)	2022	CNN	Wav2vec embeddings	SEP-28k FluencyBank	Interjection Sound Repetition Word Repetition Block Prolongation	Avg.F1 on SEP-28k: 70.60% Avg.F1 on FluencyBank 64.80

4

Time Interval Annotation of Stuttering Data

The previous chapter provided a comprehensive literature review and a background study on SED, discussing the challenges of detecting stuttering events. One of these challenges is the scarcity of annotated data. Despite encouraging results of the stuttering detection methods in previous research, these methods have several limitations. The main limitation is the reliability of the training data. The detection models are based on manually labelled UCLASS data, where the number of observations on each stuttering event and the agreement in the annotation process have not been measured. Annotation of stuttering speech is challenging, needs SLP expertise, and should be appropriately measured (Barrett et al. 2022). Therefore, two stuttering domain experts and the author were engaged in a data annotation process to ensure the reliability of the thesis's experimental data. Moreover, an annotation tool was developed to streamline and ease the annotation process.

This chapter provides two contributions. Firstly, annotations of 47 unique speakers (29 male and 18 female) from the UCLASS dataset. Secondly, an open-source annotation tool that can assist in stuttering research projects is developed, implemented and presented. This chapter explains the data preparation phase using the time interval annotation technique as illustrated in Figure 4.1. The chapter explores data segmentation approach and demonstrates data preparation and annotation details. Furthermore, it provides an overview of the annotation tool developed and used in the

Figure 4.1: Data preparation block diagram. UCLASS recordings for each speaker are divided into four-second audio segments. The speech segments are attached to a developed time interval annotation web tool to streamline and ease the annotation process. In most cases, each audio segment contains only one stuttering event.

annotation process, and it comprises the annotation procedure and agreement measurement. In addition to the UCLASS dataset, two different datasets, SEP-28k and FluencyBank, are prepared in this chapter.

4.1. Preparing UCLASS dataset

As mentioned, automatic SED is a widely recognised challenge due to diverse determinants such as the **nature of stuttering speech** and the **lack of reliable training data** (Lea et al. 2021, Kourkounakis et al. 2021). PWS tend to have fairly different kinds of stuttering events, which could be heterogenic and overlapped Ronald B. Gillam (2022). Most previous research results are based on the UCLASS dataset described in Section 2.4.1. The UCLASS corpus provides 456 records for people who stutter between 5 and 47 years old. The first release of this corpus was introduced in 2004 and included monologue speech records for 138 distinct speakers, 120 male and 18 female. Unfortunately, most of the speech audios in this dataset are not annotated, and there is no publicity available labelled data for this dataset that contains all stuttering core behaviours. Previous approaches built detection models by manually labelling the UCLASS data or selecting a limited number of samples with time-aligned and orthographic transcriptions from the original corpus. Therefore, this section presents the steps to prepare the UCLASS experimental data. The rest of this section is organised

as follows: Subsection 4.1.1 explores the data segmentation approach followed; Subsection 4.1.2 presents data preparation and annotation; Subsection 4.1.3 overviews the annotation tool; and Subsection 4.1.4 comprises the annotation procedure and agreement measurement.

4.1.1. Data segmentation

To compare experimental results with state-of-the-art methods and ensure the reliability of the experimental data generated, monologue speech records of 47 distinct speakers from the UCLASS dataset (Howell et al. 2009) were selected. Besides 29 male samples, all 18 female samples from the UCLASS release one were chosen. In general, the number of female samples in UCLASS is fewer than the number of male samples, and this is because 75% of all stuttering cases are in males (Manjula et al. 2019). The female samples were selected to observe the effect of gender on the detection task because the vocal tract and the speech rate are different between males and females. Hence, each speaker's speech was re-sampled of $16\ kHz$, divided into 4-second segments, and attached to the annotation portal, as illustrated in Figure 4.2. The duration of the data is approximately 95 minutes. In addition, the number of speech samples is 1554 $\left(Total\,number\,of\,segments\ = \frac{Duration(Speech\,sample)}{Length(Segment)}\right)$. All segments have a fixed size except the last segment in each speaker's speech. Therefore, these segments were excluded from the annotation process.

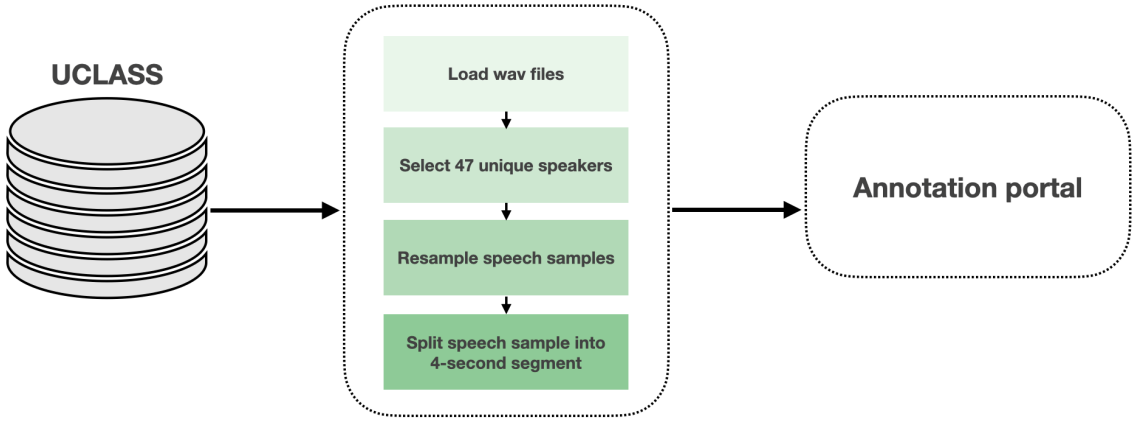


Figure 4.2: Data preprocessing block diagram. The monophonic UCLASS recordings for each speaker were sampled at 16000 Hz and divided into four-second audio segments

4.1.2. Data annotation

Model reliability and performance are correlated to the annotation process's quality. Therefore, the annotation process should be measured and achieved by SLPs (Barrett et al. 2022) due to the heterogenic nature of stuttering segments. e.g., the non-expert annotator cannot distinguish between the sound and part word repetition. In addition, few previous research reported the inter-rater agreement in the labelling process. The agreement in the annotation process may affect the result of the detection model and should be reported. To rigorously control and streamline the annotation process, a stuttering annotation tool was developed that makes the annotation process more effective. In addition, two SLPs were invited to provide annotations for speech segments, and the level of agreement between the two raters was measured as the inter-rater agreement.

4.1.3. Annotation tool

Stuttering speech annotation is a frustrating and time-consuming process. Therefore, a web-based annotation tool for stuttering labelling based on the Django framework (Django Software Foundation n.d.) was developed to ease and streamline the annotation process. The tool allows users to label stuttering data with one of eight stuttering events. The tool is designed to be used with stuttering audio data. It has two components: back-end and front-end components. The back-end manages datasets and annotators information, while the front-end is used by the users to annotate stuttering audio.

The main feature of the back-end is dataset management. The back-end allows the user to define dataset metadata and import dataset audio segments. In addition, some essential operations like searching, sorting, and filtering are provided by this component. In addition to the dataset management component, the back-end contains other core features to manage user and annotator information. On the other hand, the front-end component helps the annotator label data easily. It allows the user to select the dataset and speaker, listen to speech audio and evaluate the au-

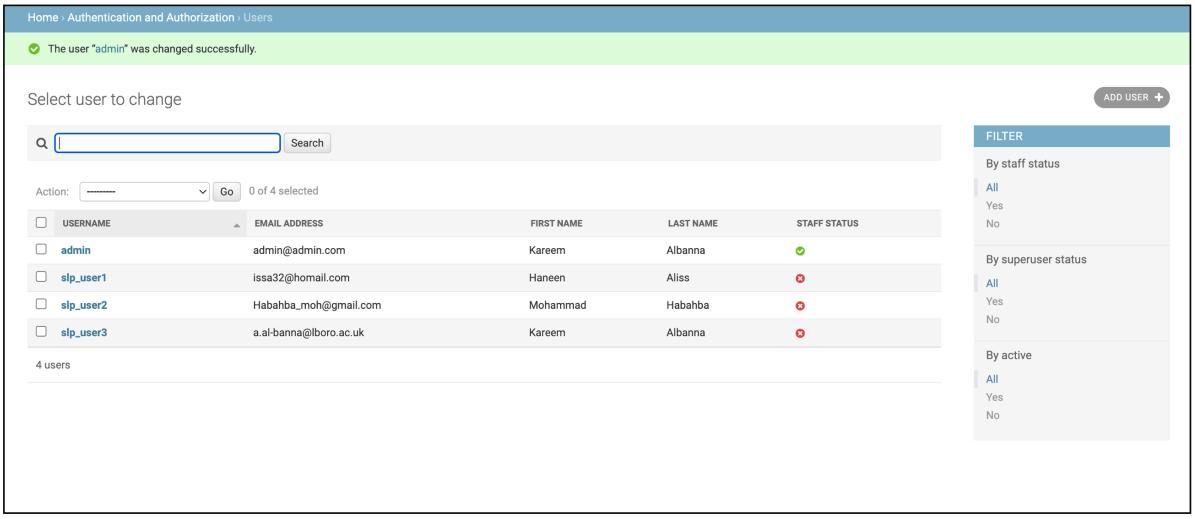


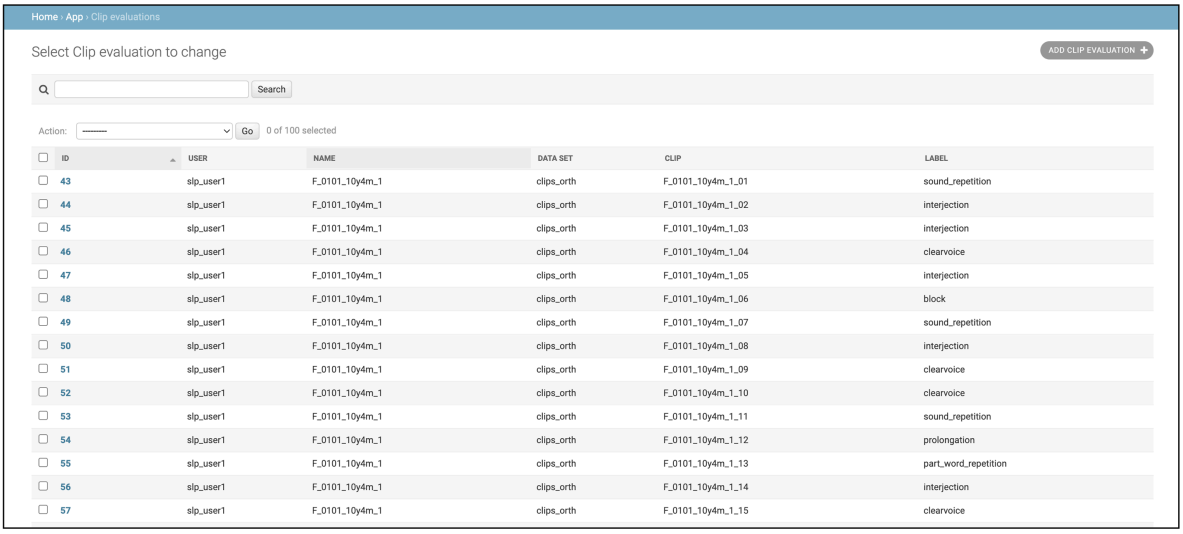
Figure 4.3: A screenshot of the back-end components that illustrate managing user and annotator information module

dio with one or more stuttering events. Figure 4.3 and 4.4 shows screen-shots of the system.

4.1.4. Annotation procedure and agreement measurement

This section explains the procedure followed to perform data annotation. In order to accomplish this task, a time-interval (Lea et al. 2021, Ingham et al. 1993) based evaluation was followed. In this evaluation, 4-second segments were imported into the portal and annotated with a binary label. In most cases, each segment contains only one stuttering event. However, it may have more than one stuttering event in a few cases. e.g., the audio clip illustrated in Figure 4.5 contains two stuttering events (word repetition and interjection); in this case, the audio will contain multiple labels. Accordingly, two registered SLPs, and the author were all involved in the annotation process. The SLPs were trained to use the annotation tool.

An experiment was conducted where three annotators, two SLPs and the author started the annotation process using the developed tool. The annotators were asked to listen to the 4-second speech segment from the previous segmentation step. Each annotator was asked to record whether they considered this segment one of the eight mutually exclusive categories (word repetition, sound repetition, part-word repetition,



ID	USER	NAME	DATA SET	CLIP	LABEL
43	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_01	sound_repetition
44	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_02	interjection
45	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_03	interjection
46	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_04	clearvoice
47	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_05	interjection
48	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_06	block
49	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_07	sound_repetition
50	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_08	interjection
51	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_09	clearvoice
52	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_10	clearvoice
53	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_11	sound_repetition
54	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_12	prolongation
55	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_13	part_word_repetition
56	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_14	interjection
57	slp_user1	F_0101_10y4m_1	clips_orth	F_0101_10y4m_1_15	clearvoice

Figure 4.4: A screenshot of the back-end components shows annotation for each speech segment.

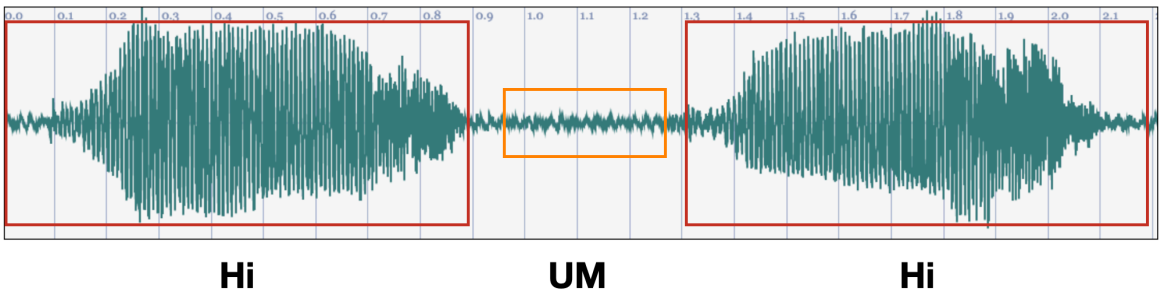


Figure 4.5: Speech sample that contains two stuttering events word repetition and interjection

interjection, prolongation, block or fluent).

To further analyse the annotation process, Fleiss' kappa (Fleiss et al. 2003) was applied to measure the inter-rater reliability agreement between the SLPs annotation. Fleiss' kappa is a statistical method used to determine the level of agreement between two or more annotators and is calculated as

$$\left(k = \frac{P_o - P_e}{1 - P_e}\right)$$

where P_o and P_e are the observed and expected agreement between the raters, respectively.

The level of agreement can be determined based on the standard kappa classification scale shown in Table 4.1. After the analysis, the results showed moderate agreement between the annotators across all the categories with a kappa value (k) of 0.487. For more details, the individual agreement on each category was computed as shown in Table 4.2.

Table 4.1: Fleiss kappa classification scale

Value of agreement	Strength of agreement
< 0.20	Poor
$0.21 - 0.40$	Fair
$0.41 - 0.60$	Moderate
$0.61 - 0.80$	Good
$0.81 - 1.00$	Very good

The results show good agreement on phrase and word repetition, moderate agreement on part-word repetition, interjection, prolongation, and fluency, and fair agreement on block and sound repetition. This result shows that even among the SLPs, there was disagreement in certain stuttering events. However, in stuttering evaluation techniques such as SSI-3, a statistical approach is used to overcome this problem by focusing on stuttering core behaviour and perceptual scaling described in Chapter 2.

Table 4.2: The individual agreement on each stuttering event among three annotators

Type	No.Samples	k value	Classification Scale
Block	256	30.2%	Fair
Sound Repetition	188	34.4%	Fair
Phrase Repetition	22	70.5%	Good
Word Repetition	87	60.6%	Good
Part-word Repetition	47	42.6%	Moderate
Interjection	172	56.0%	Moderate
Prolongation	203	53.5%	Moderate
Fluent	425	59.7%	Moderate
Total	1400	48.7%	**Moderate**

The result of the annotation process was compared with SEP28k (Lea et al. 2021) as shown in Table 4.3.

Table 4.3: Comparison between UCLASS and SEP-28k Fleiss kappa agreement values

Dataset	Block	Sound Rep	Word Rep	Interjection	Prolongation	Fluent
UCLASS	30%	34%	61%	56%	54%	60%
SEP-28k	11%	40%	62%	57%	25%	39%

4.2. Preparing Sep-28k and FluencyBank datasets

Experimental results in this thesis need to be rigorously evaluated. Thus two datasets reviewed in Chapter 2 are used in this work, SEP-28k and FluencyBank. The FluencyBank is a dataset provided for the studies of fluency development, and it is part of a more extensive open-access repository called TalkBank. This dataset includes monolingual and bilingual speech for Children and Adults Who Stutter (C/AWS), clut-

ter (C/AWC), and second language learners. Speech sample data for this dataset are collected from stuttering assessment sessions of interview and reading tasks. In the interview task, a common set of interview questions were asked by SLP to the PWS, while in the reading task, the PWS read "Friuli," a reading passage from the Stuttering Severity Instrument-4 (SSI-4) (G Riley 2009). The FluencyBank dataset comes with two separate sub-datasets, Voices-CWS and Voices-AWS. This research used the speech utterances annotated by Lea et al. (2021). The SEP-28k dataset includes $28,137$ speech segments curated from 385 episodes of eight stuttering YouTube podcasts. For each episode, three-second intervals ranging from 40 to 250 were identified near the speech pause segment and subsequently extracted and annotated. In Lea et al. (2021) also, Fleiss kappa's inter-annotator agreement was utilised to validate and annotate $4,144$ (3.5) hours of the audio clip taken from the FluencyBank dataset. Table 4.4 presents the stuttering events and their respective total number of observations and data size in hours in both datasets.

Table 4.4: The main categories and sub-categories of stuttering events and their respective total number of observations and data size in hours in FluencyBank and SEP28-k datasets

Dataset	Prolongation	Block	Sound Rep	Word Rep	Interjection	Total of Obs	Total Hours
SEP-28K	3480	1679	2517	2295	4322	14293	11.91
FluencyBank	526	292	421	361	754	2354	1.96

In the annotation process of SEP-28k and FluencyBank, which at least three well-trained annotators achieved, the inter-annotator agreement for stuttering events needed more consistency across categories. For example, word repetitions, interjections, and sound repetitions showed higher agreement levels (Fleiss kappa scores of 0.62, 0.57, and 0.40, respectively), while blocks and prolongations had only fair or slight agreement (Fleiss kappa scores of 0.25 and 0.11 respectively) (Lea et al. 2021). Therefore, in this thesis, the disagreement in the annotation is processed by taking the agreed stuttering core behaviours by three annotators. In **Chapter 7**, this reliable data has been used to improve the performance of SED as described in Table 4.5. Previous research handled this issue using concordance correlation coefficient (CCC) loss using the inter-annotator agreement for each utterance (Lea et al. 2021), while Mohapatra

et al. (2022) used the same approach followed.

Table 4.5: Description of the main categories and sub-categories of stuttering events and their respective total number of observations and data size in hours in FluencyBank and SEP28-k datasets after resolving the disagreement

Dataset	Prolongation	Block	Sound Rep	Word Rep	Interjection	Total of Obs	Total Hours
SEP-28K	1957	2096	1454	2165	1911	9583	7.985
FluencyBank	286	329	438	385	415	1853	1.544

4.3. Chapter summary

This chapter discussed the data preparation phase of the experimental data using the time interval annotation technique. The chapter explored the data segmentation approach and presented annotation details; it also comprised the annotation procedure and agreement measurement. In addition, it provided an overview of the annotation tool developed and used in the annotation process. This chapter also achieved research objectives 3 and 4 by providing two contributions. Firstly, annotations of 47 unique speakers (29 male and 18 female) from the UCLASS dataset. Secondly, design and implementation of an open-source annotation tool will help in stuttering research projects.

5

Stuttering Event Detection Using Atrous CNN

The previous chapter showed the steps followed to ensure the reliability of the experimental data. A time-interval-based annotation by stuttering domain experts was applied; in addition, Fleiss kappa has been used to determine the level of agreement between the SLPs' annotation on whether the 1400 4-second audio segment was fluent, interjection, repetition, block, or prolongation classes. The results showed good agreement on the repetition class, moderate agreement on interjection, prolongation, and fluent classes, and fair agreement on the block class. This chapter focuses on the following research question: *To which level can a robust stuttering event detection model be created, based on perceived acoustic features as a sole model input, given a limited number of reliable stuttering samples and observations?*. In addition, this chapter will demonstrate the capabilities of convolutional, recurrent architectures and hybrid approaches, such as ConvLSTM, in detecting stuttering events directly from the speech signal using mel-scale features. Moreover, this chapter suggests and evaluates a novel SED model architecture that detects stuttering events directly from the speech signal. The model uses a log mel spectrogram as a sole acoustic feature and a 2D atrous convolutional network to learn spectral and temporal features representation. To rigorously check model robustness and performance, the model was evaluated on three stuttering datasets (UCLASS, SEP-28k and FluencyBank). *The model was published in the IEEE 13^{th} International Conference on Information*

and Communication Systems 2022 (Al-Banna, Edirisinghe & Fang 2022).

5.1. Introduction

Investigating the use of different ML approaches to create a robust **SED**, based on acoustic features that directly detect stuttering events from the speech signal, is challenging because the margin of accuracy of these models is approximately 45%. In SED , acoustic features differentiate between stuttering core behaviours, with most previous models using perceived acoustic features as a sole input model. As previously summarised in Table 3.1, there are existing models on stuttering event detection that investigate how acoustic processing techniques are leveraged with ML and DL. Most of these models are based solely on the acoustic features of the sound and do not rely on any other information, such as context, language and gender. Therefore, this chapter proposes and evaluates a new stuttering detection model architecture to directly detect the core behaviours from the signal. In addition, it shows the experiments conducted to evaluate the new model rigorously. Moreover, this chapter will evaluate state-of-the-art methods such as CNN, ConvLSTM and BI-LSTM on UCLASS and FluencyBank and compare the performance of these methods with that of the proposed model architecture.

5.2. Proposed model

Different architectures based on convolutional and recurrent neural networks were evaluated and tested to build the proposed model. Seven architectures were evaluated under the same experimental conditions, acoustic features and stuttering classes. These models include a 1D convolutional model with local pooling operations, a 1D convolutional model with dilations, a 2D convolutional model with local pooling, a 2D convolutional model without local pooling, and a 2D convolutional model with dilations. Additionally, a Bi-LSTM model and a CovLSTM model were also evaluated.

The proposed stuttering detection model was designed based on the atrous neural network that will be described in Section 5.2.1. As shown in Figure 5.1, the model first

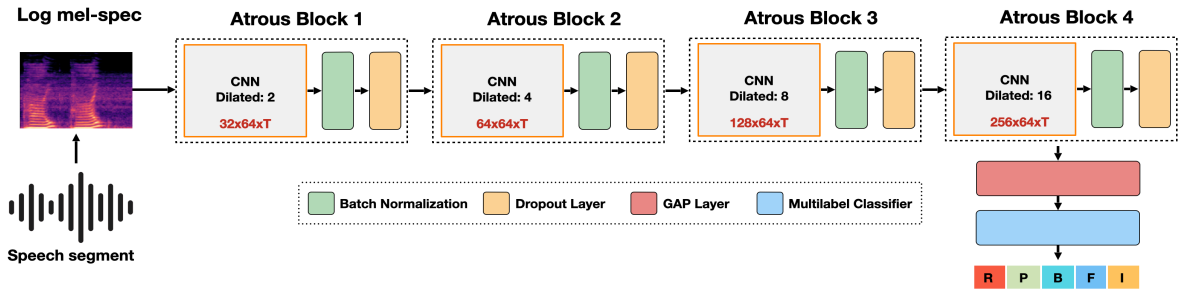


Figure 5.1: The model architecture consists of four atrous convolutional blocks. Each block contains a dilated CNN with different dilation rates, batch normalisation, and dropout. Global average pooling is applied to create one feature map for each corresponding event.

extracts a log mel-spectrogram feature from the signal. The mel-spectrograms are then passed to the atrous convolutional network, which is designed to learn spectral and temporal features. The network consists of four atrous convolution layers with different dilation rates of 2, 4, 8, and 16, respectively. Each layer is followed by batch normalisation and a dropout of 0.4 layers. The atrous block is used to maintain the dimensions of the feature maps to be the same size as the original model-spectrogram input using different dilation rates that increase the field of view exponentially, as illustrated in Figure 5.2. The final feature maps are passed to a global average pooling layer to create one feature map for each corresponding event. At the end of the network, a multi-label classifier with a sigmoid activation is employed to predict the score of each stuttering event.

5.2.1. Atrous CNN

A Convolutional Neural Network (CNN) (Arbib 1998) is a multi-layered feed-forward neural network that convolves a filter (kernel) of weights with a grid-like input data format (e.g., audio, video, images) in at least one of its layers (Goodfellow et al. 2016). A kernel is represented typically by a two-dimensional matrix (m, n) and is applied to the layer input by sliding and overlapping techniques. E.g., in image convolutional operation, each image pixel is multiplied by a kernel weights matrix to obtain new values for that pixel by sliding and overlapping the kernel over the pixel and its neighbours. These values will then be added to output a single value for each overlap. A

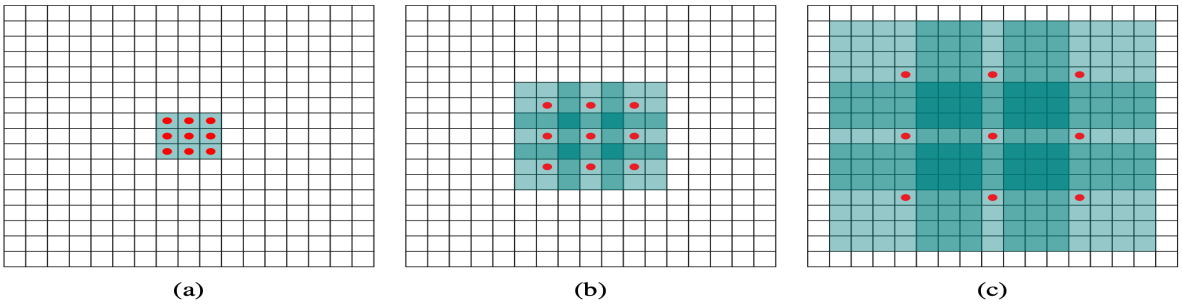


Figure 5.2: Shows the receptive field increases exponentially when using different dilation factors. d=1,d=2, and d=4. Yu & Koltun (2016)

convolutional operation is defined as

$$f(t) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n)$$

where I is a 2D input matrix, K is a kernel-convolution matrix with m and n dimensions. In the detection domain, CNNs can be employed as a feature extractor on different data types, such as audio, video, and text. It can successfully capture temporal and spatial dependencies in different data features such as shapes, gradients, and audio spectrum. This property makes it adaptable and robust for different computer vision and speech analysis problems (Habib et al. 2021). The output of the convolutional layer is a feature map $f(t)$. Usually, the feature map passes through a non-linear activation function, e.g. Rectified Linear Unit (ReLU); this function maintains the positive values and changes the negative values to zero. Typically, a pooling operation is employed after each convolutional. The pooling operation reduces the input size or the feature maps by applying a 2D kernel.

On the other hand, the dilated convolution module (Yu & Koltun 2016) is a module that uses holes ”atrous” in the convolutional operation to aggregate the multi-scale contextual information and increases the receptive field of these data without losing data resolution. The module was developed based on the atrous algorithm (Shensa 1992), which is generally used in the wavelet transformation. Using the dilation factor d as given in

$$f(t) = (F *_d k)(\mathbf{p}) = \sum_{\mathbf{s}+d\mathbf{t}=\mathbf{p}} F(\mathbf{s})k(\mathbf{t})$$

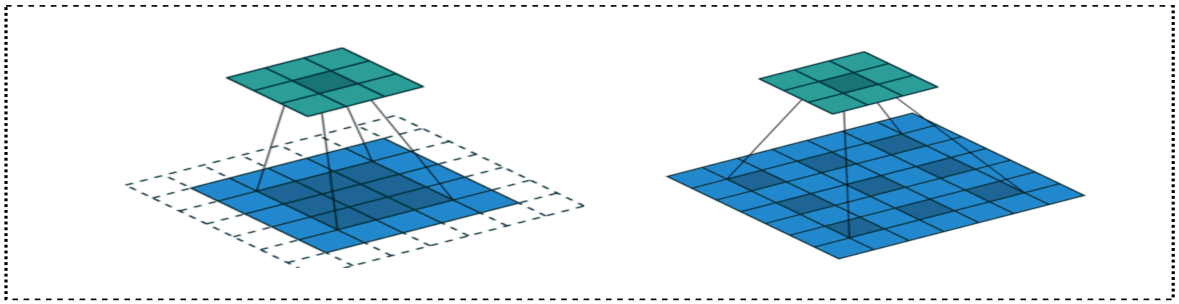


Figure 5.3: Show how the receptive field increase without losing the area coverage resolution. However, when d=1, the standard convolution will be maintained. Yu & Koltun (2016)

without pooling will exponentially increase the receptive field without losing the area coverage resolution. However, when d=1, the standard convolution will be maintained as illustrated in Figure 5.3.

5.2.2. Auditory-based spectral feature

The input of the proposed model is a log mel spectrogram. A spectrogram is a visual representation of the frequency over time. This acoustic feature can maintain the time and frequency of the signal in a single plot. Most of the DL approaches in disfluency classification and recognition use a spectrogram as a feature extraction method. In a spectrogram, the signal's energy is represented by different colours, the darker the colour, the lower the energy. A pre-emphasis filter of a 0.95 filter coefficient was applied for each audio segment to enhance the signal-to-noise ratio. The improved signal was divided into a short time frame with 64 frequency bins and $172, 130$ time frames. As a result, each audio segment is converted to a log mel-spectrogram of shape with 64×130 and 64×172 for FluencyBank and UCLASS, respectively. The following are the steps to convert the time-domain single to the log mel spectrogram.

Firstly, the pre-emphasis filter is applied to the waveform signal to enhance the signal-to-noise ratio and balance the frequency spectrum. This filter is applied to the time-domain signal (s) with α filter coefficient as

$$\gamma(t) = s(t) - \alpha s(t-1)$$

Secondly, the filtered signal $\gamma(t)$ is divided into a short time frame signal with a $25\ ms$ frame size and a $10\ ms$ frame stride. After framing, the hamming window function is applied to each frame as

$$W(n) = 0.54 - 0.46 \cos \frac{2\pi n}{N-1}$$

Thirdly, the Short-Time Fourier-Transform (STFT) is employed to each frame with N FFT, and the power spectrum is computed as

$$P_s = \frac{|FFT(x_i)|^2}{N}$$

Finally, the power spectrum converts to a mel-scale signal as

$$m = 2595 \log_{10}(1 + \frac{f}{700})$$

where 64 triangular filters are utilised to compute the filter banks on a mel-scale to the power spectrum to extract frequency bands.

5.3. Experiments

This section presents the experimental results of the proposed model for stuttering event detection. The model was trained and evaluated on two datasets on stuttering core behaviours and fluent class, UCLASS (Howell et al. 2009) and FluencyBank (Bernstein Ratner & MacWhinney 2018). Moreover, the experimental setup, evaluation metrics as explained in Chapter 2, and results of the experiments are utilised. Furthermore, a comparison of the proposed model performance with existing models and the implications and significance of the results in the context of stuttering detection are discussed.

5.3.1. Experiments

As presented in Table 5.1, four groups of experiments were conducted to evaluate the impact of various model architectures on the SED performance. Different architectures based on convolutional and recurrent neural networks were evaluated and

Table 5.1: Summary of experimental groups, model structures, and description of each model.

	Features	Model structure	Description
Group1			
	log mel-spectrogram(FluencyBank)	1D CNN with local pooling	The model consists of three 1D convolutional layers followed by local pooling.
	log mel-spectrogram(FluencyBank)	1D CNN with dilation	Three 1D dilated convolutional layers with a fixed stride and dilation rate 1,2,3.
Group2			
	log mel-spectrogram(FluencyBank)	2D with local pooling	The model consists of three 2D convolutional layers followed by local pooling.
	log mel-spectrogram(FluencyBank)	2D atrous CNN without pooling	The model consists of four atrous convolution layers with different dilation rates of 2, 4, 8, and 16, respectively
Group3			
	log mel-spectrogram(FluencyBank & UCLASS)	CNN+LSTM	The model consists of on 2D convolutional layers followed by a bidirectional LSTM layer with 64 units.
	log mel-spectrogram(FluencyBank & UCLASS)	BI-LSTM	A bidirectional LSTM layer with 64 units is then applied to the reshaped feature representation
	log mel-spectrogram(FluencyBank & UCLASS)	2D atrous CNN without pooling	The model consists of four atrous convolution layers with different dilation rates of 2, 4, 8, and 16, respectively
Group4			
	log mel-spectrogram(FluencyBank & SEP-28k)	CNN+LSTM	The model consists of on 2D convolutional layers followed by a bidirectional LSTM layer with 64 units.
	log mel-spectrogram(FluencyBank & SEP-28k)	2D atrous CNN without pooling	The model consists of four atrous convolution layers with different dilation rates of 2, 4, 8, and 16, respectively

demonstrated to build the proposed model. Seven architectures were evaluated under the same experimental conditions, acoustic features and stuttering events. These models include a 1D convolutional model with local pooling, a 1D convolutional model with dilation, a 2D convolutional model with local pooling, 2D convolutional model without pooling. Additionally, a Bi-LSTM model and a CovLSTM model were also evaluated.

5.3.2. Experiment (A): Effect of increasing the receptive field on SED performance using a 1D convolutional network.

Experiment aim

The main objective of this experiment is to investigate and evaluate the effect of increasing the field of view on the features map. Two baseline models will apply to achieve this objective. The first model consists of three 1D convolutional layers followed by local pooling, and the second model consists of three 1D dilated convolutional layers with a fixed stride.

Experimental procedure

In this experiment, two convolutional structures were created. The first model consists of three 1D convolutional layers with a kernel size = 3, stride = 1, and $32, 64, 128$ filters. Each layer is followed by local pooling, batch normalisation, dropout and ReLU activation function. The final feature map is then fed to a global pooling layer and dense

layer with sigmoid activation to predict the detection score for each class. All local pooling layers are replaced with dilation rates in the second model. A dilation rate of 1, 2, and 3 is added for each layer, respectively.

Experimental results

The F1 score results of each stuttering class on the FluencyBank dataset are shown in Table 5.2. As expected, the average F1 scores across all stuttering classes increased, by 4.02%, from 23.66% in the first model to 27.68% in the second model, when the pooling operation was replaced with dilation. In addition, the average recall increased by 2.39%, from 23.15% using local pooling to 25.54% using dilation. However, the F1 score and recall of the word repetition class did not change and remained zero in both models and surprisingly, the F1 score of the prolongation class decreased by 6.48% in the dilated model. Nevertheless, this finding is compatible with previous research (Sheikh et al. 2021) which argued that the repetition and prolongation might exceed the segment duration and cause those classes likely to be misclassified. Evidently, based on the experimental results using 1D convolution with dilation performs better than 1D convolution with local pooling, however, in this context, it could not be able to detect the main stuttering core behaviours.

Another interesting point is that the two models achieved more than 80% F1 on the fluent class, which may be because the model is trained on a sufficient number of observations of the fluent class that model learns only the fluent class. In addition, the average of UAR increased by 6%, and the average of EER decreased by 1% using 1D CNN model with dilation as reported in Table 5.3.

Conclusions

In this experiment, two convolutional structures were evaluated. The first model utilised local pooling, while the second used dilation rates. The results showed that the average F1 and recall scores across all stuttering classes, increased by 4.02% in the second model using dilation, from 23.66% to 27.68% F1 score and 2.39% recall score. In contrast, the F1 score of the prolongation class decreases, while word repetition is still challenging and undetectable. Moreover, the two models achieved more than

Table 5.2: Comparison of F1 and recall on the positive class of the designed 1D with pooling and 1D with dilation.

F1	B	F	I	P	S	W	Avg
1D CNN with pooling	10.53	81.56	21.91	26.43	1.50	0.00	23.66
1D CNN with dilation	24.13	80.81	33.86	19.95	7.35	0.00	**27.68**
Avg. recall on positive class	**B**	**F**	**I**	**P**	**S**	**W**	**Avg**
1D CNN with pooling	5.99	99.33	14.62	18.18	0.79	0.00	23.15
1D CNN with dilation	15.40	96.55	23.92	12.95	4.39	0.00	**25.54**

Table 5.3: Comparison of UAR and EER of the designed 1D CNN with pooling and 1D CNN with dilation.

Avg. UAR	B	F	I	P	S	W	Avg
1D CNN with pooling	57.33	58.30	53.83	65.59	34.93	37.14	51.19
1D CNN with dilation	65.06	56.29	62.44	63.39	63.36	37.14	**57.94**
Avg. EER	**B**	**F**	**I**	**P**	**S**	**W**	**Avg**
1D CNN with pooling	33.30	30.97	39.68	21.31	27.77	25.72	29.79
1D CNN with dilation	31.63	31.60	35.53	21.67	27.65	25.73	**28.97**

80% F1 on the fluent class, which may be because the model is trained on a sufficient number of observations of the fluent class, which make the model learns only the fluent class.

In addition, the first group of experiments demonstrated that the 1D CNN model with dilation detected the fluent class with an averag recall of 99.33% and 0.79%, and 0.00% for sound and word repetition, respectively. The results of evaluating the 1D CNN model with the dilation model suggested that its ability to detect stuttering events may be limited. In the next experiment, the 1D convolution will be replaced by 2D, and the dilation rate will be increased in time and mel-scale dimension.

5.3.3. Experiment (B): Effect of increasing the receptive field on SED performance using a 2D convolutional network.

Experiment aim

The main objective of this experiment is to evaluate the performance of SED and observe the impact of increasing the receptive field on both dimensions, time dimension and mel-scale. Therefore, all 1D dilated convolutional layers in the previous architecture described in Section 5.3.2 are replaced with 2D layers. Moreover, an extra 2D layer was added, and the dilation rate changed to 2, 4, 8, and 16, respectively.

Experimental procedure

The second experiment investigates the effect of increasing the receptive field in both time and frequency dimensions. Therefore two convolutional networks are created. The first network has the same structure as the previous 1D convolutional layers with local pooling; however, each 1D layer has been replaced with 2D convolutional layers. On the other hand, the second network consists of four atrous convolution layers with different dilation rates of 2, 4, 8, and 16, respectively. Each layer is followed by batch normalisation and a dropout of 0.4 layers. The final feature maps are passed to a global average pooling layer to create one feature map for each corresponding event. At the end of the network, a multi-label classifier with a sigmoid activation is employed

to predict the score of the stuttering events.

Experimental results

The F1 results of each stuttering event on the FluencyBank dataset illustrated in Table 5.4 show the average F1 score of 2D atrous network and 1D dilated network increased dramatically from 27.68% to 52.87%. In addition, the 2D atrous model surpasses all other models in all stuttering events except for the fluent class. However, it achieved a promising result with a 79.08% F1 score in the fluent class. In the case of using the 2D CNN approach with local pooling, the model outperformed the 1D dilated model in all aspects and fluent class. However, the atrous model outperformed the 2D with pooling with an 8% overall F1 score. As presented in Table 5.4, the atrous model surpasses all models regarding recall by 12%. In the atrous model, the detection of the sound and word repetition classes increased by 44.12% and 35.38% recall score compared to the 1D dilated model.

Moreover, the recall of interjection and block classes increased by 32.79% and 32.56%, from 15.40% to 47.96% for the block class and 23.92% to 56.71% for the interjection class using atrous model compared to the 1D dilated model. In addition, the recall of the prolongation class increased by 31%, from 12.95% to 43.52%. In contrast, the recall of the fluent class decreased by 18%, from 96.55% to 79.44%. To further analyse the evaluation results, Tables 5.6 and 5.7 are a comparison between 2D CNN and 2D CNN with dilation. The tables present the actual recall and the average of TP of SED on each stuttering event using the average across the 10-fold cross-validation. The results show that the dilated (atrous) convolution outperforms the 2D CNN with pooling by 13.21% average recall. However, the 2D model with pooling outperforms the proposed model in the fluent class, which may be because of the imbalanced nature of the training data. In addition to the improvement of average F1 scores, the atrous model achieved 64.26% and 29.07% UAR and EER, respectively, as presented in Table 5.5.

Table 5.4: Comparison of F1 and recall on the positive class of the designed 1D CNN with dilation, 2D CNN with pooling and 2D atrous CNN .

F1	B	F	I	P	S	W	Avg
1D CNN with dilation	24.13	80.81	33.86	19.95	7.35	0.00	27.68
2D CNN with pooling	38.43	**82.25**	51.07	40.06	36.55	30.04	46.40
2D Atrous CNN	**48.64**	79.08	**57.81**	**42.06**	**49.58**	**40.06**	**52.87**
Avg. recall on positive class	**B**	**F**	**I**	**P**	**S**	**W**	**Avg**
1D CNN with dilation	15.40	**96.55**	23.92	12.95	4.39	0.00	25.54
2D CNN with pooling	31.02	91.94	45.13	31.14	27.19	5.85	38.71
2D Atrous CNN	**47.96**	79.44	**56.71**	**43.52**	**48.51**	**35.38**	**51.92**

Table 5.5: Comparison of UAR and EER of the designed 1D CNN with dilation, 2D CNN with pooling and 2D atrous CNN.

Avg. UAR	B	F	I	P	S	W	Avg
1D CNN with dilation	65.06	56.29	62.44	63.39	63.36	37.14	57.94
2D CNN with pooling	61.06	68.56	64.69	63.41	67.18	60.06	64.16
2D Atrous CNN	61.96	66.15	66.34	62.58	65.74	62.81	**64.26**
Avg. EER	**B**	**F**	**I**	**P**	**S**	**W**	**Avg**
1D CNN with dilation	31.63	31.60	35.53	21.67	27.65	25.73	28.97
2D CNN with pooling	32.86	27.35	32.96	22.81	25.70	25.34	**27.84**
2D Atrous CNN	33.62	29.06	31.75	25.58	27.26	27.18	29.07

Table 5.6: Results of atrous model on stuttering events

Model	Event	No of observations	TP	Avg.recall
2D with Dilation(Atrous)	P	88	38.3	43.52
	B	137	65.7	47.96
	S	114	55.3	48.51
	W	106	37.5	35.38
	I	158	89.6	56.71
	F	284	225.6	79.44
	Avg			51.92

Table 5.7: Results of 2D CNN with pooling model on stuttering events

Model	Event	No of observations	Avg.TP	Avg.recall
2D CNN with pooling	P	88	27.6	31.14
	B	137	48.2	31.02
	S	114	30.5	27.19
	W	106	5.0	5.85
	I	158	72.9	45.13
	F	284	256.5	91.94
	Avg			38.71

Conclusions

In this experiment, two convolutional models, 2D with pooling and 2D with dilation (atrous), were evaluated and compared with the 1D dilated network. The results proved that the SED F1 score increased by 6.47% across all stuttering events as presented in Table 5.4. The F1 score of the sound and word repetition classes improved by 13.01% and 10.02%, respectively, using the atrous model compared to 2D with the pooling model. Moreover, the F1 score of interjection and block classes increased by 10.25% and 6.74%, respectively, from 38.43% to 48.64% for the block class and 51.07% to 57.81% for the interjection class. Furthermore, the F1 score of the prolongation class increased by 2%, from 40.06% to 42.06%. In contrast, the F1 score of the fluent class decreased by 3.17%, from 82.25% to 79.08%. The experiment concluded that the atrous convolutional model is suitable for stuttering events detection. The model F1 score increased by 25% compared to 1D convolution with dilation from 27.68% to 52.87%.

Additionally, the 2D atrous network outperforms the 2D with pooling with an 8% overall recall score. The results proved that increasing the receptive fields on time and mel-scale dimensions enhances the F1 and recall scores for SED. Furthermore, in the 2D atrous model, the sound and word repetition recall increased by 44.12% and 35.38% compared with the 1D dilated model, indicating the model's ability to learn stuttering core behaviours. In the next experiment, the SED performance will be evaluated on two datasets, FluencyBank and UCLASS. In addition, the following experiment will compare the performance of the proposed model with two stuttering detection methods, ConvLSTM (Lea et al. 2021), and BI-LSTM (Jouaiti & Dautenhahn 2022).

5.3.4. Experiment (C): Comparison of proposed model with LSTM and ConvLSTM on UCLASS and FluencyBank datasets.

Experiment aim

This experiment aims to compare the performance of the proposed model illustrated in Figure 5.1 with two stuttering detection models, ConvLSTM (Lea et al. 2021), BI-

LSTM (Jouaiti & Dautenhahn 2022) based on the same experimental setup, stuttering events and spectral features.

Experimental procedure

In this experiment, the repetitions are grouped in one class because the number of observations is too small in the UCLASS dataset. In addition, the models were retrained using weighted entropy and with FL.

Experimental results

To further analyse the proposed architecture, the model was compared with two stuttering detection methods, (Lea et al. 2021, Jouaiti & Dautenhahn 2022). Recall results of the proposed model against the UCLASS and FluencyBank datasets are shown in Table 5.8 indicate that the model outperforms the ConvLSTM and BI-LSTM on the prolongation class, with a recall of 46.8%and 47.6% on the UCLASS and FluencyBank, while ConvLSTM surpasses the model on the block and repetition classes in both datasets. Moreover, the model gained 6% and 4% margins on the UCLASS and FluencyBank datasets for the fluent class.

Furthermore, the F1 results reported in Table 5.9 show that the proposed model exceeds the results of ConvLSTM and Bi-LSTM on the prolongation class with an F1 of 52.0% and 44.5% on the UCLASS and FluencyBank datasets, respectively. In addition, the model gains 2.1% and 2.0% margins on the UCLASS and FluencyBank datasets for the fluent class. However, ConvLSTM outperformed the proposed model on the repetition class, with 1.5% and 5.3% margins in both datasets. BI-LSTM outperform the proposed model on the interjection class on FluencyBank dataset with a margin of 5%. In addition to the improvement of average F1 scores, the proposed model achieved 64.5% and 30.5% UAR and EER, respectively, as presented in Table 5.10.

Conclusions

This experiment compared the performance of the proposed model with two stuttering detection methods, ConvLSTM and BI-LSTM. The experimental results indicate that the proposed model outperforms the state-of-the-art methods on prolongation and flu-

Table 5.8: Experimental results on UCLASS and FluencyBank datasets

UCLASS	B	R	F	P	I	Avg.recall
Proposed model	43.2	35.1	**74.6**	**46.8**	59.0	**51.7**
2D with local pooling	**45.4**	41.0	69.3	33.1	12.7	40.3
CNN+LSTM	42.3	**42.1**	67.6	41.2	**62.7**	51.1
BI-LSTM	40.7	37.6	70.6	35.3	58.2	48.5
FluencyBank	B	R	F	P	I	Avg.recall
Proposed model	39.2	41.1	**89.3**	**47.6**	59.4	55.3
2D with local pooling	37.2	39.3	87.1	30.2	47.5	48.3
CNN+LSTM	**45.4**	**54.0**	77.1	45.2	**60.1**	**56.4**
BI-LSTM	43.1	46.7	83.3	42.1	55.7	54.2

Table 5.9: Experimental results on UCLASS and FluencyBank datasets

UCLASS	B	R	F	P	I	Avg.F1
Proposed model	46.3	37.4	**64.1**	**52.0**	35.5	**47.1**
2D with local pooling	45.1	**40.2**	62.0	40.6	16.5	40.9
CNN+LSTM	**48.1**	38.9	59.0	49.0	37.2	46.4
BI-LSTM	45.4	36.7	58.6	44.7	36.2	44.3
FluencyBank	B	R	F	P	I	Avg.F1
Proposed model	45.2	49.0	**84.3**	**44.5**	52.5	**55.1**
2D with local pooling	44.1	47.0	82.3	25.8	52.7	50.4
CNN+LSTM	40.6	**54.3**	79.0	42.3	53.3	53.9
BI-LSTM	44.2	45.0	81.4	41.0	**57.3**	53.8

Table 5.10: Experimental results on FluencyBank dataset

Avg. UAR	B	R	F	P	I	Avg
Proposed model	61.8	65.4	65.9	63.2	66.1	**64.5**
2D with local pooling	58.1	62.3	60.9	62.3	62.9	61.3
CNN+LSTM	58.4	58.8	58.0	65.9	60.3	60.3
BI-LSTM	58.6	62.8	63.4	55.4	63.6	60.7
Avg. EER	B	R	F	P	I	Avg
Proposed model	33.6	32.4	29.0	25.6	31.7	30.5
2D with local pooling	32.9	33.1	27.4	19.8	33.0	**29.2**
CNN+LSTM	40.5	40.8	40.9	33.4	39.4	39.0
BI-LSTM	58.6	33.6	27.6	22.2	33.4	35.1

ent classes on two datasets (UCLASS and FluencyBank). Furthermore, it achieved acceptable results in the block, interjection, and repetition classes. The following experiment will evaluate the model generalisation and robustness using cross datasets and an exclusive speaker test.

5.3.5. Experiment (D): Evaluate the model generalisation and robustness using cross datasets and an exclusive speaker test.

Experiment aim

This experiment aims to evaluate the model generalisation and robustness using cross datasets and an exclusive speaker test by focusing on FluencyBank and SEP-28k. In this group of experiments, different stuttering events samples were selected from the SEP-28k dataset to observe the behaviour of the proposed model in stuttering events with different data distributions.

Table 5.11: Evaluation of the model generalisation and robustness using cross datasets and an exclusive speaker test using FluencyBank without repetition grouping.

Model	UAR	F1 Score	Recall	EER
CNN	64.2	42.9	**55.6**	**27.9**
RNN	63.6	42.9	54.6	28.3
Proposed model	**64.3**	**52.9**	54.2	29.1
ConvLSTM	58.3	49.6	45.1	31.5

Experimental procedure

This group of experiments further evaluated the model generalisation and robustness using cross datasets and an exclusive speaker test, by focusing on FluencyBank and SEP-28k datasets. The SEP-28k contains 23 hours of stuttering podcasts, making it suitable to evaluate the model's performance. Three evaluation metrics were utilised to investigate the robustness of the model. In addition to the UAR, the F1 score and a recall of the minority class matrix were employed. The training process repeated on the five stuttering events and fluent class on FluencyBank using 10-fold cross-validation.

Experimental results

The model generalisation and robustness evaluation using cross datasets and an exclusive speaker test using FluencyBank without repetition class grouping, as shown in Table 5.11, shows that the proposed model still outperforms the methods by 52.9%, achieving acceptable scores of 64.3% and 54.2% in UAR and the recall, respectively. Surprisingly, the ConvLSTM model surpasses the proposed model by 2% of the recall. Therefore a set of experiments was conducted to evaluate the ConvLSTM and the proposed model against 500, 1000, 2000, and 10000 random samples from SEP-28k with different data distributions. The experiments show that the F1 score of SED significantly worsens by more than 20% in both models, as reported in Table 5.12.

Table 5.12: Evaluation of the ConvLSTM and proposed model against random samples from SEP-28k

	500 Sample				1000 Sample				2000 Sample				10000 Sample			
Model	**UAR**	**F1**	**Recall**	**EER**	**UAR**	**F1**	**Recall**	**EER**	**UAR**	**F1**	**Recall**	**EER**	**UAR**	**F1**	**Recall**	**EER**
CovLSTM	54.00	32.00	25.39	39.40	54.00	33.00	25.16	38.75	55.00	33.00	26.00	38.80	**54.00**	31.00	**24.00**	**35.40**
Proposed	**55.00**	**33.00**	**29.18**	**30.64**	**56.00**	**34.00**	**27.91**	**29.74**	**56.00**	**36.00**	**29.40**	**29.80**	53.00	**31.00**	23.9	37.80

Conclusions

The findings show that the performance of SED significantly declines by more than 20% F1 score for different model architectures using the exclusive speaker test. These experiments concluded that the perceived acoustic features, such as mel-scale features, as a sole input model, can provide valuable information for SED but may not be sufficient for accurately detecting stuttering events.

5.4. Summary and findings

The findings of this chapter aim to answer the following research question *To which level can a robust stuttering events detection model be created, based on perceived acoustic features as a sole model input, given the limited number of reliable stuttering samples and observations?*. In addition, this chapter demonstrated the capabilities of convolutional, recurrent architectures and hybrid approaches, i.e. ConvLSTM, in detecting stuttering events directly from the speech signal using perceived features, i.e., mel-scale features. Moreover, this chapter suggested and evaluated a novel SED model architecture that detects the stuttering events directly from the speech signal. The model is based on a log mel spectrogram and 2D atrous convolutional network designed to learn spectral and temporal features.

In order to answer the research question, this chapter investigated the potential of using perceived acoustic features, i.e. mel-scale features, as the sole model input to develop a robust SED under data reliability constraint. One of the challenges in developing SED is the need for more reliable stuttering samples and observations in model training. The finding of experiment D evaluates the model generalisation

and robustness using cross datasets and exclusive speaker test shows that the performance of SED significantly worsens by more than 20% in terms of F1 score for different model architectures; this drop in performance may have significant implications for the model's application in real-world scenarios, highlighting the importance of additional features to capture more information of the speech segment and speaker variability. In addition, the data reliability issue may be better to be resolved. Moreover, the mel-scale represents the frequency domain of the stuttering speech segment since it better approximates the human auditory system's sensitivity to different frequencies. The window size and overlapping parameters are essential in the feature extraction process using mel-scale and may affect model performance.

While perceived acoustic features such as mel-scale features can provide valuable information for SED, they may not be sufficient for accurately detecting stuttering events. For instance, including contextual information, pre-trained models, and time-domain features may help improve the model's accuracy by guiding the detection process. Similarly, additional features such as prosodic features (e.g., intonation, rhythm, and fundamental frequency or phonetic features (e.g., phoneme and syllable-level features) can improve model generalisation and robustness. Therefore, incorporating a diverse range of features can help enhance the accuracy and robustness of the speech recognition model, leading to improved performance in real-world scenarios.

The issue of data reliability is a critical consideration in the development of a robust SED. Given the limited number of reliable class samples and observations, there is a risk of introducing bias into the model, which can significantly impact its performance. The first group of experiments demonstrated that the 1D CNN model with dilation detected fluent class with an average recall of 99.33%; however, the average recall for sound and word repetition were 0.79% and 0.00%, respectively. The results of evaluating the 1D CNN model with the dilation model suggest that its ability to detect stuttering events may be limited. Specifically, the observed performance indicates that the model is primarily learning to detect fluent speech segments while struggling to identify and differentiate stuttering events accurately.

These findings suggest a need for further investigation on data reliability and po-

tential model improvements to enable it to learn and detect stuttering events more effectively. By addressing these limitations, the model may be better trained to provide more reliable and accurate stuttering event detection.

Regarding demonstrating the capabilities of convolutional, recurrent architectures and hybrid approaches, the previous experiments argued that 2D atrous neural networks boost the model F1 score by 25% compared to 1D convolution with dilation from 27.68% to 52.87%. Moreover, the 2D atrous network outperforms the 2D with pooling with an 8% overall F1 score. The results proved that increasing the receptive fields on time and mel-scale dimensions enhances the detection rate for SED. Furthermore, In the 2D atrous, the sound and word repetition detection F1 score increased by 44.12% and 35.38% compared with the 1D dilated model, indicating the model's ability to learn stuttering core behaviours.

6

The Impact of Stuttering Event Representation on Detection Performance

The findings of the previous chapter suggest a need for further investigation of data reliability and potential model improvements to learn and detect stuttering events more effectively. A model may be better trained to provide more reliable and accurate stuttering event detection by addressing these limitations. Although the perceived acoustic features can provide valuable information for SED, they may not be sufficient for accurately detecting stuttering events. For instance, contextual information, pre-trained models, and time-domain features may help improve SED performance.

Therefore, this chapter will answer the research question **RQ-2**: *What is the impact of stuttering event representation on detection performance?* . At the same time, the chapter will evaluate the performance of SED and observe the impact of applying ASR pre-trained features on each stuttering event. Moreover, different groups of experiments will evaluate the impact of three time-domain features, such as Zero Crossing Rate (ZCR), Spectral Flux Onset Strength Envelope (SFO) and Fundamental Frequency (FF) features, on SED performance. *Part of this chapter was published in the Journal of Information & Knowledge Management* (Al-Banna, Edirisinghe, Fang & Hadi 2022).

6.1. Introduction

Previous research have reported promising results in applying traditional ML to automate SED. However, prior studies have focused on small datasets generated by a limited number of speakers, specific stuttering events, and spectral features only. Pálfy (2014) gained 98% accuracy for SED performance, while the performance worsened by 47% in (Sheikh et al. 2021) when it was applied to 138 of UCLASS speakers; this significant reduction in accuracy reveals potential research gaps. Several factors influence SED performance, like any ML detection task, the nature of the data used, the number of observations and samples, feature representation and the train/test split (see **Chapter 3**).

This chapter will rigorously investigate the effective use of eight common ML classifiers on two relatively large-scale publicly available datasets (FluencyBank and SEP-28k) to automatically detect stuttering events using multiple objective metrics, accuracy, recall, precision, and F1 score. The contributions in this chapter are: (1) Evaluate the detection performance on small and large-scale datasets for six stuttering events. (2) Investigate the effect of spectral and temporal representation on the stuttering detection task. (3) Examine the enhancement of stuttering detection using a pre-trained ASR model.

6.2. Experiments

This section presents the experimental results for different ML architectures. These architectures were trained and evaluated on two datasets on stuttering core behaviours, SEP-28k (Lea et al. 2021) and FluencyBank (Bernstein Ratner & MacWhinney 2018). In addition, this section investigates the impact of spectral features on detection performance with different scaling techniques and the impact of spectral features on detection performance with a single-label balanced dataset. Furthermore, it compares the effect of ASR and temporal features on detection performance.

6.2.1. Experiment (A): Impact of spectral features on detection performance with mini-max scaling

Experiment aim

This experiment aims to rigorously investigate the effective use of eight well-known traditional ML classifiers on two published available datasets (FluencyBank and SEP-28k) to automatically detect stuttering events using multiple objective metrics. Support Vector Machine (SVM) is a supervised ML technique used for SED; the primary goal of SVM is to identify a hyperplane within an N-dimensional space to classify stuttering events. This experiment uses linear and a Radial Basis Function (RBF) (Cortes & Vapnik 1995, Wang et al. 2004). Moreover, Random Forest (RF) and Decision Tree (DT) (Breiman 2001, Salzberg 1994) are evaluated in this experiment; the RF is an ensemble ML technique that combines the predictions from multiple DTs to produce the forest. Gaussian Naïve Bayes (GNB) (Manning et al. 2008) is another supervised ML algorithm employed, the GNB model assumes that the MFCC coefficients follow a Gaussian distribution. In addition, the k-Nearest Neighbour (kNN) (Pedregosa et al. 2011) model is trained and evaluated in a supervised learning approach, where stuttering event labels are comprised. Furthermore, AdaBoost and Quadratic Discriminant Analysis (QDA) (Hastie et al. 2009, Pedregosa et al. 2011) classifers are trained and tested in this experiment. AdaBoost is an ensemble learning technique used to evaluate SED by combining the predictions of multiple weak learners, while QDA, which is a variation of Linear Discriminant Analysis (LDA), allows for more flexible modelling of stuttering events distributions, by considering quadratic relationships between MFCC coefficients. The approaches are evaluated with the hyperparameters listed in Table 6.1 and using 40 MFCC coefficients.

Experimental procedure

This experiment utilised a quasi-logarithmic frequency scale of MFCC, as described in Algorithm 1, where each 3-second segment with a single label is downsampled to $8\ KHz$. The downsampled time-domain segment was then converted into a frequency domain using the STFT with hop length with ($0.010\times$sampling rate) and ($0.090\times$number

Table 6.1: Hyperparameters of eight classifiers

Classifiers	Hyperparameters
SVM with RBF kernel	C=1.0, kernel='rbf', gamma='scale', max_iter=-1
SVM with a linear kernel	C=1.0, loss='squared_hinge', max_iter=1000, dual=True
DT	criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1
AdaBoost	n_estimators=50, learning_rate=1.0, algorithm='SAMME.R'
kNN	n_neighbors = 5, weights = 'distance', algorithm = 'brute', eaf_size = '30', n_jobs=4
RF	n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, max_features='auto'
QDA	priors=None, reg_param=0.0, store_covariance=False, tol=1.0e-4, store_covariances=False
GNB	priors=None, var_smoothing=1e-9

of FFT); the algorithm results in a 40 MFCC mean vector representations for each stuttering segment. The feature vector was then scaled using a MinMax scaler defined as

$$x\_scaled = \frac{(x - x\_min)}{(x\_max - x\_min)}$$

where x_scaled represents the normalised value of the MFCC coefficient, x is the original MFCC coefficient, x_min is the minimum value of the MFCC coefficient in dataset, and x_max is the maximum value of the MFCC coefficient across dataset. The MinMax scaler thus normalises the vector features into values in a range of 1 and 0. 10-fold cross-validation is used to evaluate the experiment results.

Experimental results

Table 6.2 and Table 6.3 show the evaluation results on the SEP-28k and FluencyBank datasets, the RF classifier performed best according to the prediction accuracy evaluator, and the DT classifier resulted in the worst performance. In particular, in terms of prediction accuracy, the RF outperformed, SVM with an RBF kernel,SVM with a linear kernel, kNN, AdaBoost, QDA, GNB, and DT, against FluencyBank dataset, by 0.48%, 2.89%, 5.06%, 5.43%, 6.99%, 13.75%, and 14.84%, respectively. On the SEP-28k dataset, the RF outperformed SVM with an RBF kernel, SVM with a linear kernel, AdaBoost, kNN, QDA, GNB, and DT by, 0.37%, 1.54%, 1.73%, 5.02%, 5.37%, 8.14%, and 17.29%, respectively.

Recall results of the eight classifiers against the SEP-28k and FluencyBank datasets are shown in Table 6.2 and Table 6.3. The RF achieves the best performance, with

Algorithm 1 Compute MFCC coefficients

procedure getMFCC
    S[t] $\leftarrow$ 3 seconds downsampled time domain *signal*
    S[t] $\leftarrow$ S[0:int(3 * 8000Khz)]
    $\alpha \leftarrow$ 0.95
    $\gamma(t) \leftarrow$ S(t) $- \alpha S(t-1)$
    Framing signal to **N** frames with a 90 ms frame size and 10 ms frame stride
    Apply hamming window function
    $W(n) \leftarrow 0.54 - 0.46 \cos \frac{2\pi n}{N-1}$
    Compute power spectrum for each frame x
    $P_s \leftarrow \frac{|FFT(x_i)|^2}{N}$
    Covert the power spectrum to mel-scale
    $m \leftarrow 2595 \log_{10}(1 + \frac{f}{700})$
    Apply 40 triangular filters are applied as filter banks on a mel-scale
    Apply Discrete Cosine Transform on filter-banks
    $mfcc \leftarrow$ dct(filterbanks)
    $coefficients \leftarrow$ mean(transpose(mfcc,40)) .
end procedure

Table 6.2: Results of eight classifiers on FluencyBank

Classifiers	accuracy	recall	F1	UAR	EER
SVM with RBF kernel	49.82%	50.00%	41.00%	55.29	16.79
SVM with linear kernel	47.41%	47.00%	34.00%	51.04	17.83
DT	35.46%	35.00%	36.00%	54.25	22.41
AdaBoost	44.87%	43.00%	39.00%	54.25	18.80
kNN	45.24%	44.00%	43.00%	57.69	18.63
RF	**50.30%**	**50.00%**	42.00%	56.47	**16.71**
QDA	43.31%	44.00%	**44.00%**	**59.23**	19.20
GNB	36.55%	35.00%	41.00%	53.70	22.09

Table 6.3: Results of eight classifiers on SEP-28k

Classifiers	accuracy	recall	F1	UAR	EER
SVM with RBF kernel	49.98%	49.00%	35.00%	50.00	17.01
SVM with linear kernel	48.81%	49.00%	32.00%	50.52	16.84
DT	33.06%	33.00%	33.00%	52.77	22.05
AdaBoost	48.62%	49.00%	34.00%	50.55	17.10
kNN	45.33%	44.00%	**42.00**%	**56.69**	17.90
RF	**50.35%**	**50.00%**	36.00%	51.72	**16.58**
QDA	44.62%	41.00%	41.00%	55.74	18.24
GNB	42.21%	49.00%	35.00%	51.60	19.49

a recall of 50% and 50% on the SEP-28k and FluencyBank datasets, respectively. The DT obtains the lowest value among the eight classifiers, with a recall of 33% and 35% on the SEP-28k and FluencyBank datasets, respectively.

Moreover, the F1 score results of the six classifiers against the SEP-28k and FluencyBank datasets are shown in Table 6.2 and Table 6.3. The QDA classifier achieves the best performance, with an F1 score of 44% on the FluencyBank dataset. kNN achieves the best performance, with an F1 score of 42% on the SEP-28k dataset. The SVM with a linear kernel classifier obtains the lowest value among the eight classifiers, with an F1 score of 32% and 34% on the SEP-28k and FluencyBank datasets, respectively.

Conclusions

This experiment aims to investigate the effective use of eight well-known traditional ML classifiers. This group of experiments has confirmed that the RF is the best classifier when accuracy is considered. The RF constructs multiple decision trees during training, enhancing detection results. Furthermore, The QDA classifier performs best, with an F1 score of 44% on the FluencyBank dataset. The kNN classifier performs best, with an F1 score of 42% on the SEP-28k dataset. This experiment also proved that the DL methods used in the previous chapter perform better than traditional machine learning models on the FluencyBank dataset, with 56.4% and 55.3% using ConvLSTM and atrous neural networks, respectively. The next group of experiments will evaluate the performance of the traditional ML classifiers on each stuttering event with mean normalisation scaling.

6.2.2. Experiment (B): Impact of spectral features on detection performance for each stuttering event with mean normalisation

Experiment aim

This experiment aims to evaluate the performance of each stuttering event using six classical machine learning algorithms using mean normalisation for MFCC vector ex-

tracted based on Algorithm 1 on two large-scale datasets. The experiments were conducted to observe the impact of spectral features on detection performance for each stuttering event with mean normalisation.

Experimental procedure

In this experiment, the mean normalisation defined as

$$MFCC_{norm}[i] = MFCC[i] - \frac{1}{N}\sum_{n=1}^{N} MFCC[n]$$

where $MFCC_{norm}[i]$ is the normalised MFCC feature for the i-th coefficient, $MFCC[i]$ is the original MFCC feature for the $i-th$ coefficient, N is the total number of coefficients in the MFCC feature vector.

The mean normalisation is applied on the MFCC feature vector. The experiments were repeated according to the previous experimental settings, and the accuracy, detection rate and F1 score for each stuttering event were observed.

Experimental results

The accuracy results presented in Table 6.4 indicate that the QDA performed best on the FluencyBank dataset, while the RF classifier outperformed other classifiers on the SEP-28k. The accuracy for imbalanced datasets is not a suitable evaluation method because the distribution of the observations and samples in stuttering speech is inconsistent among the classes. e.g., the number of samples belonging to the fluent class is much higher than in other classes. In contrast, block, prolongation, and sound repetition classes in both datasets are minority classes. Therefore, the recall (detection rate) and F1 score for each stuttering event was observed as tabulated in Table 6.5 and Table 6.6.

Recall results of the six classifiers against the SEP-28k and FluencyBank datasets are shown in Table 6.5. The RF performs best, with a recall of 42% and 36% on the SEP-28k and FluencyBank datasets, respectively. The DT classifier indicates the lowest recall among the six classifiers, with a recall of 20% and 27% on the SEP-28k and FluencyBank datasets, respectively.

Table 6.4: Accuracy results of Experiment 2

Classifier	FluencyBank	SEP-28k
kNN	37.00%	42.34%
DT	36.00%	30.09%
RF	43.38%	**49.79%**
GNB	28.91%	40.06%
AdaBoost	28.44%	47.59%
QDA	**43.65%**	41.06%

In addition to the recall, the F1 score is vital in stuttering detection. The F1 score results of the six classifiers against the SEP-28k and FluencyBank datasets are shown in Table 6.6. The kNN classifier achieves the best performance, with an F1 score of 33% on the FluencyBank dataset and an F1 score of 27% on the SEP-28k dataset. However, the AdaBoost classifier obtains the lowest value among the six classifiers, with an F1 score of 21% and 13% on the SEP-28k and FluencyBank datasets, respectively. In general, the performance of SED is low across all classifiers, and all classifiers struggled to detect stuttering core behaviour for two main reasons. Firstly, each speech utterance is labelled with one label without handling the agreement problem in both datasets. Secondly, the distribution of the samples in stuttering speech is inconsistent among the classes.

Detecting the block class is challenging in both datasets. The average F1 score across all classifiers is 0.085% and 0.056% on FluecyBank and SEP-28k, respectively, which may be due to two reasons. The first reason is the agreement between raters in the annotation process on block class where the agreement was 11%; since the raters are not SLPs (Barrett et al. 2022, Lea et al. 2021). In the annotation process, it is difficult to distinguish between the block and other events. The second reason is the nature of the block, which is spectral and temporal in nature, and it may overlap with other stuttering events, such as sound repetition and prolongation. In most of the previous research, the block class was discarded to enhance the detection per-

Table 6.5: Recall results on the FluencyBank and SEP28-K datasets

FluencyBank

	P	B	S	W	I	F	Avg
kNN	0.29	0.05	0.24	0.08	0.32	**0.54**	0.25
DT	0.26	0.14	0.27	0.16	0.32	0.49	0.27
RF	0.42	**0.25**	**0.46**	0	**0.55**	0.47	**0.36**
GNB	0.33	0.02	0.27	**0.25**	0.40	0.46	0.29
AdaBoost	**0.47**	0	0.25	0	0.42	0.47	0.27
QDA	0.35	0.08	0.39	0.16	0.44	0.51	0.32
Avg	**0.35**	**0.09**	**0.31**	**0.11**	**0.41**	**0.49**	**0.29**

SEP28-K

	P	B	S	W	I	F	Avg
kNN	0.25	**0.15**	0.27	0.21	0.28	**0.57**	0.29
DT	0.13	0.04	0.14	0.14	0.2	0.54	0.20
RF	**0.46**	0	**0.57**	**0.75**	**0.32**	0.42	**0.42**
GNB	0.18	0.05	0.11	0.11	0.24	0.52	0.20
AdaBoost	0.25	0	0.16	0.14	0.24	0.5	0.22
QDA	0.28	0.11	0.26	0.18	0.30	0.56	0.28
Avg	**0.26**	**0.06**	**0.25**	**0.26**	**0.26**	**0.52**	**0.27**

Table 6.6: F1 results on the FluencyBank and SEP28-K datasets

FluencyBank							
	P	**B**	**S**	**W**	**I**	**F**	**Avg**
kNN	0.24	0.06	0.22	0.6	0.3	0.54	**0.33**
DT	0.21	**0.19**	0.26	**0.16**	0.32	0.5	0.27
RF	0.15	0.11	0.28	0	**0.41**	**0.61**	0.26
GNB	0.15	0.04	0.33	0.05	0.19	0.53	0.22
Adaboost	0.25	0	0.13	0	0.29	0.6	0.21
QDA	**0.28**	0.11	0.34	0.14	0.45	0.53	0.31
AVG	**0.21**	**0.09**	**0.26**	**0.16**	**0.33**	**0.55**	**0.27**
SEP28-K							
	P	**B**	**S**	**W**	**I**	**F**	**Avg**
kNN	**0.23**	**0.1**	**0.21**	0.15	0.27	0.64	**0.27**
DT	0.14	0.05	0.13	0.14	0.21	0.53	0.20
RF	0.06	0.06	0.09	0.05	0.12	**0.68**	0.18
GNB	0.07	0.04	0.05	0.10	0.18	0.62	0.18
Adaboost	0.01	0.0	0.04	0.01	0.07	0.66	0.13
QDA	0.21	0.09	**0.21**	**0.18**	**0.23**	0.63	0.26
AVG	**0.12**	**0.06**	**0.12**	**0.11**	**0.18**	**0.63**	**0.20**

formance. The previous research considered being silent for more than $350\ ms$ as a block. Utilising other time-domain features, such as fundamental frequency, may enhance the detection rate for this event.

It can be witnessed that the sound repetition and prolongation F1 score decreased dramatically by 14% and 9%. i.e., in the counting procedure in SSI-3, the prolongation time is computed, and it considers one of the core behaviours of stuttering, and this event should be addressed in the SED task.

Despite the good agreement in the annotation process for the Interjection class, the average F1 score for Interjection decreases by 15%. This may be because PWS use these interjections between normal speech, which may overlap with the fluent class. In addition, these interjections are varied in English and other languages, such as Arabic. Therefore, Lea et al. (2021) claimed that using ASR may enhance the detection rate for this event. However, according to our observation, most of the annotated interjections in UCLASS, FluencyBank and SEP-28k datasets are ”UM, AH”. Therefore using MFCC on these datasets may achieve good results.

The word repetition F1 score decreases slightly by 5%, which may be because of the good agreement with 67% in the annotation process since normal raters easily capture this event. However, the F1 score increased by 8% from 55% to 63% in SEP-28k and FluencyBank, respectively. In addition, the ERR results presented in Table 6.7 indicate that the RF performed best on the FluencyBank and SEP-28k.

Conclusions

This experiment evaluated and compared the performance of each stuttering event using six classical machine learning algorithms using mean normalisation for MFCC on FluencyBank and SEP-28k datasets. The experiment results argued that the performance in the context of the F1 score of the prolongation class decreased in SEP-28k by 9%, the block class declined by 3%, while the sound and interjection classes dramatically declined by 14% and 15%, respectively, the word repetition decreased by 5%, and the most exciting point that fluent class increased by 8% of F1 score that because the fluent class has a good agreement in the annotation process and the

Table 6.7: UAR and EER results on the FluencyBank and SEP28-K datasets

FluencyBank	Avg.UAR	Avg.EER
kNN	0.56	0.19
DT	0.57	0.20
RF	0.56	0.**16**
GNB	0.55	0.17
Adaboost	0.54	0.21
QDA	**0.57**	0.20
SEP28-K	Avg.UAR	Avg.EER
kNN	0.55	0.19
DT	0.53	0.22
RF	0.52	**0.17**
GNB	0.50	0.17
Adaboost	0.51	0.19
QDA	**0.55**	0.18

previous models learnt only the fluent class. The following experiment will investigate the effect of spectral features on SED. The imbalanced nature of stuttering will handle by excluding the Fluent class.

6.2.3. Experiment (C): Impact of spectral features on detection performance with single-label balanced dataset

Experiment aim

This experiment aims to evaluate the performance of each stuttering event by resolving the imbalance issue by excluding the fluent class and focusing on stuttering core behaviours using a common deep learning approach in stuttering detection. The experiment assumed that solving the imbalance issue may increase the performance of SED for each stuttering event.

Experimental procedure

This experiment uses a simple shallow ConvLSTM deep neural network to detect five stuttering events, as illustrated in Figure 6.1. The network consists of a 1D convolutional layer with a kernel size of =3, stride =1 and 256 filters. This layer is followed by batch normalisation, dropout of 0.1 and Rectified Linear Unit (ReLU). The extracted features are permutated and passed to a BLSTM layer with 64 units and a time distribution layer to learn the temporal relation between the encoded features. After batch normalisation and dropout, the final feature map is fed to a 1D global pooling layer and dense layer with softmax activation to predict the detection score for each class.

Experimental results

The average F1 score of each stuttering event on the SEP-28k dataset using 10-fold cross-validation are shown in Table 6.8. The interjection class achieves the best performance with a 31.4% F1 score, and the block and prolongation classes obtain the lowest F1 score with 15.1% and 15.5%. The repetition of sound and word classes reaches 22.9% and 28.9% F1 score, respectively, as reported in Table 6.9, the F1 score of SED increased approximately by 11% across all stuttering events. In addi-

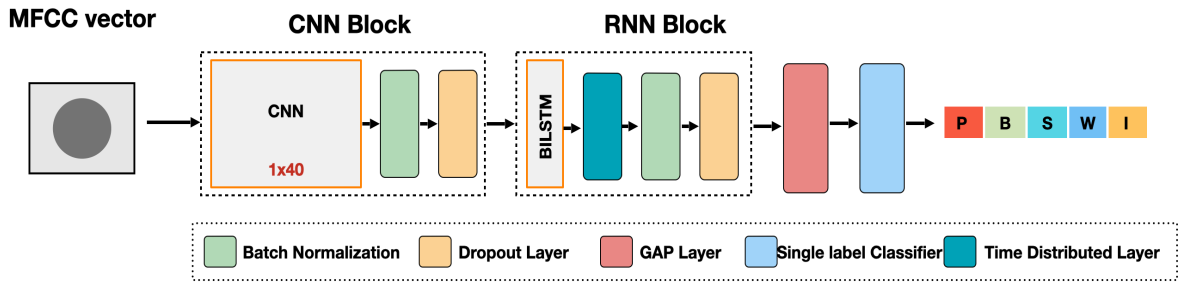


Figure 6.1: Simple shallow ConvLSTM deep neural network to detect five stuttering events.

Table 6.8: F1 results on SEP28-K dataset

Event	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Avg
P	0.136	0.178	0.148	0.141	0.146	0.156	0.184	0.201	0.106	0.155	**0.155**
B	0.164	0.148	0.116	0.164	0.173	0.140	0.149	0.123	0.139	0.197	**0.151**
S	0.287	0.185	0.24	0.215	0.23	0.184	0.278	0.206	0.231	0.232	**0.229**
W	0.280	0.302	0.287	0.310	0.279	0.295	0.296	0.290	0.286	0.270	**0.289**
I	0.308	0.329	0.296	0.337	0.321	0.319	0.300	0.294	0.310	0.325	**0.314**

tion, the F1 score of word repetition increased by 18.0%, and the prolongation class was the least with 4%. Regarding the other stuttering events, the F1 score of the block class increased by approximately 9%, from 6% to 15%, while the sound repetition increased by 11%, from 12% to 23%. In addition to the improvement of average F1 scores, the model achieved 54.9% and 24.0% UAR and EER, respectively, as presented in Table 6.10.

The main observations from these results indicate that using MFCC as an exclusive feature, is not sufficient to improve SED's performance. This may be because of the diverse variations in speech rate and differences in vocal tracts between speakers in SEP-28k dataset since we have 383 YouTube podcasts. Furthermore, the findings mentioned in the previous experiment may affect the performance results and encourage us to investigate more speech features. Therefore, employing extra temporal or pre-trained ASR features may enhance the SED performance. As noticed, detecting bock and prolongation classes is still challenging in SEP-28k dataset due to the data reliability of those events in the annotation process.

Table 6.9: Comparison between the average F1 score of all classifiers in the previous experiment and the average F1 score of SED in experiment C

	P	B	S	W	I	Avg.F1
Experiment B	0.120	0.060	0.120	0.110	0.180	0.120
Experiment C	0.155	0.151	0.229	0.289	0.314	0.228

Table 6.10: UAR and EER results on SEP28-K dataset

	P	B	S	W	I	Avg
UAR	0.524	0.525	0.559	0.558	0.578	0.549
EER	0.260	0.253	0.169	0.276	0.242	0.240

Conclusions

The above experimental results show that handling the imbalance problem and focusing on stuttering core behaviours in SEP-28k improves the SED average F1 score by 11% across all stuttering events. In addition, the detection of word repetition increased by 18.0%, and the prolongation was the least at 4% F1 score. Regarding the other stuttering events, the detection performance of the block class increased by approximately 9%, from 6% to 15%, while the sound repetition increased by 11%, from 12% to 23%. In the next experiment, we will investigate employing pre-trained ASR features in SED.

6.2.4. Experiment (D): Impact of pre-trained ASR features on detection performance

Experiment aim

The main objective of this experiment is to evaluate the performance of SED and observe the impact of applying ASR pre-trained features on each stuttering event. The investigation is based on the assumption that using ASR may enhance the detection

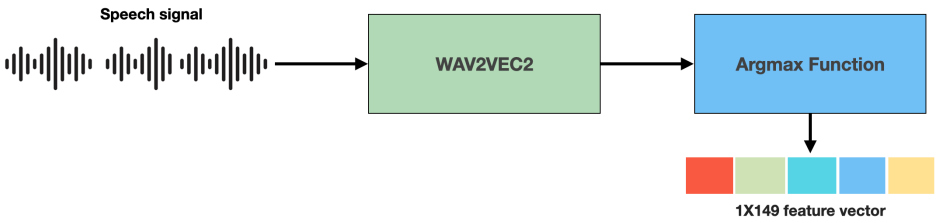


Figure 6.2: The WAV2VEC2 feature extraction diagram.

of word repetition and Interjection classes.

Experimental procedure

In this experiment, a bimodal of ConvLSTM fusion was created with multiple features. The main objective of this model is to integrate different features into a concatenated discriminative feature vector to enhance the performance of SED. Therefore, the extracted 40 MFCC described in the Algorithm 1 and 149 contextual vectors extracted from the Wav2vec2 pre-trained (Baevski et al. 2020) model, as illustrated in Figure 6.2 are utilised in this experiment. Wav2vec2 is a family of models for learning the representation of audio data. WAV2Vec2 model employs mask language modelling MLM of PERT (Devlin et al. 2019) on audio data, which involves training a model to predict masked (or ”hidden”) segments.

The same ConvLSTM architecture described in the previous experiment and the same experimental settings, as illustrated in Figure 6.1, are used in this experiment. Unlike the previous experiment, the outputs of these models are finally fused and input to a fully connected layer with a softmax to predict the score of the stuttering events.

Experimental results

The average F1 score results of each stuttering event on the SEP-28k dataset using 10 folds are shown in Table 6.11. As expected, the interjection and word repetition classes achieve the best F1 score with 39% and 37%, respectively, while block and sound repetition classes obtain the lowest F1 score with 25% and 26%. In addition, the F1 score of the Prolongation class was 27%. To investigate the effect of ASR feature representation on each stuttering event, the provided Table 6.12 compares the F1 score of SED in the previous experiment using MFCC only with the SED F1

Table 6.11: F1 results of experiment four on the SEP28-K datasets

Event	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Avg
P	0.26	0.29	0.32	0.24	0.25	0.28	0.27	0.30	0.26	0.26	**0.27**
B	0.23	0.25	0.22	0.25	0.28	0.26	0.26	0.28	0.25	0.25	**0.25**
S	0.24	0.29	0.28	0.16	0.29	0.25	0.22	0.28	0.26	0.28	**0.26**
W	0.38	0.34	0.34	0.37	0.39	0.37	0.34	0.36	0.37	0.40	**0.37**
I	0.36	0.44	0.41	0.33	0.38	0.39	0.40	0.37	0.40	0.37	**0.39**

score, with fused MFCC and ASR features.

In general, The F1 score of SED increased approximately by 8% across all stuttering events. In addition to this clear point, the detection of the prolongation class increased by 11%. The sound repetition is the least with 3%. Regarding the other stuttering events, as Table 6.12 shows, the block's F1 score increased by approximately 10%, from 15% to 25%, while the F1 score of the word repetition and interjection increased by 8%, from 29% to 37% for word repetition and from 31% to 39% for interjection. Moreover, the model achieved 56.3% and 19.3% UAR and EER, respectively, as presented in Table 6.13.

Table 6.12: Comparison between the average F1 score in the previous experiment and the average F1 score of SED in experiment D

	P	B	S	W	I	Avg.F1
MFCC	0.16	0.15	0.23	0.29	0.31	0.23
MFCC+ASR	0.27	0.25	0.26	0.37	0.39	0.31

Table 6.13: UAR and EER results on SEP28-K dataset

	P	B	S	W	I	Avg
UAR	0.540	0.527	0.565	0.583	0.602	0.563
EER	0.172	0.184	0.192	0.224	0.192	0.193

Conclusions

This experiment confirms that the fusion of spectral and ASR features improves SED F1 score across all stuttering events by %8. The most interesting point in this experiment is that not only Word repetition and Interjection classes F1 scores were enhanced, but also the F1 score for other stuttering events improved. It can be noticed that the F1 score of the block class increased by approximately 10%, from 15% to 25%, while the word repetition and interjection increased by 8%, from 29% to 37% for word repetition and from 31% to 39% for interjection class.

6.2.5. Experiment (E): Impact of temporal features on detection performance

The main objective of this group of experiments is to observe the impact of three temporal features on SED performance. The temporal features include Fundamental Frequency (FF), Zero Crossing Rate (ZCR) and Spectral Flux Onset Strength Envelope (SFO). This group assumed that quantifying and detecting unvoiced periods and the sudden change in stuttering events may enhance the detection rate for block, prolongation and sound repetition classes.

Experimental procedure

In this experiment, the ConvLSTM fusion model with multi-feature was created. The main objective of this model is to integrate different features into a concatenated discriminative feature vector to enhance the performance of SED. The same ConvLSTM architecture described in the previous experiment and the same experimental settings was repeated, as illustrated in Figure 6.1. The temporal features are commonly used to detect unvoiced periods and sudden changes in stuttering speech. The ZCR of the temporal region frames is the ratio of changing the signal sign between negative to positive values during consecutive samples (Bäckström et al. 2014), computed as

$$Z(i) = \sum_{n=\infty}^{-\infty} \frac{1}{2} \mid \mathit{sgn}[x_i(n)] - \mathit{sgn}[x_i(n-1)] \mid \omega[(n-i)],$$

where sgn() is the sign function, and x_i is a consecutive sample within a given time

Table 6.14: F1 results on SEP28-K datasets

Event	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Avg
P	0.31	0.37	0.33	0.35	0.34	0.34	0.31	0.34	0.35	0.31	**0.33**
B	0.22	0.26	0.21	0.23	0.22	0.20	0.27	0.22	0.23	0.18	**0.22**
S	0.25	0.29	0.33	0.28	0.33	0.28	0.26	0.29	0.26	0.32	**0.29**
W	0.38	0.36	0.39	0.36	0.37	0.35	0.35	0.34	0.41	0.42	**0.37**
I	0.36	0.39	0.40	0.43	0.45	0.39	0.38	0.40	0.35	0.40	**0.40**

window ω and vice versa.

The SFO is calculated by comparing the power spectrum of the frame with the power of the previous frame and the strength envelope return vector, representing the amount of increasing spectral energy at each frame. The FF is also extracted based on pYIN algorithm (Mauch & Dixon 2014); the output of this extraction process is three frames with 47 dimensions representing the F0 candidate, Viterbi decoding and voicing flags.

Experimental results

The F1 score results of each stuttering event on the SEP-28k dataset are shown in Table 6.14. The interjection and word repetition classes achieve the best average F1 scores of 40% and 37%, respectively. In contrast, block and sound repetition classes obtain the lowest F1 score of 22% and 29% scores and 33% F1 score for the prolongation class. According to the given Table 6.15, it can notice that employing temporal features in SED enhanced the performance of SED by 2% across all the stuttering events. However, the detection of block class decreased by 3%, while the detection of prolongation and sound repetition classes increased by 6% and 3%; it also noticed that the detection of interjection class increased by 1%. Moreover, the model achieved 57.3% and 18.5% UAR and EER, respectively, as presented in Table 6.16.

Table 6.15: Comparison between the average F1 score in the previous experiment and the average F1 score of SED in experiment E

	P	B	S	W	I	Avg.F1
MFCC	0.16	0.15	0.23	0.29	0.31	0.23
MFCC+ASR	0.27	0.25	0.26	0.37	0.39	0.31
MFCC+ASR+ZCR+SFO+FF	0.33	0.22	0.29	0.37	0.4	0.33

Table 6.16: UAR and EER results on SEP28-K dataset

	P	B	S	W	I	Avg
UAR	0.572	0.520	0.580	0.584	0.611	0.573
EER	0.179	0.154	0.189	0.232	0.172	0.185

Conclusions

The finding in this group of experiments is compatible with the experiment assumption, which found that employing temporal features increased the performance of SED by 2% across all stuttering events. The detection of prolongation and sound repetition classes increased by 6% and 3%. However, block class detection decreased by 3%; this result was unexpected. The most interesting point is that interjection class detection increased by 1%. It can be concluded that the fundamental frequency,zero-crossing rate and SFO are vital features of SED.

6.3. Summary and findings

In this chapter, multiple groups of experiments were conducted toward enhancing the performance of the SED and answering research question **RQ-2**. Therefore, the chapter starts by investigating the effective use of eight common machine learning classifiers on two published datasets (FluencyBank and SEP-28k) with a relatively large number of stuttering samples with single labels to observe the behaviour of SED; the

experiments evaluated the SED performance using a perceived acoustic feature, i.e., MFCC. The experiments concluded that these techniques have struggled to detect stuttering core behaviours, as explained in experiment B. The finding of this group of experiments argued that using the single label annotation in SED is not suitable because the speech segment with 3 seconds length may contain more than one stuttering core behaviour. Therefore, in order to enhance the performance of SED, the multi-label approach is more robust; the results of the next chapter proved this finding. In addition, the data reliability issue needs to be solved; in the next chapter, the reliability issue was resolved by taking the agreement between at least three annotators.

While perceived acoustic features such as MFCC can provide valuable information for SED, they may not be sufficient for accurately detecting stuttering events. For instance, including contextual information, pre-trained models, and time-domain features may help improve the model's accuracy by guiding the detection process. Similarly, additional features such as prosodic features (e.g., intonation, rhythm, and fundamental frequency or phonetic features (e.g., phoneme and syllable-level features) can improve model generalisation and robustness. Thus, incorporating a diverse range of features can help enhance the accuracy and robustness of the speech recognition model, leading to improved performance in real-world scenarios.

Therefore, the second group of experiments in this chapter investigates the effect of spectral and temporal representation on the stuttering detection task. In addition, it examines the enhancement of stuttering detection using a pre-trained ASR model. The main objective of this group is to evaluate the performance of SED and observe the impact of applying ASR pre-trained features on each stuttering event. The investigation is based on the assumption that using ASR may enhance word repetition and interjection class detection. A baseline ConvLSTM model with Multi-feature was created to prove the experiment assumptions. The experiments utilised the 40 MFCC and 149 contextual vectors extracted from the Wav2vec2 pre-trained (Baevski et al. 2020) model.

Despite the data reliability issue, this experiments group proved that the fusion of spectral and ASR features improves SED performance across all stuttering events by

8% as demonstrated in group **D**. The F1 score of the Block class increased by approximately 10%, from 15% to 25%, while the word repetition and interjection increased by 8%, from 29% to 37% for word repetition and from 31% to 39% for Interjection class. The finding of these experiments motivates exploring more features; therefore, the experiments extended to investigate the impact of three time-domain features, i.e., ZCR, SFO, and FF features on SED performance as shown in group **E**.

The main objective of this experiment was to observe the impact of three temporal features on SED performance. The experiments are based on the assumption that quantifying and detecting unvoiced periods and the sudden change in stuttering events may enhance the detection rate for block class, prolongation and sound repetition classes. Besides the ZCR, the SFO for each speech utterance and the FF of the speech single were extracted.

The finding of this experiment provided evidence that employing temporal features increased the average F1 score of SED by 2% across all stuttering events. The F1 score of the prolongation and sound repetition classes increased by 6% and 3%. However, the F1 score of the block class decreased by 3%; this result was unexpected. This result indicates that the data reliability issue may affect the detection rate of the block class and needs to be resolved; also, the window length and overlapping of the extracted features need to be refined; the experiments of **Chapter 7** proved this conclusion. Based on the findings of this chapter, the answer to **RQ-2** will be discussed in the Conclusion chapter.

The next chapter will propose a novel attention-based model to effectively learn frame-level and temporal representations by considering contextual, pitch, time-domain and auditory-based spectral features. The multi-feature fusion approach, using time-domain features (ZCR, SFO and FF), ASR embeddings, and auditory-based spectral features, is capable of improving SED performance significantly, which outperforms state-of-the-art methods. Moreover, a convolutional block with attention maps along two separate dimensions based on CBAM (Sanghyun Woo, Jongchan Park, Joon-Young Lee 2018) was introduced for SED. The proposed work demonstrates the effectiveness of this lightweight module in performing automatic feature selection by

assigning shared weights to the intermediate feature map and eventually focusing on the salient features of speech regions, leading to improved performance in SED.

7

Multi-feature based deep attention model for stuttering event detection

7.1. Introduction

This chapter proposes to utilise multiple acoustic features extracted based on different pitch, time-domain, frequency domain, and automatic speech recognition feature to detect stuttering core behaviours more accurately and reliably. In addition, both spatial and temporal attention mechanisms are exploited as well as Bi-LSTM modules, to learn better representations to improve the SED performance. The main contributions of the this chapter are

- A novel attention-based model is proposed to effectively learn frame-level and temporal representations by considering contextual, pitch, time-domain and auditory-based spectral features.
- The multi-feature fusion approach, using time-domain features (ZCR, SFO and FF), ASR embeddings, and auditory-based spectral features, is capable of improving SED performance significantly, which outperforms state-of-the-art methods.
- A convolutional block with attention maps along two separate dimensions based on CBAM Sanghyun Woo, Jongchan Park, Joon-Young Lee (2018) was introduced for SED. The proposed work demonstrates the effectiveness of this lightweight

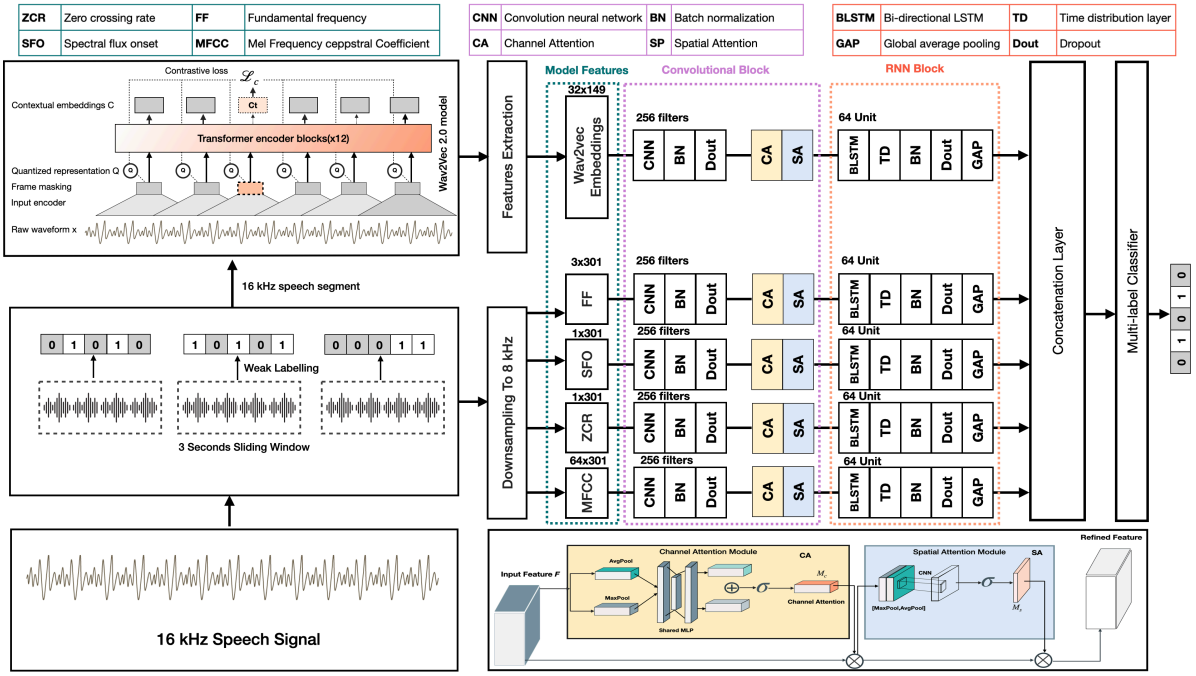


Figure 7.1: High-level overview of the proposed model architecture for stuttering events detection .

module in performing automatic feature selection by assigning shared weights to the intermediate feature map and eventually focusing on the salient features of speech regions, leading to improved performance in SED.

The rest of the chapter is structured as follows: Section 7.2 presents the proposed architecture for the stuttering events detection model; Section 7.3 presents the results and discussions of the experiments; the conclusion and findings placed in Section 7.4.

7.2. Proposed model

A high-level overview of the architecture of the proposed model for stuttering event detection is shown in Figure. 7.1. It employs a multi-feature deep learning model to effectively learn a frame-level and temporal feature representation of contextual, time-domain and auditory-based spectral features. These features are passed to the five parallel ConvLSTM branches. Combining CNN and BI-LSTM into a single architecture is suitable for SED in which temporal sequence modelling is advantageous (Mesaros et al. 2021, Kourkounakis et al. 2021). In the ConvLSTM, the CNN block is employed as a feature extractor to learn discriminative features through consecutive convolu-

tions and non-linear transformations. At the same time, the RNN block aims to learn the temporal changes of the speech signal. After each convolutional block, attention maps along two different dimensions based on CBAM (Sanghyun Woo, Jongchan Park, Joon-Young Lee 2018) has been used to improve the intermediate feature map with minimal computation and concentrate on the essential features for detecting stuttering. The CBAM is a lightweight attention mechanism that outperforms the Squeeze and Excitation (SE) method on image classification and objects detection tasks (Sanghyun Woo, Jongchan Park, Joon-Young Lee 2018). The module consists of two attention blocks as shown in Figure. 7.3, spatial and channel attention. The channel attention emphasises essential features in a feature map by weighing the importance of each channel. In addition, spatial attention highlights important regions and suppresses irrelevant regions by using a convolutional layer and skipping connection (Sanghyun Woo, Jongchan Park, Joon-Young Lee 2018). The outputs of the parallel networks are combined and fed through a dense layer with separate sigmoid activation functions for each output, producing probability for each dysfluency class. A detailed description of the model is discussed below.

7.2.1. Feature representation

A temporal acoustic region within the speech signal is annotated in a binary manner with one or more stuttering events. These annotations indicate if the stuttering events are active or inactive in that region. The acoustic regions may have more than one stuttering event. In addition, these annotations include extra metadata describing that region and contain start and end intervals in the original speech signal. In SED models, the duration of the temporal region is generally between three and five seconds. Using the fixed-sized region makes supervised learning methods suitable for SED task, where the acoustic regions and the annotations are used to train the model. Therefore, the monophonic three seconds acoustic regions were sampled at $8000\ Hz$ and $16000\ Hz$, as shown in Figure 7.1 and then fed into two layers to extract four groups of features: auditory-based spectral, pitch, temporal feature and contextual representations.

Auditory-based spectral features

The acoustic energy over a set of frequencies can be detected using spectral representation (McFee et al. 2015). This chapter's experiments extracted the auditory-based spectral feature using MFCC, a widely used feature extraction technique for speech signal analysis, which emulates the human auditory system. It involves converting audio signals into coefficients that capture spectral characteristics on a quasi-logarithmic frequency scale. The $8\ Khz$ downsampled time-domain temporal region is converted into the frequency domain using STFT with hop length with ($0.010\times$sampling rate) and ($0.025\times$ number of FFT). Subsequently, 64 filter banks were applied to the frequency domain signal to extract MFCC coefficients vector representations for each acoustic region. Despite the importance of the aforementioned spectral feature representation in detecting frequency information, pitch and time information is equally essential for SED.

Pitch and Temporal Features

Pitch and temporal features are commonly used to detect unvoiced periods and sudden changes in stuttering speech. Quantifying the unvoiced periods in stuttering speech may enhance the detection of blocks and prolongation events since voiced speech is smoother than unvoiced speech (Bäckström et al. 2022). Three pitch and temporal features were extracted from the temporal region ZCR and the spectral flux strength envelope autocorrelation. The ZCR of the temporal region frames is the ratio of changing the signal sign between negative to positive values during consecutive samples x_i within a given time window ω and vice versa (Bäckström et al. 2014).

In addition to the ZCR, The autocorrelation of a spectral flux onset (SFO) strength envelope is computed based on (Böck et al. 2012). The pYIN algorithm (Mauch & Dixon 2014) is employed to compute the fundamental frequency (F0) of the temporal region. The output of this algorithm is three frame vectors representing the F0 candidate, Viterbi decoding and voicing flags where the Viterbi decoding estimates the most likely F0 sequence and voicing flags.

Contextual features

The logits of size 32×149 produced by the WAV2Vec 2.0 model (Baevski et al. 2020) were fused with proposed features to explore the effect of transformer embedding on the experimental results. A pre-trained base WAV2Vec 2.0 model illustrated in Figure 7.2 trained on 960 of fluent speech samples was used as a feature extractor for the contextual features. Wav2vec is a family of models for learning the representation of audio data. WAV2Vec model employs mask language modelling MLM of PERT Devlin et al. (2019) on audio data, which involves training a model to predict masked (or ”hidden”) segments. In mask language modelling, the model is given an input audio signal, and a portion of the signal is masked (i.e., hidden from the model). The model is then trained to predict a token of audio data that gives the context of the unmasked portion of the signal. Wav2Vec consists of two self-supervised learning tasks, a contrastive and a quantisation task. In the contrastive task, the model is trained to maximise the similarity between related examples (positive samples) and minimise the similarity between unrelated examples (negative samples) in different masked time steps. The contrastive task is typically implemented using a contrastive $\mathcal{L}_c$ loss function, which compares the similarity between pairs of examples and adjusts the model parameters to minimise the loss. The contrastive task trains the model to perform various downstream tasks, such as stuttering events detection.

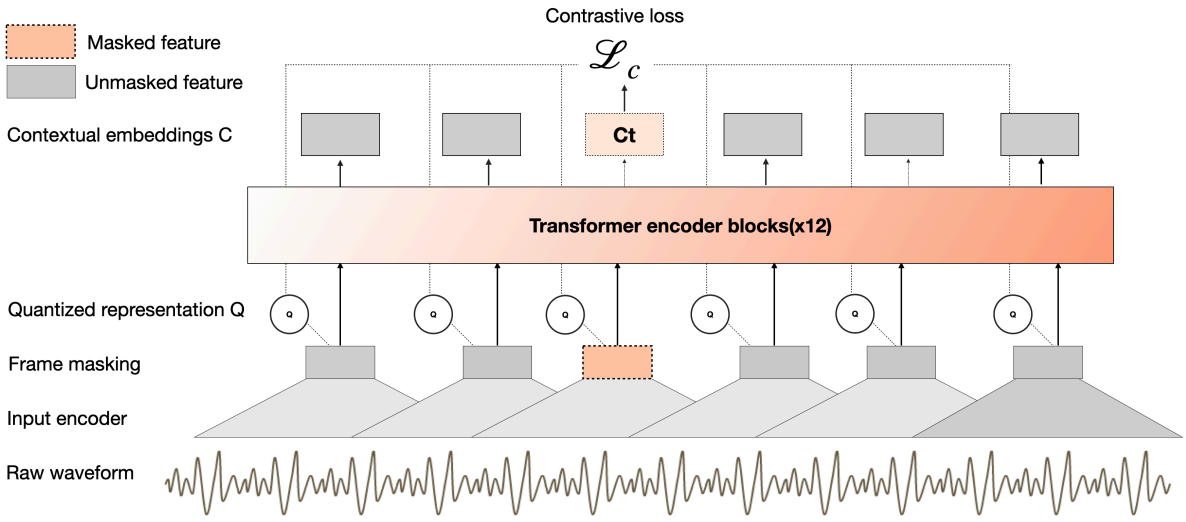


Figure 7.2: A pre-trained base WAV2Vec 2.0 model trained on 960 of fluent speech used as a feature extractor for the transformer embeddings features.

7.2.2. Convolutional block with attention maps along two separate dimensions

CNN blocks are utilised to extract discriminative features through a sequence of convolutions and non-linear transformations to learn a frame-level representation of contextual, pitch, temporal and auditory-based spectral features. This work employs five parallel convolutional blocks, both 1D and 2D. Each block consists of a single CNN layer with 256 filters and a size of 3 for 1D CNN and 3×3 for 2D CNN, followed by batch normalisation and a dropout of 0.4. A CNN block detects spatial patterns via sliding and overlapping techniques on the extracted features. The kernel operation is applied for each window of the input features to generate a feature map $f(t)$ for each window. This process is repeated for all possible windows, resulting in multiple feature maps that capture different characteristics of the input features. The feature maps pass through ReLU activation function that maintain the positive values and change the negative values to zero. The generated feature maps of each convolutional block are then channelled through attention maps along two separate dimensions based on CBAM (Sanghyun Woo, Jongchan Park, Joon-Young Lee 2018) attention mechanism to refine the intermediate feature map with minimal computation and concentrate on the essential features for detecting stuttering. The module consists of two attention modules, as illustrated in Figure 7.3, channel and spatial attention. Channel attention emphasises essential features in a feature map by weighing the importance of each channel. In addition, spatial attention highlights important regions and suppresses irrelevant regions by using a convolutional layer and skipping connection (Sanghyun Woo, Jongchan Park, Joon-Young Lee 2018). The outputs of the parallel networks are combined and fed through a dense layer with separate sigmoid activation functions for each output, producing probability for each dysfluency class.

The channel attention module processes the channel of the feature map (Sanghyun Woo, Jongchan Park, Joon-Young Lee 2018). It starts by applying maximum and average pooling to the input feature map F. The output of these pools is then fed into a multi-layer perceptron network. The sum of the two outputs is passed through a sigmoid activation function to calculate the attention values as

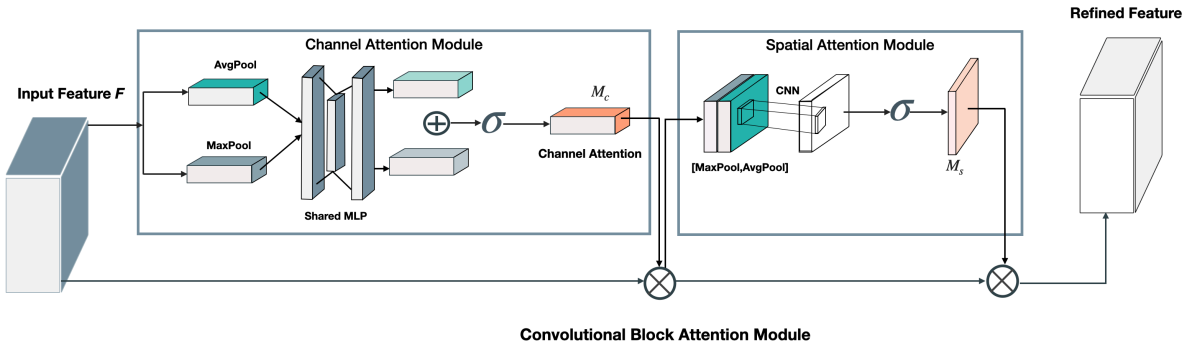


Figure 7.3: Overview of the attention maps along two separate dimensions based on CBAM .

$$M_c(F) = \sigma(MLP(max\_pool(F)) + MLP(avg\_pool(F))) = \sigma(W_1(W_0(F^c_{max})) + W_1(W_0(F^c_{avg})))$$

where σ represents the Sigmoid function, MLP is a multi-layer perceptron with one hidden layer, W_0, and W_1 are shared weights within the MLP network, and the average and maximum pooling output are represented by F^c_{avg} and F^c_{max}, respectively.

These values are then multiplied with the input features to generate the final channel attention.

The spatial attention module extracts attention in the spatial dimension by first averaging and maximum pooling the input feature map in the spatial dimension. This produces two feature maps, which are then concatenated along the channel dimension. These feature maps are then passed through a convolutional layer and a sigmoid activation function. The resulting feature values are multiplied with the original input feature to generate the final refined feature maps as

$$M_s(F) = \sigma(f^{7*7}([max\_pool(F); avg\_pool(F)])) = \sigma(f^{7*7}([F^s_{max}; F^s_{avg}]))$$

where f^{7*7} indicates that a convolution operation is being performed with a filter of size $7 * 7$ and the output denoted by F^s_{avg} corresponds to the average pooling, while F^s_{max} represents the output obtained from maximum pooling.

7.2.3. Learning temporal features using recurrent block

In order to detect stuttering events, it is vital to analyse the temporal features of speech segments. A recurrent neural network (RNN) is well suited for analysing sequential data such as speech, and can accurately identify stuttered speech events (Lea et al. 2021, Jouaiti & Dautenhahn 2022). In this work, a recurrent block is used to learn temporal features of refined feature maps. The RNN block, as illustrated in Fig 7.1, begins with permuting and reshaping the refined feature maps produced by the CBAM to be processed by a recurrent neural network (RNN). A bidirectional LSTM layer with 64 units is then applied to the reshaped feature representation. The bidirectional LSTM processes the input sequence in both forward and backward directions. It allows the model to consider past and future information, resulting in a more comprehensive understanding of the speech segment. After the LSTM layer, a TimeDistributed dense layer is employed, allowing the model to learn different feature representations at different time steps. This layer applies a fully connected dense layer to each time step of the input sequence. In addition, to improve the stability and speed of the model's training, batch normalisation is applied to the output of the time-distributed dense layer. A dropout layer with a rate of 0.4 is applied to the output of the batch normalisation layer. Dropout is a regularisation technique that can help to prevent overfitting. Finally, the output of the dropout layer is passed through a global average pooling layer, which collapses the entire 2D tensor into 1D. This is used to reduce the dimensionality of the output and make it more convenient for the final stages of the network.

7.3. Experiments

This section presents the experimental results of the proposed model for stuttering event detection. The model was trained and evaluated on two datasets on stuttering core behaviours, SEP-28k (Lea et al. 2021) and FluencyBank (Bernstein Ratner & MacWhinney 2018). In addition, the experimental setup, evaluation metrics, and results of the experiments are described. Furthermore, a comparison of the proposed model performance with existing models, and the implications and significance of the

results in the context of stuttering detection are discussed.

7.3.1. Model evaluation

The performance of the proposed model is evaluated in this chapter using the k-fold cross-validation and hold-out test. Moreover, multiple statistical metrics are considered to evaluate the proposed model.

7.3.2. Implementation details

TensorFlow 2.6 deep learning framework is utilised to build and train the model, and the Librosa library extracts features from the speech signals. In addition, the sklearn library is used to implement k-fold cross-validation. Furthermore, a pre-trained Wav2vec base model is incorporated to generate contextual features for the speech signals. Through the use of these tools, the effective processing and detection of speech signals with high accuracy are achieved. The training datasets of the temporal regions and their corresponding labels are utilised for the model's training. k-fold cross-validation with 10 folds are employed, with data shuffled before each fold and balanced class weights being used. The Adam optimiser with a learning rate of 0.0001 and binary cross-entropy loss function is utilised in training. Additionally, the best model weights during training are saved, and early stopping patience set to 5 is implemented to prevent overfitting.

7.3.3. Ablation study

In the proposed model, four groups of experiments were conducted to evaluate the impact of various feature extraction techniques on the model's performance, as presented in Table 7.1. Mel-frequency cepstral coefficients (MFCCs), logits obtained from the Wav2vec model, Zero Crossing Rate (ZCR), Spectral flux onset (SFO) and fundamental frequency were used as feature representations in the experiments. The effect of the attention mechanism was also a concern in all the experiments. The first experiment uses a baseline model of one CNN block followed by an RNN block,

Table 7.1: Summary of experimental groups, model structures, and key features.

		Features	Model structure	Description
Group1				
	Test1	MFCC	Conv Block+RNN Block	Using a single branch contains a single convolutional block and RNN block without CBAM
	Test2	MFCC	Conv Block+CBAM+RNN Block	Using a single branch contains a single convolutional block and RNN block with CBAM
Group2				
	Test1	MFCC + WAV2VEC logits	Conv Block+RNN Block	Using two branches contains a single convolutional block and RNN block without CBAM
	Test2	MFCC + WAV2VEC logits	Conv Block+CBAM+RNN Block	Using two branches contains a single convolutional block and RNN block with CBAM
Group3				
	Test1	MFCC + WAV2VEC logits+ZCR+FF+SFO	Conv Block+RNN Block	Using five branches contains single convolutional block and RNN block without CBAM
	Test2	MFCC + WAV2VEC logits+ZCR+FF+SFO	Conv Block+CBAM+RNN Block	Using five branches contains a single convolutional block and RNN block with CBAM
Group4				
	Test1	MFCC + WAV2VEC logits+ZCR+FF+SFO	Conv Block+CBAM+RNN Block	Using five branches contains single convolutional block and RNN block with CBAM attention and Focal Loss

with MFCC described in algorithm one as the sole acoustic feature. This experiment aimed to examine the behaviour of the baseline model using MFCC. Accordingly, the CNN block consists of a single 2D CNN layer with 256 filters of a 3×3 size, followed by batch normalisation and dropout of 0.4. The feature maps are then processed by attention maps based on CBAM. The RNN block includes a bidirectional LSTM layer with 64 units, a time-distributed dense layer and a global average pooling layer. The output is passed through a dense layer with sigmoid activation functions, producing probability for each stuttering event. The baseline model was tested without CBAM, as presented in group 1, to observe the impact of attention on model performance.

Table 7.2 compares the F1 score results for the baseline model and the model without the CBAM attention mechanism. As observed, using the CBAM module slightly increased the F1 score for prolongation, from 66.41% to 66.68%, and block, from 58.08% to 60.38%, which is a significant improvement. However, a decrease in the F1 score was observed for sound repetition, from 68.39% to 63.91%. Furthermore, no significant difference was observed for interjection, with the F1 score remaining at 75.4% and 75.11%, respectively. These findings suggest that incorporating the CBAM mod-

ule can lead to a significant improvement in the performance of the block, while it might have a negative effect on the performance of sound repetition. In addition, employing more features than a hand-crafted MFCC as an exclusive acoustic feature may enhance the performance of SED.

Table 7.2: F1 score comparison of four experiment and test groups with different model structures and features on SEP-28k on the test dataset.

Experiment	Prolongation	Block	Sound	Word	Interjection	Avg.F1
Group1+Test1	66.41	58.08	68.39	63.76	75.40	66.41
Group1+Test2	66.68	60.38	63.70	65.20	75.11	66.21
Group2+Test1	67.57	61.40	72.97	67.52	76.44	69.18
Group2+Test2	68.22	61.27	72.34	67.74	77.13	69.34
Group3+Test1	68.64	61.08	75.69	69.06	77.70	70.44
Group3+Test2	69.80	62.84	76.95	69.18	78.17	71.39
Group4+Test1	**72.23**	**68.66**	**79.89**	**72.07**	**79.40**	**74.45**

In the second experiment (group 2), the logits produced by the pre-trained base WAV2Vec 2.0 model were fused with MFCC to explore the effect of contextual features on the experimental results. This group combines two identical branch structures described in Table 7.1. The model structure was tested without CBAM. The results demonstrate that the model's performance improved in all the classes except prolongation. The sound repetition class recorded the most substantial improvement in the F1 score, rising by 4% from 68.39% to 72.97%. This suggests that the model could effectively learn and generalise to sound repetition. In addition, the block and the word repetition classes also recorded substantial enhancement, increasing by 3% from 58.08% to 61.4% and 63.76% to 67.52%, F1 scores, respectively. Moreover, the interjection class recorded a slight but notable improvement of the F1 score from 75.40% to 76.44%. On the other hand, the F1 score of prolongation slightly increased by less than 1% from 66.41% to 67.57%. The experiment results show that including CBAM in the model had a small but insignificant impact on its performance. Specifically, the

model's F1 score increased by 0.16%. However, the F1 score of the interjection and the prolongation recorded 1% enhancement.

Table 7.3: Results of four experiment and test groups with different model structures and features on SEP-28k on the test dataset

Experiment	Avg.ERR	Avg.UAR	Avg.recall
Group1+Test1	36.41	61.31	63.59
Group1+Test2	36.71	61.04	63.29
Group2+Test1	33.16	61.53	66.84
Group2+Test2	32.96	61.50	67.04
Group3+Test1	31.51	61.82	68.49
Group3+Test2	30.39	62.55	69.61
Group4+Test1	**23.43**	**64.29**	**76.57**

In the third experiment (group 3), multiple features were fused on the model to effectively learn a frame-level and temporal features representation of contextual, time-domain and auditory-based spectral features as described in Table 7.1. These features are passed to the five parallel branches with the same structure as the baseline model. The results demonstrate an enhancement in F1 score across all five classes of a multi-feature model with CBAM. The sound repetition class has the highest improvement of 4.61% among the stuttering events, indicating that the model struggled to detect sound repetition in the previous groups without pitch and temporal features. However, the F1 score of interjection slightly increased by 1%. Moreover, the F1 score of the block, word repetition and prolongation increased by 1.5%.

In the fourth experiment (group 4), the sigmoid activation function with a FL with $\alpha = 0.25$ and $\gamma = 2.0$ (see Section 2.5.2) was employed to modify the standard CE loss used in multi-label classifier by adding a focal weight to address the class imbalance problem. It can be clearly seen that along with the focal loss hyper tuning, the overall F1 score of the testing results increased by 3%, from 71.39% to 74.45%. In addition to the improvement of F1 score, UAR increased by 2%, EER decreased by 7% and the

recall increased approximately by 7% as reported in Table 7.3, respectively.

7.3.4. Performance comparison of the proposed model for SED

As presented in Table 7.4, the proposed multi-feature SED model based on the results presented in the ablation section surpasses the existing SEP-28k and FluencyBank datasets by 4% and 3%, with 74.44% and 71.41% overall F1 scores, respectively. Superior results indicate the consistency of using multiple features with the attention method in different stuttering events datasets. The model demonstrates improved block and word repetition F1 score results on both datasets, outperforming the current state-of-the-art models. Specifically, the F1 score of the block event achieved 68.66% on the SEP-28k and 66.20% on FluencyBank. On the other hand, for word repetition, the F1 score was 72.02% and 69.58% on SEP-28k and FluencyBank, respectively. The outperformance of existing models highlights the potential of the proposed model in stuttering event detection and demonstrates a noticeable enhancement of the F1 score of block event.

Table 7.4: F1 score comparison of four experiment and test groups with different model structures and features on SEP-28k on the test dataset.

SEP-28k	**F1**					
	Prolongation	**Block**	**Sound**	**Word**	**Interjection**	**Avg.F1**
Lea et al. (2021)	68.50	55.90	63.20	60.40	71.30	63.86
Mohapatra et al. (2022)	**73.00**	58.00	72.00	71.00	79.00	70.60
Proposed model	72.23	**68.66**	**79.89**	**72.02**	**79.40**	**74.44**

FluencyBank	**F1**					
	Prolongation	**Block**	**Sound**	**Word**	**Interjection**	**Avg.F1**
Al-Banna, Edirisinghe & Fang (2022)	44.50	45.20	49.00	NA	52.50	55.10
Mohapatra et al. (2022)	73.00	63.00	61.00	67.00	60.00	64.80
Lea et al. (2021)	67.90	56.80	**74.30**	59.30	**82.60**	68.18
Proposed model	**80.38**	**66.20**	66.41	**69.58**	74.46	**71.41**

The improved F1 score of the model on block and word repetition in both datasets is attributed to three main factors. Firstly, the proposed model addresses the reliability issues of the Sep-28k and FluencyBank datasets by using the intra-rater agreement with at least three annotators since the annotators of these datasets are not SLPS. The agreement on the block event as described in the 4.2 is 11%. The intra-rater agreement interprets the results of Al-Banna, Edirisinghe & Fang (2022) in block and other events, while Mohapatra et al. (2022) handles this issue with the same approach followed in the proposed model.

The second factor is the fusion of temporal, contextual and pitch features. This finding is compatible with the previous studies (Lea et al. 2021, Al-Banna, Edirisinghe, Fang & Hadi 2022) that argued the importance of these features to the detection block. Finally, the model effectively tackles the class imbalance problem by applying the best-fit focal weights after significant experiments. The model shows promising results and achieved state-of-the-art results only on 8 hours and 1.5 hours of reliable data of SEP-28k and FluencyBank, respectively, on all stuttering events.

Regarding the prolongation class, the model outperforms state-of-the-art by 7.38% on the FluencyBank dataset; however, Mohapatra et al. (2022) performs slightly better than the proposed model by 0.80% on SEP-28k. Concerning sound repetition, the model demonstrates promising results and surpasses the previous studies by 7.8% on SEP-28k; it also achieved 66.41% on FluencyBank. The baseline results of Al-Banna, Edirisinghe & Fang (2022) were analysed since the performance of the sound repetition is considerably less than the previous studies by 11%; this may be due to merging sound and word repetition into one core behaviour class and using F1 score on minority classes. Hence, It may be advisable to classify word and sound repetition as separate categories. Lea et al. (2021) shows strong performance on interjection on the FluencyBank dataset with 82.60% and sound repetition with 74.30%; however, it struggled to detect other stuttering events. Therefore, phoneme probabilities and articulatory vocal tract features in Lea et al. (2021) are viable for these stuttering events. In addition, using three annotators' agreements to resolve inter-rater agreements can result in missing data on interjection and sound, making them misclassified in a rela-

tively small dataset.

7.4. Summary and findings

This chapter proposes to utilise multiple acoustic features extracted based on different pitch,time-domain, frequency domain, and automatic speech recognition feature to detect stuttering core behaviours more accurately and reliably. In addition, both spatial and temporal attention mechanisms were exploited to effectively learn frame-level representations to improve the SED performance. In the proposed model, different ablation groups were conducted to evaluate the impact of various feature extraction techniques on the model's performance; the effect of attention mechanisms was also a concern in all the experiments. The proposed approach is tested on two English stuttering events datasets, SEP-28k and FluencyBank. The results demonstrate that the model is resilient and can perform well on unseen stuttering events. The experimental evaluation and analysis convincingly demonstrate that the proposed model outperforms the state-of-the-art models on both datasets, with 74.44% and 71.41% overall F1 scores on SEP-28k and FluencyBank, respectively.

An ablation study concluded that the $8\ Khz$ downsampled frequency domain signal with STFT with hope length with ($0.010\times$sampling rate, ($0.025\times$ number of FFT) and 64 filter banks MFCC coefficients are suitable for enhancing the performance of SED on a reliable large-scale dataset. The overall F1 score was 66.41%, as explained in the first group of experiments. In addition, the results of the second and third experiments groups provided evidence that the multi-feature fusion approach, using time-domain features (ZCR, SFO and FF), ASR embeddings, and auditory-based spectral features with proper parameters, is capable of improving SED performance significantly. The performance increased by 5% from 66.41% to 71.39%. Furthermore, using convolutional blocks with attention maps along two separate dimensions based on CBAM demonstrated the effectiveness of this lightweight module in performing automatic feature selection by assigning shared weights to the intermediate feature map and eventually focusing on the salient features of speech regions, leading to improved performance in SED.

The experimental evaluation and analysis convincingly demonstrate that the proposed model surpasses state-of-the-art models on two popular stuttering datasets, by 3% and 4% overall F1 scores, respectively. The superior results indicate the consistency of the proposed method, supported by both multi-feature and attention mechanisms in different stuttering events datasets.

8

Conclusion and Future Work

8.1. Thesis summary

This research has investigated different traditional ML and DL approaches on highly imbalanced and uncertain datasets to create a robust **SED** based on acoustic features. The research focused on three goals. Firstly, to investigate DL methods to create a robust stuttering detection model based on acoustic features, given the limited number of reliable stuttering observations and samples currently available for research. The research started by demonstrating the capabilities of different convolutional, recurrent architectures and hybrid approaches, on three published datasets, (manually annotated UCLASS, FluencyBank, and Sep-28k). Based on the findings of this investigation, a novel SED model architecture that detects stuttering events directly from the speech signal was proposed. The model uses a log mel spectrogram as a sole acoustic feature and a 2D atrous convolutional network to learn spectral and temporal features representation. The improvement of model generalisation and robustness using cross datasets and exclusive speaker test was conducted and showing that the performance of SED significantly worsens by more than 20% in terms of F1 score. This drop in performance may have significant implications for the model's application in real-world scenarios, highlighting the importance of the use of additional features to capture more information of the speech segment and speaker variability. Although the proposed deep learning approach showed promising results in SED, investigating the improvement of model generalisation and robustness using multi-feature and

evaluating the performance against traditional machine learning approaches is vital.

Therefore, the second goal of this thesis was to examine and evaluate the impact of the specific frequency domain, time domain and pre-trained features, on stuttering events detection using ML approaches. In order to achieve this goal, the use of eight traditional machine learning classifiers have been investigated by observing the behaviour of SED on two available datasets (FluencyBank and Sep-28k), with a relatively large number of stuttering samples with a single label annotation. The experiments concluded that these techniques have struggled to detect stuttering core behaviours (experiment B, Chapter 6). The finding of this group of experiments argued that using the single label annotation in SED is not suitable because the speech segment with 3 seconds length may contain more than stuttering core behaviour. In addition, the data reliability issue needs to be solved. While perceived acoustic features such as MFCC can provide valuable information for SED, they may not be sufficient for accurately detecting stuttering events.

Thus, the effect of spectral and temporal representation on the stuttering detection task has been investigated in this goal. In addition, it examined the enhancement of stuttering detection using a pre-trained ASR model. This group of experiments confirmed that the fusion of spectral and ASR features, improves SED F1 score across all stuttering events by 8% (group D, Chapter 6). This finding provided a motivation to explore more features; therefore, the experiments were extended to investigate the impact of three time-domain features, zero crossing rate, the auto-correlation of spectral flux onset, and fundamental frequency features on SED performance (group E, Chapter 6). The finding in this experiment provided evidence that employing temporal features increased the F1 score of SED by 2% across all stuttering events. The detection of prolongation and sound repetition increased by 6% and 3%. However, block detection decreased by 3%. This result indicates that the data reliability issue may affect the detection rate of the block event and needs to be resolved. Further, the window length and overlapping of the extracted features need to be refined.

Based on the previous goals, limitations, and findings, the final part of the thesis focused on investigating the impact of employing a multi-feature based deep atten-

tion model on the performance of the SED. This utilised multiple acoustic features extracted based on different pitch, time-domain, frequency domain, and automatic speech recognition features, to detect stuttering core behaviours more accurately and reliably. In addition, both spatial and temporal attention mechanisms were exploited, as well as Bi-LSTM modules, to learn better representations, to improve the SED performance. The experimental evaluation and analysis convincingly demonstrated that the proposed model surpasses the state-of-the-art models on two popular stuttering datasets (sep-28k and FluencyBank), with an improvement of 3% and 4% in terms of F1 scores, respectively. Moreover, a convolutional block with attention maps along two separate dimensions based on CBAM was introduced for SED. The proposed attention mechanism demonstrated the effectiveness of this lightweight module in performing automatic feature selection by assigning shared weights to the intermediate feature map and eventually focusing on the salient features of speech regions, leading to improved performance in SED.

8.2. Conclusions of thesis findings

The conclusions of the original findings of the research conducted within this thesis can be summarised as follows:

Time Interval Annotation of Stuttering Data

- In order to create a robust SED model using time interval annotation, the annotation process is better to achieved by domain field experts or SLP. Thus, the model's reliability and performance are correlated with the annotation process's quality.
- Even between domain field experts, there was a disagreement on the annotation process. Based on thesis experiments on Chapter 4, the audible block, sound repetition, and part-word repetition are challenging events; the kappa agreement for these events was 30.20%,34.40% and 42.60%, respectively. The inter-rater agreement in the annotation process may affect the result of the detection model and should be reported.

- In SED, each stuttering speech segment lasts between 3 to 5 seconds which may contain more than one stuttering event. Therefore, weak multi-label annotation is more suitable than single-label annotation. This work provided evidence that using muli-label stuttering segments enhances the SED performance dramatically.
- Stuttering speech annotation is a time-consuming process, and there is a need for open-source tools to streamline and control the annotation process; using these tools may enhance the data reliability issue.

Stuttering Event Detection Using Atrous Convolutional Neural Networks

To investigate the potential of using perceived acoustic features, i.e. mel-scale features, as the sole model input to develop a robust SED under data reliability constraint.

- The perceived acoustic features, such as mel-scale features, can provide valuable information for SED, but they may not be sufficient for accurately detecting stuttering events. The finding of experiment D in Chapter 5, which evaluated the model generalisation and robustness using cross datasets and exclusive speaker test, showed that the performance of SED significantly worsens by more than 20% F1 score for different model architectures.
- The 1D CNN model with dilation detected fluent class with an average recall of 99.33%, 0.79%, and 0.00% for sound and word repetition, respectively. The results of evaluating the 1D CNN model with the dilation model suggest that its ability to detect stuttering events may be limited. Specifically, the observed performance indicates that the model is primarily learning to detect fluent speech segments while struggling to identify and differentiate stuttering events accurately.
- The results of Chapter 5 proved that increasing the receptive fields on time and mel-scale dimensions of perceived acoustic features by employing 2D atrous with different dilation rates enhanced the detection rate for SED. The previous ablations argued that 2D atrous neural networks boost the model performance by 25% compared to 1D convolution with dilation from 27.68% to 52.87%. More-

over, the 2D atrous network outperforms the 2D with pooling with an 8% overall F1 score, indicating the model's ability to learn stuttering core behaviours.

- The main challenge in developing SED is the need for more reliable stuttering samples and observations in model training and additional features to capture more speech segment information and speaker variability information.
- The mel-scale represents the frequency domain of the stuttering speech segment since it better approximates the human auditory system's sensitivity to different frequencies. Window size and overlapping parameters are essential in the feature extraction process using mel-scale and may affect model performance.

Impact of Stuttering Event Representation on Detection Performance

Toward enhancing the performance of the SED and showing the impact of the spectral, temporal and pre-trained features in stuttering events detection, the main findings are

- Traditional ML approaches struggle to detect stuttering core behaviours on relatively large-scale datasets (experiment B, Chapter 6), compared to the results of Chapter 5 with multi-label experiments. The finding argued that using the single label annotation in SED is unsuitable because the speech segment with 3 seconds length may contain more than one stuttering core behaviour. Therefore, the multi-label approach is more robust to enhance the performance of SED.
- While perceived acoustic features, i.e., MFCC, can provide valuable information for SED, they may not be sufficient for accurately detecting stuttering events.
- Despite the data reliability issue, the experiment groups of Chapter 6 proved that the fusion of spectral and ASR features improves SED performance across all stuttering events by 8% as demonstrated in group D on chapter 6. The detection performance of the block increased by approximately 10%, from 15% to 25%, while the word repetition and interjection increased by 8%, from 29% to 37% for word repetition and from 31% to 39% for interjection.
- The quantifying and detecting unvoiced periods and the sudden change in stuttering events enhanced the detection rate for prolongation and sound repetition.

This finding provided evidence that employing temporal features increased the performance of SED by 2% across all stuttering events. The detection of prolongation and sound repetition increased by 6% and 3%. However, block detection decreased by 3%; this result was unexpected. This result indicates that the data reliability issue may affect the detection rate of the block event and needs to be resolved, and the window length and overlapping of the extracted features need to be refined.

Multi-feature based deep attention model for stuttering event detection

- The $8\ Khz$ downsampled frequency domain signal with STFT with hop length with ($0.010\times$sampling rate), ($0.025\times$ number of FFT) and 64 filter banks MFCC coefficients are suitable for enhancing the performance of SED on a reliable large-scale dataset. The overall F1 score was 66.41%, as explained in the first group of experiments in Chapter 7.
- The results of the second and third experiment groups in Chapter 7 provided evidence that the multi-feature fusion approach, using time-domain features (ZCR, SFO and FF), ASR embeddings, and auditory-based spectral features with proper parameters, is capable of improving SED performance significantly. The performance increased by 5% from 66.41% to 71.39%.
- Using convolutional blocks with attention maps along two separate dimensions based on CBAM demonstrated the effectiveness of this lightweight module in performing automatic feature selection by assigning shared weights to the intermediate feature map and eventually focusing on the salient features of speech regions, leading to improved performance in SED.

8.3. Research questions addressed

The research presented in this thesis has addressed the following research questions.

- **RQ-1** *To which level can a robust stuttering events detection model be created based on perceived acoustic features as a sole model input, given a limited*

number of reliable stuttering samples and observations?

To develop a robust SED under data reliability constraint, this work investigated employing the perceived acoustic features, i.e. mel spectrogram and MFCC on different traditional ML and DL models, as the sole model input Chapter 5 focuses on answering this question; however, Chapter 6 and Chapter 7 also tested these features with different agreement and annotation levels. The perceived features are suitable for the SED and can provide valuable information, especially when the signal window size and window overlapping are correctly identified. However, they may not be sufficient alone for accurately detecting stuttering events due to the data reliability issue and segment annotation with a single label.

The $8\ Khz$ downsampled frequency domain signal with STFT with hop length with ($0.010\times$sampling rate), ($0.025\times$ number of FFT) and 64 filter banks MFCC with multi-label archived best ConvLSTM model performance with 66.41% F1 score on relatively large-scale dataset while the best model performance of mel spectrogram was 56.4% and 50.35% using MFCC with a single label. In addition, the finding of experiment D on Chapter 5, which evaluated the model generalisation and robustness using cross datasets and exclusive speaker test, provided evidence that the performance of SED significantly worsens using the perceived acoustic features as an exclusive feature.

- **RQ-2** *What is the impact of stuttering event representation on detection performance?*

  Based on thesis experiments, stuttering events representation with a multi-label approach rather than a single label is suitable to enhance the performance of SED; the experiments showed that the performance decreased by using single label method on a perceived acoustic feature approximately by 6% using different models. Despite the data reliability issue, The experiment groups of Chapter 6 proved that employing extracted ASR pre-trained features improves SED performance across all stuttering events by 8%. In addition, the quantifying and detecting unvoiced periods and the sudden change in stuttering events en-

hanced the detection rate for prolongation and sound repetition and increased the performance of SED by 2% across all stuttering events.

- **RQ-3** *To which level can an attention-based, multi-feature fusion model architecture enhance the performance of SED?*

  Employing a multi-feature fusion approach, using time-domain features (ZCR, SFO and FF), ASR embeddings, and auditory-based spectral features on reliable data with multi-label time annotation, is capable of improving SED performance significantly, which outperforms state-of-the-art methods. The proposed multi-feature SED model in Chapter 7 surpasses existing approaches on SEP-28k and FluencyBank by 4% and 3% F1 scores, with 74.44% and 71.41% overall F1 scores, respectively. Superior results indicate the consistency of the multi-features with attention method in different stuttering events datasets. The model demonstrates improved block and word repetition performance on both datasets, outperforming the current state-of-the-art models. Specifically, the block repetition achieved 68.66% on the Sep28k and 66.20% on FluencyBank.

  On the other hand, for word repetition, the F1 score was 72.02% and 69.58% on SEP28k and FluencyBank, respectively. Using convolutional blocks with attention maps along two separate dimensions based on CBAM demonstrated the effectiveness of this lightweight module in performing automatic feature selection by assigning shared weights to the intermediate feature map and eventually focusing on the salient features of speech regions, leading to improved performance in SED.

8.4. Future work

Employing a multi-feature fusion approach, using time-domain features (ZCR, SFO and FF), and ASR embeddings, and auditory-based spectral features on reliable data with multi-label time annotation, is capable of improving SED performance significantly, which outperforms state-of-the-art methods. Despite the encouraging results of the proposed model, this model imposes two limitations. The first limitation is a data limitation. The model was trained on UCLASS, SEP-28k and FluencyBank with

a fixed-length speech segment. However, in severe stuttering cases, the repetition, block and prolongation might exceed the segment duration and cause those classes likely to be misclassified. Although the number of parameters of the proposed model is approximately eight million, the second limitation is a lack of real-time evaluation and optimisation of the proposed model's real-time performance, which is an essential consideration for many practical applications. Addressing the data limitation and lack of real-time evaluation and optimisation of SED may enhance the performance of SED and suggest directions for future research.

In this thesis, the proposed models are trained and evaluated on datasets with fixed-length segments. In Sep-28k and FluncyBank datasets, $40 - 250$ three-second intervals near the speech pause segment are extracted and annotated for each stuttering speech. In severe stuttering cases, the repetition, block and prolongation might exceed the segment duration and cause those classes likely to be misclassified. Unlike fixed-length, which may truncate or pad stuttering events, variable-length segments can capture these events and may enhance the performance of SED. Employing a multi-feature fusion approach, using time-domain features (ZCR, SFO and FF), ASR embeddings, and auditory-based spectral features with variable-length segment processing provides a future direction for this work.

In addition, the proposed features in this thesis are correlated to the nature of stuttering speech rather than speakers. It is known that any speaker generally has different speech rates and vocal tracts, which may affect speech production. Different speaker characteristics such as age, gender, and severity level may affect the performance of SED and its generalisation ability. Therefore, fusing these meta-features with the proposed approaches can be helpful in personalised therapy for PWS and provides another future direction for this thesis.

Furthermore, real-time evaluation and optimisation of the proposed model in different practical applications are necessary. Therefore, integrating, evaluating and optimising the proposed SED for live speech assistive technology or stuttering severity evaluation systems, where environmental noise, speaker variations, and speech rates are expected, provides some avenues for performance enhancement of the proposed

SED.

Developing a real-time pipeline that integrates the proposed SED in real-world scenarios involves optimising the model for low latency. Measuring the computational resources required to deploy the proposed SED and optimising factors such as processing time, memory usage, and power consumption improve the ability of the model to be applied in these practical applications. A lower computational is generally desirable, especially for mobile or embedded systems. Additionally, deploying the model on specialised hardware accelerators like GPUs or TPUs could enhance its real-time capabilities.

Analysis and evaluation SED performance in dynamic, real-time scenarios where environmental noise, speaker variations, and speech rates are vital; this analysis can help identify the model's limitations and potential areas for improvement in real-time applications. The analysis involves evaluating the model's performance under different conditions or with different input data. For example, the model's performance could be evaluated on a subset of the data with a lower signal-to-noise ratio or on data recorded from speakers with different accents or speaking styles.

Bibliography

Al-Banna, A.-K., Edirisinghe, E. & Fang, H. (2022), Stuttering detection using atrous convolutional neural networks, *in* '2022 13th International Conference on Information and Communication Systems (ICICS)', pp. 252–256.

Al-Banna, A.-K., Edirisinghe, E., Fang, H. & Hadi, W. (2022), 'Stuttering Disfluency Detection Using Machine Learning Approaches', *Journal of Information & Knowledge Management* **21**(02).

Al-Nafjan, A., Al-Wabil, A., AlMudhi, A. & Hosny, M. (2018), 'Measuring and monitoring emotional changes in children who stutter', *Computers in Biology and Medicine* **102**, 138–150.

Alharbi, S., Hasan, M., Simons, A. J., Brumfitt, S. & Green, P. (2018), A lightly supervised approach to detect stuttering in children's speech, *in* 'Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH', Vol. 2018-Septe, International Speech Communication Association, pp. 3433–3437.

Almutairi, M., Gabralla, L. A., Abubakar, S. & Chiroma, H. (2022), 'Detecting elderly behaviors based on deep learning for healthcare: Recent advances, methods, real-world applications and challenges', *IEEE Access* **10**, 69802–69821.

Arbib, M. A., ed. (1998), *The Handbook of Brain Theory and Neural Networks*, MIT Press, Cambridge, MA, USA.

Arnold, H. S., Conture, E. G. & Ohde, R. N. (2005), 'Phonological neighborhood density in the picture naming of young children who stutter: Preliminary study', *Journal of Fluency Disorders* **30**(2), 125–148.

Baevski, A., Zhou, H., Mohamed, A. & Auli, M. (2020), 'wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations'.
URL: *https://github.com/pytorch/fairseq*

Bakhtiar, M., Seifpanahi, S., Ansari, H., Ghanadzade, M. & Packman, A. (2010), 'Investigation of the reliability of the SSI-3 for preschool Persian-speaking children who stutter', *Journal of Fluency Disorders* **35**(2), 87–91.

Barrett, L., Hu, J. & Howell, P. (2022), 'Systematic Review of Machine Learning Approaches for Detecting Developmental Stuttering', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **30**, 1160–1172.
URL: *https://ieeexplore.ieee.org/document/9729855/*

Bernstein Ratner, N. & MacWhinney, B. (2018), 'Fluency Bank: A new resource for fluency research and practice', *Journal of Fluency Disorders* **56**, 69–80.

Böck, S., Krebs, F. & Schedl, M. (2012), Evaluating the online capabilities of onset detection methods., *in* 'ISMIR', pp. 49–54.

Bracewell, R. N. & Bracewell, R. N. (1986), *The Fourier transform and its applications*, Vol. 31999, McGraw-Hill New York.

Breiman, L. (2001), 'Random forests', *Machine Learning* **45**(1), 5–32.

Bäckström, T., Räsänen, O., Zewoudie, A., Zarazaga, P. P., Koivusalo, L., Das, S., Mellado, E. G., Mansali, M. B. & Ramos, D. (2014), *Introduction to Audio Analysis*, Academic Press, Oxford.
URL: *https://www.sciencedirect.com/science/article/pii/B9780080993881000091*

Bäckström, T., Räsänen, O., Zewoudie, A., Zarazaga, P. P., Koivusalo, L., Das, S., Mellado, E. G., Mansali, M. B. & Ramos, D. (2022), *Introduction to Speech Processing*, 2 edn.
URL: *https://speechprocessingbook.aalto.fi*

Cai, S., Beal, D. S., Ghosh, S. S., Tiede, M. K., Guenther, F. H. & Perkell, J. S. (2012), 'Weak Responses to Auditory Feedback Perturbation during Articulation in Persons

Who Stutter: Evidence for Abnormal Auditory-Motor Transformation'.
URL: *www.plosone.org*

Chee, L. S., Ai, O. C., Hariharan, M. & Yaacob, S. (2009*a*), Automatic detection of prolongations and repetitions using lpcc, *in* '2009 International Conference for Technical Postgraduates (TECHPOS)', pp. 1–4.

Chee, L. S., Ai, O. C., Hariharan, M. & Yaacob, S. (2009*b*), 'MFCC based recognition of repetitions and prolongations in stuttered speech using k-NN and LDA', *SCOReD2009 - Proceedings of 2009 IEEE Student Conference on Research and Development* pp. 146–149.

Chen, Q., Chen, M., Li, B. & Wang, W. (2020), Controllable Time-Delay Transformer for Real-Time Punctuation Prediction and Disfluency Detection, *in* 'ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings', Vol. 2020-May, Institute of Electrical and Electronics Engineers Inc., pp. 8069–8073.

Cieslak, M., Ingham, R. J., Ingham, J. C. & Grafton, S. T. (2015), 'Anomalous White Matter Morphology in Adults Who Stutter'.
URL: *http://www.dsi-studio.labsolver.org*

Conture, E. G., Curlee, R. F. & Curlee, R. F. (2007), *Stuttering and Related Disorders of Fluency*, Thieme Publishers Series, Thieme.
URL: *https://books.google.co.uk/books?id=BXIhAQAAMAAJ*

Cook, S. & Howell, P. P. (n.d.), Children's and parents' perspectives of psychosocial impact of stuttering and stuttering-related bullying, Technical report.
URL: *http://www.stutteringattitudes.com/documents/Presentation pdfs/Cook and Howell Parents Perspectives.pdf*

Cooper, E. B. & Cooper, C. S. (1985), 'Clinician attitudes toward stuttering: A decade of change (1973-1983)', *Journal of Fluency Disorders* **10**(1), 19–33.

Cortes, C. & Vapnik, V. N. (1995), 'Support-vector networks', *Machine Learning* **20**(3), 273–297.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y. & Belongie, S. (2019), Class-balanced loss based on effective number of samples, *in* '2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 9260–9269.

Czyzewski, A., Kaczmarek, A. & Kostek, B. (2003), *J. Intell. Inf. Syst.* **21**(2), 143–171.

Dash, A., Subramani, N., Manjunath, T., Yaragarala, V. & Tripathi, S. (2018), Speech Recognition and Correction of a Stuttered Speech, *in* '2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018', pp. 1757–1760.

Devlin, J., Chang, M., Lee, K. & Toutanova, K. (2019), BERT: pre-training of deep bidirectional transformers for language understanding, *in* J. Burstein, C. Doran & T. Solorio, eds, 'Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)', Association for Computational Linguistics, pp. 4171–4186.
URL: *https://doi.org/10.18653/v1/n19-1423*

Django Software Foundation (n.d.), 'Django'.
URL: *https://djangoproject.com*

Fayek, H. M. (2016), 'Speech processing for machine learning: Filter banks, mel-frequency cepstral coefficients (mfccs) and what's in-between'.
URL: *https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html*

Fleiss, J. L., Levin, B. & Paik, M. C. (2003), *Statistical Methods for Rates and Proportions*, Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., Hoboken, NJ, USA.
URL: *http://doi.wiley.com/10.1002/0471445428*

Fook, C. Y., MUTHUSAMY, H., CHEE, L. S., ADOM, A. H. B. & YAACOB, S. B. (2013), 'Comparison of speech parameterization techniques for the classification of speech disfluencies', *TURKISH JOURNAL OF ELECTRICAL ENGINEERING &*

COMPUTER SCIENCES **21**, 1983–1994.
URL: *https://journals.tubitak.gov.tr/elektrik/vol21/iss7/13*

G Riley, K. B. (2009), 'SSI-4: Stuttering Severity Instrument - Fourth Edition KIT Glyndon D. Riley : PRO-ED Inc. Official WebSite'.
URL: *https://www.proedinc.com/Products/13025/ssi4-stuttering-severity-instrument–fourth-edition.aspx?bCategory=ola!flu*

Garofolo, John S., Lamel, Lori F., Fisher, William M., Pallett, David S., Dahlgren, Nancy L., Zue, Victor & Fiscus, Jonathan G. (1993), 'Timit acoustic-phonetic continuous speech corpus'.
URL: *https://catalog.ldc.upenn.edu/LDC93S1*

Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep Learning*, MIT Press. `http://www.deeplearningbook.org`.

Habib, M., Faris, M., Qaddoura, R., Alomari, M., Alomari, A. & Faris, H. (2021), 'Toward an Automatic Quality Assessment of Voice-Based Telemedicine Consultations: A Deep Learning Approach', *Sensors* **21**(9), 3279.
URL: *https://www.mdpi.com/1424-8220/21/9/3279*

Hall, D. E., Lynn, J. M., Altieri, J., Segers, V. D. & Conti, D. (1987), 'Inter-intrajudge reliability of the stuttering severity instrument', *Journal of Fluency Disorders* **12**(3), 167–173.

Hampton, A. (2008), 'Non-linguistic auditory processing in stuttering: evidence from behavior and event-related brain potentials', *Elsevier* .

Hariharan, M., Chee, L. S., Ai, O. C. & Yaacob, S. (2012), 'Classification of speech dysfluencies using LPC based parameterization techniques', *Journal of Medical Systems* **36**(3), 1821–1830.

Hastie, T. J., Rosset, S., Zhu, J. & Zou, H. (2009), 'Multi-class adaboost', *Statistics and Its Interface* **2**, 349–360.
URL: *https://api.semanticscholar.org/CorpusID:11803458*

Howell, P. & Davis, S. (2011), 'Predicting persistence of and recovery from stuttering by the teenage years based on information gathered at age 8 years', *Journal of Developmental and Behavioral Pediatrics* **32**(3), 196–205.

Howell, P., S, D. & J, B. (2009), 'The University College London Archive of Stuttered Speech (UCLASS)', *Journal of speech, language, and hearing research : JSLHR* **52**(2), 556–569.
URL: *https://pubmed.ncbi.nlm.nih.gov/19339703/*

Howell, P. & Sackin, S. (1995), 'AUTOMATIC RECOGNITION OF REPETITIONS PROLONGATIONS IN STUTTERED SPEECH Peter Howell and Stevie Sackin', *Proceedings of the First World Congress on Fluency Disorders , (pp. 372-374).* .

Howell, P., Sackin, S. & Glenn, K. (1997), 'Development of a Two-Stage Procedure for the Automatic Recognition of Dysfluencies in the Speech of Children Who Stutter: II. ANN Recognition of Repetitions and Prolongations With Supplied Word Segment Markers Europe PMC Funders Group', *J Speech Lang Hear Res* **40**(5), 1085–1096.

Iimura, D. & Miyamoto, S. (2020), 'The influence of stuttering and co-occurring disorders on job difficulties among adults who stutter', *Speech, Language and Hearing* pp. 1–10.

Ingham, R. J., Cordes, A. K. & Finn, P. (1993), 'Time-Interval Measurement of Stuttering', *Journal of Speech, Language, and Hearing Research* **36**(6), 1168–1176.
URL: *http://pubs.asha.org/doi/10.1044/jshr.3606.1168*

Jeon, H., Jung, Y., Lee, S. & Jung, Y. (2020), 'Area-Efficient Short-Time Fourier Transform Processor for Time–Frequency Analysis of Non-Stationary Signals', *Applied Sciences* **10**(20), 7208.
URL: *https://www.mdpi.com/2076-3417/10/20/7208*

Jouaiti, M. & Dautenhahn, K. (2022), 'Dysfluency Classification in Stuttered Speech Using Deep Learning for Real-Time Applications', *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*

pp. 6482–6486.
URL: *https://ieeexplore.ieee.org/document/9746638/*

Kell, C. A., Neumann, K., von Kriegstein, K., Posenenske, C., von Gudenberg, A. W., Euler, H. & Giraud, A.-L. (2009), 'How the brain repairs stuttering', *Brain* **132**(10), 2747–2760.
URL: *https://academic.oup.com/brain/article-lookup/doi/10.1093/brain/awp185*

Khara, S., Singh, S. & Vir, D. (2018), A Comparative Study of the Techniques for Feature Extraction and Classification in Stuttering, *in* 'Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018', Institute of Electrical and Electronics Engineers Inc., pp. 887–893.

Kourkounakis, T., Hajavi, A. & Etemad, A. (2020), Detecting Multiple Speech Disfluencies Using a Deep Residual Network with Bidirectional Long Short-Term Memory, *in* 'ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings', Vol. 2020-May, Institute of Electrical and Electronics Engineers Inc., pp. 6089–6093.

Kourkounakis, T., Hajavi, A. & Etemad, A. (2021), 'FluentNet: End-to-End Detection of Stuttered Speech Disfluencies with Deep Learning', *IEEE/ACM Transactions on Audio Speech and Language Processing* **29**, 2986–2999.

Krishnavedala (2013), 'Mel scale vs hertz scale.', `https://upload.wikimedia.org/wikipedia/commons/a/aa/Mel-Hz_plot.svg`. Accessed: 2022-24-10.

Lea, C., Huang, Z., Jain, D., Tooley, L., Liaghat, Z., Thelapurath, S., Findlater, L. & Bigham, J. P. (2022), Nonverbal sound detection for disordered speech, *in* 'ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', pp. 7397–7401.

Lea, C., Mitra, V., Joshi, A., Kajarekar, S. & Bigham, J. P. (2021), SEP-28k: A Dataset for Stuttering Event Detection from Podcasts with People Who Stutter, *in* 'ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Pro-

cessing (ICASSP)', IEEE, pp. 6798–6802.
URL: *https://ieeexplore.ieee.org/document/9413520/*

Lewis, K. E. (1995), 'Do SSI-3 Scores Adequately Reflect Observations of Stuttering Behaviors?', *American Journal of Speech-Language Pathology* **4**(4), 46–59.
URL: *http://pubs.asha.org/doi/10.1044/1058-0360.0404.46*

Li, B., Muñoz, J. P., Rong, X., Chen, Q., Xiao, J., Tian, Y., Arditi, A. & Yousuf, M. (2019), 'Vision-based mobile indoor assistive navigation aid for blind people', *IEEE Transactions on Mobile Computing* **18**(3), 702–714.

Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. (2020), 'Focal loss for dense object detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **42**(2), 318–327.

Mahesha, P. & Vinod, D. S. (2016), 'Gaussian Mixture Model Based Classification of Stuttering Dysfluencies', *Journal of Intelligent Systems* **25**(3), 387–399.
URL: *https://www.degruyter.com/document/doi/10.1515/jisys-2014-0140/html*

Manjula, G., Shivakumar, M. & Geetha, Y. (2019), Adaptive optimization based neural network for classification of stuttered speech, *in* 'ACM International Conference Proceeding Series', pp. 93–98.

Manning, C. D., Raghavan, P. & Schütze, H. (2008), *Introduction to Information Retrieval*, Cambridge University Press.

Mauch, M. & Dixon, S. (2014), Pyin: A fundamental frequency estimator using probabilistic threshold distributions, *in* '2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', pp. 659–663.

McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E. & Nieto, O. (2015), librosa: Audio and music signal analysis in python, *in* 'SciPy'.

Mesaros, A., Heittola, T., Virtanen, T. & Plumbley, M. D. (2021), 'Sound Event Detection: A tutorial', *IEEE Signal Processing Magazine* **38**(5), 67–83.
URL: *https://ieeexplore.ieee.org/document/9524590/*

Miller, B. & Guitar, B. (2009), 'Long-Term Outcome of the Lidcombe Program for Early Stuttering Intervention', *American Journal of Speech-Language Pathology* **18**(1), 42–49.
URL: *http://pubs.asha.org/doi/10.1044/1058-0360%282008/06-0069%29*

Mitra, V., Huang, Z., Lea, C., Tooley, L., Georgiou, P., Kajarekar, S. & Bigham, J. (2021), Analysis and tuning of a voice assistant system for dysfluent speech.
URL: *https://arxiv.org/pdf/2106.11759.pdf*

Mohapatra, P., Pandey, A., Islam, B. & Zhu, Q. (2022), Speech Disfluency Detection with Contextual Representation and Data Distillation, *in* 'Proceedings of the 1st ACM International Workshop on Intelligent Acoustic Systems and Applications', ACM, New York, NY, USA, pp. 19–24.
URL: *https://dl.acm.org/doi/10.1145/3539490.3539601*

Mubarak, H., Hussein, A., Chowdhury, S. A. & Ali, A. (2021), 'Qasr: Qcri aljazeera speech resource – a large scale annotated arabic speech corpus'.
URL: *https://arxiv.org/abs/2106.13000*

Pálfy, J. (2014), Analysis of Dysfluencies by Computational Intelligence, Technical report.

Panayotov, V., Chen, G., Povey, D. & Khudanpur, S. (2015), Librispeech: An asr corpus based on public domain audio books, *in* '2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', pp. 5206–5210.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011), 'Scikit-learn: Machine learning in python', *Journal of Machine Learning Research* **12**, 2825–2830.

Ratner, N. B. & Tetnowski, J. A. (2014), *Current Issues in Stuttering Research and Practice*, Taylor & Francis.
URL: *https://books.google.co.uk/books?id=pORHAwAAQBAJ*

Ravikumar, K., Rajagopal, R. & Nagaraj, H. (2009), 'An approach for objective assessment of stuttered speech using mfcc features', *DSP Journal* **9**.

Riley, G. D. (1991), 'Response to reviews by E. Charles Healey and Donald Mowrer'.

Riley, J., Riley, G. & Maguire, G. (2004), 'Subjective Screening of Stuttering severity, locus of control and avoidance: Research edition', *Journal of Fluency Disorders* **29**(1), 51–62.

Ronald B. Gillam, T. P. M. (2022), *Communication Sciences and Disorders: From Science to Clinical Practice: From Science to Clinical Practice*, 4th edn, Jones & Bartlett Learning.

Salzberg, S. L. (1994), 'C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993', *Machine Learning* **16**(3), 235–240.
URL: *http://link.springer.com/10.1007/BF00993309*

Sanghyun Woo, Jongchan Park, Joon-Young Lee, I. S. K. (2018), *CBAM: Convolutional Block Attention Module*, Springer Nature.

Sheikh, S. A., Sahidullah, M., Hirsch, F. & Ouni, S. (2021), 'StutterNet: Stuttering Detection Using Time Delay Neural Network', *European Signal Processing Conference* **2021-August**, 426–430.

Shensa, M. (1992), 'The discrete wavelet transform: wedding the a trous and Mallat algorithms', *IEEE Transactions on Signal Processing* **40**(10), 2464–2482.
URL: *http://ieeexplore.ieee.org/document/157290/*

Sidavi, A. & Fabus, R. (2010), 'A Review of Stuttering Intervention Approaches for Preschool-Age and Elementary School-Age Children', *Contemporary Issues in Communication Science and Disorders* **37**(Spring), 14–26.
URL: *https://pubs.asha.org*

Smołka, E., Kuniszyk-Jóźkowiak, W., Wiśniewski, M. & Suszyński, W. (2007), 'Automatic detection of prolonged fricative phonemes with the Hidden Markov Models approach', *Journal of Medical Informatics & Technologies* **11**, 293–297.

Stevens, S. S., Volkmann, J. & Newman, E. B. (1937), 'A Scale for the Measurement of the Psychological Magnitude Pitch', *The Journal of the Acoustical Society of America* **8**(3), 185–190.
URL: *http://asa.scitation.org/doi/10.1121/1.1915893*

Świetlicka, I., Kuniszyk-Jóźkowiak, W. & Smołka, E. (2009), 'Artificial neural networks in the disabled speech analysis', *Advances in Intelligent and Soft Computing* **57**, 347–354.

Tahmasebi, N., Shafie, B., Karimi, H. & Mazaheri, M. (2018), 'A Persian-version of the stuttering severity instrument-version four (SSI-4): How the new additions to SSI-4 complement its stuttering severity score?', *Journal of Communication Disorders* **74**, 1–9.
URL: *https://doi.org/10.1016/j.jcomdis.2018.04.005*

Tan, T. S., Helbin-Liboh, Ariff, A. K., Ting, C. M. & Salleh, S. H. (2007), 'Application of Malay speech technology in Malay speech therapy assistance tools', *2007 International Conference on Intelligent and Advanced Systems, ICIAS 2007* pp. 330–334.

Van Riper, C. (1973), *The treatment of stuttering*, Prentice-Hall.

Villegas, B., Flores, K., Pacheco-Barrios, K. & Elias, D. (2019), Monitoring of respiratory patterns and biosignals during speech from adults who stutter and do not stutter: A comparative analysis, *in* 'International Symposium on Medical Information and Communication Technology, ISMICT', Vol. 2019-May.

Wang, J., Chen, Q. & Chen, Y. (2004), Rbf kernel based support vector machine with universal approximation and its application, *in* F.-L. Yin, J. Wang & C. Guo, eds, 'Advances in Neural Networks – ISNN 2004', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 512–517.

World Health Organisation (2010), 'International statistical classification of diseases and related health problems 10th revision (icd-10) version for 2010'. `https://icd.who.int/browse10/2010/en#/F98.5`, Accessed on 6 February 2021.

Yairi, E. & Ambrose, N. (2013), ‘Epidemiology of stuttering: 21st century advances’, *Journal of Fluency Disorders* **38**(2), 66–87.
URL: */pmc/articles/PMC3687212/ /pmc/articles/PMC3687212/?report=abstract https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3687212/*

Yu, F. & Koltun, V. (2016), Multi-scale context aggregation by dilated convolutions, *in* Y. Bengio & Y. LeCun, eds, ‘4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings’.
URL: *http://arxiv.org/abs/1511.07122*

Zayats, V., Ostendorf, M. & Hajishirzi, H. (2016), ‘Disfluency detection using a bidirectional LSTM’, *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* **08-12-September-2016**, 2523–2527.