Enhancing EEG-Based Imagined Speech Recognition Through Spatio-Temporal Feature Extraction Using Information Set Theory

An Undergraduate Project Report submitted to Manipal Academy of Higher Education in partial fulfilment of the requirement for the award of the degree of

Bachelor of Technology

in

Biomedical Engineering

Submitted by

Ashrith Sagar Yedlapalli

Reg No: 200902016

Under the guidance of

Dr. Jeevan M

Associate Professor

Department of Biomedical Engineering,

Manipal Institute of Technology, Manipal, India

MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

July, 2024

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Manipal
July 8, 2024

CERTIFICATE

This is to certify that the project titled **Enhancing EEG-Based Imagined Speech Recognition Through Spatio-Temporal Feature Extraction Using Information Set Theory** is a record of the bonafide work done by **Mr. Ashrith Sagar Yedlapalli** (Reg. No. 200902016) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech) in **Biomedical Engineering** of Manipal Institute of Technology Manipal, Karnataka, (A Constituent College of Manipal Academy of Higher Education), during the academic year 2023-24.

Dr. Jeevan Medikonda
Project Guide

Dr. Niranjana Sampathila
HOD, Dept. of Biomedical Engineering
M.I.T., MANIPAL

ACKNOWLEDGEMENTS

This project would not have been possible without the constant support and guidance received from peers and professors. I want to thank my project guide, Dr. Jeevan M (Associate Professor, Department of Biomedical Engineering), for his guidance and support throughout this project. He has been instrumental in providing insights and suggestions, and has been a key contributor to the success of this project. Further, I would like to thank Dr. Muralidhar Bairy (Professor & Former HOD, Department of Biomedical Engineering) and Dr. Niranjana Sampathila (Professor & HOD, Department of Biomedical Engineering) for their constant support and encouragement through the two years that was spent working on this project.

I also wish to express my gratitude to friends and family members for their welcoming support whenever it was needed. Their encouragement and support received has been invaluable and has been a key driving force for the completion of this project especially the numerous times when roadblocks where encountered.

I wish to show gratitude to the numerous developers involved in Unix, and Python communities, git, LaTeX, Ti*k*Z, for their invaluable contributions to the open-source community, and whose tools and resources have been such an integral part of my everyday routine.

Finally, I would like to thank the Department of Biomedical Engineering, Manipal Institute of Technology, for providing the necessary resources in the Lab to develop this project. Lastly, it was a great learning experience and has been a remarkable journey of 4 years in my life.

ABSTRACT

Imagined speech is a form of speech wherein an individual mentally articulates words without any physical movement.

In this study, we perform an Imagined speech classification task using EEG signals by utilising a novel approach to extract rich spatio-temporal features using Information set theory techniques to capture more information and improve the classification. We improve over the feature extraction and feature selection with the rich spatio-temporal features offering better differentiating power and drastically reducing the dataset size used for training without sacrificing the classification performance. In addition, this procedure allows for effectively utilising all the information in the EEG signals, reducing the risk of discarding potentially important information without adding to the computational complexity of the feature space.

The proposed approach was tested on the KaraOne database, obtaining average accuracies varying between $40\% - 90\%$ across five binary phonological tasks and trained using multiple independent classifiers, including the Hanman Classifier, Decision Tree, Random Forest, and Support Vector Machine. The best performing model is the Random Forest Classifier sampled with the Random Over Sampler yielding average accuracies of $60\% - 90\%$ in the tasks, which is a significant improvement over the KaraOne baseline machine learning method.

This demonstrates the effectiveness and improvement of the feature selection techniques in creating rich spatio-temporal features for imagined speech classification. The implementation was done on Python3 and the scikit-learn library and is publicly available on GitHub [1] under the MIT License.

Keywords: Imagined speech classification, EEG signals, Spatio-temporal features, Information set theory, Hanman classifier, Machine learning.

[1] https://github.com/AshrithSagar/EEG-Imagined-speech-recognition

LIST OF TABLES

2.1 Frequency bands of EEG waves 5
2.2 Comparison of KaraOne and FEIS datasets 8

3.1 Summary of features extracted in KaraOne database 12
3.2 Parameters for the different models 19

4.1 Performance metrics of proposed methodology 21
4.2 Confusion matrices 22
4.3 Results comparison: KaraOne vs. IFST 23
4.4 Top 10 highest mean correlations and corresponding *p*-values 24

A.1 ANOVA analysis of the five binary classification tasks 31
A.2 Summary of functions used for feature extraction 33

LIST OF FIGURES

2.1 Recording procedure used in KaraOne database 7
2.2 FEIS model building procedure 8

3.1 Proposed methodology flowchart 10
3.2 Preprocessing workflow 11
3.3 Extracting effective information from the features 14
3.4 Random Forest classifier illustration 16
3.5 Support Vector Machine illustration 17

4.1 KaraOne methodology for feature selection 23
4.2 Top 10 EEG channels with highest mean correlations 25

LIST OF ALGORITHMS

1 Effective Information 14
2 Hanman Classifier 15

LIST OF ABBREVIATIONS

***k*-NN**	*k*-Nearest Neighbors
ADASYN	Adaptive Synthetic Sampling
ALS	Amyotrophic lateral sclerosis
ANOVA	Analysis of Variance
BCI	Brain-computer interface
CNN	Convolutional Neural Network
DT	Decision Tree Classifier
DT	Decision Tree
EEG	Electroencephalography
FEIS	Fourteen-channel EEG for Imagined Speech
HC	Hanman Classifier
LSTM	Long Short-Term Memory
MCN	Modified Combinatorial Nomenclature
RF	Random Forest Classifier
RF	Random Forest
ROS	Random Over Sampler
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine

TABLE OF CONTENTS

Acknowledgements **i**

Abstract **ii**

List of Tables **iii**

List of Figures **iv**

List of Algorithms **v**

List of Abbreviations **vi**

Chapter 1 Introduction **1**
- 1.1 Area of work . . . 1
- 1.2 Rationale behind the work . . . 2
- 1.3 Target specifications . . . 2
- 1.4 Project work schedule . . . 3
- 1.5 Organisation of the report . . . 3

Chapter 2 Literature review **4**
- 2.1 Electroencephalography (EEG) . . . 4
  - 2.1.1 Frequency bands . . . 4
- 2.2 Imagined speech . . . 5
- 2.3 Information sets . . . 6
- 2.4 Datasets . . . 6
  - 2.4.1 KaraOne dataset . . . 6
  - 2.4.2 FEIS dataset . . . 7
- 2.5 Summarised outcomes . . . 8
- 2.6 Objectives . . . 8

Chapter 3 Methodology **10**
- 3.1 Dataset . . . 10
  - 3.1.1 Preprocessing . . . 10
  - 3.1.2 Feature extraction . . . 11
- 3.2 Model . . . 15
- 3.3 Classification . . . 18
  - 3.3.1 Class imbalance . . . 18

3.3.2 Evaluation metrics and Model parameters 18
3.4 Sampling 18
3.4.1 Random Over Sampler 18
3.4.2 SMOTE and ADASYN 19

Chapter 4 Results **20**
4.1 Performance metric analysis 20
4.1.1 Comparative analysis with existing literature 23
4.1.2 Correlation analysis: EEG vs. Acoustic features 24
4.2 Limitations 24
4.3 Applications 26

Chapter 5 Conclusion **27**
5.1 Conclusion 27
5.2 Future scope of work 27

References **28**

Annexures **31**
A.1 ANOVA analysis 31
A.2 Feature functions 33
A.3 Hanman classifier 35
A.4 Implementation details 37
A.4.1 Directory structure 37
A.4.2 Classifier module 37

Project details

Orginality report (Turnitin)

CHAPTER 1

INTRODUCTION

This chapter provides an overview of brain wave analysis using EEG in research, focusing on its application to understanding brain function and facilitating communication via Brain-computer interfaces (BCIs). It begins with an exploration of EEG technology's historical development and its significance in mapping brain activity. Then we discuss on the challenges and advancements in BCI systems, particularly in the context of recognizing imagined speech, highlighting the research objectives, methodology, and structure of this thesis.

1.1 Area of work

The field of brain wave analysis in the research setting has been extensively researched with the use of **Electroencephalographys (EEGs)**, which is a non-invasive method to record the electrical activity in the brain by placing electrodes on the scalp. Since the first recording of EEG signals by Hans Berger in 1924 [28], it has been applied quite successfully to understand the workings of the brain, particularly in mapping the brain function to bodily activity, which is a majorly popular area of interest. The electrodes in the conventional scalp EEG are typically placed using a standardised Internation 10-20 system [], with conductive electrode gels applied to improve conductivity and reduce noise due to motion artefacts. **Brain-computer interfaces (BCIs)** stand at the forefront of technological advancement, offering individuals with paralysis a means to interact directly with their surroundings. These systems are pivotal for enhancing the quality of life for patients and facilitating seamless interaction with the world around them. Despite their potential, existing BCI paradigms such as event-related potentials and motor imagery are constrained by their reliance on specific stimuli and limited class options for practical communication. Moreover, they have demonstrated inefficiencies in user control, highlighting the pressing need for a more intuitive and user-friendly framework [15]. A number of research studies have explored the feasibility of performing EEG classification, although currently these methods have not yet reached a level of performance for practical applications. The ability to effectively classify EEG signals remains an interesting and challenging problem, with the potential to better the lives of individuals with disabilities.

The journey of EEG technology began in the early 20th century and has since evolved into a sophisticated tool for brain research. Over the decades, BCIs have seen substantial advancements, from basic control systems to complex interfaces that can interpret various mental states. Imagined speech, in particular, has emerged as a crucial area of study, offering a direct link between thought and action without the need for physical movement. Despite significant progress, several challenges persist. Current BCI methods often struggle with accuracy and user-friendliness, primarily due to the complex nature of EEG signal interpretation. The limited class options for communication and the reliance on specific stimuli restrict the practicality of these systems. Additionally, user control remains a significant challenge, necessitating the development of more intuitive interfaces. The potential applications of advanced BCIs are vast, ranging from medical rehabilitation to everyday communication aids. Improved BCI systems could greatly enhance the independence and quality of life for individuals with disabilities. Moreover, the integration of these technologies into daily life could lead to new forms of interaction and control, making technology more accessible and responsive to individual needs.

Imagined speech entails the mental representation of words or concepts an individual intends to convey without verbal articulation [15]. This cognitive process is captured through EEG signals, a preferred method due to its non-invasive nature and ease of use and setup. Using EEG signals to recognise imagined speech from brain activity directly holds promise for facilitating more efficient control of BCIs with reduced training requirements.

1.2 Rationale behind the work

EEG signals are part of a broader framework of Brain-computer interface (BCI), which have significantly improved the lives of people affected by paralysis by offering a means to interact with their surroundings. BCI systems acquire and analyse brain signals and translate them into useful commands for the user [24]. They do not use the natural neuromuscular pathways since their main motive is to be used in cases to replace or restore the lost function of individuals with disabilities. Despite their potential, existing BCI paradigms are primarily constrained by their reliance on specific stimuli and limited class options, making them difficult for practical communication [15], highlighting the need for a more intuitive and user-friendly framework.

One area where EEG signals have been used is in recognising Imagined speech, also called Covert or Silent speech, where the individual visualises a mental representation of words or concepts they intend to convey instead of verbal articulation [15]. Using EEG signals to recognise imagined speech from brain activity is preferred due to its non-invasive nature and ease of use.

1.3 Target specifications

By introducing novel approaches to feature extraction and classification, this research seeks to advance the field of BCI and contribute to the development of more effective and user-friendly systems.

1.4 Project work schedule

In the month of January, the project scope and deliverables were identified. A comprehensive literature review was already carried out a few months prior, was extended during this period, and was organised to identify a suitable dataset to be used. In the month of February, the relevant details of the dataset were obtained, and the pre-processing parts were implemented first. The feature extraction parts in the corresponding literature were implemented, followed by the implementation of the proposed methodology and the novel feature extraction method. In March, the classification models were identified and trained on the dataset using both the literature and proposed techniques, and they were subsequently evaluated. The results were analysed in the month of April, with relevant metrics and details to be included. The documentation of the implementation of the project repository was worked on in March and continually improved and completed by the month of May. The final report and documentation were compiled during May and June, and the final project presentation was prepared.

1.5 Organisation of the report

This thesis report comprises 5 chapters, with this chapter being the Introduction and the first. Chapter 2 provides a comprehensive review of the literature on EEG classification and feature extraction techniques, and datasets for imagined speech classification. Chapter 3 thoroughly outlines the research methodology, including details on the dataset, preprocessing steps, and classification techniques utilised. Chapter 4 presents the results of the classification task and evaluates the model performance and assesses thei effectiveness. It also discusses the implications of the study, highlighting the contributions and limitations of the research. Chapter 5 serves as the conclusion of the report, summarizing the main findings and providing recommendations for future research. Finally, the last section provides references and the appendices contain additional information and supplementary materials.

CHAPTER 2
LITERATURE REVIEW

This chapter presents a comprehensive overview of the literature on EEG-based imagined speech recognition. We start with the EEG modility in general and discuss on the different frequency bands associated with it. We then move on to the concept of imagined speech and its classification using machine learning models. The introduction and application of information sets in literature are covered next. Finally, we discuss the popular datasets used in the literature for imagined speech classification, and the outcomes and objectives of this study.

2.1 Electroencephalography (EEG)

Electroencephalography (EEG) is a non-invasive and efficient modality used to record electrical brain activity by measuring electrical signals from electrodes placed on the scalp [16]. Since Hans Berger first recorded EEG signals in 1924 [28], EEG has been employed to study brain activity and has become an invaluable tool in neuroscience. EEG waveforms are generally divided into consecutive bands based on frequency ranges as Delta (0.5 to 4 Hz), Theta (4 to 8 Hz), Alpha (8 to 12 Hz), Beta (12 to 35 Hz), and Gamma (greater than 36 Hz), respectively, and are essential in EEG analysis providing insights into the brain activity at different levels of consciousness and cognitive processing [17]. The frequency bands are summarised in Table 2.1.

Preprocessing of EEG signals usually includes a combination of downsampling, filtering and windowing. Filtering can be done in the spatial, time, or frequency domains [17]. Bandpass filters are typically used to filter out the EEG signals into the range of $0.5 - 30$Hz under standard clinical recording techniques. The aim of feature extraction is to capture the relevant and meaningful information within the data, which is generally advantageous in the classification step and can be accomplished by analysing the time, frequency, and spatial domains [17].

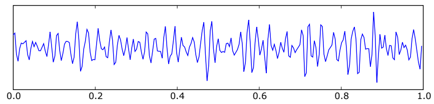
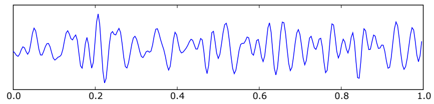
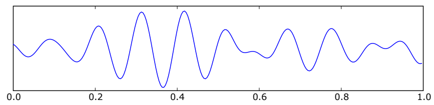
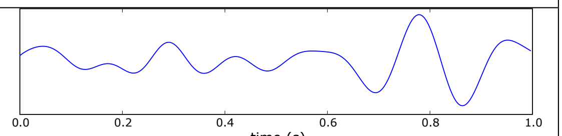
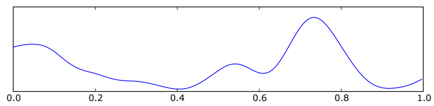
2.1.1 Frequency bands

Gamma band : Associated with higher cognitive functions, such as memory, perception, and problem-solving.

Beta band : Typically associated with active thinking, focus, and concentration. This brain activity is commonly observed bilaterally across the frontal and parietal lobes [16].

Alpha band : Associated with relaxation, meditation, and creativity. It induces the production of serotonin, which is the neuro-transmitter that increases relaxation and pain relief [16]. It is evident on both sides of the brain, with marginally greater amplitude typically observed on the

Table 2.1: EEG waves and their frequency bands

Frequency band	**Frequency range**	**State**	**Signal**
Gamma	> 35 Hz	Combination of two senses	
Beta	$12 - 35$ Hz	Active thinking, Alertness	
Alpha	$8 - 12$ Hz	Calmness, Day dream	
Theta	$4 - 8$ Hz	Deeply relaxed	
Delta	$0.5 - 4$ Hz	Deep rest, dreamless sleep	

non-dominant hemisphere. It is recorded from the occipetal and parietal regions and represents the white matter of the brain.

Theta band : Linked with deep relaxation, meditation, and sleep.

Delta band : Linked to deep sleep, unconsciousness, and healing processes. This wave exhibits the highest amplitude and is the slowest among other brain waves [16].

2.2 Imagined speech

Imagined speech, also referred to as Covert speech, entails the mental representation of words or concepts an individual intends to convey without verbal articulation [15]. BCI for imagined speech applications generally comprises four steps: signal acquisition, applying relevant signal processing techniques, feature extraction in a particular domain and classification [17]. For the classification of imagined speech, researchers have utilized both traditional machine learning methods and deep learning techniques [17]. Several traditional machine learning models have been attempted for classifying imagined speech, including 'Support Vector Machines (SVMs)', 'k-Nearest Neighbors (k-NN)' and 'Random Forest (RF)'. While these perform well, deep learning models such as 'Convolutional Neural Network (CNN)' and 'Long Short-Term Memory (LSTM)' networks have shown better performance [17]. However, they have drawbacks, needing a significantly large amount of training data and being computationally expensive. A popularly used dataset for imagined speech classification is the KaraOne database [27], which includes seven phonemic/ syllabic prompts and four words. There has been an effort to use a more cost-effective acquisition system, as carried out in the FEIS dataset [26], where they adopt a similar approach to KaraOne.

In the study done by [27], the database included seven phonemic/syllabic prompts and four words which were classified across five imagined speech binary classification tasks (See 2.4.1). In the following study done by [26], the dataset was collected using a similar approach but with a more cost-effective acquisition system (See 2.4.2).

2.3 Information sets

The applications of Information sets to extract effective information were extensively utilised in [1, 19, 18], addressing the shortcomings of fuzzy sets. An information set is created from a fuzzy set, where the values in the fuzzy set are considered as information source values, which, when multiplied with an entropy function, yields the information values. These information values constitute an information set whose sum denotes the information/uncertainty in the information set. Information sets decouple the information source values and the membership grades in a fuzzy set by computing the information values as a product and offer an extension to the entropy function to be applied either in the probabilistic or possibilistic domain or both. In each of the problems dealt with in [1, 19, 18], feature extraction was performed by transforming the data into some sort of information set, following which the effective features were extracted to enable better classification. The motivation for the proposed methodology follows this theme.

2.4 Datasets

2.4.1 KaraOne dataset

The KaraOne database [1] [27] is a publicly available multi-modal imagined speech dataset. It contains data from 14 participants (four female and eight male) over three modalities: EEG signals, facial tracking, and audio signals. A 64-channel Neuroscan Quick-cap recorded the EEG signals sampled with a sampling frequency of 1000 Hz. The cues consist of seven phonemic prompts ($/diy/$, $/iy/$, $/piy/$, $/tiy/$, $/uw/$, $/m/$, $/n/$) and four words (knew, gnaw, pat, pot), and were chosen in a way that ensured a balanced representation of vowels, plosives, nasals, and voiced and unvoiced phonemes. The experimental setup used while recording the data is given in Fig. 2.1.

Each trial involved four states:

[1] Available at 'https://www.cs.toronto.edu/~complingweb/data/karaOne/karaOne.html'

1. **Rest**: A 5-second relaxation period during which the subject was instructed to unwind and stop actively thinking.
2. **Stimulus**: A 2-second period during which the participant was presented with a visual and auditory prompt. The visual prompt was displayed on the monitor screen and consisted of a written text of the prompt, while the auditory prompt was played through the speakers. The participant was also instructed to position their articulators in this stage.
3. **Imagined speech**: A 5-second phase during which the subject during which the subjects silently imagined speaking the prompt without any verbal articulation or physical movement.
4. **Speaking**: A 2-second period during which the subject spoke the prompt aloud, while the Kinect sensor recorded the facial animation units and the microphone recorded the audio signals.

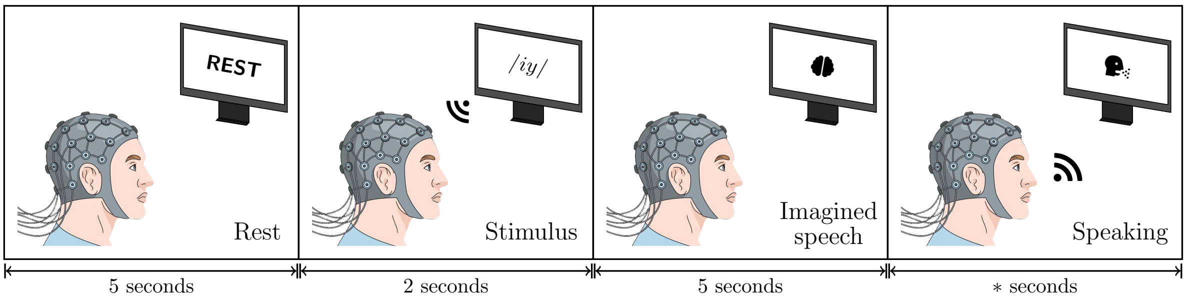


Figure 2.1: A visual representation of the recording procedure used in the KaraOne database. Each trial for each subject consisted of four states: rest, stimulus, imagined speech, and speaking.

2.4.2 FEIS dataset

The FEIS dataset [2] was collected by [26] using a cost-effective acquisition system, viz., a 14-channel Emotiv EPOC+ headset with dry electrodes with a sampling frequency of 256 Hz. The dataset consists of 16 phonemic prompts, with 10 repetitions of each prompt per participant among the 21 participants. The data was preprocessed using a notch flter using the built-in Emotiv software at 50 Hz and 60 Hz to remove powerline noise. No signal preprocessing was carried out to remove physiological artifacts (such as blinks or saccades) due to unavailibility of the ocular channels in the headset. The data was then segmented into 5-second imagined speech EEG epochs, with a 10% overlap between consecutive windows. The preprocessing steps are similar to the one carried out in the KaraOne dataset.

[2]Available at https://doi.org/10.5281/zenodo.3554128

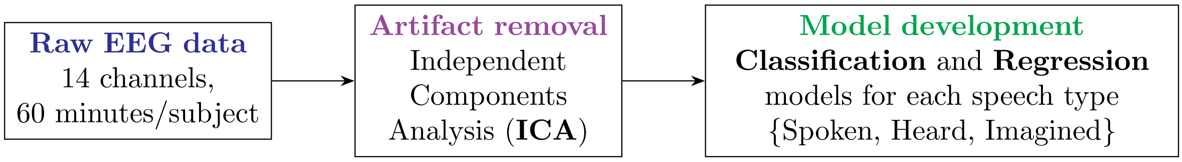


Figure 2.2: Illustration of the model-building procedure for the FEIS dataset.

Table 2.2: Comparison of KaraOne and FEIS datasets

	KaraOne [27]	**FEIS** [26]
EEG device	64 channels	14 channels
Sampling rate	1000 Hz	256 Hz
Participants	14 participants	21 participants
Duration	30 - 40 minutes	60 minutes
Prompts	7 phonemes + 4 words	16 phonemes

2.5 Summarised outcomes

- Electroencephalography (EEG) records brain electrical activity non-invasively. EEG waveforms are grouped into ‘Delta’, ‘Theta’, ‘Alpha’, ‘Beta’, and ‘Gamma’ bands according to their frequency ranges.
- Preprocessing techniques like filtering and downsampling are essential to enhance EEG signal quality for analysis, enabling meaningful feature extraction in EEG-based studies.
- Imagined speech, also known as covert speech, involves mental representation of words or concepts without verbalizing. In BCI research, a typical pipeline includes signal acquisition, applying relevant signal processing techniques, feature extraction in a particular domain and classification.
- Classification of imagined speech utilising classical machine learning models (e.g., SVM, KNN, RF) and deep learning models (e.g., CNN, LSTM) have been carried out in literature [17], with varying results.
- Popular datasets for imagined speech classification include the KaraOne database, featuring 4 words and 7 phonemic/syllabic prompts, and the cost-effective FEIS dataset.
- Information sets, exemplified in studies [1, 19, 18], enhance feature extraction by quantifying information and improving classification accuracy across various applications.

2.6 Objectives

In this work, we look into the following main **objectives**:

- Conduct a classification task utilising the EEG Imagined speech dataset with the extracted features and evaluate the performance of the proposed method, comparing it with existing techniques.
- Extract rich spatio-temporal features using Information Set theory techniques to capture more information effectively and improve classification.

CHAPTER 3

METHODOLOGY

This chapter contains the methodology used in the proposed work. Fig. 3.1 shows the flowchart of the methodology, which comprises three main parts: preprocessing of the data, feature extraction and the classification tasks. The data preprocessing involves filtering the data, filtering noise and windowing the data. Feature extraction involves extracting the features from the data, resulting in a feature matrix that is used for classification. The classification part involves training the machine learning model and evaluating its performance.

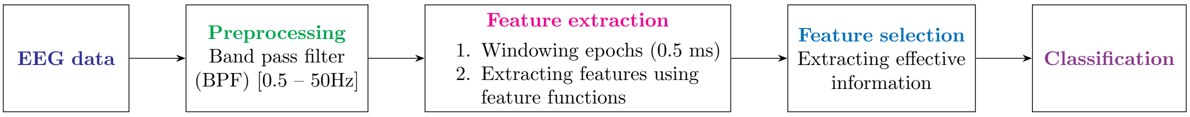


Figure 3.1: Methodology flowchart for the proposed work.

3.1 Dataset

This study focuses on the 5-second imagined speech segment, during which participants imagines speaking the prompt without any associated physical movements. The recording procedure has been given in Section 2.4.1.

Eight men and four women with a mean age of 27 ± 5 years [27] were selected from the University of Toronto to participate in the study. They were all free of neurological disorders and drug addiction, and none of them had any history of visual, auditory, or motor impairments. They also had some post-secondary education and were right-handed dominant. Given that the study was carried out in English, the participants' English language competency was also considered. Two participants fluently spoke North American English, having learnt it at an average age of six. Ten participants stated that North American English was their first language. The 14 subjects are labelled as {'MM05', 'MM08', 'MM09', 'MM10', 'MM11', 'MM12', 'MM14', 'MM15', 'MM16', 'MM18', 'MM19', 'MM20', 'MM21', 'P02'} respectively, out of which KaraOne only uses 12, while all 14 are used in this study.

3.1.1 Preprocessing

A bandpass filter filtered out the EEG signals into the range of $0.5 - 30$Hz. Baseline correction was applied to the EEG data to remove DC offset. A laplacian filter using the neighbourhood of adjacent channels was not used due to the possible loss of important EEG information [17]. The preprocessing workflow is shown in Figure 3.2.

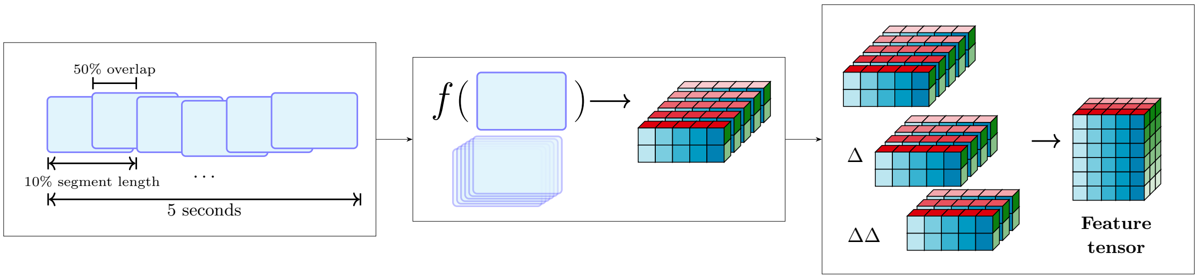


Figure 3.2: Preprocessing workflow: (a). Windowing the data. (b). Extracting features from the windows. (c). Addition of delta and double delta features.

3.1.2 Feature extraction

Windowing was performed over each imagined speech segment of 5 seconds, with each window length being 10% of the segment length and with 50% overlap between consecutive windows. This gave 19 windows per segment, with each window length being about 500 ms. The assumption is that the signals in each window will be statistically stationary over that period, which is useful for feature extraction as the features can be assumed to be time-invariant over this period. This was done to capture the temporal dynamics of the signal, as EEG signals are inherently non-stationary in nature.

A set of statistical features and few entropy functions were computed on the windows, giving a total of 27 features per window. The choice of the features used was based on [26] and [27]. The feature functions are described in Table A.2, and the extracted features summary is given in Table 3.1. In addition, the delta (differential) and double-delta (acceleration) features are also computed, resulting in tripling the number of features per window with very little computational overhead, thereby giving a total of $27 \times 3 = 81$ features per window across the 62 channels in the dataset. The initial two windows were discarded to accommodate the inclusion of these additional features. These features approximate the first and second derivates of the signal, respectively, in a simple manner without much computational overload and are useful in capturing the dynamics of the signal. The differentiators tend to amplify the noise in the signal, which is partly mitigated by considering the double differentiators. This gives a total of $17 \times 62 \times 81 = 85,374$ features per segment/epoch.

The procedure for extracting the effective information is given as follows. Consider the feature matrix X_f for the f^{th} feature in (Eqn. 3.1), which is considered as an information source matrix. The features are considered as information source values comprising an information set

Table 3.1: Features extracted from each subject in the KaraOne database

Subject	**Features shape**	**Labels shape**
MM05	(165, 17, 62, 81)	(165,)
MM08	(131, 17, 62, 81)	(131,)
MM09	(132, 17, 62, 81)	(132,)
MM10	(132, 17, 62, 81)	(132,)
MM11	(132, 17, 62, 81)	(132,)
MM12	(132, 17, 62, 81)	(132,)
MM14	(132, 17, 62, 81)	(132,)
MM15	(132, 17, 62, 81)	(132,)
MM16	(132, 17, 62, 81)	(132,)
MM18	(132, 17, 62, 81)	(132,)
MM19	(132, 17, 62, 81)	(132,)
MM20	(132, 17, 62, 81)	(132,)
MM21	(132, 17, 62, 81)	(132,)
P02	(165, 17, 62, 81)	(165,)

along the temporal and spatial dimensions.

$$X_f = \begin{bmatrix} x_{11f} & \cdots & x_{1cf} & \cdots & x_{1Cf} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{w1f} & \cdots & x_{wcf} & \cdots & x_{wCf} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{W1f} & \cdots & x_{Wcf} & \cdots & x_{WCf} \end{bmatrix}_{W \times C} \qquad \begin{array}{c} 1 \leq w \leq W \\ 1 \leq c \leq C \\ 1 \leq f \leq F \end{array} \tag{3.1}$$

The uncertainity across the temporal and spatial dimension are computed separately using a Gaussian membership function (Eqn. 3.2, Eqn. 3.3) as follows:

$$G_t\left(x_{wcf}\right) = \exp\left\{-\frac{1}{2}\left[\frac{x_{wcf} - \mu_{cf}}{\sigma_{cf}}\right]^2\right\}, \qquad \begin{array}{c} 1 \leq c \leq C \\ 1 \leq f \leq F \end{array} \tag{3.2}$$

$$G_s\left(x_{wcf}\right) = \exp\left\{-\frac{1}{2}\left[\frac{x_{wcf} - \mu_{wf}}{\sigma_{wf}}\right]^2\right\}, \qquad \begin{array}{c} 1 \leq w \leq W \\ 1 \leq f \leq F \end{array} \tag{3.3}$$

where μ_{cf} and σ_{cf} are the temporal mean and standard deviation calculated across the windows (Eqn. 3.4), while μ_{wf} and σ_{wf} are the spatial mean and standard deviation calculated across the channels (Eqn. 3.5).

$$\mu_{cf} = \frac{1}{W}\sum_{w=1}^{W} x_{wcf}, \quad \sigma_{cf} = \sqrt{\frac{1}{W}\sum_{w=1}^{W}\left(x_{wcf} - \mu_{cf}\right)^2} \tag{3.4}$$

$$\mu_{wf} = \frac{1}{C}\sum_{c=1}^{C} x_{wcf}, \quad \sigma_{wf} = \sqrt{\frac{1}{C}\sum_{c=1}^{C}\left(x_{wcf} - \mu_{wf}\right)^2} \tag{3.5}$$

The choice of the Gaussian membership function helps to determine the uncertainity in the source values, which is then used to extract the effective information.

The temporal and spatial fold informations (Eqn. 3.6) are then obtained by extracting information from the uncertainities in the source values, therby considering the set $\left\{I_f^{\lambda}\left(x_{wcf}\right)\right\}$ as an information set, which is given by,

$$I_f^{\lambda}\left(x_{wcf}\right) = x_{wcf} \cdot G_{\lambda}\left(x_{wcf}\right), \quad \lambda = \{t, s\} \tag{3.6}$$

where λ corresponds to the folds, representing the temporal (t) and spatial (s) folds respectively.

The fused information for a given window w and channel c is then computed by summing the fold informations across the corresponding window and channel (Eqn. 3.7).

$$\mathscr{I}_f\left(x_{wcf}\right) = I_f^t\left(x_{wcf}\right) + I_f^s\left(x_{wcf}\right), \quad \begin{array}{c} 1 \le w \le W \\ 1 \le c \le C \\ 1 \le f \le F \end{array} \tag{3.7}$$

The normalised effective information is then computed by taking the mean at each fused information source value along the temporal and spatial folds (Eqn. 3.8).

$$E_f = \frac{1}{WC}\sum_{w=1}^{W}\sum_{c=1}^{C} \mathscr{I}_f\left(x_{wcf}\right) \tag{3.8}$$

This gives the following $F \times 1$ effective information vector $\mathbf{E}$,

$$\mathbf{E}^T = \begin{bmatrix} E_1 & \cdots & E_f & \cdots & E_F \end{bmatrix}_{1 \times F} \tag{3.9}$$

These are the effective features that are used to represent the spatio-temporal information, thus effectively reducing the $[W \times C \times F]$ feature matrix to a F-length feature vector (Eqn. 3.8). The effective information extraction is shown in Figure 3.3. This procedure is summarised as a pseudocode in Algorithm 1.

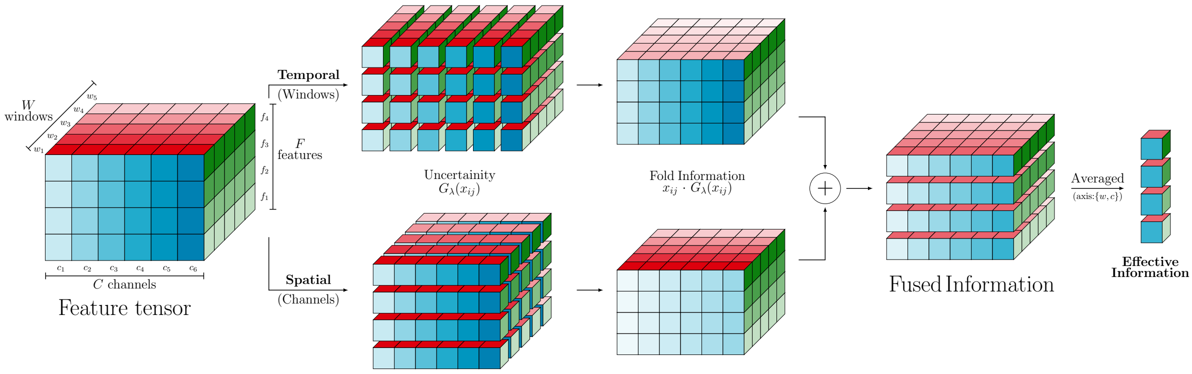


Figure 3.3: Proposed methodology: Extracting effective information from the features across both temporal and spatial dimensions resulting in rich spatio-temporal features. The two folds of information are separately computed across the temporal and spatial dimensions by extracting the uncertainty in the source values. These are then fused and averaged along the temporal and spatial dimensions to obtain the effective feature vectors.

Algorithm 1: Effective Information

Input: $[S \times W \times C \times F]$ feature tensor with S epochs, W windows, C channels, and F features

foreach *epoch in feature tensor* **do**

Compute the temporal and spatial fold informations;

$G_t\left(x_{wcf}\right) = \exp\left\{-\frac{1}{2}\left[\frac{x_{wcf}-\mu_{cf}}{\sigma_{cf}}\right]^2\right\}$, $G_s\left(x_{wcf}\right) = \exp\left\{-\frac{1}{2}\left[\frac{x_{wcf}-\mu_{wf}}{\sigma_{wf}}\right]^2\right\}$,

$I_f^\lambda\left(x_{wcf}\right) = x_{wcf} \cdot G_\lambda\left(x_{wcf}\right), \lambda = \{t, s\}$;

Compute the fused information from the fold informations;

$\mathscr{I}_f\left(x_{wcf}\right) = I_f^t\left(x_{wcf}\right) + I_f^s\left(x_{wcf}\right)$;

Compute the effective information vector $\mathbf{E}$;

$E_f = \frac{1}{WC}\sum_{w=1}^{W}\sum_{c=1}^{C}\mathscr{I}_f\left(x_{wcf}\right)$;

Output: Feature vector of length $[F]$ for the epoch

end

Result: Effective feature matrix of dimensions $[S \times F]$

3.2 Model

Four different classifiers were used to perform the classification tasks: Hanman Classifier (HC), Decision Tree Classifier (DT) Classifier, Random Forest Classifier (RF), and Support Vector Machine (SVM). Table 3.2 provides the details of the parameters for the models.

Hanman Classifier: The Hanman Classifier [19, 18] is based on Information sets. It works by computing the uncertainty of the minimum aggregated normed errors between the test sample and all the training samples for each class and then identifies the class with the lowest uncertainty as the predicted class for the test sample. The aggregation of the errors between the test and trainig samples is done using the Frank t-norm to assess similarity or dissimilarity between them. The pseudocode for the Hanman Classifier is shown in Algorithm 2.

Algorithm 2: Hanman Classifier

```
Input: Train samples, Test samples
Normalise samples along features axis;
foreach sample in test samples do
    foreach sample in train samples do
        Compute the error between the training sample and the input test sample;
    end
    foreach class do
        foreach pair of errors in class do
            Compute the t-norm of the error vectors;
        end
        Compute the minimum of the normed error pair;
        Compute the possibilistic uncertainty associated with the minimum pair;
    end
    Compute the class with the minimum uncertainty;
    Output: Predicted class for the test sample
end
Result: Predicted classes for the test samples
```

Decision Tree Classifier: The Decision Tree Classifier [22] is a straightforward and extensively employed machine learning approach. It works by approximating the features and learning them by a series of if-then rules, by recursively dividing the data into subsets based on features that maximise the information gain. Generally, the information gain is computed using either the Gini impurity function or the Shannon entropy function, given by $E_{Gi} = \sum_k p_k(1 - p_k)$ and $E_{Sh} = -\sum_k p_k \log p_k$, respectively. The complexity of a Decision Tree is closely related to it's depth. Decision Trees have a drawback that they tend to overfit and do not generalise well, and are volatile to changes in the data.

Random Forest Classifier: The Random Forest Classifier [5] is a powerful ensemble method that works by creating a whole forest of Decision Trees. The forest's outcome relies on a majority-voting principle, where it selects the most predicted class (the mode of the predictions) from the ensemble. By aggregating the predictions of multiple trees, Random Forest can achieve higher accuracy and better generalization compared to a single Decision Tree. It can handle large datasets and is robust to overfitting, overcoming the problems in Decision Trees. An illustration of a general Random Forest Classifier is shown in Figure 3.4.

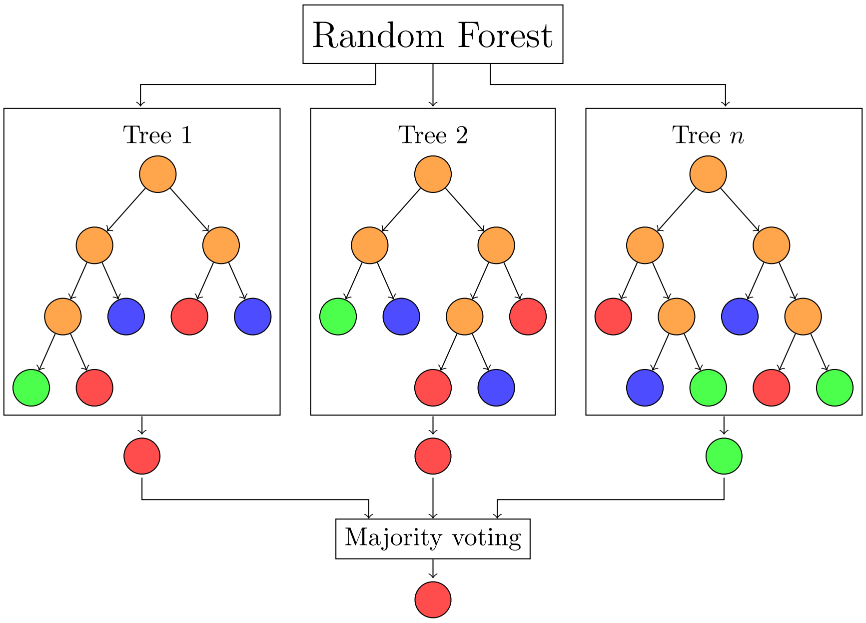


Figure 3.4: Random Forest Classifier. The classifier is an ensemble of multiple Decision Trees.

Support Vector Machines: SVMs [3, 8, 6] operate by identifying the optimal hyperplane to separate data into distinct classes effectively. This involves finding a hyperplane with the maximum margin between classes, thereby maximizing the separation between the hyperplane and the closest data points from each class. SVMs excel in high-dimensional areas and can achieve nonlinear classification by utilizing kernel functions that convert input data into feature spaces of higher dimensions. An illustration of an SVM is shown in Figure 3.5.

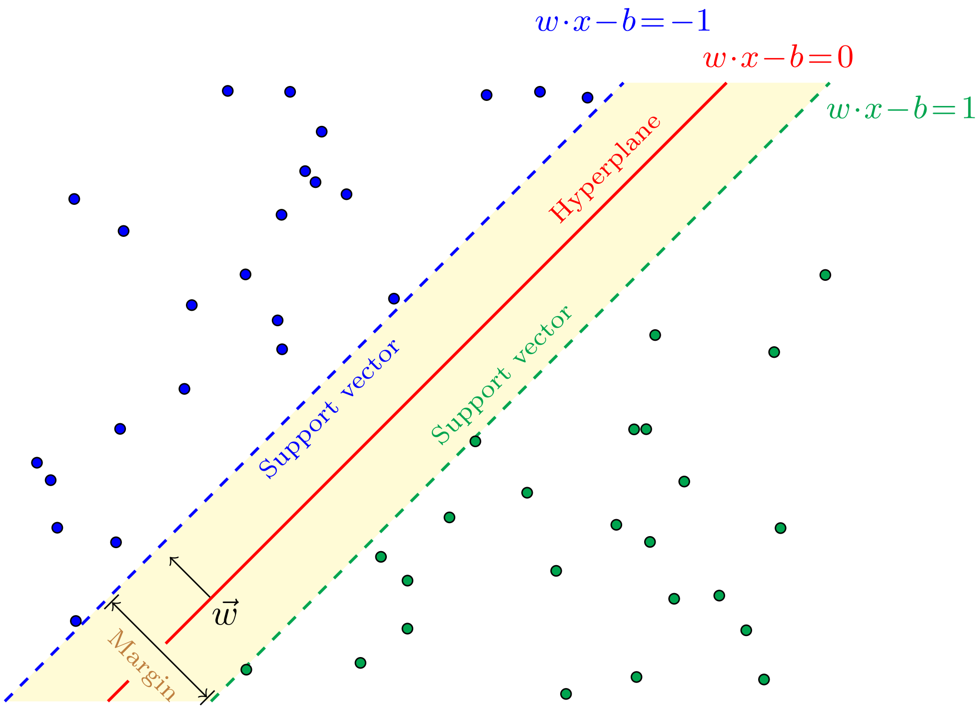


Figure 3.5: Support Vector Machine: The classifier finds the hyperplane that best separates the data into different classes.

3.3 Classification

For classification, the labels were converted to binary classes to allow for a binary classification task. Only the thinking segment of the data was used after preprocessing. Five binary classification tasks of phonological categories were performed, as in the KaraOne paper [27]: 'consonant vs. vowel-only (C/V)', presence of 'nasal ($\pm$ Nasal)', 'bilabial ($\pm$ Bilabial)', 'high-front vowel ($\pm$/iy/)', and 'high-back vowel ($\pm$/uw/)', respectively. The classification of the mental states of the speaker and multi-class classification tasks falls beyond the scope of this study.

3.3.1 Class imbalance

In addition, the dataset was resampled to balance the classes for each binary task. Even though the trials had roughly the same number of samples for each trial, and hence for each class, the binary labels for the tasks were imbalanced. The class imbalance in each of the tasks was handled by resampling the dataset to account for the missing samples in the minority class,.i.e., oversampling techniques were used to balance the classes. The dataset was resampled to balance the classes for each binary task using different oversamplers: Random Over Sampler, ADASYN and SMOTE (See Section 3.4). In addition, the case where no resampler used was also considered, to compare the results with the resampled datasets. The resampling was done on the training set only, to prevent any possible data leakage from the test set.

3.3.2 Evaluation metrics and Model parameters

Various classification metrics were used to evaluate the model: 'Accuracy', 'F1-score', 'Precision' and 'Recall'. Hyperparameter tuning was performed with a grid search along with cross-validation (CV). The CV strategy chosen was stratified k-fold performed with $k = 5$ to ensure that each fold had the same class distribution as the original dataset. The training split in each fold was resampled after the split, and the validation set is kept free from the resampled data to disallow any data leakage during CV. The metrics are reported as the average across the folds, along with the standard deviation.

3.4 Sampling

The different oversamplers are discussed here: Random Over Sampler, SMOTE and ADASYN.

3.4.1 Random Over Sampler

The Random Over Sampler (ROS) is a simple oversampling method that addresses class imbalance by randomly selecting and replicating samples from the minority class until it matches

Table 3.2: Parameters for the different models

Model	Parameters
Hanman Classifier	`alpha=0.5, beta=1, a=1, b=0, q=2`
Decision Tree Classifier	`min_samples_split=2, max_depth=None, criterion='gini'`
Random Forest Classifier	`max_depth=100, n_estimators=100, class_weight='balanced'`
Support Vector Machines	`C=1, kernel='rbf', gamma='scale', shrinking=True`

the number of samples in the majority class. This technique over-samples by duplicating original minority class samples, thereby, care must be taken to avoid overfitting.

3.4.2 SMOTE and ADASYN

'Synthetic Minority Over-sampling Technique (SMOTE)' [7] and 'Adaptive Synthetic Sampling (ADASYN)' [12] are oversampling techniques designed to address class imbalance by generating synthetic samples for the minority class. SMOTE interpolates between minority class samples to create synthetic instances, while ADASYN builds on SMOTE by considering the density distribution of the minority class. In SMOTE, a new sample x_{new} is generated by selecting one of the k nearest neighbors of x_i, denoted as x_{zi}, and computing:

$$x_{new} = x_i + \lambda \times (x_{zi} - x_i)$$

where λ is a random number between 0 and 1. The number of synthetic samples generated is proportional to the ratio of minority class to majority class samples.

ADASYN enhances SMOTE by focusing on samples near the decision boundary of the classifier, identified using an internal k-Nearest Neighbors classifier, to generate synthetic samples.

CHAPTER 4
RESULTS

4.1 Performance metric analysis

The classification metrics on the five binary classification tasks for different classifiers, along with the use of different samplers, have been reported in Table 4.1. The corresponding confusion matrices have been reported in Table 4.2. The best performing model is the Random Forest classifier sampled with the Random over sampler yielding average accuracies of $60\% - 90\%$ in the tasks, which is a significant improvement over the KaraOne SVM classifiers [27]. Overall, an improvement in the performance over the KaraOne baseline models is observed, given that some of its pre-processing steps were not carried out in this study.

The results in Table 4.1 show that there is a lot of variation among the different classifiers and samplers combinations. Since the effective features rely on the uncertainities in the EEG data, the classifiers that can handle them better are naturally able to perform better. The results also show that the use of different samplers has a significant impact on the performance of the classifiers. The ANOVA analysis of the features, given in Table A.1, shows that a majority of the features used are not much correlated, with Pearson correlation values tending to be near zero. This suggests the use of a large number of features, which are mostly uncorrelated, and hence, the use of samplers to balance the classes is essential.

It can be observed that the case with using the Hanman classifier with No Sampler (NS) and Random Over Sampler (ROS) have identical metrics due to the classifier computing the least possibilitic uncertainty among all the pairs of norm errors among all the classes. Since ROS doesn't generate new samples but only duplicates the existing ones randomly, the classifier doesn't see any new samples and hence the metrics are identical. The significant changes while using SMOTE and ADASYN samplers suggests that the Hanman classifier is quite sensistive to dataset changes due to sampling.

The RF classifier perfoms the best due to its robustness in handling the uncertainities in the data by inherently averaging multiple decision trees which helps reduce variance and overfitting.

The results in Table 4.3 show that the proposed methodology outperforms the KaraOne SVM classifiers [27] in all the tasks except for the Task 2 ($\pm$ Nasal), with RF+ROS performing the best among all the models.

Table 4.1: Performance metrics for different tasks across different classifiers and samplers. Each cell contains the mean and standard deviation of the metric across the 5 folds.
NS: No Sampler, ROS: Random Over Sampler, SMOTE, ADASYN.

Task	Metric (%)	Hanman Classifier				Decision Tree Classifier			
		NS	ROS	SMOTE	ADASYN	NS	ROS	SMOTE	ADASYN
C/V	Accuracy	77.35 ± 1.39	77.35 ± 1.39	54.08 ± 2.00	54.15 ± 3.34	69.55 ± 1.72	69.55 ± 0.78	66.27 ± 2.38	63.48 ± 1.68
	F1	87.02 ± 0.82	87.02 ± 0.82	65.67 ± 1.95	66.54 ± 3.33	80.95 ± 1.22	81.31 ± 0.57	77.90 ± 1.97	75.69 ± 1.54
	Precision	81.96 ± 0.72	81.96 ± 0.72	84.57 ± 1.18	82.44 ± 1.73	82.94 ± 0.79	81.71 ± 0.68	83.91 ± 0.80	83.12 ± 0.46
	Recall	92.77 ± 1.11	92.77 ± 1.11	53.70 ± 2.27	55.91 ± 4.53	79.06 ± 1.89	80.94 ± 1.24	72.77 ± 3.18	69.53 ± 2.65
±Nasal	Accuracy	54.63 ± 2.19	54.63 ± 2.19	48.92 ± 2.17	48.71 ± 3.23	55.89 ± 2.47	55.61 ± 2.38	54.70 ± 2.72	54.08 ± 2.05
	F1	31.85 ± 2.44	31.85 ± 2.44	40.44 ± 2.63	40.79 ± 2.22	41.53 ± 4.67	39.52 ± 3.85	41.92 ± 3.77	39.79 ± 2.79
	Precision	35.03 ± 3.11	35.03 ± 3.11	35.01 ± 2.20	35.16 ± 2.60	39.85 ± 3.69	39.04 ± 3.21	39.16 ± 3.33	37.91 ± 2.49
	Recall	29.23 ± 2.07	29.23 ± 2.07	47.88 ± 3.41	48.65 ± 1.98	43.46 ± 6.00	40.19 ± 5.14	45.19 ± 4.75	41.92 ± 3.47
±Bilabial	Accuracy	58.05 ± 2.62	58.05 ± 2.62	51.85 ± 3.40	52.68 ± 2.11	57.49 ± 2.73	55.96 ± 1.98	57.07 ± 1.07	54.36 ± 2.82
	F1	34.94 ± 4.20	34.94 ± 4.20	44.99 ± 2.78	44.90 ± 2.63	40.70 ± 4.81	40.14 ± 1.62	44.98 ± 1.72	42.58 ± 5.47
	Precision	39.92 ± 4.66	39.92 ± 4.66	38.51 ± 2.54	38.85 ± 2.06	41.05 ± 4.32	39.65 ± 1.88	41.99 ± 1.36	39.00 ± 4.08
	Recall	31.15 ± 4.15	31.15 ± 4.15	54.42 ± 5.11	53.27 ± 4.11	40.38 ± 5.30	40.77 ± 2.55	48.46 ± 2.32	47.12 ± 7.86
±/iy/	Accuracy	57.84 ± 1.96	57.84 ± 1.96	52.75 ± 2.20	54.56 ± 2.54	59.16 ± 1.35	61.25 ± 3.60	57.00 ± 1.39	57.42 ± 1.80
	F1	34.87 ± 1.63	34.87 ± 1.63	43.43 ± 1.72	45.43 ± 1.84	45.33 ± 3.17	47.58 ± 5.21	43.74 ± 1.20	44.66 ± 2.72
	Precision	39.79 ± 2.67	39.79 ± 2.67	38.40 ± 1.91	40.31 ± 2.32	43.97 ± 1.97	46.65 ± 4.71	41.65 ± 1.37	42.22 ± 2.23
	Recall	31.15 ± 1.98	31.15 ± 1.98	50.00 ± 1.72	52.12 ± 1.65	46.92 ± 4.99	48.65 ± 6.19	46.15 ± 2.51	47.50 ± 3.97
±/uw/	Accuracy	89.41 ± 1.11	89.41 ± 1.11	52.89 ± 0.41	53.24 ± 3.17	80.77 ± 2.11	82.65 ± 1.27	70.59 ± 4.20	72.20 ± 2.22
	F1	5.17 ± 4.73	5.17 ± 4.73	13.54 ± 1.58	14.58 ± 1.60	9.98 ± 3.78	9.98 ± 3.25	12.50 ± 5.05	12.47 ± 4.81
	Precision	16.67 ± 13.94	16.67 ± 13.94	8.12 ± 0.93	8.75 ± 1.02	8.86 ± 3.54	9.39 ± 2.95	8.74 ± 3.94	8.70 ± 3.29
	Recall	3.08 ± 2.88	3.08 ± 2.88	40.77 ± 5.22	43.85 ± 3.92	11.54 ± 4.21	10.77 ± 3.77	22.31 ± 6.62	22.31 ± 9.55

Task	Metric (%)	Random Forest Classifier				Support Vector Machine Classifier			
		NS	ROS	SMOTE	ADASYN	NS	ROS	SMOTE	ADASYN
C/V	Accuracy	81.74 ± 0.36	81.60 ± 0.34	76.86 ± 1.05	76.24 ± 1.53	81.88 ± 0.00	53.38 ± 22.34	53.87 ± 24.88	75.47 ± 4.15
	F1	89.95 ± 0.21	89.85 ± 0.20	86.65 ± 0.58	86.24 ± 0.96	90.04 ± 0.00	58.30 ± 30.50	56.48 ± 34.65	85.66 ± 2.90
	Precision	81.90 ± 0.14	81.92 ± 0.21	82.16 ± 0.78	81.98 ± 0.75	81.88 ± 0.00	81.72 ± 0.17	84.08 ± 3.46	81.71 ± 0.32
	Recall	99.74 ± 0.34	99.49 ± 0.42	91.66 ± 0.79	90.98 ± 1.75	100.00 ± 0.00	55.57 ± 35.27	56.00 ± 39.64	90.21 ± 6.16
±Nasal	Accuracy	62.65 ± 1.35	60.98 ± 1.15	58.82 ± 1.75	57.42 ± 1.77	63.69 ± 0.14	57.28 ± 7.51	49.83 ± 10.17	44.74 ± 9.02
	F1	16.10 ± 2.50	24.63 ± 2.95	36.75 ± 5.63	36.00 ± 3.81	0.00 ± 0.00	19.31 ± 16.12	33.61 ± 21.22	43.52 ± 17.51
	Precision	45.39 ± 6.85	41.12 ± 3.20	41.18 ± 3.81	39.52 ± 2.62	0.00 ± 0.00	35.28 ± 4.86	35.85 ± 3.51	37.12 ± 0.75
	Recall	10.00 ± 2.16	17.69 ± 2.76	33.46 ± 7.00	33.27 ± 4.96	0.00 ± 0.00	22.12 ± 29.94	51.92 ± 38.88	73.27 ± 34.49
±Bilabial	Accuracy	63.83 ± 0.97	62.86 ± 1.00	60.35 ± 0.74	61.60 ± 2.56	63.76 ± 0.00	52.06 ± 11.68	47.80 ± 9.72	54.91 ± 9.18
	F1	20.67 ± 2.65	27.18 ± 3.13	37.62 ± 1.82	39.88 ± 3.75	0.00 ± 0.00	26.27 ± 21.39	36.94 ± 17.65	25.06 ± 14.58
	Precision	51.28 ± 5.93	46.99 ± 3.09	43.80 ± 1.10	46.24 ± 4.50	0.00 ± 0.00	33.79 ± 4.36	35.00 ± 3.84	33.86 ± 4.04
	Recall	13.08 ± 2.16	19.23 ± 2.85	33.08 ± 2.76	35.19 ± 3.83	0.00 ± 0.00	40.96 ± 43.91	56.54 ± 38.00	29.62 ± 34.00
±/iy/	Accuracy	66.06 ± 1.42	64.95 ± 2.21	63.41 ± 3.65	63.48 ± 1.78	63.76 ± 0.00	51.01 ± 11.67	54.98 ± 7.83	56.03 ± 10.13
	F1	31.80 ± 4.18	39.61 ± 3.36	44.94 ± 5.13	45.17 ± 2.58	0.00 ± 0.00	30.11 ± 17.70	28.87 ± 12.17	24.33 ± 14.23
	Precision	58.17 ± 4.54	52.93 ± 5.00	49.78 ± 5.70	49.59 ± 2.85	0.00 ± 0.00	38.41 ± 8.87	36.03 ± 6.73	38.07 ± 8.98
	Recall	21.92 ± 3.35	31.73 ± 2.92	41.35 ± 5.86	41.54 ± 2.88	0.00 ± 0.00	42.88 ± 39.04	31.54 ± 28.69	28.27 ± 33.99
±/uw/	Accuracy	90.94 ± 0.00	90.94 ± 0.00	88.78 ± 0.95	88.78 ± 0.95	90.94 ± 0.00	28.78 ± 24.50	27.80 ± 23.63	40.14 ± 31.10
	F1	0.00 ± 0.00	0.00 ± 0.00	4.55 ± 2.29	4.78 ± 2.44	0.00 ± 0.00	15.23 ± 2.07	16.81 ± 0.77	13.46 ± 3.86
	Precision	0.00 ± 0.00	0.00 ± 0.00	9.02 ± 4.95	13.17 ± 10.96	0.00 ± 0.00	8.89 ± 0.14	9.82 ± 1.55	8.28 ± 1.03
	Recall	0.00 ± 0.00	0.00 ± 0.00	3.08 ± 1.54	3.08 ± 1.54	0.00 ± 0.00	74.62 ± 29.65	79.23 ± 24.40	60.77 ± 40.29

Table 4.2: Confusion matrices for different tasks across different classifiers and samplers. Each cell contains the 5 fold averaged confusion matrix values along with the corresponding standard deviation. NS: No Sampler, ROS: Random Over Sampler, SMOTE, ADASYN.

Task	Decision Tree Classifier							
	NS		ROS		SMOTE		ADASYN	
Task-0	13.80 ±1.83	38.20 ±1.83	9.40 ±2.33	42.60 ±2.33	19.20 ±2.14	32.80 ±2.14	18.80 ±1.83	33.20 ±1.83
	49.20 ±4.45	185.80 ±4.45	44.80 ±2.93	190.20 ±2.93	64.00 ±7.48	171.00 ±7.48	71.60 ±6.22	163.40 ±6.22
Task-1	115.20 ±3.97	67.80 ±3.97	117.80 ±6.79	65.20 ±6.79	110.00 ±5.83	73.00 ±5.83	111.60 ±4.18	71.40 ±4.18
	58.80 ±6.24	45.20 ±6.24	62.20 ±5.34	41.80 ±5.34	57.00 ±4.94	47.00 ±4.94	60.40 ±3.61	43.60 ±3.61
Task-2	123.00 ±2.90	60.00 ±2.90	118.20 ±6.91	64.80 ±6.91	113.40 ±1.96	69.60 ±1.96	107.00 ±6.45	76.00 ±6.45
	62.00 ±5.51	42.00 ±5.51	61.60 ±2.65	42.40 ±2.65	53.60 ±2.42	50.40 ±2.42	55.00 ±8.17	49.00 ±8.17
Task-3	121.00 ±4.24	62.00 ±4.24	125.20 ±6.05	57.80 ±6.05	115.60 ±5.71	67.40 ±5.71	115.40 ±4.88	67.60 ±4.88
	55.20 ±5.19	48.80 ±5.19	53.40 ±6.44	50.60 ±6.44	56.00 ±2.61	48.00 ±2.61	54.60 ±4.13	49.40 ±4.13
Task-4	228.80 ±5.34	32.20 ±5.34	234.40 ±4.03	26.60 ±4.03	196.80 ±10.55	64.20 ±10.55	201.40 ±7.23	59.60 ±7.23
	23.00 ±1.10	3.00 ±1.10	23.20 ±0.98	2.80 ±0.98	20.20 ±1.72	5.80 ±1.72	20.20 ±2.48	5.80 ±2.48

Task	Random Forest Classifier								Support Vector Machine Classifier							
	NS		ROS		SMOTE		ADASYN		NS		ROS		SMOTE		ADASYN	
Task-0	0.20 ±0.40	51.80 ±0.40	0.40 ±0.80	51.60 ±0.80	5.20 ±2.48	46.80 ±2.48	5.00 ±2.37	47.00 ±2.37	0.00 ±0.00	52.00 ±0.00	22.60 ±18.77	29.40 ±18.77	23.00 ±21.77	29.00 ±21.77	4.60 ±2.73	47.40 ±2.73
	0.60 ±0.80	234.40 ±0.80	1.20 ±0.98	233.80 ±0.98	19.60 ±1.85	215.40 ±1.85	21.20 ±4.12	213.80 ±4.12	0.00 ±0.00	235.00 ±0.00	104.40 ±82.88	130.60 ±82.88	103.40 ±93.16	131.60 ±93.16	23.00 ±14.48	212.00 ±14.48
Task-1	169.40 ±5.95	13.60 ±5.95	156.60 ±4.50	26.40 ±4.50	134.00 ±5.14	49.00 ±5.14	130.20 ±6.40	52.80 ±6.40	182.80 ±0.40	0.20 ±0.40	141.40 ±52.47	41.60 ±52.47	89.00 ±69.49	94.00 ±69.49	52.20 ±61.69	130.80 ±61.69
	93.60 ±2.24	10.40 ±2.24	85.60 ±2.87	18.40 ±2.87	69.20 ±7.28	34.80 ±7.28	69.40 ±5.16	34.60 ±5.16	104.00 ±0.00	0.00 ±0.00	81.00 ±31.14	23.00 ±31.14	50.00 ±40.44	54.00 ±40.44	27.80 ±35.87	76.20 ±35.87
Task-2	169.60 ±4.08	13.40 ±4.08	160.40 ±3.88	22.60 ±3.88	138.80 ±4.26	44.20 ±4.26	140.20 ±6.11	42.80 ±6.11	183.00 ±0.00	0.00 ±0.00	106.80 ±79.15	76.20 ±79.15	78.40 ±67.27	104.60 ±67.27	126.80 ±61.63	56.20 ±61.63
	90.40 ±2.24	13.60 ±2.24	84.00 ±2.97	20.00 ±2.97	69.60 ±2.87	34.40 ±2.87	67.40 ±3.98	36.60 ±3.98	104.00 ±0.00	0.00 ±0.00	61.40 ±45.67	42.60 ±45.67	45.20 ±39.52	58.80 ±39.52	73.20 ±35.36	30.80 ±35.36
Task-3	166.80 ±1.17	16.20 ±1.17	153.40 ±5.00	29.60 ±5.00	139.00 ±9.36	44.00 ±9.36	139.00 ±4.05	44.00 ±4.05	183.00 ±0.00	0.00 ±0.00	101.80 ±73.98	81.20 ±73.98	125.00 ±51.49	58.00 ±51.49	131.40 ±63.93	51.60 ±63.93
	81.20 ±3.49	22.80 ±3.49	71.00 ±3.03	33.00 ±3.03	61.00 ±6.10	43.00 ±6.10	60.80 ±2.99	43.20 ±2.99	104.00 ±0.00	0.00 ±0.00	59.40 ±40.60	44.60 ±40.60	71.20 ±29.84	32.80 ±29.84	74.60 ±35.35	29.40 ±35.35
Task-4	261.00 ±0.00	0.00 ±0.00	261.00 ±0.00	0.00 ±0.00	254.00 ±3.03	7.00 ±3.03	254.00 ±2.61	7.00 ±2.61	261.00 ±0.00	0.00 ±0.00	63.20 ±78.02	197.80 ±78.02	59.20 ±74.12	201.80 ±74.12	99.40 ±99.72	161.60 ±99.72
	26.00 ±0.00	0.00 ±0.00	26.00 ±0.00	0.00 ±0.00	25.20 ±0.40	0.80 ±0.40	25.20 ±0.40	0.80 ±0.40	26.00 ±0.00	0.00 ±0.00	6.60 ±7.71	19.40 ±7.71	5.40 ±6.34	20.60 ±6.34	10.20 ±10.48	15.80 ±10.48

Table 4.3: Average accuracies (%) across modalities and classes given the SVM-quad classifier from KaraOne methodology [27], and for select classifiers based on the proposed methodology.

Method	Classifier	Modality	Task C/V	± Nasal	± Bilabial	± /iy/	± /uw/
KaraOne	SVM-quad	**EEG**	**18.08**	63.50	56.64	59.60	**79.16**
		FAC	62.54	48.10	63.73	40.25	20.68
		AUD	81.05	40.48	39.98	37.63	18.33
		EEG+FAC	72.17	48.41	63.73	56.03	19.60
		EEG+AUD	61.13	62.72	39.99	49.15	83.75
		ALL	75.72	51.87	63.73	46.01	20.20
IFST	**RF + ROS**	**EEG**	**81.60**	60.98	62.86	64.95	**90.94**
	HC + NS / ROS		77.35	54.63	58.05	57.84	**89.41**
	DT + ROS		69.55	55.61	55.96	61.25	**82.65**
	SVM + ADASYN		**75.47**	44.74	54.91	56.03	40.14

4.1.1 Comparative analysis with existing literature

Comparative analysis with KaraOne methodology : The KaraOne study by [27] addresses the challenge of feature reduction by employing Pearson correlation coefficients to analyze the relationship between EEG features and classes. This approach is motivated by the high-dimensional nature of the feature space. Each EEG feature's correlation with the classes is individually assessed using Pearson correlation analysis. Subsequently, the top N features with the highest absolute correlation coefficients are selected for each task, where $N \in \{5, \ldots, 100\}$. Figure 4.1 illustrates this methodology.

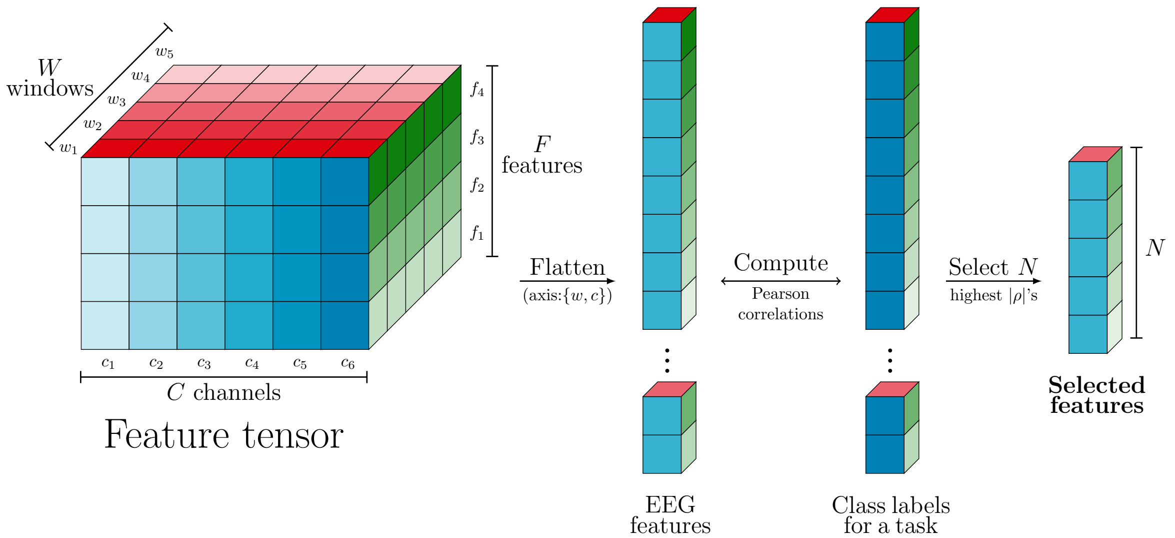


Figure 4.1: EEG features are ranked based on their Pearson correlations with the given classes for each task independently. N features are selected with the highest absolute correlation coefficients, where $N \in \{5, \ldots, 100\}$.

A limitation of the KaraOne feature selection method is that the set of features selected for each task has varying attributes and varies vastly with the choice of N. The addition of new data also affects the features that are selected. Further, most of the features are discarded and do not contribute to adding any information to the features selected. This contrasts the proposed methodology of having a fixed number of attributes that correspond to each feature and remain unchanged with adding new data or with the choice of some hyperparameter N and utilising information from all the features across all the windows and channels. Furthermore, this method requires the speaking segment of the trials to be available, while the proposed method only utilises the imagined speech segment.

4.1.2 *Correlation analysis: EEG vs. Acoustic features*

In addition, Pearson correlation coefficients are computed between the EEG and acoustic features, comparing the $17 \times 81 = 1377$ audio features with each of the 62 EEG channels across all imagined speech segments in the dataset. This analysis aimed to gauge how effectively each EEG channel predicts the corresponding audio output [27]. The top ten highest correlations are given in Table 4.4 and shown in Figure 4.2 w.r.t. the Modified Combinatorial Nomenclature.

Table 4.4: Top 10 highest mean correlations and their corresponding p-values between the acoustic and EEG features in each of the 62 channels across all the imagined speech segments in the dataset.

Channel	T7	FT7	TP7	FT8	P3	CP5	T8	P5	P7	C4
Pearson r	0.2397	0.2343	0.2297	0.2291	0.2284	0.2282	0.2281	0.2280	0.2277	0.2263
p-Value	0.0434	0.0467	0.0550	0.0521	0.0579	0.0573	0.0528	0.0568	0.0571	0.0492

4.2 Limitations

As highlighted in [17], EEG-based BCI systems for imagined speech classification face several challenges. One of the primary limitations is the restricted vocabulary size that these systems can effectively recognize. This limitation arises because the neural signals associated with imagined speech are subtle and can be easily confounded by noise or overlapping mental activities. Consequently, the classification accuracy tends to decrease as the number of target words or phrases increases. Additionally, mental repetition poses a significant challenge. In many BCI studies, participants are required to repeatedly imagine the same word or phrase to generate sufficient data for training and testing the classification algorithms. This repetitive task can lead to mental fatigue and reduced concentration, further affecting the quality of the EEG signals and, consequently, the performance of the BCI system.

This limitation is also acknowledged in the present study. Despite efforts to mitigate these issues through advanced signal processing techniques and robust machine learning

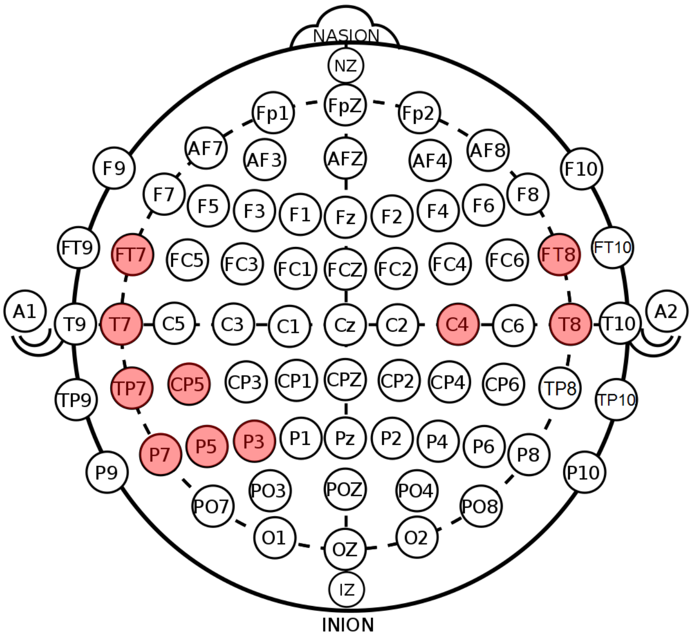


Figure 4.2: EEG electrode placement according to the 'International 10–20 system' with the 'Modified Combinatorial Nomenclature (MCN)'. The red circles indicate the top 10 EEG channels with the highest absolute correlation coefficients between the Acoustic and EEG features.

algorithms, the problem of limited vocabulary and mental repetition remains a significant barrier. The vocabulary used in this study was necessarily restricted to ensure reliable classification performance. Future research needs to explore more sophisticated methods to enhance signal clarity and develop strategies to expand the vocabulary without compromising the system's accuracy. Moreover, the variability in EEG signals across different individuals adds another layer of complexity. Each person's brain activity patterns are unique, which means that models trained on one individual's data may not generalize well to others. This individual variability necessitates the development of personalized models, which can be time-consuming and resource-intensive.

Another critical limitation is the susceptibility of EEG signals to external artifacts, such as muscle movements, eye blinks, and environmental noise. These artifacts can obscure the neural signals of interest, making it difficult for the classification algorithms to accurately interpret the imagined speech. While artifact removal techniques have improved, they are not foolproof and can sometimes result in the loss of important neural information.

Finally, the current study, like many others in the field, was conducted in controlled laboratory settings. The transition of BCI systems from the lab to real-world applications introduces additional challenges, such as varying environmental conditions and the need for more user-friendly interfaces. Ensuring the robustness and usability of EEG-based BCI systems in everyday scenarios remains an open area of research.

4.3 Applications

Potential applications include brain-computer interfaces for communication in military settings, assistive technologies for individuals with speech impairments due to various neuro-biological disorders such as Alzheimer's disease, Parkinson's disease, and 'Amyotrophic lateral sclerosis (ALS)' where there is an impairment in physical movement but not in cognitive function [4].

CHAPTER 5
CONCLUSION

5.1 Conclusion

In this work, we have explored the use of Information set theory techniques to extract rich spatio-temporal features from EEG signals for the imagined speech classification task. These features offer better differentiating power and drastically reduce the dataset size used for training without sacrificing classification performance. They address the issue of not being able to effectively utilise all the information present in the EEG signals, without sacrificing computational complexity of the feature space. This advancement paves the way for more robust and practical applications in real-world scenarios, ultimately contributing to the development of more accessible communication aids for individuals with speech impairments.

5.2 Future scope of work

The current implementation explores utilising the rich spatio-temporal features on machine learning models due to their simplicity and the fact that they have a feature extraction step. This feature extraction step is implicit in deep learning models, which are able to learn the features based on updating weights in an artificial neural network. The drawback of such methods is that they require a huge amount of data to train the model. The future work could involve exploring deep learning models for the same problem and comparing the results with the current implementation. Further, the addition of the information set-based features can be explored in a multi-class classification task.

REFERENCES

[1] Manish Aggarwal and Madasu Hanmandlu. Representing uncertainty with information sets. *IEEE Transactions on Fuzzy Systems*, 24(1):1–15, February 2016.

[2] Christoph Bandt and Bernd Pompe. Permutation entropy: A natural complexity measure for time series. *Phys. Rev. Lett.*, 88:174102, Apr 2002.

[3] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA, 1992. Association for Computing Machinery.

[4] Mariana P. Branco, Elmar G. M. Pels, Ruben H. Sars, Erik J. Aarnoutse, Nick F. Ramsey, Mariska J. Vansteensel, and Femke Nijboer. Brain-computer interfaces for communication: Preferences of individuals with locked-in syndrome. *Neurorehabilitation and Neural Repair*, 35(3):267–279, 2021. PMID: 33530868.

[5] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[6] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357, 06 2002.

[8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

[9] R. Esteller, G. Vachtsevanos, J. Echauz, and B. Litt. A comparison of waveform fractal dimension algorithms. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(2):177–183, 2001.

[10] Cindy Goh, Brahim Hamadicharef, Goeff Henderson, and Emmanuel Ifeachor. Comparison of fractal dimension algorithms for the computation of eeg biomarkers for dementia. *CIMED'05: Proc. Computational Intelligence in Medicine and Healthcare*, 06 2005.

[11] Richard Hardstone, Simon-Shlomo Poil, Giuseppina Schiavone, Rick Jansen, Vadim V. Nikulin, Huibert D. Mansvelder, and Klaus Linkenkaer-Hansen. Detrended fluctuation analysis: A scale-free view on neuronal oscillations. *Frontiers in Physiology*, 3, 2012.

[12] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008.

[13] T. Higuchi. Approach to an irregular time series on the basis of the fractal theory. *Physica D: Nonlinear Phenomena*, 31(2):277–283, 1988.

[14] T. Inouye, K. Shinosaki, H. Sakamoto, S. Toi, S. Ukai, A. Iyama, Y. Katsuda, and M. Hirano. Quantification of eeg irregularity by use of the entropy of the power spectrum. *Electroencephalography and Clinical Neurophysiology*, 79(3):204–210, 1991.

[15] Ashwin Kamble, Pradnya H. Ghare, Vinay Kumar, Ashwin Kothari, and Avinash G. Keskar. Spectral analysis of eeg signals for automatic imagined speech recognition. *IEEE Transactions on Instrumentation and Measurement*, 72:1–9, 2023.

[16] J. Satheesh Kumar and P. Bhuvaneswari. Analysis of electroencephalography (eeg) signals and its categorization–a study. *Procedia Engineering*, 38:2525–2536, 2012. INTERNATIONAL CONFERENCE ON MODELLING OPTIMIZATION AND COMPUTING.

[17] Diego Lopez-Bernal, David Balderas, Pedro Ponce, and Arturo Molina. A state-of-the-art review of eeg-based imagined speech decoding. *Frontiers in Human Neuroscience*, 16, 2022.

[18] Mamta and Madasu Hanmandlu. Robust authentication using the unconstrained infrared face images. *Expert Systems with Applications*, 41(14):6494–6511, October 2014.

[19] Jeevan Medikonda, Saurabh Bhardwaj, and Hanmandlu Madasu. An information set-based robust text-independent speaker authentication. *Soft Computing*, 24(7):5271–5287, Apr 2020.

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

[21] A. Petrosian. Kolmogorov complexity of finite sequences and recognition of different preictal eeg patterns. In *Proceedings Eighth IEEE Symposium on Computer-Based Medical Systems*, pages 212–217, 1995.

[22] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, March 1986.

[23] Joshua S. Richman and J. Randall Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6):H2039–H2049, 2000. PMID: 10843903.

[24] Jerry J. Shih, Dean J. Krusienski, and Jonathan R. Wolpaw. Brain-computer interfaces in medicine. *Mayo Clinic Proceedings*, 87(3):268–279, March 2012.

[25] Raphael Vallat. GitHub - raphaelvallat/antropy: AntroPy: entropy and complexity of (EEG) time-series in Python — github.com. https://github.com/raphaelvallat/antropy. [Accessed 29-03-2024].

[26] Scott Wellington and Jonathan Clayton. Fourteen-channel EEG with Imagined Speech (FEIS) dataset, November 2019.

[27] Shunan Zhao and Frank Rudzicz. Classifying phonological categories in imagined and articulated speech. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 992–996, 2015.

[28] Rümeysa İnce, Saliha Seda Adanır, and Fatma Sevmez. The inventor of electroencephalography (eeg): Hans berger (1873–1941). *Child's Nervous System*, 37(9):2723–2724, March 2020.

ANNEXURES

A.1 ANOVA analysis

Table A.1: ANOVA analysis of the five binary classification tasks

Index	Feature	Task-0		Task-1		Task-2		Task-3		Task-4	
		F-Statistic	p-Value	F-Statistic	p-Value	F-Statistic	p-Value	F-Statistic	p-Value	F-Statistic	p-Value
1	Mean	8.25	0.00	0.00	0.97	0.00	0.98	0.46	0.50	0.04	0.84
2	Absolute mean	3.49	0.06	1.80	0.18	1.34	0.25	1.16	0.28	0.01	0.93
3	Maximum	0.00	0.99	0.64	0.42	0.01	0.90	0.95	0.33	0.91	0.34
4	Absolute Maximum	0.14	0.71	0.82	0.37	0.52	0.47	0.80	0.37	0.06	0.80
5	Minimum	0.00	0.98	0.53	0.47	0.03	0.87	1.03	0.31	1.11	0.29
6	Absolute minimum	1.67	0.20	2.83	0.09	1.65	0.20	0.00	0.95	0.34	0.56
7	Minimum + Maximum	9.63	0.00	0.11	0.74	0.09	0.76	0.02	0.89	0.15	0.69
8	Maximum - Minimum	0.00	0.97	0.54	0.46	0.07	0.79	0.90	0.34	0.63	0.43
9	Curve length	2.88	0.09	1.67	0.20	0.02	0.90	0.86	0.35	7.06	0.01
10	Energy	2.41	0.12	0.07	0.79	1.86	0.17	0.36	0.55	0.74	0.39
11	Non-linear energy	0.00	0.95	0.02	0.88	0.01	0.93	0.00	0.97	0.73	0.39
12	Integral	8.25	0.00	0.00	0.97	0.00	0.98	0.46	0.50	0.04	0.84
13	Standard deviation	0.02	0.90	0.50	0.48	0.08	0.77	1.23	0.27	0.13	0.72
14	Variance	0.38	0.54	0.63	0.43	0.14	0.71	0.67	0.41	0.01	0.92
15	Skewness	0.61	0.44	0.26	0.61	0.47	0.50	0.44	0.51	0.23	0.63
16	Kurtosis	1.17	0.28	6.78	0.01	0.10	0.75	0.23	0.63	6.84	0.01
17	Sum	8.25	0.00	0.00	0.97	0.00	0.98	0.46	0.50	0.04	0.84
18	Spectral entropy	13.28	0.00	5.93	0.01	0.21	0.65	1.95	0.16	8.15	0.00
19	Sample entropy	10.29	0.00	0.46	0.50	0.91	0.34	1.46	0.23	4.02	0.05
20	Permutation entropy	16.58	0.00	6.43	0.01	4.57	0.03	3.46	0.06	16.92	0.00
21	SVD entropy	5.55	0.02	0.02	0.89	2.41	0.12	0.53	0.46	4.09	0.04
22	Approximate entropy	11.92	0.00	1.04	0.31	0.33	0.56	0.57	0.45	4.83	0.03
23	Petrosian fractal dimension	0.86	0.35	3.00	0.08	0.30	0.58	0.93	0.34	1.92	0.17
24	Katz fractal dimension	13.09	0.00	0.08	0.78	1.98	0.16	3.44	0.06	4.28	0.04
25	Higuchi fractal dimension	10.43	0.00	4.41	0.04	3.03	0.08	2.83	0.09	16.31	0.00
26	Root Mean Square	1.53	0.22	1.22	0.27	1.05	0.31	1.16	0.28	0.00	0.98
27	Detrended fluctuation	0.56	0.45	0.50	0.48	2.68	0.10	8.26	0.00	1.26	0.26
28	Δ Mean	9.67	0.00	10.78	0.00	1.90	0.17	0.16	0.69	3.55	0.06
29	Δ Absolute mean	1.22	0.27	2.77	0.10	0.89	0.35	0.58	0.45	0.00	0.99
30	Δ Maximum	0.76	0.38	0.44	0.51	0.16	0.69	0.95	0.33	3.35	0.07
31	Δ Absolute maximum	0.16	0.69	0.51	0.48	0.07	0.80	1.50	0.22	1.54	0.21
32	Δ Minimum	0.51	0.48	0.52	0.47	0.20	0.65	1.17	0.28	3.98	0.05
33	Δ Absolute minimum	0.08	0.78	1.12	0.29	0.03	0.85	1.57	0.21	3.11	0.08
34	Δ Minimum + Maximum	0.01	0.91	0.16	0.69	0.69	0.41	0.09	0.76	1.64	0.20
35	Δ Maximum - Minimum	0.18	0.67	0.38	0.54	0.40	0.53	1.57	0.21	3.19	0.07
36	Δ Curve length	7.52	0.01	0.06	0.81	1.52	0.22	0.53	0.47	3.39	0.07
37	Δ Energy	2.52	0.11	7.83	0.01	2.42	0.12	5.27	0.02	0.14	0.71
38	Δ Non-linear energy	3.39	0.07	0.03	0.86	1.81	0.18	0.02	0.90	0.16	0.69

Table A.1 continued from previous page

Index	Feature	Task-0		Task-1		Task-2		Task-3		Task-4	
		F-Statistic	p-Value	F-Statistic	p-Value	F-Statistic	p-Value	F-Statistic	p-Value	F-Statistic	p-Value
39	Δ Integral	10.07	0.00	10.98	0.00	1.93	0.16	0.17	0.68	3.55	0.06
40	Δ Standard deviation	0.07	0.79	0.76	0.38	0.24	0.62	1.05	0.31	2.03	0.15
41	Δ Variance	3.45	0.06	3.03	0.08	1.49	0.22	3.58	0.06	0.89	0.35
42	Δ Skewness	0.00	0.96	0.68	0.41	1.76	0.18	1.51	0.22	0.03	0.86
43	Δ Kurtosis	5.05	0.02	0.27	0.60	0.73	0.39	4.50	0.03	2.84	0.09
44	Δ Sum	9.67	0.00	10.78	0.00	1.90	0.17	0.16	0.69	3.55	0.06
45	Δ Spectral entropy	0.15	0.70	2.21	0.14	2.47	0.12	1.39	0.24	0.00	0.97
46	Δ Sample entropy	1.07	0.30	0.92	0.34	1.10	0.30	0.88	0.35	1.93	0.16
47	Δ Permutation entropy	0.12	0.73	0.06	0.81	0.02	0.89	0.46	0.50	4.17	0.04
48	Δ SVD entropy	0.16	0.69	2.53	0.11	1.32	0.25	9.16	0.00	1.58	0.21
49	Δ Approximate entropy	0.93	0.33	0.81	0.37	1.31	0.25	0.46	0.50	1.97	0.16
50	Δ Petrosian fractal dimension	1.21	0.27	0.14	0.71	0.66	0.42	0.82	0.36	2.62	0.11
51	Δ Katz fractal dimension	5.07	0.02	0.21	0.65	0.07	0.80	0.93	0.34	1.90	0.17
52	Δ Higuchi fractal dimension	3.51	0.06	2.04	0.15	0.32	0.57	0.07	0.79	4.02	0.05
53	Δ Root Mean Square	0.62	0.43	1.90	0.17	0.15	0.70	1.35	0.25	0.24	0.63
54	Δ Detrended fluctuation	2.70	0.10	0.02	0.90	0.04	0.84	0.34	0.56	7.87	0.01
55	ΔΔ mean	22.50	0.00	8.56	0.00	0.48	0.49	0.11	0.74	7.66	0.01
56	ΔΔ absmean	0.77	0.38	0.12	0.73	1.82	0.18	4.01	0.05	0.03	0.87
57	ΔΔ maximum	1.02	0.31	0.00	0.99	0.69	0.41	2.05	0.15	0.21	0.65
58	ΔΔ absmax	9.03	0.00	1.28	0.26	1.02	0.31	1.04	0.31	0.49	0.48
59	ΔΔ minimum	3.41	0.06	0.20	0.65	0.87	0.35	2.78	0.10	0.36	0.55
60	ΔΔ absmin	1.15	0.28	2.67	0.10	1.61	0.20	9.75	0.00	0.91	0.34
61	ΔΔ Minimum + Maximum	4.67	0.03	2.71	0.10	6.51	0.01	1.14	0.29	12.10	0.00
62	ΔΔ Maximum - Minimum	9.45	0.00	1.76	0.18	2.22	0.14	5.60	0.02	0.46	0.50
63	ΔΔ Curve length	3.50	0.06	2.07	0.15	1.18	0.28	1.11	0.29	0.71	0.40
64	ΔΔ Energy	1.61	0.20	0.90	0.34	0.27	0.60	2.23	0.14	0.70	0.40
65	ΔΔ Non-linear energy	2.51	0.11	1.14	0.29	0.09	0.76	0.18	0.67	0.08	0.78
66	ΔΔ Integral	22.51	0.00	8.68	0.00	0.48	0.49	0.12	0.73	7.57	0.01
67	ΔΔ Standard deviation	6.21	0.01	2.71	0.10	2.28	0.13	6.54	0.01	0.10	0.75
68	ΔΔ Variance	4.94	0.03	1.71	0.19	3.11	0.08	5.12	0.02	0.24	0.62
69	ΔΔ Skewness	1.39	0.24	1.23	0.27	0.99	0.32	0.71	0.40	0.01	0.92
70	ΔΔ Kurtosis	1.58	0.21	1.26	0.26	0.02	0.88	0.23	0.63	0.00	0.95
71	ΔΔ Sum	22.50	0.00	8.56	0.00	0.48	0.49	0.11	0.74	7.66	0.01
72	ΔΔ Spectral entropy	19.22	0.00	0.04	0.84	0.58	0.45	0.08	0.78	0.66	0.42
73	ΔΔ Sample entropy	15.10	0.00	0.22	0.64	0.09	0.76	0.66	0.42	0.04	0.85
74	ΔΔ Permutation entropy	1.15	0.28	0.07	0.79	0.06	0.81	0.04	0.84	0.84	0.36
75	ΔΔ SVD entropy	8.00	0.00	0.91	0.34	1.81	0.18	3.63	0.06	0.01	0.92
76	ΔΔ Approximate entropy	14.17	0.00	0.50	0.48	0.59	0.44	0.53	0.47	0.42	0.52
77	ΔΔ Petrosian fractal dimension	2.06	0.15	0.15	0.70	0.08	0.78	0.10	0.75	0.31	0.58
78	ΔΔ Katz fractal dimension	13.48	0.00	0.02	0.88	3.40	0.07	4.80	0.03	2.00	0.16
79	ΔΔ Higuchi fractal dimension	1.48	0.22	0.12	0.73	0.08	0.78	0.39	0.53	0.85	0.36
80	ΔΔ Root Mean Square	0.29	0.59	0.09	0.77	1.28	0.26	2.81	0.09	0.05	0.82
81	ΔΔ Detrended fluctuation	0.02	0.88	0.64	0.42	0.57	0.45	0.07	0.79	0.01	0.91

A.2 Feature functions

Table A.2: Summary of functions used for feature extraction on the windows.

Feature Function	**Computation**
Mean	$\frac{1}{n}\sum_{i=1}^{n} x_i$
Absolute mean	$\frac{1}{n}\sum_{i=1}^{n} \lvert x_i \rvert$
Maximum	$\max(x)$
Absolute maximum	$\max(\lvert x \rvert)$
Minimum	$\min(x)$
Absolute minimum	$\min(\lvert x \rvert)$
Minimum ± Maximum	$\max(x) \pm \min(x)$
Curve length	$\sum_{i=1}^{n-1} \lvert x_i - x_{i+1} \rvert$
Energy	$\sum_{i=1}^{n} x_i^2$
Nonlinear energy	$\sum_{i=2}^{n-1} x_i^2 - x_{i+1}x_{i-1}$
Integral	$\int_a^b x(t)\,dt$
Standard deviation	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mathrm{mean}(x))^2}$
Variance	$\frac{1}{n}\sum_{i=1}^{n}(x_i - \mathrm{mean}(x))^2$

Table A.2 – Continued from previous page

Feature Function	**Computation**
Skew	$\frac{n}{(n-1)(n-2)}\sum_{i=1}^{n}\left(\frac{x_i-\bar{x}}{s}\right)^3$
Kurtosis	$\frac{n(n+1)}{(n-1)(n-2)(n-3)}\sum_{i=1}^{n}\left(\frac{x_i-\bar{x}}{s}\right)^4-\frac{3(n-1)^2}{(n-2)(n-3)}$
Sum	$\sum_{i=1}^{n} x_i$
Spectral entropy [14]	$-\sum_{i=1}^{n} P_i\log(P_i)$
Sample entropy [23]	$-\log(A/B)$
Permutation entropy [2]	$-\sum_{i=1}^{N!} P_i\log(P_i)$
Singular Value Decomposition entropy [25]	$-\sum_{i=1}^{m} \sigma_i\log(\sigma_i)$
Approximate entropy [23]	$\log(C/D)$
Petrosian fractal dimension [21, 10]	$\frac{\log(N)}{\log(N)+\log(N/N_0)}$
Katz fractal dimension [9]	$\frac{\log(L)}{\log(d/L)+\log(d/L_0)}$
Higuchi fractal dimension [13]	$D=\frac{\log(L)}{\log(1/L)}$
Root Mean Square	$\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$
Detrended fluctuation [11]	$F(n)=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y(i)-\bar{y}_n)^2}$

A.3 Hanman classifier

The implementation is available at:

https://github.com/AshrithSagar/EEG-Imagined-speech-recognition/blob/master/utils/models.py

```python
import numpy as np
from joblib import Parallel, delayed
from sklearn.base import BaseEstimator, ClassifierMixin
from sklearn.preprocessing import minmax_scale
from sklearn.utils.multiclass import unique_labels
from sklearn.utils.validation import check_array, check_is_fitted, check_X_y


class HanmanClassifier(BaseEstimator, ClassifierMixin):
    def __init__(
        self, *, alpha=None, beta=None, a=None, b=None, q=None, n_jobs=1, verbose=None
    ):
        self.alpha = alpha
        self.beta = beta
        self.a = a
        self.b = b
        self.q = q
        self.n_jobs = n_jobs
        self.verbose = verbose

    def __repr__(self):
        return (
            f"HanmanClassifier("
            f"alpha={self.alpha}, beta={self.beta}, a={self.a}, b={self.b}, q={self.q}"
            f")"
        )

    @staticmethod
    def frank_t_norm(a, b, q):
        numerator = (q**a - 1) * (q**b - 1)

        denominator = q - 1
        if denominator == 0:
            return 0  # Handle division by zero

        return np.log1p(numerator / denominator) / np.log(q)

    def fit(self, X_train, y_train):
        """Fit the classifier to the training data.

        Parameters:-
        X_train : array-like of shape (n_samples, n_features)
        y_train : array-like of shape (n_samples,)

        Returns an instance of the estimator.
        """

        X_train, y_train = check_X_y(X_train, y_train)
        self.classes_ = unique_labels(y_train)
        self.X_, self.y_ = X_train, y_train
        self.n_features_in_ = X_train.shape[1]

```

```
        # Pre-compute the normalized training data for each class
        self.X_cls = [
            minmax_scale(self.X_, axis=1)[self.y_ == cls] for cls in self.classes_
        ]

        self.is_fitted_ = True
        return self

    def predict(self, X_test):
        """Predict class labels for samples in X_test.

        Parameters:-
        X_test : array-like of shape (n_samples, n_features)

        Returns:-
        ndarray of shape (n_samples,)
        """

        check_is_fitted(self, ["X_", "y_"])
        X_test = check_array(X_test)

        X_test = minmax_scale(X_test, axis=1)

        y_pred = Parallel(n_jobs=self.n_jobs)(
            delayed(self._predict_sample)(sample) for sample in X_test
        )

        return np.array(y_pred)

    def _predict_sample(self, sample):
        """Predict the class label for a single sample.

        Parameters:-
        sample : array-like of shape (n_features,)
            The input sample, assuming MinMax scaled along features.

        Returns:
        The predicted class label for the input sample.
        """

        entropies = np.zeros(len(self.classes_))
        for cls_idx, X_cls in enumerate(self.X_cls):
            error = np.abs(X_cls - sample)

            norm_error = self.frank_t_norm(error[:, None], error[None, :], self.q)
            min_norm_error = np.min(norm_error, axis=(0, 1))

            possibilistic_uncertainty = np.sum(
                min_norm_error**self.alpha
                * np.exp(-((self.a * min_norm_error + self.b) ** self.beta))
            )
            entropies[cls_idx] = possibilistic_uncertainty

        return np.argmin(entropies, axis=0)
```

A.4 Implementation details

The implementation of the proposed methodology is done using Python and the scikit-learn library [20]. The code is available on GitHub [1] and is structured in a modular way to allow for easy modification and extension.

A.4.1 Directory structure

- `utils/` contains utility classes and functions for data preprocessing, feature extraction, and classification.
- `models/` contains the implementations of the proposed models.
- `workflows/` contains the workflows for feature extraction and classification.
- `requirements.txt` contains the required Python packages.
- `README.md` contains the instructions for running the code.

A.4.2 Classifier module

The classifier module provides a framework for implementing and evaluating classifiers.

```python
from utils.classifier import EvaluateClassifier
# Load and import dataset, labels to X, y
classifier = EvaluateClassifier(X, y, save_dir='path/to/save', test_size=0.2, verbose=True)
classifier.compile(model=None, sampler=None, cv=None)  # Specify model & cross-validation strategy
classifier.evaluate(show_plots=True)  # Evaluate the model using cross-validation
classifier.save()  # Save evaluation results, metrics, plots, and model parameters
```

[1] https://github.com/AshrithSagar/EEG-Imagined-speech-recognition

PROJECT DETAILS

Student details	
Name	Ashrith Sagar Yedlapalli
Register number	200902016
Section / Roll number	NA / Roll – 09
Email address	ashrith.yedlapalli@learner.manipal.edu, ashrith.yedlapalli@gmail.com
Phone number	+91 91080 96633
Project details	
Title	Enhancing EEG-Based Imagined Speech Recognition Through Spatio-Temporal Feature Extraction Using Information Set Theory
Duration	6 months
Date of reporting	January 05, 2024
Organization details	
Organization	Manipal Institute of Technology, Manipal
Full postal address with pin code	Department of Biomedical Engineering, Manipal Institute of Technology, Manipal, Karnataka, India – 576 104
Website address	`https://www.manipal.edu/mit.html`
Guide details	
Faculty	Prof. Dr. Jeevan Medikonda
Full contact address with pin code	Department of Biomedical Engineering, Manipal Institute of Technology, Manipal, Karnataka, India – 576 104
Email address	jeevan.m@manipal.edu, jeevanmedi@gmail.com

thesis.pdf

ORIGINALITY REPORT

10%	7%	6%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	1%
2	shreyarajagopal13.files.wordpress.com Internet Source	1%
3	Zhao, Shunan, and Frank Rudzicz. "Classifying phonological categories in imagined and articulated speech", 2015 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 2015. Publication	1%
4	ebin.pub Internet Source	<1%
5	Ashwin Kamble, Pradnya Ghare, Vinay Kumar, Ashwin Kothari, Avinash Keskar. "Spectral Analysis of EEG Signals for Automatic Imagined Speech Recognition", IEEE Transactions on Instrumentation and Measurement, 2023 Publication	<1%