EMOTION RECOGNITION WITH ELECTROENCEPHALOGRAPHY USING ARTIFICIAL INTELLIGENCE

A THESIS SUBMITTED TO
THE FACULTY OF ARCHITECTURE AND ENGINEERING
OF
EPOKA UNIVERSITY

BY

ILVA XHAFERRI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JUNE, 2024

Approval sheet of the Thesis

This is to certify that we have read this thesis entitled **"Emotion Recognition with Electroencephalography using Artificial Intelligence"** and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. Arban Uka
Head of Department
Date: June, 27, 2024

Examining Committee Members:

Prof. Dr. Bekir Karlık	(Computer Engineering)	_____
Dr. Florenc Skuka	(Computer Engineering)	_____
Assoc. Prof. Dr. Arban Uka	(Computer Engineering)	_____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Surname: Ilva Xhaferri

Signature: _____

ABSTRACT

EMOTION RECOGNITION WITH ELECTROENCEPHALOGRAPHY USING ARTIFICIAL INTELLIGENCE

Xhaferri, Ilva

M.Sc., Department of Computer Engineering

Supervisor: Dr. Florenc Skuka

Emotion recognition has gained major importance in recent years, with applications in human-computer interfaces, affective computing, and numerous medical applications. To capture and analyze the emotional states, several modalities are used, where one of the most dominant is Electroencephalography (EEG). Facilitated by the advancements in EEG acquisition technologies, as well as in the Artificial intelligence field, Emotion Recognition with EEG data has attracted many researchers. This work aims to implement a subject-independent model that utilizes EEG to perform Emotion Recognition on DEAP and DREAMER datasets. It attempts to find the right combination of processing methods, feature extraction, feature selection and classifier that generalize well on unseen data without having excessive computational costs. In this thesis several Machine Learning models are implemented, along with a one-dimensional CNN model which succeeds in providing a reliable performance for the task of Emotion Recognition with EEG.

Keywords: *Emotion recognition, EEG, EEG feature extraction, Emotion classification, Inter-subject approach*

ABSTRAKT

KLASIFIKIMI I EMOCIONEVE ME ELEKTROENCEFALOGRAFI DUKE PËRDORUR INTELIGJENCËN ARTIFICIALE

Xhaferri, Ilva

Master Shkencor, Departamenti i Inxhinierisë Kompjuterike

Udhëheqësi: Dr. Florenc Skuka

Klasifikimi i emocioneve ka fituar një rëndësi të madhe vitet e fundit, me aplikime në ndërfaqet Human-Computer, procesim i emocioneve dhe në aplikime të shumta mjekësore. Për të regjistruar dhe analizuar gjendjet emocionale, përdoren disa modalitete, ku më dominantja është Elektroencefalografia (EEG). E ndihmuar nga zhvillimet në teknologjitë e regjistrimit të EEG, si dhe në fushën e inteligjencës artificiale, Klasifikim i emocioneve me anë të EEG ka tërhequr shumë studiues. Ky punim synon të krijojë një model që përdor EEG për të kryer Klasifikimin e Emocioneve me datasetet DEAP dhe DREAMER. Ai përpiqet të gjejë kombinimin e duhur të metodave të përpunimit, në nxjerrjen dhe selektimin e të dhënave më të dobishme, dhe modelit kalsifikues me qëllim që të prodhoj rezultat të mirë dhe me të dhëna qe nuk i ka hasur më parë, pa pasur llogaritje të tepërta. Në këtë temë janë zbatuar disa modele të Machine Learning, së bashku me një model 1D CNN i cili arrin të sigurojë një performancë të pëlqyeshme në lidhje me Klasifikimin e emocioneve me EEG.

Fjalët kyçe: *Njohja e emocioneve, EEG, Përzgjedhje e veçorive të EEG,Klasifikimi i emocioneve, model i pavarur nga pjesëmarrësit.*

ACKNOWLEDGEMENTS

I would like to express my profound gratitude to my supervisor, Dr. Florenc Skuka, for their guidance and support throughout the completion of this thesis. I also extend my thanks to the committee members for their significant feedback. Lastly, I am grateful to my family and friends for their unwavering support throughout these months.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Emotion recognition with EEG is an emerging multidisciplinary field with researchers from neurology, psychology, computer science, and many more. Despite the advances in both EEG recording technology and Machine Learning and Deep Learning algorithms, Emotion Recognition using EEG signals has several limitations, such as the susceptibility of EEG signal acquisition to noise and other interferences, the difficulty to capture the complex patterns given the non-linear and non-stationary nature of EEG signals, the limited generalization power of the model when tested on a cross-subject dataset, and high complexity of the algorithms utilized for the classification.

The goal of this thesis is to develop a reliable, scalable, and computationally efficient model for the task of Emotion Recognition using EEG. Specifically, the study will investigate the implementation of Machine Learning algorithms, and Deep Learning algorithms to classify time-series representation of the EEG signals for a subject-independent classification model. Moreover, this thesis will explore the robustness of different features extracted from the signals from the time and frequency domain respectively. This thesis makes use of two important EEG-ER datasets that have been utilized as benchmarks in the field of Emotion Recognition: DEAP [1] and DREAMER [2]. The results of prior studies on these datasets will be then compared to the results of our models.

The importance of this research lies in the potential increase in performance, and reduction of the computational and storage cost of the Emotion Recognition task, which facilitates real-world applications in the medical field, Human-computer-interfaces, adaptive learning environments and many more. By addressing the current limitations in feature extraction and selection, model generalization and computational complexity, this thesis aims to provide a novel approach to the filed with a scalable EEG-Emotion Recognition system.

1.2 Thesis Objective

The primary goal of this thesis is to build upon the existing methodologies and provide a new solution to the Emotion recognition with EEG task. More specifically, this thesis aims to:

1. To compare the performance of various Machine Learning and Deep Learning algorithms in terms of accuracy and computational cost.
2. To investigate on the feature extraction and selection methods that improve the classification process.
3. To evaluate the proposed system on DEAP and DREAMER datasets and compare its performance with state-of-art algorithms.

By achieving these objectives, the ultimate goal of this thesis is the contribution to the development of the field of Emotion Recognition with EEG signals. The thesis aims to provide reliable, scalable, and computationally efficient models for the ER task utilizing time-series data.

1.3 Scope of works

The scope of this thesis is to investigate on the combination of features extraction/selection and classifier implementation to achieve an effective model for ER with EEG task. This thesis EEG signals are utilized solely as time-series data, and no methodologies of 2D or 3D representations of EEG data are discussed. The reason for this is that the thesis aims to create a computationally inexpensive system that can readily be deployed in real-time, which is mostly not possible for computationally heavy processes such those including 2D and especially 3D data.

This study utilizes only the DEAP [1] and DREAMER [2] datasets of affective EEG signals, and provide comparison of the implementations with existing methods. The only emotion rating categories used in this thesis are Valence, Arousal, and Dominance as both the datasets provide these measures for each trial. Discrete classification of emotions is out of scope for this study. Moreover, this thesis solely aims to implement subject-independent modeling and therefore, subject-dependent approaches are out of the scope of this study.

1.4 Organization of the thesis

The thesis is organized into 7 chapters: in the first chapter a brief introduction of the Emotion Recognition task is provided, followed by the problem statement, thesis objectives and Scope of Works. Chapter 2 provides theoretical background information related to the fields of Emotion and Electroencephalography. For the field of Emotion, a comprehensive guide to the structure and basis of emotion is provided, along with the introduction to Emotion models and some concepts on neurophysiological mechanisms of emotion. In the second part of this theoretical background chapter, the basic concepts of Electrophysiology are discussed in order to provide the very genesis of the EEG signals. Finally, the chapter is enclosed with practical information of EEG acquisition procedures.

In Chapter 3 an extensive Literature Review of the existing methodologies of the ER with EEG task are provided, starting from emotion targets, following with data processing, feature extraction and selection. The main classification methods are discussed and a brief overview of the datasets employed is given. The chapter is concluded with an important discussion on the problem of 'domain shifting' and the possible methods to by-pass it.

Chapter 4 explains in detail the methodologies used during this study: the data processing done, the features that were selected to be representative of the EEG signals, as well as the classifiers utilized.

Chapter 5 presents the experimental results with several tables for the multiple implementations done during the study. In Chapter 6 discussions of the results are given as well as multiple plots to visualize the performance of the models. Finally, Chapter 7 gives concluding remarks and provides with future research suggestions.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Introduction

Emotion recognition refers to the detection and interpretation of emotional states that affect humans. It includes several fields such as psychology, neuroscience, and more recently computer science. The goal of Emotion recognition (ER) is to identify and understand human emotions by analyzing different mediums in which they might be showcased (i.e. facial expressions, body language, physiological signals, voice, text patterns, etc.) [3]. The application of ER is widespread, starting from medical usages, Human-computer-interfaces (HCI) especially when facilitating activities for people with disabilities, market research, etc. Given its importance, ER has been receiving a lot of interest from researchers in recent years.

Although there are a multitude of modalities from which emotion can be observed, Electroencephalography (EEG) is considered as one of the most appropriate to achieve this task. Emotion recognition with EEG involves analyzing the physiological signals originating from the activity of the individual's central nervous system [4], to infer their emotional states. Furthermore, EEG data has a high temporal resolution, reliable since it is a relatively impartial way of interpreting emotions, as well as being easily obtainable with non-invasive, affordable EEG headsets.

The emergence of research interest in Emotion Recognition has also contributed to the advancements in the Machine learning field, with the development of deep learning and AI technology that has greatly facilitated the processing speed and computing capabilities, as well as removed the need for medical expert analysis and feature extraction. Moreover, the advancement in EEG acquisition technology has also made it possible for good-quality EEG data can be acquired efficiently [5].

2.2 Emotions

To be able to design and conduct a successful experiment with EEG, a prior introduction to several concepts must be done. This section is further divided into the following topics: the definition of emotions, emotion components, Emotion theories, emotion models, emotion-evoking methods, and finally the Neurobiological basis of emotions.

The American Psychological Association (APA) defines emotions as: "Conscious mental reactions subjectively experienced as strong feelings … typically accompanied by physiological and behavioral changes in the body." A common view that emotion researchers have is that emotions are episode-like and are elicited by various stimuli [6] [7] [8]. Given their episodic nature, emotions are generally short-lived.

2.2.1 Components of emotions

In addition, researchers commonly regard emotions as having several components that are influenced by evolutionary and social contexts [6] [7] [8]. Usually, they are grouped into three major categories: subjective experience component, physiological responses, and behavioral responses. (UWA) A more complex set of components is mentioned by [9] [10] [11]: a subjective feeling component, a motor component, a physiological component, an action tendency component, and an appraisal component.

- **Subjective feeling component** is considered to have a monitoring function. In this case, the monitoring is done with respect to the individual's immediate emotional experiences, recognizing the subjective feeling (for example: fear) and applying regulatory strategies to deal with said feeling (feeling less scared).

- **Action tendency component** prioritizes actions needed in given situations. For example, students feeling curious tend to approach and ask questions, while when feeling fear there is a tendency/urge to avoid the

situation. However, action tendencies are not actions – rather they are overt behavior – so one has the choice of acting or not.

- **Appraisal component** is labeled with the meaning-making function: the cognitive evaluation of an emotional event considering the significance and the consequences of the event, predicting the emotional reactions and achieving the regulation of emotions, as well as the communication of the emotional knowledge to others.

- **The motor component** has a communicative function since it helps express feelings, for example crying to signal sadness, or smiling to communicate happiness.

- **The physiological component** acts as a support to the other components. For instance, appraisal of an event as significant is related to an increase in the activity of the amygdala.

Given that emotions are considered to have components, each of them can be assessed using different paradigms. For example, physiological changes can be measured by investigating single-neuron activity, up to the large-scale autonomic and motor system responses to emotional episodes. Motor components can be evaluated by measuring the muscle activity (from the smallest units to the expression in the entire body). Although these measuring methodologies provide information for the essence of the emotion, because of the inherently subjective nature of feelings, self-report remains the most significant tool to measure emotions.

2.2.2 Emotion Theories

To explain how the different emotion components interact with one another several emotion frameworks/theories have been developed/suggested:

Charles Darwin Evolutionary Theory of Emotion is one of the first theories of emotion that arises from the comparative studies performed by Charles Darwin [12] [13]. Darwin proposed that physiological behavior and emotions are controlled by the intellect and cortical mechanisms. He aligned the idea of instinctive behavior and natural selection, by suggesting that these behaviors are inherited and are critical to the

survival of the species. For example, the emotion of fear keeps us safe from predators [13].

The James-Lange Theory [14] suggests that emotions are a result of physiological arousal. Noticing the physiological responses, one experiences during an external event leads to the corresponding emotion. For instance, while encountering a threat the fight or flight response makes our heart rate and respiration rate increase. Becoming aware of such a change in our physiological state gives rise to the feeling of fear.

The Cannon-Bard Theory [15] contradicts James-Lange's claim that physiological arousal comes before emotional experience. Rather they claim that this process is simultaneous and independent. Given the same example of imminent threat, according to the Cannon-Bard theory the flight or fight response of the organism, and the emotion of fear occur at the same time, though the reactions are not connected. In the example of cortical processes (connected with higher cognitive functions and therefore the emotional experience part of the process) was inhibited by anesthesia, but the emotional display is still present: during the early stage of ether anesthesia sobbing (grief), laughter (happiness), or energetic aggressive reactions (rage) may be present. This serves to show that even when the cognitive appraisal of the physiological state does not happen as suggested by James-Lange, emotional experiences still occur, further proving the independent nature of these two processes.

2.2.3 Emotion Models

In current research, emotions are usually defined according to two types of models: discrete and dimensional emotion models [16]. In discrete emotion models, the emotions are described by a specific subset of emotional states. For instance, [9] categorizes emotions in eight basic states (anger, fear, sadness, disgust, surprise, happiness, trust, and expectation). Similarly, [17] claims there are six basic emotions (anger, disgust, fear, happiness, sadness, and surprise). However, these discrete models fail to capture the complexity of Human emotion and have been found to have many limitations [18].

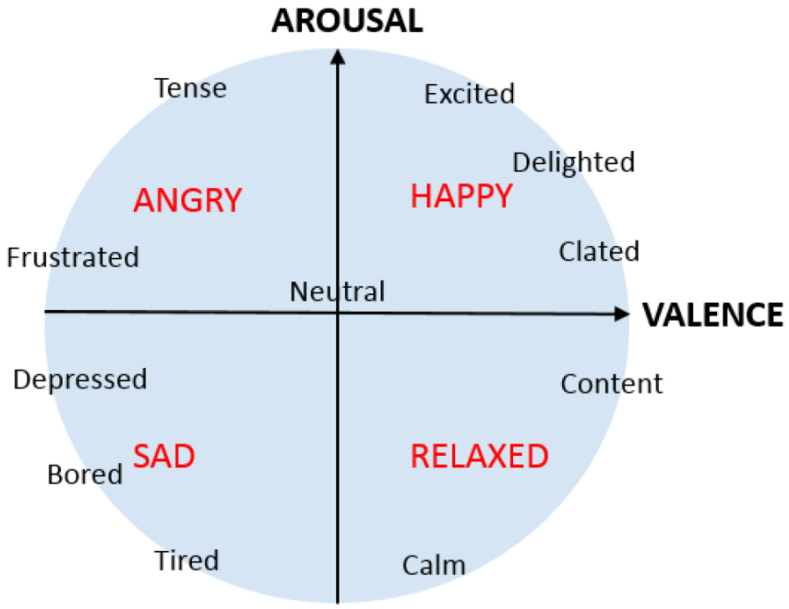

Figure 1:*Distribution of different emotions on arousal-valence 2D*

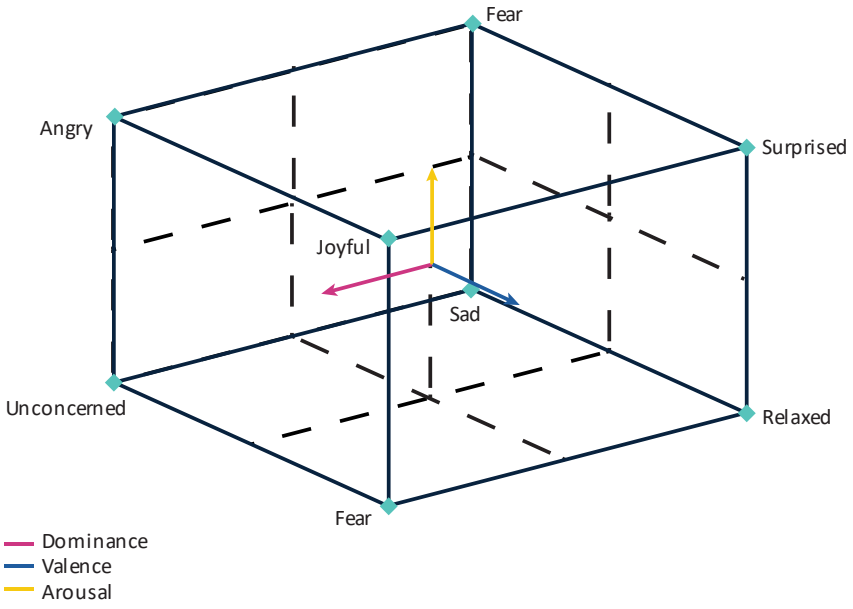

Figure 2: *Distribution of different emotions on arousal-valence-dominance 3D space.*

Dimensional models, on the other hand, are more effective for quantifying and characterizing the type and intensity of emotions. The dimensions used to describe emotions in these models have been termed valence, arousal, and dominance.

Valence refers to the emotion type ranging from unpleasant to pleasant. **Arousal** measures the intensity of the emotion, where boredom is categorized as having low arousal, while excitement has a high arousal level. **Dominance** refers to the emotion of being controlling or submissive in nature [19]. *Figure 1* and *Figure 2* showcase how these dimensions can be used to describe different emotions. Because the dimensional emotion model provides a way to quantify emotions, it is a better fit with the emotion recognition task at hand.

Choosing the right method to evoke the target emotions is one of the key elements in emotion recognition studies. Selecting the stimulation materials to be as suitable and effective has direct consequences on the quality of the collected data. The evoking methods can be divided into internal and external stimuli [20] [21], where internal stimuli refer to the recollection of personal experiences or self-imagination under a set of experimental instructions [22]. In contrast, external stimuli utilize pictures, audio, and videos for emotion evoking.

Considering the individual differences between the subjects in age, culture, gender, personal experience, and emotional perception, there are many limitations of internal stimuli as an emotion-evoking method [16]. On the other hand, evoking emotion using external stimuli, while still recognizing individual differences, seems as a more efficient approach in a laboratory environment. [23] claim that video-based stimuli provide much richer stimulation than pictures or audio separately and, therefore are very effective as emotion-inducing stimuli [24]. Recently, VR has also been used as a stimulus and has shown great results as it provides the capability to be immersed in a virtual environment. A suggestion from the review paper [25] is to keep the stimulus length in VR studies between 15-20 seconds to avoid the fluctuations of emotions.

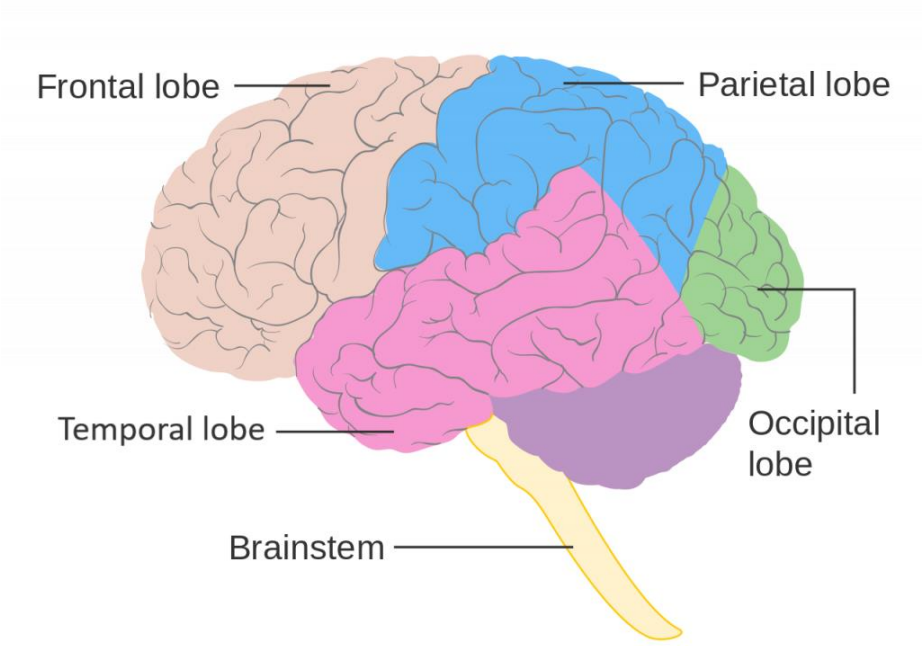

Figure 3: *Brain regions*

2.2.4 Emotion-related neurophysiological mechanism

The brain is comprised of three main components: the cerebrum, the cerebellum, and the brainstem. The cerebrum is the largest of the three and can be divided into two cerebral hemispheres [26]. The intermediate region between the cerebrum and the brainstem is termed the diencephalon. The cerebrum is composed of nerve cells with their cell bodies being in the outer part (cerebral cortex) a consistency that is termed ‘gray matter’ and the inner part is made up mainly of axons termed as ‘white matter’ [26]. The cerebral hemispheres are furthermore partitioned into five lobes: frontal, temporal, parietal, occipital, and limbic [27] depicted in *Figure 3*. There is also the insular lobe (‘insula’ meaning island) situated within the lateral fissure (sulcus) that separates the temporal lobe from the frontal and parietal lobes. Some structures of the limbic system are situated in the diencephalon, that is the thalamus, subthalamus, and hypothalamus [26]. Others like the hippocampus and amygdala (paired structure with parts on both hemispheres) are located in the medial temporal lobe [28].

The cerebral cortex is the considered as the basis for our conscious activities. Out of the lobes that constitute the cortex, the frontal, temporal and insular lobes are crucial in emotional activity. Not only are they involved in behavior such as planning, or motivation, but also, they are situated close to the limbic system (the part of the brain related to behavioral and emotional responses). The temporal lobe sits right above the hippocampus and amygdala, while the interior part of the frontal lobe - the anterior cingulate cortex takes part in emotional processing. In addition, the insula has the function of analyzing sensory impulses and associates them with emotional experience [29].

While the function of each region of the brain is sometimes claimed as being fixed, there is a division of thought regarding the neurobasis of emotions: the locationist approach, and the psychological constructionist approach. The first one hypothesizes that emotions have discrete categories, which are linked to specific parts of the brain, while the second believes that emotions are more complex and rely on the interaction of several brain regions [30].

Emotions from a locationist perspective have several basic categories such as happiness, sadness, anger, fear, disgust, etc., which cannot be further decomposed into more basic psychological components, and in turn rely on distinct brain mechanisms to exhibit themselves [31] or as networks [32] [33]. On the other hand, psychological constructionist perspective explains that discrete emotions are not established categories, but rather psychological events that emerge from the interaction of brain regions [34] [35] [36] [37]. Moreover, in some psychological constructionist views, these emotions are product of the ‘core affect’ - a state of feeling emotionally charged (feeling good/bad/energized/exhausted) that influences the body in a kinesthetic, somatovisceral, and neurochemical way [8] [30]. More generally, ‘core affect’ can be summarized as the neurophysiological state of consciously feeling emotion [8].

There are several structures mentioned by locationist hypothesists, that are directly linked to specific emotions. The amygdala for example is termed as the source of emotion ‘fear’. This hypothesis was founded on the studies on fear-learning in rats where the amygdala was observed to be stimulated [38] [39] [40] [41] [42] [43] [44] [45]. Moreover, it was viewed that individuals with some damage to the amygdala had difficulty in perceiving instances of fear [46] [47]. However, the psychological

constructionist perspective counters this, claiming that the amygdala is most likely involved when salient stimuli occur. Salient stimuli capture the attention more readily than other stimuli, given their importance and distinctiveness. The amygdala is most likely activated in order to process the uncertain stimulus, which extensively includes fear-inducing stimuli giving rise to the misconceptions [48].

Another structure that locationists hypothesize as the basis of the emotion 'disgust' is the anterior insula (insular lobe) [49] [50]. They claim that disgust is an emotion that has evolved from the reflex of rejecting food [51] or as an aversion to potential disease [52]. Moreover, they state that individuals suffering from damage to this structure have difficulty in perceiving disgust exhibited in facial or vocal expressions [53] [54].Similarly, the psychological constructionist perspective negates this exclusive relationship between the emotion of 'disgust' and the anterior insula. They claim that this structure is a key region in processing awareness of physical sensations, which most likely includes the feeling of 'disgust'. Other sensations exhibited during the stimulation of the insula include twitching, tingling and warmth in different parts of the body, feelings of movement, etc. [55] [56]. Below in *Table 1* there is a compilation of the brain structures that locationist hypothesis links with specific emotions, followed by the psychological constructionist explanation.

Table 1: *Neurobasis of emotion: two views* ***[30]***

Brain structure	**Locationist hypothesis on emotion processed**	**Psychological constructionist hypothesis**
Amygdala	fear	Detects salient stimuli
Anterior Insula	disgust	Involved in awareness of physical sensations
Orbitofrontal cortex	anger [57] [58]	Integrates sensory information form the environment and the body [59] [60]
Anterior cingulate cortex	sadness [57] [61]	Stimulated when engaged in cognitive load [30]

2.3 Electrophysiology

Electrophysiology in neuroscience is the discipline that deals with bio-electrical activity in living tissues. It investigates the electrical properties of cell membranes and their functional significance by analyzing the electrical signals produced during physiological processes [62] [63]. Electrophysiology techniques remain the main choices for analyzing neural activity and the physiological properties that give rise to this activity. Considered by neuroscientists to be the backbone of neuroscience research as it provides a precise method to investigate the activity of neurons that produce cognition and behavior [63].

Electroencephalography is a type of electrophysiological experiment that studies the electrical activity of the brain. Therefore, to describe EEG process in detail it is necessary to provide context as to what electrophysiological experiments attempt to measure.

2.3.1 Anatomy and functionality of a neuron

The nervous tissue is made up of neurons and supporting tissue (neuroglia/glia) [64]. Neurons are independent cells, highly differentiated and specialized in nature. They are excitable cells since they generate and propagate electrical signals and are connected to one another via special contact sites called synapses. Despite their differences in morphology, all neurons share features: having a cell body (soma), and the cytoplasmic prolongations dendrites and the axon (dendrites converge on the soma, while the axon emerges from the soma at the axon hillock). The morphology of a neuron is shown in *Figure 4*.The somatodendritic tree (the dendrites and the soma) is considered to be the neuron's receptor site, since they receive synaptic contacts from other neurons and in response to the stimuli generate electrical signals. The axon and axon collaterals on the other hand, are the transmission site of the neuron, given the fact that they propagate the action potential over varied distances without attenuating their amplitude. The axon and its collaterals end in synaptic boutons that create synaptic contacts with target cells (terminal arborization) [65].

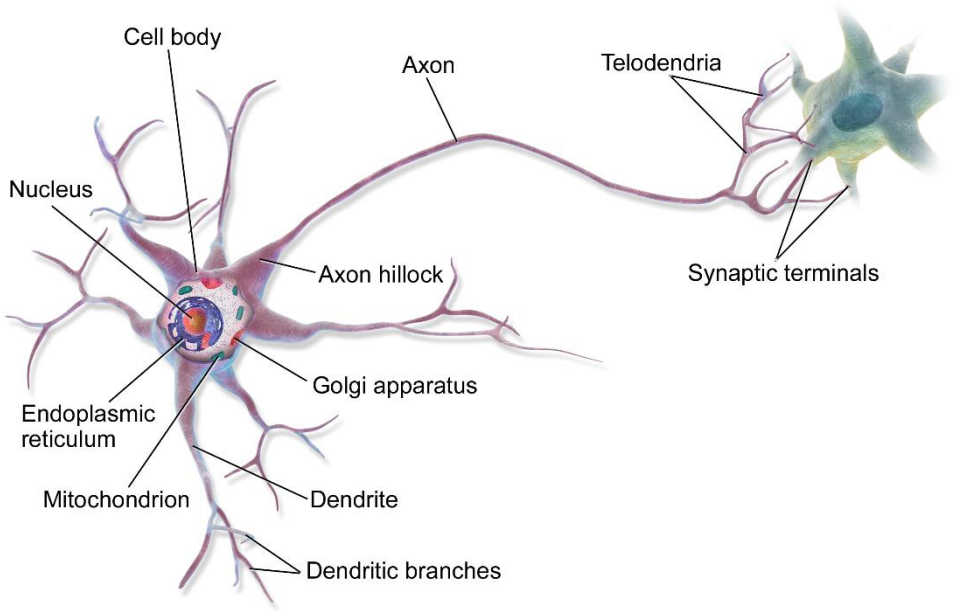

Figure 4: *Neuron structure*

The function of the neuron is the information flow throughout the neural circuitry via the transmission of the electrical impulses. To achieve this flow of signals there are two types of transmissions that occur: inter-neuronal (between adjacent neurons at the synaptic sites) and intra-neuronal (along the somatodendritic domain and axon of a single neuron). This is made possible by the bioelectrical nature of the neuron's cell membrane, which originates from the voltage difference between the cell and the extracellular fluid caused by the asymmetrical concentration and electrostatic gradients of ions, and by the presence of ion channels that affect their permeabilities. Given these variations in charge between the intracellular and extracellular parts of the membrane, an electrical potential is created. The neuron membrane is said to have two main potential states: the resting potential when the neuron cell is in a non-excited state, and the action potential when the neuron is in an excited state. Due to the differences in permeability of inorganic ions (mainly Na+, Cl-, and K+) as well as due to the Sodium-potassium-ATPase pump, the net charge across the membrane during the resting potential is approximately -70 mV [63] [65].

When the membrane potential passes the threshold of -55 mV (due to external impulses from adjacent neurons – process explained in detail in inter-neuronal transmission section below) an action potential is triggered at the axon hillock, and propagates as a wave through the axon. The local depolarization opens the voltage gated Sodium ion (Na+) channels which rush inside the cell according to the electrochemical gradient. The cytoplasm becomes less and less negative. The Na+ ions continue to flow causing the membrane to become depolarized at 0 mV. Because the voltage gated Na+ channels are not immediately deactivated, there is overshoot and the electric potential becomes 30 – 40 mV. Following this the potassium (K+) voltage-gated channels open and due to the electrochemical gradient of the potassium ions (K+) flow outside the cell and the membrane potential becomes negative again in a stage denoted as repolarization. There is a delay in the closing of the K+ voltage-gated channels, making the membrane potential even more negative (hyperpolarization). Finally, the Na+/K+ pump, that was active during the entire time, restores the membrane polarity back to the resting potential of approximately -70mV. The process is depicted in *Figure 5.*

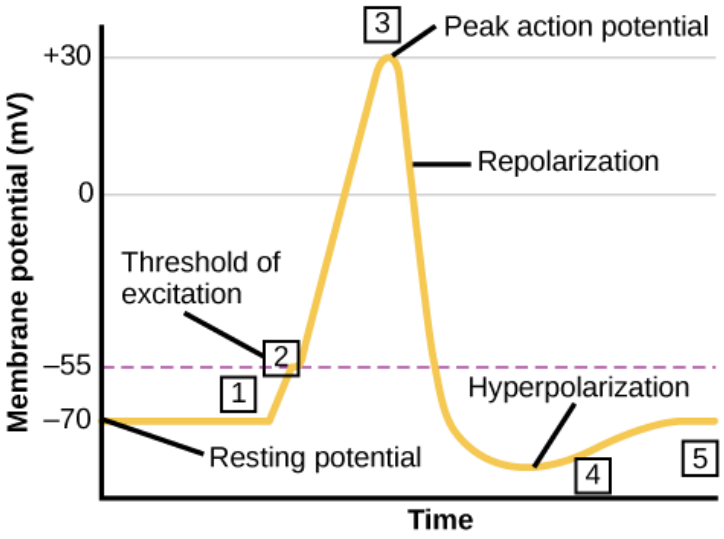

Figure 5: *Action Potential process*

Neurons communicate with one another by causing changes in the membrane potential of the adjacent neurons at the synapses. Synaptic transmission can be electrical, chemical or mixed. In electrical synapses the ions can flow directly between the cells without the mediation of the neurotransmitters. This is made possible by the clusters of intracellular channels or gap junctions between the pre-synaptic and post-synaptic neurons [65] [66] [64]. These synapses have minimal delay and are typically found in pyramidal neurons [64]. Chemical synapses have a space between the plasma membranes called the synaptic cleft, approximately 30-50 nm wide. During an action potential the presynaptic membrane is depolarized and Ca2+ ions exit the cell via the Ca2+ voltage-sensitive channels. The increased intracellular concentration of Ca2+ ions cause the synaptic vesicles to undergo exocytosis and release neurotransmitters in the synaptic cleft. Depending on the neurotransmitter molecules that bind to receptors on the post-synaptic neuron and can bring about an excitatory postsynaptic potential (EPSP) or inhibitory postsynaptic potential (IPSP) [65]. EPSP bring about the depolarization of the membrane and if the electric potential passes the threshold an action potential occurs. IPSP on the other hand promote the hyperpolarization of the membrane, that is make it more negative than the resting state (-70 mV) and therefore lowering the possibility of an action potential from occurring [62] [67]. The combination of EPSP and IPSP events is the overall signal detected in the postsynaptic neuron. When localized potentials add up and cause the depolarization of the membrane to surpass the threshold (usually -55 mV) an action potential will occur [67].

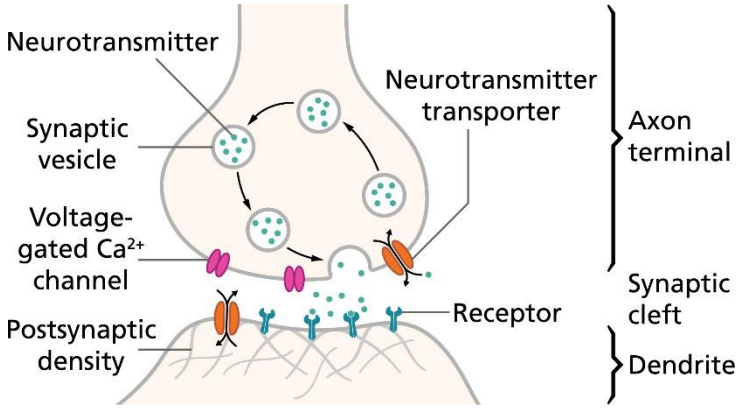

Figure 6: *The synaptic cleft during an EPSP*

2.4 Electrophysiology

After the brief introduction to the fundamentals of neuron anatomy and functioning, we can proceed with our discussion of Electroencephalography as an electrophysiological experiment. This section examines the neural basis of EEG signals, different frequencies of EEG signals and how they relate to emotional states.

EEG represents the potential difference between two cerebral locations plotted over time: the voltage of the "active" electrode where neural activity is happening is compared with the voltage in a "neutral" electrode chosen as a reference [64]. For surface EEG the measurable activity is not generated by action potentials, as they are very short in duration (1 ms) and the amplitude and electric field produced by them rapidly decreases. Therefore, EEG signals rely on detecting the postsynaptic potentials of groups of cortical neurons, as they have greater duration (10 – 40 ms) and generate stronger electric fields that can be detected from scalp electrodes [64]. Cortical neurons and their unique morphology are crucial in the generation of EEG. They are situated in deep cortical areas and have long apical dendrites perpendicular to the cortical surface, which makes them great electric dipoles. Let's consider the following cases:

1. A superficial EPSP occurs, opening the Na+ voltage-gated channels and promoting depolarization of the cell. The extracellular space becomes negative. In the deep cortical areas, the electric potential of the extracellular fluid is positive (no action potential has triggered channels to open and depolarize). By the EEG this is recorded as negative scalp potential and depicted as an upward deflection. If a deep EPSP happens, the opposite situation will unfold: positive scalp potential and negative in the distant cortical areas. This is recorded as downward deflection in EEG [68].

2. When IPSPs happen, the situation is reversed. With a deep inhibitory postsynaptic potential, the scalp polarity becomes negative, while the polarity distant to the synapse is positive. The opposite for superficial inhibitory postsynaptic potentials [68].

Thus, the EEG is a summation of all the EPSP and IPSP of over 100 cortical pyramidal neurons, measured in an area of at least 6cm2. The cortical activity has to be synchronized for the EEG signals to be detected. The more synchronized the activity of the cells is the larger the amplitude of the signal. Moreover, if this synchronous performance of the neurons is repeated numerous times, it results in rhythmic EEG waves [64].

2.4.1 Neural Oscillations – Brain Rhythm

As mentioned above when discussing the genesis of EEG signals, the summarized and synchronized activity of a large number of neurons produce electrical signals [64]. Given the pulsating nature of EPSP/IPSP, these firing mechanisms produce neural oscillations. There are three main categories of brain rhythms/oscillations [69]:

1. *Spontaneous rhythms* that occur during periods of absent sensory inputs, and have usually low frequencies (<15Hz). For example, the delta rhythm, theta, and the spindle rhythm that usually range between 4-12 Hz.

2. *Induced rhythms* are termed as ‘wake state rhythms’ as they are mostly observed to occur during conscious activities such as perceiving external stimuli (olfactory, visual, auditory, to name a few). The alpha, beta, and gamma rhythm can be included in this section, as there are studies that show the presence of fast oscillations during complex behaviors done in a conscious state.

3. *Pathological rhythms* exhibit themselves during specific neurological cases that can include epileptic seizures, and tremors.

Given that frequencies of these neural oscillations can be linked to specific functions of the brain, the same can be said for emotional states. Many studies suggest that different brain rhythms are present during specific emotions. For instance, research suggests that the Alpha activity observed regions such as the frontal lobe (right side) can be an indicator of negative emotions, whereas their activity in the left frontal lobe may indicate positive ones. It is suggested that gamma oscillations that appear to have an asymmetric nature in the temporal and parietal lobes can identify

emotions [70]. The *Table 2* gives a summarized view of the different brain rhythms and the states they are associated with.

Table 2: *Frequency ranges of the EEG signal extracted from* ***[70]***

Brainwave	**Frequency Range**	**Mental Condition**
Delta	0-4 Hz	Unconscious/deep sleep state
Theta	4-8 Hz	Relaxed state/meditative/dreaming
Low Alpha	8-10 Hz	Calmness/State of peacefulness while being awake
High Alpha	10-12 Hz	Focused/actively learning
Low Beta	12-18 Hz	Solving-problems/engaged thinking
High Beta	18-30 Hz	Alertness/agitated
Low Gamma	30-50 Hz	Self-control/compassion
High Gamma	50-70 Hz	Cognitive tasks

2.4.2 EEG acquisition

An EEG signal acquisition system is composed of EEG electrodes, analog-to-digital converters, preprocessing circuits, and EEG signal control and feedback systems [71]. This composition is shown in *Figure 7*Figure 9. Each of the elements is briefly introduced in the following subsections.

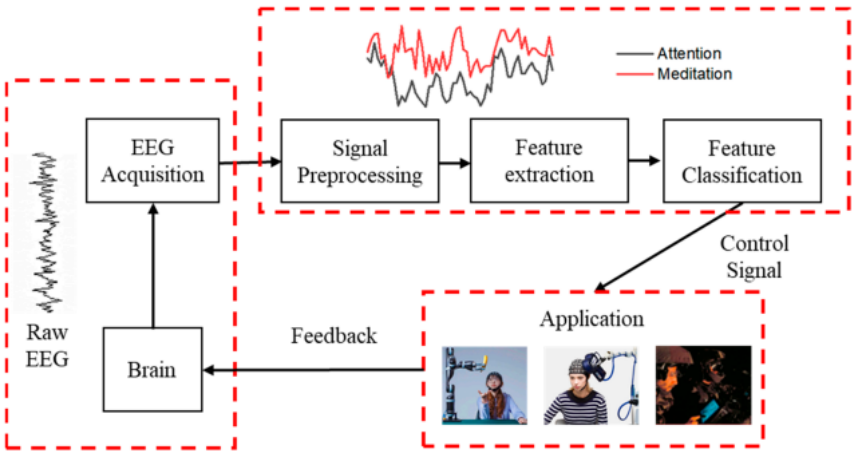

Figure 7: *EEG real-time acquisition mechanism* ***[71]***

EEG electrodes are the sensors which detect and measure the electrical brain activity. They can be categorized into groups according to some of their characteristics:

Ag/AgCl electrodes are the most commonly used, due to their excellent performance. These gel-based electrodes have high reliability, good SNR, stable signal quality, and cost-effectiveness [71].

The most common electrode placement system is the 10/20 system, a standardization proposed in 1958 by the "International Federation in Electroencephalography and Clinical Neurophysiology" [72] [73]. The 10-20 label refers to the proportional distance (in %) between the ears and nose where the positions for the electrodes are chosen.

The electrode labeling convention is done according to the brain region/lobe: F for frontal, O for occipital, P for posterior, C for Central and T for Temporal succeeded by a number. An even number indicates the electrode is in the right hemisphere, while an odd number indicates the left hemisphere. If it is followed by "Z" it stands for zero and denotes the electrodes' positions in the midline region (*Figure 8)*.

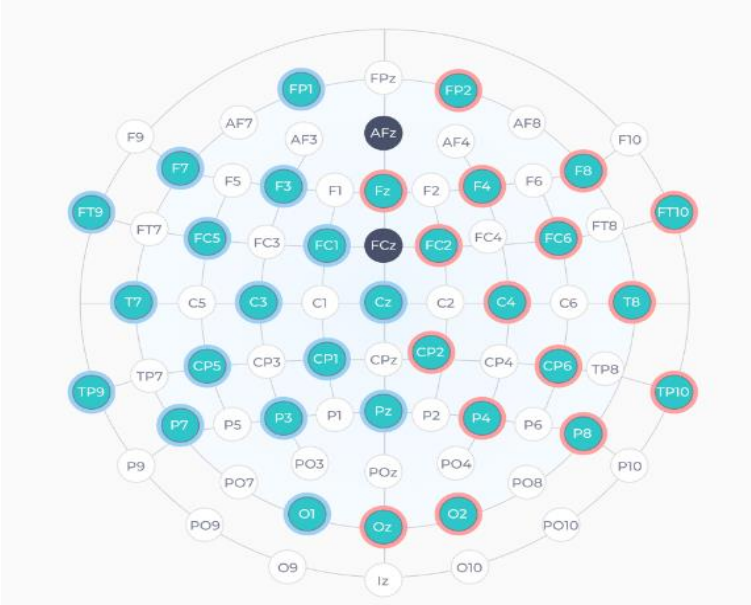

Figure 8: *Electrode placement of 32-Channel EPOC Flex Gel Sensor Kit*

Another factor that affects the process of data acquisition is the electrode impedance. Measured in Ohm(Ω), impedance is the measure of opposition to electrical flow, and it is crucial to have it as low as possible to ensure high-quality EEG data. Impedance may be increased by dead skin, scalp sweating, and oily skin secretions among some factors that obstruct brain activity from being recorded. A solution to this is to clean electrode sites with alcohol and apply an appropriate amount of conductive material [74].

Table 3: *EEG electrodes*

Category	**Description**
Material	Metal disc electrodes are typically made out of tin, gold, stainless steel, Ag/AgCl electrodes made out of silver, and a silver chloride layer [74].
Invasiveness	Non-invasive electrodes are placed on the scalp, while invasive electrodes are surgically implanted into the cerebral cortex [71].
Wet or Dry	Wet electrodes require a conductive material (gel, saline, paste). Dry electrodes can transmit the electrical signals independently [71] [73].
Cap-based or Free-from	Some headsets specify where the electrode should be placed in the cap, while Freeform electrodes can be placed freely on the scalp [73].
Active or Passive	Active electrodes have built-in amplifiers to maintain signal quality, while passive electrodes are connected via normal cables [75].

2.4.3 Analog-to-Digital Converter Circuit

After EEG data is recorded through the headset, the analog voltage signals need to be converted into digital form. This is done by the Analog-to-Digital Converter circuit (ADC). Several factors should be considered when designing an ADC circuit for a specific application since its performance greatly affects the quality of the recorded EEG data [76].

First, *ADC resolution* radically affects the acquisition quality and accuracy since having a higher resolution allows the circuit to convert minimal voltage oscillations [77]. Secondly, the ADC *sampling rate* determines the number of data

points acquired per second, so this parameter should be selected appropriately [78]. Moreover, the *noise level* of the ADC should be at a low level to ensure a high SNR. Here noise suppression techniques may also be implemented [79]. Also important is the *power-supply noise* that may influence the quality of the signals. Stable power supplies can minimize this effect, as well as a positioning of the reference electrodes that avoids this noise [80]. Finally, power consumption should be considered to fit the application, and to have reliable and prolonged data acquisition periods [71].

2.4.4 Pre-processing Circuit

Pre-processing circuits are implemented to improve the quality of th EEG signal, by removing noise and artifacts. They are composed of filters, op-amps and reference electrodes [81].

Filters are used to remove noise and artifacts while conserving the main frequency components. For instance, in [82] a Band-pass filter of 0.5 - 45 Hz in range was applied to remove signal frequencies unfit for Emotion recognition. Firstly, the Savitzky-Golay smoothing filter is applied to the signal, and then these features are subtracted from the original resulting in the average trend of the EEG signal [83] [84].

Reference electrodes are considered as zero or a known potential when calculating the potential difference at other electrodes, therefore they should be placed away from the primary sources of brain EEG signals. In the Emotiv EPOC Flex Gel sensor kit (32 channels) there are 2 reference electrodes CMS/DRL that can be configured in any 10-20 location or on ears.

2.4.5 Processor Circuit

In the system displayed in *Figure 7*Figure 9 the next element to discuss is the Processor Circuit. It is a key component that allows the real-time processing and analysis of the acquired EEG signals from the previous stages. *Table 4* gives an overview of the units that can be used.

Table 4: *Processing Circuit Options*

Type	Features	Suitable for:
FPGA **[85]**	High speed, flexible, reconfigurable	Utilizing hardware speed processing for digital filtering, waveform or spectral observations.
DSP **[86]**	High computation speed, strong real-time processing	Implementing signal processing algorithms (digital filtering, power spectrum estimation, frequency analysis, and T-F analysis).
CPU **[87] [88]**	Versatile, large storage capacity	Implement complex algorithms such as neural networks and machine learning.

2.4.6 EEG Artifacts

Artifacts are errors in the perception or representation of any data caused by the technique or equipment used. EEG artifacts are divided into two main groups: *physiologic/intrinsic artifacts*, and *extra physiologic/extrinsic artifacts* [71](*See*

Table 5). Some extra-physiologic factors can be seen in *Figure 9* [74] [73].

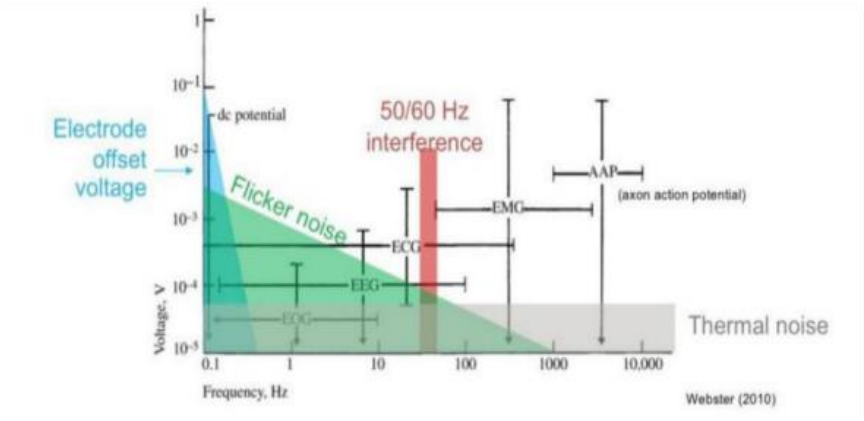

Figure 9: *Spectral densities of Interference sources*

Table 5: *EEG Artifacts*

Type	Examples
Physiological	**Muscle artifacts**: Generated from the movement of facial muscles. This activity is usually shorter than the brain activities therefore it can be identified based on duration, morphology, and frequency [74].
	Eye movements: The human eye generates electrical activity that is distinct from the brain's. It has been observed that horizontal eye movements appear in a box shape, while vertical eye movements appear to have a sinusoidal wave structure [74] [73].
	Eye blinks: also known as an electrooculogram (EOG), these signals generally have higher amplitude compared to EEG signals (same frequency) [74] [89].
Extra-physiological	**Equipment Artifacts**: Usually caused by the movement of the electrodes or headset.
	Power Line Artifacts: With a frequency of 50 Hz in Europe (60 Hz in the US), the power line noise causes artifacts especially when the electrode impedance is reduced. Since most brain activity signals have a lower frequency, these artifacts can be filtered using low-pass filters and notch filters [74] [89].
	Thermal Noise: In a similar manner as the power line noise, this type of noise caused by the random thermal motion of charge carriers in electrical conductors causes artifacts [89].
	Flicker Noise: Occurs in almost all electronic devices [89].

CHAPTER 3

LITERATURE REVIEW

3.1 Emotion models as classification labels

As mentioned in section 2. When discussing emotion models, there are two types of emotion models that can be used to classify emotions: the discrete and the continuous model. Datasets that have used the discrete model categorize emotions in several classes. For example, in SEED-IV [90] there are 4 classes (happy, sad, fear, and neutral), MAHNOB-HCI [91] has 9 discrete types, and MPED [92] uses 7 types (joy, anger, fear, funny, disgust, sad, and neutral). On the other hand, datasets that use continuous emotion models use the values of Valence, Arousal, or Dominance to classify them. Depending on the combination selected, this classifying model can be 2D (predominantly Valence-Arousal) or 3D (Valence-Arousal-Dominance). Datasets such as DEAP [1], DREAMER [2], and MANHOB-HCI use all 3 of these measures to indicate the emotional state of the subject. During the recording of the affective signals and the self-labeling by the subjects of the emotion they experience it is very common to use diagrams such as SAM (Self-Assessment-Manikin) in *Figure 10* to perform the labeling.

It is important to discuss the labeling schemes of continuous emotion models, especially when they are intended to be used in Machine Learning/Deep Learning. While discrete models have distinct classes that can be immediately used as targets of the classification, for the continuous models some processing has to be done. Depending on the rating system used in the datasets there are mainly four labeling schemes [93]:

1. One-dimensional binary labels for each of the measures (valence, arousal, and dominance). For example, in DEAP with a range of 1-9 to label each measure, High Valence (HA/HD) is everything above 4.5, and below this threshold the Low Valence (LA/LD).

2. One-dimensional three-class labels. Taking again the case of DEAP, the range 1-3.5 can be LV/LA/LD, 3.5-6.5 the neutral class for Valence/Arousal/Dominance, and the upper range 6.5-9 the HV/HA/HD classes.

3. Two-dimensional scheme for Arousal-Valence with 4 classes one per each quadrant. For DEAP this is Low Arousal-Low Valence, HA-LV, LA-HV, and HA-HV.

4. Three-dimensional label scheme for Arousal-Valence-Dominance with 8 classes. Similarly to the 2D scheme, for each binary combination of the three measures there is a class: LA-LV-LD, LA-LV-HD, LA-HV-LD, LA-HV-HD, HA-LV-LD, HA-LV-HD, HA-HV-LD, HA-HV-HD.

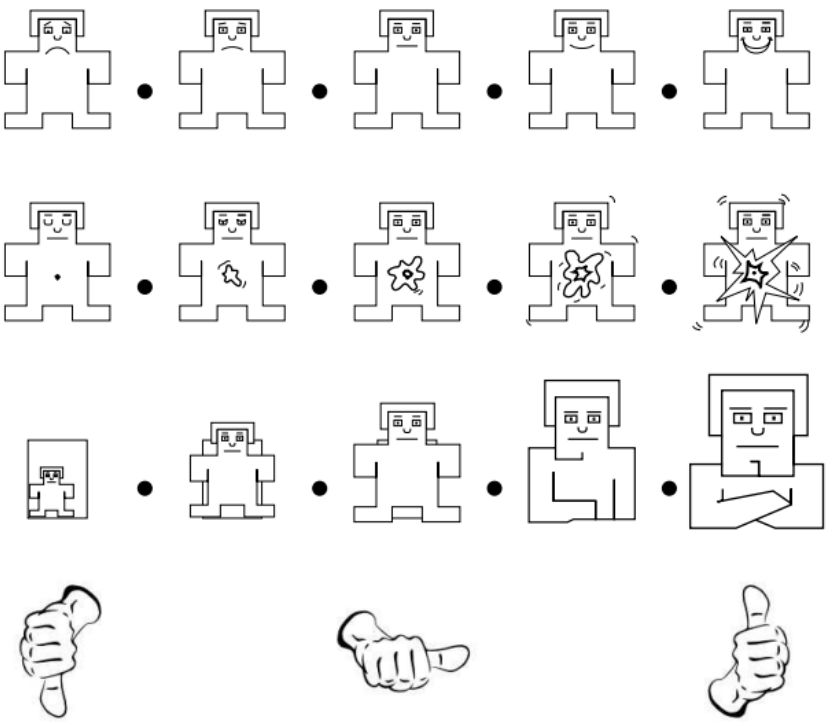

Figure 10:*Images used for self-assessment, from top: Valence SAM, Arousal SAM, Dominance SAM, Liking SAM*

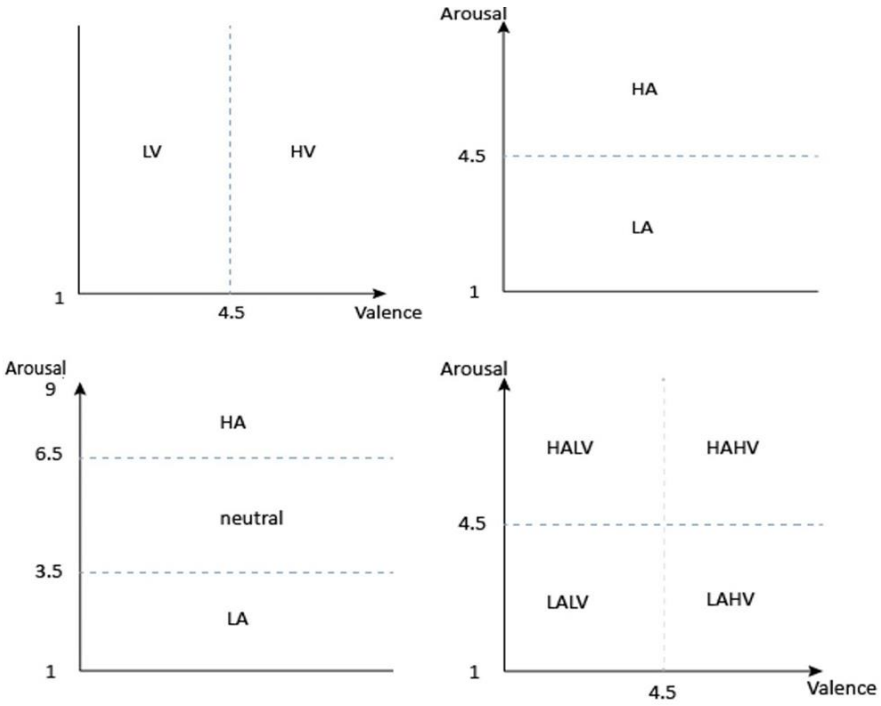

Figure 11:*Emotion labels*

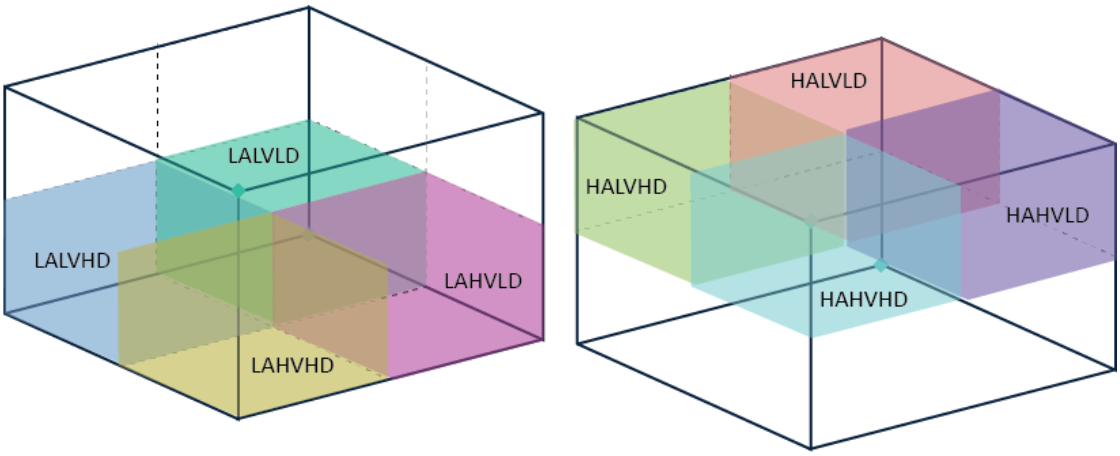

Figure 12:*Three-dimensional binary Arousal-Valence-Dominance (split into two for better visualization)*

3.2 EEG data processing

As mentioned before, EEG signals are difficult to collect and interpret. Moreover, they are usually mixed with a multitude of noisy signals coming from the body itself or the environment. Predominantly, EEG recording is done with a high sampling rate: 256 Hz, 512 Hz, or even 1024 Hz. For emotion recognition where the brain oscillations related to them are in the ranges of 8-45 Hz cite [70] [94] even the sampling rate of 128 Hz is enough, if taking into account the Nyquist-Shannon theorem where the highest frequency that can be reliably captured is 64 Hz.

To remove the noise, EEG signals can be manually cleaned from artifacts, or different filtering methods such band-pass filters (Butterworth/Chebyshev filters). Moreover, methods such as ICA, discrete wavelet can be used to separate the different signal sources, and remove the noisy ones [94].

3.3 Feature Extraction

Feature extraction aims to collect information that can be used effectively to distinguish a person's emotional state. Therefore, the accuracy of the classification algorithm is determined by the quality of the EEG features that are extracted [95]. The process of feature extraction includes transforming the signal by choosing relevant parts of it and discarding irrelevant elements [96] [95]. Moreover, it reduces the required resources for data analysis since the complexity of the data is also reduced [97]. Several forms of inputs are used in EEG classification tasks, which use different extraction methods.

1. **Raw EEG data:** Because feature engineering for EEG processing is considered a complex and time-consuming step, it has become increasingly popular to use the EEG signal as time series that are slightly processed (filtered, cleaned form artifacts, down sampled). This category of inputs is often named "Raw EEG" [98]. For the DEAP dataset, the Raw EEG signals have been preprocessed by down sampling to 128 Hz, and applying a

bandpass filter of 4-45 Hz range (common range to avoid artifacts such as ocular artifacts) [68].

2. **Time domain features:** Temporal features of EEG refer to the characteristics of the signal observed in the time domain. They provide insight into how the electrical activity measured by the headset varies over time. Analyses in the time domain features are done to extract the statistical parameters of EEG, and they primarily depict the waveform features of the EEG signal [95] [96].

3. **Frequency domain features:** The frequency domain is extracted from the time domain, and it shows the spectral (frequency) characteristics of the EEG signal. The spectrum acquired from the time domain (most commonly using Fourier Transform), is divided into several sub-bands (usually in 5 bands to represent the major brain activity frequency bands) from which features are extracted [95] [96] [97].

4. **Time-frequency domain features:** As with most biological signals, EEG has a non-stationary nature. Stationary signals have a mean and variance that does not change over time. For such signals, measuring the mean and variance over only one segment retrieves enough information to estimate the signal's true mean and variance. Therefore, these measures cannot be captured similarly for EEG physiological signals such as ECG, EMG, and EEG to name a few [98].

While the time domain analysis gives the temporal representation of the signal, it does not include the information that the frequency domain provides - frequency variations or the energy distribution of the signal. To overcome these limitations, the signals can be represented in the time-frequency domain [97].

5. **Topographic maps:** Topographic maps (topo maps) of the brain are visual-spatial illustrations of brain activity cite [99]. It illustrates the electrical activity across the scalp, depicting elements such as the amplitude or power of EEG signals across different electrode locations. Apart from the amplitude or power component of the EEG signal, other quantitative features

mentioned above (time and frequency domain features) can be used to construct topographic maps (in [99] Differential Entropy is used).

6. **Connectivity features:** Connectivity features serve as a method for quantifying connections between different brain regions based on the electrical activity recorded by the electrodes [100]. Effective connectivity depicts how the information flows between different EEG channels at a specific frequency component. The most common effective connectivity method is Granger-Causality computed in the frequency domain [101]. In two estimators of brain effective connectivity measures are used: dDTF(based on Granger-Causality), and PCD measure used extensively in neuroscience studies. The information extracted from these estimators can be represented as a matrix or graph [100].

Table 6: *List of features to extract from EEG* ***[94]***

Feature Type	**Feature**
Time-domain	Histogram analysis, Statistical measures (Mean, Variance, STD, Skewness, Kurtosis), Hjorth parameters, Fractal dimensions, Event-related potential, Entropy measures, Zero Crossing Rate (ZCR), Slope Sign Change (SSC), Willison Amplitude (WAMP)
Frequency-domain	Power spectral density (PSD), Higher-order spectrum (HOS), Logarithm energy spectrum, Approximate entropy (AnEn), Permutation entropy (PeEn)
Time-frequency domain	Discrete Wavelet Transform, Short-time Fourier transform, Hilbert Huang Transform, Wavelet packet transform, Differential entropy (DE)

3.4 Feature selection

After the extraction of the features, there are several methods that can be used to select the ones that represent the data more reliably and efficiently. There are two main methods to feature selection: manual and automatic. The first method requires some knowledge of the field to be able to choose the right features given the inherent

complexity of EEG signals. The experimentation with choosing the features manually may be too laborious for the EEG-untrained person. Therefore, the automatic feature selections are utilized in order to by-pass the complex process of selecting features personally [94].

Automatic feature selection methods can be divided into 'filter methods' and 'wrapper methods', with the first one being model-independent, while the latter requires a built model to function. Therefore, when dealing with large datasets or real-time applications the 'filter method' is often preferred. Some of the most utilized 'filter methods' are: chi-squared (χ^2) test-based approach, mutual information-based approach with mRMR(minimal-redundancy-maximal-relevance) [102] being the most widely used, ANOVA F-test approach, etc.

3.5 Classification Methods

Classifiers are algorithms used to classify data. There are numerous classification methods for EEG Emotion recognition that can be categorized into two main groups: Machine Learning (ML) algorithms and Deep Learning Algorithms. ML algorithms most commonly include Support Vector Machines (SVM), Decision Forests, Random Forest, Linear Discriminant Analysis (LDA), k-nearest Neighbors (KNN), and Naive Bayes, to name a few.

These algorithms are usually dependent on manually selected features, and are considered impractical when dealing with large datasets. The general framework of an ER with EEG using Machine Learning is showcased in *Figure 13.* Often, they are employed in subject-dependent modeling, where the data is limited to only one subject. In [21], of the classifiers reviewed, 59% SVM was used (with different kernels: RBF, linear, polynomial, and Pearson). 14% of the works used kNN, while Linear Discriminant Analysis was used 6.3%, and Quadratic Discriminant Analysis 3.2%. The final 6.35% is split equally among the Naive Bayes method and Multi-Layer Perceptron Backpropagation. Study [103] tests 5 classic ML classifiers on the DEAP dataset: Support Vector Machine, K-Nearest Neighbor, Decision Trees, Logistic

Regression, and Linear Discriminant Analysis. Firstly, the performance of the classifiers before using PCA on the data, the accuracy was 50-65%. After PCA was employed and the data dimensionality was reduced, that is the most significant channels were found (F3, C3, F4, C4, AF3, PO4, CP1), the accuracy results improved to 55-75%. On the other hand, subject-dependent implementations report satisfactory performances with accuracies reaching 73.14% for Valence and 73.06% [94] for Arousal using SVM algorithm, and 86.75% for Valence and 84.05% for Arousal using KNN [104] [105].

In the last years Deep learning algorithms have been increasingly employed as classifiers in the Emotion Recognition with EEG task. Compared to the traditional ML classifiers, Deep learning methods have reported a better performance without the need to perform extensive feature extraction that generally relies on human experts. There are different features used to train the DL models. The raw signals can be supplied, as well as the time-domain, frequency-domain, and time-frequency domain features of

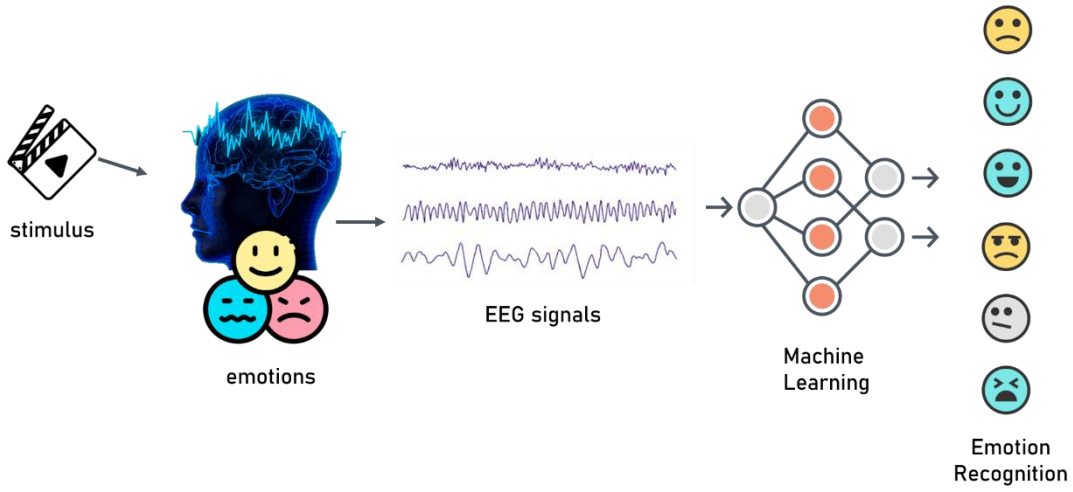

Figure 13:*The General framework of emotion recognition using Machine Learning*

the EEG signal [96]. Furthermore, spatial features such as Topographic maps, and effective connectivity measures can be provided as image inputs, especially for models that specialize in image recognition.

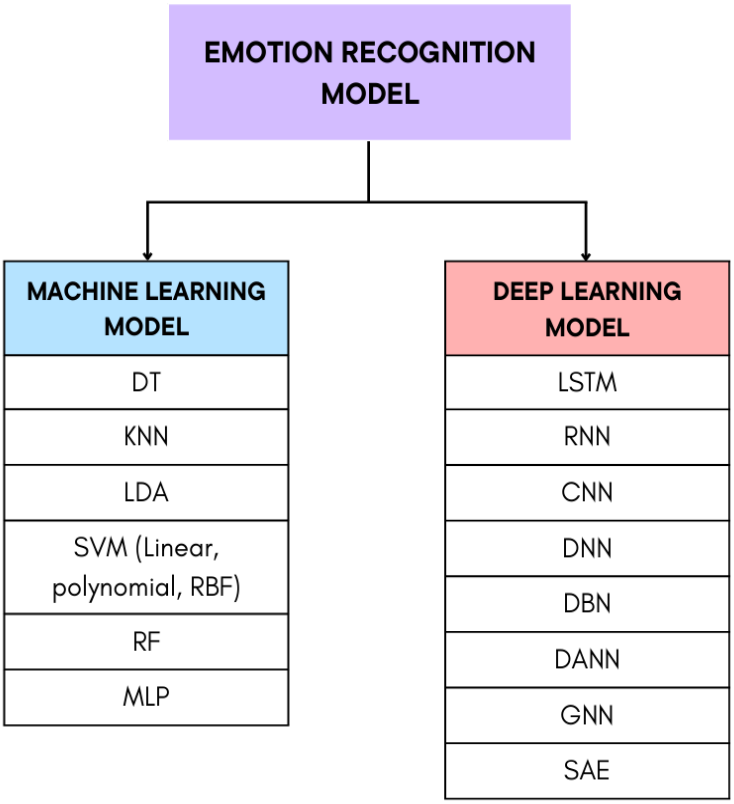

Figure 14:*Algorithms used to classify emotions*

An example of models that use RAW and processed EEG signals (STD) is the study [106] where the application of DNN (multi-layer perceptron), one-dimensional CNN (Convolutional Neural Network), LSTM (Long Short-Term Memory), and a hybrid CNN-LSTM model was investigated. The RAW data includes the data array preprocessed by the DEAP dataset where of the 32 initial channels, only 14 with the most impact was selected. On the other hand, the STD data (Spatio-temporal decomposition) is extracted from the time domain using eight statistical methods: mean, standard deviation, maximum, minimum, average absolute values of the first difference, average absolute values of second difference, skewness, and kurtosis. These statistical features are combined into an eigenvector to represent the time domain features. The labeling scheme for this study is similar to the 2D Arousal-Valence labeling scheme in the 2D Arousal-Valence model shown in *Figure 11* where the DNN is reported to have 63.99% accuracy with the RAW features, and 79% with

the STD features. The other models also achieve favorable results (67-94%), with the CNN-LSTM having the highest accuracy with the RAW feature dataset 94.17%.

Other DL models applied to the ER task are Stacked Autoencoders (SAE) [107], Deep Belief Network (DBN) [23], Recurrent Neural Network (RNN) [108], Graph Convolutional Neural Networks (GCNN) [109], Domain Adversial Neural Network (DANN) [110], 2D and 3D Convolutional Neural Networks [111]. In [100] 2D CNNs(AlexNet, VGG-19, ResNet-50, and Inception-v3) [112] trained on MAHNOB-HCI [91] dataset using dDTF images (32x32 images showing information flow between regions of the brain) were compared in all frequency bands and achieved accuracies ranging from 79.20-99.43% with ResNet-50 [112] having the highest performance over all frequencies (accuracy, precision, recall, f1-score). For the DEAP dataset [1], the accuracy ranged from 75.20-95.77% where ResNet-50 is again the model with the best performance. [113] implements a 2D-CNN with PSD features as inputs and achieves 85.5% accuracy for Valence and 87.3% for Arousal with a 10-fold cross-validation strategy. realize a 3D CNN that captures the along with the spatial information of 2D EEG frames, its temporal information too, by using 3D streams as inputs. With a 5-fold validation strategy they achieve 99% accuracy in both Valence and Arousal for subject-independent modeling.

3.6 Emotion Classification Datasets

3.6.1 DREAMER

DREAMER dataset [2] is a multi-modal database with EEG and ECG signals recorded during *affect elicitation using audio-visual stimuli*. In total, there were 23 subjects participating in the experiment, by viewing 18 video sequences labeled as ‘stimuli’ variable. Before each film clip, a neutral clip is shown which is labeled as a ‘baseline’ variable. The structure of the data for each ***i-th*** subject/participant is shown in *Table 7*.

Table 7: *DREAMER dataset* ***[2]***

Audiovisual stimuli	
Number if videos	18
Video content	Audio-video
Video-duration	65-393 s (M = 199 s)
Experiment information	
Number of participants	23
Rating scales	Arousal, Valence, Dominance
Rating values	1-5
Recorded signals	14-channel 128 Hz EEG, 256 Hz ECG

3.6.2 DEAP

DEAP Dataset [1] (See *Table 8*) is a multi-modal database with EEG signals and peripheral physiological signals of 32 participants during an audio-visual affect experiment. The signals were recorded while participants watched 40 one-minute-long sections of music videos, and provided their rating in terms of valence, arousal, dominance, like/dislike, and familiarity. The three measures that are of major interest in this study, that is valence, arousal, and dominance, where measured on a discrete scale of 1-9 with Self-Assessment-Manikins (SAM). DEAP dataset includes EEG signals measured by 32-channel 512 HZ EEG, down sampled to 128 Hz.

Table 8: *DEAP dataset* ***[1]***

Audiovisual stimuli	
Number if videos	40
Video content	Music Videos
Video-duration	63 s
Experiment information	
Number of participants	32
Rating scales	Arousal, Valence, Dominance
Rating values	1-9
Recorded signals	32-channel 128 Hz EEG

3.7 Subject-dependent/independent emotion recognition

The structure of the train/validation/test datasets is crucial when it comes to EEG signals. Given their complex and temporal nature the data splitting strategy greatly affects the performance of the model. When performing the division of data there are two approaches that researchers take: the subject-dependent and the subject-independent approach. The subject-dependent modeling achieves the process of emotion recognition by building different models for each of the subjects, given that the domain-shift between different subjects makes it substantially difficult to generalize on EEG data.

As mentioned above, the 'domain shift' hinders the ability for models to perform well with inter-subject modeling. This phenomenon is caused by the diversity of EEG signals between individuals. These contrasting characteristics may arise from several factors: culture, gender, age, genetics, etc. Studies have shown that women are more sensitive to negative emotion than men despite their age [114]. While there is plenty of research done on the socio-cultural factors that may cause variations in EEG signals, the most significant difference remains that between genders.

Therefore, the subject-independent modeling aims to build a model that generalizes well on the dataset despite the inter-subject discrepancies. The common trend for the performance of the two modeling approaches is with 70% - 100% accuracy for subject-dependent modeling, and for the subject-independent counterpart the results dropped significantly (10%-20%). Finding a way around the 'domain shift' problem is crucial to subject-independent models, and there are four main ways utilized nowadays:

1. **Calibration of signals among participants:** This can be done by using baseline recording (recordings during non-affective trials) and aligning them with one another.

2. **Aspect-oriented modelin**g: By restricting the sample space to only one gender, age or culture, for very specific modeling can produce good results. For example, [115] established an individual-specific, a gender-specific and a general model and reported increased accuracy in the individual-specific (subject-dependent) and gender-specific models.

3. **Transfer Learning-domain adaptation approach:** By using Transfer Learning EEG features are mapped into a common feature representation space, further adjusted into distinctive inter-subject features [116].

4. **Choosing subject-independent features:** Researchers are looking into finding robust EEG features that provide enough information to work around the 'domain shift'. Some of these features are the Power Spectrum Density extracted from low Alpha frequencies, Hjorth parameters extracted from the Beta rhythm, as well as the brain oscillations of subset Beta and Gamma when observed in the temporal lobe.

CHAPTER 4

METHODOLOGY

4.1 Data processing

The datasets utilized in this thesis are the DEAP (Database for Emotion Analysis using EEG, Physiological and Video Signals) [1] and DREAMER [2] (The details on their structure is given in 3.6.) The Raw EEG signals from each dataset are processed into ***numpy*** arrays and used to extract features needed for the classification task. Apart from the preliminary filtering and down sampling done to the datasets by the researchers who carried out the EEG recordings, no additional band-pass filters, or wavelet decompositions are applied to either one of them. This is done in order to have no bias when comparing the results of our model implementations with the existing literature.

Both datasets were processed by removing the baseline signals from the affect recordings: for DEAP the 3s baseline recording was duplicated into 60s, while for DREAMER the 60s baseline recording was replicated to fit the length of each of the stimuli trial lengths. The removal of the baseline is done by taking the mean value of the baseline recordings for an epoch of 1s (128 sample points) and then subtract this mean from the affect recordings sectioned into 128 sample epochs [117]. This method helps 'clean' the EEG stimuli recordings from background/non-affective signals and provides a less noisy representation of the RAW EEG recordings. However, as later seen in the results of ML algorithms tried with both the original and cleaned data, they perform better with the uncleaned signals.

4.2 Feature Extraction

The features extracted from the time-series EEG signals are part of the time domain, frequency domain, and time-frequency domain. For the Deep Learning implementations, the Euclidean distance from the center of the clusters, as well as the label predicted by the algorithm Kmeans are added as additional features. *Table 9* shows each of the features extracted from the original EEG data.

Table 9: *Extracted Features*

Feature Type	**Feature**
Time-domain	Mean, Standard deviation, Minimum, Maximum, Skewness, Kurtosis, Hjorth parameters (Activity/Complexity/Mobility) [118]
Frequency-domain	Power spectral density (PSD) for 5 frequency bands (4-8 Hz,8-12 Hz,12-16 Hz,16-45 Hz), Spectral Entropy
Other	Euclidean Distance from cluster centers, K-means predicted labels

The formulas for the statistical measures, as well as the PSD and Spectral Entropy used to extract features from the raw EEG are given below:

$$\overline{\boldsymbol{x}} = \frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{x_i} \qquad \textit{Equation 1}$$

$$\boldsymbol{\sigma} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(\boldsymbol{x_i} - \overline{\boldsymbol{x}})^{\mathbf{2}}} \qquad \textit{Equation 2}$$

$$\boldsymbol{Minimum} = \boldsymbol{min}(\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}) \qquad \textit{Equation 3}$$

$$\boldsymbol{Maximum} = \boldsymbol{max}(\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}) \qquad \textit{Equation 4}$$

$$\boldsymbol{Activity} = \frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{x_i^2} \qquad \textit{Equation 5}$$

$$\boldsymbol{Mobility} = \sqrt{\frac{\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(x_{i+1}-x_i)^2}{\boldsymbol{\mu_0}}} \qquad \textit{Equation 6}$$

$$\boldsymbol{Complexity} = \sqrt{\frac{\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(x_{i+1}-x_i)^2}{\boldsymbol{\mu_0}}} \Bigg/ \sqrt{\frac{\boldsymbol{\mu_2}}{\boldsymbol{\mu_0}}} \qquad \textit{Equation 7}$$

$$\boldsymbol{Kurtosis} = \frac{\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(x_i-\bar{x})^4}{\left(\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(x_i-\bar{x})^2\right)^2} \qquad \textit{Equation 8}$$

$$\boldsymbol{Skewness} = \frac{\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(x_i-\bar{x})^3}{\left(\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(x_i-\bar{x})^2\right)^{\frac{3}{2}}} \qquad \textit{Equation 9}$$

Spectral Density of a signal x(t) is given by:

$$\boldsymbol{S_{xx}(f)} = \lim_{\boldsymbol{T\to\infty}} \frac{\mathbf{1}}{\boldsymbol{T}}|\boldsymbol{X(f)}|^2 \qquad \textit{Equation 10}$$

Where *X(f)* is the Fourier Transform of *x(t)*.

Spectral entropy measures the complexity of the power spectrum and is defined as:

$$\boldsymbol{SE} = -\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{P(f_i)}\, \boldsymbol{log_2}\left(\boldsymbol{P(f_i)}\right) \qquad \textit{Equation 11}$$

Where $P(f_i)$ is the normalized power spectral density at frequency f_i and N is the number of frequency bins.

In addition to the type and number of the features selected to represent the EEG data, the number of channels used to record the signals is also relevant. The DEAP dataset is recorded using 32 channels, while DREAMER only 14 channels. In order to assess the influence of the number of channels on the performance of the model from the DEAP dataset the most relevant 18, 14, and 10 groups of electrodes are collected. This is done by reshaping the raw/original EEG array into (sample points) x 32 channels, and then by using the mRMR filter method for the target label scheme the 18/14/10 most representative channels will be selected.

4.3 Compilation of datasets

1. **DEAP dataset:** The time/frequency domain features are extracted from overlapping 2 second windows (256 sample points) with step size 32 sample points. The full size of the dataset is then 313160 x 480 features with features from both domains. Then this dataset is sectioned into:

    i. **DEAP time-domain:** 9 time-domain features for each 32 channels totals to 288 attributes. After the addition of the Euclidean distances and the KMeans labels the number of attributes goes to 291 for binary classification and 297 for 3D classification (8 classes).

    ii. **DEAP time-domain ML:** For the dataset used in ML implementation the number of samples had to be reduced and therefore it was reshaped into: 4680 x 4656 (binary) and 4680 x 4752 (3D).

    iii. **DEAP frequency-domain:** 6 frequency-domain features for each channel totals to 192 attributes, and after the addition of Euclidean distances and KMeans labels it goes to 195 for binary, and 201 for 3D classification.

b. **DEAP frequency-domain ML:** Similarly, the dataset is reduced in size for ML implementation to 4680 x 3120 (binary) and 4680 x 3216 (3D).

2. **DREAMER dataset:** The time/frequency domain features are extracted for each of the 14 channels over a 2s window with step size 64 sample points. It results into a dataset of dimensions 170246 x 210.

i. **DREAMER time-domain:** There are 9 time-domain features so 126 attributes, and 129 after adding the cluster distances and KMeans labels for binary classification, and 135 for 3D classification.

ii. **DREAMER time-domain ML:** The sample number is reduced by reshaping the array into 5674 x 1935 (binary), 5674 x 2025 (3D).

iii. **DREAMER frequency-domain:** There are 6 frequency-domain features so 84 attributes, and 87 after adding the cluster distances and KMeans labels for binary classification, and 93 for 3D classification.

iv. **DREAMER frequency-domain ML:** The sample number is reduced by reshaping the array into 5674 x 1305 (binary), 5674 x 1395 (3D).

Table 10: *Datasets used*

Usage	Data
DEAP-ML time-domain	4680 x4656 (4752)
DEAP-ML frequency-domain	4680 x 3120 (3216)
DEAP-CNN time-domain	313160 x 291 (297)
DEAP-CNN frequency-domain	313160 x 195 (201)
DREAMER-ML time-domain	5674 x 1935 (2025)
DREAMER-ML frequency-domain	5674 x 1305 (1395)
DREAMER-CNN time-domain	170246 x 129 (135)
DREAMER-CNN frequency-domain	170246 x 87 (93)

4.3.1 Data splitting and cross-validation

The models are trained using K-Fold cross-validation with k being 5, 10, 15, and 20. In each fold the data is split into training and validating sets, therefore trained on the training part and evaluated on the other. In the traditional ML and ensemble ML models, after performing the training with cross-validation the average accuracy is given to get a better understanding of the performance of the model.

On the other hand, the CNN datasets are not split with cross-validation. Due to the sample size of the datasets, as well as the complexity of the 1D CNN models, it is highly computationally expensive to use cross-validation. Therefore, the models undergo normal train/validation/test process. Another important aspect to mention is the fact that the data is not shuffled randomly, to ensure that some of the temporal information is learned by the model. We adapt a similar approach to [119] where every 10th sample point is allocated to the test dataset, every 4th sample point to the validation dataset, and everything else to the training dataset. This ensures that the continuous development of the emotion along those sample points is kept intact, as well as ensure that all partitions have some data from each subject to help with the domain-shift problem. When the random shuffle was used with DEAP dataset it caused over-fitting with the training accuracy drastically increasing while the validation accuracy did not improve after the first few epochs.

4.4 Classifiers

In this thesis the classifiers can be divided into the two main groups mentioned in 3.5: Machine Learning and Deep Learning models. These classifiers are used for the task of Emotion Classification for binary Arousal, Valence, and Dominance, as well as the 3D eight category-classification.

4.4.1 Machine Learning Classifiers

The Machine learning models implemented are namely: Decision Tree, K-Nearest Neighbors, Random Forest, Linear Discriminant Analysis, and Support Vector Machines. While the Deep Learning models used two versions of 1D Convolutional Neural Network (CNN).

Decision Tree is a supervised learning algorithms that is used for classification and regression. The way it works is it splits data into subsets creating a tree-like model of decisions (see *Figure 15*). The splitting is done based on the value of input features, and there are sub-trees, decision nodes terminal nodes that represent the outcome of the classification or regression process.

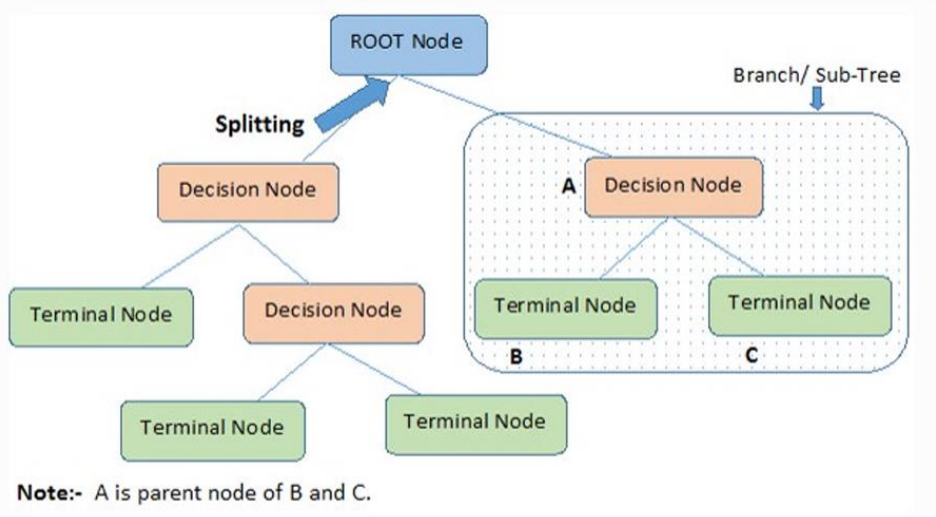

Figure 15:*Decision Tree algorithm*

Table 11: *Decision Tree Classifier Algorithm*

Algorithm 1 Decision Tree Classifier
1: **Input:** Training data $D = \{(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)\}$, feature set F
2: **Output:** Decision Tree T
3:
4: **function** BULID TREE*(D,F)*
5: **if** all examples in D belong to the same class then
6: **return** leaf node with that class
7: **end if**
8: **if** F is empty **then**
9: **return** leaf node with the majority class in D
10:**end if**
11: $f^* \leftarrow$ arg max f_F InformationGain*(f,D)*
12:$T \leftarrow$ create a decision node that splits on f^*
13:**for** each value v of f^* **do**
14: $D_v \leftarrow$ subset of D where feature f^* has value v
15: $T_v \leftarrow$ BUILD TREE $(D_v,F\backslash\{f^*\})$
16: add branch to T corresponding to $f^* = v$ and subtree T_v
17:**end for**
18:**return *T***

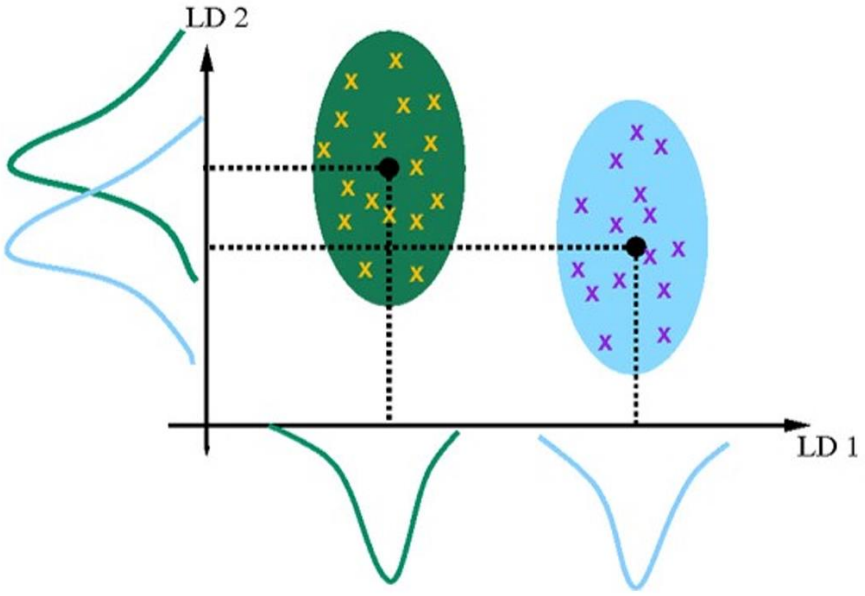

Figure 16:*Linear Discriminant Analysis Algorithm*

Linear Discriminant Analysis (LDA) is a statistical method primarily utilized for dimensionality reduction and classification. It maximizes the separation between the features of different classes. This algorithm performs the separation between classes by assuming that each class follows a Gaussian distribution and searches for a linear combination of the best features to separate these classes (*Figure 16*)

Table 12: *LDA Algorithm*

Algorithm 2 Linear Discriminant Analysis (LDA)
1: **Input:** Training data $D = \{(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)\}$
2: **Output:** Classifier $f(x)$
3:
4: **Compute** the mean vectors μ_k for each class k
5: **Compute** the within-class scatter matrix S_W
6: **Compute** the between-class scatter matrix S_B
7: **Compute** the matrix $\boldsymbol{S_W^{-1}S_B}$
8: **Find** the eigenvectors and eigenvalues of $\boldsymbol{S_W^{-1}S_B}$
9: **Select** the top k-1 eigenvectors to form a transformation matrix W
10: **Transform** the data using W
11: Classify new samples

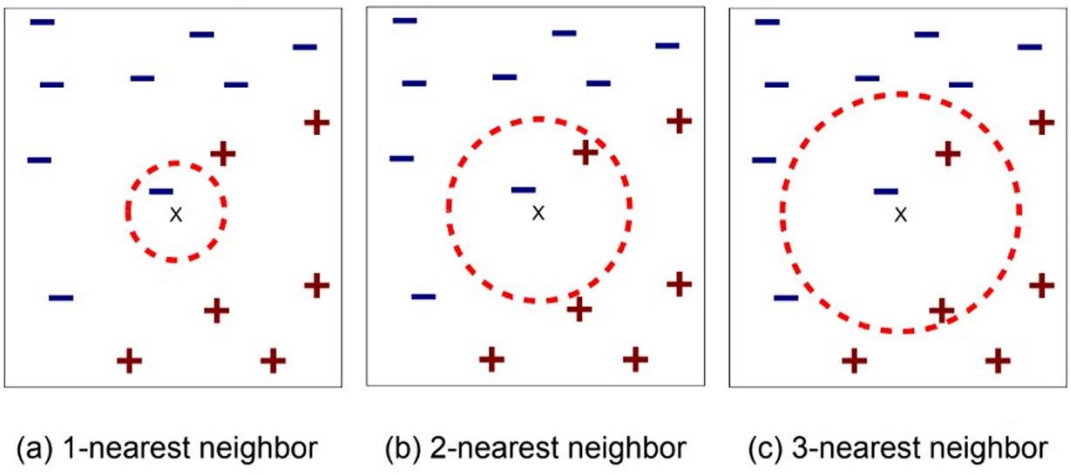

Figure 17:*K-Nearest Neighbors Algorithm*

K-Nearest Neighbor (kNN) is a supervised learning algorithm for classification and regression, which classifies data points into classes based on the majority class among its k nearest neighbors (see *Figure 17*). K is a parameter that can be set to a desired

number. The distance between datapoints is mainly measured using Euclidean distance, which makes it computationally expensive if the dataset is too large since there has to be a distance calculation for all.

Table 13: *k-NN Algorithm*

Algorithm 3 k-Nearest Neighbors (k-NN)
1: **Input:** Training data $D = \{(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)\}$, number of neighbors k, query point x_q
2: **Output:** Predicted class for x_q
3:
4: **Compute** the distance between x_q
5: **Select** the k nearest neighbors to x_q
10: **Assign** the class label to x_q based on the majority class among the k nearest neighbors

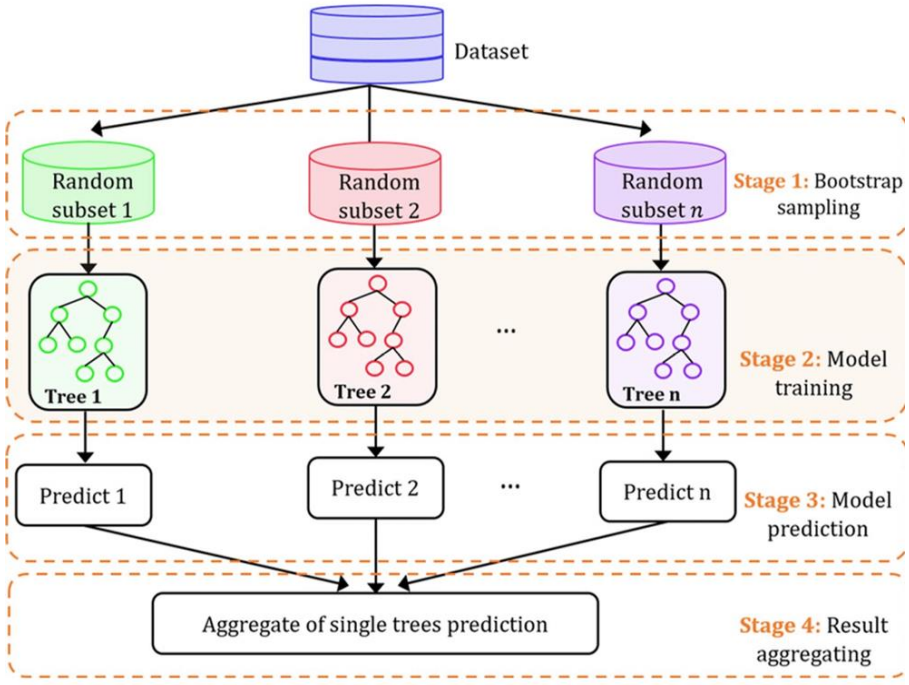

Figure 18:*Random Forest Algorithm*

Random Forest (RF) is an ensemble learning method used for classification and regression tasks. It works by constructing multiple decision trees, training them on subsets of the dataset and then getting the prediction of each of the models. The

results are gathered, the averaging/voting process occurs and then the prediction of the RF model is given. Random Forests provide good performance and known for the ability to deal with high-dimensional feature datasets(*Figure 18*).

Table 14: *Random Forest Algorithm*

Algorithm 4 Random Forest

1: **Input:** Training data $D = \{(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)\}$, number of trees T, number of features m
2: **Output:** Ensemble classifier $F(x)$
3:
4: **for** $t = 1$ to T **do**
5: Draw a bootstrap sample D_t from D
6: Train a decision tree T_t on D_t with the following modifications:
7: **for** each node in T_t **do**
8: Randomly select m features from the feature set F
9: Choose the best feature and split based on the selected m features
10: **end for**
11: **end for**
12: **Aggregate** the predictions of the T trees:
13: **For classification**: $F(x) = mode(T_1(x), T_2(x), \ldots, T_T(x))$
14: **For regression:** $\boldsymbol{F(x) = \frac{1}{T}\sum_{t=1}^{T} T_t(x)}$

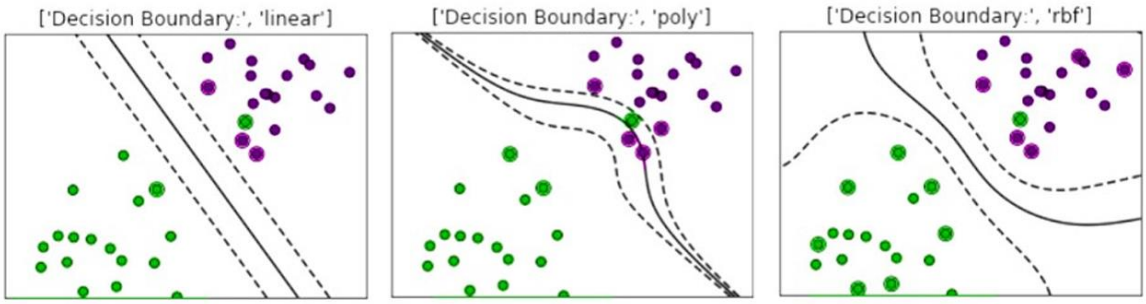

Figure 19:*SVM Algorithm with different kernels*

Support Vector Machine (SVM) is a supervised learning algorithm utilized for both classification and regression tasks. It finds the best hyperplane that separates the data into different classes in the feature space. By maximizing the margin (the

distance between the hyperplane and the nearest data points from any class) the optimal hyperplane is found

Table 15: *SVM with linear kernel algorithm*

Algorithm 5 SVM with Linear Kernel
1: **Input:** Training data $D = \{(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)\}$
2: **Output:** Classifier $f(x)$
3:
4: **Solve** the problem: $\min_{w,b} \frac{1}{2}\|w\|^2$
Subject to $y_i(w \cdot x + b) \geq 1$
i5: **Compute** decision function $f(x) = w \cdot x + b$
10: **Classify** new samples using $f(x)$

Table 16: *SVM with Polynomial kernel algorithm*

Algorithm 6 SVM with Polynomial Kernel
1: **Input:** Training data $D = \{(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)\}$ *degree d*
2: **Output:** Classifier $f(x)$
3:
4: **Solve** the optimization problem:
$$\min_{\alpha} \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j + 1)^d + b - \sum_i \alpha_i$$
Subject to $\sum_i \alpha_i y_i = 0$ and $\alpha_i \geq 0$
5: **Compute** decision function $f(x) = \sum_i \alpha_i y_i (x_i \cdot x + 1)^d + b$
10: **Classify** new samples using $f(x)$

Table 17: *SVM with RBF kernel Algorithm*

Algorithm 7 SVM with RBF Kernel

1: **Input:** Training data $D = \{(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)\}$, parameter γ

2: **Output:** Classifier $f(x)$

3:

4: **Solve** the optimization problem:

$$\min_{\alpha} \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j e^{-\gamma\|x_i - x\|^2} - \sum_i \alpha_i$$

Subject to $\sum_i \alpha_i y_i = 0$ and $\alpha_i \geq 0$

5: **Compute** decision function $f(x) = \sum_i \alpha_i y_i e^{-\gamma\|x_i - x\|^2} + b$

10: **Classify** new samples using $f(x)$

4.4.2 Deep Learning Classifiers

The Deep Learning models used in this study are commonly used with sequential data, which includes EEG signals. The CNN models implemented in this thesis are one-dimensional Convolutional Neural Networks designed to work well with time-series data such as EEG signals. There are 3 versions that will be referenced as CNN1, CNN2, and CNN3 throughout the thesis. CNN1 is taken from [119] and implemented with binary and 8 classes classification different from the original 10 classes implementation on only 3 subjects. The other two CNN models are further advancements of it.

CNN1 (see *Figure 20*)accepts input data and passes it to three convolutional Conv1D layers with batch normalization, kernel size=3, max-pooling with stride=2 and padding set to 'same'. Each of the Conv1D layers has the activation function ReLU, with the first and second layer having 128 filters, and the third layer having 64 filters.

The convolutional layers are followed by a Flatten layer making the 3D output of the last layer into a 1D vector. Three Fully Connected Dense layers follow, with dropout rate 0.2. The two first dense layers with 64 and 32 units respectively use hyperbolic tangent 'tanh' as activation function, while the last fully connected dense layer with 16 units, uses ReLU. The last dense layer is the output layer, where the number of units corresponds to the number of classes. The activation function is either softmax or sigmoid depending on the type of classification(multi-class/binary).

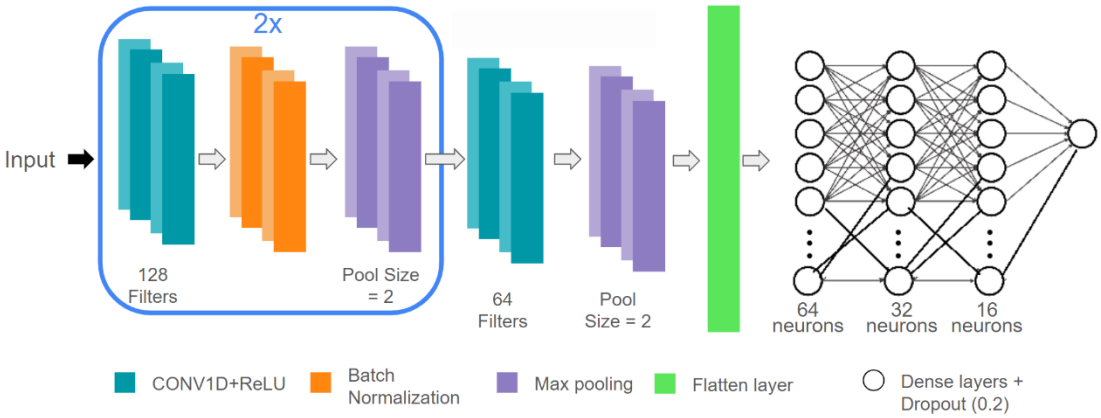

Figure 20:*CNN1 architecture*

CNN2 (see *Figure 22*) accepts input data and passes it to three convolutional blocks with a 1D convolutional layer Conv1D (kernel size=3), batch normalization, max-pooling with stride=2 and padding set to 'same'. Each of the Conv1D layers has the activation function ReLU. The first and second layer have 256 feature filters, and the third layer 128. The convolutional blocks are followed by a convolutional layer Conv1D with 128 filters and only max-pooling, without batch normalization.

The convolutions are followed by a Flatten layer making the 3D output of the last layer into a 1D vector. Four Fully Connected Dense layers follow, with dropout rate 0.2. The three first dense layers with 128, 64 and 32 units respectively use hyperbolic tangent ‘tanh’ as activation function, while the last fully connected dense layer with 16 units, uses ReLU. The last dense layer is the output layer, where the number of units corresponds to the number of classes. The activation function is either softmax or sigmoid depending on the type of classification(multi-class/binary).

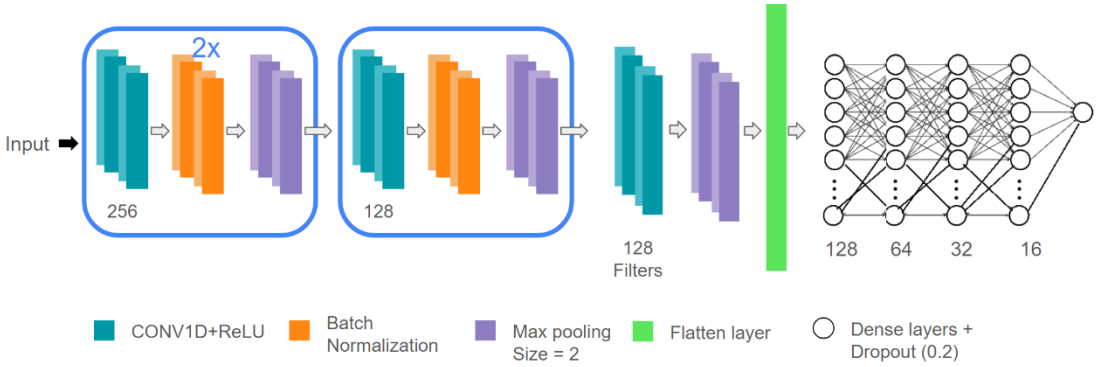

Figure 22:*CNN2 architecture*

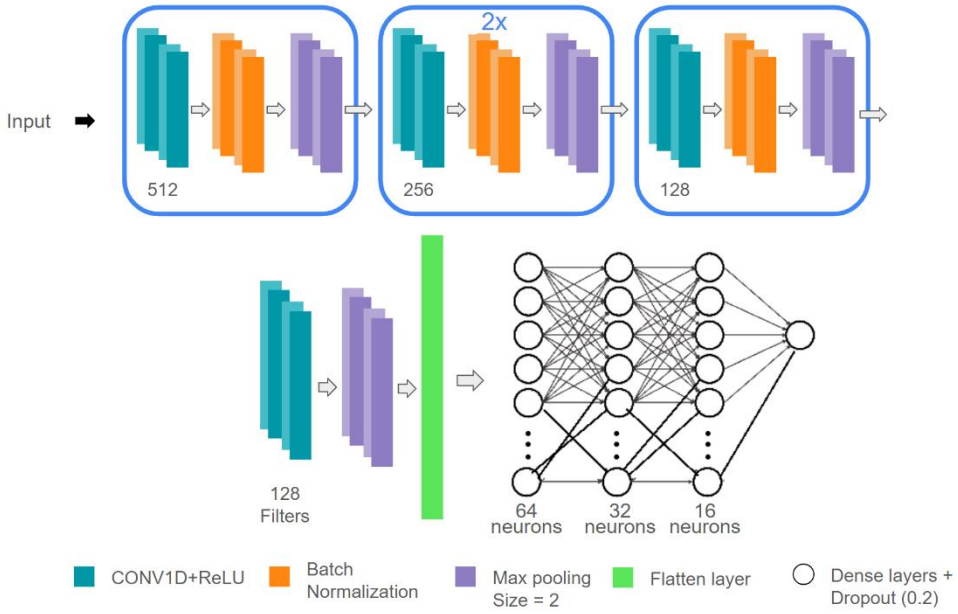

Figure 21:*CNN3 architecture*

CNN3 (See *Figure 21*) accepts input data and passes it to three convolutional blocks with a 1D convolutional layer Conv1D (kernel size=3), batch normalization, max-pooling with stride=2 and padding set to 'same'. Each of the Conv1D layers has the activation function ReLU. The first layer has 512 feature filters, the second and third layer have 256, and the fourth layer 128. The convolutional blocks are followed by a convolutional layer Conv1D with 128 filters and only max-pooling, without batch normalization.

The convolutions are followed by a Flatten layer making the 3D output of the last layer into a 1D vector. Three Fully Connected Dense layers follow, with dropout rate 0.2. The two first dense layers with 64 and 32 units respectively use hyperbolic tangent 'tanh' as activation function, while the last fully connected dense layer with 16 units, uses ReLU. The last dense layer is the output layer, where the number of units corresponds to the number of classes. The activation function is either softmax or sigmoid depending on the type of classification(multi-class/binary).

4.4.3 Optimizers and learning rates

The selection of the optimizer and learning rate are crucial to the efficient implementation of a DL model. To achieve the best utilization of the models at hand, a grid search of the best parameters is utilized. For optimizers Adam, Stochastic Gradient Descent and RMSprop were considered, while the learning rates included 0.01, 0.001, and 0.0001. The training was done for 50 epochs with model CNN1 for DEAP time CNN dataset (32 channels), and the best combination of the trials was then used for all the other implementations of CNN1 with different datasets. The pair with the best result was optimizer: Adam and learning rate:0.001.

Table 18: *Grid Search for Parameters*

Optimizers	**Learning rate**	**Accuracy**	**Precision_0**	**Racall_0**	**F1-score_0**	**Precision_1**	**Racall_1**	**F1-score_1**
Adam	0.01	0.664	0.000	0.000	0.000	0.664	1.000	0.798
Adam	**0.001**	**0.835**	**0.770**	**0.726**	**0.747**	**0.865**	**0.890**	**0.877**
Adam	0.0001	0.828	0.781	0.679	0.726	0.847	0.903	0.875
SGD	0.01	0.794	0.734	0.608	0.665	0.817	0.889	0.851
SGD	0.001	0.715	0.643	0.343	0.448	0.731	0.904	0.808
SGD	0.0001	0.676	0.610	0.102	0.175	0.680	0.967	0.790
RMSpropp	0.01	0.664	0.000	0.000	0.000	0.664	1.000	0.798
RMSprop	0.001	0.798	0.788	0.548	0.647	0.802	0.925	0.859
RMSprop	0.0001	0.817	0.743	0.700	0.721	0.852	0.877	0.865

4.4.4 Environment Details

The environments used to implement the models where Google Collaboratory and Kaggle Environment with the respective resources: 16 GB Tesla T4 GPU, 12.7 GB CPU for Google Collaboratory, 2x 15 GB T4 GPU, 29 GB CPU for Kaggle. The most crucial libraries used where TensorFlow 2.15.0, Keras 3.4.1.

4.4.5 Performance Evaluation

To evaluate the performance of the implemented models several evaluation metrics are used. For ML models the average accuracy for binary classification of Valence, Arousal, and Dominance, and the additional eight-class classification is calculated during each of the cross-validation runs. In addition to that, the average time for the implementation of the model is given in seconds. The Deep Learning models are evaluated on the accuracy, precision, recall, and f1-score metrics. The average duration of an epoch and the total time of implementation is also recorded in seconds.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \qquad \textit{Equation 12}$$

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \qquad \textit{Equation 13}$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \qquad \textit{Equation 14}$$

$$\boldsymbol{F1-Score} = \boldsymbol{2} \cdot \frac{\boldsymbol{Precision \cdot Recall}}{\boldsymbol{Precision+Recall}} \qquad \textit{Equation 15}$$

CHAPTER 5

EXPERIMENTAL RESULTS

5.1. Implementation Results

In this chapter are presented the results of all the implementations of the different classifiers for both of the datasets. There are two types of experiments carried out: the ML implementation on both time and frequency features with DEAP and DREAMER, and the Deep Learning Model with the one-dimensional CNN model. For ML implementations there are 7 classifiers that are trained and tested on 4 different cross validation trials. The DL implementation involves several trials with DEAP and DREAMER datasets. With DEAP both time and frequency domain features were applied separately on 32, 18, 14, and 10 channels of the DEAP dataset, in order to investigate on the channels' influence on performance. In addition to that, the CNN model is used for the time and frequency domain feature datasets for DREAMER, separately in this case as well. The classifications done were of the binary scheme for Valence, Arousal, and Dominance, but also including the 3D emotion labeling scheme which is also utilized. This brings the number of total implementations to 488 individual classifier implementations: for ML: 4*7*4 classifiers per dataset *4 ML datasets = 448; 4 channel configurations of DEAP x 2 datasets x 4 classification labels = 32, and finally 8 implementations from DREAMER CNN. The results of these implementations are given by the tables below, where they are grouped according to their similarities.

5.1.1 Removing baseline effect

In the following tables the effect of removing the mean baseline from each epoch is given by comparing the results of Machine Learning algorithms using the original data and the cleaned version of the dataset.

Table 19: *ML algorithms on DEAP time-domain dataset with baseline/without baseline recordings*

		Unclean			**Clean**		
Algorithm	k-fold	Val.	Aro.	Dom.	Val.	Aro.	Dom.
DT	5	0.535	0.558	0.567	0.520	0.572	0.585
LDA	5	0.590	0.641	0.617	0.612	0.590	0.611
KNN	5	0.585	0.623	0.617	0.591	0.603	0.624
RF	5	0.609	0.650	0.633	0.615	0.645	0.658
SVM_linear	5	0.589	0.632	0.621	0.600	0.589	0.604
SVM_poly	5	0.612	0.625	0.646	0.626	0.634	0.641
SVM_rbf	5	0.613	0.646	0.655	0.630	0.637	0.666
DT	10	0.534	0.550	0.590	0.546	0.561	0.595
LDA	10	0.598	0.638	0.603	0.592	0.591	0.614
KNN	10	0.584	0.625	0.625	0.591	0.609	0.611
RF	10	0.624	0.644	0.646	0.644	0.644	0.650
SVM_linear	10	0.598	0.641	0.618	0.591	0.587	0.611
SVM_poly	10	0.616	0.623	0.645	0.630	0.635	0.644
SVM_rbf	10	0.623	0.653	0.657	0.629	0.644	0.663
DT	15	0.569	0.549	0.566	0.526	0.570	0.565
LDA	15	0.591	0.626	0.593	0.592	0.607	0.606
KNN	15	0.590	0.621	0.627	0.589	0.612	0.612
RF	15	0.626	0.643	0.663	0.621	0.654	0.664
SVM_linear	15	0.593	0.637	0.623	0.584	0.595	0.600
SVM_poly	15	0.614	0.623	0.646	0.629	0.635	0.645
SVM_rbf	15	0.621	0.652	0.660	0.628	0.646	0.669
DT	20	0.537	0.533	0.546	0.537	0.578	0.591
LDA	20	0.585	0.632	0.596	0.592	0.600	0.607
KNN	20	0.590	0.618	0.634	0.591	0.607	0.616
RF	20	0.632	0.653	0.651	0.623	0.650	0.659
SVM_linear	20	0.597	0.635	0.635	0.593	0.582	0.615
SVM_poly	20	0.614	0.623	0.646	0.629	0.636	0.644
SVM_rbf	20	0.623	0.655	0.657	0.630	0.642	0.668

Table 20:*ML algorithms on DREAMER time-domain dataset with baseline/without baseline recordings*

		Unclean			**Clean**		
Algorithm	k-fold	Val.	Aro.	Dom.	Val.	Aro.	Dom.
DT	5	0.676	0.815	0.841	0.680	0.805	0.835
LDA	5	0.628	0.734	0.767	0.616	0.720	0.741
KNN	5	0.601	0.692	0.737	0.578	0.688	0.720
RF	5	0.761	0.820	0.850	0.727	0.799	0.727
SVM_linear	5	0.624	0.702	0.737	0.585	0.692	0.585
SVM_poly	5	0.618	0.771	0.802	0.627	0.773	0.627
SVM_rbf	5	0.692	0.780	0.811	0.676	0.774	0.676
DT	10	0.683	0.816	0.844	0.685	0.806	0.838
LDA	10	0.636	0.739	0.777	0.619	0.734	0.755
KNN	10	0.607	0.703	0.743	0.589	0.698	0.729
RF	10	0.760	0.822	0.852	0.740	0.803	0.740
SVM_linear	10	0.619	0.708	0.741	0.604	0.703	0.604
SVM_poly	10	0.622	0.772	0.802	0.628	0.773	0.628
SVM_rbf	10	0.698	0.781	0.812	0.678	0.776	0.678
DT	15	0.685	0.815	0.844	0.686	0.808	0.841
LDA	15	0.641	0.738	0.783	0.622	0.726	0.758
KNN	15	0.605	0.707	0.751	0.592	0.697	0.728
RF	15	0.764	0.823	0.854	0.735	0.804	0.735
SVM_linear	15	0.624	0.710	0.743	0.602	0.704	0.602
SVM_poly	15	0.624	0.772	0.803	0.627	0.772	0.627
SVM_rbf	15	0.700	0.781	0.812	0.682	0.776	0.682
DT	20	0.685	0.816	0.844	0.688	0.809	0.841
LDA	20	0.645	0.740	0.778	0.618	0.724	0.759
KNN	20	0.604	0.711	0.749	0.592	0.700	0.733
RF	20	0.766	0.820	0.854	0.734	0.805	0.734
SVM_linear	20	0.623	0.717	0.747	0.603	0.703	0.603
SVM_poly	20	0.625	0.773	0.803	0.628	0.773	0.628
SVM_rbf	20	0.699	0.781	0.814	0.682	0.777	0.682

5.1.2 Machine Learning Implementation Results

The following tables show the results for the implementations of the Machine Learning Algorithms for the time-domain and frequency-domain datasets of DEAP and DREAMER.

Table 21: *DEAP ML time-domain dataset results (cont.)*

Algorithm	kfolds	acc_val	time_val	acc_aro	time_aro	acc_dom	time_dom	acc_8	time_8
DT	5	0.535	120.426	0.558	130.518	0.567	92.758	0.22	109.48
LDA	5	0.59	71.056	0.641	71.949	0.617	89.391	0.289	96.62
KNN	5	0.585	3.43	0.623	3.64	0.617	2.401	0.274	2.677
RF	5	0.609	62.332	0.65	66.065	0.633	58.225	0.372	65.59
SVM_linear	5	0.589	140.992	0.632	183.172	0.621	166.881	0.322	316.925
SVM_poly	5	0.612	82.711	0.625	81.668	0.646	79.135	0.366	158.112
SVM_rbf	5	0.613	98.231	0.646	94.415	0.655	97.76	0.37	167.278
DT	10	0.534	268.333	0.55	303.025	0.59	224.803	0.237	260.282
LDA	10	0.598	134.315	0.638	135.689	0.603	140.304	0.291	126.747
KNN	10	0.584	3.275	0.625	4.582	0.625	3.437	0.289	3.565
RF	10	0.624	144.553	0.644	160.768	0.646	136.533	0.368	145.159
SVM_linear	10	0.598	333.482	0.641	447.23	0.618	352.185	0.332	696.656
SVM_poly	10	0.616	190.657	0.623	185.103	0.645	188.967	0.366	308.55
SVM_rbf	10	0.623	201.946	0.653	205.294	0.657	212.636	0.371	344.145

Table 22:*DEAP ML time-domain dataset results (part 2)*

Algorithm	kfolds	acc_va	time_val	acc_aro	time_aro	acc_dom	time_dom	acc_8	time_8
DT	15	0.569	493.605	0.549	533	0.566	355.802	0.229	425.547
LDA	15	0.591	218.47	0.626	199.41	0.593	179.098	0.284	194.982
KNN	15	0.59	5.883	0.621	4.085	0.627	5.555	0.286	4.859
RF	15	0.626	229.468	0.643	249.713	0.663	203.062	0.364	232.954
SVM_linear	15	0.593	552.979	0.637	728.339	0.623	527.271	0.34	1136.252
SVM_poly	15	0.614	293.756	0.623	305.511	0.646	256.133	0.363	503.223
SVM_rbf	15	0.621	308.731	0.652	307.423	0.66	282.596	0.371	544.751
DT	20	0.537	612.687	0.533	673.48	0.546	461.891	0.227	582.469
LDA	20	0.585	294.884	0.632	274.904	0.596	244.548	0.289	278.172
KNN	20	0.59	5.247	0.618	6.533	0.634	6.274	0.285	5.811
RF	20	0.632	313.5	0.653	344.875	0.651	274.173	0.361	317.553
SVM_linear	20	0.597	757.978	0.635	939.173	0.635	710.348	0.34	1534.879
SVM_poly	20	0.614	399.599	0.623	393.974	0.646	342.123	0.363	671.168
SVM_rbf	20	0.623	418.477	0.655	411.754	0.657	373.972	0.371	717.203

Table 23:*DEAP frequency-domain dataset results (part 1)*

Algorithm	**kfolds**	**acc_val**	**time_val**	**acc_aro**	**time_aro**	**acc_dom**	**time_dom**	**acc_8**	**time_8**
DT	5	0.645	46.432	0.637	41.506	0.629	39.001	0.252	47.744
LDA	5	0.651	43.917	0.654	43.788	0.661	43.174	0.27	41.686
KNN	5	0.627	1.153	0.628	1.211	0.632	2.403	0.284	2.186
RF	5	0.627	32.516	0.628	31.534	0.652	29.4	0.357	35.289
SVM_linear	5	0.63	53.983	0.639	58.012	0.656	50.019	0.365	72.767
SVM_poly	5	0.588	47.443	0.569	42.931	0.6	42.319	0.368	51.685
SVM_rbf	5	0.532	49.184	0.62	51.492	0.607	46.113	0.37	61.216
DT	10	0.644	107.708	0.61	102.27	0.635	97.247	0.245	109.54
LDA	10	0.625	71.285	0.655	71.692	0.666	73.236	0.253	87.61
KNN	10	0.623	1.589	0.637	3.215	0.633	2.884	0.284	1.93
RF	10	0.628	73.346	0.627	72.649	0.654	66.509	0.371	83.838
SVM_linear	10	0.63	129.098	0.647	139.643	0.662	116.791	0.363	258.651
SVM_poly	10	0.6	102.806	0.59	104.77	0.614	103.633	0.367	190.756
SVM_rbf	10	0.55	108.822	0.617	110.321	0.606	104.425	0.37	214.515

Table 24:*DEAP ML frequency-domain dataset results (part 2)*

Algorithm	**kfolds**	**acc_val**	**time_val**	**acc_aro**	**time_aro**	**acc_dom**	**time_dom**	**acc_8**	**time_8**
DT	15	0.642	165.871	0.599	159.127	0.637	149.059	0.262	173.068
LDA	15	0.634	110.482	0.649	116.588	0.662	118.167	0.254	135.474
KNN	15	0.627	1.961	0.618	1.996	0.643	2.24	0.283	4.275
RF	15	0.63	116.731	0.63	127.477	0.655	103.363	0.366	131.153
SVM_linear	15	0.631	207.138	0.647	227.72	0.666	184.589	0.364	402.536
SVM_poly	15	0.588	172.198	0.586	164.812	0.588	151.169	0.368	290.145
SVM_rbf	15	0.554	163.951	0.609	172.2	0.604	153.773	0.369	267.642
DT	20	0.64	217.744	0.605	220.718	0.646	200.162	0.265	229.088
LDA	20	0.633	158.039	0.655	162.743	0.666	162.147	0.249	161.163
KNN	20	0.628	2.76	0.621	2.331	0.641	2.363	0.289	2.347
RF	20	0.628	157.254	0.628	156.487	0.655	144.923	0.372	164.893
SVM_linear	20	0.63	283.298	0.648	301.44	0.664	265.859	0.365	383.228
SVM_poly	20	0.576	218.992	0.596	212.586	0.603	223.297	0.367	262.512
SVM_rbf	20	0.554	220.769	0.622	227.483	0.599	215.718	0.37	285.533

***Table 25**:DREAMER time-domain dataset results (part 1)*

Algorithm	kfolds	acc_val	time_val	acc_aro	time_aro	acc_dom	time_dom	acc_8	time_8
DT	5	0.676	52.868	0.815	102.660	0.841	103.908	0.353	57.906
LDA	5	0.628	82.740	0.734	77.783	0.767	83.172	0.162	65.571
KNN	5	0.601	2.643	0.692	1.570	0.737	1.827	0.465	1.652
RF	5	0.761	64.662	0.820	104.287	0.850	109.122	0.521	69.191
SVM_linear	5	0.624	91.221	0.702	52.179	0.737	45.523	0.399	77.478
SVM_poly	5	0.618	84.417	0.771	74.994	0.802	78.577	0.426	112.140
SVM_rbf	5	0.692	89.253	0.780	67.606	0.811	63.317	0.488	101.933
DT	10	0.683	130.931	0.816	244.128	0.844	281.253	0.367	139.651
LDA	10	0.636	199.499	0.739	191.114	0.777	204.411	0.174	168.055
KNN	10	0.607	3.710	0.703	2.751	0.743	3.820	0.470	1.981
RF	10	0.760	148.176	0.822	235.110	0.852	252.415	0.525	159.024
SVM_linear	10	0.619	207.770	0.708	114.140	0.741	102.619	0.408	186.137
SVM_poly	10	0.622	189.987	0.772	166.428	0.802	180.740	0.424	285.135
SVM_rbf	10	0.698	178.053	0.781	144.531	0.812	148.812	0.492	234.935

Table 26:*DREAMER time-domain dataset results (part 2)*

Algorithm	kfolds	acc_val	time_val	acc_aro	time_aro	acc_dom	time_dom	acc_8	time_8
DT	15	0.685	194.541	0.815	367.385	0.844	399.965	0.368	217.425
LDA	15	0.641	311.642	0.738	316.439	0.783	321.430	0.153	273.303
KNN	15	0.605	2.531	0.707	2.271	0.751	2.385	0.473	2.206
RF	15	0.764	229.082	0.823	372.135	0.854	399.482	0.536	249.310
SVM_linear	15	0.624	334.439	0.710	168.113	0.743	156.796	0.415	305.276
SVM_poly	15	0.624	270.862	0.772	255.133	0.803	278.172	0.425	447.190
SVM_rbf	15	0.700	274.769	0.781	232.423	0.812	223.347	0.497	363.159
DT	20	0.685	277.114	0.816	519.145	0.844	499.383	0.373	295.120
LDA	20	0.645	423.488	0.740	449.042	0.778	440.368	0.138	371.098
KNN	20	0.604	2.535	0.711	2.745	0.749	2.550	0.474	4.119
RF	20	0.766	316.109	0.820	521.663	0.854	541.358	0.530	339.827
SVM_linear	20	0.623	455.053	0.717	287.734	0.747	215.185	0.407	412.220
SVM_poly	20	0.625	370.853	0.773	362.142	0.803	381.893	0.426	609.308
SVM_rbf	20	0.699	407.009	0.781	293.937	0.814	303.971	0.498	497.423

Table 27:DREAMER frequency-domain dataset results (part 1)

Algorithm	kfolds	acc_val	time_val	acc_aro	time_aro	acc_dom	time_dom	acc_8	time_8
DT	5	0.649	38.905	0.727	90.308	0.762	97.152	0.364	48.628
LDA	5	0.599	62.467	0.711	59.066	0.737	60.151	0.243	52.42
KNN	5	0.702	1.448	0.821	1.835	0.851	1.471	0.476	1.438
RF	5	0.748	53.884	0.805	79.541	0.842	95.029	0.531	59.861
SVM_linear	5	0.623	67.303	0.721	38.52	0.754	38.119	0.372	66.017
SVM_poly	5	0.62	57.552	0.776	49.655	0.805	46.391	0.426	82.34
SVM_rbf	5	0.658	59.853	0.785	48.952	0.818	43.572	0.449	69.07
DT	10	0.65	91.699	0.724	176.056	0.763	212.403	0.373	106.158
LDA	10	0.603	150.24	0.719	140.318	0.75	145.373	0.179	128.72
KNN	10	0.707	1.881	0.821	1.766	0.85	1.762	0.496	2.948
RF	10	0.759	125.212	0.813	185.535	0.842	216.612	0.544	134.615
SVM_linear	10	0.626	214.888	0.731	101.034	0.763	87.744	0.376	152.891
SVM_poly	10	0.62	130.234	0.776	120.195	0.805	101.172	0.427	181.195
SVM_rbf	10	0.663	130.097	0.787	98.698	0.819	95.24	0.456	166.569

Table 28:*DREAMER frequency-domain dataset results (part 2)*

Algorithm	**kfolds**	**acc_val**	**time_val**	**acc_aro**	**time_aro**	**acc_dom**	**time_dom**	**acc_8**	**time_8**
DT	15	0.648	137.962	0.736	300.198	0.755	308.38	0.382	165.929
LDA	15	0.609	227.676	0.725	219.696	0.752	237.214	0.176	207.323
KNN	15	0.709	1.972	0.823	1.957	0.848	2.043	0.499	1.921
RF	15	0.765	195.022	0.814	289.902	0.839	346.882	0.547	214.607
SVM_linear	15	0.63	326.807	0.726	161.861	0.764	141.628	0.378	250.181
SVM_poly	15	0.621	199.347	0.778	186.428	0.806	159.134	0.426	323.158
SVM_rbf	15	0.665	199.102	0.787	149.545	0.819	142.437	0.455	272.222
DT	20	0.664	193.411	0.729	394.442	0.76	418.472	0.383	227.525
LDA	20	0.611	311.471	0.721	297.881	0.749	312.636	0.155	291.793
KNN	20	0.711	2.178	0.822	2.139	0.85	2.163	0.504	2.124
RF	20	0.757	265.524	0.819	390.138	0.844	466.624	0.551	288.126
SVM_linear	20	0.632	471.766	0.728	229.109	0.765	189.526	0.373	344.836
SVM_poly	20	0.62	268.261	0.778	262.288	0.805	215.108	0.427	415.992
SVM_rbf	20	0.665	265.157	0.787	207.077	0.819	197.095	0.454	360.118

5.1.3 Deep Learning Implementation Results

The following tables show the results for the implementations of the three CNN models for the time-domain and frequency-domain datasets of DEAP and DREAMER.

Table 29: *DEAP CNN1 results for time and frequency-domain (0: arousal, 1: valence, 2: dominance, 3:3D model)*

domain	Nr	total(s)	avg(s)	label	acc	prec.	recall	f1-score
time	32	3322	16.611	0	0.881	0.881	0.881	0.88
	18	1859	9.297	0	0.846	0.845	0.846	0.845
	14	1491	7.455	0	0.817	0.815	0.817	0.815
	10	1218	6.088	0	0.795	0.794	0.795	0.789
	32	3544	17.721	1	0.887	0.887	0.888	0.886
	18	1863	9.316	1	0.856	0.855	0.856	0.854
	14	1499	7.495	1	0.815	0.814	0.815	0.81
	10	1224	6.118	1	0.799	0.799	0.799	0.79
	32	3252	16.259	2	0.899	0.898	0.899	0.899
	18	1817	9.087	2	0.846	0.844	0.846	0.842
	14	1476	7.378	2	0.824	0.821	0.824	0.82
	10	1203	6.013	2	0.812	0.808	0.812	0.806
	32	1659	16.591	3	0.574	0.689	0.449	0.52
	18	915	9.148	3	0.697	0.781	0.623	0.685
	14	760	7.6	3	0.574	0.689	0.449	0.52
	10	631	6.305	3	0.57	0.701	0.436	0.512
freq.	32	1940	9.701	0	0.87	0.87	0.87	0.87
	18	1321	6.605	0	0.834	0.833	0.834	0.834
	14	1154	5.772	0	0.818	0.816	0.818	0.816
	10	932	4.66	0	0.802	0.8	0.802	0.798
	32	1958	9.792	1	0.884	0.884	0.884	0.882
	18	1332	6.661	1	0.846	0.845	0.846	0.843
	14	1156	5.778	1	0.831	0.829	0.831	0.828
	10	929	4.645	1	0.803	0.8	0.803	0.8
	32	1967	9.837	2	0.89	0.889	0.89	0.89
	18	1379	6.895	2	0.852	0.85	0.852	0.85
	14	1172	5.859	2	0.832	0.829	0.832	0.828
	10	940	4.702	2	0.819	0.816	0.819	0.816
	32	2473	24.73	3	0.802	0.85	0.765	0.804
	18	1421	14.215	3	0.709	0.783	0.642	0.703
	14	1225	12.26	3	0.672	0.745	0.602	0.662
	10	991	9.917	3	0.667	0.745	0.595	0.654

Table 30:*DEAP CNN2 results for time and frequency-domain (0:arousal, 1:valence, 2:dominance, 3:3D model)*

domain	Nr	total(s)	avg(s)	label	acc	prec.	recall	f1-score
time	32	3445	34.453	0	0.897	0.897	0.897	0.897
	18	1959	19.59	0	0.89	0.89	0.89	0.889
	14	1539	15.387	0	0.863	0.862	0.863	0.862
	10	1236	12.358	0	0.793	0.79	0.793	0.789
	32	3490	34.901	1	0.917	0.916	0.917	0.916
	18	1982	19.822	1	0.87	0.869	0.87	0.869
	14	1551	15.51	1	0.864	0.863	0.864	0.862
	10	1213	12.125	1	0.831	0.83	0.831	0.83
	32	3490	34.904	2	0.925	0.925	0.925	0.925
	18	1951	19.506	2	0.882	0.881	0.882	0.881
	14	1584	15.839	2	0.842	0.841	0.842	0.842
	10	1221	12.207	2	0.791	0.786	0.791	0.786
	32	3549	35.488	3	0.807	0.848	0.776	0.808
	18	2038	20.38	3	0.677	0.758	0.605	0.664
	14	1585	15.846	3	0.559	0.605	0.501	0.536
	10	1264	12.639	3	0.548	0.614	0.462	0.516
freq.	32	2293	22.933	0	0.911	0.911	0.911	0.911
	18	1399	13.991	0	0.881	0.881	0.881	0.881
	14	1190	11.897	0	0.881	0.88	0.881	0.88
	10	948	9.481	0	0.848	0.848	0.848	0.848
	32	2183	21.826	1	0.92	0.92	0.92	0.92
	18	1350	13.503	1	0.895	0.895	0.895	0.894
	14	1149	11.489	1	0.877	0.877	0.877	0.877
	10	918	9.185	1	0.866	0.865	0.866	0.864
	32	2345	23.447	2	0.925	0.925	0.925	0.925
	18	1427	14.265	2	0.885	0.885	0.885	0.885
	14	1208	12.076	2	0.883	0.881	0.883	0.882
	10	900	8.997	2	0.87	0.868	0.87	0.868
	32	2353	23.527	3	0.787	0.839	0.742	0.786
	18	1484	14.843	3	0.716	0.785	0.659	0.712
	14	1218	12.182	3	0.693	0.774	0.616	0.678
	10	998	9.976	3	0.679	0.752	0.604	0.666

Table 31:*DEAP CNN3 results for time and frequency-domain (0:arousal, 1:valence, 2:dominance, 3:3D model)*

domain	***Nr***	***total(s)***	***avg(s)***	***label***	***acc***	***prec.***	***recall***	***f1-score***
time	32	5853	58.53	0	0.925	0.925	0.924	0.924
	18	3412	34.12	0	0.911	0.911	0.91	0.911
	14	2726	27.26	0	0.898	0.897	0.898	0.897
	10	2241	22.41	0	0.881	0.881	0.881	0.881
	32	5821	58.21	1	0.932	0.932	0.932	0.932
	18	3287	32.87	1	0.919	0.919	0.919	0.919
	14	2621	26.21	1	0.887	0.887	0.887	0.887
	10	1962	19.62	1	0.874	0.874	0.874	0.874
	32	5776	57.76	2	0.935	0.935	0.935	0.935
	18	3290	32.9	2	0.88	0.88	0.88	0.88
	14	2637	26.37	2	0.892	0.893	0.892	0.892
	10	2005	20.05	2	0.875	0.875	0.875	0.875
	32	5993	59.93	3	0.821	0.846	0.803	0.822
	18	3331	33.31	3	0.734	0.769	0.708	0.736
	14	2757	27.57	3	0.759	0.796	0.728	0.759
	10	2094	20.94	3	0.735	0.78	0.697	0.735
freq.	32	3899	38.99	0	0.926	0.927	0.926	0.926
	18	2355	23.55	0	0.904	0.904	0.904	0.904
	14	1936	19.36	0	0.903	0.903	0.903	0.903
	10	1774	17.74	0	0.887	0.887	0.887	0.887
	32	3787	37.87	1	0.937	0.937	0.937	0.937
	18	2211	22.11	1	0.917	0.916	0.917	0.917
	14	1883	18.83	1	0.912	0.991	0.912	0.912
	10	1447	14.47	1	0.891	0.89	0.891	0.89
	32	3801	38.01	2	0.941	0.941	0.941	0.941
	18	2337	23.37	2	0.916	0.916	0.916	0.916
	14	2657	26.57	2	0.902	0.901	0.902	0.901
	10	2010	20.1	2	0.889	0.889	0.889	0.889
	32	3897	38.97	3	0.8	0.825	0.783	0.802
	18	2387	23.87	3	0.782	0.82	0.75	0.783
	14	2106	21.06	3	0.767	0.81	0.731	0.768
	10	1608	16.08	3	0.746	0.785	0.715	0.745

Table 32:*DREAMER CNN results for time and frequency-domain (0:arousal, 1:valence, 2:dominance, 3:3D model)*

Model	domain	total(s)	avg(s)	label	acc	prec.	recall	f1-score
CNN1	time	806	4.032	0	0.923	0.922	0.923	0.921
		805	4.023	1	0.848	0.847	0.848	0.847
		806	4.029	2	0.91	0.909	0.91	0.91
		840	4.198	3	0.727	0.788	0.67	0.719
	freq.	643	3.215	0	0.936	0.935	0.936	0.935
		650	3.248	1	0.881	0.881	0.881	0.881
		635	3.174	2	0.944	0.943	0.944	0.942
		666	3.331	3	0.814	0.844	0.79	0.814
CNN2	time	838	8.376	0	0.883	0.883	0.883	0.882
		845	8.454	1	0.688	0.709	0.688	0.697
		880	8.798	2	0.915	0.915	0.915	0.915
		889	8.892	3	0.672	0.729	0.62	0.667
	freq.	657	6.573	0	0.877	0.877	0.877	0.877
		652	6.521	1	0.924	0.923	0.924	0.924
		652	6.524	2	0.949	0.948	0.949	0.949
		692	6.925	3	0.803	0.816	0.794	0.802
CNN3	time	1440	14.4	0	0.867	0.878	0.867	0.871
		1453	14.53	1	0.825	0.823	0.825	0.823
		1440	14.4	2	0.927	0.929	0.929	0.927
		1538	15.38	3	0.635	0.697	0.605	0.635
	freq.	1095	10.95	0	0.945	0.945	0.945	0.945
		1082	10.82	1	0.884	0.885	0.884	0.884
		1094	10.94	2	0.955	0.954	0.955	0.955
		1160	11.6	3	0.778	0.785	0.774	0.779

CHAPTER 6

DISCUSSION OF RESULTS

6.1. Machine Learning Results

In this thesis we construct several classifiers (Machine Learning and Deep Learning classifiers) for the task of Emotion Recognition with the usage of EEG affective signals. The Machine Learning Classifiers Decision Tree, K-Nearest Neighbor, Linear Discriminant Analysis, Random Forest, and three Support Vector Machine classifiers (with linear/polynomial/rbf kernels) were tested on both DEAP [1] and DREAMER [2] datasets and evaluated by the accuracy metric. For DEAP the results are in par with the pre-existing implementations, since the accuracies for DEAP time-domain dataset and frequency-domain dataset did not surpass the 68% mark with cross-validation (k=5,10,15,20). On the other hand, both instances of the DREAMER dataset have a satisfactory performance with the best performing model scoring 85% only with Machine Learning classifiers. This performance could be most likely attributed to the removal of the baseline signals from the affective recordings in the pre-processing stage.

In the following histograms there are several points of interest. First of all, it is observed that across both the DREAMER and DEAP datasets the Dominance Binary Scale has the highest accuracy, followed by Arousal, leaving Valence as the less accurate binary emotion category to classify. While not very apparent in the DREAMER dataset results, when observing the DEAP counterpart, the classifiers mostly reach their highest when k=5 or 10, with a slight drop for k=20. Moreover, the graph shows that for DEAP Random Forest is the classifier with the most reliable performance.

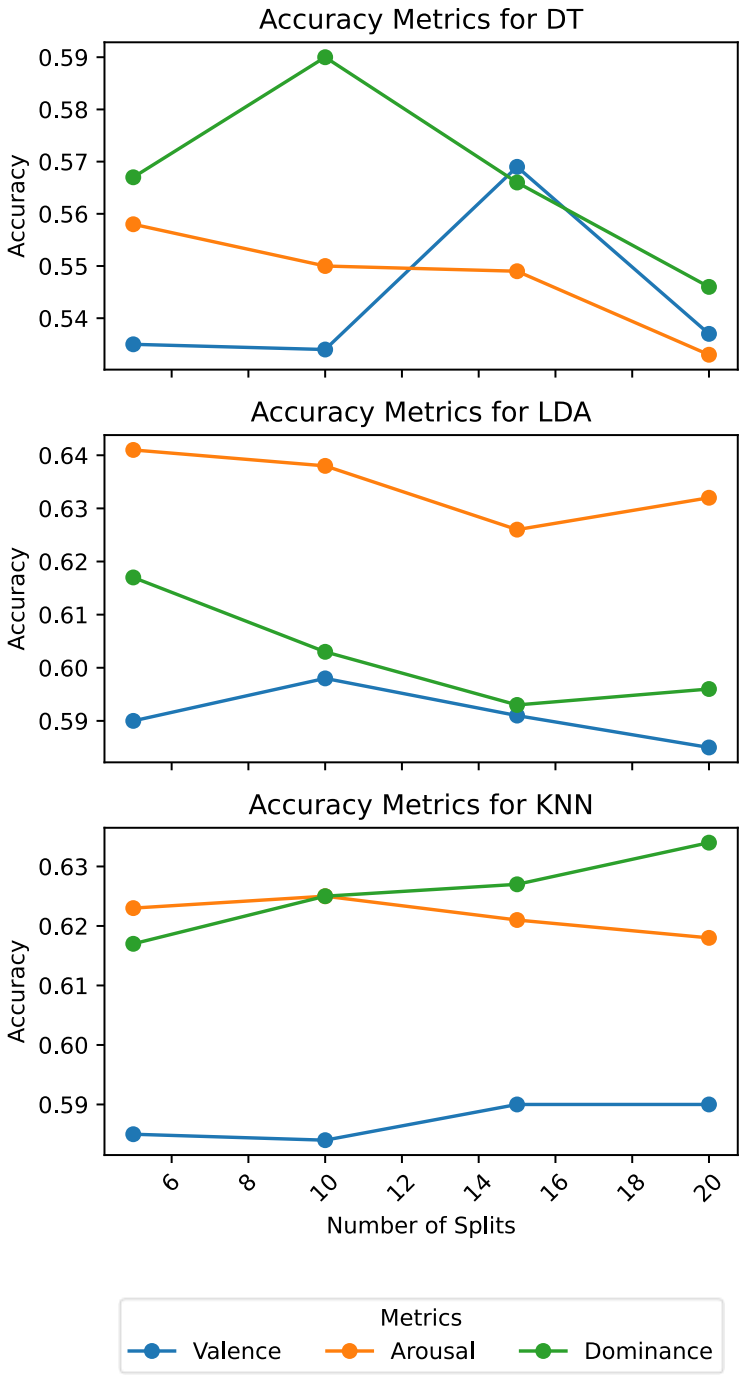

Figure 23:*DEAP time-domain Accuracy vs Nr of k-folds*

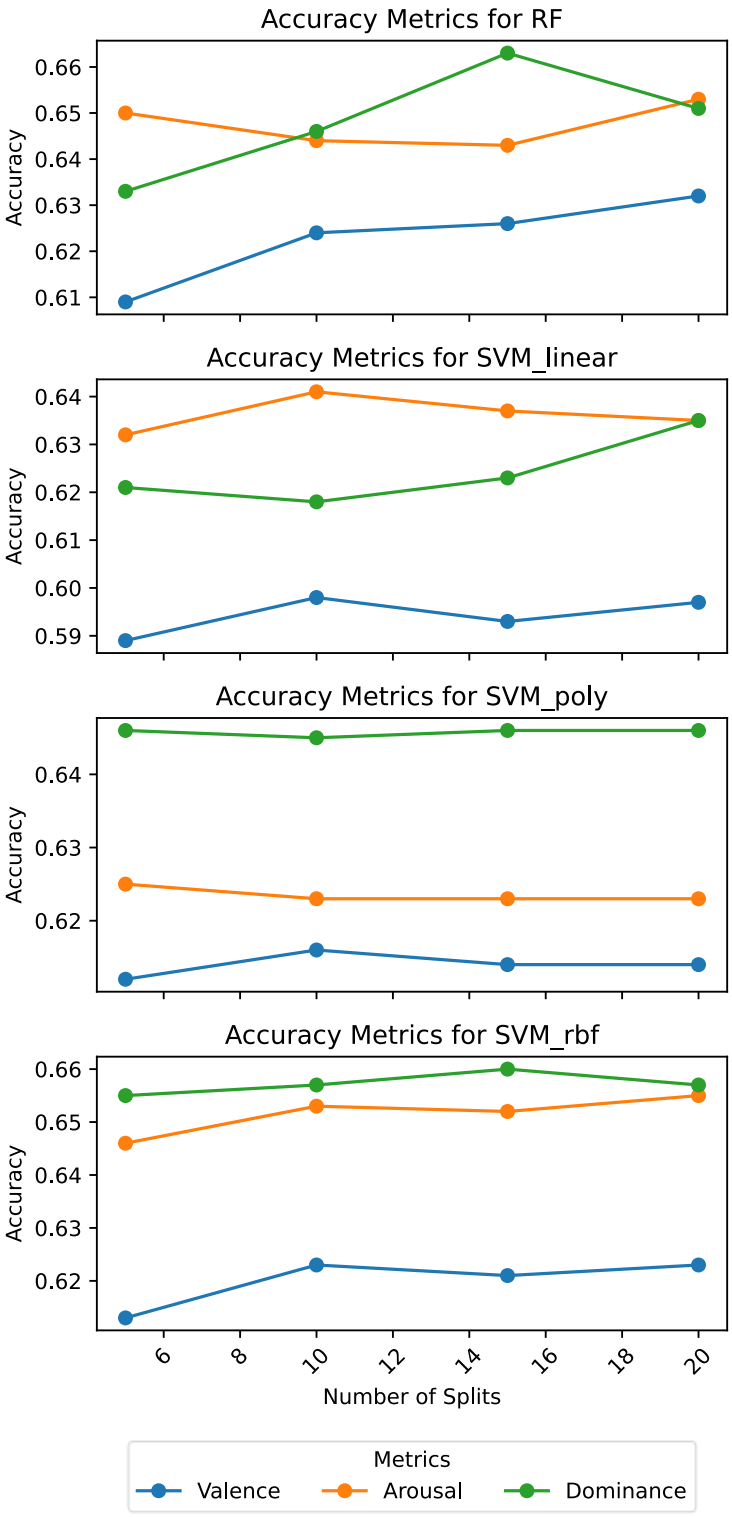

Figure 24:*DEAP time-domain Accuracy vs Nr of k-folds*

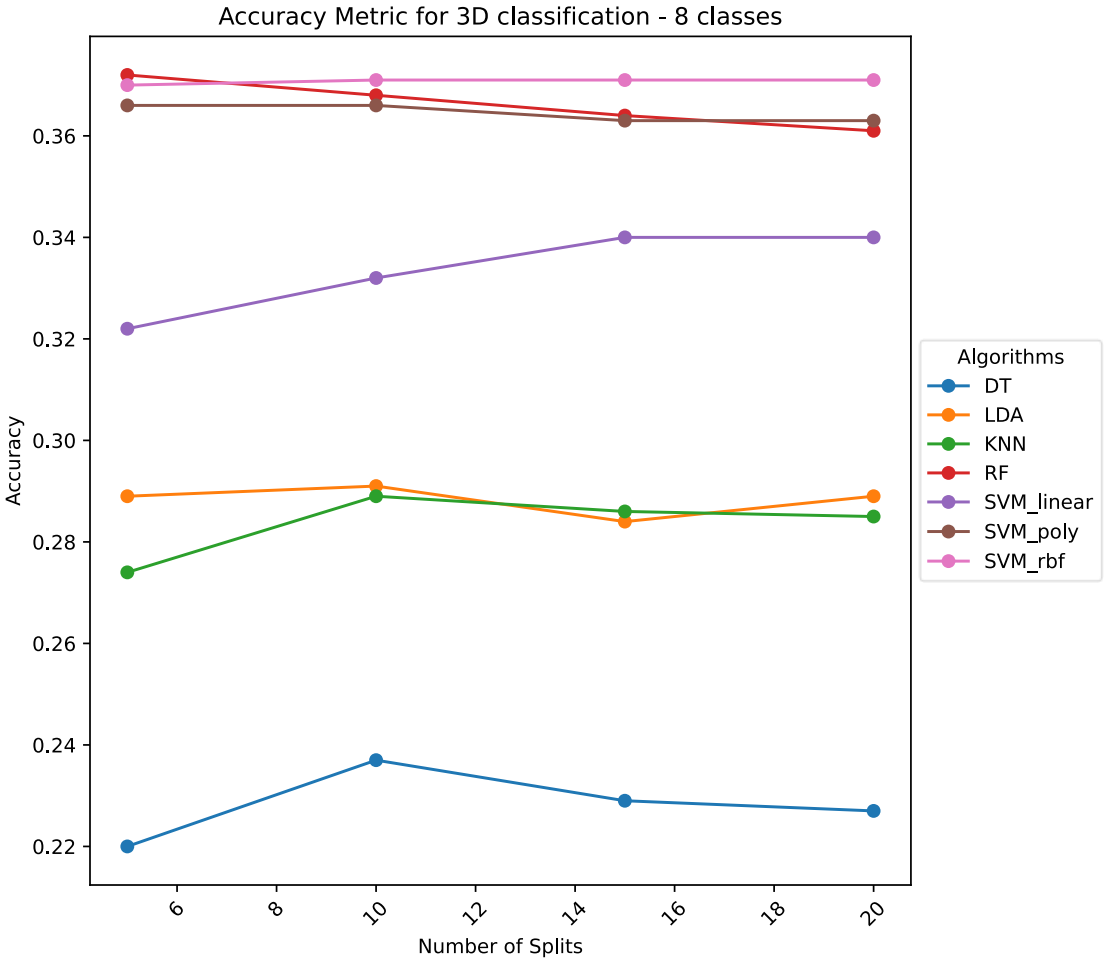

Figure 25:*DEAP time-domain Accuracy vs Nr of k-folds for 3D model*

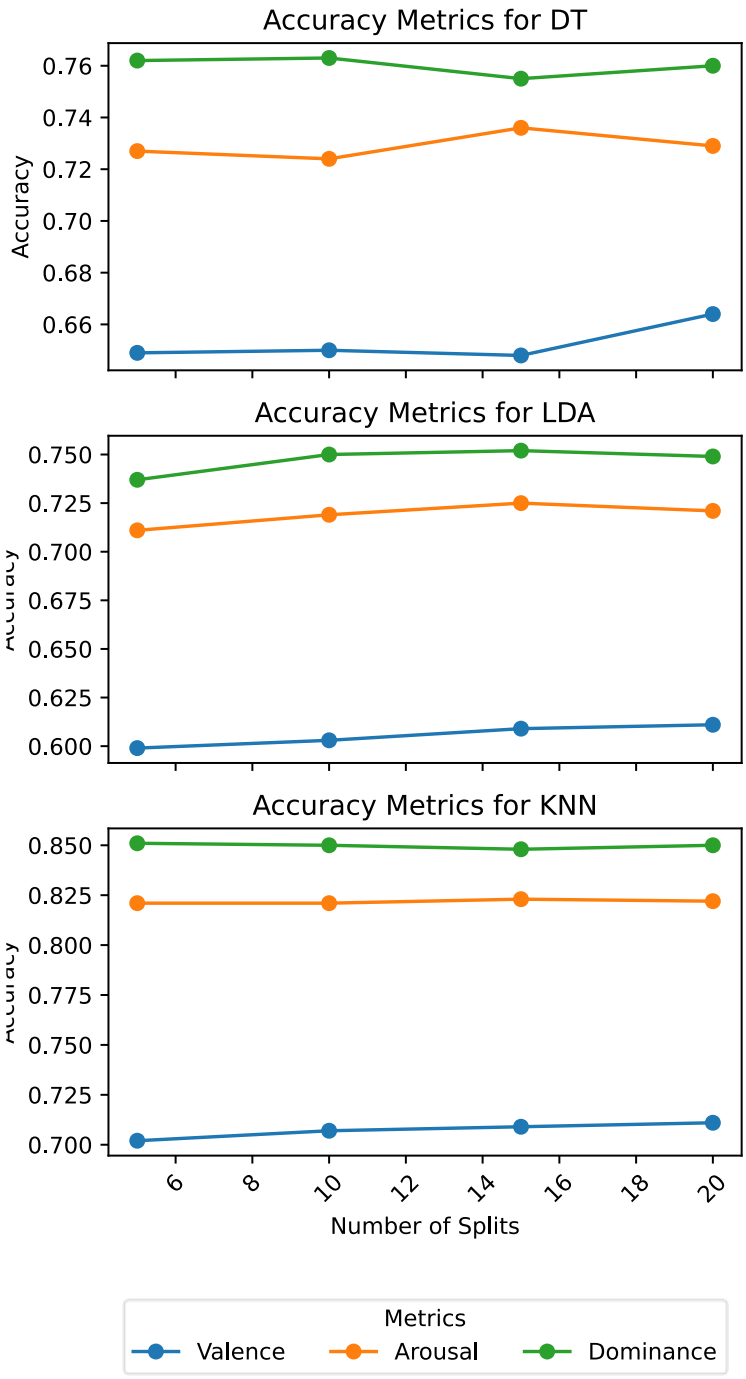

Figure 26:*DEAP frequency-domain Accuracy vs Nr of k folds*

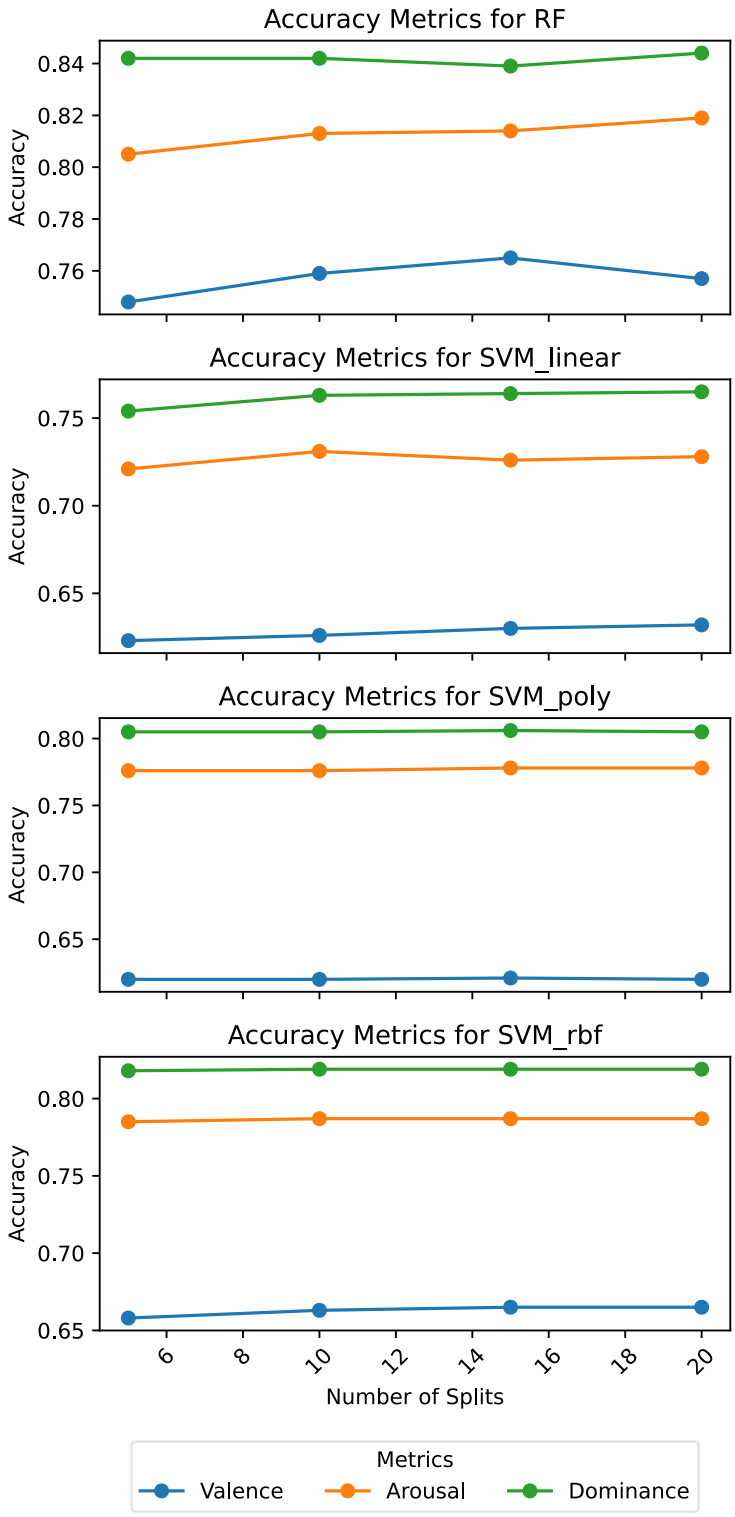

Figure 27:*DEAP frequency-domain Accuracy vs Nr of k folds*

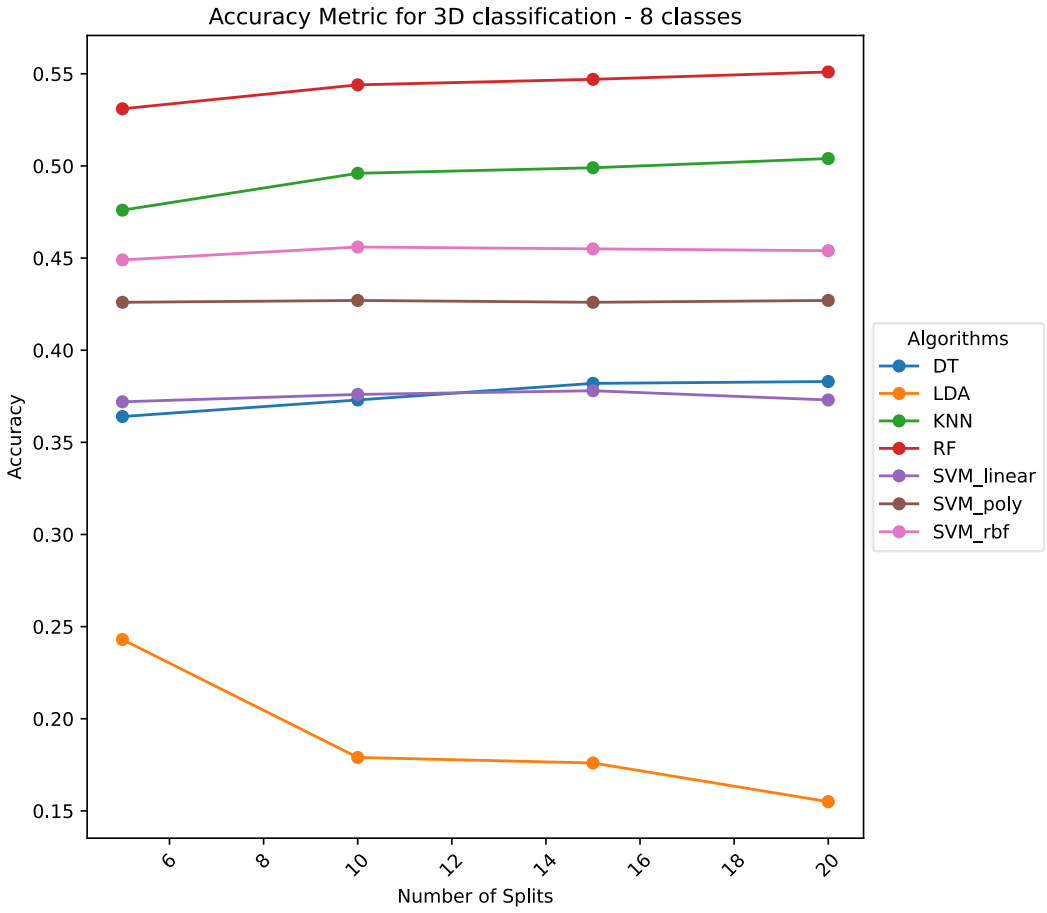

Figure 28:*DEAP frequency-domain Accuracy vs Nr of k folds for 3D model*

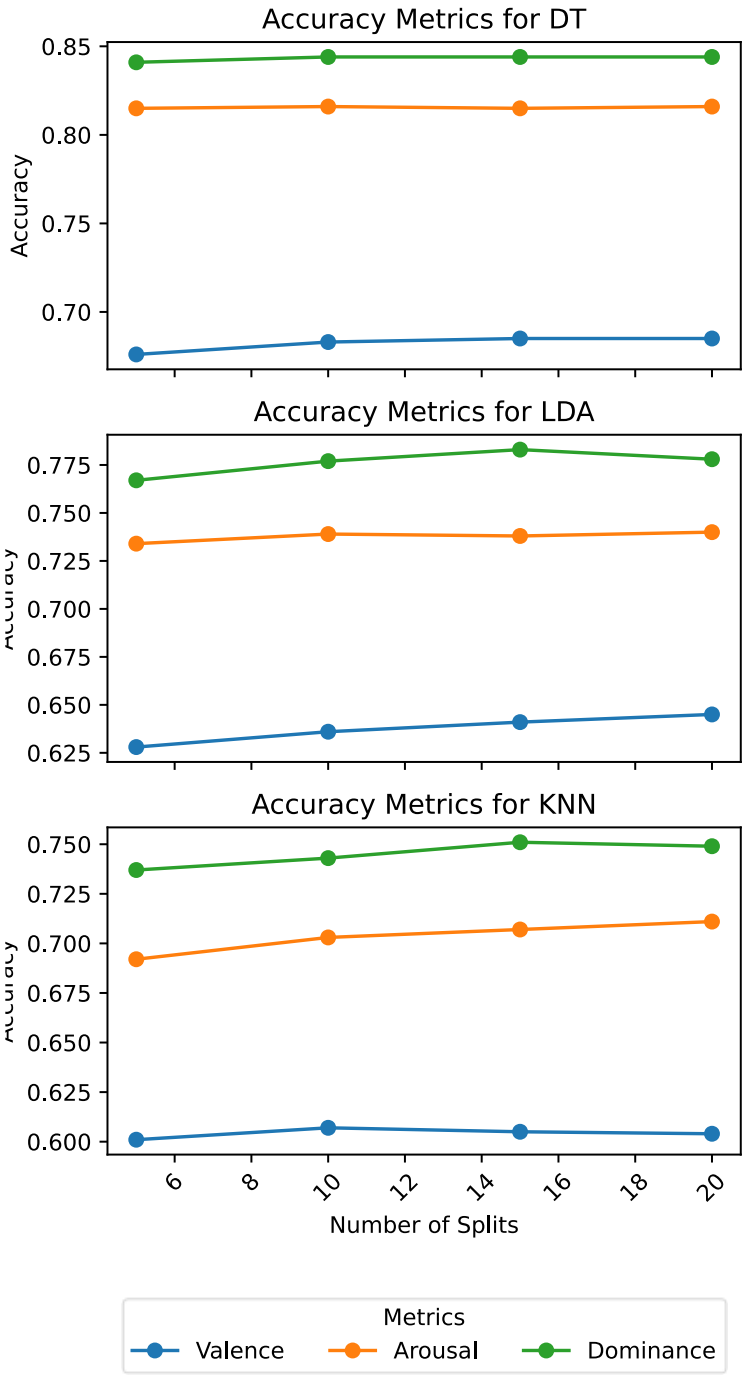

Figure 29:*DREAMER time-domain Accuracy vs Nr of k folds*

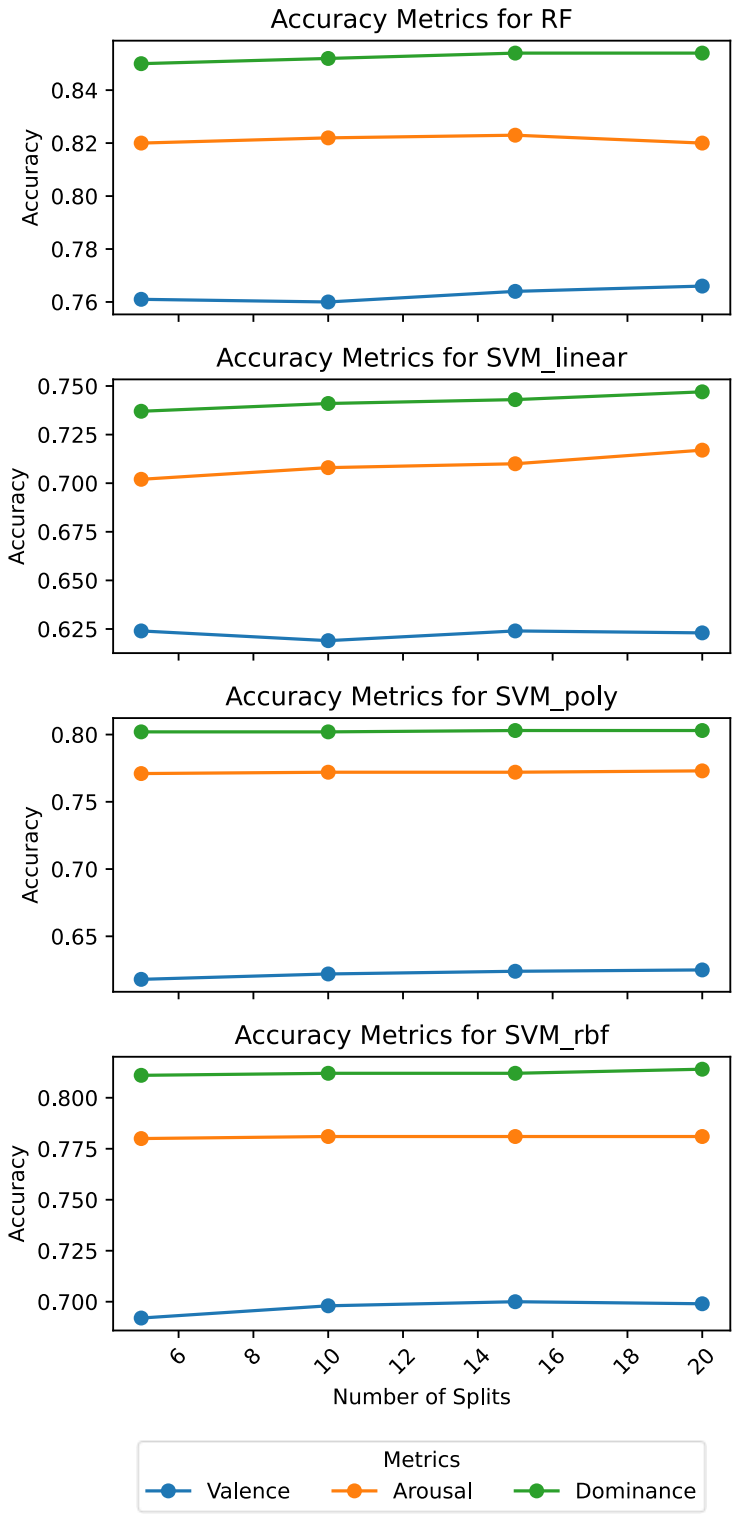

Figure 30:*DREAMER time -domain Accuracy vs Nr of k folds*

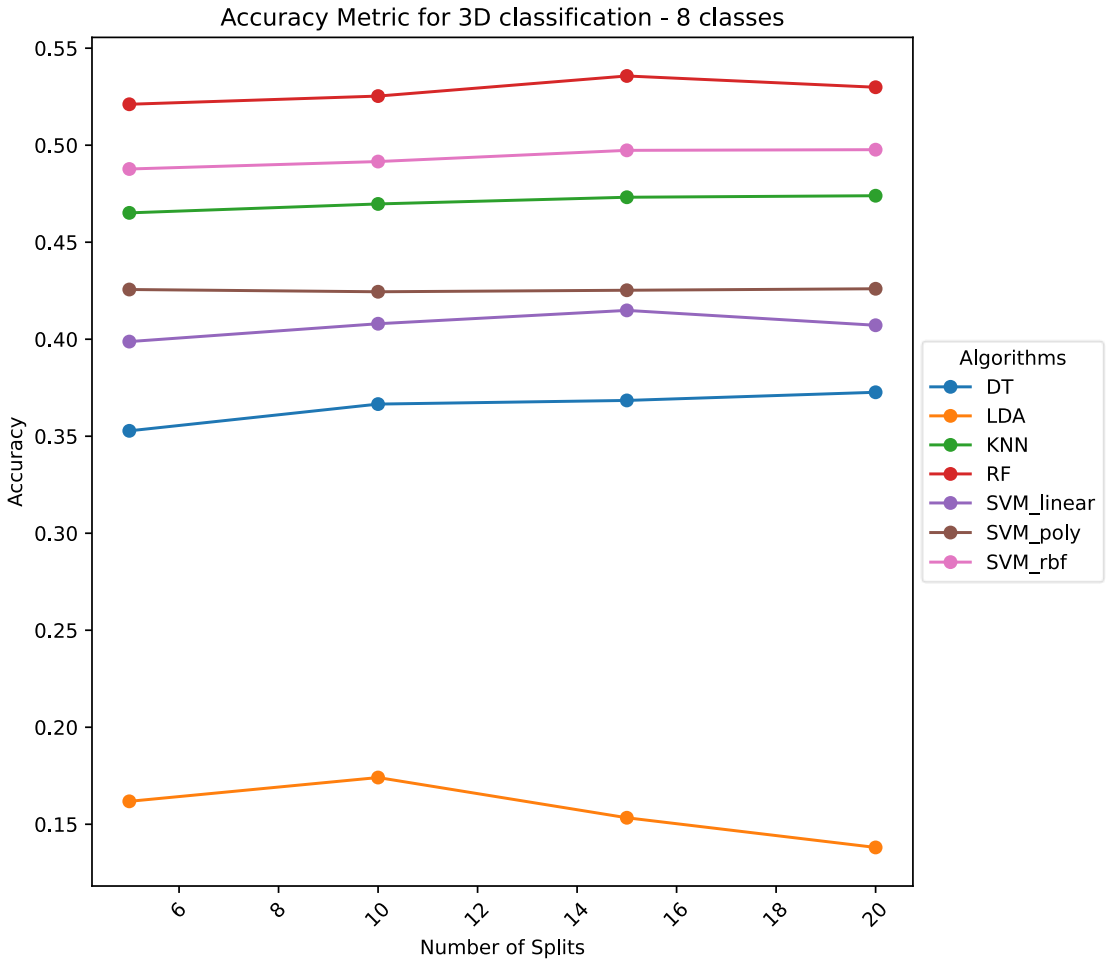

Figure 31:*DREAMER time-domain Accuracy vs Nr of k folds for 3D model*

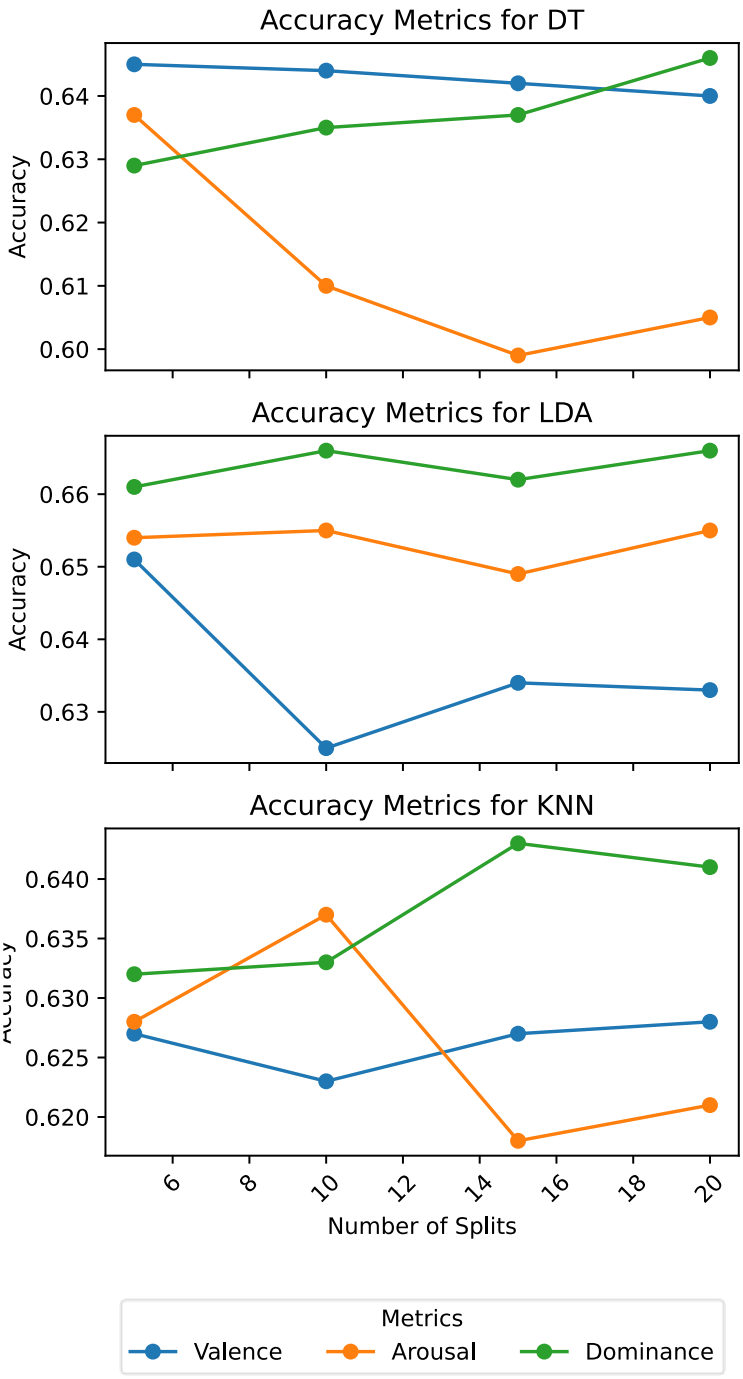

Figure 32:*DREAMER frequency-domain Accuracy vs Nr of k folds*

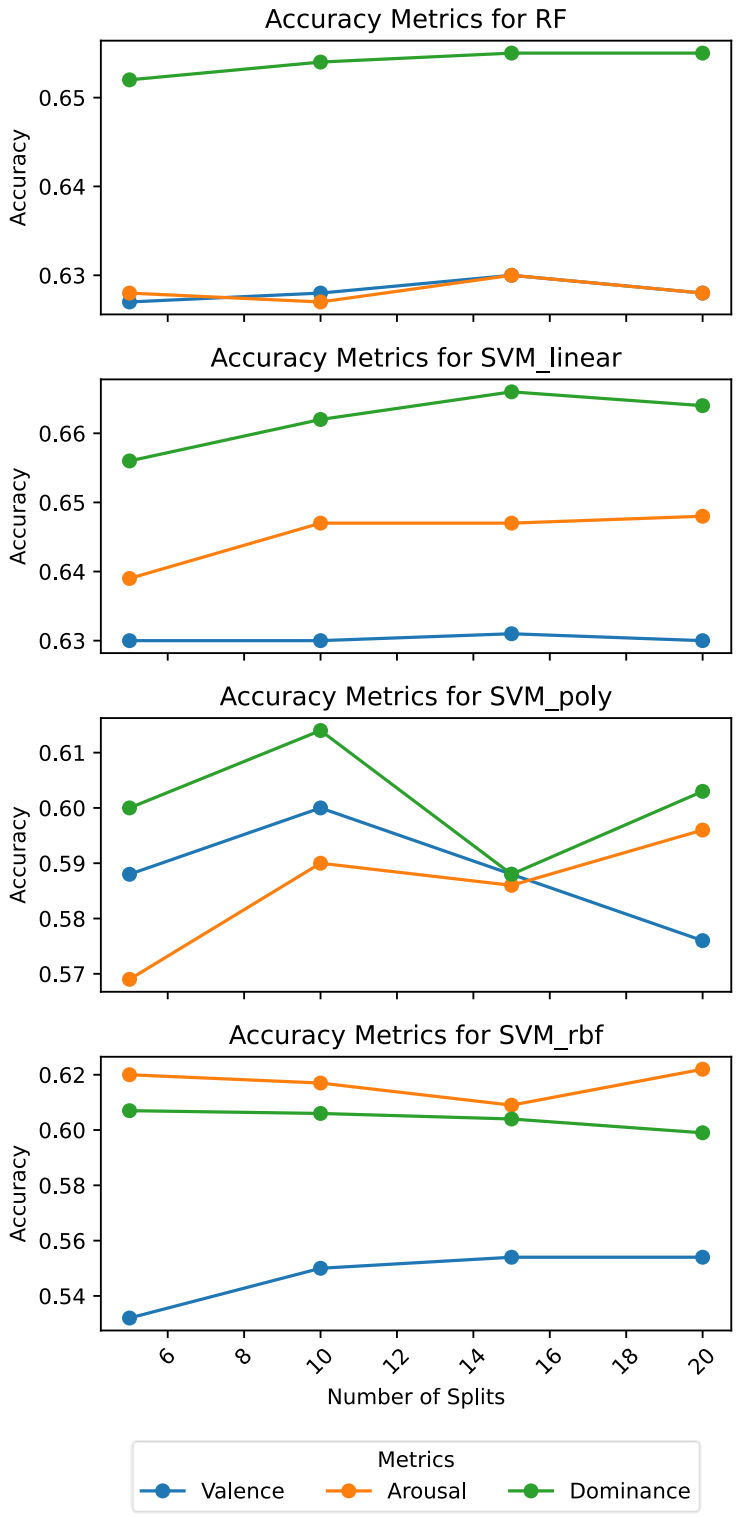

Figure 33:*DREAMER frequency-domain Accuracy vs Nr of k folds*

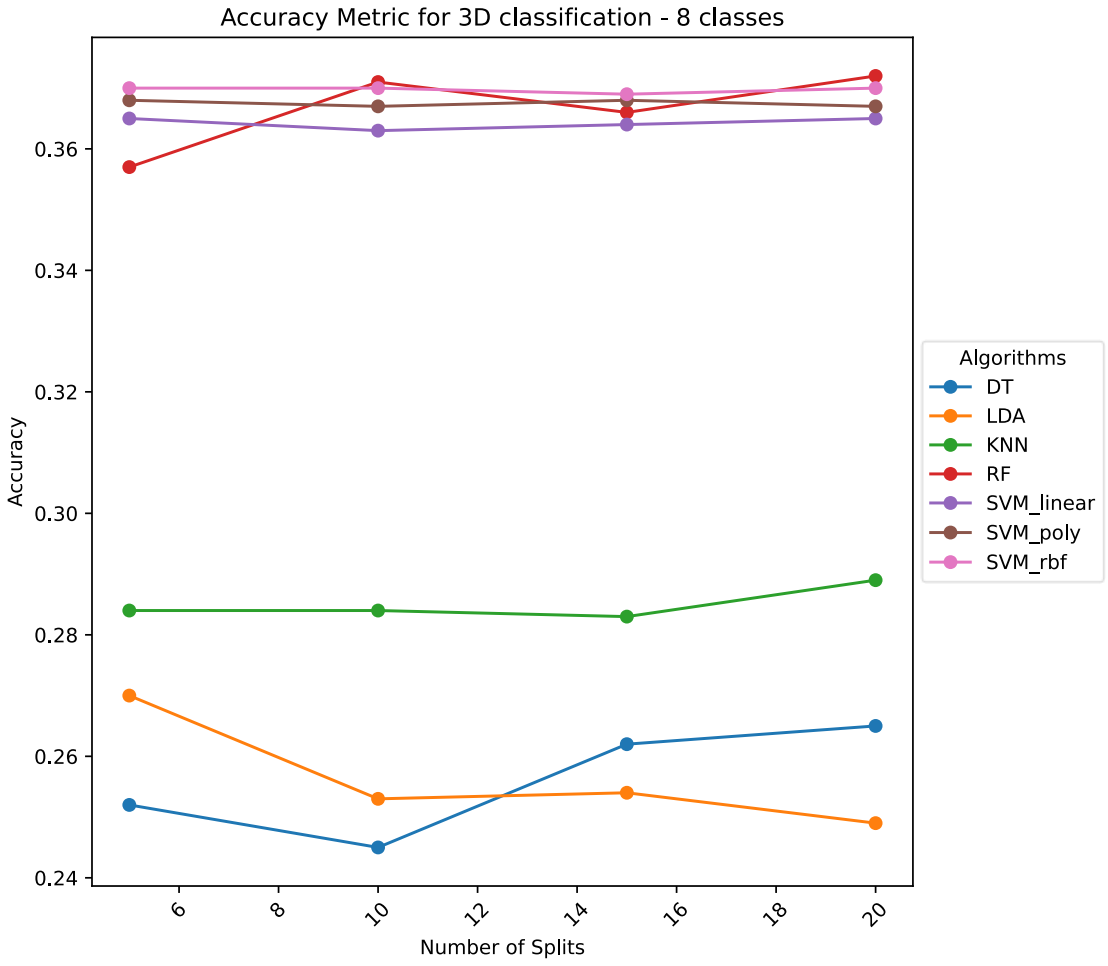

Figure 34:*DREAMER frequency-domain Accuracy vs NR of k folds for 3D model*

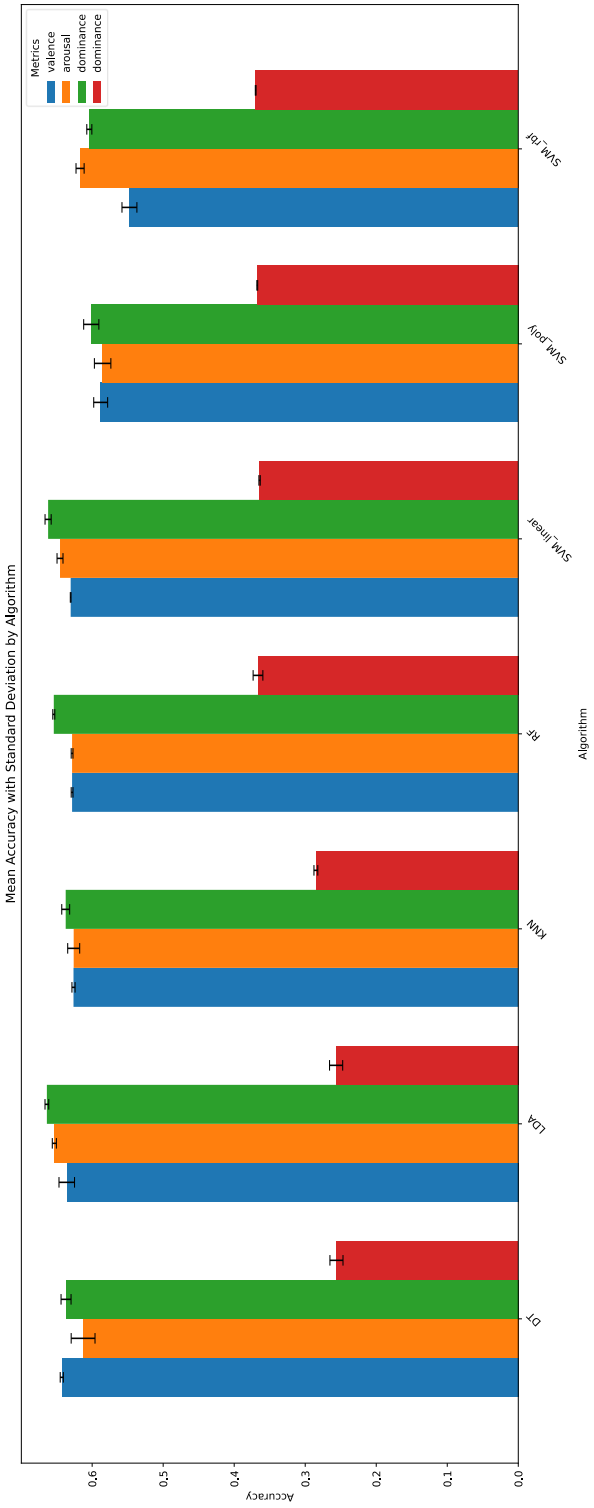

Figure 35:*DEAP time-domain ML results (acc)*

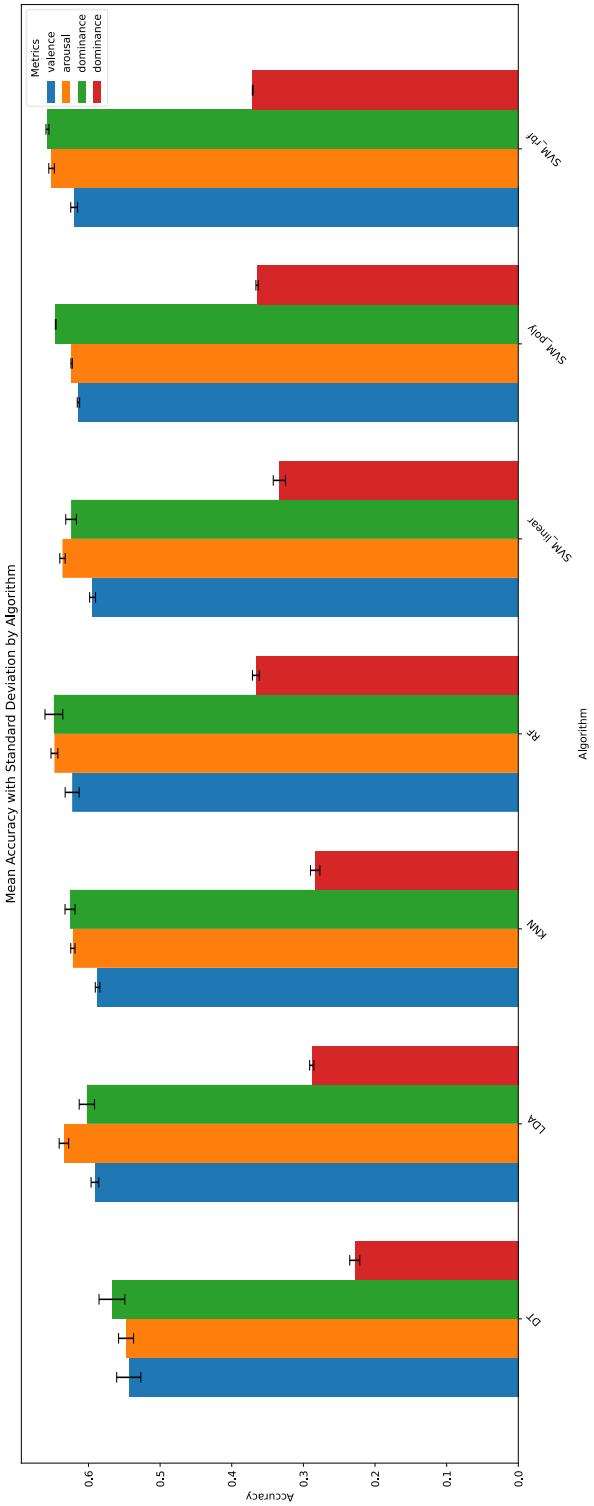

Figure 36:*DEAP frequency-domain ML results (acc)*

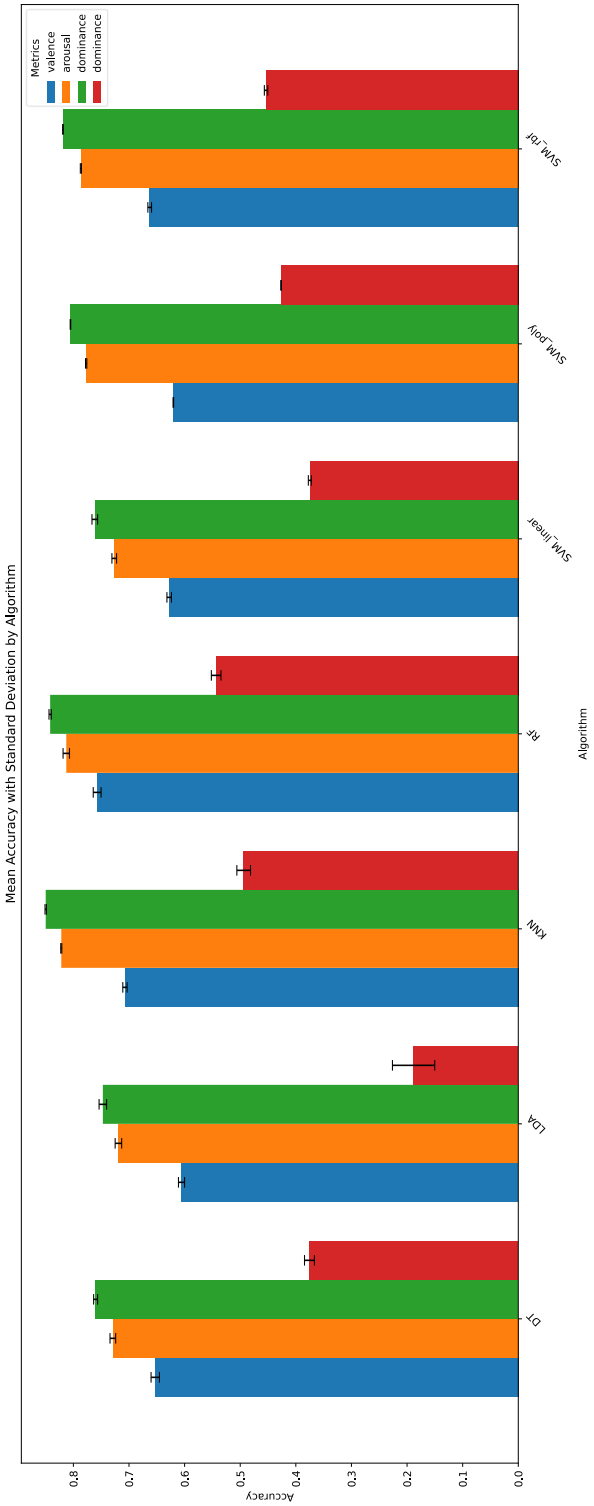

Figure 37:*DREAMER time-domain ML results (acc)*

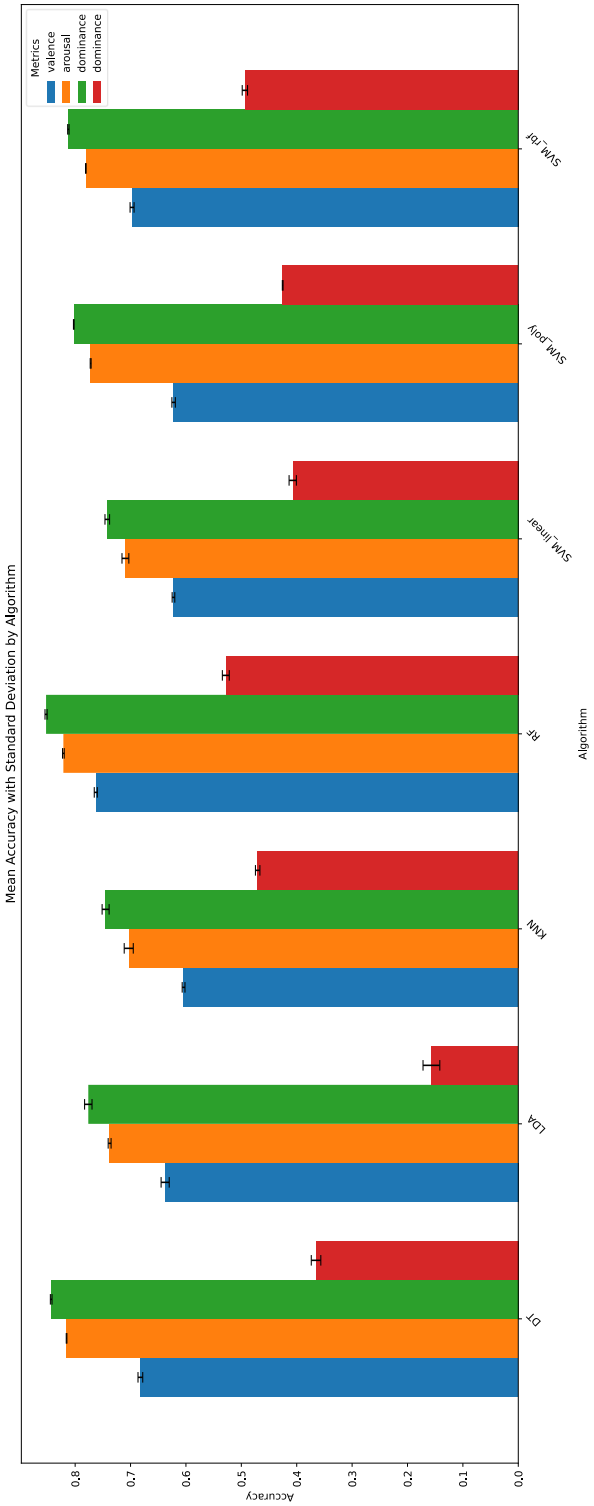

Figure 38:*DREAMER frequency-domain ML results (acc)*

The following figures (*Figure 39*, *Figure 40*, *Figure 41*, *Figure 42*) showcase all the implementations done with ML algorithms and their respective accuracy for each of the binary classifications and the 3D model multi-class one. *Figure 44* and *Figure 43* show that the models trained on frequency-domain dataset slightly outperformed those in time-frequency domain for both DEAP and DREAMER.

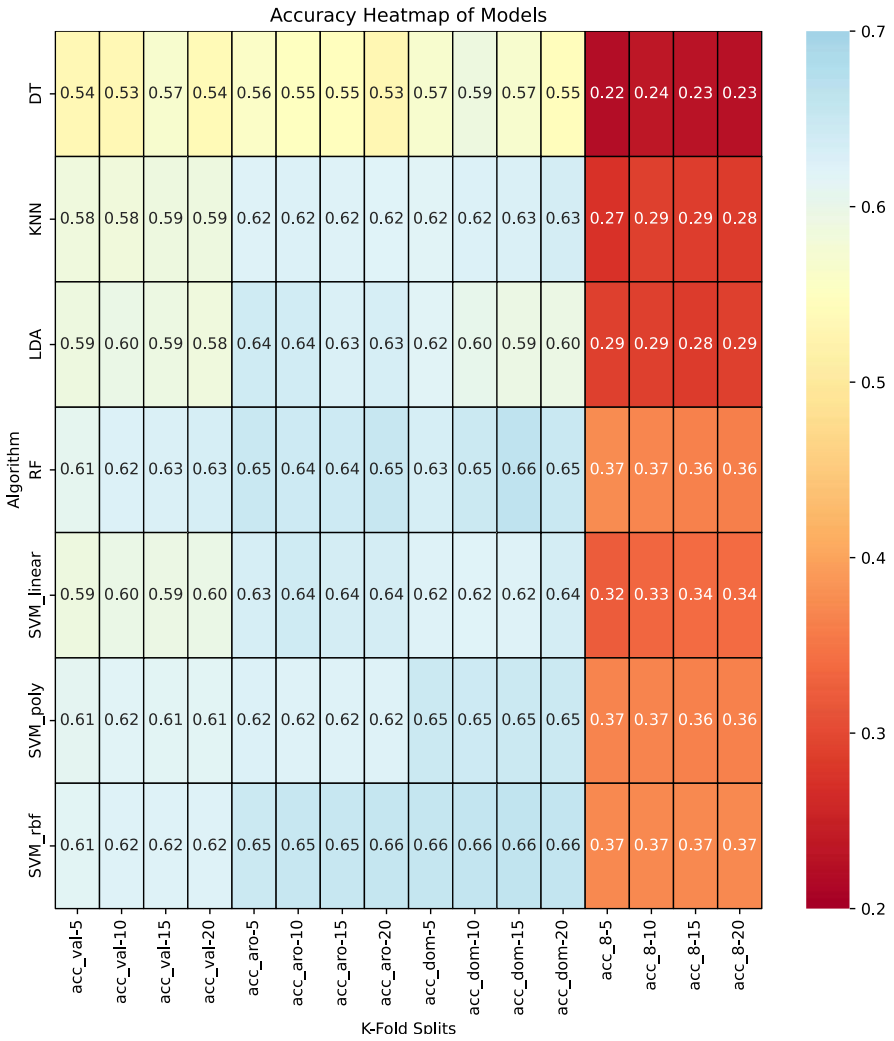

Figure 39:*DEAP time-domain heatmap of implementations (Algorithm and Accuracy)*

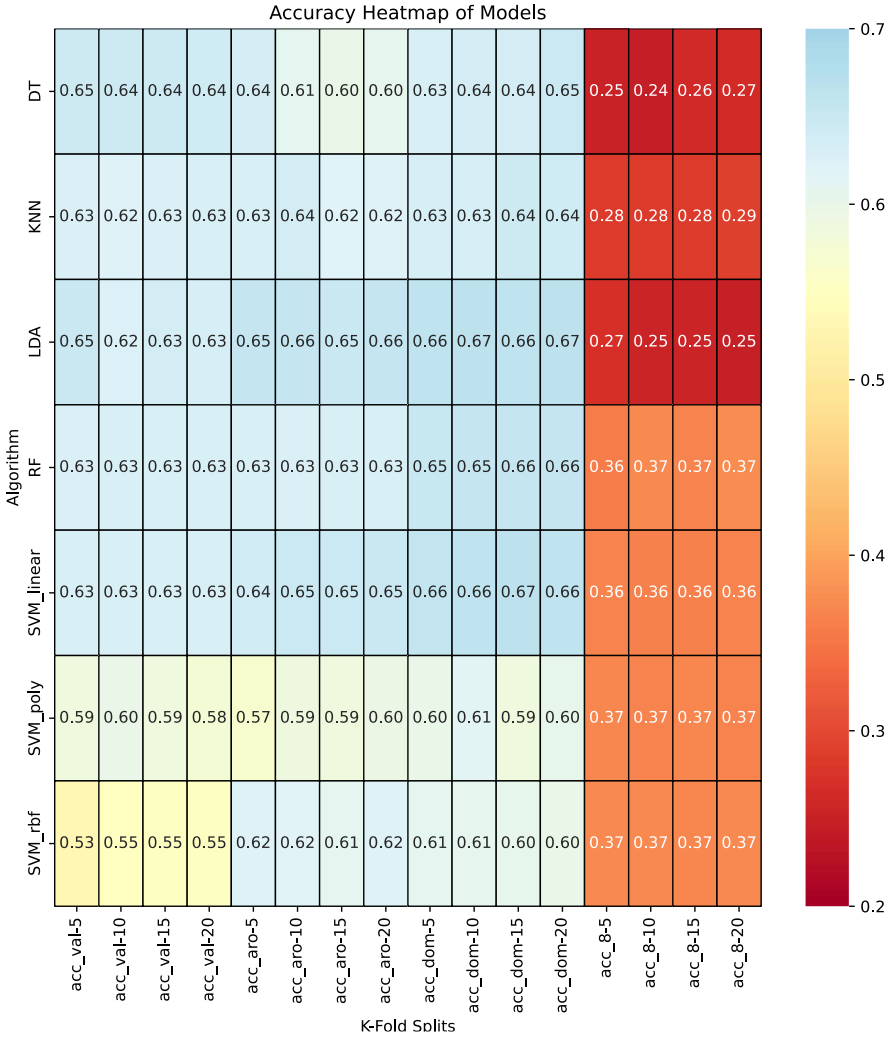

Figure 40:*DEAP frequency-domain heatmap of implementations (Algorithm and Accuracy)*

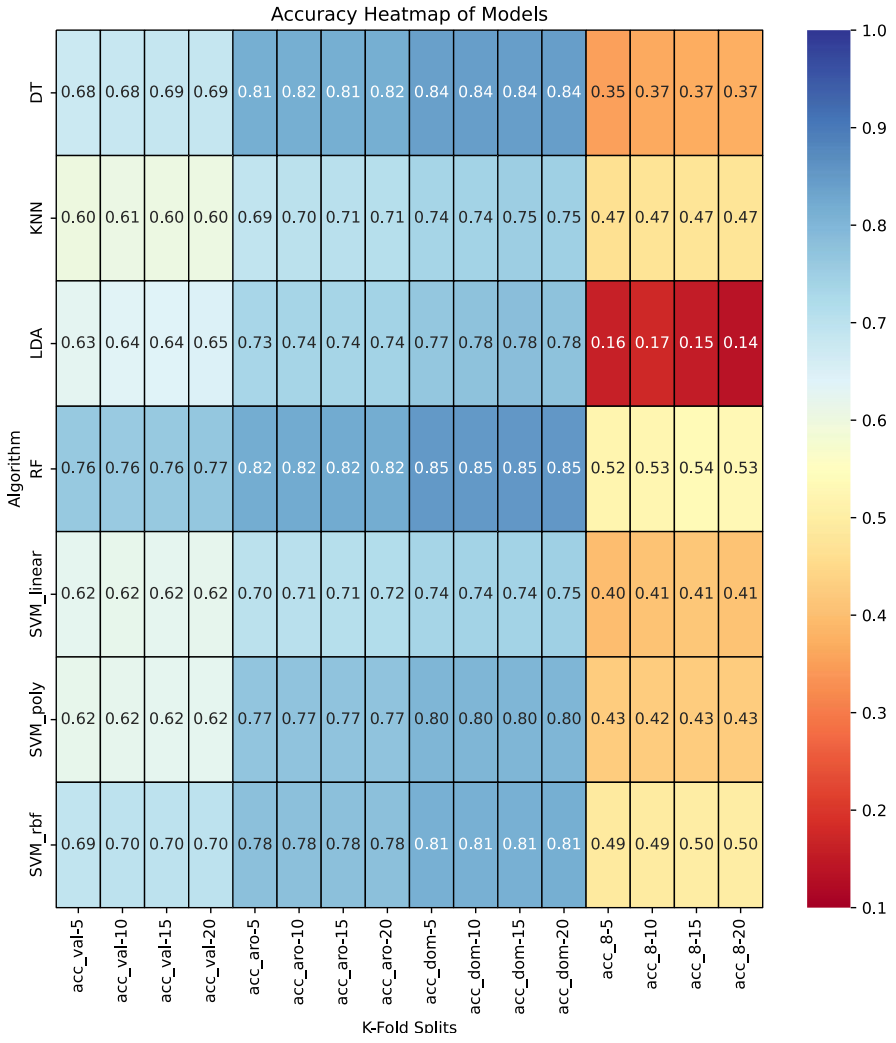

Figure 41:*DREAMER time-domain heatmap of implementations (Algorithm and Accuracy)*

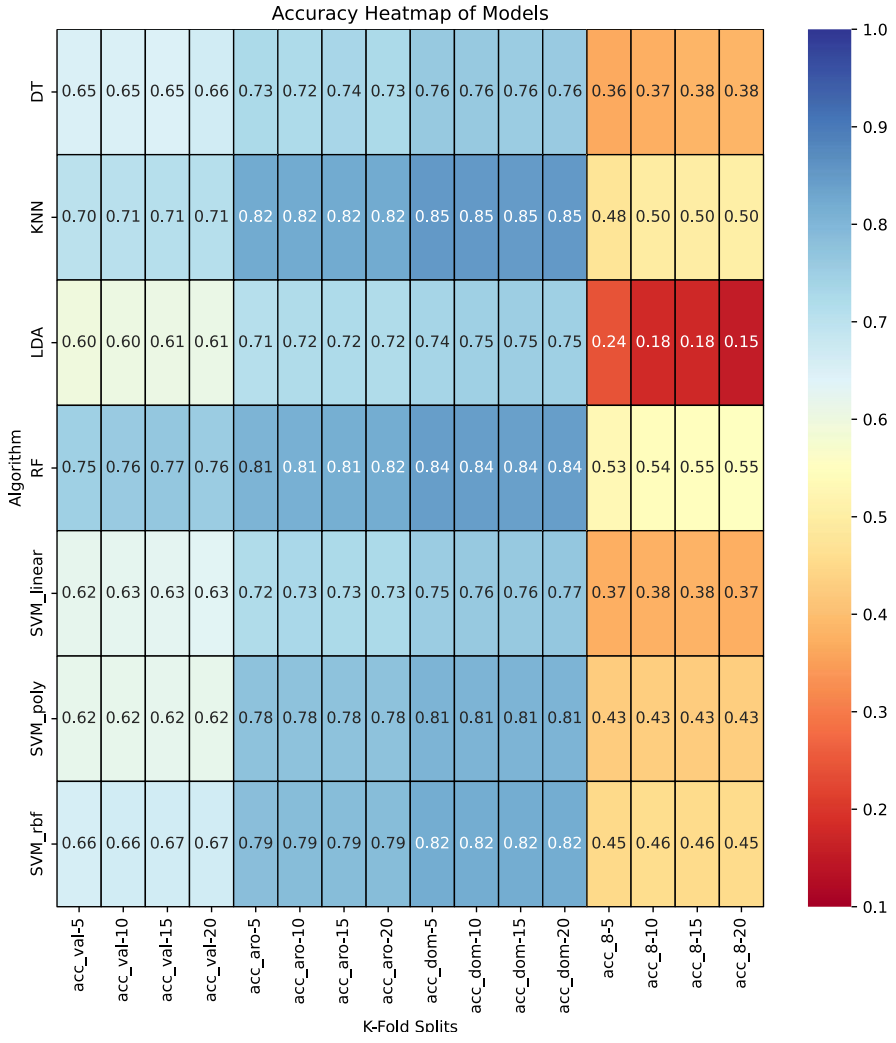

Figure 42:*DREAMER frequency-domain heatmap of implementations (Algorithm and Accuracy)*

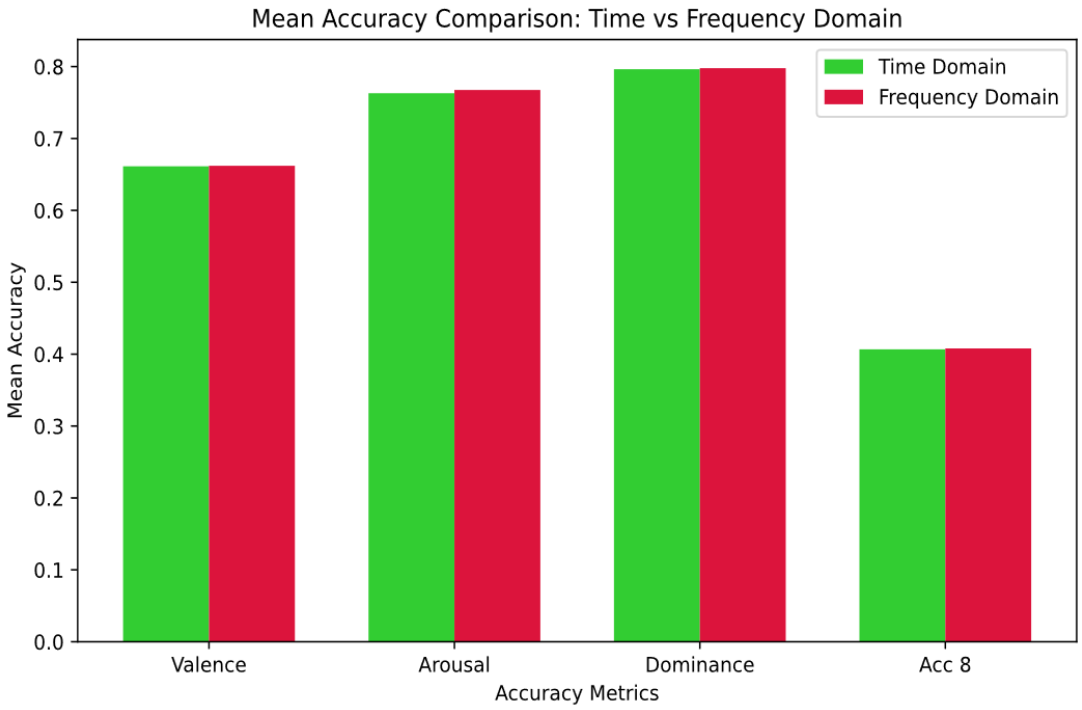

Figure 44:*DREAMER ML results time-domain vs frequency-domain Accuracies*

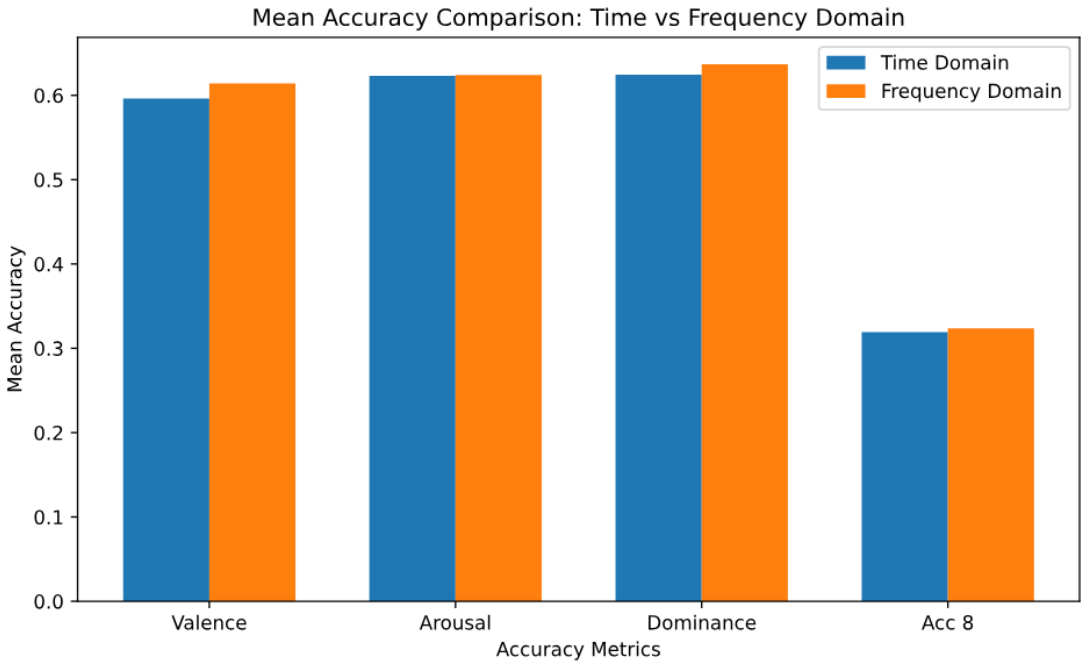

Figure 43:*DEAP ML results time-domain vs frequency-domain Accuracies*

6.2. Deep Learning Results

For the DL classifiers, the CNN models are trained on 1D time-series, which makes the training process significantly rapid, a characteristic that is important when deploying models in real-life applications. With both DEAP and DREAMER datasets the model performs well, with the latter having very reliable accuracy (over 90%) for all 4 emotion classification labels. For DEAP the additional investigation on the configuration of channels was done, and it is observed that there is a slow drop in accuracy and other metrics as the channels go from 32 to 18,14, and finally 10 for both time and frequency domain feature datasets.

The following figures (*Figure 45, Figure 47, Figure 49, Figure 51, Figure 53*) show the training-validation accuracy VS epochs during the training of the model CNN1, for different channel numbers (32, 18, 14, 10). The upper graph shows the performance of the model on time-domain features dataset, while the bottom is for frequency-domain feature dataset.

The figures (*Figure 46, Figure 48, Figure 50, Figure 52, Figure 54*) show the training-validation accuracy VS epochs during the training of the model CNN2, for different channel numbers (32, 18, 14, 10). The upper graph shows the performance of the model on time-domain features dataset, while the bottom is for frequency-domain feature dataset.

From the graphs it can be observed that while the accuracy increases slightly in CNN2, but the validation curve is very irregular and can introduce problems during the inference process.

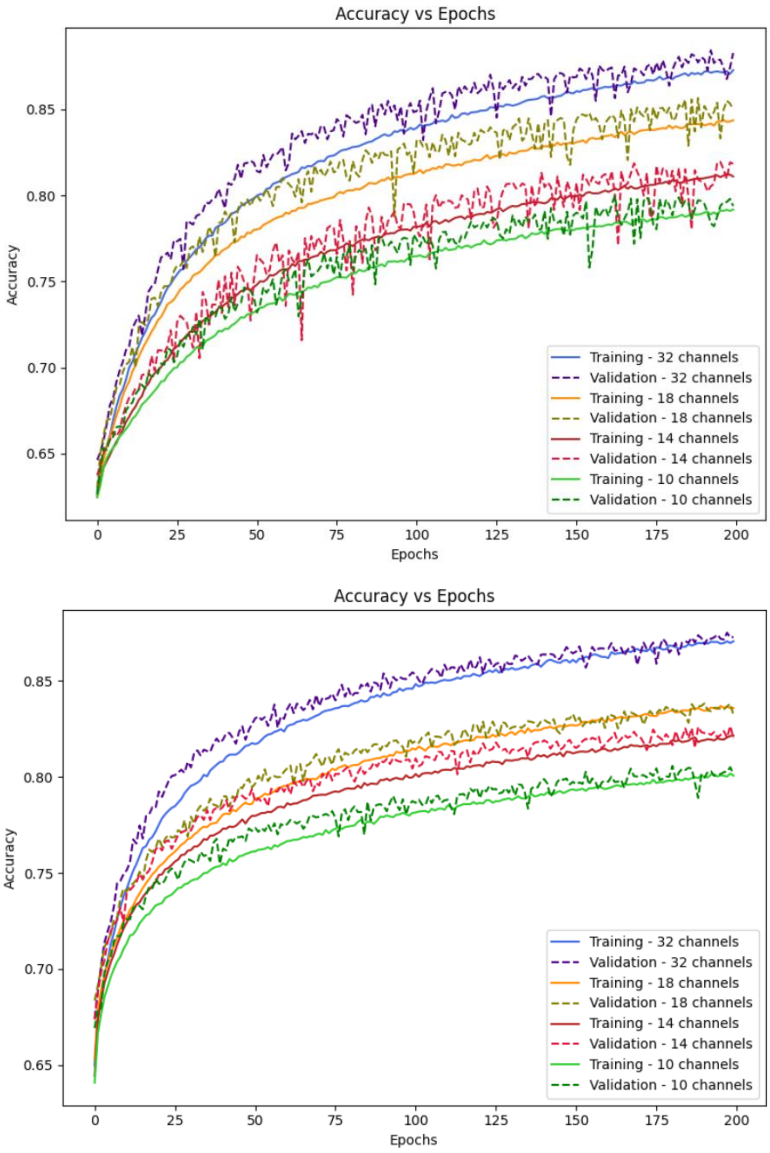

Figure 45:*CNN1 Arousal binary classification (Accuracy vs Epochs during training) top: time-domain, bottom: frequency-domain*

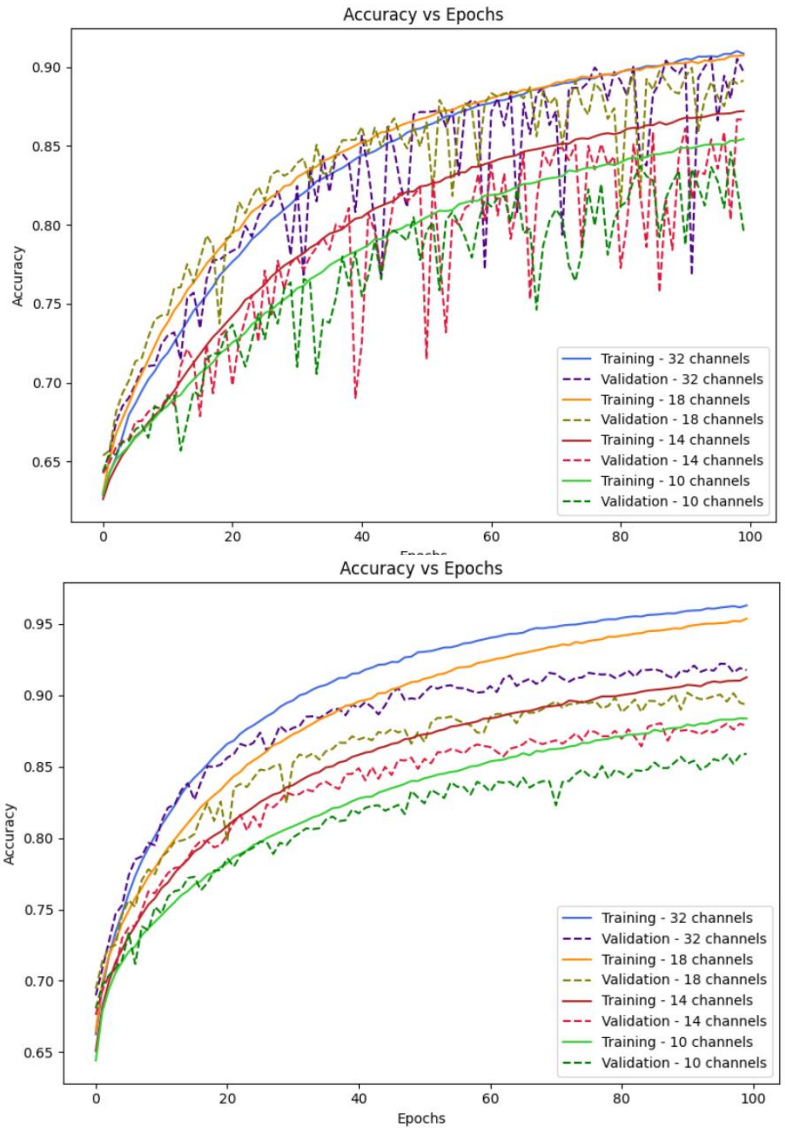

Figure 46:*CNN2 Arousal binary classification (Accuracy vs Epochs during training) top: time-domain, bottom: frequency-domain*

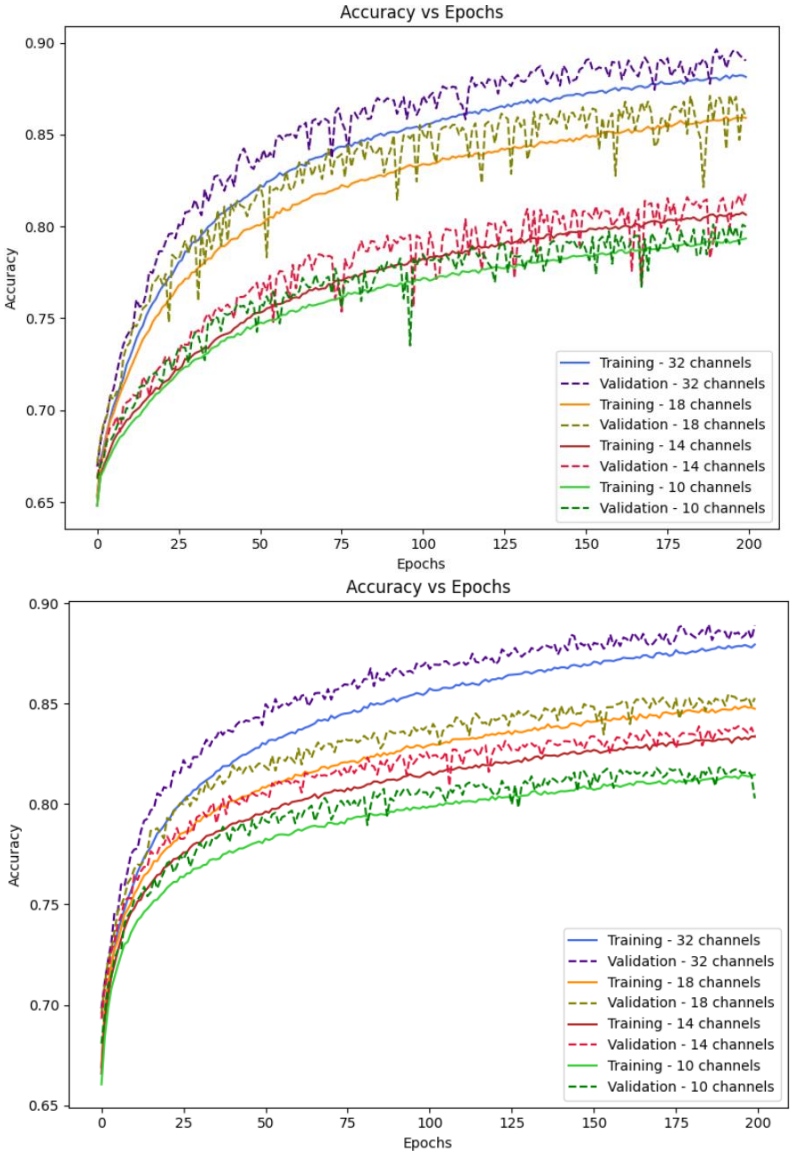

Figure 47:*CNN1 Valence binary classification (Accuracy vs Epochs during training) top: time-domain, bottom: frequency-domain*

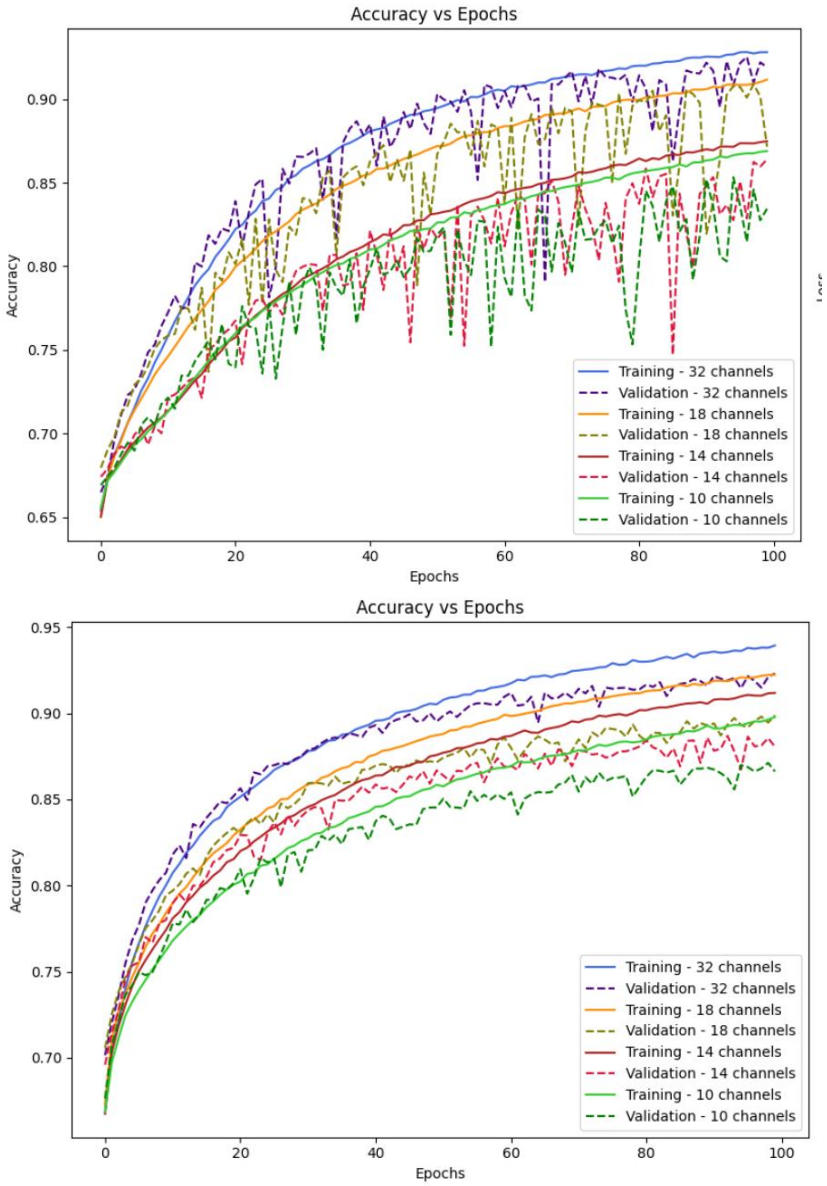

***Figure 48**:CNN2 Valence binary classification (Accuracy vs Epochs during training) top: time-domain, bottom: frequency-domain*

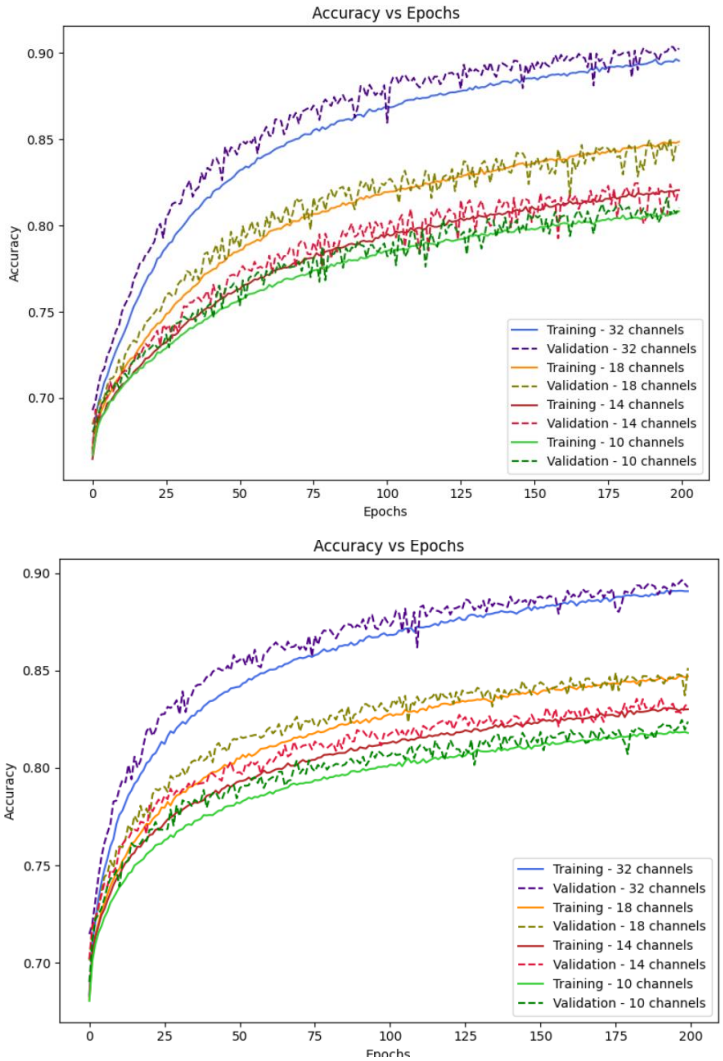

Figure 49:*CNN1 Dominance binary classification (Accuracy vs Epochs during training) top: time-domain, bottom: frequency-domain*

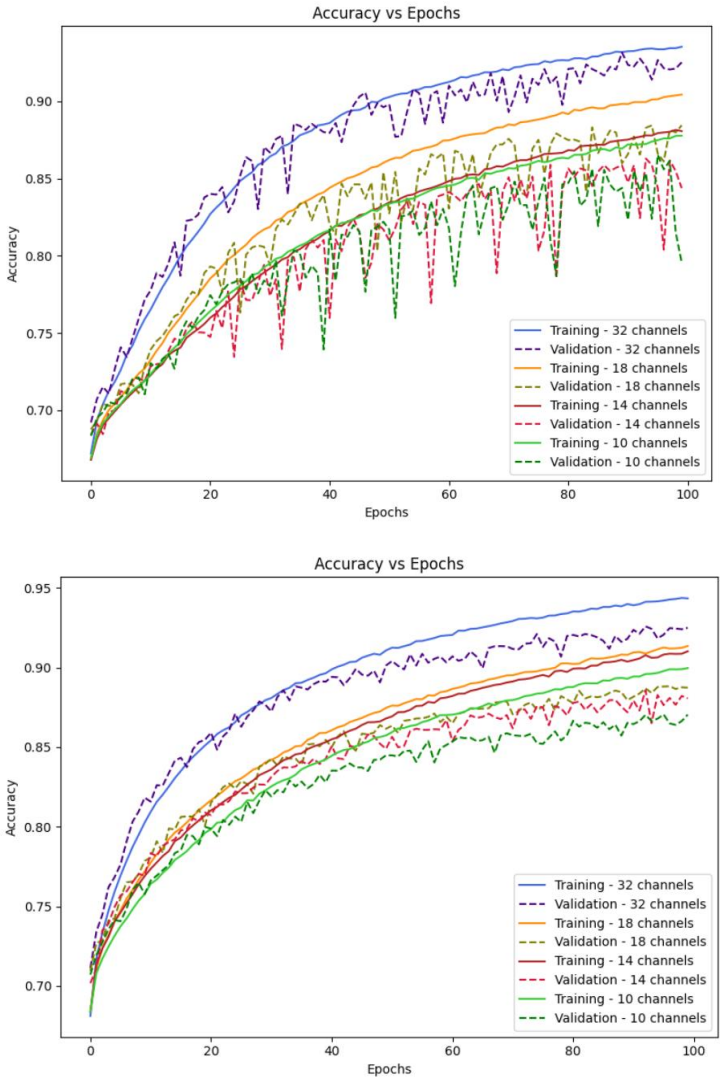

Figure 50:*CNN2 Dominance binary classification (Accuracy vs Epochs during training) top: time-domain, bottom: frequency-domain*

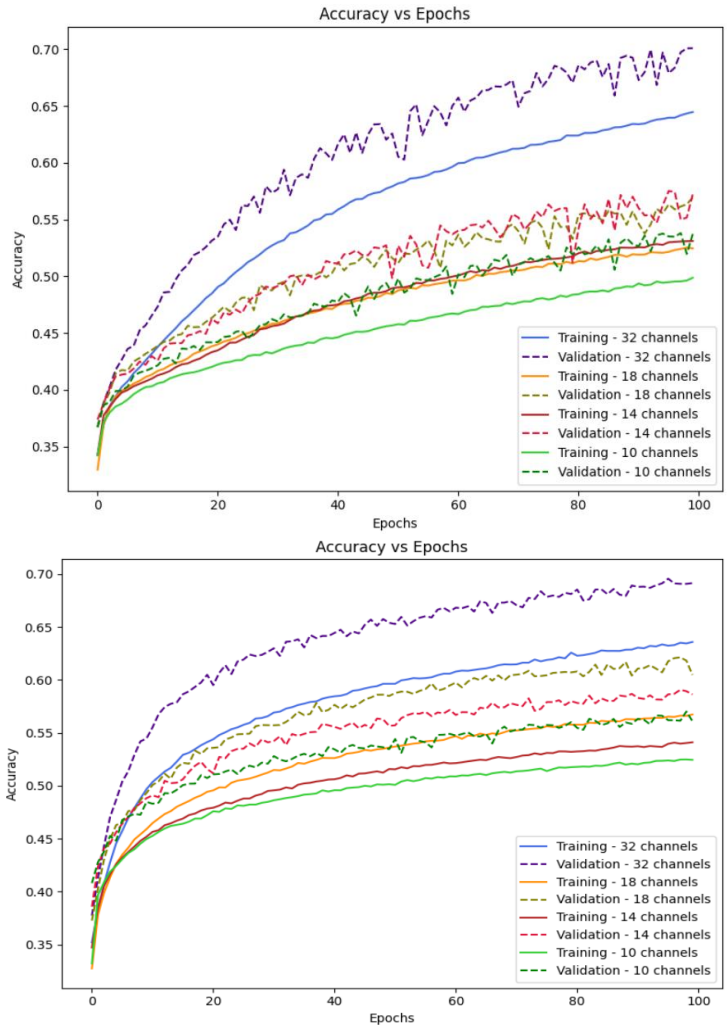

Figure 51:*CNN1 3D model classification (Accuracy vs Epochs during training) top: time-domain, bottom: frequency-domain*

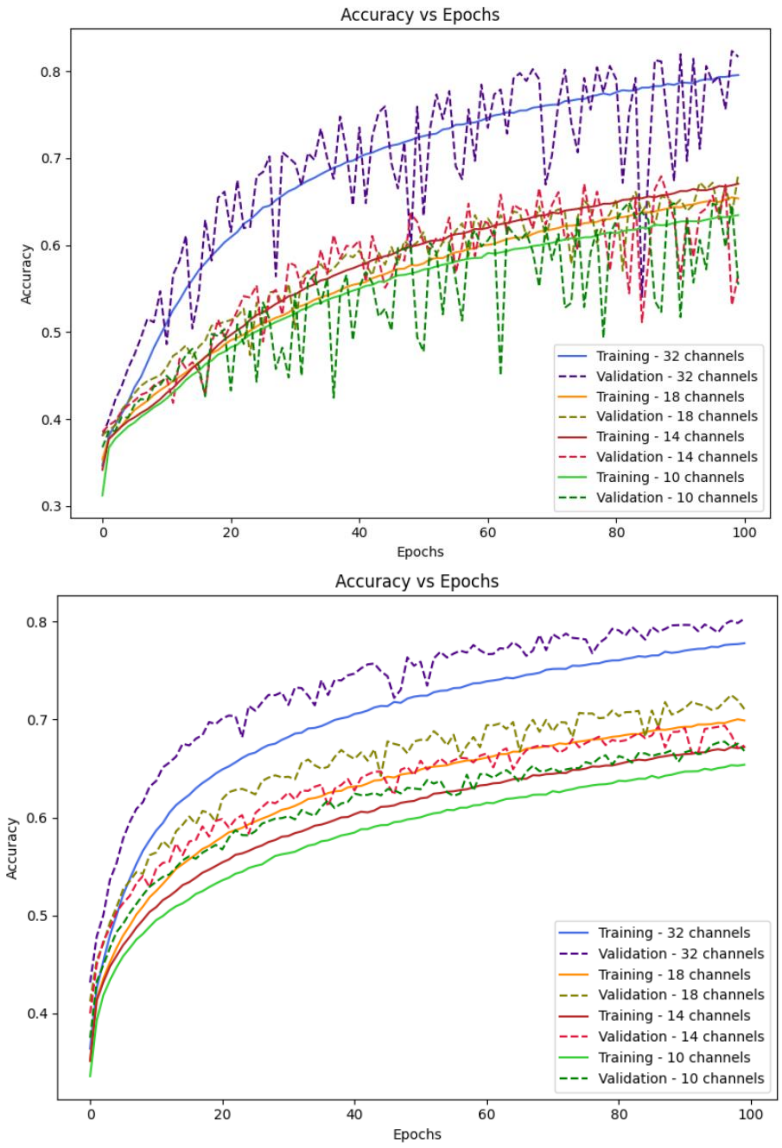

Figure 52:*CNN2 3D model classification (Accuracy vs Epochs during training) top: time-domain, bottom: frequency-domain*

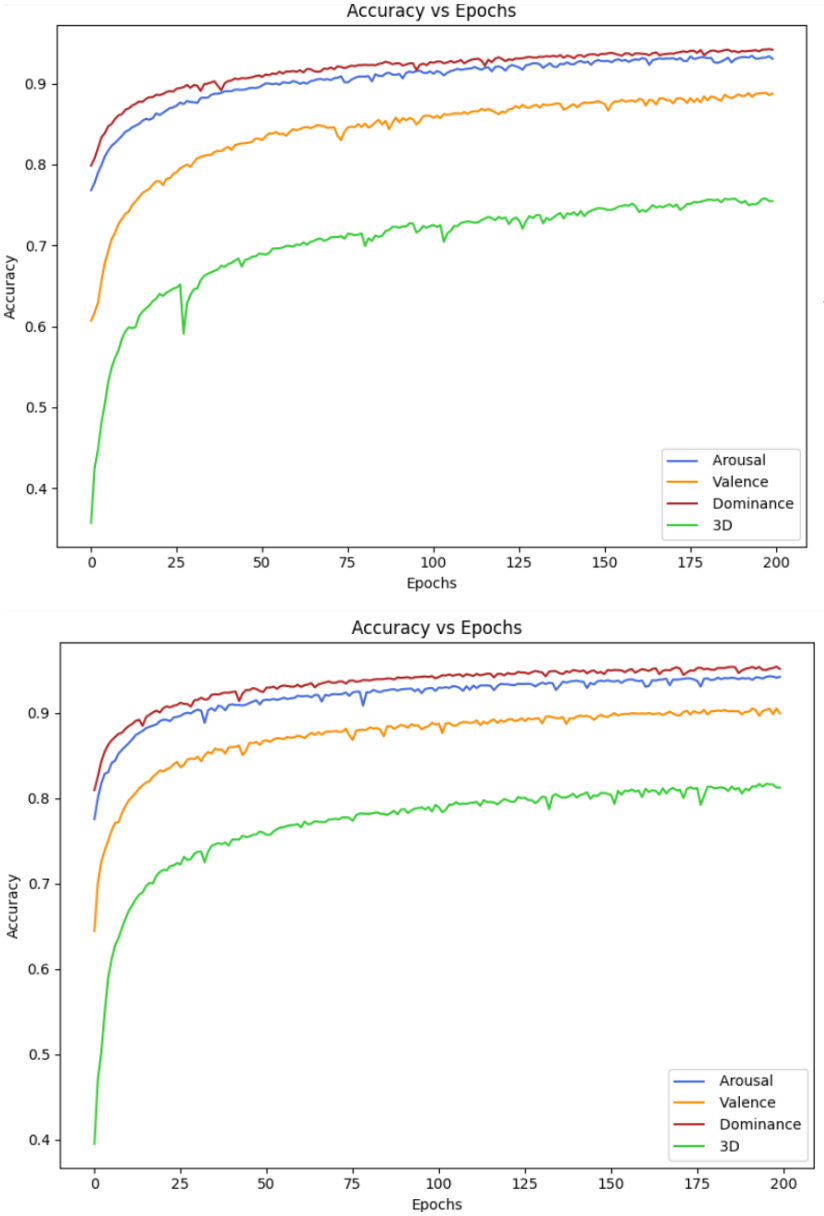

Figure 53:*CNN1 DREAMER (Accuracy vs Epochs for validation process) top: time-domain, bottom: frequency-domain*

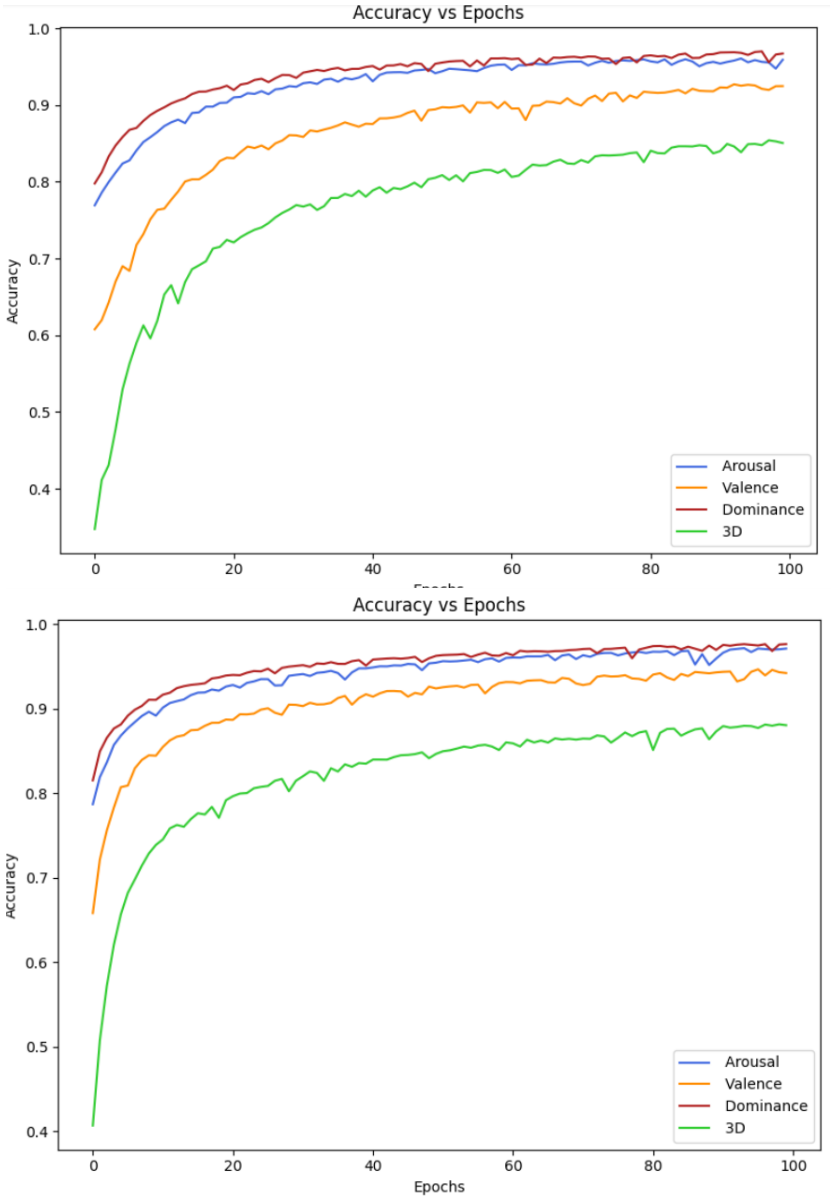

Figure 54:*CNN2 DREAMER (Accuracy vs Epochs for validation process) top: time-domain, bottom: frequency-domain*

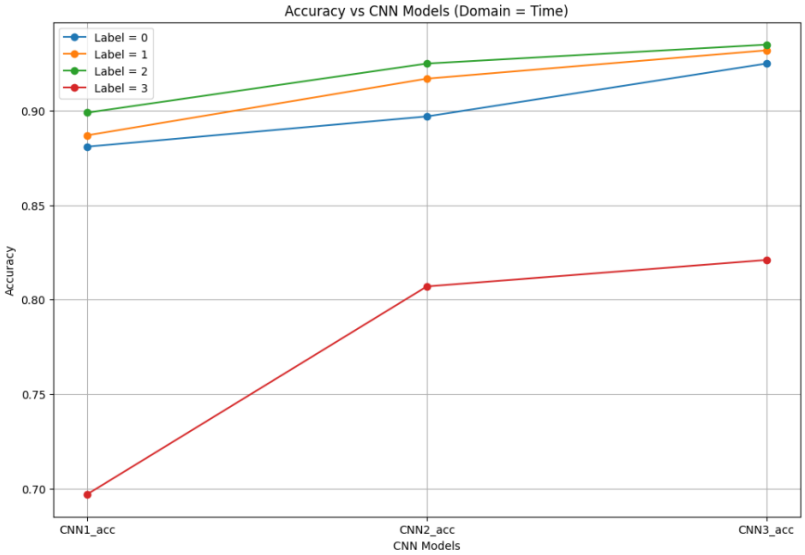

Figure 55: *DEAP CNN all models*

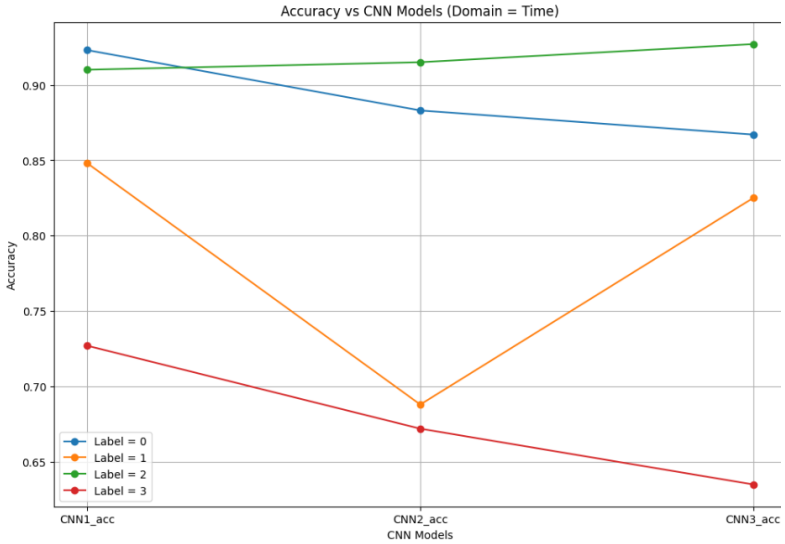

Figure 56:*DREAMER CNN all models*

For DEAP dataset the accuracy of the models increases as the complexity of the model does too, that is CNN3 has the best performance followed by CNN2 and then CNN1. In CNN1 and CNN2 it can be seen that the time domain features yield higher accuracy values for the binary classification, but for CNN3 for this type of classification frequency-domain features outperform the latter. For the DREAMER dataset the accuracy of the model decreases with CNN1 having the best performance,

followed by CNN2 and CNN3. This could be caused by the limited data or noisy recordings.

Compared to the existing models, the DEAP-CNN implementation for a subject-independent approach is satisfactory, as it is a fairly simple 1D CNN model that does not require the resources a 2D or 3D CNN approach might need.

Model	Methodology	Dataset	Binary Valence	Binary Arousal	Binary Domina
Atkinson et al. [104]	Kernel classifiers with mRMR feature	DEAP	73.14	73.06	
Li et al. [113]	CNN+RNN hybrid DL framework	DEAP	72.06	74.12	
Wang et al. [120]	EEG-specific 3D-CNN (EmotioNet)	DEAP	72.1	73.1	
Cho and Hwang [111]	2 types of 3D-CNN	DEAP	**99.11**	**99.74**	
Huang et al. [121]	BiDCNN built on 3 kinds of feature matrices	DEAP	68.14	63.94	
Liu et al. [122]	An extended domain adaptation method by introducing subject clustering	DEAP	73.9	68.8	
Bhosale et al. [123]	A few-shot adaptation method based on meta-learning	DEAP	76.46	75.81	70
Katsigiannis et al. [2]	SVM classifier	DREAMER	62.49	62.17	
CNN1(modified [119]**)**	**1D CNN**	DEAP	88.7	89.9	80.2
		DREAMER	93.6	88.1	94.4
CNN2	**1D CNN**	DEAP	91.1	92	92.5
		DREAMER	88.3	92.4	94.9
CNN3	**1D CNN**	DEAP	**92.6**	**93.7**	**94.1**
		DREAMER	**94.5**	**88.4**	**95.5**

CHAPTER 7

CONCLUSIONS

7.1. Key findings

The goal of this thesis is to implement subject-independent models for Emotion Recognition with EEG affective signals. The primary objective was developing an efficient Artificial Intelligence model that performs well with inter-subject data, but at the same time is easy to deploy and does not require excessive computations. This main objective of the thesis was successfully completed as it resulted in the implementation of three one-dimensional Convolutional Neural Networks that generalize well with inter-subject data. The results were compared to state-of-art methodologies and came out to have higher performance that a good portion of them. The few models that report higher accuracy in our case have turned out to be the complex 2D-3D CNN models that require a lot of computational power and are not fit for real-time application of Emotion Recognition.

This thesis achieves its secondary objective, that is providing a compilation of the most relevant methodologies used in the task of Emotion Recognition with EEG signals, serving as a stepping stone for further research. In addition to that, it provides a selection of robust features that can be extracted from EEG signals in order to capture the information more accurately, starting from frequency or time domain features, to the different channel configurations that can be chosen.

7.2. Limitations of the Study

Several limitations were encountered during the completion of this thesis. Despite being widely used as benchmarks for ER with EEG, the datasets DEAP and DREAMER have limited number of subjects, which is crucial when dealing with EEG signals given the inherent diversity of physiological signals as well as emotional expressions. And lastly, the complexity of the data pre-processing, feature extracting methods and computationally expensive algorithms used, hinder the time efficiency of the models so that they cannot be deployed in real-time.

7.3. Recommendations for Future Research

Given the increasing trend of ER with EEG research in the recent years, it would be important for future works to include more methodologies or processing strategies that work around the significant difference between EEG signals of two individuals. This can be done by finding robust features that appear to be shared by distinct individuals. These can be features of the time-frequency domain, features that quantify the asymmetrical activity of the brain and so on. Another suggestion would be to use domain adaptation methods, or by assembling a hybrid model from models that have been trained by similar individuals (in age, gender, culture, etc.). And most importantly investigation ways to simplify the pre-processing and implementation of the models in order to achieve real-time application of these algorithms. Lastly, the suggested approach for future research is the inclusion of fusion data, for example between EEG and ECG signals, EEG and facial expression video-recording, and so on. The combination of several mediums will provide a better understanding of the emotional processes undergoing and therefore improve upon the accuracy of the Emotion Recognition task.

7.4. Final Thoughts

To summarize, this thesis presents three 1D CNN models with subject-independent modeling that perform well in Emotion Recognition with EEG, despite not having a heavy computational load, and therefore providing a foundation on possible straightforward models that can be deployed in real-time for Human-computer interfaces. The key findings also demonstrate the great importance of choosing the right processing and feature selection methods as it can drastically decrease the complexity of the EEG data and help represent it better.

REFERENCES

[1] S. Koelstra, C. Muhl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt and I. Patras, "Deap: A database for emotion analysis; using physiological signals," *IEEE transactions on affective computing,* vol. 3, p. 18–31, 2011.

[2] S. Katsigiannis and N. Ramzan, "DREAMER: A database for emotion recognition through EEG and ECG signals from wireless low-cost off-the-shelf devices," *IEEE journal of biomedical and health informatics,* vol. 22, p. 98–107, 2017.

[3] C. Yu and M. Wang, "Survey of emotion recognition methods using EEG information," *Cognitive Robotics,* vol. 2, p. 132–146, 2022.

[4] R. Yuvaraj, P. Thagavel, J. Thomas, J. Fogarty and F. Ali, "Comprehensive Analysis of Feature Extraction Methods for Emotion Recognition from Multichannel EEG Recordings," *Sensors,* vol. 23, p. 915, 2023.

[5] J. X. Chen, P. W. Zhang, Z. J. Mao, Y. F. Huang, D. M. Jiang and Y. N. Zhang, "Accurate EEG-based emotion recognition on combined features using deep convolutional neural networks," *IEEE Access,* vol. 7, p. 44317–44328, 2019.

[6] P. Ekman, "Are there basic emotions?," 1992.

[7] P. C. Ellsworth and K. R. Scherer, "Appraisal processes in emotion," 2003.

[8] C. E. Izard, "Basic emotions, natural kinds, emotion schemas, and a new paradigm," *Perspectives on psychological science,* vol. 2, p. 260–280, 2007.

[9] R. Plutchik, "The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice," *American scientist,* vol. 89, p. 344–350, 2001.

[10] J. A. Russell, "Core affect and the psychological construction of emotion.," *Psychological review,* vol. 110, p. 145, 2003.

[11] K. R. Scherer and others, "On the nature and function of emotion: A component process approach," *Approaches to emotion,* vol. 2293, p. 31, 1984.

[12] C. Darwin, *On the origin of species, 1859,* Routledge London, 2016.

[13] C. Darwin, "The expression of the emotions in man and animals (1872)," *The Portable Darwin,* p. 364–393, 1993.

[14] W. James, "What is emotion? 1884.," 1948.

[15] W. B. Cannon, "The James-Lange theory of emotions: a critical examination and an alternative theory," *The American journal of psychology,* vol. 100, p. 567–586, 1987.

[16] H. Wanrou, G. Huang, L. Li, Z. Li, Z. Zhang and Z. Liang, "Video-triggered EEG-emotion public databases and current methods: A survey," *Brain Science Advances,* vol. 6, pp. 255-287, September 2020.

[17] P. Ekman, W. Friesen, M. O'Sullivan, A. Chan, I. Diacoyanni-Tarlatzis, K. Heider, R. Krause, W. LeCompte, T. Pitcairn, P. Ricci Bitti, K. Scherer, M. Tomita and A. Tzavaras, "Universals and Cultural Differences in the Judgments of Facial Expressions of Emotion," *Journal of Personality and Social Psychology,* vol. 53, pp. 712-7, October 1987.

[18] L. Shu, J. Xie, M. Yang, Z. Li, Z. Li, D. Liao, X. Xu and X. Yang, "A Review of Emotion Recognition Using Physiological Signals," *Sensors,* vol. 18, p. 2074, June 2018.

[19] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology,* vol. 39, p. 1161–1178, 1980.

[20] A. Etkin, C. Büchel and J. J. Gross, " The neural bases of emotion regulation.," *Nat Rev Neurosci.,* vol. 16, pp. 693-700, 2015.

[21] S. M. Alarcão and M. J. Fonseca, "Emotions Recognition Using EEG Signals: A Survey," *IEEE Transactions on Affective Computing,* vol. 10, pp. 374-393, 2019.

[22] C. Kothe, S. Makeig and J. Onton, "Emotion Recognition from EEG During Self-Paced Emotional Imagery," 2013.

[23] W.-L. Zheng and B.-L. Lu, "Investigating Critical Frequency Bands and Channels for EEG-Based Emotion Recognition with Deep Neural Networks," *IEEE Transactions on Autonomous Mental Development,* vol. 7, pp. 1-1, September 2015.

[24] B. Nakisa, N. Rastgoo, D. Tjondronegoro and V. Chandran, "Evolutionary Computation Algorithms for Feature Selection of EEG-based Emotion Recognition using Mobile Sensors," *Expert Systems with Applications,* vol. 93, October 2017.

[25] J. M. T. J. Nazmi Sofian Suhaimi, "EEG-Based Emotion Recognition: A State-of-the-Art Review of Current Trends and Opportunities," *Computational Intelligence and Neuroscience,* vol. 2020, 2020.

[26] W. L. Nowinski, "Introduction to brain anatomy," *Biomechanics of the Brain,* p. 5–40, 2011.

[27] S. M. Casillo, D. D. Luy and E. Goldschmidt, "A History of the Lobes of the Brain," *World Neurosurgery,* vol. 134, p. 353–360, February 2020.

[28] E. Britannica, *amygdala,* 2024.

[29] M. Lévêque and M. Lévêque, "The neuroanatomy of emotions," *Psychosurgery: New Techniques for Brain Disorders,* p. 49–106, 2014.

[30] K. A. Lindquist, T. D. Wager, H. Kober, E. Bliss-Moreau and L. F. Barrett, "The brain basis of emotion: a meta-analytic review," *Behavioral and brain sciences,* vol. 35, p. 121–143, 2012.

[31] P. Ekman and others, "Basic emotions," *Handbook of cognition and emotion,* vol. 98, p. 16, 1999.

[32] C. E. Izard, "Forms and functions of emotions: Matters of emotion-cognition interactions," *Emotion review,* vol. 3, p. 371–378, 2011.

[33] J. Panksepp, Affective neuroscience: The foundations of human and animal emotions, Oxford university press, 2004.

[34] L. F. Barrett, "The future of psychology: Connecting mind to brain," *Perspectives on psychological science,* vol. 4, p. 326–339, 2009.

[35] S. Duncan and L. F. Barrett, "Affect is a form of cognition: A neurobiological analysis," *Cognition and emotion,* vol. 21, p. 1184–1211, 2007.

[36] L. Pessoa, "On the relationship between emotion and cognition," *Nature reviews neuroscience,* vol. 9, p. 148–158, 2008.

[37] A. R. McIntosh, "Contexts and catalysts: a resolution of the localization and integration of function in the brain," *Neuroinformatics,* vol. 2, p. 175–181, 2004.

[38] H. A. Al Maskati and A. W. Zbrożyna, "Cardiovascular and motor components of the defence reaction elicited in rats by electrical and chemical stimulation in amygdala," *Journal of the autonomic nervous system,* vol. 28, p. 127–131, 1989.

[39] M. Davis, E. A. Antoniadis, D. G. Amaral and J. T. Winslow, "Acoustic startle reflex in rhesus monkeys: a review," *Reviews in the Neurosciences,* vol. 19, p. 171–186, 2008.

[40] A. Öhman, "Of snakes and faces: An evolutionary perspective on the psychology of fear," *Scandinavian journal of psychology,* vol. 50, p. 543–552, 2009.

[41] J. M. Hitchcock and M. Davis, "Fear-potentiated startle using an auditory conditioned stimulus: effect of lesions of the amygdala," *Physiology & behavior,* vol. 39, p. 403–408, 1987.

[42] M. Davis, "The role of the amygdala in fear and anxiety," *Annual review of neuroscience,* vol. 15, p. 353–375, 1992.

[43] M. Fendt and M. S. Fanselow, "The neuroanatomical and neurochemical basis of conditioned fear," *Neuroscience & Biobehavioral Reviews,* vol. 23, p. 743–760, 1999.

[44] M. S. Fanselow and A. M. Poulos, "The neuroscience of mammalian associative learning," *Annu. Rev. Psychol.,* vol. 56, p. 207–234, 2005.

[45] J. E. LeDoux, P. Cicchetti, A. Xagoraris and L. M. Romanski, "The lateral amygdaloid nucleus: sensory interface of the amygdala in fear conditioning," *Journal of neuroscience,* vol. 10, p. 1062–1069, 1990.

[46] K. S. LaBar, J. E. LeDoux, D. D. Spencer and E. A. Phelps, "Impaired fear conditioning following unilateral temporal lobectomy in humans," *Journal of neuroscience,* vol. 15, p. 6846–6855, 1995.

[47] A. Bechara, D. Tranel, H. Damasio, R. Adolphs, C. Rockland and A. R. Damasio, "Double dissociation of conditioning and declarative knowledge

relative to the amygdala and hippocampus in humans," *Science,* vol. 269, p. 1115–1118, 1995.

[48] P. C. Holland, M. Gallagher, P. C. Holland and M. Gallagher, "Amygdala circuitry in attentional and representational processes," *Trends in cognitive sciences,* vol. 3, p. 65–73, 1999.

[49] M. Jabbi, J. Bastiaansen and C. Keysers, "A common anterior insula representation of disgust observation, experience and imagination shows divergent functional connectivity pathways," *PloS one,* vol. 3, p. e2939, 2008.

[50] B. Wicker, C. Keysers, J. Plailly, J.-P. Royet, V. Gallese and G. Rizzolatti, "Both of us disgusted in My insula: the common neural basis of seeing and feeling disgust," *Neuron,* vol. 40, p. 655–664, 2003.

[51] P. Rozin, J. Haidt and C. R. McCauley, *Disgust In Lewis M, Haviland-Jones JM, editors,* Handbook of emotions. New York: Guilford Press, 2000.

[52] V. Curtis, R. Aunger and T. Rabie, "Evidence that disgust evolved to protect from risk of disease," *Proceedings of the Royal Society of London. Series B: biological sciences,* vol. 271, p. S131–S133, 2004.

[53] R. Adolphs, D. Tranel and A. R. Damasio, "Dissociable neural systems for recognizing emotions," *Brain and cognition,* vol. 52, p. 61–69, 2003.

[54] A. J. Calder, J. Keane, F. Manes, N. Antoun and A. W. Young, "Impaired recognition and experience of disgust following brain injury," *Nature neuroscience,* vol. 3, p. 1077–1078, 2000.

[55] W. Penfield and M. E. Faulk Jr, "The insula: further observations on its function," *Brain,* vol. 78, p. 445–470, 1955.

[56] M. Tsakiris, M. D. Hesse, C. Boy, P. Haggard and G. R. Fink, "Neural signatures of body ownership: a sensory network for bodily self-consciousness," *Cerebral cortex,* vol. 17, p. 2235–2244, 2007.

[57] F. C. Murphy, I. A. N. Nimmo-Smith and A. D. Lawrence, "Functional neuroanatomy of emotions: a meta-analysis," *Cognitive, affective, & behavioral neuroscience,* vol. 3, p. 207–233, 2003.

[58] Vytal and S. Hamann, "Neuroimaging support for discrete neural correlates of basic emotions: a voxel-based meta-analysis.," *Journal of cognitive neuroscience,* vol. 22, p. 2864–2885, 2010.

[59] H. Barbas, "Anatomic organization of basoventral and mediodorsal visual recipient prefrontal regions in the rhesus monkey," *Journal of Comparative Neurology,* vol. 276, pp. 313-342.

[60] M. KRINGELBACH, "The functional neuroanatomy of the human orbitofrontal cortex: evidence from neuroimaging and neuropsychology," *Progress in Neurobiology,* vol. 72, p. 341–372, April 2004.

[61] K. L. Phan, T. Wager, S. F. Taylor and I. Liberzon, "Functional Neuroanatomy of Emotion: A Meta-Analysis of Emotion Activation Studies in PET and fMRI," *NeuroImage,* vol. 16, p. 331–348, June 2002.

[62] J. Rettinger, S. Schwarz and W. Schwarz, Electrophysiology, Springer, 2022.

[63] J. C. Strickland, "Guide to research techniques in neuroscience," *Journal of Undergraduate Neuroscience Education,* vol. 13, p. R1, 2014.

[64] M. Brienza and O. Mecarelli, "Neurophysiological basis of EEG," *Clinical electroencephalography,* p. 9–21, 2019.

[65] C. Hammond, Cellular and Molecular Neurobiology (Deluxe Edition), Academic Press, 2015.

[66] A. E. Pereda, "Electrical synapses and their functional interactions with chemical synapses," *Nature Reviews Neuroscience,* vol. 15, p. 250–263, 2014.

[67] M. D. David Valentine, *Learning EEG,* 2020.

[68] T. Kirschstein and R. Köhling, "What is the source of the EEG?," *Clinical EEG and neuroscience,* vol. 40, p. 146–149, 2009.

[69] X.-J. Wang, "Neurophysiological and Computational Principles of Cortical Rhythms in Cognition," *Physiological Reviews*, vol. 90, p. 1195–1268, July 2010.

[70] A. Kawala-Sterniuk, N. Browarska, A. Al-Bakri, M. Pelc, J. Zygarlicki, M. Sidikova, R. Martinek and E. J. Gorzelanczyk, "Summary of over Fifty Years

with Brain-Computer Interfaces—A Review," *Brain Sciences,* vol. 11, p. 43, January 2021.

[71] Y. Qin, Y. Zhang, Y. Zhang, S. Liu and X. Guo, "Application and Development of EEG Acquisition and Feedback Technology: A Review," *Biosensors,* vol. 13, p. 930, October 2023.

[72] J. A. S. P. E. R. HH, "The ten-twenty electrode system of the international federation," *Electroenceph clin Neurophysiol,* vol. 10, p. 367–380, 1958.

[73] M. Teplan, "Fundamental of EEG Measurement," *MEASUREMENT SCIENCE REVIEW,* vol. 2, January 2002.

[74] M. Chowdhury, "EEG-based assessment of emotional well-being in smart environment," 2020.

[75] G. Di Flumeri, P. Aricò, G. Borghini, N. Sciaraffa, A. Di Florio and F. Babiloni, "The dry revolution: Evaluation of three different EEG dry electrode types in terms of signal spectral features, mental states classification and usability," *Sensors,* vol. 19, p. 1365, 2019.

[76] R. Martins, S. Selberherr and F. A. Vaz, "A CMOS IC for portable EEG acquisition systems," *IEEE Transactions on Instrumentation and Measurement,* vol. 47, p. 1191–1196, 1998.

[77] D. W.-K. Ng and S. Y. Goh, "Indirect control of an autonomous wheelchair using SSVEP BCI," *Journal of Robotics and Mechatronics,* vol. 32, p. 761–767, 2020.

[78] H. J. Landau, "Sampling, data transmission, and the Nyquist rate," *Proceedings of the IEEE,* vol. 55, pp. 1701-1706, 1967.

[79] J. Tang, M. Xu, J. Han, M. Liu, T. Dai, S. Chen and D. Ming, "Optimizing SSVEP-based BCI system towards practical high-speed spelling," *Sensors,* vol. 20, p. 4186, 2020.

[80] C.-J. Lee and J.-I. Song, "A chopper-stabilized current-feedback instrumentation amplifier for EEG acquisition applications," *IEEE Access,* vol. 7, p. 11565–11569, 2019.

[81] J. Kalra, P. Mittal, N. Mittal, A. Arora, U. Tewari, A. Chharia, R. Upadhyay, V. Kumar and L. Longo, "How Visual Stimuli Evoked P300 is Transforming

the Brain–Computer Interface Landscape: A PRISMA Compliant Systematic Review," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 1429-1439, 2023.

[82] R. Ghosh, N. Deb, K. Sengupta, A. Phukan, N. Choudhury, S. Kashyap, S. Phadikar, R. Saha, P. Das, N. Sinha and others, "SAM 40: Dataset of 40 subject EEG recordings to monitor the induced-stress while performing Stroop color-word test, arithmetic task, and mirror image recognition task," *Data in Brief*, vol. 40, p. 107772, 2022.

[83] P. Gajbhiye, N. Mingchinda, W. Chen, S. C. Mukhopadhyay, T. Wilaiprasitporn and R. K. Tripathy, "Wavelet domain optimized Savitzky–Golay filter for the removal of motion artifacts from EEG recordings," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, p. 1–11, 2020.

[84] S. Phadikar, N. Sinha and R. Ghosh, "Automatic Eyeblink Artifact Removal From EEG Signal Using Wavelet Transform With Heuristically Optimized Threshold," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, pp. 475-484, 2021.

[85] P. Kwiatkowski, "Digital-to-time converter for test equipment implemented using FPGA DSP blocks," *Measurement*, vol. 177, p. 109267, 2021.

[86] P. Kwiatkowski, "Employing FPGA DSP blocks for time-to-digital conversion," *Metrology and Measurement Systems*, vol. 26, 2019.

[87] L. Moroz, V. Samotyy, P. Gepner, M. W. and G. Nowakowski, "Power Function Algorithms Implemented in Microcontrollers and FPGAs," *Electronics*, vol. 12, p. 3399, 2023.

[88] H. Song, G. Luo, Z. Ji, R. Bo, Z. Xue, D. Yan, F. Zhang, K. Bai, J. Liu, X. Cheng and others, "Highly-integrated, miniaturized, stretchable electronic systems based on stacked multilayer network materials," *Science Advances*, vol. 8, p. eabm3785, 2022.

[89] B. Li, T. Cheng and Z. Guo, "A review of EEG acquisition, processing and application," in *Journal of Physics: Conference Series*, 2021.

[90] W. Zheng, W. Liu, Y. Lu, B. Lu and A. Cichocki, "EmotionMeter: A Multimodal Framework for Recognizing Human Emotions," *IEEE Transactions on Cybernetics*, pp. 1-13, 2018.

[91] M. Soleymani, J. Lichtenauer, T. Pun and M. Pantic, "A Multimodal Database for Affect Recognition and Implicit Tagging," *IEEE Transactions on Affective Computing*, vol. 3, pp. 42-55, 2012.

[92] T. Song, W. Zheng, C. Lu, Y. Zong, X. Zhang and Z. Cui, "MPED: A Multi-Modal Physiological Emotion Database for Discrete Emotion Recognition," *IEEE Access*, vol. 7, pp. 12177-12191, 2019.

[93] R. Alazrai, R. Homoud, H. Alwanni and M. Daoud, "EEG-Based Emotion Recognition Using Quadratic Time-Frequency Distribution," *Sensors*, vol. 18, p. 2739, August 2018.

[94] X. Li, Y. Zhang, P. Tiwari, D. Song, B. Hu, M. Yang, Z. Zhao, N. Kumar and P. Marttinen, "EEG Based Emotion Recognition: A Tutorial and Review," *ACM Computing Surveys*, vol. 55, p. 1–57, November 2022.

[95] H. Liu, Y. Zhang, Y. Li and X. Kong, "Review on emotion recognition based on electroencephalography," *Frontiers in Computational Neuroscience*, vol. 15, p. 84, 2021.

[96] J. Wang and M. Wang, "Review of the emotional feature extraction and classification using EEG signals," *Cognitive robotics*, vol. 1, p. 29–40, 2021.

[97] E. H. Houssein, A. Hammad and A. A. Ali, "Human emotion recognition from EEG-based brain–computer interface using machine learning: a comprehensive review," *Neural Computing and Applications*, vol. 34, p. 12527–12557, 2022.

[98] J. Semmlow, "Chapter 10 - Stochastic, Nonstationary, and Nonlinear Systems and Signals," in *Circuits, Signals and Systems for Bioengineers (Third Edition)*, Third Edition ed., J. Semmlow, Ed., Academic Press, 2018, pp. 449-489.

[99] D. Li, J. Liu, Y. Yang, F. Hou, H. Song, Y. Song, Q. Gao and Z. Mao, "Emotion recognition of subjects with hearing impairment based on fusion of facial expression and EEG topographic map," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, p. 437–445, 2022.

[100] S. Bagherzadeh, K. Maghooli, A. Shalbaf and A. Maghsoudi, "Emotion recognition using effective connectivity and pre-trained convolutional neural networks in EEG signals," *Cognitive Neurodynamics,* vol. 16, p. 1087–1106, 2022.

[101] T. Mullen, "Source information flow toolbox (SIFT)," *Swartz Center Comput Neurosci,* p. 1–69, 2010.

[102] C. H. R. I. S. DING and H. A. N. C. H. U. A. N. PENG, "MINIMUM REDUNDANCY FEATURE SELECTION FROM MICROARRAY GENE EXPRESSION DATA," *Journal of Bioinformatics and Computational Biology,* vol. 03, p. 185–205, April 2005.

[103] V. Doma and M. Pirouz, "A comparative analysis of machine learning methods for emotion recognition using EEG and peripheral physiological signals," *Journal of Big Data,* vol. 7, p. 1–21, 2020.

[104] J. Atkinson and D. Campos, "Improving BCI-based emotion recognition by combining EEG feature selection and kernel classifiers," *Expert Systems with Applications,* vol. 47, pp. 35-41, 2016.

[105] Z. Mohammadi, J. Frounchi and M. Amiri, "Wavelet-based emotion recognition system using EEG signal," *Neural Computing and Applications,* vol. 28, p. 1985–1990, January 2016.

[106] Y. Zhang, J. Chen, J. H. Tan, Y. Chen, Y. Chen, D. Li, L. Yang, J. Su, X. Huang and W. Che, "An investigation of deep learning models for EEG-based emotion recognition," *Frontiers in Neuroscience,* vol. 14, p. 622759, 2020.

[107] X. Xing, Z. Li, T. Xu, L. Shu, B. Hu and X. Xu, "SAE+LSTM: A New Framework for Emotion Recognition From Multi-Channel EEG," *Frontiers in Neurorobotics,* vol. 13, June 2019.

[108] A. N. M. Faisal, A. Rahman, M. T. Habib, A. Siddique, M. Hasan and M. Khan, "Neural networks based multivariate time series forecasting of solar radiation using meteorological data of different cities of Bangladesh," *Results in Engineering,* vol. 13, p. 100365, February 2022.

[109] "EEG emotion recognition using fusion model of graph convolutional neural networks and LSTM," *Applied Soft Computing,* vol. 100, p. 106954, 2021.

[110] Y. Li, W. Zheng, Z. Cui, T. Zhang and Y. Zong, "A Novel Neural Network Model based on Cerebral Hemispheric Asymmetry for EEG Emotion Recognition," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018.

[111] J. Cho and H. Hwang, "Spatio-Temporal Representation of an Electoencephalogram for Emotion Recognition Using a Three-Dimensional Convolutional Neural Network," *Sensors,* vol. 20, p. 3491, June 2020.

[112] Y. Cimtay and E. Ekmekcioglu, "Investigating the Use of Pretrained Convolutional Neural Network on Cross-Subject and Cross-Dataset EEG Emotion Recognition," *Sensors,* vol. 20, p. 2034, April 2020.

[113] X. Li, D. Song, P. Zhang, G. Yu, Y. Hou and B. Hu, "Emotion recognition from multi-channel EEG data through Convolutional Recurrent Neural Network," in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2016.

[114] M. Bilalpur, S. M. Kia, T.-S. Chua and S. Ramanathan, "Discovering gender differences in facial emotion recognition via implicit behavioral cues," *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII),* pp. 119-124, 2017.

[115] J. Bailenson, E. (. Pontikakis, I. Mauss, J. Gross, M. Jabon, C. Hutcherson, C. Nass and O. John, "Real-time classification of evoked emotions using facial feature tracking and physiological responses," *International Journal of Human-Computer Studies*, vol. 66, pp. 303-317, May 2008.

[116] Z. Lan, O. Sourina, L. Wang, R. Scherer and G. Müller-Putz, "Domain Adaptation Techniques for EEG-Based Emotion Recognition: A Comparative Study on Two Public Datasets," *IEEE Transactions on Cognitive and Developmental Systems,* vol. PP, pp. 1-1, April 2018.

[117] I. M. Agus Wirawan, R. Wardoyo, D. Lelono and S. Kusrohmaniah, "Comparison of Smoothing Methods to Remove Artifacts in Emotion Recognition based on Electroencephalogram Signals," in *2022 Seventh International Conference on Informatics and Computing (ICIC)*, 2022.

[118] B. Hjorth, "The physical significance of time domain descriptors in EEG analysis," *Electroencephalography and Clinical Neurophysiology*, vol. 34, pp. 321-325, 1973.

[119] D. Acharya, R. Jain, S. S. Panigrahi, R. Sahni, S. Jain, S. Deshmukh and A. Bhardwaj, "Multi-class Emotion Classification Using EEG Signals," *Communications in Computer and Information Science*, 2020.

[120] Y. Wang, Z. Huang, B. McCane and P. Neo, "EmotioNet: A 3-D Convolutional Neural Network for EEG-based Emotion Recognition," 2018.

[121] D. Huang, S. Chen, C. Liu, L. Zheng, Z. Tian and D. Jiang, "Differences First in Asymmetric Brain: A Bi-hemisphere Discrepancy Convolutional Neural Network for EEG Emotion Recognition," *Neurocomputing*, vol. 448, April 2021.

[122] J. Liu, X. Shen, S. Song and D. Zhang, "Domain Adaptation for Cross-Subject Emotion Recognition by Subject Clustering," *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 904-908, 2021.

[123] S. Bhosale, R. Chakraborty and S. K. Kopparapu, "Calibration free meta learning based approach for subject independent EEG emotion recognition," *Biomedical Signal Processing and Control*, vol. 72, p. 103289, February 2022.

APPENDIX A

CODE IMPLEMENTATION

1. Feature Extraction Implementation

```
!pip install git+https://github.com/forrestbao/pyeeg.git

import numpy as np
import pyeeg as pe
import pickle as pickle
import pandas as pd
import math
from sklearn import svm
from sklearn.preprocessing import normalize
import os
import time
from scipy.stats import skew, kurtosis

def Feature_Processing(sub, channel,band, window_size,
step_size,sample_rate):
    meta = []
    m = 0
    with open("/content/drive/My Drive/DEAP_ILVA/DATA/s" + sub +
'.dat', 'rb') as file:
        subject = pickle.load(file, encoding='latin1')
        print(sub)


        for i in range(0, 40):

            labels = subject["labels"][i]
            start = 0

            data = subject["data"][i]
            # data = np.transpose(data)
            print(data.shape)

            while start + window_size <= data.shape[1]:
                meta_array = np.empty((1,484))
                meta_data = np.empty((len(channel), 15))
# Initialize meta_data as an array
                measures_per_dataset = np.zeros((15,))
```

```
                for idx, j in enumerate(channel):
                    X = data[j][start : start + window_size]
                # Calculate statistics for each window
                    mean_per_window = np.mean(X,axis=0)
                    std_per_window = np.std(X, axis=0)
                    min_per_window = np.min(X, axis=0)
                    max_per_window = np.max(X, axis=0)
                    skewness_per_window = skew(X, axis=0)
                    kurtosis_per_window = kurtosis(X, axis=0)


 # Calculate Hjorth parameters
                       complex_per_window =
                       np.sqrt(np.mean(np.diff(X, axis=0)**2,
                       axis=0)) / np.std(X, axis=0)
                    mobility_per_window = np.std(np.diff(X,
axis=0), axis=0) / np.std(X, axis=0)
                    activity_per_window = np.std(X, axis=0)

# Calculate wavelet features
# wavelet_energy = calculate_wavelet_energy(X)
# wavelet_entropy = calculate_wavelet_entropy(X)

                    Y = pe.bin_power(X, band, sample_rate)

# # Calculate differential entropy
# diff_entropy = calculate_differential_entropy(X)

                    spec_entropy = calculate_spectral_entropy(X,
sample_rate)

                    Y = pe.bin_power(X, band, sample_rate)


                    # Store statistics for the current channel
                    measures_per_dataset[0] = mean_per_window
                    measures_per_dataset[1] = std_per_window
                    measures_per_dataset[2] = min_per_window
                    measures_per_dataset[3] = max_per_window
                    measures_per_dataset[4] = skewness_per_window
                    measures_per_dataset[5] = kurtosis_per_window
                    measures_per_dataset[6] = complex_per_window
                    measures_per_dataset[7] = mobility_per_window
                    measures_per_dataset[8] = activity_per_window
                    # measures_per_dataset[9] = wavelet_energy
                    # measures_per_dataset[10] = wavelet_entropy
                    # measures_per_dataset[11] = diff_entropy
```

```
                    measures_per_dataset[9] = spec_entropy
                    measures_per_dataset[10:] = Y[0]

                    # Store the calculated measures for the
                    current dataset
                    meta_data[idx] = measures_per_dataset[:]


                print(meta_data.shape)
                meta_data = meta_data.reshape((-1,))
                print(meta_data.shape)
                meta_array[0,0:480] = meta_data[:]
                meta_array[0,480:484] = labels[:]

                # meta_array.append(meta_data)
                # meta_array.append(labels)


                meta.append(meta_array)
                start = start + step_size
                print(i,j,meta_array.shape,start)


        meta = np.array(meta)
        np.save('/content/drive/My
Drive/DEAP_ILVA/Datasets/small_step/s' + sub, meta,
allow_pickle=True, fix_imports=True)

for subjects in sorted(subjectList):
    Feature_Processing (subjects, channel, band, window_size,
step_size, sample_rate)
```

2. Deep Learning Implementation

```
import numpy as np
#import pyeeg as pe
import pickle as pickle
import pandas as pd
import math

from sklearn import svm
from sklearn.preprocessing import normalize

import os
import time

import pandas as pd
import keras.backend as K
import numpy as np
import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
from keras.models import Sequential
from keras.layers import Conv1D
from keras.layers import MaxPooling1D
from keras.utils import to_categorical
from keras.layers import Flatten
from keras.layers import Dense
import numpy as np
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D,
MaxPooling2D
from keras import backend as K
from keras.models import Model
import timeit
from keras.models import Sequential
from keras.layers import Flatten, Dense, Dropout
from keras.layers import Convolution1D, MaxPooling1D,
ZeroPadding1D
from keras.optimizers import SGD
#import cv2, numpy as np
import warnings
warnings.filterwarnings('ignore')

subjectList = ['01','02','03', '04', '05', '06', '07', '08',
'09', '10', '11' , '12', '13', '14', '15', '16', '17', '18',
```

```
'19', '20', '21', '22', '23', '24', '25', '26', '27', '28', '29',
'30', '31', '32']


import numpy as np
from sklearn.model_selection import train_test_split

def split_eeg_data_by_subject_in_chunks(chunk_size,
train_size=0.7, val_size=0.15, test_size=0.15):

    # Ensure the split sizes add up to 1
    assert train_size + val_size + test_size == 1, "Split sizes
must add up to 1"

    train_data = []
    val_data = []
    test_data = []

    for subject in subjectList:
      with open('/kaggle/input/deapsmall/small_step/s' + subject
+ '.npy', 'rb') as file:
        sub1 = np.load(file,allow_pickle=True)
        #sub1 = sub1[240:,:,:]

#         sub1 = sub1[::2, :]

        print(sub1.shape)
        subject_data = sub1.reshape(sub1.shape[0],1*484)


        n_chunks = len(subject_data) // chunk_size
        subject_data = subject_data[:n_chunks * chunk_size]  #
Trim excess rows not fitting into a full chunk
        chunks = np.array_split(subject_data, n_chunks)

        # Split chunks into training and temp (validation + test)
        train_chunks, temp_chunks = train_test_split(chunks,
train_size=train_size, shuffle=False)

        # Split temp chunks into validation and test sets
        val_size_relative = val_size / (val_size + test_size)
        val_chunks, test_chunks = train_test_split(temp_chunks,
train_size=val_size_relative, shuffle=False)

        # Append chunks to the respective lists
        train_data.append(train_chunks)
        val_data.append(val_chunks)
```

```
        test_data.append(test_chunks)

      # Convert lists back to numpy arrays
      train_arr = np.concatenate(train_data)
      val_arr = np.concatenate(val_data)
      test_arr = np.concatenate(test_data)


    return train_arr, val_arr, test_arr


train_data, val_data, test_data =
split_eeg_data_by_subject_in_chunks(chunk_size=1)

train_data =
train_data.reshape((train_data.shape[0]*1,train_data.shape[2]))
val_data =
val_data.reshape((val_data.shape[0]*1,val_data.shape[2]))
test_data =
test_data.reshape((test_data.shape[0]*1,test_data.shape[2]))

# Check the resulting shapes
print(f"Train data shape: {train_data.shape}")
print(f"Validation data shape: {val_data.shape}")
print(f"Test data shape: {test_data.shape}")


import numpy as np
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from keras.utils import to_categorical


# choose the dataset 0 for time, 1 for frequency, 2 for time-
frequency

def choose_data(dataset_type,label_type, source_data_array,
select_nr):

  if select_nr == 32:
    select_channels =
[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24
,25,26,27,28,29,30,31]
  elif select_nr == 18:
    if label_type == 0:
```

```
        select_channels =
np.load('/kaggle/input/selectch/selected_channels_18_arousal.npy'
)
    elif label_type == 1:
        select_channels =
np.load('/kaggle/input/selectch/selected_channels_18.npy')
    elif label_type == 2 or label_type==3:
        select_channels =
np.load('/kaggle/input/selectch/selected_channels_18_dominance.np
y')
  elif select_nr == 14:
    if label_type == 0:
        select_channels =
np.load('/kaggle/input/selectch/selected_channels_14_arousal.npy'
)
    elif label_type == 1:
        select_channels =
np.load('/kaggle/input/selectch/selected_channels_14.npy')
    elif label_type == 2 or label_type==3:
        select_channels =
np.load('/kaggle/input/selectch/selected_channels_14_dominance.np
y')
  elif select_nr == 10:
    if label_type == 0:
        select_channels =
np.load('/kaggle/input/selectch/selected_channels_10_arousal.npy'
)
    elif label_type == 1:
        select_channels =
np.load('/kaggle/input/selectch/selected_channels_10.npy')
    elif label_type == 2 or label_type==3:
        select_channels =
np.load('/kaggle/input/selectch/selected_channels_10_dominance.np
y')



  if dataset_type == 0:
    ch = 0
    target_data_array =
np.empty((source_data_array.shape[0],select_nr,9))
    for channel in select_channels:
      target_data_array [:,ch,:] =
source_data_array[:,channel,0:9]
      ch += 1
  elif dataset_type == 1:
    ch = 0
```

```
    target_data_array =
np.empty((source_data_array.shape[0],select_nr,6))
    for channel in select_channels:
      target_data_array [:,ch,:]= source_data_array[:,channel,9:]
      ch += 1
  else:
    print('dataset_error')

  target_data_array =
target_data_array.reshape((source_data_array.shape[0],select_nr*t
arget_data_array.shape[2]))

  return target_data_array


# choose the measure to be 0 Arousal 1 Valence 3 Dominance for
deap
def choose_label(label_type, source_data_array):
  target_data_array = np.empty((source_data_array.shape[0]))

  if label_type == 0:
      target_data_array = source_data_array[:,0]
  elif label_type == 1:
      target_data_array = source_data_array[:,1]
  elif label_type == 2:
      target_data_array = source_data_array[:,2]
  elif label_type == 3:
      target_data_array = eight_class(source_data_array[:,:])
      print(target_data_array.shape)
  else:
    print('label_error')

  return target_data_array


def eight_class(label_array):
    array_classes = np.empty((label_array.shape[0],))

    print(label_array.shape)
    arousal = binary_labels(label_array[:,0])
    valence = binary_labels(label_array[:,1])
    dominance = binary_labels(label_array[:,2])

    for i in range(label_array.shape[0]):
        if arousal[i] == 0 and valence[i] == 0 and dominance[i]
== 0:
            array_classes[i] = 0
```

```
        elif arousal[i] == 0 and valence[i] == 0 and dominance[i]
== 1:
            array_classes[i] = 1
        elif arousal[i] == 0 and valence[i] == 1 and dominance[i]
== 0:
            array_classes[i] = 2
        elif arousal[i] == 0 and valence[i] == 1 and dominance[i]
== 1:
            array_classes[i] = 3
        elif arousal[i] == 1 and valence[i] == 0 and dominance[i]
== 0:
            array_classes[i] = 4
        elif arousal[i] == 1 and valence[i] == 0 and dominance[i]
== 1:
            array_classes[i] = 5
        elif arousal[i] == 1 and valence[i] == 1 and dominance[i]
== 0:
            array_classes[i] = 6
        elif arousal[i] == 1 and valence[i] == 1 and dominance[i]
== 1:
            array_classes[i] = 7
        else:
            print("error")
    return array_classes




# convert labels into binary
def binary_labels(label_array):
    array_binary = np.empty((label_array.shape[0],))

    for i in range(label_array.shape[0]):
      if 1 <= label_array[i] <= 4.55:
        array_binary[i] = 0

      elif 4.56 <= label_array[i] <=9:
        array_binary[i] = 1

      else:
        print(label_array[i],'error')

    return array_binary
# find Kmeans labels and distance from cluster centers and add to
dataset
```

```
  # y_train = to_categorical(classes_train)
  # y_val = to_categorical(classes_val)
  # y_test = to_categorical(classes_test)


def kmeans_features(x,y,n_clusters):
  # Apply K-means clustering

  kmeans = KMeans(n_clusters=n_clusters, random_state=42)
  kmeans.fit(x)

  # Transform the data by adding the distances from each cluster
center: +2 features
  X_transformed = np.hstack((x, kmeans.transform(x)))

  # Predict cluster labels for the entire dataset
  cluster_labels = kmeans.predict(x)

  # Add cluster labels as new features to the dataset
  X_transformed = np.hstack((X_transformed,
np.expand_dims(cluster_labels, axis=1)))

  return X_transformed

def dataset_kfolds(X_transformed,y):

    x_train = []
    y_train = []
    x_val = []
    y_val = []
    x_test = []
    y_test = []

    for i in range(X_transformed.shape[0]):
        if i % 10 == 0:
            x_test.append(X_transformed[i])
            y_test.append(y[i])
        elif i % 8 ==0:
            x_val.append(X_transformed[i])
            y_val.append(y[i])
        else:
            x_train.append(X_transformed[i])
            y_train.append(y[i])

    return
np.array(x_train),np.array(y_train),np.array(x_val),np.array(y_va
l),np.array(x_test),np.array(y_test)
```

```
def save_models(history, model, dataset_type, label_type,
nr_channels):

    if dataset_type == 0:
        dataset = 'time'
    elif dataset_type == 1:
        dataset = 'frequency'

    if label_type == 0:
        label = 'arousal'
    elif label_type == 1:
        label = 'valence'
    elif label_type==2:
        label = 'dominance'
    elif label_type==3:
        label= '8class'
    else:
        print('error')


    filename_model = '/kaggle/working/deap_cnn3_'+ dataset +'_' +
label + '_' + str(nr_channels) +'.h5'
    filename_history = '/kaggle/working/deap_cnn3_'+ dataset +'_'
+ label + '_' + str(nr_channels) +'.csv'

    model.save(filename_model)
    history_df = pd.DataFrame(history.history)
    history_df.to_csv(filename_history)


import numpy as np
from keras.models import load_model

def threshold_predictions(y_pred_prob, threshold):
    """
    Convert predicted probabilities to binary predictions based
on a threshold.

    Parameters:
    - y_pred_prob: Predicted probabilities (can be 1D or 2D
array)
    - threshold: Threshold value

    Returns:
    - Binary predictions
    """
```

```
    return (y_pred_prob >= threshold).astype(int)

def evaluating_model(x_test,y_test,dataset_type,label_type,
nr_channels, metrics_cnn,total_time, avg_time):

    if dataset_type == 0:
        dataset = 'time'
    elif dataset_type == 1:
        dataset = 'frequency'

    if label_type == 0:
        label = 'arousal'
    elif label_type == 1:
        label = 'valence'
    elif label_type==2:
        label = 'dominance'
    elif label_type==3:
        label= '8class'
    else:
        print('error')

    filename_model = '/kaggle/working/deap_cnn3_'+ dataset +'_' +
label + '_' + str(nr_channels) +'.h5'
    print(filename_model)
    x_test = x_test.reshape(x_test.shape[0],x_test.shape[1], 1)

    model = load_model(filename_model)
    evaluation = model.evaluate(x_test, y_test)

    # The evaluate method returns the model's loss value and
metrics values
    loss = evaluation[0]
    accuracy = evaluation[1]

    y_pred = model.predict(x_test)

    print(y_pred)

    # Example usage:
    threshold = 0.5  # Set your threshold value here
    y_pred_binary = threshold_predictions(y_pred, threshold)


    print("Loss:", loss)
    print("Accuracy:", accuracy)

    print(y_pred_binary)
```

```
    from sklearn import metrics
    precision = metrics.precision_score(y_test, y_pred_binary,
average='weighted')
    recall = metrics.recall_score(y_test, y_pred_binary,
average='weighted')
    f1_score = metrics.f1_score(y_test, y_pred_binary,
average='weighted')

    precision_0 = precision
    recall_0 = recall
    f1_0 = f1_score

    print(precision_0, recall_0, f1_0)

        # Combine all metrics into a dictionary
    individual_metrics = {
        'model':'cnn3',
        'dataset': dataset_type,
        'nr_channels' : nr_channels,
        'total_time': total_time,
        'average_time': avg_time,
        'emotion_categ' : label_type,
        'labels': '8class',
        'accuracy': accuracy,
        'precision': precision_0,
        'recall': recall_0,
        'f1-score': f1_0
        }

    print(individual_metrics)

    metrics_cnn.append(individual_metrics)



from tensorflow.keras.utils import to_categorical

def cnn_run(dataset_type,
label_type,train_data,val_data,test_data, results, channel_nr):

    # splilt data from labels
    x_train, y_train = train_data[:,:480], train_data[:,480:]
    x_val, y_val = val_data[:,:480], val_data[:,480:]
    x_test, y_test = test_data[:,:480], test_data[:,480:]

    print(x_train.shape, y_train.shape)
```

```
    print(x_val.shape, y_val.shape)
    print(x_test.shape, y_test.shape)

    # reshape them to show channels so that the dataset can be
selected
    X_train = x_train.reshape((x_train.shape[0],32,15))
    Y_train = y_train[:,:]

    X_val = x_val.reshape((x_val.shape[0],32,15))
    Y_val = y_val[:,:]

    X_test = x_test.reshape((x_test.shape[0],32,15))
    Y_test = y_test[:,:]


    # chose data
    x_train = choose_data(dataset_type,label_type,
X_train,channel_nr)
    x_val = choose_data(dataset_type,label_type, X_val,
channel_nr)
    x_test = choose_data(dataset_type,label_type,
X_test,channel_nr)

    print(x_train.shape, x_val.shape, x_test.shape)

    # choose labels
    y_train = choose_label(label_type, Y_train)
    y_val =choose_label(label_type, Y_val)
    y_test = choose_label(label_type, Y_test)
    print(y_train.shape,y_val.shape, y_test.shape)

    # turn labels into binary
    if label_type == 0 or label_type==1 or label_type==2:
        train_binary = binary_labels(y_train)
        val_binary = binary_labels(y_val)
        test_binary = binary_labels(y_test)
        n_clusters = 2
    else:
        train_binary = to_categorical(y_train[:])
        val_binary = to_categorical(y_val[:])
        test_binary = to_categorical(y_test[:])
        n_clusters = 8

    print(train_binary.shape,val_binary.shape,test_binary.shape)

    # concatenate for kmeans
    x = np.concatenate((x_train,x_val,x_test),axis=0)
```

```
    y = np.concatenate((train_binary,val_binary,test_binary),
axis=0)
    X_kmeans_add = kmeans_features(x,y,n_clusters)


    # create the datasets to be used for classification
    x_train,y_train,x_val,y_val,x_test,y_test=
dataset_kfolds(X_kmeans_add,y)
    print(x_train.shape, y_train.shape,x_val.shape, y_val.shape,
x_test.shape, y_test.shape)

#     unique_labels_train, counts_train = np.unique(y_train,
return_counts=True)
#     count_dict_train = dict(zip(unique_labels_train,
counts_train))

#     # Count occurrences of each label in validation set
#     unique_labels_val, counts_val = np.unique(y_val,
return_counts=True)
#     count_dict_val = dict(zip(unique_labels_val, counts_val))

#     # Count occurrences of each label in test set
#     unique_labels_test, counts_test = np.unique(y_test,
return_counts=True)
#     count_dict_test = dict(zip(unique_labels_test,
counts_test))

#     # Print results
#     print(f"Train set label counts: {count_dict_train}")
#     print(f"Validation set label counts: {count_dict_val}")
#     print(f"Test set label counts: {count_dict_test}")

    total_time, avg_time =
train_model(x_train,y_train,x_val,y_val,x_test,y_test,dataset_typ
e,label_type,channel_nr,n_clusters)
    evaluating_model(x_test,y_test,dataset_type,label_type,
channel_nr, results, total_time, avg_time)

metrics = []

cnn_run(1,2,train_data,val_data,test_data, metrics, 32)
cnn_run(1,2,train_data,val_data,test_data, metrics, 18)




df = pd.DataFrame(metrics)
```

```
print(df)
df.to_csv('/kaggle/working/deap_metrics_cnn3.csv',index=False)
```