Norwegian University
of Life Sciences

Master's Thesis 2023 30 ECTS
Faculty of Science and Technology

ECG-based Human Emotion Recognition Using Generative Models

Ole Gilje Gunnarshaug
MSc Data Science

Preface

With this thesis my master's degree in Data Science at the Norwegian University of Life Sciences (NMBU) is concluded. The research topic for the thesis was introduced by Associate Professor in Data Science at NMBU, Fadi Al Machot. Thanks to Fadi for both introducing me to the topic and being willing to guide me throughout the thesis. I appreciate the time and guidance provided in the process. Secondly, I would like to thank my co-supervisor, Habib Ullah, for his guidance and insightful comments on the thesis.

A thanks to my friends in TF6-206 for making the experience of writing a master's thesis to be nothing but positive. Finally, a special thanks to my good friends Herman Ellingsen and Kristian Olai Berg for their great support and for reading through the thesis providing helpful comments.

Ås, May, 2023
Ole Gilje Gunnarshaug

Abstract

Human emotion recognition (HER) is ever-evolving and has become an important research field. In autonomous driving, HER can be vital in developing autonomous vehicles. Introducing autonomous vehicles is expected to increase safety, having the potential to prevent accidents. Recognizing the passengers' emotional reactions while driving can help machine learning algorithms learn human behavior in traffic. In this thesis, the focus has been on HER using electrocardiogram (ECG) data. The effect of Autoencoders and Sparse Autoencoders in HER using ECG data has been explored and compared to the state-of-the-art. Additionally, the extent of ECG data as a single modality for HER has been discussed. Three pipelines were constructed to explore how Autoencoders and Sparse Autoencoders affect HER. All pipelines were denoised and resampled using the Pan-Tompkins algorithm. Additionally, the pipelines were all trained, validated, and tested using the Support Vector Classifier (SVC). The first pipeline uses the Pan-Tompkins processed signals as input to the SVC. In the second pipeline, the input to the SVC is features extracted from the signals using an Autoencoder. The last pipeline uses the latent space of a Sparse Autoencoder as input to the SVC. The target emotions for the classification task were based on the two-dimensional emotion model of valence and arousal, resulting in four classes. The pipeline including an Autoencoder for feature extraction outperformed the pipeline without feature extraction in addition to reducing the bias the models showed towards one class. Using a Sparse Autoencoder, the overall results were lower, but it was able to reduce the bias toward one class further. These results show that the Autoencoder has potential in ECG-based HER and could contribute to the field.

Contents

List of Figures

List of Tables

Table of Abbreviations

Abbreviations	**Meaning**
HER	Human Emotion Recognition
ECG	Electrocardiogram
EEG	Electroencephalography
GSR	Galvanic Skin Response
SVC	Support Vector Classifier
ANN	Artificial Neural Network
LR	Logistic Regression
KNN	K-Nearest Neighbors
RF	Random Forest
PT	Pan-Tompkins
DCNN	Deep Convolutional Neural Network
NB	Naïve Bayes
DT	Decision Tree
LALV	Low Arousal Low Valence
LAHV	Low Arousal High Valence
HALV	High Arousal Low Valence
HAHV	High Arousal High Valence
AI	Artificial Intelligence
SVM	Support Vector Machine
RBF	Radial Basis Function
SAM	Self-Assessment Manikins

Chapter 1

Introduction

1.1 Motivation

Human emotion recognition (HER) is an important research field that can be vital in different areas like Autonomous driving and Active and assisted living. In Autonomous driving, HER can contribute to the making of autonomous vehicles. Autonomous vehicles are expected to be safer than manual vehicles driven by humans today [1]. They are expected to be safer both when it comes to vehicle-to-vehicle and vehicle-to-infrastructure. Additionally, they will be able to help reduce the traffic in highly trafficked areas like in the cities. HER can be implemented to predict the passengers' emotions, which will be used to enhance safety. To make Autonomous driving a realistic concept, there are multiple challenges.

One big challenge is social acceptance, making people trust the idea. Another big challenge is creating a system to identify the correct emotional state accurately. A common way of making such a system is combining preprocessing with an appropriate machine learning model. The classical machine learning models often require extensive preprocessing, which can be time-consuming and challenging to implement. An alternative to the classical machine learning models is deep learning models. The deep learning models hold the advantage of learning from raw data, needing little or no preprocessing to recognize the emotions. In HER there are also different modalities to use as input for the machine learning models. One popular approach is using physiological signals like Electrocardiogram (ECG), electroencephalogram (EEG), and Galvanic skin response (GSR). The modality used in this study is ECG which has the advantage of being easy and cheap to measure [2].

1.2 Objectives

This thesis will focus on HER using the physiological signal, ECG by utilizing generative models and a classical machine learning model. The main objective is to construct an emotion recognition system that can utilize the latent space of generative models to enhance the accuracy of ECG-based emotion recognition. The experiments will be conducted on two benchmark datasets, namely the ASCERTAIN and MAHNOB datasets. The two generative models chosen for this thesis are the Autoencoder and the Sparse Autoencoder. Their influence on the performance of a classical machine learning model and its ability to explain the variability in ECG data will be reviewed. The Support Vector Classifier (SVC) was chosen to be the classifier in this study based on a review of different methods by M. Hasnul et al. in 2021 [2]. They stated that the SVC is a frequently used classifier in emotion recognition which displays the overall best performance for emotion recognition systems based on their study. A secondary objective for this thesis is to compare the performance of the generative models to the state-of-the-art and explore if ECG data can be an effective modality for single-modal HER. By comparing the results to state-of-the-art, the goal is to highlight the relevance of this study.

To summarize the objectives of this thesis, two research questions have been formulated as follows:

- RQ1: To what extent can the latent space captured by an Autoencoder and a Sparse Autoencoder explain the variability in ECG data?
- RQ2: What is the overall performance of the proposed pipelines compared to state-of-the-art, and how far can ECG be used as a single modality for HER?

1.3 Related Work

This section will review state-of-the-art methods for HER using ECG signals. Only studies after 2017 will be considered to ensure only the most relevant and up-to-date studies are included. Tables 1.1 and 1.2 summarize the different approaches reviewed in this section and their classification accuracies.

In 2019 D. Nikolova et al. published a paper on emotion recognition with ECG signals using an Artificial Neural Network (ANN) and Logistic Regression (LR) to discriminate human emotional states across various subjects [3]. With the ANN providing a 35% classification accuracy and the LR providing 40% accuracy, they concluded that ECG has a potential in affective computing if combined with other modalities. Furthermore, in [4], SVC, K-Nearest Neighbors (KNN), and Random Forest (RF) are compared using a finite impulse filter for denoising and the Discrete Cosine Transform to extract features. They concluded with SVC providing the highest average accuracy of 91%. L. Santamaria-Granados et al. used the Pan-Tompkins (PT) algorithm to transform the signals before using it as input for a Deep Convolutional Neural Network (DCNN) [5]. Their proposed method achieved 76% and 75% accuracy for arousal and valence, respectively. In [6], S. Ismail et al. compared ECG and Photoplethysmogram signals with four machine learning models: SVC, Naïve Bayes (NB), KNN, and Decision Tree (DT). For both arousal and valence, the best accuracies with ECG signals were achieved using an SVC, with an accuracy of 69% for arousal and 59% for valence. With two-dimensional target emotions combining arousal and valence, the highest accuracy was 32% using KNN.

Table 1.1: Summary of related work and their respective classification accuracies. In this table, the focus is on the related work using similar methods as in this research.

Ref	Dataset	Emotion label	Signals Processing	Classifier	Accuracy
[3]	Self-made dataset	Fear, Disgust, Neutral	Statistical features: two based on R peak amplitude and six based on length of the RR intervals	ANN and LR	ANN: 35%, LR: 40%
[4]	Self-made dataset	Happy, Exciting, Calm, Tense	Finite impulse filter and Discrete Cosine Transform	KNN, RF and SVC	KNN: 83%, RF: 82%, SVC: 91%
[5]	AMIGOS	Arousal and Valence	PT algorithm	DCNN	Arousal: 76%, Valence: 75%
[6]	Self-made dataset	Arousal, Valence and Two-dimensional	PT algorithm	SVC, NB, KNN, DT	Arousal SVC: 69%, Valence SVC: 59%, Two-dimensional KNN: 32%

In Table 1.2, research including ECG signals from either ASCERTAIN or MAHNOB is presented. In [7] M.Wiem and Z. Lachiri extracted 169 features from the peripheral physiological signals in MAHNOB and used them as input for an SVC. Their results concluded that ECG and respiration volume were the two most relevant signals for HER. The ECG signals achieved accuracies of 66% for arousal and 65% for valence. In [8] F. Panahi et al. used both ECG and GSR signals to study the effectiveness of the Fractional Fourier Transform to improve the accuracy of HER using physiological signals. They used an SVC to classify the extracted features and concluded that the phase information from the Fractional Fourier Transform using ECG signals achieved the highest accuracy of 77% for arousal and 78% for valence.

Table 1.2: Summary of related work and their respective classification accuracies. In this table, the focus is on related work using ASCERTAIN and MAHNOB as benchmark datasets. Only papers including ECG signals for emotion recognition will be considered.

Ref	Dataset	Emotion label	Signals Processing	Classifier	Accuracy
[7]	MAHNOB	Arousal and Valence	Butterworth filter and Statistical features	SVC	Arousal: 66%, Valence: 65%
[9]	MAHNOB	Arousal and Valence	Neighborhood Component Analysis Dimensionality reduction	KNN	Arousal: 66%, Valence: 65%
[10]	ASCERTAIN	Arousal and Valence	None	DCNN with Convolutional Block Attention Module	Arousal: 79%, Valence: 76%
[8]	ASCERTAIN	Arousal and Valence	Phase information of Fractional Fourier Transform	SVC	Arousal: 77%, Valence: 78%

In this thesis, the target emotions were chosen to be arousal and valence, but in contrast to most related works presented in this section, arousal and valence were combined. Instead of having separate predictions for arousal and valence with a two-part binary classification task, they were combined for a two-dimensional emotion model. The four target emotions were set to be: LALV, LAHV, HALV, and HAHV, which will be further explained in Chapter 3. Furthermore, this study explores the effect of generative models like the Autoencoder and Sparse Autoencoder for feature extraction. Both of the mentioned Autoencoders have been used with EEG signals, but this study aims to explore their effect on ECG-based emotion recognition [11, 12].

1.4 Contributions

In this thesis, the goal is to contribute to the field of HER by exploring the ability of Autoencoders and Sparse Autoencoders to capture the variability in ECG data from benchmark datasets like ASCERTAIN and MAHNOB. Three pipelines are constructed, with the first pipeline being PT-SVC, acting as a baseline feeding the preprocessed signals to an SVC. The second pipeline is PT-AE-SVC utilizing an Autoencoder for feature extraction before training an SVC. Lastly, the third pipeline is PT-SAE-SVC, where the signals are encoded using a Sparse Autoencoder and feeding the sparse latent space to an SVC. As different variations of Autoencoders have shown potential for feature extraction using EEG signals, this paper looks to study their ability to provide better results for ECG data [11, 12]. The results discovered in this study will be compared with state-of-the-art approaches to see if Autoencoders and Sparse Autoencoders can contribute to the field of HER using ECG signals.

The pipelines proposed in this thesis are all preprocessed the same way. The signals are oversampled using the random oversampling technique to get the same number of samples for each class. Next, the PT QRS-detection algorithm is applied to the signals both to reduce the noise and to resample the signals to be the same length. In PT-SVC, the signals are standardized before training the classifier, while in PT-AE-SVC and PT-SAE-SVC, the signals are normalized before training the encoders. All pipelines are also trained and tested using the same machine learning model, namely SVC. The SVCs are tuned separately finding the optimal hyperparameters of each classifier. The target emotions chosen for the experiments are based on the two-dimensional emotion model of valence and arousal with four target emotions. The ambition of using this emotion model is to contribute to predicting the level of both valence and arousal combined.

1.5 Thesis Overview

The thesis structure for the remaining chapters will be: Chapter 2, discussing the theory of emotions and how emotions are measured, the preprocessing techniques used on the signals, and machine learning models. Chapter 3 will cover the methods used for preprocessing and the three pipelines used for emotion recognition and how they were implemented. Furthermore, in Chapter 4 the benchmark datasets used for the experiments and the results for the three pipelines are presented. Chapter 5 will discuss the results in the context of the research questions and their relevance in addition to the remaining challenges in HER. Finally, Chapter 6 will provide a conclusion of this thesis.

Chapter 2

Theory

2.1 Emotions

The scientific research on emotions began in the late 1800s when the Danish physiologist C. Lange and the American psychologist W. James started their research for the book “The Emotions” [13]. Since James and Lange started their research on emotions, there have been conflicts about the definition of emotions. However, there is a consensus that emotions resemble modulatory systems which have interactions with both “higher-order” and “lower-order” systems to affect the physical behavior [14]. These interactions follow events that people experience; different people experiencing the same event might have different emotions. In addition to the disagreement on defining emotions, there are different ways of characterizing them.

2.1.1 Basic Emotions

In [15], P. Ekman and W. Friesen proposed happiness, sadness, anger, fear, surprise, disgust, and interest to be the seven universal emotions, or as they called them, the primary effects. Ekman and Friesen followed up on their proposal of the seven universal emotions in [16] by comparing the facial expressions of emotion of different cultures. Their goal for the study was to show that preliterate cultures with no to little contact with the literate culture would show similar facial behavior to members of the literate culture. The study was set in New Guinea, where they selected subjects that were highly unlikely to have been affected by facial behavior from literate culture. To study their facial behavior, the experiments involved telling stories, showing pictures, and a combination of the two, all designed to be relevant to only one emotion. While these seven emotions were proposed to be universal by Ekman and Friesen in 1969, Ekman later rejected some of their research and suggested happiness, sadness, anger, fear, disgust, and surprise to be the six basic emotions [17]. The basic emotions are considered to be biological and easier to separate and recognize than other emotions. There have also been challenges to the claim of basic emotions [18, 19]. In [18], A. Wierzbicka argues that the claim of basic emotions is biased based on the native language of the researchers.

2.1.2 Complex Emotions

Describing an emotion experienced by a subject can be a complex task due to the complexity of the human brain and emotions. In contrast to basic emotions, complex emotions can be described as intricate and challenging to describe and recognize. Complex emotions are often a combination of multiple basic emotions, and the experience can differ depending on the individual. Envy, guilt, and shame are typical examples of complex emotions. In [20], philosopher R. Wollheim described emotions as an extended mental episode occurring from something that either satisfies or frustrates a pre-existing desire. This definition differs from the formerly discussed basic emotions, relating more to the philosophical aspect of emotions, like moral psychology. In moral psychology, extending the view of emotions beyond the six basic emotions is important. The way people think and their choices are often related to complex emotions, like guilt or shame after making a mistake. In [21] P. Griffiths describes emotions as "Machiavellian" emotions. He argues that emotions are used strategically, meaning that they are expressed and possibly produced when it is advantageous in the context of social events. This proposed description of emotions further shows the complexity that lies within human emotions.

2.1.3 Arousal and Valence

Instead of describing emotions with a set of basic emotions and extending from them, a popular approach to characterize the emotions is Lang's proposed method of measuring the level of valence and arousal [22]. In Lang's characterization, valence is either pleasant (positive) or unpleasant (negative), and arousal ranks from high to low. As seen in Figure 2.1, valence and arousal can be explained as two dimensions orthogonal to each other, where valence is the emotional direction of either pleasant or unpleasant, and arousal is the intensity of the emotion. The figure also shows the seven universal emotions proposed by Ekman and Friesen mapped in context to the two-dimensional emotion space proposed by Lang. A person with positive valence and high arousal can then be assumed to be happy.

2.2 Measuring Emotions

Making an accurate measurement of the emotion that is experienced by a subject given a type of stimulus is, as mentioned earlier, not an easy task. In [23], they divide the measuring methods into self-reporting and machine assessment techniques. The self-reporting techniques are usually some kind of questionnaire for the subject to fill out about the subjective experience of the emotions. Machine assessment techniques are different methods for measuring the physiological signals sent from the human body. These machine assessment techniques can yet again be divided into two categories: non-invasive and invasive measuring methods. The non-invasive measuring methods do not include any devices connected to the subject's body where the subject might feel "intruded". An example of a non-invasive machine assessment technique is an endosomatic methodology for measuring GSR, which is not using any external current to obtain the signals [24]. In contrast to the non-invasive techniques, the invasive machine assessment techniques use tools that will be directly connected to the subject's body. An example of an invasive measuring technique is a wearable Shimmer3 ECG sensor strapped around the waist and connected to several places on the upper body.

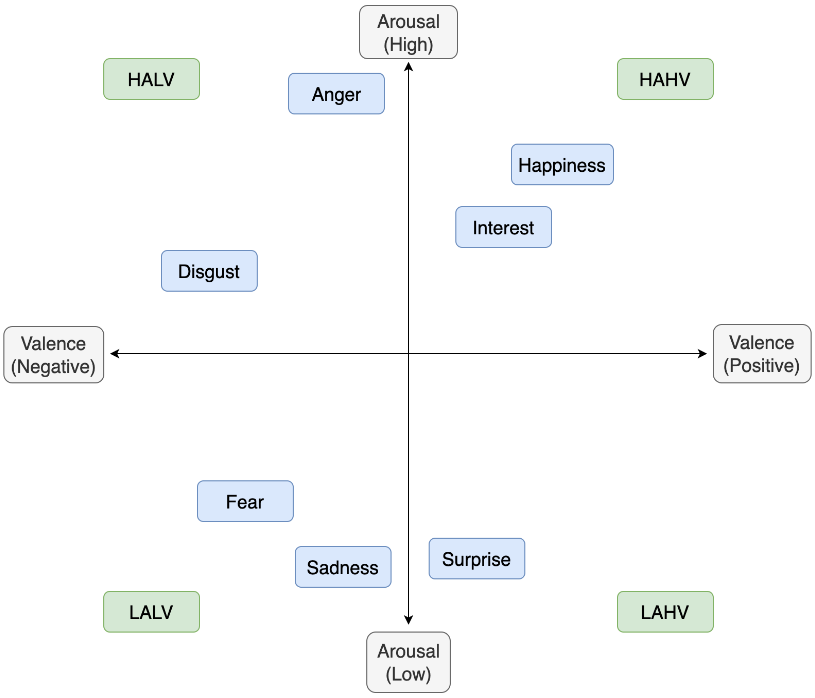


Figure 2.1: Showing the seven universal emotions proposed by Ekman and Friesen mapped in the two-dimensional space of valence and arousal. The green boxes contain the target emotions used for classification in this thesis. The target emotions are defined as: HALV: High Arosual Low Valence, HAHV: High Arousal High Valence, LALV: Low Arousal Low Valence, LAHV: Low Arousal High Valence.

2.2.1 Visual Sensors

Using visual sensors for emotion recognition is a common approach due to its low cost and that it is an effective way to collect data. The main types of visual sensors are cameras used for facial emotion recognition and photoplethysmography technology for heart rate detection [25]. Facial emotion recognition is an important and commonly used type of emotion recognition where facial expressions are recorded using visual sensors. Observing the facial behavior during some form of stimulus can provide a lot of information on the subject's emotions. For instance, a smile often refers to a positive emotion like happiness. A downside to using visual sensors is that it depends on the lighting. Having lousy lighting will provide a worse performance for emotion recognition. To overcome this challenge, there are different types of visual sensors like NIR cameras, thermal cameras, and Kinect sensors. NIR camera is a type of camera capturing near-infrared bands which goes beyond the visible spectrum. They are especially good for detecting changes in skin color and textures [26]. The thermal cameras are similar to NIR cameras, but they capture the changes in skin temperature [27]. The Kinect sensors are a combination of multiple sensors, such as RGB cameras and depth sensors. This enables it to capture changes in facial expression and track the movement of facial muscles [28].

2.2.2 Audio Sensors

Another common approach to emotion recognition is using audio sensors for speech emotion recognition [29]. Audio sensors can be handy because they capture speech, providing valuable information for recognizing emotions. After audio data is collected, feature extraction is crucial to obtaining the relevant information. The most apparent audio sensor is the microphone capturing recordings from people speaking [30]. Furthermore, the voice stress analyzer is a type of microphone capturing the change of stress in speech. This microphone is especially utilized for speech emotion recognition in lie detection [31]. Like visual sensors, audio sensors are considered a cheap and effective way to collect data, but they also have limitations. In speech, there is a lot of variability between subjects. There is no objective truth to how emotions in speech are either expressed or perceived [32]. This affects the quality of feature labeling and feature extraction. In addition, it is not always given that a sentence only contains one single emotion making it difficult for a machine to capture these emotions.

2.2.3 Physiological Signals

Physiological signals can be measured using both invasive and non-invasive machine-measuring methods. The most common physiological signals used for emotion recognition are EEG, GSR, and ECG [23]. EEG measures electrical signals from the brain and is usually measured using an electroencephalogram. The electroencephalogram consists of metal plate electrodes that are connected to the head. GSR, also known as skin conductance or electrodermal activity measures electrical signals from the skin. When a person is subjected to a stimulus affecting the emotional state, the skin responds with a sweat reaction. The change in sweat is then captured by sensors placed on the fingertips, the surface of the hands, the soles of the feet, or a combination of the three [33].

ECG signals are, as mentioned, usually measured using invasive machine assessment techniques like the wearable Shimmer3 ECG sensor. The sensor is supposed to capture the heart's electrical signals while the subject is exposed to some stimulus. The ECG signals can be depicted in a graph where each cardiac cycle generates a series of waves and deflections. In Figure 2.2, a single deflection is shown to demonstrate the directions of the signal. The waves have a baseline where the deflection is neutral. When the deflection is above the baseline, it is a positive deflection, and likewise, it is negative when the deflection is below the baseline. In addition to the direction of the deflection, the magnitude is also significant when interpreting it. The magnitude of a deflection is measured in millimeters of voltage and can be understood as how far the deflection is from the baseline. When the deflection is far from the baseline, it is considered a high magnitude, and when it is close to the baseline, it is considered a low magnitude [34].

2.3 Preprocessing

In emotion recognition using physiological signals, preprocessing is a crucial step to increase the performance of a machine learning model. There are many techniques to process physiological signals, like denoising the signals using different types of filters or various feature extraction methods. Some common feature extraction methods used for physiological signals are the discrete wavelet transform, Fourier transform, and Pan and Tompkins' QRS-detection [35–37].

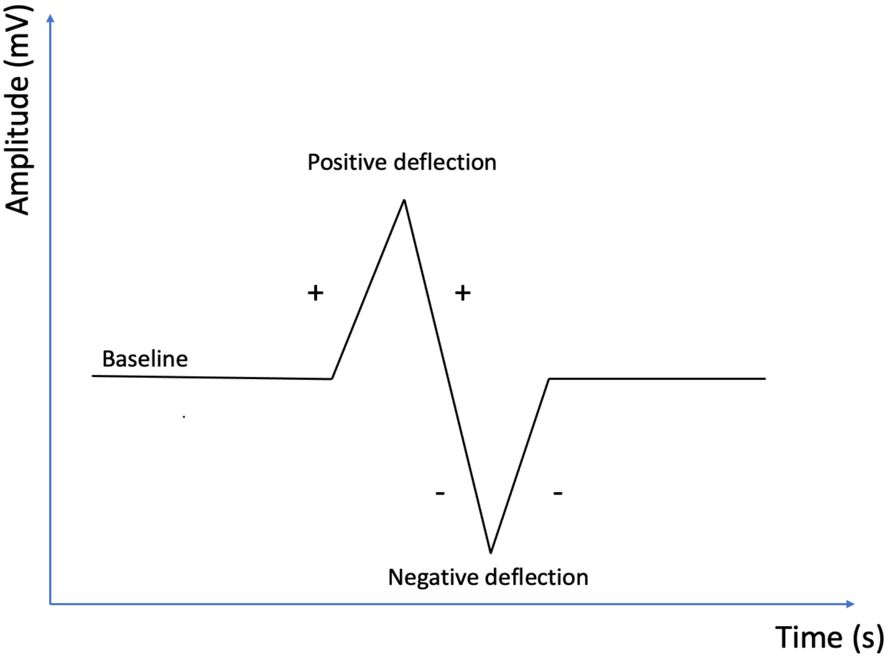


Figure 2.2: Showing the direction of a deflection from an ECG signal. When the line is above the baseline, the deflection is positive, and when it is below the baseline, it is negative.

2.3.1 Filtering

In signal preprocessing for emotion recognition, filtering refers to filtering out the noise from the raw signals. This is especially important for ECG signals due to the noise level that occurs when measuring the heart rate. The ECG signals can be disturbed by several factors, including interference from the power line, muscle movements, baseline wanders, motion artifacts, and external electrical system interference [2]. Depending on the type of noise, there are different types of filters to apply to the signals. The low-pass filter is a commonly used filter to denoise the signals with a cutoff value, which limits the maximum frequency [38]. Another popular type of filter is the high-pass filter. In contrast to the low-pass filter, the high-pass filter sets a cutoff value to limit the minimum frequency of the signal. The filter removes all noise that is either near the desired minimum frequency or below [39]. A bandpass filter is a combination of low-pass and high-pass filters. The signals are passed within a given range of frequencies [40].

2.3.2 Pan-Tompkins Algorithm

In 1985 J. Pan and W. Tompkins proposed a real-time QRS-detection algorithm for detecting the QRS complexes of ECG signals [41]. Because of all the noise produced when measuring ECG signals, QRS-detection can be challenging. In the PT QRS-detection algorithm, they reduce the influence of the different sources of noise using digital filters. They use cascaded low-pass and high-pass filters, which construct an integer-coefficient bandpass filter. Next, they have a filter that calculates the derivative of the signal to get information on the slope of the QRS complex. Furthermore, they square the differentiated signal point by point. This makes all the data points positive and emphasizes the higher frequencies because of the nonlinear amplification of the output. Lastly, they pass the signal through a moving-window integration.

The moving-window integration is supposed to acquire information on the R-wave slope and the waveform feature.

2.4 Machine Learning Theory

Machine learning is a subfield of Artificial Intelligence (AI) that revolves around structured and unstructured data. In simple terms, machine learning can be understood as algorithms built to use the knowledge that lies in data to learn from itself and make predictions. The introduction of machine learning is groundbreaking when it comes to utilizing the data available efficiently. Instead of having humans manually analyze the data and make rules, we can now rely on self-learning machines to make the decisions for us [42].

2.4.1 Classical Machine Learning

The possibility of using machines to learn from themselves was introduced in 1950 when A. Turing asked the question: “Can machines think?” [43]. In the years following Turing’s test paper, studies on machine learning were starting to take form. In 1959 A. Samuel published a paper on using the game of checkers to explore two machine learning procedures [44]. With his study on machine learning, he could conclude that it is possible to program a machine to play the game of checkers better than the creator of the program.

As machine learning is a subfield of AI, we can further divide machine learning into three categories of supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the objective is to make an algorithm that can use historical and labeled data to make predictions about future data or data unknown to the algorithm. A supervised learning algorithm can either be made to predict categories, known as classification, or to predict continuous outcomes, known as regression analysis. In both cases, the algorithm is given a set of historical data with labels that are used to train it to make predictions on new data. In contrast to supervised learning, unsupervised learning uses unlabeled data or data where the structure is unknown. Unsupervised learning is used to find data patterns and gain new knowledge based solely on the data fed into the algorithm. Lastly, reinforcement learning is based on a self-learning system that improves from feedback. While this sounds a lot like supervised learning, the agent does not learn from known labels but from a reward function that measures its performance. Simplified, reinforcement learning can be explained as a trial-and-error approach [42].

The Perceptron

The first Machine learning algorithm was proposed by Rosenblatt in 1958 [45] called The Perceptron. The Perceptron is designed to replicate the neurons in the brain where they either fire or not. The algorithm uses weights, w_j, initially set to zero or very small values close to zero. The output is then calculated, and the weights are updated based on whether the calculated output is correct or not. The weights are calculated as follows:

$$w_j := w_j + \Delta w_j \tag{2.1}$$

Furthermore, the Δw_j, used to update the weights, is calculated as in Equation 2.2. In the equation, η is the weights' learning rate, y^i denotes the true labels for the classes, $\hat{y}^i$ is the predicted class labels, and x_j^i the sample values of feature j in vector i. The weights are updated until the Perceptron converges. If the classes are not linearly separable, a maximum number of iterations or accepted misclassifications can be given to stop the algorithm from updating the weights.

$$\Delta w_j = \eta(y^i - \hat{y}^i)x_j^i \tag{2.2}$$

Support Vector Machine

Support Vector Machine (SVM) is a machine learning algorithm inspired by the Perceptron algorithm. As discussed, the Perceptron tries to minimize the number of misclassified samples. In an SVM, the goal of the objective function is to find the best decision boundaries or hyperplanes that maximize the margin between the classes. Calculating the separating hyperplane is an optimization problem which can be defined as [46]:

$$\min \frac{1}{2}\|w\|^2 \text{ subject to } t_i(w^T x_i + w_0) \geq +1, \forall i \tag{2.3}$$

In Equation 2.3 w denotes the weight vector of the coefficients for the hyperplane, x_i represents every data point, and t_i is the label of each data point. The optimization problem defined in Equation 2.3 can be solved by utilizing the Lagrange multipliers. The new optimization problem can now be defined as:

$$L_d = -\frac{1}{2}\sum_i \sum_k \alpha_i \alpha_k t_i t_k (x_i)^T (x_k) + \sum_i \alpha_i \tag{2.4}$$

α in Equation 2.4 denotes the Lagrange multipliers added to the optimization problem.

To maximize the distance between the two classes, the margin consisting of two additional hyperplanes, one positive and one negative, is calculated. The positive and negative hyperplanes are calculated as shown in Equations 2.5 and 2.6, respectively, where w^T is the transposed weight vector and x is the sample vector. However, in most real cases, achieving a perfectly separated margin while maintaining robustness is impossible. This is where the slack variable and regularization penalty come into play.

In 1995, V. Vapnik and C. Cortes introduced the slack variable to deal with non-linearly separable classes, resulting in the soft-margin classification in SVM [47]. The slack variable permits a certain amount of misclassification. To regulate the penalization, a constant C is used. This parameter penalizes samples that are misclassified. The regularization penalty establishes the order of importance between maximizing the margin and accurately classifying the samples.

Obtaining as many correctly classified samples as possible is essential. However, a smaller margin may lead to overfitting and a less robust model, whereas a larger margin may compromise classification precision. To achieve the desired trade-off between margin size and classification accuracy, choosing an appropriate penalization parameter is crucial.

$$w_0 + \mathbf{w}^T \mathbf{x}_{pos} = 1 \tag{2.5}$$

$$w_0 + \mathbf{w}^T \mathbf{x}_{neg} = -1 \tag{2.6}$$

An alternative to the linear SVM allowing a certain number of misclassifications, the kernel trick can be implemented to handle linearly inseparable data. The kernel trick is a way to map the feature space to a multidimensional feature space, separate the classes and map it back to the original feature space. The kernel trick uses the mapping function, often referred to as Φ, to project the data onto the new feature space for it to be linearly separable. The kernel trick can be implemented using different variations of this mapping function, like the Polynomial and the Radial Basis Function (RBF). Choosing a kernel is an important issue when training an SVM, but there is no easy way to know which one to use for any specific problem. In [48], G. Prajapati and A. Patle compared the performance of a Polynomial kernel and the RBF kernel and concluded that the RBF seemed better suited for larger datasets. The only other conclusion they could make after their analysis was that the choice of kernel directly impacts the accuracy of classification. In more recent studies, there is also a lack of consensus on which kernel to choose. The reports show different kernels providing the best performance in different classification and regression problems [49–51].

2.4.2 Deep Learning

Just like machine learning is a subfield of AI, deep learning is a subfield of machine learning. The theory behind deep learning can be dated back to the 1940s when Warren McCulloch and Walter Pitts proposed an explanation of how the neurons in the brain might function [52]. The field is built around the concept of imitating the learning process of the biological brain. The idea is that imitating the neurological activity in the brain could help solve complex problems. This idea was for the first time applied in Rosenblatt's perceptron in 1960 [53]. The perceptron algorithm was made to use the visual stimuli in the theoretical model, also called the perceptron, to stimulate perceptual learning, recognition, and spontaneous classification. As this was only a single-layer machine learning model, the first true multilayered neural network was not proposed until D. E Rumelhart et al. implemented the backpropagation in 1986 [54]. The idea behind adding multiple layers to the neural networks is that it can help solve the more complex problems like with image and voice recognition. Furthermore, there is also a downside to adding layers to the model. With an increased number of layers, the complexity of the model itself also increases. A highly complex deep learning model will be more costly to train and could, in some cases, overcomplicate the problem at hand.

Neural networks come in different variations and complexities. With the different problems, there are different types and levels of layers that can optimize the network. The different types of problems can also provide different types of input data and require different variations of outputs. However, all multilayered neural networks consist of an input layer, one or more hidden layers, and an output layer. A simple example of a multilayered neural network is the multilayered perceptron depicted in Figure 2.3. This network is a typical example of a feedforward network, which means that each layer receives the output of the preceding layer as its input. In this particular example, there is one input layer consisting of three input units, two hidden layers consisting of three hidden units, and an output layer with two output units. The layers are all connected through weighted coefficients, which link the units together and form a fully connected neural network. The input data is passed through the input layer, where the units calculate a weighted sum. This step is repeated, in this case only two times, until the data reaches the output layer where the output is calculated using an activation function [42].

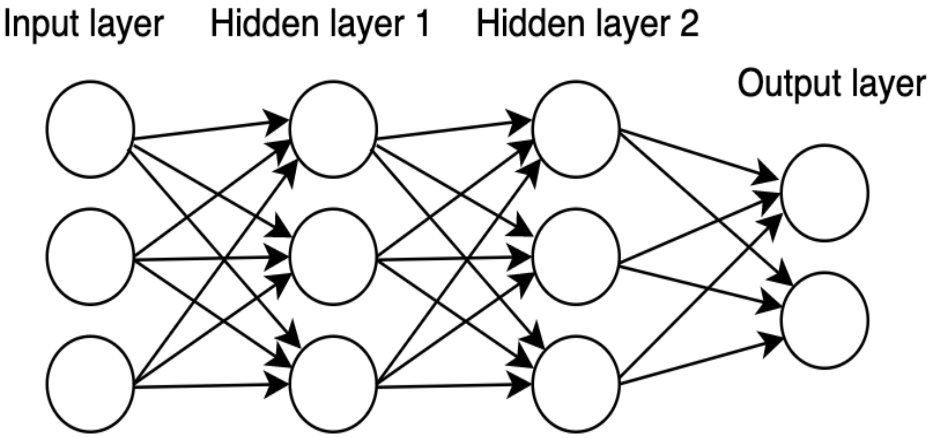


Figure 2.3: Showing the architecture of a fully connected feedforward ANN consisting of an input layer with three units, two hidden layers with three units, and an output with two units.

Most neural networks used for machine learning today are trained with the backpropagation algorithm [42]. The backpropagation algorithm is an efficient method for computing the partial derivatives of a chosen cost function in multilayered neural networks. The derivatives are calculated for the algorithm to learn the weight coefficients for parameterizing the neural networks. Because neural networks can grow very complex, containing several layers and several neurons in these layers, the computations can consequently also become very complex. To deal with the complexity, the backpropagation starts from the output by multiplying the output vector with the last weight vector resulting in a new vector to multiply with the following weight matrix. This process is repeated for all layers in the neural network. These vector-matrix multiplications are much less computationally expensive than the matrix-matrix multiplications of the weight matrices with the same process starting with the input.

Autoencoder

The Autoencoder was formally introduced in 1987 by McClelland et al. [55] and was originally intended as an unsupervised learning algorithm. Autoencoders are neural networks constructed to encode the input data to a compressed representation and then decode it to be as similar to the input data as possible. The Autoencoder network is trying to learn the function $h_{W,b}(x) \approx x$, which can be understood as an approximation to the identity function. The goal is to find structures in the data by having constraints like limiting the number of hidden units in the hidden layers. Figure 2.4 shows an example of a simple Autoencoder architecture. The network has six input units which are encoded into a compressed representation of three hidden units. The second layer is called the bottleneck layer containing the latent space of the encoded inputs. By extracting the latent space of the bottleneck layer, the Autoencoder can be used as a feature extraction technique in supervised learning. Next, it decodes the data, trying to reconstruct the original data using only the information from the hidden units' activation vectors. In the process of decoding the data, the algorithm looks for correlations between the features to help it reconstruct the input data.

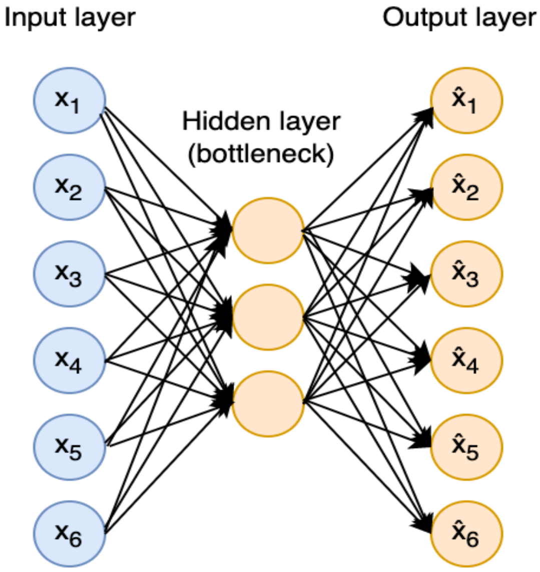


Figure 2.4: Showing the architecture of a simple Autoencoder consisting of one input layer with six input units, a hidden layer with three units, and an output layer with six output units.

Sparse Autoencoders

Giving the network the constraint of fewer units in the hidden layer than in the input layer is the most common way to avoid having it learn the identity function. Another way to restrain the model from learning the identity function is by making the activations of the hidden units sparse. Adding the sparsity constraints will mean that most nodes will be zero. Having more or the same number of hidden units as input units require a regularization term. The sparsity regularization is similar to normal regularization, but instead of applying it to the weights, it is applied to the activations. Adding the regularization term to the Autoencoder makes the overall cost function for the Sparse Autoencoder expressed as following [56]:

$$J_{sparse}(W,b) = J(W,b) + \beta \sum_{j=1}^{S_2} KL(\rho||\hat{\rho}_j) \tag{2.7}$$

In equation 2.7 β is a parameter that regulates the weight of the penalty that is added to neurons where $\hat{\rho}_j$ and ρ are significantly different. KL is the Kullback-Leibler divergence of two Bernoulli random variables with means ρ and $\hat{\rho}_j$ and measures the similarity between the two distributions. S_2 is the number of units in the hidden layer.

Chapter 3

Method

In this chapter, the methods used for the emotion recognition task will be presented. First, the process of defining target classes will be explained. Furthermore, the preprocessing techniques are described. Lastly, the pipelines constructed to explore the research questions are reviewed in detail.

3.1 General Settings

The experiments for this paper were all implemented on Google's collaborative notebooks, known as Google Colab notebooks. The Google Colab notebook is a Jupyter notebook hosted on the browser, which provides graphics processing units for computationally expensive machine learning tasks. Specifically, the Colab Pro package was used to perform the experiments efficiently. In the experimental process of creating, training, and testing the models, the following packages were used:

- Scikit-learn (Sklearn): 1.2.2 [57]
- TensorFlow: 2.12.0 [58]

3.2 Preprocessing

3.2.1 Redefining Classes

The choice of emotional labels is essential to train a good classification model for emotion recognition. As discussed in Section 2.1.3, emotions are often measured in valence and arousal. In the ASCERTAIN dataset valence ranges from 0 to 6, and arousal ranges from -3 to 3. The emotional states were chosen to be defined as either high or low for both valence and arousal. For valence, 0 to 4 was mapped to low, and 5 and 6 were considered high. The same was done for arousal, where the range -3 to 0 was mapped as low, and 1 to 3 was considered high. This was also done for the MAHNOB dataset. Both valence and arousal is ranging from 1 to 9, where 1 to 4 was mapped to low and 5 to 9 considered high. The redefining of the emotional states resulted in the following four labels, which were then used for the classification task:

- Class 1: Low Arousal and Low Valence (LALV), including emotions like depression and sad.
- Class 2: Low Arousal and High Valence (LAHV), including emotions like relaxed and calm.
- Class 3: High Arousal and Low Valence (HALV), including emotions like anger and fear.
- Class 4: High Arousal and High Valence (HAHV), including emotions like happy and excited.

In Table 3.1, the distribution of the four classes is shown for ASCERTAIN and MAHNOB. The ASCERTAIN dataset was imbalanced with only 77 samples of the class "HALV" and the majority class being "LAHV" with 735 samples. In the MAHNOB dataset, the "HAHV" class had 203 samples, while the "LALV" class only had 67 samples. For the models to learn how to classify all classes correctly, the data was oversampled using the Random oversampling technique. After the data were oversampled, the ASCERTAIN dataset contained 735 samples for all classes, and the MAHNOB dataset contained 203 samples for all classes.

Table 3.1: Distribution of samples in their respective classes for the ASCERTAIN and MAHNOB datasets.

Dataset	**LALV**	**LAHV**	**HALV**	**HAHV**
ASCERTAIN	571	736	77	571
MAHNOB	67	135	122	203

In the table, the following abbreviations are used: LALV: Low Arousal Low Valence, LAHV: Low Arousal High Valence, HALV: High Arousal Low Valence, HAHV: High Arousal High Valence.

3.2.2 Pan-Tompkins

When measuring physiological signals, the length of the signals represents time in milliseconds. As the length of stimulation varies for each video clip and person, the length of the signals is not equal for all experiments. When feeding signals into a machine learning algorithm, all signals need to be a fixed length. The PT QRS-detection algorithm was applied to the signals to deal with this issue. The PT algorithm is a denoising method detecting the QRS complexes of the ECG signals, as discussed in Section 2.3.2. In this study, the algorithm is also used for resampling the signals to get all the signals to the same length. When the signals are resampled with the PT algorithm, the length of the signals represents the frequency in hertz (Hz). For both the ASCERTAIN and MAHNOB datasets, frequencies 256 Hz and 512 Hz were explored to see how the frequencies affect emotion recognition. Figure 3.1 shows two subplots that display the raw signals in the subfigure above and the signals from the same example after the PT algorithm was applied in the subfigure below. In the first subplot, the raw signals are very noisy and range from 0 to 22923 milliseconds. In the second subplot, the signals are processed using the PT algorithm. The processed signals are less noisy and are downsampled to a signal length of 256.

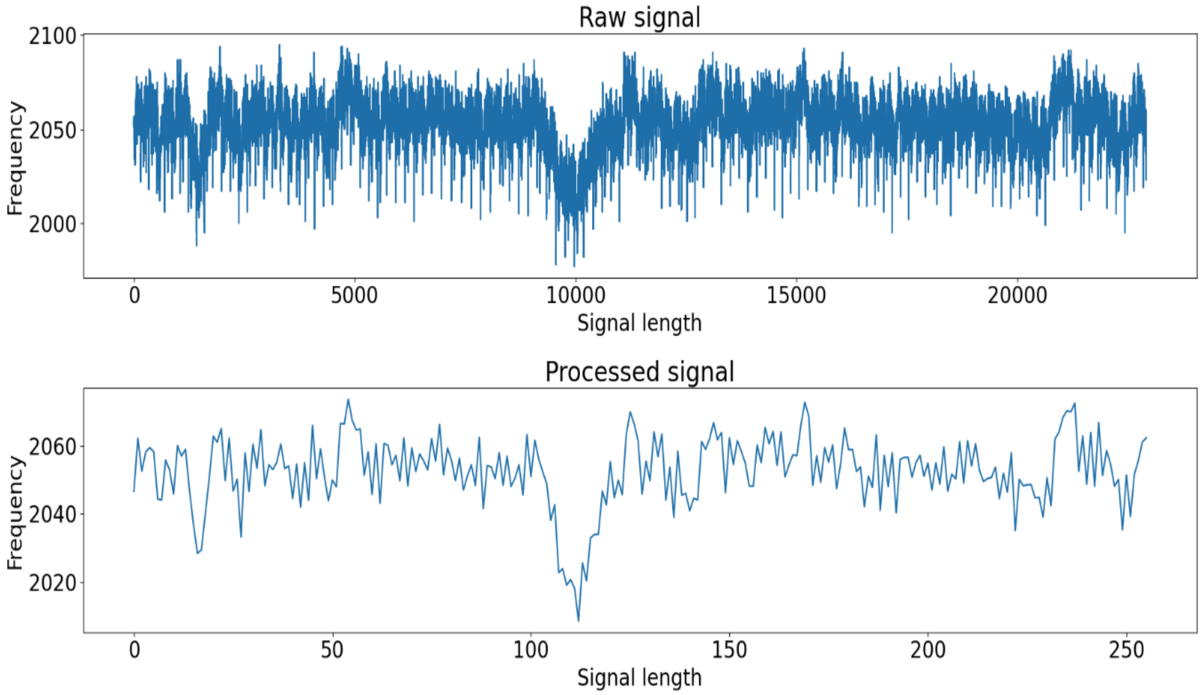


Figure 3.1: Two subplots showing examples of raw signals (above) and processed signals (below), from the ASCERTAIN dataset. The processed signals have been denoised and resampled to 256 Hz using the PT QRS-detection algorithm.

3.2.3 Splitting the Data

Choosing the size of training, validation, and test sets is an assessment that is based on the problem at hand and the number of samples available. The validation and test sets are often in the range of 10% to 40% of the dataset. When the dataset contains fewer samples, it is common to split the data where the validation and test sets are closer to 35% to ensure that the models are evaluated properly [42]. When leaving this many samples out of the training process, the models might have difficulty learning the patterns in the data. For the problem of limited training data, a common solution is splitting the data into train and test sets and using a grid search with cross-validation to identify the optimal hyperparameters for the classifier.

The ASCERTAIN and MAHNOB datasets have 2940 and 812 samples, respectively, after oversampling to get balanced class distributions. The test set was chosen to be a subset consisting of 25% of the samples. The data were then further split into the validation and training sets, where the validation set was chosen to be 35% of the remaining samples after the first split. The data were shuffled to ensure a randomized split. Additionally, the data were evenly distributed by class in the training, validation, and test sets using a stratification parameter. The training set was used to train the model, and the validation set was used to validate the model's performance. The test set was not used until the training of the model was completed to keep it fully unknown to the model.

3.3 Pipelines

In this section, three different pipelines will be presented. Figure 3.2 illustrates the steps of the three pipelines in a flowchart. As previously discussed, all pipelines have been preprocessed by applying the PT algorithm, and the datasets have been split the same way in all cases. The pipelines are also all tested with the two datasets discussed in 4.1. The three pipelines have been given a name to easier refer to them throughout this thesis. PT-SVC is the first pipeline consisting of the processed signals standardized and classified using an SVC. PT-AE-SVC is the second pipeline that has been normalized and fed to an Autoencoder network before classifying using an SVC. The last pipeline, named PT-SAE-SVC, contains the same steps as the second pipeline but with a Sparse Autoencoder network for feature extraction. All pipelines are used for both the ASCERTAIN and MAHNOB datasets with resampled frequencies of 256 Hz and 512 Hz.

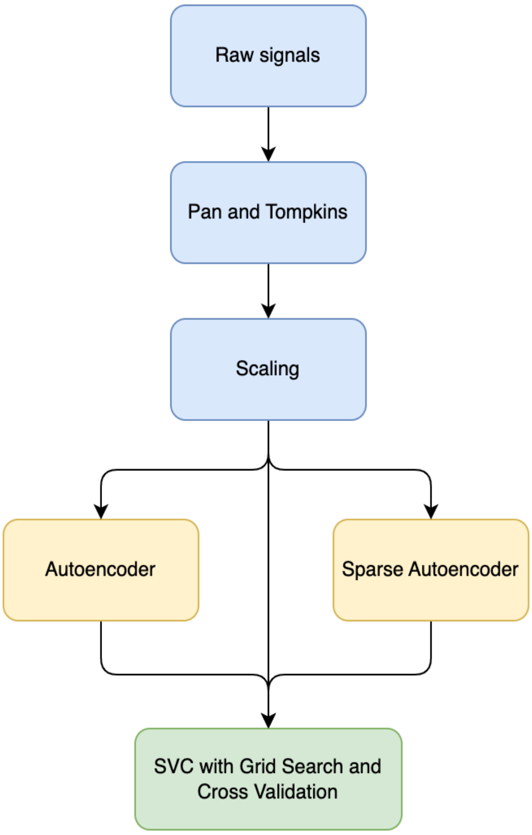


Figure 3.2: The steps of the three pipelines presented in this section from raw data to classification. The first pipeline is the PT-SVC pipeline which uses the scaled signals as input for the SVC. The second pipeline is the PT-AE-SVC, encoding the scaled signals using an Autoencoder followed by an SVC. The third pipeline is the PT-SAE-SVC using an SVC to classify the encoded data from a Sparse Autoencoder.

3.3.1 PT-SVC

In the PT-SVC pipeline, the processed signals are standardized using a standard scaler. The standard scaler scales the data independently on each feature to ensure a mutual scale for all features. Standardizing the data helps minimize any bias occurring from features having an extensive range of values. The standardization is calculated using the statistical Z-score normalization shown in equation 3.1. In the equation, Z represents the standard score of sample x, μ is the mean, and σ is the standard deviation of each sample. This calculation provides a mean of zero and a unit variance for all features. This step is especially important when using an SVC to classify the data. The classifier assumes that the data are centered around 0 and have the same variance for all features.

$$Z = \frac{x - \mu}{\sigma} \tag{3.1}$$

The standardized data is fed into an SVC. The SVC is trained using a grid search with cross-validation. The purpose of the grid search is to find the optimal parameters by cross-validating a parameter grid containing various hyperparameters. With cross-validation, the data is split into five subsets where one of the subsets is withheld in training to be used as validation. This process is repeated for all subsets to be used as validation. Furthermore, the cross-validation process is repeated for each combination of hyperparameter values in the parameter grid. The hyperparameters included in the grid search were the regularization parameter C, the parameter for stopping criterion tolerance, and the kernel coefficient gamma. As discussed in Section 2.4.1, there are several different kernel functions that can be implemented to handle nonlinearity in the data. The linear, polynomial, sigmoid, and RBF kernels were all tested and cross-validated in the grid search.

As previously mentioned, the pipeline is used on two different datasets, namely ASCERTAIN and MAHNOB. In Table 3.2, the minimum and maximum values for the chosen SVC hyperparameters are listed for the two datasets and the two frequencies of the signals. Table 3.3 shows the optimal combinations of hyperparameters for the four SVCs. The optimal choices of kernels are not listed as the RBF kernel was the best choice for all classifiers. The best combination for the ASCERTAIN signals resampled to 256 Hz was a C value of 0.01, a stopping criterion tolerance of 2, and a gamma of 48. With a frequency of 512 Hz, the best combination for the ASCERTAIN signals was a C value of 0.5, a stopping criterion tolerance of 2, and a gamma value of 34. With the MAHNOB data, the signals with a 256 Hz frequency showed the best performance with a C value of 2, a stopping criterion tolerance of 2, and a gamma value of 103. Lastly, the optimal combination for the MAHNOB signals resampled to 512 Hz was a C value of 0.001, a stopping criterion tolerance of 2, and a gamma value of 7.

Table 3.2: The tested ranges of parameter values for the SVCs trained with the signals preprocessed using the PT algorithm and a standard scaler.

Parameter	**Minimum**	**Maximum**
C	0.001	100
Tolerance	0.01	100
Gamma	0.001	100

3.3.2 PT-AE-SVC

The PT-AE-SVC pipeline normalizes the processed signals using a min-max scaler before using an Autoencoder for feature extraction and, lastly, an SVC to classify the signals. The min-max scaler scales the data to a range from zero to one. Like the standardizer used in the PT-SVC

Table 3.3: The optimal parameter values in the SVCs trained with the signals preprocessed using the PT algorithm and a standard scaler.

Parameter	ASC 256	ASC 512	MAH 256	MAH 512
C	0.01	0.5	2	0.001
Tolerance	2	2	2	2
Gamma	48	34	103	7

In the table the following abbreviations are used: ASC 256: ASCERTAIN dataset with a frequency of 256 Hz, ASC 512: ASCERTAIN dataset with a frequency of 512 Hz, MAH 256: MAHNOB dataset with a frequency of 256 Hz, MAH 512: MAHNOB dataset with a frequency of 512 Hz.

pipeline, the goal is to scale all the features to the same range of values. This can be essential for the Autoencoder to converge during training and improve its generalization. Equation 3.2 shows the equation used to calculate the normalized output. In the equation, X represents the sample, X_{min} is the minimum sample value across the given feature, and X_{max} is the maximum sample value.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.2}$$

The normalized data is used as input for the Autoencoder. As discussed in Section 2.4.2, Autoencoder is an unsupervised machine learning algorithm. As it is used to recognize patterns in the data by mapping the data to a new feature space, it can be used as a feature extraction method for supervised learning. To build the architecture for the Autoencoder, there are different layers to add where adding new layers increase the complexity of the model. Because of the difference in frequency in the two datasets, two separate Autoencoders were trained. Both models were constructed with an input layer, two hidden layers, one bottleneck layer, and an output layer. The full architecture for the Autoencoders is shown in Figure 3.3. The output from the bottleneck layer is encoded data used for feature extraction. The rest of the network is constructed to validate the model's performance while building the architecture and tuning the hyperparameters.

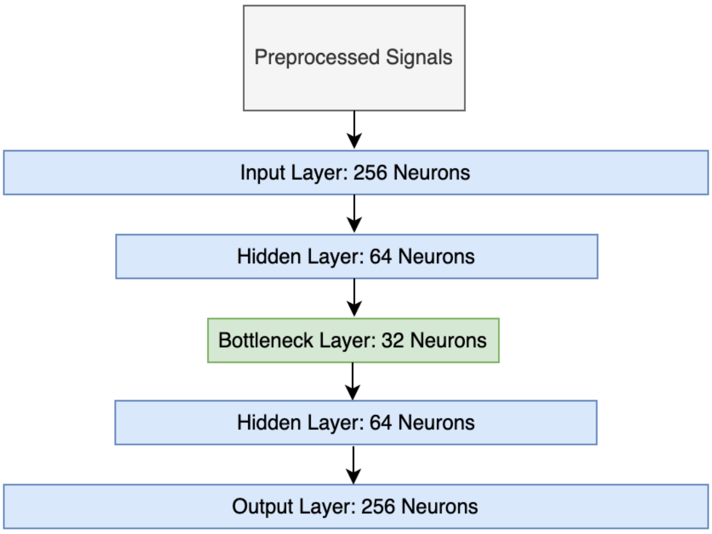


Figure 3.3: The architecture of the Autoencoders used for feature extraction.

The first layer is the input layer, where the input data is processed. The second layer is the first hidden layer using neurons to learn from the input layer. The first hidden layer is the encoder producing an encoded lower dimensional representation of the input data. The third layer, called the bottleneck layer, compresses the data into a smaller feature space containing the most relevant information of the input data. After the bottleneck layer, there is another hidden layer, which is the decoding part of the network. As discussed in 2.4.2, the decoder uses the information from the bottleneck layer to reproduce the input data. The data is mapped back to the initial feature space in the output layer. While the architecture for the two datasets was chosen to be the same, they do not necessarily have the same optimal values for the hyperparameters or number of neurons. Different combinations of neurons were tested to find the optimal number of neurons in the hidden layers and the bottleneck layer. In Table 3.4, the ranges of values for the number of neurons and the parameters are listed.

Table 3.4: The tested ranges of the number of neurons and parameter values for the Autoencoder network.

Parameter	**Minimum**	**Maximum**
Hidden Neurons	8	128
Bottleneck Neurons	8	64
Learning Rate	0.00001	0.1
Batch Size	2	16
Number of Epochs	30	200

The different datasets and frequencies were tuned separately to find the optimal combinations of neurons and parameter values. The optimal values for the respective datasets and frequencies are shown in Table 3.5. Both Autoencoders for the ASCERTAIN signals performed best with 64 hidden neurons and 32 bottleneck neurons. For the signals with 256 Hz frequency, the optimal combination of parameter values was a learning rate of 0.0005, batch size of 2 and 100 epochs. With a frequency of 512 Hz, the optimal combination was a learning rate of 0.0007, a batch size of 3, and 80 epochs. The MAHNOB dataset also performed better with 64 hidden neurons and 32 bottleneck neurons for both signal frequencies. However, with a 256 Hz frequency, the best combination of parameter values was a learning rate of 0.0005, batch size of 2, and 90 epochs. With a frequency of 512 Hz, the learning rate was also 0.0005, batch size was 3, and 100 epochs.

Table 3.5: The optimal number of neurons and parameter values in the Autoencoder network.

Parameter	**ASC 256**	**ASC 512**	**MAH 256**	**MAH 512**
Hidden Neurons	64	64	64	64
Bottleneck Neurons	32	32	32	32
Learning Rate	0.0005	0.0007	0.0005	0.0005
Batch Size	2	3	2	3
Number of Epochs	100	80	90	100

In the table the following abbreviations are used: ASC 256: ASCERTAIN dataset with a frequency of 256 Hz, ASC 512: ASCERTAIN dataset with a frequency of 512 Hz, MAH 256: MAHNOB dataset with a frequency of 256 Hz, MAH 512: MAHNOB dataset with a frequency of 512 Hz.

The features extracted from the Autoencoder were used to train an SVC to predict the respective emotions. Four different classifiers were trained, one for each set of features extracted with the Autoencoder models. Like in the PT-SVC pipeline, the SVC was trained using a grid search with cross-validation to find the optimal hyperparameter values. The hyperparameters chosen to include in the parameter grid were C, stopping criterion tolerance, and gamma, as shown in Table 3.6. In the table, the range of values that were tested in the grid search is listed.

Table 3.6: The tested ranges of parameter values for the SVCs trained with the features extracted using the Autoencoder.

Parameter	Minimum	Maximum
C	0.001	100
Tolerance	0.01	100
Gamma	0.001	100

The optimal combinations of parameter values for the four classifiers are presented in Table 3.7. For the first classifier for the ASCERTAIN signals with a frequency of 256 Hz, the optimal combination of hyperparameter values was a C value of 1, a stopping criterion tolerance of 1, and a gamma value of 130. Next, the optimal combination for the signals with a frequency of 512 Hz was a C value of 5, a stopping criterion tolerance of 1, and a gamma value of 112. For the MAHNOB signals with a frequency of 256 Hz, the best SVC hyperparameter value combination was a C value of 1, a stopping criterion tolerance of 1, and a gamma value of 11. Lastly, for the signals with a frequency of 512 Hz, the best combination was a C value of 1, a stopping criterion tolerance of 2, and a gamma value of 7.

Table 3.7: The optimal parameter values in the SVCs trained with the features extracted using the Autoencoder.

Parameter	ASC 256	ASC 512	MAH 256	MAH 512
C	1	5	1	1
Tolerance	1	1	1	2
Gamma	130	112	11	7

In the table the following abbreviations are used: ASC 256: ASCERTAIN dataset with a frequency of 256 Hz, ASC 512: ASCERTAIN dataset with a frequency of 512 Hz, MAH 256: MAHNOB dataset with a frequency of 256 Hz, MAH 512: MAHNOB dataset with a frequency of 512 Hz.

3.3.3 PT-SAE-SVC

The last pipeline, namely PT-SAE-SVC, feeds the processed signals to a Sparse Autoencoder encoding the signals as discussed in 2.4.2. The Sparse Autoencoder works the same way as the Autoencoder by encoding the signals and trying to reproduce the original signals by decoding them. The output from the encoder is used for feature extraction, and the decoder output is used to evaluate the model. The difference between the Autoencoder and the Sparse Autoencoder is the sparsity added to the Sparse Autoencoder. Instead of compressing the data into a smaller feature space, it contains more or the same number of neurons in the hidden layers and the bottleneck layer as input units. The bottleneck layer includes a regularization term to counter the potential drawbacks of high sparsity. The regularization term is an L1-regularization applied to the neurons in the bottleneck layer. Figure 3.4 shows the architecture used to train the Sparse Autoencoder. The architecture is constructed with an input layer that passes the data to a hidden layer. The output from the hidden layer is then passed to the bottleneck layer, where the sparse representation of the input data is penalized by the L1-regularization. Next, the data is reproduced with a hidden layer which passes the data to the output layer.

Table 3.8 lists the ranges for the different parameter values tested for the Sparse Autoencoder.

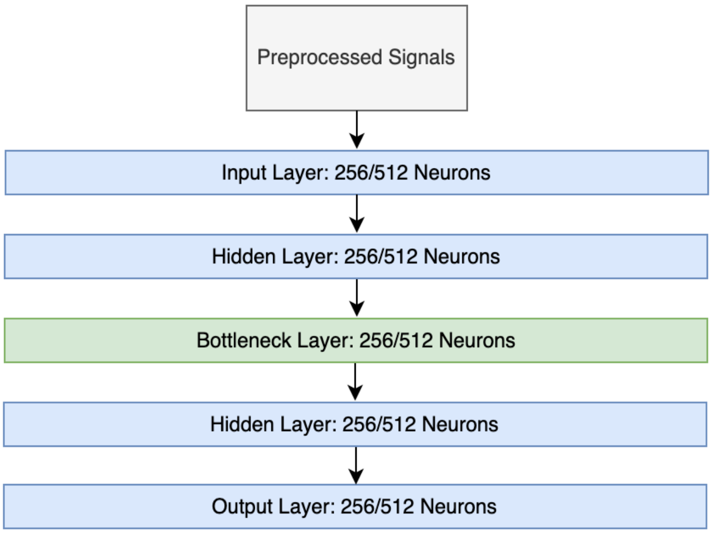

Preprocessed Signals
Input Layer: 256/512 Neurons
Hidden Layer: 256/512 Neurons
Bottleneck Layer: 256/512 Neurons
Hidden Layer: 256/512 Neurons
Output Layer: 256/512 Neurons

Figure 3.4: The architecture of the Sparse Autoencoders used for feature extraction. The number of neurons in each layer depends on the signals' frequencies.

With the Sparse Autoencoder networks, the number of hidden neurons and bottleneck neurons was chosen to be the same as the frequencies of the signals. The optimal parameter values for the four Sparse Autoencoders are presented in Table 3.9. The optimal combination for the ASCERTAIN dataset with a frequency of 256 Hz was a learning rate of 0.0008, batch size of 2, and 80 epochs. Having a frequency of 512 Hz, the optimal combination was a learning rate of 0.0005, batch size of 2 and 100 epochs. For the MAHNOB dataset, the 256 Hz frequency signals had a learning rate of 0.0004, batch size of 3, and 100 epochs as the optimal combination. With a frequency of 512 Hz, the optimal combination was a learning rate of 0.0006, batch size of 2, and 120 epochs.

Table 3.8: The tested ranges of the number of neurons and parameter values for the Sparse Autoencoder network.

Parameter	**Minimum**	**Maximum**
Learning Rate	0.00001	0.1
Batch Size	2	16
Number of Epochs	30	200

Table 3.9: The optimal number of neurons and parameter values in the Autoencoder network.

Parameter	**ASC 256**	**ASC 512**	**MAH 256**	**MAH 512**
Hidden Neurons	256	512	256	512
Bottleneck Neurons	256	512	256	512
Learning Rate	0.0008	0.0005	0.0004	0.0006
Batch Size	2	2	3	2
Number of Epochs	80	100	100	120

In the table the following abbreviations are used in the table: ASC 256: ASCERTAIN dataset with a frequency of 256 Hz, ASC 512: ASCERTAIN dataset with a frequency of 512 Hz, MAH 256: MAHNOB dataset with a frequency of 256 Hz, MAH 512: MAHNOB dataset with a frequency of 512 Hz.

The outputs of the Sparse Autoencoders are used to train four different SVCs, one for each set of output. Like the previously discussed SVCs, the classifiers are trained using a grid search and cross-validation. They are also finetuned using the same hyperparameters chosen for the other SVCs, namely C, stopping criterion tolerance, and gamma. In Table 3.10, the ranges of values that were evaluated in the grid searches are listed. The optimal values for the hyperparameters of each SVC are listed in Table 3.11. For the ASCERTAIN signals with a frequency of 256 Hz, the optimal parameter combination was a C value of 76, a stopping criterion tolerance of 2, and a gamma value of 86. With a frequency of 512 Hz, the optimal combination was a C value of 96, a stopping criterion tolerance of 2, and a gamma value of 94. The best combination for the MAHNOB signals with a frequency of 256 Hz was a C value of 66, a stopping criterion tolerance of 1, and a gamma of 7. Lastly, with a frequency of 512 Hz, the best combination was a C value of 70, a stopping criterion tolerance of 1, and a gamma value of 9.

Table 3.10: The tested ranges of parameter values for the SVCs trained with the features extracted using the Sparse Autoencoder.

Parameter	**Minimum**	**Maximum**
C	0.001	100
Tolerance	0.01	100
Gamma	0.001	100

Table 3.11: The optimal parameter values in the SVCs trained with the features extracted using the Sparse Autoencoder.

Parameter	**ASC 256**	**ASC 512**	**MAH 256**	**MAH 512**
C	76	96	66	70
Tolerance	2	2	1	1
Gamma	86	94	7	9

In the table the following abbreviations are used in the table: ASC 256: ASCERTAIN dataset with a frequency of 256 Hz, ASC 512: ASCERTAIN dataset with a frequency of 512 Hz, MAH 256: MAHNOB dataset with a frequency of 256 Hz, MAH 512: MAHNOB dataset with a frequency of 512 Hz.

Chapter 4

Results

This chapter presents the results of the experiments conducted in this study. Furthermore, the two benchmark datasets, namely ASCERTAIN and MAHNOB, will be introduced. The results include the performance of SVCs with PT-processed signals and features extracted with Autoencoders and Sparse Autoencoders as input. Using the pipelines, the goal is to answer the two research questions stated in Chapter 1, namely:

- RQ1: To what extent can the latent space captured by an Autoencoder and a Sparse Autoencoder explain the variability in ECG data?
- RQ2: What is the overall performance of the proposed pipelines compared to state-of-the-art, and how far can ECG be used as a single modality for HER?

4.1 Datasets

In this thesis, two public multimodal datasets (ASCERTAIN and MAHNOB) containing physiological signals will be used to compare different systems for HER. Both datasets contain ECG signals, which will be the physiological signals used to predict the subjects' emotional states.

4.1.1 ASCERTAIN

ASCERTAIN is a multimodal database for implicit personality and affect recognition using commercial physiological signals and was collected by R. Subramanian et al. [59]. The database contains ECG, EEG, GSR, and facial activity data of 58 subjects, in addition to big-five personality scales and emotional self-ratings. The subjects were all university students, where 21 were females, and 37 were males. They watched 36 affective movie clips while wearing off-the-shelf sensors to record the physiological signals and facial activity data. For self-rating, the participant ranked each video's level of arousal and valence, engagement, liking, and familiarity. For emotion recognition, only the arousal and valence self-reports are used, where they ranked their arousal and valence from -3 to 3 and 0 to 6, respectively. While there are several different signals, the ECG signals will be the focus of this study. The sampling frequency for the ECG signals was 256 Hz, measured on both the right and the left arm. For this study, only the signals from the left arm are used to recognize the subjects' emotional states.

4.1.2 MAHNOB

MAHNOB is, like ASCERTAIN, a multimodal database, and it was collected by M. Soleymani et al. [60]. The experiments were conducted on 30 young and healthy adults, where 17 were females, and 13 were males. The subjects watched 20 emotional video clips while their physiological signals were measured using the Biosemi Active 2 system. The emotions were evaluated using self-assessment manikins (SAM) questionnaires, where they selected their own perception on the level of valence and arousal [61]. SAM is a questionnaire visualizing the valence and arousal dimensions through manikins, as illustrated in Figure 4.1. Both valence and arousal range from 1 to 9, representing negative to positive and low to high, respectively. As in the ASCERTAIN database, there are several types of signals, but only the ECG signals are extracted for this study. They were recorded with three electrodes strapped to the chest's upper right and left corners and one below the last rib on the abdomen. Only the signals from the upper left part of the chest are considered in this paper. The signals were collected with a sampling rate of 1024 Hz but later downsampled to 256 Hz.

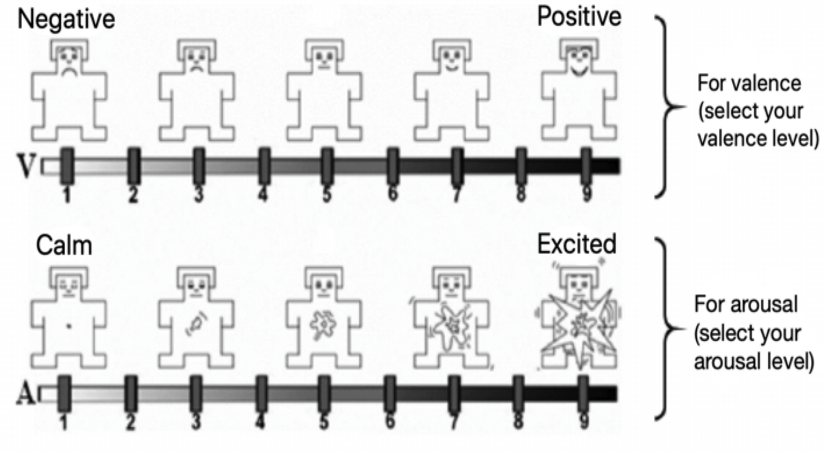

Negative
Positive
For valence (select your valence level)
V
1
2
3
4
5
6
7
8
9
Calm
Excited
For arousal (select your arousal level)
A
1
2
3
4
5
6
7
8
9

Figure 4.1: Illustration of SAM where the subjects selects their level of valence (above) and level of arousal (below) [61].

4.2 Evaluation Metrics

To evaluate the models properly, four scoring metrics were used: accuracy, precision, recall, and F1-score. Accuracy is a measure calculating the percentage of samples classified correctly, and it is calculated as shown in Equation 4.1. The calculation in Equation 4.2 determines the model's precision, which measures the number of true positives predicted correctly. Recall, also known as true positive rate, measures the model's ability to identify the positive samples in the dataset correctly. The calculation for recall is defined in Equation 4.3. Lastly, F1-score is a combination of precision and recall where the weighted averages of the two measures are calculated. The F1-score is calculated as illustrated in Equation 4.4.

The number of true positives, true negatives, false positives, and false negatives are used to calculate the evaluation metrics. True positives are the number of positives accurately predicted to be positive. True negatives are the number of negatives accurately predicted to be negative. False positives are negatives predicted falsely to be positives. False negatives are positives predicted falsely to be negatives.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{4.1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.3}$$

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4.4}$$

4.3 Presenting the Results

To answer the first research question, the results from the three pipelines will be compared. The PT-SVC pipeline acts as a baseline to observe how the results change when Autoencoders and Sparse Autoencoders are used to extract features. Tables 4.1 and 4.2 show the performance scores from the two first pipelines, PT-SVC and PT-AE-SVC, respectively. PT-AE-SVC with the Autoencoder provides generally higher accuracy, recall, and F1-score, except for the MAHNOB signals with a frequency of 512 Hz.

Table 4.1: Training and test accuracy, precision, recall, and F1-score from the PT-SVC pipeline with PT processed signals and an SVC.

Dataset	Training Accuracy	Test Accuracy	Precision	Recall	F1-Score
ASCERTAIN 256 Hz	56%	57%	73%	57%	51%
ASCERTAIN 512 Hz	56%	56%	68%	56%	51%
MAHNOB 256 Hz	60%	62%	85%	62%	62%
MAHNOB 512 Hz	63%	62%	62%	62%	61%

Table 4.2: Training and test accuracy, precision, recall, and F1-score from the PT-AE-SVC pipeline with PT, Autoencoder and SVC.

Dataset	Training Accuracy	Test Accuracy	Precision	Recall	F1-Score
ASCERTAIN 256 Hz	56%	59%	62%	59%	57%
ASCERTAIN 512 Hz	56%	59%	63%	59%	58%
MAHNOB 256 Hz	62%	64%	79%	64%	64%
MAHNOB 512 Hz	63%	58%	58%	58%	58%

Table 4.3: Training and test accuracy, precision, recall, and F1-score from the PT-SAE-SVC pipeline with PT, Sparse Autoencoder and SVC.

Dataset	Training Accuracy	Test Accuracy	Precision	Recall	F1-Score
ASCERTAIN 256 Hz	50%	50%	48%	50%	49%
ASCERTAIN 512 Hz	50%	46%	45%	46%	45%
MAHNOB 256 Hz	61%	60%	59%	60%	59%
MAHNOB 512 Hz	58%	51%	51%	51%	50%

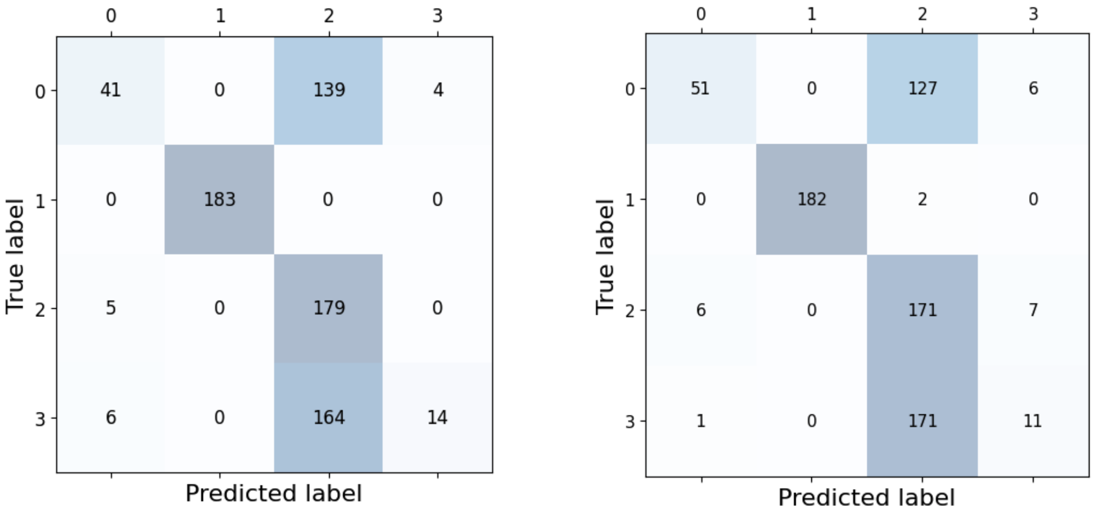


Figure 4.2: Confusion matrices of the classification results from SVC with signals processed with the PT algorithm. The signals are from the ASCERTAIN dataset with frequencies of 256 Hz (left) and 512 Hz (right). The labels 0, 1, 2 and 3 represent the classes HAHV, HALV, LAHV, and LALV, respectively.

In Figures 4.2 - 4.5, the confusion matrices for the two pipelines are presented, showing the explained variability for each class for the given datasets and signal frequencies. The main difference is the added generalization from the Autoencoder. For the PT-SVC pipeline, the confusion matrices show a clear bias towards one class with both frequency levels for ASCERTAIN and a frequency of 256 Hz for MAHNOB. The matrices for the ASCERTAIN dataset show a bias towards the ”LAHV” class, and the matrix for MAHNOB dataset with 256 Hz frequency shows a bias towards the ”LALV” class. In PT-AE-SVC, the mentioned biases are still apparent but less significant. Next, the results from the PT-SAE-SVC pipeline, listed in Table 4.3, show lower scores than the two other pipelines for all metrics. The confusion matrices for the PT-SAE-SVC pipeline are shown in Figures 4.6 and 4.7. The figures show that adding the sparsity to the feature extraction further reduces the bias towards one class, as in the PT-SVC and PT-AE-SVC pipelines. In the confusion matrices with the ASCERTAIN dataset, the “HALV” class has almost all samples correctly predicted. The remaining samples are evenly spread with only a slight bias towards the “HAHV” class.

In Section 1.3, the state-of-the-art for HER using ECG signals was presented. When comparing the results, there are several aspects to consider. One important aspect is the dataset used for the emotion recognition task. Table 1.2 lists studies using ECG signals from ASCERTAIN and MAHNOB. Another important consideration is the target emotions used to explain the subjects’ emotional state. All related work from the table predicted arousal and valence separately, while this study combined the two emotional states. The highest accuracies reported for the

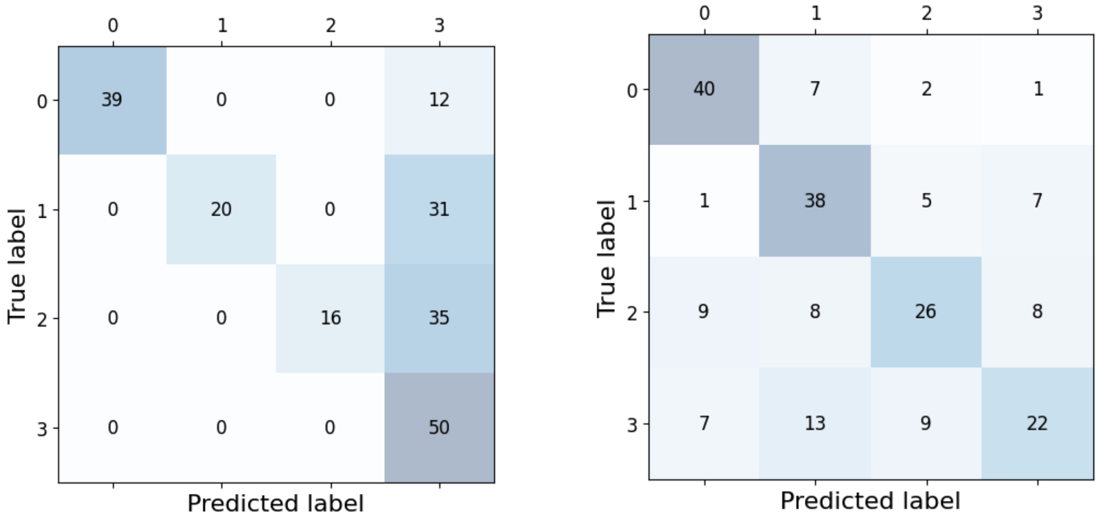


Figure 4.3: Confusion matrices of the classification results from SVC with signals processed with the PT algorithm. The signals are from the MAHNOB dataset with frequencies of 256 Hz (left) and 512 Hz (right). The labels 0, 1, 2 and 3 represent the classes HAHV, HALV, LAHV, and LALV, respectively.

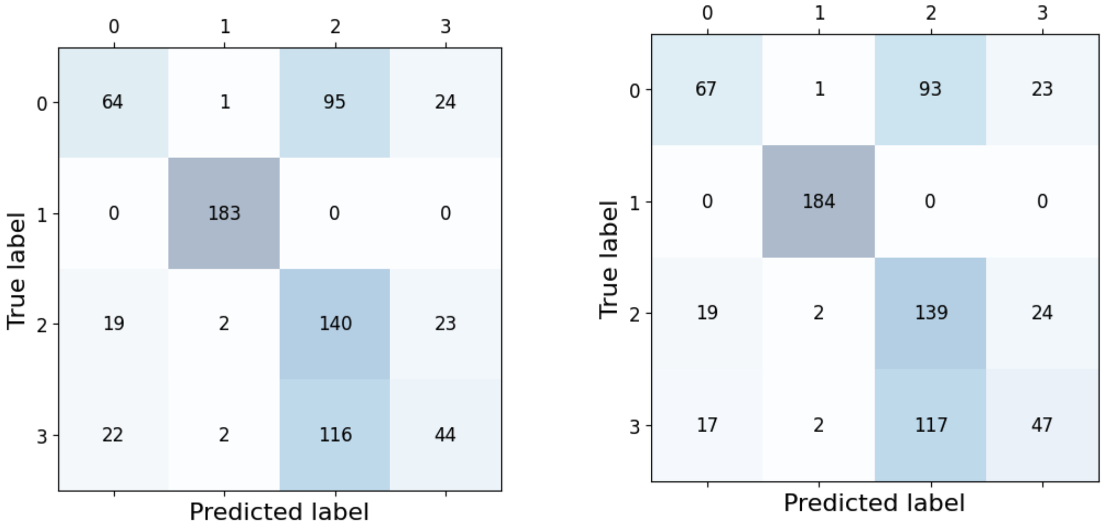
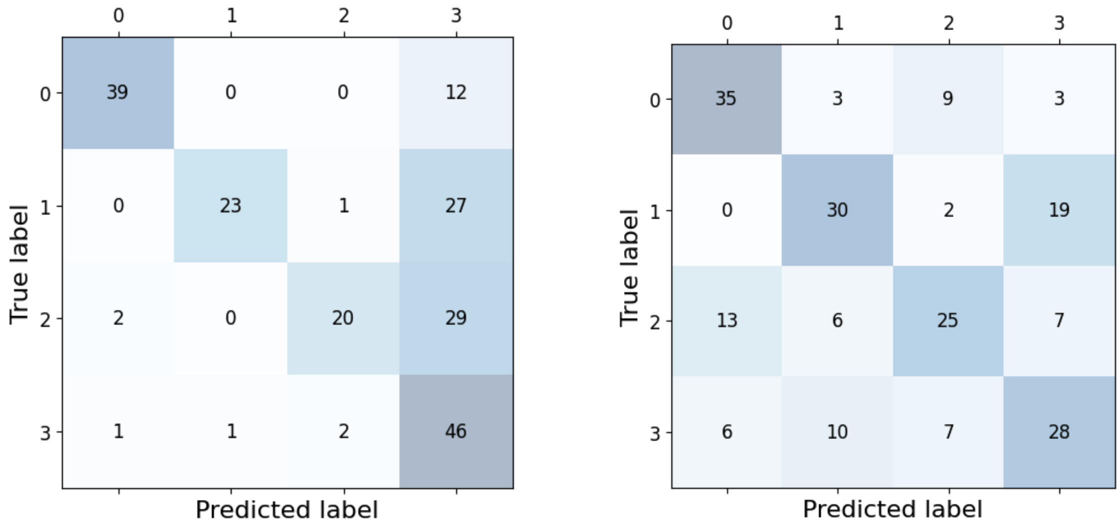


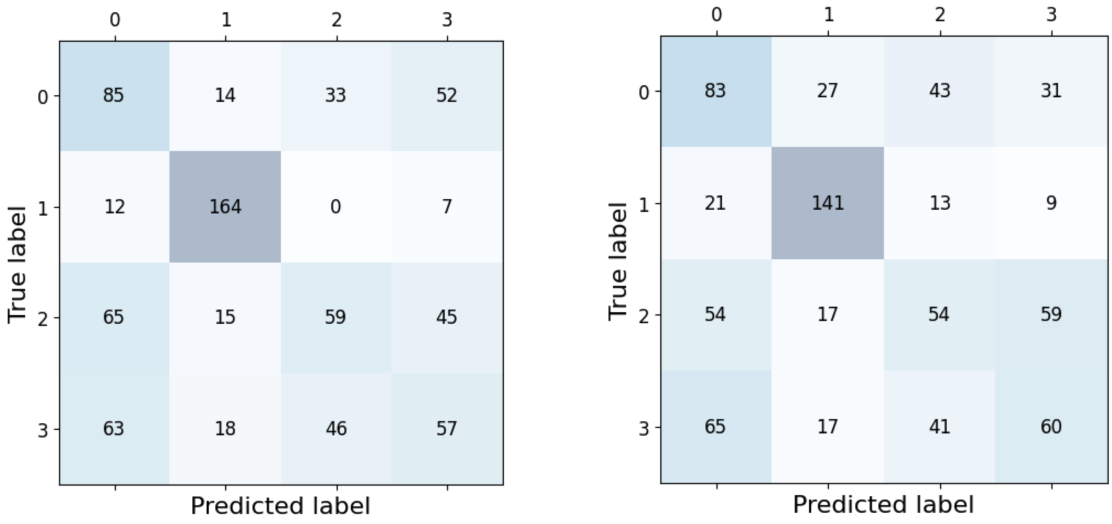
Figure 4.4: Confusion matrices of the classification results from SVC with features extracted using an Autoencoder. The signals are from the ASCERTAIN dataset with frequencies of 256 Hz (left) and 512 Hz (right). The labels 0, 1, 2 and 3 represent the classes HAHV, HALV, LAHV, and LALV, respectively.



	0	1	2	3
0	39	0	0	12
1	0	23	1	27
2	2	0	20	29
3	1	1	2	46

	0	1	2	3
0	35	3	9	3
1	0	30	2	19
2	13	6	25	7
3	6	10	7	28

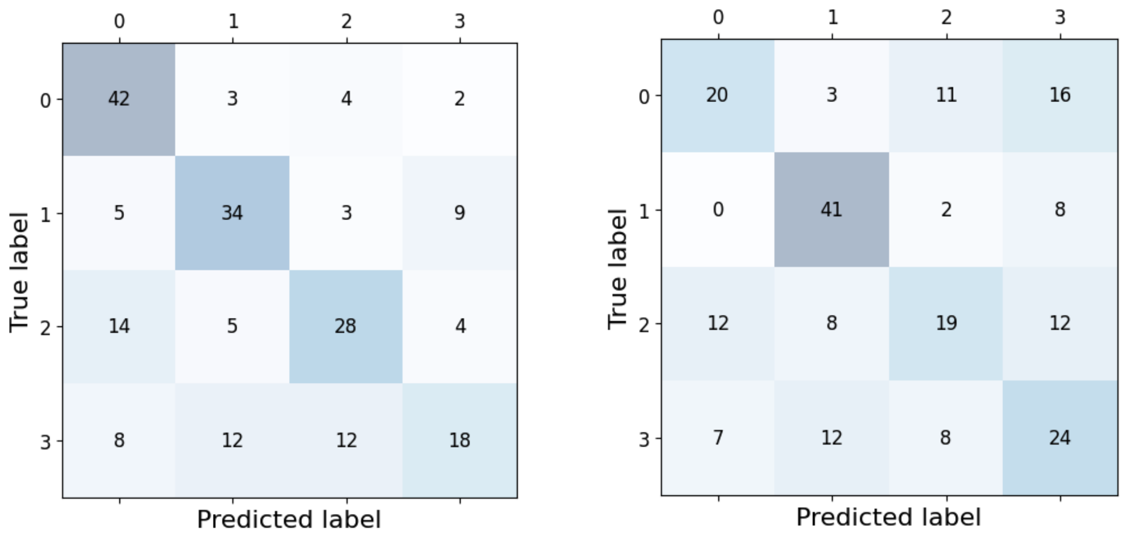
Figure 4.5: Confusion matrices of the classification results from SVC with features extracted using an Autoencoder. The signals are from the MAHNOB dataset with frequencies of 256 Hz (left) and 512 Hz (right). The labels 0, 1, 2 and 3 represent the classes HAHV, HALV, LAHV, and LALV, respectively.



	0	1	2	3
0	85	14	33	52
1	12	164	0	7
2	65	15	59	45
3	63	18	46	57

	0	1	2	3
0	83	27	43	31
1	21	141	13	9
2	54	17	54	59
3	65	17	41	60

Figure 4.6: Confusion matrices of the classification results from SVC with features extracted using a Sparse Autoencoder. The signals are from the ASCERTAIN dataset with frequencies of 256 Hz (left) and 512 Hz (right). The labels 0, 1, 2 and 3 represent the classes HAHV, HALV, LAHV, and LALV, respectively.



	0	1	2	3
0	42	3	4	2
1	5	34	3	9
2	14	5	28	4
3	8	12	12	18

	0	1	2	3
0	20	3	11	16
1	0	41	2	8
2	12	8	19	12
3	7	12	8	24

Figure 4.7: Confusion matrices of the classification results from SVC with features extracted using a Sparse Autoencoder. The signals are from the MAHNOB dataset with frequencies of 256 Hz (left) and 512 Hz (right). The labels 0, 1, 2 and 3 represent the classes HAHV, HALV, LAHV, and LALV, respectively.

ASCERTAIN dataset were 78.7% and 78.3% for arousal and valence, respectively. With the MAHNOB dataset, the highest accuracies were 66.4% for arousal and 66% for valence. In this study, the highest accuracies achieved were 59% and 64% for the ASCERTAIN and MAHNOB datasets, respectively.

4.4 Summarizing the Findings

From the results of the three pipelines, it appears that the PT-SVC pipeline and the PT-AE-SVC pipeline are getting similar scores. The PT-AE-SVC pipeline had slightly better accuracy, recall, and F1-score, while the PT-SVC pipeline got better precision, except for the MAHNOB signals with a frequency of 512 Hz. The confusion matrices show that the pipelines got a prediction bias towards one class, but the bias was reduced with the Autoencoder. With the signals from the ASCERTAIN dataset, the bias was towards the ”LAHV” class, and for the MAHNOB signals with 256 Hz frequency, the bias was towards the ”LALV” class. With the Sparse Autoencoder, the confusion matrices show no apparent bias towards any of the classes but an overall worse classification performance. The results from this study are also compared with state-of-the-art using ECG signals from the same two datasets. This study achieved lower classification accuracy than state-of-the-art, but the target emotions are defined differently.

Chapter 5

Discussion

The three pipelines show different results as presented in Chapter 4. The PT-AE-SVC pipeline generally displays the highest accuracy, recall, and F1-score. The PT-SVC pipeline provides higher precision, meaning it is better at identifying true positives from predicted ones. The MAHNOB signals with a frequency of 512 Hz were an exception showing higher accuracy, recall, and F1-score without feature extraction and higher precision with the Autoencoder. Both the mentioned pipelines are biased towards one class, namely the "LAHV" class with ASCERTAIN data and the "LALV" class with MAHNOB data. The last pipeline using Sparse Autoencoders to encode the signals shows all over worse results from the evaluation metrics. However, the confusion matrices show significantly less bias towards any classes.

5.1 Exploring the Research Questions

In this section, the results will be discussed in context of the research questions:

- RQ1: To what extent can the latent space captured by an Autoencoder and a Sparse Autoencoder explain the variability in ECG data?
- RQ2: What is the overall performance of the proposed pipelines compared to state-of-the-art, and how far can ECG be used as a single modality for HER?

Regarding RQ1, The PT-AE-SVC pipeline achieved better accuracy, recall, and F1-score, except for MAHNOB signals with a frequency of 512 Hz. As the MAHNOB signals with a frequency of 256 Hz showed similar or better scores for all metrics in all three pipelines, it could indicate that the MAHNOB signals should have a frequency of 256 Hz. Furthermore, the confusion matrices displayed a decrease in bias towards the "LAHV" and "LALV" classes with the Autoencoder, which could make the model more generalizable and better able to capture the variability in the data. With the model showing generally better performance in the evaluation metrics while reducing the bias towards the two classes, it could be a contribution to emotion recognition using ECG signals. Even though the Sparse Autoencoder had worse classification performance than both the Autoencoder and not using any feature extraction, it further decreased the bias towards the "LAHV" and "LALV" classes. Using the MAHNOB signals with a frequency of

256 Hz as input; the model showed relatively good results compared with the other pipelines while reducing the biases significantly. These results could offer great potential for the Sparse Autoencoder to help capture the variability in the data. Furthermore, the results could implicate that Sparse Autoencoders are better suited for generative pseudo-sampling.

Figure 5.1 illustrates the latent space of the MAHNOB signals with a frequency of 256 Hz after the Autoencoder was applied. The figure shows the samples in a three-dimensional scatterplot with the latent space mapped to a three-dimensional feature space using Principal Component Analysis. In the figure, the samples are scattered seemingly without any clear pattern regarding the target emotions. The lack of separability between the classes in the figure shows the challenge of predicting the subjects' emotional states. An essential element to consider is that the data is illustrated in only three dimensions. For the model to explain the variability in the data, it might require more than three features from the original 256 features. For the classification, the data is mapped to a 32-dimensional feature space, providing better results when tuning the Autoencoder.

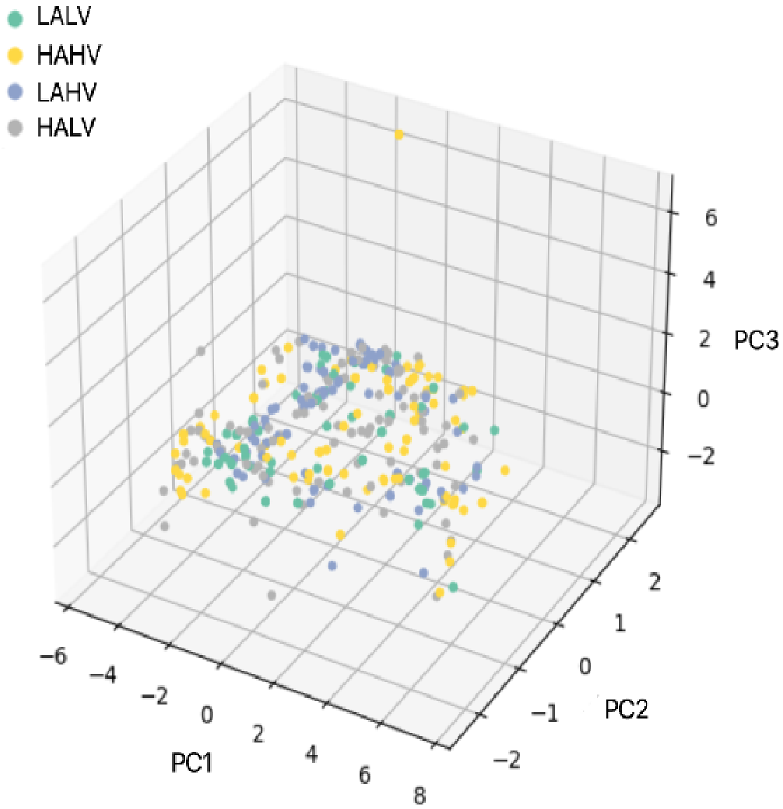


Figure 5.1: The latent space of the samples from MAHNOB with signals of a 256 Hz frequency. The data has been encoded using an Autoencoder. Principal Component Analysis has further reduced the feature space dimensionality to three principal components. The three axes represent the principal components.

In the second research question, the performance of the proposed pipelines is compared to the state-of-the-art. As discussed in Section 4.3, there are several factors to be considered when comparing the results with other studies. One crucial factor is the datasets used to train and test the models. Different datasets have different configurations for the experiments, affecting the models' ability to predict subjects' emotional states. Additionally, the subjects involved will vary, which could significantly impact the ability to predict their emotions. How people react to certain events is individual and can be affected by demographic characteristics like age, gender, or culture. Another essential aspect to consider is the target emotions. As discussed in Chapter 2, there are different ways of defining emotions. In some studies, the target emotions are chosen to be specific emotions like the seven universal emotions, while others use arousal and valence. The different definitions of target emotions make it difficult to compare the results, as it is often easier to achieve higher scores with fewer targets to predict.

Considering that this study has combined arousal and valence to be four target emotions as opposed to the studies listed in Table 1.2 where arousal and valence are predicted separately, the accuracy can be expected to be lower. Looking at the accuracies achieved with the MAHNOB dataset, this study achieved the highest accuracy of 64% while state-of-the-art achieved 66% and 65% for arousal and valence, respectively. Even though this study used four target emotions, the accuracy is still close to state-of-the-art. With the ASCERTAIN dataset, state-of-the-art shows higher accuracy with 78% for both arousal and valence, while this study achieved an accuracy of 59%. Even though it is not as close as with the MAHNOB dataset, a difference of 19% with two additional target emotions is respectable.

As discussed in Chapter 2, several different modalities can be used for HER. These modalities are often combined for a multimodal approach to HER and report better performance than with a single modality [2]. As previously discussed, it can be challenging to compare the different studies as they use various datasets and emotional models. In [2], M. Hasnul et al. reviewed ECG-based systems for HER and their applications in healthcare. They compared studies using ECG for unimodal and multimodal emotion recognition, where it is combined with other modalities. For both approaches, they found seven reports of achieving over 90% accuracy. Although it is hard to conclude based on these results due to the variability in the experiments, it shows that ECG-based emotion recognition has great potential for both unimodal and multimodal approaches.

5.2 Relevance of Obtained Results

As previously discussed, one big challenge in the field of HER today is the lack of data. To make reliable and general systems for emotion recognition, it is essential to have large datasets with enough data to train and test the systems. After oversampling, the ASCERTAIN and MAHNOB datasets consist of 2940 and 812 samples, respectively. Both datasets are relatively small, especially the MAHNOB dataset. After splitting the datasets into the train, validation, and test sets, the models were left with only 395 samples to train the models on from MAHNOB. The lack of enough data in the training process might have contributed to the pipelines not reaching their full potential. Grid search with cross-validation is a common way to solve the problem of few training samples. Cross-validation can be utilized to avoid splitting the data into three subsets and instead only splitting it into two subsets. Because of the Autoencoders using a train and test set to fit and evaluate the model, the dataset was split into three subsets to ensure that data leakage is completely avoided. The Autoencoders then use the training and evaluation sets to train and evaluate the model. These subsets are further used to train the classifiers. The third subset is not used until the training and evaluation of the Autoencoders and the classifiers are fully completed. In this study, the grid search is mainly included to speed up the process of identifying the optimal hyperparameters.

5.3 Remaining Challenges

As previously mentioned, the demographic variability in the dataset is an important aspect to consider. Most affective datasets contain a group of subjects, all from the same demographical group, like a university. Both ASCERTAIN and MAHNOB are examples of datasets with little demographical variations. ASCERTAIN conducted their experiments on 58 university students, and MAHNOB had 30 volunteers, all from the same college, namely the Imperial College in London. The cultural background of the subjects can have a significant impact on their emotional reactions to certain events. A dataset consisting of a wide variety of subjects is essential in making a system for emotion recognition capturing the variability in the population. The challenge is to create a database with enough samples from different demographical groups. The data collection for such a database would be both expensive and time-consuming.

In Chapter 2, the different emotional models were discussed. In HER, there are several ways of defining the target emotions. Some studies base their target emotions on the seven universal emotions, while others prefer the two-dimensional model of valence and arousal. Not having a standard emotional model for all datasets makes using different datasets in one study difficult. Additionally, it can be challenging to compare the results of various studies adapting to different emotion models. The problem of accurately measuring emotions was also discussed in Chapter 2. Understanding the emotional reactions to certain events is not always as straightforward as one might think. In some cases, capturing the full experience requires more than just the subject's own perception. For some datasets, questionnaires are used to annotate the data, which could lead to falsely annotated emotions.

5.4 Further Work

For future works, there are several approaches that could improve the results. As previously discussed, the sizes of the datasets are relatively small, which could affect the models' performances. Making larger datasets can be both costly and time-consuming. Instead of spending time and resources on collecting new data through experiments, augmentation can be used as an alternative method to increase the data samples. Augmentation is a way of using the data to generate new data similar to the original samples. Augmentation uses the training data to create new data by applying transformations or modifications to the existing data. Augmenting will provide more samples for the machine learning models to train on and potentially increase their ability to learn the patterns in the data. Additionally, an increased number of samples will give more samples to test the models on. In this study, augmentation was not included because of limited time.

Another approach to increasing the data samples without conducting new experiments for data collection is windowing. In addition to increasing the number of samples, windowing will ensure a fixed length for all samples. The signals and their R-waves are divided into a sequence of windows. The length of the windows can vary, but according to [62], the most common duration of the physiological variables is one minute. In further works, windowing can be helpful to improve the accuracy of emotion recognition due to more training data.

In this thesis, both datasets used for the emotion recognition task were imbalanced, having a significantly imbalanced class distribution. When predicting, the models provided biased results towards the majority classes. This issue was addressed using the random oversampling technique to get an equal distribution of the classes. In future works, different sampling techniques could potentially show better results. Various sampling techniques could include stratified sampling and cluster sampling or other techniques for oversampling. An example of another oversampling method used for ECG-based emotion recognition is SMOTE [63]. For this paper, the random oversampling techniques were chosen for simplicity, as the focus was on the feature extraction methods.

As mentioned in 5.1, the latent space of the Sparse Autoencoder showed significantly less bias towards any classes. The results in PT-SAE-SVC could implicate that Sparse Autoencoders are better suited for generative pseudo-sampling than feature extraction. In Further works a suggestion could be using Sparse Autoencoders to oversample the data instead of the random oversampling technique used in this thesis. This could potentially help improve the performance of the emotion recognition system as the Sparse Autoencoder will generate the new samples based on a learned distribution in the data.

Chapter 6

Conclusion

The main objective of this thesis has been to explore the effect of using the latent space of an Autoencoder and a Sparse Autoencoder as input to a classical machine learning model for ECG-based emotion recognition. Secondly, the performance is compared to state-of-the-art, and the potential for ECG data to be used as a single modality for HER is discussed.

The effects of adding an Autoencoder and a Sparse Autoencoder as feature extraction techniques have been tested by constructing three pipelines. The PT-SVC pipeline uses an SVC to classify based on the preprocessed ECG data. The PT-AE-SVC extracts the most significant features using an Autoencoder, which is then used as input data for an SVC. The last pipeline is the PT-SAE-SVC, where an SVC takes the latent space of a Sparse Autoencoder as input. All pipelines are oversampled to ensure balance in the dataset. The PT QRS-detection algorithm was also applied to minimize the noise in the data and resample all signals to be of the same length. The signals were resampled to frequencies of 256 Hz and 512 Hz for both datasets.

Two benchmark datasets, namely ASCERTAIN and MAHNOB, were used to train and test the three pipelines. The subjects ranked their level of valence and arousal, and the target emotions for classification were chosen to be based on the two-dimensional valence and arousal model. The target emotions used to describe the subjects' emotional states were LALV, LAHV, HALV, and HAHV, where L represents low, H represents high, A is arousal, and V is valence. In state-of-the-art, most studies choose to predict valence and arousal separately, making it difficult to draw any conclusions with the results from this thesis compared to their results.

The PT-SVC pipeline was used as a baseline to explore the impact of using an Autoencoder and a Sparse Autoencoder for feature extraction in ECG-based emotion recognition. Four evaluation metrics, namely accuracy, precision, recall, and F1-score, in addition to confusion matrices, were used to evaluate the three pipelines. The evaluation metrics revealed that the Autoencoder generally increased the performance and could potentially be a contribution to ECG-based emotion recognition. The confusion matrices displayed a clear bias towards one class, but by utilizing the Autoencoder for feature extraction, the bias was reduced. Furthermore, using a Sparse Autoencoder reduced the bias even more, but the general performance was negatively affected. This might imply that the latent space of Sparse Autoencoders could be used for generative pseudo-sampling.

Bibliography

[1] Jacopo Sini, Antonio Costantino Marceddu, and Massimo Violante. “Automatic emotion recognition for the calibration of autonomous driving functions”. In: *Electronics* 9.3 (2020), p. 518.

[2] Muhammad Anas Hasnul et al. “Electrocardiogram-based emotion recognition systems and their applications in healthcare—a review”. In: *Sensors* 21.15 (2021), p. 5015.

[3] Desislava Nikolova et al. “ECG-based human emotion recognition across multiple subjects”. In: *Future Access Enablers for Ubiquitous and Intelligent Infrastructures: 4th EAI International Conference, FABULOUS 2019, Sofia, Bulgaria, March 28-29, 2019, Proceedings 283.* Springer. 2019, pp. 25–36.

[4] Bo Sun and Zihuai Lin. “Emotion Recognition using Machine Learning and ECG signals”. In: *arXiv preprint arXiv:2203.08477* (2022).

[5] Luz Santamaria-Granados et al. “Using deep convolutional neural network for emotion detection on a physiological signals dataset (AMIGOS)”. In: *IEEE Access* 7 (2018), pp. 57–67.

[6] Sharifah Noor Masidayu Sayed Ismail, Nor Azlina Ab Aziz, and Siti Zainab Ibrahim. “A comparison of emotion recognition system using electrocardiogram (ECG) and photoplethysmogram (PPG)”. In: *Journal of King Saud University-Computer and Information Sciences* 34.6 (2022), pp. 3539–3558.

[7] Mimoun Ben Henia Wiem and Zied Lachiri. “Emotion classification in arousal valence model using MAHNOB-HCI database”. In: *International Journal of Advanced Computer Science and Applications* 8.3 (2017).

[8] Farnaz Panahi, Saeid Rashidi, and Ali Sheikhani. “Application of fractional Fourier transform in feature extraction from ELECTROCARDIOGRAM and GALVANIC SKIN RESPONSE for emotion recognition”. In: *Biomedical Signal Processing and Control* 69 (2021), p. 102863.

[9] Hany Ferdinando, Tapio Seppänen, and Esko Alasaarela. “Enhancing Emotion Recognition from ECG Signals using Supervised Dimensionality Reduction.” In: *ICPRAM.* 2017, pp. 112–118.

[10] Tianqi Fan et al. “A new deep convolutional neural network incorporating attentional mechanisms for ECG emotion recognition”. In: *Computers in Biology and Medicine* (2023), p. 106938.

[11] Hongli Zhang. “Expression-EEG based collaborative multimodal emotion recognition using deep autoencoder”. In: *IEEE Access* 8 (2020), pp. 164130–164143.

[12] Junxiu Liu et al. “EEG-based emotion classification using a deep neural network and sparse autoencoder”. In: *Frontiers in Systems Neuroscience* 14 (2020), p. 43.

[13] William James. “The emotions.” In: (1922).

[14] Lauri Nummenmaa and Heini Saarimäki. “Emotions as discrete patterns of systemic activity”. In: *Neuroscience letters* 693 (2019), pp. 3–8.

[15] Paul Ekman and Wallace V Friesen. “The repertoire of nonverbal behavior: Categories, origins, usage, and coding”. In: *semiotica* 1.1 (1969), pp. 49–98.

[16] Paul Ekman and Wallace V Friesen. “Constants across cultures in the face and emotion.” In: *Journal of personality and social psychology* 17.2 (1971), p. 124.

[17] Paul Ekman. “Are there basic emotions?” In: (1992).

[18] Anna Wierzbicka. “Human emotions: universal or culture-specific?” In: *American anthropologist* 88.3 (1986), pp. 584–594.

[19] Andrew Ortony and Terence J Turner. “What’s basic about basic emotions?” In: *Psychological review* 97.3 (1990), p. 315.

[20] Richard Wollheim. “On the emotions”. In: (1999).

[21] Paul E Griffiths. “Basic emotions, complex emotions, Machiavellian emotions”. In: (2002).

[22] Peter J Lang. “The emotion probe: Studies of motivation and attention.” In: *American psychologist* 50.5 (1995), p. 372.

[23] Andrius Dzedzickis, Artūras Kaklauskas, and Vytautas Bucinskas. “Human emotion recognition: Review of sensors and methods”. In: *Sensors* 20.3 (2020), p. 592.

[24] Roberto Zangróniz et al. “Electrodermal activity sensor for classification of calm/distress condition”. In: *Sensors* 17.10 (2017), p. 2324.

[25] Yujian Cai, Xingguang Li, and Jinsong Li. “Emotion Recognition Using Different Sensors, Emotion Models, Methods and Datasets: A Comprehensive Review”. In: *Sensors* 23.5 (2023), p. 2455.

[26] Guoying Zhao et al. “Facial expression recognition from near-infrared videos”. In: *Image and vision computing* 29.9 (2011), pp. 607–619.

[27] Abhiram Kolli et al. “Non-intrusive car driver’s emotion recognition using thermal camera”. In: *Proceedings of the Joint INDS’11 & ISTET’11*. IEEE. 2011, pp. 1–5.

[28] Qi-rong Mao et al. “Using Kinect for real-time emotion recognition via facial expressions”. In: *Frontiers of Information Technology & Electronic Engineering* 16.4 (2015), pp. 272–282.

[29] Olivier Martin et al. “The eNTERFACE’05 audio-visual emotion database”. In: *22nd international conference on data engineering workshops (ICDEW’06)*. IEEE. 2006, pp. 8–8.

[30] Kiavash Bahreini, Rob Nadolski, and Wim Westera. “Towards real-time speech emotion recognition for affective e-learning”. In: *Education and information technologies* 21 (2016), pp. 1367–1386.

[31] Srinivasan Ramakrishnan and Ibrahiem MM El Emary. “Speech emotion recognition approaches in human computer interaction”. In: *Telecommunication Systems* 52 (2013), pp. 1467–1478.

[32] Shi-wook Lee. “The generalization effect for multilingual speech emotion recognition across heterogeneous languages”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 5881–5885.

[33] Deger Ayata, Yusuf Yaslan, and Mustafa Kamaşak. "Emotion recognition via galvanic skin response: Comparison of machine learning algorithms and feature extraction methods". In: *IU-Journal of Electrical & Electronics Engineering* 17.1 (2017), pp. 3147–3156.

[34] Atul Luthra. *ECG made easy.* Jaypee Brothers Medical Publishers, 2019.

[35] VK Srivastava and Devendra Prasad. "DWT-based feature extraction from ECG signal". In: *American J. of Eng. Research (AJER)* 2.3 (2013), pp. 44–50.

[36] Sukkharak Saechia, Jeerasuda Koseeyaporn, and Paramote Wardkein. "Human identification system based ECG signal". In: *TENCON 2005-2005 IEEE Region 10 Conference.* IEEE. 2005, pp. 1–4.

[37] MAZ Fariha et al. "Analysis of Pan-Tompkins algorithm performance with noisy ECG signals". In: *Journal of Physics: Conference Series.* Vol. 1532. 1. IOP Publishing. 2020, p. 012022.

[38] Somchanok Tivatansakul and Michiko Ohkura. "Emotion recognition using ECG signals with local pattern description methods". In: *International Journal of Affective Engineering* 15.2 (2016), pp. 51–61.

[39] Arti Rawat and Pawan Kumar Mishra. "Emotion recognition through speech using neural network". In: *Int. J* 5 (2015), pp. 422–428.

[40] Monisha Chakraborty and Shreya Das. "Determination of signal to noise ratio of electrocardiograms filtered by band pass and Savitzky-Golay filters". In: *Procedia Technology* 4 (2012), pp. 830–833.

[41] Jiapu Pan and Willis J Tompkins. "A real-time QRS detection algorithm". In: *IEEE transactions on biomedical engineering* 3 (1985), pp. 230–236.

[42] Sebastian Raschka and Vahid Mirjalili. *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2.* Packt Publishing Ltd, 2019.

[43] Alan Mathison Turing. "Mind". In: *Mind* 59.236 (1950), pp. 433–460.

[44] Arthur L Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229.

[45] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.

[46] Shirish K Shevade et al. "Improvements to the SMO algorithm for SVM regression". In: *IEEE transactions on neural networks* 11.5 (2000), pp. 1188–1193.

[47] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20 (1995), pp. 273–297.

[48] Gend Lal Prajapati and Arti Patle. "On Performing Classification Using SVM with Radial Basis and Polynomial Kernel Functions". In: *2010 3rd International Conference on Emerging Trends in Engineering and Technology.* 2010, pp. 512–515.

[49] Andi Nurkholis, Debby Alita, Aris Munandar, et al. "Comparison of Kernel Support Vector Machine Multi-Class in PPKM Sentiment Analysis on Twitter". In: *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)* 6.2 (2022), pp. 227–233.

[50] MF Rohmah et al. "Comparison Four Kernels of SVR to Predict Consumer Price Index". In: *Journal of Physics: Conference Series.* Vol. 1737. 1. IOP Publishing. 2021, p. 012018.

[51] Mufni Alida and Metty Mustikasari. "Rupiah Exchange Prediction of US Dollar Using Linear, Polynomial, and Radial Basis Function Kernel in Support Vector Regression". In: *Jurnal Online Informatika* 5.1 (2020), pp. 53–60.

[52] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5 (1943), pp. 115–133.

[53] Frank Rosenblatt. “Perceptron simulation experiments”. In: *Proceedings of the IRE* 48.3 (1960), pp. 301–309.

[54] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[55] James L McClelland, David E Rumelhart, PDP Research Group, et al. *Parallel Distributed Processing, Volume 2: Explorations in the Microstructure of Cognition: Psychological and Biological Models*. Vol. 2. MIT press, 1987.

[56] Andrew Ng et al. “Sparse autoencoder”. In: *CS294A Lecture notes* 72.2011 (2011), pp. 1–19.

[57] Lars Buitinck et al. “API design for machine learning software: experiences from the scikit-learn project”. In: *arXiv preprint arXiv:1309.0238* (2013).

[58] Martın Abadi et al. “Tensorflow: a system for large-scale machine learning.” In: *Osdi*. Vol. 16. 2016. Savannah, GA, USA. 2016, pp. 265–283.

[59] Ramanathan Subramanian et al. “ASCERTAIN: Emotion and personality recognition using commercial sensors”. In: *IEEE Transactions on Affective Computing* 9.2 (2016), pp. 147–160.

[60] JEROEN Lichtenauer and MOHAMMAD Soleymani. *Mahnob-hci-tagging database*. 2011.

[61] Margaret M Bradley and Peter J Lang. “Measuring emotion: the self-assessment manikin and the semantic differential”. In: *Journal of behavior therapy and experimental psychiatry* 25.1 (1994), pp. 49–59.

[62] Sylvia D Kreibig. “Autonomic nervous system activity in emotion: A review”. In: *Biological psychology* 84.3 (2010), pp. 394–421.

[63] Retantyo Wardoyo, I Made Agus Wirawan, and I Gede Angga Pradipta. “Oversampling approach using radius-SMOTE for imbalance electroencephalography datasets”. In: *Emerging Science Journal* 6.2 (2022), pp. 382–398.

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway