National Technical University of Athens
School of Electrical and Computer Engineering
Division of Signals, Control and Robotics

Manifold Learning and Nonlinear Recurrence Dynamics for Speech Emotion Recognition on Various Timescales

DIPLOMA THESIS

EFTHYMIOS TZINIS

Supervisor : Alexandros Potamianos
Associate Professor NTUA

Athens, June 2018

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Signals, Control and Robotics

Manifold Learning and Nonlinear Recurrence Dynamics for Speech Emotion Recognition on Various Timescales

DIPLOMA THESIS

EFTHYMIOS TZINIS

Supervisor : Alexandros Potamianos
Associate Professor NTUA

Approved by the examining committee on the June 21, 2018.

Alexandros Potamianos Associate Professor NTUA	Petros Maragos Professor NTUA	Giorgos Stamou Associate Professor NTUA

Athens, June 2018

Efthymios Tzinis
Electrical and Computer Engineer

Dedicated to my math teacher Giorgos Fragkoulopoulos.
The first one who managed to foresee my true potential and believed in it,
despite the nonconforming and highly chaotic nature of my existence.

Abstract

In this work [1] we investigate Speech Emotion Recognition (SER) by following three different approaches which are outlined below. For the evaluation of each approach, we use multiple datasets and experimental setups which are also followed by the literature. Moreover, both utterance-based and segment-based classification methods are followed where each emotional utterance is represented by one feature vector and a list of vectors, respectively.

First, we explore the efficacy of various time-scales (frame, phoneme, word or utterance) for deciding the emotional content of a speech utterance for both Low Level Descriptors (LLDs) (local features) and statistical functionals (global features). By combining Recurrent Neural Networks (RNNs) and statistical functionals over segments that roughly correspond to the duration of a couple of words, we report state-of-the-art results on IEMOCAP. Purportedly, choosing the appropriate time-scale is key for high performing SER systems.

In addition, we investigate the performance of features that can capture nonlinear recurrence dynamics embedded in the speech signal for SER. Reconstruction of the phase space of each speech frame and the computation of its respective Recurrence Plot (RP) reveals complex structures which can be measured by performing Recurrence Quantification Analysis (RQA). These measures are aggregated by using statistical functionals over segment and utterance periods. We report SER results for the proposed feature set on three databases using different classification methods. When fusing the proposed features with traditional feature sets, we show an improvement in unweighted accuracy of up to 5.7% and 10.7% on Speaker-Dependent (SD) and Speaker-Independent (SI) SER tasks, respectively, over the baseline feature set. Following a segment-based approach we demonstrate state-of-the-art performance on IEMOCAP using an attention-based Bidirectional RNN.

Finally, we reduce the dimensionality of acoustic features used for SER by using manifold learning algorithms. In essence, we present a novel algorithm for nonlinear manifold learning using derivative-free optimization techniques, namely, Pattern Search MDS. By using General Pattern Search (GPS) formulation we are able to provide theoretical convergence guarantees up to first order stationary points for the proposed algorithm. Moreover, we demonstrate practical improvements of the proposed algorithm in terms of computational efficiency, convergence rate and solution accuracy on various experimental setups. Our results suggest that our algorithm is capable of finding solutions to the general problem of multidimensional scaling (MDS) under multiple setups. In accordance with our focus on SER, we evaluate Pattern Search MDS as briefly discussed next. Each emotional utterance is represented by a feature vector lying in a high dimensional space. In order to reduce the dimensionality of these emotional feature vectors, we try to approximate an underlying low-dimensional manifold in which the initial pairwise distances are also preserved. We show that a significant reduction in terms of input dimensionality and training time can be achieved by simultaneously maintaining SER accuracy at a competitive level.

Key words

Nonlinear recurrence dynamics, deep learning, machine learning, speech emotion recognition, recurrence plot, multidimensional scaling, manifold learning

[1] Papers: [1], [2], [3], [4] have been conducted under the development of this thesis.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Prof. Alexandros Potamianos not only for his invaluable guidance during the conduction of my thesis but also for his motivation. He has been continuously stimulating my research ideas and pushing me forward; compelling me to work at the best of my capabilities.

In addition, I would like to thank Prof. Petros Maragos and Prof. Giorgos Stamou for being part of my thesis committee as well as their inspirational courses at National Technical University of Athens (NTUA) which intrigued me to delve deeper into the study of dynamical systems and neural networks.

Special thanks to my fellow labmate and close friend Giorgos Paraskevopoulos for the high quality work we have produced, the insightful conversations we have exchanged and the sleepless nights during the two-digit working hours routine we have experienced together. But most profoundly, I would like to thank him as well as all the other people from the group, for the fun we had the past year.

My eternal appreciation goes to my parents for their unselfish love and support during all these years of my academic development. Specifically, I would like to thank them for instilling me the priceless virtues of perseverance and self-responsibility which will forever accompany me.

Last but not least, I would like to thank all my friends and all the people who have indulgently stood by me through the most challenging days of my life. I will never forget your limitless encouragement when I was sharing my dreams and ambitions with you.

Efthymios Tzinis,

Athens, June 21, 2018

Contents

Abstract . . . 7

Acknowledgements . . . 9

Contents . . . 11

List of Tables . . . 15

List of Figures . . . 17

List of Algorithms . . . 19

1. Introduction . . . 21
- 1.1 Emotion Perception . . . 21
- 1.2 Emotion in Speech Signals . . . 21
- 1.3 Automatic Speech Emotion Recognition . . . 23
  - 1.3.1 Motivation . . . 23
  - 1.3.2 Acoustic Features . . . 23
  - 1.3.3 Classification Methods . . . 25
  - 1.3.4 State-of-the-Art Approaches . . . 27
- 1.4 Challenges . . . 28
  - 1.4.1 Timescales of Decision . . . 28
  - 1.4.2 Nonlinear Phenomena in Speech Production . . . 30
  - 1.4.3 Curse of Dimensionality and Dimensionality Reduction . . . 33
- 1.5 Goals and Contributions . . . 37
- 1.6 Thesis Organization . . . 38

2. Technical Background . . . 41
- 2.1 Notation . . . 41
- 2.2 Classification Models . . . 41
  - 2.2.1 Loss Function . . . 41
  - 2.2.2 Support Vector Machines (SVMs) . . . 43
  - 2.2.3 Logistic Regression (LR) . . . 44
  - 2.2.4 K-Nearest Neighbors (KNNs) . . . 44
- 2.3 Recurrent Neural Networks (RNNs) . . . 45
  - 2.3.1 Long Short Term Memory (LSTM) unit . . . 46
  - 2.3.2 Bidirectional LSTM . . . 47
  - 2.3.3 Attention Mechanism . . . 48
  - 2.3.4 Attention-based Bidirectional LSTMs . . . 50
- 2.4 Phase Space Reconstruction . . . 50
  - 2.4.1 Definition . . . 50
  - 2.4.2 Average Mutual Information (AMI) . . . 50
  - 2.4.3 Takens Theorem . . . 52

2.4.4 False Nearest Neighbors (FNN) . . . 52
2.5 Recurrence Plots (RPs) . . . 53
2.6 Recurrence Quantification Analysis (RQA) . . . 54
2.6.1 RQA Measures . . . 54
2.7 Multidimensional Scaling (MDS) . . . 56
2.7.1 Classical MDS . . . 56
2.7.2 Metric MDS . . . 57
2.7.3 SMACOF . . . 57
2.8 General Pattern Search (GPS) methods . . . 58
2.8.1 GPS formulation . . . 58
2.8.2 GPS Convergence . . . 59

3. Timescales of Decision for Speech Emotion Recognition . . . 63
3.1 Motivation . . . 63
3.2 Acoustic Features . . . 63
3.2.1 Pre-Processing . . . 63
3.2.2 Voice Activity Detection . . . 64
3.2.3 Local Features . . . 67
3.2.4 Global Features . . . 68
3.3 Approaches on Different Timescales . . . 70
3.3.1 Frame-Based . . . 70
3.3.2 Segment-Based . . . 72
3.3.3 Utterance-Based . . . 73
3.4 Experimental Setup . . . 73
3.4.1 Dataset . . . 74
3.4.2 Performance Measurement . . . 74
3.4.3 LSTM Trained on Frame Level LLDs . . . 74
3.4.4 LSTM Trained on Statistical Features . . . 75
3.4.5 SVM Trained on Utterance Level Statistical Features . . . 76
3.5 Experimental Results and Discussion . . . 76
3.5.1 Optimal Timescales for LSTM Trained on Local Features . . . 76
3.5.2 Optimal Timescales for LSTM Trained on Global Features . . . 77
3.5.3 Comparison Between Timescales . . . 78
3.5.4 Comparison with the Literature . . . 79

4. Modeling Nonlinear Recurrence Dynamics for Speech Emotion Recognition . . . 81
4.1 Motivation . . . 81
4.2 Related Work . . . 82
4.3 Reconstructing Phase Spaces from Speech Signals . . . 83
4.3.1 Unfolding of the True Dynamics of Speech . . . 83
4.3.2 Definition . . . 84
4.3.3 Estimating time-lag parameter . . . 84
4.3.4 Analyzing AMIs for Different Speakers and Emotions . . . 85
4.3.5 Estimating embedding dimension parameter . . . 86
4.3.6 Reconstructing Phase Spaces for Various Speech Segments . . . 87
4.4 Recurrence Plots (RPs) from Speech Frames . . . 89
4.4.1 From Phonemes To Recurrence Plots . . . 91
4.4.2 The Effectiveness of PS's Parameters for the Extraction of RPs . . . 91
4.4.3 Emerging Chaotic Structures behind RPs of Phonemes? . . . 93
4.5 Acoustic Feature Extraction . . . 94
4.5.1 Baseline Feature Set (IS10 Set) . . . 95
4.5.2 Proposed Nonlinear Feature Set (RQA Set) . . . 95

4.5.3 Fused Feature Set (RQA + IS10 Feature Set) . . . 95
4.6 Classification Methods . . . 95
4.6.1 Utterance-Based . . . 95
4.6.2 Segment-Based . . . 96
4.7 Experiments . . . 97
4.7.1 Datasets . . . 97
4.7.2 Speaker Dependent (SD) . . . 98
4.7.3 Speaker Independent (SI) . . . 98
4.7.4 Leave One Session Out . . . 99

5. Pattern Search Multidimensional Scaling . . . 101
5.1 Motivation . . . 101
5.2 Related Work . . . 101
5.3 Algorithm Description . . . 102
5.3.1 Formulation . . . 102
5.3.2 Search Directions . . . 103
5.3.3 Move Alongside the Optimal Direction . . . 103
5.3.4 Computation of the Error . . . 104
5.3.5 Complexity . . . 104
5.4 Approximations and Optimizations . . . 104
5.4.1 Tolerance for "Bad" Moves . . . 104
5.4.2 Updating the Current Dissimilarity Matrix . . . 105
5.4.3 Randomized Direction Selection . . . 105
5.4.4 Estimating a Starting Radius . . . 105
5.4.5 Parallel Implementation . . . 105
5.5 GPS Formulation of Pattern Search MDS . . . 106
5.6 Convergence of Pattern Search MDS . . . 108
5.7 Experiments . . . 108
5.7.1 Tuning the hyperparameters . . . 108
5.7.2 Manifold Geometry Reconstruction . . . 109
5.7.3 Semantic Similarity . . . 110
5.7.4 Image classification . . . 110
5.7.5 Speed of Convergence Evaluation . . . 112
5.7.6 Robustness to Noise . . . 112
5.7.7 Dimensionality Reduction for Speaker Independent SER . . . 114
5.7.8 Comparing (Pattern Search MDS + KNN) to Utterance Level Parametric Models 120
5.8 Visualizing Emotional Manifolds from Utterance Level Acoustic Features . . . 121
5.8.1 IS10 2D Manifolds for EmoDB . . . 122
5.8.2 RQA 2D Manifolds for EmoDB . . . 123
5.8.3 Fused Feature Set (RQA + IS10) 2D Manifolds for EmoDB . . . 124
5.8.4 Fused Feature Set (RQA + IS10) 3D Manifolds for 2 Male Speakers of IEMOCAP . . . 125

6. Epilogue . . . 127
6.1 Conclusions . . . 127
6.1.1 Conclusions from Chapter 3 . . . 127
6.1.2 Conclusions from Chapter 4 . . . 127
6.1.3 Conclusions from Chapter 5 . . . 127
6.2 Future Work . . . 128

Bibliography . . . 133

Appendix 145

A. Abbreviations 145

List of Tables

3.1 Local Features . . . 67
3.2 Sets of Statistical Functionals for Global Features . . . 69
3.3 Global Features . . . 69
3.4 Accuracy of Proposed Models on Various Timescales . . . 78
3.5 Accuracy of Models in the Literature and Proposed Model . . . 79

4.1 RQA Measures extracted from each RP . . . 96
4.2 Sets of Statistical Functionals for RQA Feature Set . . . 96
4.3 SD results on SAVEE and Emo-DB. (ESR) Ensemble Softmax Regression . . . 98
4.4 SI results on SAVEE and Emo-DB. (ESR) Ensemble Softmax Regression . . . 99
4.5 LOSO results on IEMOCAP. (GFS): Glottal Flow Spectrogram, (SP): Spectrogram. . . 100

5.1 Comparison of pattern search MDS with other dimensionality reduction algorithms for the semantic similarity task with word-embeddings . . . 111
5.2 Comparison of pattern search MDS with other dimensionality reduction algorithms for the MNIST dataset . . . 112
5.3 Comparison of pattern search MDS with other dimensionality reduction methods for semantic similarity task using injected with noisy word-embeddings . . . 115
5.4 Comparison of pattern search MDS with other dimensionality reduction methods for Speaker Independent Speech Emotion Recognition using IS10 Feature Set . . . 117
5.5 Comparison of pattern search MDS with other dimensionality reduction methods for Speaker Independent Speech Emotion Recognition using RQA Feature Set . . . 118
5.6 Comparison of pattern search MDS with other dimensionality reduction methods for Speaker Independent Speech Emotion Recognition using the Fused (RQA + IS10) Feature Set . . . 119
5.7 Comparison of (Pattern Search MDS + KNN) to Utterance Level Parametric Models 121

List of Figures

1.1 Two dimensional emotion map using activation and valence [5] . . . 22
1.2 Emotional Classification using different RNN architectures. . . . 29
1.3 Human System of Speech Production . . . 31
1.4 Source Filter Model of Speech Production [6] . . . 32
1.5 A phone /a/ and its corresponding phase space exhibiting recurrence dynamics [7] . . . 32
1.6 Learned two dimensional manifolds using t-SNE with different parametrization for a variety of sounds . . . 34
1.7 2D manifold of MNIST dataset learned using t-SNE . . . 36
1.8 2D manifold of word embeddings for a variety of context words . . . 37

2.1 Unrolling of a Recurrent Neural Network with an input sequence of $T + 1$ timesteps 45
2.2 Block Diagram of a Long Short-Time Memory unit . . . 46
2.3 Bidirectional LSTM layer . . . 48
2.4 Attention Mechanism for given activations from an RNN . . . 49
2.5 Recurrence Plots for Different Types of Systems. From left to right: random noise, periodic oscillations with two frequencies, deterministic chaotic system and autoregressive process. . . . 54

3.1 WebRTC VAD evaluation on EmoDB utterances. . . . 65
3.2 OpenSMILE VAD mechanisms based on probability of voicing and pre-trained LSTM 66
3.3 OpenSMILE VAD based on different cut-offs from voice probability . . . 66
3.4 Frame-Based Approach using LSTM trained on sequences of concatenated frame-level LLDs . . . 71
3.5 Segment-Based Approach using LSTM trained sequences of segment global statistical features computed over . . . 72
3.6 Utterance-Based Approach using SVM trained on utterance statistical features . . . 73
3.7 Weighted Accuracy (WA) and Unweighted Accuracy (UA) of LSTM trained on various lengths of concatenated frames with local features . . . 77
3.8 Weighted Accuracy (WA) and Unweighted Accuracy (UA) of LSTM trained on global features over various segments durations . . . 78

4.1 Average Mutual Information for a speech-segment of 0.2 seconds . . . 84
4.2 Inverse AMI plots for Different Speakers and Emotions for the phoneme /ae/ . . . 85
4.3 Selected time-lags for an *angry* utterance of 3 seconds for various time-durations . . . 86
4.4 Percentiles of false nearest neighbors for various embedding dimensions and speech segments of different time-durations . . . 88
4.5 Reconstructed Phase Spaces of Speech Frames of Vowels . . . 89
4.6 Vowel /ae/ in time domain and its corresponding Recurrence Plot for different speakers and their emotional manifestations . . . 90
4.7 Extracting Continuous RPs for vowel /ae/ for all SAVEE speakers and their emotional manifestations . . . 92
4.8 Extracting the RP of a 30ms frame contained in the excitation of vowel /e/ and understanding the underlying dynamics by comparison . . . 94

5.1 Sphere of radius r around point $\mathbf{x}_i^{(k)}$ and possible search directions . . . 103
5.2 Convergence plot for different starting radii . . . 109
5.3 Comparison of pattern search MDS with other dimensionality reduction algorithms for reconstructing $2D$- manifolds from $3D$ artificial data . . . 111
5.4 Convergence comparison of pattern search MDS and MDS SMACOF for reconstruction of geometrical shapes and semantic similarity task . . . 113
5.5 Comparison of pattern search MDS with other dimensionality reduction methods for the reconstruction of $2D$- manifolds from $3D$ artificial data injected with Gaussian noise . . . 114
5.6 Comparison of pattern search MDS with other dimensionality reduction methods for shapes with holes and non-convex regions . . . 115
5.7 Comparison of the produced $2D$ Manifolds when applying dimensionality reduction methods to IS10 acoustic feature representations for EmoDB . . . 122
5.8 Comparison of the produced $2D$ Manifolds when applying dimensionality reduction methods to RQA acoustic feature representations for EmoDB . . . 123
5.9 Comparison of the produced $2D$ Manifolds when applying dimensionality reduction methods to Fused (RQA + IS10) acoustic feature representations for EmoDB . . . 124
5.10 Comparison of the produced $3D$ Manifolds when applying dimensionality reduction methods to Fused (RQA + IS10) acoustic feature representations for 2 Male Speakers of IEMOCAP . . . 126

6.1 Isolating periodic subgraphs from RPs in order to extract pitch-invariant representations for SER . . . 129
6.2 Bimodal Autoencoder for combining spectral and nonlinear representations of speech signals . . . 130

List of Algorithms

1 General Pattern Search (GPS) 59
2 Pattern Search MDS 103
3 Define search directions 103
4 Find optimal move for a point 104

Chapter 1

Introduction

1.1 Emotion Perception

Emotions play a crucial role on the communication interplay of human beings. Discrete emotions reflect the inner affective state of the speaker at the current moment and by being able to perceive this information, the corresponding human engaging in the dialogue is able to provide more empathetic answers [8]. Emotions are expressed using primarily facial gesticulations, body movements and vocal cues. For example, when engaging in a dialogue with a person who is crying then by interpreting the social signal of a sad face with tears and the trembling voice, one is capable of following a soothing approach in order to communicate his message.

Some of the most prominent emotions are anger, fear, disgust, happiness, sadness and surprise which are aptly described in the "Big-Six" emotion model introduced by Ekmann [9]. On the other hand, some psychological models propose the discrimination of emotions based on two axis of "valence" and "activation" [10]. In this model, each emotion is mapped to a specific area of the two-dimensional space produced by the two aforementioned axis as can be seen in Figure 1.1. Similarly, other models are creating a three dimensional space by also utilizing an "attention-rejection" axis in order to properly codify the emotional map [11].

The majority of these models are originated from a biological or a psychological point of view in order to describe emotions and human behaviors. An arousing question is how we could actually validate the efficacy of these models or even utilize them in real case scenarios and how these models are extended under various information modalities (image, audio, text, etc.).

1.2 Emotion in Speech Signals

Speech is one of the most common channels of emotion expression and sometimes is the most convenient one. For instance, in a conversation by phone or other means where one cannot see the facial expression of his/her co-speaker. During the past years, researchers tried to find correlations between characteristics of vocal cues and emotions which are conveyed. Under this method of processing emotional speech a variety of ordered relationships have been proposed between acoustic features and emotions. In essence, these relationships are qualitative indicators of how each emotion is mapped in the range of values provided by a speech feature.

Some relationships are quite intuitive such as an angry utterance would more probably yield higher energy content than a sad utterance. Of course there is an undeniable variation between different speakers but without loss in generalization we can conclude that in general emotions which are ranked higher in arousal/activation axis would also rank higher in terms of energy content [12]. In this context, SER can be viewed as a more difficult task and instead an easier one is tried to be solved which is continuous emotion recognition. Based on the previous statements about the axes of arousal and valence we could assume that each emotion could be aptly represented in the 2D plane of these continuously valued axes. Thus, we could try to predict for each emotional utterance its respective score on each one of this axis and after that we would assume that we could assign the corresponding discrete emotion. Furthermore, the fundamental frequency of the vocal tract or pitch as it is called is also a

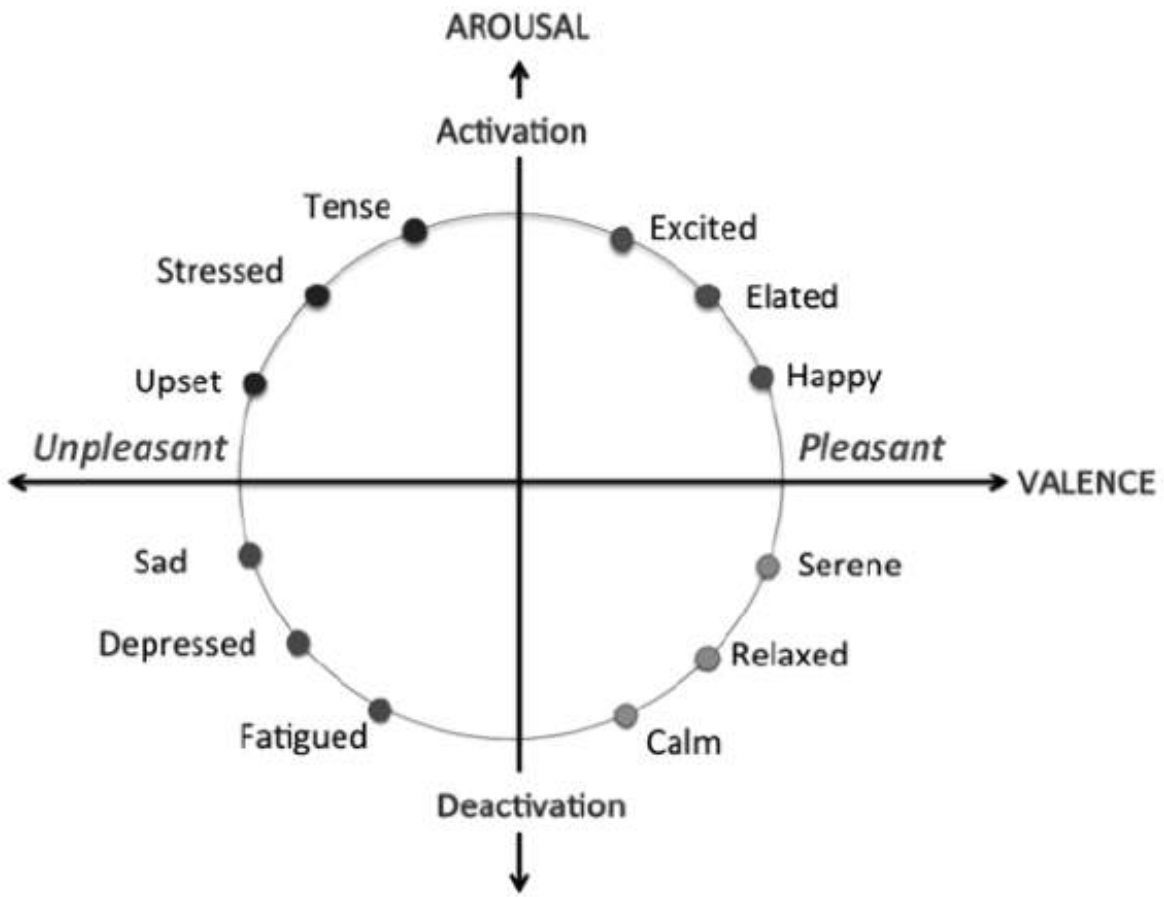


Figure 1.1: Two dimensional emotion map using activation and valence [5]

good discriminator between some basic emotions. Essentially, pitch corresponds to the frequency of the oscillations that the vocal fold produces under speech generation and thus a higher tone in voice is directly correlated to the instantaneous pitch value. Speech signals containing fear or anger tend to exhibit very high pitch values while signals containing sadness or disgust are mapped in lower values of the fundamental frequency [13]. Although these mappings cannot provide an objective criterion for discriminating emotions they were the sparkling lights in the journey of emotion recognition because of their easy interpretation.

However, emotions also correlate with specific areas of values from acoustic features which are not as intuitive as the ones described previously. For instance in [14] a thorough and exhaustive study is performed for finding correlations with a variety of acoustic features and basic emotions. For example, higher speaking rate might indicate an emotion expression of anger, fear or disgust while on the contrary lower values might correspond to sadness or happiness. Abrupt changes in pitch values over time might represent stressed conditions or anger compared to smooth transitions which are more likely to be found in utterances of happiness or fear.

In order to objectively analyze the emotional content of speech signals we cannot use these well studied yet qualitative relationships between acoustic features and underlying emotions. In such a vaguely defined area as emotion perception, qualitative discriminators are often subject to debate and cannot offer a solution in real case scenarios where the affective state of a speaker would need to be inferred. We need to be able to encode emotional information from speech in such a way that facilitates real time emotion recognition and can generalize in multiple speakers. In this context, the quantitative analysis of emotions in speech signals seems more a necessity than a subject to a philosophical argument.

1.3 Automatic Speech Emotion Recognition

In this section we provide an initial motivation from the theoretical models of emotion perception to quantified methods of affective computing and some applications of the latter in Section 1.3.1. Moreover, we discuss some of the most widely used set of acoustic features and classification methods which are proposed in literature in Sections 1.3.2 and 1.3.3, respectively. By the term method here we refer to the combination of the feature extraction strategy alongside its combination with a use of a classifier which is used to predict the corresponding emotional label for the testing utterance. Finally, in Section 1.3.4, we briefly detail some state-of-the-art approaches for Speech Emotion Recognition (SER) which are presented in literature. From now on when we use the term SER we refer to automatic speech emotion recognition which is performed in a computational setup instead of using a theoretical framework for interpreting emotional context.

1.3.1 Motivation

Extensive analysis of human perception and generation of emotion has provided substantial intuition for the implementation of artificial models which are able to capture behavioral signals [15]. Building emotionally aware Human-Machine Interfaces (HMIs) ultimately relies on understanding the underlying dynamics of the affective cues and integrating cognition in an HMI. Consequently, the goal of autonomous Speech Emotion Recognition (SER) is to build a machine which is capable of interacting with humans and adeptly interpreting and utilizing affective signals from speech. All these models belong to the wider class of Artificial Intelligence (AI) models. The latter is originated under a multidisciplinary context like neuroscience, physics, biology, psychology and last but not least computer science. In this study we follow a computational approach but we should not be oblivious to the significant contribution of other disciplines in order to be able to build SER models.

Building upon the transition from the abstract nature of emotions to a quantitative approach where all the information should be coded in a vector of zeros and ones we need to recall some of the potential usages of an SER system. For example, SER is key for building intelligent adaptive HMI's to the affective state of the user, especially in cases like call centers where no other information modality is available [16]. In general, automatic dialogue systems which are empathetic to the inner state of the user are vital assets to a vast amount of applications including surveillance and tutoring agents [17]. Going a step further, capturing the affective state of a speaker could be utilized for even higher level predictions such as engagement of a user in a certain dialogue interplay [4].

Specifically, in [4] we have studied an automatic classifier for deciding whether a child is suffering from Autism Spectrum Disorder (ASD). We integrated an innovative psychological model for determining the true engagement level for every utterance of the dyad parent-child interplay. We performed a binary classification experiment by using a multimodal Support Vector Machine (SVM) trained on acoustic, linguistic and dialogue features derived from our introduced dataset with video recordings. We showed that Typically Developing (TD) children demonstrated a more deterministic behavior model than ASD children while parent's video-related features were found to be the best predictors for child's engagement level. The resulting paper provides deeper insight in the social and cognitive structures of children with ASD. In general, the aforementioned studies display only a small fraction of what affective computing could give back to humanity [18].

1.3.2 Acoustic Features

One of the most demanding problems in SER is finding a representative set of emotion features and the optimal time-scale for emotional context extraction. Prosodic, spectral and voice quality Low Level Descriptors (LLDs), extracted from speech frames, have been extensively used for SER [19]. Some of the most widely used LLDs or local features are Mel Frequency Cepstral Coefficients (MFCCs), pitch or fundamental frequency, short-time energy from frames, Zero Crossing Rate (ZCR) and Harmonic to Noise Ratio (HNR). All these LLDs are extracted directly from speech frames corresponding to

windows of $20 - 100$ms. Thus, in order to be able to provide a temporal aggregation of the emotional information for longer periods of speech than frames, global features are used. The latter set of features is calculated as statistical functionals over local features for the given utterance or speech-segment [20].

During the past years, a variety of LLDs, extracted on a frame level and later aggregated using statistics or modeled directly as a sequence of frames, has been proposed in the literature as capable of capturing emotional content [19]. The main bulk of work in acoustic feature extraction has been revolving around paralinguistic analysis of speech signals primarily derived from prosodic and spectral representations [21]. Some other features like jitter and shimmer have also been shown to provide additional information for the classification of human speaker styles and arousal levels [22] and used later for SER [23]. All these handcrafted features have been thoroughly tested for their paralinguistic properties and their efficacy on capturing emotional content from speech. An exemplary basis of the most essential approaches existing in the paralinguistic domain is given by Schuller and Batliner in [24].

Some features sets, containing most of the aforementioned descriptors as well as a set of statistical functionals which are applied over them have yielded impressive classification accuracies on SER tasks. One of the first compact feature sets presented was a 384 acoustic feature vector in Interspeech 2009 Paralinguistic Challenge [25] (IS09 Feature Set). This set consisted of the LLDs referred previously in this section and resulted in accuracies of up to 70.1% and 65.1% for a 2-class and 5-class emotion classification tasks on Fau-Aibo dataset [26]. A much bigger set of 1582 features has been introduced in Interspeech 2010 (IS10 Feature Set) by Schuller [23] where it was tested on age, gender and level of interest recognition tasks. In this set, some extra descriptors like jitter, shimmer, voicing probability, line spectral pairs and log Mel Frequency-banks have been added to the LLDs of the previous work. Surprisingly, this set is still used and yields state-of-the-art results even in todays approaches [2], [27], [28]. To this end, a much wider set of speech features has been introduced in a subsequent paralinguistic challenge of Interspeech 2013 [29] presenting a 6373-dimensional feature vector (IS13 Feature Set) for each utterance. The proposed feature set has been assessed under recognition of emotions, autism, affect and social signals from vocal cues.

In addition, some works combine different feature sets for emotion classification by combining different modalities or feature sets extracted from a different domain. For example, in [30] three different information modalities have been utilized for feature extraction, namely, text, audio and image. The final emotional utterance representation consists of the concatenation of the three feature vectors which serves as input for the classifier. Notably, image multimodal approach is not directly related to SER as it includes an information channel which cannot be captured solely by audio but it falls under the general task of automatic emotion recognition. Moreover, linguistic information could be captured from audio by building the acoustic model and language models in order to perform Automatic Speech Recognition (ASR) [6]. In this frame, acoustic features have been concatenated with linguistic features for improving the performance of SER systems [31]. The concatenation of these feature vector is often called "early fusion" or just "fusion".

Other works are using much more compact feature sets for SER. In [32] a subset of the aforementioned IS10 feature set [23] are used for domain adaptation on SER tasks. Moreover, because of the huge dimensionality of the proposed feature sets, many approaches include a dimensionality reduction technique in order to lessen the computational load in the training process. Namely, we recall that in some works some linear dimensionality reduction techniques are employed such as Principal Component Analysis (PCA) [33]. In the later work PCA is performed over a vector of pitch and energy based features in order to minimize the dimensionality of the input vectors but to also boost the final classification accuracy. Furthermore, different feature selection techniques have also flourished under the huge dimensionality of the proposed feature sets for SER. For instance, in [34] Ensemble Random Forest to Trees (ERFTrees) have been introduced for the selection of the most representative features as well as nonlinear dimensionality reduction techniques for reducing the input space of vectors used for SER.

In the same context of minimal speech processing and feature extraction, there is a necessity of reducing the total time of feature extraction and data preprocessing in general. To this end, in [35], each emotional utterance is represented only by its spectrogram and thus minimal speech processing is required. In essence, a Fourier transformation is applied for the whole process of data preparation and feature extraction before the final training/testing process comes in front. Spectrogram representations have also been used from glottal source signals for the final emotion classification in a more general representation learning approach [36]. Another common feature extraction technique is to utilize an autoencoder in order to automatically extract salient features from spectral representations such as Mel Filterbanks (MFBs) [27], spectrograms from raw signal [37] or glottis [36].

Other less popular feature extraction methods are based on transforming the input signal using a different analysis than Fourier or Cepstrum. For instance, in [38] a wavelet packet transformation is employed for a later feature extraction. Wavelets are defined by coefficients which are parameters to be tuned during the training phase of optimizing a cost function such as the entropy between emotional classes. The wavelet coefficients are capable of capturing the information of low-level frequencies close to human speech generation frequencies [39]. Moreover, in [40] Teager Energy Operator is used for SER which is a nonlinear operator oftenly used for estimating truthfully the energy of the source filter under AM-FM modulation process [41]. Some other nonlinear features have also been tested on SER tasks which are derived from the phase space of speech signals [42] and the instantaneous phase and amplitude of the speech frame [43]. Nonlinear features for SER is actually the main concept of Chapter 4. Specifically, in Section 4.2, we discuss in much detail other approaches which are also employing nonlinear analysis of speech signals for SER.

Recently, significantly less time-consuming preprocessing approaches have been introduced where the conventional speech features are supplemented by the raw signal representations. Surprisingly, in [44] only the raw signal representation of each emotional utterance is used as the input feature vector and simultaneously competitive results have been reported on continuous emotion recognition.

1.3.3 Classification Methods

On the one hand, extracting features capable of capturing the emotional state of the speaker is a challenging task for SER but this is only a fraction of the configurations that needs to be set in order to specify the holistic approach which will be followed for the final classification. The selection of the appropriate classifier, the way it will be trained, transfer learning and other techniques between different models and other parameters are also essential parts in order to perform SER. Thus, the grid space one has to search before he concludes for their approach becomes vast, if we acknowledge the fact that new models capable of modeling speech frames as well as other models which are keenly connected to models from other domains such as text and image. For instance, Convolutional Neural Networks (CNNs) were firstly introduced for image recognition of handwritten digits [45] but have been used extensively for SER by using as input the spectrogram of emotional utterances [35]. In this section we do not aim to exhaustively present all the historical contributions of other researchers in SER but we focus on displaying some of the most prominent ones in terms of novelty. In the next section (Section 1.3.4) we will cover similar aspects but in term of efficacy and focusing mainly on findings which yield the best classification accuracies for SER.

Proposed SER approaches mainly differ on the aggregation and temporal modeling of the input sequence of LLDs or in other words local features corresponding to frame-representations. We can easily divide SER approaches to three main categories. Namely, 1) utterance-based 2) segment-based and 3) direct approaches which are explained below in much more detail. For each category we present some of the most resilient approaches in terms of performance on SER tasks.

Utterance-based approaches: In these approaches, emotional utterances are first split into overlapping frames which is a common speech processing technique in order to extract short-term features [6]. For each frame a set of LLDs are extracted and then they are aggregated over the whole utterance length by applying statistical functionals and consequently creating a global statistical representation for each emotional utterance. This final representation is used as an input for a classifier which is

trained on a selected fraction of the dataset and tested on the rest of the available utterances. Utterance-based approaches are the most straightforward to implement but combined with robust classifiers show good results for a variety of SER tasks [20]. For instance, using only utterance-level statistics from pitch-related LLDs and Gaussian Mixture Models (GMMs) for the final classification yield a recognition accuracy of up to 77% [46]. A similar approach is to use a Support Vector Machine (SVM) classifier with utterance-based statistics [47]. The IS09 feature set was tested on a variety of emotional speech databases with a multi-class SVM for the final prediction of each emotional label [25]. Moreover, IS10 feature set has been assessed on SER using multiple pair-wise SVMs for discrete emotion classification [23]. The same feature set has also been utilized for integrating gender information in the aforementioned IS10 feature set using Denoising Autoencoders (DAEs) and next an SVM is applied for SER [48]. Moreover an Ensemble Softmax Regression (ESR) classification model has been utilized with utterance-based statistics similar to IS10 feature set [49]. Finally, a multi-task learning approach where a Deep Belief Neural (DBN) network is trained on discrete SER and continuous emotion recognition as a regression on arousal and valence using as input the same IS10 feature set [28].

Segment-based approaches: An approach of this category is pretty similar with an utterance based approach despite that the statistics are not directly applied to the whole utterance length. Instead the frames of the whole utterance are grouped in speech segments which are generally shorter in duration than the whole utterance itself. In this context, each speech segment is represented by a set of statistical functionals which are applied on LLDs which are extracted from the frames which belong to the specific time duration. Segment-based approaches have showcased that computation of statistical functionals over LLDs in appropriate timescales yields a significant performance improvement for SER systems [50], [2]. However, in these approaches the input of each emotional utterance is a sequence of vectors and thus a new challenge in training process occurs. Modeling of the input sequence of feature vectors is not a trivial task on SER tasks. A majority voting technique is employed in [51] where the decision of the final emotional label is based on the posterior probabilities of a GMM classifier which is trained and testing on a subsentence level. In contrast, this assumption does not hold in general because an angry utterance is more closely related to an event driven inference rather than the expectation of noticing angry content in all frames of a given utterance. In [52], a hybrid implementation including a simple Artificial Neural Network (ANN) from utterance-based statistics as well as a Hidden Markov Model (HMM) in order to acquire information from voiced speech segments. In [53], Provost introduced a variable window length approach for capturing emotional speech in sub-utterance scale. For every window, the presence of an affective cue (Emotion Profile (EP)) was estimated and the final emotion flow was modeled with a trajectory which served as input for an HMM. The fusion of EPs and LLDs in a hierarchical correlation model with multiple layers corresponding to each type of feature drawn from a different timescales, was used in [54]. Namely, utterance, sub-utterance and frame level feature representations were extracted for each emotional utterance and used on different layers of the hierarchical models. Emotion probabilities of every layer's unit were fed in an SVM classifier proposing a naive way of performing transfer learning [55] on SER. In other words posterior probabilities corresponding to each emotional class were learned under a different timescale and used in order to perform the final classification on utterance-level.

Direct approaches: The temporal aggregation of frame-level features using statistical functionals over speech segments or the whole utterance is not an optimal way of preserving the emotional content which is conveyed by a vocal cue [27]. Thus, this category of SER approaches usually refers to the direct modeling of emotional information derived from frame-LLDs or even raw signal itself. Consequently, the complexity of this problem is closely related to the huge length of the sequence which is supposed to be modeled. Many regions inside an utterance correspond to silence or a neutral state of speaker affective state which should be ignored for the final emotion classification [56]. Firstly, HMMs have been utilized in order to model the sequence of frame-level LLDs [57]. However, with the advent of deep learning, a lot of things in the way of modeling sequences of feature vectors has changed [58]. State-of-the-art results have been reported in many tasks such as image

classification/segmentation, ASR, object detection, etc [59] all these models have been also utilized for SER. Progressively, deeper architectures which independently learned abstract emotional context from simple local features, were applied. Namely, Deep Neural Networks (DNNs) [60], Extreme Learning Machines (ELMs) [61],[62], CNNs [35] and Recurrent Neural Networks (RNNs) [63] were trained on LLDs vectors corresponding to 1 or more consecutive frames. A special RNN model which is capable of preserving long term dependencies of input sequences and called Long Short-Time Memory (LSTM) unit [64] has been extensively used for SER tasks trained on frame-level features [65], [66]. In the latter two approaches, an attention mechanism on top of the LSTM boosts the recognition accuracy by providing a saliency detection mechanism over the huge input sequence [67]. In this way, the weights (which values are learned during training) of the attention layer are used for this purpose. Thus, regions of the input sequence corresponding to silent frames or a neutral affective state are weighted-off from the process of emotional classification. LLDs which were used in the aforementioned studies are similar to the ones discussed in the previous section (Section 1.3.2) consisting of frame descriptors falling under the three main categories of voice quality, pitch and spectrum. Consequently, the decision time-scale for each emotional state was based on frame level (30ms) or phoneme level (10-30 concatenated LLDs-vectors). In other approaches such as [44], a one-dimensional CNN is used from the raw input signal for continuous emotion recognition on both activation and valence axes.

1.3.4 State-of-the-Art Approaches

Recently, DNNs alongside with ELMs were employed [60]. LLDs were extracted from frames and were concatenated in chunks of 265ms. The highest energy segments were fed into a DNN and emotion class posterior probabilities were computed for each one of them. Statistics of these probabilities were fed into the ELM kernel for utterance-level classification, while the model was tested on the Interactive Emotional Dyadic Motion Capture (IEMOCAP) [68] database. Metallinou et al. [63] demonstrated the significance of using Bi-directional LSTMs (BLSTMs) and generally RNNs for integrating long-term temporal context in SER by experimenting in activation-valence space. Lee et al. [61] proposed an RNN-Connectionist Temporal Classification (CTC) schema in which every frame can be assigned to all emotional classes and an extra NULL label for silence frames. They applied ELM over BLSTM and showed significant improvement on the IEMOCAP by relatively boosting the accuracy around 12%.

Contemporary SER work also focuses on classifying emotion by utilizing spectrogram features with minimal speech processing. In [36] Ghosh et al. introduced innovative ways for training a BLSTM, like representation and transfer learning. The former is based on glottal flow signals while the latter learns emotion representation from activation-valence space, both methods were tested on improvised and scripted utterances. In a multi-task learning setup Xia et al. [28] has successfully utilized activation-valence information and reported state of the art results with their DBN over utterance based representations. Furthermore, multi-task learning has also been utilized with DNNs on a cross corpus experimental setup showing the great generalization potential of these methods [69]. Last but not least, transfer learning has also been investigated by combining progressive neural networks in SER tasks in one of the latest studies [70].

Building upon the tremendous expressiveness of RNNs and the necessity for emphasizing emotionally-dense frames of speech, attention models were deployed for SER. In [66] promising scores were presented on IEMOCAP. An attention-based BLSTM was employed for a better conceptual of emotional sub-utterances and yielded absolute improvement of 1.46% in accuracy from no attention model. Finally, Misramadi et al. [65] extracted frame-wise raw spectral features and LLDs for training a BLSTM. They demonstrated that LLDs outperform raw spectral features and used a weighted pooling layer with attention on top, in order to deal with the voting of emotional-irrelevant frames. In addition, a variety of pooling methods was compared, over the last hidden layer, for the final utterance classification.

In a recent study [35], different deep implementations were compared. Models were trained on

spectrogram features and state-of-the-art results on IEMOCAP were reported using CNN. In this context, CNNs [27] have been employed for SER towards a utilization of a minimal feature set of MFBs capable of preserving salient representations from frame-level descriptors. Additionally, convolutional Sparse Auto-Encoders (SAE) for learning salient features from spectrograms of emotional utterances have also been studied [37] under multiple SER experimental setups. In the latter approach, the learned feature vectors corresponds to weights of the convolutional filters after the application of a kernel able to preserve local invariant feature vectors. These vectors serve as input for an SVM training and testing.

The aforementioned feature sets, classification methods and state-of-the-art approaches comprise only a fraction of the assorted methods existing in the literature. Because the purpose of this study is not solely related to an exhaustive review of all the available methods, we suggest the reader to search additional information about literature works in review publications such as [19], [20] and [21].

1.4 Challenges

Although a variety of new challenges have been continuously spawning as new models and approaches are invented in order to build more effective SER models, we are focused on three main problems which have always been confusing SER research for years. These challenges are briefly explained below as well as their arousing questions for the complexity and the potential of actually finding a "well-rounded" solution to the problem.

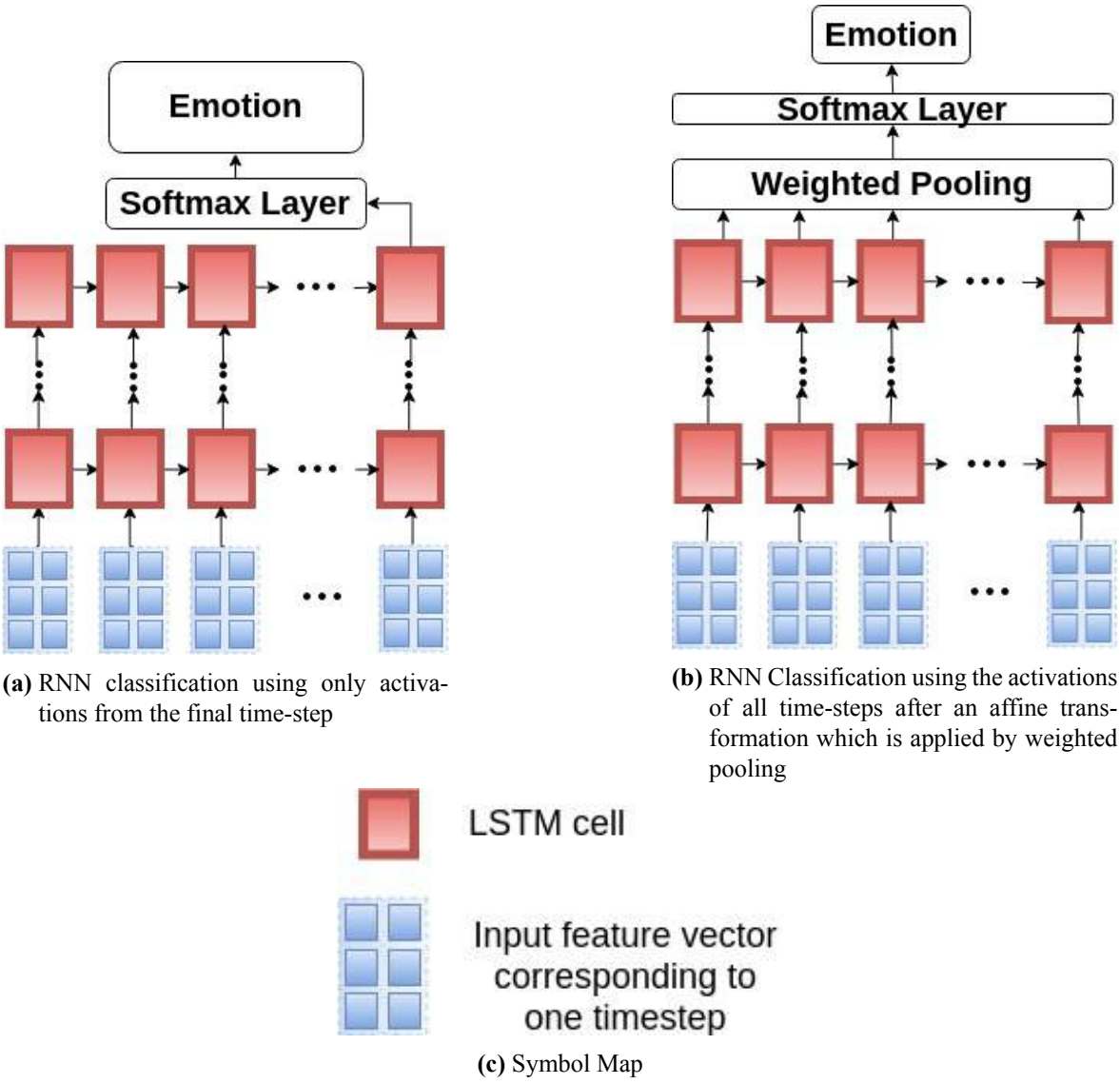
1.4.1 Timescales of Decision

In general, affective content could be captured in different timescales. The structure of the human auditory cortical processing model is highly contingent to this kind of functioning [71]. Human audio perception could be adequately modeled with a multidimensional spectro-temporal receptor of emitted sounds [71]. This is indicative of the process which brain undertakes in order to recognize vocal cues and consequently, interpret higher level information which is conveyed. Different timescales are focusing on different aspects of this information which is communicated. It is essential to understand the caveats of each timescale in order to build resilient SER models.

As we have mentioned in the previous Section 1.3.3, segment-based and direct approaches are facing the problem of modeling a sequence of features in order to train a model which can predict the emotional content of an utterance. Each time-step of the input sequence of vector could correspond to a timescale which characterizes the level from which the decision of the emotional label is drawn. This timescale which is also referred as time-level [50] could correspond to a speech duration of a frame, a phoneme, a syllable, a word, a speech segment of arbitrary length or even the utterance itself. Although the architecture of each model alters significantly the encoding of the input sequence of feature vectors, the basis of the utterance decision is nonetheless configured by the corresponding timescale of each input vector.

Recalling SER approaches in the literature, we can see that different timescales have been used for extracting acoustic features. However, the selection of each timescale for the extraction of each feature set (local or global features) is not yet analyzed carefully. This might lead to an ad-hoc selection of the respective time-scale as well as a counter intuitive way of manipulating emotional speech when the approach is based on sub-utterance frames or segments. For example, by following a segment-based approach for feature extraction and then applying an SVM on each segment in order to infer the corresponding emotion label for each one is not representative of human inference of emotions. Specifically, if an utterance consists of segments of 3 seconds and only one speech frame contains a manifestation of anger then the whole utterance should be characterized as angry despite that all the other included segments are neutral. If the selected aggregation function in order to infer the emotional label is a majority function [51] then the classification would be deemed to predict a neutral utterance instead of an angry one. The same problem exists for direct approaches where each frame of 20ms is

forced to vote for the existing emotion in this timescale [61] by running an Expectation Maximization (EM) algorithm for the classification of all frames. How could we be sure that a frame of some milliseconds could enclose discriminatory information of a happy manifestation?



(a) RNN classification using only activations from the final time-step

(b) RNN Classification using the activations of all time-steps after an affine transformation which is applied by weighted pooling

(c) Symbol Map

Figure 1.2: Emotional Classification using different RNN architectures.

Recently used models such as: HMMs [52] and RNNs [65] are modeling the input sequence of feature vectors without explicitly demanding the classification result from each time-step of the input sequence. These models during the training procedure are encoding the emotional information of the sequence by penaltizing the prediction error produced by the activations of their output layer. Although a variety of RNN architectures differ significantly from each other in terms of the temporal aggregation which is applied in order to infer the emotional content for each utterance. In figure 1.2 two of the most widely known types of RNN architectures are displayed. In Figure 1.2a, it is evident that only the activations of the final time-step contribute to the decision of the emotional class. On the contrary, in Figure 1.2bm the activations from all time-steps affect the final classification decision. Although, we should not be oblivious to the fact that RNN architectures like LSTMs could adeptly encode information of the input sequence of features in the activations of the final time-step and thus the former architecture is widely used [64]. However, if the input sequence is quite lengthy then the information encoding flow is not as smooth as it is expected to be. For example, assuming that input features correspond to LLDs extracted from 20ms frames with 0.5 overlapping ratio, then an utterance

of 4 seconds would contain a total number of 400 feature vectors. If some indicative frame level LLDs of an anger utterance are found at the beginning of the sequence and after that all the residual frames represent a neutral affective state then at the end of the sequence where the emotional decision is based the contribution of the former frames would be diminished [72]. This is why bidirectional RNNs [63] and attention mechanisms [66] are employed in order to alleviate this informational forgetting through multiple iterations of timesteps.

In essence, all of the proposed architectures in the literature could reduced to these two types of architectures [65], depicted in Figures 1.2a and 1.2b. For example, an attention model would be just another weighted pooling strategy which is used in order to specify the weights for each time-step. Consequently, how the corresponding activations of each time-step would contribute to the final emotional decision after the application of a softmax layer. The emotional content is not solely inferred by the model itself but the set of representative features which are extracted under a specific timescale. Notably, the timescale from which we draw the decision for the emotional content is closely related to the timescale of the input features despite the fact that RNNs seem to capture dependencies on multivariate timescales . The aforementioned RNN architecture do not change this timescale between the subsequent layers as CNNs do with max/mean and other pooling layers in between [73].

Considering the combination of the variety of acoustic features (local and global) which are extracted on different timescales some very interesting questions arise and are not yet fully explored.

- The timescale for inferring emotional content is different for each feature set?
- How could we determine a satisfactory timescale for each type of feature sets that could aptly capture emotional information?
- If we concatenate some feature vectors of each feature type and feed the RNN with a much smaller sequence of vectors could we boost our recognition accuracy?

1.4.2 Nonlinear Phenomena in Speech Production

The model of speech production has been extensively reviewed and analyzed through all these years [74]. In Figure 1.3 [1], the physiological system of human speech production is displayed. In essence, lungs are corresponding to the source of the power in this system which are producing the airflow. The respiratory system produces vibrations of the vocal folds in the larynx if the airflow is pressed through the glottis or a transient impulse for non-vowel verberation. The produced air vibrations are later filtered by the vocal tract which corresponds to the larynx, the pharynx and the nasal activity. Tongue, jaws, lips and velum function as articulators which alter the resonance characteristics of the upper part of the system [74].

From a more technical point of view we can model the aforementioned speech generation system with a source-filter model as defined in [6] which is also depicted in Figure 1.4. Although a variety of different models have been proposed in order to aptly reflect the true underlying dynamics of the speech generation system, we focus on the most common one which is the source-filter model [75]. The hardest component of the aforementioned model is the vocal fold in terms of analysis and formulation. Essentially, the turbulences, birfurcations and fluctuations emerging from biophysical processes result in certain chaotic phenomena in vocal production system [76]. Aperiodic vibrations of the vocal folds might be tempting to characterize them as random perturbations but instead they are exhibiting an entirely deterministic nature lying in the chaotic regime [77]. In fact, all these sounds correspond to a basin of attraction in their corresponding state-spaces (the space including all possible states of a system) [78]. The latter space is governed by periodic structures which are exhibiting recurrence patterns which are indicative of the true nature of the system under analysis [79]. The intrinsic nonlinearity of the vocal folds has also been studied for pathological conditions using narrow-band spectrograms [80]. The latter results shows that the nature of the dynamics of the vocal folds is truly nonlinear and it is not just a part of a general hypothesis of the system.

[1] Figure 1.3 was found in: http://sail.usc.edu/~lgoldste/General_Phonetics/Source_Filter/test.html

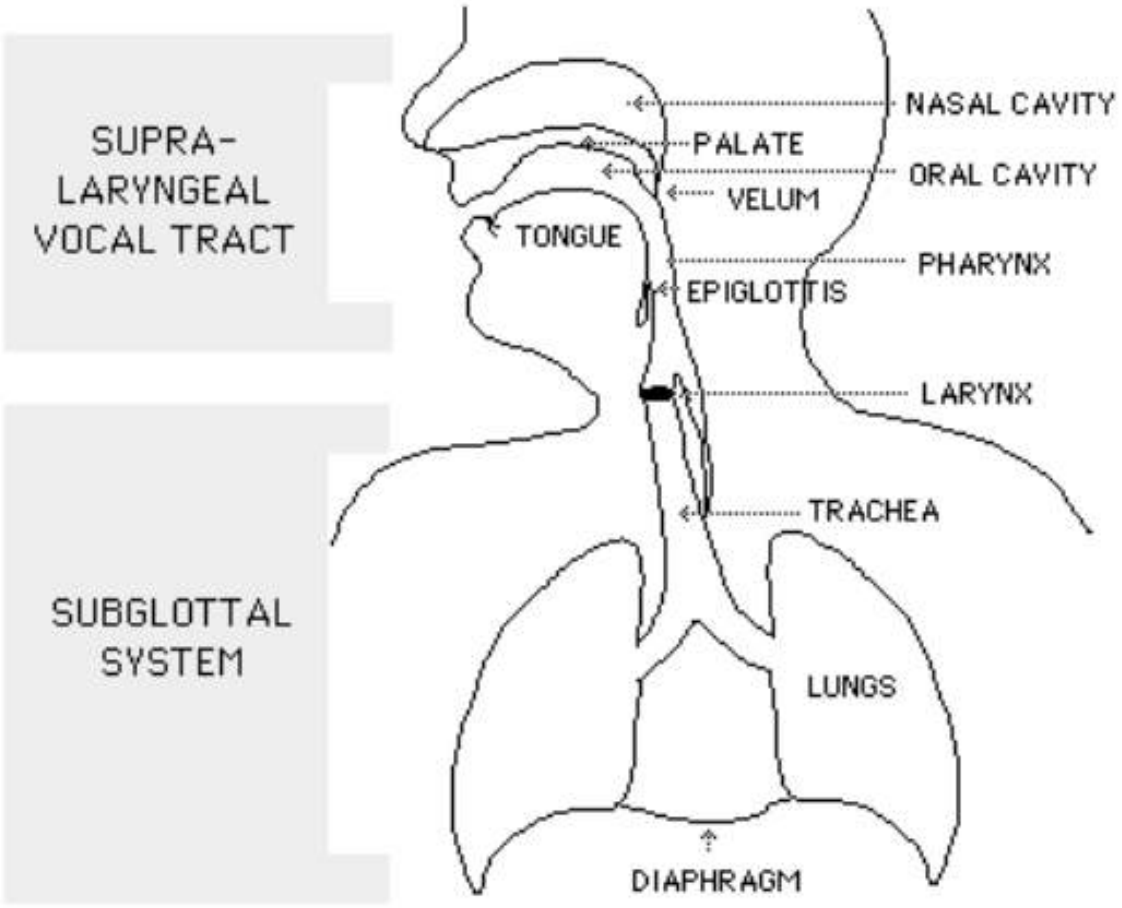


Figure 1.3: Human System of Speech Production

Most of the feature sets described in Section 1.3.2 are based on LLDs which are extracted under the assumption of a linear source-filter model of speech generation. However, vocal fold oscillations and vocal tract fluid dynamics often exhibit highly nonlinear dynamical properties which might not be aptly captured by conventional LLDs [76]. The fact that it is much easier to extract acoustic features with the assumption of linearity and stationarity of the signal, as Fourier transformation demands, makes nonlinear representations of speech to be completely neglected in traditional feature sets. Moreover, conventional feature sets contain pitch features which are related to the estimation of the fundamental frequency of the vocal tract. However, in speech production only vowels are related to periodicity regions from a dynamical system viewpoint. Silence frames between two phonemes that are pronounced correspond to random noise as well as fricative consonants in which their underlying dynamics correspond to chaotic patterns [81]. Both the aforementioned phonemes are causing noisy output in pitch related features which assume periodicity inside the extraction frame.

We could analyze the dynamical properties of the speech production system like any nonlinear dynamical system. In order to do so, one of the most prominent methods is the reconstruction of its phase space [82]. The latter space is actually an embedding of the initial signal representation in time domain to a higher dimensional space using time-delayed versions of the signal and is also called state-space. Intuitively, this is a more expressive representation of the same dynamical system which is very complex in the time domain but could be unfolded using more dimensions and appropriate time delays for the specific system [78]. For instance, in Figure 1.5 the representation of phoneme */a/* is displayed in time domain (left) and the corresponding state-space reconstruction (right). From the displayed phase space of Figure 1.5, it is quite evident that the embedded trajectories of the state-space are displaying recurrence behavior especially when the corresponding signal in time domain

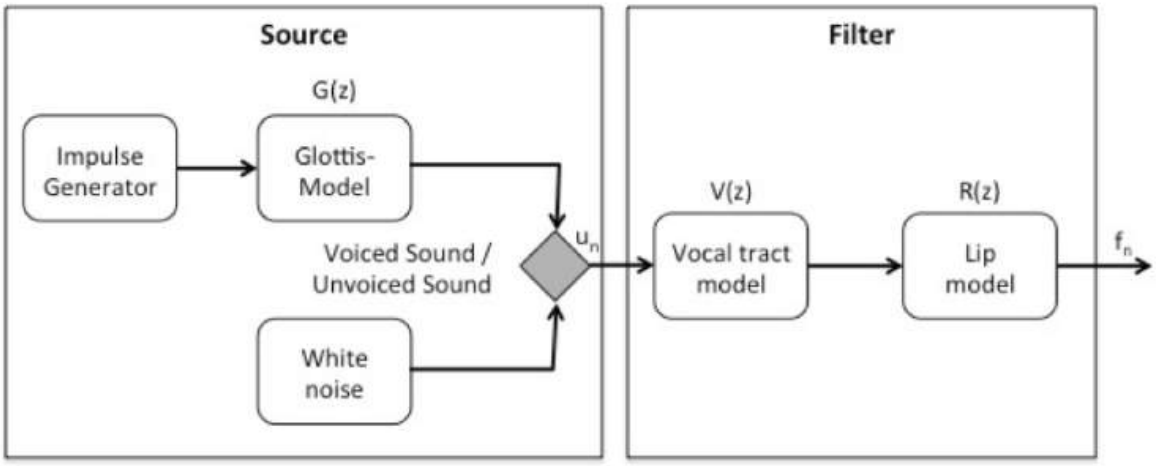


Figure 1.4: Source Filter Model of Speech Production [6]

exhibits periodicity. The trajectories which appear to be similar in the phase space or parallel under a local scale reflect the co-evolution of states for the system which is analyzed. The latter is ultimately displayed when attractors appear at the reconstructed phase space of the signal. Attractors are sets of numerical values toward which a system tends to evolve, for a wide variety of starting conditions of the system [83]. This asymptotic behavior of trajectories around strange attractors could be informative of the recurrence properties of the underlying dynamics of the system which is a speech frame in our case. However, recurrence cannot be aptly captured using only estimations of the most prominent frequencies but these visualizations underpin the exploitation of the complex structures using tools able to capture this information and utilize it for speech processing. Especially in SER tasks, where turbulences are much more prominent in emotional voice signals we should also try to capture this information for boosting the performance of our SER systems.

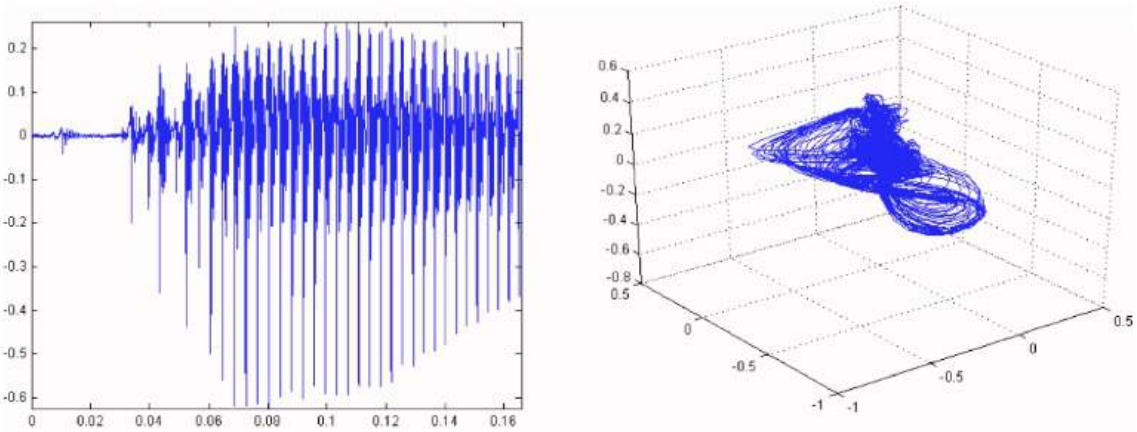


Figure 1.5: A phone /a/ and its corresponding phase space exhibiting recurrence dynamics [7]

Considering the nonlinear nature of the vocal fluid dynamics, one practical challenge is to find a set of nonlinear features which can aptly capture emotional information by taking into account the true identity of the underlying dynamics. Recurrence information of the Phase Space (PS) or state-space of speech signals is highly indicative of the true nature of the underlying dynamics and can capture non-linear dependencies of the input data [84]. Although a variety of nonlinear features has been studied for SER such as: Teager Energy Operator (TEO) [40], modulation features from instantaneous amplitude and phase [43] as well as geometrical measures from PS orbits [42], the recurrence information of the PS has not yet been investigated. Some emerging questions and challenges are the following:

- Is it possible to extract nonlinear recurrence information from speech signals in order to utilize it as acoustic information?
- How can we integrate nonlinear recurrence information for SER?
- Recurrence information and nonlinear analysis could boost the accuracy of SER systems?

1.4.3 Curse of Dimensionality and Dimensionality Reduction

Although the aforementioned feature sets in Section 1.3.2 have been shown to adeptly capture emotional content from speech signals, their feature space is quite huge and it is growing as new features are added to the previous sets. For example, the works of Schuller in [25], [23], [29] has been have been increasing the size of the proposed features for paralinguistic tasks from the previous reference. Namely, the initial 384 features [25] have been expanded to 1582 [23] and consequently resulting to 6373 feature vectors in [29]. Although the newly added features might capture significant emotional information independently, it would be quite possible that they do not offer any further discriminatory ability to the whole set of features. Of course we cannot be sure about the contribution of each feature to the recognition accuracy as the latter approach would demand an exhaustive search over the combinations produced by the power set of the feature set.

The models which these features are trained on are suffering from what is called the "Curse of Dimensionality" [85]. The latter problem refers to phenomena of decline in performance metrics, numerical instability, sparsity and dissimilarity of the input data when the model is trained on high-dimensional data. The enormous amount of feature dimensions could be catastrophic for the training process of a model because of the exponentially increased (in terms of the input dimensions) time and samples required for the process.

To this end, many feature selection techniques have been proposed [34] as well as a variety of dimensionality reduction techniques [86]. In the former approach, a subset of the whole set of features is selected using a heuristic based on average mutual information or entropy between some feature sets. On the contrary, in general dimensionality reduction techniques seek to find a lower dimensional representation $\mathbf{x} \in \mathbb{R}^n$ of the initial high dimensional features lying in $\mathbf{x} \in \mathbb{R}^D$, where usually $D >> n$ and simultaneously preserving the geometrical properties of the high dimensional space. However, reduced dimensions of the input space loose their physical meaning as the dimensionality reduction algorithm produces the higher dimensional representation irrespective of the true physical properties of the output space. Some algorithms, like PCA [33], try to produce the low dimensional representations by using linear combinations of the initial features' dimensions while other nonlinear methods suggest to reconstruct a nonlinear manifold $\mathcal{M}$ which retains the intrinsic geometry of the high dimensional space. The latter approach is often called "manifold learning" and it has been extensively used in a variety of domains where the dimensionality of the input data exceeds the computational limits posed by the architectural caveats of our hardware which is used to approximate numerically the manifold $\mathcal{M}$ [87].

If the high dimensional input data $\mathbb{R}^D$ can be described by a low dimensional manifold $\mathcal{M}$ which is embedded in a lower dimensional space of $\mathbb{R}^n$ then we can significantly lower the computational complexity and time of our models by reconstructing this manifold. This is often true for audio, text, image and multimedia data where the input vectors are described by high dimensional vectors in which the true informational units could be sparse. As a result, we could approximate the underlying data distribution $p(\mathbf{X})$, where $\mathbf{X}$ is the input data. by taking advantage of the various manifold learning methods. Interestingly, most of these manifold learning techniques are not based on the labels of the input data in order to infer the input distribution $p(\mathbf{X})$.

In rare cases where the input data distribution can be aptly approximated using 2-dimensional or 3-dimensional target spaces we gain a more intuitive representation of the distribution. Generally speaking, this is not true in high dimensional spaces where the visualization of the manifolds and the resulting spaces is missing. Without loss of generality, we focus on the visualization of different

data sources using t-Distributed Stochastic Neighbor Embedding (t-SNE) manifold learning algorithm [88].

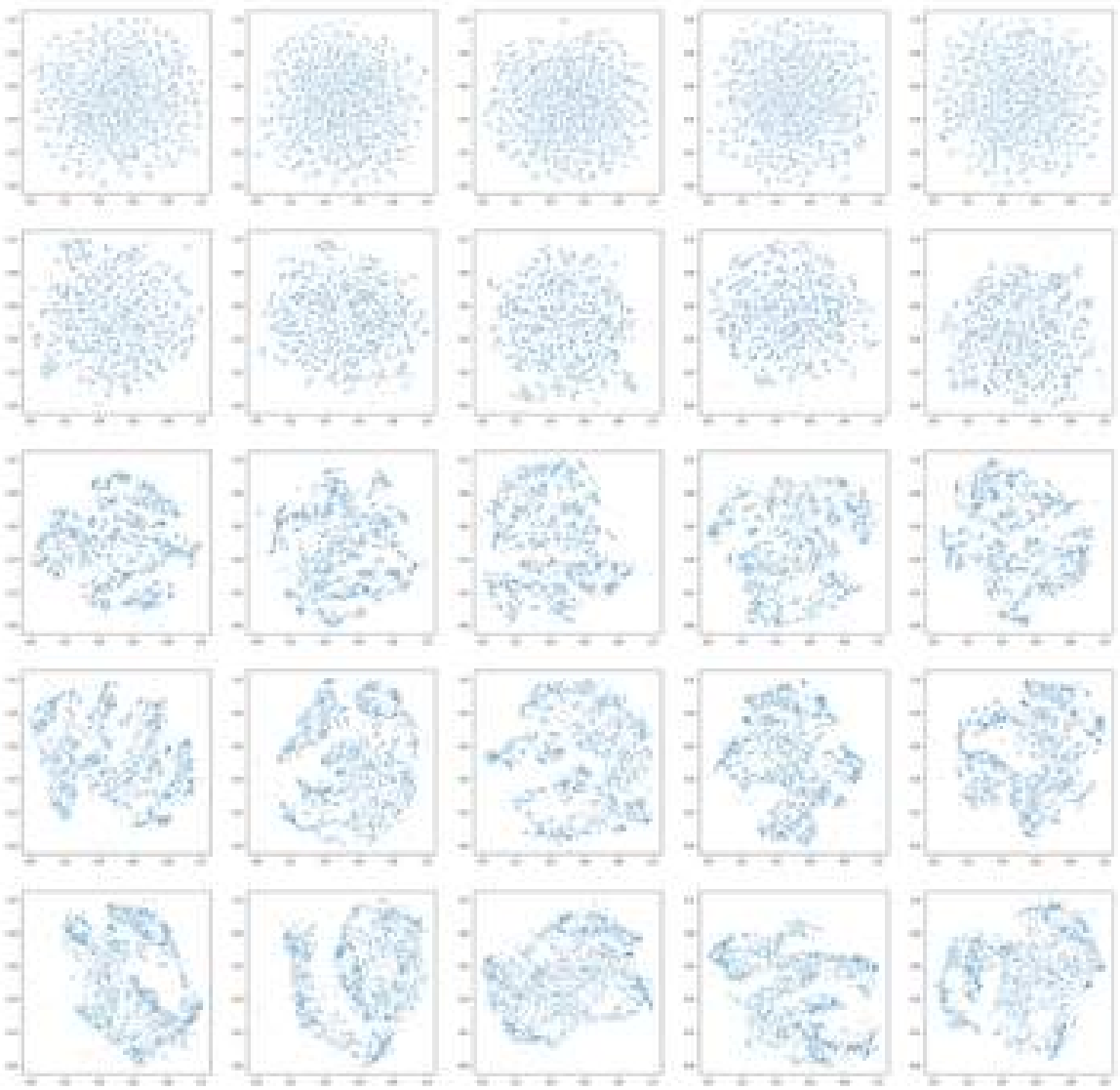


Figure 1.6: Learned two dimensional manifolds using t-SNE with different parametrization for a variety of sounds

For example, we can get a deeper insight of the underlying probability distribution of MFCCs which are probably the most common features used in speech processing. In Figure 1.6 [2] a learned 2-dimensional manifold from 512 sound files is created using t-SNE for various configurations and as input vectors statistical functionals which are applied over frame-based MFCC representations. The two parameters of t-SNE which are dispersed across the vertical and horizontal axes are the complexity and the number of iterations which will be forced to run, respectively. Now each sound waveform is represented with a 2-D vector instead of a $13 \cdot N_{stats}$, where N_{stats} is the number of statistical functionals applied on all frames represented by 13 MFCCs. It is evident that we can utilize this lower dimensional manifold which preservers the geometrical properties and structure of the high-dimensional feature space in order to perform clustering or classification with a much more squashed and less sparse feature space. In Figure 1.6 we can see that different areas of interest emerge in the two dimensional plane while points are gathering in specific areas. Presumably, points in the 2-

[2] Figure was found in: http://alexanderfabisch.github.io/t-sne-in-scikit-learn.html

dimensional planes which seem to lie close enough or lie in adjacent regions would follow a similar behavior in the high dimensional space. Consequently, we could actually visualize the discriminatory ability of MFCCs if these features are actually making the corresponding classes linearly separable in the high-dimensional space.

Building upon that, we can take advantage of the reduction of the input space and its dimensions in order to visualize the regions where the samples of each class are lying. Purportedly, if all the samples of each class are lying in areas which are easily separated on sight then the input distribution $p(\mathbf{X})$ of the high-dimensional feature vectors could be approximated using only a 2-dimensional manifold $\mathcal{M}$. Although, our focus lies in speech processing and especially SER we present the results of 2-dimensional learned manifolds by using t-SNE on both image recognition and text classification in Figures 1.7 [3] and 1.8 [4], respectively.

In Figure 1.7, the images of MNIST dataset [89] which are 28×28 images of handwritten digits are used as input for the t-SNE. The input vector of 784 pixels for each image is reduced to a 2-dimensional vector with a nonlinear manifold learning procedure. As shown in the Figure 1.7 the learned manifold does provide a strong on-sight indication that this manifold learning technique produces 2-dimensional representations where the classes of all samples are linearly separable between them. In order to offer a framework for classification we need to run the manifold learning algorithm first on the high-dimensional input, get the low-dimensional new representations learned and after that evaluate any classifier on the representations with reduced dimensions. Moreover, the same applies in other domains such as text classification where the input features correspond to vector representations of the words. In Figure 1.8, the learned 2-dimensional vectors after t-SNE application upon the initial word-embeddings is displayed for a batch of words belonging to various contexts. As it is also evident by a visual inspection of the figure, words in similar contexts are displayed similar in the 2-dimensional map. This is a strong-indication that the induced word embeddings contain discriminative information about the context in which each word belongs but under a sparse representation, in terms of the density of the included information. All in all, we can assume that we can extend these methods for SER and possibly reduce the dimensionality of the used acoustic features drastically without losing much in terms of recognition performance.

It is essential that the input data offer a sufficient sampling of the underlying distribution in order to be able to reconstruct the low dimensional manifold $\mathcal{M}$. Otherwise, manifold learning approaches like any other machine learning technique would be deemed unsuccessful. Although we utilized only t-SNE for the previous visualization analysis, there are a vast amount of different manifold algorithms presented in literature and we do not know which one would yield the most resilient solution for acoustic features used in SER tasks. There is no win-all algorithm for manifold learning but instead an interesting trade-off between expressiveness, required memory and running time is noticed [86]. Finding a robust manifold learning approach is closer to a more global machine learning problem because of the generality of solutions provided which can be adequately used in different tasks (as we saw previously with t-SNE) without any assumption about the actual identity of the source data used as input.

Nevertheless, most of the algorithms proposed in literature are based on the computation of derivatives which are computationally inefficient and less scalable on problems where smoothness cannot be assumed. Specifically, the majority of these algorithms reduce this problem to the optimization of a deterministic loss function f. Given this minimization objective, they usually employ gradient-based methods to find a global or a local optimum. In many situations, however, the loss function is non-differentiable or estimating its gradient may be computational expensive. Additionally, gradient-based algorithms usually yield a slow convergence; multiple iterations are needed in order to minimize the loss function.

All the aforementioned analysis arises a huge set of challenges which are yet to be met and questions to be answered.

[3] Figure was found in: https://medium.com/@LeonFedden/comparative-audio-analysis

[4] Figure was found in: http://colah.github.io/posts/2015-01-Visualizing-Representations/

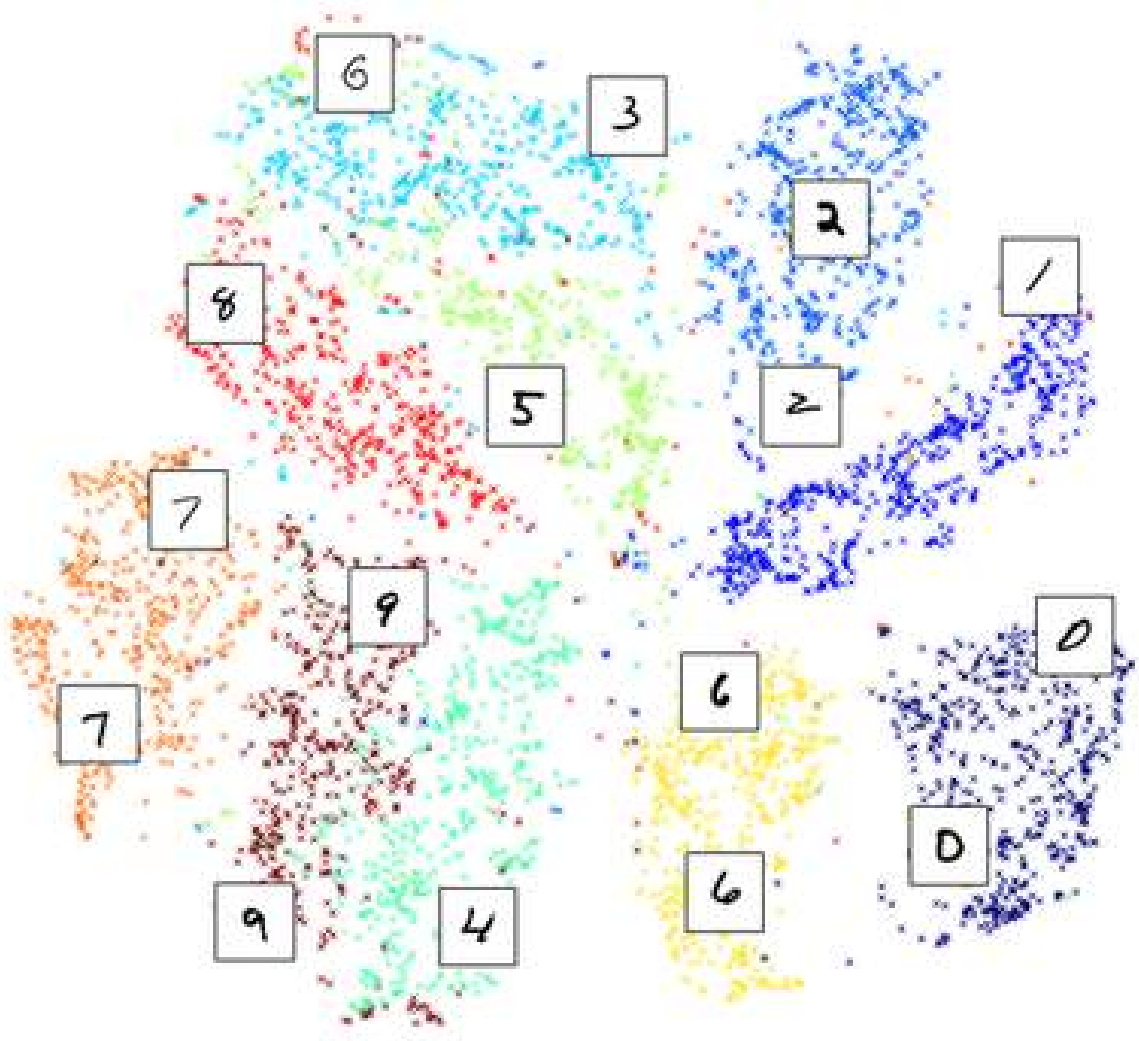


Figure 1.7: 2D manifold of MNIST dataset learned using t-SNE

- Could we provide a novel, expressive and simultaneously efficient manifold learning approach in order to perform nonlinear dimensionality reduction?
- Could we avoid the computation of derivatives without loosing the ability of the manifold approximation?
- How can we provide a theoretical proof for the convergence of an algorithm of this kind?
- Does this algorithm provide a general solution for deviant data-sources without making any assumption about the source where the feature vectors are extracted?
- Experimentation on acoustic feature sets for SER data does offer a significant improvement?
- Can we use the proposed manifold learning algorithm in order to provide visualizations of the 2-dimensional maps of different emotions, produced by a nonlinear dimensionality reduction of the input acoustic features?
- Would these emotional maps from acoustic features be similar to the ones provided in Figures 1.7 and 1.8 and provide insightful qualitative information about the features in use?

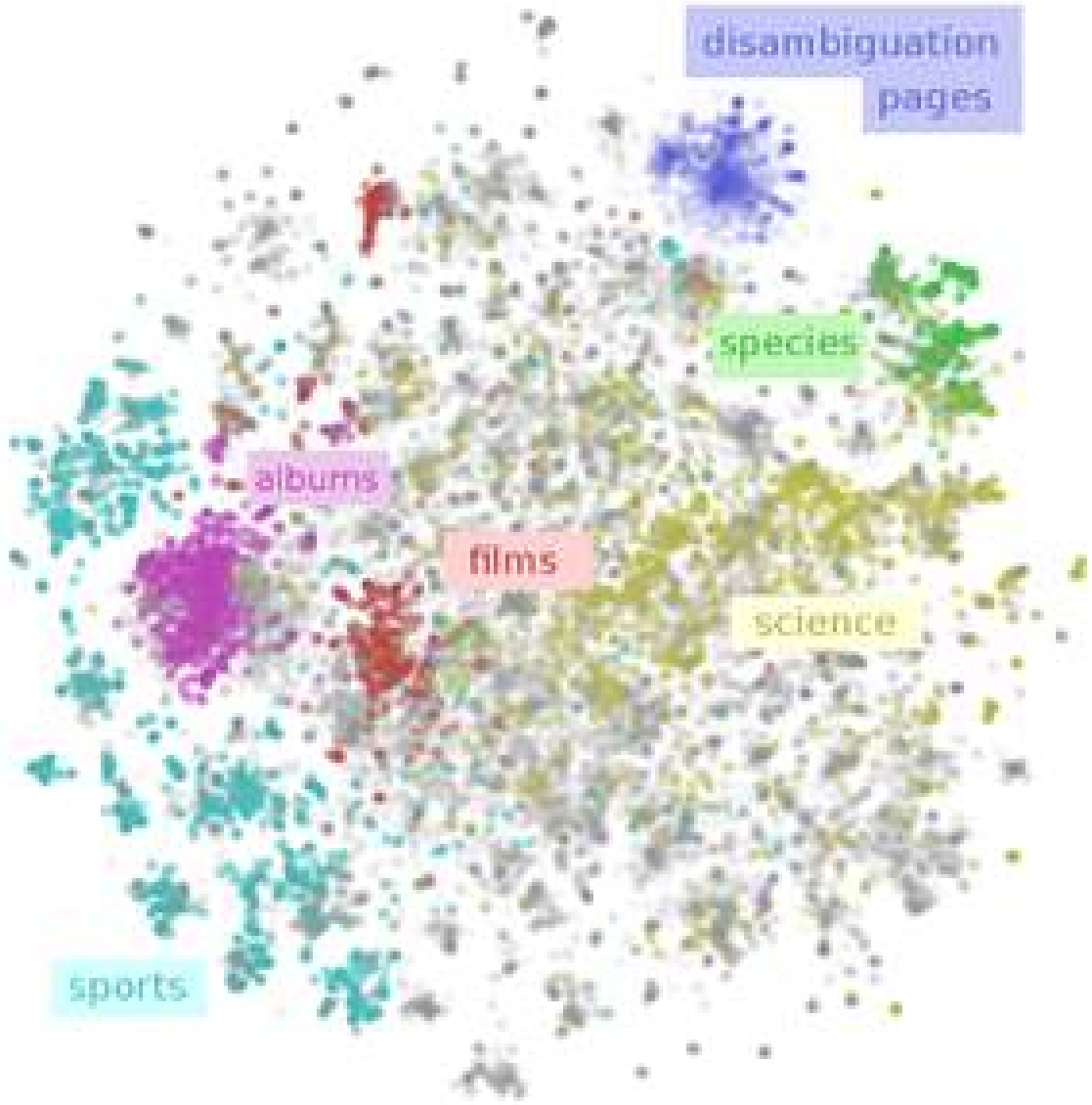


Figure 1.8: 2D manifold of word embeddings for a variety of context words

1.5 Goals and Contributions

The goals of this thesis are driven by the questions posed in Section 1.4 for all the sub-areas of interest. In other words, the success of this thesis could be rated based on:

- Whether our approaches, remarks and results give the answers to the questions posed in Section 1.4.
- If the challenges are met and tackled down for the full spectrum of our interest, as they are described in 1.4.

This thesis has a threefold contribution which can be succinctly described by the following:

1. **We investigate the effectiveness of the choice of the appropriate timescale when using RNNs for SER tasks:** We show that, the choice of the appropriate time-scale for LLDs (local features) and statistical functionals (global features) is key for a high performing SER system. We assess our approach using both local and global features and evaluate the performance at various time-scales (frame, phoneme, word or utterance). The analysis on various timescales

provides both qualitative and quantitative insights about how different feature sets capture the information on different timescales. We show that for RNN models, extracting statistical functionals over speech segments that roughly correspond to the duration of a couple of words produces optimal accuracy. We report state-of-the-art SER performance on the IEMOCAP corpus at a significantly lower model and computational complexity.

2. **We introduce a new set of nonlinear features which are able to capture patterns of recurrence dynamics and evaluate them under different models and SER tasks:** We investigate the performance of features that can capture nonlinear recurrence dynamics embedded in the speech signal for the task of Speech Emotion Recognition (SER). Reconstruction of the phase space of each speech frame and the computation of its respective RP reveals complex structures which can be measured by performing RQA. These measures are aggregated by using statistical functionals over segment and utterance periods. We report SER results for the proposed feature set on three databases using different classification methods. When fusing the proposed features with traditional feature sets, e.g., [23], we show an improvement in unweighted accuracy of up to 5.7% and 10.7% on Speaker-Dependent (SD) and Speaker-Independent (SI) SER tasks, respectively, over the baseline [23]. Following a segment-based approach we demonstrate state-of-the-art performance on IEMOCAP using a Bidirectional Recurrent Neural Network with an attention mechanism on top.

3. **We propose a novel algorithm for manifold learning which is based on derivative-free optimization instead of the conventional gradient-based approaches, namely pattern search multi-dimensional scaling (MDS)**. The latter approaches necessitate the computation of the gradient of a loss function in order to select the optimal movement in the exploration space. However, our algorithm does not require smoothness properties of the objective function which is to be minimized as well as any assumption about the differentiability of it. Specifically, we propose an extension of the classical MDS method, where instead of performing gradient descent, we sample and evaluate possible "moves" in a sphere of fixed radius for each point in the embedded space. A fixed-point convergence guarantee can be shown by formulating the proposed algorithm as an instance of General Pattern Search (GPS) framework. Evaluation on both clean and noisy synthetic datasets shows that pattern search MDS can accurately infer the intrinsic geometry of manifolds embedded in high-dimensional spaces. Additionally, experiments on real data, even under noisy conditions, demonstrate that the proposed pattern search MDS yields competitive to the state-of-the-art results on a variety of tasks such as image classification and semantic similarity of text embeddings. For SER experiments we utilize the feature sets described in Section 4 as well as their combination and visualize the learned 2-dimensional emotional maps of these acoustic feature set.

1.6 Thesis Organization

The remainder of this thesis is organized as follows. We divide the aforementioned contributions (see Section 1.5) to three Chapters 3, 4, 5. Each chapter can be considered self-contained in terms of notation, related work around the topic which analyzes, experiments and conclusions which are drawn from the results of the former. This model of organization has been selected because of the diverge methods, models and issues that are presented in each chapter.

Firstly, in Chapter 2, we introduce some basic mathematical notation (Section 2.1), we analyze some classification methods and the supervised setup within we are using them (Section 2.2). We also present the mathematical formulation and the usage of RNNs as well as how they are structured and combined with attention mechanisms and bidirectional topologies (Section 2.3). Moreover, we present the formalization for the PS reconstruction, the extraction of RPs and their corresponding RQA measures in Sections 2.4.1, 2.5 and 2.6, respectively. In Sections 2.7 and 2.8 we provide the

mathematical formulation for multidimensional scaling (MDS) and General Pattern Search (GPS) methods, correspondingly.

The first part of this work consists of the analysis of SER systems under different timescales and especially for RNN based SER systems. The aforementioned analysis is covered in Chapter 3 where we analyze the efficacy of different timescales when using RNNs for SER. Some prior motivation for this part of the work is given in Section 3.1 and an extensive review as well as a qualitative analysis of the methods for global and local acoustic feature extraction techniques is presented in Section 3.2. Furthermore, the proposed SER schemes are discussed in Section 3.3 while the preparation for conducting experiments with those architectures is conferred in Section 3.4. The final results and findings on how the decision timescale affects SER performance as well as a comparison with the literature is presented in Section 3.5.

The second part of this thesis consists Chapter 4 in which we explore the integration of nonlinear acoustic features which are reflecting the intricate recurrence dynamics of emotional speech. Specifically, in Section 4.1 we discuss the ideas which driven this approach while in Section 4.2 we list some of the most prominent approaches for utilizing information extracted from RPs and in general nonlinear speech features for SER. In Sections 4.3 and 4.4, we specify how the PS and RP are computed for each speech frame and we provide qualitative findings for these structures for different speakers and emotional manifestations. We rigorously define the acoustic feature sets that will be extracted for SER and the classification methods that we will employ in Sections 4.5 and 4.6, respectively. We present the datasets that we use for the evaluation of all three feature sets under all SER experiments as well as the final results compared with approaches in the literature in Section 4.7.

The last part of this thesis is comprised of Chapter 5 where we propose Pattern Search MDS in order to perform gradient-free nonlinear dimensionality reduction. In Section 5.1 we discuss the ideas which driven this approach while in Section 5.2 we briefly discuss some of the most prominent approaches found in literature in order to perform dimensionality reduction. We describe the proposed algorithm in section 5.3 and we also propose some of the approximations in order to make our algorithm terminate faster as well as discussing some emerging trade-offs in Section 5.4. We reduce the proposed algorithm to the general class of GPS methods in Section 5.5 and we prove its convergence theoretically using proved theorems for the unified GPS framework in Section 5.6. We assess our algorithm for assorted recognition tasks and geometry preserving tasks for the solution provided by our algorithm in Section 5.7. Specifically, we test our algorithm for SER using identical experimental setups and sets of acoustic features that were introduced in Chapter 4 (see Section 5.7.7). Last but not least, we provide a comparison for the produced manifolds in $\mathbb{R}^2$ and $\mathbb{R}^3$ from various dimensionality reduction methods in Section 5.8.

Finally, the denouement of this thesis is Chapter 6 where we summarize the conclusions we have drawn from the analysis of all the previous Chapters (see Section 6.1) as well as we point out some possible feature directions that could be based on this work (see Section 6.2).

Chapter 2

Technical Background

2.1 Notation

We denote real, integer and natural numbers as $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{N}$, respectively. Scalars are represented by no-boldface letters, vectors appear in boldface lowercase letters and matrices are indicated by boldface uppercase letters. All vectors are assumed to be column vectors unless they are explicitly defined as row vectors. For a vector $\mathbf{z} \in \mathbb{R}^n$, $\|\mathbf{z}\|_1 = \sum_{i=1}^{n} |z_i|$ is its ℓ_1 norm and $\|\mathbf{z}\|_2 = \sqrt{\sum_{i=1}^{n} z_i^2}$ is its ℓ_2 norm, where z_i is the ith element of $\mathbf{z}$. Moreover, the latter notation could be defined either as $z(i)$. By $\mathbf{A} \in \mathbb{R}^{n \times m}$ we denote a real-valued matrix with n rows and m columns. Additionally, the jth column of the matrix $\mathbf{A}$ and its entry at ith row and jth column are referenced as $\mathbf{a}_j$ and a_{ij}, respectively. We can also define the entry of a matrix $\mathbf{A}$ at ith row and jth column as $\mathbf{A}_{ij}$ but this would be specifically defined for each equation separately. The trace of the matrix $\mathbf{A}$ appears as $tr(\mathbf{A})$ and its Frobenious norm as $||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}^2}$. The square identity matrix with n rows is denoted as $\mathbf{I}_n \in \mathbb{R}^{n \times n}$. For the matrices $\mathbf{A} \in \mathbb{R}^{n \times m}$ and $\mathbf{B} \in \mathbb{R}^{n \times m}$ we indicate their Hadamard product as $\mathbf{A} \odot \mathbf{B}$. The n-ary Cartesian product over n sets $S_1, ..., S_n$ is denoted by $\{(s_1, ..., s_n) : s_i \in S_i.\ 1 \leq i \leq n\}$ Finally, $\mathbf{X}^{(k)}$ refers to the estimate of a variable $\mathbf{X}$ at the kth iteration of an algorithm. We define the conditional probability of the event Ω given that the event Ω' has already happened with: $p(\Omega|\Omega')$. In order to make the notation easier for the mapping of functions $f : \mathbb{R} \to \mathbb{R}$ in vectors of real numbers $\mathbf{x} = [x_1, ..., x_n]$, we assume that the direct application of the function to the vector we use $f(\mathbf{x})$ as the intuitive extension of function f to the $\mathbb{R}^n$ where $f(\mathbf{x}) = [f(x_1), ..., f(x_n)]$. We denote with $\mathbf{a}||\mathbf{b}$ the concatenation of vectors $\mathbf{a}$ and $\mathbf{b}$. We define the element-wise multiplication for two vectors $\mathbf{a}$ and $\mathbf{b}$ with $\mathbf{a} \odot \mathbf{b}$.

2.2 Classification Models

In this section we will analyze the classification models which are used in this thesis. In this work, when we confront a classification task then we will assume a supervised machine learning task where the label $\mathcal{Y}_i$ of each sample $\mathbf{x}_i$ of the training set is known. Generally speaking, there are many other setups where some labels are known (semi-supervised learning) as well as setups where no label of the input data is available on the training time (unsupervised learning) [58]. After we train our model, we evaluate our model using a separate subset of the available data which is called "testing set". No instances should belong to both the "training" and "testing" sets.

2.2.1 Loss Function

We optimize the parameters $\boldsymbol{\theta}$ of our model by minimizing an objective function $\mathcal{J}(\boldsymbol{\theta})$ (also called loss function). In supervised training, the loss function is also defined by the training $\mathbf{X}_{\mathbf{train}}$ and their corresponding labels $\mathcal{Y}_{train}$. In SER and other tasks we usually have more than two discrete classes for our data and thus we are using a categorical cross entropy as our objective function which we are trying to minimize. For instance, if we have N training samples and $\mathcal{C}$ emotional labels, then we can convert the categorical labels to one-hot vectors of length $\mathcal{C}$. In the latter representation, each vector

has zeros everywhere else except of the index which corresponds to the categorical label of a sample $\mathbf{x}_k$. Namely, we convert the categorical label as follows:

$$\mathcal{Y}_k \rightarrow \quad \mathbf{y}_k = [0, ..., \underbrace{1}_{corresponding\ \ label\ \ index\ \ c}, ..., 0] \tag{2.1}$$

Now index c in this vectorized representation represents the same class as $\mathcal{Y}_k$ which is the true annotated label corresponding to sample $\mathbf{x}_k$. In this supervised learning setup, each sample corresponds to a tuple of 1) a data-vector representation $\mathbf{x}_k$ and 2) its vectorized label representation $\mathbf{y}_k$.

Assuming that our model is trained on a set of N data samples $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{N}$ the prediction $\hat{\mathbf{y}}_k$ for kth tuple is defined as the posterior probability for all the available classes $\mathcal{C}$ given the parameters of the model $\boldsymbol{\theta}$ and its data-vector representation $\mathbf{x}_k$. If we define as $\mathbf{h}(\mathbf{x}_k, \boldsymbol{\theta})$ the activations of our model which are used for the inference of the label of a given sample $\mathbf{x}_k$ we can assume that the predicted distribution vector $\hat{\mathbf{y}}_k$ is defined element wise by the respective posteriors probabilities:

$$\hat{\mathbf{y}}_k = [p(\hat{\mathbf{y}}_{k,1}|\boldsymbol{\theta}, \mathbf{x}_k), ..., p(\hat{\mathbf{y}}_{k,\mathcal{C}}|\boldsymbol{\theta}, \mathbf{x}_k)] \tag{2.2}$$

Hence, the loss function for the kth sample is shown next:

$$\mathcal{J}(\boldsymbol{\theta})_k = -\mathbf{y}_k \cdot log(\hat{\mathbf{y}}_k) \tag{2.3}$$

More or less, the prediction vector of our model corresponds to the likelihood of the corresponding posteriors for each class. If we would like to optimize the parameters of our model by maximizing the likelihood of our model we could easily do it by reversely minimizing the average of the negative log likelihood from all the available N training samples. The objective function is shown in the following equation:

$$\mathcal{J}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{k=1}^{N} \mathbf{y}_k \cdot log(\hat{\mathbf{y}}_k) \tag{2.4}$$

Where N is the number of the training samples. Essentially, the aforementioned equation is particularly useful when training NNs using the value of the loss function $\mathcal{J}(\boldsymbol{\theta})$ for computing the error at the final error of the NN in terms of the classification decisions which made for the N samples. In most cases we do not use the whole dataset in every iteration of the training process but instead we compute the error over subsets of the training set (these sets are often called batches when our model is a neural network). In order to learn the optimal parameters for our models we could employ any parameter optimizer which is based on the computation of the gradient $\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})$ for finding a local minima or either tries to find the local minima by directly moving points (see next Section 2.8). If our model is an NN the most widely used algorithm in order to optimize its weights is backpropagation [90]. In brief, the error computed by the partial derivatives of each layer configures the alteration of the weights of the subsequent layer by following the exact similar procedure computing chains of partials derivatives in respect of the parameters to be tuned.

All in all, the loss function $\mathcal{J}(\boldsymbol{\theta})$ is just a heuristic method in order to be able to approximate a local minima for the nonlinear minimization problem which generally it cannot be solved by any algorithm in polynomial time (these types of problems are called NP-problems). Moreover, the feature space of the data samples $\{\mathbf{x}_i\}_{i=1}^{N}$ is generally complex and nonlinear. The extracted feature-vectors are entangled in subspaces where the vectors which are lying in these subspaces belong to different classes. Hence, the problem of finding the optimal discriminatory regions of the high-dimensional input space becomes non-trivial and requires a careful consideration of both feature extraction and model optimization [59]. For instance, in Figure 1.7 it is evident that the feature vectors which correspond to digits of "9" and "4" are not linearly separable and thus the discriminatory region is much harder to find than other regions for other digit-pairs such as "0" and "7" which seem to be completely unrelated from their representation on the 2-dimensional plane.

Building upon that, a variety of different classification models have been employed in order to find these discriminatory regions over any feature space. We will provide the mathematical formulation of the classification models which we are using in this work as well as a brief explanation of how they work.

2.2.2 Support Vector Machines (SVMs)

As we have previously mentioned, feature vectors of each class live entangled with feature vectors from other classes and thus they are not linearly separable. In other words, one cannot easily find a hyperplane of the input feature space serving as a classification boundary for data belonging to each class of the training set. SVMs are trying to find maximum-margin hyperplanes in order to create these classification boundaries between the vectors of each class [91]. Hence, each binary classification problem can be reduced to finding the separating hyperplane which has the maximum margin between the distance of the nearest points of each class (A distance of a point to a hyperplane is just the length of the projection of this point vertically to the hyperplane).

In a more rigorous way we can define a binary classification problem for a sequence of N training samples $\{(\mathbf{x}_k, \mathcal{Y}_k)\}_{k=1}^{N}$ where $\mathcal{Y}_k \in \{-1, 1\}$. We can assume that the parameters $\boldsymbol{\theta}$ of an SVM is the coefficients of the decision hyperplane which are to be found. SVMs are solving the following problem:

$$\begin{aligned} &\underset{\boldsymbol{\theta},\boldsymbol{b},\boldsymbol{\zeta}}{min} \tfrac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\theta} + C\sum_{i=1}^{N}\zeta_i \\ &constrains: \qquad\qquad \mathcal{Y}_i(\boldsymbol{\theta}^T\phi(\mathbf{x}_i) + b) \geq 1 - \zeta_i \end{aligned} \tag{2.5}$$

where $C > 0$ is a regularization term for configuring the penalty term of wrongly classified instances (practically feature vectors $\mathbf{x}$ which do not follow the constraints of Equation 2.5 and $\phi(\cdot)$ is a nonlinear map which is used in order to transform the input data-vectors. To this end, in order to acquire nonlinear relations between data points we can perform the kernel trick [92] using the aforementioned maps $\phi(\cdot)$. This allows the SVM training algorithm to fit the maximum-margin hyperplane in a transformed feature space which the boundaries between the two classes are much easier to be found than in the initial input space. Furthermore, as the classification boundaries are hyperplanes then the support vectors would be parallel with the alteration of the constant ($\boldsymbol{\theta}^T\phi(\mathbf{x}_i) + b = 1$ and $\boldsymbol{\theta}^T\phi(\mathbf{x}_i) + b = -1$). Then the distance between those support vectors would be $\frac{2}{||\boldsymbol{\theta}||}$. This is an extra verification of why Equation 2.5 serves as the objective function of an SVM.

Since we apply the kernel trick $\mathcal{K}(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x_i})^T\phi(\mathbf{x_j}))$, we can convert the problem defined above to its dual and also use the new transformations of the feature space.

$$\begin{aligned} &\underset{\boldsymbol{\alpha}}{min} \tfrac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\theta} - \mathbf{1}_N^T\boldsymbol{\alpha} \\ &constrains: \qquad \sum_{i=1}^{N}\mathcal{Y}_i\alpha_i = 0 \\ &\qquad\qquad\quad 0 \leq \alpha_i \leq C, \;\; i \in \{1, .., N\} \end{aligned} \tag{2.6}$$

where $\mathbf{Q}_{i,j} = \mathcal{Y}_i\mathcal{Y}_j\mathcal{K}(\mathbf{x_i}, \mathbf{x_j})$ and $\mathbf{1}_N = [1, 1, ..., 1]$ is a vector of ones in $\mathbb{R}^N$ space.
Hence, the decision boundary would be defined by the following equation:

$$sgn(\sum_{i=1}^{N}\mathcal{Y}_i\alpha_i\mathcal{K}(\mathbf{x}_i, \mathbf{x}) + b)) \tag{2.7}$$

where b is the intercept of the defined model and $sgn(\cdot)$ is the function of sign where -1 corresponds to negative values and 1 to positive ones as well as 0 for zero-values.

A list of kernel functions $\mathcal{K}(\mathbf{x_i}, \mathbf{x_j})$ which are widely used:

- Linear kernel: $\langle \mathbf{x_i}, \mathbf{x_j} \rangle$
- Polynomial kernel of degree d and a bias parameter r: $(\gamma\langle \mathbf{x_i}, \mathbf{x_j} \rangle + r)^d$

- Radial Base Function (RBF) kernel regularization parameter γ: $exp(-\gamma||\mathbf{x_i} - \mathbf{x_j}||^2)$
- Sigmoid kernel with regularization parameter γ and a bias parameter r: $tanh(\gamma\langle\mathbf{x_i}, \mathbf{x_j}\rangle + r)$

We can easily extend the previous formulation of binary decision SVMs in multi-class problems by simply training separate binary classifiers for all the classes available in the training data. After that, the binary classifier with the maximum confidence score is selected as the final decision of the multinomial SVM. This method is also called "one-versus-rest".

2.2.3 Logistic Regression (LR)

A simpler classification algorithm is Logistic Regression (LR) classifier in which the parameters of the model $\boldsymbol{\theta}$ again correspond to coefficients of a hyperplane likewise in SVMs (see Section 2.2.2). We can utilize the formulation defined in the previous subsections of Section 2.2 in order to portray the mathematical foundations of LR classifier.

We will again describe the simple problem for a binary classifier which has a similar setup and formulation as described in Section 2.2.2. The N training samples $\{(\mathbf{x}_k, \mathcal{Y}_k)\}_{k=1}^{N}$ consist the input feature vectors and labels where $\mathcal{Y}_k \in \{0, 1\}$. The activation of the LR classifier is determined by applying a sigmoid function (see Equation 2.8 below) over the fitted line in order to get the final classification decision.

$$sigmoid(z) = \frac{1}{1 + e^{-z}} \tag{2.8}$$

The activation of LR for a given vector $\mathbf{x}$ would be defined as follows:

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = sigmoid(\boldsymbol{\theta}^T\mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T\mathbf{x}}} \tag{2.9}$$

The loss function to be minimized while fitting the LR to the training data is the following:

$$\frac{1}{2}||\boldsymbol{\theta}||_q + C\sum_{i=1}^{N} log(exp(-\mathcal{Y}_i(\boldsymbol{\theta}^T\mathbf{x}_i + b)) + 1) \tag{2.10}$$

where $C > 0$ and b represent the coefficients of penaltization of wrong classifications and the intercept of the hyperplane (same as Section 2.2.2). $||\cdot||_q$ is the norm used to define the distance between feature vectors which is to be penaltized during the training process (for $q = 1$ and $q = 2$ we compute Manhattan ℓ_1 and Euclidean ℓ_2, respectively). The second part of the above objective function is the exact same as the one described in Section 2.2.1 while the first one regularizes the amplitude of the parameters in order to avoid numerical stability errors due to the large numerical values which might be produced during the optimal line inference. In order to find the parameter values for the LR classifier we then again need to minimize the loss function (see Equation 2.10) by computing the gradient of this loss function and try to fit LR under maximum likelihood with stochastic gradient descent [93].

2.2.4 K-Nearest Neighbors (KNNs)

KNN is a non-parametric classifier. By this we mean that no parameters of this model are available to tune or optimize as well as that KNN does not make any assumptions about the probability distribution of the input data. There is actually no training process at all. For a given vector $\mathbf{x}'$ we would like to estimate its corresponding label $\mathcal{Y}'$. In order to do this we just need to compute the K nearest neighbors to the test vector $\mathbf{x}'$. We can define the set of the K nearest neighbors as shown next:

$$\mathcal{U} = \underset{i}{argmin}||\mathbf{x}_i - \mathbf{x}'||_q \tag{2.11}$$

where $\mathbf{x}_i$ is the ith sample of the training data. $||\cdot||_q$ is the norm used to define the distance metric between the test vector and any training sample (for $q = 1$, $q = 2$ or $q = \infty$ we compute Manhattan, Euclidean or Supremum norm, respectively).

Now for the aforementioned set of nearest neighbors we find the label which is represented by the majority of the data vectors $\{\mathbf{x}_i \mid i \in \mathcal{U}\}$. Hence, the corresponding label data of the input vector is estimated using the labels of the nearest neighbors, in terms of the distance metric we have selected, inside the space of the training feature vectors.

2.3 Recurrent Neural Networks (RNNs)

Going a step beyond simple models which require fixed-length representations of the input data, RRNs show their supremacy when they are used with input sequences. Namely, we alter the definition of the supervised problem by converting the input for the ith sample from just a vector $\mathbf{x}_i$ to a sequence of vectors $\{\mathbf{x}_{ij}\}_{j=1}^{T}$ where T is the number of timesteps or the length of the input sequence of feature vectors. This is particularly useful when modeling input data like audio and text where the underlying time dependencies completely alter the procedure of inferring information in classification setups and more [59]. The core functionality of an RNN is to infer the underlying dynamics of the input sequence temporal behavior by keeping an internal state and updating it for different timesteps.

In essence, RNNs are creating replicas of a cell generator with different parameters over different timesteps. In this way, the recurrence connections could be seen as connections between hidden nodes of the same level while input and output represent connections with subsequent layers. This is exactly the topology of any typical DNN except that RNN can be optimized using specific timesteps and not across the full timelength of the unrolled model. RNNs need to be set with the maximum length which could be used either as test or training in prior to the utilization of the model itself.

A simple visualization of an RNN is displayed in Figure 2.1. The input vector $\mathbf{x}$ could also be seen as a sequence of vectors $\{\mathbf{x}_j\}_{j=0}^{T}$ which are parts of the initial feature vector. It is evident that the recurrent connections are transferring information for the State $\mathbf{S}$ vectors as new timesteps are computed. In this way, every activation corresponding to the instance of the RNN cell at timestep t is computed using information from the previous state $\mathbf{S}_{t-1}$ as well as the current input vector $\mathbf{x}_t$. So actually unrolling an RNN is just a way of better visualizing the behavior and the underlying mechanisms of this NN but the core functionality is the same in both representations. The overall architecture of an RNN is formed as a directed graph along the sequence of the input vectors.

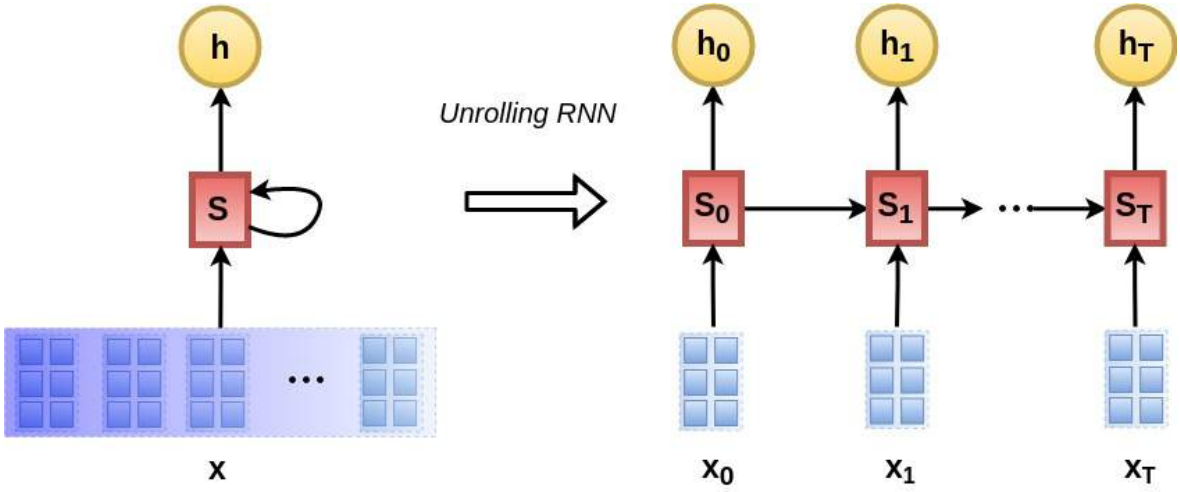


Figure 2.1: Unrolling of a Recurrent Neural Network with an input sequence of $T + 1$ timesteps

It is easy to infer that we could stack piles of RNN layers on top of each other by simply connecting the activation of each cell at timestep t, which is $\mathbf{h}_t$, as an input to the next RNN layer for the same

timestep. By this we create deep recurrent networks which are able to better encode the informational flow between the vectors of the input sequence and infer higher level attributes which might not be adequately inferred using shallow networks [59].

2.3.1 Long Short Term Memory (LSTM) unit

Building upon the RNN topology, it is limpid that every aspect of this NN facilitates the temporal modeling of the input sequence. However, the long-term dependencies of the input sequence $\{\mathbf{x}_j\}_{j=0}^{T}$ can be problematic as the computation which is performed arises highly complex composition of functions and nonlinear behavior. The recurrent topology of the network entails the computation of the gradient and its flow over multiple timesteps which yields the famous problem of vanishing or exploding gradient when the error is tried to be propagated backwards for the optimization of the network parameters [72]. One way for tackling this problem and enabling an effective way of construction and training of an RNN is the LSTM unit [64] which has a forget gate in order to be able to clip very small and very high gradient values. In Figure 2.2 the block diagram of an LSTM cell is displayed which will be analyzed further below.

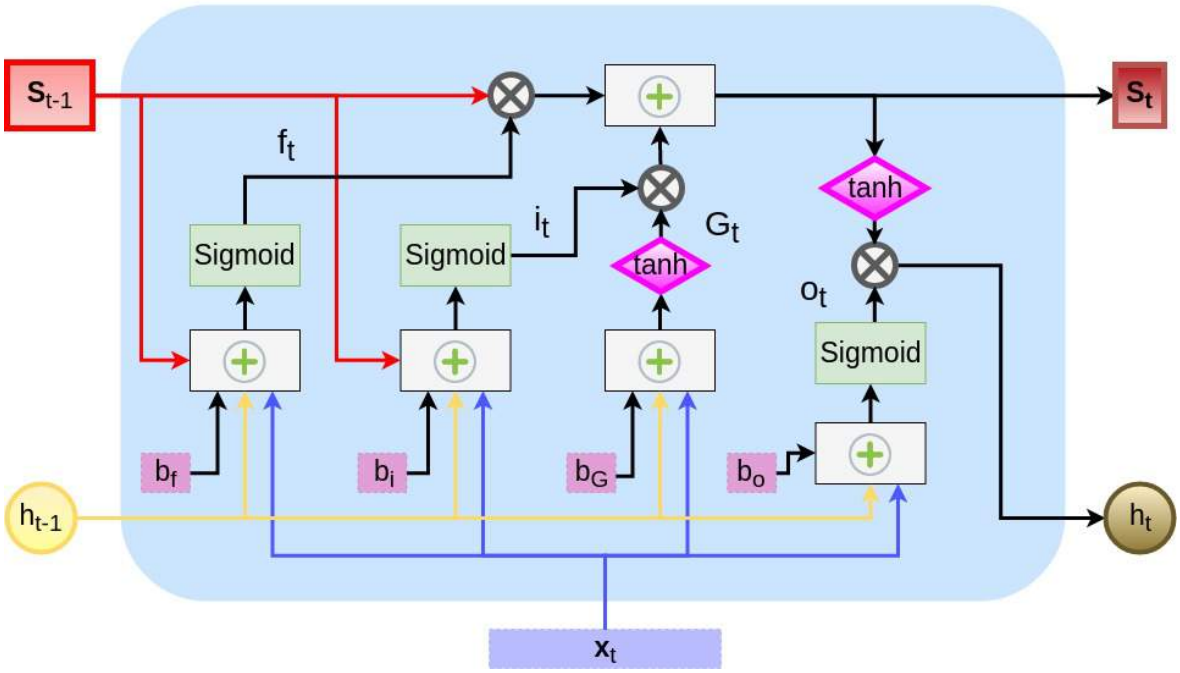


Figure 2.2: Block Diagram of a Long Short-Time Memory unit

The multipliers represented with gray "x" inscribed in circles consist the core elements of the LSTM architecture are the following:

1. The forget gate controls the informational flow for the history from previous timesteps $\mathbf{S}_{t-1}$
2. The input gate controls the informational flow for the input vector $\mathbf{x}_t$ at the current timestep
3. The output gate controls the information flow from the current state $\mathbf{S}_t$ (which has been already computed) to the final activation of the cell $\mathbf{h}_t$ at the current timestep

Going a step deeper in the functionality of the LSTM cell which will be used extensively for the experimental setup of this thesis we will analyze the architecture displayed in Figure 2.2.

First of all the multiplications are performed element-wise for the input vectors. Moreover the rectangle with a green plus represents any aggregation function for two or more input vectors like: concatenation, element-wise average, multiplication, adding, maximum or minimum. The application

of the nonlinear functions $sigmoid(\cdot)$ and $tanh(\cdot)$ are represented by a green rectangle and a purple rhombus, respectively. The activations for the current timestep $\mathbf{h}_t$ and the previous timestep $\mathbf{h}_{t-1}$ serve as the output and one of the inputs of this cell, respectively. In addition they are depicted using yellow circles. Finally, the state vectors $\mathbf{S}_t$ and $\mathbf{S}_{t-1}$ represent the weights of the neural network which are able to save internally the state for each timestep and follow a similar input/output explanations as previously. Moreover, the current timesteps are colored using darker colors comparing to the same vectors which serve as input for the cell at timestep t in order to show the time dependence between these pairs. The biases $\mathbf{b}$ are just vectors with numbers which are added to the final vector representation when the latter are passed from an aggregation function. The same schematic representation is followed in all figures in this Section. We will also elaborate on each element of the block diagram using strict mathematical formulation below.

The forget gate is modeled as follows:

$$\mathbf{f}_t = sigmoid(\mathbf{W}_f \cdot (\mathbf{h}_{t-1}||\mathbf{x}_t) + \mathbf{b}_f) \tag{2.12}$$

where $\mathbf{W}_f$ represents the weights vectors corresponding to the forget gate, and $\mathbf{b}_f$ is the respective bias. In this way, the output will be a number between zero and one which the former corresponds to forget the input of the previous activation. In contrast, a value of 1 in this gate represents that the information of the previous activation $\mathbf{h}_{t-1}$ alongside with the information from the current input vector $\mathbf{x}_t$ would be fully considered for the computation of the state of this LSTM. The same applies for all the other gates which are displayed in Figure 2.2.

The input gate controls the information which will flow from the activation of the previous timestep $\mathbf{h}_{t-1}$ alongside with the information from the current input vector $\mathbf{x}_t$ when we are trying to update the current state weights $\mathbf{S}_t$. These are modeled using the following equations:

$$\mathbf{i}_t = sigmoid(\mathbf{W}_i \cdot (\mathbf{h}_{t-1}||\mathbf{x}_t) + \mathbf{b}_i) \tag{2.13}$$

$$\mathbf{G}_t = tanh(\mathbf{W}_G \cdot (\mathbf{h}_{t-1}||\mathbf{x}_t) + \mathbf{b}_G) \tag{2.14}$$

where $\mathbf{W}_G$ and $\mathbf{W}_i$ represent the weights of the network for the gate of the input informational flow control as well as the respective biases $\mathbf{b}_i$ and $\mathbf{b}_G$.

In order to be able to compute the updates on for the current state $\mathbf{S}_t$ we are adding the information which has been controlled by the aforementioned gates (input and forget gates). Namely the current state is computed as shown next:

$$\mathbf{S}_t = \mathbf{i}_t \odot \mathbf{G}_t + \mathbf{f}_t \odot \mathbf{S}_{t-1} \tag{2.15}$$

Next in order to compute the output at the current timestep t, we need to combine the precomputed state vector $\mathbf{S}_t$ after a nonlinear function $tanh(\cdot)$ is applied as well as information from the input vector and the activation from the previous timestep at the output gate multiplier. Specifically, the following equations express the activation vector at the current timestep:

$$\mathbf{o}_t = tanh(\mathbf{W}_o \cdot (\mathbf{h}_{t-1}||\mathbf{x}_t) + \mathbf{b}_o) \tag{2.16}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{S}_t) \tag{2.17}$$

2.3.2 Bidirectional LSTM

Although LSTMs are able to encode temporal dependencies from input sequences of vectors it is quite common that the length of the input sequence is quite long in order to be trained adequately using backpropagation. The main problem is that the temporal dependence when training very long sequences is diminishing as sequential computations over multiple timesteps alter the gradient flow [65] and consequently the informational flow. In order to tackle this problem we can encode the input

sequence from the beginning to the end (forward RNN) and also in reverse (backward RNN) and in the end combine the activations of the two RNN layers in order to find the output activation for each timestep. We will focus only on Bidirectional LSTMs (BLSTMs) because we only use the LSTM unit for our RNN models.

In Figure 2.3, a BLSTM architecture is depicted with both forward and backward LSTM layers combined in order to produce the corresponding activation. The red rectangles with solid perimeter correspond to the forward architecture which is also evident from the direction of the recurrent activations and consequently the informational flow. On the other hand, backward LSTM cells for each timestep are represented using green rectangles with dotted lines as perimeter.

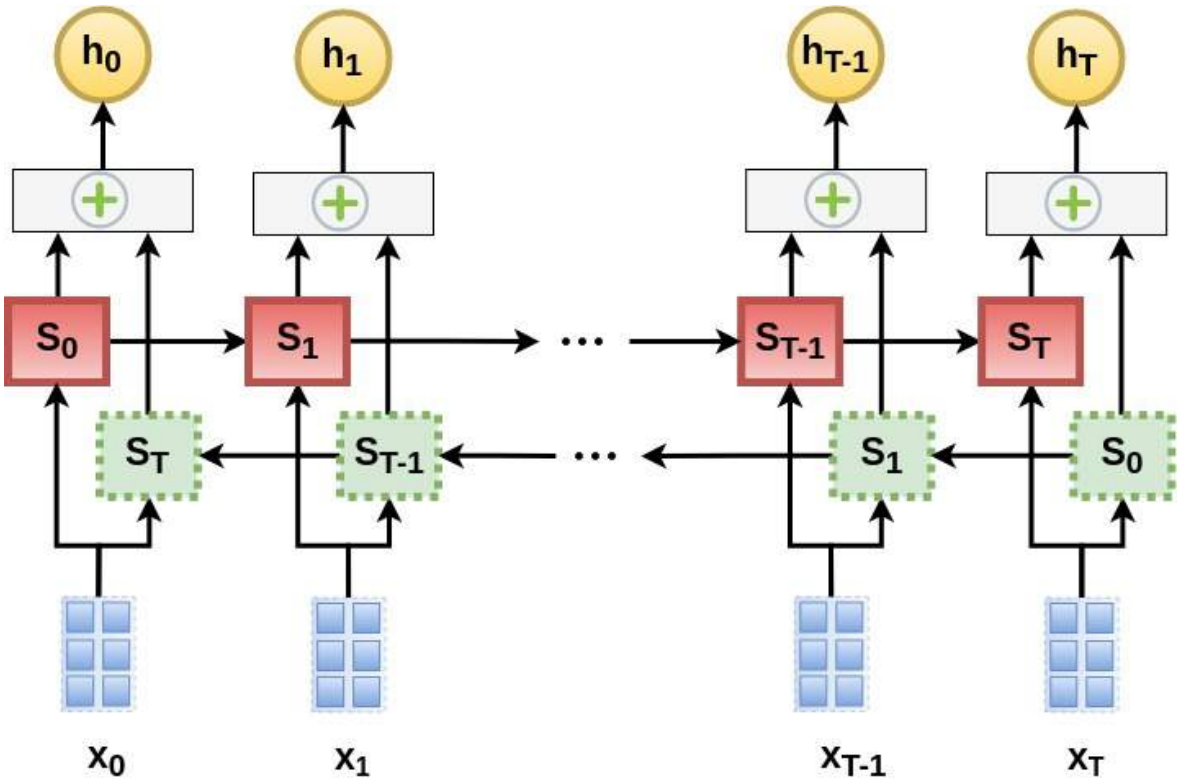


Figure 2.3: Bidirectional LSTM layer

Specifically, we combine the two architectures by separately computing the forward activation $\overrightarrow{\mathbf{h}}_t$ at timestep t as well as the corresponding backward activation $\overleftarrow{\mathbf{h}}_{T-t}$ and concatenating them for computing the final activation at each timestep. To this end, the activation at timestep t is simply the concatenation of the aforementioned vectors, namely: $\mathbf{h}_t = \overrightarrow{\mathbf{h}}_t || \overleftarrow{\mathbf{h}}_{T-t}$. The same applies for all the $T+1$ timesteps of the input sequence. We can extend the BLSTM architecture topology by stacking extra BLSTM layers on top by just connecting the activations from the forward architecture $\overrightarrow{\mathbf{h}}_t$ to the input of the next forward LSTM layer. We do the same for the backward architecture by connecting $\overleftarrow{\mathbf{h}}_{T-t}$ to the input of an LSTM layer which is one level higher. Finally, we postpone the concatenation of both activations at the current timestep to the last layer of the BLSTM architecture.

2.3.3 Attention Mechanism

An attention mechanism is applied over the sequence of hidden state vector which are drawn from the top layer of an RNN in order to provide a mechanism for focusing on specific parts of the input sequence which might be more indicative for the final classification or regression. Specifically a context vector $\mathbf{C}$ is used in order to enable the attention mechanism to learn what to attend based on the current input sequence as well as the encoded information so far [67]. Hence, the output of this

layer will be able to neglect useless activations from timesteps which are completely uninformative for the classification or regression purpose we utilize the RNN architecture. For example, in SER we expect that input sequences of short segment feature vectors will contain irrelevent information at the beginning and the ending timesteps which might be neglected by the attention mechanism by learning weights which are close to zero. These segments might correspond to silence durations which are typical to the start and end times of utterances.

In Figure 2.4 an attention mechanism is depicted by using a context vector **C** and the input sequence of the RNN activations $\{\mathbf{h}_t\}_{t=0}^{T}$. We represent the softmax function with a light red rectangle without perimeter. The softmax value of an input vector **x** with length $T+1$ would be given element-wise by the following equation:

$$softmax(\mathbf{x})_k = \frac{e^{x_k}}{\sum_{t=0}^{T} e^{x_t}}, \quad 0 \leq k \leq T \tag{2.18}$$

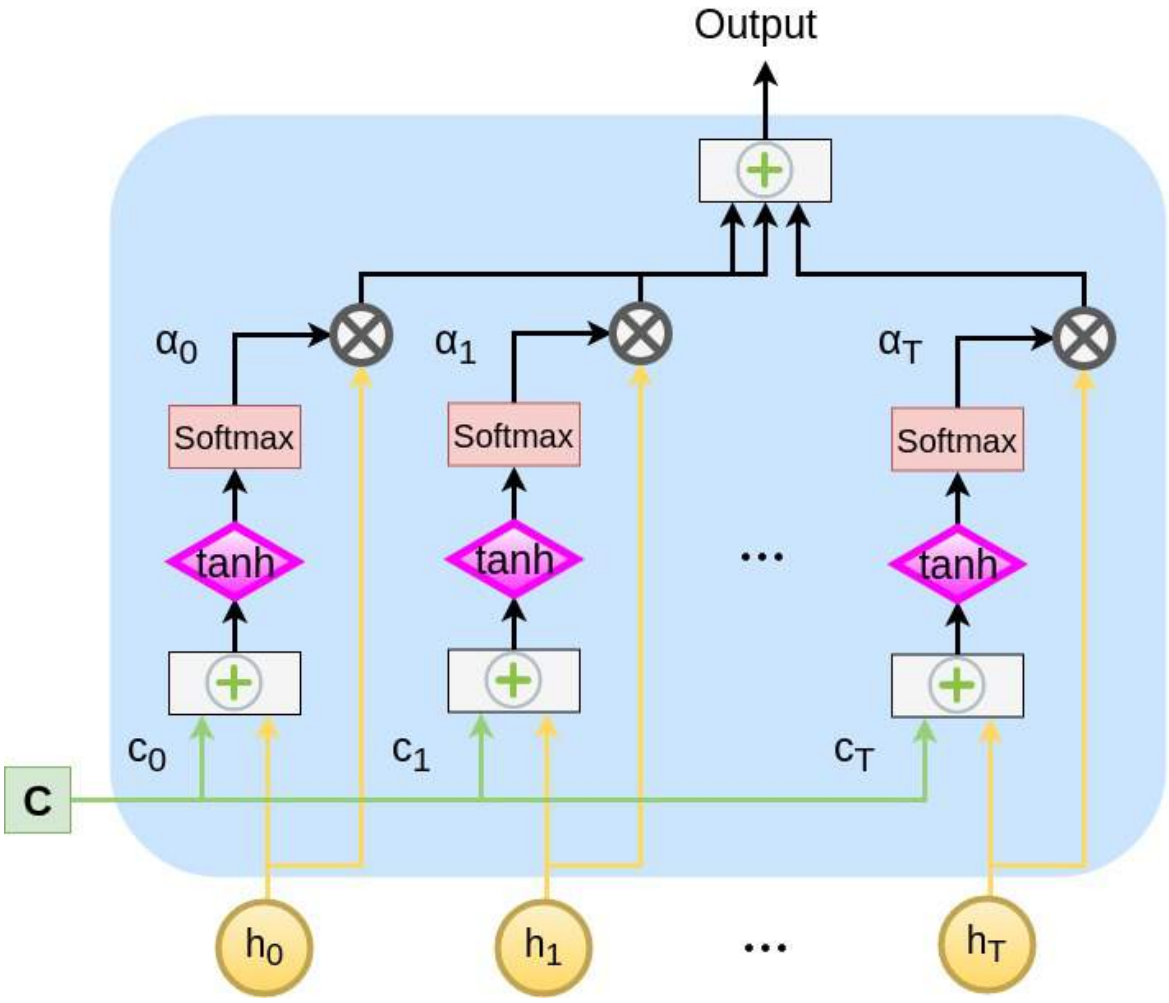


Figure 2.4: Attention Mechanism for given activations from an RNN

In order to truly understand how the attention mechanism work it is essential to specify how the context vector is computed. The context vector **C** is computed as the average of all activations from all timesteps, formally we would write:

$$\mathbf{C} = \frac{1}{T+1} \sum_{t=0}^{T} \mathbf{h}_t \tag{2.19}$$

For each timestep we combine the context vector and the corresponding input hidden state vector as follows in order to get the internal representation r:

$$\mathbf{r} = tanh(\mathbf{W}_c \cdot (\mathbf{h}_t || \mathbf{c}_t) + \mathbf{b}_c) \tag{2.20}$$

where $\mathbf{W}_c$ and $\mathbf{b}_c$ are the trainable parameters of the attention layer.

The attention layer learns the corresponding attentive weights α for all the timesteps using the previously defined internal vector $\mathbf{r}$, as follows:

$$\mathbf{a} = softmax(\mathbf{r}) \tag{2.21}$$

Hence the vector α will contain the corresponding weights in order to multiply for getting the final output representation of the attention layer. Namely, the output would be defined as: $\sum_{t=0}^{T} a_t \odot \mathbf{h}_t$. It easy to infer that if we sum all the values from the attention vector α we would get one as a result. This is indicative of the process we follow where all the vectors of the input sequence are normalized to one like a probability vector.

2.3.4 Attention-based Bidirectional LSTMs

If we combine the BLSTM architecture which was presented in Section 2.3.2 with an attention mechanism (see previous Section 2.3.4) we are able to create an Attention-based BLSTM (A-BLSTM). Specifically, we need to connect the outputs of the final layer of a BLSTM to the input of the attention layer in order to get the final output representation vector for the input sequence. Now we can forward the output vector of the overall architecture to subsequent layers in order to perform classification or regression using discrete or continuous labels, respectively. Following the notation from the previous Sections, we combine the hidden state of the final timestep $\mathbf{h}_T = \overrightarrow{\mathbf{h}}_T || \overleftarrow{\mathbf{h}}_0$ (which is able to encode the information from the input sequence) with the output vector of the attention mechanism in order to get the final representation of the input sequence.

2.4 Phase Space Reconstruction

2.4.1 Definition

$$\mathbf{x}(i) = [s(i), s(i+\tau), ..., s(i+(d_e-1)\tau)] \tag{2.22}$$

where d_e is the embedding dimension of the reconstructed PS and τ is the time lag. If the embedding theorem holds and the aforementioned parameters are set appropriately, then the orbit defined by the points $\{\mathbf{x}(i)\}_{i=1}^{N}$ would truthfully preserve invariant quantities of the true underlying dynamics which are assumed to be unknown [82]. In the study of dynamical systems, the delay embedding theorem specifies the conditions under which the dynamics of the original system $\mathbf{s}^\star$ can be reconstructed from a sequence of observations $\{\mathbf{x}(i)\}_{i=1}^{N}$ of the state of the dynamical system. A successful reconstruction preserves the properties of the true dynamical system $\mathbf{s}^\star$ that do not change under smooth changes in its coordinates such as diffeomorphisms. However it is not assured that the geometrical properties of the attractor are not identical to the ones of the approximated manifold $\mathcal{M}$.

Multitude of researches have been trying to identify optimal parameters for the reconstruction of the PS. In accordance with [78], parameters τ and d_e for each speech frame can be estimated individually by using Average Mutual Information (AMI) [94] and False Nearest Neighbors (FNN) [95], respectively. Both approaches and extensive analysis for the selection of the aforementioned parameters will be presented in next sections.

2.4.2 Average Mutual Information (AMI)

First of all, Average Mutual Information (AMI) is a measure of nonlinear correlation between the given signal $\{s(i)\}_{i=1}^{N}$ and a time delayed version of this signal by τ samples e.g., $\{s(i+\tau)\}_{i=1}^{N-\tau}$. If

we want to be precise we have to use the same number of samples for each signal and thus we consider the signals $\{s(i)\}_{i=\tau+1}^{N}$ and $\{s(i+\tau)\}_{i=1}^{N-\tau}$.

Formally, the AMI for a signal $\{s(i)\}_{i=1}^{N}$ and a specified time-lag τ as shown next:

$$\mathcal{I}(\{s(i)\}_{i=1}^{N}, \tau) = \sum_{i=1}^{N-\tau} p_b(s(i), s(i+\tau)) \cdot log_2[\frac{p_b(s(i), s(i+\tau))}{p_b(s(i)) \cdot p_b(s(i+\tau))}] \tag{2.23}$$

where $p_b(s(i), s(i+\tau))$ defines the joint probability function which is derived from the histogram of values of the signal $\{s(i)\}_{i=1}^{N}$ as well as the marginal probabilities $p_b(s(i))$ and $p_b(s(i+\tau))$. Specifically, the values of these probabilities are computed over some specified number of bins. In this work, we use $N_{bins} = 32$. Any digital signal has discrete integer values, thus, we notate s_{max} and $s_{min} = -s_{min}$ the maximum and minimum integer values that a discretized signal could take, respectively. Without loss of generality we could assume that the values that a digital signal could take are equally distributed for negative and positive integers. In order to extract the probability distribution we need the histogram of the initial signal $\mathcal{H}(\{s(i)\}_{i=1}^{N}, s_{max}, N_{bins})$ which is defined for each bin as the number of indexes of the input signal for which the values of the signal are lying inside that bin. Prior to that we define the following sets which divide the signal into the aforementioned N_{bins}:

$$\mathcal{A}_k = \{i \ \ \forall i \in \{1, ..., N\} \ \mid \ [s(i) + s_{max}] \ \ div \ \ 2s_{max} = k - 1\} \tag{2.24}$$

where $a \ \ div \ \ b$ defines the integer division between the numbers a and b and $k \in \{1, ..., N_{bins}\}$ is the index of the bin. Moreover, in order to sustain a more easier to read notation we define the corresponding bin-index for each sample of the input signal $s(i)$ using the following notation:

$$ind_{\mathcal{A}}(s(i)) = [s(i) + s_{max}] \ \ div \ \ 2s_{max} + 1 \tag{2.25}$$

Consequently, we describe equivalently the sets $\{\mathcal{A}_k\}_{k=1}^{N_{bins}}$ using the latter notation as shown below:

$$\mathcal{A}_k = \{i \ \ \forall i \in \{1, ..., N\} \ \mid \ ind_{\mathcal{A}}(s(i)) = k\} \tag{2.26}$$

Additionally, we could define the value for the kth bin as follows:

$$\mathcal{H}(\{s(i)\}_{i=1}^{N}, s_{max}, N_{bins}, k) = card(\mathcal{A}_k) \tag{2.27}$$

where $card(\mathcal{A}_k)$ defines the cardinality of the set or the number of the elements of the set $\mathcal{A}_k$ and again $k \in \{1, ..., N_{bins}\}$ is the index of the corresponding bin. However, in our case we also need to create the corresponding set for all the available pairs of indexes i and $i + \tau$ which are reflecting the correlation of the values of the signal for the specified time-lag Similarly, we define a set for all the available pairs of the input signal as defined below:

$$\mathcal{B}_{jk} = \{i \ \ \forall i \in \{1, ..., N\} \ \mid \ ind_{\mathcal{A}}(s(i)) = j, \ \ ind_{\mathcal{A}}(s(i+\tau)) = k\} \tag{2.28}$$

where $k \in \{1, ..., N_{bins}\}$ is again the index of the corresponding bin.

In this context, we can easily define the marginal probabilities $p_b(s(i))$ for all the indexes of the input signal $\{s(i)\}_{i=1}^{N}$ individually using the aforementioned sets $\{\mathcal{A}_k\}_{k=1}^{N_{bins}}$, as shown next:

$$p_b(s(i)) = \frac{card(\mathcal{A}_{ind_{\mathcal{A}}(s(i))})}{\sum_{k=1}^{N_{bins}} card(\mathcal{A}_k)} \tag{2.29}$$

Likewise in the previous equation we can specify also the joint probabilities $p_b(s(i), s(i+\tau))$ for all the indexes of the input signal and their corresponding index-pairs which are delayed by τ samples as shown below:

$$p_b(s(i), s(i+\tau)) = \frac{card(\mathcal{B}_{ind_{\mathcal{A}}(s(i)), ind_{\mathcal{A}}(s(i+\tau))})}{\sum_{j=1}^{N_{bins}} \sum_{k=1}^{N_{bins}} card(\mathcal{B}_{jk})} \tag{2.30}$$

2.4.3 Takens Theorem

In order to successfully unfold the dynamics of the system we need to specify the embedding dimension d_e for the points of the reconstructed PS. In order to do so we should be careful that for the selected embedding dimension the aforementioned "embedding theorem" holds. This theorem is also called "Takens-theorem" which suggests that if we increase the embedding dimension d_e or equivalently the number of time-lagged versions of the given signal then the dynamics of the approximated manifold $\mathcal{M}$ which is comprised of the points $\{\mathbf{x}(i)\}_{i=1}^{N}$ becomes much more deterministic. Specifically, there should be a diffeomorphism $\phi : \mathcal{M} \rightarrow \mathbf{R}^{d_e}$ that maps the approximated manifold of points $\{\mathbf{x}(i)\}_{i=1}^{N}$ into $\mathbf{R}^{d_e}$ by simultaneously preserving that the gradient of the mapping ϕ has full rank [82]. The aforementioned statements hold if:

$$d_e \geq 2d_{BCD}(\mathcal{M}) + 1 \tag{2.31}$$

where d_{BCD} corresponds to the box counting dimension of the approximated smooth manifold $\mathcal{M}$:

$$d_{BCD}(\mathcal{M}) = \lim_{\epsilon \rightarrow 0} \frac{log(N(\epsilon))}{log(\frac{1}{\epsilon})} \tag{2.32}$$

where $N(\epsilon)$ is the number of boxes of side length equal to ϵ which are required to cover the manifold $\mathcal{M}$.

In other words, the embedding dimension d_e should be large enough in order to successfully unfold the dynamics of the state-space of the original system. Otherwise, it would be possible that trajectories of the reconstructed phase space to be collapsed in close regions and thus the points $\{\mathbf{x}(i)\}_{i=1}^{N}$ lying on the resulting manifold $\mathcal{M}$ could mistakenly appear as neighbors while this is not true for the original system.

2.4.4 False Nearest Neighbors (FNN)

In order to specify the embedding dimension d_e we will use the criterion of false nearest neighbors [95]. By using this criterion we gradually increase the embedding dimension and compute a ratio which is indicative of the number of neighbors for each point of the trajectory. In essence, we compute the distances $\mathbf{D}_{\hat{d_e}}(\mathbf{x}(i), \mathbf{x}(j))$ and $\mathbf{D}_{\hat{d_e}+1}(\mathbf{x}(i), \mathbf{x}(j))$. These distances correspond to the values of the distance metrics between the points $\mathbf{x}(i)$ and $\mathbf{x}(j)$ of the reconstructed phase space trajectory for different embedding dimensions. If we assume that the approximated manifold $\mathcal{M}$ will be embedded in the euclidean space $\mathbb{R}^{\hat{d_e}}$ then we could define the distance metric between the points $\mathbf{x}(i)$ and $\mathbf{x}(j)$ as follows:

$$\mathbf{D}_{\hat{d_e}}(\mathbf{x}(i), \mathbf{x}(j)) = ||\mathbf{x}(i) - \mathbf{x}(j)||_2 = \sqrt{\sum_{k=0}^{\hat{d_e}-1} [s(i + k \cdot \tau) - s(i + k \cdot \tau)]^2} \tag{2.33}$$

Now for all the points we compute the relative percentage of change for the distance between any pair of points on the phase space trajectory when we use an extra dimension for the embedding dimension. Namely we compute:

$$R_{FNN}^{\hat{d_e}}(\mathbf{x}(i), \mathbf{x}(j)) = \frac{\mathbf{D}_{\hat{d_e}+1}(\mathbf{x}(i), \mathbf{x}(j)) - \mathbf{D}_{\hat{d_e}}(\mathbf{x}(i), \mathbf{x}(j))}{\mathbf{D}_{\hat{d_e}}(\mathbf{x}(i), \mathbf{x}(j))} \tag{2.34}$$

When the average value of the above ratio matrix $\mathbf{R}_{FNN}^{\hat{d_e}}$ over all its values exceeds a value of $(0.10 - 0.15)$ then this means that there is a big difference on the distance between the points of the PS orbit $\mathbf{x}(i)$ and $\mathbf{x}(j)$ when we add an extra dimension. As a result, this means that point $\mathbf{x}(i)$ has a false neighbor when we use an embedding dimension of $\hat{d_e}$ instead of a bigger one $\hat{d_e} + 1$. Ideally, we would like to estimate an embedding dimension d_e which achieves a number of false nearest neighbors

which is as close to zero as possible. Although there is a trade-off because as we increase the number of dimensions which are used for the embedding we also increase the dimensions of the input data to be analyzed. This is also an example of the general problem of the "curse of dimensionality" (see Section 1.4.3). Generally speaking, a ratio value of false nearest neighbors for a selected embedding dimension around 10% seems sufficient for the reconstruction of the dynamics [78].

2.5 Recurrence Plots (RPs)

Given a PS trajectory $\{\mathbf{x}(i)\}_{i=1}^{N}$ we analyze the recurrence properties of these states by calculating the pairwise distances and thresholding these values in order to compute the corresponding RP [96].

Sometimes we do not proceed with the thresholding in order to produce plots with continuous values which are basically the visualization of locally normalized distance matrices. These plots sometimes are also called unthresholded recurrence plots or continuous recurrence plots [84]. Formally, we could write the distance between the two points as follows:

$$\mathbf{D}_q(\mathbf{x}(i), \mathbf{x}(j)) = ||\mathbf{x}(i) - \mathbf{x}(j)||_q \tag{2.35}$$

where $||\cdot||_q$ is the norm used to define the distance between any two trajectory points $\mathbf{x}(i)$ and $\mathbf{x}(j)$. Specifically, for $q = 1$, $q = 2$ or $q = \infty$ we compute Manhattan, Euclidean or Supremum norm, respectively. Generally speaking, the distance matrix could represent any valid norm.

RPs are binary square matrices and are defined element-wise as shown next:

$$\mathbf{R}_{i,j}(\epsilon, q) = \Theta(\epsilon - ||\mathbf{x}(i) - \mathbf{x}(j)||_q) \tag{2.36}$$

where $\Theta(\cdot)$ is the Heaviside function and ϵ is the thresholding value. Thus, matrix $\mathbf{R}$ consists of ones in areas where the states of the orbit are close and zero elsewhere. The measure of proximity is defined by threshold ϵ for which multiple selection criteria have been studied [97]. We consider three criteria depending on: 1) a fixed ad-hoc threshold value, 2) a fixed Recurrence Rate (RR) as defined in the next Equation 2.40 (e.g., For $RR = 0.15$ we set ϵ according to a fixed probability of the pairwise distances of PS's points $P(||\mathbf{x}(i) - \mathbf{x}(j)||_q < \epsilon) = 0.15,\ \ 1 \leq i, j, \leq N$), and 3) a fixed ratio of the standard deviation σ of points $\{\mathbf{x}(i)\}_{i=1}^{N}$, e.g., $\epsilon = 5\sigma$ [98]. For fixed values of ϵ and q we denote as $\mathbf{R}_{i,j}$ the respective entry of the RP matrix for simplicity of notation.

The main structures emerging in RPs and we are able to capture consist of lines and point of the binary matrix. It is essential to understand the main categories of lines which are able to quantify information about an RP:

An L-length diagonal line (of ones) is defined by:

$$(1 - \mathbf{R}_{i-1,j-1})(1 - \mathbf{R}_{i+L+1,j+L+1}) \prod_{k=1}^{k=L} \mathbf{R}_{i+k,j+k} = 1 \tag{2.37}$$

An L-length vertical line is described by:

$$(1 - \mathbf{R}_{i,j-1})(1 - \mathbf{R}_{i,j+L+1}) \prod_{k=1}^{k=L} \mathbf{R}_{i,j+k} = 1 \tag{2.38}$$

An L-length white vertical line (of zeros) is defined as:

$$\mathbf{R}_{i,j-1}\mathbf{R}_{i,j+L+1} \prod_{k=1}^{k=L} (1 - \mathbf{R}_{i,j+k}) = 1 \tag{2.39}$$

Of course a single point would correspond to an entry of the binary matrix $\mathbf{R}$ but without belonging to one the previously mentioned categories of lines. So any structure of RPs based on single

points would correspond to a noisy image filled with isolated points similar to images produced when inducing pepper noise.

We also denote with $P_d(l)$, $P_v(l)$ and $P_w(l)$ the histogram distributions of lengths of diagonal, vertical and white vertical lines, respectively. Hence, the total number of these lines are correspondingly $N_d = \sum_{l \geq d_m} P_d(l)$, $N_v = \sum_{l \geq v_m} P_v(l)$ and $N_w = \sum_{l \geq w_m} P_w(l)$, where $d_m = 2$, $v_m = 2$ and $w_m = 1$ define the minimum lengths for each type of line [84].

All these structures reflect the recurrence dynamics of the system under analysis by the diverging patterns for each system type. In Figure 2.5 [1] we can see four different type of systems that are captured from their one-dimensional time-series representations and how their corresponding RPs look like. It is evident that RPs reflect the dynamics of each type of system by different structures on lines and points. Noisy data would also result in RPs with uncorrelated structure (noisy images). Periodicity is also evident in the second image for both frequencies by watching the parallel to the diagonal lines which reflect the co-evolution of the orbits of the PS's trajectories for subsequent time-frames. Finally, the autoregressive process and determinism of the time-series are evident by the existence of extreme events around the binary plot by some areas which are activated and others who are not. Generally speaking, the texture of these images are representative of the underlying system.

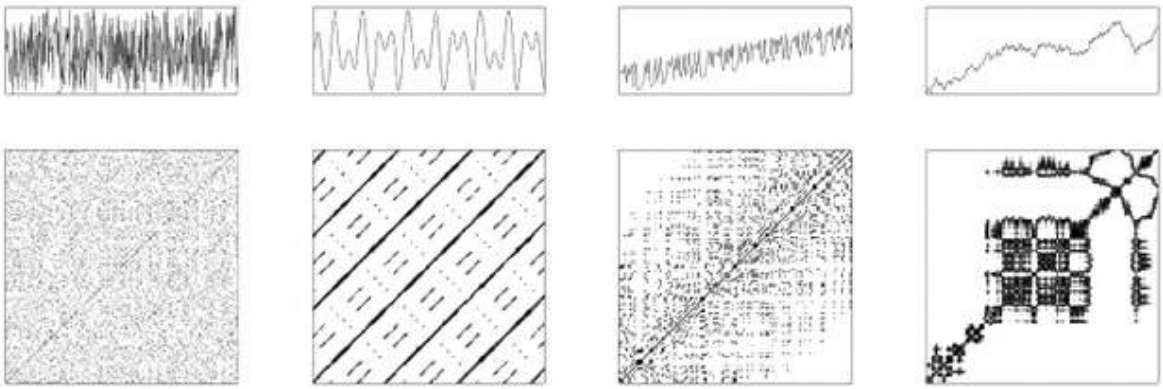


Figure 2.5: Recurrence Plots for Different Types of Systems. From left to right: random noise, periodic oscillations with two frequencies, deterministic chaotic system and autoregressive process.

2.6 Recurrence Quantification Analysis (RQA)

As we have discussed in the previous Section 2.5 RPs reflect some properties of the systems under analysis which are closely related to the recurrence dynamics which the system exhibits as evolving in time. However, these images have could be used for qualitative analysis of the true identity of the system but could be also utilized for extracting quantitative measurements in order to discriminate the underlying systems which will be analyzed.

2.6.1 RQA Measures

The measures we extract from the RPs are based on patterns from diagonal, vertical lines similar to the definitions of the previous Section 2.5. According to this we define some RQA measures which will be extracted from each $N \times N$ RP and will be used in this work.

Recurrence Rate (RR): which is a measure of the density of points in the RP. RR defines the

[1] Figure was found in: https://en.wikipedia.org/wiki/Recurrence_plot

probability that a similar state recurs to its neighbourhood in PS.

$$RR = \frac{1}{N^2} \sum_{i,j=1}^{N} \mathbf{R}_{i,j} \tag{2.40}$$

Determinism (DET): the ratio of diagonal lines of at least length l_{min} to the total number of points of the RP. Processes with less correlated time-series or or time-series with chaotic behavior would exhibit very short diagonals on their RPs. On the other hand, purely deterministic processes would lead to longer diagonals and less isolated points on their RPs.

$$DET = \frac{\sum_{l=d_m}^{N} lP_d(l)}{\sum_{l=1}^{N} lP_d(l)} \tag{2.41}$$

Max Diagonal Length ($\mathbf{L_{max}}$): the maximum length of a diagonal line found. L_{max} reflects the inverse of the exponential divergence of the trajectories of the PS. For longer diagonals. If the trajectories of the PS are diverging fast then we expect to see shorter diagonals structures and consequently lower values for L_{max}.

$$L_{max} = max(\{l_i\}_{i=1}^{N_d}) \tag{2.42}$$

Average Diagonal Length (L): the average length of all diagonal lines found on the RP. Intuitively, a diagonal line of length l means that trajectories are co-evolving during l samples but they correspond to different times of the system evolution. These lines indicate how different trajectories diverge during the evolution of the system and as time passes by. This average length is actually the mean time that we can predict the next recurrence of states from the state we observe now.

$$L = \frac{\sum_{l=d_m}^{N} lP_d(l)}{\sum_{l=d_m}^{N} P_d(l)} \tag{2.43}$$

Diagonal Entropy (DENTR): refers to the Shannon entropy of the probability of finding a diagonal line of length exactly equal to l. This entropy of diagonal lines reflects the complexity of the RP. Hence, we would expect that for unvoiced speech segments or white noise regions this value would be close to zero as well as the complexity of the system under analysis.

$$DENTR = \sum_{l=d_m}^{N} \frac{P_d(l)}{N_d} ln(\frac{N_d}{P_d(l)}) \tag{2.44}$$

Laminarity (LAM): the ratio of vertical lines of at least length u_{min} to the total number of points of the RP. These vertical structures represent the existence of laminar phases that the underlying system exhibits without indicating the time-periods which they occur.

$$LAM = \frac{\sum_{l=v_m}^{N} lP_v(l)}{\sum_{l=1}^{N} lP_v(l)} \tag{2.45}$$

Max Vertical Length ($\mathbf{V_{max}}$): the maximum length of all vertical lines found on the RP. Indicates the maximum time-period that the underlying system will stay in a laminar state.

$$V_{max} = max(\{l_i\}_{i=1}^{N_v}) \tag{2.46}$$

Trapping Time (TT): the average length of all vertical lines found on the RP. Reflects the mean time of the laminar phases of the system recurring under the specified frame of analysis.

$$TT = \frac{\sum_{l=v_m}^{N} lP_v(l)}{\sum_{l=v_m}^{N} P_v(l)} \tag{2.47}$$

Vertical Entropy (VENTR): refers to the Shannon entropy of the probability of finding a vertical line of length exactly equal to l. This entropy of vertical lines reflects the distribution of time-periods for which the system abides in laminar phases.

$$VENTR = \sum_{l=v_m}^{N} \frac{P_v(l)}{N_v} ln(\frac{N_v}{P_v(l)}) \tag{2.48}$$

Max White Vertical Length (W_{max}): the maximum length of all white lines found on the RP. White areas or band are indicative of abrupt change of dynamics. Thus, longer white vertical lines indicate the existence of rare events on the RP that might not be predicted using information from previous trajectories of the PS.

$$W_{max} = max(\{l_i\}_{i=1}^{N_w}) \tag{2.49}$$

Average White Vertical Length (AWVL): the average length of all white lines found on the RP. The average time-periods between abrupt changes of the system, from one state to the other.

$$AWVL = \frac{\sum_{l=w_m}^{N} lP_w(l)}{\sum_{l=w_m}^{N} P_w(l)} \tag{2.50}$$

White Vertical Entropy (WENTR): refers to the Shannon entropy of the probability of finding a white vertical line of length exactly equal to l. This entropy of white vertical reflects the distribution of time-periods that abrupt changes of the dynamics of the underlying system occur.

$$WENTR = \sum_{l=w_m}^{N} \frac{P_w(l)}{N_w} ln(\frac{N_w}{P_w(l)}) \tag{2.51}$$

2.7 Multidimensional Scaling (MDS)

2.7.1 Classical MDS

Classical MDS was first introduced in [99] and can be formalized as follows. Given the matrix consisting of pairwise distances or dissimilarities $\{\delta_{ij}\}_{1\leq i,j\leq N}$ between N points in a high dimensional space, the solution to Classical MDS is given by a set of points $\{\mathbf{x}_i\}_{i=1}^{N}$ which lie on the manifold $\mathcal{M} \in \mathbb{R}^L$ and their pairwise distances are able to preserve the given dissimilarities $\{\delta_{ij}\}_{1\leq i,j\leq N}$ as faithfully as possible. Each point $\mathbf{x}_i \in \mathbb{R}^L,\ 1 \leq i \leq N$ corresponds to a column of the matrix $\mathbf{X}^T \in \mathbb{R}^{L\times N}$. The embedding dimension L is selected as small as possible in order to obtain the maximum dimensionality reduction but also to be able to approximate the given dissimilarities δ_{ij} by the Euclidean distances $d_{ij}(\mathbf{X}) = ||\mathbf{x}_i - \mathbf{x}_j||_2 = \sqrt{\sum_{k=1}^{L}(x_{ik} - x_{jk})^2}$ in the embedded space $\mathbb{R}^L$.

The proposed algorithm uses a centering matrix $\mathbf{H} = \mathbf{I}_N - \frac{1}{N}\mathbf{1}_N^T\mathbf{1}_N$ in order to subtract the mean of the columns and the rows for each element. Where $\mathbf{1}_N = [1, 1, ..., 1]$ a vector of ones in $\mathbb{R}^N$ space. By applying the double centering to the Hadamard product of the given dissimilarities, the Gram matrix $\mathbf{B}$ is constructed as follows:

$$\mathbf{B} = -\frac{1}{2}\mathbf{H}^T(\mathbf{\Delta} \odot \mathbf{\Delta})\mathbf{H} \tag{2.52}$$

It can be shown (Ch. 12 [100]) that classical MDS minimizes the Strain algebraic criterion in Equation 2.53 below:

$$||\mathbf{X}\mathbf{X}^T - \mathbf{B}||_F^2 \tag{2.53}$$

The eigendecomposition of the symmetric matrix $\mathbf{B}$ gives us $\mathbf{B} = \mathbf{V}\mathbf{V}^T$ and thus the new set of points consisting the embedding in $\mathbb{R}^L$ are given by the first L positive eigenvalues of , namely $\mathbf{X} = \mathbf{V}_L$. This solution provides the same result as Principal Component Analysis (PCA) applied on the vector

in the high dimensional space [101]. Classic MDS was originally proposed for dissimilarity matrices which can be embedded with good approximation accuracy in a low-dimensional Euclidean space. However, matrices which correspond to embeddings in Euclidean sub-spaces [102], Poincare disks [103] and constant-curvature Riemannian spaces [104] have also been studied.

2.7.2 Metric MDS

Metric MDS describes a superset of optimization problems containing classical MDS. Shepard has introduced heuristic methods to enable transformations of the given dissimilarities δ_{ij} [105], [106] but did not provide any loss function in order to model them [107]. Kruskal in [108] and [109] formalized the metric MDS as a least squares optimization problem of minimizing the non-convex Stress-1 function defined in Equation 2.54 shown next:

$$\sigma_1(\mathbf{X}, \hat{\mathbf{D}}) = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(\hat{d_{ij}} - d_{ij}(\mathbf{X}))}{\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2(\mathbf{X})}} \tag{2.54}$$

where matrix $\hat{\mathbf{D}}$ with elements $\hat{d_{ij}}$ represents all the pairs of the transformed dissimilarities δ_{ij} that are used to fit the embedded distance pairs $d_{ij}(\mathbf{X})$.

In essence, $\hat{d_{ij}} = \mathcal{F}(\delta_{ij})$ where $\mathcal{F}$ is usually an affine transformation $\hat{d_{ij}} = \alpha + \beta\delta_{ij}$ for unknown α and β. Additionally, monotone and polynomial regression transformations are employed for nonmetric-MDS, as well as, a wider family of transformations [110]. Kruskal proposed an iterative gradient-based algorithm for the minimization of σ_1 since the solution cannot be expressed in closed form. Assuming that $d_{ij} \hat{=} \delta_{ij}$ the algorithm iteratively tries to find the coordinates of points $\mathbf{X}$ which are lying in the low embedding space $\mathbb{R}^L$. Trivial solutions ($\mathbf{X} = \mathbf{0}$ and $\hat{\mathbf{D}} = \mathbf{0}$) are avoided by the denominator term in Equation 2.54.

A weighted MDS raw Stress function is defined as:

$$\sigma_{raw}^2(\mathbf{X}, \hat{\mathbf{D}}) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(\hat{d_{ij}} - d_{ij}(\mathbf{X}))^2 \tag{2.55}$$

where the weights w_{ij} are restricted to be non-negative; for missing data the weights are set equal to zero. By setting $w_{ij} = 1, \forall 1 \leq i, j \leq N$ one can model an equal contribution to the Metric-MDS solution for all the elements.

2.7.3 SMACOF

SMACOF which stands for Scaling by Majorizing a Complex Function is a state-of-the-art algorithm for solving metric MDS and was introduced in [111]. By setting $\hat{d_{ij}} = \delta_{ij}$ in raw stress function defined in Equation 2.55, SMACOF minimizes the resulting stress function $\sigma_{raw}^2(\mathbf{X})$.

$$\sigma^2(\mathbf{X}) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(\delta_{ij}^2 - 2\delta_{ij}d_{ij}(\mathbf{X}) + d_{ij}^2(\mathbf{X})) \tag{2.56}$$

The algorithm proceeds iteratively and decreases stress monotonically up to a fixed point by optimizing a convex function which serves as an upper bound for the non-convex stress function in Equation 2.56. An extensive description of SMACOF can be found in [100] while its convergence for a Euclidean embedded space $\mathbb{R}^L$ has been proven in [112].

Let matrices $\mathbf{U}$ and $\mathbf{R}(\mathbf{X})$ be defined element-wise as follows:

$$u_{ij} = \begin{cases} -w_{ij} & i \neq j \\ \sum_{k \neq i} w_{ik} & i = j \end{cases} \tag{2.57}$$

$$r_{ij} = \begin{cases} -w_{ij}\delta_{ij}d_{ij}^{-1}(\mathbf{X}) & i \neq j, d_{ij}(\mathbf{X}) \neq 0 \\ 0 & i \neq j, d_{ij}(\mathbf{X}) = 0 \\ \sum_{k \neq i} r_{ik} & i = j \end{cases} \tag{2.58}$$

The stress function in Equation 2.56 is converted to the following quadratic form:

$$\sigma^2(\mathbf{X}) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\delta_{ij}^2 - 2tr(\mathbf{X}^T\mathbf{R}(\mathbf{X})\mathbf{X}) + tr(\mathbf{X}^T\mathbf{U}\mathbf{X}) \tag{2.59}$$

The quadratic can be minimized iteratively as follows:

$$T(\mathbf{X}, \hat{\mathbf{X}}^{(k)}) = \sum_{j=1}^{N} w_{ij}\delta_{ij}^2 - 2tr(\mathbf{X}^T\mathbf{R}(\hat{\mathbf{X}}^{(k)})\hat{\mathbf{X}}^{(k)}) + tr(\mathbf{X}^T\mathbf{U}\mathbf{X}) \tag{2.60}$$

$$\hat{\mathbf{X}}^{(k+1)} = \underset{\mathbf{X}}{\text{argmin}}\ T(\mathbf{X}, \hat{\mathbf{X}}^{(k)}) = \mathbf{U}^{\dagger}\mathbf{R}(\hat{\mathbf{X}}^{(k)})\hat{\mathbf{X}}^{(k)} \tag{2.61}$$

where $\hat{\mathbf{X}}^{(k)}$ is the estimate of matrix $\mathbf{X}$ at the kth iteration and $\mathbf{U}^{\dagger}$ is Moore-Penrose pseudoinverse of $\mathbf{U}$. Noticeably, the term $\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\delta_{ij}^2$ corresponds to a constant which is defined only by the initial weight matrix $\mathbf{W}$ and the given dissimilarities δ_{ij}. At iteration k the convex majorizing convex function touches the surface of σ at the point $\hat{\mathbf{X}}^{(k)}$. By minimizing this simple quadratic function in Equation 2.60 we find the next update which serves as a starting point for the next iteration $k+1$. The solution to the minimization problem is shown in Equation 2.61. The algorithm stops when the new update yields a decrease $\sigma^2(\hat{\mathbf{X}}^{(k+1)}) - \sigma^2(\hat{\mathbf{X}}^{(k)})$ that is smaller than a threshold value.

2.8 General Pattern Search (GPS) methods

2.8.1 GPS formulation

The unconstrained problem of minimizing a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is formally described as

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathbb{R}^n}{\text{argmin}}\ f(\mathbf{x}) \tag{2.62}$$

Next we present a short description of iterative GPS minimization of Equation 2.62 based on [113, 114]. First we have to define the following components:

- A basis matrix that could be any nonsingular matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$.
- A matrix $\mathbf{C}^{(k)}$ for generating all the possible moves for the kth iteration of the minimization algorithm
$$\mathbf{C}^{(k)} = [\mathbf{M}^{(k)} \ \ -\mathbf{M}^{(k)} \ \ \mathbf{L}^{(k)}] = [\mathbf{\Psi}^{(k)} \ \ \mathbf{L}^{(k)}] \tag{2.63}$$
where the columns of $\mathbf{M}^{(k)} \in \mathbb{Z}^{n\times n}$ form a positive span of $\mathbb{R}^n$ and $\mathbf{L}^{(k)}$ contains at least the zero column of the search space $\mathbb{R}^n$.
- A pattern matrix $\mathbf{P}^{(k)}$ defined as
$$\mathbf{P}^{(k)} = \mathbf{B}\mathbf{C}^{(k)} = [\mathbf{B}\mathbf{M}^{(k)} \ \ -\mathbf{B}\mathbf{M}^{(k)} \ \ \mathbf{B}\mathbf{L}^{(k)}] \tag{2.64}$$
where the submatrix $\mathbf{B}\mathbf{M}^{(k)}$ forms a basis of $\mathbb{R}^n$.

In each iteration k, we define a set of steps $\{\mathbf{s}_i^{(k)}\}_{i=1}^m$ generated by the pattern matrix $\mathbf{P}^{(k)}$ as shown next:

$$\mathbf{s}_i^{(k)} = \Delta^{(k)}\mathbf{p}_i^{(k)}, \quad \mathbf{P}^{(k)} = [\mathbf{p}_1^{(k)}, ..., \mathbf{p}_m^{(k)}] \in \mathbb{R}^{n\times m} \tag{2.65}$$

where $\mathbf{p}_i^{(k)}$ is the ith column of $\mathbf{P}^{(k)}$ and defines the direction of the new step, while $\Delta^{(k)}$ configures the length towards this direction. If the pattern matrix $\mathbf{P}^{(k)}$ contains m columns, then $m \geq n+1$ in order to positively span the search space $\mathbb{R}^n$. Thus, a new trial point of GPS algorithm towards this step would be $\mathbf{x}_i^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}_i^{(k)}$ where we evaluate the value of the function f to minimize. The success of a new trial point is decided based on the condition that it takes a step towards further minimizing the function f, i.e., $f(\mathbf{x}^{(k)} + \mathbf{s}_i^{(k)}) > f(\mathbf{x}^{(k+1)})$. The steps of a GPS method are presented in Algorithm 1.

Algorithm 1 General Pattern Search (GPS)

```
1:  procedure GPS_SOLVER(x^(0), Δ^(0), C^(0), B)
2:      k = −1
3:      do
4:          k = k + 1
5:          s^(k) = EXPLORE_MOVES(BC^(k), x^(k), Δ^(k))
6:          ρ^(k) = f(x^(k) + s^(k)) − f(x^(k))
7:          if ρ^(k) < 0 then
8:              x^(k+1) = x^(k) + s^(k)                ▷ Successful iteration
9:          else
10:             x^(k+1) = x^(k)                        ▷ Unsuccessful iteration
11:         Δ^(k+1), C^(k+1) = UPDATE(C^(k), Δ^(k), ρ^(k))
12:     while convergence criterion == False
```

To initialize the algorithm we select a point $\mathbf{x}^{(0)} \in \mathbb{R}^n$ and a positive step length parameter $\Delta^{(0)} > 0$. In each iteration k, we explore a set of moves defined by the `EXPLORE_MOVES()` subroutine at line 5 of the algorithm. Pattern search methods described using a GPS formalism mainly differ on the heuristics used for the selection of exploratory moves. If a new exploratory point lowers the value of the function f, iteration k is successful and the starting point of the next iteration is updated $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$ as shown in line 8, else there is no update. The step length parameter $\Delta^{(k)}$ is modified by the `UPDATE()` subroutine in line 11. For successful iterations, i.e., $\rho^{(k)} < 0$, the step length is forced to increase in a determistic way as follows:

$$\begin{aligned} &\Delta^{(k+1)} = \lambda^{(k)}\Delta^{(k)}, \quad \lambda^{(k)} \in \Lambda = \{\tau^{w_1}, ..., \tau^{w_{|\Lambda|}}\} \\ &\tau > 1, \quad \{w_1, ..., w_{|\Lambda|}\} \subset \mathbb{N}, \quad |\Lambda| < +\infty \end{aligned} \tag{2.66}$$

where τ and w_i are predefined constants that are used for the ith successive successful iteration. For unsuccessful iterations the step length parameter is decreased, i.e., $\Delta^{(k+1)} \leq \Delta^{(k)}$ as follows:

$$\Delta^{(k+1)} = \theta\Delta^{(k)}, \quad \theta = \tau^{w_0}, \quad \tau > 1, \quad w_0 < 0, \tag{2.67}$$

where τ and the negative integer w_0 determine the fixed ratio of step reduction. Note that the generating matrix $\mathbf{C}^{(k+1)}$ could be also updated for unsuccessful/successful iterations in order to contain more/less search directions, respectively.

2.8.2 GPS Convergence

GPS methods under the aforementioned defined framework have some important convergence properties shown in [113, 114, 115, 116, 117] and summarized here. For any GPS method which satisfies the specifications of Hypothesis 1 on the exploratory moves one may be able to show convergence for Algorithm 1.

Hypothesis 1 (Weak Hypothesis on Exploratory Moves): *The subroutine* `EXPLORE_MOVES()` *as defined in Algorithm 1, line 5 guarantees the following:*

- *The exploratory step direction for iteration k is selected from the columns of the pattern matrix $\mathbf{P}^{(k)}$ as defined in Equation 2.65 and the exploratory step length is $\Delta^{(k)}$ as defined in Equations 2.66, 2.67.*
- *If among the exploratory moves $\mathbf{a}^{(k)}$ at iteration k selected from the columns of the matrix $\Delta^{(k)}\mathbf{B}[\mathbf{M}^{(k)} - \mathbf{M}^{(k)}]$ exist at least one move that leads to success, i.e., $f(\mathbf{x}^{(k)} + \mathbf{a}) < f(\mathbf{x}^{(k)})$, then the* `EXPLORE_MOVES()` *subroutine will return a move $\mathbf{s}^{(k)}$ such that $f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)}) < f(\mathbf{x}^{(k)})$.*

Hypothesis 1 enforces some mild constraints on the configuration of the exploratory moves produced by Algorithm 1, line 5. Essentially, the suggested step $\mathbf{s}^{(k)}$ is derived from the pattern matrix $\mathbf{P}^{(k)}$, while the algorithm needs to provide a simple decrease for the objective function f. Specifically, the only way to accept an unsuccessful iteration would be if none of the steps from the columns of the matrix $\Delta^{(k)}\mathbf{B}[\mathbf{M}^{(k)} - \mathbf{M}^{(k)}]$ lead to a decrease of the objective function f. Based on this hypothesis one can formulate Theorem. 1 as follows:

Theorem 1: *Let $L(\mathbf{x}^{(0)}) = \{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}^{(0)})\}$ be closed and bounded and f continuously differentiable on a neighborhood of $L(\mathbf{x}^{(0)})$, namely on the union of the open balls $\bigcup_{\mathbf{a} \in L(\mathbf{x}^{(0)})} B(\mathbf{a}, \eta)$ where $\eta > 0$. If a GPS method is formulated as described in Section 2.8.1 and Hypothesis 1 holds then for the sequence of iterations $\{\mathbf{x}^{(k)}\}$ produced by Algorithm 1*

$$\lim_{k \to +\infty} \inf \, ||\nabla f(\mathbf{x}^{(k)})|| = 0$$

For the proof of this Theorem we refer the reader to [113].

As shown in [118] one can construct a continuously differentiable objective function and a GPS method with infinite many limit points with non-zero gradients and thus even Theorem. 1 holds, the convergence of $||\nabla f(\mathbf{x}^{(k)})||$ is not assured. However, the convergence properties of GPS methods can be further strengthened if additional criteria are met. Specifically, a stronger hypothesis on exploratory moves Hypothesis 2 regulates the measure of decrease of the objective function for each step produced by the GPS method, as follows:

Hypothesis 2 (Strong Hypothesis on Exploratory Moves): *The subroutine* `EXPLORE_MOVES()` *as defined in Algorithm 1, line 5 guarantees the following:*

- *The exploratory step direction for iteration k is selected from the columns of the pattern matrix $\mathbf{P}^{(k)}$ as defined in Equation 2.65 and the exploratory step length is $\Delta^{(k)}$ as defined in Equations 2.66, 2.67.*
- *If among the exploratory moves $\mathbf{a}^{(k)}$ at iteration k selected from the columns of the matrix $\Delta^{(k)}\mathbf{B}[\mathbf{M}^{(k)} - \mathbf{M}^{(k)}]$ exist at least one move that leads to success, i.e., $f(\mathbf{x}^{(k)} + \mathbf{a}) < f(\mathbf{x}^{(k)})$, then the* `EXPLORE_MOVES()` *subroutine will return a move $\mathbf{s}^{(k)}$ such that:*
  $f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)}) \leq \min_{\mathbf{a}^{(k)}} f(\mathbf{x}^{(k)} + \mathbf{a}^{(k)})$.

Hypothesis 2 enforces the additional strong constraint on the configuration of the exploratory moves, namely that the subroutine `EXPLORE_MOVES()` will do no worse than produce the best exploratory move from the columns of the matrix $\Delta^{(k)}\mathbf{B}[\mathbf{M}^{(k)} - \mathbf{M}^{(k)}]$. Based on this hypothesis and by adding requirements restricting the exploration step direction and length for the GPS method, one can formulate Theorem. 2 which is also presented here without proof.

Theorem 2: *Let $L(\mathbf{x}^{(0)}) = \{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}^{(0)})\}$ be closed and bounded and f continuously differentiable on a neighborhood of $L(\mathbf{x}^{(0)})$, namely on the union of the open balls $\bigcup_{\mathbf{a} \in L(\mathbf{x}^{(0)})} B(\mathbf{a}, \eta)$ where*

$\eta > 0$. *If a GPS method is formulated as described in Section 2.8.1,* $\lim_{k \to +\infty} \Delta^{(k)} = 0$, *the columns of the generating matrices* $\mathbf{C}^{(k)}$ *are bounded by norm and Hypothesis 2 holds then for the sequence of iterations* $\{\mathbf{x}^{(k)}\}$ *produced by Algorithm 1*

$$\lim_{k \to +\infty} ||\nabla f(\mathbf{x}^{(k)})|| = 0$$

The additional requirements specify that: 1) the generating matrix $\mathbf{C}^{(k)}$ should be norm bounded in order to produce trial steps from Equation 2.65 that are bounded by the step length parameter $\Delta^{(k)}$ and 2) $\lim_{k \to +\infty} \Delta^{(k)} = 0$ that can be easily met by selecting $\Lambda = \{1\}$ in Equation 2.67; this also guarantees a non increasing sequence of $\Delta^{(k)}$ steps [113]. Although these criteria provide much stronger convergence properties, we are faced with a trade off between the theoretical proof of convergence and the efficiency of heuristics in finding a local optimum.

Both theorems 1 and 2 provide a first order optimality condition if their specifications hold. Although the latter theorem premises much stronger convergence results, step-length control parameter $\Delta^{(k)}$, provides a reliable asymptotic measure of first-order stationarity when it is reduced after unsuccessful iterations [114].

Chapter 3

Timescales of Decision for Speech Emotion Recognition

This chapter is an extended version of the published paper [2] in International Conference on Affective Computing and Intelligent Interaction (ACII) which has taken place in San Antonio, Texas, USA, October 23-26 2017. If the reader needs to cite parts of this chapter then it would be preferred to use the following reference:

- *Efthymios Tzinis and Alexandros Potamianos. "Segment-based speech emotion recognition using recurrent neural networks." In Affective Computing and Intelligent Interaction (ACII), 2017 Seventh International Conference on, pp. 190-195. IEEE, 2017.*

3.1 Motivation

As we have previously discussed in Section 1.4 a speech signal encloses information in different timescales. focusing on SER task we can easily infer that different sets of acoustic features could be activated in different timescales. Hence, the performance of the SER system in use is sensitive to this configuration. Although the timescale at which frame-based features are concatenated or statistical segment-based features are extracted has substantial implications for SER performance, little progress has been made towards this direction [53], [54]. Considering the expressiveness of RNNs when they are modeling series of feature vectors and their close relation with the timescale which is correspondent to the input feature vectors, it is essential to analyze the effectiveness of different timescales on RNN-based SER systems. Specifically, the appropriate decision timescale from which RNNs can explore a more abstract sequential representation either from local or global features.

3.2 Acoustic Features

In this section we will present some methods we used in order to extract the acoustic local and global features sets for our approach, as described in Sections 3.2.3 and 3.2.4. Local features correspond to raw values of LLDs and model each utterance using these feature vectors directly. On the other hand, in order to extract global features we need to first extract LLDs and after that apply statistical functionals over speech-segments. Now In this context, each emotional utterance would be modeled using a sequence of one or more statistical representations of the included segments. In Section 3.2.1 we are highlighting some preprocessing strategies we follow and experiment on, prior to the feature extraction process based on LLDs from frames. Additionally, in Section 3.2.2 we analyze the key mechanism of Voice Activity Detection (VAD) for OpenSMILE and other frameworks. In order to extract the aforementioned features we used OpenSMILE framework [119] which is also used in many other works in literature as well.

3.2.1 Pre-Processing

Conventional acoustic feature extraction techniques are based on breaking the signal which is to be analyzed into overlapping frames of (10ms-100ms) in order to extract the required LLDs [120]. For

each speech frame we are applying a Hamming window of the same length as shown next.

$$s_w(i) = \begin{cases} s(i) \cdot w(i) & c - \frac{W}{2} \leq i \leq c + \frac{W}{2} \\ 0 & otherwise \end{cases} \tag{3.1}$$

where s_w corresponds to the windowed signal, c is the center index around which the window function will be applied. The corresponding window has a length of W samples. It is easy to infer that the corresponding frame would also have a length equal to W as all the other areas would be zeroed. Although a variety of windows have been proposed for digital signal processing, in this work we will use a hamming window which is defined as follows:

$$w_H(i) = \begin{cases} 0.54 + 0.46 \cdot cos(\frac{\pi \cdot i}{W}) & c - \frac{W}{2} \leq i \leq c + \frac{W}{2} \\ 0 & otherwise \end{cases} \tag{3.2}$$

As well as a Gaussian window which could be approximated as shown next:

$$w_G(i) = \begin{cases} G(i) - \frac{G(-0.5)[G(i+\frac{W}{2}) + G(i-\frac{W}{2})]}{G(-0.5+\frac{W}{2}) + G(-0.5-\frac{W}{2})} & c - \frac{W}{2} \leq i \leq c + \frac{W}{2} \\ 0 & otherwise \end{cases} \tag{3.3}$$

where $G(\cdot)$ is defined as the Gaussian in the next Equation:

$$G(i) = e^{(\frac{i - \frac{W}{4} + 0.5}{0.2})^2} \tag{3.4}$$

Again, W defines the length of the specified window.

After the aforementioned procedure we can accumulate the windowed speech frames and process them individually. In this context, we assume that these frames contain frequency information below 4kHz, which is an acceptable assumption for speech signals [6]. We use a pre-emphasis filter in order to boost the information which is conveyed in higher frequencies and suppress the respective in lower ones which could be areas of silence or noise. The pre-emphasis filtering is defined as shown next:

$$s_{pe}(i) = s(i) - \alpha \cdot s(i-1) \tag{3.5}$$

where α is considered constant and equal to 0.975 [23].

After this process we are able to perform various feature extraction methods which are based on spectral, cepstral or voice quality features domain in order to get a static feature representation of LLDs for each frame. The process we described above is similar for all the LLDs extracted in all the subsequent sections of this thesis but the configurations of the frame length or the overlapping between the frames could be changed. Thus, we will refer to this configuration by changing the required parameters.

3.2.2 Voice Activity Detection

Silence frames also exist inside an a speech signal which might produce misleading values in their feature vector representations and thus leading the SER system to falsely classify the whole utterance. In order to nullify the effect of silence frames in the sequence of feature vectors we need a Voice Activity Detection (VAD) mechanism. There are many implementations of VAD like WebRTC Voice Activity Detector [1] or the one which also included inside OpenSMILE toolkit [119].

We can see the effectiveness of Google's WebRTC when we assess it on utterances from the BERLIN emotional speech database (EmoDB) [121] in Figure 3.1. In the aforementioned figure, a variety of VAD masks produced under different selections of the parameters "Fd" and "Agg" are displayed. These parameters correspond to the time in ms the frames will be processes and the aggressiveness on which a frame would be considered a speech frame or a silent one. We can see that

[1] Source code has been developed by Google and is freely available for download and non-commercial usage here: https://github.com/wiseman/py-webrtcvad

different parametrizations produce different VAD masks for each of the 4 utterances which are displayed. A longer window takes into account a wider and consequently a more smooth region of the speech signal in order to infer if speech is included and thus it would be more likely to produce a positive result. On the contrary, shorter frames might not contain speech at all and thus some regions of non-speech might be reflected in the VAD mask although a consonant is emitted. An aggressive technique would provide much more regions of voice because fricative and low-energy consonants might be confusing for the VAD system. On the other hand, a less aggressive technique might produce regions with non-speech as vocal ones. The latter result could be alleviated by introducing smoothing filtering before the final result of the VAD mask.

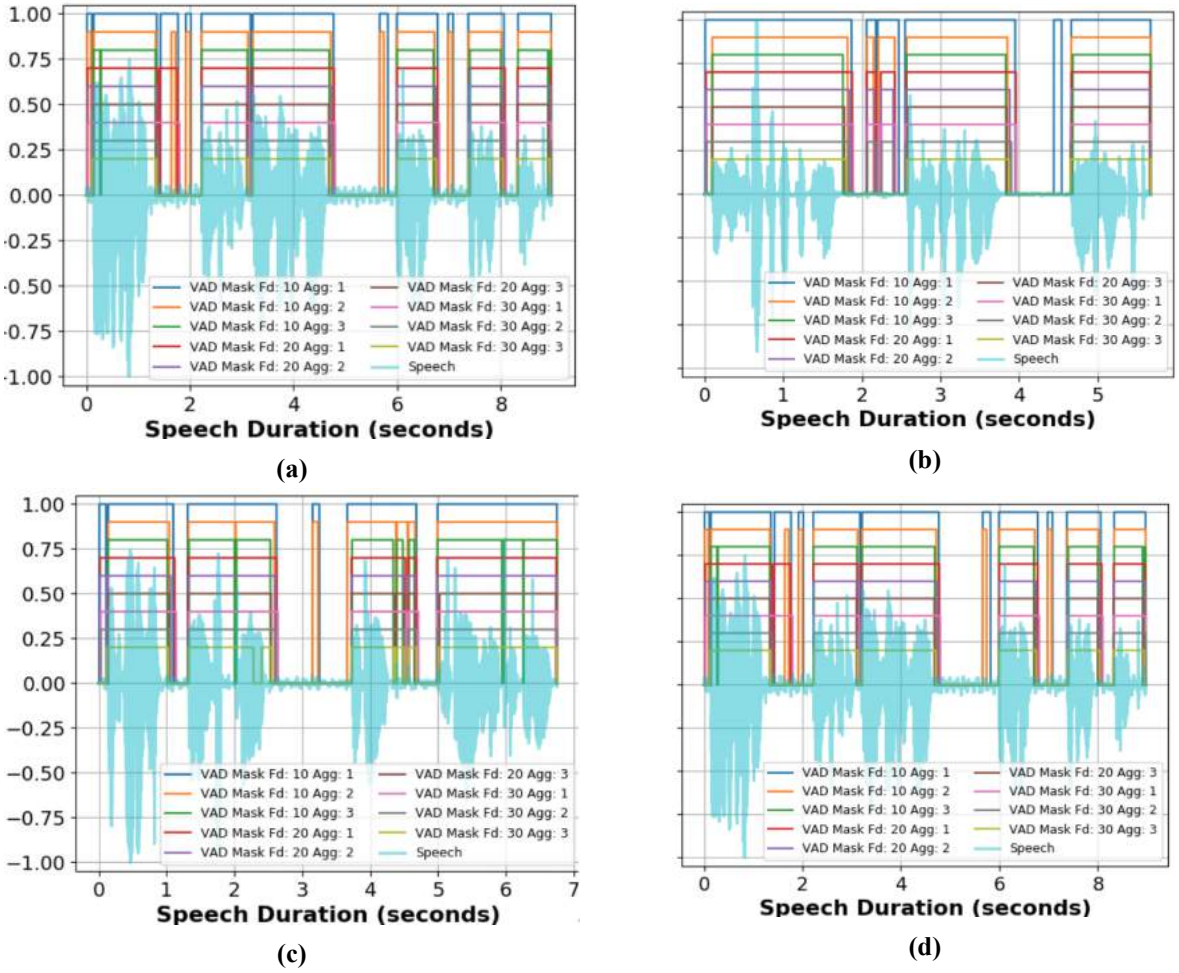


Figure 3.1: WebRTC VAD evaluation on EmoDB utterances.

As we have mentioned earlier, in our work we utilize OpenSMILE in order to extract our acoustic feature sets. To this end, we use two of the built-in VAD mechanisms. The first one is based on an LSTM unit for the inference using pitch and energy LLDs on frame level and it has been trained on a movie dataset. On the other hand, a more naive approach is to extract the probability of voice for each frame and then threshold it in order to acquire the regions of speech inside the signal. Both of these VAD architectures are depicted in Figure 3.2. It is evident that the LSTM-based method suffers from the missing data of this dataset which is evaluated. Namely, the LSTM VAD has been trained on different samples with presumably different noise and recording conditions rather than the ones existing in EmoDB. This is why we select the more naive VAD mechanism which is based on voicing probability and a threshold in order to infer the final mask for voice/silent regions of the each emotional utterance. Moreover, we notice that frame level pitch descriptor is higher in the same areas as probability of voicing. This indicates that even ad-hoc thresholds applied on the latter value

can aptly describe regions of speech and discriminate between areas of silence. The selected voice probability cut-off threshold is specifically set so high (0.85) in Figure 3.2, in order to provide the time-series of the latter feature under a suboptimal value. It is clear that with a qualitative analysis of a threshold we can aptly adjust the value of the threshold for our purposes.

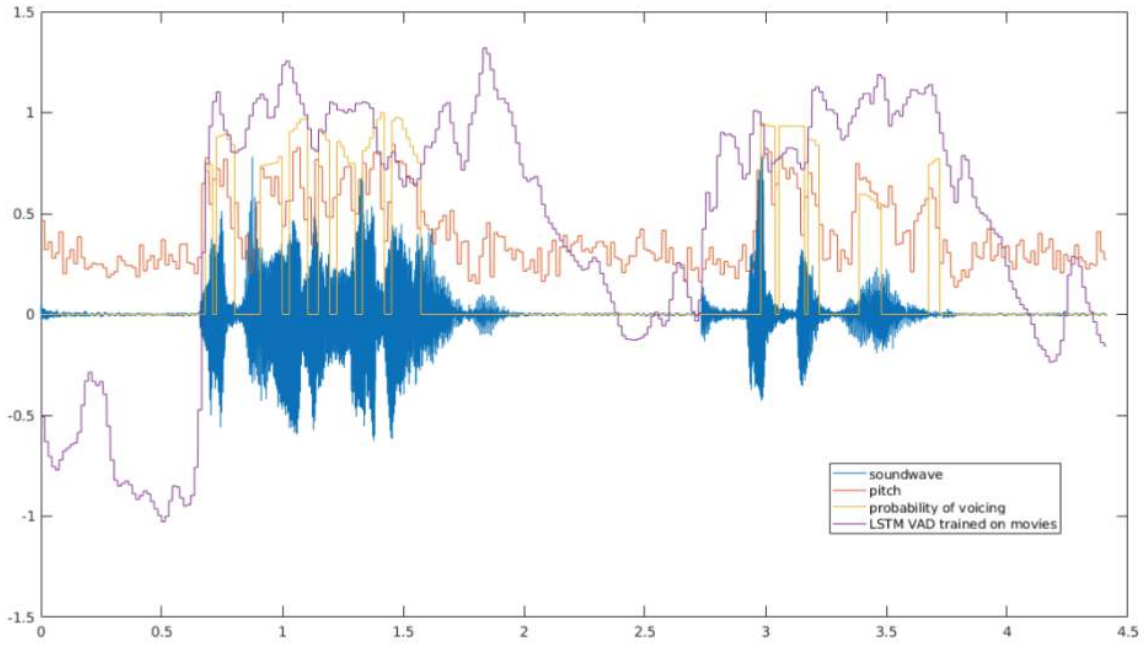
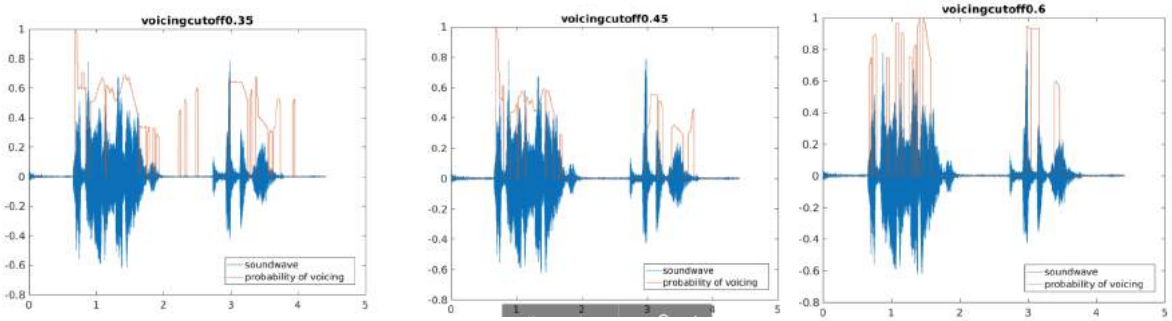


Figure 3.2: OpenSMILE VAD mechanisms based on probability of voicing and pre-trained LSTM

We also present a brief analysis over some values of thresholds for the frame-level descriptor of probability of voicing under the same utterance. In Figure 3.3 we can see the speech signal plotted alongside its corresponding voicing probability feature series for all its frames of (30ms). As we raise the threshold we can see that regions of speech are neglected and the produced VAD mask, based on the value of probability of voicing, does not include them. In this context we believe that a threshold value close to 0.35 would adequately satisfy our purpose. It should be noted, that this is another heuristic method of finding a threshold where it could suffer significantly for other speech databases. However, the purpose of this study is not related to search every parameter in the grid space emerging from the feature extraction procedure and thus we will not analyze this subject further.



(a) voicing cut-off 0.35 **(b)** voicing cut-off 0.45 **(c)** voicing cut-off 0.6

Figure 3.3: OpenSMILE VAD based on different cut-offs from voice probability

Table 3.1: Local Features

(RMS) Root Mean Square, (ZCR) Zero Crossing Rate, (HNR) Harmonics to Noise Ratio, (DDP) Difference of Difference of Periods, (LSP) Line Spectral Pairs, (SHS) Sub-Harmonic Sum, (ACF) Autocorrelation Function, (MFCCs) Mel Frequency Cepstral Coefficients.

Selected LLDs	Including 1st Delta
RMS Energy	✓
Quality of Voice	✓
ZCR	✓
Jitter Local	✗
Jitter DDP	✗
Shimmer Local	✗
F0 by SHS	✓
Loudness	✓
Probability of Voicing	✓
HNR by ACF	✓
MFCCs[0-14]	✓
Total: 25	Total: 22

3.2.3 Local Features

In this chapter we refer to Local features as the raw LLDs which are extracted on a frame-level in order to model each emotional utterance. In this section we will provide a brief analysis on how different LLDs are extracted and which of them will be used for our experiments. It is significant to note that multiple LLDs are extracted in this work. In this section we are only considered about LLDs which are used directly as an input to our SER models.

We break each emotional utterance in 30ms frames with 15ms step size and window them (see Equation 3.1) with a Hamming window as defined in Equation 3.2. After windowing we also apply a pre-emphasis filtering as defined in Equation 3.5.

We extract 25 LLDs for every frame and we also extract the derivatives for all of them (except jitter and shimmer). The latter features already describe perturbation measures of frequency and amplitude instability, respectively. Thus they are closely related with the notion of the difference which is the discrete equivalent of derivative (practically a delta value). Taking a delta of a delta value might produce much noisy and extravagant values which can be catastrophic for the training process. However, for jitter we extract a similar measure which is computed as a difference of difference for the value of jitter across different periods.

The selected LLDs are drawn from cepstral, spectral, energy and voice quality which are the most prominent clusters of acoustic features. Specifically, we extract: Root Mean Square (RMS) energy of the windowed frame, quality of voicing, Zero Crossing Rate (ZCR), Harmonics to Noise Ratio (HNR) by using Autocorrelation Function (ACF) of the speech frame, jitter and jitter Difference of Difference of Periods (DDP). Moreover, we extract: local shimmer, Fundamental Frequency (F0) using Sub-Harmonic Sum (SHS), loudness, probability of voicing and Mel Frequency Band (MFB) and MFCCs. The aforementioned selected features are also presented in Table 3.1 in a more compact format. The second column indicates if we also append the delta value of the corresponding feature between this value and the corresponding contained in the previous frame. A check-mark symbol ✓ indicates that the delta value is also appended in the final feature vector while an x-mark ✗ reflects the opposite.

Now every frame feature vector is endowed with 47 local-attributes. Because of the large number of local features, the most prominent LLDs are selected [61], [65], [27]. we refer the reader to extenssive descriptions of how these aforementioned LLDs are extracted [19], [20], [120]. Conse-

quently, each emotional utterance will be represented as a sequence of LLD-vectors extracted from all frames included in it. Of course the length of this sequence of feature vectors would not be the same across different utterances.

3.2.4 Global Features

In this chapter we refer to Global Features as the values of statistical functionals over speech segments of any duration which are applied over LLDs extracted from frames. Obviously, the frame-level feature extraction process would be similar to the one described in the previous section (see Section 3.2.3). After the feature extraction on a frame level we are applying sets of statistical functionals in order to get the final sequence of statistical feature vector representations. The length and the stride of the segments would define the number of feature vectors in each sequence. In the edge case where the segment durations would be equal to the total duration of each utterance, then the sequence would be deprecated into one feature vector. Generally speaking, a speech segment could be consider any time interval inside an utterance which is shorter than the utterance itself and longer than a duration corresponding to a typical frame of 10ms-100ms.

We follow a similar procedure with Schuller et al. [23] who extracted a robust set of emotional features. The attributes for the corresponding speech segments are based on LLDs which are extracted from frames, enriched with other LLDs as well. This extension of contained LLDs is applicable to this case because of the temporal aggregation of these values using statistical functionals. We could not include all these features in the previous set of Local features because of the increased dimensionality of the input sequence which would be caused.

Different LLDs require different window sizes for proper extraction. In short, we extracted pitch by using a Gaussian window of 60ms and 10ms step size, while for all the other LLDs we use a Hamming window of 25ms and 10ms step size. This process is followed because pitch requires a much wider window length for a good estimation of the prominent frequency of the speech signal. We could not follow a similar approach in the previous section (Section 3.2.3) mainly because the modeled sequence obtained by local features had to be extracted under a predefined window length. Although in this approach we can easily extract LLDs using multi-scale approaches in order to adapt each window size to a specific LLD.

To this end, we apply statistical functions on frame-level LLDs. In order to obtain the global features per speech-segment we use a variety of statistical functionals which are applied over each LLD value across the duration of the selected segment. The statistical functionals which are used correspond to ranges, position of extreme values, percentiles and other statistical values which are able to capture how each attribute or frame-level LLD is changing its value inside the selected segment. An overview of the selected statistical functionals to be applied are presented in Table 3.2 alongside with their corresponding grouping. We will use the symbol of each grouping in order to specify later the set of the statistical functionals which are applied to each frame-level LLD.

Now that we have the sets of the statistical functionals which will be applied towards the extraction of the statistical representations of the speech segments we are able to apply them over frame-level LLDs as well as their corresponding delta-values. Namely, the delta values are first computed between consecutive frames inside the speech segment which is to be analyzed and after that one statistical value corresponds to a value from the sequence of frame-level LLDs in this segment.

We use similar frame-level LLDs as in the previous section 3.2.3 and after that we apply the statistical functionals. However, in order to follow the feature extraction method described in [23], we use the same LLDs and we excluded RMS short time energy, quality of voicing and ZCR. We also appended other LLDs such as the first 8 Line Spectral Pairs (LSP), the first 8 coefficients from the Mel Filter-banks (MFBs) and the envelope of the fundamental frequency F0. For some of the aforementioned LLDs we also extract their deltas from their consecutive frames which is also evident in Table 3.3. The procedure of feature extraction is similar to the one described previously while only the last part alters here where the statistical functionals are applied for the last step of the feature extraction procedure.

Table 3.2: Sets of Statistical Functionals for Global Features

Statistical Functions	Set
position of maximum position of minimum arithmetic mean standard deviation skewness kurtosis linear regression coefficient 1/2 Quadratic regression error Absolute linear regression error quartile 1/2/3 quartile ranges 2-1/3-2/3-1 percentile 99 up-level time 75/90	A
percentile 1 percentile range 1-99	B
OnSets Number Duration	C

Table 3.3: Global Features

(HNR) Harmonics to Noise Ratio, (DDP) Difference of Difference of Periods, (LSP) Line Spectral Pairs, (SHS) Sub-Harmonic Sum, (ACF) Autocorrelation Function, (MFCCs) Mel Frequency Cepstral Coefficients and (MFBs) Mel Filter-banks.

Selected LLDs	1st Delta	Functional Sets*
Jitter Local	✗	A
Jitter DDP	✗	A
Shimmer Local	✗	A
F0 by SHS	✓	A,C
Loudness	✓	A,B
Probability of Voicing	✓	A,B
HNR by ACF	✓	A,B
MFCCs[0-14]	✓	A,B
LSP Frequency [0-7]	✓	A,B
log MFB [0-7]	✓	A,B
F0 Envelope	✓	A,B

**Statistical Functional Sets (A,B,C) are defined in Table 3.2.*

The list of the selected LLDs alongside with their statistical functional sets are displayed in Table 3.3. In the first column, the selected LLDs are listed which will be extracted from each frame. The second column indicates if we also append the delta value of the corresponding feature between this value and the corresponding contained in the previous frame. A check-mark symbol ✓ indicates that the delta value is also appended in the final feature vector while an x-mark ✗ reflects the opposite. Under the column "Functional Sets" we can see the corresponding symbol defined in the previous Table 3.2 that indicates the respective set of statistical functions which will be applied on each sequence of frame-level LLDs. As a result, every speech segment is now represented with a fixed-length vector of 1582 features which is the same as the one proposed in [23]. For an utterance-based classification method (see Section 1.3.3) the sequence of vectors would be deprecated into a single 1582-d vector.

3.3 Approaches on Different Timescales

The core of our approach lies in investigating emotion decision scales for different combinations of feature sets and timescales which they are aggregated. Consecutive frames with local features can be concatenated and consequently change the emotion decision-scale. For example, consider that a concatenation of 5 frames may represent a *sad* emotion but each frame independently might not. Accordingly, emotion decision-scale is sensitive to the speech segment's duration on which global features are extracted. A speech segment can be considered as a timespan longer than a frame (e.g. the whole speech utterance could serve as a segment). We focus on the impact that number of concatenated frames and segments' duration have on SER systems which are based on RNN architectures and especially LSTMs. Thus, we find the appropriate decision time-scale for local and global features, respectively.

Next, we divide our approaches to three setups which are practically corresponding to the three main subcategories of SER classification approaches as described in Section 1.3.3.

1. In the first one, we extract Local features on a frame-level (as described in Section 3.2.3) and we try to infer the emotional utterance directly using the input sequence to feed our LSTM. This approach is described in Section 3.3.1 and corresponds to the "discrete" approach following the terminology of Section 1.3.3.

2. In the second one, we extract Global features (as described in Section 3.2.4) over various segment durations of an utterance and try to infer the emotional utterance using the input sequence of segment statistics to feed our LSTM. This approach is described in Section 3.3.2 and corresponds to the "Segment-Based" approach following the terminology of Section 1.3.3.

3. In the last one, we extract Global features (as described in Section 3.2.4) over the whole utterance. Now we only have one feature vector that we use as input for an SVM classifier. This approach is described in Section 3.3.3 and corresponds to the "Segment-Based" approach following the terminology of Section 1.3.3.

3.3.1 Frame-Based

We employ an LSTM RNN as described in [64] and train it with multiple timesteps which correspond in concatenated LLDs. In essence, the input sequence for our LSTM consists of consecutive frame-level features which are concatenated over a selected timescale. For example, if we want to aggregate LLDs over a timescale corresponding to a phoneme then after the extraction of local features (see Section 3.2.3) for each frame, we concatenate all frame-level LLDs using windows of 100ms. Specifically, because the local features are extracted using frames of 30ms with 15ms overlap, each window would be a concatenated vector of 6 frame-level LLD feature vectors.

Each window of concatenated LLDs would correspond to a timestep of the input sequence. These fixed-length vectors of attributes correspond to different timesteps and vary between utterances. For each sequence of timesteps, representing an emotional utterance, the expected output is an emotion label (this is often called many-to-one training [66]). During the training process we train our LSTM using instances from all the available emotions. Likewise in any supervised Machine Learning (ML) problem, in the training process we are given the sequence of the concatenated LLDs $\{\mathbf{x}_0, ..., \mathbf{x}_{T-1}\}$ as input and a discrete emotion y which is the corresponding label. We consider T to be equal to the number of concatenated LLD-vectors which are included in a specific utterance. By using multiple input sequences and backpropagating the error of each prediction during the training process we are able to train our RNN and test it on samples which were not used for training.

RNNs can adequately encode the information enclosed in a sequence of timesteps and produce the expected output on the last timestep. In our case, information about the emotional content would be encoded over all concatenated LLD feature vectors and on the final timestep, the model makes a prediction about the most prominent emotion of a given utterance. When LSTM layers are stacked,

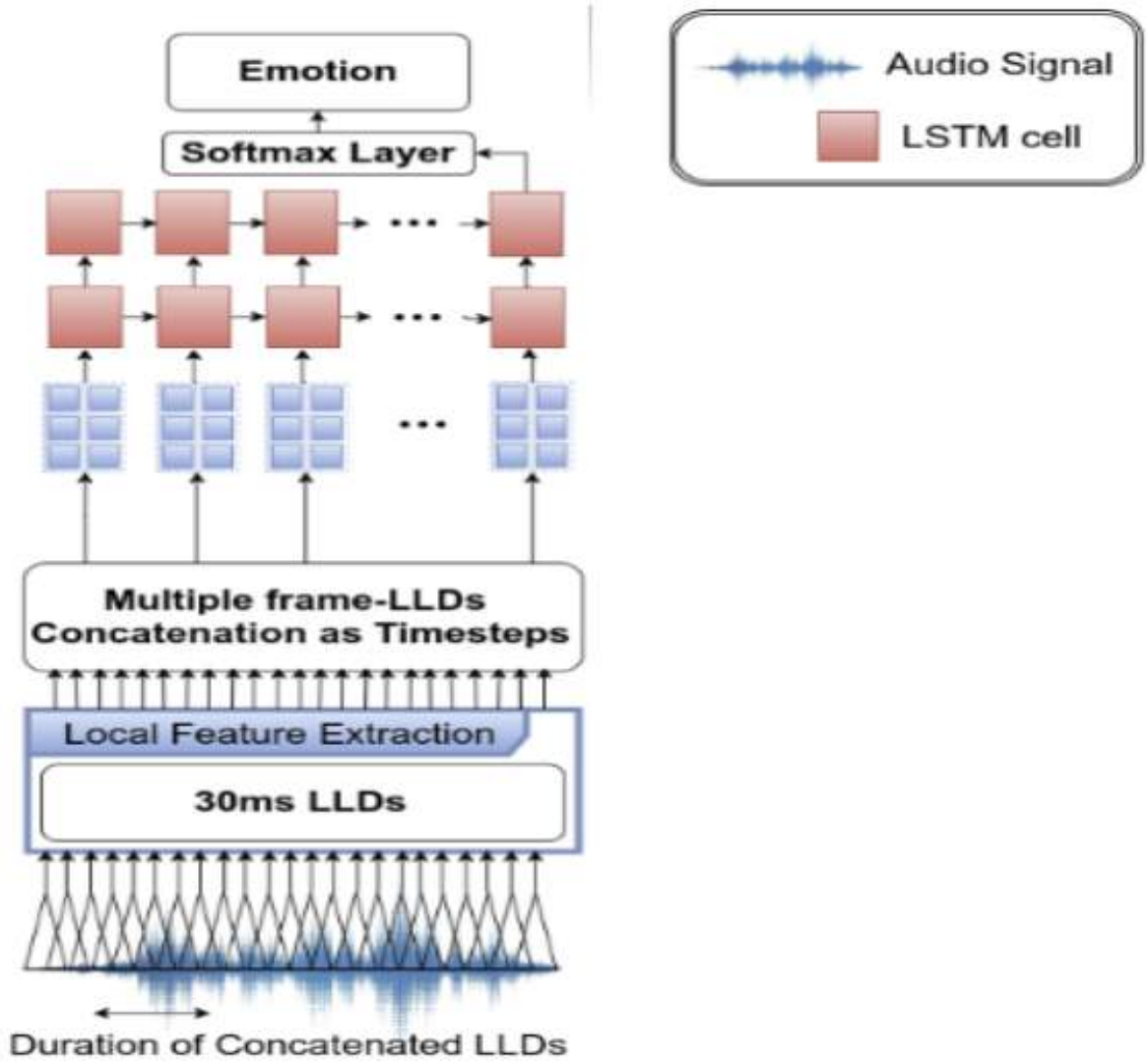


Figure 3.4: Frame-Based Approach using LSTM trained on sequences of concatenated frame-level LLDs

the output of every layer is fed as an input in the subsequent layer. Our LSTM has 2 hidden layers and practically each layer activates the next one in this sequence. On top of the final timestep representation there is a dense output layer which leads to a softmax layer in order to infer the emotional category of the input sequence.

An overview of the whole process is presented in Figure 3.4. Each small blue square corresponds to a vector of local features, extracted from a frame. The cyan rectangle containing multiple LLD-vectors (6 in Figure) corresponds to a timestep of the input sequence $\mathbf{x}_i$. A red square with the connections with the previous and the subsequent layers as well as the recurrent connections indicates an unrolled LSTM unit over the timesteps of the input sequence. The activations of the last timestep are given as input to a dense layer with a number of hidden nodes N_h equal to the number of emotional classes. The prediction is based on the output of the latter layer (which is practically N_h posterior probabilities corresponding to each emotional class) after the application of a softmax function in order to get an one-hot prediction vector. The predicted class will be activated (1 in the respective entry of the vector) while all the other classes would have zeros at their respective indexes.

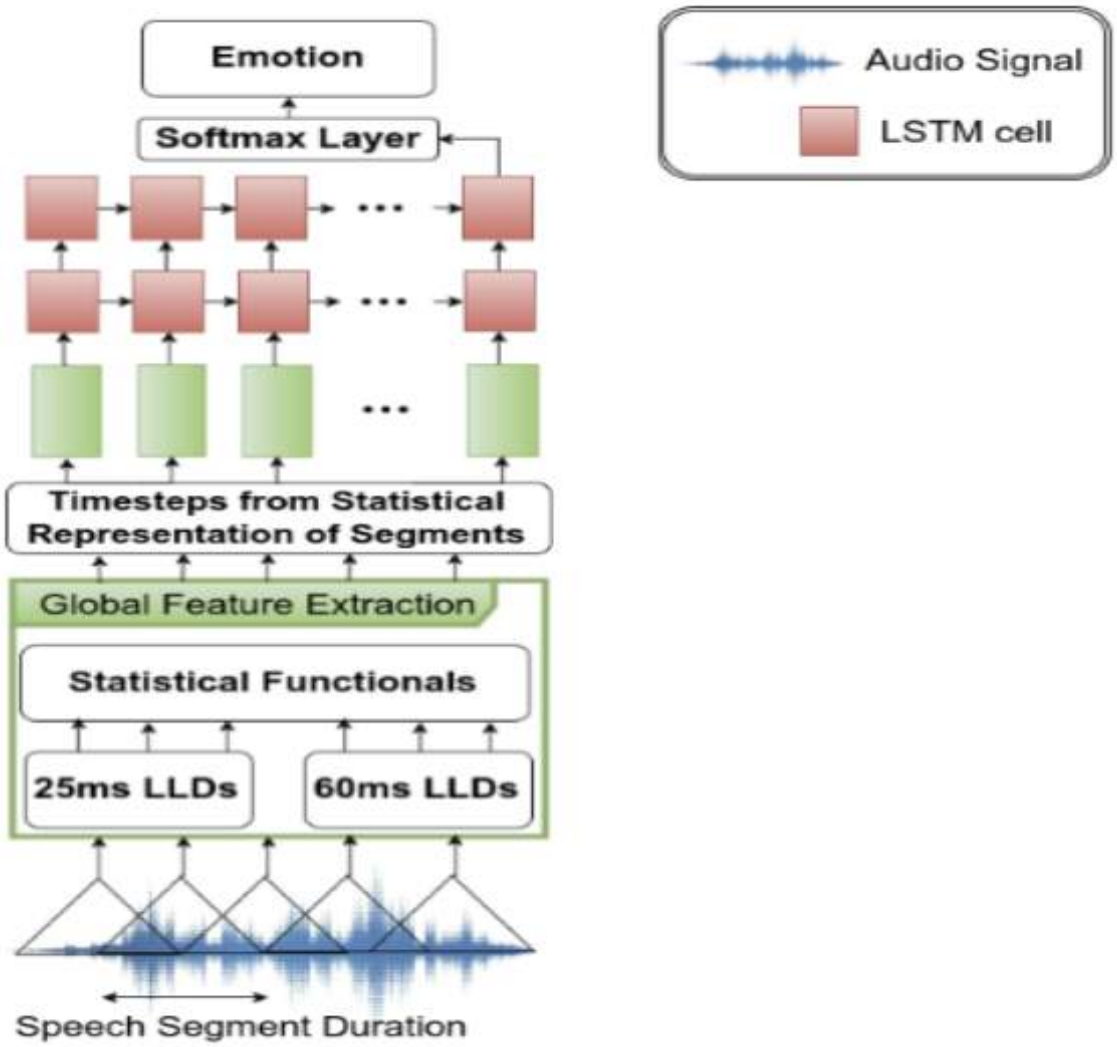


Figure 3.5: Segment-Based Approach using LSTM trained sequences of segment global statistical features computed over

3.3.2 Segment-Based

An LSTM is trained in this approach which has the exact same architecture described in the previous Section 3.3.1. Although the differences between the two approaches revolve around the feature extraction procedure and the formation of the input sequence. In the Segment-Based approach we firstly create overlapping segments from the utterance of longer durations than the ones used for the extraction of LLDs. Next, we align all the frames which belong to each segment and extract LLDs similar to the ones described in the previous section. Specifically, each segment is represented with a global feature vector after the global feature extraction process is applied (see Section 3.2.4).

The overall classification scheme with the breaking into segments, the global feature extraction and the3 final classification using LSTM, is depicted in Figure 3.5. It is evident that the LSTM architecture is similar to the one presented for the LSTM trained on local features from concatenated LLDs (see Figure 3.4) but the input sequence formation has been altered. Specifically, each input timestep corresponds to a 1582-dimensional vector as described in 3.2.4 which is represented with a green rectangle. We notice that the timescale on which the decision of the emotional content is drawn, after the encoding of the input sequence, is the exact same as the one under which the global features are extracted.

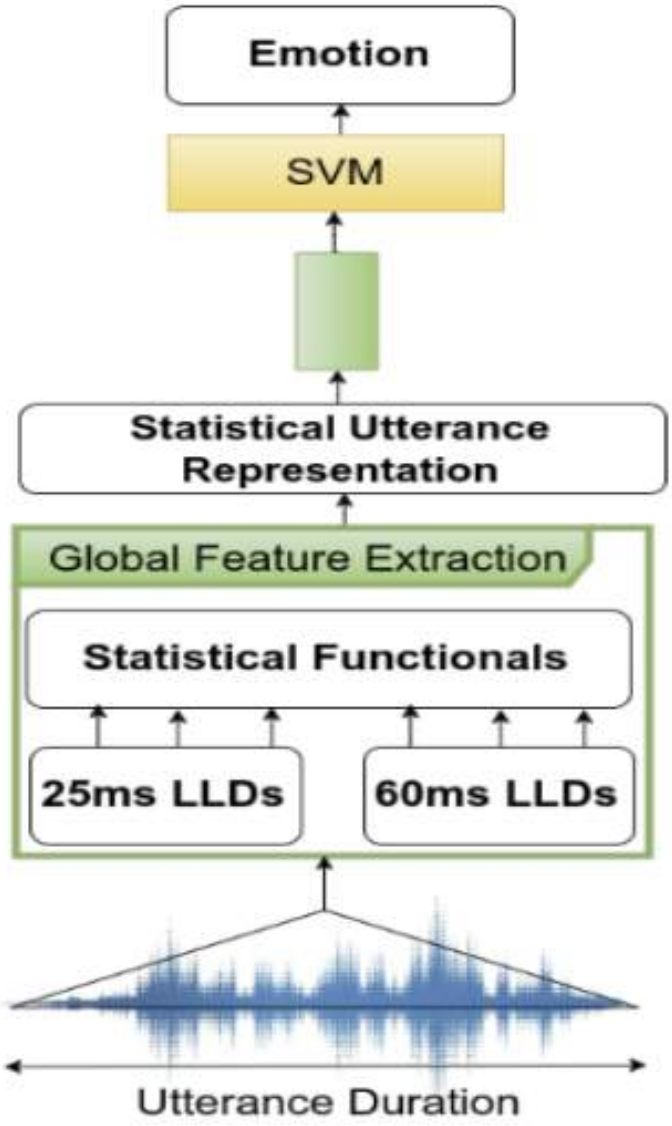


Figure 3.6: Utterance-Based Approach using SVM trained on utterance statistical features

3.3.3 Utterance-Based

In this approach we extract global features as previously mentioned (see Sections 3.3.2, 3.2.4) but we do not break each utterance into overlapping segments. We apply statistical functionals over the LLD-vectors sequence of the whole utterance. Now each emotional utterance is represented with only one feature vector consisting of 1582 attributes. An SVM with a Radial Base Function (RBF) kernel is built and trained on these feature vectors. The overall classification scheme is presented in Figure 3.6.

3.4 Experimental Setup

In this section we provide the experimental setup that we follow in our experiments in order to evaluate our methods on different timescales. We describe the dataset we use and the evaluation scheme we use in order to measure the performance of our systems in Sections 3.4.1 and 3.4.2, respectively. Moreover, we discuss the experimental framework for each of the three timescale-approaches, described in the previous section 3.3, in Sections 3.4.3, 3.4.4 and 3.4.5.

3.4.1 Dataset

For the experiments we use the IEMOCAP database. This database contains audio and visual data from 5 sessions. In each session 2 people (one Male and one Female) perform an acted or an improvised dialogue. For each utterance, 3 human annotators labeled it with a categorical emotion label. We choose only the utterances for which at least 2 out of 3 annotators had similar opinion. Specifically, our dataset comprises of audio signals from 4 emotional categories: *angry* (1103 utterances), *sad* (1084 utterances), *happy* (595 utterances) and *neutral* (1708 utterances). In total we have 4490 emotional utterances. The emotions we kept are the same setup as other experimental setups in the literature [66], [36], [61].

3.4.2 Performance Measurement

We employ a Leave One Session Out (LOSO) schema for testing our models, same as [66], [61], [36]. In each fold (5 in total), one session is used for test and the remaining 4 for train. For every session we test, one speaker is used for validation set and the other for testing. We repeat the experiment by reversing the validation and test sets. The average accuracy from the two speakers is included for the final assessment. All features are normalized by the global mean and the standard deviation from the samples of the dataset. Evaluation is performed by using:

1. **Weighted Accuracy (WA)** which is the percentage of correct classification decisions over the test set
2. **Unweighted Accuracy (UA)** which is the average percentage of accuracies obtained over each emotional class

The latter performance metric is used in order to provide a much more reliable accuracy metric for an imbalanced dataset like the one we use here. For example, in the edge case that we have a classifier that naively votes "neutral" for all emotional utterances, this is often called majority classifier and is widely used for binary classification problems [4]. In this case, a majority classifier for the aforementioned subset of IEMOCAP database (see previous Section 3.4.1) would yield a 38.04% WA but only 25% UA.

3.4.3 LSTM Trained on Frame Level LLDs

As we have previously mentioned in Section 3.3.1 we train an LSTM with two hidden layers. In Section 2.3.1 we offered an extensive explanation of the main LSTM block and how it can be stacked in order to construct deeper architectures like the ones we use here. The two hidden layers have cell sizes of 512 and 256 neurons each. A softmax layer for classifying the 4 emotional categories is placed on top of the final timestep as it is shown in Figure 3.4. We have noticed that performance does not increase when we add extra hidden layers. If we increase the number of neurons on each cell we also increase quadratically the complexity of our model because of the connections between the subsequent layers as well as the probability of overfitting of our model [122]. Overfitting is practically to tune the parameters of our LSTM optimally for the training set and thus the discriminatory regions of the learned parameters are closely tightened to optimize the regions of the training set. In this way, the learned weights of our LSTM would loose their generalization attribute. Practically, the instances of the test set would be classified wrongly because the discriminatory regions of the classes are optimally trained for the training set but cannot be extended in order to include the instances of the test set.

In order to prevent overfitting in our LSTM, we chose to regularize our model by applying dropout on the connections of our Neural Network (NN). Applying dropout on the connections of our NN can be viewed as a procedure where a subset of NN's weights is selected randomly and all units in this set are dropped as well as their connections during training. It has been shown that dropout is an efficient technique for preventing a NN from overfitting [123]. Nevertheless, applying dropout ratio in RNNs is not as easy as all the other architectures such as DNNs and CNNs where only feed-forward

connections are existent. In accordance with [124], we regularize our LSTM by applying a dropout rate of *Dr=0.5*, only on the non-feedback connections of the LSTM. Furthermore, the performance of our LSTM is not further increased when we try different dropout rates.

The model is optimized by minimizing the categorical cross-entropy metric, by using Nadam optimizer [125], which is basically Adam optimizer with Nesterov momentum. Base value of learning rate is set to $\alpha_{base} = 0.002$. The number of training epochs is set to 100 but early stopping is applied when the loss function of the validation set does not improve for 10 consecutive epochs. The configuration which yields the best sum of WA and UA on the validation set is selected. Moreover, batch normalization layers are discarded because of their undesirable effects on faster convergence when they are applied on recurrence connections of an LSTM [126]. Batch normalization is essentially the idea of using statistical values of the activations produced on the forward step between batches of training or testing data. In each layer these statistics are used in order to z-normalize the activations of the current layer and consequently lead all values to a range where gradients are not exploding neither converge to zero. The samples used for training are not vast in order to use all these techniques for faster convergence of our network. On the contrary, as we describe next, our main problem is how to avoid overfitting because our LSTM converges immensely fast. Additionally, batch size is set equal to 400, in order to fit in the memory of our Graphical Processor Unit (GPU). The aforementioned LSTM architecture was implemented using Keras [127] over a Theano [128] back-end.

After the extraction of LLDs as described in Section 3.2.3 we concatenate them in chunks corresponding to longer durations (see Figure 3.4). Consecutive frame-vectors are concatenated in chunks of different lengths and are fed in the LSTM. The length of each chunk is the number of consecutive frame-vectors that it contains. Evaluation is performed for chunk-lengths that correspond to frame (30ms), phoneme (90ms-300ms) and longer segments of speech (400ms-8sec) time-scales for deducing the emotional label. The learning rate for local extracted LLDs is given by the following equation:

$$\alpha_{local} = \frac{\alpha_{base}}{N_{Frames}} \tag{3.6}$$

where the initial or base learning rate is set to $\alpha_{base} = 0.002$ and N_{Frames} correspond to the number of frames which are included in each chunk of LLD-vectors.

This learning rate is employed because when we increase the number of LLDs in each chunk then a reverse proportional decrease would be reflected on the length of the timesteps given as input to our LSTM. In this context, the number of connections in our NN is stabilized and we reduce the number of the timesteps. By doing this, we notice that our LSTM quickly overfits. Hence we employed the aforementioned heuristic technique in order to avoid this problem on all the corresponding timescales on which we concatenate the frame-level LLDs.

3.4.4 LSTM Trained on Statistical Features

For this experiment, global-statistical features are extracted as described in Section 3.2.4. We are using segments of lengths (0.5s-8s) with an overlap ratio of *OL=0.5* between them. The training procedure which is followed here is the similar to the one described in the previous Section 3.4.3. The architecture of the LSTM used is actually the same as the one we use for training with concatenated frame-level LLDs, which is also evident by comparing the Figures 3.4 and 3.5. However, we select a different method in order to adapt the learning rate for global-statistical features. Using the same $\alpha_{base} = 0.002$, the initial learning rate is given by the following equation:

$$\alpha_{global} = \frac{\alpha_{base} \cdot OL}{100 \cdot T_{segment}} \tag{3.7}$$

where $T_{segment}$ is the duration in seconds of each segment which is used to extract the global features from.

3.4.5 SVM Trained on Utterance Level Statistical Features

As described in Section 3.2.4 statistical features are extracted from the whole utterance which now serves as the only segment. We use an RBF kernel and regularize gamma coefficient γ with the number of features for every utterance. Namely:

$$\gamma = \frac{1}{1582} \tag{3.8}$$

For an extensive explanation of how all these parameters configure the multiclass SVM classifier we refer the reader to the previous Section 2.2.2. In order to optimize our model we select the grid space of our SVM to be equal to the set of values for cost coefficient C which lie in the $[0.001, 60]$ interval. For every test speaker, the cost coefficient C value which yields the best sum of WA and UA for the corresponding validation speaker is chosen. All other hyperparameters are set to their default values (we refer the reader to the exact LibSVM implementation [129]).

3.5 Experimental Results and Discussion

3.5.1 Optimal Timescales for LSTM Trained on Local Features

Results for the local feature training experiment are displayed on Figure 3.7. It is evident that a concatenation of 5 frames' LLDs yields the best performance in both WA and UA metrics. This corresponds to phoneme time-scale (100ms) decision for the emotional context. As we are increasing the number of frames in every chunk, we change the decision scale to larger scales (1-8s) and notice a gradual decline in both metrics WA and UA. We assume that this decline in performance metrics might be occurring because LLDs misrepresent silent frames of speech and when they are concatenated for long speech durations, they drive the RNN to misleading abstractions for the emotional manifestation of utterances. Moreover, when the decision is made on frame level, silent frames are mistakenly characterized by the same emotion label as higher energy frames. It is also evident that because of the imbalanced distributions of the emotional utterances we always achieve slightly better WA compared to the UA.

We could assume that the peak for both performance metrics of WA 59.14% and UA 54.2%, when training the LSTM with chunks of LLDs corresponding to time durations of 100ms consist of an optimal decision timescale for this type of features. Presumably, when using chunks of only one frame in each chunk, then we widen the length of the input sequence $\{\mathbf{x}_i\}_{i=0}^{T}$ for a sequence with $T+1$ timsteps. This could cause the problem we have previously discussed in Sections 2.3.3 and 2.3.2 where the informational flow for the emotional dynamics are vanishing through the multitude of timesteps leading to probably a much less resiliet SER system because of the loss when trying to tune the weights using backpropagation on all these timesteps. Although, the core idea of our approach is not to implement the most resilient RNN architecture with extra mechanisms with attention mechanisms or multiple RNNs in order to cope with this problem. These experimental results offer a valuable qualitative insight on how we can obtain superior performance when we use simple RNN models. To this end, the combination of the aforementioned mechanisms and architecture with the optimal timescale of the input features could potentially lead to a robust estimate for the selection of the SER model when using frame-level LLDs.

We should also not be oblivious to the fact that because the computation of the gradient and the update of weights is performed sequentially through the timesteps of the input sequence it would be much more convenient for the training process to have as much fewer timesteps as we could. However, when the vector of each timestep is very high in dimensions we expect to see that the network cannot aptly encode the dependencies between all those features in different timesteps with the same number of trainable parameters. This is also evident from the results we present in figure 3.7 where the architecture of the RNN remains stable while we concatenate multiple frame-level LLD vectors in each timestep. In the edge case of 533 concatenated frames we see that we obtain very low

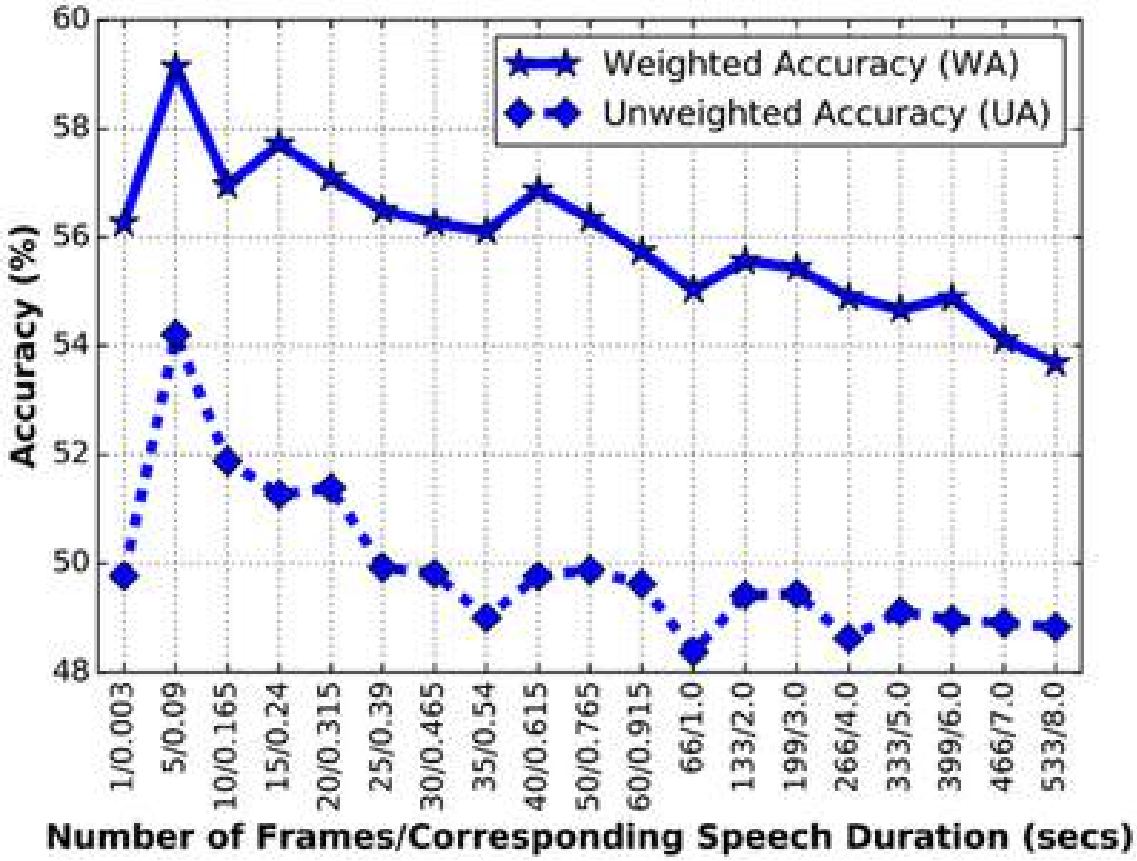


Figure 3.7: Weighted Accuracy (WA) and Unweighted Accuracy (UA) of LSTM trained on various lengths of concatenated frames with local features

performance on both metrics WA and UA mainly because of the reason we mentioned earlier. On the contrary, phoneme scales seem to be adequately preserve emotional information and provide a good solution for the trade-off between the length of the modeling sequence and the accuracy of the model.

3.5.2 Optimal Timescales for LSTM Trained on Global Features

The results of the experiment where the LSTM is trained on global features are shown in Figure 3.8. Results in Figure 3.8 demonstrate the effect of training an LSTM with global features over different time-scales. Statistical features on syllable (0.5s) and utterance (6-8s) time-scales do not perform as well as words (3-4 seconds) time-scales. Speech segments which downscale to phoneme level durations do not enclose sufficient emotional information when we represent them with global statistical features. On the other hand, when we extract global features from nearly the whole utterance, poor results are obtained. It is important to notice that if an utterance exceeds the time duration of the selected segment we zero pad the former in order to create signal vectors with length at least equal to the one of the selected segment. The misrepresentation of global features for longer durations are probably due to the statistical misrepresentation of the whole emotion utterance when using so little in numbers timesteps for our input sequences from which we want to infer the emotional content. If the number of the input timesteps are relatively low then it would be impossible to learn the long and short term dependencies between those feature vectors for each timestep. This is practically the same trade off which we discussed earlier where we would like to obtain the least number of timesteps which can facilitate the faster convergence but without leading to misrepresentations of the emotional content for the utterance.

Deriving higher level statistical functions from multiple LLDs over speech segments, leads to a more salient representation of underlying emotional dynamics. Moreover, by combining the efficacy

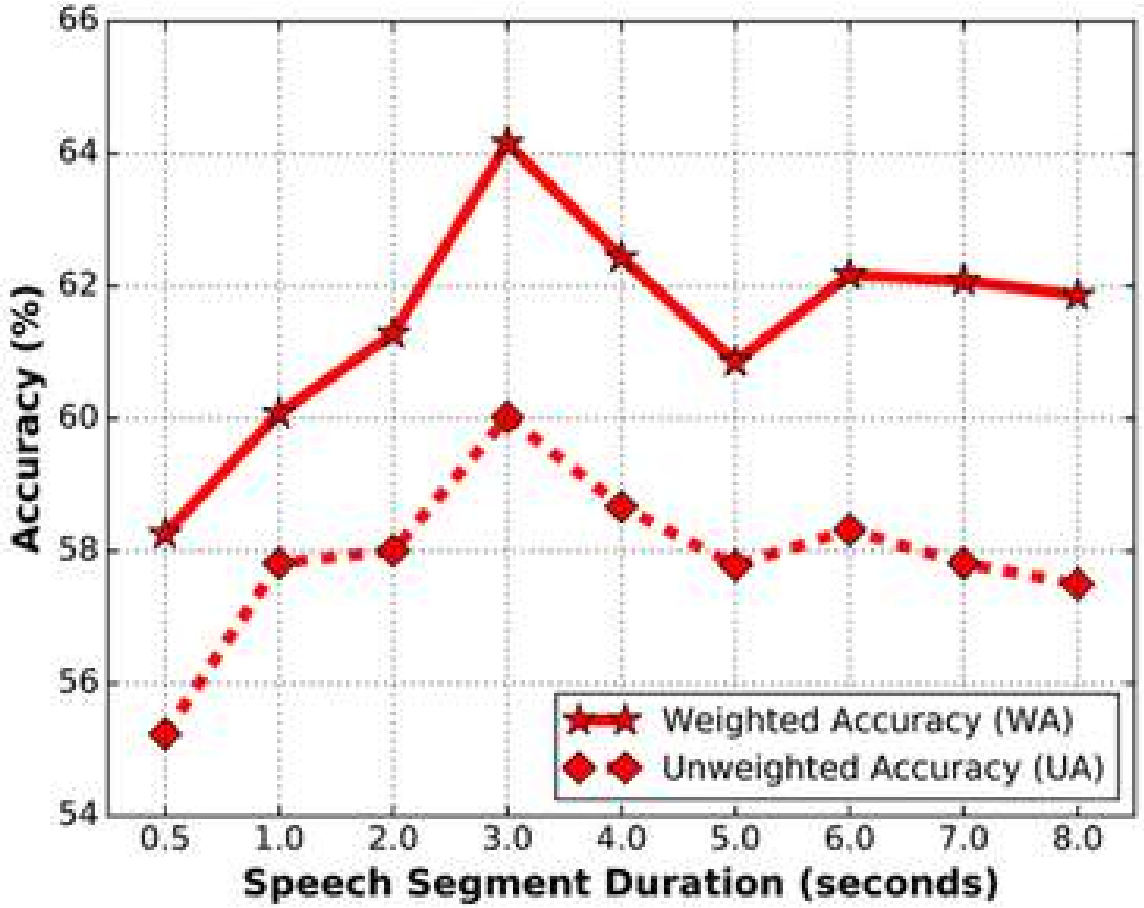


Figure 3.8: Weighted Accuracy (WA) and Unweighted Accuracy (UA) of LSTM trained on global features over various segments durations

of an LSTM to interpret the long-short term dependencies and the aforementioned statistical representation over speech segments on word-timescales, we obtain the best performance in both WA and UA metrics. This is another indication that the notion of choosing the appropriate timescales is key for SER systems when they follow a segment-based approach with global features.

3.5.3 Comparison Between Timescales

As we have extensively discussed in previous sections we follow all three approaches of a more direct LSTM trained on frame-level LLDs, a segment based approach on global features extracted from speech-segments as well as an utterance based approach using SVMs. The results of the best models for each category are presented in Table 3.4.

Table 3.4: Accuracy of Proposed Models on Various Timescales

Model	Type of Features	WA (%)	UA(%)
SVM	Statistical over the whole utterance	53.54	49.23
LSTM	LLDs chunks of 90ms	59.14	54.2
LSTM	Statistical over 3 seconds segments	**64.16**	**60.02**

It is evident that the segment-based approach provides a far more resilient SER model. Overall, concatenation of multiple LLDs at SER tasks necessitates the existence of a mechanism that confines the influence of non-emotional frames [60], [66]. The need of such a mechanism can be avoided by employing statistical representation on a word time-scale. Intuitively, this resembles the human

Table 3.5: Accuracy of Models in the Literature and Proposed Model

Model	Type of Features	WA (%)	UA(%)
Best LSTM [35]	Spectrogram	61.71	58.05
BLSTM-SUA [66]	LLDs	59.33	49.96
BLSTM-WPA [65]	LLDs	63.5	58.8
BLSTM-ELM [61]	LLDs chunks of 250ms	62.85	**63.89**
LSTM	Global features Section 3.2.4 over 3 seconds	**64.16**	60.02

Weighted Pooling Attention (WPA), Sub-Utterance Attention (SUA)

system of emotional deduction by taking into consideration information from a couple of words.

Our proposed LSTM model, trained on statistical features over a 3 seconds speech segment, obtains a relative improvement of 5.02% in WA (59.14% $\rightarrow$ 64.16%) and 5.82% (54.2% $\rightarrow$ 60.02%) in UA from the best performing LSTM trained directly on LLDs and 10.62% in WA (53.54% $\rightarrow$ 64.16%) and 10.79% (49.23% $\rightarrow$ 60.02%) in UA from an utterance level decision SVM. (see Table 3.4).

3.5.4 Comparison with the Literature

Comparable literature models are presented in Table 3.5. Our model surpasses all attention based RNNs: weighted pooling attention BLSTM [65] in both WA and UA by 0.66% and 1.22%, sub-utterance attention based BLSTM [66] in both WA and UA by 4.83% and 10.06%, respectively. Authors in [61] test their model only on improvised scripts which is a subset of the IEMOCAP database and as it is demonstrated in [36] they tend to give higher scores in WA and UA. Moreover, they test only on one of the speakers for each IEMOCAP session without specifying which one is used for the final assessment. Despite that, our simple LSTM scores 1.31% higher in WA from their proposed BLSTM-ELM model. Finally, the current state-of-the-art approach in [35] was obtained by exclusively using the final session as a validation set and testing only on the other 4. We should also notice that in this publication, authors have merged the excitement class with the happy instances which is beneficial in terms of performance in UA metric. Nevertheless, their proposed CNN architecture yields very similar results (64.78% WA and 60.89% UA) to ours (64.16% WA and 60.02% UA). Irregardless of the exclusion of sessions, our model clearly outperforms all simple RNN architectures (LSTM, BLSTM) reported in the literature. Comparing with the best RNN-LSTM architecture reported in [35], we obtain a relative improvement of 2.45% in WA and 1.97% in UA.

Chapter 4

Modeling Nonlinear Recurrence Dynamics for Speech Emotion Recognition

This chapter is an extended version of the paper [1] which it has been selected for oral presentation at Interspeech 2018: Conference of the International Speech Communication Association which has been planed to take place in Hyderabad, India, September 2-6 2018. If the reader needs to cite parts of this chapter then it would be preferred to use the following reference (or the most updated one under the same title and author specified):

- *Efthymios Tzinis †, Giorgos Paraskevopoulos †, Christos Baziotis, and Alexandros Potamianos, "Integrating recurrence dynamics for speech emotion recognition," in Proceedings of INTERSPEECH, (in press), 2018.*
  †Both authors contributed equally in this work

4.1 Motivation

As it has already been mentioned previously in Section 1.4.2 nonlinear phenomena in speech production mechanisms are key aspects of understanding the variations in speech generation process and utilize them for our purposes. Specifically, highly nonlinear oscillations of the vocal folds constitute analysis of the nonlinear dynamics of voice a key aspect to understand speech and emotions which the latter might carry. Some of these phenomena like modulations oh the speech airflow and various turbulences, octave jumps, noise concentrations as well as biphonation which is actually the existence of sub-harmonics in the frequency domain encompass some of the most indicative facts that the linearity assumption of the source-filter model might not hold. Consequently, the stationarity of the speech frames in order to perform Fourier analysis is vital for the latter approach but questionable in general. It might be particularly beneficial for SER systems to exploit this kind of nonlinear information in order to achieve better performance.

Building upon capturing the nonlinear dynamics of speech in short-term windows, it is crucial to underpin the process of nonlinear analysis from speech signals. Nonlinear analysis of a speech signal through the reconstruction of its corresponding Phase Space (PS) lies in embedding the signal in a higher dimensional space where its dynamics are unfolded [78]. The state-space is constructed by using time-delayed versions of the signal, aligning them, concatenating them and then using the concatenated vector as the reconstructed PS representation. The reconstructed PS of the signal is just another approximation of the true dynamics of the signal for which we always try to best estimate the parameters of the time-delay and embedding dimension in order to make a successful unfolding of the underlying dynamics. Recurrent patterns of these orbits are indicative attributes of system's behavior (see discussion in Section 1.4.2 and especially Figure 1.5 in order to validate the recurrence dynamics of a speech frame) and can be analyzed using Recurrence Plots (RPs) [96]. RPs are just local distance matrices which are able to preserve the recurrence information of the system under analysis. To this end, Recurrence Quantification Analysis (RQA) provides complexity measures for an RP which are capable of identifying system's transitions between chaotic and order regimes [84].

4.2 Related Work

Despite the great progress in SER (see Section 1.3.4 for some state of the art approaches), most contemporary approaches completely neglect the nonlinear aspects voice dynamics for the sake of making the feature extraction process less time consuming and following more direct approaches with minimum speech processing involved. However, there are various works where nonlinear analysis using PS and other methods have been utilized for SER which are briefly discussed below.

Apart from the PS analysis, other approaches follow different methods in order to extract nonlinear features from emotional utterances. In [130] modulation spectral has been extracted in order to capture both acoustic frequency, and the temporal modulation frequency components over the instantaneous phase and amplitude for SER tasks. The combination with conventional feature sets was quite beneficial for the performance of the models. To this end, in [43] micro-modulation features from instantaneous amplitude and phase have been investigated and evaluated on Berlin database [121] as well as their combination with MFCCs. In the same context of modulations of speech signals Teager Energy Operator (TEO) has been utilized for SER [40]. TEO is a nonlinear operator applied on time-series and especially speech signals. It has had many interesting applications in various fields of digital signal processing as well as a simple and efficient definition and implementation. On of the most successful applications of TEO has been the demodulation of AM-FM signals which has been a fruitful area of research during the previous years [41].

Considering the classical analysis of time series using reconstructed phase spaces we can see a vast amount of work which has been focused on how to apply this analysis for SER. In [131] a neural network was trained using a set of handcrafted features which are drawn from the representation of the signal in the reconstructed PS are used. The features used for training the NN are actually statistical functionals applied on frame-level nonlinear descriptors. The descriptors used were some of them which have been widely used in ASR [78]. Correlation entropy, Lempel-Ziv complexity, the first minimum of the mutual information, Shannon entropy, Correlation dimension and the Hurst exponent were used for discrimination between anger, fear and neutral emotional utterances from Berlin database [121]. A qualitative finding of this study was that emotional utterances associated with fear, and anger show more noisy and complex structures than the ones associated with neutral state. Moreover, in [42] a variety of geometrical measures from reconstructed PS orbits have reported significant improvement on SER when combined with conventional feature sets. Specifically from each PS on frame-level some measures have been calculated such as: distance to the centroid of the attractor, the length of trajectory legs, current angle between consecutive trajectory legs and the distance of points to the identity line and after that sets of statistical functionals have been applied for the final classification using SVM.

Although the latter approaches are quite interesting and provide insightful results about the nonlinear behavior of speech signals of emotional utterances, the recurrence properties of the latter are still not utilized for SER. RPs from reconstructed phase spaces are able to capture some of the most prominent aspects of the underlying dynamical system [84] and thus they could be used for the analysis of emotional speech utterances as well. However, the information from RPs have not yet been utilized for SER tasks. As an exception to the previous statement in [132] RQA measures which are extracted from RPs of reconstructed phase spaces of speech frames have been shown to be statistically significant for the discrimination of emotions. Although the statistical analysis involved different parameters of the reconstructed PS and RQA measures used widely for analyzing RPs [79], an actual SER experimental setup is missing. This is quintessential because the statistical significance shown in a study could be quite misleading for a real case where one needs an autonomous system performing emotion classification. Specifically, the pair wise statistical analysis cannot substitute the true experimental setup using all features and specific models which can be used in the real world.

On the contrary, the recurrence information from RPs and especially feature obtained by performing RQA have been extensively used in many other areas and applications which are also related with emotion recognition but not from speech. For instance, in [133] an analysis of time-series and complex networks using recurrence plots has been presented. Complex network measures were shown

to represent structural aspects of dynamical systems that are complementary to those characterized by other methods of time series analysis. Moreover, in [134] RQA was utilized to measure structural coupling and synchronization in natural and clinical verbal interactions. In [135] body motion data were analyzed for finding deception using dynamical signatures extracted from RPs. their results indicated that continuous fluctuations of deceptive movements in the upper face as well as the arms, are characterized by dynamical properties of less stability and more complexity which was reflected in the respective RQA measures. Additionally, in [136] RQA has been employed for identifying dynamical patterns of brain stimuli and emotion classification of the respective ElectroEncephalogram Signals (EEGs). Interestingly, PS trajectories display higher periodicity during exciting negative while in positive emotions the underlying system is highly chaotic. The findings of the aforementioned study suggests that RQA has the potential to discover differences of brain signal features in response to an emotional stimulus which is quite intriguing if one thinks if this could also apply to emotional signals from speech.

In contemporary approaches, various other approaches have been studied for the efficient extraction of the recurrence properties of signals. For instance, in [137] new RPs are extracted for different frequency scales in order to analyze the recurrence properties of the input signal by utilizing a multi-timescale approach. In other works, nonlinear recurrence information has also been extracted directly from RPs without the utilization of RQA measures. Specifically, in [138] a CNN was trained using RPs for general time-series classification task and shown that this method provides a resilient method for classifying both stationary and non-stationary time-series.

4.3 Reconstructing Phase Spaces from Speech Signals

4.3.1 Unfolding of the True Dynamics of Speech

In general, it is assumed that speech signals since they are generated from such a complicated and highly nonlinear system such as the human speech production mechanism (see Section 1.4.2) they carry latent characteristics of the underlying dynamics of the generator system which is still unknown [78]. In other words, it is assumed that the sampled speech signal $\{s(i)\}_{i=1}^{N}$ which is a 1-dimensional function of time could represent a projection of the true system $\{\mathbf{s}^{\star}(i)\}_{i=1}^{N}$ where $\mathbf{s}^{\star}(i) \in \mathbb{R}^{d^{\star}} \ \forall 1 \leq i \leq N$. We assume that $d_{\star}$ corresponds to the dimensions of the vector which is the representative of the true dynamical system under analysis. Essentially, we would like to reconstruct the dynamics and invariant properties of the unknown system $\{\mathbf{s}^{\star}(i)\}_{i=1}^{N}$ by using only its one-dimensional projection $\{s(i)\}_{i=1}^{N}$.

Equivalently, we could gain a more intuitive insight for the concepts of the reconstruction of the true dynamics by presenting a paradigm in real-life. Imagine a sphere with radius ρ lying in a 3-dimensional space and then projecting it onto a 2-dimensional plane. This would result in a circle with also radius ρ. If one could only see the projection in a 2-dimensional plane, it would not be possible to understand that this projection corresponds to a sphere lying in a 3-dimensional space. Moreover, if we project vertically a cup of tee with a circular bottom of radius ρ lying in a 3-dimensional space onto the same plane and checking the resulting circle we would end up with an identical shape like the one produced by the projection of the sphere. It would also be impossible to distinguish the initial 3-dimensional object from which this circle has been derived as its projection. Now imagine that the initial 3-dimensional object represents the true dynamics of each type of system and the projection on the two dimensional plane is what we can see. In this case, we have two types of systems: the cup system and the sphere system but the observable representation is common (circle on the plane) and we cannot be sure about which system generated this projection. Although, it would quite useful to gain information about which object has been used for the projection with just observing projections with nuance differences which can be utilized towards the determination of the corresponding original system.

Going back to our purpose, we seek to disclose the true dynamical aspects of the speech production

when emotions are in play. Purportedly, if we could capture dynamical aspects such as recurrence patterns from the attractors of emotional speech frames and aggregate them we could build more resilient SER systems. In this context, we need to find ways in order to be able to reconstruct, partially at least, the true dynamics of the system by unfolding them in higher dimensions. It is evident that this process is not trivial and careful consideration of all the variables for this reconstruction should be made. As shown next we will define how we can reconstruct locally the phase space of each speech-frame in order to use this information for extracting emotional content from the viewpoint of dynamical signatures.

4.3.2 Definition

Given a speech frame with N samples $\{s(i)\}_{i=1}^{N}$ we reconstruct its corresponding PS trajectory by computing d_e time-delayed versions of the original speech frame by multiples of time lag τ and creating the vectors lying in $\mathbb{R}^{d_e}$ as specified in the Equation 2.22.

In order to reconstruct the PS of each speech frame we need to specify the two parameters of time-lag τ and the embedding dimension d_e. In the following Sections 4.3.3 and 4.3.5 we will analyze fully how these parameters are selected as well as some qualitative results by changing these parameters for different durations of speech segments. In Section 4.3.6 we provide some PS visualization of various segments.

4.3.3 Estimating time-lag parameter

In order to estimate the time-lag parameter τ for the number of samples of delay that will be enforced for the input signal $\{s(i)\}_{i=1}^{N}$ we are using the AMI criterion $\mathcal{I}(\tau)$ (see definition in Section 2.4.2) in order to select an adequate time-lag τ for our input signal. The function $\mathcal{I}(\tau) : \mathbb{N} \rightarrow \mathbb{R}$ would map all selected integer values for τ to real numbers, indicating the corresponding mutual information for each time-lag.

In Figure 4.1 we provide an example of AMI for a speech segment of 200ms. It is evident that for small time-lags there is generally higher mutual information while for very big values we can see that the opposite is true. Moreover, the periodicity in AMI plot corresponds to recurrence properties of the speech-segment for certain time-lags which indicate not only linear but also nonlinear correlations.

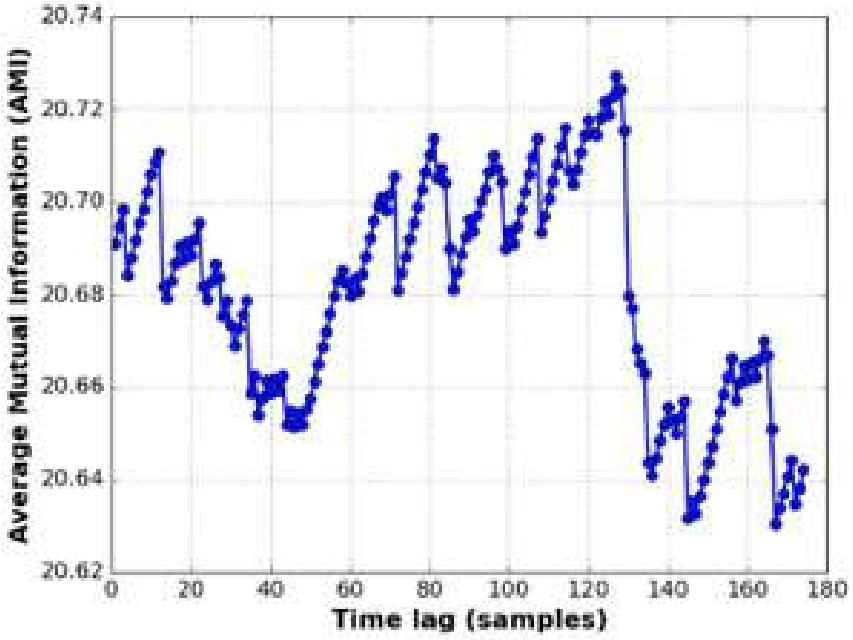


Figure 4.1: Average Mutual Information for a speech-segment of 0.2 seconds

4.3.4 Analyzing AMIs for Different Speakers and Emotions

In Figure 4.2 a variety of inverse AMI plots are displayed for different speakers and emotions for the phoneme /ae/. The utterances used for this Figure are drawn from Surrey Audio-Visual Expressed Emotion (SAVEE) Database [139] by following the same aliases for the speakers and available emotions. We can see that the AMIS are decreasing (in the plot the inverse of AMIs is displayed) because of the declining correlations of the input signals. The similarities among speakers or emotions of the AMI function are evident. for example, for the speaker *JK* the *sad* and *neutral* expressions of /ae/ are much more similar than the other two emotions displayed. Additionally, the *angry* utterances for both speakers *DC* and *JK* seem quite similar. However, these functions are extracted as measures for estimating the parameters of the PS. Hence, these heuristic algorithms for estimating parameters cannot be used directly for the classification of emotions.

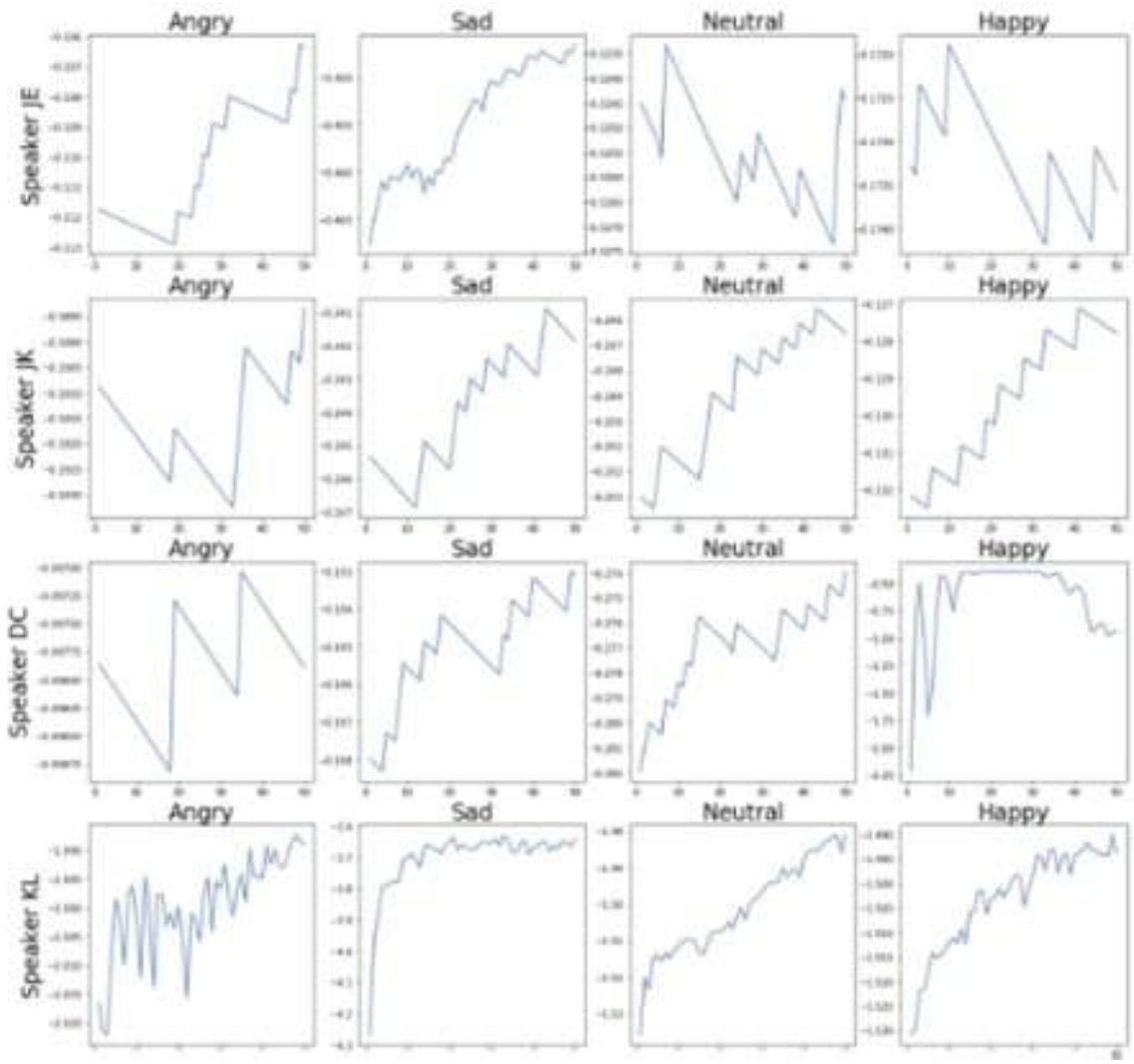


Figure 4.2: Inverse AMI plots for Different Speakers and Emotions for the phoneme /ae/

In order to avoid extreme events of ultimate correlations of the values of $\{s(i)\}_{i=1}^{N}$ and $\{s(i+\tau)\}_{i=1}^{N}$ or cases where the two signals are completely uncorrelated. These two types of cases typically correspond to values that are very small or very big, respectively. In order to avoid such extreme cases, we select the time-lag τ based on the first local minima of the AMI $\mathcal{I}(\tau)$. More formally, we select the following time-lag:

$$\tau = min\{\hat{\tau} \mid \mathcal{I}(\hat{\tau}) < min(\mathcal{I}(\hat{\tau}+1), \mathcal{I}(\hat{\tau}) < \mathcal{I}(\hat{\tau}-1))\} \tag{4.1}$$

This process of selecting the first τ which locally minimizes the AMI function $\mathcal{I}(\tau)$ would produce different results for each length of a speech segment. Thus, if we try to break each utterance in frames or segments we will work independently on each of these time-periods for the reconstruction of the corresponding PS. thus it would be important to further analyze how the time-lag parameter would be estimated using various durations for breaking into speech segments/frames.

In Figure 4.3 we have selected an *angry* utterance of 3 seconds and we run the previous algorithm in order to estimate the optimal time-lag for the reconstruction of the phase spaces for various time-durations of speech segments and frames. The initial signal was sampled at 44100 kHz because it is also a sample of the SAVEE dataset. We have confined the search for the optimal τ by setting a maximum $\tau_{max} = 16$. As the time duration of the segments increases, we can see that either $\tau = 1$ or $\tau = 16$because the correlation in between the same signal is defined by a monotonous function which is not creating any local minimum in such a small number of available time-lags to search in. However, the most prominent results of this figure is that for the timescales which correspond to time-frames of around 20ms (which are the ones which are also used as well in traditional speech processing [6]) the distribution of the selected time lags is not dispersed over the two extreme values for time-lags. This is indicative of the different time-lags that will be selected for different frames which might contain silence periods, vowels or fricative sounds.

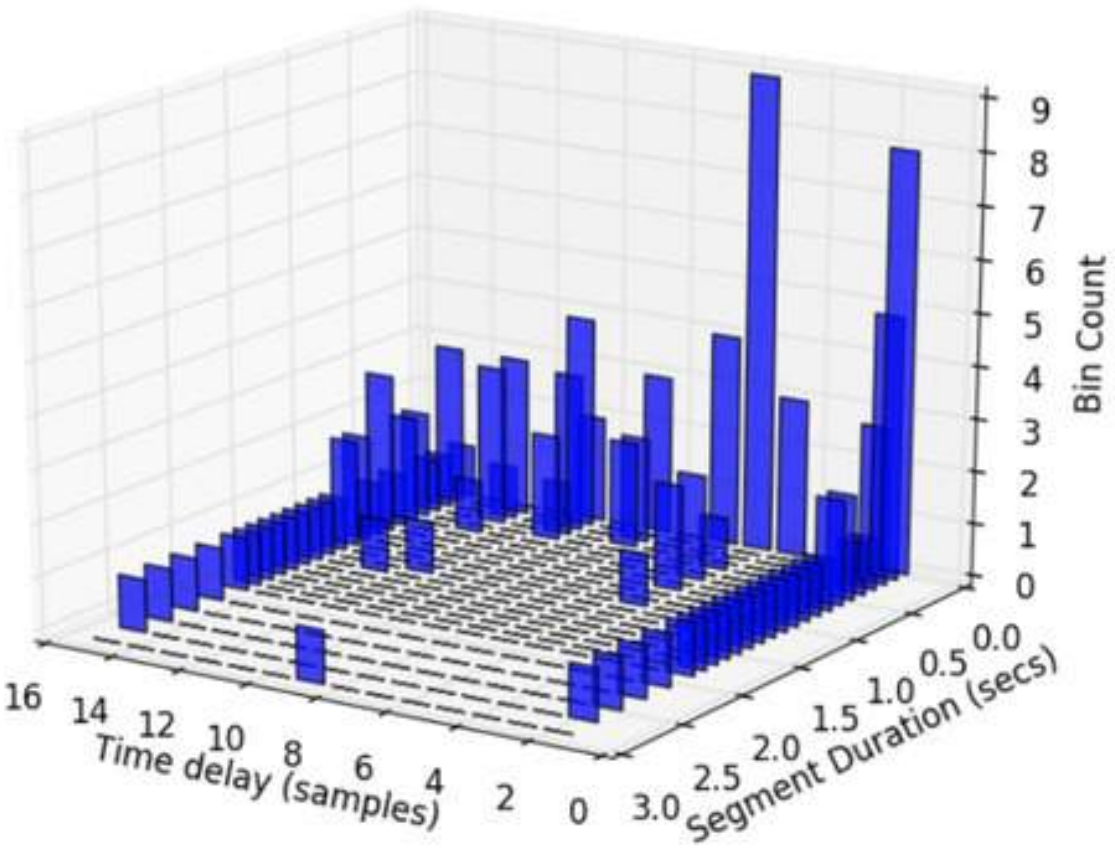


Figure 4.3: Selected time-lags for an *angry* utterance of 3 seconds for various time-durations

4.3.5 Estimating embedding dimension parameter

For this section we will assume that the time-lag τ is also estimated as it has been shown in the previous Section 4.3.3 or it is estimated otherwise. Now that we have set the time-lag τ we should also specify the number of the multitude of the time-lagged vectors that we will use for the reconstruction of the

phase space of the given signal $\{s(i)\}_{i=1}^{N}$. This crucial because as we have previously mentioned (see Sections 2.4.4 and 4.3) the unfolding dynamics of the input signal should aptly preserve the invariant quantities of the original complex generator system $\mathbf{s}^{\star}$. This is applicable only if the selected dimensions d_e of the reconstructed phase space are able to unfold the dynamics without collapsing trajectories of the approximated manifold $\mathcal{M}$.

For the estimation of the embedding dimensions we will use the False Nearest Neighbors (FNN) heuristic method as it described in the previous Section 2.4.4. In order to gain an insight of how the percentile of the false nearest neighbors changes as we add more dimensions for the reconstruction of the phase space we plot these numbers for different lengths of speech-segments. In Figure 4.4, we present the percentiles of the total number of the false nearest neighbors as we add an extra dimension to the reconstruction of the PS from the previous number of dimensions. This plot refers to the segmentation of the initial emotional utterance of 3 seconds which was also used in our previous experimentation for estimating the selected time-lag τ for each segment individually (see Figure 4.3). Specifically, in Figures 4.4a and 4.4b we present the percentiles of the FNN for an estimation of the time-lag by the first local minimum of AMI and an ad-hoc selection of time-lag $\tau = 1$, respectively. It is quite clear that in the first method 4.4a the percentiles of the false nearest neighbors is quite smaller than the ones presented in 4.4b because of the estimation of the time-lag parameter τ based on the local minimum of AMI instead of using ad-hoc values. Noticeably, this statement holds for most of the durations of segments presented. In both diagrams we can see that as we increase the number of dimensions, the number of FNN converges to zero, which is in accordance with our intuition as well as the mathematical description of the algorithm (see Section 2.4.4). Additionally, we can qualitatively reassure our belief that for $d_e = 3$ we have an adequate number of FNNs which is smaller than 20%.

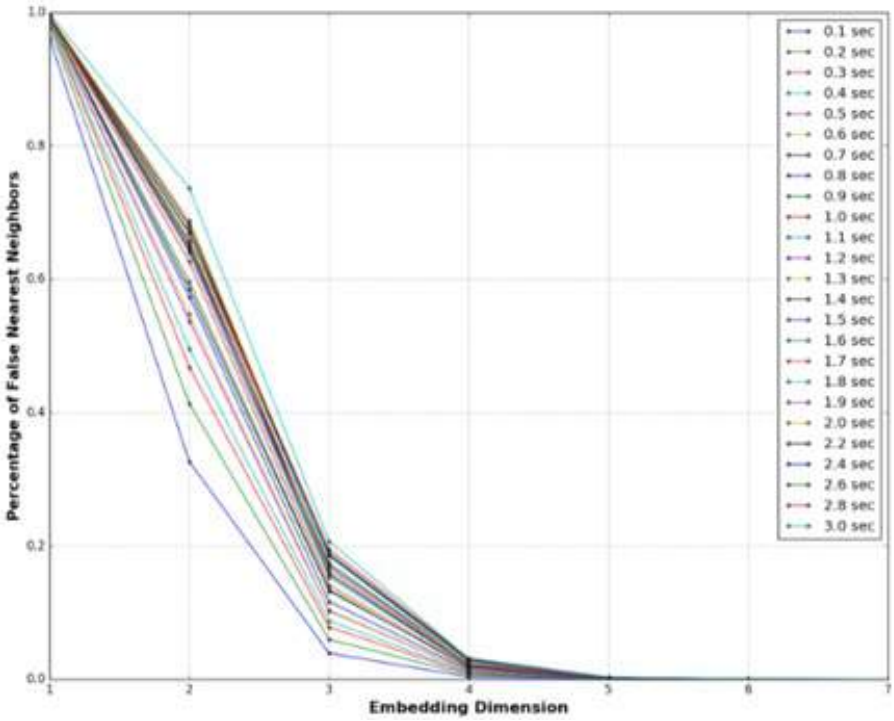
In general, we do not care about longer speech segments than the ones which correspond to the frame-timescales cause this is the timescale from which we would like to extract the local PS orbits of the speech frames. This is quite reassuring as our method indicates that we could actually reconstruct the local recurrence dynamics from each speech frame by using only 3 dimensions for the reconstruction of its local PS. In other words, as the number of FNN does not decrease significantly by adding an extra 4th dimension we can assume that the dynamics of the true system $\mathbf{s}^{\star}$ would be aptly unfolded and described using an embedded manifold $\mathcal{M} \in \mathbf{R}^3$. The latter result is also crucial as we can visualize these unfolded dynamics.

4.3.6 Reconstructing Phase Spaces for Various Speech Segments

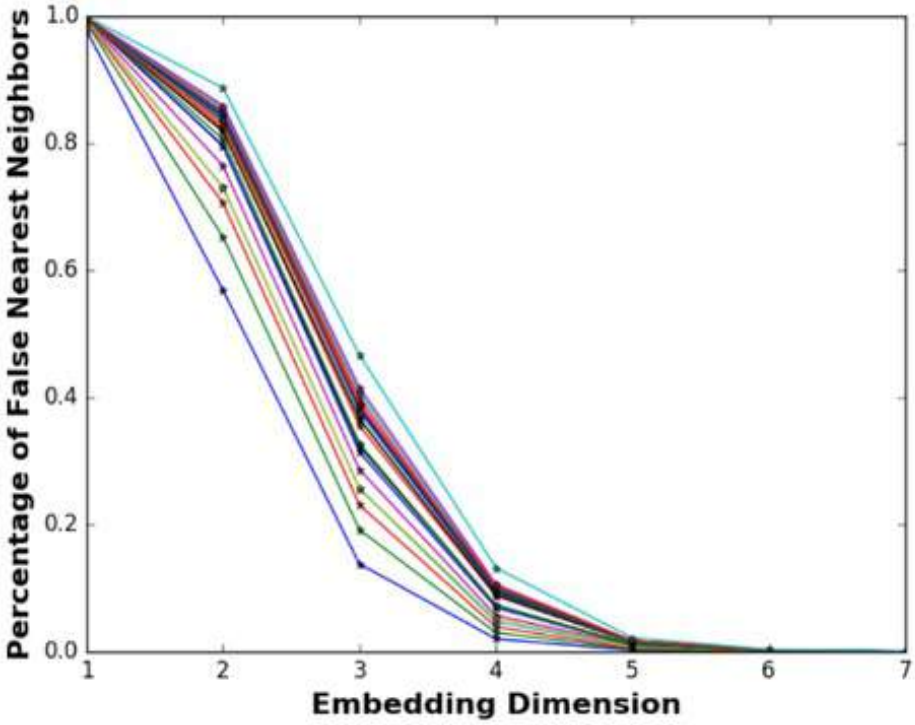
After we have set the parameters of each PS we want to approximate, we should proceed with the actual reconstruction of this attractor. We employ the Equation 2.22 and we use the previously defined parameters of time-lag τ and embedding dimensions d_e in order to represent each sample with a d_e-dimensional vector lying on the reconstructed manifold $\mathcal{M} \in \mathbf{R}^3$.

In Figure 4.5, a variety of different reconstructed phase space trajectories are displayed for various phonemes of speech. From this digram we can clearly see the reconstructed attractors as there are areas that the trajectories are asymptotically evolving around. Each attractor also indicates the recurrence underlying dynamics of each vowel that are actually very similar from a topological point of view but completely different from the perspective of dynamical systems. In other words some topological signatures such as the wholes of each attractor might remain the same between all 4 reconstructed phase spaces but these structures are quite dissimilar from the viewpoint of how the orbits are evolving through time. For example, in Figure 4.5b the states of the PS are evolving parallel to each other by creating a "vortex" at the center of the attractor which could also be captured by a projection on a two-dimensional plane. On the contrary, the reconstructed dynamics of Figures 4.5a and 4.5c have states which are evolving in parallel but also vertically displaying a different aspect of dynamics than the previous example. Finally, we can see that the reconstructed PS of Figure 4.5 has orbits that display no harmonic oscillations or evolution. This is an indicative measure of a noisy dynamical system or in the worst case scenario an unsuccessful unfolding of the dynamics of the underlying system.

In either case, it is evident that the recurrence properties of all these points lying on the recon-



(a) Selected time-lag τ based on the first local minimum of AMI



(b) Ad-hoc selection of time-lag $\tau = 1$

Figure 4.4: Percentiles of false nearest neighbors for various embedding dimensions and speech segments of different time-durations

structed manifold $\mathcal{M} \in \mathbf{R}^3$ are very indicative of the nature of each dynamical system and might be exploited for extracting nonlinear dependencies for speech signals in general and consequently for SER.

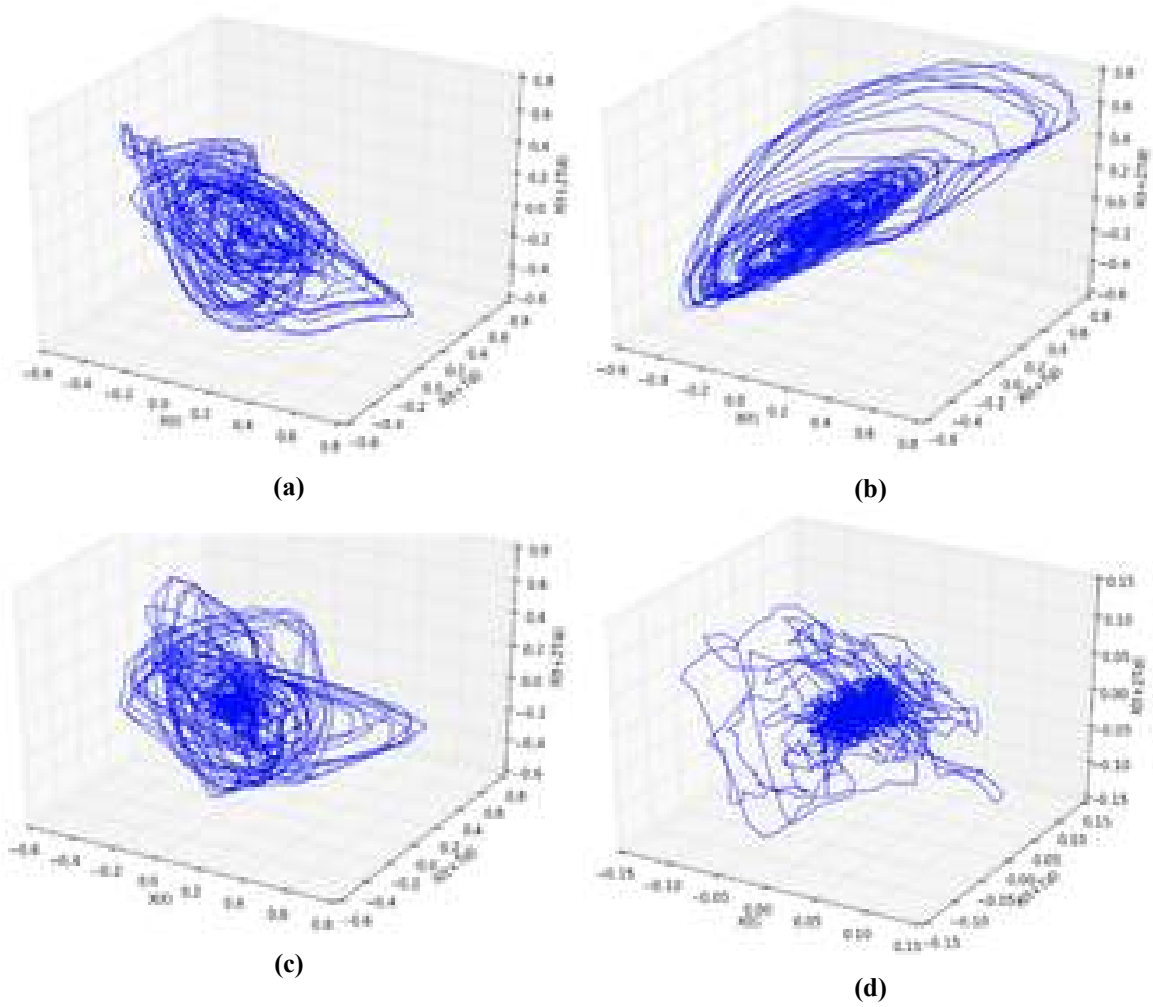


Figure 4.5: Reconstructed Phase Spaces of Speech Frames of Vowels

4.4 Recurrence Plots (RPs) from Speech Frames

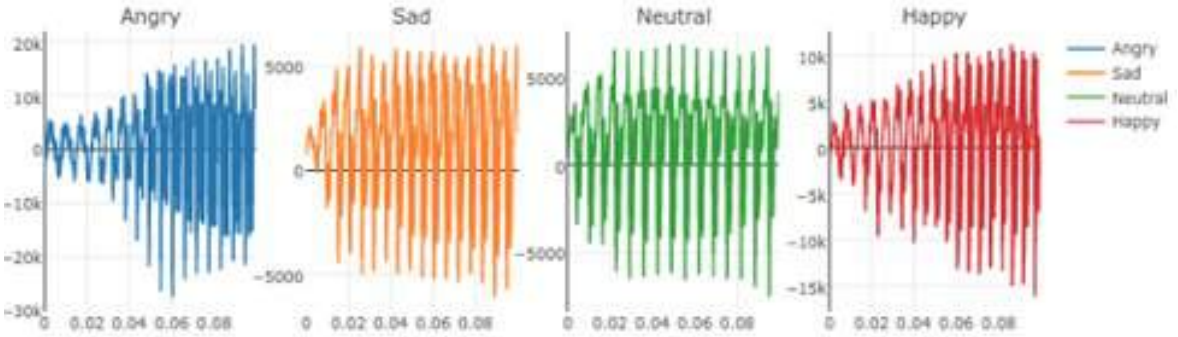
After we have gained an insight of how the reconstructed PS orbits would look like on the embedded space, we should consider ways for extracting useful information about them by exploiting the emerging recurrence structures. One of the most appropriate tools in order to visualize the recurrence properties that each PS exhibits are Recurrent Plots (RPs) [84]. In this short analysis of how RPs are extracted from local reconstructed PSs, we will analyze both unthresholded or continuous RPs and binary or thresholded RPs (see definition in Section 2.5).

In essence, the continuous diagrams are equal to the distance matrix under any specified norm of the PS (e.g., Euclidean, Supremum, Manhattan). The continuity is inferred from the fact that the latter diagrams are comprised of values which are continuous based on the selected metric. Specifically, we would like to normalize the output of these continuous RPs in range $[0, 1]$. In order to do so we can normalize each continuous RP by the maximum distance from all the pair-wise distances between any two trajectory points $\mathbf{x}(i)$ and $\mathbf{x}(j)$. Formally, we notate the continuous RP with **CR** and we define this symmetric matrix element-wise as follows:

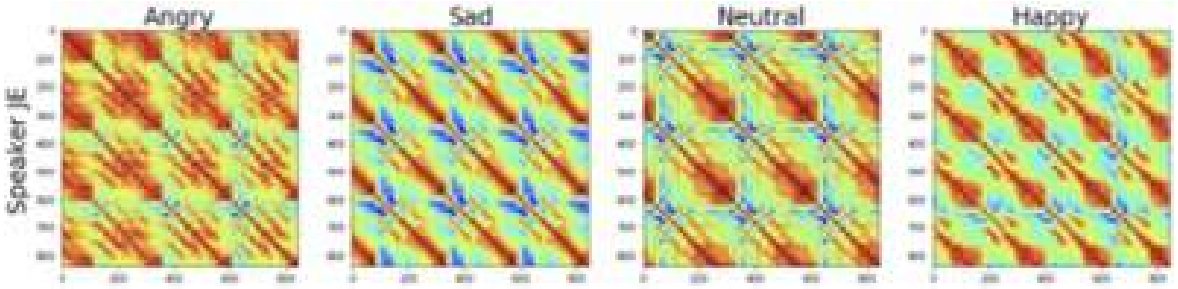
$$\mathbf{CR}_{i,j} = \frac{1 - \mathbf{D}_q(\mathbf{x}(i), \mathbf{x}(j))}{\underset{1 \leq \hat{i}, \hat{j} \leq N-(d_e-1)\tau}{max} \{\mathbf{D}_q(\mathbf{x}(\hat{i}), \mathbf{x}(\hat{j}))\}} = \frac{1 - ||\mathbf{x}(i) - \mathbf{x}(j)||_q}{\underset{1 \leq \hat{i}, \hat{j} \leq N-(d_e-1)\tau}{max} \{||\mathbf{x}(\hat{i}) - \mathbf{x}(\hat{j})||_q\}} \tag{4.2}$$

where $|| \cdot ||_q$ is the norm used to define the distance between any two trajectory points $\mathbf{x}(i)$ and $\mathbf{x}(j)$. Specifically, for $q = 1$, $q = 2$ or $q = \infty$ we compute Manhattan, Euclidean or Supremum

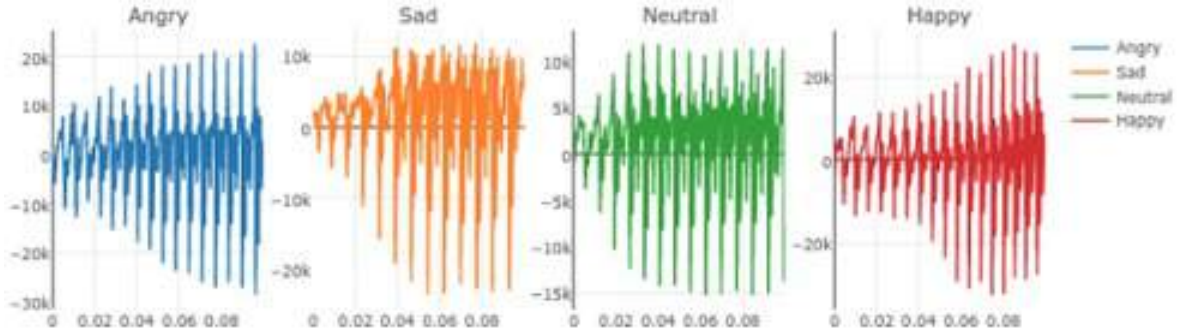
norm, respectively. If we would like to extract the resulting binary RPs from the previous continuous RPs we simply threshold these values depending on one of the aforementioned criteria for selecting the thresholding value ϵ (see Section 2.6 for a more extensive discussion about various methods for specifying the way of thresholding as well as the value of threshold).



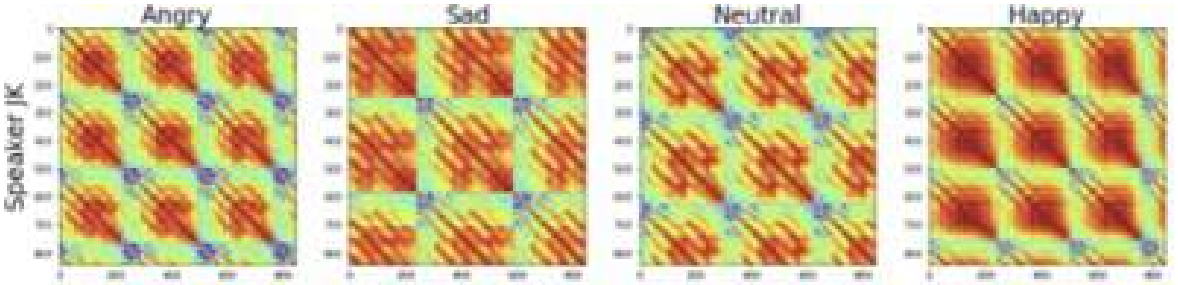
(a) /ae/ Phoneme of Speaker *JE* in time domain for each emotion type



(b) Continuous Recurrence Plot for /ae/ Phoneme of Speaker *JE* for each emotion type



(c) /ae/ Phoneme of Speaker *JK* in time domain for each emotion type



(d) Continuous Recurrence Plot for /ae/ Phoneme of Speaker *JK* for each emotion type

Figure 4.6: Vowel /ae/ in time domain and its corresponding Recurrence Plot for different speakers and their emotional manifestations

4.4.1 From Phonemes To Recurrence Plots

We focus on the visualization of the Continuous RPs for speech-frames timescales that correspond to phonemes and more specifically vowels that exhibit highly recurrence dynamics. In order to gain a more intuitive insight of how these speech signals are represented on the time-domain and on their continuous RPs images we need to compare them side by side.

In Figure 4.6, we compare the representations of /ae/ syllable for both time-domain and continuous RPs representations for two speakers of SAVEE dataset [139] for 4 different emotions, namely: *angry*, *sad*, *neutral* and *happy* which correspond to the respective columns of the grid of the images. It is evident that for both speakers the /ae/ representation in time-domain is pretty much alike. This also holds for the similarities of the time-domain representations between different emotions.

On the other hand, the continuous RPs representations of the speech signals is quite different for both speakers *JK* and *JE*. We notice some interesting patterns on the texture of the resulting images which are periodic with a pattern which is repeated across the diagonal line. It is limpid that the small subgraph of these images which is repeated across the main diagonal exhibits a spatial periodicity equal to the fundamental frequency of the voice (pitch). This is exemplary of the main problem of RP representations which remain strictly correlated with the fundamental frequencies existing on the speech signal. We can also see some more substructures which are displayed on these periodic subgraph images and display the existence of subharmonic oscillations of voice which is often called biphonation and is a highly nonlinear phenomenon on the vocal tract (see Section 1.4.2). However, if we focus on the representations of continuous RPs for speakers *JE* and *JK* individually, it is quite clear that the displayed structures are distinct among distinct emotions.

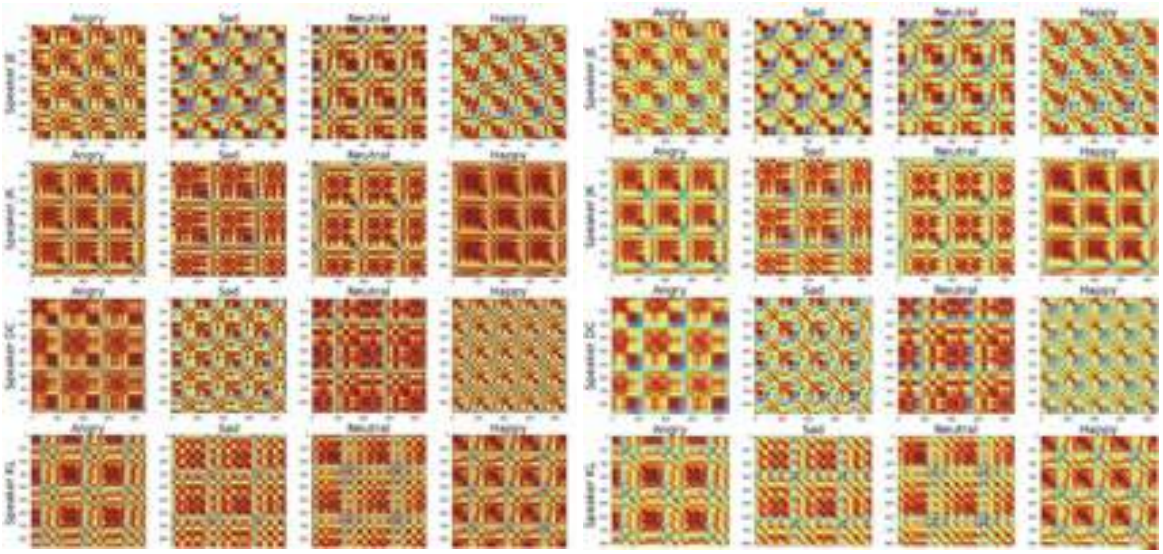
The displayed subgraph-image which is periodically repeated across the image has a structure which is strongly related with the identity of the speaker. Namely, for speaker *JE* we see that this substructure resembles a small fork while for the other speaker *JK* it is much more similar to a cactus. If we could draw some more general conclusions from these visualizations we could see that in general the *happy* and the *angry* RP representations for both speakers seem to be quite more dense than the other emotions for which /ae/ vowel is emitted. This could be indicative of the recurrence properties of how the vowel is phoned across different emotional conditions and consequently a key to build a better SER system by utilizing such information.

4.4.2 The Effectiveness of PS's Parameters for the Extraction of RPs

We should also analyze the efficacy of the parameters of the reconstructed PS which might also be reflected on the extraction of the corresponding RPs. Because the process is performed sequentially for a given signal we need to be careful of how the parameters of each processing block are creating misleading representations of the subsequent processing blocks. In a previous study [42], it was shown that the embedding dimension d_e is not a crucial parameter for the RP representations and especially for the binary ones. Presumably, this happens because the RP representations capture the recurrence properties of the time-series under study without even requiring the reconstruction of the PS. However, the unfolding of the dynamics of the time-series using higher dimensions disentangles the recurrence patterns which are displayed over RPs mainly because the periodic parts of the orbits would be displayed more correlated comparing to the uncorrelated parts of the orbits if the image is normalized. For instance, if we have d_e numbers which occur approximately with the same periodicity as the time-lag τ, which we have already selected for the reconstruction of the PS, then the corresponding entries of the continuous RP would be much closer to one than before. This amplifying of the edges of the image around entries with higher periodicity is further enhanced because other points which in general are not correlated with each other, the unfolding of the dynamics by the PS reconstruction pushes these differences to be much more evident. However, the time-lag parameter of τ samples between the time-lagged versions of the signal which are used to embed the dynamics play an important role in the final continuous RP representation.

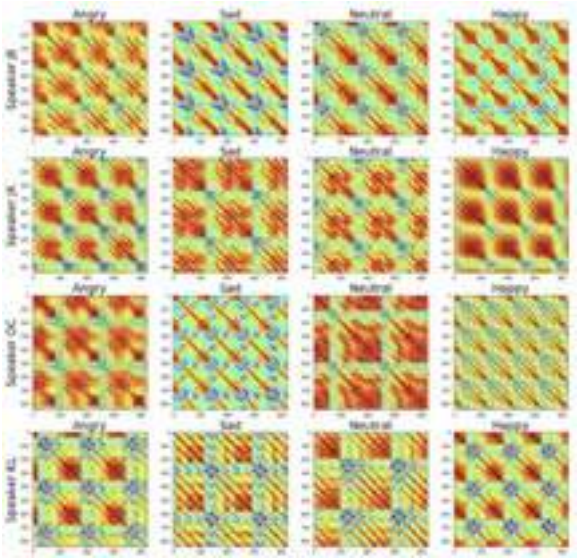
In Figure 4.7, we use all the speakers from SAVEE database [139] for the same utterance and we

segment the utterance in order to acquire the aligned frames for the excitation of vowel /ae/ under different emotional conditions. Again, the selected emotional labels are the same the ones we used in the previous analysis, namely: *angry*, *sad*, *neutral* and *happy*. All the continuous RP representations are extracted using the orbits of a 3-dimensional PS of the input frame corresponding to the vowel /ae/. If we do not use a time-lag (see Figure 4.7a) in order to embed the dynamics then the RP representations seem to be much more correlated with each other for all points even though we perform a normalization with the longest distance of these local pair-wise distances. Moreover, we see that as we increment the selected time-lag parameter tau we see that all the points are becoming much more uncorrelated until only the most correlated patterns are close to one (brown areas on Figure 4.7d) with all the others to be much closer to zero (blue areas on Figure 4.7d) than in the other Figures 4.7a, 4.7b and 4.7c. Presumably this is the indication that the true dynamics of the input signal still remain collapsed without the appropriate embedding. If we further increase the time-delay then we expect to see ones only in the of the main diagonal which is the opposite case. Hence, the time-lag parameter has a significant impact on how the resulting RP will look like.

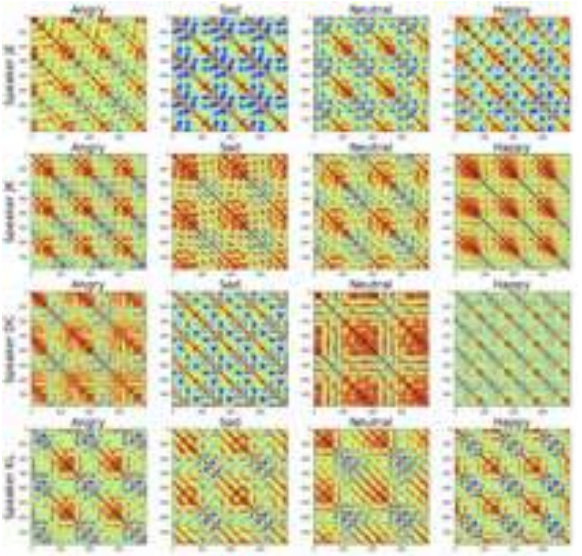


(a) Recurrence Plots using $\tau = 0$

(b) Recurrence Plots using $\tau = 7$



(c) Recurrence Plots using $\tau = 25$



(d) Recurrence Plots using $\tau = 50$

Figure 4.7: Extracting Continuous RPs for vowel /ae/ for all SAVEE speakers and their emotional manifestations

4.4.3 Emerging Chaotic Structures behind RPs of Phonemes?

As we have previously mentioned, the recurrence signatures of time-series are important for understanding the true underlying dynamics and identities of these signals. As we have previously mentioned in Section 2.5, these classes of systems could be divided to autoregressive, random noise, periodic and chaotic dynamics. So the question is how the overall process of RP extraction works and if we can actually acquire information from those recurrence patterns about the dynamics of the true system $\mathbf{s}^{\star}$?

In Figure 4.8, the overall process of extracting a binary RP is displayed as well as a comparison of two RPS which are extracted from different kind of systems. Specifically, we segment an *angry* emotional utterance and isolate a 30ms speech frame as shown in Figure 4.8a that is included inside the time-span of the excitation of vowel /e/. After that, we reconstruct the PS of the given frame as explained in previous Sections 2.5 and 4.3 by estimating the parameters of time-lag τ and embedding dimension d_e. In addition, we apply the equation of binary RPs (see Equation 2.36) by selecting an ad-hoc threshold value of $\epsilon = 0.15$ in order to produce the final binary image. This process until here presents a full visualization of the whole sequential process of the nonlinear feature extraction which is introduced in this Section.

We have also simulated a known system for its complex dynamics that it displays under different configurations and we created its binary RP representation by using the evolution through time of the states of the variables of the system. We have used Lorenz96 system [140] by using $N_{Lor} = 50$ variables for modeling it with an initial position of zero and a force of $F = 5$ units which is applied at a specified position when the system is in equilibrium. If we let the system dynamics evolve through time we expect to see a chaotic behavior after a few units of discrete time. The dynamical model of Lorenz96 is formulated using the equations derived from the following equation:

$$\frac{dz(i)}{dt} = [z(i+1) - z(i-2)]z(i-1) - z(i) + F \tag{4.3}$$

If we want to simulate in a computer the above equation, we have to convert the time derivative to discrete difference of consecutive by just inserting an index about the number of time units that each variable exists. Namely, if we are at the tth evolution of the system the vector of variables would be represented as $\mathbf{z}_t$. Moreover, the discretized equation would be similarly defined by the following expression:

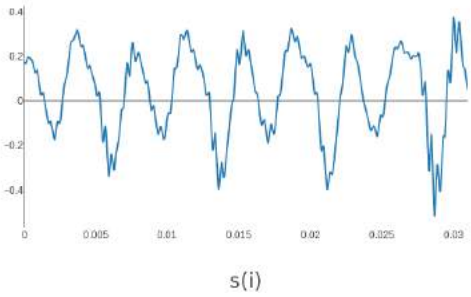
$$z_t(i) = [z_{t-1}(i+1) - z_{t-1}(i-2)]z_{t-1}(i-1) + F \tag{4.4}$$

where the following initial conditions are preserved

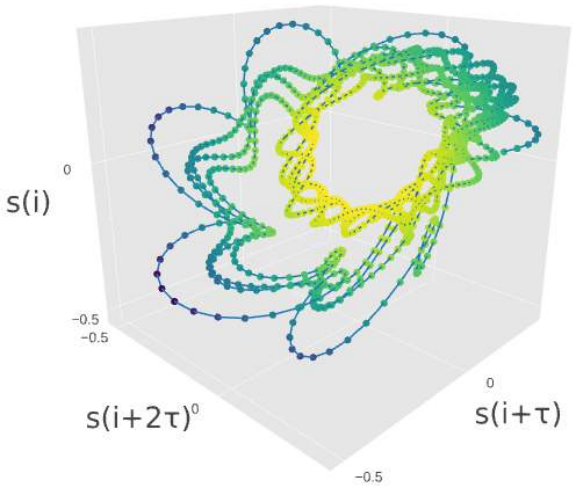
$$z(i) = z(i \mod N_{Lor}), \quad i \in \mathbb{N} \cap [0, N_{Lor}] \tag{4.5}$$

where $a \mod b$ defines the reminder from the integer division between the numbers a and b. Finally, the corresponding binary RP with an ad-hoc threshold $\epsilon = 0.1$ is displayed in Figure 4.8d. As it has been shown the recurrence properties of this binary image displays small intermittent diagonals parallel to the main diagonal which is indicative of the existence of chaotic regimes that the Lorenz96 system explores [140].

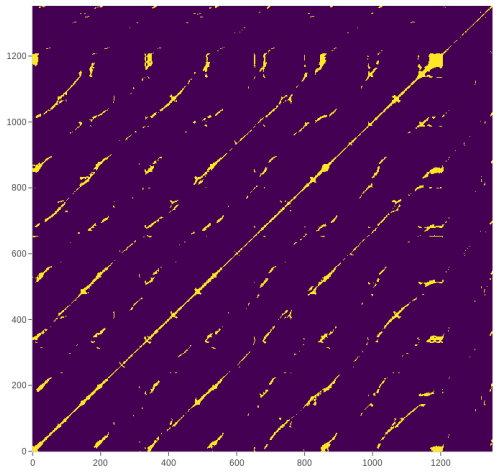
All in all, emerging small-scale structures based on lines of ones or zeros reflect the dynamic behavior of the system. For instance, diagonal lines indicate both similar evolution of states for different parts of PS's orbit and deterministic chaotic dynamics of the system [84]. This is also depicted in Figures 4.8c and 4.8d when comparing them side by side. It is evident that both systems have diagonals parallel to the main diagonal with a fundamental frequency which is prominent for both systems. In addition, these intermittent diagonal lines in both systems are indications of highly nonlinear phenomena in the oscillations of both systems. Thus, RP representations reveal common aspects for the dynamics of different systems using a similar display on the binary image. In other words, different dynamical systems display similar behavior in terms of dynamics and recurrence properties which



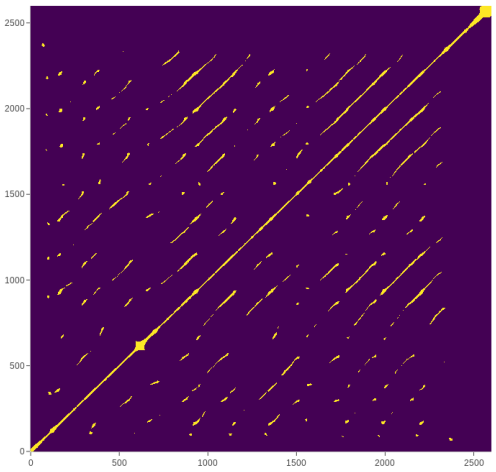
(a) Time-domain representation of the speech frame



(b) Reconstructed Phase Space ($m = 3$, $\tau = 7$)



(c) Binary Recurrence Plot using a threshold of $\epsilon = 0.15$ and a Manhattan norm for measuring the distance between points on the PS



(d) Binary Recurrence Plot of Lorenz96 system displaying chaotic behavior

Figure 4.8: Extracting the RP of a 30ms frame contained in the excitation of vowel /e/ and understanding the underlying dynamics by comparison

they are displayed aptly on RP representations. In this case, the production of the human speech under an "angry" emotional stimulation seems to resemble the dynamics of the theoretical Lorenz96 system which displays chaotic nature in its dynamics.

4.5 Acoustic Feature Extraction

Likewise in the previous Chapter 3 we use acoustic feature sets in order to perform the final SER. We extract the acoustic features for two methods of classification, namely: utterance and segment-level. Both classification methods are identical to the ones we described in previous Sections 3.3.2 and 3.3.3 in terms of the feature extraction but in this setup we use different classification models. We introduce the new nonlinear feature set (RQA feature set) based on RQA measures which is described in Section 4.5.2. As a baseline feature set we extract the same features as described in Section 3.2.4 and we call

it "IS10" feature set [23]. Finally, the fused feature set is described in Section 4.5.3 and includes the features from both the previously aforementioned feature sets.

4.5.1 Baseline Feature Set (IS10 Set)

We use the IS10 feature set [23], in which 1582 features are extracted corresponding to statistical functionals applied on various LLDs. The extraction is performed for both segment and utterance based approaches using openSMILE toolkit [119]. A further analysis on this feature set is provided in a previous Section 3.2.4.

4.5.2 Proposed Nonlinear Feature Set (RQA Set)

The proposed RQA feature set for a given speech segment or utterance is extracted as described next.

First, we break the given speech signal into overlapping frames with an overlap ratio of $OL = 0.5$. For each frame we reconstruct its PS trajectory as defined in Equation 2.22. We select for the reconstruction of each PS the time-lag parameter τ and the embedding dimension d_e as described in Sections 2.4.2 and 2.4.4, respectively.

Next, for each PS orbit, its respective RP is computed as a binary image from the distances between the points of the trajectory as explained in Section 2.5. In order to select the threshold value ϵ in order to create its binary image, we use one of the following criteria based on: 1) a fixed ad-hoc threshold value, 2) a fixed Recurrence Rate (RR) or 3) a fixed ratio of the standard deviation σ of points on the PS trajectory (a more lengthy explanation of the criteria has been given in Section 2.5 but can be also found in the literature [84]).

In order to quantify the complex structures of the RP, a list of RQA measures is extracted in order to create a static-length RQA representation for each frame. The selected RQA measures are presented in Table 4.1 while their corresponding qualitative analysis of these measures ahas already been given in Section 2.6.1. Now each speech-frame is represented by a 12-dimensional feature vector corresponding to the values of the aforementioned RQA measures.

Representations for speech-segments and utterances containing multiple frames are obtained by applying a set of 18 statistical functionals over all 12-dimensional frame-attributes from the included speech frames as well as their deltas. The selected 18 statistical functionals which are used are presented in Table 4.2. Thus, a $432 = 18 \cdot 24$ feature-vector is obtained for each speech-segment or utterance of emotional speech.

4.5.3 Fused Feature Set (RQA + IS10 Feature Set)

Because the two feature sets (RQA, IS10) are extracted for any given duration of a speech-frame or utterance they can be combined using a simple concatenation of the two statistical representations. Namely, when we refer to the fused feature set (RQA + IS10 Feature Set) we would refer to a 2014-dimensional feature vector which has been created by the aligned concatenation of the IS10 (1582 features) and RQA (432 features) feature vectors.

4.6 Classification Methods

In order to assess the performance of the proposed RQA and IS10 feature sets as well as their fusion, we experiment on approaches on various time-scales. Specifically, we investigate both utterance-based and segment-based SER as outlined below.

4.6.1 Utterance-Based

For each utterance we obtain its statistical representation by extracting the corresponding feature set as described in Section 4.5. For the final emotion classification we employ an SVM with Radial

Table 4.1: RQA Measures extracted from each RP

Name	Formulation
Recurrence Rate	$\frac{1}{N^2}\sum_{i,j=1}^{N}\mathbf{R}_{i,j}$
Determinism	$\frac{\sum_{l=d_m}^{N} lP_d(l)}{\sum_{l=d_m}^{N} lP_d(l)}$
Max Diagonal Length	$max(\{l_i\}_{i=1}^{N_d})$
Average Diagonal Length	$\frac{\sum_{l=d_m}^{N} lP_d(l)}{\sum_{l=d_m}^{N} P_d(l)}$
Diagonal Entropy	$\sum_{l=d_m}^{N}\frac{P_d(l)}{N_d}ln(\frac{N_d}{P_d(l)})$
Laminarity	$\frac{\sum_{l=v_m}^{N} lP_v(l)}{\sum_{l=v_m}^{N} lP_v(l)}$
Max Vertical Length	$max(\{l_i\}_{i=1}^{N_v})$
Trapping Time	$\frac{\sum_{l=v_m}^{N} lP_v(l)}{\sum_{l=v_m}^{N} P_v(l)}$
Vertical Entropy	$\sum_{l=v_m}^{N}\frac{P_v(l)}{N_v}ln(\frac{N_v}{P_v(l)})$
Max White Vertical Length	$max(\{l_i\}_{i=1}^{N_w})$
Average White Vertical Length	$\frac{\sum_{l=w_m}^{N} lP_w(l)}{\sum_{l=w_m}^{N} P_w(l)}$
White Vertical Entropy	$\sum_{l=w_m}^{N}\frac{P_w(l)}{N_w}ln(\frac{N_w}{P_w(l)})$

Table 4.2: Sets of Statistical Functionals for RQA Feature Set

Statistical Functions
min
max
arithmetic mean
median
variance
skewness
kurtosis
range
1_{st}, 5_{th}, 25_{th}, 50_{th}, 75_{th}, 95_{th}, 99_{th} percentiles
$25-50$, $50-75$ and $25-75$ quartile ranges

Base Function (RBF) kernel and one-versus-rest LR classifier. An extensive analysis as well as the mathematical formulation of both of these models can be found in Section 2.2.2 for the SVM and in Section 2.2.3 for the LR classifier. Cost coefficient C lies in the interval $[0.001, 30]$ for both SVM and LR models which is the only hyper-parameter to be tuned. Both models are implemented using scikit framework [141].

4.6.2 Segment-Based

We break each utterance in segments of 1.0 seconds duration and 0.5 seconds stride, in accordance with the segment-based approach of the previous Section 3.3.2. For each speech segment we extract the feature sets described in Section 4.5 and as a result each utterance is now represented by a sequence of statistical vectors corresponding to different time steps. This sequence is fed as an input to a Long Short Time Memory (LSTM) unit for emotion classification. An extensive description of an LSTM

model can be found in 2.3.1. SER can be formulated as a many-to-one sequence learning where the expected output of each sequence of segment features is an emotional label derived from the activations of the last hidden layer [66].

Specifically, we employ an Attention-based Bidirectional LSTM (A-BLSTM) architecture [65] where the decision for the emotional label is derived from a weighted aggregation of all timesteps. An extensive analysis of this model can be found in Sections 2.3.4. Furthermore, the network topology of a Bidirectional LSTM is described in Section 2.3.2 as well as how the attention mechanism works when applied on top of an RNN architecture which has been analyzed in Section 2.3.3. We implement this architecture in pytorch [142]. In addition, the grid space of hyper-parameters consists of the following parameters:

- number of layers $\{1, 2\}$
- number of hidden nodes $\{128, 256\}$
- input noise $[0.3, 0.8]$
- dropout rate $[0.3, 0.8]$
- learning rate $[0.0002, 0.002]$.

4.7 Experiments

We evaluate our proposed feature set under three different SER tasks described next. We also compare our results with the most relevant experimental setups reported in literature. For all tasks, we report: Weighted Accuracy (WA) and Unweighted Accuracy (UA) which are describe in the previous Section 3.4.2.

We test a variety of frame durations $\{20, 30, 50\}$ ms from which RPs are computed. For the configuration of the RP we explore the sets of parameters detailed in Section 2.5. Specifically, we test Manhattan, Euclidean and Supremum norms as well as multiple selection criteria for the threshold value ϵ, depending on ad-hoc threshold setting, fixed recurrence rate and fixed σ. The respective ratio parameter for the three aforementioned criteria lies in $[0.05, 0.5]$.

After an extensive study of the RQA Feature set configuration parameters, we conclude that best results on SER tasks are obtained using a frame duration of 20 ms for extracting RPs. In addition, the best performing parameters for the RP configuration seem to be a Manhattan norm with a threshold setting depending on a fixed recurrence rate lying in $[0.1, 0.2]$.

4.7.1 Datasets

The following databases are used in our experiments:

1. **SAVEE:** Surrey Audio-Visual Expressed Emotion (SAVEE) Database [139] is composed of emotional speech voiced by 4 male actors. SAVEE includes 480 utterances (120 utterances per actor) of 7 emotions i.e., 60 anger, 60 disgust, 60 fear, 60 happiness, 60 sadness, 60 surprise and 120 neutral.

2. **Emo-DB:** Berlin Database of Emotional Speech (Emo-DB) [121] contains 535 emotional sentences in German, voiced by 10 actors (5 male and 5 female). Specifically, 7 emotions are included i.e., 127 anger, 45 disgust, 70 fear, 71 joy, 60 sadness, 81 boredom and 70 neutral.

3. **IEMOCAP:** IEMOCAP database [68] contains 12 hours of video data of scripted and improvised dialog recorded from 10 actors. Utterances are organized in 5 sessions of dyadic interactions between 2 actors. For our experiments we consider 5531 utterances including 4 emotions (1103 angry, 1636 happy, 1708 neutral and 1084 sad), where we merge excitement and happiness class into the latter one [27], [28], [35], [36].

4.7.2 Speaker Dependent (SD)

We evaluate RQA features on SAVEE and Emo-DB following the utterance-based approach described in Section 4.6.1. In this setup we apply per-speaker z-normalization (PS-N) and split randomly utterances in train and test sets. Accuracies using 5-fold cross-validation are summarized on Table 4.3 for the best performing classifier hyper-parameter values.

The fused set achieves significant performance improvement over the baseline IS10 feature set for both datasets. On SAVEE, WA is improved by 3.1% (77.1% $\rightarrow$ 80.2%) and UA by 3.4% (74.5% $\rightarrow$ 77.9%). We also achieve an improvement of 4.9% (88.4% $\rightarrow$ 93.3%) and 5.7% (87.2% $\rightarrow$ 92.9%) for WA and UA, respectively on Emo-DB. The feature set used in [49] is extracted over cepstral, spectral and prosodical LLDs similar to the ones used in IS10 [23]. Noticeably, they achieve similar performance to ours when we use only IS10 but our fused set with LR outperforms on both Emo-DB (5% in UA and 4.6% in WA) and SAVEE (4.5% in UA and 3.9% in WA). The proposed combination of features and LR also surpasses a Convolutional SAE approach [37] in terms of WA by 5% on Emo-DB and 4.8% on SAVEE. Presumably, RQA measures contain information closely related to speaker-specific emotional dynamics not captured by conventional features.

Table 4.3: SD results on SAVEE and Emo-DB. (ESR) Ensemble Softmax Regression

Features	Model	SAVEE		Emo-DB	
		WA	UA	WA	UA
IS10	SVM	77.1	74.5	88.4	87.2
	LR	74.4	71.8	87.4	86.3
RQA	SVM	66.0	63.0	81.8	80.4
	LR	64.4	61.1	81.9	79.9
RQA+IS10	SVM	77.3	75.5	90.1	88.9
	LR	**80.2**	**77.9**	**93.3**	**92.9**
[37] Spectrogram	SAE	75.4	-	88.3	-
[49] LLDs Stats	ESR	76.3	73.4	88.7	87.9

4.7.3 Speaker Independent (SI)

Again, we follow the utterance-based approach described in Section 4.6.1 on both SAVEE and Emo-DB datasets but we do not make any assumptions for the identity of the user during training. We use leave-one-speaker-out cross validation, where one speaker is kept for testing and the rest for training. The mean and standard deviation are calculated only on training data and used for z-normalization on all data. From now on we refer to this normalization as Per Fold-Normalization (PF-N). Table 4.4 presents accuracies averaged over all folds for the best performing classifier hyper-parameter values.

In comparison with the baseline IS10 feature set, the fused feature set obtains an absolute improvement of 5.5% and 8.2% on SAVEE as well as 2.4% and 3.2% on Emo-DB in terms of WA and UA, respectively. Furthermore, our fused set achieves higher performance on SAVEE (3.5% in WA and 4.5% in UA) and slightly lower in Emo-DB compared to [49]. In [143] Weighted Spectral Features based on Hu Moments (WSFHM) are fused with IS10 on utterance-level which is similar to our approach. In direct comparison using the same model (SVM) we surpass the reported performance in terms of WA by 2.5% and 0.4% on SAVEE and Emo-DB, respectively. In addition, both RQA and IS10 sets achieve quite low performance on SAVEE. However, their combination yields an impressive performance improvement of 5.5% (48.5% $\rightarrow$ 54.0%) in WA and 10.7% (43.1% $\rightarrow$ 53.8%) in UA over IS10 when we use LR. Our results suggest that RQA measures preserve invariant aspects of nonlinear dynamics occurring in emotional speech and are shared across different speakers.

Table 4.4: SI results on SAVEE and Emo-DB. (ESR) Ensemble Softmax Regression

Features	Model	SAVEE		Emo-DB	
		WA	UA	WA	UA
IS10	SVM	47.5	45.6	79.7	74.3
	LR	48.5	43.1	76.1	71.9
RQA	SVM	45.6	41.1	70.9	64.2
	LR	47.7	42.3	71.1	67.1
RQA+IS10	SVM	52.5	50.6	82.1	76.9
	LR	**54.0**	**53.8**	80.1	77.5
[49] LLDs Stats	ESR	51.5	49.3	**82.4**	**78.7**
[143] WSFHM+IS10	SVM	50.0	-	81.7	-

4.7.4 Leave One Session Out

In this task, we assume that the test-speaker identity is unknown but we are able to train our model considering other speakers who are recorded in similar conditions. We evaluate on both utterance and segment-based methods (described in Section 4.6.2) on IEMOCAP. Given our assumption, we treat each of the 5 sessions as a speaker group [68]. We use LOSO in order to create train and test folds. In each fold, we use 4 sessions for training and the remaining 1 for testing. For the testing session we use one speaker as testing set and the other for tuning the hyper-parameters of our models. We repeat the evaluation by reversing the roles of the two speakers. In the final assessment, we report the average performance obtained in terms of WA and UA obtained from all speakers [27], [28], [36]. In order to be easily comparable with the literature we follow three different normalization schemes. We use the aforementioned PS-N and PF-N schemes as well as Global z-normalization (G-N). In G-N we calculate the global mean and standard deviation from all the available samples in the dataset and perform z-normalization over them. Results on IEMOCAP for the three different normalization schemes are demonstrated on Table 4.5.

A consistent performance improvement is shown for all combinations of normalization techniques and employed models when the fused set is used instead of IS10. Specifically, for SVM the fused set yields a relative improvement varying from 0.3% to 1.0% in WA and from 0.2% to 0.9% in UA under all normalization strategies. The same applies for LR (in WA from 0.8% to 1.0% and in UA from 0.3% to 1.0%) as well as for A-BLSTM (in WA from 0.1% to 0.7% and in UA from 0.2% to 0.7%). In accordance with our intuition [2], a segment-based approach using A-BLSTM surpasses all utterance-based ones in WA from 3.4% to 8.4% and in UA from 3.8% to 6.8% for all normalization schemes, when the fused set is used. This is really important as the introduced RQA set provides a resilient feature representation of emotional utterances on a variety of time-scale approaches.

In [27] low level Mel Filterbank (MFB) features are fed directly to a CNN. In [36] a stacked autoencoder is used to extract feature representations from spectrograms of glottal flow signals and then a BLSTM is used for classification. We surpass both reported results by 0.2% in UA for [27] and by a margin of 8.7% in WA and 8.5% in UA for [36], respectively even with simple models. Compared to a multi-task DBN trained for both discrete emotion classification and for valence-activation in [28], we report 2.0% higher WA and 3.1% higher UA. We also report 4.6% higher UA and 1.9% lower WA compared to CNNs over spectrograms [35]. We assume that this inconsistency in performance metrics occurs because a slightly different experimental setup is followed where the final session is excluded from testing [35].

Table 4.5: LOSO results on IEMOCAP. (GFS): Glottal Flow Spectrogram, (SP): Spectrogram.

Features	Model	PS-N		PF-N		G-N	
		WA	UA	WA	UA	WA	UA
IS10	SVM	58.3	60.9	58.9	60.1	59.2	60.5
	LR	57.5	61.2	54.6	57.9	53.5	57.5
	A-BLSTM	62.0	65.1	62.6	65.0	62.8	65.0
RQA	SVM	52.9	54.6	53.1	53.8	53.1	53.7
	LR	52.2	54.8	52.6	54.0	52.8	54.3
	A-BLSTM	55.6	59.3	56.6	58.3	56.7	58.7
RQA + IS10	SVM	59.3	61.8	59.2	60.4	59.5	60.7
	LR	58.3	62.0	55.6	58.7	54.5	58.7
	A-BLSTM	**62.7**	**65.8**	**63.0**	**65.2**	62.9	**65.5**
[27] MFB	CNN	-	61.8	-	-	-	-
[28] IS10	DBN	-	-	-	-	60.9	62.4
[35] SP	CNN	-	-	-	-	**64.8**	60.9
[36] GFS	BLSTM	-	-	50.5	51.9	-	-

Chapter 5

Pattern Search Multidimensional Scaling

This chapter is an extended version of the paper [3] which can be found online at https://arxiv.org/abs/1806.00416. If the reader needs to cite parts of this chapter then it would be preferred to use the following reference (or the most updated one under the same title and authors specified):

- Giorgos Paraskevopoulos †, *Efthymios Tzinis †, Emmanuel-Vasileios Vlatakis-Gkaragkounis, and Alexandros Potamianos, "Pattern Search Multidimensional Scaling," arXiv:1806.00416, 2018.*
  †Both authors contributed equally in this work

5.1 Motivation

As we have previously discussed in Section 1.4, the features we use for all classification problems could lead to huge vector representations which makes it extremely challenging to train our models in terms of the time and memory which is required for the overall process. The same applies to our problem which is emotional recognition. In previous Sections 4.5.1, 4.5.2 and 4.5.3 we have seen that each speech-segment or utterance uses vectors lying in $\mathbb{R}^{1582}$ (IS10 Feature set), $\mathbb{R}^{432}$ (RQA Feature set) and $\mathbb{R}^{2014}$ (RQA + IS10 Feature set), respectively. In this context, we seek to find low-dimensional representations which are able to preserve the initial pair-wise distances of the high-dimensional representations. In this way, we seek to approximate a low-dimensional manifold $\mathcal{M} \in \mathbb{R}^L$ where $L < 20$ and we are still able to get the same results in terms of accuracy performance. In order to aptly describe this low-dimensional representation which can capture the pair-wise correlations between the high-dimensional vectors we need to also integrate the nonlinear dependencies of the input feature space. This problem is known as non-metric multidimensional scaling (MDS) or nonlinear dimensionality reduction (NLDR) task. The majority of these nonlinear dimensionality reduction algorithms are actually minimizing a loss function f. Given this minimization objective, they usually employ gradient-based methods to find a global or a local optimum. In many situations, however, the loss function is non-differentiable or estimating its gradient may be computational expensive. Additionally, gradient-based algorithms usually yield a slow convergence; multiple iterations are needed in order to minimize the loss function. Inspired by the recent progress in derivative-free optimization tools, we propose an iterative algorithm which treats the non-metric MDS task as a derivative-free optimization problem.

5.2 Related Work

Some dimensionality reduction algorithms suppose that the low-dimensional representations of data can be obtained using linear dimensionality reduction techniques like Principle Components Analysis (PCA) [144] but this is not true when we deal with data for real cases such as SER. In real case scenarios, such a linearity assumption may be too strong and can lead to misleading representations of the input data. In this context, a significant progress has been made towards manifold learning algorithm for taking into account the nonlinear structure of the input data.

Representative manifold learning algorithms include Isometric Feature Mapping (ISOMAP) [145, 146, 147, 148, 149], Landmark ISOMAP [150, 151],Locally Linear Embedding (LLE) [152, 153,

154, 155, 156], Modified LLE [157] Hessian LLE [158, 159], Semidefinite Embedding [160], [161], [162], [163], Laplacian Eigenmaps (LE) [164, 152, 165], Local Tangent Space Alignment (LTSA) [166], Spectral Clustering [167], t-SNE [88] etc. ISOMAP uses a geodesic distance to measure the geometric information within a manifold. LLE assumes that a manifold can be approximated in a Euclidean space and the reconstruction coefficients of neighbors can be preserved in the low-dimensional space. LE uses an undirected weighted graph to preserve local neighbor relationships. Hessian LLE obtains low-dimensional representations by applying eigenanalysis on a Hessian coefficient matrix. LTSA utilizes local tangent information to represent the manifold geometry and extends this to global coordinates. Finally, SDE attempts to maximize the distance between points that do not belong in a local neighborhood. In addition, a common nonlinear method for dimensionality reduction is the kernel extension of PCA [168] which is also similar to the Spectral Clustering in which a Gaussian Kernel element-wise is used and after that a K-means algorithm is applied in order to divide data into clusters. Finally, t-SNE is well suited for the visualization of the learned manifolds (for $2D$ and $3D$ spaces) from high-dimensional data by iteratively minimizing a non-convex objective function through gradient descent.

A wide class of derivative-free algorithms for nonlinear optimization has been studied and analyzed in [169] and [170]. GPS methods consist a subset of the aforementioned algorithms which do not require the explicit computation of the gradient in each iteration-step. Some GPS algorithms are: the original Hooke and Jeeves pattern search algorithm [171], the evolutionary operation by utilizing factorial design [172] and the multi-directional search algorithm [173], [174]. In [113], a unified theoretical formulation of GPS algorithms under a common notation model has been presented as well as an extensive analysis of their global convergence properties. Local convergence properties have been studied later by [114]. Notably, the theoretical framework as well as the convergence properties of GPS methods have been extended in cases with linear constrains [117], boundary constrains [115] and general Lagrangian formulation [116].

5.3 Algorithm Description

5.3.1 Formulation

The key idea behind the proposed algorithm is to treat MDS as a derivative-free problem, using a variant of general pattern search optimization to minimize a loss function. The input to pattern search MDS is a $N \times N$ target dissimilarity matrix $\mathbf{T}$ and the target dimension L of the embedding space. An overview of the algorithm shown in Algorithm 2 is presented next.

The initialization process of the algorithm consists of: 1) random sampling of N points in the embedded space and construction of the matrix $\mathbf{X}^{(0)} = [\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, ..., \mathbf{x}_N^{(0)}] \in \mathbb{R}^{N \times L}$, 2) computing the embedded space dissimilarity matrix $\mathbf{D}^{(0)}$, where the element $d_{ij}^{(0)}$ is the Euclidean distance between vectors $\mathbf{x}_i^{(0)}$ and $\mathbf{x}_j^{(0)}$ of $\mathbf{X}^{(0)}$, and 3) computing the initial approximation error $e^{(0)} = f(\mathbf{T}, \mathbf{D}^{(0)})$, where e is the element-wise mean squared error (MSE) between the two matrices. The functional f that we attempt to minimize is the normalized square of the Frobenius norm of the matrix $\mathbf{T} - \mathbf{D}$, i.e., $f(\mathbf{T}, \mathbf{D}) = (1/N^2)||\mathbf{T} - \mathbf{D}||_F^2$. Equivalently one may express f element-wise as follows:

$$f(\mathbf{T}, \mathbf{D}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (t_{ij} - d_{ij})^2, \quad \text{where } \mathbf{T}, \mathbf{D} \in \mathbb{R}^{N \times N} \tag{5.1}$$

Algorithm 2 Pattern Search MDS

procedure MDS($\mathbf{T}$, L, $r^{(0)}$)
  $k \leftarrow 0$ ▷ k is the number of epochs
  $\mathbf{X}^{(k)} \leftarrow$ UNIFORM($N \times L$)
  $\mathbf{D}^{(k)} \leftarrow$ DISTANCE_MATRIX($\mathbf{X}^{(k)}$)
  $e^{(k)} \leftarrow f(\mathbf{T}, \mathbf{D}^{(k)})$
  $e^{(k-1)} \leftarrow +\infty$
  $r^{(k)} \leftarrow r^{(0)}$
  while $r^{(k)} > \delta$ **do**
    if $e^{(k-1)} - e^{(k)} \leq \epsilon \cdot e^{(k)}$ **then**
      $r^{(k)} \leftarrow \frac{r^{(k)}}{2}$
    $\mathbf{S} \leftarrow$ SEARCH_DIRECTIONS($r^{(k)}$, L)
    for all $x \in \mathbf{X}^{(k)}$ **do**
      $\mathbf{X}^*, e^* \leftarrow$ OPTIMAL_MOVE($\mathbf{X}^{(k)}$,x,$\mathbf{S}$,$e^{(k)}$)
      $e^{(k-1)} \leftarrow e^{(k)}$
      $e^{(k)} \leftarrow e^*$
      $\mathbf{X}^{(k)} \leftarrow \mathbf{X}^*$
    $k = k + 1$

5.3.2 Search Directions

Following the initialization steps, in each epoch (iteration), we consider the surface of a hypersphere of radius r around each point $\mathbf{x}_i^{(k)}$. The possible search directions lie on the surface of a hypersphere along the orthogonal basis of the space, e.g., in the case of 3-dimensional space along the directions $\pm x, \pm y, \pm z$ on the sphere shown in Figure 5.1. This creates the search directions matrix S and is summarized in Algorithm 3

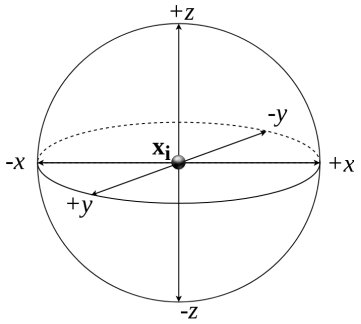


Figure 5.1: Sphere of radius r around point $\mathbf{x}_\mathbf{i}^{(\mathbf{k})}$ and possible search directions

Algorithm 3 Define search directions

function SEARCH_DIRECTIONS(r, L)
  $\mathbf{S}^+ \leftarrow r \cdot \mathbf{I}_L$
  $\mathbf{S}^- \leftarrow -r \cdot \mathbf{I}_L$
  $\mathbf{S} \leftarrow [\frac{\mathbf{S}^+}{\mathbf{S}^-}]$
  return S

5.3.3 Move Alongside the Optimal Direction

Each point is moved greedily along the dimension that produces the minimum error. At this stage, we only consider moves that yield a monotonic decrease in the error function. Algorithm 4 finds the

optimal move that minimizes $e^{(k)} = f(\mathbf{T}, \mathbf{D}^{(k)})$ for each new point $\tilde{x}$ and moves $\mathbf{X}$ in that direction. Note that when writing $s \in \mathbf{S}$, the matrix $\mathbf{S}$ is considered to be a set of row vectors.

Algorithm 4 Find optimal move for a point

```
1:  function OPTIMAL_MOVE(X^(k), x, S, e)
2:      e* ← e
3:      for all s ∈ S do
4:          x̃ ← x + s
5:          X̃ ← UPDATE_POINT(X^(k), x, x̃)        ▷ Update x point of X^(k) with x̃
6:          D ← DISTANCE_MATRIX(X̃)
7:          ẽ ← f(T, D)
8:          if ẽ < e* then
9:              e* ← ẽ
10:             X* ← X̃
11:     return X*, e*
```

5.3.4 Computation of the Error

The resulting error e^* is computed after performing the optimal move for each point in $\mathbf{X}^{(k)}$. If the error decrease hits a plateau, we halve the search radius and proceed to the next epoch. This is expressed as shown next:

$$e^{(k)} - e^* < \epsilon \cdot e^{(k)} \tag{5.2}$$

where ϵ is a small positive constant, namely the error decrease becomes very small in relation to $e^{(k)}$. The process stops when the search radius r becomes very small, namely $r < \delta$, where δ is a small constant, as shown in Algorithm 2.

5.3.5 Complexity

The complexity of the core algorithm is $\mathcal{O}(N^2 L)$. This is explained as: for each epoch we search across $2L$ dimensions for N points. In each search we also need $\mathcal{O}(N)$ operations to update the distance matrix as we move all points independently for each epoch.

5.4 Approximations and Optimizations

Next, a set of algorithmic optimizations are presented which improve the execution time or the solution quality of Algorithm 2. Noticeably, in some cases both optimizations in time and evaluation performance is achieved using these optimizations/approximations. We also present ways to improve the execution time by searching for an approximate solution, as well as, discuss ways to utilize parallel computation for parts of the algorithm.

5.4.1 Tolerance for "Bad" Moves

In the main algorithm we propose (see Section 5.3) we restrict the accepted moves so that the error decreases monotonically. This is a reasonable restriction that also provides us with theoretical guarantees of convergence. Nonetheless, in our experimental setting, we observed that if we relax this restriction and allow each point to always make the optimal move, regardless if the error (temporarily) increases the algorithm converges faster to better solutions. The idea of allowing greedy algorithms to make some "bad" moves in hope to get over local minima can be found in other optimization algorithms, Simulated Annealing [175] being the most popular. To implement this one can modify line 13 in Algorithm 2 to:

$\mathbf{X}^*, e^* \leftarrow$ OPTIMAL_MOVE($\mathbf{X}^{(k)}$, x, S, $+\infty$)

5.4.2 Updating the Current Dissimilarity Matrix

In line 6 of Algorithm 4 we observe that we recompute the dissimilarity matrix after we make a move for each point. This can be avoided because each move modifies only one point $\mathbf{x}_i^{(k)}$, therefore only the row $\mathbf{d}_{i,:}^{(k)}$ and column $\mathbf{d}_{:,i}^{(k)}$ of the dissimilarity matrix $\mathbf{D}^{(k)}$ are affected. Furthermore only one dimension l of the vector $\mathbf{x}_i^{(k)}$ is modified by the move, i.e., only element $x_{i,l}^{(k)}$ of matrix $\mathbf{X}^{(k)}$. In detail, the element $d_{i,j}$ that stores the dissimilarity between points $\mathbf{x}_i$ and $\mathbf{x}_j$ should be updated as follows for the move from $x_{i,l}^{(k)}$ to $x_{i,l}^{(k+1)}$ for $i \neq j$:

$$d_{i,j}^{(k+1)} = \sqrt{\left(d_{i,j}^{(k)}\right)^2 - \left(x_{i,l}^{(k)} - x_{j,l}^{(k)}\right)^2 + \left(x_{i,l}^{(k+1)} - x_{j,l}^{(k+1)}\right)^2} \tag{5.3}$$

5.4.3 Randomized Direction Selection

It follows from the need to search for the optimal move across the embedding dimensions L, that the complexity of the algorithm has a linear dependency on L. A large value of L will harmfully affect the execution time of the algorithm. An approximate technique to alleviate this problem is to perform a random sampling of $K < L$ directions over all possible directions in the L dimensional space. After that, we define the optimal move for each point in the same way as before but by comparing the errors on the loss function using only the selected K directions. In this way, we are able to select a "good" direction instead of the optimal. Instead of $2L$ moves per epoch one would consider only $2K$ directions in order to compute the new estimate of the error. In this case, the overall complexity per epoch would be $\mathcal{O}(N^2K)$ instead of $\mathcal{O}(N^2L)$.

Presumably, there might be better strategies in order to select the K directions than the naive random sampling of all possible directions in the L dimensional space. As the geometry of the embedding space starts becoming apparent, after a few epochs of the algorithm, it makes sense to increasingly bias the search towards the principal component vectors of the neighborhood of the point that is being moved. For instance, in Hooke and Jeeves pattern search algorithm [171] an increased step is performed towards the direction which yielded a sufficient decrease on the loss function from the previous epoch.

5.4.4 Estimating a Starting Radius

An important parameter for our algorithm is the starting radius $r^{(0)}$. This parameter controls how broad the search will be initially and has an effect similar to the learning rate of gradient-based optimization algorithms. A conservative choice for initial radius will lead the algorithm to converge slowly to a local optimum. Whereas, a high value would most probably cause the error to overshoot by simultaneously making the algorithm harder to converge to a local minima. A simple technique to automatically find a good starting radius is to use binary search between zero and an adequately large value. In particular, we set the starting radius to an arbitrary value, perform a dry run of the algorithm for one epoch and observe the effect on error. If the error increases we halve the radius. Otherwise we double it and repeat the process. This process is allowed to run for a small number of epochs. The starting radius found using this technique is a not too pessimistic or too optimistic estimate of the best parameter value.

5.4.5 Parallel Implementation

Another way to boost the execution time is to utilize parallel computation to speed up parts of the algorithm. In our case we can parallelize the search for the optimal moves across the embedding dimensions using the map-reduce parallelization pattern. Specifically, we can map the search for

candidate moves to run in different threads and store the error for each candidate move in an array $\mathbf{e} = [e_1, e_2.., e_{2L}]$. In other words, the process of computing the optimal direction for each point is completely unrelated to all the other points and thus it can be completely parallelized. After the search completes we can perform a reduction operation (min) to find the optimal move and the optimal error $\mathbf{X}^*, e^*$. For our implementation we used the OpenMP parallelization framework [176]. The parallel implementation of the algorithm led to a significant speedup in execution time by reducing it to $25 - 50\%$ of the initial time required to run.

5.5 GPS Formulation of Pattern Search MDS

Pattern Search MDS belongs to the general class of GPS methods and can be expressed using the unified GPS formulation introduced in Section 2.8. Next, we express our proposed algorithm and associated objective function under this formalism.

First, we restate the problem of MDS in a vectorized form. We use matrix with elements $\{\delta_{ij}\}_{1 \leq i,j \leq N}$ that expresses the dissimilarities between N points in the high dimensional space. The set of points $\{\mathbf{x}_i\}_{i=1}^N$ lie on the low dimensional manifold $\mathcal{M} \in \mathbb{R}^L$ and form the column set of matrix $\mathbf{X}^T$. The matrix $\mathbf{X} \in \mathbb{R}^{N \times L}$ will be now vectorized as an one column vector as shown next:

$$\begin{aligned} \mathbf{x}_i &= [x_{i1}, ..., x_{iL}]^T \in \mathbb{R}^L, 1 \leq i \leq N \\ \mathbf{z} &= vec(\mathbf{X}^T) = [x_{11}, ..., x_{1L}, ..., x_{N1}, ..., x_{NL}]^T \end{aligned} \tag{5.4}$$

Now our new variable $\mathbf{z}$ lies in the search space $\mathbb{R}^{N \cdot L}$. The distance between any two points $\mathbf{x}_i$ and $\mathbf{x}_j$ of the manifold $\mathcal{M}$ remains the same but is now expressed as a function of the vectorized variable $\mathbf{z}$. Namely, $d_{ij}(\mathbf{X}) = ||\mathbf{x}_i - \mathbf{x}_j|| = \sqrt{\sum_{k=1}^{L}(x_{ik} - x_{jk})^2} = d_{ij}(\mathbf{z})$. To this end, our new objective function to minimize g is the MSE between the given dissimilarities δ_{ij} and the euclidean distances d_{ij} in the low dimensional manifold $\mathcal{M}$ as defined in Equation 5.5 shown next:

$$g(\mathbf{z}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (d_{ij}(\mathbf{z}) - \delta_{ij})^2, \;\; \mathbf{z} \in \mathbb{R}^{N \cdot L} \tag{5.5}$$

Consequently, the initial MDS is now expressed as an unconstrained non-convex optimization problem which is expressed by minimizing the function g over the search space of $\mathbb{R}^{N \cdot L}$ (Equation 5.6). Specifically, the L coordinates for all N points on the manifold $\mathcal{M}$ now serve as degrees of freedom for our solution.

$$\mathbf{z}^* = \min_{\mathbf{z} \in \mathbb{R}^{N \cdot L}} g(\mathbf{z}) \tag{5.6}$$

Now that we have formulated the problem and the variable $\mathbf{z}$ in the appropriate format we can match each epoch of our initial algorithm with an iteration of a GPS method. Therefore, the moves produced by our algorithm form a sequence of points $\{\mathbf{z}^{(k)}\}$. Moreover, we are going to define the matrices $\mathbf{B}, \mathbf{C}^{(k)}, \mathbf{P}^{(k)}$ for our algorithm as in Equations 2.63, 2.64. The choice of our basis matrix $\mathbf{B}$ is the identity matrix as shown in Equation 5.8.

$$\mathbf{e}_i = [0, .., \underbrace{1}_{index \;\; i}, ..., 0]^T, 1 \leq i \leq N \cdot L \tag{5.7}$$

$$\mathbf{B} = \mathbf{I}_{N \cdot L} = [\mathbf{e}_1, ..., \mathbf{e}_{N \cdot L}] \tag{5.8}$$

While the identity matrix is non singular and its columns span positively the search space $\mathbb{R}^{N \cdot L}$, we also define $\mathbf{M}^{(k)}$ as the identity matrix. In Equation 5.9 matrix $\mathbf{\Psi}^{(k)}$ represents the movement alongside the unit coordinate vectors of $\mathbb{R}^{N \cdot L}$. Nevertheless, our generating matrix $\hat{\mathbf{C}}$ also comprises of all the remaining possible directions which are generated by the set $\{-1, 0, 1\}$. In total, we have

$3^{N \cdot L} - 2 \cdot N \cdot L$ extra direction vectors inside the corresponding matrix $\mathbf{L}^{(k)}$ as it is shown in Equation 5.10.

$$\begin{aligned} \mathbf{M}^{(k)} &= \hat{\mathbf{M}} = \mathbf{I}_{N \cdot L} \in \mathbb{Z}^{N \cdot L \times N \cdot L} \\ \mathbf{\Psi}^{(k)} &= \hat{\mathbf{\Psi}} = [\hat{\mathbf{M}} \quad -\hat{\mathbf{M}}] \end{aligned} \tag{5.9}$$

$$\begin{aligned} \hat{S} &= \{-1, 0, 1\} \\ \mathbf{L}^{(k)} &= \hat{\mathbf{L}} \\ \hat{\mathbf{L}} &= \{\hat{v} : \hat{v} \in \underbrace{\hat{S} \times ... \times \hat{S}}_{N \cdot L} \wedge \hat{v} \notin \{\mathbf{e}_1, ..., \mathbf{e}_{N \cdot L}\}\} \end{aligned} \tag{5.10}$$

According to Equations 5.9, 5.10, we construct the full pattern matrix $\mathbf{P}^{(k)}$ in Equation 5.11 in a similar way to Equation 2.64. For our algorithm the pattern matrix is is equal to our generating matrix $\mathbf{C}^{(k)} = \hat{\mathbf{C}}$ which is also fixed for all iterations. Conceptually, the generating matrix $\hat{\mathbf{C}}$ contains all the possible exploratory moves while a heuristic is utilized for evaluating the objective function g only for a subset of them.

$$\begin{aligned} \mathbf{C}^{(k)} = \hat{\mathbf{C}} &= [\hat{\mathbf{\Psi}} \quad \hat{\mathbf{L}}] = [\hat{\mathbf{M}} \quad -\hat{\mathbf{M}} \quad \hat{\mathbf{L}}] \\ \mathbf{P}^{(k)} &= \hat{\mathbf{P}} \equiv \mathbf{B}\hat{\mathbf{C}} \equiv \hat{\mathbf{C}} \end{aligned} \tag{5.11}$$

Finally, we configure the updates of the step length parameter for each class of both successful and unsuccessful iterations as they were previously described in Equations 2.66, 2.67, respectively. Recalling the notation of Section 2.8.1, $\hat{\mathbf{s}}^{(k)}$ is the step which is returned from our exploratory moves subroutine at kth iteration. For the successful iterates $g(\mathbf{z}^{(k)} + \hat{\mathbf{s}}^{(k)}) < g(\mathbf{z}^{(k)})$ we do not further increase the length of our moves by limiting $\Lambda = \{1\}$ as follows:

$$\Delta^{(k+1)} = \Delta^{(k)}, \quad \text{if} \quad f(\mathbf{z}^{(k)} + \hat{\mathbf{s}}^{(k)}) < f(\mathbf{z}^{(k)}) \tag{5.12}$$

Similarly, for the unsuccessful iterations $g(\mathbf{z}^{(k)} + \hat{\mathbf{s}}^{(k)}) \geq g(\mathbf{z}^{(k)})$ we halve the distance by a factor of 2 by setting $\theta = \frac{1}{2}$ as it is shown next:

$$\Delta^{(k+1)} = \frac{1}{2}\Delta^{(k)}, \quad \text{if} \quad f(\mathbf{z}^{(k)} + \hat{\mathbf{s}}^{(k)}) >= f(\mathbf{z}^{(k)}) \tag{5.13}$$

A short description of our algorithm as a GPS method for solving the problem stated in Equation 5.6 follows: In each iteration, we fix the optimal coordinate direction for each one of the points lying on the low dimensional manifold $\mathbf{x}_i \in \mathcal{M}, \ 1 \leq i \leq N$. For each internal iteration of Algorithm 4, if the optimal direction produces a lower value for our objective function g we accumulate this direction and move alongside this coordinate of the $\mathbb{R}^{N \cdot L}$. Otherwise, we remain at the same position. As a result, the exploration of coordinates for the new point $\mathbf{x}_{i+1}$ begins from this temporary position. This greedy approach provides a potential one-hot vector as described in Equation 5.7 if the iterate is successful or otherwise, the zero vector $\mathbf{0} \in \mathbb{R}^{N \cdot L}$. The final direction vector $\hat{\mathbf{s}}^{(k)}$ for kth iteration is computed by summing these one-hot or zero vectors. At the kth iteration, the movement would be given by a scalar multiplication of the step length parameter $\Delta^{(k)}$ with the final direction vector in a similar way as defined in Equation 2.65. This provides a simple decrease for the objective function g or in the worst case represent a zero movement in the search space $\mathbb{R}_{N \cdot L}$. Regarding the movement across $\hat{\mathbf{s}}^{(k)}$, it is trivial to show that this reduction of the objective function g is an associative operation. In other words, accumulating all best coordinate steps for each point $\{\mathbf{x}_i\}_{i=1}^{N}$ and performing the movement at the end of the kth iteration (as GPS method formulation requires) produces the same result as taking each coordinate step individually. Finally, pattern search MDS terminates when the step length parameter $\Delta^{(k)}$ becomes smaller than a predefined threshold.

5.6 Convergence of Pattern Search MDS

Now that we have expressed pattern search MDS using the unified GPS framework we can also utilize the Theorems stated in Section 2.8 in order to prove the convergence properties of the proposed algorithm.

First of all, the objective function g is indeed continuously differentiable around all points of the search space $\mathbb{R}^{N \cdot L}$ except of the points where $\mathbf{x}_i = \mathbf{x}_j$. But this case can be handled by alterating a bit the set of minimizers $\mathcal{Z}_\star$ in order to include the points of $L(\mathbf{z}_0)$ where $\mathbf{x}_i = \mathbf{x}_j$, where g is not differentiable [177]. In essense, the theorem will still hold as the objective function would be continously differentiable on the set of open balls $\bigcup_{\mathbf{a} \in L(\mathbf{z}^{(0)})} B(\mathbf{a}, \eta)$ where $\eta > 0$. Moreover, the pattern matrix $\hat{\mathbf{P}}$ in Equation 5.11 contains all the possible step vectors provided by our exploratory moves routine. Thus, all of our exploratory moves are defined by Equation 2.65. In each iteration we evaluate the trial steps alongside all coordinates for all the points $\mathbf{x}_i \in \mathcal{M},\ 1 \leq i \leq N$. In our restated problem definition (see Section 5.5), this is translated to searching all over the identity matrices $\mathbf{I}_{N \cdot L}$ and $-\mathbf{I}_{N \cdot L}$ of the search space $\mathbb{R}^{N \cdot L}$. But from our definition of the first columns of our generating matrix in Equation 5.9 this corresponds to checking all the potential coordinate steps provided by $\hat{\mathbf{\Psi}} = [\mathbf{I}_{N \cdot L} \ \ -\mathbf{I}_{N \cdot L}]$. Consequently, if there exists a simple decrease when moving towards any of the directions provided by the columns of $\hat{\mathbf{\Psi}}$ then our algorithm also provides a simple decrease. This result verifies that Hypothesis 1 is true for the exploratory moves. By combining the differentiability of our objective function g and Hypothesis 1, Theorem 1 holds for pattern search MDS. Hence, $\lim_{k \to +\infty} \inf \ ||\nabla f(\mathbf{z}^{(k)})|| = 0$ is guaranteed.

Trying to further strengthen the convergence properties of the proposed algorithm, we note that most of the requirements of Theorem 2 are met but we fail to meet the specifications of Hypothesis 2 for the minimum decrease provided by the the columns of $\hat{\mathbf{\Psi}}$. However, our generating matrix $\hat{\mathbf{C}} = [\hat{\mathbf{c}}_1, ..., \hat{\mathbf{c}}_{3^{N \cdot L}}]$ is indeed bounded by norm because $||\hat{\mathbf{c}}_j||_1 \leq N \cdot L,\ 1 \leq j \leq 3^{N \cdot L}$. By halving the step length parameter for the unsuccessful iterations we also ensure that $lim_{k \to +\infty} \Delta^{(k)}$. In order to meet the specifications of Theorem 2 we would need a quadratic complexity of $\mathcal{O}((N \cdot L)^2)$ in order to ensure that each iteration provides the same decrease in function g as the decrease provided by the "best" column of $\hat{\mathbf{\Psi}}$. This is formally stated at the second part of Hypothesis 2. If we modify our algorithm in order to meet these requirements we would not be able to implement all the optimizations proposed in Section 5.4 and the overall runtime would be dramatically increased.

5.7 Experiments

5.7.1 Tuning the hyperparameters

We present some guidelines on how to set the hyperparameters for the proposed algorithm and report the values used in the experiments that follow. Specifically:

- The constant ϵ in line 9 of Algorithm 2 determines when the move radius r is decreased. By setting ϵ to a value very close to 0, e.g., 10^{-10}, the search will take more epochs but the solution will be closer to the local optimum. If we relax ϵ to a value around 10^{-2}, we can do a coarse exploration of the search space that will produce a rough solution in a small number of epochs. In our experiments we set $\epsilon = 10^{-4}$ that provides a good trade-off between solution quality and fast convergence for the datasets used. However, the selection of this value has been made empirically.

- We experimentally found that if L is large, we may only search 50% of the search dimensions and still get a good solution, while significantly reducing the execution time. In this context, we randomly sample a new search space for each epoch.

- The proposed algorithm is relatively robust to the choice of the initial size of the move search radius. However, the choice of $r^{(0)}$ does affect convergence speed. In this context, We show the convergence for an example run of the classical Swissroll for various cases of a starting radius. Evidently, the best-case seems to be $r^{(0)} = 32$, a pessimistic one would be $r^{(0)} = 1$ and an optimistic one: ($r^{(0)} = 65536$) starting radius in Figure 5.2.

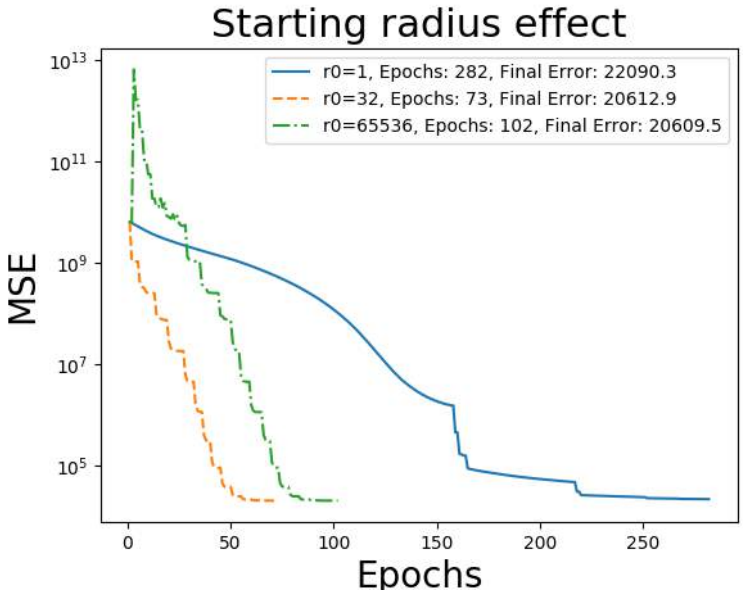


Figure 5.2: Convergence plot for different starting radii

5.7.2 Manifold Geometry Reconstruction

The key assumption in manifold learning is that input data lie on a low-dimensional, non-linear manifold, embedded in a high-dimensional space. Thus non-linear dimensionality reduction techniques aim to extract the low-dimensional manifold from the high dimensional space. For our experiments we use a variety of geometric manifold shapes and compare the proposed MDS to other, well-established dimensionality reduction techniques.

One should note that MDS algorithms with Euclidean distance matrices as input, might not be able to infer data geometry, thus we need to provide as input a *geodesic distance matrix*. This matrix is computed by running Djikstra's shortest path algorithm on the Nearest Neighbors graph trained on the input data. For our experiments we sample 3000 points on 11 $3D$ shapes and reduce them to 2 dimensions using pattern search MDS, SMACOF MDS [111], truncated SVD [178], Isomap [145, 146, 147, 148, 149], Local Linear Embedding (LLE) [152, 153, 154, 155, 156], Hessian LLE [158, 159], modified LLE [157] and Local Tangent Space Alignment (LTSA) [166].

The geodesic distance matrices provided to pattern search MDS and SMACOF MDS is computed using Djikstra's algorithm on KNN (nearest neighbor) graphs (An extensive review of KNN algorithm has been given in a previous section 2.2.4). We list the times it took each method to run. Noticeably, pattern search MDS is faster than SMACOF MDS for all of the experiments we conduct.

We present 4 characteristic shapes selected from the ones we tested which are lying entangled on $\mathbb{R}^3$ and we seek to find $2D$ manifolds which preserve the geometry of the data. In Figure 5.3 we provide the results we obtain by the reconstructed manifolds of all dimensionality reduction algorithms as well as pattern search MDS, for each one of the selected 4 shapes.

On the first shape, we examine the classical swissroll, where a $2D$ plane is "rolled" in $3D$ space and the target is to extract the original $2D$ plane. Results are presented in Figure 5.3a. We observe that linear dimensionality reduction techniques like truncated SVD have trouble unrolling the swissroll.

Also LLE introduces a lot of distortion to the constructed plane. Similarly, in Figure 5.3c, Truncated SVD also has a problem because of the nonlinear entanglement of the data on $3D$ space. LLE also has trouble because it introduces curvature to plane because of the local coordinate system which is constructed for each point.

Next, we examine how the algorithms handle sparse distance matrices. To this end, we generate a dataset of $3D$ non-overlapping clusters with a line connecting the centroids, where sparsity of the distance matrix follows because the vast majority of the points are very closely sampled inside the clusters. A good mapping should preserve the cluster structure in lower dimensions. In Figure 5.3b we see that the truncated SVD and the MDS family of algorithms (proposed, SMACOF, Isomap) produce good results, while the LLE variants can't handle sparsity in distance matrices very well. In particular Hessian LLE and LTSA do not produce any output because of numerical instability. Specifically, in Hessian LLE the matrices used for the computation of the null space become singular, while in LTSA the resulting point coordinates become infinite as we seek to perform eigenvalue decomposition. Pattern search MDS does not rely on eigenvalue computation or equation system solvers and therefore it is numerically stable.

Finally, we showcase how the algorithms perform with transitions from dense to sparse regions with a toroidal helix shape in Figure 5.3d. We can see that five methods, including pattern search MDS, unroll the shape into the expected $2D$ circle, while truncated SVD provides a daisy-like shape. Hessian LLE and LTSA collapse the helix into multiple overlapping circles.

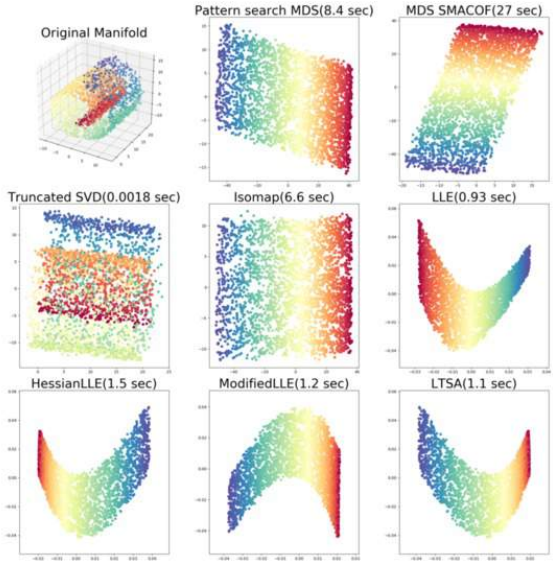
5.7.3 Semantic Similarity

Construction of semantic network models consists of representing concepts (e.g., words, audio, etc.) as vectors in a, possibly high-dimensional, space $\mathbb{R}^n$. The relations between concepts are quantified as the distances, or inversely the cosine similarities, between semantic vectors. The semantic similarity task aims to evaluate the correlation of the similarities between concepts in a given semantic space against a set of ground truth similarity values provided by human annotators.

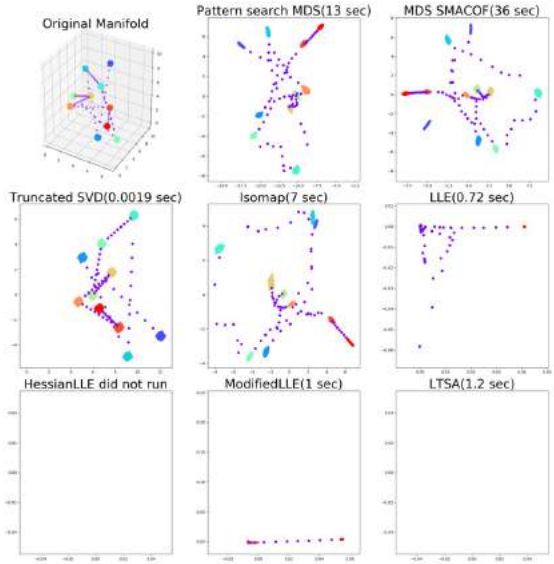
We evaluate the performance of the dimensionality techniques investigated also in Section 5.7.2 for the semantic similarity task. We use the MEN [179] and SimLex-999 [180] semantic datasets as ground truth. Both datasets are provided in the form of lists of word pairs, where each pair is associated with a similarity score. This score was computed by averaging the similarities provided by human annotators. We consider these annotations as ground truth labels in order to evaluate and compare the efficacy of our algorithm on how it preserves the similarity between word-embeddings on the learned low-dimensional manifold. The initial high-dimensional semantic word vectors are 300-dimensional GloVe vectors constructed by [181] using a large Twitter corpus. We reduce the dimensionality of the vectors to the target dimension $L < 300$ and calculate the Spearman correlation coefficient between the human provided and the automatically computed similarity scores. Results are summarized in Table 5.1 for $L = 10$. We observe that LLE yields the best results for MEN, while pattern search MDS performs best for SimLex-999. In addition, we observe that nonlinear dimensionality reduction techniques can significantly improve the performance of the semantic vectors in some cases.

5.7.4 Image classification

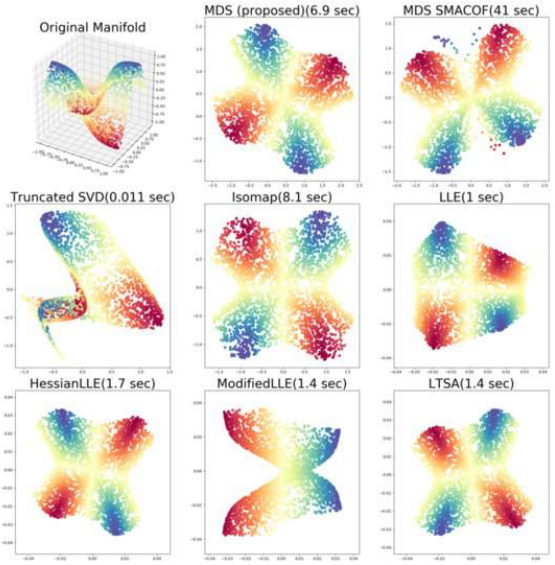
The next set of experiments aims to compare the proposed algorithm to other dimensionality reduction methods for KNN classification (see Section 2.2.4 for an extensive description of KNN nonparametric classification) using a real image dataset. We choose to use MNIST as a benchmark dataset which contains $70,000$ handwritten digit images. We selected a random subset of 1000 images and reduced the dimensionality from 784 to 20. Performance of the models is evaluated on $1-NN$ classification and using 10-fold cross-validation. The evaluation metric is macro-averaged F1 score. Table 5.2 summarizes the results. Observe that dimensionality reduction using pattern search MDS and Truncated SVD can improve classification performance over the original high-dimensional data. Pattern



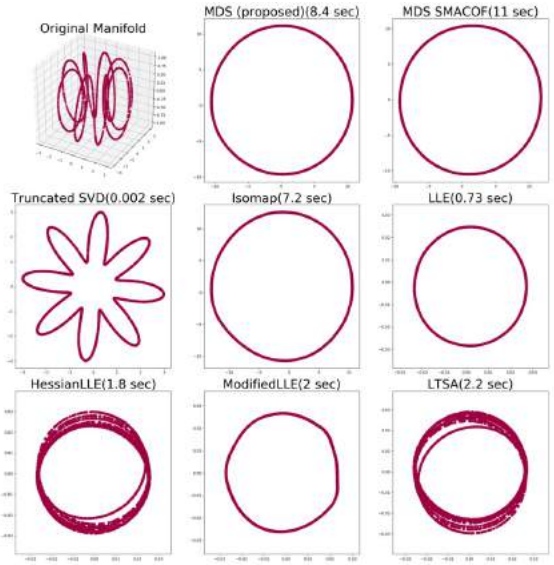
(a) Swissroll



(b) Clusters of Points in 3 dimensions



(c) Twin Peaks



(d) Toroid-Helix

Figure 5.3: Comparison of pattern search MDS with other dimensionality reduction algorithms for reconstructing $2D$- manifolds from $3D$ artificial data

Dimensionality reduction	Dimensions	**MEN**	**SimLex-999**
-	300	0.635	0.177
Pattern Search MDS	10	0.596	**0.242**
MDS SMACOF	10	0.632	0.221
Isomap	10	0.625	0.132
Truncated SVD	10	0.562	0.140
LLE	10	**0.657**	0.172
Hessian LLE	10	0.157	0.004
Modified LLE	10	0.643	0.158
LTSA	10	0.154	0.004

Table 5.1: Comparison of pattern search MDS with other dimensionality reduction algorithms for the semantic similarity task with word-embeddings

search MDS yields the best results overall. Hessian LLE, Modified LLE and LTSA did not run due to numerical instability.

Method	Dimensions	**MNIST 1-NN F1 score**
Original MNIST	784	0.861
Pattern Search MDS	20	**0.878**
MDS SMACOF	20	0.857
Isomap	20	0.829
Truncated SVD	20	0.871
LLE	20	0.813
Hessian LLE	20	–
Modified LLE	20	–
LTSA	20	–

Table 5.2: Comparison of pattern search MDS with other dimensionality reduction algorithms for the MNIST dataset

5.7.5 Speed of Convergence Evaluation

Next, we compare speed of convergence of pattern search MDS and MDS SMACOF, in terms of numbers of epochs. To this end we will consider the experiments of Sections 5.7.2 and 5.7.3 and present comparative convergence plots.

We see the convergence plots for the cases of swissroll, $3D$ clusters, toroid helix in Figure 5.4a, 5.4b and 5.4c, respectively. The convergence plot for the word semantic similarity task is shown in Figure 5.4d. The plots are presented in y-axis logarithmic scale because the starting error is many orders of magnitude larger than the local minimum reached by the algorithms.

For all cases, we observe that pattern search MDS converges very quickly to a similar or better local optimum while MDS SMACOF hits regions where the convergence slows down and then recovers. These saw-like structure of the pattern search plots are due to the fact that we allow for "bad moves" as detailed in Section 5.4.1.

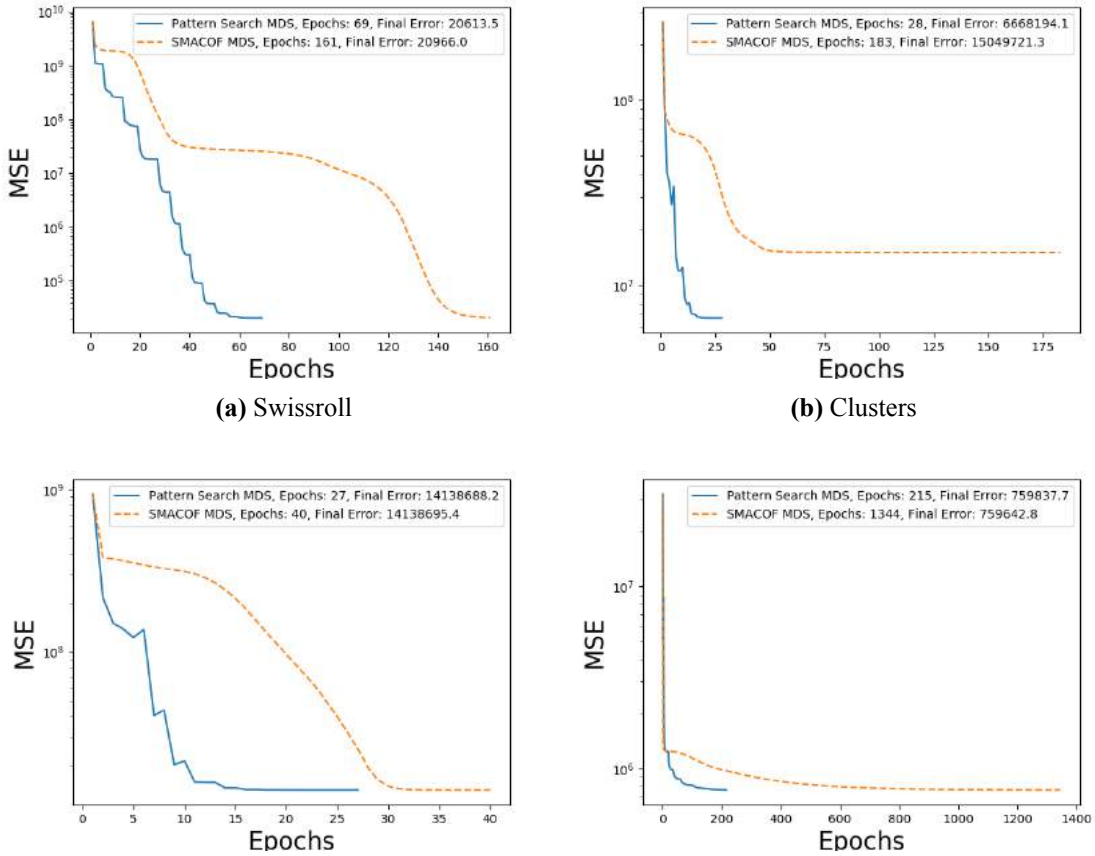
5.7.6 Robustness to Noise

In this set of experiments we aim to demonstrate the robustness of pattern search MDS when the input data are corrupted or noisy. To this end two cases of data corruption are considered: additive noise as well as missing data.

Robustness to Additive Noise

In this set of experiments, we inject Gaussian noise of standard deviation (σ) to the input data and use the dissimilarity matrix calculated on the noisy data as input to each one of the dimensionality reduction algorithms evaluated.

Specifically, for the synthetic data of Section 5.7.2, we follow a qualitative evaluation by showing the unrolled manifolds for high levels of noise. We perform dimensionality reduction for swissroll, toroid helix and $3D$ clusters for increasing noise levels. We report results for the highest possible noise deviation where one or more techniques still produce meaningful manifolds. Beyond these values of σ the original manifolds become corrupted and the output of all methods is dominated by noise. Figs. 5.5a, 5.5b, 5.5c show the results for noisy swissroll with $\sigma = 0.3$, $3D$ clusters with



(a) Swissroll **(b)** Clusters

(c) Toroid-Helix **(d)** Semantic Similarity

Figure 5.4: Convergence comparison of pattern search MDS and MDS SMACOF for reconstruction of geometrical shapes and semantic similarity task

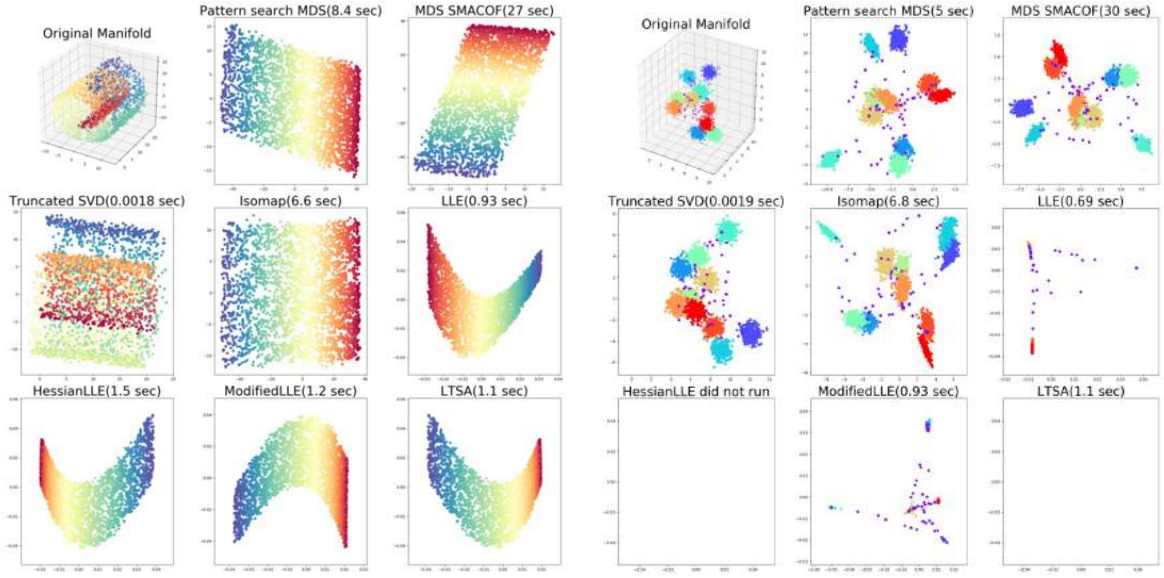
$\sigma = 0.4$ and toroid helix with $\sigma = 0.07$ respectively. Overall, the pattern search MDS, followed by SMACOF MDS and Isomap are more robust to additive noise.

For the experiments for noise injection for semantic similarity task, we inject different levels of Gaussian noise in the original word vectors and evaluate the correlation on both MEN and Simlex-999 datasets (same as the ones we used in the experiments with the clean word-embeddings in Section 5.7.3). Results are presented in Table 5.3. We observe that the relative performance of the algorithms is maintained under noise injection, except for LLE which cannot handle high amounts of noise. LLE is achieving the best correlation values on MEN at $\sigma = 0.01$ and $\sigma = 0.1$, while pattern search MDS achieving the best performance on Simlex-999.

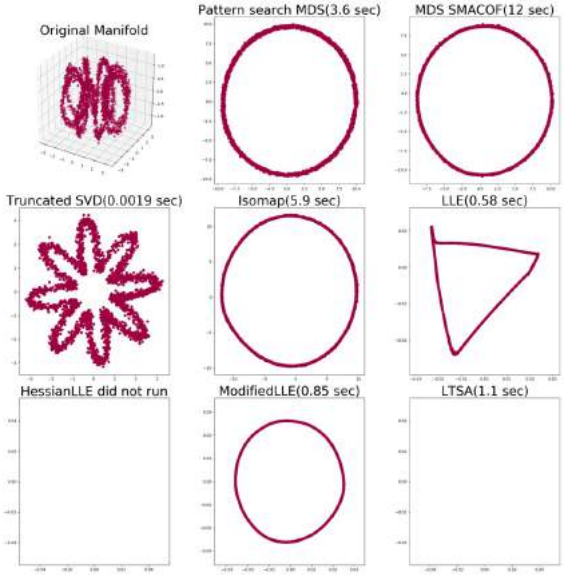
Robustness to Missing Data

In many real case scenarios a part of the data might not be available during training time and they are missing. We need to estimate the robustness of pattern search MDS when a part of the data is not available.

For this experiment, we create two new synthetic datasets, namely a dense and a sparse swissroll with a hole as shown in Figure 5.6. In Fig 5.6a, we show the performance of the various algorithms applied to a dense swissroll with a hole in the middle. As we can see only Hessian LLE, modified LLE and LTSA are able to reconstruct the shape correctly, while MDS algorithms result in distortion around the hole. This is due to the non-convexity we introduced to the space when adding the hole. This distortion can still be observed (to a lesser degree) in the sparse variation shown in Figure 5.6b.



(a) Swissroll + Gaussian Noise **(b)** Clusters + Gaussian Noise



(c) Toroid Helix + Gaussian Noise

Figure 5.5: Comparison of pattern search MDS with other dimensionality reduction methods for the reconstruction of $2D$- manifolds from $3D$ artificial data injected with Gaussian noise

For the sparse data case, we observe that LLE methods result in distortion around the edges.

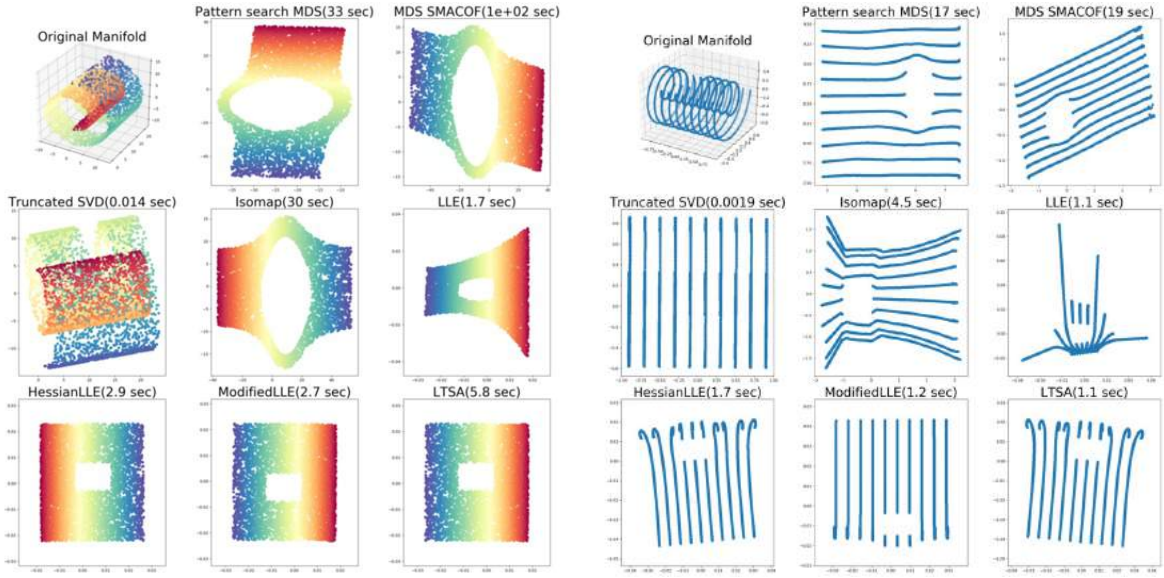
These preliminary experiments indicate that LLE variations can handle better non-convexities in input data, while MDS variations can handle sparse data better. This is because LLE methods are based on inferring and combining local data geometry, while MDS methods are inferring global geometry.

5.7.7 Dimensionality Reduction for Speaker Independent SER

In this section, we apply pattern search MDS for reducing the dimensionality of the acoustic feature sets presented in Sections 4.5.1 (IS10 Feature set: 1582 Features), 4.5.2 (RQA Feature set 432 Features) and 4.5.3 (Fused Feature set: 2014 Features). We evaluate the proposed NLDR algorithm compared to other algorithms for SER under Speaker Independent (SI) experimental setup on an utterance level. For a more extensive description of this setup we refer to the previous Section 4.6.1. For our experiments we use the EmoDB [121] emotional database as described in Section 4.7.1 con-

Method	Dimensions	MEN			SimLex-999		
		$\sigma = 0.01$	$\sigma = 0.1$	$\sigma = 0.5$	$\sigma = 0.01$	$\sigma = 0.1$	$\sigma = 0.5$
Original GloVe	300	0.635	0.619	0.431	0.178	0.169	0.077
pattern search MDS	10	0.593	0.597	0.462	**0.249**	**0.315**	**0.204**
MDS SMACOF	10	0.633	0.620	0.462	0.229	0.222	0.123
Isomap	10	0.622	0.613	**0.497**	0.134	0.124	0.079
Truncated SVD	10	0.562	0.551	0.380	0.140	0.136	0.039
LLE	10	**0.659**	**0.649**	0.369	0.175	0.166	0.052
Hessian LLE	10	0.156	0.144	0.023	0.005	0.04	0.018
Modified LLE	10	0.635	0.633	0.489	0.158	0.162	0.096
LTSA	10	0.155	0.141	0.020	0.06	0.04	0.002

Table 5.3: Comparison of pattern search MDS with other dimensionality reduction methods for semantic similarity task using injected with noisy word-embeddings



(a) Swisshole **(b)** Spiral Hole

Figure 5.6: Comparison of pattern search MDS with other dimensionality reduction methods for shapes with holes and non-convex regions

sisting of 7 emotions. We seek to lower the dimensions of these utterance-level representations by simultaneously not loosing much of the descriptive nature of the input features. In order to identify the true discriminatory ability of the input features we use a nonparametric model in order to assess the learned low-dimensional manifold $\mathcal{M}$. Specifically, we use KNN (see Section 2.2.4) with various numbers of neighbors in order to infer the label for an unknown sample. In essence in order to evaluate the accuracy of the low-dimensional manifold representations, we use a variety of values for the parameter K of nearest neighbors that we take into consideration during testing. Namely we choose K to lie in the set: $K \in \{1, 5, 9, 13, 17, 21\}$. Then the unknown label is inferred form the respective labels of its neighbors. We report both performance metric of WA and UA similar to the previous experimental setup presented in Section 4.7.3. All in all, for each experiment described below, we run each dimensionality reduction algorithm without knowing the labels of the dataset. Next, for each fold we consider one speaker as the test speaker and all the others for training. KNN is applied with the training labels to be the ones of the speakers from the training set. We report the average accuracy obtained from each fold when all the utterances from all speakers have been assessed without taking

care of the number of samples for each speaker.

Using IS10 Feature Set

In Table 5.4, we present the results from a variety of dimensionality reduction algorithms when using the IS10 feature set for SI SER for a variety of configurations of the target dimension L and the number of nearest neighbors K. *IS10 Features* under the column method, means that no other dimensionality reduction is performed and KNN is applied directly over the 1582 feature space. For the row of the matrix with zeros, the respective dimensionality reduction algorithm did not produce any results using the default parameters. For each target dimension $L \in \{2, 10, 25\}$ we notate with bold the highest results obtained when testing all dimensionality reductions under a specified number of K selected nearest neighbors. We also highlight with **blue** the best results that are obtained over all possible configurations of dimensionality reduction methods and K selected nearest neighbors when looking for the inference of an unknown sample from the test set.

It is evident that this feature set offers quite descriptive representations across different speakers for recognizing emotional manifestations. Using KNN from this feature space produces results of up to 69.9% in WA and 64.1% in UA. However, we can achieve similar performances to the initial feature set by using only 10 dimensions or even surpass this performance on both WA and UA by using only 25 dimensions, instead of using 1582, when we use the appropriate dimensionality reduction algorithm before applying KNN. In 2 dimensions, the best performing algorithms for dimensionality reduction are Spectral Clustering for WA and LLE for UA in most of the cases. The 2-dimensional manifolds learned from the former method achieves a WA of up to 53.1% when using $K = 13$ while when we search $K = 17$ neighbors over the manifolds of the latter method we obtain an accuracy of 47.4% in UA. Moreover, when we increase the target dimensions to 10 we notice that the best results in WA are again obtained when employing Spectral Clustering with $K = 17$ (67.4%) but also in UA with $K = 17$ (61.2%). Although, for different values of K neighbors the best results are obtained by different methods. When we further increase the target dimensions to 25, we acquire the best performance in WA with $K = 17$ (71.4%) and in UA with $K = 17$ (65.5%) when using SMACOF. The aforementioned performance scores surpass the best performance of the initial feature space of 1582 features by 1.5% and 2.8% in WA and UA, respectively. These results indicate that the input features can be aptly described by manifolds of lower dimensions and still provide discriminatory ability for SI SER.

Using RQA Feature Set

In Table 5.5, we present the results from a variety of dimensionality reduction algorithms when using the RQA feature set for SI SER for a variety of configurations of the target dimension L and the number of nearest neighbors K. The notation is similar to the one explained before.

Comparing the RQA feature set with the IS10 feature set we conclude that the former provides less information for discriminating emotions, which is similar to the findings in Section 4.7.3 for SI SER. Using KNN from this feature space produces results of up to 56.9% in WA and 48.4% in UA. However, we can easily surpass this performance on both WA and UA by using only 10 dimensions, instead of using 432, when we use the appropriate dimensionality reduction algorithm before applying KNN. In 2 dimensions, the best performing algorithm for dimensionality reduction seems to be LLE for both WA and UA metrics in most of the cases except of some cases that spectral Clustering performs slightly better. The 2-dimensional manifolds learned from the LLE method achieve a WA of up to 48.2% when using $K = 13$. On the other hand, Spectral Clustering provides the best results for UA (41.5%) when we search $K = 21$ neighbors over the manifolds of the latter method. We should not be oblivious to the fact that when we increase significantly the number of neighbors we do not provide any useful results for the discriminatory ability of the features even if we achieve higher results. In essence, when we dramatically increase the neighbors we search the data-representations might be collapsed with one

Method	L	Number of Nearest Neighbors											
		Weighted Accuracy (WA)						Unweighted Accuracy (UA)					
		1	5	9	13	17	21	1	5	9	13	17	21
IS10 Features	1582	61.0	68.8	69.7	69.1	69.9	69.9	58.2	62.2	64.1	62.4	62.7	62.5
Pattern Search MDS	2	43.4	44.9	46.2	46.5	49.5	49.9	40.9	40.0	41.1	41.2	44.9	44.9
Modified LLE		40.4	44.7	46.3	48.5	49.6	49.7	40.3	41.1	42.1	44.4	44.6	42.8
Spectral Clustering		42.4	**49.2**	**51.8**	**53.1**	50.9	**52.5**	37.3	42.2	45.3	**47.3**	43.8	44.8
LTSA		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MDS SMACOF		39.7	46.7	44.7	47.0	48.7	49.5	37.4	42.5	38.9	41.5	43.0	43.8
Truncated SVD		41.7	44.6	44.8	46.8	47.1	49.3	39.7	41.0	40.6	41.8	42.1	44.1
Hessian LLE		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ISOMAP		**44.6**	48.4	49.7	50.2	48.7	49.0	**41.2**	43.5	44.9	44.0	41.3	41.6
LLE		41.3	47.6	51.1	51.2	**51.6**	52.1	38.6	**44.0**	**45.7**	46.3	**47.4**	**46.1**
Pattern Search MDS	10	55.2	64.5	64.7	63.2	65.7	64.3	50.5	58.4	**58.6**	57.7	59.9	58.0
Modified LLE		**59.6**	63.4	62.4	63.2	63.8	64.4	54.8	56.0	55.6	56.1	57.3	57.1
Spectral Clustering		59.4	64.1	**66.1**	65.4	**67.4**	66.1	**55.7**	58.5	60.8	58.9	**61.2**	59.2
LTSA		48.5	49.1	52.5	53.2	53.8	53.2	43.9	43.4	46.6	47.9	47.2	47.0
MDS SMACOF		53.6	62.9	64.2	63.7	64.7	**67.3**	49.3	56.3	57.1	57.5	58.6	**60.9**
Truncated SVD		56.3	63.0	64.1	**66.1**	64.6	64.6	52.3	56.9	57.7	**59.7**	58.8	58.1
Hessian LLE		48.8	49.3	52.5	53.2	53.8	53.2	44.1	43.6	46.6	47.9	47.2	47.0
ISOMAP		57.6	**64.9**	64.7	64.2	64.5	62.9	52.3	**59.1**	56.7	56.3	56.7	55.2
LLE		54.3	58.2	57.8	59.5	58.9	60.5	49.4	53.4	51.7	53.5	52.8	54.4
Pattern Search MDS	25	59.0	66.3	69.3	**70.4**	69.7	71.0	54.9	59.3	62.4	**63.5**	62.2	63.8
Modified LLE		55.2	58.7	65.3	67.0	66.2	66.4	51.2	53.4	62.9	61.0	63.2	62.7
Spectral Clustering		54.2	60.2	61.2	61.7	61.7	62.9	51.7	55.7	55.9	56.5	55.2	57.2
LTSA		55.2	57.0	60.2	59.1	62.2	59.2	50.6	50.3	53.1	51.8	54.9	52.3
MDS SMACOF		**62.2**	**69.9**	**69.8**	69.4	**71.4**	**71.3**	**57.0**	**62.9**	**62.5**	62.7	**65.5**	**65.5**
Truncated SVD		61.6	66.5	68.4	67.2	66.4	68.9	56.2	60.1	61.3	60.6	60.1	61.6
Hessian LLE		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ISOMAP		58.9	64.5	65.9	65.3	67.0	66.0	54.1	59.0	58.3	57.5	58.5	58.6
LLE		54.9	58.0	60.8	62.5	62.2	63.4	50.1	52.9	55.4	56.9	57.1	57.5

Table 5.4: Comparison of pattern search MDS with other dimensionality reduction methods for Speaker Independent Speech Emotion Recognition using IS10 Feature Set

another and still achieve better results. This means that a more sophisticated parametric model could still not be able to disentangle the geometries in order to aptly classify samples from the test set.

Moreover, when we increase the target dimensions to 10 we notice that we achieve the best results in both WA and UA when using the proposed algorithm Patter Search MDS. Specifically, we obtain WA of up to 60.0% with $K = 17$ as well as 52.7% in UA with $K = 13$. For values of of $K = 1$ and $K = 5$ LLE seems to properly capture the geometry of the data. These performance scores significantly outperform the best performance scores of the initial feature space of 432 features by 3.1% and 2.3% in WA and UA, respectively. In this context, we notice the opposite phenomenon of the collapsed data representations which is might be that the low dimensional manifolds might be comprised of small connected components belonging to the same emotional class without being able to preserve the global structure of the data. This could be the case that LLE might perform slightly better than pattern search MDS but overall the representations of the latter might be proven to be more

Method	L	Number of Nearest Neighbors											
		Weighted Accuracy (WA)						Unweighted Accuracy (UA)					
		1	5	9	13	17	21	1	5	9	13	17	21
RQA Feature Set	432	51.8	56.5	56.2	56.5	56.9	55.4	47.3	48.4	47.8	47.1	48.1	46.9
Pattern Search MDS	2	34.8	40.6	39.4	43.1	43.5	44.1	29.4	32.4	31.3	34.1	35.7	34.7
Modified LLE		35.5	40.5	39.7	41.4	41.3	40.8	30.0	34.4	33.2	36.1	36.6	35.9
Spectral Clustering		37.6	42.1	43.6	44.9	45.6	**47.9**	35.1	36.2	**36.7**	38.4	38.5	**41.5**
LTSA		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MDS SMACOF		32.6	40.7	38.4	41.2	42.7	42.9	27.7	34.3	30.9	32.7	33.9	33.7
Truncated SVD		29.2	42.8	43.5	43.7	45.0	45.2	25.9	**36.7**	35.1	35.9	37.0	39.0
Hessian LLE		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ISOMAP		**39.8**	**43.1**	**44.8**	**48.2**	**47.5**	**47.9**	**35.4**	35.5	36.1	**40.1**	**38.7**	39.1
LLE		31.8	34.3	37.1	38.3	37.8	38.5	28.4	27.2	28.9	29.8	28.9	29.2
Pattern Search MDS	10	48.1	57.5	**58.5**	**59.9**	**60.0**	**59.4**	41.5	48.7	**51.3**	**52.7**	**52.5**	**51.4**
Modified LLE		48.6	54.0	57.0	55.5	54.4	54.2	44.5	45.6	49.5	47.1	44.9	43.6
Spectral Clustering		47.3	53.3	52.7	54.6	55.4	55.2	42.9	44.9	44.0	47.5	47.1	47.5
LTSA		34.9	41.1	43.3	43.2	45.2	43.9	28.5	33.7	35.7	34.6	36.9	35.6
MDS SMACOF		47.6	55.8	57.0	57.5	57.7	57.5	41.6	46.8	48.5	49.1	49.7	48.5
Truncated SVD		47.6	54.3	53.5	54.4	55.1	53.5	43.2	46.6	47.0	44.6	45.6	44.2
Hessian LLE		34.9	41.1	43.3	43.2	45.2	43.9	28.5	33.7	35.7	34.6	36.9	35.6
ISOMAP		48.1	51.3	52.2	52.5	51.9	52.6	43.6	42.8	42.7	44.5	43.6	44.3
LLE		**51.2**	**58.0**	57.1	57.6	58.8	58.2	**45.2**	**48.8**	47.5	49.5	50.4	48.0
Pattern Search MDS	25	**51.2**	56.0	**58.6**	**58.5**	**58.8**	58.1	45.1	48.6	**49.8**	49.3	**50.9**	48.7
Modified LLE		46.8	55.2	55.8	55.7	56.1	56.7	41.5	49.2	49.1	48.8	48.8	49.7
Spectral Clustering		49.3	55.1	54.0	57.2	56.0	56.6	42.8	47.7	45.9	48.3	48.3	49.2
LTSA		50.0	53.5	54.0	53.9	56.5	56.2	44.3	45.0	45.7	45.6	47.4	46.9
MDS SMACOF		50.5	**58.1**	57.3	57.4	56.7	57.2	**45.6**	**51.1**	48.3	48.6	46.8	47.7
Truncated SVD		49.5	57.4	57.9	57.9	57.8	58.5	42.7	49.6	49.2	48.9	48.8	51.5
Hessian LLE		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ISOMAP		50.4	54.1	53.1	52.6	53.8	54.0	44.1	46.9	46.1	44.3	45.1	46.4
LLE		49.5	52.8	55.8	56.3	56.6	**58.7**	44.5	46.1	49.4	**50.0**	49.6	**52.0**

Table 5.5: Comparison of pattern search MDS with other dimensionality reduction methods for Speaker Independent Speech Emotion Recognition using RQA Feature Set

resilient and scalable for training using multiple data.

When we further increase the target dimensions to 25, we notice a small decline on performance in both WA and UA metrics. The best performance for these dimensions is obtained with $K = 17$ (58.8%) using pattern search MDS and with $K = 21$ (52.0%) using LLE, for WA and UA, respectively. The aforementioned performance scores still surpass the best performance of the initial feature space of 432 features by 1.9% and 1.6% in WA and UA, respectively. These results indicate that the input features can be aptly described by manifolds of much lower dimensions and still provide discriminatory ability for SI SER.

Using the Fused Feature Set (RQA + IS10)

In Table 5.6, we present the results from a variety of dimensionality reduction algorithms when using the Fused feature set (RQA + IS10) for SI SER under a variety of configurations of the target dimension

L and the number of nearest neighbors K. The notation is similar to the one explained before.

Method	L	Number of Nearest Neighbors											
		Weighted Accuracy (WA)						Unweighted Accuracy (UA)					
		1	5	9	13	17	21	1	5	9	13	17	21
RQA + IS10	2014	62.4	69.3	72.3	70.9	72.4	72.0	58.4	63.7	65.9	63.8	65.1	65.0
Pattern Search MDS	2	**45.7**	**51.7**	50.3	**51.9**	51.5	50.3	39.5	**44.7**	42.8	44.9	44.4	42.9
Modified LLE		42.2	46.6	46.8	49.5	51.1	48.7	39.6	40.4	39.9	44.1	47.9	41.6
Spectral Clustering		43.0	48.0	52.0	51.8	52.5	52.2	39.5	41.3	45.7	45.1	45.1	45.0
LTSA		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MDS SMACOF		43.5	50.1	50.1	51.2	50.5	50.9	39.4	44.5	43.3	44.5	44.2	44.1
Truncated SVD		41.5	41.7	47.2	48.6	49.6	52.0	36.4	36.3	41.9	44.5	43.2	45.2
Hessian LLE		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ISOMAP		42.5	46.4	45.6	46.9	47.4	47.8	38.0	39.5	38.8	39.3	39.8	39.7
LLE		43.5	47.3	**53.1**	51.3	**54.2**	**53.5**	**40.6**	44.0	**48.1**	**46.9**	**49.7**	**48.6**
Pattern Search MDS	10	56.7	66.1	**69.1**	**68.8**	**69.7**	**69.9**	51.5	59.6	**63.1**	**61.6**	**62.6**	**62.9**
Modified LLE		59.3	66.9	66.5	68.0	66.4	66.1	53.5	59.3	58.5	60.5	58.7	58.5
Spectral Clustering		55.6	64.5	65.2	67.6	66.9	65.9	51.5	59.2	59.1	61.1	60.3	59.0
LTSA		39.2	44.4	47.1	47.5	48.9	48.9	34.0	37.3	39.5	40.2	42.4	41.2
MDS SMACOF		58.2	66.1	65.9	66.6	67.6	68.5	52.6	58.4	57.8	58.8	59.4	61.4
Truncated SVD		57.9	65.9	66.2	66.9	66.8	66.8	52.3	58.6	58.8	59.2	59.4	59.1
Hessian LLE		39.2	44.4	47.1	47.5	48.9	48.9	34.0	37.3	39.5	40.2	42.4	41.2
ISOMAP		**62.5**	**68.0**	67.5	67.3	67.8	66.7	**57.7**	**61.1**	57.7	58.1	58.7	57.6
LLE		60.9	62.4	62.2	64.1	62.7	63.0	55.7	54.5	53.9	56.5	55.1	54.2
Pattern Search MDS	25	62.1	68.3	71.4	73.0	**74.4**	**74.1**	57.6	61.3	65.4	66.5	**68.8**	**68.6**
Modified LLE		59.7	62.4	66.5	66.6	66.7	66.7	55.4	57.5	61.0	60.9	59.4	59.6
Spectral Clustering		59.3	65.4	67.7	66.3	67.4	67.3	53.7	59.1	60.5	58.6	61.2	59.3
LTSA		57.4	61.3	63.8	62.9	61.8	61.0	54.6	52.9	55.9	54.3	53.7	52.4
MDS SMACOF		60.8	70.7	**73.6**	**73.9**	74.0	72.9	57.2	64.0	**66.7**	**66.7**	66.2	65.0
Truncated SVD		61.4	**70.9**	72.5	73.0	71.9	71.4	55.5	**64.4**	64.3	65.3	64.3	63.5
Hessian LLE		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ISOMAP		**64.4**	69.8	68.3	68.5	68.1	67.7	**58.5**	62.5	60.3	58.9	59.6	58.7
LLE		59.6	62.0	61.6	62.5	63.3	64.5	55.0	56.0	56.1	57.9	57.2	59.4

Table 5.6: Comparison of pattern search MDS with other dimensionality reduction methods for Speaker Independent Speech Emotion Recognition using the Fused (RQA + IS10) Feature Set

In accordance to our findings in previous Section 4.7.3 for SI SER, we notice that we still get a better performance on both accuracy metrics of WA and UA compared to the previous feature sets when using the Fused Feature set. Applying KNN directly on the representations of the fused set produces results of up to 72.4% in WA and 65.9% in UA. However, we can easily surpass this performance on both WA and UA by using only 25 dimensions, instead of using 2014, when we use the pattern search MDS before applying KNN. In 2 dimensions, the best performing algorithms for dimensionality reduction seems to be pattern search MDS and LLE for both WA and UA metrics. The 2-dimensional manifolds learned from the LLE method achieve a WA of up to 54.2% when using $K = 17$ and a UA of 49.7% for an equal number of neighbors. This huge decline in performance metrics is indicative that 2 dimensions cannot capture the full discriminatory ability of the data under any configuration of

nearest neighbors and dimensionality reduction algorithm.

Moreover, when we increase the target dimensions to 10 we notice that we achieve much better results in both WA and UA when using the proposed algorithm Patter Search MDS for most of the values of K, similar to the ones of the previous results for RQA Set solely (see Table 5.5). Specifically, we obtain WA of up to 69.9% with $K = 21$ as well as 63.1% in UA with $K = 9$ using pattern search MDS. For values of of $K = 1$ and $K = 5$ LLE seems to better capture the geometry of the data than the proposed algorithm. This indicates that pattern search MDS is a resilient method to preserve emotional information from utterance level representations even if the target space is quite low on dimensions.

When we further increase the target dimensions to 25, we acquire the best performance in WA with $K = 17$ (74.4%) and in UA with $K = 17$ (68.8%) when using pattern search MDS before applying KNN. These performance scores outperform the best performance scores of the initial feature space of 2014 features by 2.0% and 2.9% in WA and UA, respectively. Recalling the best performance we obtained under the same experimental setup of SI SER experiments using EmoDB database for evaluation, as presented in Table 4.4, we obtained 82.1% WA using SVM and 77.5% using LR. Comparing the results achieved by our best configurations here, the combination of a NLDR algorithm and KNN yields a lower performance by 7.7% in WA and 8.7% in UA. However, this can be compensated when looking at the number of features where the SER system was trained (25 instead of 2014) and that KNN is one of the simplest models compared to much more complex classifiers such as SVM and LR. Hence, these results indicate that even one of the simplest non-parametric models which is KNN (see Section 2.2.4 for explanation) can perform competitively against parametric models such as SVM and LR (see Sections 2.2.2 and 2.2.3 for the description of these models) when using NLDR algorithms prior to classification. Noticeably, these findings suggest that more sophisticated SER models could use as a preprocessing step NLDR algorithms in order to facilitate the training procedure of their parameters without loosing the expressiveness of the input features and focusing only on the salient parts of these representations.

5.7.8 Comparing (Pattern Search MDS + KNN) to Utterance Level Parametric Models

In this section, we present the best results which are obtained from the combination of Pattern Search MDS with KNN classifier for SER and compare them with the results presented in previous Sections 4.7.3 and 4.7.4. The presented results include all the combinations of feature sets of RQA and IS10 as presented in previous Sections. Except of the experiments presented before about SI SER using EmoDB, we also perform similar experiments to (Leave One Session Out) LOSO with IEMOCAP database using the same utterance level feature extraction pipelines (see Section 4.7.4). We perform Global Normalization (GN) before applying pattern search MDS for reducing the dimensions.

We present the results of all the experiments by using Pattern Search MDS combined with KNN for SER as well as we transfer the results on the same utterance level experimental setups from Sections 4.7.3 and 4.7.4 in Table 5.7. The feature sets are the same as the ones which were used in previous Sections: 4.7.2, 4.7.3 and 4.7.4, namely: IS10, RQA and the Fused Feature Set (RQA + IS10). The first two rows corresponding to each feature set refers to the best results obtained for EmoDB and IEMOCAP under Speaker Independent (SI) and Leave One Session Out (LOSO) utterance level experimental setups presented in previous Sections. In the latter two rows corresponding to each feature set, the best results obtained by a KNN with $K \in \{k \mid k \leq 40 and\ k\ mod\ 4 = 1\}$ after applying Pattern Search MDS in order to reduce the dimensions from the initial feature set to 10 or 25. L corresponds to the target dimension of pattern search MDS while for the first two rows, where no dimensionality reduction is applied (this is notated by a "-" under the "dimensionality Reduction Method" column), this column specifies the dimensions of each utterance level representation.

The results displayed on Table 5.7 signify the importance of applying proper dimensionality reduction before applying an utterance level classifier for SER. In general, the combination of pattern search MDS and the non-parametric classifier KNN does not yield better results than the application

Table 5.7: Comparison of (Pattern Search MDS + KNN) to Utterance Level Parametric Models

Features	Dimensionality Reduction Method	L	Classifier	EmoDB		IEMOCAP	
				WA	UA	WA	UA
IS10	-	1582	SVM	79.7	74.3	59.2	60.5
	-	1582	LR	76.1	71.9	53.5	57.5
	-	1582	KNN	69.9	64.1	53.1	55.7
	Pattern Search MDS	10	KNN	65.7	59.9	53.8	55.2
	Pattern Search MDS	25	KNN	70.4	63.5	**54.5**	56.8
RQA	-	432	SVM	70.9	64.2	53.1	53.7
	-	432	LR	71.1	67.1	52.8	54.3
	-	432	KNN	56.9	48.4	46.9	48.8
	Pattern Search MDS	10	KNN	60.0	52.7	46.4	47.2
	Pattern Search MDS	25	KNN	58.8	50.9	47.6	49.3
RQA+IS10	-	2014	SVM	82.1	76.9	59.5	60.7
	-	2014	LR	80.1	77.5	54.5	58.7
	-	2014	KNN	72.4	65.9	52.6	55.1
	Pattern Search MDS	10	KNN	69.9	63.1	52.9	54.4
	Pattern Search MDS	25	KNN	74.4	68.8	**54.9**	57.2

of an SVM or LR classifier over the initial utterance level feature vectors. For all feature set we notice that by an SVM classifier over the initial feature vectors outperform the best combination of Pattern Search MDS and KNN by margins of $7.7\% - 10.9\%$ in WA and $8.1 - 11.5\%$ in UA for EmoDB SI experiment as well as by $4.7 - 5.5\%$ in WA and $3.5 - 4.4\%$ in UA for IEMOCAP. Likewise, when an LR model is applied over the initial feature vectors instead, we get similar margins that the latter approach outperforms the best combination of Pattern Search MDS and KNN. Although, there are some cases where the proposed algorithm with KNN combined produce better results for IEMOCAP experiment compared to LR. Namely, in WA $53.5\% \rightarrow 54.5\%$ when using IS10 Feature Set and again in WA $54.5\% \rightarrow 54.9\%$ when using the Fused Feature Set. This is quite intriguing if we take into consideration that with 10 or 25 dimensions of the input features and employing only a nonparametric KNN for SER does not perform any kernel tricks like SVM and LR models in order to create discriminatory hyperplanes. Thus, a decline in performance is acceptable when we reduce the feature space to $\approx 0.5 - 1\%$ of its initial dimensionality.

Additionally, the decline in both performance metrics of WA and UA when using Pattern Search MDS with KNN instead of more sophisticated utterance level models like SVM and LR is much more evident in EmoDB experiment while in IEMOCAP the performance obtained is much more comparable to the best one. Presumably, this can be explained because in EMoDB experiment we only have 535 samples in order to construct the low-dimensional manifold representation with 7 emotional classes instead of the IEMOCAP experiment where 5531 instances are available for only 4 emotional classes. In this context, the distribution of the input data which we try to approximate in low-dimensions using pattern search MDS would be sparsely sampled and thus the reconstruction of the manifold would most probably be harder to perform. Noticeably, in all cases, the simple KNN over the reduced in size representations yields better performance in both metrics compared to the case where we apply KNN directly over the initial feature representations.

5.8 Visualizing Emotional Manifolds from Utterance Level Acoustic Features

In this Section we visualize some $2D$ and $3D$ manifolds which were produced after applying NLDR algorithms over the acoustic feature sets that were used in SI SER experiments in the previous Section

5.7.7. We would like to present an insight of how these representations look like in order to draw some qualitative reassurances of the results presented in Tables 5.4, 5.5, 5.6 and 5.7 as well as how each dimensionality reduction method works, especially for the proposed one. For each visualization we select the algorithm that yielded the best performances for the previous SI SER experiments on EmoDB (see Tables 5.4, 5.5, 5.6). Namely, we analyze the manifolds produced by *Pattern Search MDS*, *MDS SMACOF*, *Spectral Clustering*, *LLE*, *ISOMAP* and *Truncated SVD* from left to right and from top to bottom.

5.8.1 IS10 2D Manifolds for EmoDB

In Figure 5.7, we present the 2-dimensional manifolds obtained using the IS10 utterance-level feature set on all utterances of EmoDB (all 7 emotions are included).

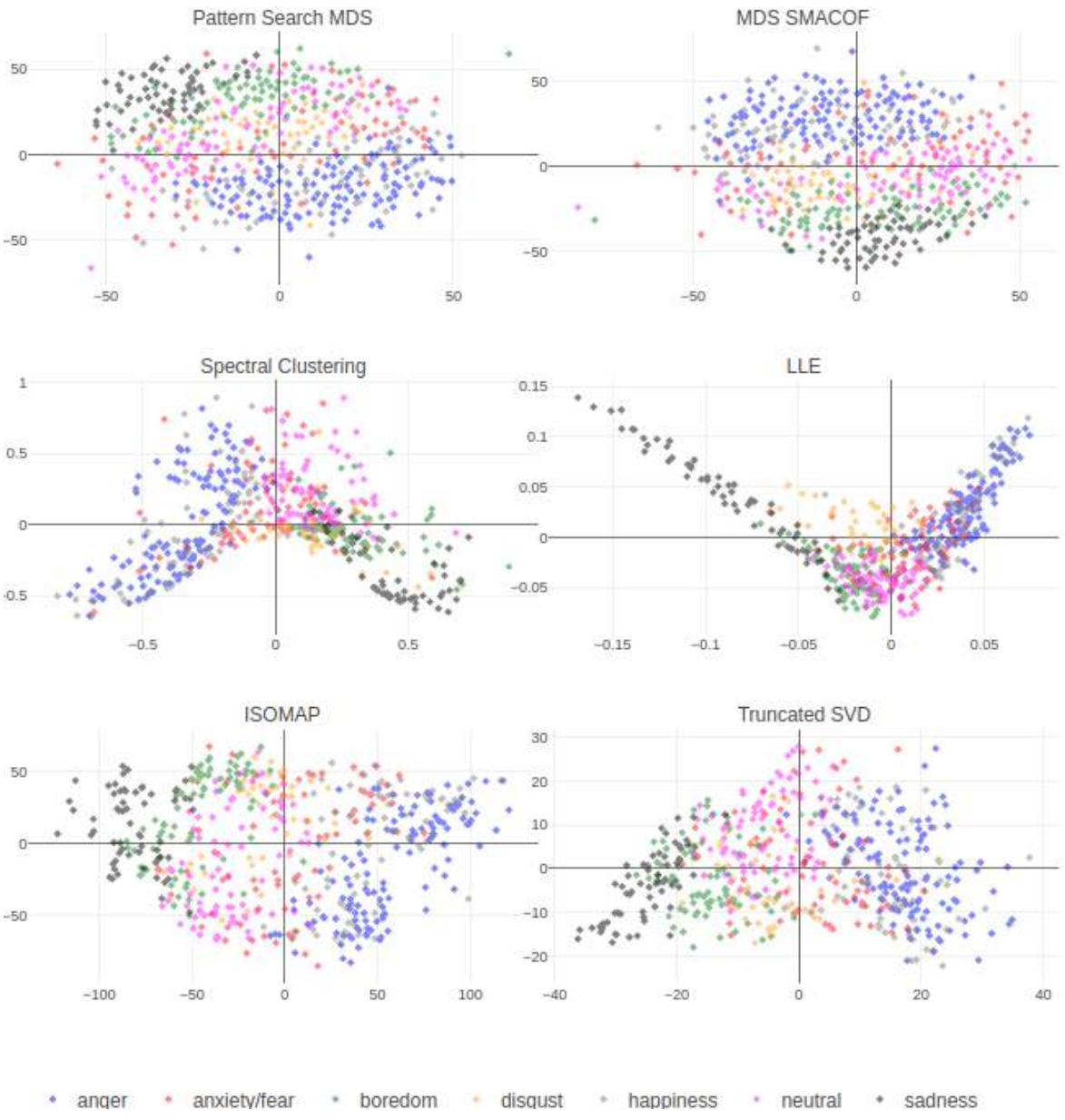


Figure 5.7: Comparison of the produced $2D$ Manifolds when applying dimensionality reduction methods to IS10 acoustic feature representations for EmoDB

All the methods displayed produce manifolds where manyclasses are collapsed to one another. For example, the instances of *happiness* and *anger* seem to overlap in all learned manifolds. Moreover,

some classes like *anxiety/fear* seem to be dispersed over the manifold indicating that the dynamics of this class have not been disentangled using this feature set in order to reflect this on a 2-dimensional map. LLE and Spectral Clustering seem to provide orthogonal and curved manifolds, respectively, which are comprised with regions of discrete emotions that are quite discrete in between the different emotional classes without avoiding the overlap between some classes. this is also depicted in the performance obtained by these methods using KNN in Table 5.7. In addition, Pattern Search MDS produces a manifold very similar to MDS SMACOF, in terms of topology and how all emotional classes are represented.

5.8.2 RQA 2D Manifolds for EmoDB

In Figure 5.8, we present the 2-dimensional manifolds obtained using the RQA utterance-level feature set on all utterances of EmoDB (all 7 emotions are included).

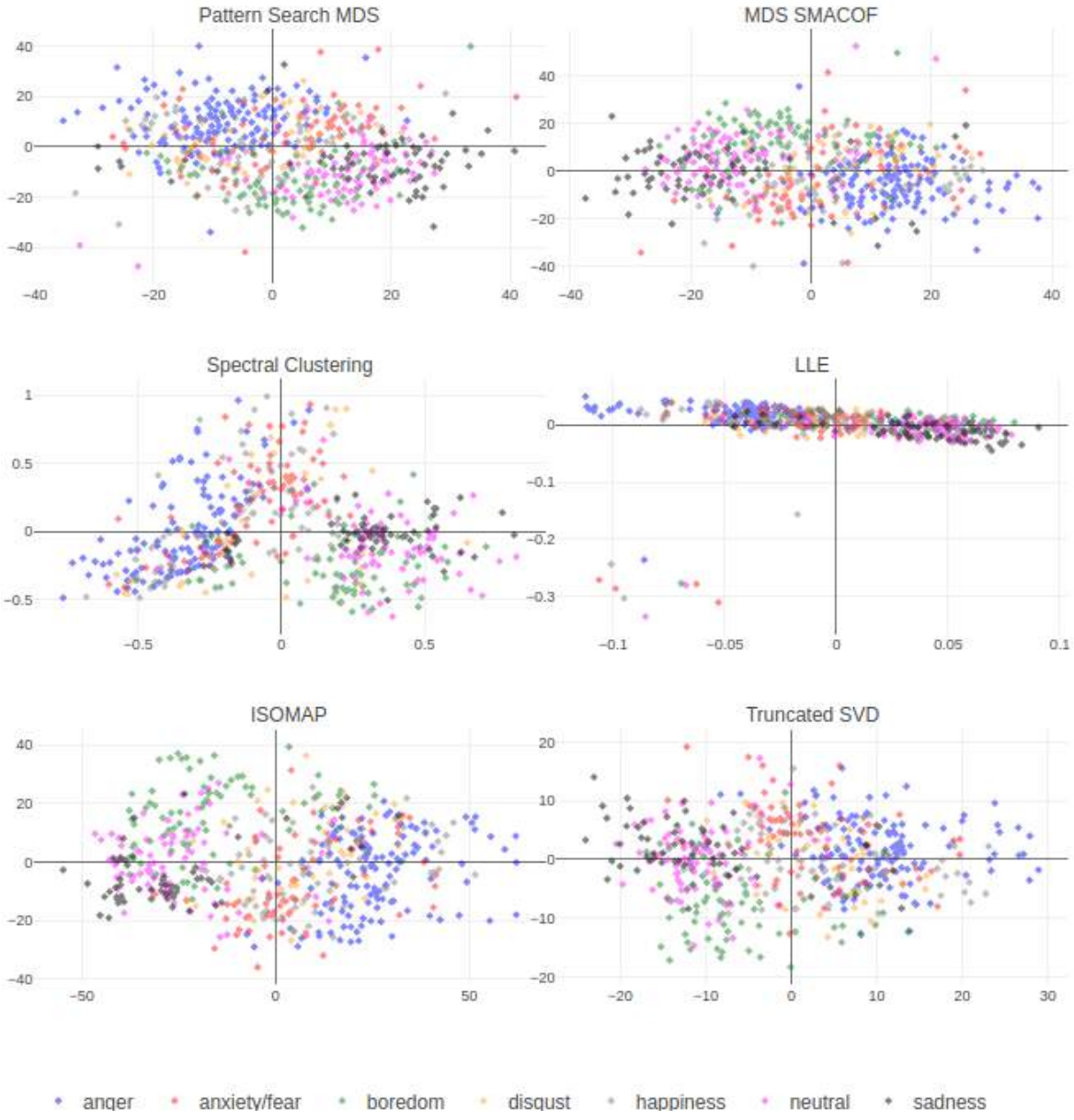


Figure 5.8: Comparison of the produced $2D$ Manifolds when applying dimensionality reduction methods to RQA acoustic feature representations for EmoDB

In general, these manifolds are consisted of much more emotional regions that overlap than the

ones obtained when using IS10. Although, this is completely related to the expressiveness of each feature type and in accordance with our findings in previous Sections of this thesis we have concluded that RQA is less descriptive than IS10 feature set when they are evaluated individually (see Tables 4.4, 4.5 and 5.7). There is a complete collapse of the samples distribution for LLE method which is also evident by the performance obtained using KNN (see Table 5.5). It is quite evident that for RQA representations, ISOMAP performs significantly well in creating regions with merely only one or two emotions and this is also depicted in both WA and UA performance metrics in Table 5.5. The regions of the same emotional samples are not so homogeneous for Pattern Search MDS, Truncated SVD and MDS SMACOF.

5.8.3 Fused Feature Set (RQA + IS10) 2D Manifolds for EmoDB

In Figure 5.9, we present the 2-dimensional manifolds obtained using the Fused (RQA + IS10) utterance-level feature set on all utterances of EmoDB (all 7 emotions are included).

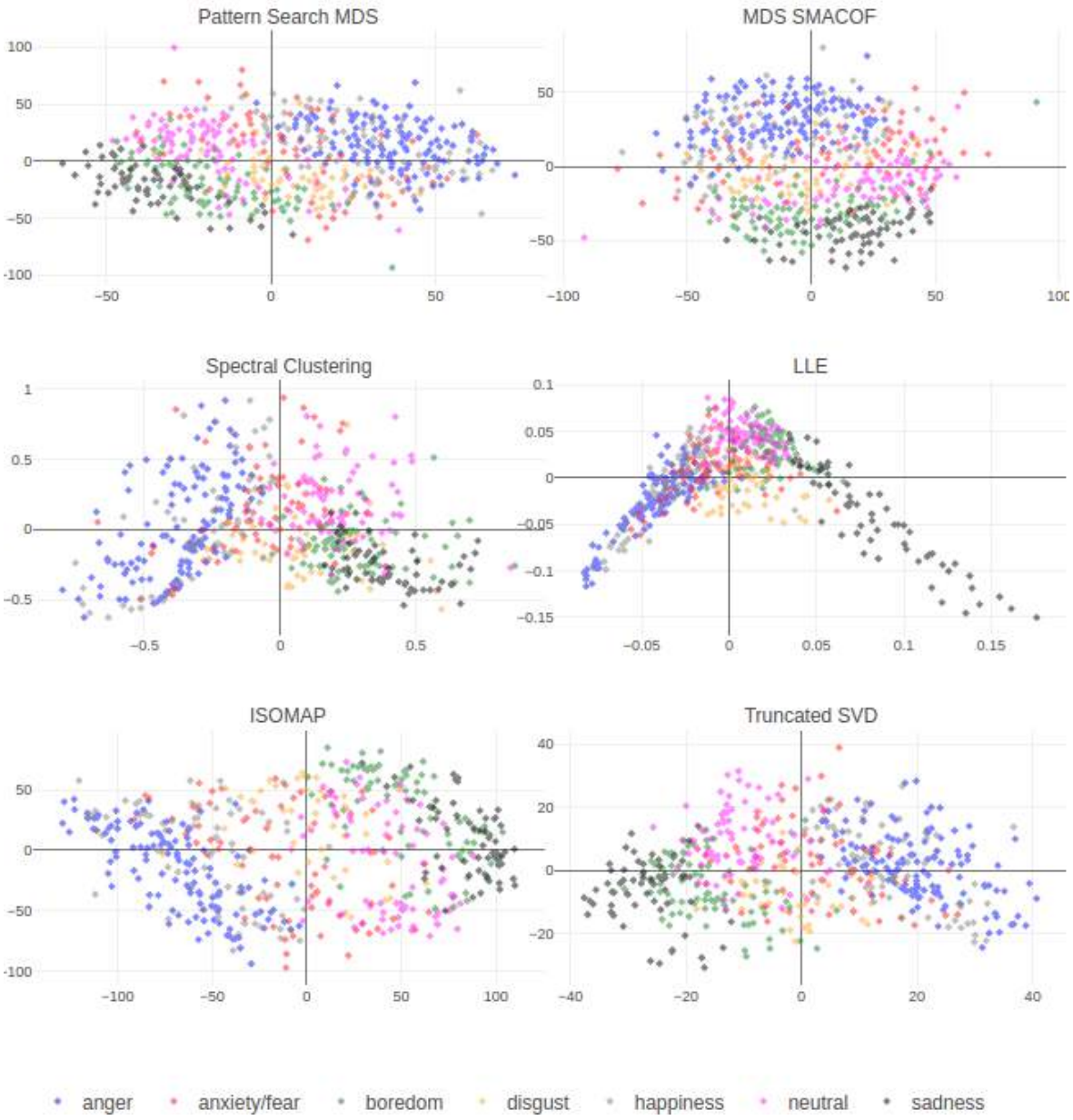


Figure 5.9: Comparison of the produced $2D$ Manifolds when applying dimensionality reduction methods to Fused (RQA + IS10) acoustic feature representations for EmoDB

For the Fused feature set we notice that all methods produce 2-dimensional representations which are comprised of quite disentangled regions of emotional instances. However, Pattern Search MDS, LLE and Spectral Clustering seem to provide the best one because of the adjacency of samples that belong to each class. This is also verified by the performance of KNN classification presented in Table 5.6. Although, ISOMAP produces well interconnected regions of emotional classes, overall the produced emotional map contains large holes and sparse regions which is also depicted in its performance. Presumably, this is caused due to the misrepresentations of geodesic distances over the distances computed directly from the distances of the points on the Euclidean Space. Finally, Truncated SVD also produces a map in which the regions for each emotional class are quite dense within classes but without avoiding the overlapping regions between similar expressed emotions.

5.8.4 Fused Feature Set (RQA + IS10) 3D Manifolds for 2 Male Speakers of IEMOCAP

Building upon drawing some qualitative conclusions about how these low-dimensional manifolds are constructed for each dimensionality reduction algorithm, we will add one more dimension to the target dimension L. We would also like to see the behavior of these algorithms with a higher number of samples and under different speakers. We isolate two male speakers from IEMOCAP database, namely the male speakers from the first two sessions and we plot the learned 3-dimensional manifolds for all the aforementioned dimensionality reduction methods in Figure 5.10. We follow the same experimental setup as described in previous Section 4.7.1 for taking a subset of IEMOCAP dataset considering only 5531 utterances including 4 emotions (1103 angry, 1636 happy, 1708 neutral and 1084 sad), where we merge excitement and happiness class into the latter one. Moreover, we only consider the fused feature set (RQA + IS10) because of its descriptive nature for SER. In this way, we can focus on these representations by taking into consideration many more samples from the distribution.

In this setup we notate with discrete symbols of "x" and "spheres" the emotional representations corresponding to each male speaker from the first and the second IEMOCAP session, respectively. We can see that for example, ISOMAP fails to map the distributions of the same emotional classes for both speakers. There are two disconnected regions that correspond to different speakers if we look closer in Figure 5.10 (down and left). Presumably, these disconnected regions are created because of some dominant features like pitch and energy that dominate other features as well in high dimensional feature space. In this context, when we compute the geodesic distances over all points utterances of the same speaker contain speaker related information that yields a closer connectivity between samples of the same speaker. This could also lead to significant decline in performance of SER system for which, the generalization beyond speaker related information is a necessity for accurate inference.

In addition, MDS SMACOF and Pattern Search MDS again produce very similar results to ellipsoids which is in accordance to the previous findings about the learned 2-dimensional manifolds when using the Fused Feature Set (see Figure 5.9). In this context, LLE also seems to construct the low dimensional manifold using a translation, rotation, and rescaling of local data which are exactly the operations to which the weights that LLE defines for each data point are invariant. Spectral clustering follows a similar pattern when reconstructing the manifold on $3D$ to the previously shown manifold on $2D$ with the curved surface manifold over the samples (see Figure 5.9). This is directly related to the handcrafted kernel that spectral Clustering uses in order to normalize the global affinity matrix between the points of the high-dimensional space.

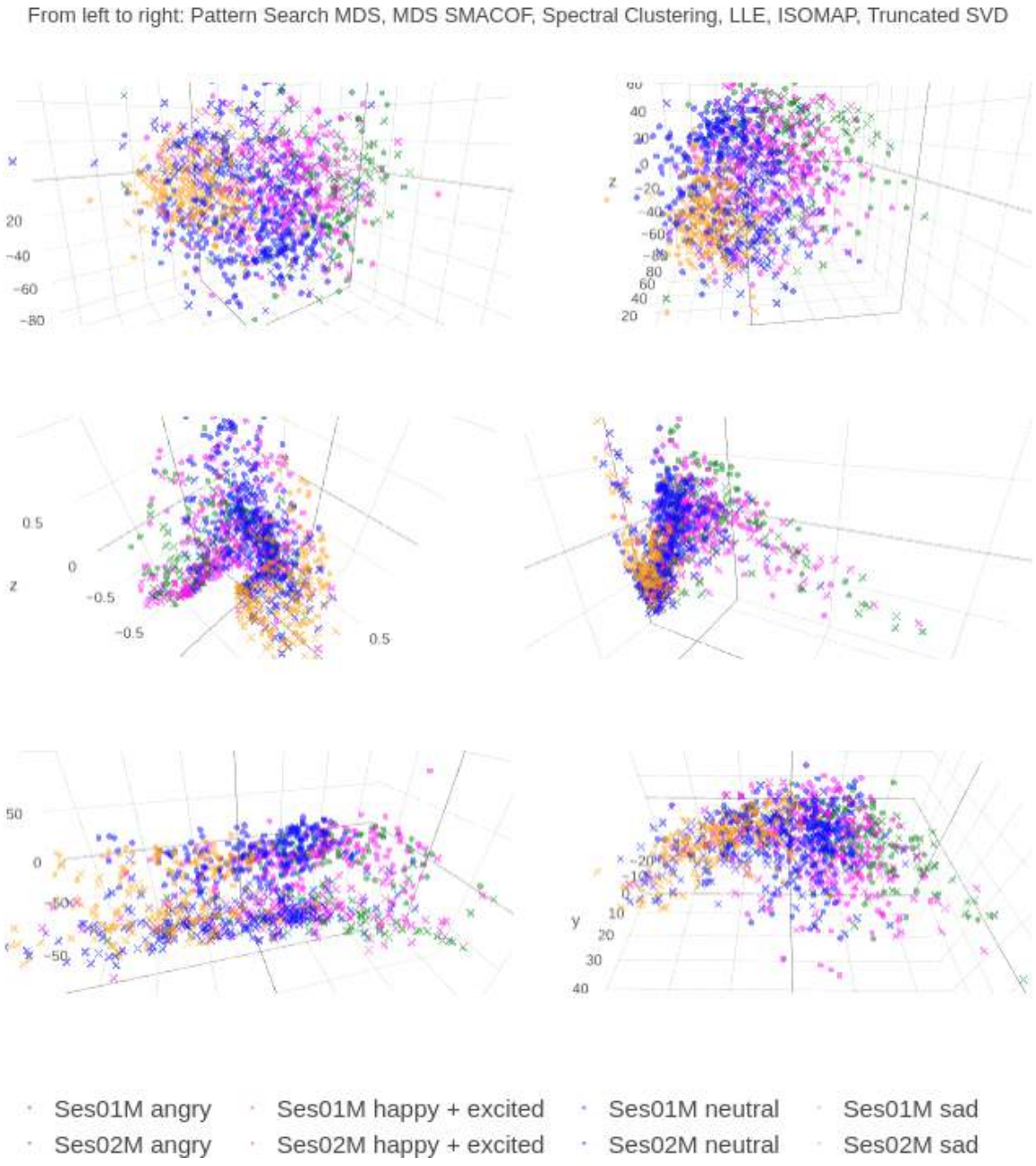


Figure 5.10: Comparison of the produced $3D$ Manifolds when applying dimensionality reduction methods to Fused (RQA + IS10) acoustic feature representations for 2 Male Speakers of IEMOCAP

Chapter 6

Epilogue

6.1 Conclusions

We draw some conclusions from this work which can be divided into three main categories corresponding to Chapters 3, 4 and 5, respectively. Hence, we will list the most profound conclusions we drew from the work presented in this thesis in respect to the Chapters referred above. The conclusions below are ultimately related to the questions posed in the corresponding Sections 1.4.1, 1.4.2, 1.4.3. specifically, the conclusions below are presented in such a way that some answers could be given to aforementioned questions, considering the work which has been made towards the conduction of this diploma thesis.

6.1.1 Conclusions from Chapter 3

We have shown that the time-scale on which we extract features for SER, has profound effect on RNNs performance. LLDs work well on a time-scale that roughly corresponds to phoneme level. Conversely, statistical features remarkably encode emotional context on a time-scale that corresponds to word level. In our proposed LSTM model, we extract statistical features over segments of speech which correspond to a couple of words and we report state-of-the-art results on the IEMOCAP database, with a much lower computational complexity and without employing extra attention mechanisms.

6.1.2 Conclusions from Chapter 4

We investigated the usage of nonlinear RQA measures extracted from RPs for SER. The effectiveness of these features has been tested under both utterance-based and segment-based approaches across three emotion databases. The fusion of nonlinear and conventional feature sets yields significant performance improvement over traditional feature sets for all SER tasks; the performance improvement is especially large when speaker identity is unknown. The fused data set improves on the state-of-the-art for SER under most testing conditions, classification methods and datasets. Noticeably, the experiments show that the proposed RQA feature set can be adequately used on its own for SER and without any utilization of a feature selection scheme or dimensionality reduction algorithm. As a result, we believe that, recurrence analysis of speech signals is a promising direction for SER research.

6.1.3 Conclusions from Chapter 5

We propose pattern search MDS, a novel algorithm for nonlinear dimensionality reduction, inspired by gradient-free optimization methods. Pattern search MDS is formulated as an instance of the wider family of GPS methods, thus providing theoretical guarantees of convergence up to a fixed point. Additional optimizations further improve the performance of our algorithm in terms of computational efficiency, robustness and solution quality. The qualitative evaluation against other popular dimensionality reduction techniques for both clean and noisy manifold geometry shapes indicates that pattern search MDS can accurately infer the intrinsic geometry of manifolds embedded in high-dimensional spaces. Furthermore, the comparison of convergence characteristics against SMACOF MDS show

that pattern search MDS converges in fewer epochs to similar or better solutions. Experiments on real data yield comparable to state-of-the-art results both for a lexical semantic similarity task and on MNIST for KNN classification. Open-source implementations of pattern search MDS and the data generation process are provided to facilitate the reproducibility of our results.

When we consider an experimental setup for SI SER, our findings suggest that the proposed algorithm can aptly capture the dissimilarities of the input feature space and embed it in a low-dimensional manifold. Compared to other dimensionality reduction algorithms we notice that in most cases pattern search MDS provides the best results among all the other methods for SI SER. Presumably, RQA representations have a highly nonlinear structure between different classes of emotions and this is why linear dimensionality reduction algorithms fail to preserve those invariants on low dimensional representations. The learned representations obtained from NLDR methods can still preserve the discriminatory ability of the high-dimensional space for all configurations and feature sets and even surpass it on WA and UA performance metrics using KNN. The same applies when we assess our algorithm towards LOSO experiments using IEMOCAP dataset. All these experiments have been performed for all the combinations of feature sets proposed in in Sections 4.5.1 (IS10 Feature set: 1582 Features), 4.5.2 (RQA Feature set 432 Features) and 4.5.3 (Fused Feature set: 2014 Features). We consistently notice that KNN classifier performs better when we use the proposed algorithm compared to the initial high-dimensional representations of emotional utterances. Last but not least, we provide some visualizations from the learned manifolds of emotional utterances of $2D$ and $3D$ Euclidean spaces in order to get a qualitative insight of the pros and the cons of applying each dimensionality reduction algorithm for reducing the dimensions of utterance level representations.

6.2 Future Work

In the context of extracting better nonlinear representations from RPs in order to improve the performance of SER systems, we could try to isolate the invariant features from these representations which is the small periodic subgraph of the images. As we have previously mentioned, one of the main challenges is to capture invariant dynamical signatures of each speech frame (see Section 1.4.2). Vowels are dominated by the fundamental frequency of the speaker and thus, we would like to get rid off these speaker related features in our nonlinear representations from RPs. An outline of the proposed feature extraction scheme is presented in Figure 6.1. First, we perform vowel segmentation in order to focus only on the most frequency dominated regimes of the speech signal. The rest feature extraction pipeline is identical to the one which has been presented in previous Section 4.5.2 until the part of extracting the periodic subgraph from each RP. After that, each speech segment or even utterance of emotional speech can be represented by a sequence of these RP subgraphs which would hopefully preserve structures indicative of the nature of each emotional manifestation. Last but not least, continuous RP representations are already energy normalized and under the proposed feature extraction algorithm, we could extract much more resilient nonlinear features for SER without dominating our feature representations with speaker related attributes.

In addition, we can also try to better fuse the information from different acoustic feature sets for building more robust SER systems. As we have seen in Sections 4.7.2, 4.7.3, 4.7.4 and 5.7.7, the Fused feature set (RQA + IS10) yielded significant improvement over IS10 and RQA feature sets. However, the simple concatenation of the two feature representations is the most naive way of combining information which is extracted from different sources. In this context, we can build upon the previous idea of extracting a sequence of periodic-free RP subgraphs from the vowels of each speech segment or utterance and combine them with spectral representations (essentially a sequence of Fourier transformations from all the speech frames included in the speech segment). The combination of both information channels (linear and nonlinear representations) could be effectively made by using a bimodal autoencoder like the one presented in Figure 6.2. We train an autoencoder which is based on the reconstruction of both modalities when only only one modality is available each time. With this technique, the proposed framework would be able not only to reconstruct the missing information

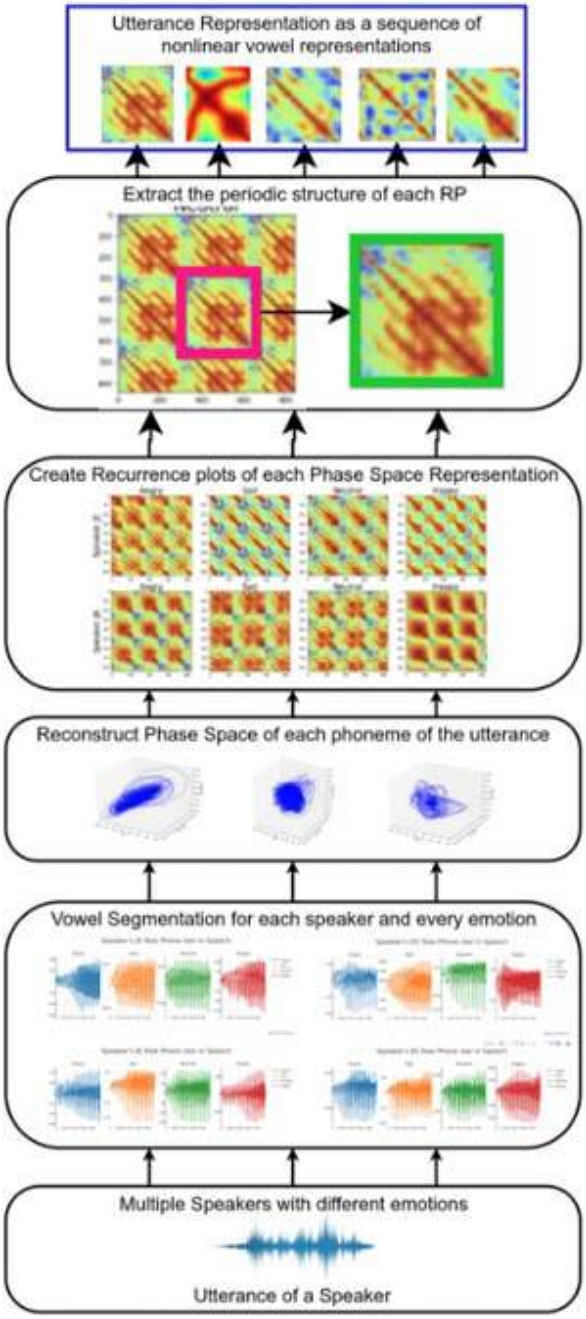


Figure 6.1: Isolating periodic subgraphs from RPs in order to extract pitch-invariant representations for SER

modality but would also obtain a resilient representation within the intermediate layer called "Shared Utterance Encoded Representation" which would condense salient information from both channels. Each of one of the two parts of the Autoencoder could implement any DNN or RNN architecture but should be identical among its mirror counterparts. Namely, the *blue* part of the autoencoder (down-left and bottom-right) would be penaltized not only by the error of reconstructing itself ("Linear Modality Reconstruction") but also from the reconstruction of the dual information channel ("Nonlinear Modality Reconstruction"). Similar things would apply to the nonlinear dual counterpart of the network (*green*). Hopefully, the intermediate representation would encode salient emotional information at the end of the training process and could be used as an input vector for subsequent layers of deeper SER systems.

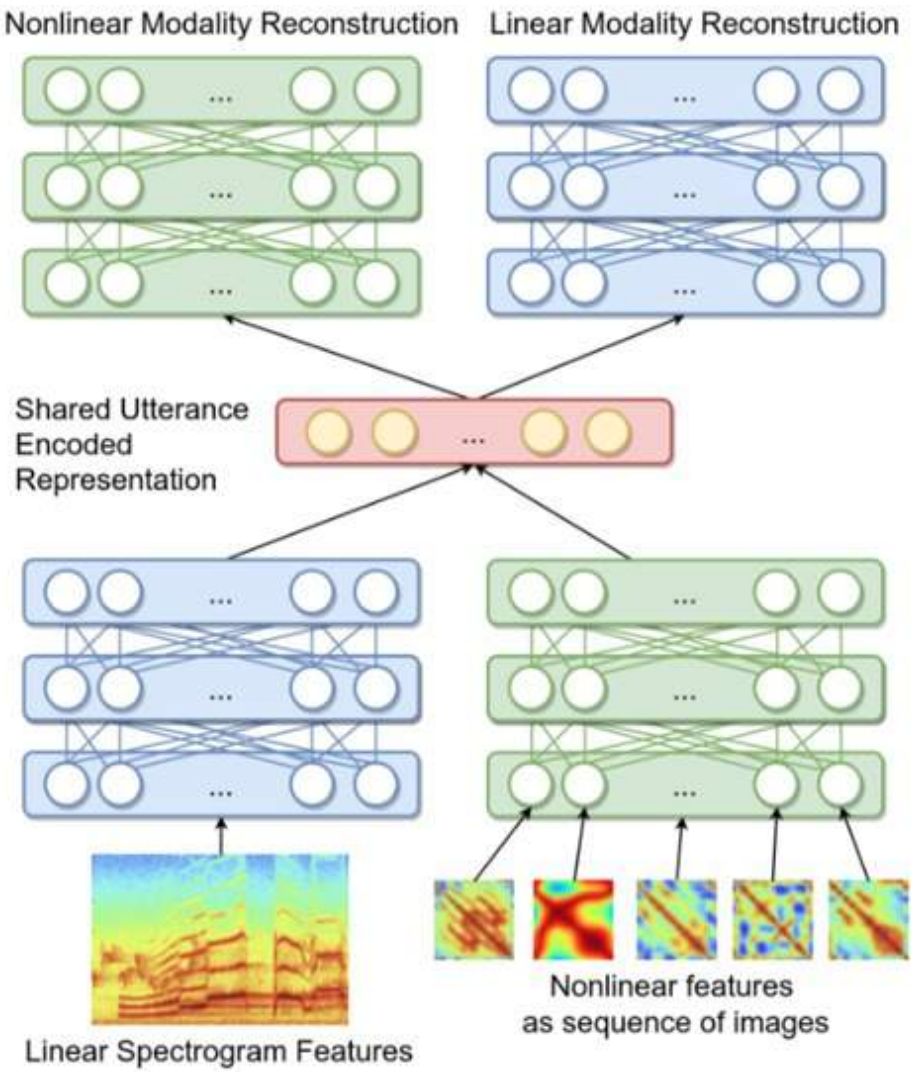


Figure 6.2: Bimodal Autoencoder for combining spectral and nonlinear representations of speech signals

Considering some next steps for the proposed Pattern Search MDS 5.3, future work will also focus on improving runtime performance and scalability of the proposed algorithm. Specifically, an approach for decreasing the computational complexity per epoch would be to narrow down the search space of possible moves as the geometry of the embedding space becomes more apparent by biasing the moves towards the principal component vectors of the neighborhood of the point that is being moved. This can be viewed as a combination of pattern search and gradient descent, where the search space of moves is wide at the beginning and then gets increasingly biased towards the direction of the gradient. Our algorithm can scale to large numbers of points by utilizing Landmark points [150] or fast approximations to MDS [182]. These approaches aim to alleviate the computational and memory cost of computing the full distance matrix, by approximating the data geometry using smaller submatrices. Moreover, stochastic approximations like stochastic SMACOF [183] can be adapted to pattern search MDS.

We also plan to provide more in-depth theoretical insights and ways to enable pattern search MDS to capture complex geometrical properties of input data. We aim to perform a detailed analysis on how heuristics and especially allowing for "bad moves" affect the performance of pattern search MDS. Furthermore, in Sections 5.7.2 and 5.7.6 we showcased that MDS can better handle sparse data and LLE can better handle non-convexity and missing data. This makes sense, as MDS takes into account the global geometry of the embedding space, while LLE focuses on the geometry of local neighborhoods.

We plan to combine the cost functions of these approaches to infer both global and local geometry of the low dimensional data manifold. Another way to increase the expressiveness of the algorithm is to investigate a wider variety of distance metrics, and specifically non-symmetrical distance metrics, motivated by cognitive sciences [184].

Bibliography

[1] E. Tzinis, G. Paraskevopoulos, C. Baziotis, and A. Potamianos, "Integrating recurrence dynamics for speech emotion recognition," in *Proceedings of INTERSPEECH (in press)*, 2018.

[2] E. Tzinis and A. Potamianos, "Segment-based speech emotion recognition using recurrent neural networks," in *Affective Computing and Intelligent Interaction (ACII), 2017 Seventh International Conference on*. IEEE, 2017, pp. 190–195.

[3] G. Paraskevopoulos, E. Tzinis, E.-V. Vlatakis-Gkaragkounis, and A. Potamianos, "Pattern search multidimensional scaling," *arXiv:1806.00416*, 2018.

[4] A. Chorianopoulou, E. Tzinis, E. Iosif, A. Papoulidi, C. Papailiou, and A. Potamianos, "Engagement detection for children with autism spectrum disorder," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5055–5059.

[5] K. Nesbitt, K. Blackmore, G. Hookham, F. Kay-Lambkin, and P. Walla, "Using the startle eyeblink to measure affect in players," in *Serious Games Analytics*. Springer, 2015, pp. 401–434.

[6] L. R. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. PTR Prentice Hall Englewood Cliffs, 1993, vol. 14.

[7] A. Jafari, F. Almasganj, and M. N. Bidhendi, "Statistical modeling of speech poincaré sections in combination of frequency analysis to improve speech recognition performance," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 3, p. 033106, 2010.

[8] P. M. Niedenthal, L. W. Barsalou, P. Winkielman, S. Krauth-Gruber, and F. Ric, "Embodiment in attitudes, social perception, and emotion," *Personality and social psychology review*, vol. 9, no. 3, pp. 184–211, 2005.

[9] R. R. Cornelius, *The science of emotion: Research and tradition in the psychology of emotions.* Prentice-Hall, Inc, 1996.

[10] T. C. Schneirla, "An evolutionary and developmental theory of biphasic processes underlying approach and withdrawal." 1959.

[11] H. Schlosberg, "Three dimensions of emotion." *Psychological review*, vol. 61, no. 2, p. 81, 1954.

[12] K. R. Scherer, "Vocal affect expression: A review and a model for future research." *Psychological bulletin*, vol. 99, no. 2, p. 143, 1986.

[13] I. Fónagy and K. Magdics, "Emotional patterns in intonation and music," *STUF-Language Typology and Universals*, vol. 16, no. 1-4, pp. 293–326, 1963.

[14] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor, "Emotion recognition in human-computer interaction," *IEEE Signal processing magazine*, vol. 18, no. 1, pp. 32–80, 2001.

[15] C. E. Izard, *Human emotions*. Springer Science & Business Media, 2013.

[16] V. Petrushin, “Emotion in speech: Recognition and application to call centers,” in *Proceedings of Artificial Neural Networks in Engineering*, vol. 710, 1999.

[17] C. M. Lee and S. S. Narayanan, “Toward detecting emotions in spoken dialogs,” *IEEE Transactions on speech and audio processing*, vol. 13, no. 2, pp. 293–303, 2005.

[18] R. W. Picard *et al.*, “Affective computing,” 1995.

[19] M. El Ayadi, M. S. Kamel, and F. Karray, “Survey on speech emotion recognition: Features, classification schemes, and databases,” *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[20] S. G. Koolagudi and K. S. Rao, “Emotion recognition from speech: a review,” *International journal of speech technology*, vol. 15, no. 2, pp. 99–117, 2012.

[21] C.-N. Anagnostopoulos, T. Iliou, and I. Giannoukos, “Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011,” *Artificial Intelligence Review*, vol. 43, no. 2, pp. 155–177, 2015.

[22] X. Li, J. Tao, M. T. Johnson, J. Soltis, A. Savage, K. M. Leong, and J. D. Newman, “Stress and emotion classification using jitter and shimmer features,” in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4. IEEE, 2007, pp. IV–1081.

[23] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, and S. Narayanan, “The interspeech 2010 paralinguistic challenge,” in *Proceedings of INTERSPEECH*, 2010, pp. 2794–2797.

[24] B. Schuller and A. Batliner, *Computational paralinguistics: emotion, affect and personality in speech and language processing*. John Wiley & Sons, 2013.

[25] B. Schuller, S. Steidl, and A. Batliner, “The interspeech 2009 emotion challenge,” in *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[26] S. Steidl, *Automatic classification of emotion related user states in spontaneous children’s speech*. University of Erlangen-Nuremberg Erlangen, Germany, 2009.

[27] Z. Aldeneh and E. M. Provost, “Using regional saliency for speech emotion recognition,” in *Proceedings of ICASSP*, 2017, pp. 2741–2745.

[28] R. Xia and Y. Liu, “A multi-task learning framework for emotion recognition using 2d continuous space,” *IEEE Transactions on Affective Computing*, vol. 8, no. 1, pp. 3–14, 2017.

[29] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi *et al.*, “The interspeech 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism,” in *Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France*, 2013.

[30] E. Mower, M. J. Mataric, and S. Narayanan, “A framework for automatic human emotion classification using emotion profiles,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1057–1070, 2011.

[31] B. Schuller, R. Müller, M. Lang, and G. Rigoll, “Speaker independent emotion recognition by early fusion of acoustic and linguistic features within ensembles,” in *Ninth European Conference on Speech Communication and Technology*, 2005.

[32] J. Deng, Z. Zhang, F. Eyben, and B. Schuller, "Autoencoder-based unsupervised domain adaptation for speech emotion recognition," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1068–1072, 2014.

[33] C. M. Lee, S. Narayanan, and R. Pieraccini, "Recognition of negative emotions from the speech signal," in *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on*. IEEE, 2001, pp. 240–243.

[34] J. Rong, G. Li, and Y.-P. P. Chen, "Acoustic feature selection for automatic emotion recognition from speech," *Information processing & management*, vol. 45, no. 3, pp. 315–328, 2009.

[35] H. M. Fayek, M. Lech, and L. Cavedon, "Evaluating deep learning architectures for speech emotion recognition," *Neural Networks*, vol. 92, pp. 60–68, 2017.

[36] S. Ghosh, E. Laksana, L.-P. Morency, and S. Scherer, "Representation learning for speech emotion recognition." in *Proceedings of INTERSPEECH*, 2016, pp. 3603–3607.

[37] Q. Mao, M. Dong, Z. Huang, and Y. Zhan, "Learning salient features for speech emotion recognition using convolutional neural networks," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2203–2213, 2014.

[38] J. C. Vasquez-Correa, T. Arias-Vergara, J. R. Orozco-Arroyave, J. Vargas-Bonilla, and E. Noeth, "Wavelet-based time-frequency representations for automatic recognition of emotions from speech," in *Proceedings of the 12. ITG Symposium on Speech Communication*, 2016, pp. 1–5.

[39] F. Shah *et al.*, "Wavelet packets for speech emotion recognition," in *Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2017 Third International Conference on*. IEEE, 2017, pp. 479–481.

[40] R. Sun and E. Moore, "Investigating glottal parameters and teager energy operators in emotion recognition," in *Affective Computing and Intelligent Interaction*, 2011, pp. 425–434.

[41] D. Dimitriadis, P. Maragos, and A. Potamianos, "Auditory teager energy cepstrum coefficients for robust speech recognition," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[42] A. Shahzadi, A. Ahmadyfard, A. Harimi, and K. Yaghmaie, "Speech emotion recognition using nonlinear dynamics features," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 23, no. Sup. 1, pp. 2056–2073, 2015.

[43] T. Chaspari, D. Dimitriadis, and P. Maragos, "Emotion classification of speech using modulation features," in *Proceedings of Signal Processing Conference (EUSIPCO)*, 2014, pp. 1552–1556.

[44] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5200–5204.

[45] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[46] C. Busso, S. Lee, and S. Narayanan, "Analysis of emotionally salient aspects of fundamental frequency for emotion detection," *IEEE transactions on audio, speech, and language processing*, vol. 17, no. 4, pp. 582–596, 2009.

[47] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. M. Lee, A. Kazemzadeh, S. Lee, U. Neumann, and S. Narayanan, "Analysis of emotion recognition using facial expressions, speech and multimodal information," in *Proceedings of the 6th international conference on Multimodal interfaces*. ACM, 2004, pp. 205–211.

[48] R. Xia, J. Deng, B. Schuller, and Y. Liu, "Modeling gender information for emotion recognition using denoising autoencoder," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 990–994.

[49] Y. Sun and G. Wen, "Ensemble softmax regression model for speech emotion recognition," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 8305–8328, 2017.

[50] B. Schuller and G. Rigoll, "Timing levels in segment-based speech emotion recognition," in *Proceedings of INTERSPEECH*, 2006.

[51] J. H. Jeon, R. Xia, and Y. Liu, "Sentence level emotion recognition based on decisions from subsentence segments," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4940–4943.

[52] X. Mao, L. Chen, and L. Fu, "Multi-level speech emotion recognition based on hmm and ann," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 7. IEEE, 2009, pp. 225–229.

[53] E. M. Provost, "Identifying salient sub-utterance emotion dynamics using flexible units and estimates of affective flow," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3682–3686.

[54] C.-H. Wu, W.-B. Liang, K.-C. Cheng, and J.-C. Lin, "Hierarchical modeling of temporal course in emotional expression for speech emotion recognition," in *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*. IEEE, 2015, pp. 810–814.

[55] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[56] A. Chorianopoulou, P. Koutsakis, and A. Potamianos, "Speech emotion recognition using affective saliency." in *INTERSPEECH*, 2016, pp. 500–504.

[57] A. Nogueiras, A. Moreno, A. Bonafonte, and J. B. Mariño, "Speech emotion recognition using hidden markov models," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[58] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[59] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[60] K. Han, D. Yu, and I. Tashev, "Speech emotion recognition using deep neural network and extreme learning machine," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[61] J. Lee and I. Tashev, "High-level feature representation using recurrent neural network for speech emotion recognition," 2015.

[62] Z.-Q. Wang and I. Tashev, "Learning utterance-level representations for speech emotion and age/gender recognition using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5150–5154.

[63] A. Metallinou, M. Wollmer, A. Katsamanis, F. Eyben, B. Schuller, and S. Narayanan, "Context-sensitive learning for enhanced audiovisual emotion classification," *IEEE Transactions on Affective Computing*, vol. 3, no. 2, pp. 184–198, 2012.

[64] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[65] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *Proceedings of ICASSP*, 2017, pp. 2227–2231.

[66] C.-W. Huang and S. S. Narayanan, "Attention assisted discovery of sub-utterance structure in speech emotion recognition." in *Proceedings of INTERSPEECH*, 2016, pp. 1387–1391.

[67] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[68] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, p. 335, 2008.

[69] B. Zhang, E. M. Provost, and G. Essl, "Cross-corpus acoustic emotion recognition with multi-task learning: Seeking common ground while preserving differences," *IEEE Transactions on Affective Computing*, 2017.

[70] J. Gideon, S. Khorram, Z. Aldeneh, D. Dimitriadis, and E. M. Provost, "Progressive neural networks for transfer learning in emotion recognition," *arXiv preprint arXiv:1706.03256*, 2017.

[71] N. Mesgarani, M. Slaney, and S. A. Shamma, "Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 920–930, 2006.

[72] Y. Bengio and P. Frasconi, "Credit assignment through time: Alternatives to backpropagation," in *Advances in Neural Information Processing Systems*, 1994, pp. 75–82.

[73] N. van Noord and E. Postma, "Learning scale-variant and scale-invariant features for deep image classification," *Pattern Recognition*, vol. 61, pp. 583–592, 2017.

[74] K. Honda, "Physiological processes of speech production," in *Springer handbook of speech processing*. Springer, 2008, pp. 7–26.

[75] M. A. Little, P. E. McSharry, S. J. Roberts, D. A. Costello, and I. M. Moroz, "Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection," *BioMedical Engineering OnLine*, vol. 6, no. 1, p. 23, 2007.

[76] H. Herzel, "Bifurcations and chaos in voice signals," *Applied Mechanics Reviews*, vol. 46, no. 7, pp. 399–413, 1993.

[77] W. T. Fitch, J. Neubauer, and H. Herzel, "Calls out of chaos: the adaptive significance of nonlinear phenomena in mammalian vocal production," *Animal behaviour*, vol. 63, no. 3, pp. 407–418, 2002.

[78] V. Pitsikalis and P. Maragos, "Analysis and classification of speech signals by generalized fractal dimension features," *Speech Communication*, vol. 51, no. 12, pp. 1206–1223, 2009.

[79] C. L. Webber Jr and J. P. Zbilut, "Recurrence quantification analysis of nonlinear dynamical systems," *Tutorials in contemporary nonlinear methods for the behavioral sciences*, pp. 26–94, 2005.

[80] H. Herzel, D. Berry, I. Titze, and I. Steinecke, "Nonlinear dynamics of the voice: signal analysis and biomechanical modeling," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 5, no. 1, pp. 30–34, 1995.

[81] S. S. Narayanan and A. A. Alwan, "A nonlinear dynamical systems analysis of fricative consonants," *The Journal of the Acoustical Society of America*, vol. 97, no. 4, pp. 2511–2524, 1995.

[82] T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," *Journal of statistical Physics*, vol. 65, no. 3-4, pp. 579–616, 1991.

[83] G. Boeing, "Visual analysis of nonlinear dynamical systems: Chaos, fractals, self-similarity and the limits of prediction," *Systems*, vol. 4, no. 4, 2016.

[84] N. Marwan, M. C. Romano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," *Physics reports*, vol. 438, no. 5-6, pp. 237–329, 2007.

[85] R. Bellman, "Adaptative control processes," 1961.

[86] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: a comparative," *J Mach Learn Res*, vol. 10, pp. 66–71, 2009.

[87] L. Cayton, "Algorithms for manifold learning," *Univ. of California at San Diego Tech. Rep*, vol. 12, no. 1-17, p. 1, 2005.

[88] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[89] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[90] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*. Elsevier, 1992, pp. 65–93.

[91] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[92] M. Aizerman, E. M. Braverman, and L. Rozonoer, "Theoretical foundations of potential function method in pattern recognition," *Automation and Remote Control*, vol. 25, pp. 917–936, 1964.

[93] M. B. Christopher, *PATTERN RECOGNITION AND MACHINE LEARNING*. Springer-Verlag New York, 2016.

[94] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical review A*, vol. 33, no. 2, p. 1134, 1986.

[95] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Phys. Rev. A*, vol. 45, pp. 3403–3411, 1992.

[96] J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle, "Recurrence plots of dynamical systems," *EPL (Europhysics Letters)*, vol. 4, no. 9, p. 973, 1987.

[97] S. Schinkel, O. Dimigen, and N. Marwan, "Selection of recurrence threshold for signal detection," *The european physical journal special topics*, vol. 164, no. 1, pp. 45–53, 2008.

[98] M. Thiel, M. C. Romano, J. Kurths, R. Meucci, E. Allaria, and F. T. Arecchi, "Influence of observational noise on the recurrence quantification analysis," *Physica D: Nonlinear Phenomena*, vol. 171, no. 3, pp. 138–152, 2002.

[99] W. S. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, Dec 1952.

[100] I. Borg and P. J. F. Groenen, *Modern multidimensional scaling: theory and applications*. Springer, 2005.

[101] J. C. Gower, "Some distance properties of latent root and vector methods used in multivariate analysis," *Biometrika*, vol. 53, no. 3-4, pp. 325–338, 1966.

[102] M. A. A. Cox and T. F. Cox, "Multidimensional scaling on the sphere," in *Compstat*, D. Edwards and N. E. Raun, Eds. Heidelberg: Physica-Verlag HD, 1988, pp. 323–328.

[103] A. Cvetkovski and M. Crovella, "Low-stress data embedding in the hyperbolic plane using multidimensional scaling," *Appl. Math*, vol. 11, no. 1, pp. 5–12, 2017.

[104] H. Lindman and T. Caelli, "Constant curvature riemannian scaling," *Journal of Mathematical Psychology*, vol. 17, no. 2, pp. 89–109, 1978.

[105] R. N. Shepard, "The analysis of proximities: Multidimensional scaling with an unknown distance function. i." *Psychometrika*, vol. 27, no. 2, pp. 125–140, Jun 1962.

[106] ——, "The analysis of proximities: Multidimensional scaling with an unknown distance function. ii," *Psychometrika*, vol. 27, no. 3, pp. 219–246, Sep 1962.

[107] P. Groenen and I. Borg, "Past, present, and future of multidimensional scaling," *Visualization and Verbalization of Data*, pp. 95–117, 2014.

[108] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, Mar 1964.

[109] ——, "Nonmetric multidimensional scaling: A numerical method," *Psychometrika*, vol. 29, no. 2, pp. 115–129, Jun 1964.

[110] S. L. France and J. D. Carroll, "Two-way multidimensional scaling: A review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 5, pp. 644–661, 2011.

[111] J. D. Leeuw, I. J. R. Barra, F. Brodeau, G. Romier, and B. V. C. (eds, "Applications of convex analysis to multidimensional scaling," in *Recent Developments in Statistics*. North Holland Publishing Company, 1977, pp. 133–146.

[112] J. de Leeuw, "Convergence of the majorization method for multidimensional scaling," *Journal of Classification*, vol. 5, no. 2, pp. 163–180, Sep 1988.

[113] V. Torczon, "On the convergence of pattern search algorithms," *SIAM Journal on optimization*, vol. 7, no. 1, pp. 1–25, 1997.

[114] E. D. Dolan, R. M. Lewis, and V. Torczon, "On the local convergence of pattern search," *SIAM Journal on Optimization*, vol. 14, no. 2, pp. 567–583, 2003.

[115] R. M. Lewis and V. Torczon, "Pattern search algorithms for bound constrained minimization," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 1082–1099, 1999.

[116] A. R. Conn, N. I. M. Gould, and P. Toint, "A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds," *SIAM Journal on Numerical Analysis*, vol. 28, no. 2, pp. 545–572, 1991.

[117] R. M. Lewis and V. Torczon, "Pattern search methods for linearly constrained minimization," *SIAM Journal on Optimization*, vol. 10, no. 3, pp. 917–941, 2000.

[118] C. Audet, "Convergence results for generalized pattern search algorithms are tight," *Optimization and Engineering*, vol. 5, no. 2, pp. 101–122, Jun 2004.

[119] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM International Conference on Multimedia*, 2013, pp. 835–838.

[120] L. R. Rabiner, R. W. Schafer *et al.*, "Introduction to digital speech processing," *Foundations and Trends® in Signal Processing*, vol. 1, no. 1–2, pp. 1–194, 2007.

[121] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of german emotional speech," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[122] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.

[123] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[124] J. Bayer, C. Osendorfer, D. Korhammer, N. Chen, S. Urban, and P. van der Smagt, "On fast dropout and its applicability to recurrent networks," *arXiv preprint arXiv:1311.0701*, 2013.

[125] T. Dozat, "Incorporating nesterov momentum into adam," 2016.

[126] C. Laurent, G. Pereyra, P. Brakel, Y. Zhang, and Y. Bengio, "Batch normalized recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2657–2661.

[127] F. Chollet *et al.*, "Keras," 2015.

[128] J. Bergstra, F. Bastien, O. Breuleux, P. Lamblin, R. Pascanu, O. Delalleau, G. Desjardins, D. Warde-Farley, I. Goodfellow, A. Bergeron *et al.*, "Theano: Deep learning on gpus with python," in *NIPS 2011, BigLearning Workshop, Granada, Spain*, vol. 3. Citeseer, 2011.

[129] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[130] S. Wu, T. H. Falk, and W.-Y. Chan, "Automatic speech emotion recognition using modulation spectral features," *Speech communication*, vol. 53, no. 5, pp. 768–785, 2011.

[131] P. Henríquez, J. B. Alonso, M. A. Ferrer, C. M. Travieso, and J. R. Orozco-Arroyave, "Nonlinear dynamics characterization of emotional speech," *Neurocomputing*, vol. 132, pp. 126–135, 2014.

[132] A. Lombardi, P. Guccione, and C. Guaragnella, "Exploring recurrence properties of vowels for analysis of emotions in speech," *Sensors & Transducers*, vol. 204, no. 9, p. 45, 2016.

[133] R. V. Donner, M. Small, J. F. Donges, N. Marwan, Y. Zou, R. Xiang, and J. Kurths, "Recurrence-based time series analysis by means of complex network methods," *International Journal of Bifurcation and Chaos*, vol. 21, no. 04, pp. 1019–1046, 2011.

[134] F. Orsucci, R. Petrosino, G. Paoloni, L. Canestri, E. Conte, M. A. Reda, and M. Fulcheri, "Prosody and synchronization in cognitive neuroscience," *EPJ Nonlinear Biomedical Physics*, vol. 1, no. 1, p. 6, 2013.

[135] N. D. Duran, R. Dale, C. T. Kello, C. N. Street, and D. C. Richardson, "Exploring the movement dynamics of deception," *Frontiers in psychology*, vol. 4, p. 140, 2013.

[136] A. Goshvarpour, A. Abbasi, and A. Goshvarpour, "Dynamical analysis of emotional states from electroencephalogram signals," *Biomedical Engineering: Applications, Basis and Communications*, vol. 28, no. 02, p. 1650015, 2016.

[137] T. Tošić, K. K. Sellers, F. Fröhlich, M. Fedotenkova, A. Hutt *et al.*, "Statistical frequency-dependent analysis of trial-to-trial variability in single time series by recurrence plots," *Frontiers in systems neuroscience*, vol. 9, p. 184, 2016.

[138] N. Hatami, Y. Gavet, and J. Debayle, "Classification of time-series images using deep convolutional neural networks," in *Tenth International Conference on Machine Vision (ICMV 2017)*, vol. 10696. International Society for Optics and Photonics, 2018, p. 106960Y.

[139] S. Haq and P. Jackson, "Speaker-dependent audio-visual emotion recognition," in *Proceedings Int. Conf. on Auditory-Visual Speech Processing (AVSP'08), Norwich, UK*, Sept. 2009.

[140] N. Marwan, J. Kurths, and S. Foerster, "Analysing spatially extended high-dimensional dynamics by recurrence plots," *Physics Letters A*, vol. 379, no. 10, pp. 894 – 900, 2015.

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

[142] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[143] Y. Sun, G. Wen, and J. Wang, "Weighted spectral features based on local hu moments for speech emotion recognition," *Biomedical signal processing and control*, vol. 18, pp. 80–90, 2015.

[144] K. F. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[145] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[146] M. Bernstein, V. D. Silva, J. C. Langford, and J. B. Tenenbaum, "Graph approximations to geodesics on embedded manifolds," 2000.

[147] H. Zha and Z. Zhang, "Isometric embedding and continuum isomap," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 864–871.

[148] D. L. Donoho and C. Grimes, "Image manifolds which are isometric to euclidean space," *Journal of Mathematical Imaging and Vision*, vol. 23, no. 1, pp. 5–24, Jul 2005.

[149] R. Pless, "Image spaces and video trajectories: Using isomap to explore video sequences." in *ICCV*, vol. 3, 2003, pp. 1433–1440.

[150] V. Silva and J. B. Tenenbaum, "Sparse multidimensional scaling using landmark points," 01 2004.

[151] V. D. Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," in *Advances in Neural Information Processing Systems 15*. MIT Press, 2003, pp. 705–712.

[152] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

[153] L. Cayton and S. Dasgupta, "Robust euclidean embedding," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 169–176.

[154] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *Journal of Machine Learning Research*, vol. 4, pp. 119–155, 2003.

[155] F. Sha and L. K. Saul, "Analysis and extension of spectral methods for nonlinear dimensionality reduction," in *Proceedings of the 22Nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: ACM, 2005, pp. 784–791.

[156] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, 2002, pp. 585–591.

[157] Z. Zhang and J. Wang, "Mlle: Modified locally linear embedding using multiple weights," in *Advances in neural information processing systems*, 2007, pp. 1593–1600.

[158] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM journal on scientific computing*, vol. 26, no. 1, pp. 313–338, 2004.

[159] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, 2003.

[160] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *International journal of computer vision*, vol. 70, no. 1, pp. 77–90, 2006.

[161] K. Q. Weinberger, B. Packer, and L. K. Saul, "Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization." in *AISTATS*. Citeseer, 2005.

[162] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, Mar. 1996.

[163] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[164] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in neural information processing systems*, 2002, pp. 585–591.

[165] ——, "Convergence of laplacian eigenmaps," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 129–136.

[166] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM J. Sci. Comput.*, vol. 26, no. 1, pp. 313–338, Jan. 2005.

[167] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[168] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[169] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *Journal of Global Optimization*, vol. 56, no. 3, pp. 1247–1293, Jul 2013.

[170] M. Avriel, *Nonlinear programming: analysis and methods*. Courier Corporation, 2003.

[171] R. Hooke and T. A. Jeeves, "" direct search" solution of numerical and statistical problems," *J. ACM*, vol. 8, no. 2, pp. 212–229, Apr. 1961.

[172] G. E. Box, "Evolutionary operation: A method for increasing industrial productivity," *Applied statistics*, pp. 81–101, 1957.

[173] V. J. Torczon, "Multidirectional search: a direct search algorithm for parallel machines," Ph.D. dissertation, Rice University, 1989.

[174] J. J. E. Dennis and V. Torczon, "Direct search methods on parallel machines," *SIAM Journal on Optimization*, vol. 1, no. 4, pp. 448–474, 1991.

[175] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[176] L. Dagum and R. Menon, "Openmp: an industry standard api for shared-memory programming," *IEEE computational science and engineering*, vol. 5, no. 1, pp. 46–55, 1998.

[177] V. Torczon, "On the convergence of the multidirectional search algorithm," *SIAM journal on Optimization*, vol. 1, no. 1, pp. 123–145, 1991.

[178] G. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, vol. 2, no. 2, pp. 205–224, 1965.

[179] E. Bruni, N. K. Tran, and M. Baroni, "Multimodal distributional semantics," *J. Artif. Int. Res.*, vol. 49, no. 1, pp. 1–47, Jan. 2014.

[180] F. Hill, R. Reichart, and A. Korhonen, "Simlex-999: Evaluating semantic models with (genuine) similarity estimation," *Computational Linguistics*, 2015.

[181] C. Baziotis, N. Pelekis, and C. Doulkeridis, "Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, August 2017, pp. 747–754.

[182] T. Yang, J. Liu, L. Mcmillan, and W. Wang, "A fast approximation to multidimensional scaling," in *In Proc. of the IEEE Workshop on Computation Intensive Methods for Computer Vision*, 2006.

[183] K. Rajawat and S. Kumar, "Stochastic multidimensional scaling," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 2, pp. 360–375, 2017.

[184] E. Pothos and J. Busemeyer, "A quantum probability explanation for violations of symmetry in similarity judgments," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 33, no. 33, 2011.

Appendix A

Abbreviations

(ACF): Autocorrelation Function
(AI): Artificial Intelligence
(AMI):Average Mutual Information
(ANN): Artificial Neural Network
(ASD): Autism Spectrum Disorder
(ASR):Automatic Speech Recognition
(AWVL): Average White Vertical Length
(BLSTMs): Bidirectional Long Short Time Memory units
(CNNs): Convolutional Neural Networks
(CTC): Connectionist Temporal Classification
(DAEs): Denoising Autoencoders
(DBN): Deep Belief Network
(DDP): Difference of Difference of Periods
(DENTR): Diagonal Entropy, recurrence quantification analysis measure
(DET): Determinism, recurrence quantification analysis measure
(DNNs): Deep Neural Networks
(EEGs): ElectroEncephalogram Signals
(ELMs): Extreme Learning Machines
(EM): Expectation Maximization algorithm
(EP): Emotion Profile
(ESR): Ensemble Softmax Regression classifier
(FNN): False Nearest Neighbors heuristic method
(GFS): Glottal Flow Spectrogram
(GMMs): Gaussian Mixture Models
(GN): Global Normalization
(GPS): General Pattern Search
(GPU): Graphical Processor Unit
(HMIs): Human-Machine Interfaces
(HMM): Hidden Markov Model
(HNR): Harmonics to Noise Ratio frame level feature
(IEMOCAP): Interactive Emotional Dyadic Motion Capture database
(ISOMAP): Isometric Feature Mapping nonlinear dimensionality reduction algorithm
(KNNs): K-Nearest Neighbors nonparametric classifier
(L): Average Diagonal Length, recurrence quantification analysis measure
(LAM): Laminarity, recurrence quantification analysis measure
(LE): Laplacian Eigenmaps
(LLDs): Low Level Descriptors extracted from speech frames
(LLE): Local Linear Embedding
(LOSO): Leave One Session Out evaluation scheme
(LR): Logistic Regression classifier
(LSP): Line Spectral Pairs

(LSTM): Long Short Time Memory unit
(LTSA): Local Tangent Space Alignment
(MDS): Multidimensional Scaling algorithm for nonlinear dimensionality reduction
(MFBs): Mel Filterbanks, frame level feature
(MFCCs): Mel Frequency Cepstral Coefficients frame level feature
(ML): Machine Learning
(MSE): Mean Squared Error loss function
(NLDR): Nonlinear Dimensionality Reduction task
(NN): Neural Network
(PCA): Principle Components Analysis dimensionality reduction algorithm
(PS): Phase Space or state space of a system under analysis
(RBF): Radial Base Function kernel
(RMS): Root Mean Square loss function
(RNNs): Recurrent Neural Networks
(RP): Recurrence Plot
(RQA): Recurrence Quantification Analysis, complexity measures extracted from a recurrence plot
(RR): Recurrence Rate
(SAE): Sparse Auto-Encoders
(SAVEE): Surrey Audio-Visual Expressed Emotion Database
(SD): Speaker Dependent evaluation scheme
(SER): Speech Emotion Recognition task
(SHS): Sub-Harmonic Sum
(SI): Speaker Independent evaluation scheme
(SP): Spectrogram
(SUA): Sub-Utterance Attention for a recurrent neural network
(SVD): Singular Value Decomposition
(SVM): Support Vector Machine classifier
(TD): Typically Developing children
(TEO): Teager Energy Operator
(TSNE): t-Distributed Stochastic Neighboring Embedding
(TT): Trapping Time, recurrence quantification analysis measure
(UA): Unweighted Accuracy performance metric
(VAD): Voice Activity Detection mechanism
(VENTR): Vertical Entropy, recurrence quantification analysis measure
(WA): Weighted Accuracy performance metric
(WENTR): White Vertical Entropy, recurrence quantification analysis measure
(WPA): Weighted Pooling Attention mechanism for recurrent neural networks
(WSFHM): Weighted Spectral Features based on Hu Moments
(ZCR): Zero Crossing Rate