NEW DIMENSION REDUCTION TECHNIQUE FOR BRAIN DECODING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ARMAN AFRASIYABI

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

SEPTEMBER 2015

Approval of the thesis:

NEW DIMENSION REDUCTION TECHNIQUE FOR BRAIN DECODING

submitted by **ARMAN AFRASIYABI** in partial fulfillment of the requirements for the degree of **Master of Science in Biomedical Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Işık Hakan Tarman _____
Head of Department, **Biomedical Engineering**

Prof. Dr. Fatoş Tünay Yarman Vural _____
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Uğur Halıcı
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Fatoş Tünay Yarman Vural
Computer Engineering Department, METU _____

Prof. Dr. Gerhard Wilhelm Weber
Institute Of Applied Mathematics, METU _____

Assoc. Prof. Dr. Ilkay Ulusoy Parnas
Electrical and Electronics Engineering Dept., METU _____

Assist. Prof. Dr. Tolga Çukur
Electrical and Electronics Engineering Dept., Bilkent University _____

Date: _____

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

Name, Last Name: ARMAN AFRASIYABI

Signature :

ABSTRACT

NEW DIMENSION REDUCTION TECHNIQUE FOR BRAIN DECODING

Afrasiyabi, Arman
M.S., Department of Biomedical Engineering
Supervisor : Prof. Dr. Fatoş Tünay Yarman Vural

September 2015, 93 pages

A new architecture for dimension reduction, analyzing and decoding the discriminative information, distributed in function Magnetic Resonance Imaging (fMRI) data, is proposed. This architecture called Sparse Temporal Mesh Model (STMM) which consists of three phases with a visualization tool. In phase A, a univariate voxel selection method, based on the assumption that voxels are independent, is used to select the informative voxels among the whole brain voxels. For this purpose, one of feature selection methods namely one way analysis of variance (ANOVA) or mutual information (MI) is employed. Then, in phase B, a multivariate voxel selection method, based on the multivariate form of the brain, known as recursive feature elimination (RFE) is employed. The last phase, phase C, contains two parts. In phase C.1, a local mesh with fix size around each voxel called *seed voxel* is formed. Next, the relationships, called arc weights, between the *seed voxel* and the neighbouring voxels are estimated. In phase C.2, ANOVA feature selection method is used to eliminate the unnecessary arc weights. Additionally, a visualization tool known as t-Distributed Stochastic Neighbor Embedding (tSNE) is used to analyse the effect of each phase. The results indicate that STMM can successfully use for brain decoding purpose.

Keywords: Sparse Temporal Mesh Model (STMM), Brain Decoding, Univariate Voxel Selection, Multivariate Voxel Selection, tSNE, fMRI

ÖZ

BEYİN OKUMA İÇİN YENİ BİR BOYUT KÜÇÜLTME TEKNİĞİ

Afrasiyabi, Arman
Yüksek Lisans, Biyomedikal Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Fatoş Tünay Yarman Vural

Eylül 2015 , 93 sayfa

Bu çalışmada, Fonksiyonel Manyetik Rezonans Görüntüleme(fMRG) verileri üzerinde boyut küçültme, analiz ve ayrımcı bilgileri deşifre işlemlerini gerçekleştiren yeni bir yapı önerilmiştir. Seyrek Zamansal Örgü Modeli (SZÖM) adı verilen bu yapı üç aşama ve bir görselleme aracından oluşmaktadır. Yapının A aşamasında, voksellerin birbirinden bağımsız olduğu varsayımına dayanarak, tüm voksellerin arasından bilgilendirici olanları bulmayı amaçlayan tek değişkenli voksel seçim modeli kullanılmıştır. Bu amaç için, tek yönlü varyans analizi (VA) veya karşılıklı bilgi (KB) yöntemlerinden yararlanılmıştır. Daha sonra, B aşamasında, beynin çok değişkenli yapısına dayanarak, Özyinelemeli Boyut Eliminasyon (ÖBE) olarak bilinen çok değişkenli voksel seçim yöntemi kullanılmaktadır. Son aşama olan C aşaması ise kendi içerisinde iki alt aşamadan oluşmaktadır. C.1 alt aşamasında, belirlenmiş ve tohum voksel adı verilmiş vokseller etrafında sabit boyutlu, bölgesel örgü modelleri kurulmaktadır. Kurulan bölgesel örgü modeller neticesinde, tohum vokseller ile komşu vokselleri arasında yay ağırlıkları adı verilen ilişkiler kestirilmektedir. C.2 alt aşamasında, VA özellik seçim yöntemi kullanılarak gereksiz yay ağırlıkları ortadan kaldırılmaktadır. Çok aşamalı bu yapının yanısıra, t-Dağıtılmış Stokastik Komşu Gömme (tSKG) isimli görüntüleme yöntemi her aşamanın etkisini analiz etmek için kullanılmatadır. Elde edilen sonuçlar, geliştirilmiş olan Seyrek Zamansal Örgü Modeli'nin, beyin okuma amacı için başarılı bir şekilde kullanılabileceğini göstermektedir.

Anahtar Kelimeler: Seyrek Zamansal Örgü Modeli (SZÖM), Beyin Okuma, Tek Değişkenli Voksel Seçimi, Çok Değişkenli Voksel Seçimi, tSKG, fMRG.

To Fatoş Tünay Yarman Vural and My Family

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor Prof. Dr. Fatoş Tünay Yarman Vural for all the support and encouragement she gave me. I have been amazingly fortunate to have Fatoş hoca as my advisor who gave me the freedom to explore on my own and at the same time the guidance to recover when my steps faltered. Without her guidance and precious support, it would not be possible to conduct this research.

My sincere thanks also goes to Assoc. Prof. Dr. İlkay Ulusoy Parnas, who provided me an opportunity to join her research group and recommended me to Prof. Dr. Fatoş Tünay Yarman Vural. My thanks also go out to the support I received from Dr. Mete Ozay and Assist. Prof. Ilke Öztekin.

I am very grateful to my colleagues Itır Önal, Hazal Mogultay, Burak Velioglu, Emre Aksan, Orhan Fırat, Sarper Alkan, Baris Nasir and Günes Sucu. I really value the knowledge and insight you have, and your willingness to share it with me.

For the non-scientific side of my thesis, I particularly want to thank my dearest friends Shahrokh Rahimi, Yousef Rahimi, Nemat Sorayyaband, Arash Ebrahimi and Abbas Abbasov for their motivation and friendship.

My deepest gratitude goes to my family for their unflagging love and unconditional support throughout my studies and my life. I would like to express my heart-felt gratitude to my dearest mom, Simin and my dearest dad, Esmail who has aided and encouraged me throughout this work. I am also indebted to my dearest mother-in-law Ziba, my dearest father-in-law Rostam, my precious brother Amir and brother-in-law Yahya my kind sisters Narmin and Roya for their support and care which helped me overcome setbacks and stays focused on my graduate study. My cute nephew Sana, thank you for your video calls during this work.

Last but not the least, I would like to thank my lovely wife Veedaa for standing beside me throughout this adventurous journey.

I acknowledge the support of TÜBİTAK (The Scientific and Technological Research Council of Turkey) by the project number 112E315.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

fMRI	function Magnetic Resonance Imaging
STMM	Sparse Temporal Mesh Model
TMM	Temporal Mesh Model
ANOVA	Analysis Of Variance
MI	Mutual Information
RFE	Recursive Feature Elimination
tSNE	t-Distributed Stochastic Neighbor Embedding
LMM-TM	Local Mesh Model with Temporal Measurements
MVPA	Multi-variate Pattern Analysis
KNN	K-Nearest Neighbors
SVM	Support Vector Machines

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

Although the human brain seems to be far from a complete uncovering, some of the abstract physiological and functional questions are partially answered about it. The results of current studies are beheld to the previous curiosities about the brain. Perhaps, the oldest interest backs to four century B.C. At that era, Aristotle considered the brain as a place for spirit circulation and a secondary organ which provides cooling and heating for the body. In one of his famous writings he says: *" There is nothing in the intellect that is not in the senses "*. However, Alexandrian anatomists, such as Rufus of Ephesus, worked on the general physical description, and they found some basic building blocks of the brain such as pia and dura matters. Avicenna (980–1037), the Islamic philosopher and medical writer, wrote in the early eleventh century that human brain is housed in the " faculty of fantasy ", " receiving all the forms which are imprinted on the five senses ".

Several centuries later, Gustav Theodor Fritsch (1838-1927) and Julius Eduard Hitzig (1838-1907) were two German physiologists who are counted as the pioneers on the mapping of the brain cortex with electrical stimulation of several animals, specially that of dogs. This work inspired the British physician David Ferrier (1843-1928) to stimulate the cortex of dogs and monkeys. The resulted map uncovered 29 functionally different parts of the cortex. However, with the advent of new technologies such as brain imagining, the researchers could be able to have a better understanding of the brain. One of the latest evolved neuroimaging technologies is functional

form of Magnetic Resonance Imaging (fMRI). fMRI provides sequential information about brain volumes, and it contributes to the uncovering of the brain functions by providing its images in high special resolution. Several characteristics of fMRI such as non-invasiveness, generation of dimensional and high special resolution images, bring popularity to fMRI, nowadays.

In order to examine the data acquired by fMRI, different approaches are proposed. Among them the most popular one is the prediction of the brain states under the " brain decoding " or " mind reading " research area. The ultimate goal of this approach is to predict what the subject thinks at the time of data acquisition using fMRI. It is expected that the most of unknown brain functions will be understood by achieving the ultimate goal of brain decoding. Additionally, decoding the brain would help science to understand the branch from both neuroscientific and medical point of views. On the one hand, brain decoding helps us to develop the effective methods which are based on the brain. For instance, effective Artificial Intelligent (AI) algorithms with high precision and speed can be developed by mimicking the brain functions. On the other hand, based on the findings of brain decoding we can design and build effective devices that provide comfortable life to diseased people.

The biggest challenge of brain decoding is the identification of the brain functions during a cognitive task. In order to achieve this challenge, one possible solution is to consider the voxels of brain as features, where they intensity values vary across different a cognitive stimuli. Then, a machine learning algorithm such as support vector machine (SVM) can be trained using that voxels. In this case, the feature space is formed by a set of voxels. Therefore, it is obvious that the selection of informative voxels is critical in order to increase the power of brain decoding. Additionally, the examination of all voxels (in the order of tens of thousands) to predict the brain states (in the order of hundreds) seems inconvenient. The source of this inconvenience stems from the fact that only limited parts of brain are responsible for a certain cognitive task. Additionally, taking into count of all voxels increases the possibility of incorrect prediction in machine learning methods. The source of this incorrectness is the family of problems which fall into the sub-category of the main problem known as " curse of dimensionality ".

In order to reduce the effects of curse of dimensionality problem, several methods are proposed in the literature under the voxel selection research area. In this work, two different approaches have been used for voxel selection purpose. In the first type, the voxels are considered to be independent from each other, and the selection is done by considering each voxel as a independent random variable. These methods are called univariate voxels selection methods. Two most popular univariate voxel selection methods are analysis of variance (ANOVA) and mutual information (MI), and both of them are used in this study. The other approach based on the examination of all voxels to select the most discriminative ones. The methods in this sub category fall into multivariate voxel selection group. Recursive Feature Elimination (RFE) is one of the popular multivariate methods, and it is also employed in this thesis.

In addition to voxel selection, there is another problem with the classical brain decoding approaches. We know that voxels as the fMRI elements contain the neural information of a group of neurons. Furthermore, it is obvious that neurons have a strong interactions with each other. Therefore, considering only the intensity values of voxels omits the relations among them. Therefore, building a robust model based on the relationships among the voxels is required to represent the nature of brain. It is expected that building a brain network could help us to find a brain decoding model with a high accuracy. In this network, the voxels can be taken as vertices, and their relationships can be considered as edges. Learning a mesh model is offered for the first time by M. Ozay et al. [59], and it is developed under the network assumption. Temporal mesh model (TMM), which is proposed by Onal et al. [67], is the later version of learning mesh model. We used TMM in this study. Similar to its old version, TMM estimates the edges between the voxels and use them instead of voxel intensity for brain decoding purpose. These edges are called arch weights, and the number of them is higher than the number of voxels. In other words, the new feature space is made up of estimated arc weights in TMM, and this feature space is larger than the previous one which is made up of voxel intensities. As a result, most probably, TMM suffers from curse of dimensionality problem. The detailed information of the this problem is given in Chapter 2.

1.2 Proposed Reduced Dimension Network Architecture: Sparse Temporal Mesh Model (STMM)

The proposed brain decoding architecture is called Sparse Temporal Mesh Model (STMM). STMM is aimed to increase the precision of brain decoding by reducing the curse of dimensionality problem using feature selection methods beside the use of powerful brain decoding method known as temporal mesh model (TMM). In brain decoding using fMRI recordings, the change in the intensity values of voxels are measured while brain is stimulated by watching a scene, remembering something, hearing a piece music and etc. The features are considered to be the voxels, and the samples are the time series obtained at each voxel during a stimulus. Therefore, during a stimulus a set of brain volumes (voxels) are recorded for each sample. The number of stimulus determines the number of samples in the set. This set of brain volumes (voxels) and samples are used to train one of the machine learning classifiers. Then, the trained classifier is tested by the samples which are not used for training purpose. Unfortunately, there is a problem which lays in the nature of fMRI recordings. The brain volumes captured in the fMRI recordings consist of 20,000 to 100,000 voxels, and the brain stimulus is in the order of hundred. In machine learning literature, this problem is known as curse of dimensionality . In order to reduce the dimension of the feature space there are dozens of voxel selection methods in the literature for brain decoding purpose [49], [30] and [31].

M. Ozay et. al, [59] illustrated that, when the peak intensity of voxels are replaced with the estimated relationships among them which are called arc weights, then the arc weights have more discriminative power compared to the peak intensity of voxels. Upon this finding, mesh learning model is proposed which estimates and replaces each voxel called *seed voxel* by the relationships of that voxel with its nearest neighbouring voxels. Later, Onal et. al, [67] developed the updated version of local mesh model which uses the discrete form of voxel's intensity changes instead of just single peak value. In this thesis, the name temporal mesh model (TMM) implies Local Mesh Model with Temporal Measurements (LMM-TM).

Due to the computational complexity and curse of dimensionality problem, the previous studies of using arc weights could only be applied in a predefined anatomical

region. This approach omits the discriminative voxel from the ignored parts of the brain. Furthermore, all of the voxels from the selected anatomical regions are considered which may cause of assumed to be achieved which results in considering unnecessary and less discriminant features (voxels).

In order solve the above problems, an architecture called sparse temporal mesh model (STMM) is proposed. STMM uses feature selection methods in the first and second phases of proposed architecture (STMM) for voxel selection purpose. Mutual information (MI) and one way analysis of variance (ANOVA) based voxel selection methods are used to select the most stimulus related features(voxels) among the whole brain voxels. In this study, it is shown that both MI and ANOVA select nearly the same voxels, and both methods take the voxels from the anatomical regions of brain which are believed to be related with the experimental task. Recursive Feature Elimination (RFE) is used in the second phase of the architecture. Contrary to the MI and ANOVA feature selections that are univariate, RFE selects voxels by considering all of them in the multivariate form. The purpose of this phase is to select voxels by considering the multivariate nature of brain. The third phase consists of two parts: implementation of temporal mesh model (TMM) and pruning the arc weights. After voxels are selected as the output of the second phase, they feed to TMM for arc weights estimation. In the second part of third phase, the arc weights are pruned using discussed ANOVA feature selection methods. Finally, a visualization method known as t-Distributed Stochastic Neighbor Embedding (tSNE) [32] is connected to all of the previous phases to provide 2D map of the samples in the feature space and visualize the effects of the phases on the feature space.

The fMRI dataset used in this work is a binary type visual stimulation of participant while measuring the brain functions with fMRI. The stimulus are belonging to the images of flowers and birds categories. During the experiment the subjects are asked to recognize the category of presented stimulus. The experiment is done in six runs, and each run contains 36 stimulations (samples) of participants.

In summary, a new architecture called sparse temporal mesh model (STMM) is proposed. The architecture is made up three phases. The whole brain voxel inputs to the architecture and the ones with high discriminative power are selected in the phases

A and B. In phase C, after implementation of TMM on the selected voxels, the estimated arc weights are pruned by using ANOVA feature selection method. The impact of each phase on feature space is visualized using tSNE method. Additionally, SVM and KNN are used to decode and predict the states of brain.

1.3 Contribution

- Previously, the implementation of TMM was limited to the extraction of few anatomical regions of the brain which are responsible for generating cognitive task of visualizing objects. One problem with this approach was that the voxels outside of the region of interest (ROI) are ignored to consider for brain decoding. Additionally, considering all of the voxels within the ROI was another limitation of previous approach. Because, in Neuroscience literature, it is shown that the specific anatomic regions do not fully active under a stimulus. In other words, the selected predefined RIOs may have inactive voxels. In this study, two univariate and a multivariate feature selection methods are combined to select the most informative voxels in the entire brain. The results indicate that although most of the selected voxels are from the functionally task related regions of the brain, the selected voxels may fall the outside of the predefined ROIs. Additionally, the performances of brain decoding get better by the elimination of irrelevant feature (voxels) compared to the the whole brain performances.

- In previous researches on mesh learning model, the number of neighbours for each voxels was either optimized or determined to be a fix number. In other words, the mesh learning model were only examined by considering fix number of arc weights for all of the voxels. Considering this assumption which all of the voxels are connected to each other with the same degree may contradict with the nature of brain. Most probably, the connectivity among the voxels can vary depending on the cognitive process and the location of the voxels. On possible solution for this problem is to optimize the number of related neighbours for each voxel. Obviously, such optimization problem is unsolvable due to the high number of voxels (in the order of tens of thousands) and different degree

of unknown interactions. In this study, a pruning method for the arc weights is proposed as an alternative solution. ANOVA feature selection method is used in order to achieve this goal. The results declare that the feature selection methods successfully capture and select the discriminative arc weights.

- In the literature, several dimension reduction techniques are used to visualize the data points in the feature space. Among these methods, tSNE was shown to have a better performance compared to the other methods in some cases such as hand writing recognition. In this study, tSNE is used to visualize the samples in the feature space and see the effect of voxel selection methods, TMM and discussed proposed solution in the previous paragraph. tSNE helps us to have better knowledge about the feature space of each phase in the proposed architecture and see its effect.

- The main goal of this study is to increase the power of brain decoding by using the voxel selection methods and mesh learning model. The experimental results of the proposed architecture showed that the results are repeatable, and the proposed architecture is a promising technique for brain decoding task.

1.4 Outline of the Thesis

Chapter 2 covers a brief literature survey about the brain decoding and dimension reduction techniques. After discussing the idea and basic fundamentals of fMRI, two popular brain decoding methods known as MVPA and local mesh model are overviewed. Next, one of the serious problems known as "curse of dimensionality" and some of its effects are discussed. Then, several possible solutions are explained. Finally, information about KNN and SVM classifiers are presented, which both of them are used in this study.

Chapter 3 introduces the proposed new dimension reduction technique for brain decoding. This chapter contains the detailed information about the employed methods in the architecture. In the first part of the chapter, two univariate voxel selection methods namely mutual information (MI) [30] and ANOVA [31] based feature selection

methods are explained. Then, a multivariate voxel selection method known as Recursive Feature Elimination (RFE) [49] is covered. The third section presents Temporal Mesh Model (TMM) which estimates the arc weights as relationships between a *seed voxel* and its local neighbours. Finally, the visualization of the feature space by tSNE is provided.

Chapter 4 illustrates the analysis and experimental results of the voxel selection methods and proposed architecture. It consists of several analyses, such as the anatomical location of selected voxels, degree of the connectivity for each voxel etc. Additionally, the classification performance of proposed method and the other ones are discussed in the chapter.

Finally, in Chapter 5, the overall outcomes of the study and possible future works are discussed.

CHAPTER 2

BRAIN DECODING AND DIMENSION REDUCTION TECHNIQUES FOR FMRI

In this chapter, an overview about the current state of art for "brain decoding" and related problems are provided. After discussing fundamentals of functional Magnetic Resonance Imaging (fMRI), the current state of art on brain decoding techniques are covered which connects researchers from several discipline under the scope of fMRI data analysis. Then, two different approaches for decoding the brain states are discussed namely Multi-Voxel Pattern Analysis (MVPA) and Mesh Learning model. In MVPA, the brain states are estimated using the intensities of the voxels. On the other hand, the estimated linear relations between voxels are replaced with the intensities in the case of mesh learning model. Then, the problem known as "curse of dimensionality" is provided that steams from the nature of high dimension and low sample of fMRI datasets. Finally, some of the possible solutions in the literature for curse of dimensionality is discussed.

2.1 functional Magnetic Resonance Imaging (fMRI) for data acquiring

One of the important physical phenomenon which has a huge impact on the science was the discovery of Nuclear Magnetic Resonance (NMR). NMR was the result of innovative and fundamental works of famous scientists such as Walter Gerlach (1889 – 1979), Otto Stern (1888 – 1969) and Isidor Rabi (1898–1988). In this event, the nuclei absorb and consequently emit electromagnetic radiation in a strongly enough magnetic field. In 1974, Paul C. Lauterbur and Peter Mansfield opened a new chapter

in the history of science using this phenomenon. Although they worked separately without the knowledge of each other, they were able to describe the application of gradients of magnetic fields for spatial localization of NMR signals-which were unknown before that time. This discovery called Magnetic Resonance Imaging (MRI) technique which was great enough to persuade the committee of Noble Prize in Physiology or Medicine to honor both scientists with the prize in 2003 [1].

The researches have been soared in MRI since 1970's and leaded to discovery of functional Magnetic Resonance Imaging (fMRI) by Seiji Ogawa[2]. fMRI is a neuroimaging technique which provides sequential information from the brain volumes over time which helps us to understand and study the dynamic changes of brain. Additionally, it is a technique which enables the researchers to study the brain activations in non-invasive and in vivo manner either when the brain is in its rest status or stimulated with a certain task. The sequential and 3D Magnetic Resonance Images (MRIs) come together to make the fMRI data that is made up of the elements known as voxels. Voxels are the elements of uniformly spaced volume which contain the data of a group of neural activities. A typical fMRI data is made up nearly 100,000 voxels that each of them shows the spatial distribution of the nuclear spin density in the form of intensity [4]. Without having an abstract knowledge of MRI as the base of fMRI, it not be possible to have true understanding it. Four fundamental units of MRI procedure are: magnet, gradient coils, radio frequency coils and computer system as shown in the Fig 2.1 (a).

1. Magnet

   A large magnet is employed by MRI machine that can produce a static magnetic field in the range of 3 to 4 Tesla (magnetic flux density unit). This is a quite high magnetic force compared to that of earth and even the polar field of sun which are 0.00005(T) and 0.0002(T) respectively [6], and it is strong enough to pick up a car. It is also strong enough to align the random spinning of hydrogen nuclei or protons in the body at the direction of the field and bring them to a status known as equilibrium state.

2. Gradient Coils

   Gradient Coils, electro-magnetic coils, are located in the middle layer of MRI scanner and generate a loud noises. This part enables the technicians to change the produced magnetic field by the first part (magnet) in the precise manner in both spatial and temporal manner by altering the magnetic strength. The result of this coil is the slice selection and localization ability in three dimensional space which is called special coding of magnetic resonance images. There are three coils which make this part, and each of them gives the three dimensional imaging ability to MRI along the x, y and z axises.

3. Radio Frequency Coils

   Radio frequency coils are the most inner part, and they are used to send a radio frequency or focused form of RF pulses into the scanner chamber. These coils are specifically designed for various parts of bodies such as knee, shoulder, wrist and head. This specification is done in order to increase the signal to noise ratio and have more diagnostic images. The introduced RF pulses to the aligned protons causes to stimulation and spin out them from equilibrium state; consequently, they forced against the magnetic field. At this time, a simply turn off the pulse (RF) gives the chance for detectors to detect the energy produced as the protons realign to the magnetic fields and back to their equilibrium state. The realignment time changes with respect to environment and the chemical nature of tissue.

4. Computer System

   The analogue to digital conversion of RF signal is done by a computer. Additionally, the computer system performs several image processing techniques to produce the final diagnostic images of the MRI.

While MRI techniques provide only the structural images of anatomical regions, fMRI generates the image of metabolic functions. [5]. It provides information about the neural activities that have the relationships with the physiological activities. In other words, fMRI is not a directional neuroimaging technique; instead, it follows the physiological changes in indirect way under the assumption of having correlations

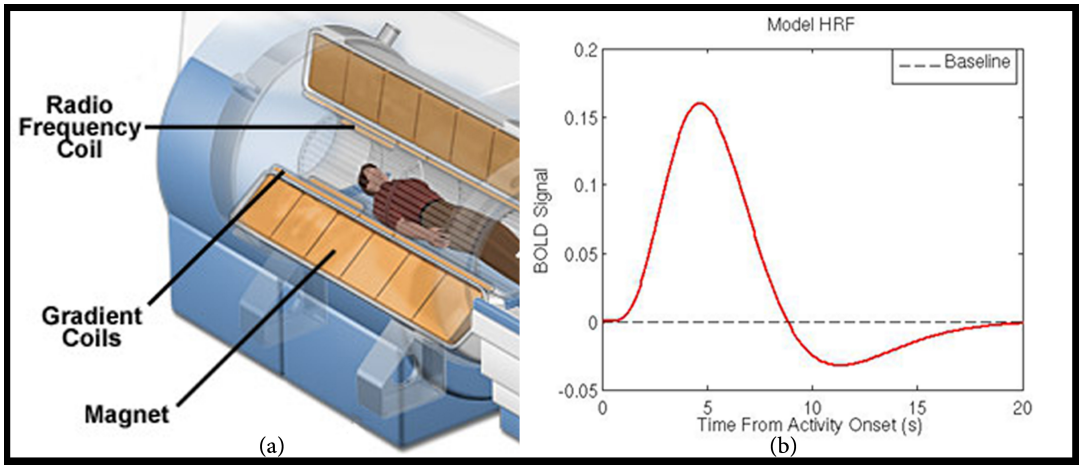

Figure 2.1: a) Three main components of Magnetic Resonance Imaging (MRI) b) the model of Hemodynamic Response Function

with neural activities. Interestingly, the origin of fMRI backs to the 1936 before the discovery of Nuclear Magnetic Resonance (NMR). In that year, Pauling and Coryell at [8] discovered an important magnetic characteristic of hemoglobin which is used as the base of later discovery of fMRI. In their research [9], they showed carbonmonoxyhemoglobin and oxyhemoglobin does not have unpaired electrons. Additionally, they found that ferrihemoglobin, the hemoglobin itself, contains four unpaired electrons. The unpaired property of oxyhemoglobin, hemoglobin combined with oxygen molecule, enables the molecule to have zero moment magnetic fields. On the other hand, the four unpaired electrons in the oxidized hemoglobin during the decomposition of the blood which is called ferrihemoglobin is paramagnetic. This means, oxidized hemoglobin has huge magnetic moment compared to the previous one. In 1982, Thulbron et al. [10] showed the variations of relaxation rate (T2) under the magnetic field of hemoglobin contained blood. However, the breakthrough discovery comes from 1990's when the measurable changes in blood oxygenation was detected at magnetic resonance signals. Using gradient-echo and under the strong magnetic fields (7 and 8.4 T), Ogawa et al. [11] [12] could produce the images of the mice brain and illustrated the existence of variations in the blood oxygenation at the activated areas of brain. The primary suggestion was that deoxyhemoglobin is the cause of increase by additional remove of oxygen from the blood during the activation of the monitored region. However, further researches showed the opposite pattern which was the decrease in the deoxyhemoglobin concentration with activation. The reason of this phenomenon was the large variations in the local cerebral blood flow (CBF).

This finding helps Turner et al. [14] to detect the small changes in MR signal which were related to the blood oxygenation in the cat's brain. The resulted small variation in MR signal due to the changes in oxygen level of hemoglobin is called Blood Oxygenation Level Dependent or briefly BOLD effect.

However, the big question was still remained unanswered: "what exactly was the causes of BOLD effect?" In fact, one should have enough knowledge of molecular and cell biology to answer this question. In its equilibrium state, the inside potential of a nerve cell is more negative compared to the outside of the cell membrane due to the existence of highly concentrated sodium ion (Na^+) at the outside of the membrane. The activation process occurs when a neuron is stimulated by another one in a series of chemical interactions. In fact, all of the exchanging processes in activation phase are downhill thermodynamically; that is, no energy is needed. However, the uphill processes are needed to back the molecules to their equilibrium state after the activation. Adenosine triphosphate (ATP) and adenosine diphosphate (ADP) are the main source of free energy which is used by neurons to do their recovery. In ATP/ADP system the glucose and oxygen molecules goes under the oxidative metabolism and change to carbon dioxide and water. Due to the fact that the brain does not have energy storage mechanism, the blood vessels play important role by taking both glucose and oxygen to the part of the brain which deals with a task. The neural activations cause the formation of a bound between deoxyhemoglobin and oxygen molecules [8]. The occurrences of this chemical bounding decrease ratio of deoxyhemoglobin to oxygenated hemoglobin form in the blood [15]. This decrease has adverse effects on the MR signals and leads to the increase of signals in the activated brain regions. However, this contrast is captured by fMRI which is known as Blood Oxygenated Level Dependent (BOLD) contrast.

Whole of exchange processes is done by Hemodynamic Response (HR) that is instantaneous delivery of blood-contained glucose and oxygen in this case- to the activated neural region. Generally, the pick of BOLD signal remains up to 4-6 seconds after the onset of neural activities, then it backs to its baseline within the 8 to 12 seconds after the onset. The dynamic patterns of signal shown at the voxel level are known as the voxel's Hemodynamic Response Function, or HRF [17]. An important factor in the fMRI studies relies on the estimation of HR by HRF [16]. A HRF example is demonstrated in Fig 2.1 (b), and a common modelling of this would be a sum of

two Gamma distributions. While one distribution models the initial peak, the other models the undershooting of the BOLD signal [17].

2.2 Brain Decoding: Multi-Voxel Pattern Analysis

The advent of functional Magnetic Resonance Imagining (fMRI) which offers the BOLD signal caused to exponentially growth of interest and curiosity among researchers. Several scientific areas of studies come together to make multidisciplinary research groups to understand and interpret the fMRI data. Perhaps, the major motivation to us as engineers comes from the curiosity of building a model that can decode the brain and predict its states accurately[29]. Finding such model does not just help the neuroscientific society, but the others such as engineers as well. The model like this, for instance, would help computer sciences to generate efficient Artificial Intelligence (AI) algorithms that are based on the decoded brain. Although making a perfect and complete model seems to be a fiction nowadays, the progress in this subject increases exponentially.

The starting point of interests in application of Artificial Intelligence (AI) algorithms on brain data backs to 1990's where some researches such as Kippenhan at [19] and N. Morch at [20] applied neural network classifier on PET scan datasets. The majority of early work is based on the detection and recognition of specific diseases such as Alzheimer. However, after the discovery, fMRI has been providing new opportunities for researchers by offering more precise data compared to techniques such as electroencephalogram (EEG). The early work on fMRI is about following the magnitude of the variations in the BOLD signal at different areas of brain with different task paradigms, and these studies fall into voxel-wise analysis. In these studies, the goal was not to extract the differences in the activation patterns of voxels [21]. General Linear Model (GLM) was the fundamental base of these studies, and the goal was to recognize and find the meaningful variations of voxels by using the statistical threshold and averaging. However, this approach had some limitations and needed to be addressed. The major disadvantage of voxel-wise analysis was missing the useful information and patterns in the relationship between the voxels. As an example, it was impossible to decode the fMRI data obtained from the brain regions related to

the visual cortex in the study shown by Kamitani and Tong [22]. The reason of this was that the scale of orientation was very small compared to the number of voxels in the fMRI data [21].

In order to solve the problems of uni-variate voxel analysis, the method known as Multi-Voxel Pattern Analysis (MVPA) is proposed for analysis of data acquired by (fMRI). Unlike the uni-variate voxel analysis which was based on the focusing on the individual voxels, the idea of MVPA is to capture and decode multi-voxel patterns in the fMRI data by implementation of advanced and powerful pattern classification algorithms. This outstanding idea was the starting point in the understanding of the most complex and unknown organ (the brain). One of the pioneered studies was done by Haxby et al. [24]. In this study, which a part of it shown in Fig. 2.2, he could distinguish the various cognitive states using MVPA, and he was able to illustrate the different patterns among voxels within Ventral Temporal (VT) cortex. In the other study, Tom M. Mitchell et al. [25] were able to train the classifiers to decode the brain states in the fMRI data of subjects who had stimulated by looking to the pictures or reading ambiguous or non-ambiguous words.

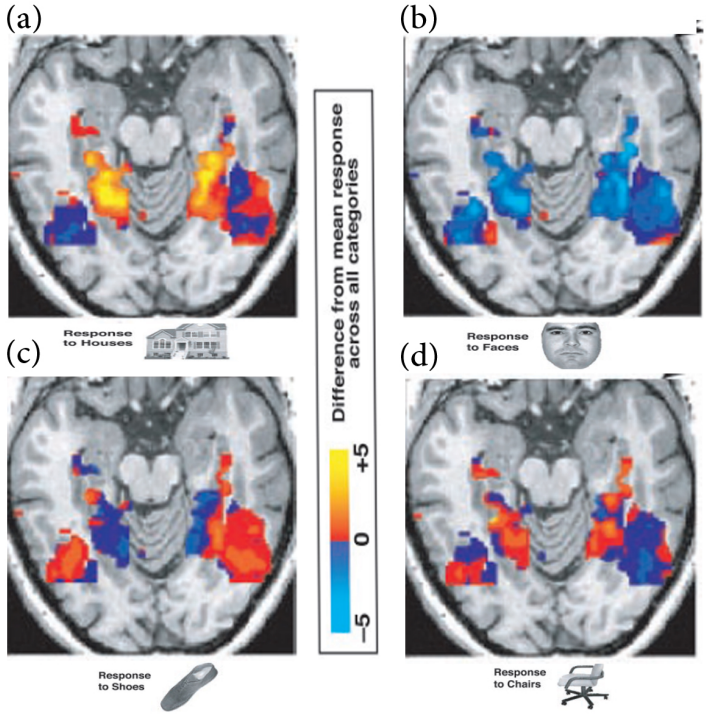

Figure 2.2: a) brain response to house picture stimulus, b) brain response to human face picture stimulus, c) brain response to shoes picture stimulus, d) brain response to chair picture stimulus.

MVPA does not have the averaging problem of uni-variate voxel methods; therefore, it increases the sensitivity in the recognition of cognitive states. Additionally, MVPA provides the examination of presence and absence of cognitive states upon the short period of time (few seconds). This approach increases the temporal resolution in the understanding of cognitive states. Additionally, MVPA gives the chance of vigorous analyses in the representation of cognitive states within the specific regions of brain [25].

Before any implementation of pattern-classification algorithms, there are two important steps: experimental design and preprocessing. The experimental design is done based on the physiological, neurological and psychological knowledge. Two general form of experiments are resting-state and task-based. In the first one, the aim is to find functional architecture of the brain when spontaneous low-frequency variations in BOLD signals are observed. Generally, this type of experiment is designed to distinguish the difference in brain activation pattern between normal and diseased people. In the task-based type, more common in brain decoding, the subject is stimulated while the BOLD signal is recorded. The ultimate goal of task-based studies is to uncover and predict the different states of brain [26]. However, the preprocessing step which is done on the output of experimental fMRI data is crucial, and it has direct impact on the later MVPA results. This step falls into three phases [27]:

1. Realignment of Images

   In realignment phase, spatial transformation is implemented in order to align the time series of images to correct the possible movement artifices during the data acquisition.

2. Co-Registration

   After realignment, the functional data co-registration to structural data in order to maximize the mutual information between them.

3. Smoothing

   Smoothing involves convolving the 3-dimensional fMRI signal by a low pass Gaussian filter. Researchers disagree about this part. Some of them, such as K. A. Norman et al. [25], states that smoothing may possibly decrease some important spatial patterns and information. On the other hand, Kriegeskorte et

al. [28] examine the effect of smoothing and states that a smoothing with right parameters is a necessary for denoising and reducing the effect of salt paper noise.

4. Anatomic Region Selection

   This step of preprocessing is based on the experimental study of the brain. If a researcher is interested to analyse the specific region(s) of the brain, then toolboxes such as MARSBAR toolbox [29] can be used to select the region of interests (ROI's). In this thesis, the fMRI data is analysed by using wholistic approaches.

Apart from the experimental design and preprocessing, in general, the Multi-Voxel Pattern Analysis- MVPA falls into four steps: Feature Selection, Pattern Assembly, Learning the Hypothesis and Validations [25]:

- Feature/Voxel Selection

  Probably feature selection is the most important step in MVPA. Feature selection involves the search for the most discriminative voxels. The methods for feature selection will be elaborated in the next section.

- Pattern Assembly-training and test matrices

  In this step, the time series of voxels which are selected in the previous section is concatenated to form the input vectors of a classifier. For example, two matrices for training and test are formed were rows and columns indicate the samples and voxels, respectively.

- Learning the Hypothesis-training the classifier

  In this step, one of pattern-classification algorithms such as k-Nearest Neighbors (KNN) or Support Vector Machines (SVM) is trained using the training matrix and related label vector.

- Validation and Measuring the generalization performance

  Finally, the validity of the trained classifier is measured using the test metrics and related label vector. Additionally, the generalization performance which is the ability of trained classifier to correctly predict the unseen is measured in this phase.

2.3 Brain Decoding: Mesh Model

For the first time, M. Ozay et al. [59] observed that changes in the intensity value of neighbouring voxels are larger than the variation of row voxel intensities across the classes. Based on this finding, a new learning method known as mesh learning Model is proposed instead of MVPA. The goal of mesh learning Model is to overcome the representation problem of brain patterns in MVPA of fMRI data sets. This method is similar to searchlight which considers the BOLD signal of a voxel and its neighbourhood. However, unlike searchlight which discriminative voxels are found based on the combination of all voxels signal in the predefined region, Mesh Leaning method models the neural activity of neighbouring voxels which subsequently fed to the classifiers. In general, this approach falls in to five steps [59]:

1. The optimum number of neighbouring voxels, p-nearest neighbor, is found.
2. A star topology is made around the seed voxel and nearest p neighbor voxels.
3. Least square estimation method is used to compute the weights between the seed voxels and its p nearest neighbours.
4. The new features are made by concatenation of weight vector which are called Mesh Arc Descriptor (MAD). The size of MAD is $N \times p$, where p is the size of mesh and N is the number of active voxel.
5. Classification and validation are measured by some of well-known algorithms such as Support Vector Machines (SVM), k Nearest Neighbors (KNN) or etc.

The first core of Mesh Learning Model which is developed by Ozay was called "Local Mesh Model". In this model, voxel at time t and in location j are shown in the form of $v(t_i, \bar{s}_j)$ where, $j = 1, 2, ..., M$ and $i = 1, 2, ..., N$ where N and M represents the number of samples and voxels respectively. Therefore, around each seed voxel $v(t_i, \bar{s}_j)$, a star mesh is formed which is shown by $\{v(t_i, \bar{s}_j)\}_{k=1}^{p}$. Although, Ozay et al. [59] used special metric to determine the neighbourhood, Firat et al. [60] used functionality criteria for neighbourhood specification.
In the local mesh model case, the arc weights $a_{i,j,k}$ are the edges which connects the

vertexes that are composed of p nearest neighbours and seed voxel. However, the goal of local mesh model is to find out the arc weights $a_{i,j,k}$. and linear regression equation is used to estimate these arc weights as follows;

$$v(t_i, \bar{s}_j) = \sum_{\bar{s_k} \in \eta_p} a_{i,j,k} v(t_i, \bar{s}_k) + \varepsilon_{i,j} \ , \tag{2.1}$$

where, $\varepsilon_{i,j}$ is the estimation error of $a_{i,j,k}$ for the voxel $v(t_i, \bar{s}_j)$ of at time t_i. Then, Levinson Durbin recursion [66] is used to minimize the squared error of $\varepsilon_{i,j}^2$ as follow;

$$\varepsilon_{i,j}^2 = \left(v(t_i, \bar{s}_j) - \sum_{\bar{s_k} \in \eta_p} a_{i,j,k} v(t_i, \bar{s}_k) \right)^2 . \tag{2.2}$$

The estimated arc weights for each voxel are concatenated to make a $1 \times p$ arc vector in the form of;

$$\bar{a_i} = [a_{i,j,1}, a_{i,j,1}, ..., a_{i,j,p}] \ ,$$

similarly, $\bar{a_i}$ are concatenated to make the sample vector A_i in the form of;

$$A_i = [\bar{a_{i,1}}, \bar{a_{i,2}}, ..., \bar{a_{i,M}}] \ ,$$

Finally, the sample vectors are combined to each other to make the feature space F which is equal to;

$$F = [A_1^T, A_1^T, ..., A_M^T] \ .$$

which is in the format of $N \times (p \times M)$ for each subject.

2.4 Curse of Dimensionality: Low Sample − High Dimension Obstacle

One of the fundamental problems in pattern recognition and machine learning is analysing the feature space in high dimension. In this problem, the dimension of the feature space is very high compared to the number of samples. Richard E. Bellman, American mathematician [1920-1984] coined the term “curse of dimensionality" for this problem [61], [62]. Although there are several approaches to solve this problem,

finding a robust solution is not a trivial task. The problem manifest itself in our case, where the number of samples is very low (in the order of hundreds) compared to the number features or voxels (in the order of thousands). However, there are many negative effects of "curse of dimensionality" which three well-known of them are: computational complexity, overfitting and sparse volume problems.

2.4.1 Computational Complexity

Computational complexity of an algorithm is proportional to the dimension of feature space, and it is determined by the number of mathematical operations such as addition and subtraction. In order to clarify, the complexity issue, Gaussian prior based maximum likelihood estimation, bellow is analysed as follows [63];

$$g(x) = -\frac{1}{2}(x-\hat{\mu})^t \hat{\sum}^{-1}(x-\hat{\mu}) - \frac{d}{2}ln2\pi - \frac{1}{2}ln|\hat{\sum}| + lnP(w). \tag{2.3}$$

In the above formulation, O would be,:

$$\begin{aligned}
\hat{\mu} &= O(dn), \\
\hat{\sum}^{-1} &= O(nd^2), \\
\tfrac{d}{2}ln2\pi &= O(1), \\
\tfrac{1}{2}ln|\hat{\sum}| &= O(d^2n) \text{ and} \\
lnP(w) &= O(n).
\end{aligned}$$

where, d represents the dimension. Note that as dimension increases the time complexity also increases.

2.4.2 Overfitting

In addition to the computational complexity, "curse of dimensionality" problem is also the cause of another serious problem, called overfitting. In an overfitted model, the space is divided according to the noise samples instead of real ones (shown in the Fig. 2.3). Subsequently, the generalization performance which is defined as the performance of a model on the unseen samples is decreasing, when it is overfitted. [64].

2.4.3 Sparse Volume

"Curse of dimensionality" problem is also the source of space sparsity which occurs when most of feature space is empty. It means, as dimension increases, the amount of obtained samples became small compared to the space volume. Therefore, the space volume became sparse as a result of dimension increase [65].

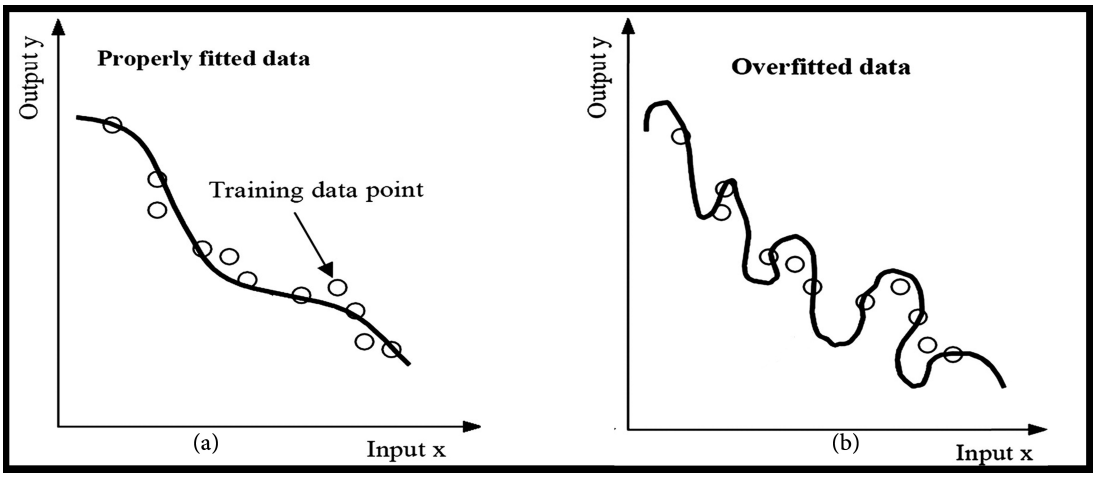

Figure 2.3: a) the correct fitted predictive model, b) the overfitted model.

In order to solve the overfitting and sparsity problems, we require a robust dimension reduction method. In fact the curse of dimensionality lays in the nature of fMRI datasets where the number of samples (in couple of hundreds) is very low compared to the dimension (couple of thousands). Therefore, finding a way to select the number of effective voxels, is very important. In the following sections the popular dimension reduction methods, will be overviewed.

2.5 Dimension Reduction using Feature Selection Methods

Designing a feature space with a set of discriminative samples is a very important task in pattern recognition. There are a variety of ways to solve this problem. The available methods are categorized in two main groups based on their output feature space: feature selection and feature extraction methods. The goal of feature selection methods is to maintain the original feature space while the elimination of less discriminative features. For example, a popular feature selection method is the Recursive Feature Elimination (RFE). On the other hand, mapping the original space into

another space is done in the feature extraction methods. The classical method called Principle Component Analysis (PCA) is considered in these group of methods. In another categorization, the feature selection methods fall into two groups based on their mechanism in the fMRI research. First, univariate feature selection methods, which voxels are analysed independent of each other to determine their predictively power [50]. They are based on different approaches such as univariate statistical tests and information theoretic criteria. Second, multivariate voxel selection methods which discards the voxels by analysing the group of them. In the following sub-sections we provide a brief overview for these methods.

2.5.1 Analysis of Variance(ANOVA)

One-way ANOVA F-test is commonly used univariate statistical test as a feature selection method [51]. This statistical test is used to distinguish and measure the impact of an even such as stimulation of brain (which is shown in the class label vector) on the intensity change of a voxel. The hypothesis is based on the meaningful variance between the class label vector and a feature vector which both of them counts as independent random variables. Hypothesis is a statement or claim about a property of an object, and hypothesis test is a procedure in standard format to test the claim about that property. ANOVA based on the statistics between a feature and class label vector which is computed by the equation [52];

$$F(v_j) = \frac{(Sum\ of\ Squares\ Between\ Groups)/(related\ degree\ of\ freedom)}{(Sum\ of\ Squares\ Within\ Groups)/(related\ degree\ of\ freedom)}\ , \tag{2.4}$$

where $F(v_j)$ is the F-value of jth voxel. F-Value which is computed for all of the voxels, is used to rank them and then select the most informative ones.

2.5.1.1 Mutual Information based Voxel Selection

Similar to ANOVA, Mutual Information (MI) based feature selection is the other univariate method that can be used to rank voxels. In this case, the mutual information of feature V_j is calculated with respect to class labels vector Y [31]. Generally, MI is used to quantify the statistical dependency of two random variables. The assumption

of this method is to consider each dimension and class label as two random variables. Therefore, MI can be used to measure the statistical dependency of them. High value of MI shows more information between the j^{th} voxel (V_j) and class conditions Y or label vector [54], [55], [56]. The mutual information between class label vector Y and feature vector V_j is defined as;

$$\mathbf{MI}(Y; V_j) = \sum_{y=1}^{C} \int_{V_j} p(y, v_j)\ log \frac{p(y, v_j)}{p(y)p(v_j)}\ , \tag{2.5}$$

where,

C is the number of class conditions, and $C = 2$ in our case.

$p(y, v_j)$, the short-handed of $(Y; V_j)$, is the joint distribution of variables of Y and V_j. Here, the scoring function of MI based feature selection ranks voxels with respect to their $MI(y, x)$ values.

2.5.2 Recursive Feature Elimination as Multivariate Method

Contradict to the univariate approaches, the multivariate methods base on the consideration of features dependency. Among the multivariate feature selection methods Recursive Feature Elimination (RFE) is the most popular method [49]. RFE based on Support Vector Machine (SVM) employs the generalized nature of SVM in order to rank the voxels according to their contribution in the classifiers obtained model. It also falls into the feature selection methods category known as wrapper methods [48]. In this method, first, SVM is trained using a fMRI training set. Next, the obtained model provides us a weights of features in the model's hyperplane. Then, the voxels are ranked according to their contributions in this weight vector obtained after minimizing the objective function of SVM [49];

$$minimize\ \ J(w) = \frac{1}{2} w^T w + a \sum_{ni}^{l} \zeta_i\ , \tag{2.6}$$

where, $\zeta_i, i = 1, ..., T_{trian}$, are slack variables that counts on the errors of training set and a is a positive real constant. After ranking, the voxels are eliminated by using a threshold. Next, the same discarded voxels from the training set are also eliminated from the test set. Finally, the training and test sets are used to train and measure the performance of the main classifier.

2.6 Dimension Reduction using Feature Extraction Methods

As discussed above feature extractions methods can also be used to reduce the dimension. The feature selection methods which only to select the subset of features in the original feature space, the feature extraction methods change the original d dimension feature space into the new feature space with dimension r, where $r < d$. In this thesis, the dimension reduction technique called t−Distributed Stochastic Neighbor Embedding (t−SNE) is used for dimension reduction purpose. By using tSNE, we are interested to visualizing the feature space.

2.6.1 t-Distributed Stochastic Neighbor Embedding (tSNE)

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a updated version of Stochastic Neighbor Embedding (SNE) [32] which has several superiorities over it. It is easier to optimize and decrease the problematic tendency of crowding points at the center of final map. Additionally, it keeps the original structure of high dimensional data in the low dimension map which resulted in better performance of both visualization and dimension reduction [33] [34] [35].
Imagine objects $x_1, ..., x_n$ in a very high dimension space like shown in the Fig. 2.4, and we want to see the exact manifold or feel the true arrangement of objects. Perhaps, making a two or three dimensional map of the high dimensional space is the fundamental goal in visualization of the data. In other words, we want to build a map of points $y_1, ..., y_n$ such that the similar objects $x_i's$ in high dimensional space to be represented by nearby points $y_i's$ in final map. To achieve this goal, the original structure of data should keep in the final map, but this is not a trivial problem to be solve. Almost all of the algorithms fail to keep the ideal structure. However, the idea of t-SNE is to minimize an objective function to reach the goal of keeping original structure. This objective function should be able to measure the discrepancy between similarities of the objects in the original space and transferred space.

PCA as a traditional method finds the linear projection of the original data points in the map such that the variance of projected data is maximized. However, most

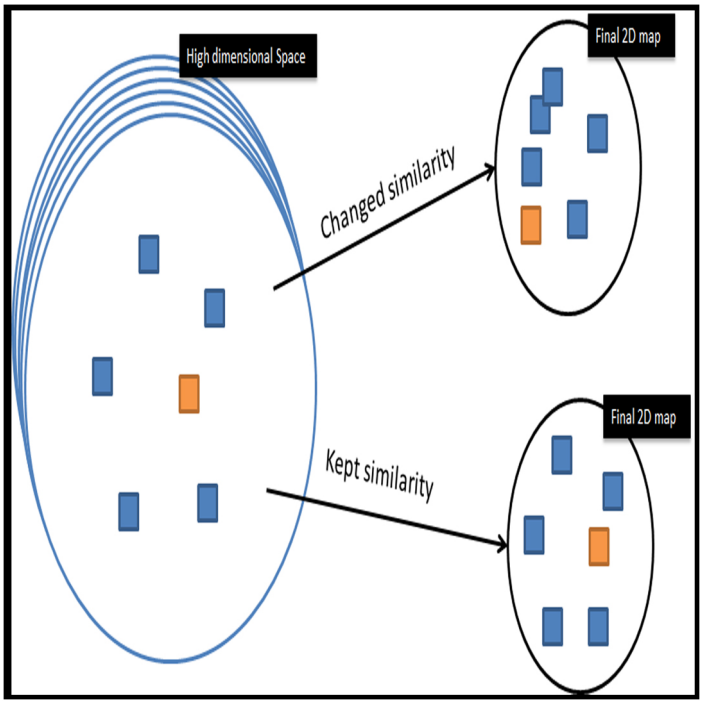

Figure 2.4: The goal of non-linear dimension reduction algorithm tSNE maintains the original structure of the data in the mapped low dimension space.

of real world data have non-linear manifold. Therefore, PCA fail to preserve the similarity of the original data. After understanding the problem, researchers come up with some nonlinear techniques such as Isomap and Locally Linear Embedding (LLE). Isomap maintains the global non-linear geometry of the data by preserving the geodesic distance which is the shortest route between two points on the surface of manifold [36]. LLE is another non-linear technique which collapses bunch of data into a single point, but this is problem in the maintainers of similarity in the original data[37]. Although, these techniques are better than PCA, they still are not well enough in some cases. The general idea of tSNE is given in the following algorithm:

Algorithm 1: t-Distributed Stochastic Neighbor Embedding (tSNE)

1 **for** *(all pairs of different objects i and j)* **do**

2 Calculate the similarity of objects x_i and x_j using joint probability $p_{i,j}$ in the high dimensional input space.

3 Measure the similarity of points y_i and y_j using $q_{i,j}$ in the output space.

4 Minimize the difference between $p_{i,j}$ and $q_{i,j}$.

tSNE measure the local similarity of objects in high dimensions within the nearby points. Imagine the red point shown in the Fig. 2.4 to be an object in the high dimensional space. Particularly, it tries to convert the high-dimensional Euclidean distance between objects into the conditional probability which represents the similarity.

2.7 Classifiers

k-Nearest Neighbors (kNN) and Support Vector Machine(SVM) algorithms are used for classification purpose in thesis. Although several other algorithms are also used in the literature for brain decoding purpose, we choose SVM and kNN to measure the performance of the implemented dimension reduction techniques. The reason of this choice was the successfulness of SVM and KNN in the high dimensional feature space.Similar to several other classifiers, kNN and SVM are shown to have reasonable performances in high dimensional feature space [31]. In this section, we are going to discuss these classifiers.

2.7.1 k-Nearest Neighbors(kNN)

kNN is a simple and very efficient classifier which is used in the most of the studies. This algorithm is based on the Euclidean distances between samples. Imagine, there is a $m - dimensionl$ feature vector with an arbitrary sample s, and lets $f_r(s)$ be the r^{th} attribute of sample s as follow [44],[45];

$$< f_1(s), f_2(s), ..., f_m(s) > ,$$

In this case the distance between s_1 and s_2 is defined as

$$dist(s_1, s_2) = \sqrt{\sum_{r=1}^{m} (f_r(s_1) - f_r(s_2))^2} \ . \tag{2.7}$$

The sample is mapped to a class label in C where $C = \{c_1, c_2, ..., c_n\}$ using target function $f : R^m \longrightarrow C$. The classification of new sample s_q is done by Algorithm 1 which gives the class label.

Algorithm 2: k Nearest Neighbors algorithm

1 **for** *(s_q does not classified)* **do**

2 let $s_1, s_2, ..., s_3$ be k nearest samples to the s_q from the training examples

3 $\hat{f}(s_q) \longleftarrow \max_{v \subseteq V}(\sum_i^k = \delta(v, f(s_q)))$ where $\delta(a, b) = 1$ if a=b and $\delta(a, b) = 0$ otherwise

2.7.2 Support Vector Machine (SVM)

SVM is anther well-known classifier. In this study, SVM architecture is used for the Recursive Feature elimination (RFE) of fMRI data, where the weight vectors are used to rank the voxels. Lets overview the basics of SVM as suggested by Vladimir Vapnik [46].

Support Vector Machine (2 class case)

Support vector machine (SVM) is a supervised learning algorithm which finds the hyperplane that maximizes the margin of the separating place between two classes. Consider a binary linear classification problem with labels $y\{-1, 1\}$ and features x. Then, the problem is to find w and b parameter such that;

$$h_{w,b} = g(w^t x + b) \ , \tag{2.8}$$

where,

$$g(z) = \begin{cases} +1 & \text{if z} \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

In order to, solve this problem, SVM employs both functional and geometric margin information. The functional margin would be formalized as;

$$\hat{\gamma} = \gamma(i)(w^t x + b) \ . \tag{2.9}$$

In the linear case, functional margin has a confidence problem because changing w and b values will not effect $h_{w,b}$, and it would only change the sign of $h_{w,b}$. On the other hand, geometric margin can be defined as the distance between decision boundary and samples. It depends on w and b values, along with the vector w. The

geometric margin can be formalized as follows;

$$\hat{\gamma} = \gamma(i)((\frac{w}{\|w\|})^t x^{(i)} + \frac{b}{\|w\|}) \ . \tag{2.10}$$

Given a training set,

$$S = \{(x^{(i)}, y^{(i)}) \ ; i = 1, 2, ..., m\} \ .$$

In order to reach this goal, it is necessary to find smallest geometric margin with respect to S. Therefore, the geometric margin is defined as,

$$\hat{\gamma} = minimize\{ \ \gamma^{(i)} \ ; i = 1, 2, ..., m\} \ . \tag{2.11}$$

As a result, geometric margin fits properly to the training data. Therefore, SVM finds a decision boundary which maximizes the geometric margin to make a confident classification. Under the assumption of linearly separable data set, the hyperplane which partitions the samples into two regions can be formalized by optimal margin classifier in the following form;

$$\min_{\gamma,w,b} \frac{1}{2}\|w\|^2 \ , \tag{2.12}$$

With the following constraint;

$$y^{(i)}(w^t x^{(i)} + b) \geq 1 \ ; i = 1, 2, ..., m \ .$$

The points with the smallest margin α_i (three points in the example below) are called support vectors. It is expected that the number of support vectors is smaller than the size of training set. The constraint can be reformulated to formalise the optimal margin classifier as following;

$$g_i(w) = y^{(i)}(w^t x^{(i)} + b) + 1 \leq 0 \ . \tag{2.13}$$

In order to satisfy the above inequality, Lagrange multipliers are employed as follows;

$$L(w, b, a) = \frac{1}{2}\|w\| - \alpha_i[y^{(i)}(w^t x^{(i)} + b) - 1] \ . \tag{2.14}$$

Equivalently, a dual optimization problem can be formulated as follow;

$$\max_{\alpha} \; W(\alpha) = \sum_{i=1} \alpha_i - \sum_{i,j=1} \frac{1}{2} y_{(i)} y_{(j)} \; \alpha_i \; \alpha_i \langle x_{(i)}, x_{(j)} \rangle \; , \tag{2.15}$$

Subjected to the following constraints;

$$\alpha_i \geq 0 \; ; i = 1, 2, ..., m,$$
$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \; .$$

The algorithm of SVM for the linearly separable datasets satisfy eq.2.20. However, the data can be non-separable where the dimension of the feature space is high. In order to build the algorithm that can work for non-linearly separable datasets, we need to change the algorithm as follows (considering functional margin less than 1);

$$\max_{\gamma, w, b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i, i = 1, 2, ..., m, \tag{2.16}$$

Subjected to the following constraints;

$$y^{(i)}(w^t x^{(i)} + b) \geq 1 - \xi_i, \; i = \{1, 2, ..., m\},$$

$$\xi_i \geq 0, i = 1, 2, ..., m.$$

After solving the above set of equations, the SVM classifier predicts "1" if $w^t x + b \geq 0$ and "-1" otherwise. The decision boundary is given by the line $w^t x + b = 0$.

2.8 Summary of the Chapter

In this chapter, first a brief fundamentals of functional Magnetic Resonance Imaging (fMRI) is discussed. Then, two well-known method known as multi-voxel pattern analysis (MVPA) and mesh learning model are explained which are trying to model the brain activities and function under the brain decoding concept. Next, the famous problem of machine learning techniques in this research area known as "curse of dimensionality " and its effects on the fMRI analysis is overviewed. In order to solve this problem and reduce the destructive effects of "curse of dimensionality", several popular feature selection methods in machine learning is provided in the section 2.5 and 2.6. Finally, two well known classifiers which are used in this thesis, namely kNN and SVM are explained in the last section .

CHAPTER 3

A NEW DIMENSION REDUCTION ARCHITECTURE FOR BRAIN DECODING

In the previous chapter, the need for dimension reduction in the fMRI data sets is explained. Some feature selection methods are discussed to reduce the dimension and solve the "curse of dimensionality" problem for brain decoding purpose. All of the dimension reduction methods are almost subjective and situation based. In other words, they work properly in the specific situations and conditions. As an example, if the input feature space contains unnecessary features, then the output space map would be less discriminative in the case of feature extraction methods such as tSNE. Therefore, finding out a suitable and generic architecture which takes the advantages of dimension reduction methods would be beneficial for brain decoding.

In this chapter, an architecture called Sparse Temporal Mesh Model(STMM) is proposed which combines previously discussed methods. The schematic overview of the proposed architecture is shown in Fig. 3.1. At the top layer of this architecture, in phase A, a univariate feature selection method, is used to eliminate the noisy and destructive voxels. The goal of this step is to eliminate the less informative voxels in order to increase the accuracy and speed of the later phases. In phase B, the most discriminative voxels are selected by using multivariate feature selection method which considers the dependency of voxels on each others. Then, the temporal mesh model (TMM) is applied on the selected voxels to estimate their relationships known as arc weights. This phase is followed by a feature selection method to prune the "unnecessary" arc weights in terms of brain decoding. In each phase, the performance map is

measured by using popular learning algorithms, namely SVM and KNN classifiers. Additionally, in each phase, the data in the feature space is visualized in 2D using tSNE.

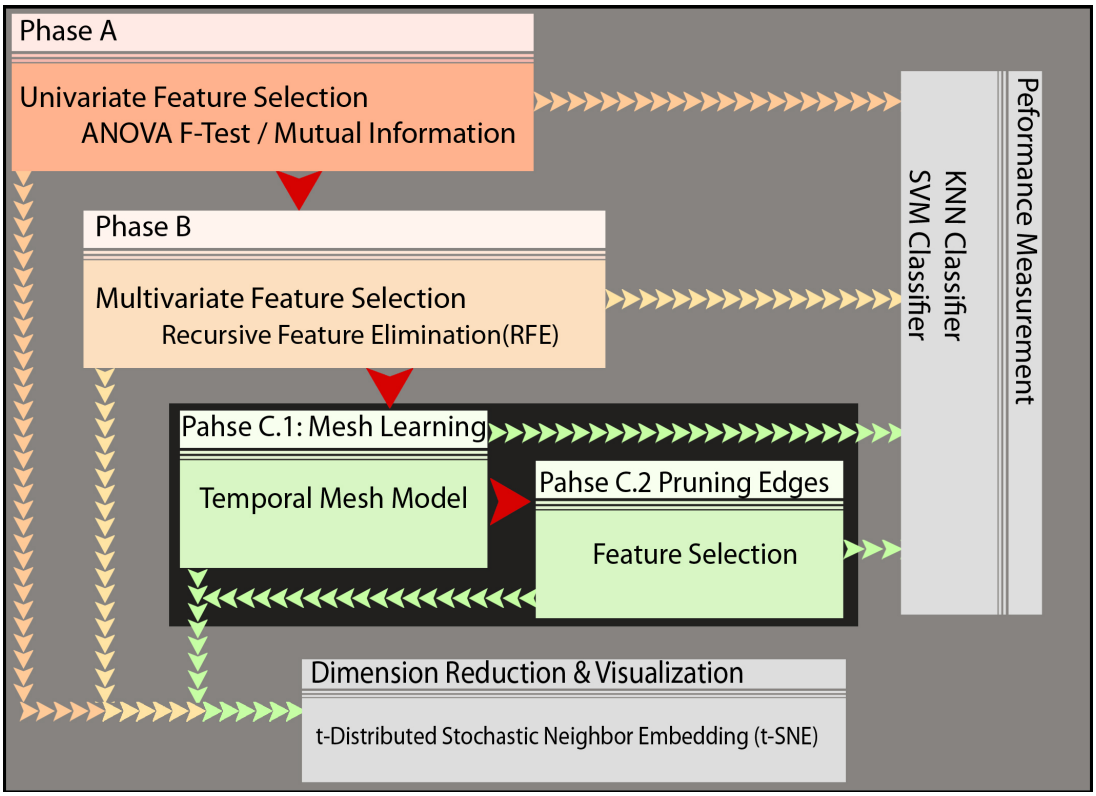

Figure 3.1: The abstract scheme of the proposed architecture known as sparse temporal mesh model (STMM) as the backbone of the thesis. After elimination of the noisy features in the phase A, a multivariate feature selection method is used to eliminate less discriminative features/voxels in the phase B. Then, Temporal Mesh Model (TMM) is applied to the selected voxels, and it is followed by pruning of useless arc weights in the Phase C. Finally, tSNE is used to reduce the dimension in order to visualize the feature space.

3.1 Phase A. Univariate Voxel Selection

It seems that there is a trade off in the application of univariate feature selection methods for brain decoding purpose. On the one hand, we know that voxels interact with each other in a cognitive task. This contradicts, with the nature of univariate voxel selection methods. The source of this paradox comes from the hypothesis of univariate methods which analyse voxels independently. On the other hand, it is believed that only a fraction of the voxels contribute to the formation of a cognitive

process. Therefore, the dependencies of voxels are limited, and a voxel is not affected by all of the other voxels. Additionally, most of the advanced multivariate feature selection and extraction methods fail to reduce the dimension properly in the case of noisy input feature space. Furthermore, the advanced methods suffer from high time complexity because of high dimensional input space. Therefore, univariate feature selection methods could be effective as a initial step to increase the precision and to speed up the further brain decoding processes. In this thesis, two types of univariate methods; namely, analysis of variance (ANOVA) and mutual information (MI) based feature selection methods, are used for voxel selection purpose.

Figure 3.2 shows the schematic overview of dataset obtained from fMRI. The dataset is a $M \times N$ matrix. M represents the the number of samples or brain response to the stimulations. N represents the number of fMRI elements or voxels which considered as features in brain decoding problem.

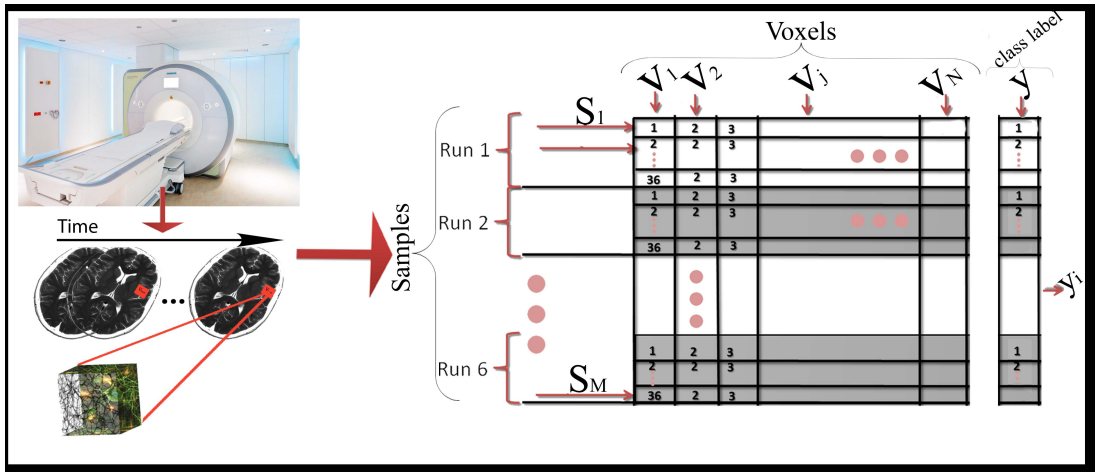

Figure 3.2: The abstract scheme of obtained dataset from fMRI machine which is used for brain decoding purpose. The rows of matrix consists of samples, and the columns are made of the voxels. The class label y is a vector which consists of the labels of samples.

3.1.1 Analysis of Variance (ANOVA) F-Test based feature selection

In one way analysis of variance ANOVA, *F*-test, both voxel vector v_j and class label vector y are considered as two independent random variables, and the impact of y on the v_i is measured by the *F*-value using Algorithm 3.

In summary, after calculation of individual sum of squares, the within group sum of square (SS_{wg}) is calculated for both class conditional class label vector y_i and the j^{th}

voxel v_j. Then, the total sum of square (SS_T) is computed by concatenating of y_i and v_j. Next, the between group (SS_{bg}) sum of square is calculated by using both within and total sum of squares. Mean square values of between and within groups are obtained by diving both between and within sum of squares to the related degree of freedoms. Finally, the *F*-value for v_j is computed by the dividing the between to within the mean squares. After calculating *F*-value for all of the voxels, they rank according to their *F*-value. Finally, top ranked voxels are picked as the selected voxels [30].

3.1.2 Mutual Information based feature selection

As the second univariate voxel selection method, mutual information (**MI**) between the class label vector y where $y = \{1, ..., C\}$ and the j^{th} voxel v_j is considered as a feature selection criterion. Mathematically, $\mathbf{MI}(y; V_j)$ is defined as;

$$\mathbf{MI}(y; v_j) = \sum_{y=1}^{C} \int_{V_j} p(y, v_j)\ log \frac{p(y, v_j)}{p(y)p(v_j)}\ . \tag{3.1}$$

where,
C is the number of different class conditions in label vector y.
$p(y, v_j)$ is the joint distribution of the random variables of y and v_j.

In convenient form, $p(y, v_j)$ can be calculated using chain rule, as follow;

$$p(y, v_j) = p(y)p(v_j|y). \tag{3.2}$$

$p(v_j|y)$ in equation (3.2) can be estimated by a kernel based technique, called Parzen-Rosenblatt window method,

$$\hat{p}(v_j|y) = \left(\sum_{j=1}^{M} \delta_y(v_j)\ \ \kappa\Big(\frac{v_j - v_{ij}}{w}\Big) \right) \Bigg/ \left(w \sum_{j=1}^{M} \delta_y(v_j) \right), \tag{3.3}$$

where,

δ_y is the kronecker delta function in the form of;

$$\delta_y(v_j) = \begin{cases} 1 & \text{if x = y} \\ 0 & \text{otherwise} \end{cases}\ .$$

Algorithm 3: ANOVA based Voxel Selection algorithm

Input: *DATA* matrix in the size of $M \times N$ where N is the number of voxels, and the number of voxels to be selected, V_{no}

1: **for** $j = 1$ to N **do**

2: $v_j \longleftarrow the\ j^{th}$ voxel of DATA matrix

3: Calculate the squared sum (SS) of differences between each condition in (y) and its mean as follow;

$$\text{SS}_y \longleftarrow \sum(y_i - \mu_y)^2$$

where y is class label vector and $y = \{y_1, ...y_M\}$. μ_y represents the mean of class label vector y.

4: Calculate the squared sum (SS) of differences between each sample in (v_j) and its mean μ_{v_j} as follow;

$$\text{SS}_{vj} \longleftarrow \sum(v_{ij} - \mu_{v_j})^2$$

where v_{ji} is the i^{th} sample of j^{th} voxel.

5: Calculate the sum of squares within the group "SS_{wg}"; SS_{wg} = SS_y +SS_{vj}

6: Concatenate y and v_j into a single vector X;

$X \longleftarrow (y\ ;\ v_j)$; where $X = \{x_1, ..., x_i, ..., x_{2M}\}$ and has $2 \times M$ elements.

7: Calculate the total sum of squares "SS_T";

$$SS_T \longleftarrow \sum(x_i - \mu_X)^2$$

where μ_X is the mean of X.

8: Calculate sum of squares between the groups;

$$SS_{bg} \longleftarrow SS_T - SS_{wg}$$

9: Calculate the degree of freedoms "df" for total, between and within groups (note: N_T is the total number of observations in X and k is the number of groups): $df_T \longleftarrow N_T$; $df_{bg} \longleftarrow k - 1$; $df_{wg} \longleftarrow N_T - k$

10: Calculate the relative mean-square values for between and within groups:

$$MS_{bg} \longleftarrow \frac{SSbg}{df_{bg}} \quad \text{and} \quad MS_{wg} \longleftarrow \frac{SSwg}{df_{wg}}$$

11: Calculate F Value as:

$$F_{vj} \longleftarrow \frac{MS_{bg}}{MS_{wg}}$$

12: **end for**

13: Sort all of the voxels according to their corresponding *F* values

$D_{sort} \longleftarrow \text{SORT}_{ANO.}(\text{DATA})$

14: **return** Matrix DATA_{sf} with selected voxels in the size of $N \times k$

$\kappa(.)$ is the kernel. Participially, $\kappa(.)$ is chosen to be Gaussian kernel.

w is the bandwidth, and w is standard deviation in the case of Gaussian kernel.

After estimation of conditional probability $\hat{p}(v_j|y)$ using Parzen-Rosenblatt window method, $p(y)$ and $p(v_j)$ in equation (3.3) can be estimated by marginalizing out X_j as follows;

$$p(y) = \int_{v_j} p(y)p(v_j|y). \tag{3.4}$$

and

$$p(v_i) = \sum_{i=1}^{C} p(y)p(v_j|y). \tag{3.5}$$

After calculating $\mathbf{MI}(Y; V_j)$ where $j \in \{1, .., N\}$ for the voxels, the voxels with high mutual information are assumed to be most informative ones. Therefore, they are selected as the output of this feature selection method. The pseudo code of discussed **MI** based feature selection is given in the following Algorithm 4.

3.2 Phase B. Multivariate Voxel selection using Recursive Feature Elimination (RFE)

After Phase A which employs the univariate voxel selection methods, the recursive feature elimination (RFE) method selects voxels by examining all of the input feature space. This method is a type of multivariate voxel selection methods. The schematic overview of RFE is shown in the Fig. 3.3. The support vector machine (SVM) classifier is trained over the training set $X = \{x_1, ...x_n\}$, and the obtained weight vector is used to rank the voxels.

For a linearly separable dataset SVM finds the following discriminative function with a bias term b;

$$g(x_i) = w.x_i + b\ , \tag{3.6}$$

and as discussed in the section 2.7.2 in binary case;

$$g(x_i) = \begin{cases} y_i = +1 & \text{if } g(x_i) \geq 0 \\ y_i = -1 & \text{otherwise.} \end{cases}$$

Algorithm 4: Mutual Information (MI) based Voxel Selection algorithm

Input: *DATA* matrix in the size of $M \times N$ where N is the number of voxels.

Input: Number of voxels to be selected, V_{no}

1: **for** $j = 1$ to N **do**

2: $V_j \longleftarrow the\ j^{th}$ voxel of DATA

3: Estimate conditional probability $\hat{p}_{(v_j|y)}$ using *Gaussian* kernel based Parzen-Rosenblatt Window method; $p_{(v_j|y)} \longleftarrow \hat{p}_{(v_j|y)}$.

4: Compute the probability of class condition $p(y)$ by marginalizing Vj; $p(y) \longleftarrow \int_{V_j} p(y)p(v_i|y)$

5: Compute the probability of v_i $p(x_j) \longleftarrow p(y)p(v_i|y)$ for all classes $\sum_{i=1}^{C} p(y)\ p(v_i|y)$

6: Compute the joint probability $p(y, x_j)$ $p(y, x_j) \longleftarrow p(y)\ p_(v_j|y)$

7: Compute the *MI* between y and v_j ***MI***(y, v_j) by importing $p(y, x_j)$, $p(c)$ and $p(v_j)$ in equation (3.1)

8: **end for**

9: Sort all of the voxels according to their corresponding *MI* values $D_{sort} \longleftarrow$ SORT$_{MI}$(DATA)

10: DATA$_{sf} \longleftarrow V_{no}$ of D_{sort}

11: **return** Matrix DATA$_{sf}$ with selected voxels in the size of $N \times k$.

However, the slack variables ζ_i is added for the non-linear datasets, and the discriminative function is changed to;

$$y_i(w.x_i + b) \geq 1 - \zeta_i \ , \tag{3.7}$$

where, $\zeta_i \geq 0$ and ζ_i calculates the deviation of data points from the optimal hyperplane (Vapnik, 1998). In order to achieve this task, SVM minimizes the following cost function;

$$\Phi(w, \zeta) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\zeta_i \ , \tag{3.8}$$

which is subjected to the constraint of (3.7).

At each step, RFE measures the weight vector w on the training set. This weight vector is interpreted as the contribution of a specific feature/voxel for each class in the obtained SVM model. However, the absolute value of the weight vectors w;

$$w \leftarrow |w_c|; \quad c = 1, ..., C,$$

is considered in the scoring function of i^{th} voxel S_{vi} for all classes C;

$$S_{vi} = \frac{\sum_{\forall c} w}{C} \ .$$

Finally, the voxels with high value of S_{vi} are selected in each iteration that is shown in Figure (3.3) as RFE Loop.

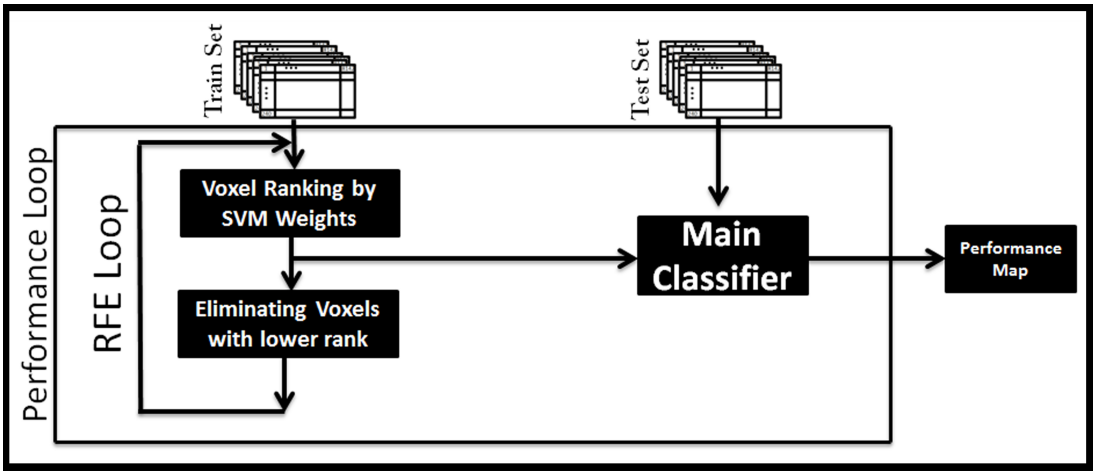

Figure 3.3: The abstract scheme of combining SVM and RFE. Voxels are eliminated iteratively through the process called recursive feature elimination (RFE) after they are ranked by SVM classifier.The main classifier trained by the output of RFE and tested by test set to obtain the performance maps.

3.3 Brain Decoding Using Temporal Mesh Model

After selecting the informative voxels, temporal mesh model (TMM) proposed by Onal et al. [60] is implemented on the selected voxels to increase the accuracy of brain decoding. TMM changes the input feature space (voxels are features) to the new one. The new features are defined as the linear estimation of a given voxel v_i by its neighboring voxels. The estimated new feature space by TMM is always larger than the input feature space. This may cause to curse of dimensionality problem due to the increase in the estimated feature space by TMM. Therefore, the use of dimension reduction methods could again be useful which is discussed in the second sub-section.

3.3.1 Temporal Mesh Model

As we discussed in the previous chapter, brain decoding using mesh model estimates the arc weights and use them as features instead of voxel intensities. These arc weights are estimated for each voxel by minimizing the objective function defined over a local mesh around each voxel. For this purpose, the following squared error $\varepsilon_{i,j}^2$ is minimized with respect to the arc weights $a_{i,j,k}$:

$$\varepsilon_{i,j}^2 = \left(v(t_i, \bar{s}_j) - \sum_{\bar{s}_k \in \eta_p} a_{i,j,k} v(t_i, \bar{s}_k) \right)^2 . \tag{3.9}$$

This model is developed on the discrete form of all hemodynamic response of a voxel instead of getting only the pick value. Onal et al. [60] extended equation (3.9) to define temporal mesh model (TMM).

In both case, however, one mesh is constructed around each voxel v_j, and the voxel v_j at the center of mesh is known as *seed voxel*. Euclidean distance between *seed voxel* and other local neighbouring voxels is used to define a Local Mesh Model with Temporal Measurement (LMM-TM) which we briefly call it temporal mesh model (TMM). Fig. 3.4 shows the red coloured seed voxel $\bar{r}(s_i, \bar{l_j})$ at $\bar{l_j}$ coordinate for the sample s_i which is connected to its neighbouring voxels (shown with blue color). $\bar{r}(s_i, \bar{l_j})$ contains the values of voxel intensities in the form of a vector and defined as:

$$\bar{r}(s_i, \bar{l_j}) = [v(s_i, t_1, \bar{l_j}), v(s_i, t_2, \bar{l_j}), ..., v(s_i, t_\tau, \bar{l_j})]^T. \tag{3.10}$$

where, t_τ represents the time at τ instance. Therefore, $\bar{r}(s_i, \bar{l_j}) \in R^\tau$ is made up of the voxel intensities that are measured at time t_τ and $w \in \{1, ..., \tau\}$. TMM based on the assumption that the arc weights between *seed voxel* and its *p-spatial nearest neighbours* can be estimated by a linear model in the form of:

$$\bar{r}(s_N, \bar{l_j}) = \sum_{\bar{l_b} \in \eta_p} a_{i,j,k} \bar{r}(s_i, \bar{l_b}) + \bar{\varepsilon}_{i,j} \ , \tag{3.11}$$

where the error vector $\bar{\varepsilon}_{i,j}$ has the following forms;

$$\bar{\varepsilon}_{i,j} = (\varepsilon_{i,1,j}, \varepsilon_{i,2,j}, ..., \varepsilon_{i,\tau,j}) \ ,$$

and each voxel has intensity components $\bar{r}(s_N, \bar{l_o})$. Additionally, the locally *p* nearest neighbours of the *seed voxel* at coordinates $\bar{l_b}$ is shown as $\bar{r}(s_i, \bar{l_b})$ where $\bar{l_b} \in \eta_p$.

Similar to the previous form of mesh model, TMM estimates the arc weights by minimizing the following expected square error;

$$E\Big(\varepsilon_{i,j}^2\Big) = \left(\Big(\bar{r}(s_N, \bar{l_j}) - \sum_{\bar{l_b} \in \eta_p} a_{i,j,k} \bar{r}(s_i, \bar{l_b}) + \bar{\varepsilon}_{i,j}\Big)^2\right) , \tag{3.12}$$

where,

$E(.)$ is the expectation function that is implemented on discrete form of hemodynamic response that is related to a stimulus period τ .

$$\begin{aligned}
&\bar{\varepsilon}_{i,j} = (\varepsilon_{i,1,j}, \varepsilon_{i,2,j}, ..., \varepsilon_{i,\tau,j}), \\
&\bar{r}(s_i, \bar{l_j}) = \Big(v(s_i, t_2, \bar{l_j}), v(s_i, t_1, \bar{l_j}), ..., v(s_i, t_\tau, \bar{l_j})\Big)^T, \text{ and} \\
&\bar{r}(s_i, \bar{l_b}) = \Big(v(s_i, t_2, \bar{l_b}), v(s_i, t_1, \bar{l_b}), ..., v(s_i, t_\tau, \bar{l_b})\Big)^T \ .
\end{aligned}$$

TMM uses the ridge regression as (3.13) estimates the arc vector $\bar{a}_{i,j} = [a_{i,j,1}, a_{i,j,2}, ..., a_{i,j,p}]$, directly from the following closed form equation.

$$a_{i,j} = (Q_{i,j}^T Q_{i,j} + \lambda I)^{-1} \ Q_{i,j}^T \bar{r}(s_i, \bar{l_j}). \tag{3.13}$$

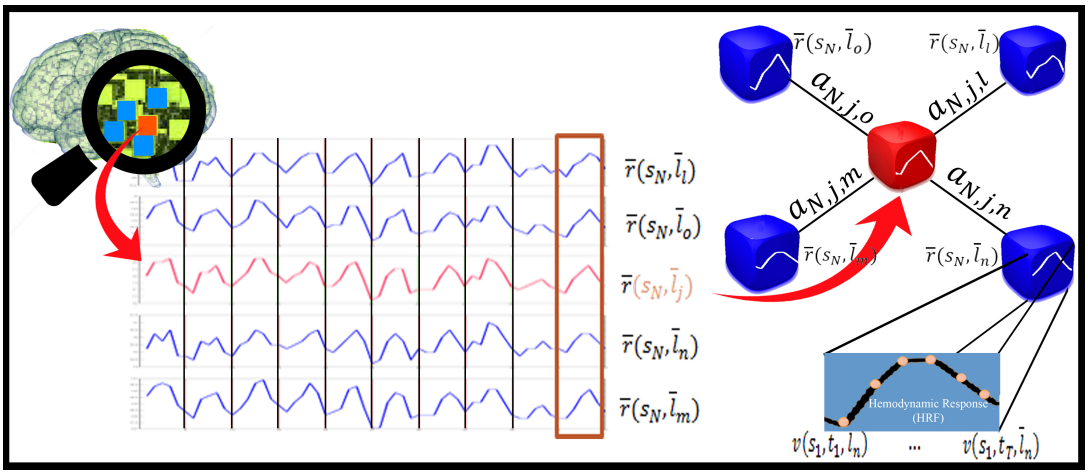

Figure 3.4: The abstract overview of Temporal Mesh Model. The zoomed red voxel is called seed voxel, and it is taken with its 4 neighbors (shown with the blue color). The arc weights between seed voxel and neighbors are estimated using six discretized form of hemodynamic responses

3.3.2 Pruning Edges

"Pruning Edges" is an important part of the proposed architecture (STMM). The goal of this phase is to solve the problem of TMM. Note that, TMM changes the input feature space. The dimension in the new space increases from N (number of voxels) to $p \times N$, where p is the number of considered neighbours around *seed voxel*. For instance, if the similarity metric of mesh model were adjusted to examine four neighbours (p=4, such as the example shown Fig. 3.4, then the dimension of feature space would be increased four times. In the case where we select 500 voxels as the output of feature selection, the new feature space would be 500×4 for the given example. However, the approach has two problems. First, TMM accelerates the "curse of dimensionality" problem. The reason of this is the mapped dimension has higher number of dimensions compared to the input feature space. Furthermore, counting a constant number of neighbours for all of the voxels is somehow *"rough"* assumption, and the degree of relationships for voxels v_i and v_j could be different. While some of the voxels are connected by high degrees to their neighbours, the others may have lower degree of connections. In order to solve this problem, again,

a feature selection method which is employed previously can be used to prune the estimated arc weights.

Therefore, we used ANOVA based feature selection method in order to select the most discriminative features. Contrary to the previous case which features were the voxels, the features are the arc weights in this case. 3 is used to select the most discriminative arc weights and prune the edges in the estimated network by TMM.

3.4 Dimension Reduction using t-Distributed Stochastic Neighbor Embedding (tSNE)

Among the several of visualization methods such as PCA, LLE and Isomap, we choose t-Distributed Stochastic Neighbor Embedding (tSNE) to visualize the feature space. The reason of this choice was the superiorities of tSNE which are discussed in the section 2.6.1. tSNE is the other part of the proposed STMM architecture (see Fig. 3.1). tSNE enables us to map the output of different phases in the architecture into the two dimensional feature space for visualization purpose, which we can feel the data points and feature space.

The idea of tSNE is to calculate the local similarity in the original feature space by the following steps:

1. A Gaussian kernel is fitted at the center of all objects in the original space.
2. The density over all the other points under this Gaussian is estimated.
3. Finally, the similarity is normalized by summation of the similarities of the other points.

Mathematically, the similarity between x_i and x_j is defined by;

$$p_{ij} = \frac{exp(-||x_i - x_j||^2/2\sigma^2)}{\sum_k \sum_{l \neq k} exp(-||x_k - x_l||^2/2\sigma^2)} , \tag{3.14}$$

where $p_{i,j}$ is defined as the new similarity metric between point i and j . Intuitively, if the point x_i and x_j are close together in the original high dimensional space, then the value of $p_{i,j}$ would be large. Conversely, $p_{i,j}$ would be small in the case of large distance between x_i and x_j.

However, there is a problem with this similarity measurement. Imagine the case where all pairwise Euclidean distances $||x_i - x_j||^2$ are large when the Gaussian is centered at x_i. In this situation the values of all joint probabilities p_{ij} would be extremely small. As a result, the cost- function would not have significant impact on the data points of the low-dimensional map.
In order to compensate the effect of this problem, the conditional probability $p_{i|j}$ is computed by using the joint probability p_{ij}. The conditional probability has a difference with the joint one in the normalization part (denominator). It is not taken all of the pair of points, but it only covers the pair points that involve the point x_i. The conditional probability $p_{i|j}$ is calculated as follows:

$$p_{i|j} = \frac{exp(-||x_i - x_j||^2/2\sigma^2)}{\sum_{j' \neq k} exp(-||x_i - x_{j'}||^2/2\sigma_i^2)} \ . \tag{3.15}$$

This approach also provides to set different bandwidth σ_i for each point, and the bandwidth is set in such a way that the conditional probability $p_{i|j}$ has a fixed perplexity. In other words, the variance σ of Gaussian kernel is selected such that a fixed number of points fall under the Gaussian function. This stems from the fact that different parts of space may have different densities. However, the joint probability (the similarity metric) is obtained by symmetrizing the obtained conditional probabilities of $p_{i|j}$ and $p_{j|i}$;

$$p_{i,j} = \frac{p_{i|j} + p_{j|i}}{2N} \ . \tag{3.16}$$

Up to now, the similarity in the original feature space is examined, particularly between each pair, x_i and x_j. Now, we turn our focus on the low dimensional space. Similar to the original space, the problem of lower dimension is to find out the similarity metric. The idea of tSNE is to use the same similarity metric in the output feature space as follow;

$$q_{ij} = \frac{(1 + ||y_i - y_j||)^{-1}}{\sum_k \sum_{l \neq k} 1 + (||y_k - y_l||^2)^{-1}} \ . \tag{3.17}$$

Where, y_i and y_j are the corresponding low dimension data points of high dimensional object x_i and x_j. Therefore, q_{ij} is the similarity metric of the pair data between y_i and y_j in the final low dimensional map. Note that, the Gaussian kernel is replaced with the student-t distribution with one degree of freedom in the computation of q_{ij}. This distribution has longer heavy tails than the Gaussian one, and this gives an important property to the tSNE. In the case where the data is intrinsically high-dimension, the student t-distribution results to well separation of dissimilar points in the final map.

Remember the goal which was to keep the similarity between the data points in the original space and final map. This goal can be achieved by minimizing the difference between q_{ij} and p_{ij}. tSNE measures the difference between q_{ij} and p_{ij} using following Kullback-Leibler Divergence;

$$C(\varepsilon) = KL(P||Q) = \sum_i \sum_{i \neq j} p_{ij} log \frac{p_{ij}}{q_{ij}} \ . \tag{3.18}$$

The above cost function should moves around in the final map such that the $KL(P||Q)$ would be minimized. This objective function is non convex in the embedding ε, and it is minimized by descending along the gradient (3.18):

$$\frac{\partial C}{\partial y_i} = 4 \sum_{i \neq j} (y_i - y_j)(p_{ij} - q_{ij})(1 + ||y_i - y_j||^2)^{-1}. \tag{3.19}$$

This equation enable us to move a single point y_i in the final map to get lower value of KL. Notice that the equation (3.19) is made up two terms. The first term, $(y_i - y_j)$, plays the role of a spring between a pair of points y_i and y_j in the final visualization map. The second term,$(p_{ij} - q_{ij})$ controls the spring (first term). Particularly, if the final map would be perfect, then $(p_{ij} - q_{ij})$ in the second term would be zero. In this case, the effect on the spring or the first term would be zero. The summation term ($\sum_{i \neq j}$) counts all of the effects from the points in the final map. In other words, moving a point in the final map is controlled by all the other point.

In order to initialize the gradient descent, an isotropic Gaussian with small variance is used to randomly sample the map points. Additionally, the gradient is added with a relatively large momentum term $\alpha(t)$ in order to decrease the probability of poor

local minima. Particularly, the sum of previous gradients,which are exponentially decreasing, is added to the current gradient at each iteration in order to specify the variations in the coordinates of $y's$ at the final map. Mathematically, the update of gradient which contains the momentum term is in the following from;

$$\gamma^{(t)} = \gamma^{(t-1)} + \eta\frac{\delta C}{\delta\gamma} + \alpha(t)\Big(\gamma^{(t-1)} - \gamma^{(t-2)}\Big). \tag{3.20}$$

where,

γ^t is the solution at the tth iteration,

η is the learning rate, and

$\alpha(t)$ is the momentum at iteration t.

The pseudo code of discussed dimension reduction method (tSNE) is given in Algorithm 5. Additionally, the two example of tSNE output is shown in Figure 3.5.

Algorithm 5: Pseudo code of t-Distributed Stochastic Neighbour Embedding

Input: *DATA* matrix in the size of $M \times N$ where N is the number of voxels.

for All of the data pints **do**
  Compute the pairwise conditional probabilities $p_{i|j}$ and $p_{j|i}$ using Eq. (3.14) in the high dimension input space
  $p_{ij} \longleftarrow (p_{i|j} + p_{j|i})/2n$
  $\gamma^{(0)} \longleftarrow \{y_1, y_2, ..., y_n\}$ using isotropic Gaussian with small variance
  for $t = 1$ to T **do**
    Compute the pairwise similarities $q_{i|j}$ and $q_{j|i}$ by Eq. (3.17) in the final map.
    Compute gradient $\frac{\delta C}{\delta\gamma}$ using Eq. (3.19)
    $\gamma^{(t)} \longleftarrow \gamma^{(t-1)} + \eta(\frac{\delta C}{\delta\gamma}) + \alpha(t)(\gamma^{(t-1)} - \gamma^{(t-2)})$
  end for
end for
return The low-dimensional map $\gamma^{(T)}$ of DATA.

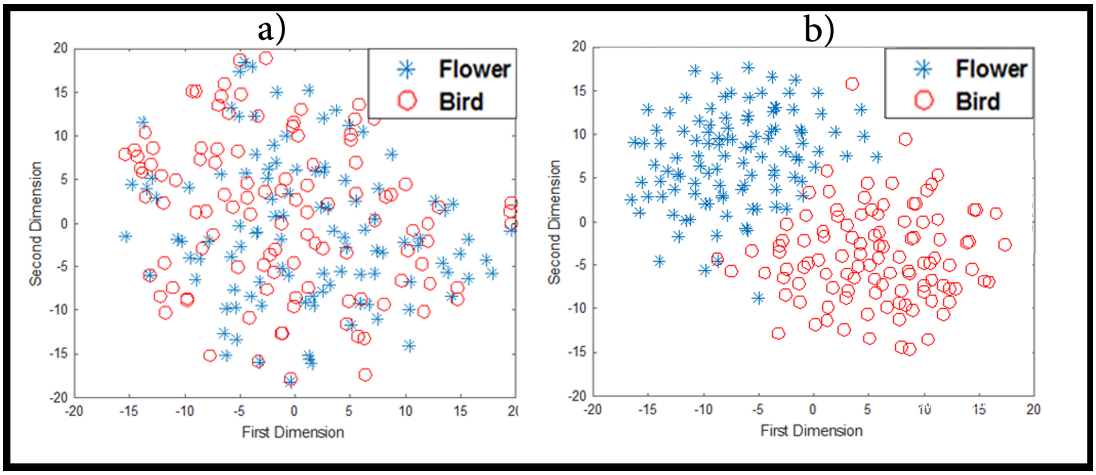

Figure 3.5: Two output example of 2D visualization using tSNE. a) an example of MVPA visualization of fMRI data with binary class conditions , b) an example of visualizing the output of STMM.

3.5 Chapter Summary

In this chapter, the architecture called STMM which aim to decode the brain states have been proposed. In the first phase of the proposed architecture, the fMRI data which contain tens of thousand voxels, are simplified by selecting only a proportion of them. The main goal of this step is to discard the noisy voxels and reduce the "curse of dimensionality" problem. This approach helps us to develop a powerful brain decoding model with high accuracy and speed. As shown in Fig. 3.1, the model uses both univariate and multivariate feature selection methods to select the informative voxels. After this step, the newly developed brain decoding method which is called temporal mesh model (TMM) is applied on the selected voxels to estimate the arc weights between the seed voxel and its neighbours. However, contrary to the dimension reduction techniques, TMM increases the feature space, and again a feature selection method is used to solve this problem. Finally, the dimension reduction technique called tSNE is applied to visualize the data points in the feature space of the different phases in STMM.

CHAPTER 4

EXPERIMENTS AND RESULTS

In this chapter, first, the details of data acquisition and fMRI recordings are explained. Then, two analyses known as "Intersection" and "Anatomical" analyses are accomplished for evaluating the voxel selection methods. The goal of these analyses is to show the consensus of the univariate and multivariate voxel selection methods. Next, the classifier performance of employed voxel selection methods in the proposed architecture are illustrated and discussed. After discussing the neighbourhood analysis for Temporal Mesh Model (TMM), the results of proposed architecture are shown and discussed in the following sections. Finally, 2D visualization results of tSNE are presented in the last section.

4.1 Data Acquisition and fMRI Recordings

The data set which is used in this study is acquired by "Pattern Analysis of functional Magnetic Resonance Imaging" group of Computer Engineering Department at Middle East Technical University. This data set is task related, and the subjects are stimulated during the fMRI recordings. The visually presented stimuli are images which fall into two categories of birds and flowers. These images contain the pictures of flowers or birds on the gray background. Fig. 4.1 shows the schematic setup of the experiment performed during data recordings. First, the participants are stimulated visually with an image up to four seconds. Then, a resting state is followed after stimulation phase. The time of resting state varies randomly among 8, 10 or 12 seconds. This resting state is designed in order to get back the stimulated brain to its baseline.

During the experiment and after the stimuli, the subject is asked to detect the category of the object. For example, if a flower image is followed by another flower image, the participant is expected to say that these objects are from the same category. The question is answered by pushing a bottom. Whole fMRI recordings is divided into six parts that are called runs. In each run the participant is stimulated 36 times. This means that we have 36 samples for each run and 216 samples for all runs combined. The dataset is binary class type with 108 samples for each class.

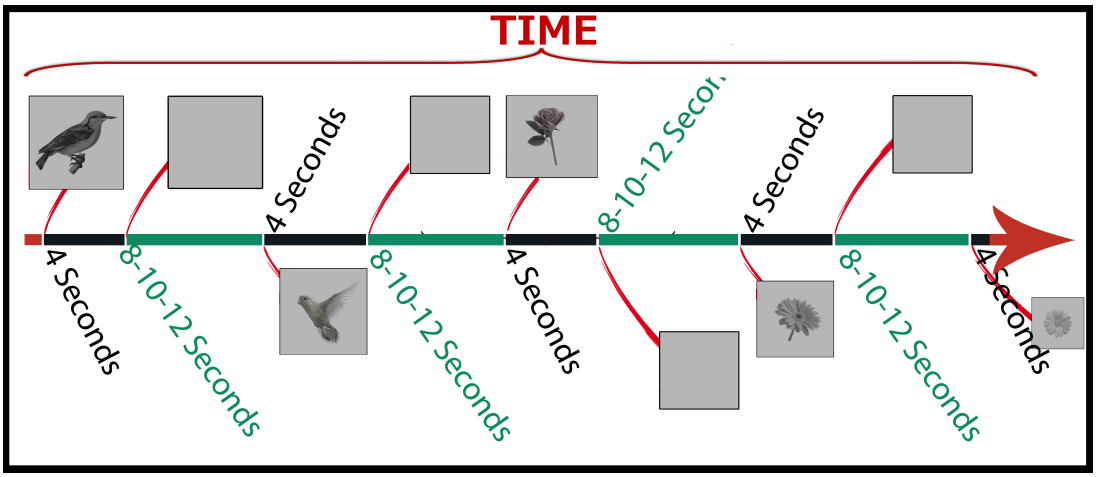

Figure 4.1: The abstract overview of the experiment which data is gathered. 8, 10 or 12 seconds of resting without any stimulus follows the four seconds of visually stimulation of brain.

Six subjects have been participated in this fMRI recordings. The first subject is used to set the fMRI machine. Additionally, the fMRI data of fourth subject was ruined during the recording. Therefore, the data of subjects 002, 003, 005 and 006 are used in this study. The number of voxels is different among the subjects. Specifically, the number of voxels for subjects 002, 003, 005 and 006 are 20177, 19962, 19311 and 19757 receptively.

4.2 Analysing Feature Selection Methods

As discussed in the previous chapter, the first and second phases of the proposed architecture STMM are voxel selection methods. Before the use of them in STMM, the consensus of them should be analysed. In order to measure the validity and rationality of the methods, two analyses are designed. In the first analysis, we examine

whether the voxel selection methods select the same set of voxels or not. In fact, if the employed methods are consensus, then we expect them to select the same voxels. Therefore, the analysis called "Intersection", which is explained in the next subsection, is designed. In the second analysis, the anatomical regions of selected voxels are examined. The purpose of this analysis is to show the existence of the relationships between the anatomical locations of the selected voxels and neuroscientific information. This analysis which we call it "Anatomical" analysis is done by examining the experiment which fMRI data is recorded and the anatomical locations of selected voxels which are represented by using CEREBRA toolbox [68].

4.2.1 Intersection Analysis

As we mentioned previously, the goal of voxel selection methods was to rank and select the informative voxels. At the output of these methods, voxels are ranked from high to low according to their discriminative power, then the low ranked voxels are eliminated using a threshold. In order to find out the validity of the feature selection methods, the analysis known as "Intersection" is designed. The purpose of this analysis is to find out whether the selected features in different methods are common or not. The 3D coordinate of the voxels in the brain are used to specify the intersections of the methods. The coordinates of the voxels consists of three elements which specify x,y and z axis in the brain. For example, if a voxel with $[4, 70, 36]$ coordinate is selected by the Mutual Information (MI) based feature selection, then another method (such as ANOVA or RFE) is checked to see whether the same voxel is selected or not. After checking all of the voxels the percentage of intersection is simply calculated by dividing the number of commonly selected features to the total number of voxels.

The results of "Intersection" analysis are shown in Fig. 4.2. In this figure the x axis shows the number of selected voxels, and y axis corresponds to the intersection percentage between the methods. The figures show that there is an increasing trend in the intersection percentage as the number of voxels decreases up to a point. Additionally, note that both of the univariate feature selection methods (MI and ANOVA) have higher intersection percentage compared to the intersections between univariate with multivariate (RFE). Therefore, the results of this analysis show that some features are

detectable with more than one voxel selection methods, and this can be interpreted as the validity of the methods.

4.2.2 Anatomical Analysis

In this analysis, the anatomical regions are examined to inspect the location of selected voxels . The goal of this analyse is to track the anatomical location of eliminated and selected voxels. Particularly, the analysis is done to check the relationships between the anatomic regions and the location of selected voxels depending on the nature of the experiment performed during the fMRI recording.

As we discussed in section 4.1, the subjects are stimulated visually by the images and asked to remember the category of the images during the fMRI recording. From the anatomical and functional neuroscientific information about the brain, we expect that the active voxels to be in the anatomical parts of brain which are related to vision and visual memory. In order to implement this analysis, the different number of voxels were selected with different methods. Then, the anatomical regions of selected voxels are visualized by using CEREBRA toolbox [68].The results of ANOVA and RFE voxel selection methods for Subject 2 is shown in Fig. 4.4 and 4.5, and the results of RFE and ANOVA for Subject 6 is shown in Fig. 4.10 and 4.11. However, an abstract knowledge about the brain anatomy and function would be beneficial in order to interpret the results.

Generally, brain is made of three parts: brain stem, cerebellum and cerebrum. Brain stem is a bundle of nerve tissue at the base of the brain, and it is responsible for breathing, body temperature, blood pressure, heart rate and hunger and thirst. Cerebellum, on the other hand, is located in the back of the brain under the cerebrum. The functions of cerebellum includes movement, posture, balance, reflexes, complex actions (such as walking, talking), collecting sensory information from the body. However, both of brain stem and cerebellum are not examined for brain decoding tasks. Cerebrum is the only part which is examined for brain decoding purpose in this study. It is the largest part of the brain which is divided in to two parts left and right cerebral hemispheres. A bridge of nerve fibers called corpus callosum connects these two

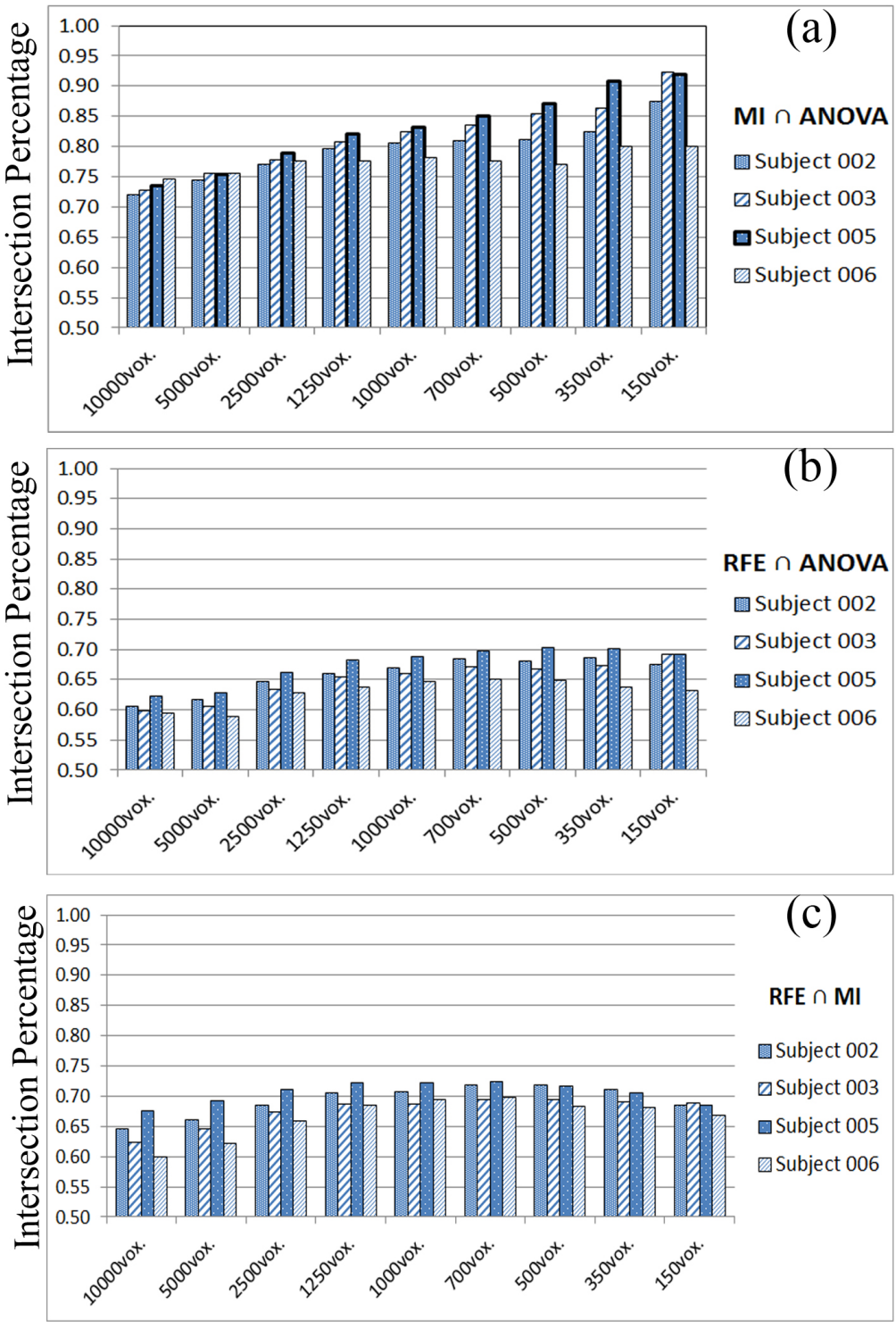

Figure 4.2: The results of *Intersection* analysis between ANOVA, MI and RFE feature selection methods in different number of selected voxels. The x axis represents the number of selected voxels, and the y axis shows the intersection percentage between methods. a) the intersection between univariate methods (MI and ANOVA), b) the intersection between RFE and ANOVA and c) the intersection between RFE and MI.

parts. Functionally, the cerebrum is divided into four regions: frontal (front), parietal (top), temporal (side), and occipital (back) lobes (see Fig.4.3).

- Frontal lobe controls the body movement, speech, behaviour, memory, emotions and intellectual functioning such as decision making.
- Parietal lobe is responsible for sensation such as touch, pain and etc.
- Temporal lobe controls hearing, memory, visual memory and emotion.
- Occipital lobe controls vision via functional visual areas.

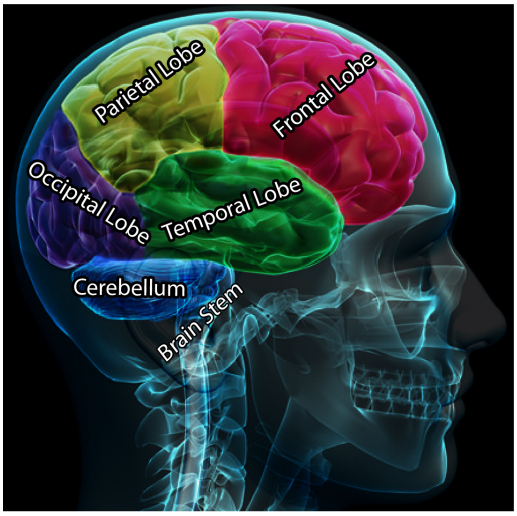

Figure 4.3: The functional parts of brain. Four main parts of cerebrum which are labelled by frontal, parietal, temporal, and occipital lobes and the other two brain parts which are cerebellum and brain stem.

The result of the "Anatomical" analysis for Subjects 002, 003, 005 and 006 are shown in Figures 4.4, 4.5,4.6, 4.7,4.8, 4.9 4.10 and 4.11. The rows of the figures show different number of selected voxels, and columns illustrate the results of ANOVA and RFE voxel selection methods. It is obvious that the tendency of both ANOVA and RFE voxel selection methods is to select the voxels from occipital and temporal lobs as the number of selected voxel decreases to below 1000 voxels.

As mentioned above, one of the the major functions of occipital lobe is vision control. This matches with the results of this analysis shown in Figures 4.4, 4.5,4.6, 4.7,4.8,

4.9 4.10 and 4.11. It is obvious that one of the brain regions which contains the selected voxels (shown in red color) is occipital lobe. Additionally, the other brain region which also contains the major number of selected voxels is temporal lobe. This also complement with the function of temporal lobe which contains memory control. Remember, that the subjects are asked to memorize the category of objects in the fMRI recording experiment. The results of mutual information (MI) based feature selection method is not shown in the figures. In fact, most of the selected voxels with MI are also selected by ANOVA. This can be inferred from the "Intersection" analysis (see Fig. 4.2). Therefore, the anatomical visualization of the selected voxels with MI would be very similar to ANOVA, and the results of MI does not shown due to similarity in the results of ANOVA and MI.

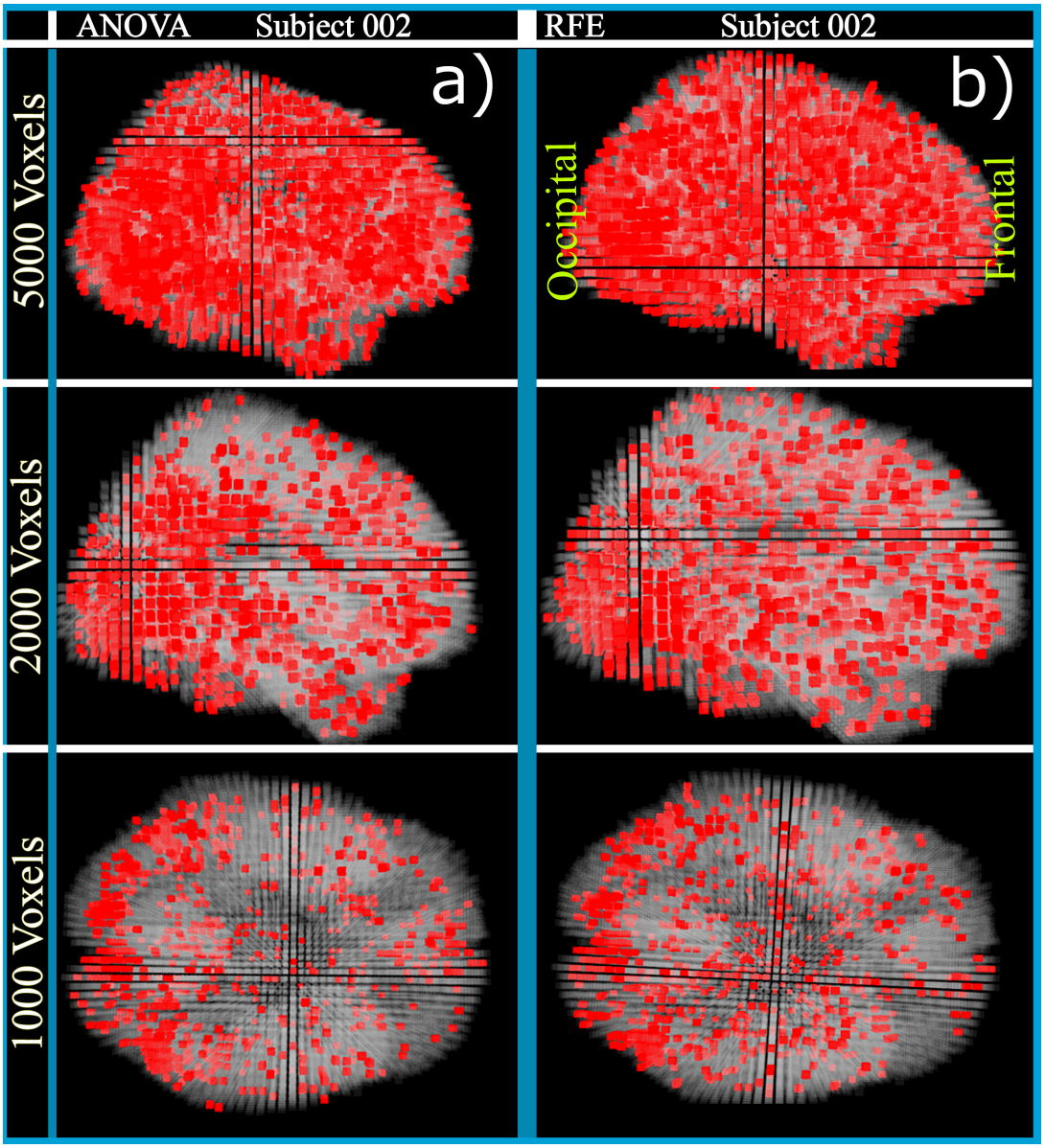

Figure 4.4: The anatomical regions of the different number of selected voxels (5000, 2000, 1000) by ANOVA and RFE in Subject002. Column (a) shows the results of ANOVA, and Column (b) shows the results of RFE voxel selection methods. The rows illustrate the different number of voxels which are selected by ANOVA and RFE feature selection methods.

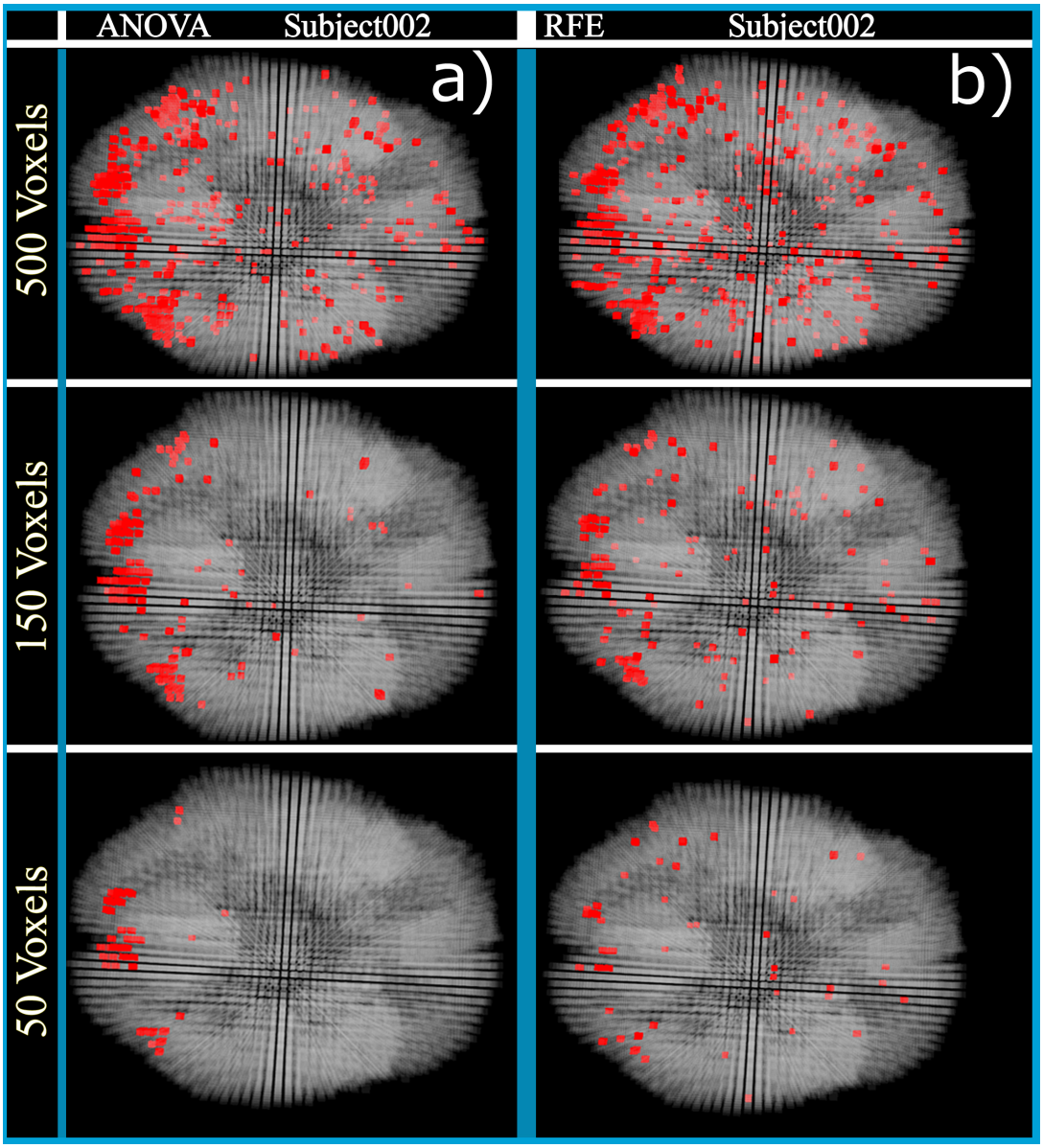

Figure 4.5: The anatomical regions of the different number of selected voxels (500, 150, 50) by ANOVA and RFE in Subject002. Column (a) shows the results of ANOVA, and Column (b) shows the results of RFE voxel selection methods. The rows illustrate the different number of voxels which are selected by ANOVA and RFE feature selection methods.

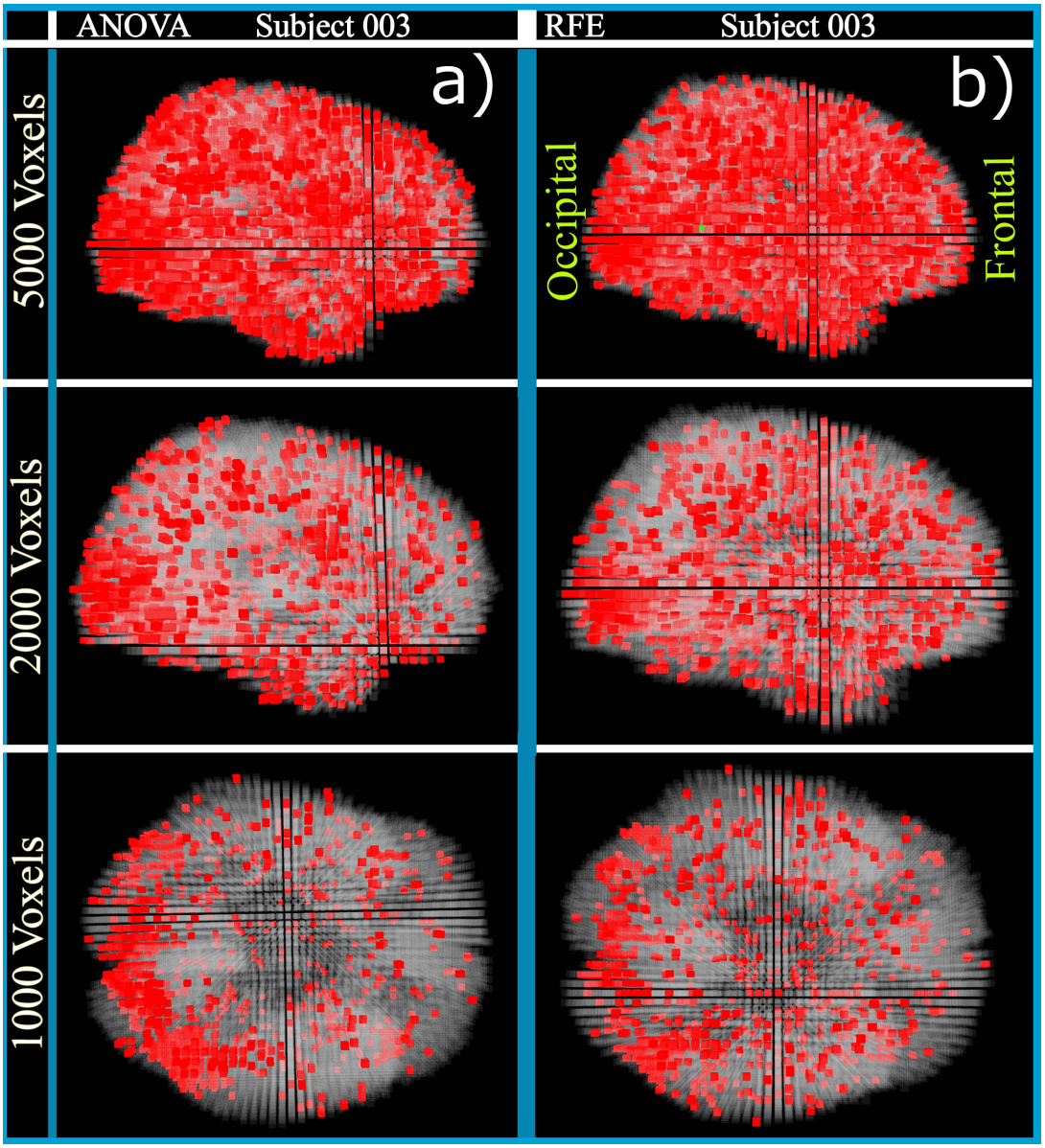

Figure 4.6: The anatomical regions of the different number of selected voxels (5000, 2000, 1000) by ANOVA and RFE in Subject003. Column (a) shows the results of ANOVA, and Column (b) shows the results of RFE voxel selection methods. The rows illustrate the different number of voxels which are selected by ANOVA and RFE feature selection methods.

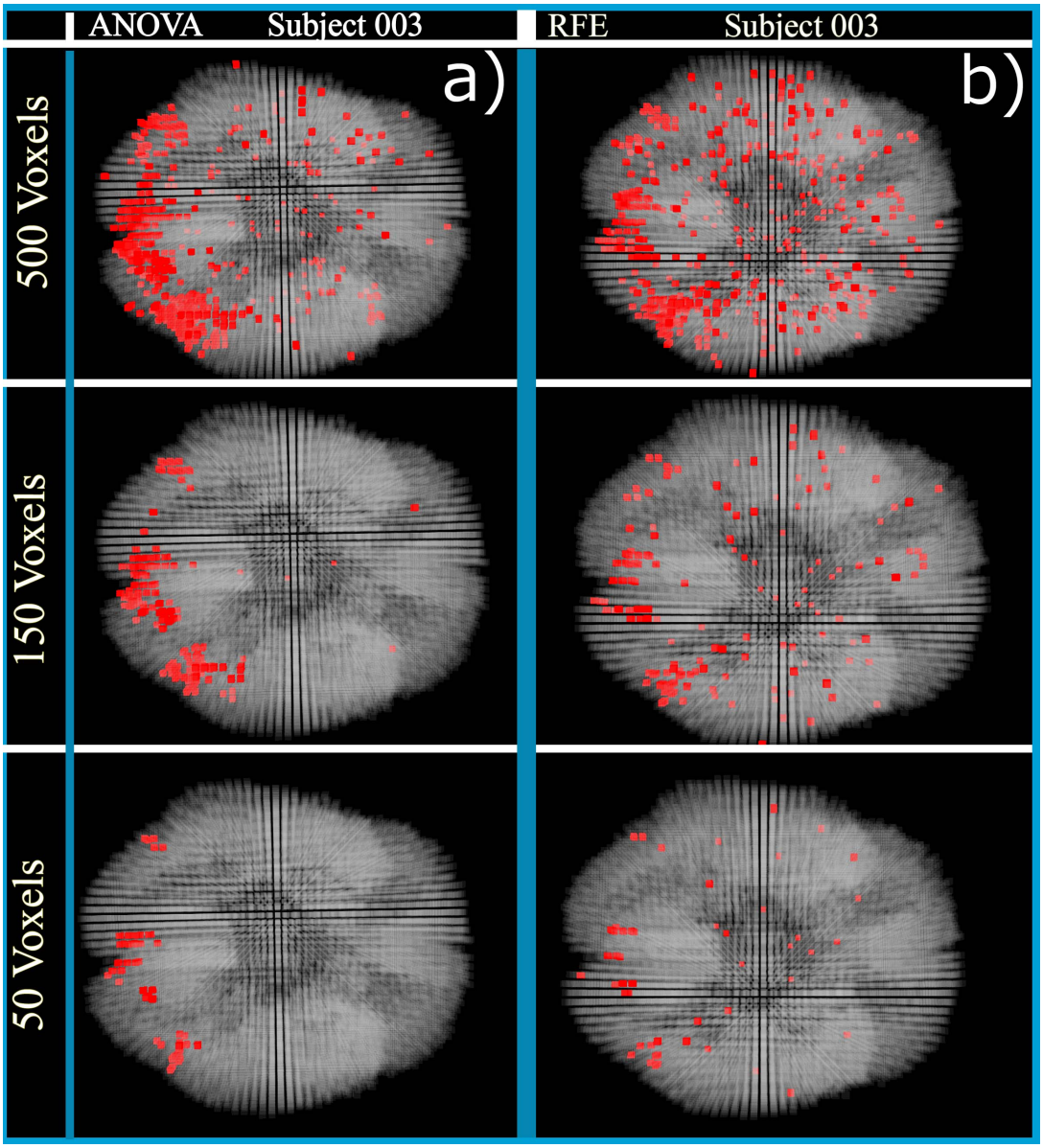

Figure 4.7: The anatomical regions of the different number of selected voxels (500, 150, 50) by ANOVA and RFE in Subject003. Column (a) shows the results of ANOVA, and Column (b) shows the results of RFE voxel selection methods. The rows illustrate the different number of voxels which are selected by ANOVA and RFE feature selection methods.

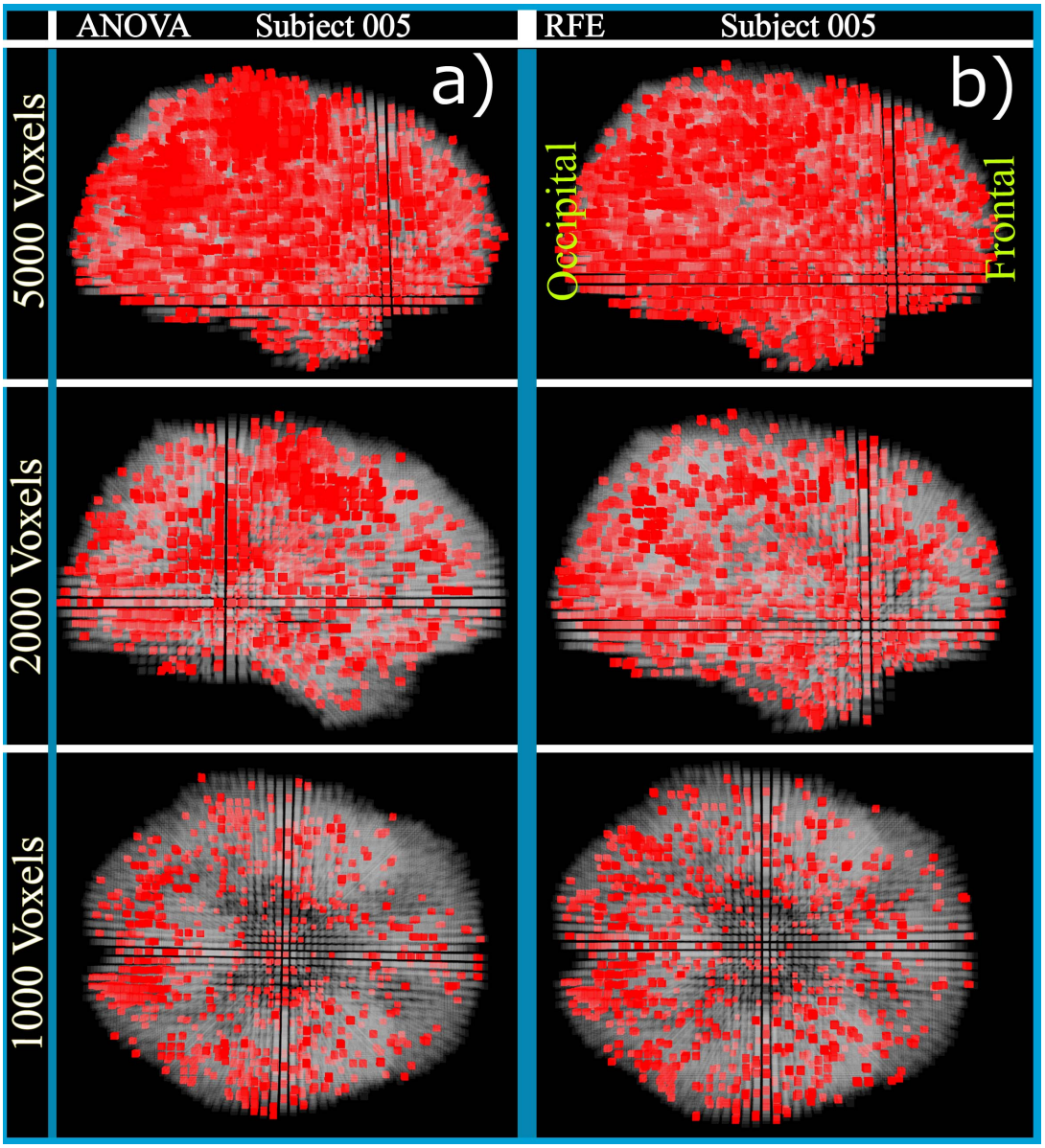

Figure 4.8: The anatomical regions of the different number of selected voxels (5000, 2000, 1000) by ANOVA and RFE in Subject005. Column (a) shows the results of ANOVA, and Column (b) shows the results of RFE voxel selection methods. The rows illustrate the different number of voxels which are selected by ANOVA and RFE feature selection methods.

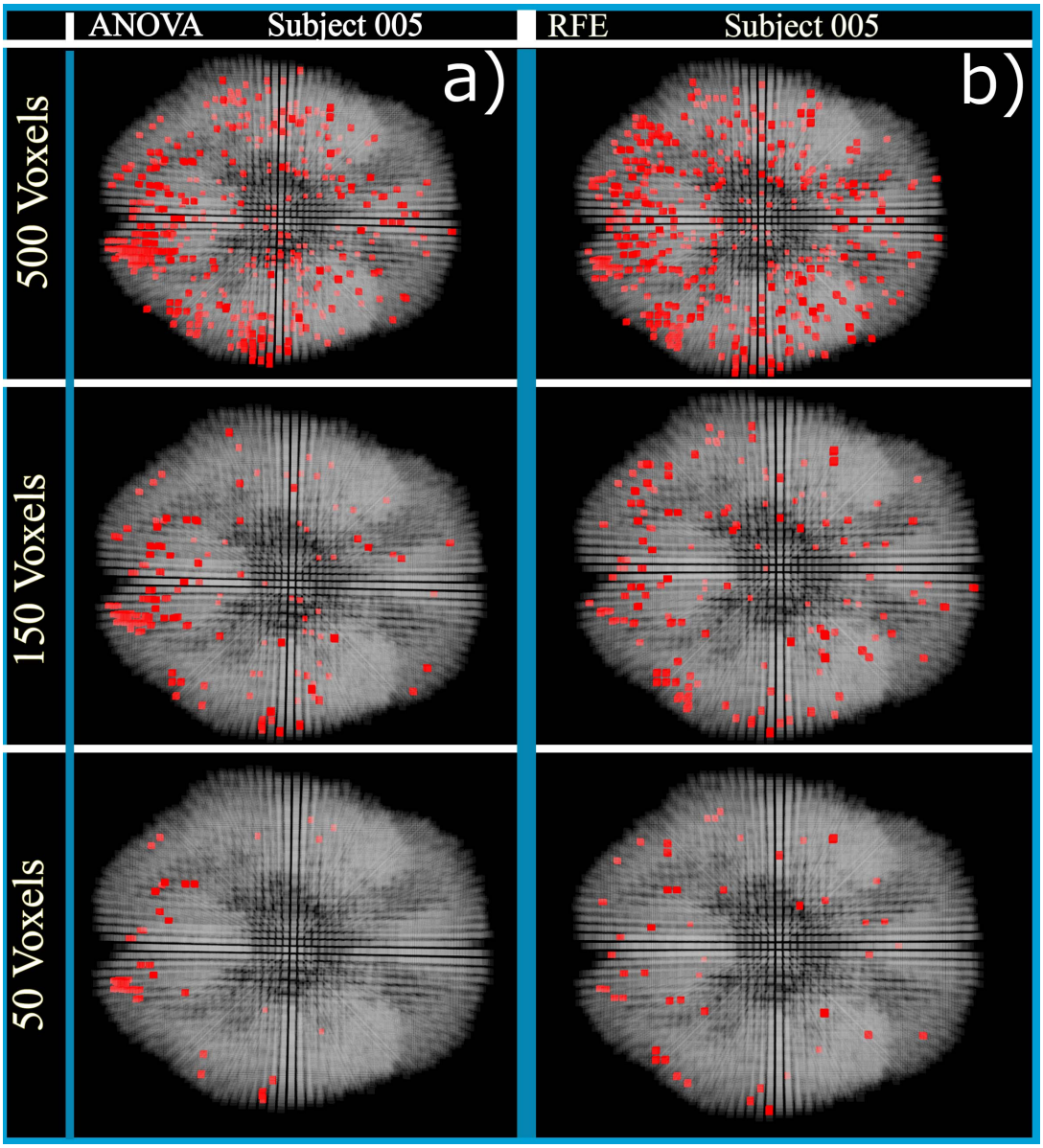

Figure 4.9: The anatomical regions of the different number of selected voxels (500, 150, 50) by ANOVA and RFE in Subject005. Column (a) shows the results of ANOVA, and Column (b) shows the results of RFE voxel selection methods. The rows illustrate the different number of voxels which are selected by ANOVA and RFE feature selection methods.

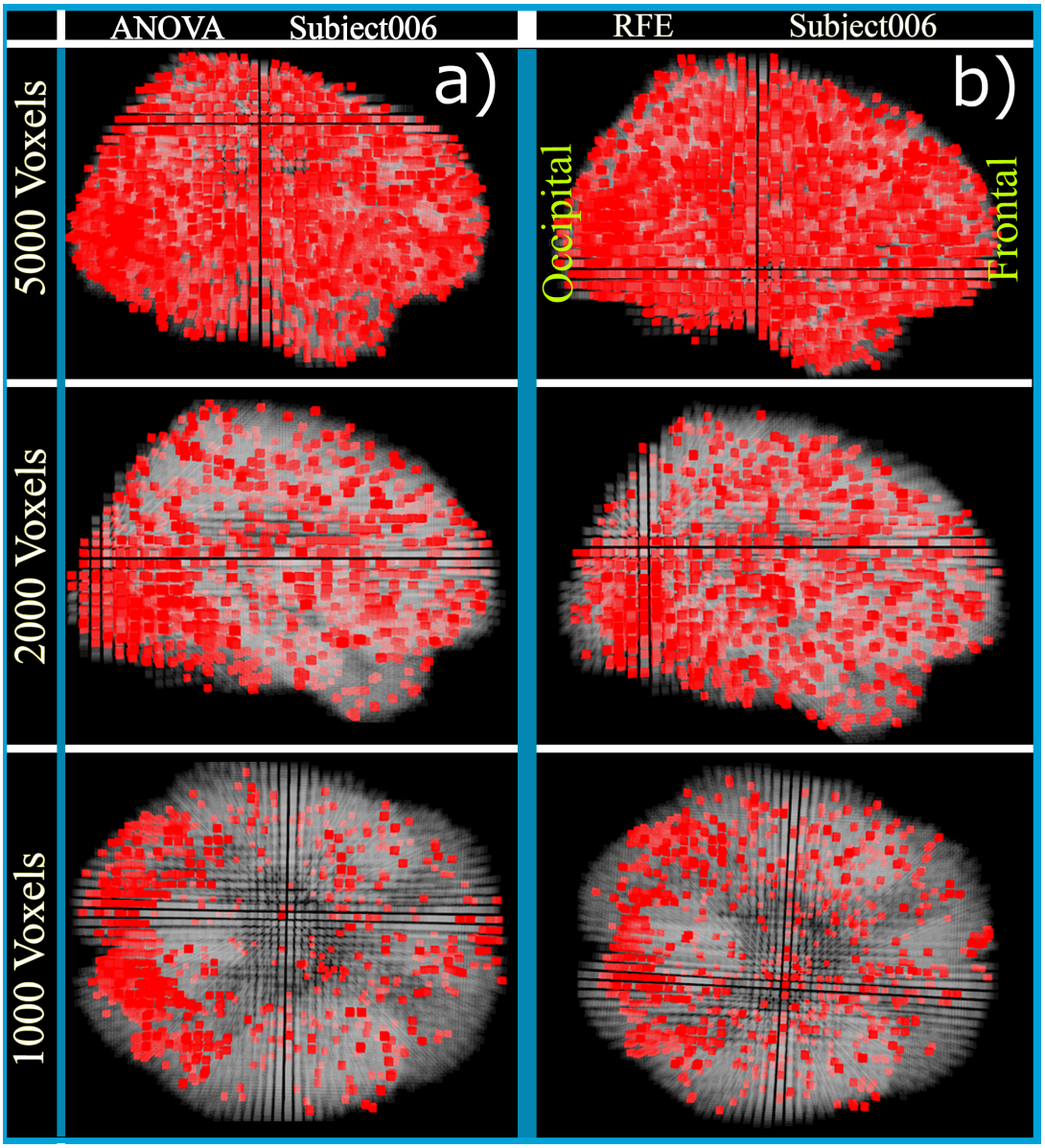

Figure 4.10: The anatomical regions of the different number of selected voxels (5000, 2000, 1000) by ANOVA and RFE in Subject006. Column (a) shows the results of ANOVA, and Column (b) shows the results of RFE voxel selection methods. The rows illustrate the different number of voxels which are selected by ANOVA and RFE feature selection methods.

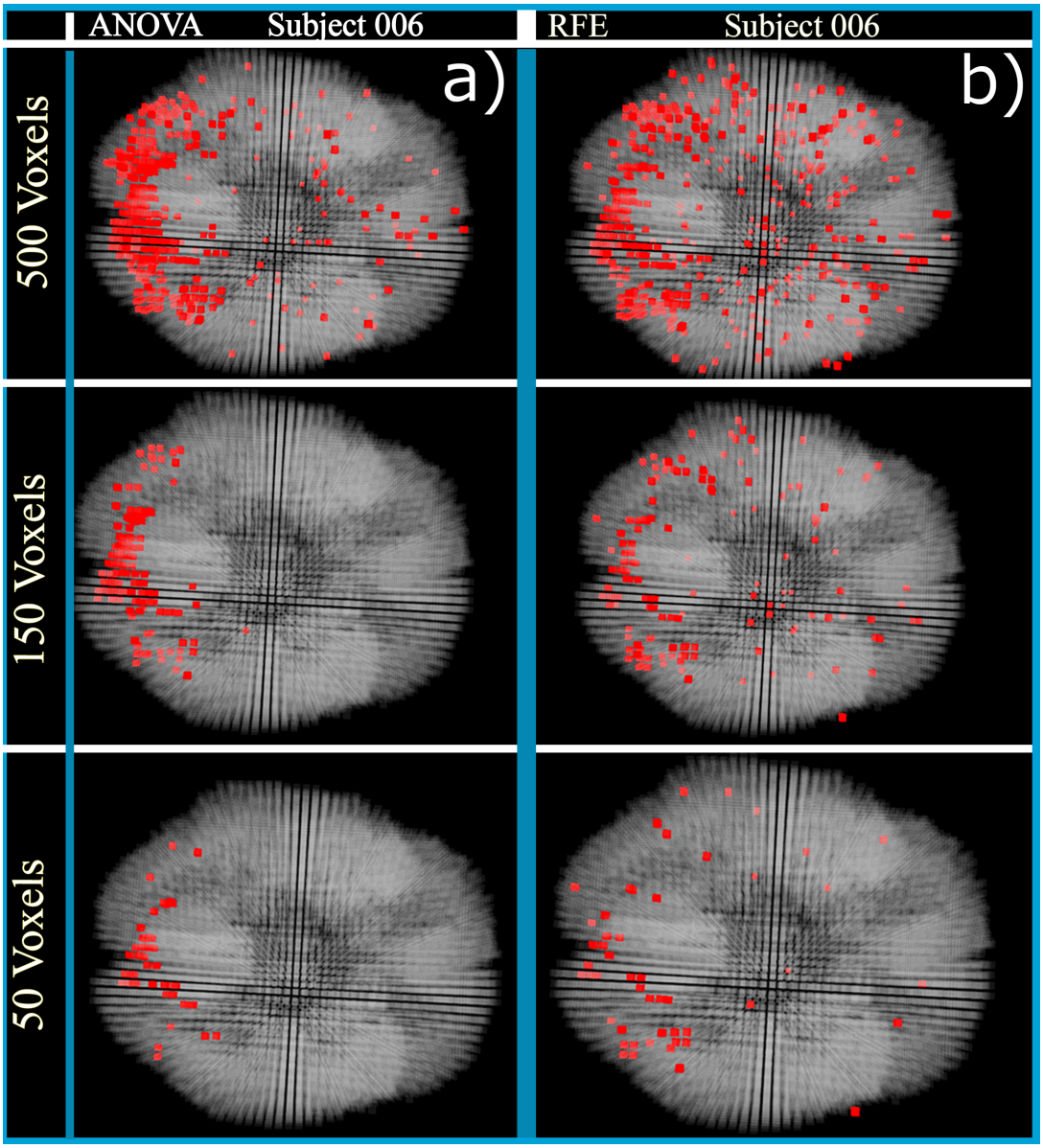

Figure 4.11: The anatomical regions of the different number of selected voxels (500, 150, 50) by ANOVA and RFE in Subject006. Column (a) shows the results of ANOVA, and Column (b) shows the results of RFE voxel selection methods. The rows illustrate the different number of voxels which are selected by ANOVA and RFE feature selection methods.

4.3 Measuring the Effects of Feature Selection Methods on Brain Decoding

In this section, the impacts of discussed voxel selection methods (ANOVA, MI and RFE) on the brain decoding are measured using classification performance of KNN and SVM algorithms. The following steps are followed to measure the classification performance for different selected voxel numbers in different subjects. The first step is to divide the data into two parts namely training set and test set. As mentioned in section 4.1, the fMRI data which is used in this research, contain six runs. We divided this fMRI data set according to the runs. Three runs (Run1, Run3, Run5) are used as the training sets, and the others (Run2, Run4, Run6) as the test sets. In the second step, the voxel selection methods are implemented on the training sets to rank and select the most informative voxels. Next, the 3D coordinates of the selected voxels from the training set are used to select the same voxels with the same 3D coordinates from the test set. Finally, KNN and SVM classifiers are trained using training sets, and their classification performance is measured using test sets. The classification performance is equal to the number of truly predicted labels divided by the whole number of labels or class conditions. For instance, if the SVM classifier truly predict 70 class conditions out of 108 ones during the testing of classifier, then the classification performance or accuracy would be 0.65 in this example.

Tables 4.1 and 4.2 show the classification performances of KNN and SVM classifiers in different number of selected voxels. The results of Multivariate Pattern Analysis (MVPA) on whole brain are obtained without any voxel selections and shown in the third row of the tables. The number of selected voxels are shown in the first column of tables (from 15000 to 500). These number of voxels are selected so that the corresponding classification performance changes as much as possible. In other words, in order to avoid repetition and similarity of performances, only the specific range of selected voxels (from 15000 to 500) are shown in the tables. Additionally, the performances of low voxel numbers (bellow the 500) are not shown due to the same reason. In this study, the kernel of SVM classifier is linear, and the C parameter of the classifier is optimized by grid search. In KNN classifier, the euclidean distance is used, and the K parameter is optimized by k fold cross validation in the training set.

Notice, an increasing classification performance trend exists up to a specific number of voxels for each method in each subject. For example, the KNN performance of ANOVA (ANO.) at subject 006 increases from 15000 to 1000 number of voxels, and it reaches to its optimal point 0.87 when 1000 voxels are selected using ANOVA. Then, it starts to decrease as the number of voxels decreases. However, the results show the contribution of the voxel selection methods in the performances of classifiers. Additionally, it is notable that the SVM performances surpass the KNN ones in subjects 003, 005 and 006.

4.4 "p" value analysis for TMM

After the first and second phases which select the voxels, the third phase of sparse temporal mesh model (STMM) architecture (see Fig. 3.1), is the implementation of TMM. As Onal et al. [67], and [44] showed finding the optimum number of local neighbours (p) for each subject is critical in order to reach the optimum classification performance. As mentioned in Chapter 3, TMM estimates the arc weights between the *seed voxel* and its locally neighbouring voxels. However, the important question in TMM is: *"How many neighbouring voxels must be chosen?"*. One possible solution is to examine a range of the number of neighbours or p's and then estimate the optimum number of neighbours $\hat{p}$ based on the classification performance.

In this section, the effect of different number of neighbouring voxels p on the classification performance is examined. In this analysis, the selected voxels at the output of second phase in the proposed architecture are given to the TMM. In other words, after RFE (in the second phase of STMM) selected the most discriminative voxels among the previous selected voxels by the first phase (MI or ANOVA) in the STMM, they used as the input of Temporal Mesh Model (TMM) in the third phase.

Of course, the other important issue which must be consider here is how many voxels should be selected in the first and second phases of STMM. In other words, the critial question is how many voxels should be selected in the univariate voxel selection phase (phase A), and how many of them should be selected in the multivariate voxel selection phase (phase B) In order to answer this question, different combina-

Table 4.1: KNN classifier performances of MVPA, Mutual Information(MI), ANOVA (ANO.) and Recursive Feature Elimination (RFE) methods on Subject002, Subject003, Subject005 and Subject006.

KNN	**Subject002**			**Subject003**			**Subject005**			**Subject006**		
Vox. No.	MI	ANO.	RFE	MI	ANO.	RFE	MI	ANO.	RFE	MI	ANO.	RFE
MVPA	0.58	0.58	0.58	0.56	0.56	0.56	0.54	0.54	0.54	0.59	0.59	0.59
15000	0.62	0.47	0.47	0.55	0.54	0.55	0.53	0.47	0.51	0.62	0.64	0.60
12500	0.63	0.60	0.49	0.53	0.55	0.52	0.53	0.50	0.52	0.64	0.62	0.68
10000	0.63	0.64	0.67	0.54	0.56	0.55	0.51	0.50	0.48	0.59	0.62	0.67
5000	0.66	0.69	0.65	0.56	0.55	0.61	0.57	0.62	0.59	0.67	0.72	0.76
3500	0.67	0.69	0.71	0.62	0.63	0.63	0.63	0.67	0.67	0.70	0.76	0.74
2500	0.68	0.76	0.77	0.60	0.63	0.72	0.68	0.63	0.71	0.72	0.83	0.78
2000	0.74	0.78	0.73	0.61	0.66	0.77	0.67	0.69	0.73	0.76	0.81	0.77
1500	0.71	0.80	0.73	0.67	0.67	0.76	0.72	0.69	0.75	0.81	0.82	0.82
1000	0.81	0.76	0.83	0.70	0.73	0.81	0.74	0.76	0.78	0.81	0.87	0.81
700	0.84	0.78	0.81	0.72	0.78	0.83	0.78	0.79	0.75	0.82	0.80	0.86
500	0.83	0.83	0.81	0.77	0.83	0.84	0.77	0.75	0.78	0.87	0.81	0.85

Table 4.2: SVM classifier performances of MVPA, Mutual Information(MI), ANOVA (ANO.) and Recursive Feature Elimination (RFE) methods on Subject002, Subject003, Subject005 and Subject006.

SVM	**Subject002**			**Subject003**			**Subject005**			**Subject006**		
Vox. No.	MI	ANO.	RFE	MI	ANO.	RFE	MI	ANO.	RFE	MI	ANO.	RFE
MVPA	0.72	0.72	0.72	0.77	0.77	0.77	0.81	0.81	0.81	0.83	0.83	0.83
15000	0.76	0.76	0.72	0.78	0.80	0.77	0.81	0.79	0.81	0.81	0.85	0.83
12500	0.78	0.76	0.74	0.81	0.81	0.78	0.81	0.79	0.81	0.84	0.85	0.82
10000	0.78	0.77	0.73	0.83	0.80	0.78	0.80	0.79	0.81	0.84	0.85	0.82
5000	0.79	0.79	0.76	0.86	0.84	0.82	0.80	0.80	0.81	0.89	0.86	0.83
3500	0.81	0.80	0.77	0.91	0.88	0.82	0.78	0.81	0.80	0.89	0.87	0.86
2500	0.81	0.80	0.78	0.89	0.86	0.83	0.79	0.81	0.81	0.87	0.88	0.87
2000	0.80	0.81	0.80	0.90	0.88	0.84	0.79	0.82	0.81	0.87	0.89	0.88
1500	0.81	0.80	0.80	0.88	0.84	0.87	0.79	0.82	0.81	0.89	0.89	0.89
1000	0.81	0.81	0.81	0.88	0.91	0.88	0.79	0.80	0.82	0.91	0.89	0.90
700	0.82	0.84	0.81	0.91	0.92	0.90	0.78	0.81	0.81	0.89	0.90	0.88
500	0.81	0.85	0.81	0.91	0.90	0.90	0.80	0.81	0.81	0.90	0.91	0.89

tion of voxels for the first and second phase of the architecture are examined. These combinations were selected to be near 1000 voxels which the feature selection methods have shown the maximum performances (see Tables 4.1 and 4.2). Fortunately, the examined combinations had similar classification performances, so only one combination of results are used to show in this section. Particularly, first 1000 voxels are selected using ANOVA, then 500 voxels are chosen among them using RFE in the second phase. Plots of next page shows KNN and SVM classification performances of subjects as the number of neighbouring voxels changes.

The classification performances of KNN and SVM classifiers for the range of $p \in \{2, 4, 6, ..., 110\}$ are plotted in Figures 4.12, 4.13, 4.14,4.15. As the figures show, generally, the accuracy increases as the number of neighbours (p) increase up to a specific point for each subject.

As figures show, the optimal number of neighbouring voxels p's varies among the subjects. For example, while the optimal classification performance for Subject 002 is near the 80's, the optimal classification performance is near 90's for Subject 003. Due to the fact that all of the subjects do not show meaning full changes in the classification performances near 100 neighbours, we determined to set the number of neighbours to be fix in this study. Therefore, we choose to work in the fix p, where $p = 100$. It is important to note that the number of neighbouring voxels (p) is relatively high with respect to previous studies such as [67], and [44]. The cause of this phenomena may have one or both of the following reasons. Firstly, voxel selection losses somehow the locality information of voxels, and this may cause to high connection of the *seed voxel* its neighbourhood voxels. The other reason may come from the nature of discriminative voxels. None of previous studies were applied TMM on the selected voxels. Therefore, one possible reason could be that the discriminative selected voxels have high degree of relations with each others.

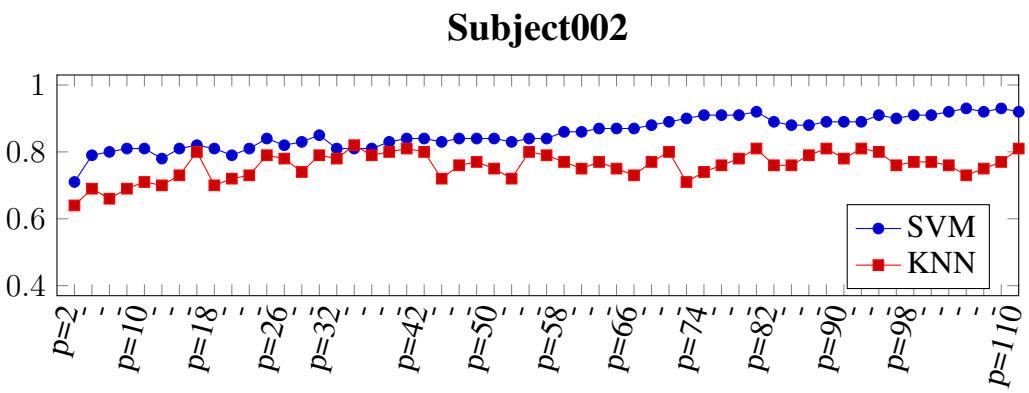

Figure 4.12: SVM and KNN classification performances of (TMM) at the phase C.1 of STMM for different number of neighbours p in subject 002.

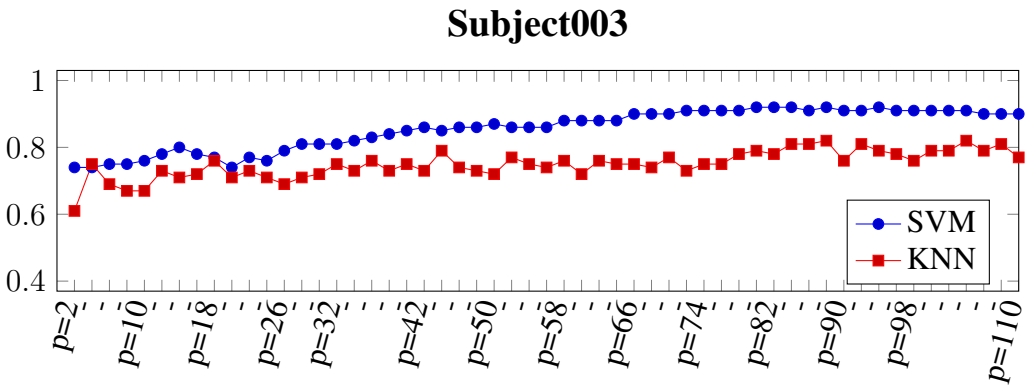

Figure 4.13: SVM and KNN classification performances of (TMM) at the phase C.1 of STMM for different number of neighbours p in subject 003.

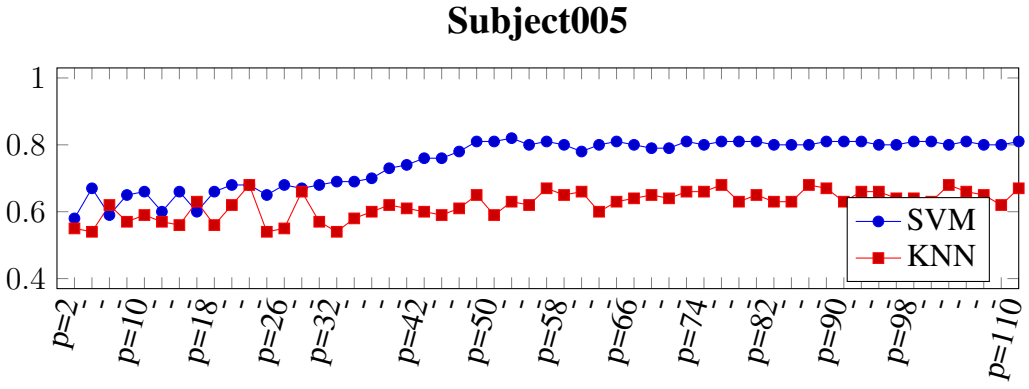

Figure 4.14: SVM and KNN classification performances of (TMM) at the phase C.1 of STMM for different number of neighbours p in subject 005.

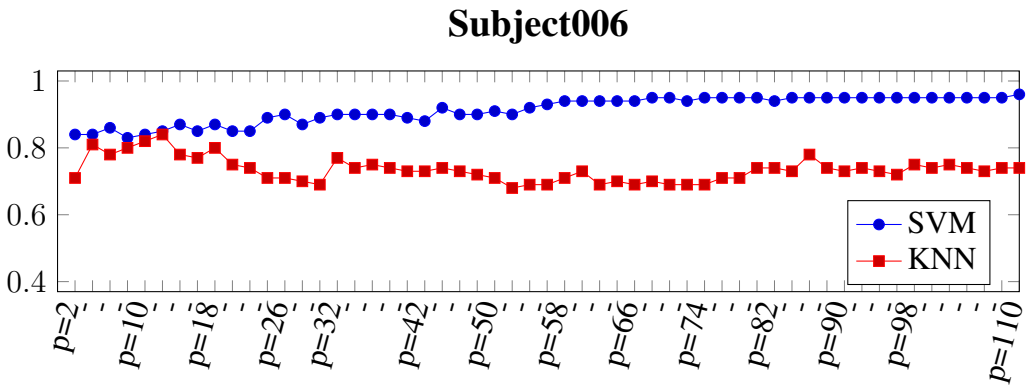

Figure 4.15: SVM and KNN classification performances of (TMM) at the phase C.1 of STMM for different number of neighbours p in subject 006.

4.5 Sparse Temporal Mesh Model (STMM) Results

As mentioned in the Chapter 3, Sparse Temporal Mesh Model (STMM) is made up of three phases. The first phase of STMM is a univariate voxel selection method (MI or ANOVA). This phase is followed by a multivariate voxel selection method (RFE) in the second phase. The selected voxel as the output of second phase is given to the third phase which consists of two steps, formation of the Temporal Mesh Model (TMM) and pruning the edgesof the TMM (see Fig. 3.1).

This section covers the classification performances of KNN and SVM for all the overall STMM architecture. One important fact is that the first phase has direct impact on the second and third phases. In other words, the features/voxels of first phase chosen by ANOVA or MI are used in the following phases. Therefore, we tested both ANOVA and MI to select the most discriminative voxels to see the effects of them on the classification performance.

Let's start with discussing the first univariate method where ANOVA is used, then RFE is implemented on the selected voxels by ANOVA. Finally, TMM and pruning edges have been implemented on the selected voxels at the output of RFE. Fig 4.16 shows the results of KNN classifier for the subjects, and Fig. 4.17 illustrates the results of SVM algorithm. As the results of the tables 4.1 and 4.2 show, the optimum performances are obtained when 2000 or less voxels are selected for all subjects. Therefore, the first phase of architecture (in this path ANOVA) is set to select 2000 or low number of voxels. On the other hand, we need to specify the a rang for the

second phase. In other words, the number of voxels that are going to select in the multivariate voxel selection method by RFE must be specified. One obvious way to reduce the number of voxels iteratively by specific range. For example, after choosing 2000 voxels by one of the univariate methods in the phase A, we can iteratively eliminate 5 voxels in each iteration by RFE in the phase B. However, tens of thousands classification performances would be examine by following this approach, and the time complexity of STMM would increase. Additionally, the purpose multivariate voxel selection method, RFE, in the phase B is not the selection of voxels directly. Instead,it is designed to select the most discriminative voxel by examining the output of univariate voxel selection methods in the multivariate form. Therefore, the number of voxels to be eliminated in the second phase of architecture is set to be 25 percent of the numbers of selected voxels in the first phase. We choose five combinations of numbers of selected voxels for ANOVA/RFE, and these combinations are (2000/1500), (1500/1125), (1200/900), (1000/700) and (850/640) to measure the performance of the proposed architecture (STMM).

The results of first and second phases of first path (first ANOVA, then RFE) are shown in the first and second column of Fig. 4.16 and 4.17. It is observed that there is no significant performance improvement in the second phase over the first one.

The third column in Figures 4.16 and 4.17 shows the results of Temporal Mesh Model (TMM). Due to the reason which is discussed in the previous section, 100 local neighbours are considered for the *seed voxel*. Except few increases, the classifier performances decreases in most of the cases in TMM compared to the first two phases. The performance decrease may have two reasons. On the one hand, the dimension of feature space increases 100 times by TMM due to consideration of estimated arc weight as new features instead of voxels. For instance, in our case, the dimension of input space (the output of second phase) increases from 1500 to 150000 in the case of 2000/1500 combination of voxels for ANOVA/RFE, which causes curse of dimensionality problem. On the other hand, although Onal et al. [44] optimized the number of neighbours for each voxel in the classical local mesh model, but TMM estimates the arc weights under the assumption of fix number of neighbours for all voxels. As mentioned previously, this may contradict with the natural functioning of the brain which the connections between the voxels can vary.

In order to obtain a sparse representation, ANOVA based feature selection is used to select and prune the most valuable and discriminative arc weights among all of the estimated ones. Particularly, this is called "Pruning Edges" in proposed architecture (STMM) (see Phase C.2 in Fig. 3.1). The results are shown in the right part (from the 5^{th} column to the last column) of Fig. 4.16 and Fig. 4.17 under the *"ANOVA on TMM"* label.

The number of whole brain voxels which are used in this study for each subject is nearly 20000 voxels. We used the univariate and multivariate voxel selection methods in phase A and phase B to select voxel in order to reduce the dimension. However, the TMM in the phase C.1 increases the dimension 100 times, because the number of neighbours is set to be 100. Due to the fact that we had originally about 20000 voxels or dimensions, therefore, the arc weights are pruned in specific ranges less than the 20000 which are shown in the figures.

However, the optimal performance of the pruned edges (presented under the *"ANOVA on TMM"*) are shown in the fourth column under the *"Max"* label which represents the optimum performance of STMM. *"Max"* is simply taken to be highest performance in overall ANOVA/RFE voxel number combinations. This is done in order to compare the classification performance of ANOVA, RFE, TMM and STMM. Almost in all of the cases, the *"Max"* surpass TMM's performances, and this shows the effectiveness of feature selection after the implementation of TMM. Furthermore, the results of KNN classifier (that is presented in Fig. 4.16) show that except the Subject005 (S5) which is nearly equal, *"Max"* passes the performances of first and second phases in Subject002, Subject003 and Subject006. However, the similar classification performance trend is shown in the case of SVM at Fig. 4.17.

As you noticed, there are hundreds of classification performances in this study. Therefore, we choose the *bar plot* to represent the performances. But, the quantitative illustration of the results could be beneficial in order to examine and see the advantage of the proposed architecture (STMM). The MVPA and obtained optimum results of both classifiers (KNN and SVM) for all subjects up to this point are summarized in Table 4.3. The optimum performances of MI, ANOVA and RFE are chosen to be the maximum performances of feature selection methods in Tables 4.1 and 4.2.

Additionally, the right side of the table (under the ANOVA/RFE title) show the optimal performance of ANOVA feature selection on the TMM, and the maximum of these performances are shown under the *"Max(Opt)"* label which are calculated by simply taking the maximum performances of five ANOVA/ RFE combinations. The *"Max(Opt)"*, the optimal performance of STMM, surpass the feature selection methods and MVPA results.

Up to now, we present only the STMM performances by using ANOVA to specify the TMM method. We accomplish similar results for the by MI in the Phase A of STMM, in Figures 4.18 and 4.19. Additionally, like the first path, the summary results of second path are shown in Table 4.4. However, it is notable that the optimal number of features (shown in Tables 4.3 4.4 inside the parenthesis under the label of "Opt ANO. on TMM") for SVM are higher than KNN. In other words, it seems that SVM gives its optimal results in the higher dimension compared to KNN. We will survey this phenomena in the next section (Visualizing with tSNE).

4.6 Visualizing with tSNE

2D maps of the data points from the high dimensional feature space (in the order of thousands) is not a trivial task. Maaten et al. [33] and [34] showed that t-distributed Stochastic Neighbor Embedding (tSNE) properly can visualize the high dimensional feature space in 2D maps. In this section, the results of implementation of tSNE on different phases of proposed architecture , STMM, is presented. Figures 4.20,4.21,4.22 and 4.23 contains the 2D visualization of whole brain MVPA and the results of three phases when the optimal classification performances are obtained for different subjects. It is notable that tSNE is an unsupervised methods, and the color and label of data are added to the output of tSNE to have detectable images.

The images labelled with (a) in Figures 4.20, 4.21, 4.22 and 4.23 show the results of MVPA for different subjects. It seems that the data points are randomly and non linearly distributed in the whole brain MVPA feature space. Most probably, this happens due to the noise caused by redundant features/voxels. However, the data points after voxel selection are somehow more separated from each other compared to the

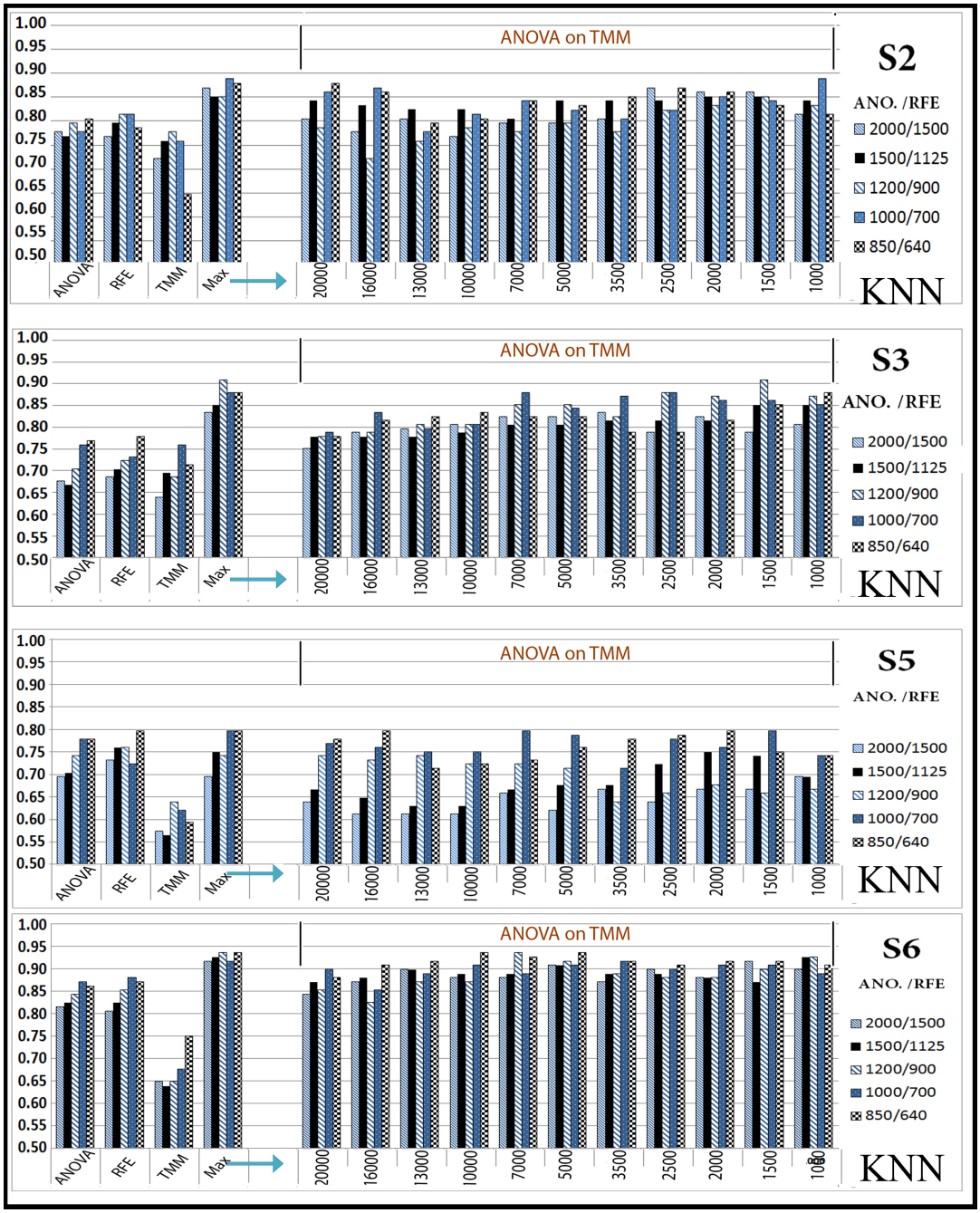

Figure 4.16: The KNN Performance using ANOVA as univariate voxel selection in the Phase A. Five different voxel combinations are used for Phase A and Phase B which are shown in different colors and labelled under the Subjects (S*) label on the right side of figures. The first, second and third columns show the KNN classification performances of Phase A,B and C.1 (See Fig. 3.1). The results of Phase C.2 ("Pruning Edges") are shown in the column 5 to the last one (under the label of "ANOVA on TMM") which each column shows different number of selected arc weights (from 20000 to 1000) using ANOVA. The optimum results of "ANOVA on TMM" is shown in the fourth which is labelled as "Max".

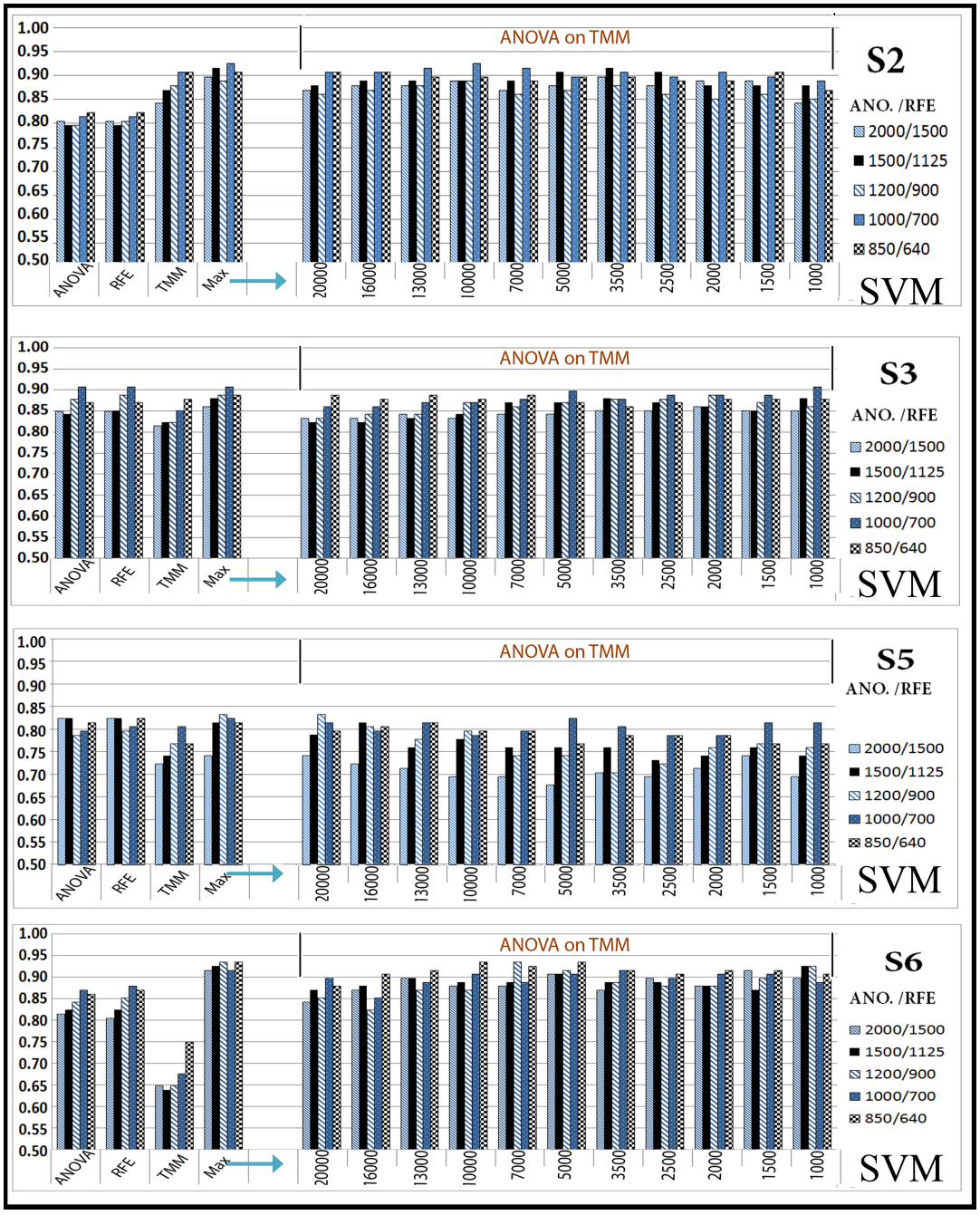

Figure 4.17: The SVM Performance using ANOVA as univariate voxel selection in the Phase A. Five different voxel combinations are used for Phase A and Phase B which are shown in different colors and labelled under the Subjects (S*) label on the right side of figures. The first, second and third columns show the SVM classification performances of Phase A,B and C.1 (See Fig. 3.1). The results of Phase C.2 ("Pruning Edges") are shown in the column 5 to the last one (under the label of "ANOVA on TMM") which each column shows different number of selected arc weights (from 20000 to 1000) using ANOVA. The optimum results of "ANOVA on TMM" is shown in the fourth which is labelled as "Max".

Table 4.3: The summary of optimal performances up to this point and following the first path when ANOVA is used in the Phase A as the univariate voxel selection. The results under the labels of MVPA, MI, ANOVA and RFE are the optimum performance of classifiers obtained without implementation of STMM architecture. In other words, these results are the maximum performances of Tables 4.1 and 4.2. The performance under the "Opt ANO. on TMM" shows the optimal quantitative performances of Figs. 4.16,4.17,4.18 and 4.19

						ANOVA/ RFE				
KNN (MI/ANOVA/RFE Max Performances)						2000/1500	1500/11250	1200/900	1000/700	850/640
	MVPA	MI	ANOVA	RFE	Max(Opt)	Opt ANO. on TMM				
Subject002	0.58	0.84	0.83	0.83	0.89	0.87 (2500)	0.85(2000)	0.85(1500)	0.89(1000)	0.88(20000)
Subject003	0.56	0.77	0.83	0.84	0.91	0.83 (3500)	0.85(1500)	0.91(1500)	0.88(7000)	0.88(1000)
Subject005	0.54	0.78	0.79	0.78	0.80	0.69 (1000)	0.75(2000)	0.74(13000)	0.80(1500)	0.80(16000)
Subject006	0.59	0.87	0.87	0.86	0.94	0.92(1500)	0.93(1000)	0.94(7000)	0.92(3500)	0.94(10000)
SVM (MI/ANOVA/RFE Max Performances)						2000/1500	1500/11250	1200/900	1000/700	850/640
	MVPA	MI	ANOVA	RFE	Max(Opt)	Opt ANO. on TMM				
Subject002	0.72	0.82	0.85	0.81	0.93	0.90(3500)	0.92(3500)	0.89(10000)	0.93(10000)	0.91(20000)
Subject003	0.77	0.91	0.92	0.9	0.91	0.86(2000)	0.88(3500)	0.89(2000)	0.91(1000)	0.89(2000)
Subject005	0.81	0.81	0.82	0.82	0.83	0.74(20000)	0.81(16000)	0.83(20000)	0.82(5000)	0.81(16000)
Subject006	0.83	0.91	0.91	0.9	0.98	0.94(16000)	0.94(16000)	0.95(16000)	0.96(13000)	0.98(10000)

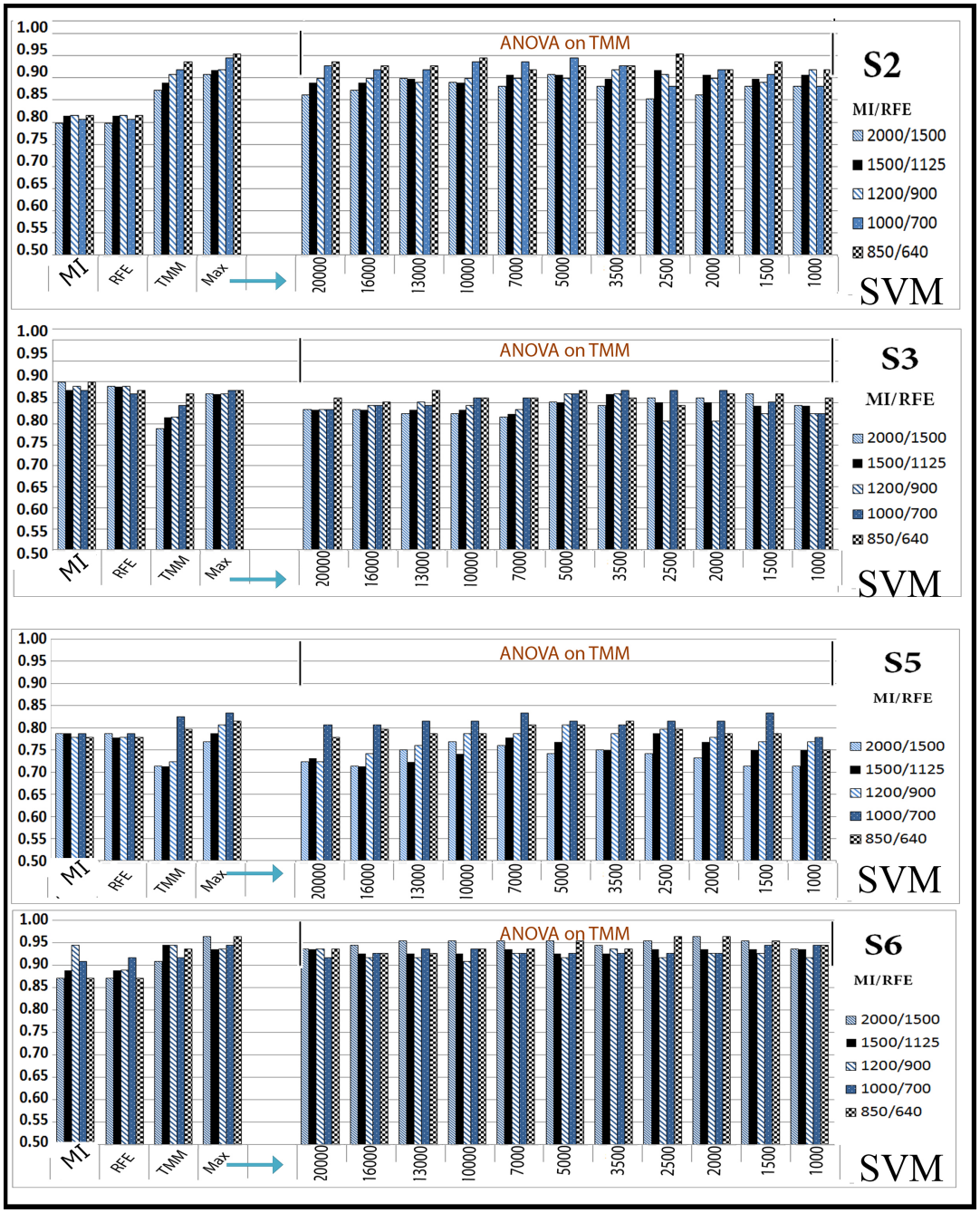

Figure 4.18: The KNN Performance using MI as univariate voxel selection in the Phase A. Five different voxel combinations are used for Phase A and Phase B which are shown in different colors and labelled under the Subjects (S*) label on the right side of the figures. The first, second and third columns show the KNN classification performances of Phase A,B and C.1 (See Fig. 3.1). The results of Phase C.2 ("Pruning Edges") are shown in the column 5 to the last one (under the label of "ANOVA on TMM") which each column shows different number of selected arc weights (from 20000 to 1000) using ANOVA. The optimum results of "ANOVA on TMM" is shown in the fourth which is labelled as "Max".

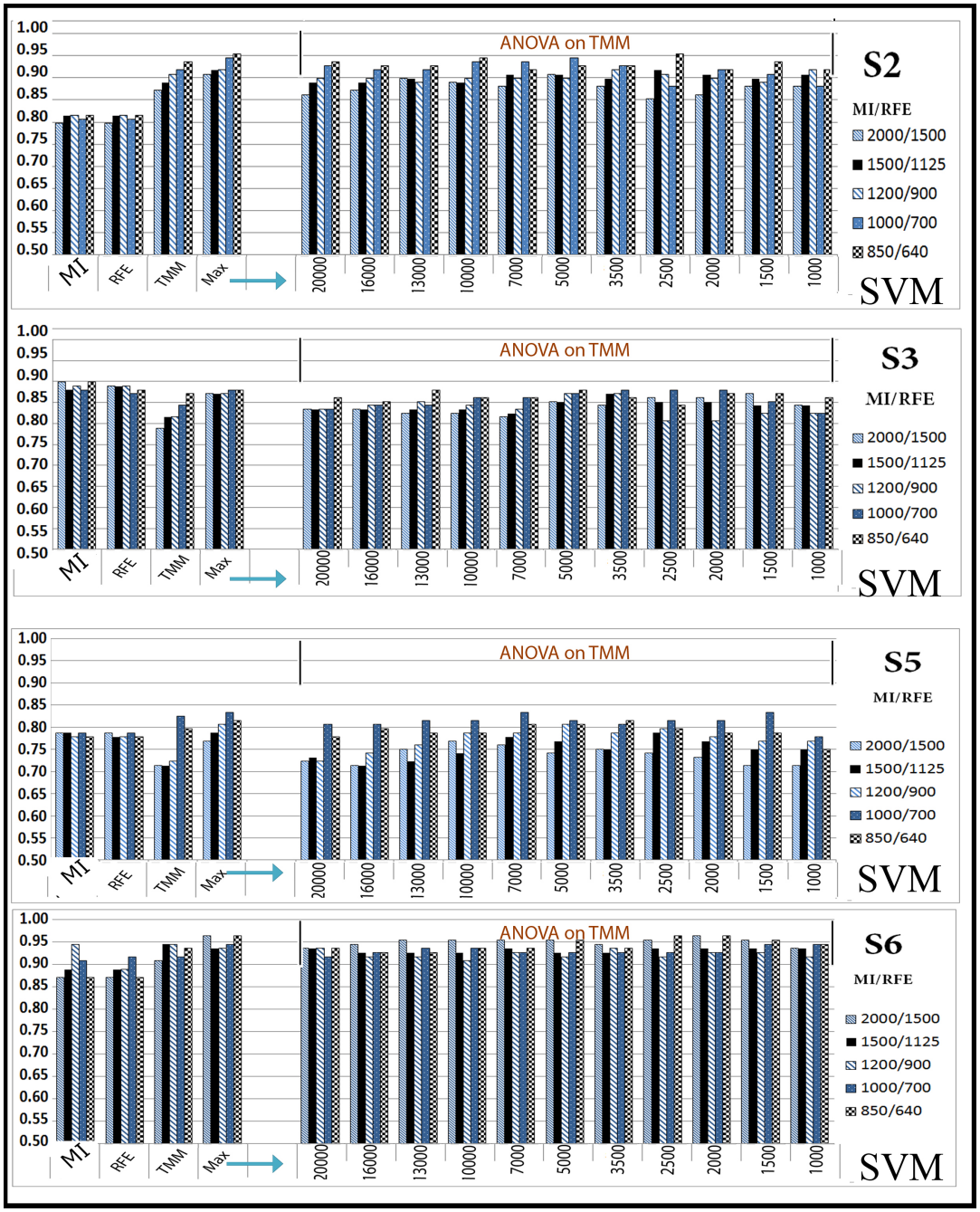

Figure 4.19: The SVM Performance using MI as univariate voxel selection in the Phase A. Five different voxel combinations are used for Phase A and Phase B which are shown in different colors and labelled under the Subjects (S*) label on the right side of the figures. The first, second and third columns show the SVM classification performances of Phase A,B and C.1 (See Fig. 3.1). The results of Phase C.2 ("Pruning Edges") are shown in the column 5 to the last one (under the label of "ANOVA on TMM") which each column shows different number of selected arc weights (from 20000 to 1000) using ANOVA. The optimum results of "ANOVA on TMM" is shown in the fourth which is labelled as "Max".

Table 4.4: The summary of optimal performances up to this point and following the second path when MI is used in the Phase A as the univariate voxel selection. The results under the labels of MVPA, MI, ANOVA and RFE are the optimum performance of classifiers obtained without implementation of STMM architecture. In other words, these results are the maximum performances of Tables 4.1 and 4.2. The performance under the "Opt ANO. on TMM" shows the optimal quantitative performances of Figs. 4.16,4.17,4.18 and 4.19

						MI/ RFE				
KNN (MI/ANOVA/RFE Max Performances)						2000/1500	1500/11250	1200/900	1000/700	850/640
	MVPA	MI	ANOVA	RFE	Max(Opt)	Opt ANO. on TMM				
Subject002	0.58	0.84	0.83	0.83	0.95	0.89(10000)	0.89(7000)	0.86(3500)	0.95(2500)	0.93(3500)
Subject003	0.56	0.77	0.83	0.84	0.88	0.85(1500)	0.88(3500)	0.83(3500)	0.87(1000)	0.87(1500)
Subject005	0.54	0.78	0.79	0.78	0.84	0.73(20000)	0.81(3500)	0.84(2500)	0.83(2000)	0.79(7000)
Subject006	0.59	0.87	0.87	0.86	0.95	0.90(3500)	0.88(7000)	0.90(2500)	0.90(1500)	0.95(1500)
SVM (MI/ANOVA/RFE Max Performances)						2000/1500	1500/11250	1200/900	1000/700	850/640
	MVPA	MI	ANOVA	RFE	Max(Opt)	Opt ANO. on TMM				
Subject002	0.72	0.82	0.85	0.81	0.95	0.91(5000)	0.92(2500)	0.92(3500)	0.94(10000)	0.95(2500)
Subject003	0.77	0.91	0.92	0.9	0.88	0.87(1500)	0.87(3500)	0.87(5000)	0.88(3500)	0.88(13000)
Subject005	0.81	0.81	0.82	0.82	0.83	0.77(10000)	0.79(2500)	0.81(5000)	0.83(7000)	0.81(7000)
Subject006	0.83	0.91	0.91	0.9	0.96	0.96(2000)	0.94(20000)	0.94(20000)	0.94(13000)	0.96(2500)

whole brain data. This is shown in the sub-images labelled with (b) and (c) in the figures. The labelled images with (b) show the result of tSNE on the optimum results of first phase (univariate voxel selection particularly ANOVA) which is obtained at 1000 voxels. The (c) sub-images illustrate the optimum results of RFE obtained by selection of 25 percent of 850 voxels which had been selected by ANOVA.

The (d) sub-images show the 2D visualization after the implementation of Temporal Mesh Model on the selected voxels at the second phase. Again, the results are shown according to the optimal classification performances. The presented results are obtained when 700 voxels are selected by RFE among the previously 1000 selected voxels by ANOVA. However, clearly, it can be seen the negative effect of increase in the feature space by TMM in the (d) sub-images. The last two sub-images (e) and (f) show the 2D visualization of data points after the pruning of TMM's arc weights (see phase C.2 "Pruning Edges" at Fig. 3.1). In fact these two sub-images are the output of the proposed architecture. The 2D maps are obtained at the optimal performance which is the maximum classification performance of SVM and KNN classifier. It is clear that the data points are almost linearly separable, and the distance between classes (flowers and birds) is higher than previous ones. Additionally, it is clear that a classifier such as KNN or SVM will not suffer the problem of "curse of dimensionality" to classify the classes. As you note, SVM classifier has higher classification performances compared to KNN, and their difference is high in some cases. For example, the results of SVM on TMM in Table 4.3 are higher than the results of KNN. This also manifest itself in the summary Table 4.4.

Except the (e) and (f) sub-images of Figures 4.20,4.21,4.22 and 4.23 the data is not separable. Therefore, one reason of high SVM performance can be the prosperity of SVM over KNN in the non-linear feature space of fMRI data sets. Secondly, although the classification performances are obtained as the result of testing the classifiers, but they obtained in subjective based condition where the number of samples are small. Therefore, the classifiers may be over-fitted. However, the proposed architecture decreases second reason by two facts. The first fact is that the architecture changes the feature space to almost linearly separable space this is shown by the tSNE visualization of the feature space at the (e) and (f) sub-images. Additionally, the differences between the results KNN and SVM classifiers are lower than compared to STMM.

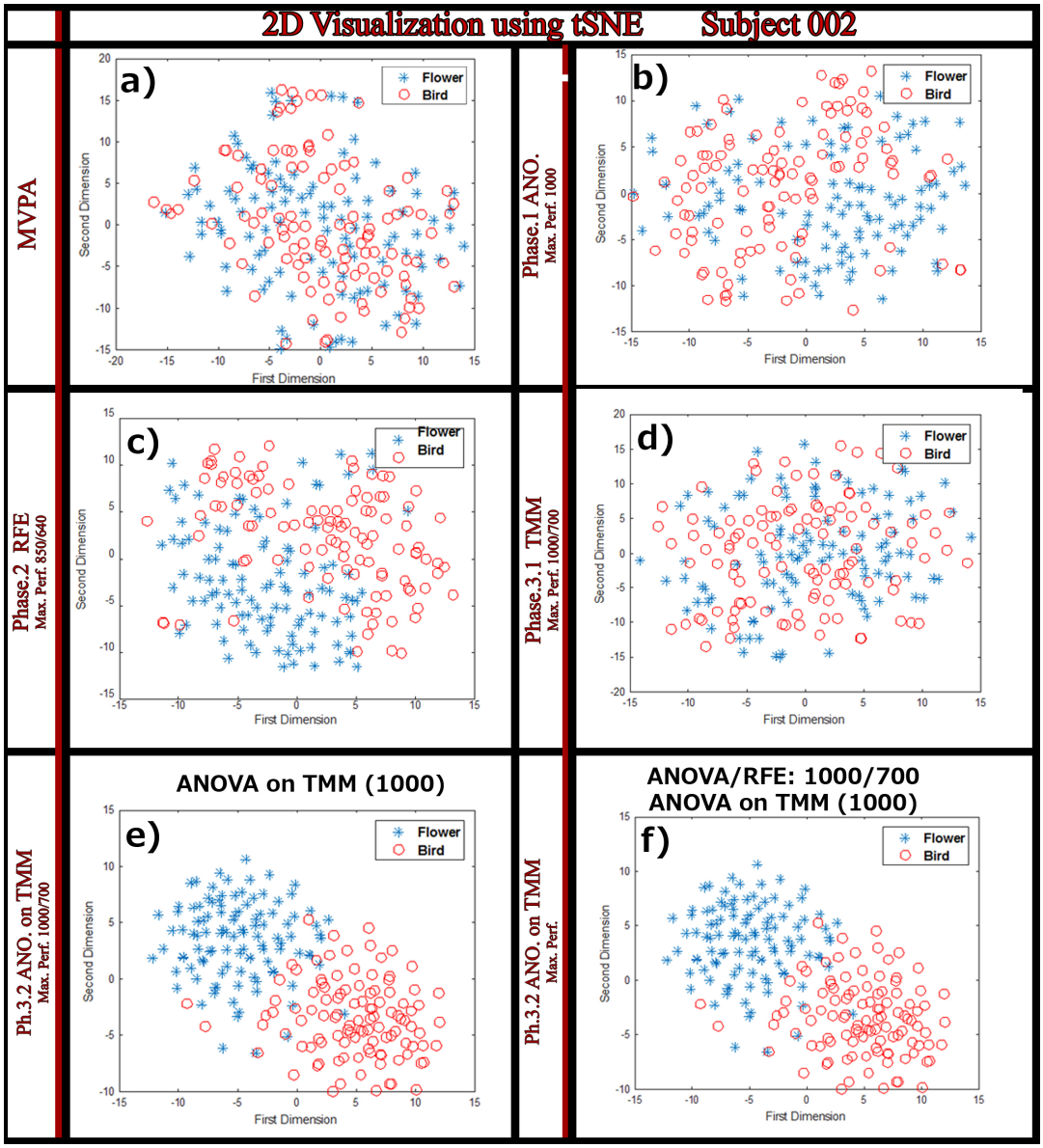

Figure 4.20: tSNE visualization of Subject 002. a) 2D visualization of whole brain multivariate pattern analysis (MVPA). b) tSNE result of phase A in the proposed architecture when optimal classification performance is obtained. At 1000 voxels selected by ANOVA. c) The 2D map of second phase at the optimal performance. The optimal performance of second phase is obtained when 640 voxels are selected by RFE from the previously selected 850 voxels with ANOVA voxel selection method. d) tSNE implementation of the first part of phase C which is TMM. This result is obtained at the optimal classification performance when 1000/700 number of voxels are selected in ANOVA/RFE at A/B phases respectively. e) the 2D map of after ANOVA is used to prune the less informative arc weights of TMM. This result is obtained when optimal number of arc weights (1000 is shown on top of (e)) are chosen on the optimal TMM which is shown in (d). f) the tSNE 2D map of the overall optimal point of the "Pruning Edge" which is obtained when: first 1000 voxels are selected with ANOVA, then, 700 out of that voxels are selected using RFE, and, finally, 1000 arc weights are chosen using ANOVA feature selection method. Note that the results of part e) and f) are similar for this subject, because both maps are obtained in the optimal 1000/700/1000 combination feature numbers for phases A/B/C2, respectively.

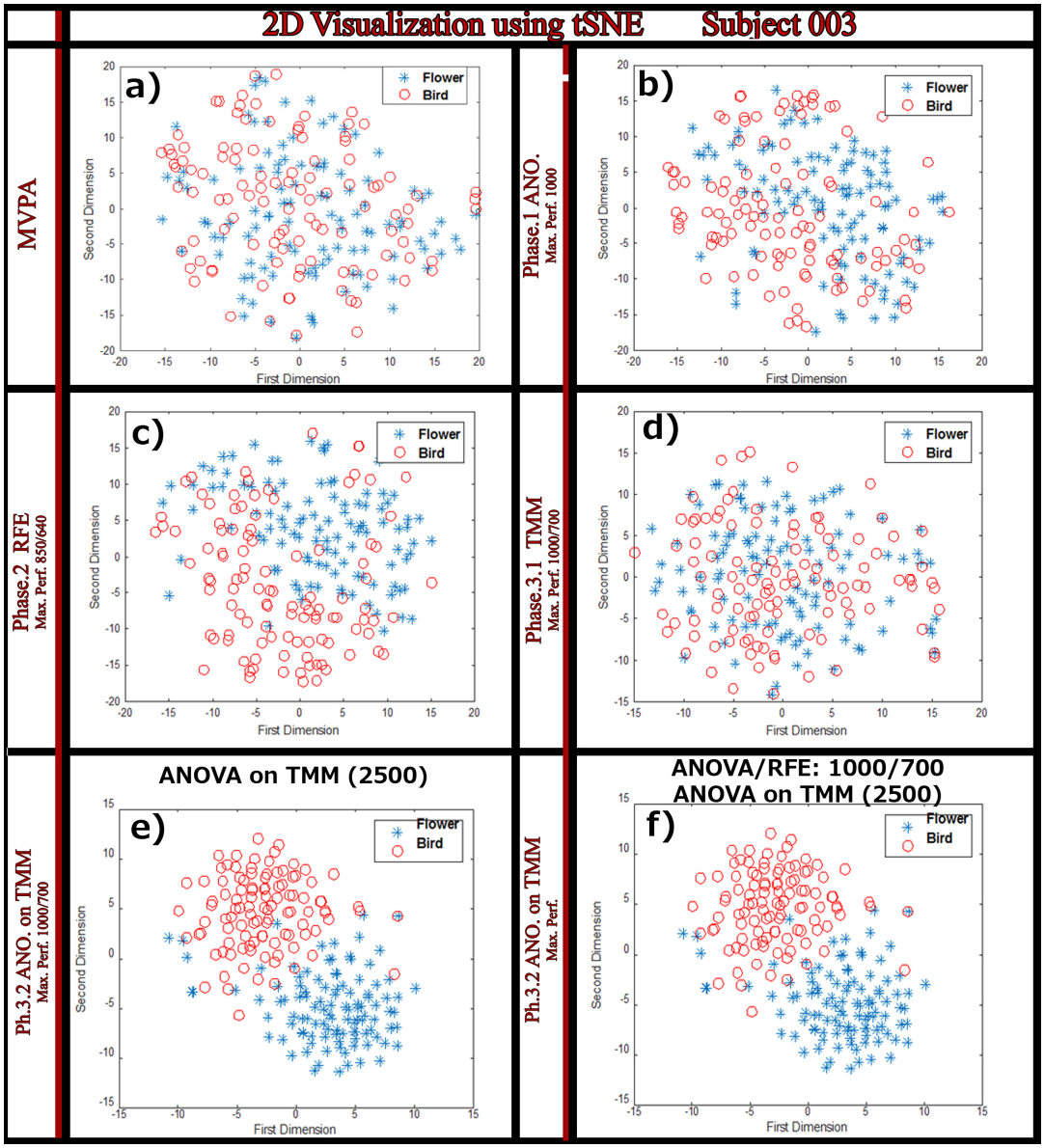

Figure 4.21: tSNE visualization of Subject 003. a) 2D visualization of whole brain multivariate pattern analysis (MVPA). b) tSNE result of phase A in the proposed architecture when optimal classification performance is obtained. At 1000 voxels selected by ANOVA. c) The 2D map of second phase at the optimal performance. The optimal performance of second phase is obtained when 640 voxels are selected by RFE from the previously selected 850 voxels with ANOVA voxel selection method. d) tSNE implementation of the first part of phase C which is TMM. This result is obtained at the optimal classification performance when 1000/700 number of voxels are selected in ANOVA/RFE at A/B phases respectively. e) the 2D map of after ANOVA is used to prune the less informative arc weights of TMM. This result is obtained when optimal number of arc weights (2500 is shown on top of (e)) are chosen on the optimal TMM which is shown in (d). f) the tSNE 2D map of the overall optimal point of the "Pruning Edge" which is obtained when first 1000 voxels are selected with ANOVA, then, 700 out of that voxels are selected using RFE, and, finally, 2500 arc weights are chosen using ANOVA feature selection method. Note that the results of part e) and f) are similar for this subject, because both maps are obtained in the optimal 1000/700/2500 combination feature numbers for phases A/B/C2, respectively.

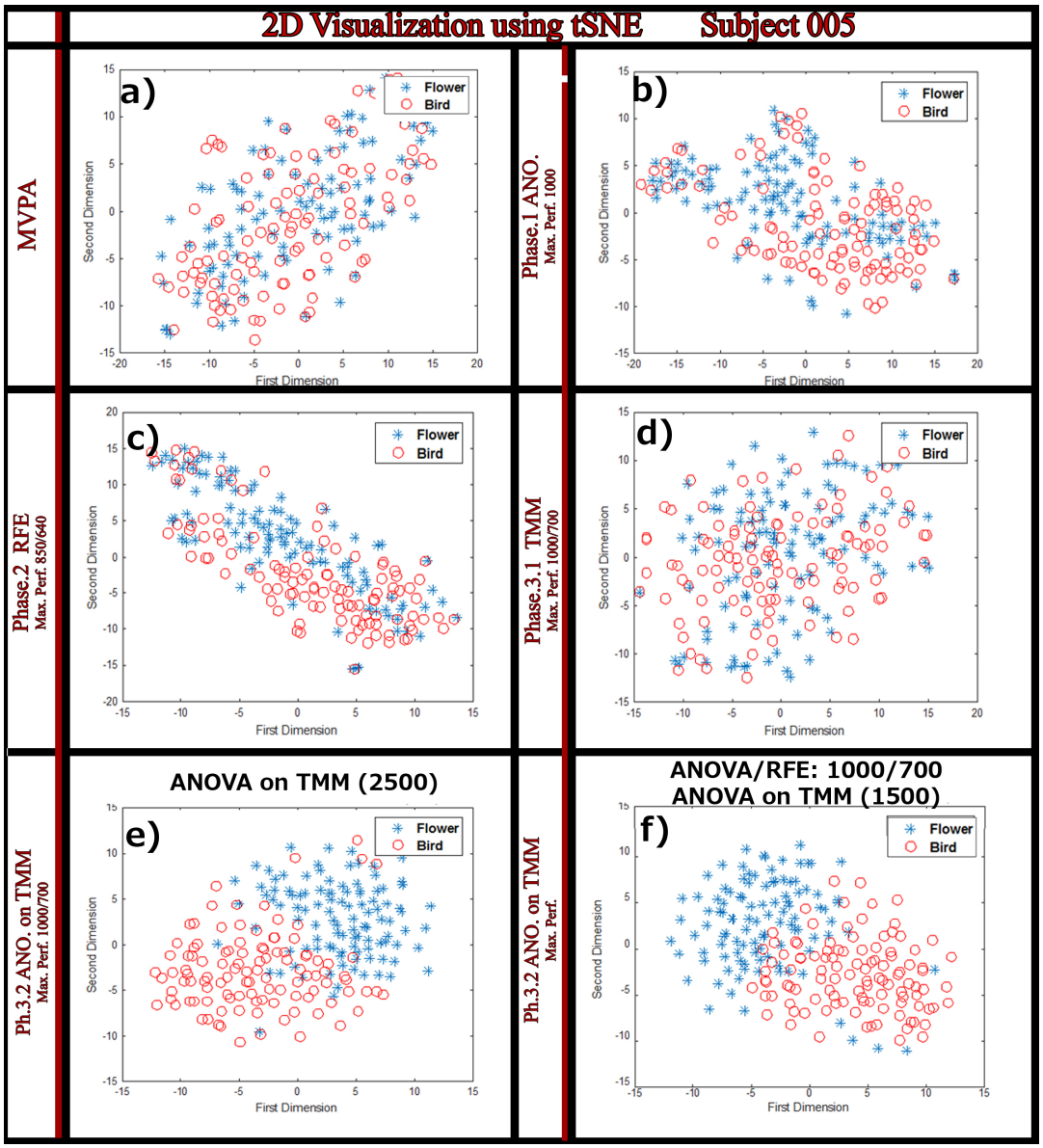

Figure 4.22: tSNE visualization of Subject 005. a) 2D visualization of whole brain multivariate pattern analysis (MVPA). b) tSNE result of phase A in the proposed architecture when optimal classification performance is obtained. At 1000 voxels selected by ANOVA. c) The 2D map of second phase at the optimal performance. The optimal performance of second phase is obtained when 640 voxels are selected by RFE from the previously selected 850 voxels with ANOVA voxel selection method. d) tSNE implementation of the first part of phase C which is TMM. This result is obtained at the optimal classification performance when 1000/700 number of voxels are selected in ANOVA/RFE at A/B phases respectively. e) the 2D map of after ANOVA is used to prune the less informative arc weights of TMM. This result is obtained when optimal number of arc weights (2500 is shown on top of (e)) are chosen on the optimal TMM which is shown in (d). f) the tSNE 2D map of the overall optimal point of the "Pruning Edge" which is obtained when first 1000 voxels are selected with ANOVA, then, 700 out of that voxels are selected using RFE, and, finally, 1500 arc weights are chosen using ANOVA feature selection method.

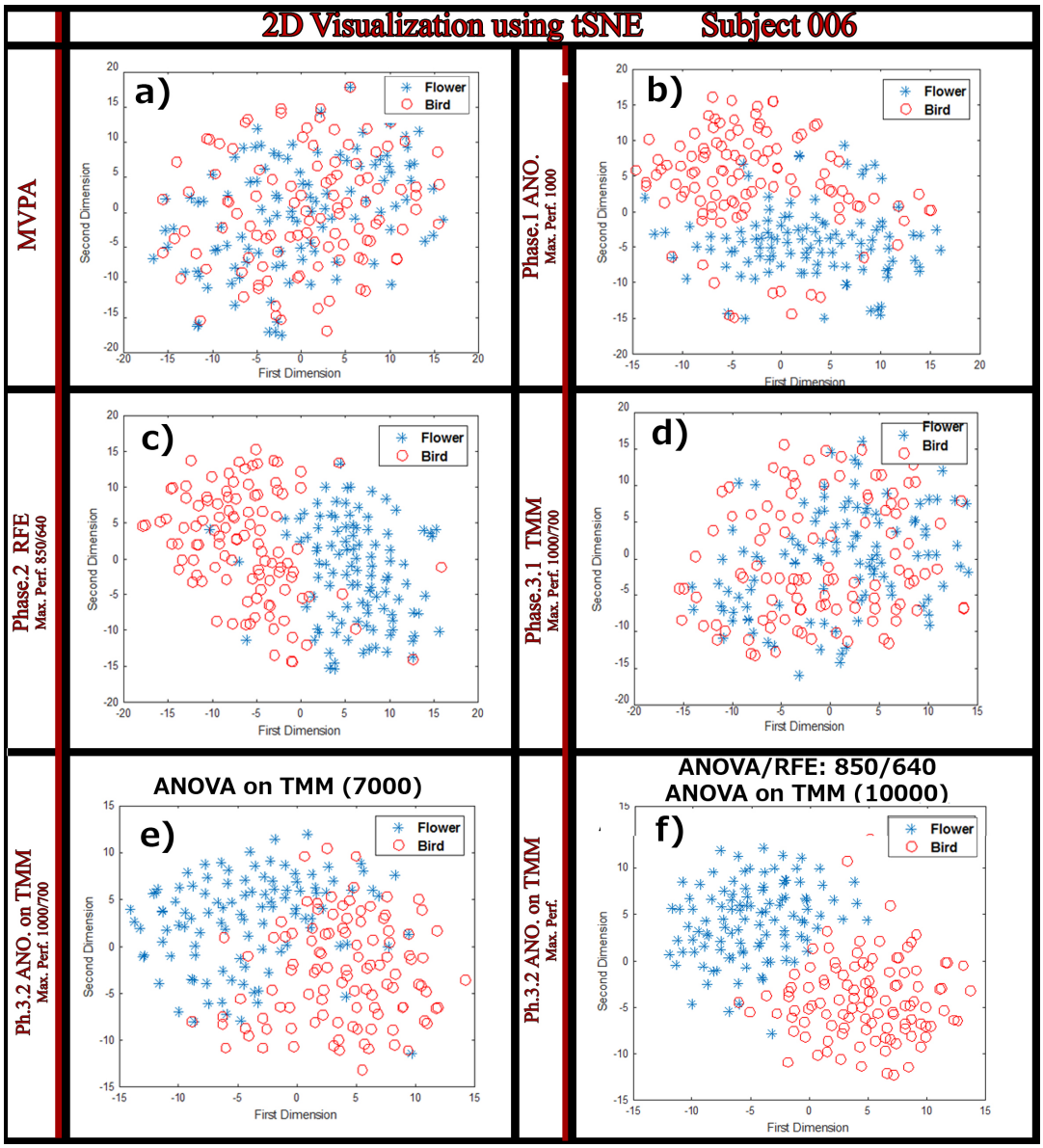

Figure 4.23: tSNE visualization of Subject 005. a) 2D visualization of whole brain multivariate pattern analysis (MVPA). b) tSNE result of phase A in the proposed architecture when optimal classification performance is obtained. At 1000 voxels selected by ANOVA. c) The 2D map of second phase at the optimal performance. The optimal performance of second phase is obtained when 640 voxels are selected by RFE from the previously selected 850 voxels with ANOVA voxel selection method. d) tSNE implementation of the first part of phase C which is TMM. This result is obtained at the optimal classification performance when 1000/700 number of voxels are selected in ANOVA/RFE at A/B phases respectively. e) the 2D map of after ANOVA is used to prune the less informative arc weights of TMM. This result is obtained when optimal number of arc weights (7000 is shown on top of (e)) are chosen on the optimal TMM which is shown in (d). f) the tSNE 2D map of the overall optimal point of the "Pruning Edge" which is obtained when first 1000 voxels are selected with ANOVA, then, 700 out of that voxels are selected using RFE, and, finally, 10000 arc weights are chosen using ANOVA feature selection method.

4.7 Discussion

This chapter covers the experiments that are designed to measure the performances of proposed architecture known as sparse temporal mesh model (STMM) and compared it to popular feature selection methods given in Chapter 2.

The first group of experiment include the analysis namely: "Intersection", "Anatomical" and "p value analysis for TMM". The result of "Intersection" and "Anatomical" analyses show that ANOVA, MI and RFE feature selection methods can be consider as the valid methods for voxel selection. This validity comes from two facts. Firstly, the results of "Intersection" analysis show that most of the selected voxels are common between the voxel selection mehtods, and the second fact is that the results of "Anatomical" analysis shows the relationships between fMRI recordings and neuroscientific findings. In other words, we expected that the active and discriminative voxel would be in occipital and temporal lobs based on the fMRI recordings. This is observed where the voxel selection methods selected the voxels from these lobs (see Figures 4.4,4.5, 4.10 and 4.11)

In the second set of experiments, the impact of voxel selection methods on "brain decoding" problem is measured by the accuracy performance of two well known classifiers, namely KNN and SVM. This experiment illustrates the effectiveness of voxel selection methods on "brain decoding". All three voxel selection methods, ANOVA, MI and RFE, increase the performance of the classifiers by eliminating the less informative voxels.

The performance of proposed architecture, STMM, is measured in the third set of experiments.KNN and SVM classifiers are used for brain decoding purpose. This experiment consists of measuring the performance of classifiers in the various phases of STMM. It is shown that the classification performances of final phase in STMM, phase C.2 "Pruning Edges", commonly higher than the performances of other phases and the results of second types of experiments. Finally, the 2D maps of all three phases in STMM are examined to see the effect of each phase in the feature space. The result of this experiment shows the effectiveness of phase A over MVPA, and it also illustrates the effectiveness of phase C.2 over all phases and MVPA.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In the first part of this chapter, the overall outcomes of the proposed architecture, STMM, is discussed. In the next part, the possible future developments of proposed architecture is covered to increase the accuracy of brain decoding.

5.1 Discussion on STMM architecture

In this study, a new architecture called sparse temporal mesh model (STMM) is proposed which consists of three phases based on univariate and multivariate analysis and a mesh model. STMM employs several feature selection methods and temporal mesh model (TMM) to increase the accuracy of brain decoding. The proposed new technique uses feature selection methods to select the voxels, then it implements TMM on the selected voxels in order to estimate the relationships between them. Additionally, it prunes the estimated arc weights of TMM in order to solve the problem of TMM. In order to visualize the effect of each phase on the feature space, the method known as t-distributed Stochastic Neighbor Embedding (tSNE) is used in STMM.

STMM is based on TMM which is proposed by Onal et al. [67]. TMM considers the discretized form of voxel's hemodynamic responses for estimation purpose. The purpose of this work is to increase the performance in brain decoding by decreasing the possible effects of "curse of dimensionality" problem. In order to testify the architecture, the data set which contains the visually stimulation of four participants are used. The results indicate the necessary of dimension reduction in the decoding of cognitive states.

STMM is composed of three phases A, B and C. In phases A and B the most discriminative voxels are selected using both univariate and multivariate voxel selection methods. One of the univariate methods (MI or ANOVA) is used to select the most discriminative features or voxels among the whole brain voxels (which are in the order of tens of thousands). In phase *B*, following the first one, a multivariate voxel selection method known as RFE is used to eliminate unnecessary voxels among the output of first phase. This is done by examination of all feature space in the multivariate form. The aim of this phase is to consider the multivariate nature of brain during the voxel selection. Phase *C* consists of the implementation of TMM and pruning the edges. After the estimation of arc weights (which represents the relationships between voxels) using TMM, ANOVA feature selection method (used in the first phase) again used to prune and discard the useless features or arc weights. Each of discussed phases are connected to the visualization method known as tSNE, in order to have an idea about the change in the feature space.

The analyses of voxel selection methods showed that they are successful to find the informative voxels from both neuroscientific and brain decoding point of views. Due to the visual stimulation of participants in the fMRI recordings, it was expected that both occipital and temporal lobes to have relations with the recordings. This expectation comes from the functions of these two lobs where occipital deals with vision and temporal lobe contains visual memory function. Additionally, the analysis called "Intersection" showed us that all of the discussed feature selection methods can commonly select the same voxels.

The performance of STMM is compared to previous brain decoding methods known as MVPA and TMM. The classification performances indicate the successfulness of the architectures over the MVPA in all of the cases. The reason of increase in the classification performance of STMM over MVPA is due to the two main reasons. First, the most discriminative voxels are identified in the phase A and B. In other words, the noisy voxels or features are eliminated and the feature space become more separable. Second, it uses the temporal mesh model (TMM) and estimates the relationships

between the voxels. On the other hand, the reason of increase in the classification performances of STMM compared to the TMM is that the estimated arc weights are pruned in the phase C.2. The pruning edges again helps to have more informative and separable feature space.

In the first path, ANOVA feature selection method is used as univariate voxel selection method. In this path, the average KNN and SVM accuracies (for four participants) showed that the performances of STMM pass the whole brain MVPA (with out voxel selection) by 32% in the case of KNN and 13% in the cause of SVM. However, in the second path where MI is used in phase A, the STMM average KNN and SVM performances of four subjects pass the average performances of whole brain MVPA by 34% and 12% respectively (see Tables 4.3 and 4.4).

Apart from the classification performances, the visualization results of tSNE, as a part of STMM, indicate the necessity of voxel selection or generally dimension reduction in brain decoding. The 2D maps of STMM outputs show that the feature space is more separable compared to the classic MVPA. Additionally, the arc weight selection using a feature selection method would result in the change of non-linear feature space into linearly one which the distance between class distributions increases. As a result, it decreases the possibility of generating an overfitted model.

5.2 Future Work

In this work, recursive feature elimination (RFE) is used in phase B of STMM in order to eliminate unnecessary voxels among the ones which are selected in phase A. However, a new multivariate voxel selection method can be designed based on the functioning of the brain. For instance, the new multivariate voxel selection method can also count the locality of voxels during their elimination because neuroscientific information shows us that the locally near voxels have strong relationships.

In this thesis, STMM examines only the spatially local neighbouring voxels. Meaning that, the neighbouring voxels of the *seed voxel* are determined spatially, and the neighbourhood voxels are selected to be the ones which are spatially closer to the *seed*

voxel. However, Firat et al. [60] illustrated that counting on the functionally nearest neighbours of the *seed voxel* increases the predictability accuracy of brain decoding. The idea of functional neighbouring voxels which have higher correlations with the *seed voxel* can be used as an alternative to the spatially local neighbouring voxels used in TMM.

Finally, it seems that more advanced graph based algorithms can be used to prune the arc weights in phase C.2 ("Pruning Edges") of STMM. Therefore, the examination of popular graph based algorithm or proposing such an algorithm would be beneficial in the case of arc weight pruning.

REFERENCES

[1] *Tal Geva, MD. Magnetic Resonance Imaging: Historical Perspective. Journal of Cardiovascular Magnetic Resonance, Department of Cardiology, Children's Hospital, and the Department of Pediatrics, Harvard Medical School, Boston, MA, USA, 8 (4) , Pages 573-580, 2006.*

[2] *S. A. Huettel, A. W. Song,and G. McCarthy. Functional Magnetic Resonance Imaging (2 ed.), Massachusetts: Sinauer, ISBN 978-0-87893-286-3, 2009.*

[3] *S. Ogawa, T.-M. Lee, A. R. Kay, and D. W. Tank. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. Proc. Natl. Acad. Sci. USA, Volume. 87, pp. 9868-9872, 1990.*

[4] *A. Lindquist. Martin ,and Tor D. Wager. Principles of functional Magnetic Resonance Imaging. Department of Biostatistics Johns Hopkins University and Department of Psychology and Neuroscience University of Colorado at Boulder.*

[5] *http://web.csulb.edu/ cwallis/482/fmri/fmri.html. Last accessed September, 2015.*

[6] *Sun/Earth Comparison, Author/Curator: Dr. David R. Williams, NSSDCA, Mail Code 690.1 NASA Goddard Space Flight Center Greenbelt, MD 20771*

[7] *http://www.nibib.nih.gov/science-education/science-topics/magnetic-resonance-imaging-mri. Last accessed September, 2015.*

[8] *Kamil Uludag, David J. Dubowitz,and Richard B. Buxton. BASIC PRINCIPLES OF FUNCTIONAL MRI.*

[9] *L. Pauling, and C. D. Coryell. The magnetic properties and structure of hemoglobin, oxyhemoglobin and carbonmonoxyhemoglobin. In Proc. Natl. Acad. Sci. U.S.A., volume 22, pages 210 – 236, 1936.*

[10] *KR. Thulborn, JC. Waterton, PM. Matthews, and GK. Radda. Oxygenation dependence of the transverse relaxation time of water protons in whole blood at high field. Biochim Biophys Acta, 714:265-270, 1982.*

[11] *S. Ogawa, TM. Lee, AS. Nayak, and P. Glynn. Oxygenation-sensitive contrast in magnetic resonance image of rodent brain at high magnetic fields. Magn Reson Med, 14:68-78, 1990.*

[12] *S. Ogawa, and T.-M. Lee. Magnetic resonance imaging of blood vessels at high fields: In vivo and in vitro measurements and image simulation. Magnetic Resonance in Medicine, 16 (1):9-18, 1990.*

[13] *S. Ogawa, DW. Tank, R. Menon, JM. Ellermann, SG. Kim, H. Merkle, and K. Ugurbil. Intrinsic signal changes accompanying sensory stimulation. functional brain mapping with magnetic resonance imaging. Proc Natl Acad Sci USA 89 (13):5951-5955, 1992.*

[14] *R. Turner, D. Le Bihan, CT.W. Moonen, D. Despres, and J. Frank. Echo-planar time course MRI of cat brain oxygenation changes. Magn Reson Med 22:159-166, 1991.*

[15] *C. S. Roy, and C. S. Sherrington. On the regulation of the blood-supply of the brain. Journal of Physiology, 11(1 -2): [85]–108, 158-7-158-17, 1890.*

[16] *A. Lindquist. Martin, Ji Meng. Loh, Lauren Y. Atlas,and Tor D. Wager. Modeling the hemodynamic response function in fMRI: Efficiency, bias and mis-modeling, 45 (1 Suppl): S187-S198, 2009.*

[17] *D. Stansbury. fMRI in Neuroscience: Estimating Voxel Selectivity and the General Linear Model (GLM), November 2012.*

[18] *Google Speech of Prof. Fatos. T. Yarman Vural*

[19] *JS1. Kippenhan, WW. Barker, S. Pascal, J. Nagel, and R. Duara. Evaluation of a neural-network classifier for PET scans of normal and Alzheimers Disease, J Nucl Med, 33(8): 1459-67, 1992.*

[20] *Neils J.S. Mørch, Ulrik. Kjems, Lars Kai. Hansen, C. Svarer, I. Law, B. Lautrup, S. Strother, and K. Rehm. Visualization of Neural Networks using Salienc maps, Neural Networks. Proceedings., IEEE International Conference on volume 4 IEEE, Pages 2085-2090, 1995.*

[21] *Z. Yang, F. Fang, and X. Weng. Recent developments in multivariate pattern analysis for functional mri. Neuroscience Bulletin, 28(4):399 – 408, 2012.*

[22] *Peter. A. Bandettini. What's new in neuroimaging methods?. Ann N Y Acad Sci, 1156: 260–293, 2009.*

[23] *Y. Kamitani, and F. Tong. Decoding the visual and subjective contents of the human brain. Nat Neurosci, 8: 679–685, 2005.*

[24] *J. V. Haxby, M. I. Gobbini, M. L. Furey, A. Ishai, J. L. Schouten, and P. Pietrini. Distributed and overlapping representations of faces and objects in ventral temporal corte.Science, 293(5539):2425 –30, 2001.*

[25] *K. A. Norman, S. M. Polyn, G. J. Detre, and J. V. Haxby. Beyond mind-reading: multivoxel pattern analysis of fmri data. TRENDS in Cognitive Sciences, 10(9):424 – 430 2006.*

[26] *M.H. Lee, C,D. Smyser, and J.S.Shimony. Resting State fMRI: A Review of Methods and Clinical Applications.AJNR AM J Neuroradiol 34:1866-72, 2013.*

[27] *http://www.fil.ion.ucl.ac.uk/spm/. Last accessed September, 2015.*

[28] *N. Kriegeskorte, R. Goebel, and P. Bandettini, "Information-based functional brain mapping," PNAS, vol. 103, no. 10, pp. 3863 – 3868, 2006.*

[29] *M. Brett, J.-L. Anton, R. Valabregue, and J.-B. Poline. "Region of interest analysis using an spm toolbox," in 8th International Conference on Functional Mapping of the Human Brain, 2002.*

[30] *V. Michel, C. Damon, and B. Thirion.Mutual information-based feature selection enhances fMRI brain activity classification, Biomedical Imaging: From Nano to Macro. ISBI 2008. 5th IEEE International Symposium on Paris, Pages 592-595, 2008.*

[31] *TOM M. MITCHELL, R. HUTCHINSON, RADU S. NICULESCU, F. PEREIRA,and X. WANG. Learning to Decode Cognitive States from Brain Images. Kluwer Academic Publishers. Manufactured in The Netherlands. Machine Learning, 57, 145–175, 2004.*

[32] *G.E.Hinton, and ST. Roweis. "Stochastic neighbor embedding". Advances in neural information Processing Systems, 15, MIT Press, Cambridge, MA, 2002.*

[33] *L. Van der Maaten, and G. Hinton. "Visualizing data using tSNE" Journal of Machine Learning Research,9, 2579-2605, 2008.*

[34] *L.J. Van der Maaten, E. O. Postma, and H. J. Van den Herik."Dimensionality reduction: A comparative review" , Journal of Machine Learning Research,10(1-41), 66-71, 2009.*

[35] *L.J. Van der Maaten. Google Talk on Visualizing Data Using t-SNE, June 24, 2013.*

[36] *J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science, 4; 295(5552): 2319- 23, 2002.*

[37] *Nonlinear Dimensionality Reduction by Locally Linear Embedding. Sam T. Roweis, and Lawrence K. Saul. Science 290 (5500): 2319-2323, 22 December 2000.*

[38] *I.T. Jolliffe. Principal Component Analysis Series: Springer Series in Statistics. 2nd ed. 2002.*

[39] *Frank Nielsen. Accuracy of Distance Metric Learning Algorithms. École Polytechnique, DMMT '09 Proceedings of the 2nd Workshop on Data Mining using Matrices and Tensors, ISBN: 978-1-60558-673-1, 2009.*

[40] *EP. Xing, Andrew Y. Ng, MI Jordan,and S Russell. Distance metric learning with application to clustering with side-information. Advances in Neural Information Processing System Conference, pages 505-512, 2002.*

[41] *S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood Components Analysis. J. GoldbergerAdvances in Neural Information Processing System, Volume 17, 2004.*

[42] *A. Globerson and S. Roweis. Metric Learning by Collapsing Classes, Neural Information Processing Systems 18 (NIPS'05). Pp. 451-458, 2005.*

[43] *Kilian Q. Weinberger, J. Blitzer, and Lawrence K. Saul. Distance Metric Learning for Large Margin. Nearest Neighbor Classification. In Y. Weiss, B. Schoelkopf, and J. Platt (eds.),Advances in Neural Information Processing Systems 18 (NIPS-18). MIT Press: Cambridge, MA, 2005.*

[44] *Itir Onal. Ms. Thesis,AN INFORMATION THEORETIC REPRESENTATION OF BRAIN CONNECTIVITY FOR COGNITIVE STATE CLASSIFICATION USING FUNCTIONAL MAGNETIC RESONANCE IMAGING, computer engineering department, Middle East Technical University, Ankara, Turkey, Sep.2013.*

[45] *T. Mitchell. Machine Learning. McGraw - Hill, 1997.*

[46] *VN. Vapnik. Statistical learning theory, 1998.*

[47] *N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot. "Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain," NeuroImage, 15(1), pp. 273–289, 2002.*

[48] *X. Chen and JC. Jeong. Minimum reference set based feature selection for small sample classifications. Proceedings of the 24th international conference on Machine learning, ACM, pages 153-160, 2007.*

[49] *Federico D. Martino, G. Valente, N. Staeren, J. Ashburner, R. Goebel, E. Formisano. Combining multivariate voxel selection and support vector machines for mapping and classification of fMRI spatial patterns. NeuroImage, 42 Pages 44-58, 2008.*

[50] *Henryk Blasinski. Voxel selection algorithms for fMRI.December 14, 2012.*

[51] *K. A. Norman, S. M. Polyn, G. J. Detre, and J. V. Haxby. Beyond mind-reading: multivoxel pattern analysis of fMRI data. Trends Cogn. Sci. Sep. 10 (9), 424–430, 2006.*

[52] *http://vassarstats.net/textbook/ch14pt1.html. Last accessed September, 2015.*

[53] *J. D. Haynes and G. Rees. Decoding mental states from brain activity in humans. Nat. Rev. Neurosci. 7 (7), 523–534, 2006.*

[54] *Chun-An Chou, K. Kampa, Sonya H. Mehta, Rosalia F. Tungaraza, W. Art Chaovalitwongse*, Senior Member, IEEE, and Thomas J. Grabowski.Voxel Selection Framework in Multi Voxel Pattern Analysis of fMRI Data for Prediction of Neural Response to Visual Stimuli. IEEE Transactions on Medical Imageing, vol. 33, No. 4, 2014.*

[55] *S. C. Shannon. "A mathematical theory of communication," The Bell Syst. Tech. J., vol. 27, pp. 379–423, 623–656, 1948.*

[56] *S. Kullback. Information Theory and Statistics. Mineola, NY: Dover, 1997.*

[57] *D. Xu and S. L. Albin, "Manufacturing start-up problem solved by mixed-integer quadratic programming and multivariate statistical modeling," Int. J. Prod. Res., vol. 40, no. 3, pp. 625–640, 2002.*

[58] *T. M. Phuong, Z. Lin et R. B. Altman. Choosing SNPs using feature selection. Proceedings. IEEE Computational Systems Bioinformatics Conference, CSB. IEEE Computational Systems Bioinformatics Conference, pages 301-309, 2005.*

[59] *M. Özay, I. Öztekin, U. Öztekin, F. T. Y. Vural. Mesh Learning for Classifying Cognitive Processes. 2012.*

[60] *O. Firat, M. Ozay, I. Onal, I. Oztekin, and F. T. Y. Vural. Functional mesh learning for pattern analysis of cognitive processes. In 12th IEEE International Conference on Cognitive Informatics and Cognitive Computing (ICCI*CC), 2003.*

[61] *Richard Ernest Bellman. Rand Corporation. Dynamic programming. Princeton University Press, 1957.*

[62] *Richard Ernest Bellman. Adaptive control processes: a guided tour. Princeton University Press, 1961.*

[63] *Richard O. Duda, Peter E. Hart, and David G. Stork. Pattern Classification, 2nd Edition, ISBN:0471056693, 2003.*

[64] *Christopher Bishop. Pattern Recognition and Machine Learning (Information Science and Statistics), 2007.*

[65] *T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning: Data Mining, Inference, and Prediction (2nd edition), 2009.*

[66] *P. P. Vaidyanathan. The Theory of Linear Prediction. Morgan and Claypool Publishers, 2008.*

[67] *I. Onal, M. Ozay, and F. T. Y. Vural. Modeling voxel connectivity for brain decoding. PRNI,Stanford, 2015.*

[68] *neuro.ceng.metu.edu.tr/fmri-prj/codes.html/. Last accessed September, 2015.*