UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Mari Liis Velner

Analyzing Activity of the Human Brain During Decision Making

Master's Thesis (30 ECTS)

Supervisor: Raul Vicente Zafra, PhD

Tartu 2019

Analyzing Activity of the Human Brain During Decision Making

Abstract:

The orbitofrontal cortex (OFC) is a region sitting at the front of the brain which function is not fully understood. However, it has been clearly implicated in decision making as shown by many neuroimaging studies over the last decades. Recent work by Saez et al. [1] has found evidence that OFC activity of high frequency (HFA) between 70-200 Hz is directly related to behavioral responses during decision making tasks. In particular, Saez et al. showed that some modulations of HFA correlated with the human choice and outcome in a simple betting game. Saez et al. conducted their analysis with univariate linear regression, predicting HFA values from one task-related parameter at a time to find electrodes which encode decision making information. This Thesis focused on extending these results and analyses of Saez et al. by applying multivariate methods to discover complex signals and important patterns in the neural data. For this, canonical correlation analysis and biclustering were used on 600 different datasets to find evidence of patterns in electrode responses and complicated combinations of behavioral responses encoded in the human brain signals. In addition, machine learning methods were used to analyze the patients' behavioral tendencies towards risk-taking in a gambling task and to predict task-related events such as winning, losing and gambling from the neural data. Moderate to good performance was achieved with most methods, but in-depth analysis is still necessary to gain a full understanding of how activity in orbitofrontal cortex gives rise to human behavior in decision making tasks.

Keywords:

Decision making, orbitofrontal cortex, machine learning, canonical correlation analysis, bi-clustering

CERCS: P170, P176, B110

Aju aktiivsuse analüüs otsuste tegemise vältel

Lühikokkuvõte:

Orbitofrontaalne ajukoor (OFC) on aju ees istuv piirkond, mille toimimist ei ole suudetud täielikult mõista. Siiski on see selgelt seotud otsuste tegemisega, nagu on näidatud paljudes viimastel aastakümnetel läbi viidud neuroloogiauuringutes. Saez jt [1] on oma viimases uuringus leidnud tõendeid selle kohta, et OFC kõrge sagedusega aktiivsus (HFA) 70-200 Hz vahel on otseselt seotud käitumisreaktsioonidega otsuste tegemisel. Näiteks näitasid Saez jt, et mõned HFA modulatsioonid korreleeruvad inimese valikuga ja tagajärgedega lihtsa kihlveo mängus. Saez jt viisid läbi analüüsi ühe muutujaga lineaarse regressiooniga, ennustades HFA väärtusi korraga ühest ülesandega seotud parameetrist, et leida elektroode, mis kodeerivad otsuste tegemisega seotud informatsiooni. Antud magistritöö keskendus Saez jt tulemuste ja analüüsi laiendamisele, rakendades mitmemõõtmelisi meetodeid, et avastada keerulisi signaale ja olulisi mustreid neuroloogilistes andmetes. Selleks kasutati 600 erineval andmekogumil kanoonilist korrelatsioonianalüüsi ja klasterdamist, et leida mustreid elektroodide aktiivsusmõõdetes ja käitumuslike reaktsioonide keerukaid kombinatsioone kodeerituna inimaju signaalides. Lisaks kasutati masinõppemeetodeid, et analüüsida patsientide käitumissuundumusi riskivõtmise suhtes hasartmänguülesandes ja ennustada närviandmetest ülesandega seotud sündmusi nagu võitmine, kaotamine ja riskivõtmine. Enamiku meetoditega saavutati mõõdukad kuni head tulemused, kuid põhjalikum analüüs on siiski vajalik, et saada täielik arusaam sellest, kuidas orbitofrontaalse ajukoore aktiivsus põhjustab inimkäitumist otsuste tegemisel.

Võtmesõnad:

Otsuste tegemine, orbitofrontaalkorteks, masinõpe, kanooniline korrelatsioonianalüüs, klasterdamine

CERCS: P170, P176, B110

Table of Contents

1 Introduction

Predicting the best outcome of a situation and making an appropriate decision is vital to the survival of animals and humans. During a decision making process the risks and rewards of potential outcomes need to be evaluated to select an action [1-4]. Although some theories have been developed about how the brain does this [5], the detailed process still remains elusive [1]. However, findings about the neural basis of these computations have been steadily growing over the past decade. Most prevalently, the **human orbitofrontal cortex** (OFC) has been shown to encode a multitude of values in its neural activity that are connected to decision making, including **reward value**, **risk**, **probability of success** and **regret** [1, 2, 4, 5, 6].

One of the latest studies on this topic is the 2018 article by I. Saez et al. titled "Encoding of Multiple Reward-Related Computations in Transient and Sustained High-Frequency Activity in Human OFC" [1]. Researches from the University of California, Berkeley and Stanford tackled the problem of decision making in the OFC. The authors had 10 adult patients play a simple gambling task for 200 trials while recording their brain activity directly from the OFC using **electrocorticography** (ECoG), a technique which allows to measure electrical potentials of brain activity directly from the cerebral cortex [7]. After collecting and analyzing the ECoG and behavioral data, they found two types of valuation signals reflected in the **high-frequency activity** (HFA) (70-200 Hz) of the OFC [1]. Firstly, fast signals, which contained information on the **current trial's choice and outcome processing**, e.g. anticipated risk, and secondly, **prolonged signals**, which encoded the information about the **previous trial** [1]. In their research, they used linear regression to predict an electrode's average HFA value from individual variables (e.g. win probability) [1]. Their results give further assurance about the importance of OFC in decision making computations [1].

Using the same data collected by Saez et al. for their 2018 research on the OFC and decision making [6], this Thesis expands on their research and **presents further analysis on OFC and decision making**. The main goal was to use multivariate methods to find **if and how groups of electrodes** (i.e. channels of HFA) and **regressors** (task-related variables, like win probability) **are correlated**. The research questions of this Thesis are the following:

Q1. Is there correlation between groups of electrodes and behavioral regressors?

Q2. What are the behavioral tendencies of each subject in regards to risk-taking?

Q3. Can we predict winning, losing or gambling from HFA in the OFC?

Q4. Are there groups of electrodes that behave similarly for some sets of trials?

Addressing Q1 adds to the research of [1] by looking at the combined effects of regressors on electrode signals, instead of a one-on-one correlation between an electrode and a variable.

In regards to Q2, this Thesis takes a simple approach by evaluating whether a patient was more motivated to take risks or play safe, using machine learning models on two specific choice-related variables.

To answer Q3, the same models are used to predict if a subject gambled, won the gamble or lost from HFA. Many datasets, using different types of parameters, were created to test, which kind of signal best reflects the three task-related regressors.

Finally, this Thesis uses clustering techniques to analyze patterns in the HFA data and answer Q4. Brain cells can react to a situation in a manner that depends on the conditions, for example a reaction of some neurons to losing a trial is different than that of others. Using **biclustering**, we can find groups of electrodes that "behave" similarly for specific groups of trials.

This Thesis uses multivariate methods to address and answer the four research questions. Specifically, the analysis consists of the following parts:

- **canonical correlation analysis**: Q1;
- systematic comparison of machine learning methods: Q2 and Q3;
- biclustering: Q4.

More details and explanation in regards to the methodology and the datasets are given in Section 3.

Figuring out the exact mechanisms of decision-making in the human brain could help tackle many issues related to **neuroeconomics**, which studies the neurobiological and computational principles of decision making [4]. It aims to understand what and how the brain computes to make value-based decisions [4]. Psychiatry is an important area, where knowledge from neuroeconomics could be useful, as we could gain a better understanding of many psychiatric disorders which involve failures in decision making processes, for example obsessive-compulsive disorder. This could result in improved diagnosis and treatment of these disorders [4].

The structure of this Thesis is the following. The **Background** section will provide a brief overview of the previous work and findings on OFC and its significance in decision making. In **Methodology**, the datasets and methods used in this Thesis are explained in detail and the **Results** section presents the results of the analysis. The **Discussion** section discusses the results in the light of the findings in [1], and the limitations and future work regarding this Thesis. Finally, the Thesis is summarized in **Conclusion**.

2 Background

This section reviews the basic concepts of OFC, intracranial recording (ECoG) and HFA, and gives an overview of findings in different studies concerning decision making in the brain.

2.1 Basics of HFA, OFC and ECoG

The communication between millions of neurons in the brain is what enables humans to think, feel and take action [9]. When masses of neurons communicate, they produce synchronized electrical pulses, or **brainwaves** [9]. These oscillations are divided into **frequency bands** that have a functional significance [10]. The fastest non-pathological (non-seizure) oscillations occur in the **high frequency** (**HFA**) **band** 70-200Hz [1, 10].

Figure 1 shows the location of OFC in the human brain.

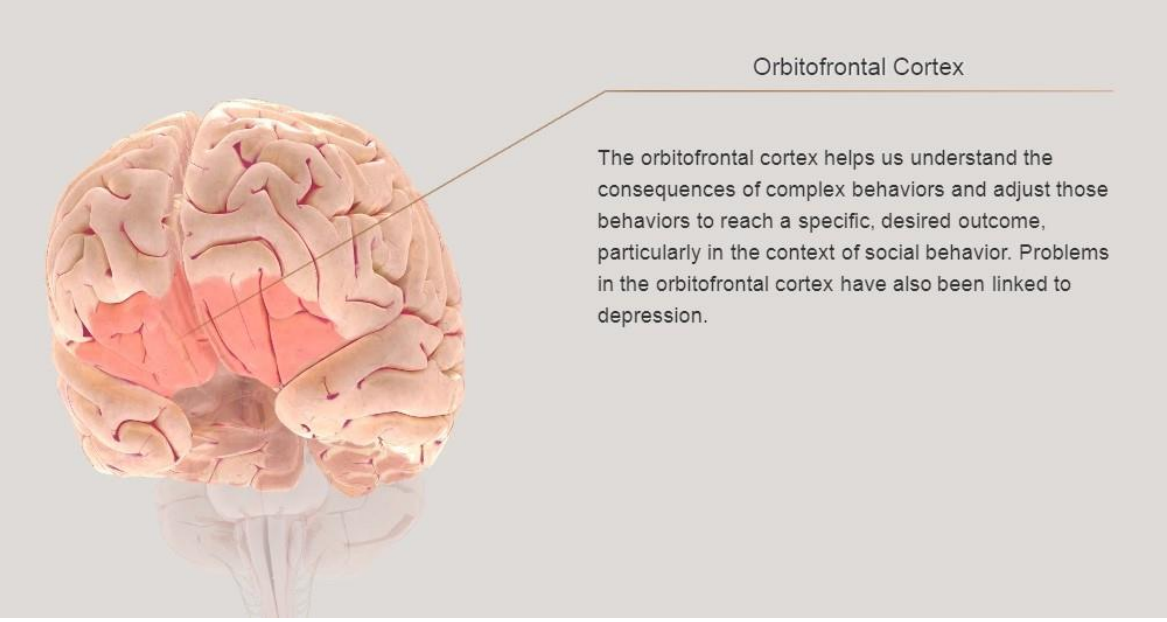

Figure 1. *The Orbitofrontal Cortex [11].*

As part of the **prefrontal cortex** (PFC), OFC is a region of the brain that lies on top of our eye orbits behind the forehead [5]. It has a wide range of connections to every type of sensory area, as well as to areas related to memory, learning and attention [12].

Intracranial EEG, or **ECoG** is a very efficient method for measuring brain activity in the OFC [1, 2]. It is a very precise method in terms of both temporal and spatial resolution, circumventing the limitations of other recording techniques like fMRI and scalp EEG [2].

2.2 The Decision Making Problem

Making decisions often requires evaluating "prospects" – **rewards** and **costs** that occur with some probability [4]. Decision making based on incomplete information is a difficult task that involves encoding computations of many different types of **reward-related values** into a signal across prefrontal cortical areas [1]. Here "reward" does not necessarily mean a monetary prize, but some positive value that a person or an animal has assigned to an action or object [3]. Studies using neuroimaging have shown that decision making involving uncertainty causes several brain areas to activate, including OFC and several other parts of the PFC, with a large part of research concentrating on OFC [5, 6].

One of the biggest questions is how exactly does the brain implement computations concerning risky decisions [4]? As stated before, some areas have been identified that are involved in this process, but there is no consensus on the nature of these computations [1]. One view on this is that the brain assigns a value for every potential outcome and then weighs the possibilities with a probability function [4]. The second view – that is considered in this Thesis and which has been implicated in many human fMRI studies – is that the brain computes some **parameters** for potential outcomes or actions (e.g. statistical moments like win probability or expected value) and then aggregates them into a value signal [1, 4, 5]. For example, neurophysiology studies have revealed that to get the expected value for some action, neurons in multiple areas, including MPFC, parietal cortex and dopamine neurons, combine information about the reward amount and the probability of acquiring it [5]. This view is in itself very natural, as behavior is often driven by multiple aspects of a reward, for example by how likely we are to acquire a reward (a pragmatic value), versus how much we want it (a hedonic value) [5].

2.3 Recent Results

In [1], the authors used several task-related variables or **regressors** to "decode" the brain signal. These included 4 variables related to making the decision – **probability of winning**, the **expected reward value**, the considered **risk** and a binary value reflecting the **decision** itself, whether or not the subject gambled [1]. The other 4 regressors concerned the processing of the game's outcome: **whether the subject won**, whether they **lost** (both of these values are 0 if the subject did not gamble), the **regret** and **reward prediction error** [1]. The importance of these regressors in OFC signals during decision making has been implied in

many studies. For instance, neuroimaging has shown that human OFC is activated by abstract rewards and punishments, like winning or losing money [5]. Several human fMRI studies have found that activations in the OFC reflect expected value and risk signals [1, 2, 4, 6]. In addition, reward probability is known to be coded in OFC, as proven by many human fMRI studies [1].

The research by Saez et al. [1] further confirmed these results, finding robust evidence for these regressors within the HFA frequency band. The signals that reflected these regressors were also **time-locked to particular time windows**: the choice-related regressors were most prevalent in the time between game presentation and the moment of choice, while outcome-related variables encoded strongest in the HFA signal that followed revealing the outcome of the game. They also found evidence of **past information** encoding: regressors from the immediately previous trial were represented in the current trial's signal, with some electrodes encoding both current and past trial information.

There is growing evidence that HFA is an **important reflector of brain activity in the cortex**, while broadband activity combines interactions from broadly distributed networks [1, 8]. Furthermore, in [1] the authors also reported weaker evidence of the regressors being encoded in broadband power.

3 Methodology

The code for the data preprocessing and analysis in this Thesis was written in Python. The implementations for the machine learning methods, canonical correlation analysis (CCA) and clustering were all taken from the **scikit-learn library**. The library was chosen for several reasons:

- the author is most familiar with this data analysis library and has used it many times during their studies,
- it is a well-known and thoroughly documented library,
- it is built on other essential and useful tools such as NumPy, SciPy and Matplotlib,
- it is easy to use.

All of the code written for this Thesis is available in the following GitHub repository: https://github.com/mariliisvelner/ofc-analysis.

The next subsections will describe the data and the analytic methods that are used in this Thesis.

3.1 Datasets

This Thesis uses two dataset collections **D1** and **D2** that were created using the **electrophysiological** (EP; the brain activity, as measured by ECoG) and **behavioral** data (describes when and how the subjects made their decisions) collected by the authors of [1]. Because the behavioral data is too specific in its information, important decision making variables were extracted to create the **regressor dataset**. D1 is used in classification and clustering, while D2 is used in CCA. Additionally, the behavioral and regressor datasets were also used in each of these analyses.

The EP and behavioral dataset were downloaded from CRCNS.org [8], one of the largest repositories of neural data, while the regressor data was obtained by contacting the authors of the research.

The first four of the following subchapters will describe the subjects who played the task, the details of the gambling task and the two datasets provided by the authors of [1]. The final two subsections cover the dataset collections D1 and D2, created during preprocessing in this Thesis.

Subject Task

In [1], the authors collected data from 10 (4 female) adult subjects with intractable epilepsy, a seizure disorder where the patient's seizures cannot be completely controlled with the aid of medication [13]. Electrode grids were implanted directly on the cortex to detect the location of the epileptogenic focus [1, 8]. Therefore, the placement of the electrodes depended wholly on the medical needs of the patient [1, 8].

The combined placement of all the electrodes in the brain is shown on Figure 2.

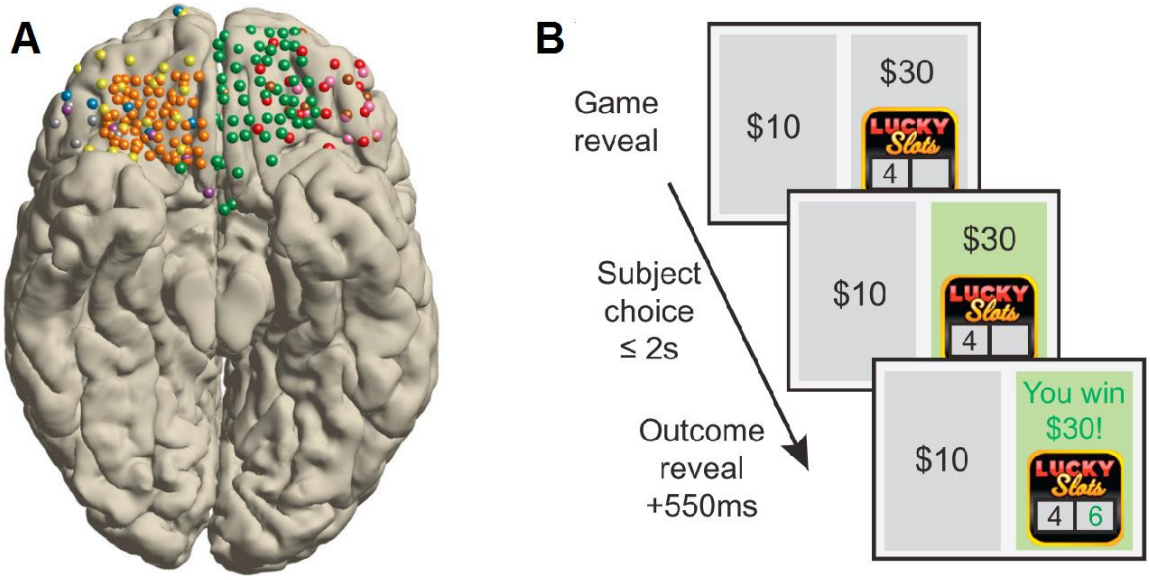

Figure 2. *Electrode location in the OFC and three game screens. Figure taken and modified from [1].*
(A) The placement of 192 electrodes of the 10 subjects combined in an anatomical reconstruction of the brain.
(B) Game screens. The image furthest in the background shows the screen during the game presentation. The screen in the middle is shown after the subject has made their choice, which is then highlighted in green (in this case, the patient chose to gamble). The screen at the front depicts the outcome reveal (i.e. the second random number and the game result).

The entire experiment consisted of 200 trials, during which the patients were off epileptic medications, fully alert and cooperative. At the beginning of each trial a **fixation cross** was shown (time t = 0), after which the game presentation screen appeared (t = 750ms). The game presentation screen displayed two numbers: a uniformly random integer (**presentation number**) from 0-10 and a prize amount (uniformly random, ranging from $10 to $30

with $5 increments). Then, the subjects had 2 seconds to choose between a sure, fixed prize (always $10) and a risky prize. If the subject did not manage to choose within the time limit, then the round was considered a timeout and the subject did not receive any reward. These trials were not used in the analysis. The outcome of the trial was shown regardless of the patient's choice: 550ms after the subject clicked on their choice, a second integer (**outcome number**; random, from 0-10, but the presentation number excluded) was shown. If the subject had chosen to gamble, they won if the second number was larger than the first [1, 8].

Table 1 shows the number of electrodes and trials included in the analysis for every subject. The number of trials varies due to different number of timeouts that occurred for each subject.

Table 1. *The number of electrodes and trials for each subject.*

Subject	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**
Electrodes	5	6	59	5	61	7	11	10	19	16
Trials	180	188	194	108	179	187	181	200	200	136

The entire experiment run of 200 trials was the same for every patient. Being fully aware of the (fair) task structure from the beginning, each patient also trained prior to the actual experiment by playing at least 10 rounds until they felt they understood the game completely. At this point they started the game, which lasted altogether about 12-15min [1, 8].

This gambling task has a very simplistic structure, where the risk and reward information are explicitly given, and working memory load and learning are minimized. This allows to analyze important decision making variables, like risk, expected reward and prediction error that have been used in previous studies [1, 8].

Electrophysiological Data

The ECoG data was sampled at a rate of 1KHz (i.e. the electrical potentials in the brain activity were measured 1000 times per second), then analyzed and visually inspected, after which electrodes with low signal-to-noise ratio (SNR) were found and removed. The data was filtered into **high frequency activity** (HFA; 70-200Hz) [1, 8].

In this Thesis, two types of EP datasets are used for every subject in preprocessing. They contain the HFA measurements in micro Volts (μV). Firstly, **the game events data**, which is an *nTrials* × *nTimePoints* × *nElectrodes* matrix that contains the HFA measurements of a patient. Specifically, *nTrials* and *nElectrodes* depend on the subject (refer to Table 1), while *nTimePoints* is always 3001 (i.e. each trial was 3 seconds long). The trials are aligned so that the data is centered [-1, 2] seconds around each game presentation event [1, 8]. That is, each trial in the data is organized so that the game presentation event of a trial is exactly **at the 1 second time mark** from the start of the trial. The remaining 2 seconds of the trial follow after the event.

The second EP dataset is the **button press data**, which has the same dimensions as the game events data matrix, but the data is centered around each button press event (i.e. the moment of choice) [1, 8].

Behavioral Data

Behavioral data, which contained exact **event timings** (in milliseconds) of each trial, was provided for every subject. There are 9 features per trial, including the start time of the trial, the game presentation time, the time when the subject pressed the button (made their choice), the outcome reveal time (when the second number was shown), the choice type (*Gamble*, *SafeBet* or *Timeout*) and the outcome of the trial. The latter could be *Win* (if the subject gambled and won), *Loss* (gambled and lost), *WouldHaveWon* (played safe, but would have won by gambling), or *WouldHaveLost* (played safe, would have lost by gambling) [1, 8].

Additionally, the authors provided a file which contained the metadata of the trials, including the risky prize amount and the presentation number [8].

Regressor Data

The regressor data contains the **regressor values for all trials and all subjects**. It is built on top of the behavioral data to analyze important **task-related features**. These variables were created because they describe the behavioral data on a higher level, providing more useful information than the behavioral dataset, which contains the exact timings of events in milliseconds. This data is too low-level and detailed to find evidence of in the noisy neural activity, while the regressors represent abstract and important aspects of the task and decisions.

The regressors can be divided into two categories: **choice-related variables**, which carry information that is important during deliberation, and **outcome evaluation related regressors**, which describe the result of the trial. The choice-related regressors are the following [1]:

1. the probability that the subject wins the gamble (**win probability**, p_{win}),
2. the **risk** (represents the variance; measures how far the current presentation number is from the mean (5)),
3. the **expected value** of the choice (EV; i.e. \$10 for all safe bet choices, $p_{win} \cdot prize$ otherwise),
4. **gamble indicator** (a binary value indicating whether the subject gambled or not),

The outcome evaluation-related regressors are [1]:

5. **win indicator** (a binary value indicating whether the subject won or not);
6. **loss indicator** (a binary value, 1 if the subject lost the trial, 0 otherwise);
7. **reward prediction error** (RPE; i.e. the obtained amount minus the expected amount (EV));
8. **regret** (the amount of additional money that the subject could have won, if they had chosen differently; i.e. the maximum reward subtracted from the received amount).

For example, if the presentation number is 4 and the risky prize amount \$20, then the probability of winning is 0.6 (as 4 itself is excluded when generating the outcome number and there are 6 possible outcome numbers out of 10 that would guarantee a win) and the risk is 0.24. Risk reaches its greatest value (0.25) when the probability of winning is the same as probability of losing the gamble (i.e. when the presentation number is 5), and its lowest value (0) when winning or losing is certain with gambling (i.e. the presentation number is 0 or 10). The expected value and gamble indicator depend on the subject's choice: if the subject decides to play safe, then the expected value is 10 and gamble indicator 0, otherwise their values are 12 (calculated as $0.6 \cdot 20$) and 1 respectively.

Continuing the example, let the outcome number be 7. If the subject had decided to play safe, the win and loss indicators would both be 0, the reward prediction error would be 0 (as the expected amount was \$10 and the person obtained \$10) and regret would be -10 (the obtained \$10 minus the possible \$20). On the other hand, if the subject gambled, then regret would be 0, the win indicator 1, loss indicator 0 and the reward prediction error 8 (the obtained \$20 minus the expected value 12).

There are a total of 16 regressors in used in this Thesis, 8 for the **current trial** and 8 for the **immediately previous trial**. The regressors from the immediately previous trial will be referred to as **'past' regressors** in the Thesis. That is, 'risk' would be the risk value from the current trial, while 'past risk' refers to the value of risk in the immediately previous trial.

Dataset Collection D1

This dataset collection was created in this Thesis using the EP and behavioral datasets. D1 was used in **classification** and **clustering**.

The collection contains many different datasets for each subject, created using **every combination** of the following parameter values:

1. **time window type**:
    a. the HFA during 1.5 seconds after the outcome reveal (type *outcome*),
    b. or the HFA between game presentation and button press event (type *choice*);
2. **time windows per electrode**: 1, 2 or 3 time windows per electrode;
3. **trial type**:
    a. all trials,
    b. win trials,
    c. loss trials,
    d. or gamble trials.

To create a dataset in this collection for a certain subject, we select **three parameter values**. For example, if we select time window type *outcome* and take two time windows per electrode and all trials, then for subject 2 the dataset is a 188 × 12 matrix (*nTrials* × (*nElectrodes* · *nTimeWindowsPerElectrode*)). In the matrix, each row corresponds to a **trial** and each column corresponds to a **time window average from an electrode**. There are 188 trials and 6 electrodes for subject 2 (refer to Table 1). If we want to have 2 time windows per electrode, we take the recordings of an electrode in the specified time window (1.5 seconds after the outcome reveal for type *outcome*), divide this time period equally into two parts and take the average of those parts. Thus, for subject 2 we get $6 \cdot 2 = 12$ features (time window average values). Similarly, if we wanted 3 time windows per electrode, then we would have $6 \cdot 3 = 18$ features for subject 2.

Taking into account the number of possible parameter values, there are $2 \cdot 3 \cdot 4 = 24$ datasets for every subject in this collection, i.e. 240 datasets in total.

We use these two particular types of time windows because there are two types of regressors: choice- and outcome-related. That is, choice-related regressors should be prevalent in the brain activity during the deliberation period (between game presentation and pressing the button), while outcome-related regressors presumably are encoded in the signal after revealing the result of the game. The 1.5 second duration was chosen as there is about 1.75 seconds from the outcome reveal until the next trial's game presentation, which gives enough time to discover the effects of the regressors in the signal, while not being influenced by the information from the new trial.

Using multiple time windows can be more informative than taking the average of the whole time period. As an example, we can look at the activity of electrode 17 of subject 9 on Figure 3.

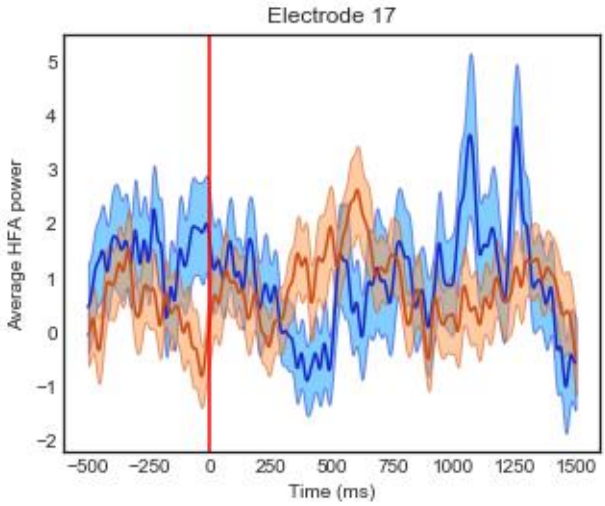

Figure 3*. The average HFA values of win trials (blue line) and all other trials (red line). The shading is SEM (standard error mean). The time is locked to the outcome reveal (represented by the red vertical line).*

We can see in the figure that the average activity fluctuates during the 1.5 seconds following the outcome reveal, with win trial average being lower during the 250-750ms time period, but spiking high around the 1s time mark. These smaller time windows seem to hold important information, as they show a significant difference in activity between different types of trials. If we were to take the average of the entire 1.5s after the outcome reveal, these changes would collapse and thus, we would lose information encoded by this electrode. Instead, we could generate more features by calculating the average HFA values of smaller

time windows, **dividing the 1.5s window into several equal windows**. Thus, to avoid information loss, 1, 2 and 3 time windows per electrode were used to create the datasets.

Dataset Collection D2

This dataset collection was created for the **canonical correlation analysis**, using the EP and behavioral datasets. CCA was initially tried with dataset D1, but as it produced mostly low correlations, new possibilities were explored that could help improve the results, but not invalidate the analysis with strong bias at the same time. The biggest problem was CCA overfitting on the training set, especially on datasets with a small number of samples or a large number of electrodes. To solve the first issue, no trial type other than 'all' was utilized in this analysis (as opposed to D1). Secondly, the number of electrodes was lessened systematically. As a result of this preprocessing, D2 was created.

Similarly with collection D1, many different datasets were created for all subjects with varying parameter values. The parameters with their optional values are:

1. **time window type**:
    a. the HFA of 1.5 seconds after the outcome reveal (type *outcome*),
    b. or the HFA between game presentation and button press event (type *choice*);
2. **electrodes**:
    a. all electrodes of the subject,
    b. best electrodes selected with **single regressor correlation**,
    c. best electrodes selected with **group regressor correlation**, requires choosing a percentage $p \in \{20\%, 40\%, 60\%, 80\%\}$;
3. **time windows per electrode**: 1, 2, or 3.

Therefore, **three parameter values** are necessary to create a dataset for a subject: time window type, electrode type and the number of time windows per electrode. Thus, there are $2 \cdot 6 \cdot 3 = 36$ different datasets per subject, 360 in total.

Aside from using all electrodes, there were two selection methods – **single and group regressor correlation** – which chose specific electrodes for each subject using CCA and are explained in more detail in Section 3.3.

As an example, if the time window type is *choice* and all electrodes were included, with 1 time window per electrode, then the dataset for subject 2 would be a 188 × 6 matrix (*nTrials* × *(nElectrodes* · *nTimeWindowsPerElectrode)*, where a row represents a trial and every

feature corresponds to an average HFA value between the game presentation and moment of choice measured by one electrode for that particular trial.

3.2 Classifiers

Two of the research questions are addressed with classification models: (Q2) what are the behavioral tendencies of each subject in regards to risk-taking, and (Q3) can we predict winning, losing or gambling, from HFA in the OFC? This Thesis explored multiple machine learning methods, and afterwards focused on two – **logistic regression** (LR) and **linear support vector classification** (LSVC) – to answer Q2 and Q3.

To answer Q2, this Thesis uses LR and LSVC on two features – the **risk prize** and the **presentation number** (i.e. the first random number) – to predict whether the subject gambled or not. Specifically, the aim is to find out how much the prize, as a more hedonic value, influences the subject's decisions, compared to the presentation number, which essentially represents the win probability – a more pragmatic aspect to the choice.

It is hard to find a good trade-off between the poorer odds of winning a large prize and a sure, albeit a smaller prize. Using these two variables to predict a person's decision should in essence combine the tempting and pragmatic angles of the subject's deliberation. Comparing the absolute values of their coefficients obtained by the model, we can analyze how important was either aspect to the patient's decision. We can additionally compare different subjects to see if some were significantly more risky in their behavior than others.

While this limits the behavior of a subject to two simple variables, it is the simplest way to analyze the rationale behind a person's behavior as these numbers are the only information given to them about the game.

To address Q3, datasets in the dataset collection D1, which include all trials of a patient (trial type is 'all'), are used to predict gamble, win and loss indicators. It is important to note that the negation of win does not automatically mean loss, as the subject did not win (or lose) in any of the safe bet trials either.

Several machine learning models were tested from scikit-learn using default parameters. Table 2 lists the specific models that were used and in which part of the classification analysis it was tested.

Table 2. *The methods used in the behavioral (Q2) and event prediction (Q3) analysis.*

Model	**Analysis**
Logistic regression	Q2, Q3
Linear support vector classification	Q2, Q3
Decision tree classifier	Q2, Q3
Random forest classifier	Q2, Q3
SGDClassifier (linear classifier with SGD training)	Q3
Support vector classification (non-linear)	Q3

In addition to LR and LSVC, **decision tree classifier**, **random forest classifier, the *SGDClassifier* and support vector classification (with non-linear kernel)** were also tested in behavioral (Q2) and event prediction (Q3) analysis, using the scikit-learn implementations (*DecisionTreeClassifier, RandomForestClassifier, SGDClassifier* and *SVC*) with their default parameters described in [14], [15], [16] and [17] respectively. The two linear models – **logistic regression** and **linear support vector classifier** – were kept because (1) they showed the best results with 5-fold cross-validation, and (2) they are easier to fine-tune and interpret in terms of feature importance, unlike for example random forest and decision trees, where the result (and thus, feature importance) depends on many parameters like measure of impurity, maximum depth of the tree, number of estimators, minimum number of samples per leaf node, etc. LR and LSVC are fast classifiers that produce coefficients for the features, which can be interpreted as feature importance.

The following subchapters describe the two selected models in more detail along with the research questions they addressed and the data processing that was necessary to apply these algorithms.

Logistic Regression

Logistic regression is a statistical method that uses the weighted sum of a set of variables to classify a sample into a binary class [18].

The regression coefficient of a variable determines how much a unit-size change in the given variable affects the probability of the positive class [19, 20]. This coefficient can then be used to analyze how much influence a feature has on the sample's class, or, in this case, the person's decision to gamble or not.

The formula used to transform a data sample $(x_1, \dots, x_n)$ to a probability of the positive class is the following:

$$y = \frac{e^z}{1 + e^z}$$

where $z = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$, the coefficients are $b_1, \dots, b_n$, and b_0 is the bias term. If $y \geq 0.5$, the sample is classified into the positive class.

What one has to consider when analyzing a variable's impact on the label with logistic regression coefficients is the scale and units of the variables. Namely, if two variables are in a vastly different scale and do not measure the same value, then the coefficients are not directly comparable [21]. For example, we cannot directly compare coefficients of temperatures in Celsius with coefficients of rainfall in centimeters. Neither can we compare random integers from 0 to 10 with prizes in dollars ranging from 15 to 30. Therefore, the data needs to be standardized [21]. In this Thesis, **mean normalization** is used. It involves subtracting the mean of a feature and dividing the result by the standard deviation. This way, the data is centered around zero and each value deviates from zero by some number of standard deviations, which allows to compare the effect of the variables using their coefficients [19, 21].

The *LogisticRegression* model of scikit-learn is used as the classifier. The only parameter values that were changed during the analysis were the *penalty*, inverse regularization strength *C*, *dual* and *fit_intercept*. The latter was set to False so that the model would be forced to fit only the coefficients of the features, without using the intercept. This makes interpreting the coefficient, feature importance easier as the result only depends on the two variables. The parameter *dual* was also set to False, as it is the recommended value if the amount of samples exceeds the number of features, which is the case in this Thesis [22].

It was decided to test different values of the penalty and *C* because regularization helps the model to generalize and reduce overfitting and thus is an important contributor to the model's performance [23]. The two tested penalization methods were L1 and L2 norm.

L1 norm takes the absolute value of all the feature weights and sums them up to a term, which is then added to the cost function [24]. Adding the L1 norm to the loss function is

otherwise known as **lasso regularization**. The loss function that the *LogisticRegression* algorithm minimizes when the regularization method is lasso, is the following [25]:

$$\min_{w,c} \ ||w||_1 + C \sum_{i=1}^{n} \log(\exp\left(-y_i(X_i^T w + b)\right) + 1)$$

where y_i is the actual class of the sample $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$, $w = (w_1, \dots, w_n)$ are the feature coefficients, b is the intercept and C the regularization parameter [25]. $||w||_1$ is the L1 norm and is calculated as $||w||_1 = |w_1| + \dots + |w_n|$, while the **logistic loss** is multiplied with C [24, 25].

Ridge regularization involves adding the **L2 norm** to the loss function. L2 norm is the sum of weight squares: $w^T w$. The loss function with ridge regularization is therefore the following [25]:

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp\left(-y_i(X_i^T w + b)\right) + 1)$$

with the same parameters as the previous formula.

Whether it is better to use L1 or L2 norm in regularization depends on the nature of the task and data. While L1 is computationally more expensive, as it cannot be solved with matrix calculations like L2, it is more robust than L2, meaning that it is more resistant to outliers in the data (cost of outliers increases linearly with absolute value, exponentially with squaring) [24].

The other parameters were kept at their default values by following the recommendations in the model's documentation and are further described in [22].

Linear Support Vector Classification

Linear support vector classification (LSVC) is essentially a support vector machine (SVM) that classifies samples using a linear kernel [26]. That is, it tries to find a hyperplane that best separates the classes by maximizing the hyperplane's distance from the two closest points from opposing classes [27].

After training, the LSVC model produces coefficients for the features. New samples are predicted into a class by calculating the dot product of these weights and the sample's feature values. If the product is larger than 0, then the sample is classified as positive, otherwise negative. This means that as with logistic regression, the class of the data sample depends

on the absolute values of the coefficients which determine the amount of influence each feature has on the dot product. We use the LSVC in addition to logistic regression to see if they achieve similar results, if the feature importance is similar with the two linear methods.

To be able to compare the coefficients of the prize and the presentation number, **mean standardization** is applied to the data.

LinearSVC was chosen from the support vector classification models that the scikit-learn library offers. As with the *LogisticRegression* model, the *fit_intercept* and *dual* parameters were set to False, and different values parameter *C* were tested, along with L1 and L2 norm for the penalty.

While LR optimizes logistic loss, LSVC uses average **squared hinge loss** over all samples, which is calculated as follows [28]:

$$L_{squared\_hinge} = \frac{1}{n}\sum_{i=1}^{n}(\max\{1 - y_i\hat{y}_i, 0\})^2$$

where y_i is the actual class of the sample i and $\hat{y}_i$ is the sample's predicted score (not the label). The regularization term (L1 or L2 norm) is added to the loss.

All other parameters maintained their default values to keep the model as simple as possible and their detailed descriptions can be found in [26].

Figure 4 shows a comparison of the squared hinge loss and logistic loss on a machine learning model.

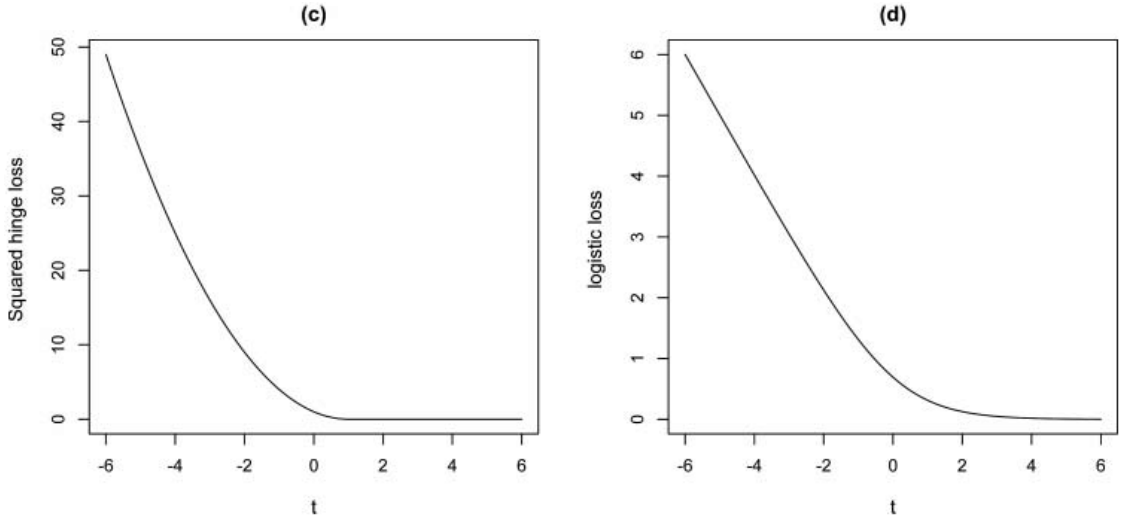

***Figure 4**. The curve of squared hinge loss (c) and logistic loss (d). Figure adapted from* [29].

We can see that the losses are different in their scale but very similar in the curve. Logistic loss, however, never reaches zero, which might affect the accuracy.

3.3 Canonical Correlation Analysis

Canonical correlation analysis (CCA) is a multivariate method of analysis which tries to maximize the **correlation** between **weighted sums of variables**. Namely, suppose we have two datasets X and Y with $\boldsymbol{x_i} \in X$, $\boldsymbol{y_i} \in Y$ $(i = 1 \dots t)$, where $\boldsymbol{x_i} = (x_{i1}, x_{i2}, \dots, x_{in})$, $\boldsymbol{y_i} = (y_{i1}, y_{i2}, \dots, y_{im})$, and $|X| = |Y| = t$. CCA finds vectors $\boldsymbol{a} = (a_1, a_2, \dots, a_n)$ and $\boldsymbol{b} = (b_1, b_2, \dots, b_m)$ such that the correlation between vectors $\boldsymbol{a}X^T$ and $\boldsymbol{b}Y^T$ is maximal [30].

CCA is used in this Thesis to answer research question Q1: **is there correlation between groups of electrodes and regressors**? We conduct the analysis for every subject separately, where X is a dataset of HFA values (from dataset collection D2) and Y is a dataset of regressor values, and samples $\boldsymbol{x_i}$ and $\boldsymbol{y_i}$ correspond to the same trial. An entry x_{ij} in dataset X represents an average HFA value from a time window recorded by a particular electrode during trial i. Therefore, the number of features in vector $\boldsymbol{x_i}$ is $n = n_{electrodes} \cdot n_{time\_windows\_per\_electrode}$. For example, if we used one time window per electrode for subject 1, we would have $n = 5$ features in X (refer to Table 1). The number of features in Y is the number of regressors chosen for the analysis, as the regressors have one particular value for every trial.

Using CCA, we can look at weighted sums of electrode signals and regressor values to find if there is a strong association between the two constructs. The authors in [1] found correlations between individual electrodes and regressors and with CCA we can find if there is evidence of a more complex signal encoded in OFC.

To assess if there is a combined signal in OFC, every dataset in collection D2 was tested with every possible combination of current and past trial regressors in their corresponding time windows. That is, if a dataset with time window type *outcome* was picked from the collection, then Y only contained outcome-related regressors, either from past or current trial. On the other hand, if the dataset contained HFA values from time windows of type *choice*, then the regressors in Y were choice-related.

It is important to note that past regressors are used with the current trials and thus, when performing CCA with past regressors, the first trial is excluded from the dataset, as it has no previous trial.

The collection D2 contained datasets that did not use every electrode of the patient. Two selection methods were used to select specific, more informative electrodes for each subject. The following two subsections describe these methods.

Single Regressor Correlation

To select the best electrodes for a subject, given the regressor data Y, this method chooses **one electrode for each regressor in $\boldsymbol{Y}$**. Specifically, it chooses the electrode with which the regressor has the largest correlation with over all of the trials. The correlation was calculated as the **Pearson correlation coefficient** which measures the strength and direction of a linear relationship in the data [31].

This correlation is calculated between two vectors of size *nTrials*, which depends on the subject (see Table 1). One vector contains one regressor's values for all of the subject's trials. The other vector depends on the type of the regressor. If it is an outcome-related variable, then the vector contains the average HFA values of *outcome*-type time windows (1.5s after outcome reveal) for every trial. If the regressor is choice-related, then the HFA average is taken between the game presentation and button press events (time window type *choice*).

Group Regressor Correlation

This method chooses electrodes based on the **whole set of regressors in dataset $\boldsymbol{Y}$**. Given a set of regressors and a percentage $p \in \{20\%, 40\%, 60\%, 80\%\}$, the **$\boldsymbol{p}$ percent of electrodes** that have the largest correlation with the regressors over all trials are chosen.

To calculate this correlation between a set of regressors and an electrode, CCA is used between the vector X (of length *nTrials*), which contains the electrode's HFA values, and dataset Y (dimensions *nTrials* × *nRegressors*). Having calculated the correlations for all electrodes of a subject, the group regressor correlation method then chooses the p percent of the best electrodes based on their correlation with the regressors.

3.4 Clustering and Visualization Methods

Clustering is another tool to find patterns in the data. It assigns data samples to groups (i.e. clusters) based on how similar they are, compared to other samples. The following subsections describe the various clustering methods that were used in this Thesis.

Biclustering

Biclustering is a non-supervised method for finding patterns in the data. Its goal is to **partition the rows** (observations) **and columns** (features) in the data so that the resulting matrix reveals patterns [32]. Figure 5 shows biclustering on a synthetic dataset.

Using biclustering algorithms on the **datasets in collection D1**, we can find if there are any **groups of electrodes** (features, columns) **that behave similarly for some group of trials** (rows). Therefore, we can use this method to answer research question Q4.

There are several biclustering algorithms, which differ in the way they define biclusters and assign rows and columns to them [33]. We apply the two biclustering algorithms from scikit-learn on the datasets, namely *SpectralCoclustering* and *SpectralBiclustering.*

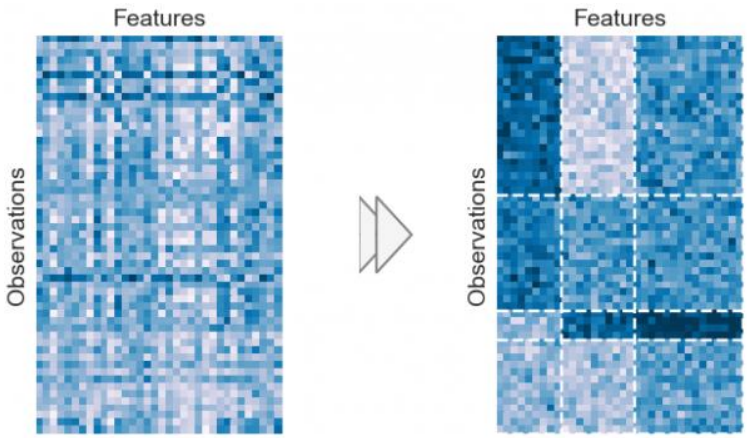

Figure 5. *Illustration of biclustering on synthetic data* [32]. *The matrix on the left shows the non-transformed, original dataset, while the matrix on the right shows the dataset after biclustering. The values in the dataset are colored such that the low values are lighter and high values darker.*

SpectralCoclustering, which implements the spectral co-clustering algorithm, partitions rows and columns such that **each row and column belongs to only one bicluster** [33]. The algorithm finds biclusters that have values higher than the other blocks in the same row and column [33]. Figure 6 shows a generated synthetic dataset before and after applying the spectral co-clustering algorithm.

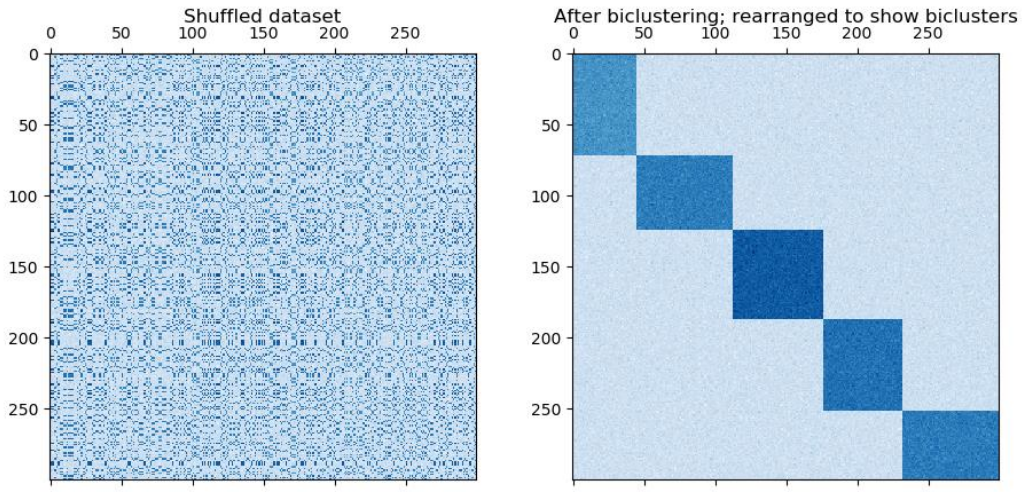

Figure 6*. Applying SpectralCoclustering to a synthetic dataset* [34]*. The dataset after random shuffling is shown on the left matrix, while the right matrix shows the arranged dataset after using spectral co-clustering.*

The *SpectralBiclustering* algorithm partitions the data into a **checkerboard structure** [33]. For example, if the rows are divided into four parts and the columns into three parts, then after applying *SpectralBiclustering*, each column belongs to four biclusters and each row belongs to three biclusters. Figure 7 shows the result of biclustering on a synthetic dataset which has an internal checkerboard structure.

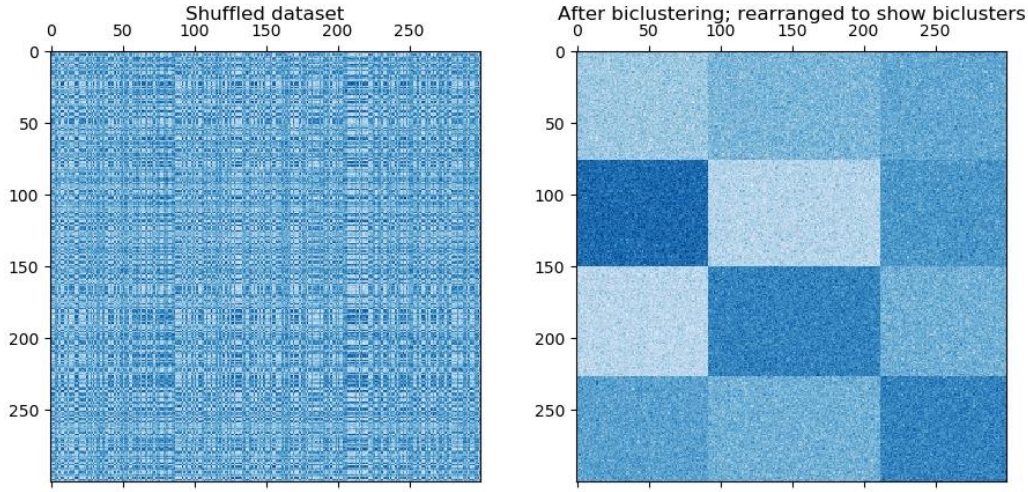

Figure 7. *Applying SpectralBiclustering to a synthetic dataset* [35]*. On the left is the dataset in a shuffled form, on the right is the result of biclustering. The generated synthetic dataset is structured internally as a checkerboard, which is why the algorithm works well.*

The two algorithms were tested with varying *n_clusters* and *svd_method* values. The first parameter specifies the number of biclusters to find, while the other is a method for **singular value decomposition** (SVD), an internal algorithm that factorizes a matrix into three submatrices [36]. Specifically, SVD provides the row and column partitions that divide the matrix into biclusters [33]. This Thesis tested both of the available SVD methods in scikit-learn, which are *randomized* and *arpack*. While the former is generally faster and works better on large datasets, *arpack* is more accurate [36, 37].

Biclustering is a NP-hard problem for which no universal method exists [38]. Many algorithms have been developed, each using its own internal method for optimizing the biclusters [39].

After applying the algorithms to the data, it is necessary to evaluate the quality the biclustering results. There are two ways to measure this: internally and externally. External measuring presumes that there exists some external ground truth that we can compare the biclustering result to [33]. In this Thesis, as with most real data, this true solution is unknown. Therefore, we must use an internal measure – something which uses the data and the result only. As scikit-learn does not have a function for any internal bicluster measure, this had to be implemented independently.

The **Average Spearman's Rho** (ASR) evaluation measure was chosen as the one to score biclusters with. It is based on the **Spearman's rank correlation**, which measures the direction and strength of a monotonic relationship in the given data [38]. Figure 8 depicts examples of monotonic and non-monotonic relationships.

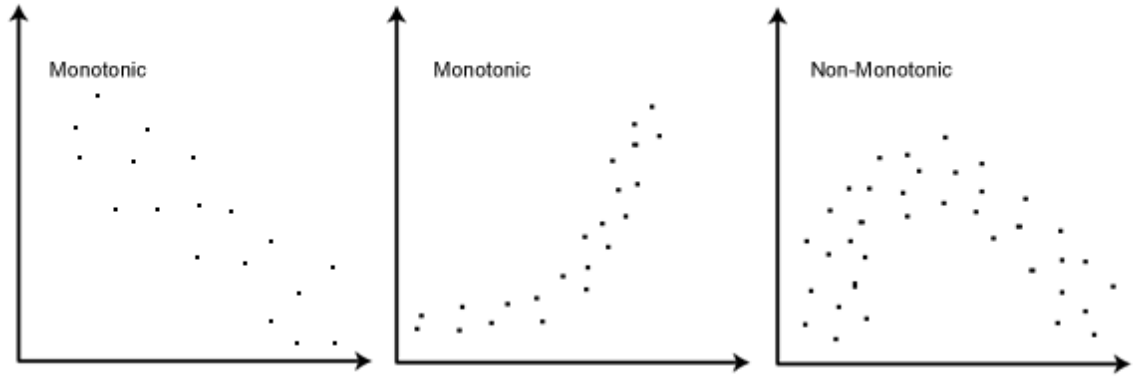

Figure 8. *Examples of monotonic and non-monotonic relationships [31]. The left plot shows a linear and a monotonic relationship, the middle one depicts a non-linear (exponential)*

monotonic relationship, while the plot on the right shows a non-monotonic (non-linear) relationship. Spearman's rank correlation would be close to -1 in the left plot (one variable increases, as the other decreases), close to 1 on the middle (a variable increases as the other does) and 0 for the last.

ASR is calculated for one bicluster X as $ASR(X) = 2 \cdot \max\{r_{rows}, r_{cols}\}$, where

$$r_{rows} = \frac{\sum_{i=1}^{n} \sum_{j=i+1}^{n} spearman(row_i, row_j)}{n(n-1)}$$

and

$$r_{cols} = \frac{\sum_{k=1}^{m} \sum_{l=k+1}^{m} spearman(col_k, col_l)}{m(m-1)}$$

where n and m are the number of rows and columns in bicluster X respectively and $spearman(a, b)$ calculates the Spearman's rank correlation between two vectors a and b [38]. In other words, the correlations between rows and columns in the bicluster are aggregated and averaged separately and the final ASR value is the double of the maximum of these values.

In this Thesis, to evaluate a biclustering result, the **ASR values of every bicluster in the dataset were averaged**. ASR is measured from -1 to 1, where ASR values close to 1 indicate a strong correlation between the trials/electrodes in the bicluster, while a score near -1 implies a weak correlation [38]. The measure was chosen for this Thesis as it has been shown to perform well in many studies [38, 39].

t-SNE

t-distributed Stochastic Neighbor Embedding, or **t-SNE,** is a technique for **visualizing high-dimensional datasets** [40]. The algorithm converts data in a high-dimensional space to a lower dimensional space while keeping the similar data samples clustered together and samples, which differ greatly, far from each other [40]. It is a probabilistic algorithm which works simultaneously in the high- and low-dimensional spaces, trying to minimize the distance between two probability distributions [40]. One distribution measures the pairwise similarities of the samples in the original data space, while the other measures pairwise similarities in the lower dimensional data space to where the objects will be converted when the algorithm finishes [40].

It is a well-known algorithm that is applied in many areas where high-dimensional data is processed, for example bioinformatics (genomic data), image and speech processing [40]. In this Thesis, t-SNE is used to visualize the raw HFA datasets to find if similar types of trials (win, loss or gamble) group together.

The scikit-learn implementation of the t-SNE algorithm is used in this Thesis. All parameter values except for *perplexity* were kept to their defaults which are further described in [41]. The **perplexity** value is essentially an estimate for the number of close neighbors of each data point [42]. While the scikit-learn documentation states that recommended values lie between 5 and 50, it is a more complex problem, as the correct value depends on the concrete dataset [42]. Multiple *perplexity* values ranging from 2 to 100 were tested in this Thesis with every dataset.

The algorithm's documentation states that using another dimensionality reduction method before applying t-SNE on the dataset will reduce noise and the computation time and is thus highly recommended [41]. This Thesis uses the scikit-learn's *TruncatedSVD* method to reduce the size of all datasets, which have more than 35 features, to a 30-feature dataset. *TruncatedSVD* uses a variant of singular value decomposition in the dimensionality reduction and works well on sparse matrices [43]. Default parameters were used for the algorithm and are specified in [44].

4 Results

This section describes the results of the analysis conducted in this Thesis. Section 4.1 (**Exploring Neural Responses**) will present various figures that describe the data and compare them to visualizations in [1] to explore the data and validate that the datasets are processed correctly. Section 4.2 (**Behavioral Analysis**) tackles research questions **Q2** and **Q3**, describing the results for the behavioral analyses, in which the patient's risk behavior was evaluated and task-related events (e.g. winning or not) were predicted with linear methods. Section 4.3 (**Electrode-Regressor Group Correlation**) addresses research question **Q1**, which aimed to find correlation between groups of electrodes and regressors, and presents the results of CCA. Finally, Section 4.4 (**Visualizing Trial-Electrode Patterns**) will show the biclustering results and visualizations with t-SNE to evaluate, whether there are any patterns in the HFA data between trials and electrodes, answering **Q4**.

4.1 Exploring Neural Responses

This section will **explore the data** by showing electrodes that differ in their responses in time windows and compare visualizations to findings in [1]. The first subsection will concentrate on comparing the activity of **win trials and other trials**, and the second subsection will present the differences between **gamble and safe bet trials**.

Win Trials and Other Trials

To validate the dataset and preprocessing pipeline, four figures from [1] were replicated to test if the results in their analysis and this Thesis match. The original plots are shown on Figure 9. The authors plotted two heat maps of trial activity from one electrode (which was not specified), setting the color map range from -20 to 20μV (plot A). The x-axis represents the time points, with t=0 being the time of outcome reveal while the trials are on the y-axis. Plot B shows the averaged activity of win trials and all other trials, and plot C shows the **explained variance** (%EV). Explained variance is measured from 0 to 1 and it describes how much the variance of one variable affects the variance of another [45]. In other words, how well do the two variables covariate.

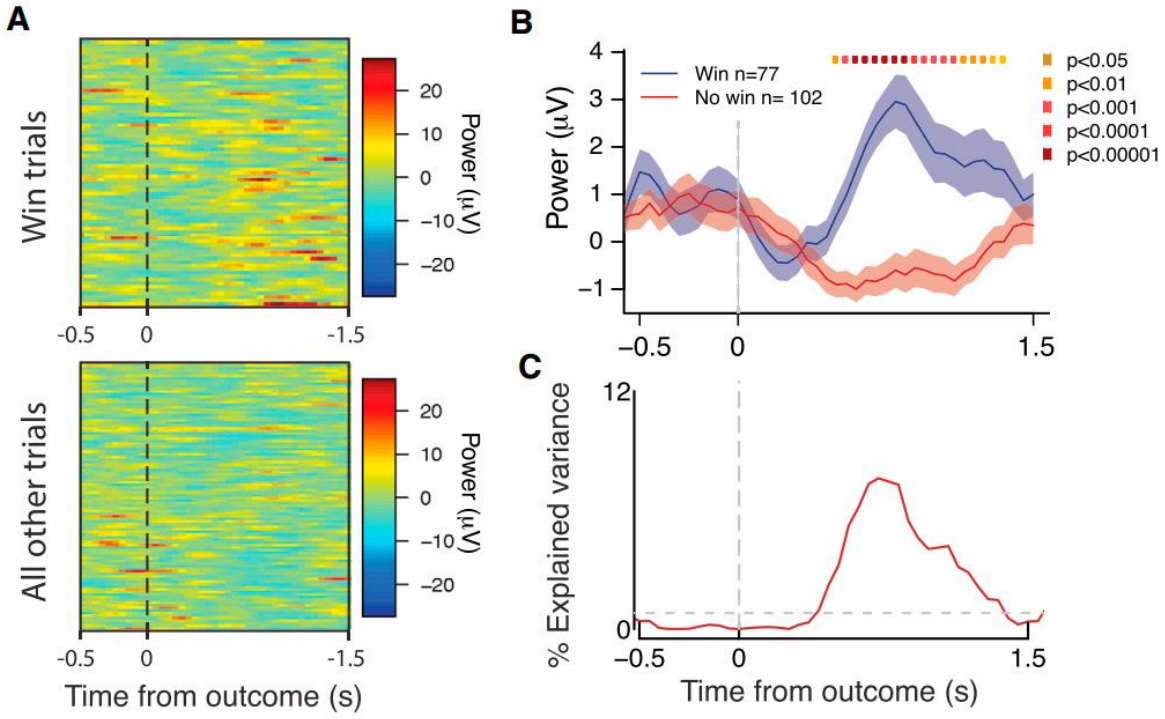

Figure 9. *Four plots from [1] depicting HFA across all trials for one electrode.*
(A) Heat map of recorded trials from one example electrode that encoded win information. The x-axis is locked to outcome reveal (t=0), the plot at the top depicts the HFA values of win trials, while the lower plot depicts all other trials.
(B) The HFA power averaged across win (blue line) and other trials (red) that are shown in (A), where the shading is SEM (standard error mean).
(C) The percentage of variance that the win indicator explains in the HFA signal in (A).

These plots show clearly that this electrode is highly activated during the win trials, after the outcome has been revealed. We can see from panel A that this electrode emits higher values more frequently during win trials, compared to others. This fact is also reflected in plot B, where the average activity across win trials is significantly higher than that of other trials during the 1.5s that follows the outcome reveal. Finally, the results in C also confirm this: the changes in HFA are most explained by the win indicator around the 1 second time mark after outcome reveal.

To validate this analysis with our pipeline, we had to analyze the 199 electrodes in the dataset. We found several electrodes which showed a similar large gap between the win and other trials, in particular the electrode 29 from subject 5. The plots for this electrode are depicted in Figure 10.

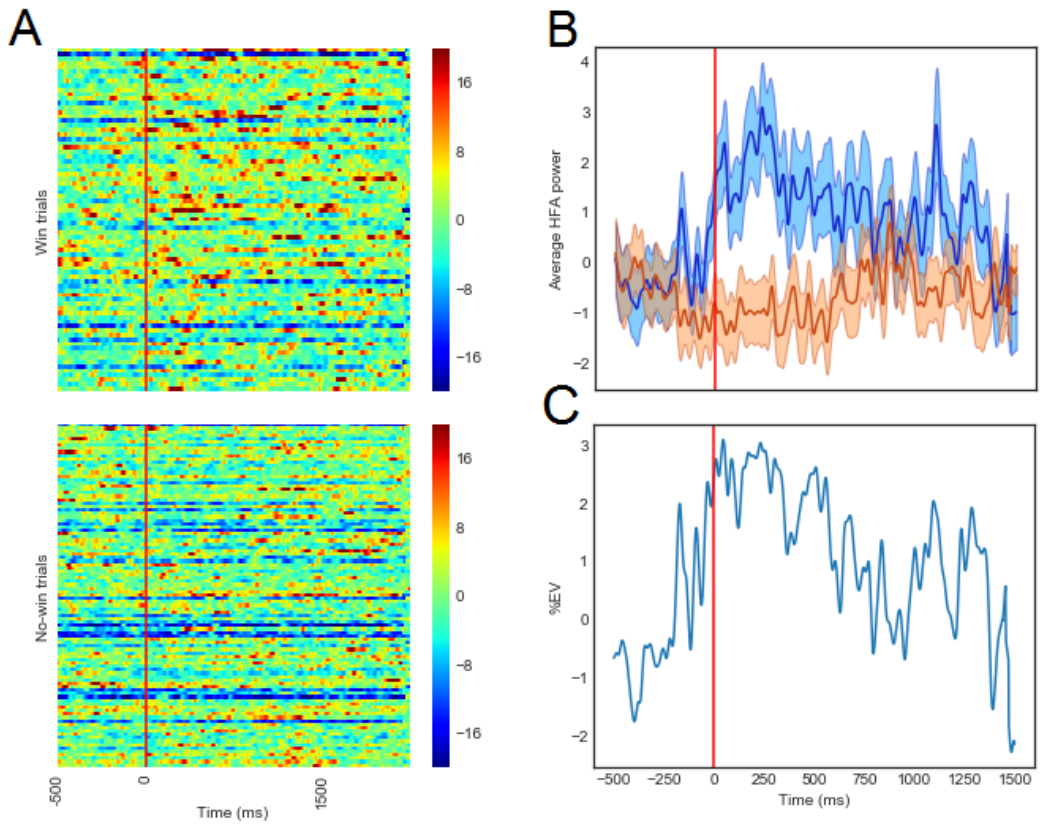

Figure 10. *The four plots visualized for another electrode encoding win information (subject 5, electrode 29). The trials in all plots are time-locked to the outcome reveal (t = 0).*
(A) Two heat maps, the top plot depicting win trials, the bottom one all other trials.
(B) The average HFA values for win (blue) and other (red) trials, with SEM (standard error mean) shading.
(C) The explained variance in the HFA signal.

This Thesis explores the **raw signals** from the electrodes. Comparing plots on Figure 10 to those in Figure 9, we can see the difference in the smoothness of the curves, with plots B and C in Figure 9 showing a much smoother signal than their counterparts in Figure 10. This difference probably comes from a preprocessing procedure which involves averaging the neural activity over small time windows for every data point. As this time window length was not specified in [1], and the raw signals still show clear tendencies in behavior, this Thesis will not additionally preprocess the data after averaging the signal or calculating explained variance.

We can see from Figure 10, plot B that this electrode shows a significant difference in HFA when comparing win trials to no-win trials. This is also reflected in plot C which shows a

similar tendency to its counterpart in Figure 9, with %EV increasing when the difference between the average HFA values of the two trial types increases. The win trials in plot A seem to also show more activity during the after-outcome time period than the other trials, just as in Figure 9.

This was an example of an electrode that encodes *win* information, where the activity for the win trials is higher than for the other trials. More examples of such electrodes in the dataset can be seen on Figure 11 which depicts electrodes from subjects 3, 4 and 5.

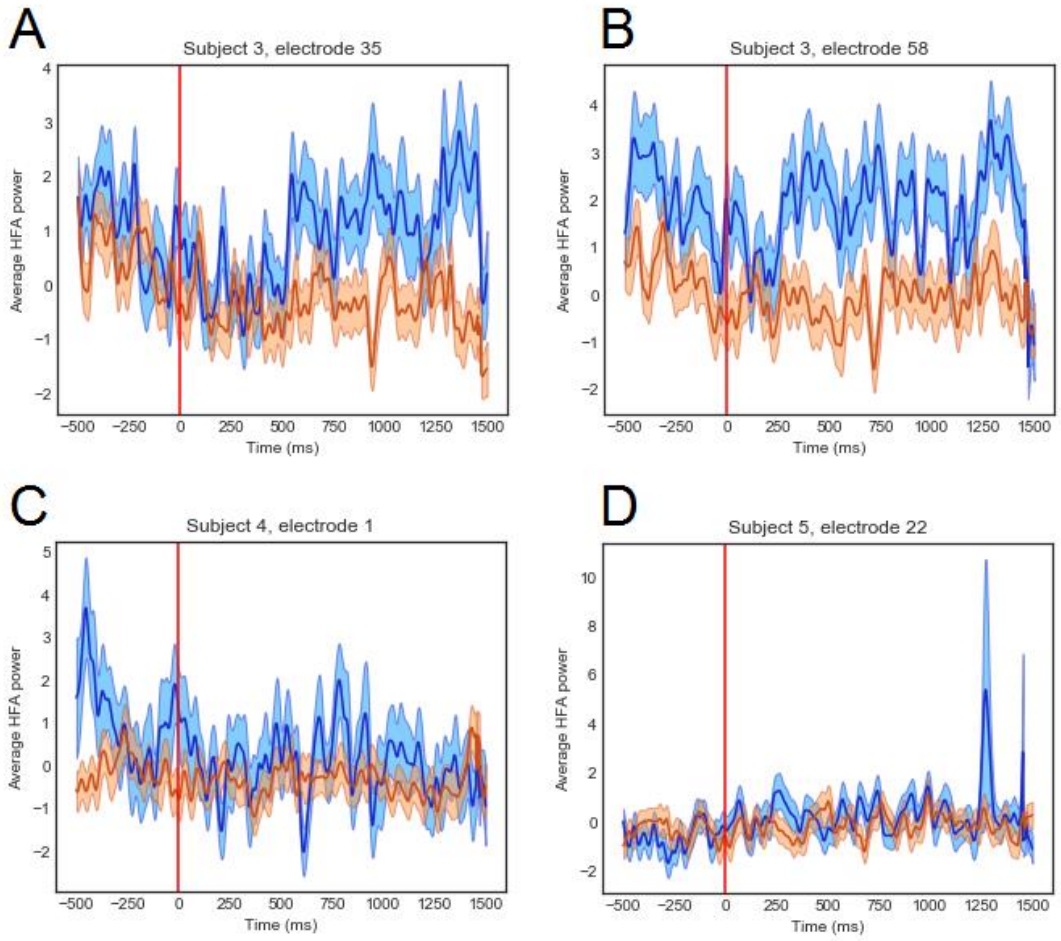

Figure 11. *Four win information encoding electrodes with average HFA values for win (blue) and other (red) trials, shading is SEM (standard error mean). All trials are time-locked to the outcome reveal (t = 0).*

We can see from Figure 11 that electrodes 35 and 58 from subject 3 (plots A and B) show considerable difference in the mean HFA value between win and other trials, while for electrode 1 from subject 4 (plot C) and electrode 22 from subject 5 (plot D) the mean values coalesce more. The differences between the types of trials for these two electrodes (in C and

D) appear in other ways: in plot C, we can see that for the win trials, the activity fluctuates much more than for other trials, and in plot D, the win trials show an unusual spike at the 1250 and 1500 time mark. These results imply that the ***win* information is encoded in different ways in different electrodes** (cortex regions) in the dataset.

There are also electrodes where the **no-win indicator** had higher neural activity than the win indicator (no-win indicator means the win indicator negated: 1 if the subject did not win, 0 otherwise). One example of this is electrode 18 from subject 9, the activity of which is depicted in Figure 12.

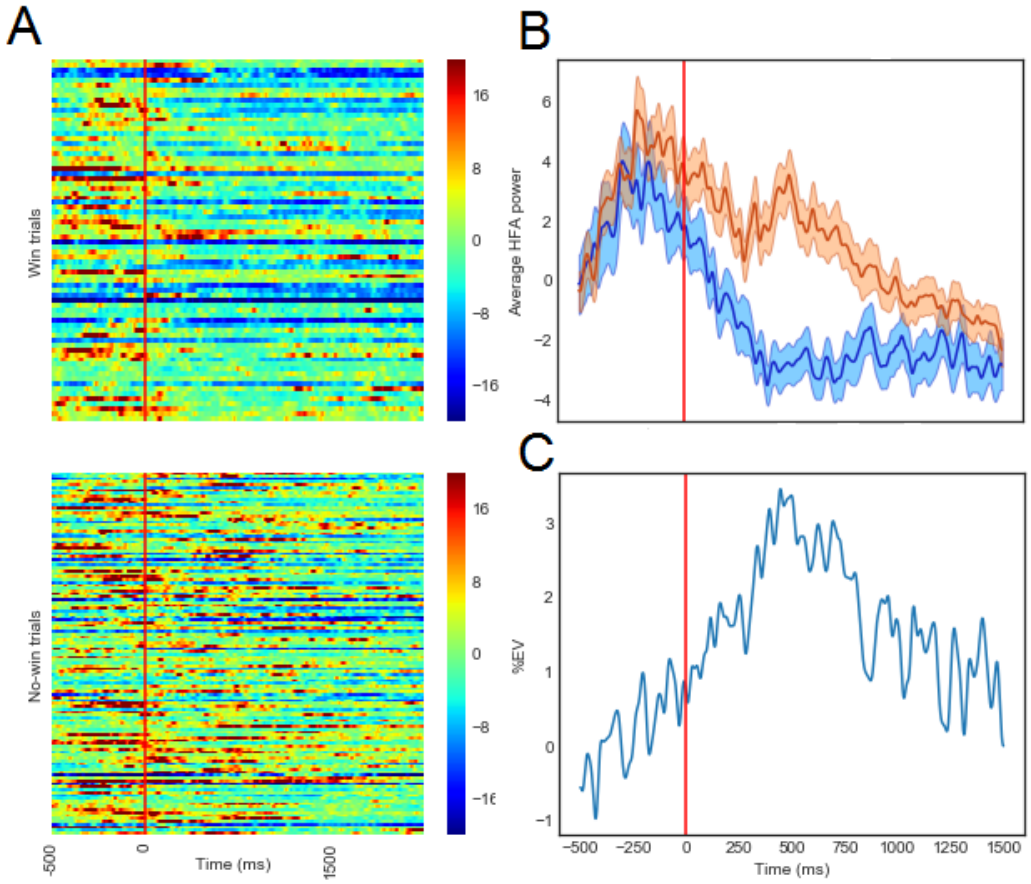

Figure 12. *HFA visualized for electrode 18 of subject 9, which encodes win information where the no-win trials have a higher activity than win trials. The trials in all plots are time-locked to the outcome reveal (t = 0).*

(A) Two heat maps, the top plot depicting win trials, the bottom one all other trials.

(B) The average HFA values for win (blue) and other (red) trials, with SEM (standard error mean) shading.

(C) The explained variance in the HFA signal.

In Figure 12, we can see that the activity is very **clearly separated** between the win and other trials in plots A and B. In the heat maps, the time window after the outcome reveal shows much higher values of HFA for the other trials which is reflected also in the average value in plot B. The explained variance in plot C also clearly increases when the average activity increases, confirming that the electrode encodes information about not winning (and winning) the trial during the few seconds after the outcome reveal.

Figure 13 depicts four more electrodes that encode the win information in a similar way, where no-win trials have higher HFA power values.

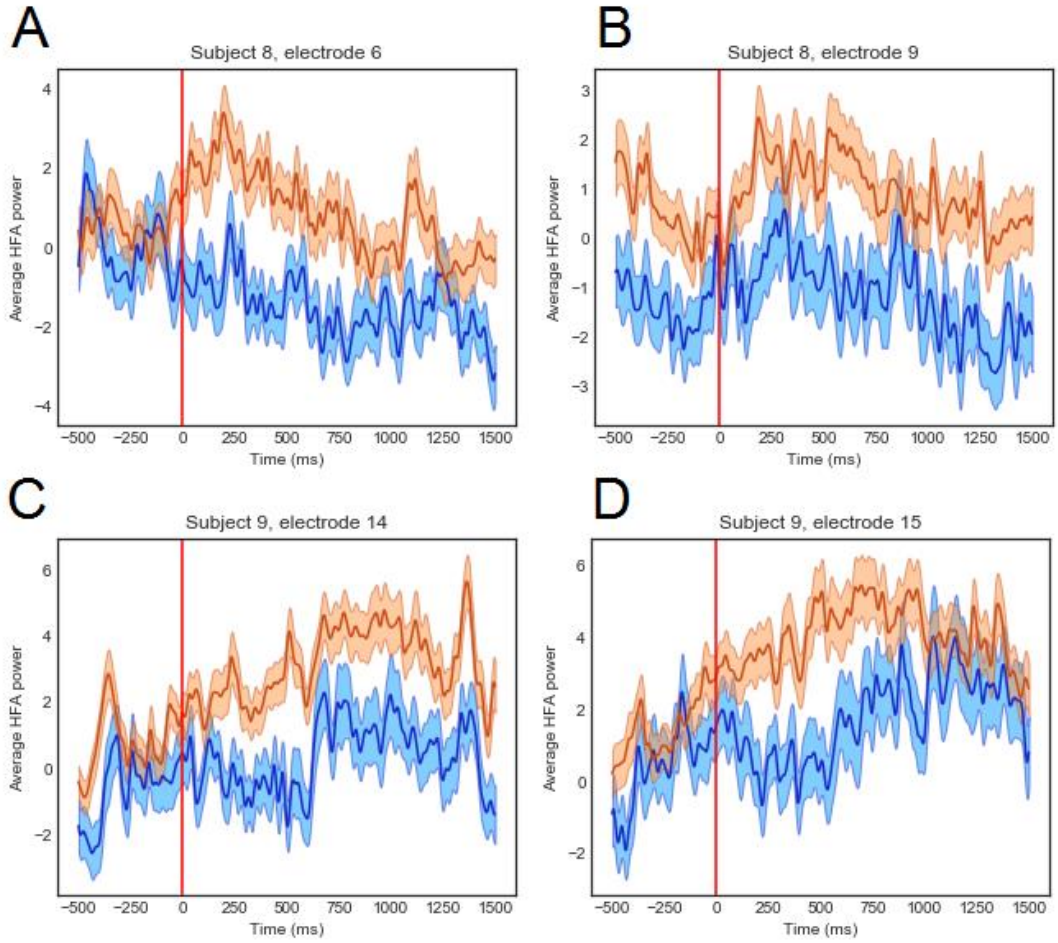

Figure 13. *Four win information encoding electrodes with average HFA values for win (blue) and other (red) trials, shading is SEM (standard error mean). All trials are time-locked to the outcome reveal (t = 0).*

All of these four electrodes from subject 8 and 9 show that the HFA mean values of the other trials are significantly larger than the win trials' average HFA.

Gamble and Safe Bet Trials

We can also compare trials based on the **gamble indicator** – a binary value which indicates whether or not the subject gambled in a given trial. Figure 14 shows an electrode from subject 2 which encodes information about the subject gambling or not.

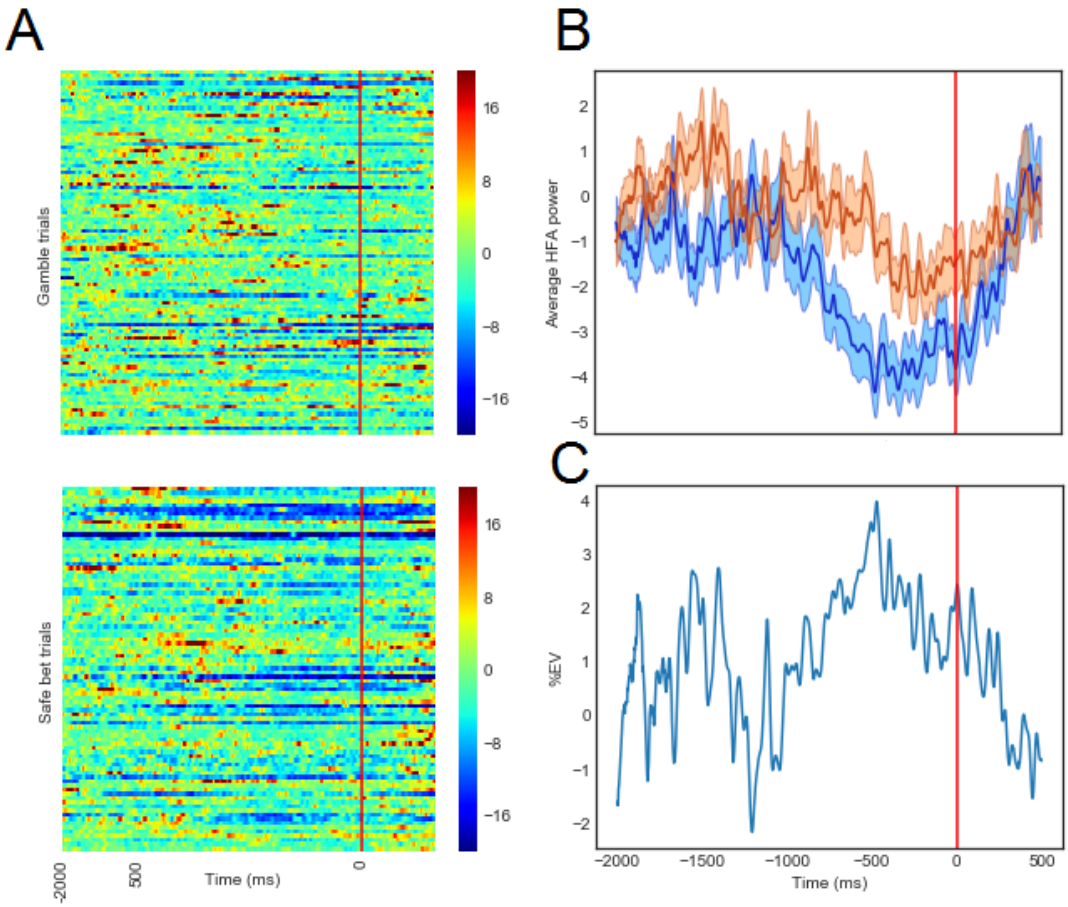

Figure 14. *The HFA analysis for electrode 5 of subject 2, which encodes the gamble indicator. The trials in all plots are time-locked to the choice event (t=0), which is marked with a red vertical line.*

(A) Two heat maps, the top plot depicting gamble trials, the bottom one safe bet trials.

(B) The average HFA values for gamble (red) and safe bet (blue) trials, with SEM (standard error mean) shading.

(C) The explained variance in the HFA signal.

The gamble and safe bet trials are analyzed in the **deliberation period** – the 2 seconds before the button press event (i.e. the moment of choice; marked with a red vertical line in the plots).

In Figure 14, plot A, we can see the gamble trials on the upper subplot and the safe bet trials on the lower subplot. The safe bet trials clearly show **decreased activation** compared to the gamble trials during the seconds immediately preceding the choice event. This is also reflected in plot B, which shows that the average HFA values for gamble trials are **significantly higher** than those of safe bet trials. The explained variance in plot C confirms this, as it is **increased simultaneously with the increase of average activity.**

More examples of electrodes with increased activity for gamble trials during deliberation are shown in Figure 15.

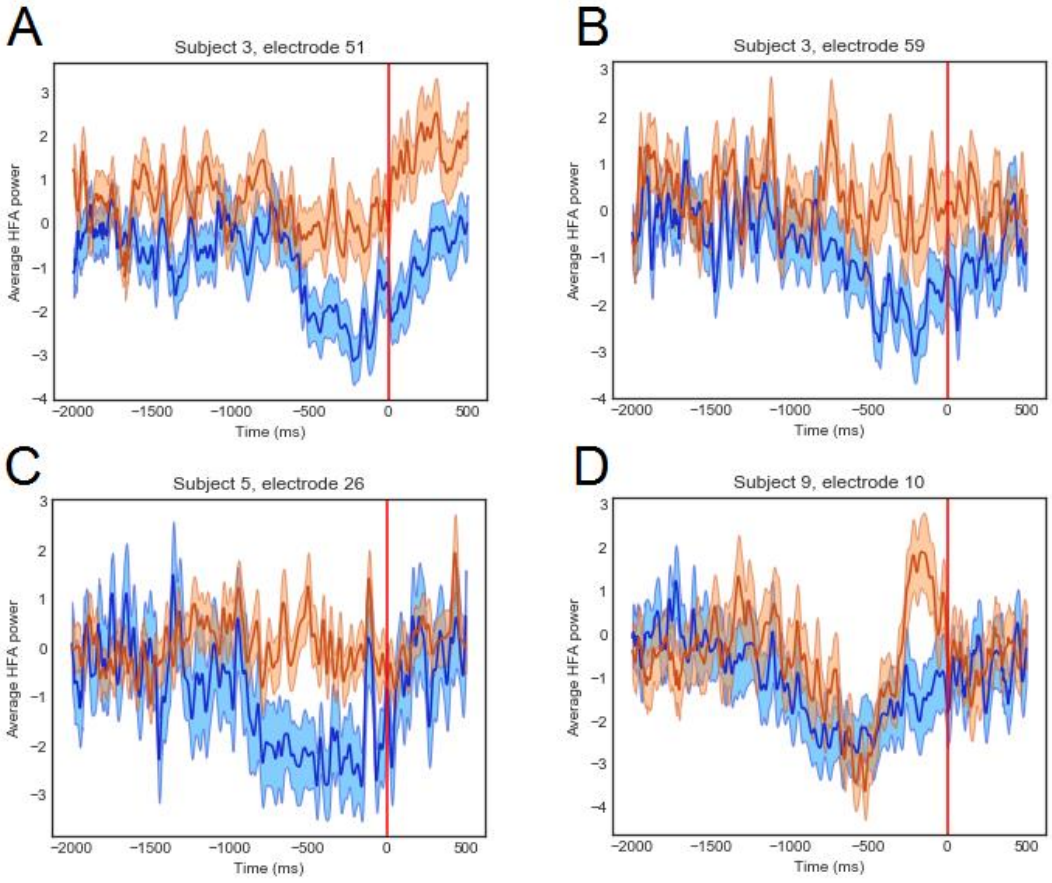

Figure 15. *Four gamble indicator-encoding electrodes with average HFA values for gamble (red) and blue (safe bet) trials, shading is SEM (standard error mean). All trials are time-locked to the button press event (t = 0).*

In Figure 15 we see 4 electrodes with average HFA value **significantly higher for gamble trials**, with a clear spike of activity for the 10th electrode of subject 9, which might indicate the moment the patient chose to risk before pressing the button.

There are also electrodes in the data, which encode the '**safe bet indicator**' (i.e. the negated gamble indicator) with higher activity values. One example is depicted in Figure 16, namely electrode 9 of subject 8.

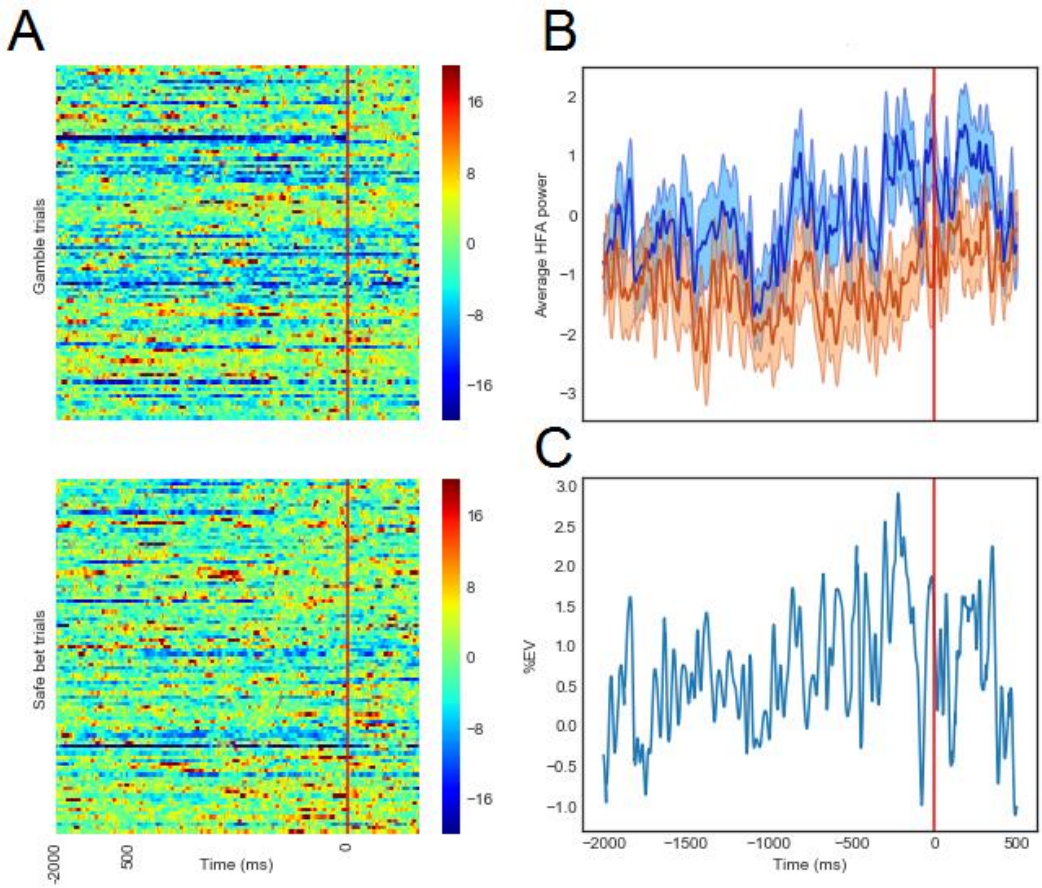

Figure 16. *The HFA analysis for electrode 9 of subject 8 which encodes the safe bet indicator (i.e. negated gamble indicator) with higher HFA values. The trials in all plots are time-locked to the choice event (t=0).*

(A) Two heat maps, the top plot depicting gamble trials, the bottom one safe bet trials.

(B) The average HFA values for gamble (red) and safe bet (blue) trials, with SEM (standard error mean) shading.

(C) The explained variance in the HFA signal.

From Figure 16, we can clearly see that the electrode encodes safe-betting (and gambling) information, as the lower heat map in panel A, depicting the HFA of safe bet trials, shows **higher HFA values during the deliberation period**, as does the average safe bet HFA signal in plot B. Plot C covariates with plot B, also indicating that the gamble information is encoded by the electrode.

Four other electrodes that encode the gamble information with a higher activity for safe bet trials are depicted in Figure 17.

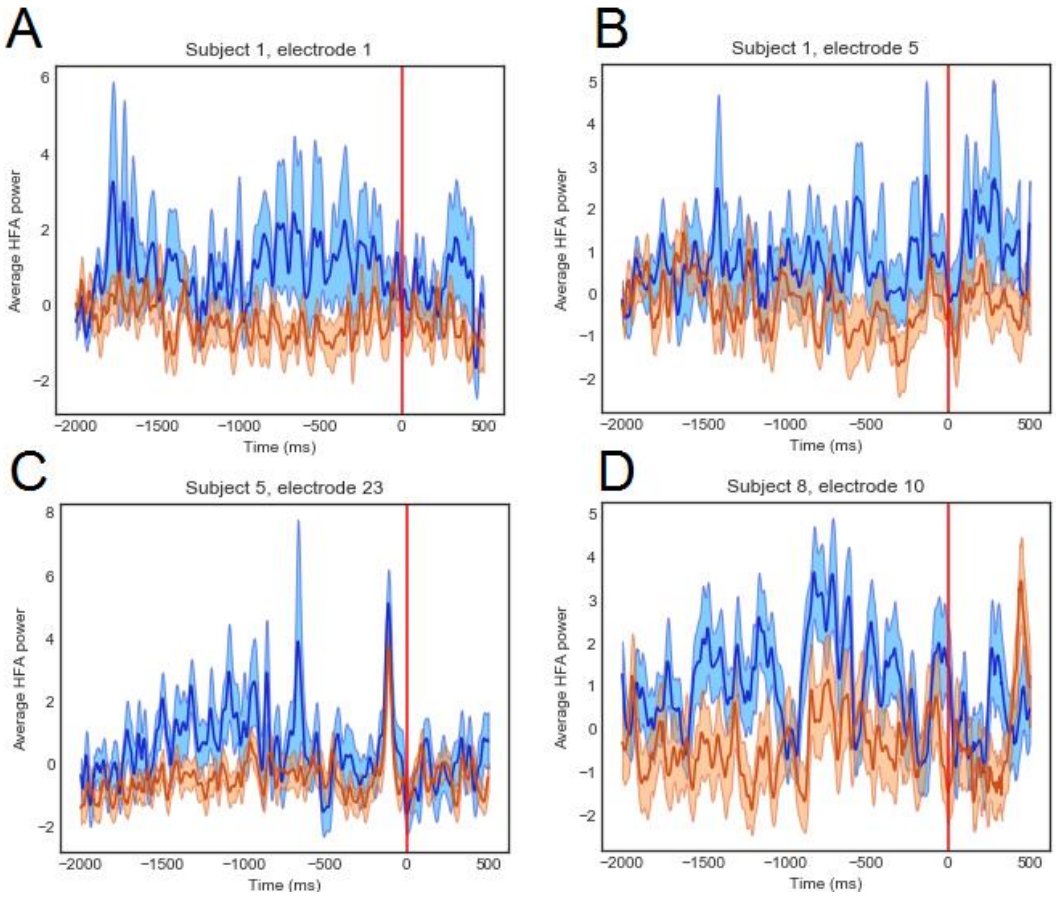

Figure 17. *Four electrodes that encode the no-gamble indicator with average HFA values for gamble (red) and blue (safe bet) trials, shading is SEM (standard error mean). All trials are time-locked to the button press event (t = 0).*

All of these four electrodes in Figure 17 show an **increased HFA average signal** for the **safe bet trials** during the deliberation period, indicating that they encode information about when the subjects decided to play safe.

Two other interesting electrodes are depicted in Figure 18, their average HFA values time-locked to choice event.

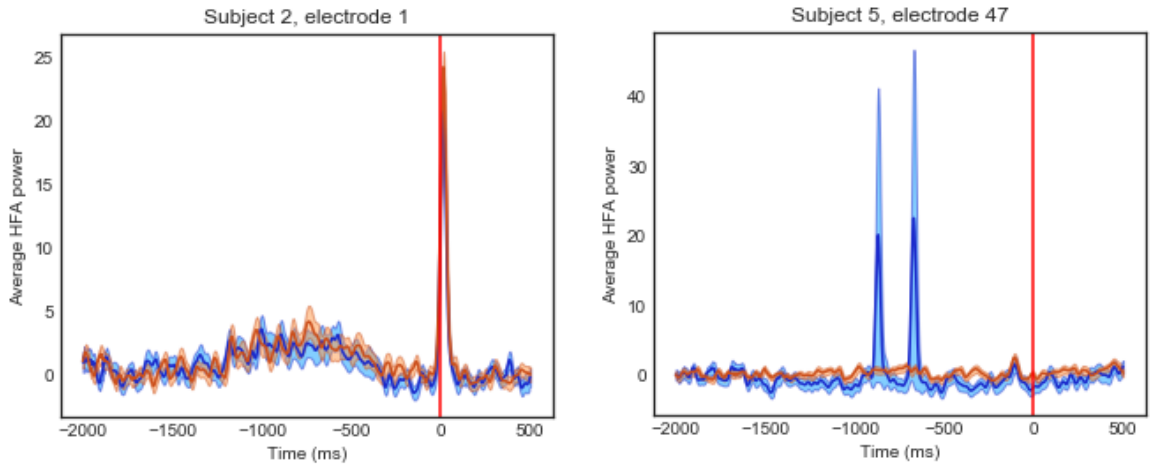

Figure 18. *The average HFA values for gamble (red) and safe bet (blue) trials of two electrodes, with a very specific type of behavior during the trials.*

The HFA signal of **electrode 1, subject 2** (on the left plot) shows a clear spike for both gamble and safe bet trials during the moment of choice, implying that it encodes the physical action of making the choice (pressing the button). **Electrode 47 of subject 5** shows two peculiar spikes during the deliberation period for the safe bet trials, with the HFA power for gamble trials being constantly near zero. This electrode might be activated due to some specific processes during deliberation, considering that the activity is spiked at two precise moments in time.

To conclude, this data exploration has shown that there is a variety of electrode responses in the data that seem to encode different kinds of information, which gives a basis for further analysis on patients' behavior, pattern-finding and correlation in the data.

4.2 Behavioral Analysis

In this section, the results for behavioral analysis are presented. Machine learning methods were applied to collection D1 to answer research questions **Q2** and **Q3**. The following subchapters will focus on the two questions, how they were addressed and what were the results.

Behavioral Tendencies

This subsection will focus on analyzing the **patients' behavior** by looking at **how the risky prize affects their choices, compared to the presentation number**. Both variables represent different aspects of the task and analyzing the coefficients that the linear models produce for the variables will help to provide insights to a subject's behavior and compare it to other subjects, thus answering research question **Q2**.

To analyze and compare the subject's behavioral tendencies towards risk-taking or safe-betting, **5-fold cross-validation** with either **logistic regression** (LR) or **linear support vector classification** (LSVC) was applied to the normalized dataset consisting of the risky prize and the presentation number to predict the gamble indicator. Several other machine learning methods from scikit-learn were tested (refer to Table 2), but are excluded from further analysis due to poorer accuracies and the complexity of interpreting their results and feature importance, which depend on many parameters (e.g. for random forests the maximum depth of the tree, the measure for the quality of a split, the number of trees in the forest, and many more).

Averaging 5-fold cross-validation results with LR and LSVC over all subjects showed that the best penalization method was **L2-norm** with $\boldsymbol{C = 1.0}$ **as** the regularization strength. The tested *C* values were from 0.1 to 1 with increments of 0.1 (i.e. 10 values in total). In addition to mean accuracy, the average **f-score** was also calculated over all folds. F-score is a good measure for determining a classifier's quality: it takes the harmonic mean of precision and recall, which means that it places importance on both minimizing false positives (precision) and identifying as many positive instances as it can (recall) [46].

Figure 10 shows the results from cross-validation, using L2-norm as penalty and 1.0 as regularization strength. From figures A and B, we can see that **overfitting is minimal**, as the **train and test accuracies and f-scores are almost identical** for both models. The images also show that for most patients, the accuracy is quite good – over 70%, except for subjects 4, 5 and 10. These are the subjects with the least amount of trials, with 4 and 5 having significantly less trials included in the analysis (108 and 136 respectively) than the rest of the subjects. This can explain the bad accuracy as there is little data to derive behavior from. However, subject 10 has 179 trials, which makes for quite a lot of data, considering that for example subject 1 has 180 trials, and a prediction accuracy about 20% higher than subject 10. This might imply that the behavior of subject 10 is hard to predict with a linear model.

The results of the two models are very similar, with both the accuracies and the feature coefficients being almost equal.

Panel C in Figure 19 shows the **absolute values of the mean coefficients** of **risky prize** and **presentation number** plotted together. This plot allows to compare the behavior of different subjects. However, the cross-validation accuracy for subjects 4, 5, and 10 was very

low, which means that the coefficients provided by the models are not a trustworthy indicator of behavior. Therefore, these subjects cannot be reliably compared with the other subjects in the analysis.

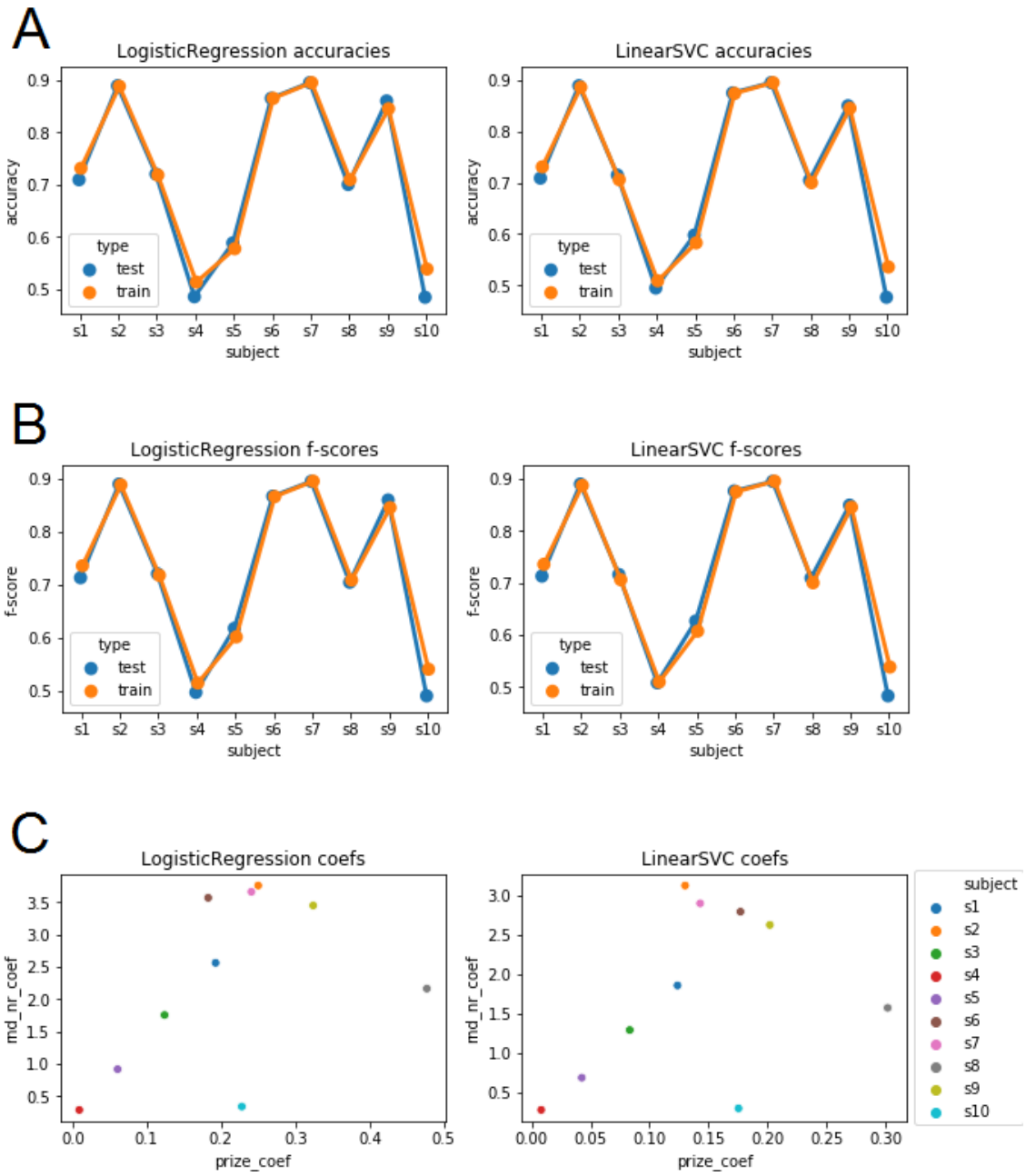

Figure 19. *The behavioral risk analysis results for each subject.*

(A) The average cross-validation accuracies for test and train set.

(B) The average cross-validation f-scores for test and train set.

(C) The absolute values of the mean coefficients of risky prize and presentation number on the x- and y-axis respectively. The average was taken over the 5 coefficient values from cross-validation folds.

To help with the analysis, we can look at Table 3, which shows the percentage of trials where subjects gambled and did not gamble.

Table 3. *Percentage of gamble and non-gamble trials for every subject.*

%Trials	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Gamble	64%	54%	47%	45%	73%	58%	49%	50%	48%	58%
Safe bet	36%	46%	53%	55%	27%	42%	51%	50%	52%	42%

In Figure 19, we can see that from the remaining seven patients, subjects 2, 6, 7 and 9 are quite similar in their behavior, as the corresponding points are positioned at a close range and there is small difference in the percentage of gamble trials (as seen in Table 3). Notably, these subjects have very high accuracies and f-scores – all nearly 0.9 – which suggests that their coefficients reflect their behavior well. Judging from the positioning, these subjects seem to apply a much higher emphasis on the win probability than the risky prize, indicating a more safe-bet reasoning.

The coefficients for subjects 1 and 3 indicate a more moderate behavior as they lie on the middle of the spectrum, not very close to any extremes. Inversely, subject 8 places strong emphasis on the risky prize compared to the presentation number, implying that the monetary aspect heavily impacts their decision.

In conclusion, this section answers **Q2** by showing that the spectrum of behavior among the participants is wide, ranging from moderate to extremes, with some preferring to prioritize the monetary aspect, risky prize, and others the probability of winning. The models were accurate for most of the subjects, except for subjects 4, 5 and 10. This might have been caused by the scarcity of data (for subjects 4 and 5), or the model's inability to capture unexpected behavior.

Predicting Event Indicators

This subsection presents the classification results for research question **Q3**, which tackles **predicting task-related events** from brain activity in the OFC. One predicted variable is **gamble indicator**, which is connected to the person's own behavior, while the others – **win**

and loss indicators – are external events, only derived from the reactions of the subjects to the outcome reveal.

Datasets in collection **D1** were used for this analysis, running **5-fold cross-validation** with both **logistic regression** (LR) and **linear support vector classification** (LSVC), predicting the loss, win or gamble indicator. The dataset collection D1 also contains datasets with many types of trials – all, loss, win, and gamble. Of these, only those were used which contain **all** of the subject's trials (as the predicted indicators are already implying, which type of a trial it is). As there are 2 time window types and 3 different numbers of time windows, then there are 6 datasets for every subject that are used in this analysis, 60 in total. Half of these are outcome-type, for which the models are run twice – with target variables win indicator and loss indicator. Then, altogether $30 + 30 + 30 = 90$ different analyses were run on each model.

As the loss indicator and win indicator are outcome-related variables, only datasets with time window type *outcome* are used for predicting them, whereas gamble indicator is a choice-related variable and thus, requires datasets with *choice*-type time windows.

Table 4 shows the mean metrics for each predicted variable.

Table 4. *Mean accuracy, precision, recall and f-score for different target variables. These are the average metrics over the cross-validation scores of all datasets. The best results for both methods are colored green.*

Target	Model	Accuracy	Precision	Recall	F-score
Loss indicator	LR	0.816	0.678	0.816	0.738
Win indicator	LR	0.636	0.461	0.636	0.514
Gamble indicator	LR	0.56	0.493	0.56	0.483
Loss indicator	LSVC	0.814	0.688	0.814	0.739
Win indicator	LSVC	0.629	0.549	0.629	0.549
Gamble indicator	LSVC	0.569	0.538	0.569	0.519

We can see from Table 4 that the **loss indicator** is the only variable which shows good results for each metric. The other indicators might be classified correctly for some particular subjects, but overall they have low to mediocre results. The two models do not show particularly large differences in results, with LSVC being a little more successful in balancing precision and recall.

Figure 20 shows the **performance of different indicators for each subject**.

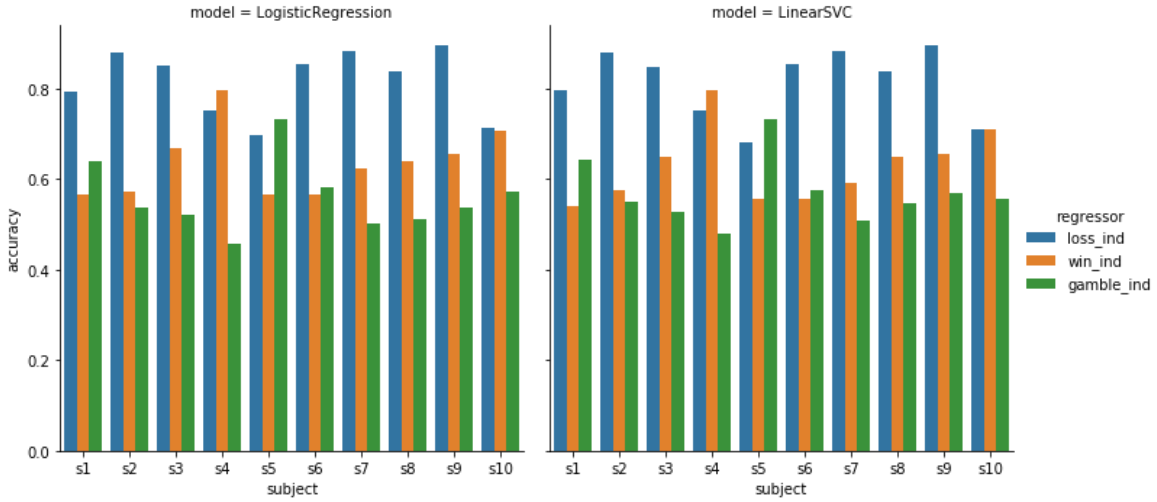

Figure 20. *The average cross-validation accuracies per subject for different target variables. The left plot shows results for LR, while the results for LSVC are on the right.*

From Figure 20 we can see the **accuracies of predicting different regressors**, grouped by the subject included in the dataset. It is clear from the plot **that loss indicator is the most successfully predicted variable**. As before, the results are very similar between the two algorithms. The only two subjects with whom another indicator was more successful, are subjects 4 and 5, with win indicator and gamble indicator having the best results respectively. Other than with subjects 5 and 1, predicting the gamble indicator was unsuccessful. The win indicator generally has slightly better results for the subjects than the gamble indicator, staying mostly close to 60%-70% accuracy.

Figure 21 shows the average **accuracies for a different number of time windows**, grouped by the regressor that was predicted by the model. We can see that the accuracies of time window values are nearly equal, indicating that there is no preference of time window numbers among any of the predicted variables.

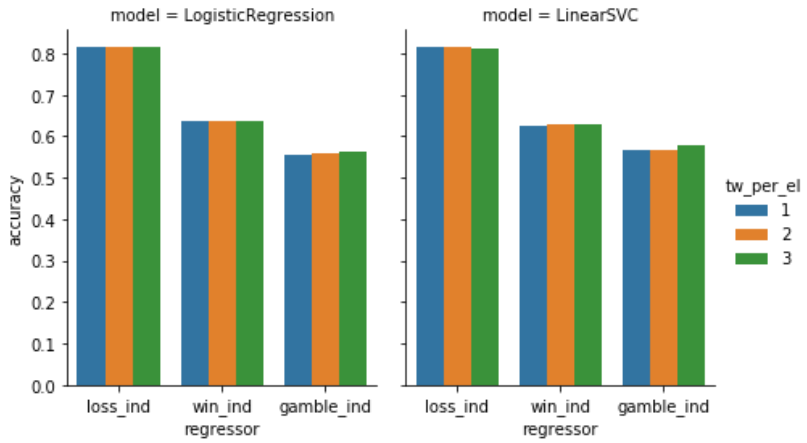

Figure 21. *The average cross-validation accuracies for different number of time windows per electrode, categorized by the target regressor. The left plot shows the average accuracies for the LR model, while the right plot depicts the LSVC results.*

To conclude, the analysis had very good results for **loss indicator** across all subjects, while mostly mediocre accuracies for the other indicators, depending on the patient.

4.3 Electrode-Regressor Group Correlation

In this section, the results for CCA are presented, which tackled the research question Q1: **is there correlation between groups of electrodes and regressors**? Namely, it tries to find separate weights for a dataset of HFA values and a dataset of regressor values to maximize the correlation between their linear combinations. High correlations would indicate **a complicated construction of HFA signals** in the brain to encode multiple task-related values.

The first subchapter describes the **process** of performing CCA on the data, while the second specifies some **notation** that is necessary to follow the rest of the analysis. The third subsection presents the **results**.

Processing Pipeline

CCA was applied to all datasets in collection D2 and all combinations of different regressors by performing **5-fold cross-validation** on them. Specifically, if given one D2 dataset X and one regressor dataset Y, then for every fold, CCA was run on the training sets X_{train} and $Y_{train}$to obtain the feature weights a and b for HFA features and regressors respectively. Then, the dot products of the weight vectors (a and b) and the corresponding test datasets (X_{test} and Y_{test}) were calculated, resulting in two vectors $v_1 := aX_{test}^T$ and $v_2 := bY_{test}^T$.

The test score of one fold was calculated as the Pearson correlation coefficient between vectors v_1 and v_2.

It is important to measure the correlation on a test set because CCA could overfit heavily on the data. Cross-validation makes use of the whole dataset to the full extent – by taking the average test (and train) correlation of all folds, we can get a general idea of how the method performs on the data.

It has to be considered that for some subjects (e.g. subject 4 with only 108 trials) one fold contains only about 20 trials, which makes it harder to find correlation in the data. CCA also tackles a more complex problem than trying to predict one variable, optimizing weights for two datasets to maximize the correlation between their dot products. In this case, finding the best solution is further complicated by the extreme noisiness of neural data. The signals may contain information that is not related to the regressors at all, for example OFC is known to code even the specific identity of sensory stimuli [1].

To perform CCA, **a HFA and a regressor dataset** need to be selected. This is equivalent to choosing the **six parameter values**, which determine the nature of the datasets. These parameters are described in Table 5.

Table 5. *The parameters of CCA. To perform CCA, one option was selected for each of these parameters. These determined the datasets that CCA would be performed on.*

Parameter	**Values**
Subjects	10 subjects
Electrode selection	3 options/methods: 1. use all electrodes of the subject, 2. use electrodes selected by single regressor correlation, 3. use electrodes selected by group regressor correlation, with a percentage $p \in \{20\%, 40\%, 60\%, 80\%\}$
Time window type	2 types: choice or outcome

Time windows per electrode	3 options: 1, 2 or 3 time windows per electrode
Regressor trial type	2 types: current or past
Regressor combinations	11 combinations per trial type and time window type ($2 \cdot 2 \cdot 11 = 44$ combinations in total)

To select one HFA dataset from collection D2, the necessary parameters are the subject, electrode selection method, time window type and number of time windows per electrode. The dataset is a matrix of dimensions *nTrials* × *(nElectrodes* · *nTimeWindowsPerElectrode),* where *nTrials* and *nElectrodes* depend on the chosen subject (see Table 1). There are a total of 360 datasets in D2.

The regressor dataset depends on the selected HFA dataset's **time window type**. If it is *choice,* then only choice-type regressors are selected; similarly with type *outcome*. The regressors could be either **only current** or **only past** trial regressors. This depends on the chosen **regressor trial type**. Therefore, combining the time window and trial types, there are 4 possible sets where to choose regressors from: current outcome, current choice, past outcome or past choice regressors. To make a regressor dataset, 2-4 regressors were chosen as features from one of these sets. Thus, the regressor dataset is an *nTrials* × *nRegressors* dataset, where *nTrials* depends on the subject (refer to Table 1).

Choosing *past* regressors for a dataset means that for every trial in the HFA matrix correspond the regressors of the **immediately previous trial**. Therefore, as the first trial does not have a previous trial, it is always excluded from the HFA dataset if *past* regressors are used in the regressor dataset.

As an example, the 2-element combinations for current outcome regressors are: $\{regret, rpe\}, \{regret, win\ indicator\}, \{regret, loss\ indicator\}, \{rpe, win\ indicator\},$ $\{rpe, loss\ indicator\}, \{win\ indicator, loss\ indicator\}$, as the current outcome regressors are regret, RPE, win and loss indicators.

By combinatorics, there are 11 possible combinations of regressors from a set of 4 that consist of at least 2 elements. As there are 2 sets of outcome regressors (current outcome and

previous outcome) and 2 sets of choice regressors, there are 22 possible *outcome* regressor combinations and 22 possible *choice* regressor combinations.

As there are 360 HFA datasets in D2, where every dataset has a certain time window type and there can be 22 regressor combinations per time window type, there are a total of $360 \cdot 22 = 7920$ possibilities of HFA and regressor dataset duos (i.e. parameter value combinations in Table 5) to perform CCA on. All of these 7920 analyses were conducted and analyzed collectively.

As an example, we select subject 3, choose electrodes based on group regressor correlation with p = 20%, time window type *choice* and 2 time windows per electrode. For the regressor dataset, we choose two past choice regressors: past risk and past win probability. Subject 3 has 59 electrodes and 194 trials altogether. The resulting HFA dataset is a 193 × (59 * 0.2 * 2) matrix, as 20% of the best electrodes are selected from the 59 total and 2 time windows are calculated per electrode. The regressor dataset is a 193 × 2 matrix with past risk and win probability values for every trial. The number of trials is 193 because we are using **past regressors** and as the very first trial of the experiment does not have a previous trial, it is excluded from the dataset.

Notation

For the sake of brevity and concreteness, these parameter combinations will be referred to in brackets in the following order: **[subject, trial type, time windows per electrode, set of regressors, electrode selection method].** The time window type is omitted because the regressors already imply the used time window (i.e. *choice*-related (outcome-related) regressors always use the *choice* (outcome) time window type).

Table 6 specifies the notation for each of the parameters.

Table 6. *The notation for the parameters used in CCA. The notations are given with an example case (where necessary).*

Parameter	**Notation**	**Example**
Subject	'S' + the number of the subject	Subject 10 is referred to as S10.
Trial type	'current' or 'past'	-

Time windows per electrode	'TWPE' + the number of time windows per electrode	If the number of time windows per electrode is 2, then the parameter is marked as TWPE2.
Combination of regressors	The set of regressors is given in braces. **NB!** If the regressors are of past trials, then this is not marked explicitly in the combination, as the trial type already specifies this.	If the regressors are risk and win probability, then the combination is written as {risk, win probability}.
Electrode selection method	1. If all electrodes were used, then 'all'. 2. If single regressor correlation was used, then 'single'. 3. If group regressor correlation was used, then 'group' + the percentage of electrodes that were used.	If group regressor correlation was used with 20% of the best electrodes, then this will be referred to as 'group 20%'.

As an example, if we wanted to refer to a parameter combination, which used subject 5, the past trial regressors win indicator and loss indicator, with 3 time windows per electrode and the electrode selection method was single regressor correlation, then it would be marked as **[S5, past, TWPE3, {win indicator, loss indicator}, single]**.

Note that for the regressors, the word 'past' is not repeated to be concise.

Electrode-Regressor Analysis

Due to the noisiness of the data, most of the combinations resulted in a low test score, with only 60 combinations of the 3960 using **current trial regressors** resulting in an average

test correlation above 0.3, and of them only one having an average of at least 0.4 test correlation. This means about 1% of all current trial combinations resulted in a test correlation that indicates a possible existence of a complex signal in the OFC.

Surprisingly, the **past trial regressors** had much more successful results: 329 combinations out of 3960 resulted in a test correlation above 0.3, with 37 combinations having a test score above 0.6.

This section will further concentrate on the results that scored an **average test correlation above 0.3.**

Figure 22 shows the percentage of combinations that each regressor appeared in, where the combinations' test correlations were above 0.3. The percentage was calculated separately for combinations with *past* regressors and *current* regressors.

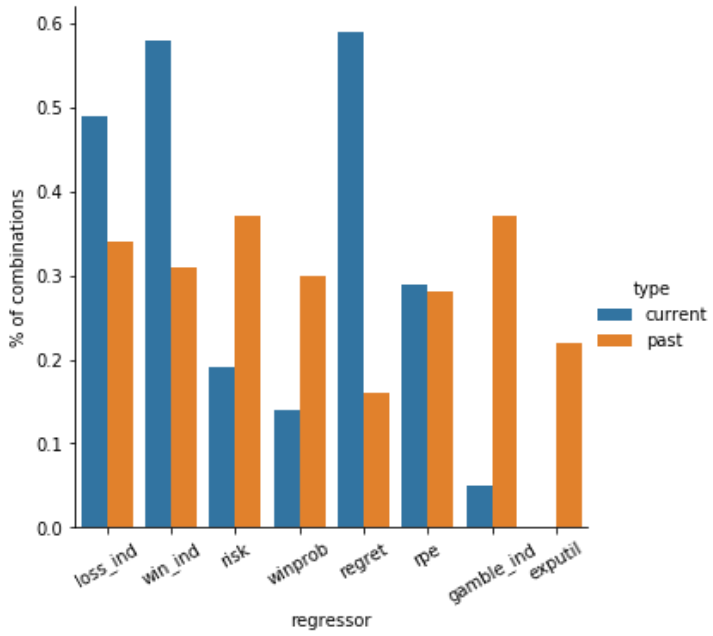

Figure 22. *The percentage of the current/past trial combinations with test score larger than 0.3 that the regressors appeared in.*

We can see that the **loss** and **win indicators** appeared very often in the successful *current* trial combinations compared to other regressors (~50% and ~58% respectively), with the **expected value** being the only one not to be in any of the top *current* trial combinations. Overall, it seems that the *current outcome*-related regressors gave significantly better results than the *current choice*-related regressors, whereas the division with *past* trial regressors is

not that clear. For the *past* trial regressors, the results are more mixed, with no regressor particularly standing out, and **regret** being the only one with less than 20% of combinations..

Figure 23 shows the percentage of top current/past combinations where each subject appeared in. With the *current* regressor combinations, **subject 5** appears to be the most successful, with also subject 3 and 10 being prevalent. For both *current* and *past* regressors, subjects 6 and 7 did not achieve a test correlation above 0.3 with any parameter combination. With the *past* regressor combinations, it appears that **subject 2** had the most combinations of parameters with good test correlations. The other subjects appear far less, with only subject 4 being in over 20% of the top *past* regressor combinations. Subject 1 did not appear in any top *past* regressor combinations.

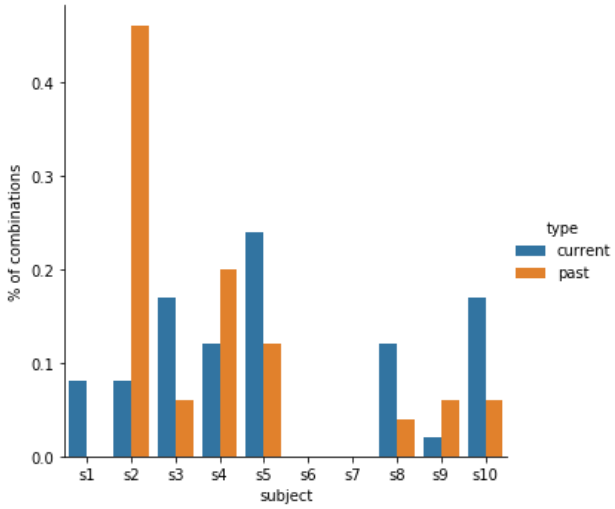

Figure 23. *The percentage of current/past trial combinations with test correlation larger than 0.3 that each subject appeared in.*

We can also compare the electrode selection options in the top combinations, which are depicted in Figure 24.

Figure 24 shows that the best option for both *current* and *past* regressor top combinations was choosing 20% of the best electrodes with **group regressor correlation**. Selecting electrodes based on single group correlation is also very successful for *current* regressor combinations.

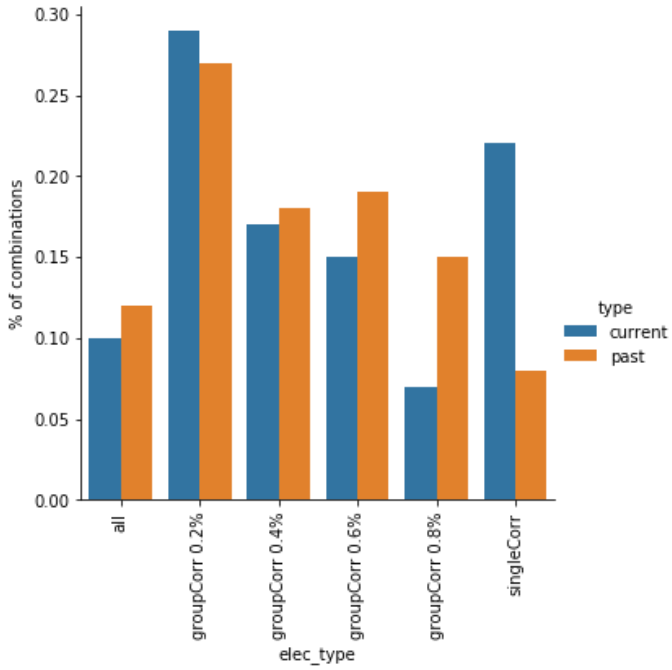

Figure 24. *The percentage of current/past trial combinations with test correlation larger than 0.3 that each electrode selection method appeared in. 'all' – all electrodes, 'singleCorr' – selected with single regressor correlation method, 'groupCorr x%' – selected with group regressor correlation method, using x percent of the best electrodes.*

Finally, we can also compare the number of time windows per electrode by the number of top combinations they appeared in.

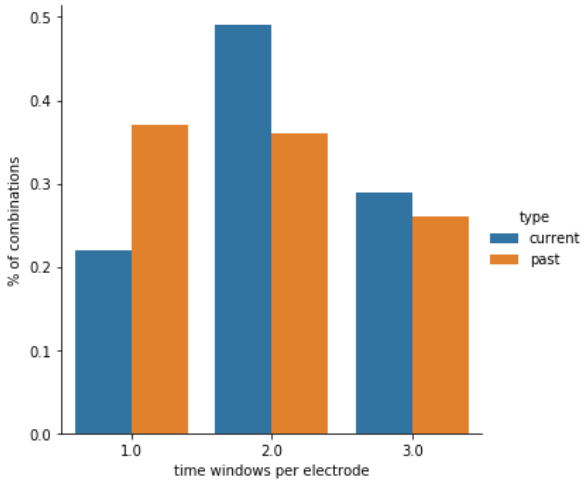

Figure 25. *The percentage of current/past trial combinations with test correlation larger than 0.3 that each number of time windows appeared in.*

From Figure 25 we can see that the tendency for the *current* and *past* regressor results is reversed: while for *current* regressor combinations, it was better to use more than one time window per electrode, then for *past* regressors the number of top combinations decreases as the number of time windows increases. We can gather that the best option for *current* regressors was using **2 time windows per electrode**, while for *past* regressors it was the **entire time window's average**.

We can also analyze specific combinations by looking at the weighted sums of the test and train data, where the weights are the ones provided by CCA. These weights are multiplied with the HFA and regressor dataset used in CCA, which produces two vectors of length *nTrials*. The points in these vectors can be combined as coordinates and plotted to see if there is a visible correlation in the dataset, given the weights calculated by CCA.

In other words, one data point in this plot represents one trial in the dataset (train or test). The x-coordinate is the weighted sum of the trial's HFA values, while the y-coordinate represents the weighted combination of the regressor values. The plotted train and test set are the ones of the best fold in cross-validation, i.e. the fold with the highest test correlation.

Firstly, we look at **current regressor combinations**. Figure 26 shows the three of the best combinations of *current* regressor parameter combinations.

We can see from the plot A in Figure 26 that there are three distinct stripes of points. This is due to the fact that loss and win indicator have only two values, 0 and 1, and thus, there are only three values which a linear combination of them can achieve: when both indicators are 0, when win is 1 and loss is 0, and the other way around (both cannot be 1 at the same time). Therefore, this is not a good combination of regressors where we can analyze the existence of a complex signal.

The second plot shows a stronger linear relationship between the different trials. While the points are quite scattered along the x-axis, there is still an inclination of a correlation between the weighted sums of HFA and regressor values. The third plot is like a mixture of the previous two, showing some relationship between the trials, but is heavily influenced by the indicators taking only two values.

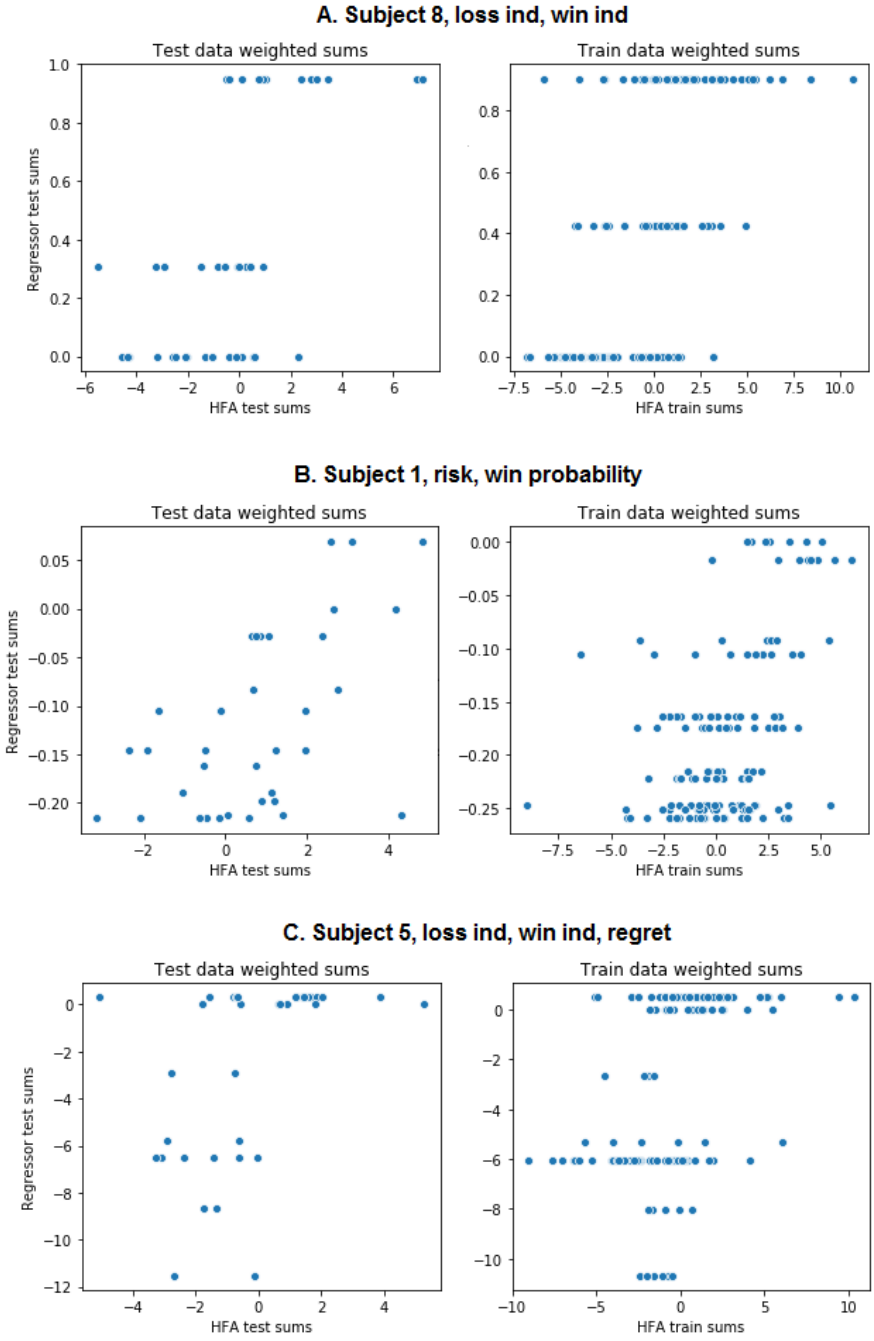

Figure 26. *The weighted sums of three combinations which had a high average test correlation during cross-validation.*

(A) The combination [S8, current, TWPE3, {loss indicator, win indicator}, group 60%]. The average test correlation was 0.4, while the train correlation was 0.55.

(B) The combination [S1, current, TWPE2, {risk, win probability}, group 60%]. Average test correlation 0.39, train 0.41.

(C) The combination [S5, current, TWPE3, {loss indicator, win indicator, regret}, single]. Average test correlation 0.37, train correlation 0.43.

The rest of the combinations for *current* trial regressors are with lower scores and show weaker relationships in the data and thus will not be described further in here.

Secondly, we analyze the **past regressor best combinations** (test score above 0.3). As before, we plot the weighted sums of the train and test data for the best combinations, where train and test data split is chosen based on the fold that had the highest test correlation among the 5 folds.

We can see from the plots in Figure 27 that the data points are gathered into two clusters at the bottom left and top right of the plots. As the gamble indicator takes only two values (0 or 1) and in plot A, the win probability is the only other variable in the set, then it is clear why these clusters are forming. If the gamble indicator is 0, then it would be presumable that the win probability is low as well, resulting in a low regressor test sum. The weighted sum is large when the probability of winning is great and the gamble indicator is thus 1. Otherwise, the points are quite scattered in the clusters, which is especially visible in the train data subplot (in panel A) and thus it would be too presumptive to claim evidence for a complex signal combining gamble indicator and win probability. The same applies to panels B and C, where the binary regressor is the one increasing the correlation, no matter the other regressors in the combination.

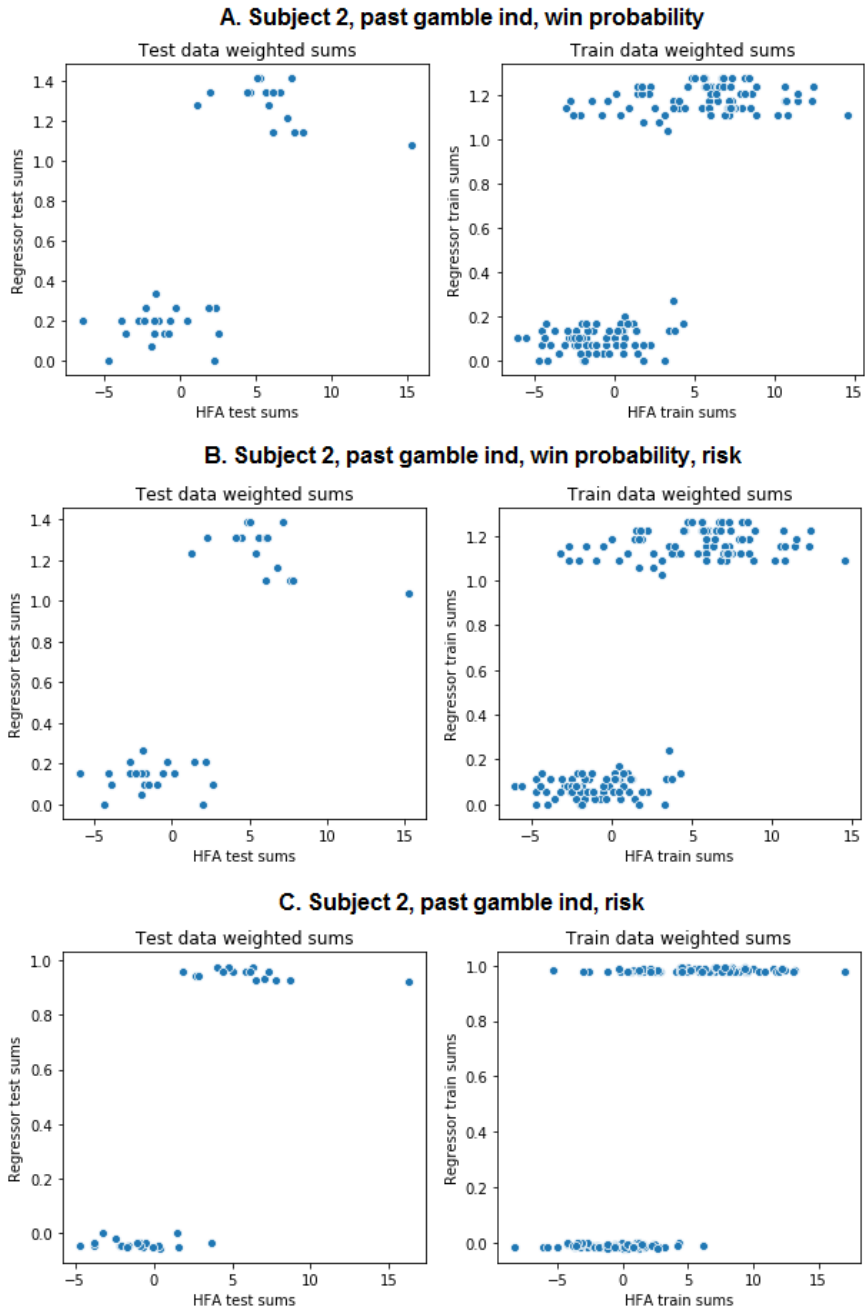

Figure 27. *The weighted sums of test and train data trials for three best past regressor combinations.*

(A) Combination [S2, past, TWPE2, {gamble indicator, win probability}, all]. Average test correlation 0.66, average train correlation 0.71.

(B) Combination [S2, past, TWPE2, {gamble indicator, risk, win probability}, all]. Test correlation 0.66, average train correlation 0.71.

(C) Combination [S2, past, TWPE2, {gamble indicator, risk}, group 40%]. Both test and train correlation 0.66.

Overall, the *past* regressor combinations have the best results mostly with **subject 2** and the **gamble indicator**. In fact, the best 109 combinations of *past* regressors are with using a subject 2 dataset and the 30 best combinations of *past* regressors all contain gamble indicator as a regressor.

Looking at the best combinations of **other subjects**, the binary indicators are prevalent there as well. Figure 28 shows three combinations of other subjects which had a test correlation higher than 0.3 with *past* regressors.

From plots A and B we can see horizontal stripes of points, where the loss and win indicator gain the value 0 or 1, and other than these orderly lines, the points do not exhibit a correlation, which could allow to make more speculations about the signal. We can see a clearer linear correlation in plot C, where the sums for RPE and win indicator are plotted with the linear combination of HFA values from a subject 10 dataset. Notably, all of these combinations achieved the results with **2 time windows per electrode**, using **group regressor correlation** to select **40%** of the best electrodes.

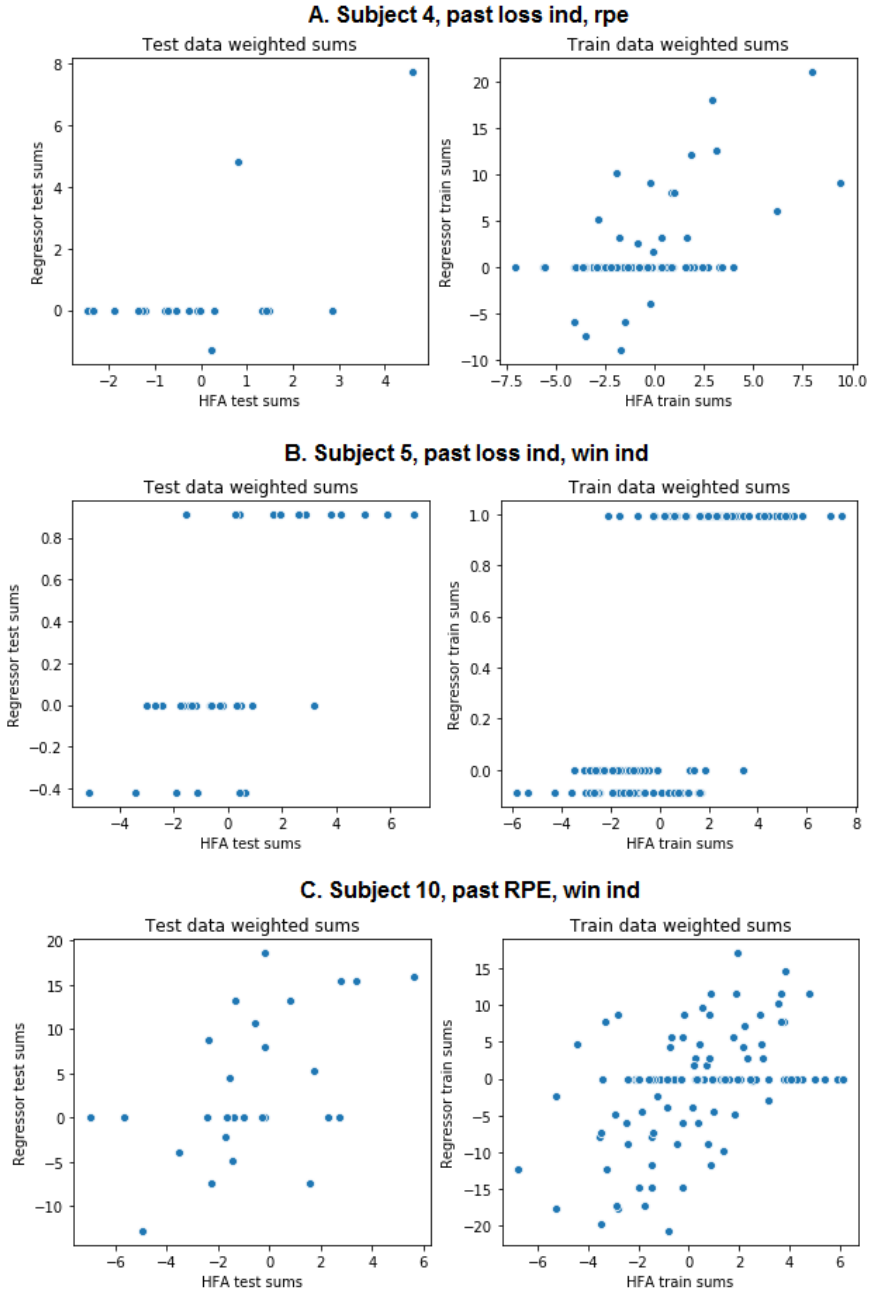

Figure 28. *The weighted sums of test and train data of three combinations of various subjects. These were some of the few past regressor combinations with other subjects than 2, which got average test correlation above 0.3.*

(A) Combination [S4, past, TWPE2, {loss indicator, RPE}, single]. Average test correlation for combination 0.38, train correlation 0.44.

(B) Combination [S5, past, TWPE3, {loss indicator, win indicator}, group 20%]. Average test correlation 0.44, train correlations 0.7.

(C) Combination [S10, past, TWPE1, {RPE, win indicator}, group 60%]. Average test correlation 0.36, average train correlation 0.45.

We can look at more interesting combinations of *past* regressors, where all regressors **take more than 2 values**. Figure 29 shows three such *past* regressor combinations for **subject 2**.

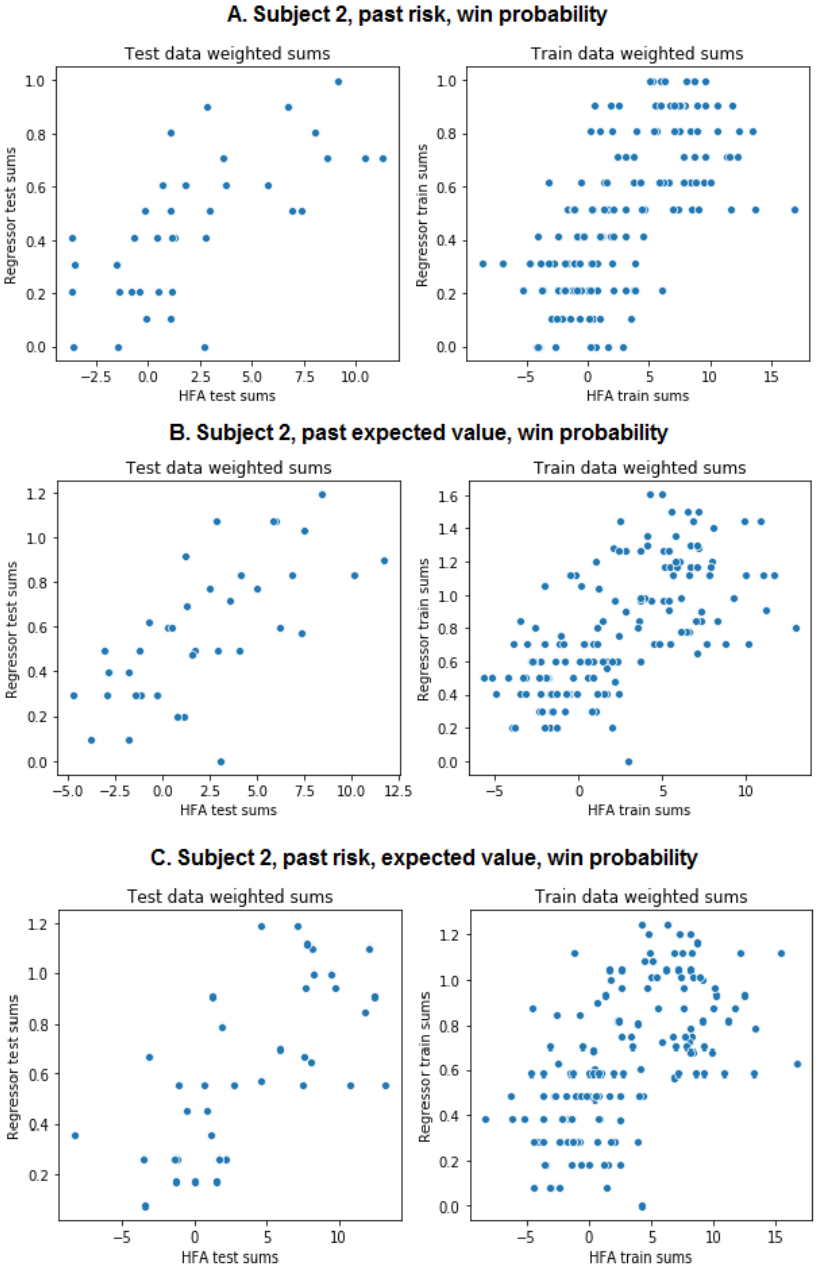

Figure 29. *The weighted sums of test and train data of three combinations for subject 2.*
(A) Combination [S2, past, TWPE2, {risk, win probability}, group 60%]. Average test correlation 0.61, train 0.63.
(B) Combination [S2, past, TWPE2, {expected value, win probability}, group 80%]. Average test correlation 0.57, train 0.62.
(C) Combination [S2, past, TWPE2, {expected value, risk, win probability}, group 40%]. Average test correlation 0.59, train 0.57.

In Figure 29 we can see evidence of a clear strong linear relationship in the data, which implies a more complicated signal in the OFC, combining several choice-related variables. Such strong relationships in the data also exist for other subjects, as demonstrated in Figure 30.

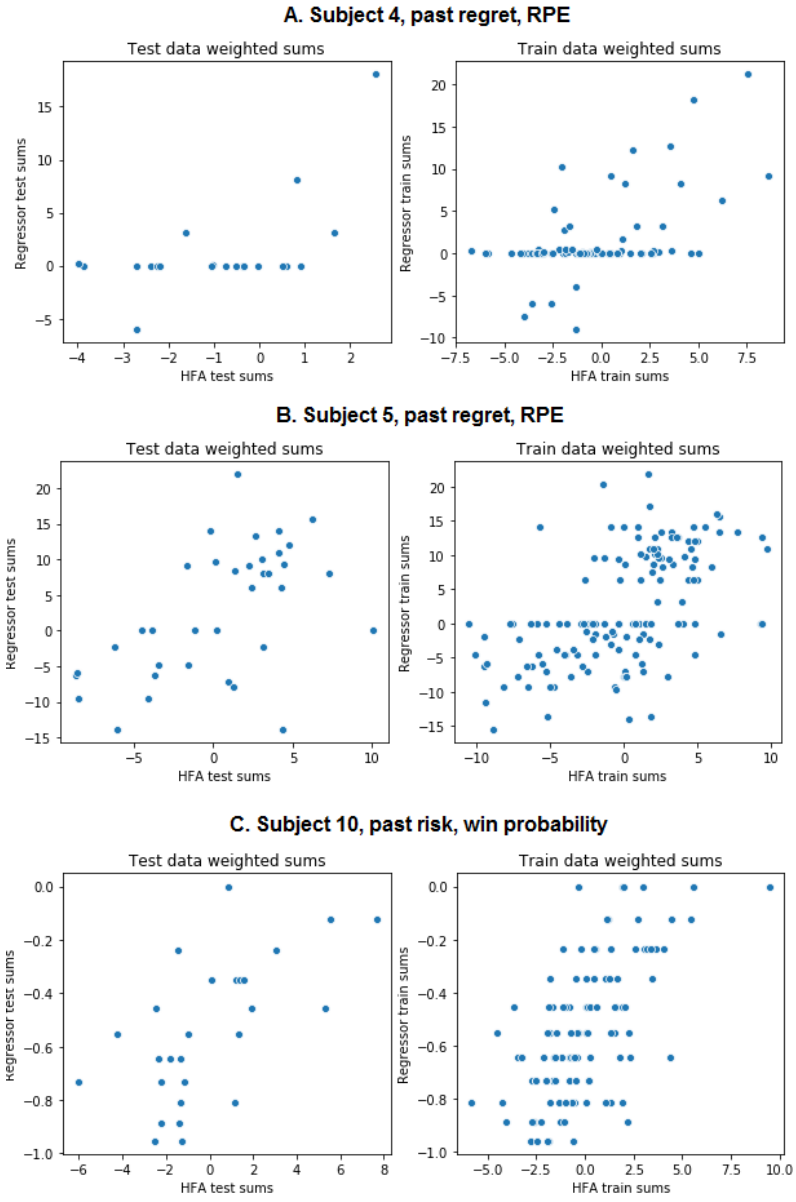

Figure 30. *The weighted sums of test and train data of three combinations for various subjects.*

(A) Combination [S4, past, TWPE2, {regret, RPE}, group 60%]. Average test correlation 0.4, train 0.41.

(B) Combination [S5, past, TWPE1, {regret, RPE}, group 20%]. Average test correlation 0.39, train 0.5.

(C) Combination [S10, TWPE2, {risk, win probability}, group 60%]. Average test correlation 0.35, train 0.56.

We can see that compared to the results of subject 2, the test correlations are noticeably lower, but the underlying linear relationship in the data is still clearly visible in the plots.

These results hint at evidence of a complex signal propagated in the OFC during decision making and give basis for further analysis of these results. However, these results have to be taken with caution as each model in the cross-validation is trained and tested only once. Even though there are many models with different variables tested in this Thesis, a multiple comparison correction analysis should be done in the future to obtain a proper statistical significance [47].

4.4 Visualizing Trial-Electrode Patterns

In this section we are interested in **detecting patterns between the trials and electrodes** in the data. In the first subsection, **biclustering** tools are applied to the data, after which the clusters are visually and mathematically evaluated to discover patterns. The second subsection presents the **visualizations of the t-SNE algorithm**, which is used on each patient's HFA data. Specifically, the aim is to test **if trials of the same type** – win, loss or safe bet trials **are clustered together**.

Biclustering

This subsection presents the biclustering results, which aim to answer **Q4**: **are there groups of electrodes that behave similarly for some sets of trials?** We use biclustering on data, where the rows correspond to a subject's trials and columns to electrodes. Applying biclustering to this data can reveal electrodes that behave similarly for some subsets of trials, e.g. during loss trials or gamble trials.

Both *SpectralCoclustering* and *SpectralBiclustering* algorithms were applied to **every dataset in collection D2** and their performance over various parameters was evaluated.

Table 7 shows the parameters and their values which were used in the biclustering analysis. These were used to select datasets from the collection D2 and to provide parameter values for the biclustering algorithms.

Table 7. *The parameters and their values which were used in the biclustering analysis to produce data and provide parameters for the algorithms.*

Parameter	**Values**
Subject	1-10
Trial types	all, gamble, loss and win
Time windows per electrode	1, 2 or 3
Time window types	choice and outcome
SVD methods	*randomized* and *arpack*
n_clusters	2-7 **NB!** For the *SpectralCoclustering* algorithm the parameter *n_clusters* directly indicates the number of clusters that are calculated in the data. For *SpectralBiclustering*, it marks the number of row and column clusters, making the total number of biclusters produced by the algorithm $n\_clusters^2$.

Taking the values from Table 7, the number of different biclustering analyses that were conducted with all possible parameter combinations is $10 \cdot 4 \cdot 3 \cdot 2 \cdot 2 \cdot 6 = 2880$.

Figure 31 shows the statistical results of the biclustering analysis. The plots show the mean of **average Spearman rho** (ASR) values calculated over all the dataset and parameter combinations that were used in the biclustering. ASR is a measure for evaluating the quality of a bicluster based on only the data and the biclustering results.

From all the plots in Figure 31, we can see that in all but a few cases the *SpectralBiclustering* algorithm gave better results than *SpectralCoclustering*, as for most parameter values, the average ASR score is higher for the former algorithm.

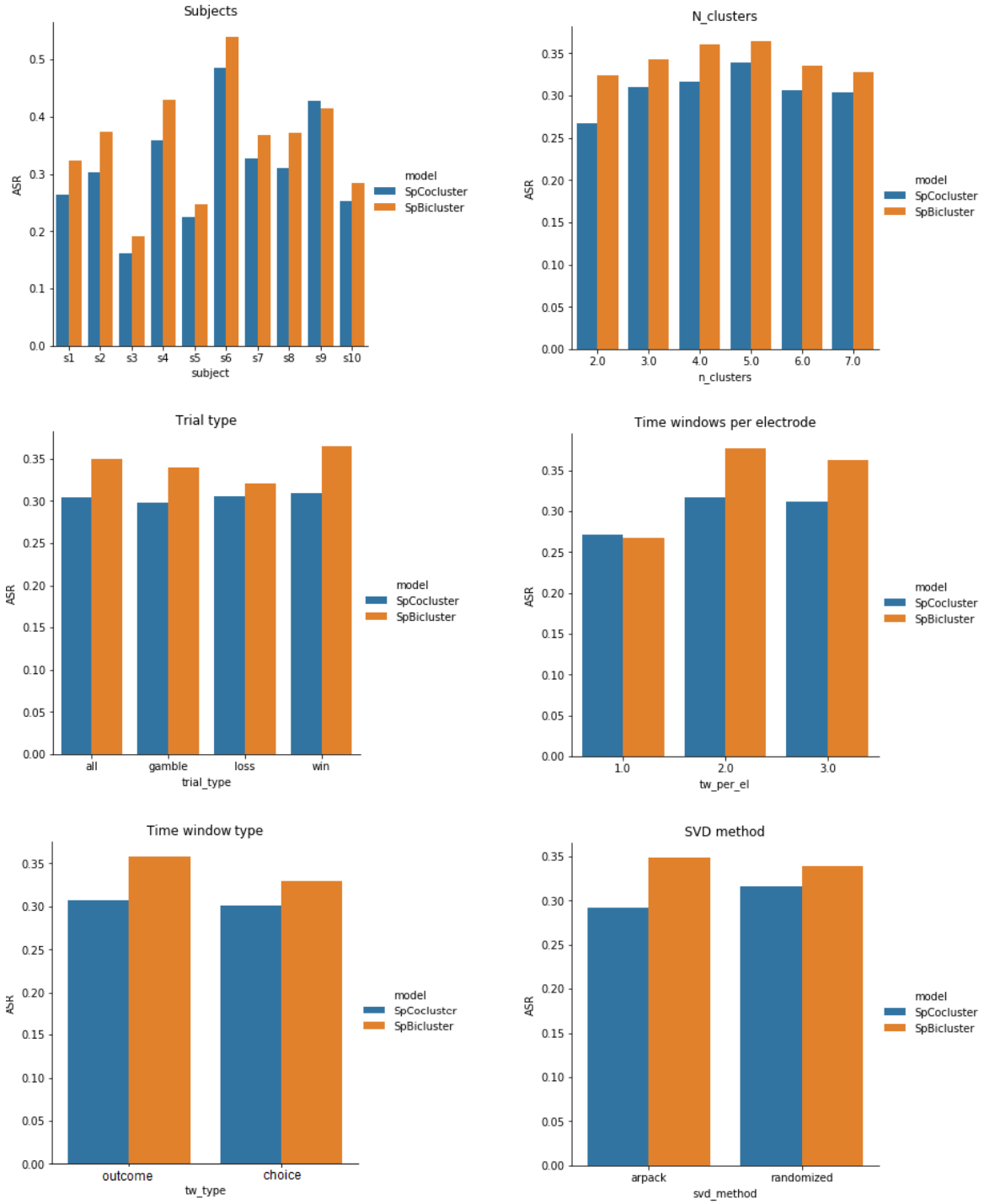

Figure 31. *The average ASR scores for different parameters. From top left to bottom right, the subject, n_clusters, trial type, time windows per electrode, time window type and SVD method.*

Looking specifically at each parameter, we can see that **subject 6** has significantly higher results from the others, while subject 3 has the lowest scores, with both algorithms giving an average result below 0.2. The poor results can be explained by the large amount of data

that subject 3 has: 59 electrodes and 194 trials. Increasing the number of time windows per electrode increases the number of features such that it is very close to the number of samples or even more, if taking a subset of the trials (win, loss or gamble trials). A large dataset with dense values is harder to bicluster, especially considering that brain activity data has a lot of noise.

The ASR scores for the *n_clusters* parameter show a similar tendency, inverted U-curve for both of the algorithms, **the best value being 5**. This number indicates the total number of biclusters for *SpectralCoclustering*, and the number of row and column clusters for *SpectralBiclustering*, making $5 \cdot 5 = 25$ biclusters altogether for the latter.

Time window and trial types, as well as the SVD method **do not show a particular difference** in the average ASR values for *SpectralCoclustering*, staying close to 0.3. However, using one time window per electrode gives noticeably worse results than using two or three, which is even more evident with *SpectralBiclustering.* This implies that dividing the two time window types into smaller units gives more valuable information. One possibility is that the smaller time windows average to more extreme values that are easier to cluster.

For *SpectralBiclustering*, it is interesting that the **win trials** appear to have the best results while using all trials has a slightly lower score. While the *outcome*-type time windows and SVD method *arpack* had higher results compared to the other options, the difference between the scores is not significant.

These are the biclustering results when averaging over all combinations that use these parameter values. We can also group results of some parameter value by another. Plot A on Figure 32 shows the **ASR average scores of time window types for different trial types**.

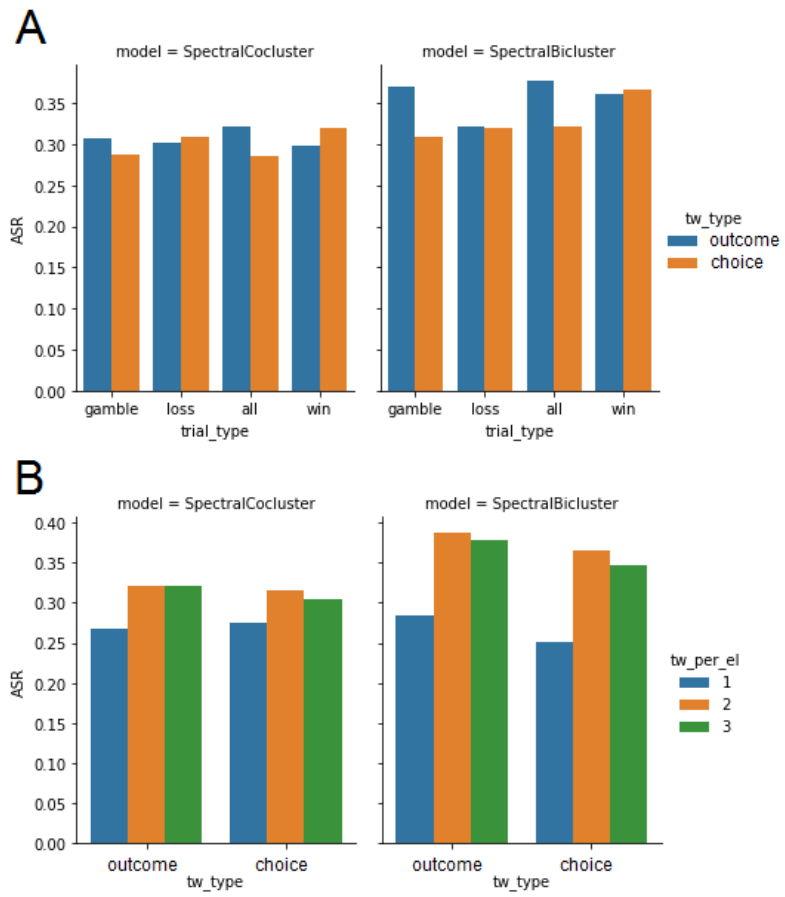

Figure 32. *Average ASR scores for different time window types (panel A) and time windows per electrode (panel B), when grouping the results by trial type and time window type respectively.*

(A) Trial type as category on the x-axis, with the y-axis reflecting the average ASR scores for different time window types. The left subplot is for the SpectralCoclustering algorithm, while the right plot is for SpectralBiclustering.

(B) Similar to plot A, but time window types are on the x-axis and the mean ASR scores for different number of time windows per electrode on the y-axis.

The results show that for both algorithms, clustering gamble trials which have *outcome*-type time windows gives higher results than clustering with *choice*-type time windows. The reason might be that an important difference between gamble trials is whether they are win or loss trials and as post-outcome time windows reflect this, it might be easier to cluster gamble trials with *outcome*-type time windows.

It is more difficult to reason the results of win and loss trials which more or less have equal ASR scores with both time window types, *choice* having only slightly higher results. On

one side, higher results for type *outcome* would imply a significant difference in post-outcome reaction that allows the algorithms to make clear subsets in the dataset. As trials have already been filtered according to the win indicator and loss indicator is always zero with win trials, this difference would have to come from other (unknown) factors. However, it is hard to reason why *choice*-type of time window would be helpful in biclustering win and loss trials, as there is no information on winning or losing in the deliberation period.

In addition to the average results we can look at particular combinations of parameters which yielded a good result and visualize the biclusters.

Table 8 shows the best results for the **spectral co-clustering algorithm**.

Table 8. *SpectralCoclustering best parameter combinations. The first five rows show the top 5 combinations, all of which are for subject 6. The last two rows show the only parameter combinations that did not have subject 6 and which resulted an ASR score larger than 0.7.*

Subject	**Tw type**	**Trials**	**Tw per el.**	***n_clusters***	**SVD**	**ASR**
S6	outcome	all	1	3	arpack	0.915
S6	choice	win	1	3	arpack	0.89
S6	choice	win	1	4	arpack	0.87
S6	outcome	all	2	2	randomized	0.841
S6	outcome	all	3	6	arpack	0.824
S4	choice	win	2	5	arpack	0.756
S9	choice	loss	1	2	randomized	0.717

Out of all the 31 parameter combinations which yielded an ASR score above 0.7 with *SpectralCoclustering*, **subject 6** was in 29. Two subjects which had a dataset that gained over 0.7 score after biclustering were **subjects 4 and 9**. The exact parameters of these combinations can be seen in Table 8.

Figure 33 shows the visualizations of biclustering the best combinations for subjects 6, 4, and 9, as shown in Table 8.

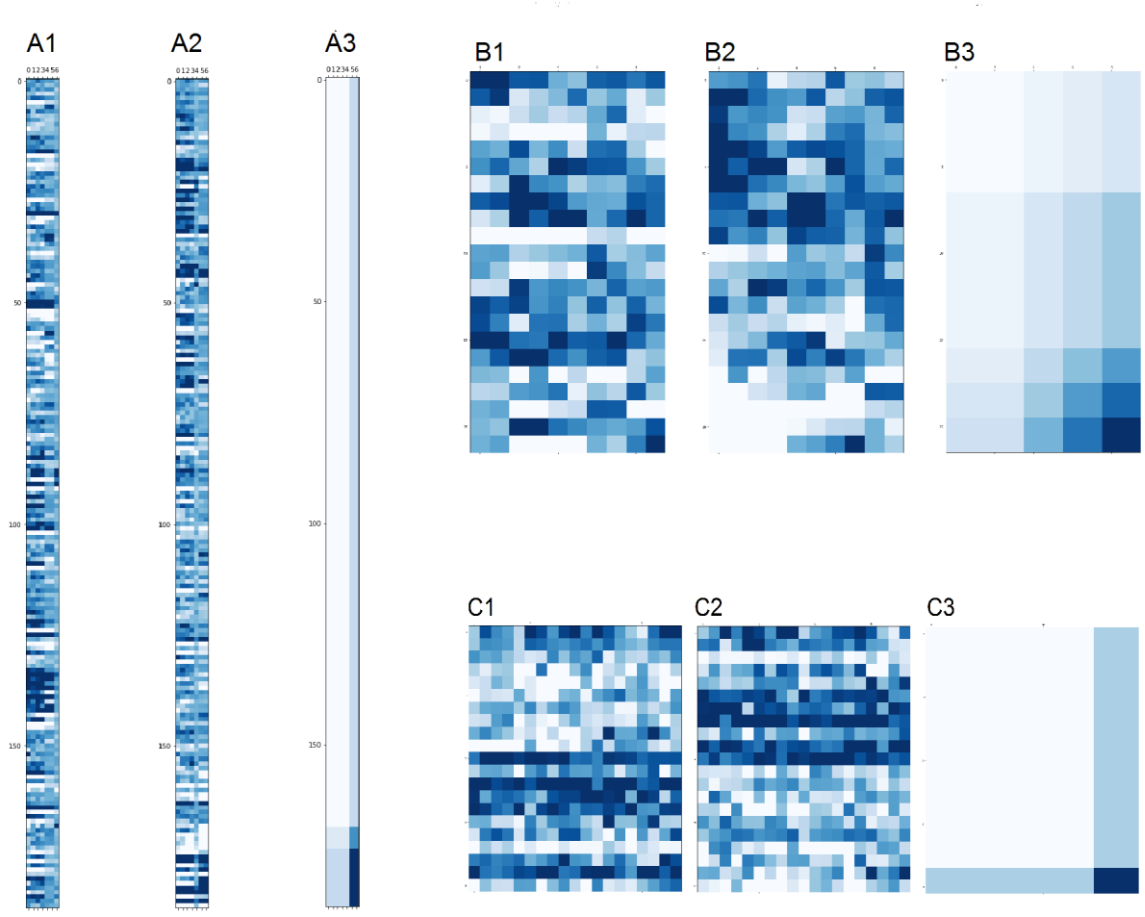

Figure 33. *The SpectralCoclustering results for top combinations of subjects 6 (plots A1, A2, A3), 4 (B1- B3) and 9 (C1-C3). The first plot of a trio shows the original dataset before biclustering, the second shows the transformed, reordered data after applying the Spectral-Coclustering algorithm. The third plot shows the outlines of the clusters, as they are positioned in the second plot.*

The biclustering algorithm should organize the data in a way that reveals some patterns in the rows and in the columns. If we closely look at the biclustering in plot A2, we notice that there are vertical "stripes" throughout the biclusters that are colored similarly. This means that similar HFA values are aligned through electrodes and trials, i.e. they are **strongly correlated**. Strong correlation in the rows and columns of the biclusters results in a high ASR score which indicates that electrodes have a very similar activity during most of the trials. This sets the basis for **further analysis** which involves analyzing the specific electrodes and trials in the biclusters with the aid of behavioral and regressor data.

The other clustering results depicted in Figure 33 are visually not so impressive, in particular the result in plot B2. There are a lot of biclusters (25 in total) and few (win) trials and electrode time windows (10 altogether) so the ASR was calculated on very small biclusters, which usually does not reveal significant patterns in the data.

The biclustered data in plot C2, Figure 33 is also peculiar as it visually shows a different structure than the one referred in C3. This might be due to the small cluster in the bottom right corner exhibiting a stronger pattern, which might be the reason why it was chosen.

Finally, we analyze the ***SpectralBiclustering*** **results**. Table 9 shows the best parameter combinations for the *SpectralBiclustering* algorithm.

Table 9. *SpectralBiclustering best parameter combinations. The first five rows show the top 5 combinations, all of which are for subject 6. The last two rows show two of the parameter combinations which did not have subject 6 and which resulted in ASR score higher than 0.7.*

Subject	**Tw type**	**Trials**	**Tw per el.**	***n_clusters***	**SVD**	**ASR**
S6	outcome	win	2	3	randomized	0.926
S6	outcome	win	2	3	arpack	0.926
S6	outcome	win	2	5	randomized	0.92
S6	outcome	win	2	5	arpack	0.92
S6	choice	win	1	2	randomized	0.908
S1	outcome	all	2	5	randomized	0.706
S4	choice	gamble	2	5	randomized	0.703

With *SpectralBiclustering,* 68 combinations resulted in an average ASR score higher than 0.7. As with *SpectralCoclustering,* **subject 6** was predominantly among these combinations, specifically in 64, while both **subject 1 and 4** managed to score over 0.7 twice. The best combinations of these two subjects are shown in Table 9.

While for *SpectralCoclustering*, the top combinations contained datasets with win and all trials, the *SpectralBiclustering* **top results only contain win trials**, with varying number of *n_clusters* and time windows per electrode. From the other subjects, subject 9 did not have a result above 0.7 with *SpectralBiclustering*, as opposed to the previous algorithm and is replaced by subject 1 in the list. While subject 4 appears in both top combinations, the dataset is different, with gamble trials for *SpectralBiclustering* and win trials for *SpectralCoclustering*.

Figure 34 shows the visualizations of biclustering with the *SpectralBiclustering* algorithm for the best combinations of subjects 6, 1, and 4 (as seen in Table 9).

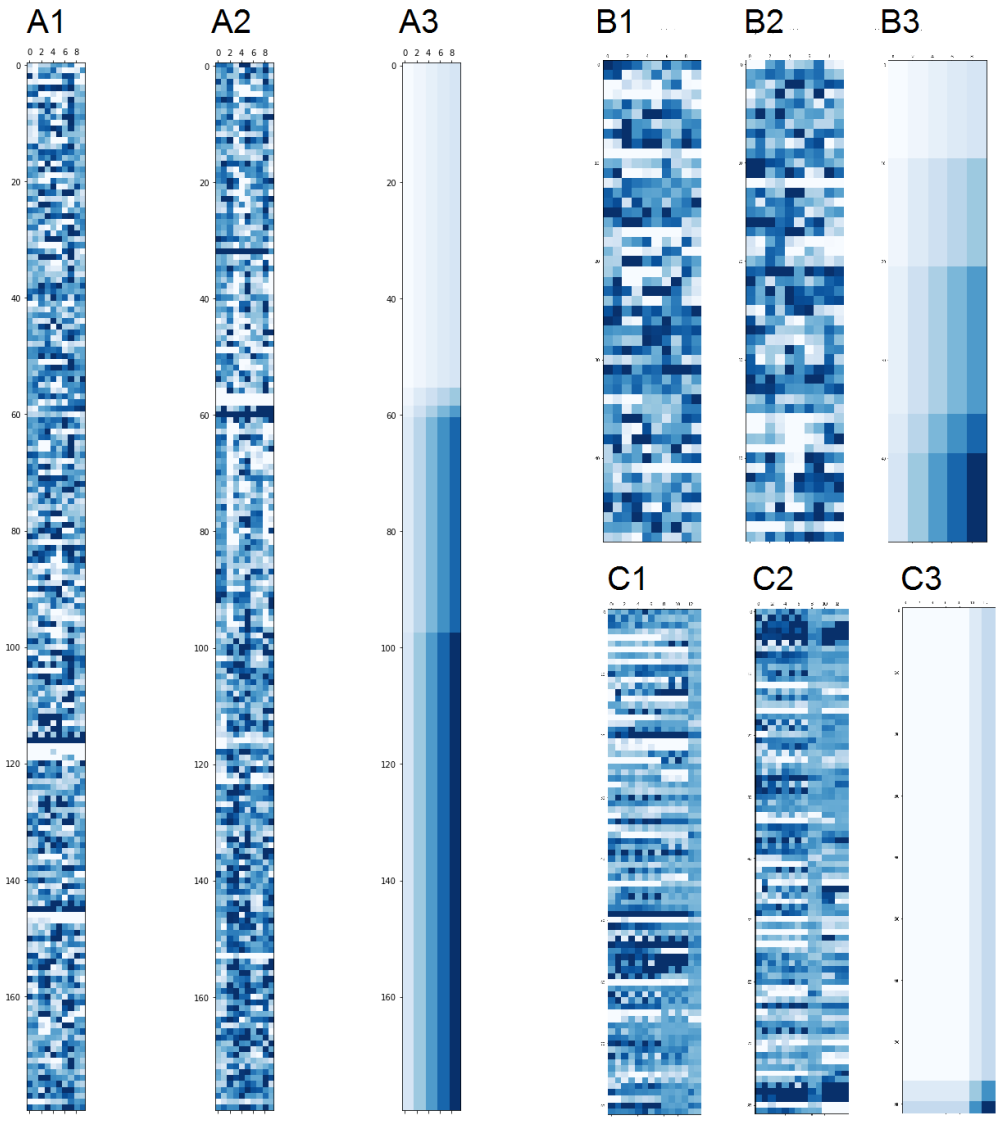

Figure 34. *The SpectralBiclustering results for top combinations of subjects 1 (A1- A3), 4 (B1-B3) and 6 (plots C1-C3), as given in Table 9.*

In Figure 34 we can see that while these specific combinations for subjects 1 and 4 resulted in high ASR scores, the visualizations do not reveal clear patterns. This is probably due to the number of biclusters – 25 in both cases – which divides the very few electrode time windows (10 columns altogether) into groups of two.

The plot C2 for the best combination (with subject 6) does show more correlation between the columns, indicating a similar response between the electrodes.

This concludes the biclustering analysis. Some of these results give solid basis for further analysis, especially for subject 6, but most of the subjects did not manage to achieve a high enough score to analyze.

Visualizations with t-SNE

This section describes the visualization results of the t-SNE algorithm. t-SNE was applied to HFA datasets of each subject (from dataset collection **D1**) to visualize trials and observe, **if similar type of trials form clusters**. t-SNE is a tool for visualizing high dimensional data. It converts objects in the original data space to a lower dimensional space, where they are grouped together (or apart) based on their similarities and differences to the other samples.

When visualizing data, it is useful to label the samples to see if the same type of objects are clustered together on the plot. This Thesis uses t-SNE to visualize trials by labelling them in two ways: (1) **gamble trials and safe bet trials**, (2) **win, loss and safe bet trials**. For the two cases, different types of datasets are selected. Namely, for option (1) the HFA dataset is with *choice*-type time windows, while for the second option, the datasets contained the average HFA values for *outcome*-type time windows. This is due to the fact that **gamble indicator** (which indicates whether a trial is a gamble or a safe bet trial) is a *choice*-related variable, while **win** and **loss indicators** are *outcome*-related.

Only datasets with **all of the trials** were selected from collection D1, while **the number of time windows per electrode was varied** (i.e. the tested values were 1, 2 and 3 time windows per electrode). Therefore, the number of samples in a selected dataset is *nTrials*, while the number of features for a dataset is $nElectrodes \cdot nTimeWindowsPerElectrode$.

Different *perplexity* values were tested with each dataset. Perplexity is an internal parameter for t-SNE that is used to estimate the number of close points for each data point [42]. It is complicated to tune correctly, as there is no universally good value [42]. Rather, it depends

on the concrete dataset. t-SNE results are generally hard to interpret as they might show some clusters that are actually random noise.

For *perplexity*, a wide range of different values from 2 to 100 (specifically, 2, 5, 10, 15, 20, 25, 30, 60, and 100) were tested. Thus, for each subject, **54 different types of analyses** were run with t-SNE: 3 different datasets (by *nTimeWindowsPerElectrode*) for cases (1) and (2), using 9 different perplexity values. Therefore, there were 540 t-SNE visualizations altogether.

All of the 540 visualizations were observed to see if any significant clusters are formed between the same types of trials. The general observation was that **in no particular case was t-SNE successful in clustering same types of trials together**. For most of the analyses, the trials were quite intermingled, with very small clusters of same types of trials forming that were deemed insignificant.

Firstly, the best number for *perplexity* was searched for. This was very hard to determine, as for every subject there was no *perplexity* value which achieved a result which did not appear as random noise. An example of all *perplexity* values plotted with one dataset for subject 8 is given in Figure 35.

We can see that with the smallest perplexity values the points are seemingly clustered together into small groups. While this could imply that 2 and 5 are good *perplexity* values to use with this dataset, this can also be very misleading as stated in [42] where it was demonstrated that t-SNE with small *perplexity* values can make the same kind of small clusters using randomly generated data.

These kinds of **structures** in the visualizations were **similar for different subjects**. While the size and spread of the whole set of data points varied across subjects, the inter-subject tendencies with using different number of *perplexity* values was similar. That is, many small clusters appeared with two smallest *perplexity* values (2 and 5), while for *perplexity* values 10 and larger there did not appear any significant differences in the structure of the data points.

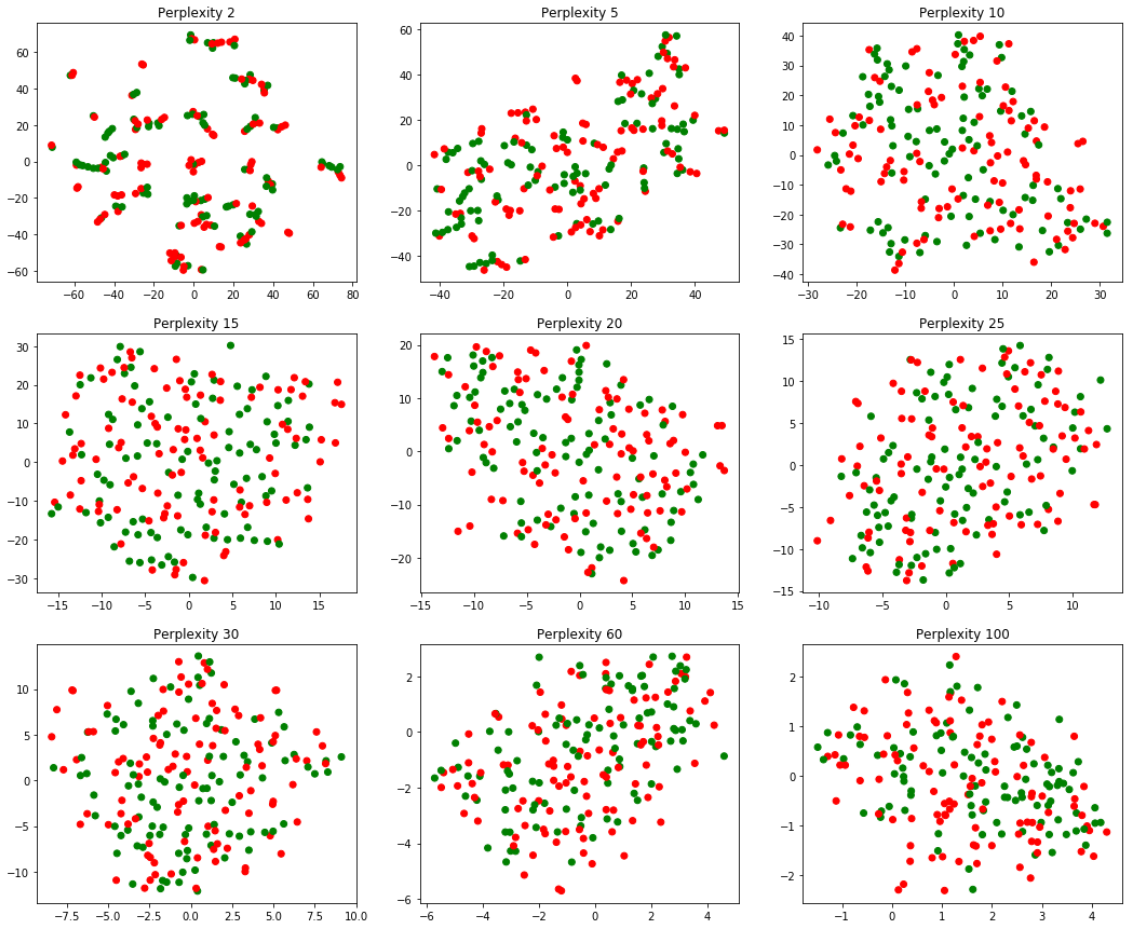

Figure 35. t-SNE visualizations of gamble (red) and safe bet (green) trials for subject 8 with different perplexity values.

For the rest of the analysis, the default parameter value 30 was kept as the *perplexity* value as it lies on the middle of the spectrum and there was not any particular *perplexity* value that was generally successful for every subject.

Figure 36 shows the t-SNE visualizations of **gamble and safe bet trials** for subjects **4** and **5**. On the upper row we can see the plots for subject 4 when using different number of time windows per electrode, and similarly for subject 5 on the bottom row. Each data point represents a trial, where red points correspond to gamble trials and green for safe bet trials.

These two subjects have datasets of vastly different magnitude: while subject 4 has 5 electrodes and 108 trials, then subject 5 has 61 electrodes and 194 trials (see Table 1). Moreover, while subject 4 gambled for 45% of the trials, subject 5 gambled during 73% (refer to Table 3).

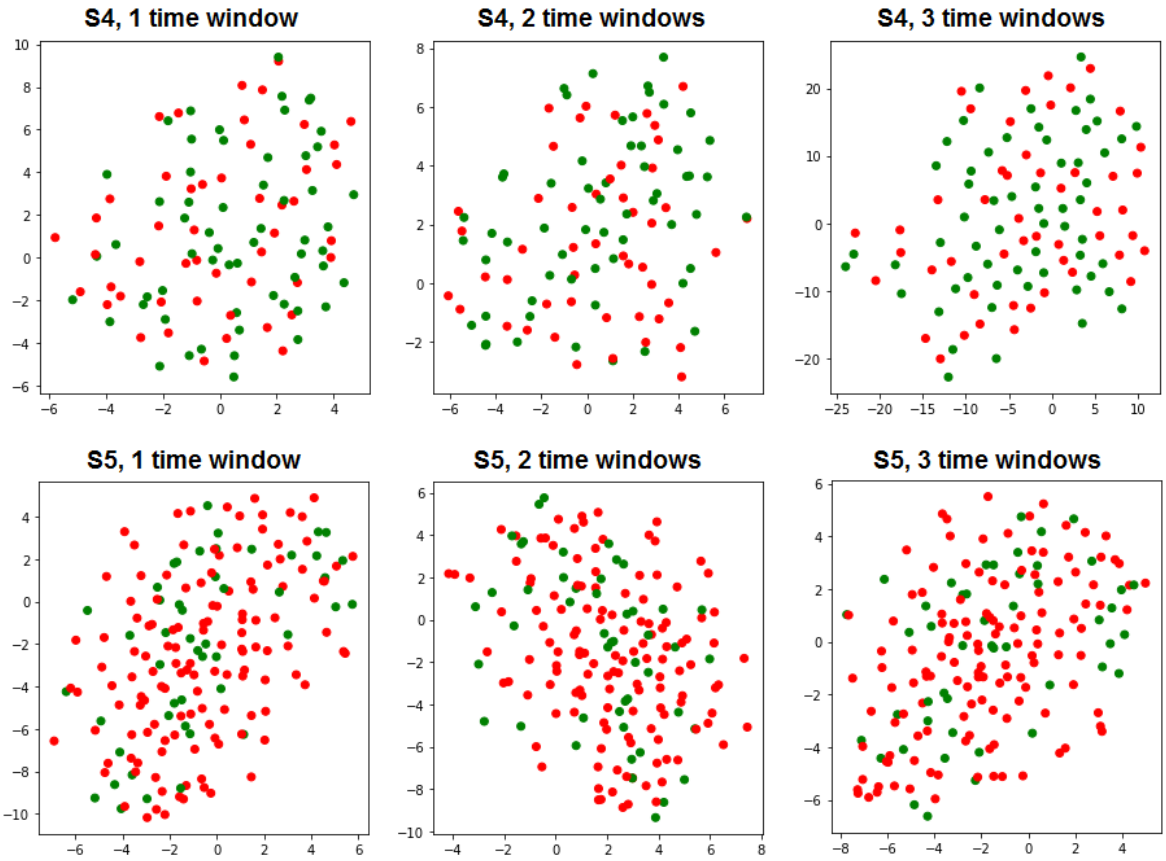

Figure 36. *t-SNE visualizations of gamble and safe bet trials for subjects 4 (the upper row) and 5 (the bottom row). From left to right, the visualized datasets are with 1, 2, and 3 time windows per electrode. The data points represent the trials in the dataset, where gamble trials are colored red and safe bet trials are colored green.*

We can see from the plots that neither subject's dataset exhibits any significant clusters of trials, with no particular changes from increasing the number of time windows. This tendency was similar across all subjects.

Figure 37 shows the t-SNE visualizations for **subjects 1 and 3**, but labelling the trials as ***win, loss*** **and** ***safe bet***. To compare, subject 1 has 5 electrodes and 180 trials, while subject 3 has 59 electrodes and 179 trials. Having almost equal amount of trials, but a vastly different number of features, both of these subjects do not show any noticeable clusters, with all types of trials very intermingled.

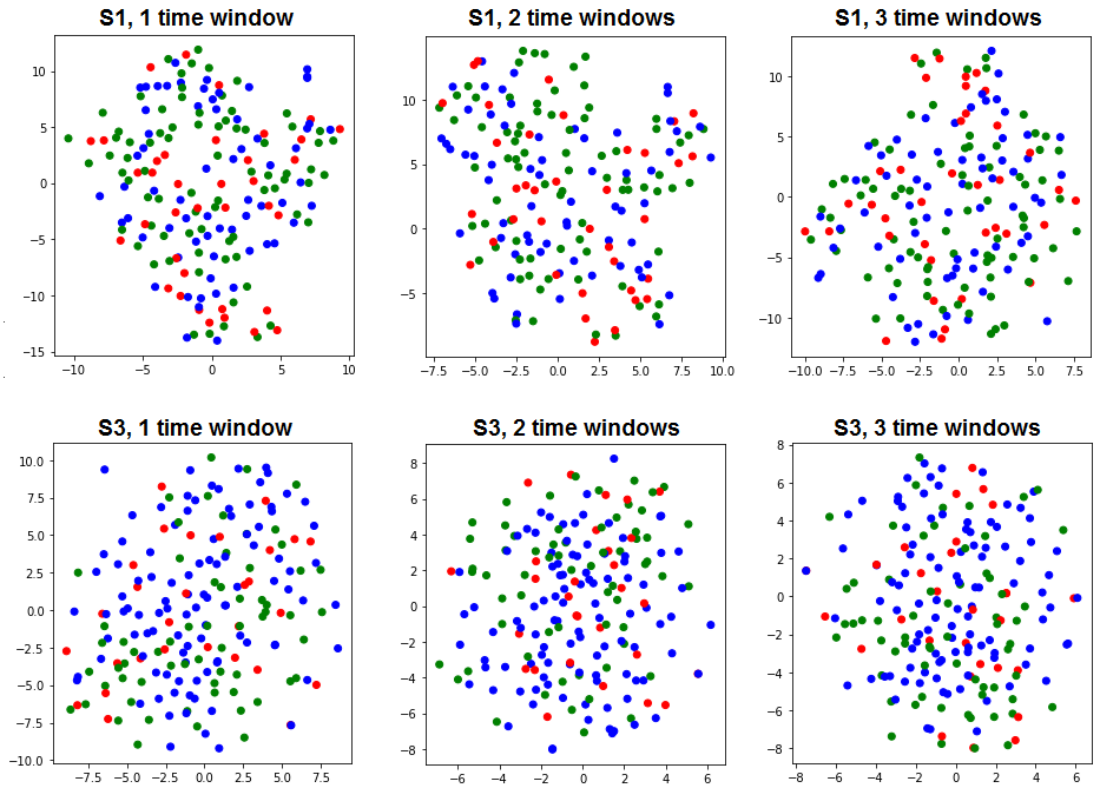

Figure 37. *t-SNE visualizations of win, loss and safe bet trials for subjects 1 (the upper row) and 3 (the bottom row). From left to right, the visualized datasets are with 1, 2, and 3 time windows per electrode. The data points represent the trials in the dataset, where win trials are colored green, loss trials red and all others (safe bet) blue.*

The only subject which showed any **significant difference** between the visualizations of datasets with a different number of time windows, was **subject 6**, as depicted in Figure 38.

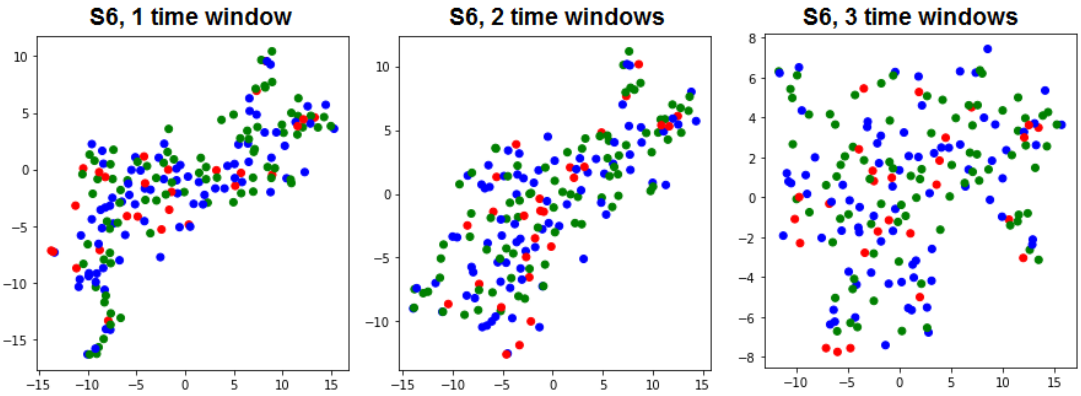

Figure 38. *t-SNE visualizations of win, loss and safe bet trials for subject 6. From left to right, the visualized datasets are with 1, 2, and 3 time windows per electrode. The data points represent the trials in the dataset, where win trials are colored green, loss trials red and all others (safe bet) blue.*

We can see a significant difference in the **spread of the data points (trials)**, when increasing the number of time windows. On the left plot, where the visualized dataset contains one time window per electrode, we can see some clustering at the bottom left corner. Overall, the data points are much more densely packed for subject 6 than for most subjects. Comparing the plot on the right with the one on the left, a significant change can be seen, as the points have become much more scattered. This might imply that increasing dimensionality makes the samples more difficult to compare and cluster with t-SNE.

In general, the t-SNE visualization **did not provide great insights into the similarities** between the trials' activity. One possibility is that they can be grouped by some specific time window which was not processed in this Thesis. Another thing to consider is the amount of noise in the neural data which makes any kind of comparison between samples, especially in high dimensions, very difficult.

5 Discussion

This section discusses the results in the light of the findings in [1], as the research also tackled the decision making in OFC, using the same dataset and several uni- and multivariate methods. In addition, the limitations and future work related to this Thesis are described.

5.1 Comparing Results

Decision making has been implied to be a process that is largely encoded in the human OFC [1]. Many fMRI studies have shown that brain signals in the OFC reflect multiple task-related values that are used to estimate the value of possible actions [1]. This Thesis expanded on the research in [1] by analyzing the behavior of the patients, using machine learning to predict task-related events, visualizing patterns in the data with biclustering algorithms and using canonical correlation analysis to discover evidence of a complex signal, which combines several variables involved in decision making.

The classification analysis confirmed the loss signal to be encoded in the *outcome*-type time window to such an extent that it can be predicted from the brain activity using linear methods. Predicting win and gamble indicator with logistic regression and linear support vector classification gave mostly mediocre results, but the conducted data exploration (Section 4.1) found several electrodes as in [1] that encode the win and gamble information.

Compared to the research in [1], an important addition in this Thesis is the usage of multivariate methods to analyze the signals and regressors. The regressor-electrode group analysis found evidence of a complex signal where a weighted combination of electrode time windows encode a weighted sum of several regressor values. These results varied by subject and regressor combination, with the *past* trial regressors having much more combinations with a high score than the *current* trial regressors. As stated in [1], this preserving of past information in a task where the trials are structurally independent might be counter-intuitive, but the brain may find it useful to keep the information "just in case" something in the situation state changes.

Biclustering was another method which added to the previous research by looking for visual and statistical evidence of electrodes behaving similarly for a subset of trials. This produced a vast number of results that were analyzed in this Thesis on a general level, finding subject 6 to have a lot of correlation between the electrodes.

5.2 Limitations

The main limitation related to this Thesis is the small amount of trials in the data. With only 10 subjects, for some whom nearly half of the trials were excluded from the analysis, the dataset contains a small number of trials. This restricts the behavioral analysis and CCA in particular as there is not enough trials to use in machine learning, derive behavior from or to find complex correlations. In addition, the number of electrodes was less than 10 for half of the subjects, which especially limits the performance of biclustering analysis since the number of features will limit the number of biclusters the algorithm can be tested with.

With the CCA results, we have to consider the possibility of good correlations occurring due to chance, as multiple comparisons with p-value significance were not calculated. This is certainly a task for future analysis to confirm these results. In addition, the CCA processes that used single regressor correlation to select electrodes from a subject have to be taken with caution, as it could have been that multiple regressors were encoded by the same electrode.

5.3 Future Work

Many analysis methods were used in this Thesis that produced a very large amount of results which can be analyzed even further using behavioral data to find patterns. The clustering analysis found that for some subjects, there exist patterns between the electrodes and trials. This is a solid basis for future work which would analyze the specific electrodes and trials in the biclusters to discover underlying reasons for the biclustering results.

In this Thesis, the classification analysis tested only a few non-linear methods (decision trees and random forest) due to time limitations, but in the future, more models could be explored with thorough hyperparameter search and tuning.

CCA also produced a wide variety of results which could be analyzed in more depth. One possibility is analyzing one parameter when grouping by another, e.g. which number of time windows worked best for which regressor individually. Some regressors might be encoded by the average of the entire time window, while for others dividing the time window in several parts gives more information about the signal.

In addition, the two dataset collections D1 and D2 which were created for this Thesis can be used to test more methods as they cover a wide variety of different parameters. For example, with CCA, the *current* and *past* regressors could be used in combination for the

regressor dataset (i.e. using different subsets of the *current* and *past outcome* (or *choice*) variables mixed together). This was decided against in this Thesis due to time limitations as it would produce more than 200 subsets of regressors to analyze.

6 Conclusion

The aim of this Thesis was to **find evidence of patterns and complex signals in the human OFC that are related to decision making**. Several machine learning, visualization and correlation methods were applied to a variety of neural datasets to expand on the results of the 2018 research by Saez et al. [1] titled "Encoding of Multiple Reward-Related Computations in Transient and Sustained High-Frequency Activity in Human OFC". The authors tackled the problem of decision making in the OFC and found robust evidence of 8 task-related variables or **regressors** being encoded in the **high-frequency activity** (HFA, 70-200Hz) of OFC [1]. As the authors used linear regression to find correlations between a single electrode and a regressor, this Thesis took a step further to analyze the data with **multivariate** methods.

The **four research questions** that were stated at the beginning of the Thesis are the following:

Q1. Is there correlation between groups of electrodes and behavioral regressors?
Q2. What are the behavioral tendencies of each subject in regards to risk-taking?
Q3. Can we predict winning, losing or gambling from HFA in the OFC?
Q4. Are there groups of electrodes that behave similarly for some sets of trials?

To answer these questions, the following methods were used:

- canonical correlation analysis (CCA) for Q1,
- systematic comparison of machine learning methods for Q2 and Q3,
- and biclustering for Q4.

Additionally, the t-SNE algorithm was used to visualize different types of datasets to test if similar types of trials are clustered together.

The Thesis used the OFC neural data and behavioral data from [1] to **build new collections of datasets** on which to apply multivariate analysis. In [1], the original data was collected from an experiment run of 200 game trials conducted with **10 patients**. During one trial, a subject was presented with a simple choice **to gamble or not** based on the risky prize amount and a random number which determined the probability of winning the gamble. Safe betting always resulted in a $10 prize, while risky prizes varied from $15 to $30.

Two dataset collections **D1** and **D2** were created in this Thesis during preprocessing. D1 was used in machine learning, biclustering and visualization, while D2 was used in CCA.

Q1 was tackled by CCA, which was applied on various types of HFA and regressor datasets to find the weights that would maximize the correlation between the weighted sums of HFA and regressor values. A strong correlation between the aggregated brain signals and regressor values could hint at evidence of a complex signal in the OFC which carries a combined value of multiple regressors. While most of the combinations resulted in a low test correlation score, some implications of such signals were found, with the **strongest evidence for *past* trial regressor combinations**. This would need to be confirmed in future work, using multiple comparison correction analysis on these results.

For **Q2,** this Thesis used two linear methods – **logistic regression** and **linear support vector classification** to predict whether the subject gambled or not based on two choice-related variables – presentation number and risky prize. The Thesis used the **feature coefficients** to analyze the effect of each variable on the person's decision, allowing to evaluate the greedy and pragmatic tendencies of the subjects' behavior. The **analysis revealed a wide spectrum of behavior among the subjects**, where some subjects prioritized the monetary aspect over the probabilistic, while others maintained a more cautious approach.

To answer **Q3**, the same linear methods were applied to neural data to predict whether the subject gambled during the trials, won or lost. The models were most successful with predicting the **loss indicator** for all subjects, while predicting winning or gambling resulted in mostly mediocre results, depending on the subject.

Finally, **Q4** was addressed by using **biclustering** on many types of neural data to find if there are patterns in the electrodes reactions in different trials. Two biclustering algorithms from scikit-learn were applied to all datasets in collection D1 and the results were evaluated by calculating the **Average Spearman's Rho** (ASR) for all clusters and averaging the values. For most subjects, no combination resulted in an average ASR value above 0.7. **Some subjects were successful**, including subjects 1, 4 and 9, but with very few combinations. The best ASR values occurred with many combinations of **subject 6**, reaching an ASR score above 0.9. This gives a basis for further analysis that involves trying to find the behavioral patterns in the trials that might explain these results.

In addition, the **t-SNE algorithm** was used to visualize different types of datasets for each subject and the results visually inspected, which did not reveal any significant clusters among the trials.

Analyzing the neural basis of decision making is important as it could give us more understanding of important processes in the brain and thus, could lead to better treatment and diagnosis of disorders that are related to the decision making process, for example obsessive compulsive disorder [4].

These results are a basis for future work, as the brain is an extremely complicated structure, conducting complex processes that have not been fully explained or understood. This Thesis leaves a wide variety of different datasets and results to be used in further analysis to tackle the problem of decision making in the OFC.

7 References

[1] I. Saez *et al.*, "Encoding of Multiple Reward-Related Computations in Transient and Sustained High-Frequency Activity in Human OFC," *Curr. Biol.*, vol. 28, no. 18, pp. 2889-2899.e3, 2018.

[2] Y. Li, G. Vanni-Mercier, J. Isnard, F. Mauguière, and J. C. Dreher, "The neural dynamics of reward value and risk coding in the human orbitofrontal cortex," *Brain*, vol. 139, no. 4, pp. 1295–1309, 2016.

[3] W. Schultz, P. Dayan, and P. R. Montague, "Neural Substrate of Prediction and Reward," *Adv. Sci.*, vol. 275, no. 5306, pp. 1593–1599, 2012.

[4] A. Rangel, C. Camerer, and P. R. Montague, "A framework for studying the neurobiology of value-based decision making," *Nat. Rev. Neurosci.*, vol. 9, no. 7, pp. 545–556, 2008.

[5] J. D. Wallis, "Orbitofrontal Cortex and Its Contribution to Decision-Making," *Annu. Rev. Neurosci.*, vol. 30, no. 1, pp. 31–56, 2007.

[6] M. O'Neill and W. Schultz, "Coding of reward risk by orbitofrontal neurons is mostly distinct from coding of reward value," *Neuron*, vol. 68, no. 4, pp. 789–800, 2010.

[7] "Electrocorticography - an overview | ScienceDirect Topics." [Online]. Available: https://www.sciencedirect.com/topics/neuroscience/electrocorticography. [Accessed: 10-Apr-2019].

[8] I. Saez *et al.*, "High-frequency activity of human orbitofrontal sites during decision-making play." CRCNS.org, 2018.

[9] "What are Brainwaves ? Types of Brain waves | EEG sensor and brain wave – UK." [Online]. Available: https://brainworksneurotherapy.com/what-are-brainwaves. [Accessed: 05-May-2019].

[10] S. X. Moffett, S. M. O'Malley, S. Man, D. Hong, and J. V. Martin, "Dynamics of high frequency brain activity," *Sci. Rep.*, vol. 7, no. 1, p. 15758, Dec. 2017.

[11] Society for Neuroscience, "Orbitofrontal Cortex." 2017.

[12] N. W. Schuck *et al.*, "Human Orbitofrontal Cortex Represents a Cognitive Map of

State Space," *Neuron*, vol. 91, no. 6, pp. 1402–1412, 2016.

[13] University of Rochester Medical Center Rochester, "Intractable Epilepsy - Conditions - For Patients - UR Neurosurgery - University of Rochester Medical Center." [Online]. Available: https://www.urmc.rochester.edu/neurosurgery/for-patients/conditions/intractable-epilepsy.aspx. [Accessed: 11-Apr-2019].

[14] "sklearn.tree.DecisionTreeClassifier — scikit-learn 0.21.0 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier. [Accessed: 14-May-2019].

[15] "3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn 0.21.0 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed: 14-May-2019].

[16] "sklearn.linear_model.SGDClassifier — scikit-learn 0.21.1 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn.linear_model.SGDClassifier. [Accessed: 15-May-2019].

[17] "sklearn.svm.SVC — scikit-learn 0.21.1 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC. [Accessed: 15-May-2019].

[18] J. Brownlee, "Logistic Regression for Machine Learning," 2016. [Online]. Available: https://machinelearningmastery.com/logistic-regression-for-machine-learning/. [Accessed: 17-Apr-2019].

[19] UCLA: Statistical Consulting Group, "FAQ: How do I interpret odds ratios in logistic regression?" [Online]. Available: https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faq-how-do-i-interpret-odds-ratios-in-logistic-regression/. [Accessed: 16-Apr-2019].

[20] T. Bock, "How to Interpret Logistic Regression Coefficients | Displayr." [Online]. Available: https://www.displayr.com/how-to-interpret-logistic-regression-coefficients/. [Accessed: 08-May-2019].

[21] “How to Identify the Most Important Predictor Variables in Regression Models,” *The Minitab Blog*, 2016. [Online]. Available: https://blog.minitab.com/blog/adventures-in-statistics-2/how-to-identify-the-most-important-predictor-variables-in-regression-models. [Accessed: 16-Apr-2019].

[22] “sklearn.linear_model.LogisticRegression — scikit-learn 0.20.3 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. [Accessed: 17-Apr-2019].

[23] P. Gupta, “Regularization in Machine Learning – Towards Data Science,” 2017. [Online]. Available: https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a. [Accessed: 29-Apr-2019].

[24] A. Bilogur, “L1 Norms versus L2 Norms | Kaggle,” 2018. [Online]. Available: https://www.kaggle.com/residentmario/l1-norms-versus-l2-norms. [Accessed: 10-May-2019].

[25] “1.1. Generalized Linear Models — scikit-learn 0.20.3 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression. [Accessed: 10-May-2019].

[26] “sklearn.svm.LinearSVC — scikit-learn 0.20.3 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC. [Accessed: 18-Apr-2019].

[27] StatSoft, “Support Vector Machines (SVM),” in *Electronic Statistics Textbook*, Tulsa, OK: Statsoft, 2013.

[28] “3.3. Model evaluation: quantifying the quality of predictions.” [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html. [Accessed: 10-May-2019].

[29] “An Efficient Algorithm for Computing the HHSVM and Its Generalizations - Scientific Figure on ResearchGate.” [Online]. Available: https://www.researchgate.net/figure/a-The-Huberized-hinge-loss-function-with-d-2-b-the-Huberized-hinge-loss-function_fig1_254295505. [Accessed: 10-May-2019].

[30] Statistics Solutions, “Conduct and Interpret a Canonical Correlation - Statistics

Solutions." [Online]. Available: https://www.statisticssolutions.com/canonical-correlation/. [Accessed: 22-Apr-2019].

[31] "Spearman's Rank-Order Correlation - A guide to when to use it, what it does and what the assumptions are." [Online]. Available: https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php. [Accessed: 11-May-2019].

[32] jramos, "Biclustering time series | Quantdare," 2018. [Online]. Available: https://quantdare.com/biclustering-time-series/. [Accessed: 25-Apr-2019].

[33] "2.4. Biclustering — scikit-learn 0.20.3 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/biclustering.html#spectral-coclustering. [Accessed: 25-Apr-2019].

[34] "A demo of the Spectral Co-Clustering algorithm — scikit-learn 0.20.3 documentation." [Online]. Available: https://scikit-learn.org/stable/auto_examples/bicluster/plot_spectral_coclustering.html#sphx-glr-auto-examples-bicluster-plot-spectral-coclustering-py. [Accessed: 26-Apr-2019].

[35] "A demo of the Spectral Biclustering algorithm — scikit-learn 0.20.3 documentation." [Online]. Available: https://scikit-learn.org/stable/auto_examples/bicluster/plot_spectral_biclustering.html#sphx-glr-auto-examples-bicluster-plot-spectral-biclustering-py. [Accessed: 26-Apr-2019].

[36] "sklearn.cluster.bicluster.SpectralBiclustering — scikit-learn 0.20.3 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.bicluster.SpectralBiclustering.html#sklearn.cluster.bicluster.SpectralBiclustering. [Accessed: 06-May-2019].

[37] C. Dhanjal, "Benchmarking the Singular Value Decomposition," 2016. [Online]. Available: https://simplyml.com/benchmarking-the-singular-value-decomposition/. [Accessed: 10-May-2019].

[38] W. Ayadi, M. Elloumi, and J.-K. Hao, "A biclustering algorithm based on a bicluster enumeration tree: application to DNA microarray data.," *BioData Min.*, vol. 2, p. 9, Dec. 2009.

[39] J. Dale, A. Nishimoto, and T. Obafemi-Ajayi, "Performance Evaluation and Enhancement of Biclustering Algorithms," no. Icpram, pp. 202–213, 2018.

[40] M. Pathak, “Introduction to t-SNE (article) - DataCamp,” 2018. [Online]. Available: https://www.datacamp.com/community/tutorials/introduction-t-sne. [Accessed: 29-Apr-2019].

[41] “sklearn.manifold.TSNE — scikit-learn 0.21.0 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html. [Accessed: 12-May-2019].

[42] M. Wattenberg, F. Viégas, and I. Johnson, “How to Use t-SNE Effectively,” *Distill*, vol. 1, no. 10, p. e2, Oct. 2016.

[43] “2.5. Decomposing signals in components (matrix factorization problems) — scikit-learn 0.21.0 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/decomposition.html. [Accessed: 12-May-2019].

[44] “sklearn.decomposition.TruncatedSVD — scikit-learn 0.21.0 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html. [Accessed: 12-May-2019].

[45] J. W. Schneider, “Two visualizations for explaining ‘variance explained’ | Assessing Psyche, Engaging Gauss, Seeking Sophia,” 2014. [Online]. Available: https://assessingpsyche.wordpress.com/2014/07/10/two-visualizations-for-explaining-variance-explained/. [Accessed: 03-May-2019].

[46] S. Ghoneim, “Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on?” [Online]. Available: https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124. [Accessed: 01-May-2019].

[47] Y. Benjamini and Y. Hochberg, “Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing,” *J. R. Stat. Soc.*, vol. 57, no. 1, pp. 289–300, 1995.

Appendix

I. License

Non-exclusive license to reproduce thesis and make thesis public

I, **Mari Liis Velner**,

1. herewith grant the University of Tartu a free permit (non-exclusive license) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

Analyzing Activity of the Human Brain During Decision Making,

supervised by Raul Vicente Zafra, PhD,

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons license CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive license does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Mari Liis Velner

16/05/2019