Towards an Affordable Brain Computer Interface for the Assessment of Programmers' Mental Workload

Makrina Viola Kosti , Kostas Georgiadis , Dimitrios A. Adamos , Nikos Laskaris , Diomidis Spinellis , Lefteris Angelis

PII: S1071-5819(18)30093-4
DOI: 10.1016/j.ijhcs.2018.03.002
Reference: YIJHC 2192

To appear in: *International Journal of Human-Computer Studies*

Received date: 19 March 2017
Revised date: 3 March 2018
Accepted date: 6 March 2018

Please cite this article as: Makrina Viola Kosti , Kostas Georgiadis , Dimitrios A. Adamos , Nikos Laskaris , Diomidis Spinellis , Lefteris Angelis , Towards an Affordable Brain Computer Interface for the Assessment of Programmers' Mental Workload, *International Journal of Human-Computer Studies* (2018), doi: 10.1016/j.ijhcs.2018.03.002

Highlights

- Understand and characterize programmers' mental effort using a low cost EEG device.
- Form biomarkers that reflect the mental workload using brain activation and functional connectivity patterns.
- Estimate the programmers' experienced difficulty during code comprehension.

Towards an Affordable Brain Computer Interface for the Assessment of Programmers' Mental Workload

Makrina Viola Kosti [a], Kostas Georgiadis [a], Dimitrios A. Adamos [b], Nikos Laskaris [a], Diomidis Spinellis [c], Lefteris Angelis [a, *]

[a] School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
[b] School of Music Studies, Aristotle University of Thessaloniki, Thessaloniki, Greece
[c] Department of Management Science and Technology, Athens University of Economics and Business, Greece

Abstract

This paper provides a proof of concept for the use of wearable technology, and specifically wearable Electroencephalography (EEG), in the field of *Empirical Software Engineering.* Particularly, we investigated the brain activity of Software Engineers (SEngs) while performing two distinct but related mental tasks: understanding and inspecting code for syntax errors. By comparing the emerging EEG patterns of activity and neural synchrony, we identified brain signatures that are specific to code comprehension. Moreover, using the programmer's rating about the difficulty of each code snippet shown, we identified neural correlates of subjective difficulty during code comprehension. Finally, we attempted to build a model of subjective difficulty based on the recorded brainwave patterns. The reported results show promise towards novel alternatives to programmers' training and education. Findings of this kind may eventually lead to various technical and methodological improvements in various aspects of software development like programming languages, building platforms for teams, and team working schemes.

Keywords: brainwaves; wearable EEG; neural synchrony; human factor; software engineering; neuroergonomics

* Corresponding author at: School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece. Tel.: +30 2310 998230.

E-mail addresses: mkosti@csd.auth.gr (M. V. Kosti), georgiaki@csd.auth.gr (K. Georgiadis), dadam@mus.auth.gr (D. A. Adamos), laskaris@aiia.csd.auth.gr (N.Laskaris), dds@aueb.gr(D. Spinellis), lef@csd.auth.gr (L. Angelis)

1 Introduction

According to the prolific software engineering researcher Robert Glass, "The most important factor in software work is not the tools and the techniques used by the programmers, but rather the quality of the programmers themselves" (Glass, 2002). In support to this statement, a significant number of studies advocate that a way to improve software developers' productivity and software quality is to focus on people (Boehm, 1988; Google Inc., 2014; Lee and Shneiderman, 2014; Sammet, 1983). Having this as common denominator, effort has been made to analyze from different perspectives the role of the human factor in software development. For instance, several empirical studies have emphasized the impact of the human factor in software engineering (Capretz et al., 2015; Kosti et al., 2016; Kosti et al., 2014). These studies use psychometric measurements in order to find connections between factors, such as personality, job attitude and performance on one side, and preferences or project outcomes or effects on the other side. Software developers employ a number of distinct cognitive processes when engaged in one of the various tasks of software development, such as coding, debugging and code comprehension. This has driven a number of researchers to employ cognitive neuroscience in order to better characterize and understand programmers' mental effort (Fritz et al., 2014; Nakagawa et al., 2014; Siegmund et al., 2014).

Registering brain activity, and subsequently decode it, appears to be a highly appealing procedure, since it opens the possibility to track the workings of the programmer's brain in action. The opportunities extend even further. Such a brain-centered approach may allow the empirical validation of theories regarding the cognitive processes associated with programming (Soloway and Ehrlich, 1984), may offer novel alternatives to programmers' training and education and can also lead to technical and methodological innovations in the field, such as improved programming languages, APIs, or development platforms. If researchers manage to measure and interpret brainwave patterns in terms of workload induced by the different software development activities, then, it would become feasible to detect the types of activities that cause particular stress or to compare alternatives for achieving a given goal in terms of brain workload. Overall, monitoring the mind of programmers could lead to novel or enhanced practices in SE. Taking advantage of recent technological advances in mobile EEG scanners, we attempted to characterize the electrical brain signals, recorded over the head and in an unobstructed way while programmers were engaged in some of their usual activities. The signals, known to form brainwaves, have been extensively studied for understanding cognition and were expected to directly reflect the underlying mental efforts.

The goal of this study is to demonstrate an inexpensive technology (mobile EEG scanners) as a way to monitor a programmer's mental effort. Hence neural correlates of programmers' workload were sought as a means to derive brain signatures indicating, in an objective manner, that they were experiencing some difficulty in performing the assigned task. If this exact sense of difficulty expressed by a programmer could be related to the brain signatures, then we could use this information to quantify the

effectiveness or usability of programming languages, APIs, and development tools. The particular task we tried to explore was program comprehension, which is an integral part in contemporary SE practice, i.e. in the context of code reuse. It requires various cognitive processes, such as working memory and attention. In accordance with a recent study (Siegmund et al., 2014) we contrast this task with the task of inspecting code for syntax errors, which is a simpler one but of similar nature and hence can provide us with a suitable baseline. More specifically, in our experiment (see Figures 1-2), the subjects performed tasks that came in pairs. The execution of a single task is referred to as a trial.

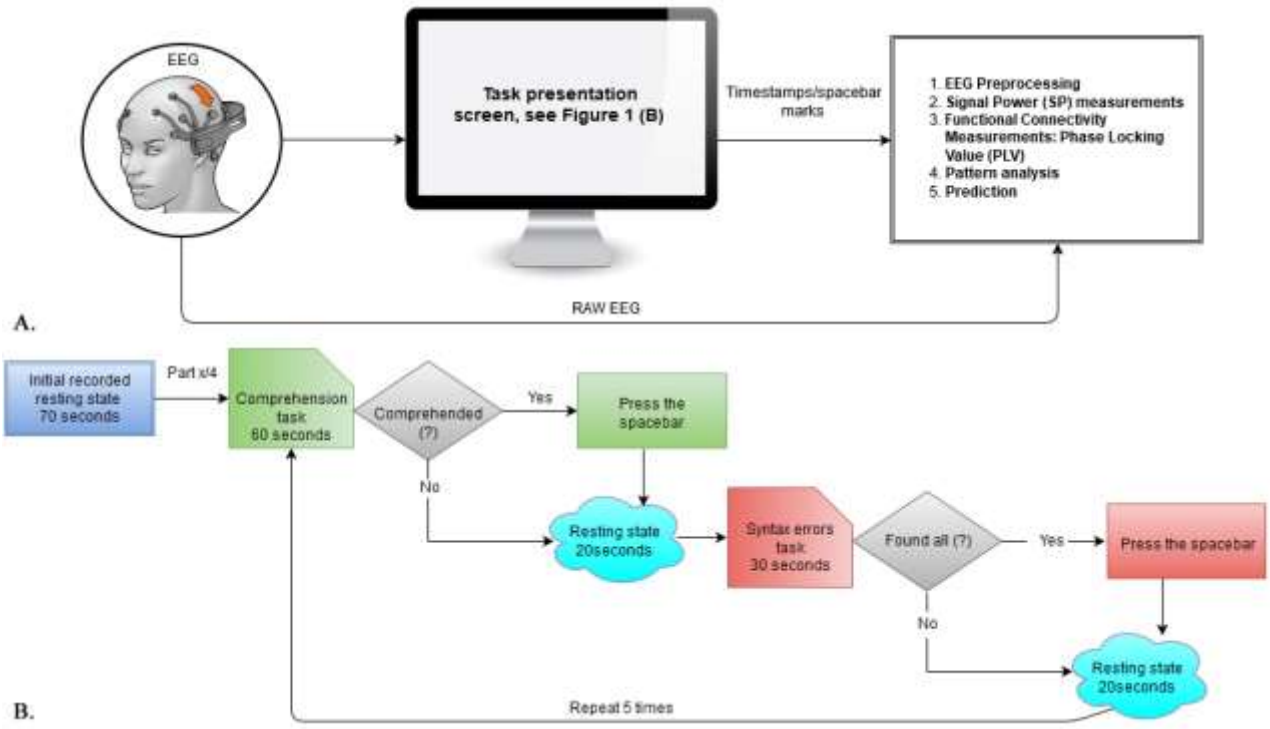
Regarding the choice of these cognitive processes, they are preferable than other tasks in the software development life cycle or testing, with the most critical reason being the inherent technical issues in EEG-recordings. A "good-quality" EEG signal can be obtained only when the subject avoids excessive movements (as we discuss in sections 3.3 and 5). The incorporation of the selected tasks leads to an experimental procedure, which is easier to be controlled as the programmer fixates on the screen for a certain amount of time. On the other hand, tasks as debugging or testing require physical involvement of the subject and uncontrolled movements. Other tasks are of great interest as well, but many steps remain to be taken and problems regarding the experimental conditions need to be solved before targeting such an ambitious goal.

During the first trial of the paired tasks, participants had to comprehend the presented code snippets (comprehension task), while during the second trial they had to detect the syntax errors injected in one of the code snippets (syntax task). In a separate session, each participant provided a description of the code snippets and rated them regarding the difficulty in understanding. The signal analysis pipeline included standard steps leading from the multichannel signals to the detection of activation and co-activation patterns (see Figure 3) that were then compared between tasks. Moreover, we proceed with a multivariate analysis, in order to model the relation between neural activity traits and task difficulty.

The contribution of this paper is threefold:

- It serves as a feasibility study about recording brain activity by means of a low cost, commercial EEG device and analyzing the brainwaves for the purpose of understanding and characterizing programmers' mental effort.
- It demonstrates that patterns of brain activations and functional connectivity can be used to form biomarkers that reflect the mental workload induced in a programmer.
- It introduces a method and a multivariate regression model that can be used to estimate the programmer's experienced difficulty during code comprehension.

The next section serves as an introduction to EEG, compares it with other neuroimaging techniques and discusses their use in the study of programmers' brain functionality. Section 2 discusses work having a similar direction to ours. Section 3 describes the experimental setup, the adopted signal-analytic methodology and the multivariate regression prediction model methodology. Section 4 is devoted to



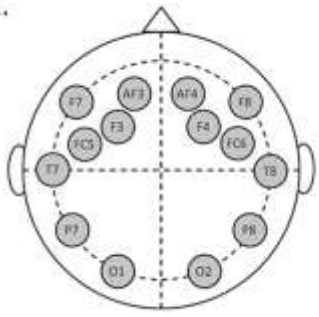
results. Section 5 discusses possible shortcomings of our approach. Section 6 includes a summary of this study and a discussion about future developments.

Figure 1. A) The employed Brain Computer Interface. B) The overall experimental procedure

Figure 2. A) Emotiv EPOC electrode positions. B) An example of a syntax task as used in the experiment.

C) A participant embedded in our BCI.

A.



B.

```
#include<stdio.h>
void main() {
   int num1 = 5;
   int num2 = 3;
   int num3 = 10;

   if (num1 > num2 & num1 < num3)
    printf("Result is %d", num1);
   else if (num2 > num1 & num2 < num3)
    printf("Result is %d", num2);
   else if (num3 > num1 & num3 < num2)
    printf("Result is %d, num3);
}
```

C.

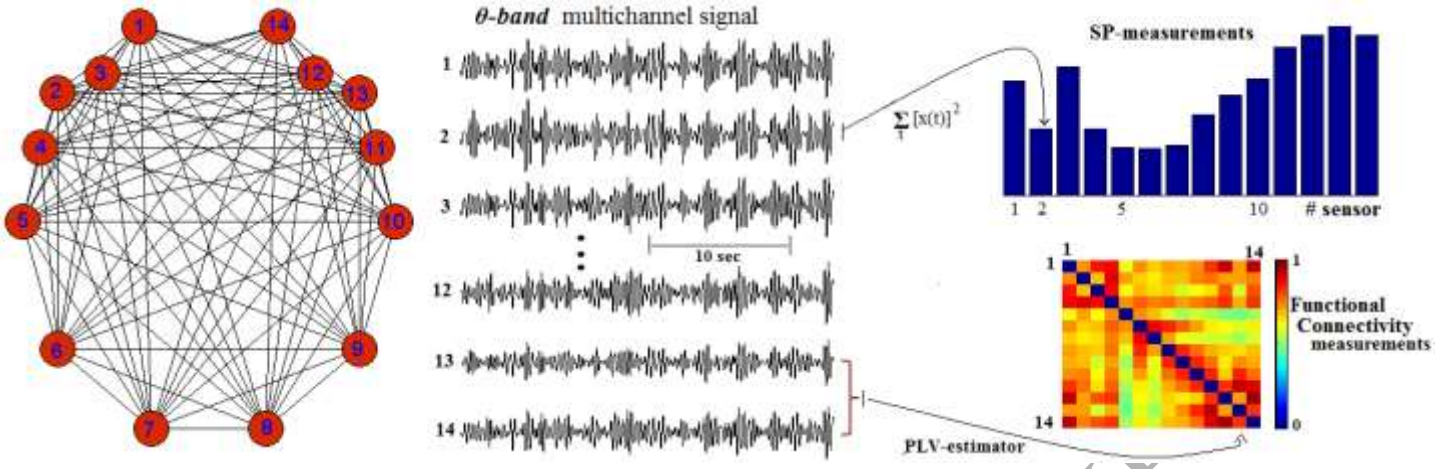


Figure 3. Exemplifying the signal-analytic procedure leading from the multichannel signal to the detection of activation and coactivation patterns based on a trial from a comprehension task. The wiring diagram (left) indicates the 14 sites at which the SP measurements are estimated and the 91 sensor pairs for which functional coupling estimates are derived. The traces (middle) correspond to θ-band brainwaves

Background

EEG and Cognitive Load

Since 1924, when studies of the human electroencephalography (EEG) began, research has mostly focused on clinical settings. Monitoring in epilepsy and sleep disorders constitute the most typical examples of the usage of this popular neuroimaging technique (Niedermeyer and da Silva, 2005).

Surface EEG is usually defined as the electrophysiological, non– invasive, measurement of (the net sum of) electrical activity of the brain. In a simple EEG setup, tiny electrodes are placed on the scalp, after having been moisturized with a conductive gel, and detect/measure electrical signals that are reflecting brain activity. Some systems though, use caps or nets into which the aforementioned electrodes are embedded. In any case, the basic principle of EEG instrumentation is the transduction of (a net flux of) ionic currents among neurons into voltage fluctuations (falling in the range of microvolts), which are measurable at the electrode positions over the cortex (Niedermeyer and da Silva, 2005).

The recorded signals present non-stationary oscillatory activity, which is usually characterized based on its spectral content (by means of Fourier transform) and its spatial distribution (reflecting instantaneous electrical fields). Even in the absence of any task, the EEG signals reveal a continuously active brain, with a background activity (the "ongoing" EEG) that is accompanied by remarkably reproducible dynamic patterns. Due to the inherent EEG dynamics, the recent cognitive studies focus on the task induced modulations of background brain activity. Conventionally the rhythmic activity ("brainwaves" or "brain rhythms") is studied independently within one of the 6 frequency bands, tagged with a small Greek letter, and isolated by means of band pass filtering. The following table (Table 1) includes the standard partition of brainwaves into frequency bands. It also accompanies them with a "typical" physiological meaning, which only provides a very crude approximation that hides the true complexity of the underlying neural processes

Table 1. Description of the standard EEG frequency bands

Band	Description
Delta **(δ)** [1-4] Hz	Brainwaves associated with unconsciousness and deep sleep.
Theta **(θ)** [4-8] Hz	State of somnolence and reduced consciousness.
Alpha **(α)** [8-12] Hz	Represents the state of physical and mental relaxation, although with awareness of what is happening around us. A bridge between our conscious thinking and subconscious mind.
Beta **(β_1)** [13-20] **(β_2)** [20-30]	High frequency low amplitude brainwaves. Observed when awake. Involved in conscious thought and logical thinking. They tend to have a stimulating affect. The right amount of beta waves allows us to focus and complete school or work-based tasks easily.
Gamma **(γ)**, [30-100]	Involved in higher processing tasks and cognitive functioning. Important for learning, memory and information processing.

EEG has been widely used to in order to analyze, evaluate and assess cognitive load (aka mental workload or simply workload) (Berka et al., 2007; Das et al., 2013; Ferreira et al., 2014; Kumar, N. and Kumar, J., 2016; Lee, 2014; Zarjam et al., 2011; Zarjam et al., 2010). Cognitive load relates to the load on working memory when performing a mental task (Paas et al., 2003). The human brain has limited processing capacity and endurance, which means that increased task difficulty will lead in reduced working memory accessibility and elevated cognitive load.

Direct measures of mental workload with the use of EEG have not only been proven to be possible (Coyne et al., 2009) but have also found practical application in many scientific areas such as adaptive training, visualization effectiveness, video game learning rates etc. (Coyne et al., 2009; Anderson et al., 2011; Mathewson et al., 2012). Zarjam et al. (2010) investigated the possibility of mental workload assessment during reading tasks. They suggested simple time-domain features of EEG signal that can reliably indicate the level of induced cognitive load. Such affirmation was also provided by later studies that employed spectral features of EEG (Ferreira et al., 2014; Das et al., 2013; Kumar, N. and Kumar, J., 2016; Zarjam et al., 2011). More recent studies have reported more advanced feature engineering techniques towards the efficient and effective measurement of cognitive load (e.g. Dimitriadis et al., 2015).

The main advantage of EEG, over the other two popular noninvasive neuroimaging techniques (fMRI: functional magnetic resonance imaging, and fNIRS: functional near infrared spectroscopy) is that it provides a direct measurement of the processes within the brain instead of an indirect measurement of either the blood flow or the metabolic activity. Additionally, EEG-based methods are found to be more suitable due to the high sensitivity EEG exhibits to cognitive states and task difficulty alternations (Antonenko et al., 2010). Moreover, it is associated with low cost and features high temporal resolution (Michel and Murray, 2012). Finally, it is becoming widely accessible as a part of the flow of wearable technology products (Das et al., 2013).

Related Studies on Program Comprehension

A number of studies have adopted techniques for monitoring brain activity in order to explore code understanding and program comprehension. A recent work by Siegmund et al. (2014) employed fMRI with the scope of gaining insights into brain activation during program comprehension. fMRI is a neuroimaging procedure which captures brain activity based on the hemodynamic response of the brain, that is the change in blood flow related to energy use by brain cells (oxygenation changes in the brain). Particular brain areas, known to be associated with specific cognitive processes were identified by comparing the Blood Oxygenation Level Dependent (BOLD) signals in comprehension and syntax tasks. Additionally, Floyd et al. (in press) performed a controlled experiment involving 29 participants using fMRI. They examined code comprehension, code review and prose review and concluded that

neural representations of programming languages are different from the representation of natural languages. Additionally, they provided evidence that this differentiation is modified with experience.

Nakagawa et al. (2014) employed fNIRS using a wearable device. fNIRS is also a noninvasive neuroimaging technique that localizes the hemodynamic response to a limited depth from the skull surface. The authors demonstrated high cerebral blood flow while understanding code and suggested the use of fNIRS during program development to measure mental workload. Ikutani and Uwano (2014) also used fNIRS and identified significant differences in brain activity, when the participants tried to understand code snippets that required variable memorization. The study concludes with the observation that the increased frontal pole activations reflected workload related with short term memory.

In the study of Kluthe (2014, Master's Thesis), EEG was used to explore the activations in alpha and theta bands with the goal of detecting different levels of expertise in undergraduate students. A later study of Crk et al. (2016) also used EEG to investigate the role of expertise in programming language comprehension. The study showed that the electrical brain activity was able to reflect prior programming experience and correlated with the self-reported experience levels. Finally, Lee et al. (2016), independently confirmed that the EEG activity can reflect expert ability in program comprehension.

Related Studies on Mental Workload

A recent study by Fritz et al. (2014) employed single channel EEG, among other physiological measurements, in an attempt to classify, in a binary way, the difficulty of code comprehension tasks. To achieve this goal, a Naïve Bayes classifier was trained to predict the difficulty of code comprehension tasks as reported by the participants. Müller also examined the possibility of predicting, by means of a classifier, whether a given code-comprehension task was perceived as easy or difficult using biometric data (Müller, 2015). A year later, Fritz and Müller (2016) trained a Naïve Bayes classifier to predict the perceived task difficulty combining biometric data from three sensors (eye tracker, EDA: electro-dermal activity and EEG). Finally, Lee et al. (2017) trained a Support Vector Machine classifier to predict task difficulty using both eye-tracker and EEG.

While our work shares with all the previous ones the goal of predicting the difficulty in a comprehension task, it differs with respect to the following methodological aspects: To begin with, we attempt to predict the experienced difficulty, and hence to quantify the involved mental workload using more than two discrete levels. To achieve this goal, we use ordinal regression, which is a discriminative model technique, instead of employing generative classifiers. We consider this approach more suitable due to the ordinal nature of the dependent variable. Finally, apart from activation patterns, we also analyze phase synchrony patterns as these are captured by means of a quite affordable, wireless and wearable EEG device.

Material and Methods

1.1 Outline-Motivation

Our study was built over the experimental design of an earlier work (Siegmund et al., 2014). However, we recorded EEG instead of fMRI activity (for the reasons discussed in section 2.1); mainly, due to the reduced inconvenience imposed by the measurement itself. As an example, fMRI requires subjects to be motionless, which is inconvenient and difficult to achieve. An important contribution of this work is the use of an affordable (around $800) consumer-grade EEG device (Emotiv EPOC[1]). A further advantage of this device is the fact that the recorded data are transmitted wirelessly, giving mobility tolerance to the participants during the experiment. This device was used in previous cognitive load measurement studies (Anderson et al., 2011). Such wearable devices may bridge the gap between clinical EEG and activities in a working environment and can also facilitate novel human-computer interactions for both practical purposes and scientific explorations. Regarding the feature-engineering step, apart from the conventionally used spectral features, we derived patterns of functional connectivity, which are considered to reflect additional aspects of the neural substrate supporting human cognition. Finally, instead of training a classifier, we tried to build a prediction model using ordinal regression. Our goal was to predict experienced difficulty, quantified in more than 2 discrete levels. Deviating from the relevant studies that employ generative classifiers, here we used ordinal regression, which is a discriminative model technique that takes into account the ordinal nature of the dependent variable.

Subjects

Ten volunteers participated in this study. They all had some kind of connection with the Aristotle University of Thessaloniki (AUTH), either as present or past students or as employees. Specifically, 8 of our subjects were males and 2 females. Their ages ranged from 25 to 37 years old. They all had experience with the C programming language, either from their studies or from their professional occupations.

Experimental Procedure

Previous to the experiment, participants were carefully instructed about the recording scheme and its requirements. Moreover, we provided them with a printed instruction pamphlet (tutorial), for which they were given time to read thoroughly. After reading the pamphlet and our oral instructions, the subjects were motivated to make questions in order to eliminate as much as possible any misunderstandings or miscomprehensions about the process. Finally, they gave us written consent in

[1] http://emotiv.com

order to be able to go on with the procedure. The main components of our procedure are depicted in Figure 1.

Before placing the headset and starting the recording, the subject sat comfortably in an armchair. We motivated them to try different seating positions in order to achieve the most comfortable one. Since EEG measurements are very sensitive to movements, which add noise to the recorded signals, we asked the participants to avoid as much as possible excessive body movements, head movements and facial expressions. When necessary, they were instructed to confine these activities during the resting periods between trials. Furthermore, during the recording session we carefully observed the participants and rated their facial movements and expressions in order to use that information later, during the data selection (or artifact elimination) step. We operated our experiments, or recordings, at different time slots of the day. More specifically we split the days into 3 time slots, namely: morning (8:00 - 12:00), noon (12:00 - 16:00) and afternoon (16:00 - 20:00). Our subjects were allowed to choose the time slot where they felt they would perform best.

Finally, at the end of the recording session, each participant was asked to answer a questionnaire. It is standard practice in EEG-recordings to not interrupt the flow of recording for various technical reasons (e.g., the battery exhaustion) and also to ensure that the brain activity data of interest are isolated from rest activations (e.g., those that would be invoked during an intermediate reporting of ratings). These were the reasons we chose to give the questionnaire to the subjects after the actual experiment.

This questionnaire specifically contained questions regarding the difficulty of each code snippet presented to the subjects. This means that every subject, after the experiment, had to rate from a [1-5] range all the code snippets presented during the comprehension task. The collected ratings were used (1) to correlate the subjective level of difficulty with the recorded activity and (2) to build a model that can directly translate the EEG-related measurements into mental workload levels. Apart from the individual ratings, the participants also provided a short description about the purpose and function of each snippet that was presented to them during the experiment. These answers were evaluated by an expert and used to investigate the correlation between the subjective difficulty experienced by the programmer during code comprehension and his or her final performance in interpreting the code.

Experimental Sessions and Tasks

In a series of pilot experiments (without the use of EEG scanner), we had tested various parameters of the experimental set up, such as the necessary time interval for completing a task, the minimal sufficient rest period between successive tasks and the independence between comprehension and

Session	Trial	Comprehension	Syntax
1	1	Factorial	Least common multiple

	2	Find max in a list of numbers	Reverse string
	3	Cross sum, sum of digits	Sum from 1 to n
	4	Prime test	Double entries of array
	5	Find middle number of three numbers	Median on sorted data
	6	Power	Factorial
	7	Swap	Reverse entries of array
2	8	Reverse string	Greatest common divisor
	9	Decimal to binary	Count same characters at same positions in String
	10	Reverse entries of array	Binary search
	11	Median on sorted data	Swap
	12	Count same characters at same positions in String	Decimal to binary
3	13	Sum from 1 to n	Bubble Sort
	14	Check palindrome	Prime test
	15	Double entries of array	Cross sum, sum of digits
	16	Greatest common divisor	Check palindrome
	17	Bubble Sort	Power
4	18	Binary search	Find max in a list of numbers
	19	Matrix multiplication	Find middle number of three numbers
	20	Least common multiple	Matrix multiplication

Table 2. Pairs of comprehension and syntax tasks, per session, per trial

syntax tasks. All parameters were finally set in a way to keep the whole recoding procedure as quick and simple as possible. Each individual experiment lasted no more than 60 minutes, with every session

lasting no longer than 11 minutes. All the recordings were carried out in the Department of Informatics at AUTH. Exceptional care was taken to conduct the experiments in a quiet room where no outside noise could distract the subjects.

During the experiment, the subjects seated in a typical working desk, as found in a typical working environment. Before the actual experiment, each participant performed a few pilot trials to familiarize with the experimental procedure. The tasks presented to the participants comprised 20 basic algorithms encoded in C, which are widely used in university classes, such as, least common multiple, matrix multiplication, bubble sort, etc..

In order to be able to create pairs of comprehension and syntax tasks we injected syntax errors into these 20 code snippets. Each subject had to perform either of the two tasks, denoted hereafter as comprehension and syntax, which came in pairs. The pairs presented to each subject, by session and by task, are shown in Table 2.

Therefore, our experimentation process was divided into 4 sessions (see Figure 1). Each session of the experiment comprised 5-paired trials, that is, 5 pairs of comprehension and syntax tasks. In a nutshell, the general scheme of each experimental round was as follows: (a) Comprehension task (60 seconds), (b) Period of rest (20 seconds), (c) Syntax task (30 seconds) and (d) Period of rest (20 seconds). Before each task, an appropriate sign was presented to the subject to reduce the possibility of confusing the type of upcoming task. The intervals of 60 seconds and 30 seconds for the comprehension and syntax task, respectively, had been set -somehow arbitrarily- to the typical (average) temporal duration necessary for a programmer to accomplish the employed tasks. This was a rough estimate derived from our independent preliminary experimentation with the recording protocol.

Recordings

EEG signals were recorded using the headset Emotiv EPOC, which includes 14 active electrodes, according to the 10-20 international system (with a spatial arrangement as shown in Figure 2A). The name 10-20 refers to the actual distances between adjacent electrodes. They either are 10% or 20% of the total front-back distance of the skull or 10% or 20% of the right-left one. In Figure 2A, each letter identifies the lobe locations and each number identifies the hemisphere location. The letters F, T, C, P and O stand for frontal lobe, temporal lobe, central lobe, parietal lobe and occipital lobe respectively. Even numbers refer to the electrode placement on the right hemisphere and the odd numbers on the other hand represent those placed on the left hemisphere. The signals were recorded with a sampling frequency of 128 Hz and filtered within [0.5-45] Hz. OpenSesame software (Mathôt et al., 2012) was used for automating the experiment setup. The synchronization procedure (event-triggering/marking) between OpenSesame and Emotiv's EEG recording software was implemented as in (Adamos et al., 2016).

There was an initial recording of subject's resting state for 70 seconds. Then the recording of the two cyclically repeated tasks was performed. Different images conveying the comprehension and syntax

tasks were randomly chosen from separate pools. The images for the comprehension task were presented for 60 seconds, while the images for the syntax task for 30 seconds. There was an interleaved period of rest lasting for 20 seconds, during which a counter was indicating the time left before the next task and the type of the task. During that period the subject was free to blink, shallow, etc. An example of a syntax task is depicted in Figure 2B. When a subject had the feeling that he or she had already comprehended the presented snippet or found all the injected errors, he/she should press the spacebar.

In Figure 2C we present the setup of the experiment with the subject in place, in front of the screen and with the headset attached.

Data Analysis

EEG Preprocessing: During offline processing the continuously recorded signals, from each recording session, were segmented into trials based on the timestamps associated with the triggers. This step resulted into 1 trial of resting state activity, 20 trials of brain activity during the comprehension task and 20 trials of brain activity during the syntax task for each subject. Using an automated procedure (Laskaris et al., 1997), trials contaminated by artifacts (e.g. eye movement) were detected and excluded from further analysis. The standard EEG frequency bands were defined as shown in Table 1). Band-limited brain activity was derived by applying a third-order Butterworth filter (Temes and LaPatra, 1977) (in zero-phase mode). With this step, the brain activity associated with each of the 6 distinct brains rhythms was treated independently. For each brain rhythm, two distinct brainwave patterns were derived from the multichannel signal, described as follows.

Signal Power (SP) measurements: At the first stage of analysis we measured, for each sensor separately, the total signal energy residing within each frequency band during every trial (see Figure 3). By taking into account the duration of each trial, we transformed the signal energy measurements to power estimates. The SP measurements were averaged across trials so as to compare the activation between syntax and comprehension tasks.

Results, expressed as relative differences in the form

$$\frac{SP_A - SP_B}{SP_B}, \quad \text{A: "comprehension" and B: "syntax"} \qquad (1)$$

were first computed for each subject independently and then averaged across subjects. The SP measurements within a particular frequency band (or brain rhythm), were additionally treated as activation patterns, which reflected globally the neural activity during a trial. These were 14 dimensional (14D) vectors, with each attribute corresponding to the SP at a specific sensor (as indexed in Figure 3).

Functional Connectivity measurements: At the second stage of analysis, we encountered phase synchrony measurements as an advanced way to characterize the cognitive processes that underlie the performance to the delivered tasks. Phase synchrony is known to reflect the coordination among

distinct neural subsystems that is necessary for performing various cognitive tasks. Its role in cognition is well established and various measures have been introduced so as to quantify its presence. This kind of estimators operates on the multichannel signals (most often by searching for statistical (functional) dependencies among sensors in pairwise fashion) and detects specific patterns of functional connectivity associated with the execution of a particular task. In the framework of this study, phase synchrony was examined from the perspective of describing the brain network(s) during code comprehension and while debugging code.

For each brain rhythm, the signals from the 14 recording sites were used to estimate a connectivity pattern that reflected the neural synchrony among brain areas and characterized the subject's cognitive performance for a given set of trials (i.e. from syntax task, comprehension task) (Dimitriadis et al., 2012).

Phase Locking Value (PLV): To detect phase synchrony based on the bandpass filtered signals from two recording sites, we adopted the PLV estimator (Lachaux et al., 2000). PLV ranges between 0 and 1, and quantifies the phase interrelations between rhythmic activities registered at different sensors. It is applied in pairwise fashion and has proven efficient for sketching the connectivity pattern of the underlying neural network(s). In our case it was applied to all pairs formed among the 14 sensors. Since it is by definition a symmetric measure (i.e. $PLV(sensor_1,sensor_2)= PLV(sensor_2,sensor_1)$), its application resulted to connectivity patterns residing in a 91-dimensional space ($14 \times 13/2=91$; see Figure 3). The pattern formation step was repeated for each trial independently and the trial dependent patterns were stored for each task separately. In this way, subject–specific groups of functional connectivity patterns were formed for each condition (syntax/comprehension) and for each brain rhythm. These groups of patterns were contrasted in pairwise manner as described below.

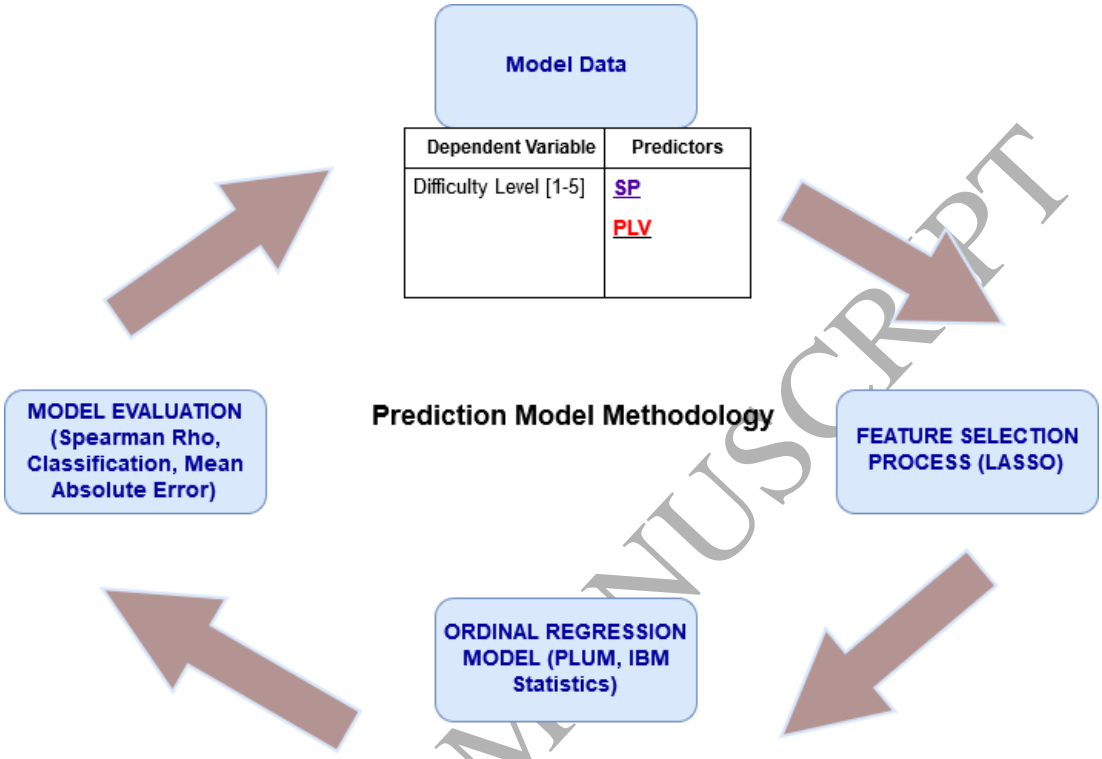
Pattern Analysis: We derived a discriminability score for the attributes included in both types of utilized brainwave patterns, i.e. the activation patterns stemmed from the SP-measurement and the connectivity patterns emerged from the PLV-measurements. To this end we employed the feature ordering scheme implemented in MATLAB (MATLAB, 2013), i.e., the *rankfeatures*[2] command using the "Wilcoxon" criterion. To facilitate the comparison between the 6 brain rhythms, we presented the results in direct contrast. In order to gain some insights into the neural mechanisms, the results were also presented topographically.

From brainwave patterns to workload estimates: At this point we will present and explain the methodology (see Figure 4) followed in order to build a model that can predict the difficulty ratings provided by our subjects. These are based on the EEG derived measurements, which reflect the workload of a programmer trying to comprehend code.

In a nutshell, the steps shown in Figure 4 are as follows:

[2] http://www.mathworks.com/help/bioinfo/ref/rankfeatures.html

- Step 1. Use LASSO (as implemented in Matlab) to derive a subset of possible predictors.
- Step 2. Use PLUM (Ordinal Regression procedure in IBM Statistics) with the subset given from LASSO.



- Step 3. Check predictors' statistical significance.
- Step 4. Exclude one by one the less significant predictors from the model until a trade-off is achieved between the significance of the predictors and the final predictive power of the model. (Model evaluation)
- Step 5. Repeat 1, 2, 3 and 4 until the optimal model is achieved.

Figure 4. Model construction and improvement cycle

The dependent variable of the model, derived by the answers given by the subjects and related to the difficulty of each comprehension task, is of ordinal nature and has values from 1 to 5, with 1 meaning "Very Easy" and 5 "Very Difficult". The collected ratings are shown in Figure 10.a and 10.b and explained in Section 4.3. This was the reason of opting for Ordinal Regression (OR) to model the desired relation.

The OR method is a generalization of the Linear Regression (LR) method. The OR method is used to model the relationship between an ordinal (dependent variable) and a set of predictors variables, either

categorical or continuous. The categorical values of an ordinal variable are represented by sequential integers, with the lowest one representing the first category and so on. The procedure builds a separate equation for each category and each equation provides us with a predicted probability for each category. The cumulative probability of the first category is always 1, therefore there is no need for a prediction equation for the last category. The set of prediction equations of the OR technique are of the form:

$$link(\gamma_j) = \theta_j - \sum_{i=1}^{k} \beta_i x_i, \quad (2)$$

where γ_j is the cumulative probability for the j^{th} category, θ_j is the threshold for the j^{th} category, $\beta_1 ... \beta_k$ are the regression coefficients, $x_1 ... x_k$ are the predictor variables and k is the number of predictors. The left side of Equation 2 represents the *link function* of the model. The link function is the function of the probabilities that results in a linear model in the parameters. It defines what goes on the left side of the equation and it is the link between the random component on the left side of the equation and the systematic component on the right. The component on the right side determines the location of the model (different from the lobe location described in other parts of the paper). In our study we used the Logit function, $\ln\left(\frac{\gamma}{1-\gamma}\right)$, as a link one (Harrell, 2015).

More precisely, we used IBM's statistics PLUM, which is the equivalent procedure for running OR. This procedure makes use of a general class of models and can analyze the relations between a polytomous ordinal dependent variable and a set of predictors (IBM Corp., 2013).

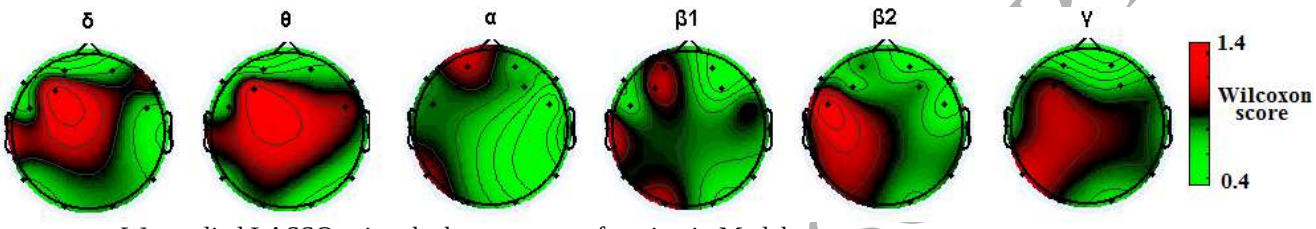
Experimentations were carried out using the SP and PLV measurements described above. Table 3 shows the number of predictors per each type of EEG measurements. In the former case (SP: ''brain activation''), as it can be seen in Table 3, 84 independent variables were used. Every one of them related to the signal power measurement at every brain sensor (out of 14) and for every frequency band (6 in total). In the latter case (PLV: ''functional connectivity''), 91 measurements per frequency band were

Table 3. Number of predictors per used dataset

Measurements used	No. of predictors
SP	84
PLV	546

used (a number resulting by the possible pairwise combinations of the 14 sensors attached to the subjects' head. Since ordinal regression itself does not provide an automatic predictor selection method, e.g., similar to stepwise regression, we had to choose a method that would operate separately

from ordinal regression so as to significantly decrease the number of possible predictors. To serve this scope, we exploited LASSO (Tibshirani, 1996). The LASSO step precedes that of PLUM application and that of the exclusion of those PLUM predictors that had significance levels under the adopted baseline of $p<0$. LASSO is a regularization technique that, in addition to minimizing the sum of the squared errors accomplished by a SLR (Standard Linear Regression), also introduces an additional term to the minimization problem. Namely, instead of solely minimizing the sum of squared errors, it also minimizes the sum of the absolute value of the regression coefficients, which are multiplied by a weight parameter, known as λ (lambda). Parameter λ can take values from zero to one. As the LASSO procedure progresses, the λ parameter increases in size while the regression coefficients shrink towards zero. We applied LASSO using the homonymous function in Matlab.

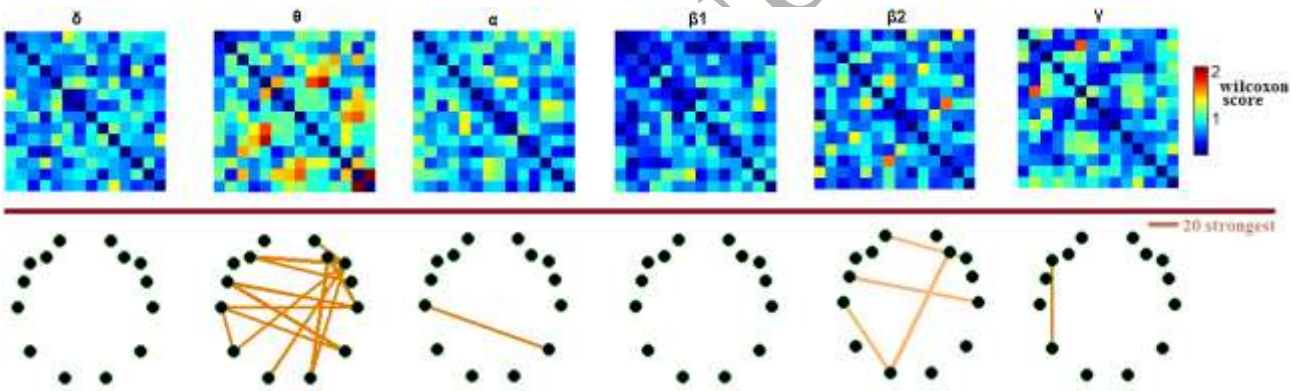


Finally, we evaluated our models comparing the actual difficulty ratings to the predicted ones primarily by using the Spearman correlation and, additionally, Mean Absolute Error (MAE). We believe that

Spearman rho provides a more appropriate prediction power evaluation metric because of the nature of our dependent variable. As our subjects used ordered ranks to express the difficulty of the comprehension process, our aim is to check if our model continues to respect that ranking. Moreover, we used cross tabulation to check the relationship between the actual and predicted difficulties along with a Chi-Square Statistic. More specifically, by applying cross tabulation, we actually assess the percentages of the predicted difficulties that were classified in the correct difficulty category. To make the notion of this evaluation metric simpler, it can be paralleled with the adjusted R squared of the SLR (Simple Linear Regression). In Table 3 this evaluation measure, namely the classification percentage, is presented in the third column of the table under the name *Class*..

Figure 5. Topographical representation of Wilcoxon score contrasting the SP measurements between comprehension and syntax task.

Figure 6. Top: Tabular representation of Wilcoxon score contrasting the phase synchrony between comprehension and syntax task. Bottom: Topographical arrangement of the top 20 links.



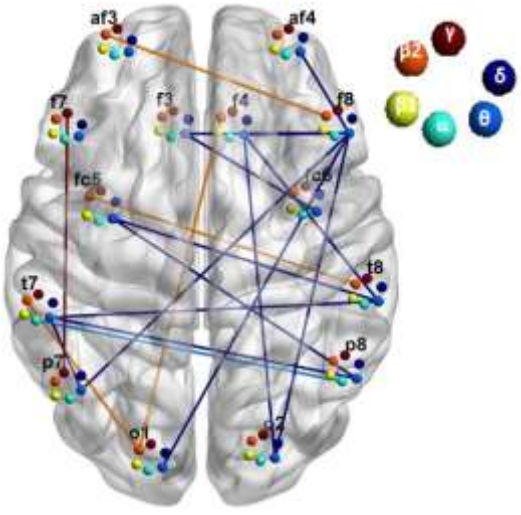


Figure 7. Comprehension vs. Syntax task: Topographical representation of the 20 most discriminative pairwise phase interactions

2 Results

Comprehension vs. Syntax Task

The comparison of SP measurements indicated that Comprehension is in general a more demanding task. Brain activity is heightened over frontal areas, mainly in the β2 band. Figure 5 (see above) includes the topographical representation of the discriminability score, which reflects the existence of consistent statistical differences between trials of Comprehension and Syntax task (the higher the score the more different the level of activation between the two tasks).

The comparison between the synchrony patterns leads to an even more clear distinction between the Comprehension and Syntax tasks. Based on the connectivity patterns (91D vectors or equivalently [14x14] matrices) corresponding to the trials of either task, we derived a [14x14] matrix tabulating the discriminability scores for all sensor-pairs (for each subject independently). The top row of Figure 6 depicts the obtained matrices, after across-subjects averaging, for all 6 frequency bands. It is important to notice that the maximum entry reaches the value 2.4 for the Wilcoxon score, which is significantly higher than the maximum reached through the SP measurements (shown in Figure 5).

To bring these results in a more neuroscientific context, we identified the 20 strongest entries (across all frequency bands) and drew them as links between the corresponding sensors over the scalp. The bottom row of Figure 6 clearly indicates that the aspects of phase synchrony that distinguish mostly the processes of Comprehension and Syntax include interhemispheric interactions within θ and β2 bands.

Finally, to further facilitate insightful observations about the functional organization of the brain, we adopted a novel topographical representation that was based on the idea to "bring all rhythms" in a common topography and exploited BrainNet Viewer[3]. Figure 7 includes all the selected links, color-coded according to the band in which they were formed. Interestingly, between the 2 concurrent pairs, there is a pair of θ and β2 links corresponding to interhemispheric interactions. Considering that the activity in θ-band is often interpreted as a signature of working memory (Jensen and Tesche, 2002) and the activity of β-band is usually associated with concentration and increased mental effort, it becomes clear that there is a difference between the level of efforts required for the comprehension and syntax task. This difference is clearer in the pattern of neural synchrony than in the pattern of neural activations.

By using PLV measurements we were able to extract richer information regarding the activity of the brain when a SEng performs these two tasks. This empirical finding reflects, at the neural level, that the syntax task, being actually a search for rule violation, is executed more easily than the code comprehension task, which calls for the mental simulation of code execution.

[3] https://www.nitrc.org/projects/bnv/

Neural Correlates of Programmer's Workload

Prior to the presentation and report of our model it is interesting to demonstrate that the programmer's workload (as registered during the briefing after each experiment) correlates with the pattern of brain's activation (i.e. SP measurements) and the pattern of functional connectivity (i.e. PLV measurements).

Observing Figure 8 we can clearly see in the heat map that the ''general trend'' is a strong correlation between brain activation in the higher bands (β1, β2 and gamma) and a programmer's workload. Figure 9, on the other hand indicates a more complex relation between connectivity and workload. There are positive correlations between couplings and workload within the two lowest bands and within β2. There are also significant negative correlations between coupling within the 4 lowest bands and the programmer's workload. These observations indicate that during code comprehension, the coordination between distinct brain areas follows a more composite scheme with only few of them engaged together through phase synchrony.

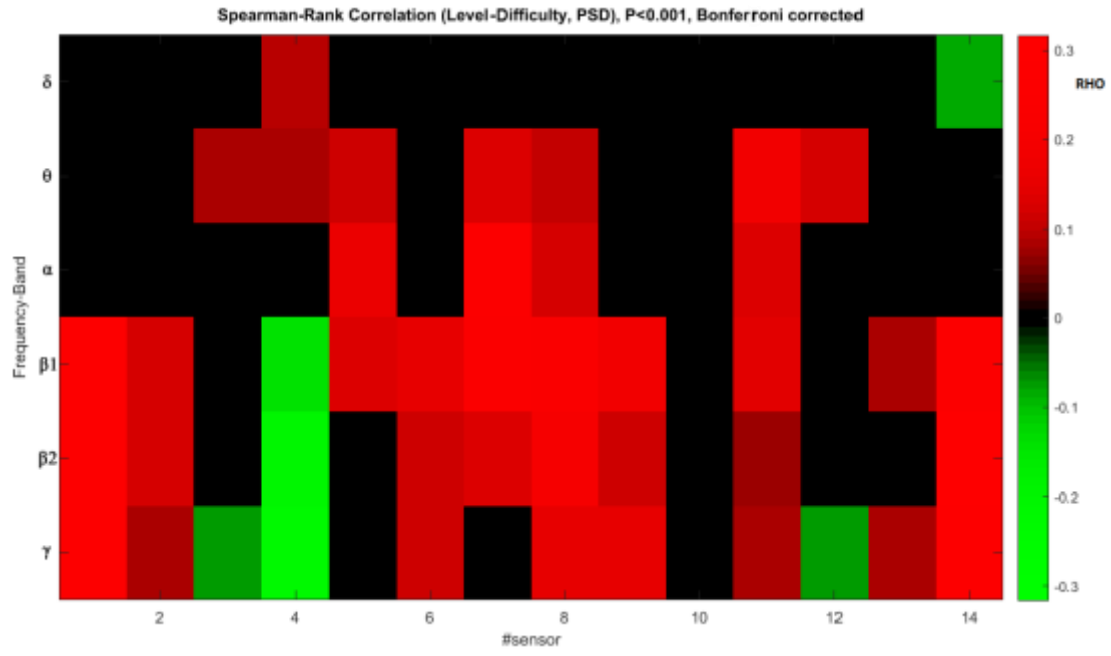


Figure 8. Spearman's RHO correlation between SP (at a sensor and frequency band) and programmer's reported difficulty in code understanding. Only significant activations (P<0.001; Bonferroni corrected) are shown

Figure 9. Spearman's RHO correlation between PLV (at every sensor-pair) and programmer's reported difficulty in code understanding. Only significant couplings (P<0.01; Bonferroni corrected) are shown.

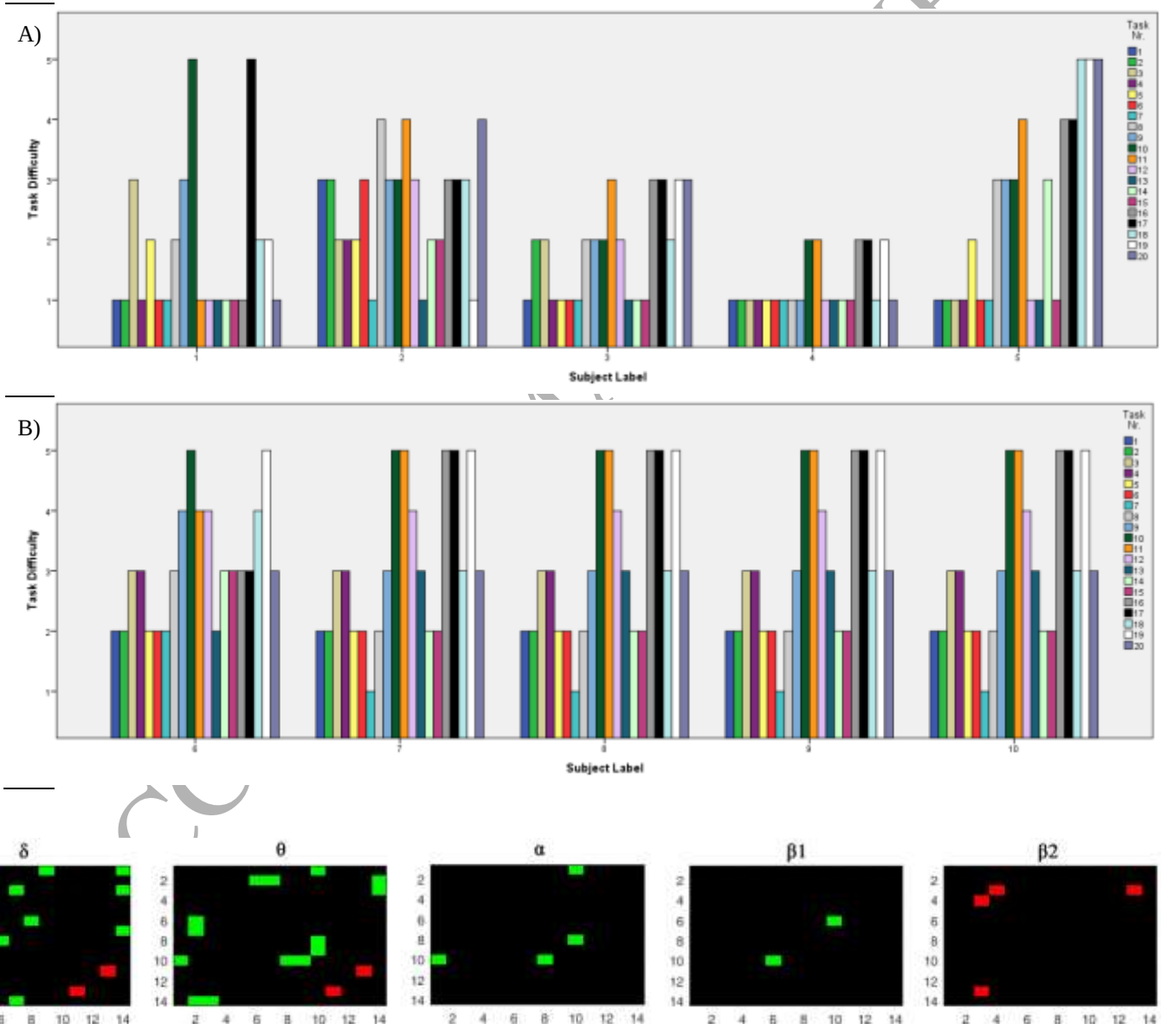


Figure 10. a) The ratings of subjects 1-5 for the 20 given comprehension tasks and b) The ratings of subjects 6-10 for the same tasks

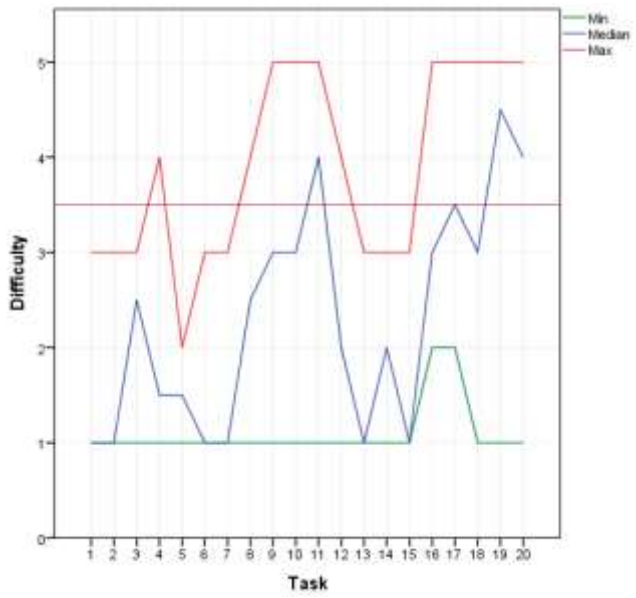


Figure 11. Comprehension task ratings across subjects

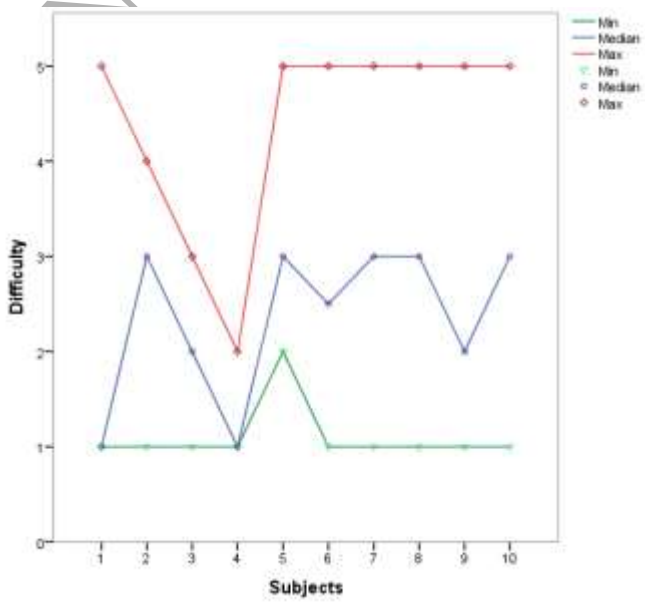


Figure 12. The ratings given by the subjects across tasks

Predicting Workload

The primary goal of our study was to model the relation of the subjective sense of difficulty (regarding a delivered comprehension task) with brain activity signatures reflecting the mental workload (induced while accomplishing this task). The subjective sense of difficulty was quantified by asking each participant to rate the comprehension tasks in a range from 1 to 5. The collected ratings for each subject are shown in Figure 10.a and 10.b below. As we can observe form the two figures, subjects 1, 3 and 4 were those that found the tasks easier than the others.

In order to provide more information and additional statistics on the way our subjects responded regarding the difficulty of the comprehension tasks presented to them during the experimentation sessions, we present two extra figures representing the ratings of the task difficulties across subjects (see Figure 11) and the task difficulty ratings across tasks (see Figure 12). The red, blue and green lines represent the maximum, median and minimum rating values respectively in each figure. From Figure 11 we understand that the most difficult tasks of our experiment, as rated by the subjects based on Median>3.5, were Task 11, 17, 19 and 20 corresponding to the *median on sorted data*, *bubble sort*, *matrix multiplication* and the *least common multiple* respectively (see Table 2 for number-tasks associations). Moreover, from Figure 12 and examining all three lines we can conclude that the Subjects having more difficulties in the comprehension of the tasks were Subjects 6, 7, 8, 9 and 10.

Furthermore, with the aim of accomplishing the abovementioned modeling, we experimented thoroughly using distinct types of brain activity measurements (SP and PLV). These measurements have been already presented and explained in section 3.5. Moreover, we used an additional predictor in our models. We added a categorical variable representing every subject with values from 1-10. The reason we used this extra variable was our need to ensure that the developed model would be a global one (working independently, regardless of the subjects participated in the experiment).

Table 4 shows the best models attained by the aforementioned procedure for the two datasets used. All models were statistically significant ($p<<0.0001$).

Table 4. Evaluation of the three best models for each dataset, the values of each column (Spearman rho, Classification percentage (Class.), MAE (Mean Absolute Error)) were calculated comparing the actual difficulty values with the ones predicted by each prediction model

	Spearman rho ($p<<0.0001$)	Class.	MAE	Nr. of predictors
SP model	0.76	44%	0.58	28/85

PLV model	**0.78**	**55%**	**0.51**	17/547

Table 5. Parameter Estimates of the SP model (Diff = rated Difficulty)

		Estimates	*Sig.*	*Location*	*Estimate*	*Sig.*
Threshold(θ_j)	**[Diff = 1]**	-4.042	0.000			
	[Diff = 2]	-2.611	0.000			
	[Diff = 3]	-1.088	0.068			
	[Diff = 4]	0.092	0.877			
Location(X)	**ln_γ(4)**	-0.42	0.00	**ln_γ(9)**	0.26	0.00
	ln_α(7)	0.41	0.00	**ln_β1(12)**	0.21	0.00
	ln_α(4)	-0.36	0.00	**ln_δ(14)**	-0.12	0.01
	ln_α(2)	0.36	0.00	**ln_β1(11)**	0.21	0.01
	ln_γ(11)	0.41	0.00	**ln_α(1)**	-0.16	0.01
	ln_β2(3)	-0.36	0.00	**ln_θ(10)**	-0.12	0.02
	ln_δ(4)	0.18	0.00	**ln_β2(6)**	0.21	0.02
	ln_β2(5)	0.33	0.00	**ln_α(9)**	0.15	0.03
	ln_α(13)	-0.25	0.00	**ln_δ(8)**	0.10	0.04
	ln_γ(8)	-0.30	0.00	**[subject=1]**	-2.01	0.00
	ln_β1(5)	0.31	0.00	**[subject=2]**	-1.78	0.00
	ln_γ(6)	-0.37	0.00	**[subject=3]**	-2.39	0.00
	ln_γ(1)	0.25	0.00	**[subject=4]**	-2.42	0.00

ln_δ(7)	-0.21	0.00	**[subject=5]**	-0.33	0.15
ln_δ(11)	-0.17	0.00	**[subject=6]**	-0.74	0.00
ln_β1(3)	-0.26	0.00	**[subject=7]**	-0.14	0.40
ln_β2(10)	-0.22	0.00	**[subject=8]**	0.37	0.08
ln_δ(6)	0.17	0.00	**[subject=9]**	-0.32	0.05
ln_δ(1)	0.13	0.00	**[subject=10]**	0[a]	

Table 6. Parameter Estimates of the PLV model (Diff = rated Difficulty)

		Estimates	*Sig.*	*Location*	*Estimate*	*Sig.*
***Threshold*(θ_j)**	**[Diff = 1]**	-10.67	0.00			
	[Diff = 2]	-8.60	0.00			
	[Diff = 3]	-6.21	0.00			
	[Diff = 4]	-4.34	0.01			
Location(X)	**δ(3,7)**	-8.86	0.00	**α(4,9)**	-6.54	0.01
	β2(5,6)	10.00	0.00	**α(4,8)**	-5.55	0.01
	β1(11,14)	-13.84	0.00	**α(5,13)**	6.67	0.01
	γ(4,6)	13.91	0.00	**β1(7,11)**	7.65	0.02
	β2(2,10)	6.85	0.00	**θ(4,14)**	-7.31	0.02
	β1(4,10)	-9.59	0.00	**β2(9,10)**	-3.47	0.03
	γ(4,13)	4.37	0.01	**α(6,13)**	-4.56	0.04
	θ(4,13)	-5.55	0.01	**δ(9,11)**	-4.22	0.05
	δ(9,14)	-5.12	0.01			

The predictor coefficients for each model are shown in Tables 5 and 6, under the *Estimate* column. The abbreviations used for the predictors in Tables 5 and 6 refer to the band of the estimator using the corresponding letter (see the mapping on Table 1) followed by a parenthesis containing the sensor number (see Figure 2) or the sensor pair ($sensor_i$, $sensor_j$), for the SP and PLV models respectively.

Since the predictors in the case of the SP model were logarithmically transformed, using the natural logarithm $\log_e(x)$, the abbreviations of predictors in this model are preceded by an (ln_) part, i.e. the ln_γ(4) variable corresponds to the logarithmically transformed Signal Powers of the gamma band in sensor$_4$. Moreover, as shown in both tables, under the *Sig.* column, all the participating predictors were statistically significant ($p \le 0.05$).

Based on the *Threshold* and *Location* the equation of the SP model is:

$$\text{link}(\gamma_j) = \theta_j - (-0.42 * \ln\_\gamma(4) + 0.41 * \ln\_\alpha(7) - 0.36 * \ln\_\alpha(4) + \ldots + 0.1 * \ln\_\delta(8) + subject_n), \quad (3)$$

for $j = 1,2,3,4$ and $n = 1,2,3,4,5,6,7,8,9,10$,

where, $\theta_j = \begin{Bmatrix} -4.042 & \text{if} \quad j=1 \\ -2.611 & \text{if} \quad j=2 \\ -1.088 & \text{if} \quad j=3 \\ 0.092 & \text{if} \quad j=4 \end{Bmatrix}$ and $subject_n = \begin{Bmatrix} -2.01 & \text{if} \quad n=1 \\ -1.78 & \text{if} \quad n=2 \\ -2.39 & \text{if} \quad n=3 \\ -2.42 & \text{if} \quad n=4 \\ -0.33 & \text{if} \quad n=5 \\ -0.77 & \text{if} \quad n=6 \\ -0.14 & \text{if} \quad n=7 \\ 0.37 & \text{if} \quad n=8 \\ -0.32 & \text{if} \quad n=9 \\ 0 & \text{if} \quad n=10 \end{Bmatrix}$

Likewise, the PLV model is reflected in the following equation:

$$\text{link}(\gamma_j) = \theta_j - (-8.86 * \delta(3,7) + 10 * \beta 2(5,6) - 13.84 * \beta 1(11,14) + 13.91 * \gamma(4,6) + \ldots - 0.42 * \delta(9,11)), \quad (4)$$

for $j = 1,2,3,4$,

where $\theta_j = \begin{Bmatrix} -10.67 & \text{if} \quad j=1 \\ -8.600 & \text{if} \quad j=2 \\ -6.21 & \text{if} \quad j=3 \\ -4.34 & \text{if} \quad j=4 \end{Bmatrix}$

Positive coefficients of the predictors' coefficients (*β*-values) show that, as the values of the independent variables increase, the likelihood of larger scores of the dependent variable increases as well. Examining the predictors that participate in both models, SP and PLV, we cannot jump to conclusions on whether a specific band affects more the total assessed workload, either positively or negatively, or whether a specific lobe participates more than the others in the estimation of mental effort based on code comprehension. This is because we have predictors in the model from a variety of bands and brain location.

The PLV-related model, in contrast to the SP-based one, did not include the subject category variable as a predictor, which in addition to its predictive power is appointed as the preferred model achieved in

our study, as it assesses mental workload independently of the subjects' participation in the experiment. From Table 4 it is clear that the use of the PLV measurements expresses better the relation between the difficulty of a comprehension task and the workload of a software engineer.

Finally, it is important to mention here the results from correlating the reported difficulty levels with the evaluation of the code descriptions, both provided during the briefing session after the recordings (see Section 3.3). We found a strong and statistically significant correlation between the two measurements. These results are important as they practically indicate that the subjects' perception of difficulty corresponded to their concrete level of understanding about each presented code snippet. Moreover, this shows that the use of their ratings in the model had a sound basis and showed the true difficulty, which reflected in their code snippets.

Threats to Validity

In studies involving measurements of human processes and performance, limitations are inevitable. Our study is no exception and some notes are in order regarding some practical issues that relate to the sensitivity of EEG in its "mobile" version. EEG is sensitive to head/body movements and even face expressions. Such activity, whenever present, is translated into noise that contaminates the brain signal. A real-time de-noising algorithm (e.g. Hsu et al., 2016) should therefore precede the estimation of workload.

Furthermore, one of the advantages of using an off-the-shelf wireless EEG device is the provided mobility and flexibility it offers to the participants and the researchers regarding the place it will be used and the manner it will be managed by the participants. The whole idea of such a setting is to faithfully represent a programmer's working environment. This benefit of working under normal conditions was somewhat compromised when we asked our participants to constrain their movements and facial expressions. However, this is the standard practice in EEG studies for registering brain activity reflecting mainly the cognitive task under investigation.

Moreover, regarding the device chosen for the experiment, despite its low cost and provided flexibility, it cannot replace the high accuracy given from ambulatory EEG or other ambulatory devices, which on the other hand are much more expensive for academic use.

The generalizability of the obtained results is also difficult to ascertain. Consequently, although it would be tempting to infer that the process can be used to evaluate the mental workload associated with arbitrary software development tasks, more studies are needed in order to be able to make this claim.

Discussion and Conclusions

In this paper we report results from one of the few attempts to associate brainwaves with the cognitive process during the mental activities of a programmer using EEG. Our study showed that the estimates of functional connectivity could be used to form a biomarker that relates directly with the mental workload induced in a programmer. Although there is recent literature that examines EEG functional connectivity in relation to mental workload (Dimitriadis et al., 2015; Dimitriadis et al., 2012; Dimitriadis et al., 2010;), this is the first study that addresses this issue in the field of software engineering and presents a methodology to model the relation of brainwaves with the mental workload of a programmer in the course of code comprehension.

After analyzing the workload of a programmer during code comprehension versus that of trying to find syntax errors, we found clear differences between the observed effort levels required for the comprehension and syntax task respectively. Figure 5 illustrates this contrast showing higher brain activation (red area) during comprehension in θ-and β2-bands. In practice, this means that a search for rule violation (syntax task) is executed more easily than code comprehension, which calls for the mental simulation of code execution. Consequently, we demonstrated that our method could be used for assessing and comparing the mental effort associated with programming tasks.

By means of SP contrasts (see Figure 6) we found significant brain activation in β and γ rhythms. It is important to stress here that functional connectivity measurements were found superior to the conventional signal power related ones (which have already been encountered in the field (Fritz et al., 2014; Ikutani and Uwano, 2014)).

We concluded our analysis proposing a statistically significant regression model to predict task difficulty during comprehension by means of brain activity measurements. We applied a discriminative technique (ordinal regression) to build our model (see Table 4) in which the assumptions made are weaker than in generative ones (i.e. Naïve Bayes), which overall leads to lower bias. Naïve Bayes for example assume that the features are conditionally independent, which is not acceptable when talking about real data. This limitation is not met in the case of ordinal regression, which works better in the case of inter-feature correlations (Ng and Jordan, 2002; Vapnik, V. N. and Vapnik, V., 1998).

The ability to monitor and estimate mental workload in a software development setting can be useful in a number of ways. First, it can be used to identify tasks that required a high-level of mental effort, e.g., by appropriately annotating the relevant resulting artifacts, such as code, designs, requirements, test cases, or documentation. Based on this markup, software development processes can be devised to direct additional scrutiny to these artifacts, for example through peer reviews, more comprehensive testing, prototyping, or static analysis and thus prevent faults (Lee et al., 2017; Müller, 2015). Mental effort measurements can be used by scientists to compare software development tools, processes, formalisms in order to find those that can be used to achieve the same task with less developer stress.

Moreover, as human computer interaction (HCI) systems are becoming omnipresent and being used to achieve significant tasks in a plethora of domains, the need of being able to estimate the instigated cognitive load by such systems turns out to be important before putting them at use (Berka et al., 2007; Kumar N. and Kumar, J., 2016). Thus, these data may prove invaluable for studying a wide variety of issues related to human factors in software engineering, to improve both the quality of the software and the working conditions of the people who develop it. In addition they can be used also to evaluate the interactive (developed) systems themselves, in order to measure their cognitive load on the users with respect to ease of use, efficiency, effectiveness, learnability, memorability and user satisfaction.

Finally we would like to emphasize that, while this study may be a testimony that the use of consumer-grade brain activity monitoring devices might be useful in assessing efficiency of processes or the level of expertise of a programmer (derived from the fact that it can potentially provide indications regarding the workload of their brain), this message should not be taken without the necessary precaution. As pointed out earlier, we envisage that this technology could be used as a *neuroergonomics* tool in order to improve the quality of programmers' working conditions and reduce their stress. It is also important to notice that such a technology also unwraps scenarios that we would considered unethical, e.g. using brain scanners in the context of a recruitment process or as a means of employee re-evaluation. Hence, it is important that a careful ethical framework is put in place to define the boundaries of acceptable use of these technologies.

References

Adamos, D. A., Dimitriadis, S. I., & Laskaris, N. A., 2016. Towards the bio-personalization of music recommendation systems: A single-sensor EEG biomarker of subjective music preference. *Information Sciences*, *343*, 94-108.

Anderson, E. W., Potter, K. C., Matzen, L. E., Shepherd, J. F., Preston, G. A., & Silva, C. T., 2011, June. A user study of visualization effectiveness using EEG and cognitive load. In *Computer Graphics Forum* (Vol. 30, No. 3, pp. 791-800). Blackwell Publishing Ltd.

Antonenko, P., Paas, F., Grabner, R., & Van Gog, T., 2010. Using electroencephalography to measure cognitive load. *Educational Psychology Review*, *22*(4), 425-438.

Berka, C., Levendowski, D. J., Lumicao, M. N., Yau, A., Davis, G., Zivkovic, V. T., ... & Craven, P. L., 2007. EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks. *Aviation, space, and environmental medicine*, *78*(5), B231-B244.

Boehm, B. W., 1988. Understanding and controlling software costs. *Journal of Parametrics*, *8*(1), 32-68.

Capretz, L. F., Varona, D., & Raza, A., 2015. Influence of personality types in software tasks choices. *Computers in Human Behavior*, *52*, 373-378.

Coyne, J. T., Baldwin, C., Cole, A., Sibley, C., & Roberts, D. M., 2009, July. Applying real time physiological measures of cognitive load to improve training. In *International Conference on Foundations of Augmented Cognition* (pp. 469-478). Springer Berlin Heidelberg.

Crk, I., Kluthe, T., & Stefik, A., 2016. Understanding programming expertise: an empirical study of phasic brain wave changes. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *23*(1), 2.

Das, D., Chatterjee, D., & Sinha, A., 2013, November. Unsupervised approach for measurement of cognitive load using EEG signals. In *Bioinformatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on* (pp. 1-6). IEEE.

Dimitriadis, S. I., Sun, Y. U., Kwok, K., Laskaris, N. A., Thakor, N., & Bezerianos, A., 2015. Cognitive workload assessment based on the tensorial treatment of EEG estimates of cross-frequency phase interactions. *Annals of biomedical engineering*, *43*(4), 977-989.

Dimitriadis, S. I., Kanatsouli, K., Laskaris, N. A., Tsirka, V., Vourkas, M., & Micheloyannis, S., 2012. Surface EEG shows that functional segregation via phase coupling contributes to the neural substrate of mental calculations. *Brain and cognition*, *80*(1), 45-52.

Dimitriadis, S. I., Laskaris, N. A., Tsirka, V., Vourkas, M., & Micheloyannis, S., 2010. What does delta band tell us about cognitive processes: a mental calculation study. *Neuroscience letters*, *483*(1), 11-15.

Ferreira, E., Ferreira, D., Kim, S., Siirtola, P., Röning, J., Forlizzi, J. F., & Dey, A. K., 2014, December. Assessing real-time cognitive load based on psycho-physiological measures for younger and older adults. In *Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB), 2014 IEEE Symposium on* (pp. 39-48). IEEE.

Floyd, B., Santander, T., & Weimer, W. Decoding the representation of code in the brain: An fMRI study of code review and expertise (In press, ICSE 2017).

Fritz, T. & Müller, S. C., 2016, March. Leveraging biometric data to boost software developer productivity. In *Software Analysis, Evolution, and Reengineering (SANER), 2016 IEEE 23rd International Conference on* (Vol. 5, pp. 66-77). IEEE.

Fritz, T., Begel, A., Müller, S. C., Yigit-Elliott, S., & Züger, M., 2014, May. Using psycho-physiological measures to assess task difficulty in software development. In *Proceedings of the 36th International Conference on Software Engineering* (pp. 402-413). ACM.

Glass, R. L., 2002. *Facts and fallacies of software engineering*. Addison-Wesley Professional.

Google Inc., 2014. Benefits—Google Jobs. http://perma.cc/TC89-Q6JD *ACM SIGCHI Bulletin*, *14*(4), 19-20.

Harrell Jr, F. E. (2015). *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. Springer.

Hsu, S. H., Mullen, T. R., Jung, T. P., & Cauwenberghs, G., 2016. Real-time adaptive EEG source separation using online recursive independent component analysis. *IEEE transactions on neural systems and rehabilitation engineering*, *24*(3), 309-319.

IBM Corp. Released 2013. IBM SPSS Statistics for Windows, Version 22.0. Armonk, NY: IBM Corp.

Ikutani, Y., & Uwano, H., 2014, June. Brain activity measurement during program comprehension with NIRS. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2014 15th IEEE/ACIS International Conference on* (pp. 1-6). IEEE.

Jensen, O., & Tesche, C. D., 2002. Frontal theta activity in humans increases with memory load in a working memory task. *European journal of Neuroscience*, *15*(8), 1395-1399.

Kluthe, T., 2014. "Measurement of Programming Language Comprehension Using p-BCI: An Empirical Study on Phasic Changes in Alpha and Theta Brain Waves (Master's thesis).". Retrieved from http://pqdtopen.proquest.com/pubnum/1560916.html

Kosti, M. V., Feldt, R., & Angelis, L., 2016. Archetypal personalities of software engineers and their work preferences: a new perspective for empirical studies. *Empirical Software Engineering*, *21*(4), 1509-1532.

Kosti, M. V., Feldt, R., & Angelis, L., 2014. Personality, emotional intelligence and work preferences in software engineering: An empirical study. *Information and Software Technology*, *56*(8), 973-990.

Kumar, N., & Kumar, J., 2016. Measurement of Cognitive Load in HCI Systems Using EEG Power Spectrum: An Experimental Study. *Procedia Computer Science*, *84*, 70-78.

Lachaux, J. P., Rodriguez, E., Le Van Quyen, M., Lutz, A., Martinerie, J., & Varela, F. J., 2000. Studying single-trials of phase synchronous activity in the brain. *International Journal of Bifurcation and Chaos*, *10*(10), 2429-2439.

Laskaris, N., Fotopoulos, S., Papathanasopoulos, P., & Bezerianos, A., 1997. Robust moving averages, with Hopfield neural network implementation, for monitoring evoked potential signals. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, *104*(2), 151-156.

Lee, H., 2014. Measuring cognitive load with electroencephalography and self-report: focus on the effect of English-medium learning for Korean students. *Educational Psychology*, *34*(7), 838-848.

Lee, J. M., & Shneiderman, B., 1978. January. Personality and programming: Time-sharing vs. batch preference. In *Proceedings of the 1978 annual conference-Volume 2* (pp. 561-569). ACM.

Lee, S., Hooshyar, D., Ji, H., Nam, K., & Lim, H., 2017. Mining biometric data to predict programmer expertise and task difficulty. *Cluster Computing*, 1-11.

Lee, S., Matteson, A., Hooshyar, D., Kim, S., Jung, J., Nam, G., & Lim, H., 2016 (October). Comparing Programming Language Comprehension between Novice and Expert Programmers Using EEG Analysis. In *Bioinformatics and Bioengineering (BIBE), 2016 IEEE 16th International Conference on* (pp. 350-355). IEEE.

Mathewson, K. E., Basak, C., Maclin, E. L., Low, K. A., Boot, W. R., Kramer, A. F., ... & Gratton, G., 2012. Different slopes for different folks: Alpha and delta EEG power predict subsequent video game learning rate and improvements in cognitive control tasks. *Psychophysiology*, *49*(12), 1558-1570.

Mathôt, S., Schreij, D., & Theeuwes, J., 2012. OpenSesame: An open-source, graphical experiment builder for the social sciences. *Behavior research methods*, *44*(2), 314-324.

MATLAB, V., 2013. 8.1. 0.604 (R2013a). MathWorks, Natick, MA.

Michel, C. M., & Murray, M. M., 2012. Towards the utilization of EEG as a brain imaging tool. *Neuroimage*, *61*(2), 371-385.

Müller, S. C., 2015, May. Measuring software developers' perceived difficulty with biometric sensors. In *Proceedings of the 37th International Conference on Software Engineering-Volume 2* (pp. 887-890). IEEE Press.

Nakagawa, T., Kamei, Y., Uwano, H., Monden, A., Matsumoto, K., & German, D. M., 2014, May. Quantifying programmers' mental workload during program comprehension based on cerebral blood flow measurement: a controlled experiment. In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 448-451). ACM.

Ng, A. Y., & Jordan, M. I., 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, *2*, 841-848.

Niedermeyer, E., & da Silva, F. L. (Eds.)., 2005. *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins.

Paas, F., Tuovinen, J. E., Tabbers, H., & Van Gerven, P. W., 2003. Cognitive load measurement as a means to advance cognitive load theory. *Educational psychologist*, *38*(1), 63-71.

Parnin, C., 2011, June. Subvocalization-toward hearing the inner thoughts of developers. In *Program Comprehension (ICPC), 2011 IEEE 19th International Conference on* (pp. 197-200). IEEE.

Sammet, J. E., 1983. Software psychology: human factors in computer and information systems.

Siegmund, J., Kästner, C., Apel, S., Parnin, C., Bethmann, A., Leich, T., ... & Brechmann, A., 2014, May. Understanding understanding source code with functional magnetic resonance imaging. In *Proceedings of the 36th International Conference on Software Engineering* (pp. 378-389). ACM.49. Soloway, E., & Ehrlich, K., 1984. Empirical studies of programming knowledge. *IEEE Transactions on software engineering*, (5), 595-609.

Soloway, E., & Ehrlich, K., 1984. Empirical studies of programming knowledge. *IEEE Transactions on software engineering*, (5), 595-609.

Temes, G. C., & LaPatra, J. W., 1977. *Introduction to circuit synthesis and design*. McGraw-Hill Companies.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267-288.

Vapnik, V. N., & Vapnik, V., 1998. *Statistical learning theory* (Vol. 1). New York: Wiley.

Zarjam, P., Epps, J., & Chen, F., 2011, August. Spectral EEG featuresfor evaluating cognitive load. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE* (pp. 3841-3844). IEEE.

Zarjam, P., Epps, J., & Chen, F., 2010. Evaluation of working memory load using EEG signals. In *Proc. APSIPA Annual Summit and Conference* (pp. 715-719).

Makrina Viola Kosti received her BSc degree in Informatics from the Aristotle University of Thessaloniki (AUTh) and she has completed the postgraduate program of *"Computer Science and Business Administration"* also in AUTh. She is currently a PhD student, working with STAINS (STAtistics and INformation Systems) research group on multivariate statistical methods for prediction with particular interest of research on the human factor.

Konstantinos Georgiadis holds a BSc degree in Computer Science from the University of Crete (2013) and a MSc in Informatics with specialization in Digital Media from the Department of Informatics, Aristotle University of Thessaloniki (2015). He is currently a PhD candidate in the Department of Informatics at the Aristotle University of Thessaloniki. Since July 2015, he has been working as a research assistant at the Informatics and Telematics Institute (ITI) of the Centre of Research & Technology Hellas (CERTH). His research interests among others include Biomedical Signal and Image Processing, Machine Learning and Brain Computer Interfaces.

Dimitrios A. Adamos is a senior teaching / research fellow at the School of Music Studies, Aristotle University of Thessaloniki (AUTh) and a member of the Neuroinformatics GRoup. He also has work experience as a senior network engineer. He holds a Dipl. in Electrical & Computer Engineering, an MSc in Medical informatics from the School of Medicine and a PhD in Neuroinformatics from the School of Biology of AUTh. His research interests include neuroinformatics, wearable electroencephalography, machine learning and novel forms of human-computer interaction.

Nikos Laskaris is an assistant professor at the Department of Informatics, Aristotle University, Greece. He is a member of AIIA lab and leads the NeuroInformatis.GRoup. He is a co-author of more than 100 scientific publications. His current research interests include computational intelligence, soft computing, data mining, nonlinear dynamics and their applications in biomedicine and neuroscience.

Diomidis Spinellis is a Professor in the Department of Management Science and Technology at the Athens University of Economics and Business, Greece. From January 2015 he is serving as the Editor-in-Chief for *IEEE Software*. His latest book is *Effective Debugging: 66 Specific Ways to Debug Software and Systems* (Addison-Wesley, 2016).

Lefteris Angelis studied Mathematics and received his Ph.D. degree in Statistics from Aristotle University of Thessaloniki (A.U.Th.). He is currently a Professor at the Department of Informatics of A.U.Th. and coordinator of the STAINS (STAtistics and INformation Systems) research group. His research interests involve statistical methods with applications to information systems and software engineering, especially regarding the research on the human factor, computational methods in mathematics and statistics, planning of experiments and simulation techniques.