Measuring Code Difficulty by Examining Psycho-Physiological Signals

Seminar: SE Research in the Neuroage

Niklas Britz

Advisors: Prof. Sven Apel,
Sebastian Böhm

Saarland Informatics Campus,
Saarland University

1 Introduction

A major aspect of Software Engineering is the correctness of software. Bugs can cause unhappy customers and cost tons of money [1]. Therefore it is important to keep code clean and understandable. How can we minimize those flaws?
The human itself plays a principal role in the success of software projects. If humans are stressed out or have difficulties solving a particular problem, the process of software engineering can stagnate, and bugs can sneak in. Even today, we know little about how the human brain when it comes to software development [23] [21] [8].
In this paper, we focus on how the brain and body react to different difficulty levels of coding tasks and discuss how biometric signals indicate those levels - independent of empirical difficulty metrics of code. In this paper we discuss the use of *Lightweight Biometric Sensors* (LBS) [10] and compares them with functional Magnetic Resonance Imaging (fMRI).
Finally, we present a study idea that aims to classify code difficulty based on psycho-physiological measures with the help of Machine Learning. The study is based on an experiment by Fritz et al. [9] but proposes some significant alterations. Our study aims to improve their study by considering novel knowledge and using different equipment like fNIRS and heart-related sensors.

This paper

- presents an fMRI study on Code vs. Prose Comprehension by Floyd et al. [8] and compares it to a replication study by Fucci et al. [10] using lightweight biometric sensors. The study shows that using LBS are sensible for classification tasks with high ecological validity and low costs and that they are superior to fMRI studies in a specific context.
- uses this information to present a study idea that aims at improving a study by Fritz et al. [9]. In the study, the researchers classify psycho-physiological

[1] https://medium.com/@99tests/21-infamous-expensive-software-bugs-f678827f94a6

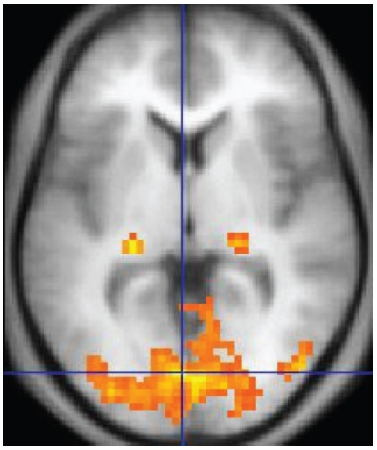
signals into 'comprehending simple vs. difficult code' with LBS. By including novel knowledge, other devices, and techniques and enriching the original study by the distinguishment of novices and experts, we aim to enhance and improve their study.

1.1 Background

Recent years have brought us superior medical imaging techniques and various possibilities to analyze biological signals of humans. We can use those devices and techniques to gain knowledge about the cognitive state of a person.

One tool that is particularly suitable for a deep understanding of neural processes is fMRI. MRI is a medical imaging technique that reveals the morphology and composition of biological tissues.
In the 1990s, researchers discovered that with MRI, one could observe different levels and changes in oxygenated blood. Oxygenated blood does not interact with a magnetic field as much as deoxygenated blood does. The activation of a specific brain area requires oxygen. Therefore the ratio between oxygenated and deoxygenated blood increases compared to a baseline state. This is called the *BOLD* effect. From the differences in the BOLD response, one can infer which brain areas are activated during specific tasks. [21]

Fig. 1. fMRI image. Highlighted area displays increased activity compared with a baseline image



In 2014 Siegmund et al. [23] published a seminal paper on using fMRI in the context of SE. The team proposed fMRI as a suitable tool for empirical software engineering and found that fMRI can be used to gain a deeper understanding of code comprehension. Since then, many other research teams have examined the human factor in software engineering using fMRI [8] [5] [14] [17].

It seems that fMRI, due to its high spatial locality, is the silver bullet when it comes to answering neuroscientific questions in Software Engineering. However, fMRI is expensive (about 500$ per hour [8]), and researchers argue that the results of fMRI studies have low ecological validity[2] [10].
In this study, we focus on devices and methods that do not have these disadvantages and can be used to measure the cognitive state of a person. Methods can either directly measure brain activity or the 'symptoms' of what is happening inside the brain. Fear, for example, can lead to an elevated heart rate and increased skin conductance level [2, p. 372]. When measuring a few of those 'symptoms' and putting them into context, one can estimate the cognitive state of a person.
This paper mainly focus on what Fucci et al. call [10] call *lightweight biometric sensors*. They characterize them as "non-intrusive, wearable, and affordable devices—to measure human physiology" [10, p. 311]. The researchers argue that those devices have the advantage of being cheaper and their results being more ecologically valid, i.e., more applicable to a real-world setting, than fMRI.
To our best knowledge, Chris Parnin [20] was the first to use LBS, namely a Electromyogram (EMG), in the context of software engineering. He used the EMG to detect sub-vocal utterances ("electrical signals sent to the tongue, lips, or vocal cords" [20, p. 197]) and then relate them to a specific area of code the subjects had looked at.
The study by Fritz et al. [9], that we base our experiment on, used eye tracking, EEG, and EDA, which can be categorized as LBS, to classify developer's biometric signals into 'understanding simple source code' vs. 'understanding hard source code'. Their results were promising but improvable. The model's precision for classifying the task difficulty was below 70% - given a novel developer with a novel task. In Part 3 of this paper, we discuss the alterations in greater detail.

2 Are LBS Sensible for the Classification of Biometric Signals? Is fMRI Not the Silver Bullet?

To better understand whether data gathered from lightweight biometric sensors is suitable for SE related classification tasks, we discuss an exemplary study by Floyd et al. [8] (fMRI) and its replication study by Fucci et al. [10] (LBS).

2.1 Using fMRI

Motivation. Floyd et al. [8] examined the brain activity of developers while understanding code and reviewing prose. In this paper, we exclude their findings on code review because code review was also left out in the replication study. For that they used an fMRI scanner and argued that fMRI has advantages over EEG, PET, etc. because it monitors the brain's complete activity with outstanding precision.

[2] https://www.britannica.com/science/ecological-validity

Subjects. The final sample of developers consisted of 18 men and 11 women with different GPAs and experiences (e.g., graduate or undergraduate).

Tasks. In the fMRI scanner, the participants were provided with code, assertions about it, and English prose text with edits. With a button in hand, the developers had to indicate whether they approved of an assertion/edit or not. The researchers collected data until the decision was made.

Data Processing and Machine Learning. Then they used the fMRI device's four-dimensional data (3D + time). After preprocessing the data, the researchers obtained a time series with three-dimensional (spatial) voxels where the value of a voxel describes the activation of this area.
For machine learning, they used Gaussian Process Classification. Because fMRI data is extensive and the training set relatively small, they applied a kernel to the spacial voxels to reduce dimensions. Floyd et al. used leave-one-run-out cross-validation to test their classifier performance.

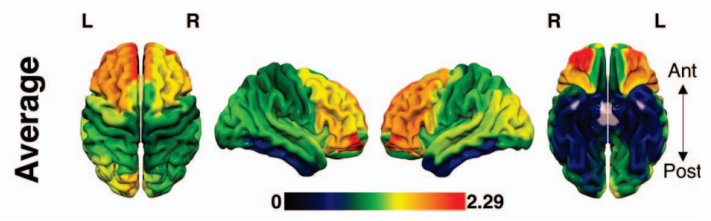
Results. The classifier could discriminate neural representations of prose review and code comprehension with a Balanced Accuracy Score (BAC) of 79.17%. Nevertheless, what about the activated regions? When overlapping data from code comprehension and prose review and considering that classification is possible, it becomes clear that the regions with a big difference in BOLD activation influence the model stronger than those with few deviations. The team then averaged the voxel weights over the number of voxels in the region and divided this 'contribution strength' [8] by the sum of strengths for all regions. The value can be interpreted as regional importance. Fig. 2 shows that especially the regions around the prefrontal cortex had huge importance for the classifier. The prefrontal cortex is responsible for short term memory [15] and higher order cognition [8]. Floyd et al. do also find that there is a high regional importance "near Wernicke's area in the temporoparietal cortex — a region classically associated with language comprehension" [8, p. 182].

It seems that we can accurately distinguish code comprehension tasks from prose reviewing tasks and have insights into what happens inside the brain. So why use other devices?

2.2 Using Lightweight Biometric Sensors

Motivation. Fucci et al. [10] wanted to replicate the study of Fritz et al. [9] by using lightweight biometric sensors. They identified the costs and the low ecological validity of the original study as the major issues. The use of the less invasive sensors, they argued, would fix those problems because they cost less

Fig. 2. Average regional importance map. "Hot" colors indicate areas containing a greater proportion of the total classification weight. Taken from [8]



(500$/hour vs. 2000$ one-time) and could be worn in a real-world setting. This is also the motivation for the use of LBS in our experiment.

Research questions. The researchers slightly altered the research questions of Fritz et al. and came up with the following two questions.

- RQ_{Clf}: Can we classify which task a participant is undertaking based on *signals collected from lightweight biometric sensors*?
- RQ_{Exp}: Can we relate expertise to classification accuracy?

Setup and Tools. In contrast to Fritz et al.'s setup, the team replaced prose review with high school prose comprehension tasks (simple assertions about a text), so the activities are comparable. This binary classification of comprehension tasks could have made more sense in Fritz et al.'s study, but it was left out because of the code review part of their study.
Fucci et al. decided to use the electronic wristband Empatica E4 to measure the developers' electrodermal activity (EDA) and heart activity in terms of blood volume pressure (BVP). The BVP can be used to calculate the heart rate (HR) and the heart rate variability (HRV). The EDA consists of a tonic component which indicates the skin's level of electrical conductivity (SCL), and a phasic component indicating changes in the conductivity or the skin's conductance response [10] [3].
They also used an EEG device that can be strapped around the head, called BrainLink. This devices measures cerebral waves and categorizes them into specific wavelengths (delta (<4Hz), theta (4-7.5Hz), alpha (4-12.5Hz), beta (13-30Hz), and gamma (>30Hz)). Specific wavelengths can be attributed to specific cognitive states. For example, alpha waves typically represent an awake but non-focused, relaxed state [3]. After a short pre-experimental briefing, the devices were calibrated by letting the developers watch a fish tank video. That method is used to collect physiological baselines [10]. Studies have indicated that there is "a link

[3] https://www.rsu.edu/wp-content/uploads/2015/06/TheElectroencephalogramEEGCorticalArousal.pdf

between viewing fish in aquariums and benefits such as reduced blood pressure and increased relaxation"[6, p. 4].
In each of the three sessions, which constituted the actual experiment, the developers then had to assess three code and six prose comprehension tasks displayed randomly. By pressing the arrow keys on a keyboard, they indicated if they approve or decline an assertion displayed for the respective example.

Fig. 3. Empatica E4 wristwatch and BrainLink EEG

Processing of data. To process the collected data, the team synchronized the collected signals with the posed questions regarding time from the beginning to the submission of an answer. Furthermore, they normalized the signals concerning the physiological data fetched in the last 30 seconds of the fish tank video, which they used as a baseline.
In order to extract features out of the EEG signal for machine learning, the researchers used bandpass filters at different intervals (alpha, beta, gamma, delta, theta) to decompose the bare signal. They used information from those frequency bins as features.
Those methods were similarly proposed by Canento et al. [4]. They also applied a bandpass filter (1-8Hz) to the rare BVP and collected information such as min, max, mean, and differences between the baseline signal and the task signal.
For extracting tonic and phasic components of the EDA Fucci et al. used the cvxEDA algorithm [11]. They extracted features such as mean tonic signal, min, max, mean, area under the receiving operator curve, etc.

Machine Learning. In order to classify the physiological data correctly, the team decided to use eight popular machine learning models (Naive Bayes (nb), kNN, Decision tree (J48), SVM with a linear kernel, Multi-layer Perceptron (mlp), Random Forest, rule-based optimizer (Jrip), boosted Decision Tree (C5.0)).
For testing, the researchers used a LORO and a Hold-out setting which they repeatably applied to achieve a reliable score. Because of the imbalanced training data (more prose than code comprehension), they propose a Balanced Accuracy

Score (BAC) as the measure for the classifier's performance.
In order to improve their results, Fucci et al. used hyperparameter tuning with a GridSearch approach.

Results. Concerning RQ_{Clf}, the researchers found out that there was no classifier that outperformed the others independently from the considered signals significantly, even though, according to the researchers, the Naive Bayes and the kNN seem inappropriate for the task. They found that the BAC is the highest when only considering heart-related signals. Heart + EEG + EDA does not improve performance significantly. The BAC of classifiers considering EEG only is the worst in both settings (LORO and Hold-out). Tables 1 and 2 display the results. Conclusively, they argued, lightweight biometric sensors could accurately differentiate between code and prose comprehension tasks.

Regarding RQ_{Exp}, Fucci et al. find no significant correlation between expertise (measured by GPA) and classifier performance using the Kendall tau correlation coefficient.

Signal	Best Classifier	Precision	Recall	F1	BAC
EEG	mlp	0.72	0.66	0.62	0.66
EDA	rf	0.78	0.71	0.71	0.71
Heart	mlp	0.91	0.87	0.87	**0.87**
EEG + EDA	C5.0	0.75	0.72	0.72	0.72
EEG + Heart	Jrip	0.90	0.86	0.87	0.86
EDA + Heart	mlp	0.91	0.83	0.86	0.83
EEG + EDA + Heart	Jrip	0.88	0.86	0.86	0.86

Table 1. Best machine learning classifier in LORO setting. Taken from [10].

Signal	Best Classifier	Precision	Recall	F1	BAC
EEG	rf	0.70	0.67	0.68	0.67
EDA	Knn	0.83	0.74	0.77	0.74
Heart	mlp	0.95	0.90	0.92	**0.90**
EEG + EDA	mlp	0.75	0.75	0.75	0.75
EEG + Heart	C5.0	0.90	0.89	0.90	0.89
EDA + Heart	svm	0.93	0.87	0.89	0.87
EEG + EDA + Heart	C5.0	0.92	0.89	0.90	0.89

Table 2. Best machine learning classifier in LORO setting. Taken from [10].

2.3 What does this show?

We saw that the lightweight biometric sensors outperformed the fMRI classification. Those sensors are sensible when deciding whether a developer is trying to understand prose or code. The benefit of this study is that the ecological validity is higher than with fMRI. Because the wristwatch is also affordable, it can be worn by developers on a daily basis. By integrating measured signals of LBS into software, it could help to fix imbalances in developers' daily work. Developers that have a high coding workload could be assigned assistance.
Conclusively, this exemplary study shows that LBS are reasonable for classification tasks with high ecological validity and it makes sense to use lightweight biometric sensors for our problem.

3 LBS, LBS on the Body, What's the Task's Difficulty? A Study Idea

As seen in the section above, LBS are practical tools when it comes to the distinguishment of biometric signals. That is why Fritz et al. [9] used those devices in 2014 to discriminate developer's psycho-physiological signals when comprehending simple code from those when comprehending difficult code. Their results are promising but, as we will see, improvable.
Our study aims to improve their results. In the following we will analyze their study and propose alterations and suggestions.

3.1 Original Study

Fritz et al. [9] wanted to figure out

- if they can acquire psycho-physiological measures from eye-tracking, EDA and EEG sensors to accurately predict whether a task is difficult or easy.
- RQ_2: which combination of psycho-physiological sensors and associated features best predicts task difficulty.

The researchers collected psycho-physiological signals from 15 developers who are professional software developers from the area around Seattle.
They have presented ten tasks of two different kinds. The first program assigned coordinated for the corners of two rectangles. The developers then had to answer if the rectangles overlapped (yes or no). There were three instances of the program. One was a simple practicing instance. The second one used local variables with single letters. The third one contained "randomized and interleaved assignments of the corner coordinates for both rectangles" [9, p. 405].
The second program created four shapes and printed them in some order. After the code snippet's display, the researchers presented a multi-choice question that dealt with what shapes were drawn and in what order. The instances of that

program differed in the used variable names, order of shape initialization and drawing, iterable array of shapes vs. singular items, calling a separate function, etc. After each question, the developers had to answer a NASA TLX [13] survey instrument, which asked them to rate the previously seen task from 1 – 20 for six categories: mental demand, physical demand, temporal demand, performance, effort, and frustration. Then the subjects had to rank those categories among each other by their importance. They combined the answers to receive an overall score. The developers also had to subjectively rank the perceived task difficulty for each task. The tasks with low overall ranks were classified as easy and with high overall ranks as difficult. The ones in the middle could be assigned to a category by looking at the developers' comments, which were unambiguous in 98% of the cases.

Results. After data cleaning and feature extraction from the biometric signals, they applied a Naive Bayes ML model. The researchers argued that Naive Bayes' training could be easily updated on-the-fly. They did three different types of predictions (by participant, by task, by participant-task). We focus on the 'by participant' classifier, which is the most useful in practice [9] because it can be applied to the biometric signal to an unknown person and an unknown task. The results can be seen in Table 3.

Signal	Precision	Recall	F-Measure
Eye	**0.69**	0.66	0.65
EDA	0.55	0.56	0.52
EEG	0.53	0.57	0.51
EYE + EDA	0.68	0.64	0.62
EYE + EEG	0.69	0.63	0.61
EDA + EEG	0.68	0.65	0.62
EYE + EDA + EEG	0.65	0.65	0.62

Table 3. Best machine learning classifier. Taken from [9].

We see that the best classifier has 'only' 69% classification accuracy. The study of Fritz et al. was published in 2014. Since then, many new studies in this area have been published. Our study aims to improve the original study's results (see below) because, as motivated, the problem is still relevant. With better results, one could better understand what characterizes task difficulty in SE from a psycho-physiological perspective. Moreover, for practical purposes, one would have a more accurate classifier.

3.2 EEG vs. fNIRS

In both Fritz et al.'s [9] and Fucci et al.'s [10] study, we can see that EEG has only a poor influence on the classifier performance (Table 1, 2, 3). In fact, in all

studies, EEG performed the worst. When looking at 'easy vs. hard code comprehension', EEG has a precision of around 53%. Considering that the training set is imbalanced (51 difficult and 65 easy tasks), one can argue that guessing would be better.
We cannot say for sure why the classifiers trained with EEG data perform so bad compared to other data sources. We see that EEG performs better when classified by task in Fritz et al.'s study. This could be because every person's brain is unique, and it is challenging to build a generalizable model. Another reason could be that it is difficult to learn from 31 EEG features with only a relatively small data set.
In order to capture cognitive load better we could either use a larger training set, which would not necessarily ensure a better predictor, or we could try to use another device.
fNIRS is a medical imaging technique that works complementary to EEG.
While EEG records cellular currents associated with neuronal activity, fNIRS quantifies the brain's activity by measuring cerebral blood flow. Because oxygenated (oxyHb) and deoxygenated blood have characteristic optical properties, fNIRS instruments can send and receive reflected near-infrared light to calculate the amount of oxygenated blood and thus the neural activity [1].

> "The main advantage of fNIRS over EEG/ERP is the localization of responses. In fNIRS, the effects are localized within 1–2 cm of the area activated, allowing for more accurate identification of the areas from which cortical responses were obtained than electrophysiological techniques." [25, p. 266]

We hope that better localization of signals could lead to better classification.
In the fMRI study by Floyd et al. [8], we saw that the neural representations of code comprehensions and prose review were less indistinguishable with a growing expertise. That means that for easy code comprehension tasks (cf. expertise), neural representations could be similar to when thinking about prose. In contrast, neural representations of demanding code comprehension tasks could match with 'normal code comprehension' in Floyd et al.'s study.
In Floyd et al.'s study, the researchers used brain localization, especially in the prefrontal cortex to receive their results. fNIRS can also use localize the origin of signals (although not as precise as fMRI and only 1cm brain depth [25]). fNIRS is also cost-effective and portable [1].

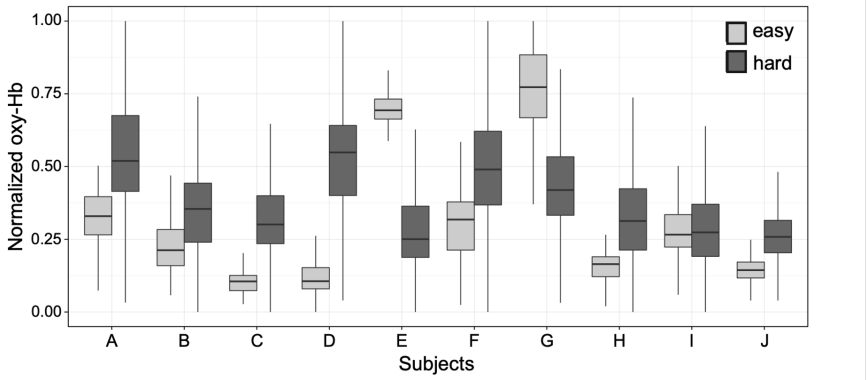
fNIRS in SE Studies. fNIRS was also part of Software Engineering studies. In particular, it has been used to measure and quantify brain activity when developers comprehend code.
Nakagawa et al. [18] were, for instance, one of the first to use fNIRS in a SE context. They invited ten graduate students to answer questions for specific code snippets of two difficulty levels. The researchers presented three different algorithms with two levels of difficulty each. The difficult tasks contained obfuscating

code and quickly changing variables. The developers then had to simulate the execution in their heads and had to evaluate variables at certain checkpoints. During the subjects performed the tasks, Nakagawa et al. measured the cerebral blood flow with fNIRS. Then they normalized the measured value of oxygenated blood with the following equation:

$$normalized\ oxyHb = \frac{oxyHb - min(s)}{max(s) - min(s)}$$

min(s) and max(s) are the maximum and minimum values of oxyHb throughout all tasks of each subject s. In Figure 4, we can see the researchers' results for the different subjects.

Fig. 4. Distributed oxyHb. Taken from [18]



Also, one can see that the variance within all hard tasks per participant is higher than within simple tasks. The researchers suggest that there are phases of low cognitive effort even during the comprehension of difficult code.

Another study by Ikutani and Uwano [15] measured the effect variables and controls in source code have on the brain's activity using fNIRS during program comprehension. They found that variable memorization leads to activation in the prefrontal cortex without the influence of arithmetics.

The two studies showed that fNIRS could be used to capture cognitive workload in a program comprehension setting. Especially Nakagawa et al. proved that fNIRS could be sensible in a task difficulty classification.

3.3 BVP

To further improve the study results of Fritz et al., we suggest introducing heart-related signals into the psycho-physiological measurement of developers. Neuroscientific research suggests a link between cognitive processes, attention, emotions, and heart rate variability [7] [16]. In the study of Fucci et al. [10] the signals that led to the best classifier between prose comprehension and code comprehension were the heart-related signals, i.e., heart rate and heart rate variability [7] [16]. Although Nickel and Nachreiner argued that "HRV is an indicator for time pressure or emotional strain, not for mental workload" [19, p. 575], difficult tasks can lead to the former.
We suggest incorporating the Empatica E4 watch, which provided good results for Fucci et al. in their study. Also, this watch can perform an EDA so that we can get rid of the EDA device of the original study.

3.4 Introducing Programmer's Experience in the Study

Another significant change to the original study is the introduction of programmers' experience to our study.
In contrast to the proposal of Floyd et al. [8] and Fucci et al. [10], we propose to use the results of practical programming projects of a mandatory course at Saarland University as a measure of programming experience. Although the general GPA correlates with learning and academic skills [12], the GPA is slightly influenced by programming experience.
It is interesting to see if an experienced programmers still show the same biometric signals as a less experienced programmer. A negative correlation between received points and classifier accuracy implies that the more experienced one is, the less indistinguishable the psycho-physiological signals are concerning task difficulty. The more interesting case is probably be a non-significant, non-negative correlation. That would imply that even though one might be a good programmer, one could still classify the psycho-physiological signals as if one was a 'beginner'.

3.5 Study Design

Research Questions

- RQ_1: Can we accurately distinguish easy and difficult coding tasks by developers' biometric signals (fNIRS, heart-related signals, EDA, eye-related signals)?
- RQ_2: Is there a correlation between programmers' expertise and classification accuracy?

Subjects. In order to be comparable to the original study, we invite around 20-25 subjects - bearing in mind that some will not show up. In contrast to the original study, we do not invite software professionals but university students

from Saarland University that have completed their introductory programming courses 'Programming 1' and 'Programming 2'. As mentioned above, this is justified because of the introduction of programmers' experience in the study. Optimally the participation in this mandatory programming course should not date back further than two years. All participants should be right-handed, have normal or corrected-to-normal vision, and report no history of neuropsychological disorders. Subjects should be monetarily rewarded.

Data Capture. The data-capturing tools of the study comprise an eye-tracker as described in the original study. Since the study is eight years old, technology has developed, and eye trackers that are capable of delivering good results cost only a few hundred dollars. Those devices can also deliver "real-time data streams including gaze point, eye position, pupil diameter, user presence, and head pose" [4].
Furthermore, we include an EDA / Heart sensor in the study. Analogously to Fucci et al.'s [10] we propose to use the Empatica E4 [5], which has been motivated and described in the paper. Moreover, we use an fNIRS device, which at least covers the prefrontal cortex area. Although the price of a research-suitable fNIRS device starts from a lower five-digit number, one can rent such a device for a few sessions.

Experimental Tasks. Because the subjects should all be familiar with Java, they receive eight different code snippets of Java Code based on two challenges. They are based on the tasks that Fritz et al. [9] posed.
The first program creates two rectangles, assigns the coordinates of the corners, and prints the rectangles on the screen (but the developer cannot see the printing). The developer then has to state if the rectangles overlap. Fritz et al. used three incremental instances: practice version, use of non-mnemonic variables, interleaving assignments and randomized assignments. They argued that the program would stress the developer regarding spatial relations, visual object grouping, and working memory.
In another incremental instance, we include a method that receives a coordinate (x, y) that swaps x and y if a specific arithmetic formula is unequal to zero. This strains the developers regarding program flow comprehension and mathematical/logical thinking. Figures 5 and 6 show an example of a practice and a challenging task.

The second program of Fritz et al. creates four shapes and draws them on the screen in a particular order. From a set of possible answers, the developer has to decide which shapes were drawn and in what order. The researchers created seven instances that differed in the order between initialization and drawing, variable names (generic vs. mnemonic), iterable array of shapes vs. single shapes, separated functions vs. single function, and mathematical operator in control flow

[4] https://tech.tobii.com/products/eye-tracker-5l/
[5] https://www.empatica.com/en-eu/research/e4/

Fig. 5. Practice Task 1, Example

```
public class TaskOne{
    public void run() {
        C lB1 = new C(0, 0);
        C rB1 = new C(1, 0);
        C lT1 = new C(0, 1);
        C rT1 = new C(1, 1);//lB1:___, rB1:___,lT1:___,rT1:___

        C lB2 = new C(5, 0);
        C rB2 = new C(6, 0);
        C lT2 = new C(5, 1);
        C rT2 = new C(6, 1);//lB2:___, rB2:___,lT2:___,rT2:___

        R one = new R(lB1, rB1, lT1, rT1);
        R two = new R(lB2, rB2, lT2, rT2);
        draw(one); draw(two);
    }
    public void draw(R r) { /* foo */ }
}
```

Fig. 6. Hard Task 1, Example

```
public void run() {
    C a = new C(0, 0);
    C b = new C(1, 0);
    C c = new C(a.x, 2);
    C d = new C(0, b.y); // a:___, b:___,c:___,d:___
    C e = new C(1, c.y);
    C f = new C(5, 12);
    C g = new C(b.x, 1);
    C h = new C(6, 0);
    foo(f); // e:___, f:___,g:___,h:___
    R one = new R(a, c, g, f);
    R two = new R(e, d, b, h);
    draw(one); draw(two);
}
public void foo(C c) {
    if (((c.x - c.y) * 7) % 2 != 0) {
        int y = c.y;
        c.y = c.x;
        c.x = y;
    }
}
```

vs. no arithmetic task.
To increase the working memory workload, I include instances where shapes are built out of other shapes (e.g., three coordinates, the middle of a circle, a corner or a rectangle, and a corner of a triangle, constitute a new triangle or just a single line if one point lies in the line of the two other points).
In order to diversify the tasks and not concentrate on only a few static code snippets, we suggest mixing the different instances in tasks one and two. For example, one instance has, among other things, mnemonic variables but an iterable array of shapes and separate functions. In contrast, another instance has generic variables, shapes not stored in an array, and separated functions.

Experimental Procedure. The experimental procedure is very similar to the one Fritz et al. proposed in their study [9].
Before the experiment, developers get three example tasks to familiarize them with the format. Then we proceed with the actual experiment, in which tasks are displayed randomly. Before each task, the developers watch a fish tank video. As mentioned above, this fosters bodily relaxation and ease of the mind [6]. When the psycho-physiological signals return to their baseline, we take those signals as a reference to the signals collected during the task. Therefore, Fucci et al. proposed to use the last 30 seconds of the fish tank video [10].
In contrast to the original study, one must collect the heart rate, heart rate variability, and oxy-Hb baseline. After the task, the subjects fill out the NASA TLX survey that was motivated and explained above (3.1. Original Study).

Data Cleaning and Transformation. The Data Cleaning and Transformation does not significantly differ from the original study. Especially the eye-tracking part could be taken on without significant alteration. For the EDA and EEG, we adopt the proposals of Fucci et al. to extract features.
For fNIRS, we have to think of something own. If we used the normalizations of Nakagawa et al. [18], we get similar results as shown in Figure 4. Those values would be unreliable as features. They do rarely take the baseline of the subject into account. Nakagawa et al. just measured the signals during the task. Just taking the minimum as a reference to the current oxyHb is not sufficient.
Instead we propose to use the following formula:

$$normalized\ oxyHb = \frac{oxyHb - meanBase(s)}{max(s) - min(s)}$$

meanBase(s) quantifies the mean of oxyHb during the last 30 seconds of the fish tank video and serves as a reference to a subject's baseline. The domain of normalized oxyHb is probably $[-\epsilon_1; 1 - \epsilon_2]$ with $0 < \epsilon_1, \epsilon_2 << 1$ because meanBase(s) is probably closer to min(s) then max(s). As features, we extract the mean of *normalized oxyHb* and the variance. Furthermore we would include the minimum and maximum of both attention and meditation (fish-tank video) phases.

Outcome Measures. As the measure for the task difficulty, we adapt Fritz et al.'s proposal of using the NASA TLX survey, which the developers have to fill out after each task. The NASA TLX overall score and the subjective ranking (0-20) that the subjects have to fill out cause the task to be classified as simple or hard. With regard to Fritz et al.'s study, we assume that the vast majority of programs can be unambiguously assigned.

Machine Learning. For Machine Learning, we stick closer to Fucci et al.'s proposals than to the original study's ones. Although Naive Bayes' training 'can be updated on-the-fly' [9, p. 408], in Fucci et al.'s classification of psycho-physiological signals, Naive Bayes' underperformed. Their approach to trying out different classical machine learning models (including Neural Networks) could potentially bring up models that outperform a Naive Bayes classifier. Also, it is sensible to tune the model's parameters - for example, with a grid search. In contrast to Fritz et al.'s original study, we suggest presenting the models' results using BAC because the dataset is likely to be imbalanced.

4 Discussion: fMRI vs. Lightweight Biometric Sensors

By discussing and presenting state-of-the-art research in this paper, we have seen that LBS are more than appropriate for psycho-physiological classification problems in SE. In terms of pure classification accuracy, we saw that they could outperform fMRI, even though fMRI displays the brain's activity accurately to millimeters. With our study, we aim to improve the results of Fritz et al. even further. fNIRS and EEG have way better temporal resolution than fMRI and the other measures are apparently an excellent indicator of cognitive strain.
In this paper, the use of LBS has been motivated and praised. Nevertheless, there are, of course, certain limitations and disadvantages of LBS.
fMRI and LBS can be compared regarding classification accuracy, but they have very distinct capabilities and applications. With fMRI, researchers can examine the brain's activity in detail and thus can explain observable phenomena. For example, we can measure that the pupils widen when we look at tasks of higher difficulty [24]. fMRI can help us to understand why this is the case and what brain regions cause this effect to happen [22].
All in all, there is no better or worse - it always depends on the application. Joint research is, therefore, essential to explain mutual relationships.

5 Conclusion

Knowing if a developer faces difficulty with a particular task is crucial. We discussed why using lightweight biometric sensors supersedes other techniques for accurately classifying biometric responses to problems in SE by the example of code comprehension vs. prose comprehension classification. Then, we have seen that portable, lightweight, and comparably cheap devices have the capability to

classify the difficulty of tasks. We presented our own study based on a study by Fritz et al. [9], published in 2014, which aimed at accurately classifying simple vs. hard coding problems concerning psycho-physiological signals. Our study focuses on improving the original study by using fNIRS and heart-related sensors and enriches it by introducing programmers' experience. Further details of the original study are altered and optimized. Last, we discussed using lightweight biometric sensors over fMRI and why joint research with those two tools is indispensable.

References

1. Ahn, S., Jun, S.C.: Multi-modal integration of eeg-fnirs for brain-computer interfaces – current limitations and future directions. Frontiers in Human Neuroscience 11, 503–509 (2017)
2. Boucsein, W.: Electrodermal activity: Second edition. Springer Science and Business Media (2013)
3. Braithwaite, J.J., Watson, D.P.Z., Jones, R.O., Rowe, M.A.: Guide for analysing electrodermal activity & skin conductance responses for psychological experiments. CTIT technical reports series pp. 1017–1034 (2013)
4. Canento, F., Fred, A., Silva, H., Gamboa, H., Lourenço, A.: Multimodal biosignal sensor data handling for emotion recognition. In: SENSORS, 2011 IEEE. pp. 647–650 (2011)
5. Castelhano, J., Duarte, I., Ferreira, C., Duraes, J., Madeira, H., Castelo-Branco, M.: The role of the insula in intuitive expert bug detection in computer code: an fmri study. Brain Imaging and Behavior 13, 623–637 (2019)
6. Clements, H., Valentin, S., Jenkins, N., Rankin, J., Baker, J., Gee, N., Snellgrove, D., Sloman, K.: The effects of interacting with fish in aquariums on human health and well-being: A systematic review. PLOS ONE 14, e0220524 (07 2019)
7. Colzato, L., Steenbergen, L.: High vagally mediated resting-state heart rate variability is associated with superior action cascading. Neuropsychologia 106, 1–6 (2017)
8. Floyd, B., Santander, T., Weimer, W.: Decoding the representation of code in the brain: An fmri study of code review and expertise. In: 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE). pp. 175–186. IEEE Press (2017)
9. Fritz, T., Begel, A., Müller, S.C., Yigit-Elliott, S., Züger, M.: Using psycho-physiological measures to assess task difficulty in software development. In: Proceedings of the 36th International Conference on Software Engineering. p. 402–413. Association for Computing Machinery (2014)
10. Fucci, D., Girardi, D., Novielli, N., Quaranta, L., Lanubile, F.: A replication study on code comprehension and expertise using lightweight biometric sensors. In: 2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC). pp. 311–322. IEEE Press (2019)
11. Greco, A., Valenza, G., lanatà, A., Scilingo, E., Citi, L.: cvxeda: A convex optimization approach to electrodermal activity processing. IEEE Transactions on Biomedical Engineering 63, 797–804 (2016)
12. Grove, W., Wasserman, T., Grodner, A.: Choosing a proxy for academic aptitude. Journal of Economic Education 37, 131–147 (2006)

13. Hart, S.G., Staveland, L.E.: Development of nasa-tlx (task load index): Results of empirical and theoretical research. In: Hancock, P.A., Meshkati, N. (eds.) Human Mental Workload, Advances in Psychology, vol. 52, pp. 139–183. North-Holland (1988)
14. Huang, Y., Liu, X., Krueger, R., Santander, T., Hu, X., Leach, K., Weimer, W.: Distilling neural representations of data structure manipulation using fmri and fnirs. In: 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE) (2019)
15. Ikutani, Y., Uwano, H.: Brain activity measurement during program comprehension with nirs. In: 15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). pp. 1–6. IEEE Press (2014)
16. Jönsson, P.: Respiratory sinus arrhythmia as a function of state anxiety in healthy individuals. International journal of psychophysiology : official journal of the International Organization of Psychophysiology 63, 48–54 (2007)
17. Krueger, R., Huang, Y., Liu, X., Santander, T., Weimer, W., Leach, K.: Neurological divide: An fmri study of prose and code writing. In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering. p. 678–690. Association for Computing Machinery (2020)
18. Nakagawa, T., Kamei, Y., Uwano, H., Monden, A., Matsumoto, K., German, D.M.: 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. p. 448–451. Association for Computing Machinery, New York, NY, USA (2014)
19. Nickel, P., Nachreiner, F.: Sensitivity and diagnosticity of the 0.1-hz component of heart rate variability as an indicator of mental workload. Human factors The Journal of the Human Factors and Ergonomics Society 45, 575–590 (2003)
20. Parnin, C.: Subvocalization - toward hearing the inner thoughts of developers. In: 2011 IEEE 19th International Conference on Program Comprehension. pp. 197–200 (2011)
21. Peitek, N., Siegmund, J., Apel, S., Kästner, C., Parnin, C., Bethmann, A., Leich, T., Saake, G., Brechmann, A.: A look into programmers' heads. IEEE Transactions on Software Engineering (4), 442–462 (2020)
22. Siegle, G., Steinhauer, S., Stenger, V., Konecky, R., Carter, C.: Use of concurrent pupil dilation assessment to inform interpretation and analysis of fmri data. NeuroImage 20, 114–124 (2003)
23. Siegmund, J., Kästner, C., Apel, S., Parnin, C., Bethmann, A., Leich, T., Saake, G., Brechmann, A.: Understanding understanding source code with functional magnetic resonance imaging. In: Proceedings of the 36th International Conference on Software Engineering. p. 378–389. Association for Computing Machinery (2014)
24. Wel, P., Steenbergen, H.: Pupil dilation as an index of effort in cognitive control tasks: A review. Psychonomic Bulletin Review 25, 2005–2015 (2018)
25. Wilcox, T., Biondi, M.: fnirs in the developmental sciences. Wiley interdisciplinary reviews. Cognitive science 63, 263–283 (2015)