Assessing Task Difficulty in Software Testing using Biometric Measures

Daryl Camilleri
Department of Computer Science
University of Malta
daryl.camilleri.11@um.edu.mt

Chris Porter
Department of Computer Information Systems
University of Malta
chris.porter@um.edu.mt

Mark Micallef
Department of Computer Science
University of Malta
mark.micallef@um.edu.mt

In this paper, we investigate the extent to which we could classify task difficulty in the software testing domain, using psycho-physiological sensors. Following a literature review, we selected and adapted the work of Fritz et al. (2014) among software developers, and transposed it to the testing domain. We present the results of a study conducted with 16 professional software testers carrying out predefined tasks in a lab setting, while we collected eye tracking, electroencephalogram (EEG) and electrodermal activity (EDA) data. On average, each participant took part in a two-hour data-collection session. Throughout our study, we captured approximately 14Gb of biometric data, consisting of more than 120 million data points.

Using this data, we trained 21 naive Bayes classifiers to predict task difficulty from three perspectives (by participant, by task, by participant-task) and using the seven possible combinations of sensors. Our results confirm that we can predict task difficulty for a new tester with a precision of 74.4% and a recall of 72.5% using just an eye tracker, and for a new task with a precision of 72.2% and a recall of 70.0% using eye tracking and electrodermal activity. The results achieved are largely consistent with the work of Fritz et al. (2014). We conclude by providing insights as to which combinations of sensors would provide the best results, and how this work could be used to enhance well-being and workflow support tools in an industry setting.

1. INTRODUCTION

Software testing is a hard problem. Vanitha and Alagarsamy (2019) define it as “one of the five main technical activity areas of the software engineering life-cycle that still poses substantial challenge”. Contrary to being a simple process of checking a sample of runs, software testing encompasses various intricate challenges and enforces a mixture of activities and techniques. In a survey of 400 job adverts across 33 countries, Florea and Stray (2019) identified as many as 40 distinct skills that companies look for when hiring testers. They also identified a multitude of requirements pertaining to specific tools, development processes and domain specific knowledge.

In this context, and considering the fact that testers tend to hail from a variety of training backgrounds (Micallef et al. 2016), it would be useful to be able to gauge the difficulty of testing tasks as experienced by individual testers while carrying out that task. Work to achieve this using psycho-physiological measures has been done in other fields and in software development. However, to the best of our knowledge, this has not been carried out in the context of software testing.

In this paper, we seek to understand how psycho-physiological measures could be leveraged to assess the difficulty experienced by software testers carrying out testing tasks. In particular, we build on the work by Fritz et al. (2014), who developed a novel approach for classifying the difficulty of software development tasks using several psycho-physiological sensors in such a way that allowed them to detect developers who were struggling and experiencing difficulty. We reproduce these experiments with two differences: (1) recruiting professional testers as participants; and (2) having participants carry out testing activities as tasks for the experiment.

In particular, through this study, we attempt to answer the following research questions:

RQ1 To what extent could we accurately predict task difficulty of software test engineers using psycho-physiological measures from EEG, EDA and eye-tracking sensors?

RQ2 What combination of psycho-physiological sensors and associated characteristics better predict whether a task is easy or difficult?

RQ3 To what extent could we assess task difficulty while software test engineers are working using psycho-physiological measures?

RQ4 To what extent are the results achieved for software testing tasks comparable to those of existing literature in the field of software development?
To this end, we recruited 16 professional software testers and asked them to carry out eight specifically designed software testing tasks. As they did this, we monitored their electroencephalographic (EEG) activity along the forehead, electrodermal activity (EDA) and eye movements using an eye tracker. This produced 14Gb of raw data, which was then cleaned, transformed and analysed towards answering the research questions above. Through this work, we make the following contributions to knowledge:

1. An investigation into what combinations of biometric measures can be used to classify task difficulty in the testing domain;

2. A reproduction of the approach used by Fritz et al. (2014) to shed more light on the validity of their approach when applied to a different domain;

3. A rich dataset of eight tasks, plus psycho-physiological measures from 16 two-hour data-collection sessions, and analysis scripts for reproduction or independent analysis.

The rest of this paper is organised into four further sections. More precisely, in Section 2, we provide the reader with an overview of existing work in this field before we discuss our specific research methodology in Section 3. In Section 4 we present the results and discuss them in light of our research questions. Finally, in Section 5, we make our concluding arguments and propose future steps in this area.

2. RELATED WORK

Recent years have seen an increased use of biometric data in understanding challenges faced by practitioners in the field of software engineering. Such challenges are mainly driven by the rapid advancements in technology, which have affected the progress, performance and workflow of software engineers. Psycho-physiology has proved promising for understanding (1) cognitive processes such as attention and language comprehension (Siegmund et al. 2017); (2) emotions, such as stress (Ostberg et al. 2017); and (3) frustration (Wróbel 2013). Earlier studies of psychological aspects could only be done through holistic processes, which limited the accuracy of the analysis.

To overcome such challenges, researchers have carried out extensive studies with the objective of understanding some of the difficulties that software engineers encounter during their day-to-day tasks. Researchers explored the use of various human-computer interaction (HCI) techniques, in particular, the movement of eyes and the related emotions, and concluded that such methods provide reliable insights into the problems generally experienced by practitioners. A study conducted by Shaw (2004) on the role of emotions in the workplace, among 12 students working on a semester-implementation project, found that self-reported emotions can change drastically within 48 hours. Correspondingly, Müller (2015) conducted a separate study on developers’ emotions. The survey indicated that software developers experienced a series of negative and positive emotions while programming. Moreover, he found frustration to be the main negative feeling that affected the performance of software developers. He also noted that, with some programmers, negative feelings such as stress had a positive impact on performance.

Various other studies were undertaken to investigate whether emotions might have any impact on the performance of developers. While some of these examined the types of emotions that developers experience (Wróbel 2013)(Shaw 2004), other studies focused on the impact that such emotions could have on the performance of the programmer (Graziotin et al. 2013)(Khan et al. 2011). At the same time, other studies sought to examine whether such emotions could be predicted using interactive logs (Carter and Dewan 2010)(Khan et al. 2013). Using this literature, Müller and Fritz (2015) discovered that developers often face frustration, happiness, anger, and enthusiasm. Khan et al. (2011), on the other hand, found that emotions affect performance in terms of debugging and programming. Therefore, these papers indicate that emotions could be qualified through the output of the developers.

Further studies were conducted to determine the difficulties that developers tend to experience. In one study, Carter and Dewan (2010) examined whether interaction logs could be possibly used to predict emotions of the developer. This was successful, essentially indicating where the programmer was getting stuck. Crosby and Stelovsky (1990), explored the concept of fixed eyes to indicate difficulties. In this case, researchers found that experienced developers spent more time on complex statements, while the inexperienced programmers spent more time on the comment areas. Moreover, Parnin (2011) also found that a measure of sub-vocal utterances could suggest that a programmer might be experiencing difficulties.

All of the above contribute to the premise that it is actually possible to detect whether a programmer is stuck or is experiencing difficulties. Indeed, the observations we made in the course of monitoring the test engineers also pointed in this direction.
Biometric sensors were then explored in a different study by Fritz et al. (2014) to determine challenges in small coding problems. Fritz et al. (2014) have shown that psycho-physiological measures could be used to assess task difficulty in software development. They demonstrated that new software engineering tools could be developed towards detecting code fragments with which developers struggle, and to help prevent interruptions during particularly difficult tasks. To do so, they used eye, brain (EEG) and skin (EDA) measures to determine task difficulty in both a post hoc scenario and as developers were carrying out tasks.

We decided to adapt the methodology proposed in Fritz et al. (2014) by replicating it as faithfully as possible when applying it to the testing domain.

3. RESEARCH METHODOLOGY

In this section, we report on the design and execution of the methodology from set-up through to data analysis. The data-collection methodology centred on a laboratory experiment, for which professional software test engineers were recruited and asked to carry out 10 software testing tasks while wearing biometric sensors. In order to minimise threats to validity, we opted to follow the approach designed by Fritz et al. (2014) in their work on measuring software-development task difficulty.

3.1. Participant Recruitment

Participants were recruited from a pool of professional software test engineers living within a 40km distance from our labs, who expressed an interest in participating in this study. Potential participants were contacted through social media groups, and partner companies who regularly help us with such endeavours. A screening questionnaire selected 20 of these candidates who had at least two years of software testing experience, knew how to develop in C# or Java, and were familiar with currently standard testing frameworks, such as Selenium and SpecFlow.

Throughout the screening process, individuals who needed to wear bifocal or trifocal glasses could not be shortlisted, since such glasses are known to cause problems when using eye-tracking technology.

3.2. Data Collection

The study data was recorded using three psycho-physiological sensors, namely: EEG to detect human brain activity, EDA to detect attention arousal, and eye tracking to understand the ebb and flow of participant attention. We also recorded the participants’ think-aloud narrative, and recorded a video and a screen capture of the entire experiment. The participants were asked to complete a pre-experiment questionnaire, a written NASA-TLX (NASA Task Load Index) survey after completing each of the experimental tasks, and a post-experiment questionnaire after the entire experiment that required them to rate each of the tasks by relative perceived difficulty. During the experiment, we also took hand-written notes.

With regard to sensor selection, we decided to balance accuracy, cost and data clean-up efforts, while making the experience as non-invasive as possible for participants. This led us to the following choices: the Muse headband for collecting EEG data (Figure 1a); the BioSignalsPlux EDA sensor for EDA data (Figure 1b); and the Eye Tribe eye tracker for eye tracking data (Figure 1c).

3.3. Task Design

Each participant was required to work through 10 tasks: 2 practice questions and 8 tasks, from which data was collected. Following feedback from initial pilot studies, each task was designed to take a short amount of time (3 to 5 minutes). Tasks revolved around four categories of testing activities: (1) test analysis and design; (2) test implementation; (3) bug finding; and (4) adequacy analysis. During each task, participants were required to read a concise specification or a short snippet of C# code presented on the screen in a Google form. They were then asked to verbally explain how they would
complete a specific task related to what they were observing on the screen. For example, after seeing a specification for a feature, participants were asked to outline how many tests would be required to test an implementation of that specific feature.

In order to reduce susceptibility to learning effects, we chose to vary the domain of each question and to change the order in which these tasks were presented to participants.

3.4. Data Collection Procedure

The data collection procedure consisted of three stages: (1) briefing and set-up; (2) data collection; and (3) debriefing.

3.4.1. Briefing and Set-up

When each participant first entered the lab, they were asked to fill out a consent form and a pre-questionnaire requesting demographic information. We then explained each of the devices and started by attaching the EDA patches to the pointing and ring fingers of the participants’ non-dominant hand. This was usually the opposite hand that the subject used to move the cursor on the screen using a mouse. We then connected the EDA sensor to the data recording computer via Bluetooth and verified that a signal was being received to the live monitoring display. Since the EDA sensor works by detecting electrodermal skin response events, it sometimes fails to operate if the participant does not perspire. For those few participants who did not register any signal, we asked them to walk up and down a flight of stairs or to take a walk around the laboratory building to cause them to perspire a little. This sufficed for all participants, and so we were eventually able to record the EDA signal of each one.

3.4.2. Data Collection

Prior to the first task, and in between each one, participants were asked to watch a two-minute, calming video of fish swimming in a fish tank, and were advised to rest their minds. This helped the participants enter a fresh psycho-physiological state prior to starting each task. Participants then went through the sequence of tasks, with researchers placing appropriate markers on data to delineate the start and end of a task. After each task, participants were asked to rate the task from 1 to 20 along six dimensions: mental demand, physical demand, temporal demand, performance, effort and frustration.

3.4.3. Debriefing

After participants completed their final task, we turned off the recordings and removed the biometric sensors. We then provided participants with a short description of each task to refresh their memory, and asked them to complete: (1) the last part of the NASA-TLX survey; and (2) a short exit survey providing feedback about their experience. After giving them the opportunity to ask any questions they might have had, the data collection session was concluded.

3.5. Data Analysis

We collected psycho-psychological measurements for a total of 128 tasks. A list of measurements used from each sensor is presented in Table 1. For every completed task, we took note of the completion time and also collected the NASA-TLX score and the difficulty rating provided by the participant at the end of the study. Video recordings and the think-aloud protocols were used to correct any unintended errors made during data analysis, which we outline in the following subsections.

3.6. Data Cleaning and Transformation

Data collected from biometric sensors is notoriously noisy and contains a considerable amount of invalid data that must be cleaned before it could be analysed. Pandas\(^1\) is one of the most widely used open-source data-science libraries for the Python programming. In our study, Pandas and other data-science libraries, such as Numpy, SciPy and

\(^1\)https://pandas.pydata.org
Matplotlib, were utilised in cleaning, transformation, and making sense of our datasets.

### 3.6.1. Brain-related Metrics

Data from the Muse EEG headband provided four channels of recording, namely TP9, AF7, AF8, and TP10, that are referred to at a 5th channel situated at FPZ. Muse yields a signal sampled at 500Hz. Using a mobile app\(^2\), the raw signal data was streamed from the Muse EEG system straight to our phone. A notch that filters frequency at 60Hz was applied by default to all raw values, in order to filter and remove any unwanted noise in the signal.

The absolute band power for each of the five familiar brainwave frequency bands: delta (\(\delta\)) (1–4 Hz), theta (\(\theta\)) (4–8 Hz), alpha (\(\alpha\)) (7.5–13 Hz), beta (\(\beta\)) (13–30 Hz) and gamma (\(\gamma\)) (30–44 Hz) were computed independently by the Mind Monitor application, based on the logarithm of the power spectral density (PSD) of the EEG data for each channel. Since each person possesses a distinctive power spectrum distribution, using the procedure outlined by Fritz et al. (2014), we calculated the ratio of each frequency band with one another to compare values between individuals. Additionally, we computed \(\text{beta}(\beta) / (\text{alpha}(\alpha) + \text{theta}(\theta))\) and \(\text{theta}(\theta) / (\text{alpha}(\alpha) + \text{beta}(\beta))\) as extra measures of task difficulty, as recommended by Kramer (1991) and Lee (Lee and Tan 2006).

The Muse headband is placed on an individual’s forehead. Thus, the sensors could receive motor signals of the face, such as blinking, brow furrowing and eyebrow motions. These motor activities produce a low frequency, high amplitude signals, which are easily distinguishable from others related to neuronal activity. According to Brookings et al. (1996), an individual’s blinking rate reduces significantly when tasks becomes harder. Taking advantage of the eye blinks detected by Mind Monitor, we computed the number of blinks per minute and then deducted the baseline number of blinks identified throughout the participant’s previous one-minute viewing of the fish-tank video before starting the task.

\[
\text{Valence} = \left((\alpha AF8/\beta AF8) - (\alpha AF7/\beta AF7)\right)
\]

\[
\text{Arousal} = \left((\alpha AF7 + \alpha AF8)/(\beta AF7 + \beta AF8)\right)
\]

Finally, we calculated valence and arousal from the alpha and beta band powers. Valence can be interpreted as a personal feeling of happiness or unhappiness, while arousal is the subjective transition from feeling calm to excited (Barrett 1998). The latter is distinguished by a higher beta power and consistency in the parietal lobe of the human brain, and by lower alpha activity. Beta waves are correlated with an alert or excited state of mind, while alpha waves are predominantly present in a state of relaxation. Therefore, the beta/alpha ratio is a plausible indicator of the excitation state of a person (Blaieh et al. 2013). Furthermore, the prefrontal lobe serves as an essential role in regulating emotion and conscious experience (Blaieh et al. 2013). Inactivation of the left frontal lobe designates negative emotion, whereas inactivation of the right frontal lobe reveals a positive emotion. Therefore, to determine the valence, as inspired by Ramirez et al. (2015), we computed the levels of the two cortical hemispheres. For our analysis, we calculated the minimum, maximum, mean, median and standard deviation for both valence and arousal.

### 3.6.2. Skin-related Metrics

Electrodermal activity (EDA) readings were obtained by measuring the voltage between two BioSignals electrodes placed on the participant’s fingers, across which a low-level current was applied. EDA signals comprise of two main components, namely: a low-frequency tonic signal that varies in minutes, and a higher-frequency phasic signal which rises in 1 to 2 seconds and falls in between 2 to 6 seconds (Fritz et al. 2014).

The tonic element of the EDA signal, otherwise known as skin conductance level (SCL), is mostly utilised to reflect general changes and to measure autonomic arousal. The phasic component reverberates responses that are grounded by external stimuli (Schmidt and Walach 2000). To clean and extract the tonic signal from our EDA data, we used a 4th-order low-pass filter set to 0.05Hz. More specifically, we used the Butterworth filter, considering its ability to

---

\(^2\)https://mind-monitor.com

| Sensors | Psycho-physiological Measurements |
|---------|----------------------------------|
| Min Valence, Max Valence | Mean Valence, StDev Valence | StDev Valence, Min Valence, Max Valence |
| Min Arousal, Max Arousal | Mean Arousal, StDev Arousal |
| Saccade Rate Per Minute | Fixation Rate Per Minute | Pupil Fixation Size, Pupil Saccade Size, Fixation Size, Forehead Size |
| Alpha Tonic Duration, Median Tonic Duration, StDev Tonic Duration | Alpha Phasic Duration, Median Phasic Duration, StDev Phasic Duration |
| EDA (T) | Record Fixation Rate Per Minute, StDev Fixation Rate Per Minute |
| Eye-Tracking (T) | Record Pupil Fixation Size, Pupil Saccade Size, Fixation Size, Forehead Size |

| TABLE 1: Psycho-physiological measurements used from each of the three sensors |
|---------------------------|
| **Valence** | **Arousal** |
| \(\text{Valence} = \left((\alpha AF8/\beta AF8) - (\alpha AF7/\beta AF7)\right)\) | \(\text{Arousal} = \left((\alpha AF7 + \alpha AF8)/(\beta AF7 + \beta AF8)\right)\) |
generate the flattest possible frequency response. In view of the substantial noise in the signal, we applied a 350-point moving average filter. This worked by averaging several points from the input signal to produce each point in the output signal (Smith 1997). Similarly, we applied a high-pass version of the same Butterworth filter to extract the phasic signal from our EDA data. Next, we measured the tonic SCL value relative to a recent baseline value. We computed it by deducting the mean SCL of the EDA signal as the participant watched the last minute of the fish-tank video from the one measured while the participant performed each task.

It would be relevant to add that researchers tend to concentrate on the amplitudes and latency of the phasic bursts in association to the stimulus onset, when examining changes in the EDA signal in response to sensory stimuli such as videos, sounds, and images (Fowles et al. 1981)(Salimpoor et al. 2009)(Anders et al. 2004). When there are notable changes in EDA activity in response to stimuli, it is referred to as an event-related skin conductance response (ER-SCR). These responses, also known as EDA peaks, can give information concerning emotional response to a stimulus. Other EDA activity peaks that are not related to the stimulus presentation are known as non-specific skin conductance responses (NS-SCR).

In our analysis, SCRs were detected by finding two consecutive zero-crossings, from positive to negative and from negative to positive (Kim et al. 2004). The SCR amplitudes were obtained by computing the maximum value amidst the two zero-crossings. While NS-SCRs always occur, the only external stimuli that the participants could have encountered must have come from what they read through the experimentation tasks. Therefore, the likely number of ER-SCRs was computed by deducting the number of peaks detected during the experimental tasks from the previous one-minute time period while they were watching the fish-tank video. The mean values of the SCR amplitudes and duration, the area under curve (AUC), number of SCR occurrences, and peak amplitudes in the signal were extracted from the EDA as features and standardised by time. SCRs that were detected with an amplitude lower than 10% of the maximum SCR amplitude were removed.

3.6.3. Eye-gaze-related Metrics

Prior to the implementation of our gaze-based measures, eye-movement data was first extracted in a preprocessing step to remove invalid data points and heartbeat measures, as identified by the Eye Tribe eye tracker. Eye Tribe provides various measures in each frame, including the x and y coordinates, the size and the centre of the pupil of both eyes. As with the findings by Fritz et al. (2014), we observed that some of the first pupil-size measurements of a fixation occurring after a blink were abnormally bigger than the subsequent one. Pupils tend to widen in low-light situations to allow more light into the eyes, thus compensating for the insufficient lighting. We removed this anomaly by eliding each of the pupil size measurements after a blink. An indication of cognitive load by the pupil's diameter can be traced back to Hess and Poll (1960). They demonstrated that there is a strong correlation between task difficulty and pupil dilation. Pupil sizes enlarge typically up to 0.5mm under cognitive load, especially when reading complex material (Fritz et al. 2014). We found these events by calculating the number of peaks in the pupil size signal where the pupil size increased by 0.1mm, 0.2mm and 0.4mm over the baseline counts of the previous one-minute fish-tank video.

Eye movements can be categorised into saccades (which occur in response to the performance of large eye movements) and fixations (which arise in response to looking at a given point). By combining saccade and fixation information, we could form a heat map of the stimuli regions that generated the most interest among the participants. Following the process of Fritz et al. (2014), we extracted the number and duration of each participant's eye fixations and saccades to gain insight into the way their eye movement was influenced while reading material with different mental demands. Since every participant worked at their own pace, we normalised most of these measures by time to make them comparable between participants.

3.6.4. Outcome Measures

The NASA-TLX (Hart and Staveland 1988) and a subjective ranking of tasks were the two outcome measures used in our tasks. The NASA-TLX form was filled by a member of the research team after each task, whereas the subjective ranking of tasks based on post hoc discernment was filled by participants at the end of the experimental session in the form of a post-questionnaire. NASA-TLX is regularly used as a subjective measure for evaluating cognitive load (Haapalainen Ferreira et al. 2010). After every task, the participant provided a difficulty rating on six 20-point scales: Performance (good/poor), Mental Demand (low/high), Physical Demand (low/high), Temporal Demand (low/high), Effort (low/high) and Frustration (low/high). Every scale was defined for the participant using Hart and Staveland's instructions (Hart 1986) along with an explanation of their meaning before the experiment.

After finishing all of the experimental tasks, the participants were instructed to reflect on each
possible pair of scale names on the NASA-TLX survey and to encircle the scale name that was the more significant to their workload. We computed the overall final score to be the summation of the products of each rating and the sum of the number of times it was chosen as more important - and then divided by fifteen, which is the number of possible pairs in the list of comparison boxes.

While the NASA-TLX score gave us an insight into the mental workload of each participant for every task, we were also interested in the participant’s summative assessment of task difficulty. To this end, we requested every participant to rank the experimental tasks from easy to hard on a scale of 1 to 10. Additionally, participants were asked to write down some comments to clarify their rating and how they thought about the task difficulty (‘I found the experiment to be very interesting. I think the questions were clear and I was given a very good brief before starting the experiment. The hardest part for me was that I couldn’t write and had to explain the scenarios verbally.’).

Next, to simplify the prediction process for our machine learning algorithm, we nominalised each task difficulty rank as ‘easy’ or ‘difficult’. Low ranks were labelled ‘easy’, whereas high ranks were marked ‘difficult’. For scores in the middle, we found out that in all but 12 cases out of 128, participants clearly expressed an easy or difficult gap in their additional comments based on their perception of the task difficulty. For the other 12 cases, we nominalised the task difficulty rank using the NASA-TLX score, as the score was completely unambiguous. Our set of data consisted of 83 easy tasks and 45 difficult tasks once we finished nominalising the task difficulty ranking.

To validate our set of nominalised ranks, we verified that there was a significant correlation between the task difficulty ranks and the NASA-TLX scores. Using the same techniques as Fritz et al. (2014), we verified our data using Spearman’s rank order correlation and a chi-square test. A Spearman’s correlation confirms that the participants’ task difficulty ranking is correlated with the NASA-TLX scores ($r[128] = 0.547$, $p$-value < 0.001). Likewise, the easy/difficult boolean of the NASA-TLX was further correlated to the easy/difficult boolean of the task difficulty rankings ($x^2(1, 128) = 84.787$, $p$-value < 0.001).

As a final step, since time is likewise a proxy for difficulty (Haapalainen Ferreira et al. 2010), we performed another Spearman correlation between the task difficulty ranks and the task completion time ($r[128] = 0.440$, $p$-value < 0.001). This also confirmed that there was a correlation between

### 3.6.5. Data Classification using Machine Learning

We used three types of predictions to stratify our datasets into test-and-training sets, namely: by participant, by task and by participant-task. In practice, the by-participant classifier is the most useful one, as it could be trained on a small set of individuals while still accurately predicting task difficulty on other persons working on new tasks. On the other hand, the by-task classifier works well when trained on a group of individuals performing new tasks, as it can predict task difficulty of any of those individuals working on a new task. Lastly, when trained on a group of people doing tasks, the by-participant-task classifier can predict task difficulty as perceived by any of those people working on a new task that other people already did.

As regards the data classification algorithm, we considered naïve Bayes, J48 decision tree and support vector model, and agreed with Fritz et al. (2014) that naïve Bayes was the ideal choice for the purpose of this study. This was mainly due to the ease with which the training set could be updated on the fly, enhancing its performance as it adapts to its user (Fritz et al. 2014). Next, in order to create a classifier that could be used while a test engineer would be working on a task, we divided our datasets into several time windows. As the test engineer made progress in the task at hand, the classifier would adapt to any changing physiological conditions before the end of the task. Datasets were split into sliding time windows of 10 seconds, with 5-second offsets.

Finally, it was necessary to decide upon the features to be used to train our model. As in Fritz et al. (2014), we chose to experiment with measurements extracted from every combination of the three biometric sensors, to a total of seven possible sets of features. Measurements that correlated almost perfectly with measurements we left in were removed to ensure the right performance for naïve Bayes.

### 4. RESULTS AND DISCUSSION

In this section, we present and discuss our results with reference to the research questions posed in Section 1. All data, support material and analysis scripts are available on our OSF repository.\(^3\)

---

\(^3\)https://osf.io/vk8rh/?view_only=ad9fc75ab08744b39346da52503b600f
Following initial screening of potential participants, we selected 20 individuals, four of whom did not make it to the lab. A further participant felt unwell towards the end of the experiment but the session still provided usable data.

On average, it took the participants 3 minutes 38 seconds to complete each task, with a standard deviation of 57 seconds. The fastest task was completed by a participant in 1 minute 23 seconds, whereas the slowest task was completed in 6 minutes 18 seconds. Each participant took approximately 2 hours to complete the entire experiment. No difficulties with equipment malfunction or measurement anomalies were encountered.

4.1. Task Difficulty Classification (RQ1)

To evaluate whether we could classify task difficulty as 'easy' or 'difficult' using psycho-physiological measures (RQ1), we trained 21 Bayes classifiers to categorise task difficulty (1) by participant; (2) by task; and (3) by participant-task. We used all seven sensor combinations (see Table 2) and utilised a leave-one-out strategy to create an exhaustive set of test and training folds.

4.1.1. ‘By Participant’ Classification

This group of classifiers was trained on data grouped by participant over all tasks, with the intent of classifying task difficulty for an individual participant at a point in time. Here, the best performance was achieved by using the eye-tracking sensors in isolation with a precision of 74.4%, recall of 72.5% and F-measure of 73.0%.

4.1.2. ‘By Task’ Classification

This group of classifiers was trained on data grouped by task over all participants. Therefore, we were testing a classifier’s ability to deliver a verdict on data from a new participant who was attempting a task that had been attempted by others in the training set. Here, the best performance was achieved by using the eye-tracking sensors in combination with EDA sensors with a precision of 72.2%, recall of 70.0% and F-measure of 70.6%. It is worth noting that this only marginally outperformed the classifier that only utilised eye-tracking data (71.7% precision, 70.0% recall and 70.5% F-measure).

4.1.3. ‘By Participant-Task’ Classification

This group of classifiers was trained on data grouped by participant and task over data from all participants and all tasks. The best performance here was again achieved by eye-tracking sensors in isolation, with a precision of 73.3%, recall of 71.7% and an F-measure of 72.2%.

4.2. Sensor Effectiveness (RQ2)

We were also interested in investigating which combination of the three biometric sensors available to us could best predict task difficulty (RQ2). As could be seen in Table 2, the best results were achieved with the influence of data from eye-tracking sensors. In two groups of classifiers (‘by participant’ and ‘by participant-task’), it was eye-tracking data in isolation that provided the best results, while in the ‘by task’ group, it was the combination of eye-tracking and EDA data (albeit marginally). If one considers classifiers that used one type of sensor in isolation, eye-tracking sensors provided more than 70% F-measure in each case, while other sensors achieved an F-measure of around 55% (arguably close to random classification). Furthermore, we have noted that whenever EEG was combined with eye-tracking data, it significantly deteriorated results which eye-tracking had achieved in isolation.

4.2.1. Influence of Individual Features

To investigate whether we could improve the accuracy of our classifiers, we used Weka’s CfsSubsetEval algorithm to analyse and retain features of each sensor combination that highly correlated with the outcome variable, and which have low intercorrelation with other features. In some cases, this produced better performance on our data. For instance, when predicting a new participant, the F-measure increased from 58.4% to 70.5% using eye-tracking and EEG data. On the other hand, when using EDA and EEG, the F-measure dropped from 56.4% to 51.9%. To examine whether there might be any significant differences between the original and fewer set of features considered by the CfsSubsetEval algorithm, we conducted several ANOVA tests on the Weka output results. As with the results obtained by Fritz et al. (2014), our tests failed to show any significant differences between the two groups. A p-value of 0.606 confirmed that
the differences we observed in our results may have been due to some artefacts in our dataset.

4.3. Classifying using Time Windows (RQ3)

Whereas in RQ1 we were concerned with identifying whether or not a test engineer found a task difficult after the fact, in RQ3 we attempted to classify the task difficulty at the time when the task was being carried out. To do this, we built and trained a set of classifiers using sliding time windows. We experimented with different window sizes ranging from 10 to 100 seconds and eventually opted to use 10-second time windows with a 5-second offset.

The results led us to make two observations. Firstly, the best results using sliding time windows are 15% to 17% less effective than the best results achieved when classifying over all collected data. Secondly, contrary to the results presented in Table 2, sliding time windows seem to provide better results when they combine multiple types of sensors. In fact, the best results achieved with sliding time windows involved combining data from all three sensors. Having said that, the combined use of eye tracking, EDA and EEG resulted in approximately 2% improvement in precision and recall over the use of eye tracking alone. Given the more intrusive nature of EDA and EEG sensors, one needs to consider whether this improvement is truly justified.

4.4. Consistency with Previous Work (RQ4)

As stated in Section 2, the methodology used in this work was adapted from work by Fritz et al. (2014), who focused their efforts on software development tasks. Although the results demonstrated that it is indeed possible to classify task difficulty in the testing domain using this methodology, it would be worth looking closer at the similarities and differences between our results and those of Fritz et al. (2014). From a purely empirical perspective, the results from the testing domain outperformed results from software development in ‘by participant’ classification (+12% F-Measure) and ‘by participant-task’ classification (+6% F-measure). Conversely, our work under-performed when it came to the ‘by task’ classification (-4% F-measure). However, we believe that such a comparison could have limited utility and would be difficult to explain, since it could be conditioned by a number of factors other than the domain itself (e.g., different equipment, different lab setting, the relatively small number of participants, etc.). It is worth noting that the best performance in all cases was achieved when an eye-tracking component was used. This is particularly encouraging because eye-tracking technology is arguably the least intrusive, and hence offers potential should researchers consider taking such experiments out of the lab and into an industry setting. Therefore, we are suggesting that this work further validates the approach proposed by Fritz et al. (2014).

5. CONCLUSION

In this work, we set out to investigate the possibility of classifying task difficulty in the software testing domain using psycho-physiological measures. Following a review of literature, we opted for the approach taken by Fritz et al. (2014) and, while replicating the methodology as faithfully as possible, adapted it to the software testing domain. Our results demonstrate that it is indeed possible to classify task difficulty with a high level of confidence. Furthermore, this was done using low-cost, off-the-shelf biosensors to develop a set of classifiers that could classify task difficulty with accuracy.

The results also suggest that it would suffice to do away with EEG and EDA sensors, and to focus solely on eye-tracking technologies. The implications here are encouraging in terms of the applicability of the approach in industry settings. For instance, by installing eye trackers on the equipment used by software testers, it could be possible for such systems to raise alerts when individuals are experiencing difficulty. It could also help improve workflow by minimising interruptions. For example, the technology could suppress notifications from e-mail and communication tools, if it realises that the person is currently concentrating on a difficult task.

5.1. Future Work

This work presents a number of opportunities for future work in (1) tool support for employee productivity and well-being; (2) taking such experiments outside the confines of lab studies; and (3) exploring the possibility using classification techniques other than Bayes classifiers. In particular, the resurgence in machine learning and data science over the last decade has brought about many new techniques that may not have been available when Fritz et al. (2014) were developing their methodology.

REFERENCES

Anders, S., Lotze, M., Erb, M., Grodd, W. and Birbaumer, N. (2004), ‘Brain activity underlying emotional valence and arousal: A response-related fmri study’, Human brain mapping 23, 200–9.

Barrett, L. (1998), ‘Discrete emotions or dimensions? the role of valence focus and arousal focus’, Cognition and Emotion 12, 579–599.

Blaiech, H., Neji, M., Wali, A. and Alimi, A. (2013), Emotion recognition by analysis of eeg signals, pp. 312–318.
Brookings, J., F. Wilson, G. and R. Swain, C. (1996), ‘Psychophysiological responses to changes in workload during simulated air traffic control’, Biological psychology 42, 361–77.

Carter, J. and Dewan, P. (2010), Design, implementation, and evaluation of an approach for determining when programmers are having difficulty, pp. 215–224.

Crosby, M. and Stelovsky, J. (1990), ‘How do we read algorithms?: A case study’, Computer 23, 25–35.

Florea, R. and Stray, V. (2019), ‘The skills that employers look for in software testers’, Software Quality Journal 27(4), 1449–1479.

Fowles, D. C., Christie, M. J., Edelberg, R., GRINGS, W. W., Lykken, D. T. and Venables, P. H. (1981), ‘Publication recommendations for electrodermal measurements’, Psychophysiology 18(3), 232–239.

Fritz, T., Begel, A., C. Müller, S., Yigit-Elliott, S. and Züger, M. (2014), ‘Using psycho-physiological measures to assess task difficulty in software development’, Proceedings of the 36th International Conference on Software Engineering, ACM.

Graziotin, D., Wang, X. and Abrahamsson, P. (2013), Are happy developers more productive? the correlation of affective states of software developers and their self-assessed productivity.

Haapalainen Ferreira, E., Kim, S., Forlizzi, J. and Dey, A. (2010), Psycho-physiological measures for assessing cognitive load, pp. 301–310.

Hart, S. G. (1986), ‘Nasa task load index (tlx): Paper and pencil package’, NASA Ames Research Center, Moffett Field, CA United States.

Hart, S. G. and Staveland, L. E. (1988), Development of nasa-tlx (task load index): Results of empirical and theoretical research, in P. A. Hancock and N. Meshkati, eds, ‘Human Mental Workload’, Vol. 52 of Advances in Psychology, North-Holland, pp. 139 – 183.

Hess, E. H. and Pollt, J. M. (1960), ‘Pupil size as related to interest value of visual stimuli’, Science 132(3423), 349–350.

Khan, I., Brinkman, W.-P. and Hierons, R. (2011), ‘Do moods affect programmers’ debug performance?’, Cognition, Technology & Work 13, 245–258.

Khan, I., Brinkman, W.-P. and Hierons, R. (2013), ‘Towards estimating computer users’ mood from interaction behaviour with keyboard and mouse’, Frontiers of Computer Science.

Kim, K., Bang, S. and Kim, S. (2004), ‘Emotion recognition system using short-term monitoring of physiological signals’, Medical biological engineering computing 42, 419–27.

Kramer, A. F. (1991), ‘Physiological metrics of mental workload: A review of recent progress’, pp. 279–328.

Lee, J. and Tan, D. (2006), Using a low-cost electroencephalograph for task classification in hci research, pp. 81–90.

Micallef, M., Porter, C. and Borg, A. (2016), Do exploratory testers need formal training? an investigation using hci techniques, in ‘2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)’, IEEE, pp. 305–314.

Müller, S. C. (2015), Measuring software developers’ perceived difficulty with biometric sensors, in ‘2015 IEEE/ACM 37th IEEE International Conference on Software Engineering’, Vol. 2, pp. 887–890.

Müller, S. C. and Fritz, T. (2015), Stuck and frustrated or in flow and happy: Sensing developers’ emotions and progress, in ‘2015 IEEE/ACM 37th IEEE International Conference on Software Engineering’, Vol. 1, pp. 688–699.

Ostberg, J., Graziotin, D., Wagner, S. and Derntl, B. (2017), Towards the assessment of stress and emotional responses of a salutogenesis-enhanced software tool using psychophysiological measurements, in ‘2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SE-motion)’, pp. 22–25.

Parnin, C. (2011), Subvocalization - toward hearing the inner thoughts of developers, pp. 197 – 200.

Ramirez, R., Palencia-Lefler, M., Giraldo, S. and Vamvakousis, Z. (2015), ‘Musical neurofeedback for treating depression in elderly people’, Frontiers in Neuroscience 9.

Salimpoor, V. N., Benovoy, M., Longo, G., Cooperstock, J. R. and Zatorre, R. J. (2009), ‘The rewarding aspects of music listening are related to degree of emotional arousal’, PLOS ONE 4(10), 1–14.

Schmidt, S. and Walach, H. (2000), ‘Electrodermal activity (eda) - state-of-the-art measurement and techniques for parapsychological purposes’, Journal of Parapsychology 64, 139–163.

Shaw, T. (2004), The emotions of systems developers: an empirical study of affective events theory, pp. 124–126.

Siegmund, J., Peitek, N., Parnin, C., Apel, S., Hofmeister, J., Kästner, C., Begel, A., Bethmann, A. and Brechmann, A. (2017), Measuring neural efficiency of program comprehension, in ‘Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering’, ESEC/FSE 2017, Association for Computing Machinery, New York, NY, USA, p. 140–150.

Smith, S. W. (1997), The Scientist and Engineer’s Guide to Digital Signal Processing, California Technical Publishing, USA.

Vanitha, A. and Alagarsamy, K. (2019), ‘Software testing in cloud platform: A survey’.

Wróbel, M. (2013), Emotions in the software development process, pp. 518–523.