Using Personality Detection Tools for Software Engineering Research: How Far Can We Go?

FABIO CALEFATO and FILIPPO LANUBILE, University of Bari, Italy

Assessing the personality of software engineers may help match individual traits with the characteristics of development activities such as code review and testing, as well as support managers in team composition. However, self-assessment questionnaires are not a practical solution for collecting multiple observations on a large scale. Instead, automatic personality detection, while overcoming these limitations, is based on off-the-shelf solutions trained on non-technical corpora, which might not be readily applicable to technical domains like software engineering. In this paper, we first assess the performance of general-purpose personality detection tools when applied to a technical corpus of developers' emails retrieved from the public archives of the Apache Software Foundation. We observe a general low accuracy of predictions and an overall disagreement among the tools. Second, we replicate two previous research studies in software engineering by replacing the personality detection tool used to infer developers' personalities from pull-request discussions and emails. We observe that the original results are not confirmed, i.e., changing the tool used in the original study leads to diverging conclusions. Our results suggest a need for personality detection tools specially targeted for the software engineering domain.

CCS Concepts: • **Human-centered computing → Empirical studies in collaborative and social computing**.

Additional Key Words and Phrases: Computational personality detection, automatic personality recognition, Big Five, Five-Factor Model, replication, negative results, LIWC, IBM Personality Insights.

1 INTRODUCTION

Software engineers' personality has been a subject of interest for researchers since the 1970s when it was first hypothesized that personality traits might influence how developers interact [108, 122]. Cruz et al. [25] identified 90 studies published between 1970 and 2010, most of which published after 2002; Barroso et al. [10] found 21 studies, published between 2003 and 2016, studying the effect of personality on professional developers. The main reasons for this growing interest in personality-focused research lie in the many practical implications existing at both individual and team level. For example, previous studies on the personality of developers involved in agile development have revealed a positive association of conscientiousness (i.e., being organized, dependable) and openness to experience with their pair programming performance [101, 102]; in addition, evidence suggests that testers are significantly higher on conscientiousness than other software development practitioners [46] and that managers are more extroverted [111]. Regarding

Authors' address: Fabio Calefato, fabio.calefato@uniba.it; Filippo Lanubile, filippo.lanubile@uniba.it, University of Bari, Dipartimento di Informatica, via E. Orabona 4, Bari, Italy, 70125.

https://doi.org/10.1145/3491039

performance, teams whose members are more extroverted have been found to release higher-quality software products [2, 3]. Licorish et al. [54] have built a prototype to assist project managers in agile team formation by providing lightweight support for personality assessment. However, these studies barely scratched the surface of the potential scenarios of applying personality to software engineering, given its profound socio-technical nature [68].

Most of the previous research on personality has been conducted using self-assessment questionnaires. Albeit reliable, detecting the personality through psychometric questionnaires has some drawbacks, such as low return rates—especially in the software engineering domain [110]—and the limited number of occasions (typically one) to perform data collection [124]. The drawbacks of self-assessment questionnaires can be overcome by computational personality detection, which is the task of automatically inferring personality traits from conversation transcripts and written text [6, 29, 120].

With the proliferation of collaborative development environments such as GitHub, social media like Twitter, and communication platforms like Slack, developers' discussions have become easily accessible. Thanks to the availability of such a wealth of communication traces, software engineering researchers have begun developing solutions for automatically detecting developers' personalities from written text, thus making it possible to study on a large scale whether and how specific aspects might influence the outcome of development activities. For example, previous research has relied on automatic tools to investigate how personality varies with the level of developers' contribution and prominence in the Apache [97], Stack Overflow [11], and GitHub [95] ecosystems; other studies have focused on clustering similar personality profiles among the developers of the Eclipse [55, 86] and Apache [17] projects.

The research on computational personality detection has resulted in the release of several general-purpose prediction tools and models (e.g., [20, 60, 63, 64, 87]). However, while there has been prior work that assessed the psychometric validity of questionnaires showing good correlations among the instruments (e.g., [105]), when it comes to studies on automatic personality detection we found no previous assessment of general-purpose tool performance used as off-the-shelf components to analyze the communication traces left in a technical domain such as software engineering. Therefore, to fill this gap, in this paper we investigate the problem of using computational personality detection tools trained on non-technical domains for software engineering research. In particular, we first ask:

- **RQ1** – *How do off-the-shelf personality detection tools perform in the software engineering domain?*

The findings of this study show that when general-purpose personality detection tools are used off-the-shelf for software engineering research, their performance is far from acceptable as (i) neither they agree with self-reported ratings nor with each other; (ii) their prediction accuracy is lower compared to the results from prior work in non-technical domains.

The low agreement and accuracy rates of personality predictions suggest that the conclusions based on the application of these tools in the software engineering domain might be affected by the choice of one specific tool over the others. Therefore, to understand whether using different personality detection tools affects the validity of the results when employed in the same software engineering study, we ask:

- **RQ2** – *How does the choice of a personality detection tool affect the validity of previous results in software engineering research?*

We conduct the replications of two former studies and—after changing the tool used therein—find that the choice of a specific personality detection tool does affect the validity of the previously published results in software engineering research as we generally fail to reproduce them.

From a research perspective, our study makes a first attempt at benchmarking the performance of the tools available for personality detection from technical text. The study advances the state of the art on computational personality detection for software engineering research by suggesting that, to cope with a domain-specific lexicon, we need to develop software engineering-specific tools given that the existing solutions cannot be fine-tuned because the prediction models are generally not retrainable.

From a practical perspective, this study furthers our understanding of the current limitations when reusing computational personality detection tools as off-the-shelf components. We show that tools underperform in presence of technical text and warn that the choice of specific personality detection tools may lead to contradictory results. Another practical contribution is the replication package that we share with the research community, including both the scripts for automatically re-executing the entire experimental workflow and an anonymized gold standard consisting of an email corpus matched with developers' self-reported personality scores.

The remainder of this paper is organized as follows. In Sect. 2, we illustrate the study background. Sect. 3 presents the two-phase research framework followed to carry out our work. In Sect. 4, we study the performance of the selected personality prediction tools. In Sect. 5, we perform the replication of two previously published, software engineering studies. The results are discussed in Sect. 6. Finally, we present the related work in Sect. 7 and conclude in Sect. 8.

2 BACKGROUND

In Sect. 2.1, we first provide an overview of the fundamental concepts and theories related to personality. Then, in Sect. 2.2 we review some of the instruments used for personality measurement. Finally, in Sect. 2.3, we review prior work that focused on developers' personalities in the software engineering field.

2.1 Personality Theories

Personality is defined by psychologists as the set of all the behavioral, temperamental, emotional, and mental attributes that characterize a unique individual [100]. Personality has been conceptualized from a variety of theoretical perspectives and at various levels of abstractions. One level that has been often studied is *personality traits* [44]. Given the complexity of its nature, psychologists have developed several taxonomies of traits, that is, descriptive models of personality useful for organizing, distinguishing, and summarizing the major individual differences among the numerous existing in human beings.

Many theories of personality traits have been proposed since the 1930s. Such theories often disagreed regarding the number of traits and their nature [38]. However, after decades of research and the compelling amount of empirical evidence collected, since the 1970s a general consensus was achieved on the validity of a taxonomy of five orthogonal personality traits, called the *Big Five*. A brief description of the five factors is reported in Table 1. The name was first proposed by Goldberg [37] to emphasize that five dimensions are sufficient to capture the main dispositional characteristics and high-level differences of individuals. These five personality traits have been obtained through repeated studies by applying factor analyses to various lists of trait adjectives used in self-descriptions and self-rating questionnaires for personality assessment. These studies are based on the *lexical hypothesis* [5], a psycholinguistic conjecture according to which the most important individual characteristics and differences in personality have been encoded over time as words in the natural language, and the more important a characteristic or difference, the more likely it is for individuals to express it using associated words [44].

Several independent studies (see Digman [27] for a compendium) performed repeated factor analyses on questionnaires based on different personality taxonomies only to find consistent

Table 1. Overview of the Big Five traits and how their low vs. high levels characterize individuals. The traits are often referred to by the mnemonic OCEAN (adapted from Plotnik and Kouyoumdjian [90]).

Factor	**Description**	
	Low	**High**
Openness	Refers to the extent to which a person is open to experiencing a variety of activities, proactively seeking and appreciating unfamiliar experiences	
	Is conservative and close-minded	*Is open to novel experiences*
Conscientiousness	Refers to the tendency to plan in advance, act in an organized or thoughtful way as well as the degree of persistence and motivation in goal-directed behavior	
	Is impulsive and careless	*Is responsible and dependable*
Extroversion	Refers to the tendency to seek stimulation in the company of others, thus assessing a person's amount of interpersonal interaction and activity level	
	Is reserved and solitary	*Is outgoing and decisive*
Agreeableness	Refers to a person's tendency to be cooperative with others and compassionate about their thoughts, feelings, and actions	
	Is unfriendly and cold	*Is warm and good-natured*
Neuroticism	Refers to the extent to which one lacks emotional stability and is prone to psychological distress, anxiety, excessive cravings, or urges	
	Is stable and calm	*Is nervous and emotionally unstable*

evidence of the existence of a latent personality structure consisting of five main factors. In other words, the extracted models showed minor differences at the higher level—i.e., the traits, albeit labeled differently, could be easily mapped onto each other [38]. Hence, trait psychologists combined these findings on the general ubiquity of five factors across various instruments with the results from the studies on the lexical hypothesis to argue that any personality traits model had to encompass at some level the same Big Five dimensions [37].

Albeit the Big Five is considered a synonym with the *Five-Factor Model* (FFM) [23, 71], the two models are slightly different. Big Five is a term used to refer in general to personality frameworks consisting of five high-level dimensions and the Big Five model only describes the five traits at a broad level. Instead, the FFM is a personality framework that further refines the five high-level traits into multiple lower-level facets. For the sake of simplicity, from now on we will use Big Five and FFM interchangeably.

2.2 Instruments for Personality Detection

Psychometrics is the development of measurement instruments and the assessment of whether these instruments are reliable and valid forms of measurement [34].

Self-rating Questionnaires. Personality traits are usually assessed using self-assessment questionnaires, which present a variable number of items (up to hundreds) that describe common situations and behaviors. Subjects take these self-reporting tests by rating on a Likert scale the extent to which an item applies to them, where each item is either positively or negatively associated with a specific trait. Finally, a numeric score is computed for each trait by aggregating (e.g., summing) all the values assigned to its related answers.

There are several instruments for the self-assessment of personality, such as the *Myers-Briggs Type Indicator* (MBTI) [78] and the *Keirsey Temperament Sorter* (KTS) [49], based on Jung's theories. However, their psychometric validity has been questioned over the years [1, 12]

The most popular instruments for measuring the Big Five traits are the *NEO-PI* [23] and the *NEO-PI-R* [24]. McCrae [69, 70] has found NEO-PI-R to be reliable even after translating and administrating it across 36 countries, also showing that it is possible to use trait mean values to capture systematic differences. Another questionnaire intended to measure the five dimensions of personality is the *Big Five Inventory* (BFI) used by Schmitt et al. [105] in a large study on 56 nations. The results showed a robust five-factor structure across geographical and cultural regions as well as a high cross-instrument correlation with the NEO-PI-R scales.

Because all the instruments above are proprietary, psychologists have developed and validated the *International Personality Item Pool* (IPIP) and its follow-up IPIP-NEO (*International Personality Item Pool Representation of the NEO PI-R*), two alternative and open Big Five inventories freely available to researchers [39].

In summary, given the evidence of the validity gained by the Big Five inventories in general and the lack of psychometric reliability of the other instruments, in this work, we focus only on the FFM and the related instruments.

Computational Personality Detection from Text. Self-report inventories are the most popular psychometric instruments to assess personality because of their validity and ease of use. However, in addition to its semantic content, written text is also capable of conveying information about the writer, such as cues to individual personality. Psychologists have been able to identify correlations between specific linguistic markers and personality traits [88]. Computational personality detection [29], also referred to as automatic personality recognition [120] is the task of inferring people's personality from their digital footprints, such as social media content (e.g., videos, pictures, likes), and written text (e.g., conversation transcripts, blog posts, emails). A machine-learning algorithm uses features extracted from the analyzed content as cues to predict personality. The features and how they are associated with the traits vary depending on the type of corpora analyzed; for instance, audio recordings allow the use of acoustic cues, such as voice inflection, which are lacking in textual corpora where instead lexical features, such as the count of positive vs. negative words, part-of-speech tags, and *n*-grams, can be leveraged as personality markers. Further details on how computational personality detection works are provided later in Section 4.1, where the selected tools are introduced.

To date, there is a limited yet steadily growing amount of work on the automatic detection of personality [47]. In particular, thanks to the recent advances in machine learning, recently developed tools (e.g., [20, 60]) are leveraging deep-learning techniques for processing several cues extracted from large text corpora. Tools can be grouped into top-down and bottom-up solutions [21]. Top-down (or closed-vocabulary) solutions rely on external resources (e.g., psycholinguistic databases) and text is processed through such pre-determined dictionaries that specify meaningful word categories associated *a priori* with personality traits. Instead, bottom-up (or open-vocabulary) [107] do not specify in advance the relationship between features and allow linguistic cues (i.e., meaningful words and phrases) associated with personality traits to emerge from data.

In Sect. 4.1, we review in detail some of the tools available as off-the-shelf solutions to automatically recognize the Big Five personality traits from text.

2.2.1 Personality Datasets.

Because automatic personality recognition approaches are inherently data-driven, the availability of experimental datasets plays a crucial role. According to Novikov et al. [83], more than 40% of studies on personality involve the collection of new datasets that remain private. The remaining studies rely on a few shared and reusable datasets. Here we briefly review those based on textual documents.

The essay dataset [88] consists of nearly 2,500 essays (i.e., unedited pieces of text) written in a controlled setting by students who had also taken the BFI personality test. Originally used to validate the LIWC tool (see Sect. 4.1), it is one of the first corpora utilized in personality prediction research [6, 63] and is still very much in use [72, 103].

Given the growing amount of digital traces left by users in social networks, it is not surprising that most of the personality datasets collect documents from social media (e.g., [40, 61, 93]). The largest dataset used in personality prediction research is myPersonality [113], which contains data of Facebook users who filled in a personality questionnaire. The anonymized dataset has been freely shared with researchers for non-commercial academic purposes until it was retired in 2018. Another example of personality-annotated datasets of posts from social networks is the PAN-AP-2015 corpus [94], consisting of Twitter posts in English, Spanish, Italian, and Dutch from users who also took the BFI test.

To the best of our knowledge, there is no personality-annotated dataset of text documents (e.g., emails, issue reports, commit messages) collected from technical domains such as software engineering.

2.3 Personality Detection in Software Engineering

Early in the development of the software engineering field, it was recognized that, in addition to technological factors, researchers also had to consider the humans involved in the development process [108, 122]. Lenberg et al. [53] have proposed the term Behavioral Software Engineering to refer to the interdisciplinary study of cognitive, behavioral, and social aspects of software engineering as performed by individuals and groups.

In the following, we review the most relevant previous studies on personality in software engineering. We restrict our review to studies published since 2016—for earlier studies, refer to the SLRs reported in [10, 25]. Also, given the compelling amount of evidence on its validity, we focus our review only on studies that leveraged the Big Five model.

Kosti et al. [51] conducted a study using a clustering technique, which identified four archetypal personality profiles characterized by the levels of *extraversion* and *conscientiousness*. Mellblom et al. [74] analyzed the response from 47 participants in a survey aimed at revealing the relationship between specific personality traits and burnout in professional software developers. Through regression analysis, they uncovered a strong link between *neuroticism* and burnout. Mendes et al. [75] surveyed 63 Brazilian developers and found that the *agreeableness* trait is significantly associated with the variation in the decision-making style. Smith et al. [111] analyzed the characteristics of professional developers' personalities based on their roles. They found managers to be more *conscientious* and *extroverted*, and agile developers to be more *neurotic* and *extroverted*. Akarsu et al. [4] studied the personality traits by administering the BFI to 18 agile teams. They found that high levels of *agreeableness* and *conscientiousness* were common in most teams. Moreover, they observed a low level of *extraversion* in isolated teams that had fewer contacts with customers. In [121], Vishnubhotla et al. investigated the association between the Big Five traits and the factors related to team climate within eight agile teams. Through regression analysis, they found that *openness* has a statistically significant positive correlation with support for innovation; also *agreeableness* is positively correlated with the overall team climate. Finally, an interesting attempt was made by Yilmaz et al. [125] who developed a psychometric questionnaire, based on the BFI and specifically adapted it to the software engineering domain, to explore how practitioners' personality traits are associated with effective software teams. The results indicated that effective teams are characterized by low *neuroticism* and high levels of *agreeableness*, *extraversion*, and *conscientiousness*.

All the studies reviewed above rely on questionnaires for psychometric assessment. Yet, several studies also rely on the automatic recognition of developers' personality traits from text. In the Related Work (Sect. 7), we provide an in-depth review of such studies.

3 RESEARCH FRAMEWORK

In this work, we follow a two-phase research framework, as depicted in Fig. 1.

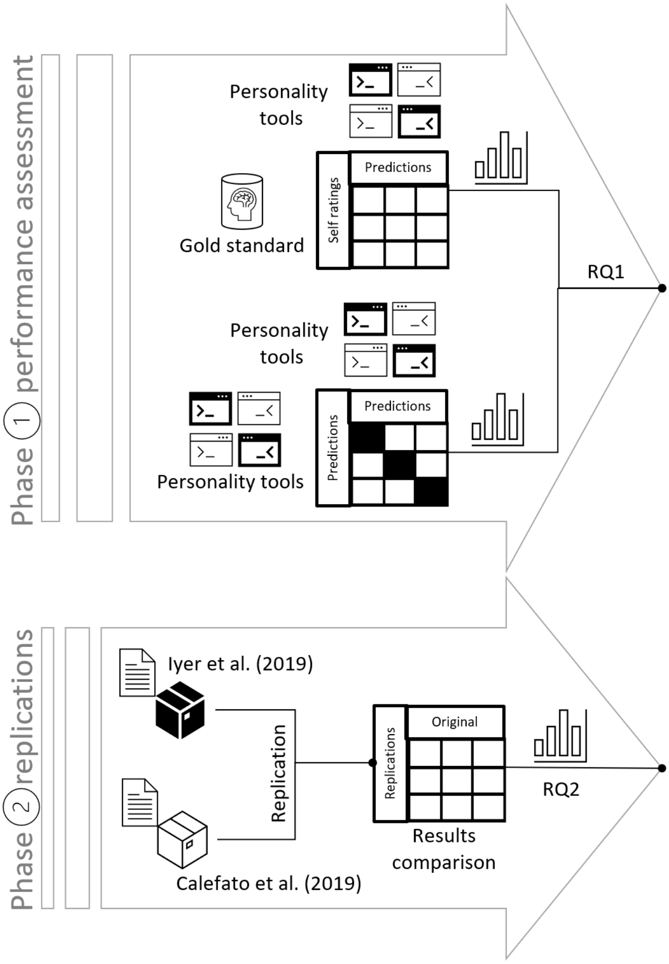


Fig. 1. Overview of the research framework.

In *Phase 1*, we assess the performance of personality detection tools for software engineering. Accordingly, to answer RQ1, we first select four tools built upon the Big Five model. Then, we set the ground truth by collecting the responses to a self-assessment personality questionnaire from 50 Apache Software Foundation (ASF) developers. Contextually, we build a dataset of over 1,500 emails written by the ASF developers. We apply the selected tools to the email dataset to extract the personality scores and compare their predictions against the self-ratings. Finally, we complement the previous analysis by assessing the agreement between the selected tools. As such, we make pairwise comparisons between the personality scores obtained by running the tools on the email corpus.

In *Phase 2*, we study whether the choice of one personality detection tool over the others affects the validity of results from software engineering studies. Accordingly, to answer RQ2, we select two recent, large-scale studies, respectively by Iyer et al. [42] and Calefato et al. [17] We choose to replicate these two studies because they both used the same tool (i.e., IBM Personality Insights) to extract developers' personality profiles from different datasets of technical content—respectively,

Table 2. The selected tools for computational personality detection on a continuous numerical scale. In column *Solution*, TD stands for Top-Down (closed vocabulary), BU for Bottom-Up (open vocabulary). The column *Scale points* indicates how the output is measured.

Tool	**License**	**Approach**	**Technique**	**Features**	**Dataset**	**Validation (ground truth)**	**Scale points**
LIWC	Commercial	TD	Word category frequencies	Closed vocabulary	2,479 essays	BFI	word-trait correlations*
IBM PI	Commercial	BU	Unspecified ML technique	Open vocabulary, word embedding	1,550-2,000 tweets	Unspecified Big 5 questionnaire	5
Personality Recognizer	Research prototype	TD	Regression models	LIWC, MRC	2,479 essays	LIWC dataset	7◇
TwitPersonality	Apache 2.0	BU	SVM	Word embedding	18,473 tweets, 9,913 FB posts	BFI, myPersonality dataset	5

* Transformed into trait scores using the formulae reported in Yarkoni [124] and scaled into the range 1 − 5.
◇ Rescaled into the range 1 − 5

pull-request discussions obtained from GitHub and emails retrieved from the ASF public archives. Finally, both studies provide a replication package, which we adapt to replicate them using another personality detection tool; the availability of complete replication packages allows us to minimize the risk of errors in executing the replications.

The two-phase framework is inspired by the work of Jongeling et al. [45], who conducted a study on the use of sentiment analysis tools for software engineering research and arranged their research questions sequentially so that, after observing disagreement among tool predictions, they could explore the effects on conclusion validity in prior work replications after switching tools.

We provide a complete replication package for the work presented here, which allows other researchers to re-execute all the steps in the research workflow. To reinforce the replicability, we provide a script that fully automates the whole experiment pipeline. Further details and instructions are available in Appendix A.

4 PHASE 1 – ASSESSMENT OF PREDICTION PERFORMANCE

This section is structured as follows. In Sect. 4.1, we describe the process followed to select the sample of tools for computational personality detection from text; we also provide details about their features, configuration, and output. In Sect. 4.2, we describe how we built a gold standard by administering a self-assessment personality questionnaire to software developers. In Sect. 4.3, we illustrate the process followed to build the dataset of emails written by the same subjects who answered the personality questionnaire. Finally, In Sections 4.4 and 4.5, respectively, we describe the evaluation metrics and report the results to answer RQ1.

4.1 Tools Selection

In this section, we review in detail the personality detection tools selected for the analysis of prediction accuracy and agreement.

To build a list of candidate tools, we started from those listed in recent studies [17, 73] containing reviews of solutions for automatic personality recognition from text, including both commercial tools and research prototypes. Then, from the resource identified, we sought more candidates using a snowball method. At the end of the search process, we identified 19 candidates.[1] From these, we filtered out those for which the tool has not been shared or made available as an off-the-shelf solution for testing the model on other datasets. Tools vary largely in terms of the type of prediction task. According to Schwartz et al. [107], prediction on a continuous numeric scale (i.e., traits are measured on a given numeric range) is a more appropriate task for studies on automatic personality

[1] The complete list of the tools identified along with the exclusion criteria is available at https://doi.org/10.6084/m9.figshare.15086391.

recognition. Therefore, we also filtered out those tools that intend personality trait prediction as a binary (e.g., yes/no) or multi-class (e.g., low/medium/high) task instead of measuring the outcome on continuous numeric scales. Eventually, we obtained four candidate tools.

To enrich the list of potential candidates, we complemented the previous search by also looking for tools in GitHub. We used four search strings obtained by combining "*personality*" with "*prediction*" (258 entries), "*detection*" (87), "*assessment*" (47), and "*recognition*" (38). Then, other than the filters applied earlier, we filtered out the repositories that (i) did not have an associated `README.md` file with installation and execution instructions and (ii) reported using personality models other than the FFM; (iii) were supplementing material to references research papers (to exclude student projects). We identified only one candidate repository, which however was already included in the previous list.

Next, we review the four selected tools. An overview is available in Table 2, where we also report the dataset, techniques, and features used to develop the prediction model, as well as the instrument adopted to establish the ground truth.

Finally, while we cannot claim completeness—the systematic review of this research field is outside the scope of this study—this section provides nonetheless a valuable, up-to-date overview of the state of the art in the field of computational personality detection on continuous numeric scales.

4.1.1 LIWC. The Linguistic Inquiry and Word Count (pronounced *luke*) [87] is a commercial, text-analysis program that counts words in psychologically meaningful, predetermined categories. It adopts a top-down approach, therefore analyzing the text looking for the occurrence of predetermined linguistic cues associated with personality traits. LIWC is arguably the most well-known resource in this category, often used as an external psycholinguistic database by other tools. Pennebaker and King [88] used LIWC to count the word categories of 2,479 essays (i.e., unedited pieces of text) written by volunteers who had also taken the BFI test as ground truth. In line with the lexical hypothesis, they found significant associations between the linguistic features of LIWC and the Big Five traits, thus providing evidence of existing connections between language use and personality [115].

Output interpretation. When analyzing a piece of text, LIWC returns word-category frequencies. We apply the formulae proposed by Yarkoni [124], one for each of the big five traits, which leverage the quantified connections between personality and word use, and transform such frequencies into numerical trait scores in the range $1 - 5$.

Tool setup. We used the standalone version of LIWC, which is also available from the web using the Receptivity[2] API. Because the tool now supports both the 2007 and 2015 versions of the vocabulary, we opted for the first one, because it was the version used in [124] to derive the formulae. Nonetheless, despite a few differences among the categories defined, in our internal tests, the scores generated with the two dictionaries have very strong Pearson correlations (between .92 and .97).

4.1.2 IBM Personality Insights. It is a commercial tool that uses an unspecified machine-learning model with a bottom-up, open-vocabulary approach. Earlier versions of the service (i.e., before December 2016) instead used the top-down, closed-vocabulary approach and relied on the LIWC dictionary. Models based on the open-vocabulary approach have been found to work well also in presence of small amounts of text such as tweets [7]. Also, as per IBM release note,[3] this version

[2] https://receptiviti.com

[3] https://cloud.ibm.com/docs/personality-insights/science.html#researchPrecise

of IBM PI reportedly outperformed the previous LIWC-based model. In November 2020, IBM announced[4] that the service had been deprecated and that it would be retired at the end of 2021.

Output interpretation. After analyzing a piece of text, IBM PI returns a JSON response. We parsed the JSON and retrieve five raw scores—one for each of the big five traits—defined in $[0, 1]$, which we rescaled in the range $1 - 5$.

Tool setup. Software Development Kits in multiple programming languages are available for the IBM PI service. In particular, we chose the Python API.

4.1.3 *Personality Recognizer.*

Top-down solutions make heavy use of external resources and test the correlations between those resources and personality traits. The seminal work for top-down solutions is Personality Recognizer,[5] a tool developed by Mairesse et al. [63] who conducted a series of experiments where multiple statistical models were benchmarked. With a supervised learning approach, they developed multiple prediction models using the same annotated dataset of essays employed for the development of LIWC. However, other than using LIWC features, they augmented the models with other dimensions from the Medical Research Council (MRC) psycholinguistic database [22].

Output interpretation. Personality Recognizer issues trait predictions on a continuous, seven-point scale. Therefore, to allow for comparison with other tools, we rescaled the output in the range $1 - 5$.

Tool setup. Personality Recognizer requires the MRC database and the LIWC 2001 dictionary. In our experiment, for consistency, we used the 2007 edition of LIWC. In addition, Personality Recognizer supports different models for computing scores; we opted for the default option, i.e., Support Vector Machine (SVM) with Linear kernel (SMOreg). Finally, the tool can be optimized for the analysis of spoken or written language. Given the nature of our dataset, we chose the second option.

4.1.4 *TwitPersonality.*

Another solution relying on the bottom-up approach is TwitPersonality.[6] To develop the tool, Carducci et al. [20] used a supervised learning approach. They first built a word-vector representation of Facebook posts (using the myPersonality dataset [113]) and then used it to train five SVM models, one for each trait. They also tested the models using a smaller corpus of tweets collected from 24 Twitter users, who also took the BFI test. Albeit with some tinkering, TwitPersonality is the only tool among those benchmarked in this study, whose model can be retrained on new data.

Output interpretation. TwitPersonality issues out-of-the-box trait predictions on a continuous, five-point scale. Therefore, no further transformations were necessary.

Tool setup. We used the default settings present in the source code. TwitPersonality can be used in two modes: *user-wise*, i.e., the written 'documents' of one author are aggregated and analyzed to extract the trait scores; *post-wise*, i.e., the trait scores are inferred from the documents individually, and then the average scores are computed. We opted for the former mode, for the sake of consistency with the other tools, albeit our internal tests showed that the differences between the two modes, when present, are negligible.

4.2 Gold Standard

In this section, we describe the process followed to build the gold standard and set the ground truth with a self-assessment questionnaire. We retrieved the publicly available mailing lists archives

[4] https://cloud.ibm.com/docs/personality-insights?topic=personality-insights-release-notes

[5] http://s3.amazonaws.com/mairesse/research/personality/recognizer.html

[6] https://github.com/D2KLab/twitpersonality

of the projects belonging to the ASF,[7] as of Jan. 2018. Our decision to investigate the ASF was motivated by the observation that all the projects within the ecosystem—albeit varying in size, scope, and technology stack—share the same code of conduct,[8] which enforces a shared set of guidelines that also regulate written interaction. Accordingly, we focused on analyzing the `dev` mailing lists because they are intended to host developers' discussions.

We retrieved the list of all email addresses present in the archives and randomly selected 1,000 among those who had contributed at least 10,000 words in their emails, to ensure they had contributed enough text for the analysis. We manually vetted the list to exclude the presence of emails automatically generated by bots such as the project's version control system or the mailing-list software. Finally, we sent invitations by email to the selected developers to take a personality test.

To collect the responses, we developed an electronic version of the 20-item Mini-IPIP [28] questionnaire—the shortest, valid personality instruments available—in an attempt to increase the notoriously low response rate of surveys in the software engineering domain [110]. The form collected the responses and, following the specifications provided in [28], transformed them into trait scores in the range $1 - 5$. In addition to the test questions, we inserted a couple of attention items and also measured the time taken to complete the tests. No monetary incentives were given to the test participants.

In the 2010s, personal data belonging to millions of Facebook users was collected without their consent by the consulting firm Cambridge Analytica and used during the 2016 US presidential campaigns for *psychological targeting*, i.e., the extraction of psychological profiles from social-media digital footprints to influence the attitudes, emotions, and behaviors of large groups of people. This scandal increased the public attention to the privacy risks of personal data misuse, creating still persistent social stigma attached to personality-related research and concerns deriving from participating in related studies [67]. Therefore, given the sensitive nature of the data collected in the study, both the invitation emails and the website contained a detailed description of the goal of the study and its academic-only interest. We ensured the developers taking the test that the results would only be presented in aggregate and that no resource would be shared, which could allow third parties to match the test results to their identities. For further details on anonymity and data protection measures adopted during data collection, please refer to the replication package in Appendix A.

We received 61 responses (6% response rate), of which 50 were deemed valid. Seven responses were excluded because the respondents failed the attention checks and four because of the short time spent in taking the test (less than two minutes over an average of nearly eight). The survey respondents belong to 34 different ASF projects,[9] including `lucene` (4 developers), `maven` (4), `couchdb` (3), `log4net` (3), `kafka` (2), `cassandra` (2), and `openmeetings` (2). From Fig. 2, we observe that the participants tend to be *open* (mean 4.33, SD .63), exhibit average levels of *conscientiousness* (3.70, .75) and *agreeableness* (3.73, .80), and are neither very *extroverted* (2.73, .84) or *neurotic* (2.77, .92).

4.3 Experimental Dataset

We matched the self-assessed personality profiles with a corpus of all the emails written by the developers who took the test. As a result, we were able to run the selected tools to infer the

[7] https://mail-archives.apache.org/mod_mbox

[8] www.apache.org/foundation/policies/conduct.html

[9] The complete list of projects and respondents is available at https://doi.org/10.6084/m9.figshare.15066564.

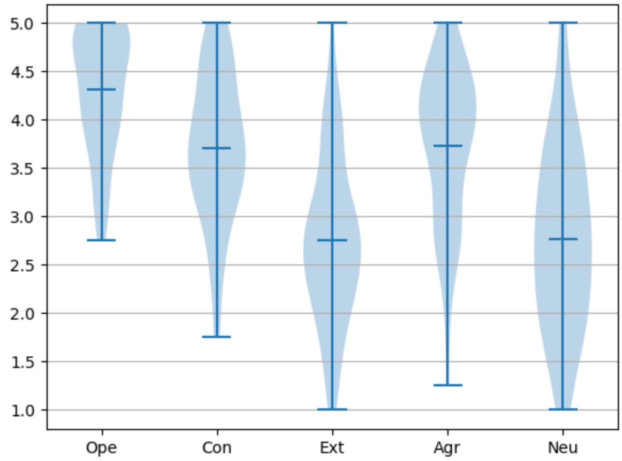


Fig. 2. Distribution of the self-ratings from the personality questionnaire.

personality scores from the email corpus and compare the predictions against the self-ratings from the questionnaire (gold standard).

To build the corpus, we aggregated all the email bodies sent by each subject and applied a series of filters to clean the data. In particular, we first used the `email-reply-parser`[10] library to ensure that only the text written by the email author was retrained while discarding the signature and any text coming from replying and forwarding. Then, we removed the lines of code using the R package `NLoN`, developed by Mäntylä et al. [65]. `NLoN` was trained and tested on various email corpora—including one from Apache developers—with good results. In addition, we used `polyglot`[11] to remove any non-English words. Finally, we lower-cased the text and removed all the stopwords using `NLTK`.[12]

Overall, the 50 subjects contributed 1,543 emails, with an average of 30.86 per developer (min 15, max 55, median 33, SD 8.45). After collating the email bodies and performing the data cleaning, we found that each developer has contributed on average 1,111.04 words (min 747, max 1764, median 1098, SD 180.14). We notice that these dimensions are in line with those of other datasets used in training and testing the selected tools. For example, IBM PI recommends providing at least between 600 and 1,200 words to enable the analysis,[13] albeit it is not disclosed what kind of preprocessing is applied. The dataset used by LIWC and Personality Recognizer contains 2,479 essays with an average length of 652 words. TwitPersonality employed a dataset of tweets, most of which are typically a hundred characters long.[14]

4.4 Evaluation Metrics

Research on the Big Five model has consistently considered personality detection as a set of five separate trait-prediction tasks. However, the formulation of the prediction tasks can differ

[10] https://pypi.org/project/email-reply-parser
[11] https://pypi.org/project/polyglot
[12] www.nltk.org
[13] https://cloud.ibm.com/docs/personality-insights?topic=personality-insights-input
[14] https://blog.twitter.com/official/en_us/topics/product/2017/tweetingmadeeasier.html

Table 3. Reference values in terms of Pearson r and Spearman ρ correlations from prior work that employed text-based datasets (larger is better).

Study (best results)	Approach	Technique	Subjects	Dataset	Validation (ground truth)	Pearson r				
						O	C	E	A	N
Lynn et al. [62]	BU	Message-level attention model	68,687	myPersonality	myPersonality (IPIP)	0.66	0.54	0.58	0.56	0.56
Laleh and Shahram [52]	TD	Regression analysis	92,225	myPersonality	myPersonality (IPIP)	0.38	0.29	0.34	0.33	0.27
Arnoux et al. [7]	TD	Regression analysis	1,323	Private	IPIP	0.29	0.33	0.27	0.42	0.37
Kosinski et al. [50]	TD	Regression analysis	54,373	myPersonality	myPersonality (IPIP)	0.43	0.29	0.40	0.30	0.30
Tomlinson et al. [118]	TD	Regression analysis	250	myPersonality	myPersonality (IPIP)	-	0.27	-	-	-
Study (best results)	**Approach**	**Technique**	**Subjects**	**Dataset**	**Validation (ground truth)**	**Spearman ρ**				
						O	**C**	**E**	**A**	**N**
Hall and Caton [40]	TD	Regression analysis	509	Private	Unspecified Big 5 questionnaire	0.62	0.68	0.70	0.68	0.61

TD stands for Top-Down (closed vocabulary), BU for Bottom-Up (open vocabulary).

drastically [84]: they can be approached as binary classification tasks, using the mean or median as thresholds to discretize the numerical scores; alternatively, after removing the observations in the middle, they approach the binary classification as limited to the upper and lower groups, albeit this is less than ideal with bell-shaped distributions.

The choice of the metrics for evaluating performance also varies. In the case of personality predictions on a continuous scale, following [107], in our analysis, we included the following performance metrics: *Pearson* product-moment correlation (r) and *Spearman* rank correlation (ρ), measured between the predicted trait scores (by each tool) and the actual scores (from the gold standard, self-assessment questionnaire); *Mean Absolute Error* (MAE),[15] the average of the absolute value of the difference between the actual and predicted scores; *Root Mean Squared Error* (RMSE),[16] the standard deviation of the residuals, i.e., the prediction errors.

Pearson correlation evaluates the linear relationship between two continuous variables. Coefficients can range from -1 (perfect negative) to +1 (perfect positive), with values close to 0 indicating the lack of correlation. Assessing the predictive performance in automatic personality detection means estimating the convergent validity, i.e., the degree to which the measures of the same construct correlate with each other. One problem with using correlations is how to interpret the results. A common interpretation of the observed correlation magnitude is: $.00 - .10$ negligible, $.10 - .39$ weak, $.39 - .69$ moderate, $.69 - .89$ strong, $.89 - 1.0$ very strong [106]. However, as observed by the authors, these cutoff points are arbitrary and should be used judiciously. In particular, values in the middle are disputable and their interpretation as weak, moderate, or strong varies with the applied rule of thumb. Achieving correlations of $r > .30$ in psychology studies is challenging—even the simple axiom according to which people's past behavior is predictive of future actions has been found to produce a correlation coefficient of $r \approx .39$ [76]. Rather than relying on the conventional cut-off points used for interpreting correlation coefficients in other fields, Meyer et al. [76] and Roberts et al. [98] have argued that research investigating psychological constructs should use baselines in the order of magnitude of correlations independently measured in related work. In other words, they have called for adjusting the norms that researchers hold for what the strength of relationships is in psychology and related fields. For example, IBM PI reportedly achieved for English an average Pearson correlation coefficient $r \approx .33$ in an internal assessment study. Some studies on psychological and behavioral constructs (e.g., [98]) have reported Pearson correlations with a small to medium magnitude in the range $.10 - .40$. In a survey of over 200 papers on personality published since 2017, Novikov et al. [83] found that the reported Pearson correlation coefficients

[15] $MAE_t = \frac{\sum_{i=i}^{n} |y_i - \widehat{y_i}|}{n}$ where t is the trait, and y_i and $\widehat{y_i}$ are the ground truth and predicted scores for subject $i = 1..n$.

[16] $RMSE_t = \sqrt{\frac{\sum_{i=i}^{n} (y_i - \widehat{y_i})^2}{n}}$ where t is the trait, and y_i and $\widehat{y_i}$ are the ground truth and predicted scores for subject $i = 1..n$.

Table 4. Reference values in terms of Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) from prior work that employed text-based datasets (smaller is better).

Study (best results)	**Approach**	**Technique**	**Subjects**	**Dataset**	**Validation (ground truth)**	**MAE**				
						O	**C**	**E**	**A**	**N**
IBM PI	BU	Unspecified	-	Private	Unspecified Big 5 questionnaire	Avg. ≈ 0.120				
Goldbeck et al. [36]	TD	Regression analysis	167	Private	BFI	0.099	0.104	0.124	0.109	0.117
Golbeck et al. [35]	TD	Regression analysis	50	Private	BFI	0.130	0.146	0.160	0.182	0.119
Arnoux et al. [7]	BU	Regression analysis	1,323	Private	IPIP	Avg. ≈ 0.120				
Study (best results)	**Approach**	**Technique**	**Subjects**	**Dataset**	**Validation (ground truth)**	**RMSE**				
						O	**C**	**E**	**A**	**N**
Carducci et al. [20] (TwitPersonality)	BU	SVM	24/250	Private/ myPersonality	myPersonality (IPIP)	0.332	0.530	0.708	0.448	0.557
Quercia et al. [91]	TD	Regression analysis	335	Private	myPersonality (IPIP)	0.690	0.760	0.880	0.790	0.850
Laleh and Shahram [52]	TD	Regression analysis	92,225	myPersonality	myPersonality (IPIP)	0.155	0.173	0.195	0.173	0.195

TD stands for Top-Down (closed vocabulary), BU for Bottom-Up (open vocabulary).

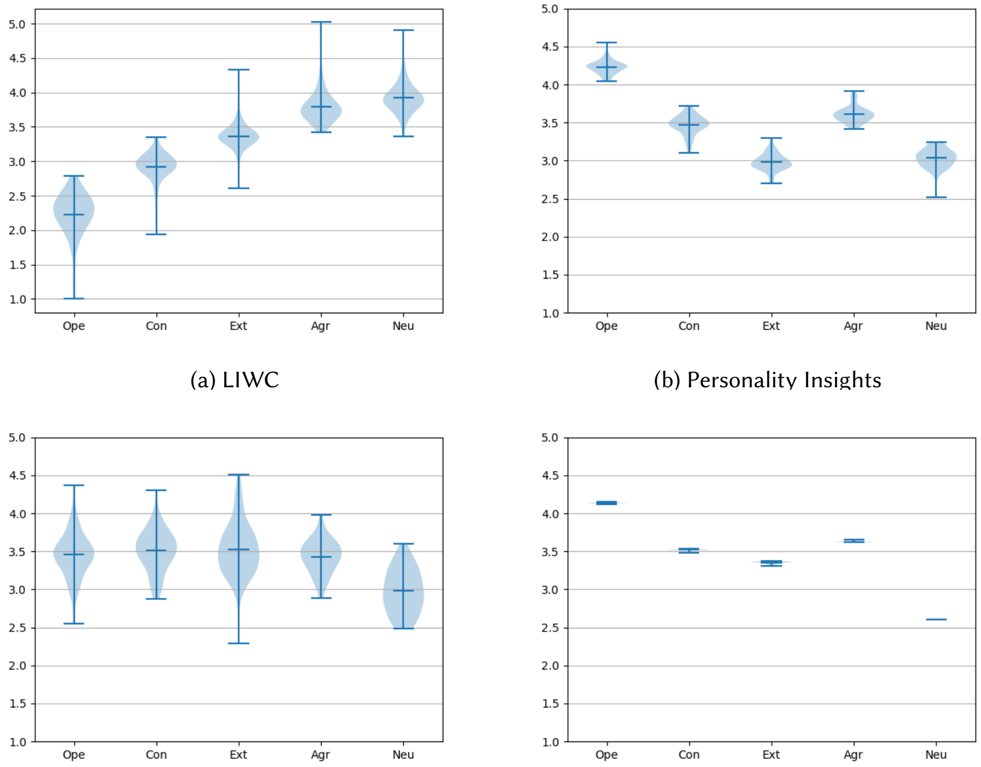
between predicted and self-reported personality traits are upper limited by values near .50. One exception is represented by the work of Lynn et al. [62], which reports a score exceeding 0.60 in the case of *openness*. Table 3 lists some of the Pearson correlation coefficients reported in recent prior work. We will use these values as baselines to assess the performance of the personality prediction tools involved in our study. We notice that most of the studies that reported Person correlation metrics adopted a top-down approach, using regression analyses for analyzing data from the myPersonality dataset.

Spearman correlation coefficient also varies between -1 and +1. Unlike Pearson r, however, Spearman ρ is a non-parametric measure that does not make any assumption regarding the normality, linearity, and homoscedasticity of distributions. The Spearman coefficient is based on the ranked values for each variable rather than the raw data. Fowler [32] found Spearman rank correlations to be more robust and outperforming Pearson r in cases of non-normal distributions. Only a few studies (e.g., [40]) have assessed prediction performance using Spearman ρ, which appears to be used more frequently with multimedia datasets. A complete analysis of correlation coefficients reported in prior work is out of the scope of this work. For more, please refer to the meta-analyses performed by Azucar et al. [8] and Marengo and Montag [66].

The MAE and RMSE measures are also very popular. Novikov et al. [83] found that 65 out of the 218 studies analyzed report personality prediction performance using either measure. Nonetheless, these metrics are not without problems. Their interpretation is largely dependent on the scale of the data and their estimates tend to be over-optimistic since self-ratings in gold standards tend to be normally distributed, with most observations close to the mean [114]. Given their popularity, we include these metrics for the sake of comparison with prior work. Table 4 provides an overview of the best MAE and RMSE results reported in recent and relevant studies, thus providing us with a performance baseline. We notice that the top-down and bottom-up approaches are almost equally distributed and that these studies mostly relied on regression analyses. However, while the studies reporting RMSE typically used myPersonality as the data source and ground truth, the others that relied on MAE built private datasets, using the IPIP or BFI questionnaires for validation.

4.5 Results

In Figures 3a-d we report the violin plots of the trait score distributions inferred by the tools. As compared to the distributions of the gold standard scores reported earlier in Fig. 2, we notice that the tool predictions are far less spread out. This can be also observed from the small standard



(a) LIWC

(b) Personality Insights

(c) Personality Recognizer

(d) TwitPersonality

Fig. 3. Violin plots of the trait predictions for each of the selected tools.

deviations reported along with other descriptive statistics in Appendix B (see Table 23). In particular, we observe that TwitPersonality issues predictions for each score that are clustered around the mean, and even constant in the case of *neuroticism*. The Q-Q plots in Figures 5-9 (see Appendix B) show that while linear relationships exist in the distributions of the self-ratings as well as the tool predictions, they are not normal, thus violating one of the assumptions to apply Pearson r correlation.

Agreement Analysis. We assessed the level of agreement by comparing the personality scores predicted by the tools, respectively, against the self-reported ratings and between each other. The Pearson and Spearman correlation coefficients are reported in Table 5. We point out that the correlation coefficients could not be calculated for *neuroticism* in the case of TwitPersonality because the tool issues a constant value for all the subjects.

Overall, we notice that no pair does consistently better than any other in terms of either correlation coefficient. Regarding Pearson correlation, the highest r coefficient is computed between LIWC and TwitPersonality for the *extraversion* trait ($r = .34$). However, most coefficients are approximately equal to .10 or smaller, and several are negative. Consistently, we observe that the largest r coefficients between all pairs and across all traits in Table 5 ($r \approx .05 - .34$) are smaller

Table 5. Pearson r and Spearman ρ pairwise correlations for each of the five traits between the gold standard (GS) and each tool, and between tools. The best result (highest positive correlation) for each trait is reported in **bold**.

Pair	Pearson r					Spearman ρ				
	O	C	E	A	N	O	C	E	A	N
GS – LIWC	-0.063	-0.057	-0.325	-0.151	-0.090	0.030	-0.056	-0.151	**0.134**	-0.048
GS – IBM PI	-0.050	**0.016**	-0.156	-0.001	-0.073	-0.048	**0.140**	-0.200	0.009	-0.107
GS – Pers.Rec.	-0.036	-0.028	**0.063**	-0.120	**0.142**	**0.033**	-0.125	**0.034**	0.041	**0.083**
GS – TwitPers.	**0.016**	-0.148	-0.114	**0.150**	-	-0.011	-0.148	-0.137	0.131	-
LIWC – IBM PI	-0.169	**0.046**	0.112	-0.144	**0.173**	-0.134	**0.003**	0.213	-0.148	**0.132**
LIWC – Pers.Rec.	**0.212**	-0.185	-0.025	**0.310**	-0.175	**0.102**	-0.073	0.035	**0.100**	-0.035
LIWC – TwitPers.	0.055	-0.006	**0.343**	-0.023	-	0.052	-0.037	**0.217**	-0.100	-
IBM PI – Pers.Rec.	-0.037	-0.072	0.191	-0.064	-0.003	-0.037	-0.099	0.159	-0.092	-0.015
IBM PI – TwitPers.	-0.195	-0.224	-0.050	-0.104	-	-0.183	-0.232	0.000	-0.198	-
Pers.Rec. – TwitPers.	-0.073	-0.043	-0.175	0.090	-	-0.077	-0.027	-0.082	0.099	-

Table 6. Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The best (smallest) result for each trait is reported in **bold**.

Tools	MAE					RMSE				
	O	C	E	A	N	O	C	E	A	N
LIWC	2.643	1.268	**0.709**	**0.668**	1.200	7.441	2.146	**0.759**	**0.653**	2.059
IBM PI	**0.554**	0.632	0.720	0.678	0.798	**0.424**	0.632	0.800	0.674	0.952
Pers. Rec.	0.955	0.628	1.011	0.770	**0.767**	1.233	0.650	1.376	0.795	0.925
TwitPers.	0.566	**0.611**	0.882	0.676	0.769	0.431	**0.606**	1.065	0.663	**0.889**

than those found in prior work and reported earlier in Table 3 ($r \approx .40 - .60$). In terms of Spearman correlation, the same observations still hold, with ρ coefficients in the range $.10 - .34$, therefore smaller than those reported in prior work ($\rho \approx .60 - .70$).

Concerning the correlations between the gold standard and the tools, we cannot identify any pattern. Instead, regarding the correlations between tools, we notice that most of the largest r and ρ coefficients are obtained for pairs including LIWC. In the case of LIWC and Personality Recognizer, this result is not surprising since the latter uses the LIWC dictionary to build the prediction model and, therefore, the two tools share some linguistic features.

Accuracy analysis. To complete the performance assessment, we complemented the agreement analysis by making sense of the prediction accuracy. Table 6 reports the MAE and RMSE metrics for each tool. First, we observe that there is no best tool in absolute, which outperforms all the others. LIWC does better for the *extraversion* and *agreeableness* traits; TwitPersonality performs better at estimating *conscientiousness* and *neuroticism*. However, a known limitation of MAE and RMSE is failing to capture the performance accurately when distributions containing observations that are mostly clustered around the mean, as in the case of TwitPersonality. Nevertheless, the best results in this study (MAE $\approx .55 - .77$, RMSE $\approx .42 - .90$) are considerably larger (worse) than the values found in prior work and reported earlier in Table 4 (MAE $\approx .10 - .18$, RMSE $\approx .16 - .88$).

4.6 Threats to Validity

The results of the analyses should be interpreted in light of the following limitations.

First, while using individual self-ratings as gold standards to set ground truth is the norm, psychology research considers the definition of a ‘true’ personality profile out of reach [123]. Indeed, personality is an elusive concept whose assessment makes it a complex activity for any rater, whether self, external observer, or computer. Any form of personality rating is a proxy measure and, as such, it comes with its limitations. Despite their validity, self-assessment questionnaires are subjective and biased towards social desirability [13]; external judgments are limited by raters' idiosyncrasies and machine learning algorithms, while free of human prejudice, reflect biases present in the data [116]. Also, albeit highly correlated, there are differences between personality constructs based on self-ratings and external observers' judgments [77].

In line with recent meta-reviews of personality prediction model performance [8, 66], we used correlation coefficients (both Pearson and Spearman) as reference metrics. We compared studies that were trained and tested on different datasets (e.g., essays, social media posts). The heterogeneity of the data sources might have influenced the comparability of the results. However, this was intentional to assess how personality detection tools can be used across domains as off-the-shelf solutions, and specifically for software engineering research.

Another potential issue is related to the use of English as lingua franca in emails, i.e., some developers did not communicate using their native language. A limited vocabulary may have arguably prevented some lexical cues from emerging from the text, as argued in the lexical hypothesis.

Finally, we acknowledge the relatively low number of subjects (50 developers) and documents (631 emails) in our experimental sample. However, it is not uncommon to find previous studies on personality detection using samples of a similar size. For instance, Carducci et al. [20] tested their personality detection tool using a corpus of Twitter posts from 24 volunteers. Similarly, Arnoux et al. [7] report the Big Five traits extracted from the social media posts of 55 volunteers.

Phase 1 – Summary

When off-the-shelf personality detection tools are applied out of domain, their performance is far from acceptable. Indeed, the tools neither agree with self-reported ratings nor with each other when used for software engineering research. In addition, their prediction accuracy is considerably worse as compared to the results from prior work on automatic personality detection from text.

5 PHASE 2 – IMPLICATIONS ON EARLIER STUDIES

The results of Phase 1 suggest that, due to the limited level of agreement and accuracy of tool predictions, the choice of a personality detection tool *might* affect the validity of previous results. Indeed, these disagreements do not necessarily imply that conclusions based on the application of these tools in the software engineering domain are affected by the choice of one specific tool over the others.

Accordingly, to answer RQ2, in this section we investigate whether the choice of a specific personality tool introduces threats to conclusion validity by replicating two previous studies in software engineering, which relied on computational personality detection. Because our goal is to assess whether the effects reported in previous studies still hold when a different personality detection tool is used, here we perform two *exact dependent replications* [109] in which we keep the same experimental setup of the original studies while changing only the tool.

5.1 Replicated Studies

We choose to replicate two previous studies, respectively by Iyer et al. [42] and Calefato et al. [17], which have analyzed the personalities of open-source software developers. Both studies provide a replication package. Therefore, we can apply the chosen methodology and perform an exact replication of the studies with the only deviation from the original work being the different personality detection tool used. Since both original studies used IBM PI here we choose to replace it with LIWC because it is the most used psychometric tool, adopted also in previous studies on personality in software engineering (e.g., [11, 95, 97]).

5.1.1 Iyer et al. (2019). The first study that we replicate is by Iyer et al. [42]. The authors applied IBM PI to examine the influence of personality traits of developers on the pull request evaluation process in GitHub. They first extracted the Big Five personality traits of 16,935 developers from trace data on GitHub, such as commit messages, issue and pull request comments. Then, they assessed their relative importance in the pull request evaluation process as compared to other non-personality factors from past research. Overall, they evaluated 501,327 pull requests from 1,860 projects and found that the effect of personality traits is significant and comparable to technical factors (e.g., number of files changed, presence of tests), albeit social factors (e.g., prior interaction, following each other) are more influential on the likelihood of pull request acceptance. In particular, they found that pull requests authored by developers (requesters) who are more *open* and *conscientious*, but less *extroverted*, have a higher chance of acceptance. Furthermore, pull requests that are closed by developers (closers) who are more *conscientious*, *extroverted*, and *neurotic*, have a higher likelihood of acceptance. Additionally, the larger the difference in personality traits between the requester and the closer, the more positive effect it has on pull request approval.

For the re-execution, we adapted the original scripts provided in the replication package. However, since the package did not include the collection of traced data used for the analysis, we followed the description of the data collection process in the paper to recreate the dataset ourselves. This lead to minor differences in the number of projects found (1,853) as compared to those reported in the original study (1,860). Finally, we applied LIWC to the reconstructed dataset to infer the developers' personality scores.

5.1.2 Calefato et al. (2019a). The second study that we replicate is by Calefato et al. [17]. The authors applied IBM PI to perform an analysis at the ecosystem-level of code commits and email messages contributed by 211 developers working on ASF projects. They found that there are three common types of personality profiles among Apache developers, characterized in particular by their level of *agreeableness* and *neuroticism.* They also found that developers with higher levels of *openness* are more likely to become contributors to ASF projects. In addition, they confirmed that developers' personality is stable over time and that the five traits do not vary significantly with their role, membership, and extent of contribution to the projects.

For the replication, we adapt the same scripts and use the same experimental dataset from the original study on which we apply LIWC.

5.2 Replication Results

To distinguish from those of this work, the research questions of the original studies are formatted in italic and lower-cased (e.g., *rq1*, *rq2*, ...).

5.2.1 Replication of Iyer et al. (2019). In this section, we answer the four research questions *rq0-3* of the original study by Iyer et al. [42] and recreate the same tables for comparison.

rq0—Replication of base model with the reconstructed dataset.

Table 7. Odds ratio of the basic models without personality traits from the original study by Iyer et al. [42] and the replication. Significant values are shown in **bold** (sig: *$p < .05$, **$p < .01$, ***$p < .001$).

Variables	Original study	Replication
(Intercept)	2.81	2.47 ***
test_file	**1.08** ***	**1.07** ***
total_churn	**0.90** ***	**0.90** ***
social_distance	**2.35** ***	**2.37** ***
num_comments	**0.68** ***	**0.68** ***
prior_interaction	**1.53** ***	**1.45** ***
followers_current	1.00	0.98
main_team_member	**1.16** ***	**1.19** ***
age_current	0.91	**1.01** ***
team_size	0.99	0.93
stars_current	**0.53** ***	**0.53** ***
test_file x num_comments	**1.12** ***	**1.13** ***
total_churn x num_comments	**1.06** ***	**1.07** ***
social_distance x num_comments	**0.92** ***	**0.93** ***
num_comments x prior_inter	**1.05** ***	**1.05** ***
R^2m	0.11	0.11
R^2c	0.58	0.59
AIC	394,718	340,207

Given the slight differences between the original and reconstructed datasets, we replicate Iyer et al.'s findings of the baseline model—including only technical and social factors—on a more recently mined dataset to determine if the results still hold. Creating a baseline model is useful in comparing other personality models with social, technical, and personality factors. In addition, GitHub has tremendous yearly growth rates, and over 60M new repositories have been created since the original study was carried out.[17] As such, the replication of the baseline model provides insights on the generalizability of Iyer et al.'s results to a dataset extracted at a different point in time.

We use the same modeling technique as in the original study—a mixed-effects logistic regression—on the reconstructed dataset to assess the effects of social and technical factors on pull request acceptance. Table 7 provides a comparison between our results and the original ones by Iyer et al. in terms of odds ratios. All factors have similar overall influences and even the model fit, both marginal (R^2m) and conditional (R^2c), is similar. The only exception is the project age factor, for which we found a significant effect. We speculate that this is due to the extraction of projects that have been active for a longer time in the reconstructed dataset.

Overall, although there are small fluctuations in the odds ratio of all the features, the original results still hold. This increases the confidence in the goodness of the reconstructed dataset and that any statistically significant, different results in the replication of the other research questions are due to the different personality detection tool used rather than differences in the data.

rq1—Does the personality of a requester affect the likelihood of the pull request being accepted?

The author of a pull request (a *requester* hereinafter) can be either a member of a projects' core development team or an outside contributor. Iyer et al. examined the personality of requesters to understand whether specific traits lead to a higher likelihood of pull request acceptance. As in the

[17] https://octoverse.github.com, accessed in March 2021.

Table 8. Odds ratio of the mixed-effect model with requester's personalities from the original study by Iyer et al. [42] and the replication. Significant results are shown in **bold** (sig: $^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$).

Variables	Original study Single run Odds Ratio	Original study Bootstrap. 95% CI	Replication Single run Odds Ratio	Replication Bootstrap. 95% CI
(Intercept)	2.87 ***	-	2.72 ***	-
test_file	**1.08** ***	[1.05, 1.11]	**1.06** ***	[1.02, 1.10]
total_churn	**0.90** ***	[0.89, 0.90]	**0.90** ***	[0.88, 0.91]
social_distance	**2.35** ***	[2.59, 2.80]	**2.38** ***	[2.63, 2.99]
num_comments	**0.68** ***	[0.65, 0.69]	**0.69** ***	[0.67, 0.69]
prior_interaction	**1.52** ***	[1.51, 1.57]	**1.45** ***	[1.43, 1.50]
followers_current	0.99	[0.95, 0.98]	0.98	[0.94, 0.99]
main_team_member	**1.15** ***	[1.12, 1.20]	**1.19** ***	[1.15, 1.22]
age_current	0.91	[0.83, 0.93]	1.01	[0.98, 1.05]
team_size	0.99	[0.85, 0.98]	0.92	[0.84, 0.97]
stars_current	**0.54** ***	[0.45, 0.49]	**0.54** ***	[0.46, 0.51]
openness	**1.07** ***	[1.05, 1.08]	**1.34** ***	[1.31, 1.57]
conscientiousness	**1.05** ***	[1.03, 1.07]	**0.77** ***	[0.68, 0.79]
extraversion	**0.94** ***	[0.93, 0.95]	**1.31** ***	[1.29, 1.46]
agreeableness	1.01	[1.00, 1.02]	**1.39** ***	[1.45, 1.60]
neuroticism	0.97	[0.94, 0.98]	**0.81** ***	[0.71, 0.84]
test_file x num_comments	**1.13** ***	[1.11, 1.15]	**1.12** ***	[1.10, 1.17]
total_churn x num_comments	**1.06** ***	[1.06, 1.08]	**1.07** ***	[1.06, 1.09]
social_connection x num_comments	**0.92** ***	[0.90, 0.96]	**0.93** ***	[0.90, 0.97]
num_comments x prior_interaction	**1.05** ***	[1.06, 1.08]	**1.05** ***	[1.05, 1.08]
R^2m	0.11		0.13	
R^2c	0.55		0.59	
AIC	394,635		333,552	

original study, we replicate *rq1* using a mixed-effects logistic regression to model the personality traits of the requester from the new dataset, along with the features already used in *rq0*.

Table 8 shows the comparison of the odd ratios from our replication against those from the original study. Unlike Iyer et al.'s results, in our replication *agreeableness* and *neuroticism* have a significant effect, respectively positive (1.39) and negative (0.81). Regarding *openness*, we find that the trait has a positive and significant influence in both studies, albeit the effect is smaller in Iyer et al. (1.34 vs. 1.07). Instead, we find a significant yet opposite effect for both *conscientiousness* (0.77 vs. 1.05) and *extraversion* (1.31 vs. 0.94).

rq2—Does the personality of a closer affect the likelihood of the pull request being accepted?

Pull requests that are closed by developers (*closers*) who are always part of the core team. By analyzing the closers' personalities, Iyer et al. aimed to understand whether specific traits affect the likelihood of the pull request getting accepted. As in the original study, we replicate *rq2* by using the requesters' personality traits instead of closers' and modeled them along with the factors used in *rq0*.

The results are reported in Table 9. *Openness* has a significant and positive effect on the pull request acceptance in both the replication (1.18) and the original study (1.05). We also observe a significant result for *conscientiousness* in both studies, albeit with an opposite direction (0.92 vs.

Table 9. Odds ratio of the mixed-effect models with closer's personalities from the original study by Iyer et al. [42] and the replication. Significant results are shown in **bold** (sig: $^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$).

	Original study		Replication	
Variables	**Single run**	**Bootstrap.**	**Single run**	**Bootstrap.**
	Odds ratio	**95% CI**	**Odds ratio**	**95% CI**
(Intercept)	2.87 ***	-	2.45 ***	-
test_file	**1.08** ***	[1.05, 1.11]	**1.06** ***	[1.03, 1.10]
total_churn	**0.90** ***	[0.89, 0.91]	**0.90** ***	[0.88, 0.90]
social_distance	**2.35** ***	[2.35, 2.83]	**2.41** ***	[2.66, 3.03]
num_comments	**0.68** ***	[0.65, 0.68]	**0.69** ***	[0.67, 0.69]
prior_interaction	**1.49** ***	[1.52, 1.56]	**1.45** ***	[1.43, 1.50]
followers_current	0.98	[0.94, 0.99]	0.98	[0.94, 0.99]
main_team_member	**1.16** ***	[1.12, 1.21]	**1.19** ***	[1.14, 1.22]
age_current	**0.92** ***	[0.86, 0.95]	1.01	[0.98, 1.05]
team_size	0.97	[0.88, 1.00]	0.97	[0.88, 1.00]
stars_current	**0.54** ***	[0.44, 0.51]	**0.54** ***	[0.46, 0.50]
openness	**1.05** *	[1.02, 1.10]	**1.18** ***	[1.15, 1.27]
conscientiousness	**1.12** ***	[1.11, 1.18]	**0.92** ***	[0.86, 0.95]
extraversion	**1.06** ***	[1.06, 1.13]	1.03	[0.99, 1.05]
agreeableness	1.01	[0.99, 1.04]	**1.13** ***	[1.13, 1.21]
neuroticism	**1.08** ***	[1.06, 1.14]	1.00	[0.94, 1.06]
test_file x num_comments	**1.12** ***	[1.08, 1.16]	**1.13** ***	[1.10, 1.17]
total_churn x num_comments	**1.06** ***	[1.05, 1.08]	**1.07** ***	[1.06, 1.09]
social_connection x num_comments	**0.92** ***	[0.88, 0.95]	**0.93** ***	[0.90, 0.97]
num_comments x prior_interaction	**1.05** ***	[1.05, 1.08]	**1.05** ***	[1.05, 1.07]
R^2m	0.12		0.11	
R^2c	0.57		0.59	
AIC	394,548		333,965	

1.12). Instead, we find completely contrasting findings regarding *extraversion*, *agreeableness*, and *neuroticism*.

rq3—Does the difference in personality between the requester and the closer affect the likelihood of the pull request being accepted?

Finally, Iyer et al. analyzed the differences in the personality traits between the requester and the closer to understand whether they hinder or facilitate pull request acceptance. As in the original study, we replicate *rq3* by considering the effects of personality differences in the model by adding the absolute differences between the personality traits of requesters and closers along with the other socio-technical features used in *rq0*.

The results are reported in Table 10. Regarding *openness*, the difference between the requester and the closer is positive and significant (1.07) only in the replication. We observe consistent results regarding the difference in the levels of *conscientiousness* and *extraversion*, which have a positive effect in both studies. Instead, we observe contrasting results for the difference in the levels of *agreeableness* and *neuroticism*—negative in the replication (respectively, 0.94 and 0.93) and positive in the original study (1.02 and 1.22).

Table 10. Odds ratio of the mixed-effect model with personality differences between the requester and closer from the original study by Iyer et al. [42] and the replication. Significant results are shown in **bold** (sig: $^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$).

	Original study		Replication	
Variables	**Single run**	**Bootstrap.**	**Single run**	**Bootstrap.**
	Odds ratio	**95% CI**	**Odds ratio**	**95% CI**
(Intercept)	3.34 ***	-	2.68 ***	-
test_file	**1.09** ***	[1.05, 1.12]	**1.09** ***	[1.05, 1.12]
total_churn	**0.92** ***	[0.90, 0.93]	**0.90** ***	[0.89, 0.91]
social_distance	**1.81** ***	[1.86, 2.03]	**2.34** ***	[2.55, 2.91]
num_comments	**0.66** ***	[0.65, 0.67]	**0.68** ***	[0.67, 0.68]
prior_interaction	**1.66** ***	[1.63, 1.69]	**1.48** ***	[1.46, 1.53]
followers_current	**1.07** ***	[1.06, 1.11]	**1.07** ***	[1.06, 1.11]
main_team_member	**1.27** ***	[1.23, 1.31]	**1.22** ***	[1.17, 1.25]
age_current	**0.92** ***	[0.87, 0.93]	1.01	[0.95, 1.05]
team_size	0.96	[0.89, 1.00]	0.94	[0.87, 0.99]
stars_current	**0.55** ***	[0.44, 0.50]	**0.53** ***	[0.46, 0.49]
diff_openness_abs	1.01	[1.01, 1.04]	**1.07** ***	[1.06, 1.14]
diff_conscientiousness_abs	**1.29** ***	[1.29, 1.35]	**1.25** ***	[1.28, 1.34]
diff_extraversion_abs	**1.12** ***	[1.11, 1.16]	**1.31** ***	[1.32, 1.44]
diff_agreeableness_abs	**1.02** **	[1.00, 1.04]	**0.94** ***	[0.89, 0.95]
diff_neuroticism_abs	**1.22** ***	[1.21, 1.27]	**0.93** ***	[0.88, 0.94]
test_file x num_comments	**1.11** ***	[1.09, 1.15]	**1.13** ***	[1.10, 1.17]
total_churn x num_comments	**1.06** ***	[1.05, 1.07]	**1.07** ***	[1.06, 1.09]
social_connection x num_comments	**0.93** ***	[0.89, 0.97]	**0.93** ***	[0.90, 0.98]
num_comments x prior_interaction	**1.05** ***	[1.05, 1.08]	**1.05** ***	[1.05, 1.07]
R^2m	0.13		0.12	
R^2c	0.56		0.58	
AIC	390,495		333,114	

5.2.2 *Replication of Calefato et al. (2019a).*

In this section, we answer the six research questions *rq1-6* presented in the original study by Calefato et al. [17]. In the replication, we recreate the related figures and tables for comparison.

Calefato et al. conducted a preliminary analysis to rule out changes in personality over time. For each of the $N = 211$ developers in the dataset, they computed monthly-based personality scores, then split the set by date into two subsets of approximately the same size. For each trait, they averaged the scores in each subset, thus obtaining two observations for each developer (i.e., early vs. later). Finally, for each trait, they performed a Wilcoxon Signed-Rank test to verify the null hypothesis that the median difference between pairs of observations (i.e., for each developer) was not significantly different from zero. Table 11 reports the results from the five paired tests in both the original study and the replication. We replicate the same tests after replacing the original dataset, containing personality scores obtained from IBM PI, with the new dataset, containing the scores obtained using LIWC. The results show no significant differences between the distributions (all adjusted p-values > 0.05 after Bonferroni correction for multiple tests), thus confirming the stability of personality traits over time with both personality tools.

rq1—Are there groupings of similar developers according to their personality profile?

Table 11. Results of the Wilcoxon Signed-Rank tests for assessing changes in mean personality traits over time in the original study by Calefato et al. [17] and the replication (all p-values > 0.05 after Bonferroni correction)

Trait	**Original study**			**Replication**		
	V	**p-value**	**95% CI**	**V**	**p-value**	**95% CI**
Openness	6,109	0.589	[-0.002, -0.003]	9,330	1,000	[-0.006, 0.029]
Conscientiousness	5,575	0.661	[-0.004, -0.003]	8,320	1,000	[-0.014, 0.014]
Extraversion	5,839	0.964	[-0.003, -0.003]	7,751	1,000	[-0.022, 0.008]
Agreeableness	5,871	0.917	[-0.003, -0.003]	7,199	0.448	[-0.028, 0.002]
Neuroticism	5,915	0.853	[-0.003, -0.004]	9,075	1,000	[-0.008, 0.023]

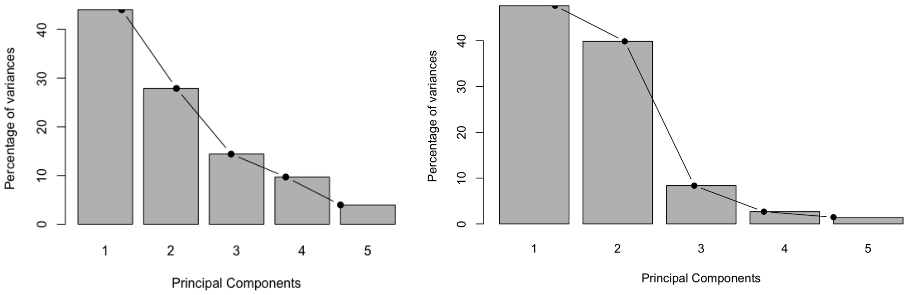
To answer the first research question, Calefato et al. applied several techniques to reveal the presence of natural groupings of personalities within the dataset of $N = 211$ developers. We replicate the same analyses presented in Calefato et al. [17] on the new dataset.

First, to ensure that original data was suitable for structure detection, Calefato et al. computed the Kaiser-Meyer-Olkin measure (0.5, the minimum value recommended in literature [31]) and Barlett's test of sphericity (χ^2 = 4088.32, p < 0.001). We obtain similar results with the new dataset (KMO = 0.5; χ^2 = 900, p < 0.001). Accordingly, we proceed with the analyses to uncover latent factors.

The first analysis performed was the Principal Component Analysis (PCA), a statistical procedure that converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables, i.e., the principal components.

The scree plots in Fig. 4 show the percentage of variance in the data for each of the five components extracted from the data. In the original study, the first three components accounted for most of the variance in the data (86%, see Fig. 4a), whereas in the replication the first two account for nearly 88% (Fig. 4b). The analysis of the eigenvalues in Table 12 shows that only the first two components in each replication have a value over Kaiser's criterion of 1, the cut-off point typically used to retain principal components. Eigenvalues correspond to the amount of the variation explained by each principal component; therefore, a latent component has an eigenvalue > 1 when it accounts for more variance than its accounted for by the original variables in a dataset. Next, we check how the traits load on the two extracted components. The loadings from the original study and the replication are reported in Table 13, from which we observe inconsistent results. In the original study, *openness* and *neuroticism* load on the first component, whereas *conscientiousness*, *extraversion*, and *agreeableness* on the second. Conversely, in the replication *openness*, *extraversion*, and *agreeableness* load on the first component, whereas *conscientiousness* and *neuroticism* load on the second; we also observe that *openness* and *conscientiousness* load negatively on their respective component.

After applying PCA, Calefato et al. applied the k-means clustering algorithm to extract clusters of developers' personalities. We replicate the analysis on the new dataset and use the 'elbow' method to identify the optimal number of clusters. The elbow point corresponds to the smallest k value after which is not observed a large decrease in the within-group heterogeneity—measured using the sum of squares—with the increase of the number of clusters. The scree plots from both studies are reported in Appendix C, Fig. 10. In the original study, Calefato et al. selected $k = 3$, whereas we choose $k=2$. Table 14 shows the distribution of the developers across the personality clusters extracted in the two studies. The developers are fairly evenly distributed across the clusters in the original study. In the replication, the first cluster is twice the size of the second, though we



(a) Original study

(b) Replication

Fig. 4. Percent of variance explained by the principal components in the original study by Calefato et al. [17] (a) and the replication (b).

Table 12. Eigenvalues returned by PCA in the original study and the replication. Only the components in **bold** with eigenvalue >1 are retained.

	Original study		**Replication**	
	Eigenvalue	**% of variance**	**Eigenvalue**	**% of variance**
Component 1	**2.201**	44.023	**2.382**	47.63
Component 2	**1.394**	27.885	**1.993**	39.86
Component 3	0.721	14.419	0.418	8.36
Component 4	0.485	9.705	0.134	2.67
Component 5	0.198	3.967	0.073	1.47

Table 13. Standardized loadings for the extracted principal components in the original study by Calefato et al. [17] and the replication.

Trait	**Original study**		**Replication**	
	Component 1	**Component 2**	**Component 1**	**Component 2**
Openness	**0.79**	0.03	**-0.861**	0.414
Conscientiousness	**0.69**	0.44	0.118	**0.942**
Extraversion	0.27	**0.74**	**0.849**	0.132
Agreeableness	-0.15	**0.92**	**0.867**	0.357
Neuroticism	**0.89**	0.04	0.001	**-0.971**

observed an even larger imbalance with larger k values. The table also reports the coordinates of the centroids—the average position of the elements assigned to a cluster. All the values are z-score standardized, with positive (negative) values above (below) the overall means. Then, Calefato et al. performed five non-parametric Kruskal-Wallis tests (one per trait), to understand whether the trait distributions in the clusters are significantly different, followed by a Tukey and Kramer (Nemenyi) *post hoc* test for multiple pairwise comparisons, to understand which pairs are indeed different. Table 15 shows the results of the Kruskal-Wallis tests, after applying Bonferroni correction to the p-values for repeated tests. In the original study, the Kruskal-Wallis test for each of the five traits was statistically significant ($p < 0.001$) with a relatively strong ($\epsilon^2 \geq 0.16$) or strong effect size

Table 14. Size and centers of the three clusters extracted with k-means in the original study by Calefato et al. [17] and the replication. The highest ▲ and lowest ▼ values per trait are shown in **bold**.

Original study

Cluster (size)	**Openness**	**Conscient.**	**Extraver.**	**Agreeabl.**	**Neurotic.**
Cluster 1 (76)	**-0.74▼**	**-0.69▼**	-0.06	0.37	**-0.84▼**
Cluster 2 (55)	**0.90▲**	**0.86▲**	**0.99▲**	**0.45▲**	**0.81▲**
Cluster 3 (80)	0.08	0.07	**-0.62▼**	**-0.67▼**	0.25

Replication

Cluster (size)	**Openness**	**Conscient.**	**Extraver.**	**Agreeabl.**	**Neurotic.**
Cluster 1 (156)	**0.50▲**	**0.16▲**	**-0.36▼**	**-0.32▼**	**-0.22▼**
Cluster 2 (76)	**-1.03▼**	**-0.33▼**	**0.77▲**	**0.65▲**	**0.45▲**

Table 15. Results of the Kruskal-Wallis tests for the comparisons of the distributions of each personality trait scores across the clusters in the original study by Calefato et al. [17] and the replication. All p-values < 0.001 after Bonferroni correction.

Trait	**Original study**				**Replication**			
	χ^2	**p-value**	ϵ^2	**95% CI**	χ^2	**p-value**	ϵ^2	**95% CI**
Openness	87.836	<0.001	0.418	[0.297, 0.532]	136	<0.001	0.590	[0.505, 0.658]
Conscienti.	78.777	<0.001	0.375	[0.257, 0.495]	17	<0.001	0.073	[0.021, 0.149]
Extraversion	94.554	<0.001	0.450	[0.354, 0.547]	84	<0.001	0.362	[0.264, 0.463]
Agreeablen.	61.248	<0.001	0.292	[0.197, 0.401]	62	<0.001	0.270	[0.170, 0.377]
Neuroticism	107.560	<0.001	0.512	[0.410, 0.613]	24	<0.001	0.104	[0.041, 0.185]

($\epsilon^2 \geq 0.36$) [96]; however the *post hoc* tests showed that only clusters 1 & 3 and clusters 2 & 3 were significantly different from each other. In the replicated study, the Kruskal-Wallis tests are also all significant ($p < 0.001$), with the effect size ranging between moderate ($\epsilon^2 \geq 0.04$) and strong. Because there are only two clusters, there is no need to run a *post hoc* test to affirm that there are significant differences among the trait distributions.

Finally, Calefato et al. applied Archetypal Analysis. In Appendix C, Fig. 11 we report the scree plots used to identify the optimal number of archetypes to extract with the elbow criterion. The two plots show the fraction of total variance in the data explained by the number of extracted archetypes. In the original study, Calefato et al. extracted three archetypes; in the replication, the function also plateaus after three archetypes. Therefore, also for the ease of comparison with the original study, we opt for extracting three archetypes. Table 16 shows the trait coordinates, standardized for ease of comparison, for the archetypes extracted in the two studies. Looking at the trait coordinates, the archetypal analyses in the two studies do not extract similar phenotypes of developers' personalities.

rq2—Do developers' personality traits vary with the type of contributors (i.e., core vs. peripheral?)

In the second research question, Calefato et al. investigated whether the members of projects' core development teams exhibit different personality traits. Accordingly, they first filtered the personality scores retaining only the $N = 118$ commit authors and then split this set into two subgroups, namely peripheral developers (i.e., external contributors without commit access to the repositories, $N = 62$) and core developers (i.e., project members with write access to the source

Table 16. The archetypes extracted in the original study by Calefato et al. [17] and the replication. The highest ▲ and lowest ▼ standardized values per trait are shown in **bold**.

Original study

Archetype	Openness	Conscient.	Extraver.	Agreeabl.	Neuroticism
Archetype 1	0.51	-0.13	**-0.81▼**	**-1.09▼**	**0.61▲**
Archetype 2	**0.64▲**	**1.06▲**	**1.12▲**	**0.87▲**	0.54
Archetype 3	**-1.15▼**	**-0.93 ▼**	-0.31	0.23	**-1.15▼**

Replication

Archetype	Openness	Conscient.	Extraver.	Agreeabl.	Neuroticism
Archetype 1	**0.32▼**	**1.30▲**	**-0.68▲**	**0.35▲**	**-1.28▼**
Archetype 2	**1.10▲**	**0.72▼**	**-1.36▼**	**-0.62▼**	**0.17▲**
Archetype 3	1.08	1.03	-1.01	-0.31	-0.80

Table 17. Results of the Wilcoxon Rank Sum tests for the unpaired comparison of median personality trait scores between core and peripheral developers in the original study by Calefato et al. [17] and the replication. All p-values > 0.05 after Bonferroni correction.

Trait	Original study			Replication		
	W	p-value	95% CI	W	p-value	95% CI
Openness	1,583	1.000	[-0.009, 0.008]	2,233	0.500	[-0.009, 0.087]
Conscientiousness	1,625	1.000	[-0.010, 0.011]	1,989	1.000	[-0.022, 0.034]
Extraversion	1,575	1.000	[-0.010, 0.008]	1,902	1.000	[-0.041, 0.041]
Agreeableness	1,273	0.271	[-0.017, 0.000]	1,685	1.000	[-0.064, 0.018]
Neuroticism	2,051	0.063	[0.004, 0.027]	1,751	1.000	[-0.049, 0.021]

code repository, $N = 56$). To replicate the analysis, we perform for each trait a Wilcoxon Rank Sum test for unpaired group comparison on the dataset with the new personality scores.

The results from both the original study and the replication are reported in Table 17. Consistently, in both studies, we observe no significant differences (i.e., all adjusted p-values $>$ 0.05, after Bonferroni correction) across all five traits. As such, the two studies consistently find that, on average, the personalities of core developers are not significantly different from those of peripheral developers.

rq3—Do developers' personality traits change after becoming a core member of a project's development team?

In the third research question, Calefato et al. investigated whether developers exhibit different personality traits after becoming members of a project's core development team. Accordingly, for each of the $N = 56$ core developers with write access to source code repositories, they first retrieved the date of the first commit accepted and integrated by them, as an approximation of the moment when they have become a member of a project's core development team. Then, for any of the projects they gained membership for, they used that date to split the personality trait scores of the developers into two paired groups, i.e., before vs. after becoming a project's core team member. We replicate the same analyses on the new dataset.

In Appendix C, Fig. 12 we report the boxplots of the five personality scores across the two groups in both the original study and the replication. Also, Table 18 reports the results of the five Wilcoxon Signed- Rank tests executed (one per trait). No significant differences are returned by the tests in

Table 18. Results of the Wilcoxon Signed-Rank tests for the paired comparison of mean personality trait scores of developers before vs. after becoming members of a project's core team in the original study by Calefato et al. [17] and the replication. All p-values > 0.05 after Bonferroni correction.

Trait	Original study			Replication		
	V	p-value	95% CI	V	p-value	95% CI
Openness	39	1.000	[-0.011, 0.034]	62	1.000	[-0.074, 0.141]
Conscientiousness	40	1.000	[-0.008, 0.031]	76	0.765	[-0.014, 0.112]
Extraversion	17	1.000	[-0.019, 0.019]	46	1.000	[-0.126, 0.071]
Agreeableness	15	1.000	[-0.038, 0.011]	49	1.000	[-0.121, 0.081]
Neuroticism	43	0.654	[-0.005, 0.048]	35	1.000	[-0.118, 0.018]

both studies (all adjusted p-values > 0.05 after Bonferroni correction). As such, the two studies consistently find that developers' personality does not change significantly after becoming a core developer.

rq4—Do developers' personality traits vary with the degree of development activity?

Calefato et al. investigated whether more productive developers are characterized by specific personality trait levels. Using the same groups of core ($N = 56$) and peripheral ($N = 62$) developers created earlier for *rq2*, they further split the two sets according to the level of development activity. Specifically, they found the mean number of commits authored by each developer in the peripheral group and split it into two subsets, i.e., authored-commits-high ($N = 17$) and authored-commits-low ($N = 45$). Similarly, they created the integrated-commits-high ($N = 44$) and integrated-commits-low ($N = 8$) subgroups considering the mean number of commits integrated (i.e., accepted) by the developers in the core group. We replicate this research question on the new dataset by applying a series of Wilcoxon Rank Sum tests to make unpaired comparisons of the median personality scores between high- and low-activity developers.

The results from the original study and the replications are shown in Table 19. The test results reveal no cases of statistically significant differences between the pairs of trait distributions in both studies (i.e., adjusted p-values > 0.05 after Bonferroni correction). As such, the two studies consistently find that developers' personality does not vary significantly with the level of development activity.

rq5—What personality traits are associated with the likelihood of becoming a project contributor?

To answer the fifth research question, Calefato et al. built a contribution likelihood model, that is, they fit a logistic regression model to study the associations between the personality traits of developers and the likelihood of having a contribution accepted. As the response variable, they used a dichotomous yes/no variable indicating whether a developer has authored at least one commit successfully integrated into a project repository. They also included a couple of control variables, namely word_count, a proxy for the extent of communication and social activity of a developer in the community through email messages from which personality traits are extracted, and project_age, measured as number of days. In the replication, we follow the same process described in the original. The dataset of $N = 211$ developers is fairly balanced with respect to the response variable, as 118 developers have at least one commit and 93 have no commits. Before fitting the model to the new dataset, we first check for the presence of high Pearson correlations between the updated personality predictors; We drop *neuroticism* and *extraversion* because they show a high correlation (.70) with *conscientiousness* and *agreeableness*, respectively. Also, the Variance Inflation Factor (VIF) computed on the resulting model reveals no collinearity issues for the retained predictors (all values < 3). Then, we evaluate the model fit using McFadden's pseudo-R^2 measure,

Table 19. Results of the Wilcoxon Rank Sum test for the unpaired comparison of median personality trait scores between developers with high vs. low degree of development activity in the original study by Calefato et al. [17] and the replication. All p-values > 0.05 after Bonferroni correction.

Original study

	Trait	W	p-value	95% CI
High vs. low commit authors (peripheral devs)	Openness	476	1.000	[-0.004, 0.021]
	Conscientiousness	449	1.000	[-0.008, 0.024]
	Extraversion	383	1.000	[-0.017, 0.017]
	Agreeableness	341	1.000	[-0.018, 0.009]
	Neuroticism	408	1.000	[-0.013, 0.017]
High vs. low commit integrators (core devs)	Openness	193	1.000	[-0.014, 0.020]
	Conscientiousness	163	1.000	[-0.029, 0.019]
	Extraversion	129	1.000	[-0.028, 0.006]
	Agreeableness	204	1.000	[-0.013, 0.025]
	Neuroticism	151	1.000	[-0.040, 0.017]

Replication

	Trait	W	p-value	95% CI
High vs. low commit authors (peripheral devs)	Openness	520	1.000	[-0.039, 0.122]
	Conscientiousness	557	1.000	[-0.021, 0.078]
	Extraversion	520	1.000	[-0.115, 0.041]
	Agreeableness	489	1.000	[-0.078, 0.100]
	Neuroticism	389	1.000	[-0.084, 0.035]
High vs. low commit integrators (core devs)	Openness	165	1.000	[-0.139, 0.046]
	Conscientiousness	253	1.000	[-0.035, 0.121]
	Extraversion	242	1.000	[-0.050, 0.084]
	Agreeableness	242	1.000	[-0.052, 0.102]
	Neuroticism	183	1.000	[-0.105, 0.038]

which describes the proportion of variance in the response variable explained by the model, and the Area Under the ROC curve (AUC), to assess the classification ability of the contribution model as compared to random guessing.

Table 20 includes the results of the two logistic regression models. We observe that the control variable project_age is statistically significant ($p < 0.001$) and has a negative effect in both the original study (-0.420) and the replication (-0.394). In the original study, the only statistically significant predictor was openness (54.09, $p < 0.01$), whereas in the replication no personality factor has a significant effect. Also, in terms of goodness of fit, the model in Calefato et al. fit the data slightly better ($R^2 = 0.397$) than the model in the replication ($R^2 = 0.270$). Finally, we replicate the assessment of the model prediction performance computing the AUC. As in the original study, we use a stratified sampling technique to split the dataset into training (70%) and test (30%) sets. The AUC performance of the logistic model in the original study is also better than the replication (0.89 vs 0.69). Overall, while the results from the original study indicated that higher *openness* scores are associated with better chances for developers to become project contributors, the replication suggests instead that personality traits have no effect.

rq6—What personality traits are associated with a higher number of contributions successfully accepted?

Table 20. Logistic regression model of the contribution likelihood as explained by personality traits in the original study by Calefato et al. [17] and the replication. Significant results are shown in **bold** (sig: $^{**}p < 0.01$, $^{***}p < 0.001$).

Original study

	Coef. Est.	**Std. Error**	**z-value**
(Intercept)	-29.523	20.175	-1.44
project_age (days)	**-0.420** ***	0.113	-3.71
log(word_count)	0.199	0.204	0.98
openness	**54.092** **	23.338	2.32
conscientiousness	-18.994	26.623	-0.71
extraversion	-4.652	16.939	-0.27
agreeableness	18.620	22.525	0.83
neuroticism	-19.710	16.939	-1.07
N=211, McFadden Pseudo-R^2=0.397, AUC=0.89			

Replication

	Coef. Est.	**Std. Error**	**z-value**
(Intercept)	1.310	12.867	0.10
project_age (days)	**-0.366** ***	0.080	-4.55
log(word_count)	-0.058	0.146	-0.40
openness	-0.532	2.823	-0.19
conscientiousness	1.027	3.021	0.34
extraversion	0.538	2.505	0.34
N=211, McFadden Pseudo-R^2=0.270, AUC=0.69			

To answer the last research question, Calefato et al. performed a regression analysis to evaluate the association between the personality traits of developers and the number of contributions (i.e., commits) that they got accepted (i.e., merged) into the project repository. As the independent variables, they used the same personality predictors used in the previous logistic regression analysis. Regarding the control variables, in addition to the count of words in emails and the age of projects, they added two more (is_integrator and track_record) to control for, respectively, core members and long-time contributors. The dependent variable used is the number of merged commits, i.e., the count of commits authored by a developer that have been successfully merged. Because the dependent variable takes non-negative integer values only, rather than fitting a linear model, Calefato et al. performed a count-data regression analysis, which handles non-negative observations. Here we follow the same process described in the original work. Different count data models can be used for estimations, depending on the characteristics of the data. Poisson distributions have a strong assumption on equidispersion, that is, the equality of mean and variance of the count-dependent variable. Alternatively, it is possible to use a negative binomial distribution, a generalization of the Poisson distribution with an additional parameter to accommodate the overdispersion. We perform the Likelihood Ratio Test (LRT) of overdispersion and find out that, as in the original study, the negative binomial model (LogLik = -1023, $\chi^2 = 635$, $p < 0.001$) provides a better fit to the data than the Poisson model (LogLik =1340).

Table 21 shows the results of the count-data regression analysis with the negative binomial models from the two studies. We observe that, except for word_count, all the control variables have a statistically significant effect in both studies. Instead, while in the original study none of

Table 21. Developers productivity model in the original study by Calefato et al. [17] and the replication. The response is the count of commits successfully merged. The number of observations (commit data) is $N = 471$, coming from 211 developers, of whom 118 have made at least one commit. Significant results are shown in **bold** (sig: * $p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$).

Original study

	Coef. Estimate	Std. Error	z value
(Intercept)	0.807	0.234	3.43
project_age (days)	**-0.068** *	0.044	-1.56
dev_is_integrator=TRUE	**0.648** **	0.221	2.93
dev_track_record (days)	**0.544** ***	0.033	16.21
log(word_count)	0.003	0.030	0.12
openness	0.036	0.068	0.53
conscientiousness	0.005	0.072	0.08
extraversion	0.046	0.066	0.71
agreeableness	-0.039	0.054	-1.80
neuroticism	0.141	0.078	-1.80
N=471, LogLik=-917, LRT χ^2=514 McFadden Pseudo-R^2=0.115			

Replication

	Coef. Estimate	Std. Error	z value
(Intercept)	1.169	0.175	6.67
project_age (days)	**-0.087** *	0.042	-2.06
dev_is_integrator=TRUE	**0.474** *	0.204	2.33
dev_track_record (days)	**0.566** ***	0.033	17.11
log(word_count)	-0.042	0.024	-1.76
openness	0.004	0.054	0.07
conscientiousness	**0.123** *	0.051	2.42
extraversion	-0.054	0.058	-0.92
N=471, LogLik=-1022.984, LRT χ^2=635 McFadden Pseudo-R^2=0.109			

the five predictors related to personality has a significant effect, in the replication we find that *conscientiousness* is significantly and positively associated with a higher number of integrated commits (0.123, $p < 0.05$). Finally, both the model in the original study ($R^2 = 0.115$) and the replication ($R^2 = 0.109$) fit the data marginally. Overall, while the results from the original study indicated that personality traits do not affect commit productivity, the replication suggests instead that higher *conscientiousness* scores are associated with a higher number of accepted commits.

5.3 Threats to Validity

Because in the replications we followed the same methodologies presented in the original studies, we have also inherited some of the threats to the validity of those papers, e.g., that the datasets used in Iyer et al. [42] and Calefato et al. [17] are respectively not representative of GitHub and the Apache ecosystem as a whole. Also, albeit one may argue that some of the statistics applied in Sect. 5.2 may not be the preferred approach, we applied them to support the comparative aspects of the replication.

Phase 2 – Summary

The choice of a personality detection tool does affect the validity of previously published results. The replication of the first study led to contrasting findings in all the three original research questions aimed to assess the effects of personality on pull-request acceptance. When replicating the second study, we were able to obtain consistent findings for only three original research questions out of six.

6 DISCUSSION

The challenges and potential of computational personality detection in software engineering research. There has been considerable interest in applying natural language processing (NLP) and computational linguistics to recent software engineering research. In particular, prior work on *sentiment analysis* (also referred to as *opinion mining*) has focused on analyzing corpora of technical text, such as emails, commit comments, code-review discussions, and app reviews, to detect the polarity [14, 82], emotions [16, 81], opinions [56, 119], and intentions [26, 41] in software developers' interactions.

One of the reasons for such widespread interest is that sentiment analysis is a highly restricted NLP problem because, to solve it, tools do not need to fully understand the semantics of each sentence or document but only some aspects of it, i.e., positive or negative sentiments and their target entities or topics [59]. Computational personality detection is also an NLP problem as it touches every aspect of the research field, e.g., co-reference resolution, negation handling, named-entity recognition, and word-sense disambiguation [18]. However, when the NLP aspects are addressed, the related constructs are then used to support the further analyses needed to extract personality profiles, which represents a separate problem. The broad scope of the problem arguably explains the poor agreement among the currently available personality detection tools and the negative results of the replications, discussed next in this section. Also, analogously to sentiment analysis, we expect that computational personality detection in software engineering will benefit from future breakthroughs and advancements in NLP [104].

Furthermore, while sentiment analysis and personality detection are under the same umbrella of *affective computing* research and, therefore, adopt similar technological solutions, the 'affective phenomena' they study vary in duration, ranging from short-lived feelings, emotions, and opinions to long-lived, slowly changing personality characteristics [89]. As such, the two research fields can complement each other also when applied to software engineering research, with sentiment analysis concentrating on transient feelings related to entities (e.g., others, themselves, objects, and events) and computational personality detection focusing on intrinsic, long-lasting dispositions (of developers). For example, previous work on sentiment analysis in software engineering has also looked into anger [33] and toxicity [92] detection, and identified cases of developers who lashed out at others during technical discussions. Computational personality detection could complement this research and tell us if those episodes were extemporaneous or rather the effect of a personal inclination of developers who, despite their technical skills and knowledge, might not be recommended, for example, for tasks such as mentoring newcomers. Prior research on onboarding has developed recommender systems that look at developers' social and technical aspects to help newcomers identify mentors in OSS projects [19, 112]; given that personality mismatch between mentors and mentees has been identified as one of the social barriers to onboarding [9], a potential follow-up study could take into account the most relevant traits for the task according to personality theories (e.g., being more agreeable and open to collaboration) to identify candidate developers as suitable for mentoring. Another potential scenario of usage concerns code review. As prior work has shown the impact of human factors in performing such activity [99], one might envision the development of a recommender system that assigns developers to code-review tasks also based on their personality profiles, e.g., by preferring those who exhibit a high level of conscientiousness, a trait associated with carrying out tasks precisely and thoroughly.

Phase 1 – Performance Assessment. Because model-based predictions aim to provide an assessment of personality, it is important to establish their convergent validity with self-report scores. Accordingly, to answer RQ1 (*How do off-the-shelf personality detection tools perform in the software engineering domain*), in Phase 1 of the study we have built a dataset of emails written by 50 Apache Software Foundation developers and compared the predictions of four personality detection tools against the self-ratings collected through a questionnaire. In addition, we have made further pairwise comparisons between the tool predictions.

As can be observed from Table 5, the results of the correlation analysis indicate that the tools analyzed neither agree with self-reported personality ratings nor with each other when used in the software engineering domain. The coefficients are worse than those reported in the literature (see Table 3). Consistently, as can be observed from Table 6, the performance accuracy in terms of prediction errors is considerably worse as compared to the results from prior work on automatic personality detection from text (see Table 4). These results should warn the research community about the current limitations when using general-purpose tools for personality prediction in software engineering research.

The disagreements among tools arguable explain why it is hard to synthesize the results from prior work on computational personality detection in software engineering (see the related work presented next in Sect. 7). We currently ignore the reasons for these disagreements, though. On the one hand, a manual error analysis is impracticable in our case. Even if for simplicity we resorted to using trait predictions based on discrete labels (e.g., low vs. high *openness*), the number of words that the tools need to reliably infer traits is in the hundreds or thousands, too large for us to analyze and reason about the root causes of misclassification. On the contrary, sentiment analysis tasks such as polarity detection, which infers the positive vs. negative polarity conveyed through text, rely on machine learning and natural language processing techniques similar to those used for building personality prediction models [79] but typically analyze input text at the sentence level; therefore, through manual analysis, researchers on sentiment analysis in software engineering have identified domain-specific errors limiting model accuracy due to the use of technical words such as *patch* and jargon like *kill a process*, which do not express any valence [80]. On the other hand, the assessed personality detection tools do not explain their outcomes—they are applied in a black-box manner and no information is provided about what steers their model predictions. The complexity and performance of machine learning models have increased over the years at the expense of interpretability. Model interpretability is not a monolithic concept and has many facets [58]. It can be applied at the model level, to refer to algorithmic transparency—a property that is usually very low in deep learning methods whose behavior is notoriously hard to 'mentally simulate' by humans; alternatively, it can be applied at the local level, when models offer *post hoc* explanations—like images, text, or examples—for the output generated in response to a single input instance. Although the need for model interpretability in computational personality detection may not be mandatory because there are no ethical concerns and accountability as in other domains like healthcare and finance, this lack of transparency is nonetheless a drawback that hinders progress in the field. We argue that the research field on computational personality detection might advance and benefit from the development of transparent prediction models that allow for error analysis.

Phase 2 – Replications. Replications play a key role in empirical software engineering so that the research community can build knowledge about which results or observations hold under which conditions [109]. To answer RQ2 (*How does the choice of a personality detection tool affect the validity of previous results in software engineering research?*), in Phase 2 we performed two exact dependent replications in which we kept the same experimental set-up of the original studies while only replacing IBM PI with LIWC to infer the developers' personality scores from written documents.

When replicating the first study by Iyer et al. [42], we have been unable to confirm any of the findings concerning the effect of specific personality traits on the likelihood of merging pull requests in GitHub. In replicating the second study by Calefato et al. [17], we have found consistent results only for three out of six research questions. In particular, we have been able to replicate those research questions that failed to find differences in the distributions of median trait scores between subgroups of developers (e.g., core vs. peripheral, high- vs. low-activity). Instead, we have failed to replicate the other research questions similar to those reported in Iyer et al. [42], where a couple of regression models were built to uncover the effects of specific personality traits on the likelihood of becoming a contributor and the productivity level.

As noted by Ferguson and Heene [30], replicability in science cannot be meaningful without the potential acknowledgment of failed replications. Negative results may happen when experimental results fail to meet expectations due to a lack of effect rather than misaligned expectations or a lack of methodological rigor in poorly designed experiments. Negative results are uncommon in the literature, even rare in software engineering where only recently there have been specific conference tracks or journals' special issues organized to present such results [85]. However, negative results are fundamental in software engineering to embrace the nature of experimentation. In fact, negative results are just as useful as positive ones because, by pointing out what has not worked, they eliminate useless hypotheses and directions, thus redirecting future experimental efforts towards alternative approaches that might pay off [117].

Therefore, though we have failed to replicate the experiments from the two studies, we argue that these negative results can be beneficial to enhance the state of the art of computational personality detection in software engineering. Arguably, the main implication of our findings is that the validity of previous studies, including but not limited to the ones by Iyer et al. [42] and Calefato et al. [17], should be questioned and possibly reassessed. However, this reassessment, as well as future studies on personality detection in software engineering, are possible only after developing and testing a reliable, SE-specific personality detection tool. In fact, the tools used in both the original studies and our replications introduced a threat to validity since all the instruments available for computational personality detection have been trained on non-software engineering-specific text documents, such as essays and social media posts. Hence, the re-assessment should ideally happen using personality detection tools specifically trained on software engineering-specific text corpora. Previous research on sentiment analysis in software engineering has highlighted the benefits (e.g., the reduced misclassification rate of neutral and positive content as emotionally negative) deriving from the use of tools specifically trained on technical documents retrieved from sources like Stack Overflow and Jira [14].

Furthermore, in this work, we have used personality detection tools as off-the-shelf components, i.e., without any tuning or training. Recent work on sentiment analysis has shown that the fine-tuning of tools to the software engineering domain might not be enough to improve accuracy, and that retraining has the potential to adjust the model performance to the shifts in lexical semantics due to different jargon and conventions used in data sources [57, 82]. Instead, we observe a trend in recent work on computational personality detection focused on using state-of-the-art, Deep Learning techniques focused more on outperforming baselines (e.g., [43, 48, 64, 72]) than on releasing reusable, possibly retrainable models that can be transferred to other domains.

7 RELATED WORK

In this section, we focus on reviewing previous studies that investigated the Big Five personality model in the software engineering domain by using tools for automatically extracting personality profiles from communication traces, such as emails, Q&A posts, and code-review comments.

Rigby and Hassan [97] were the first to automatically analyze the personality traits of developers. They studied the personality traits of the four top developers of the Apache `httpd` project against a baseline built by applying LIWC on the entire mailing-list corpus. They found that two of the developers responsible for the major releases have similar personalities, which are also different from the baseline of all other project members.

Licorish and MacDonell [55] combined social network analysis with computational personality detection. They used LIWC to analyze the communication traces of 146 practitioners from the IBM Rational Jazz projects involved in global software development activities and found that those who occupy critical roles in knowledge diffusion demonstrate more *openness* to experience.

Rastogi and Nagappan [95] used LIWC to analyze the personality profiles of nearly 400 GitHub developers. They found that those with different levels of contributions have different personality profiles, i.e., those with high or low levels of contributions are more neurotic. Also, the personality profiles of most active contributors were found to change across two consecutive years, evolving as more conscientious, more extrovert, and less agreeable.

Paruma Pabón et al. [86] used IBM PI to extract the personality traits from e-mails sent by the committers to six Eclipse projects. They found three personality clusters: the first personality groups the committers with the highest scores in *extraversion* and *neuroticism*; the second cluster groups the committers with moderate levels of *neuroticism*; the third cluster groups the committers with low values in *neuroticism.* The three personality clusters are different from those identified by Calefato et al. [17] after analyzing with the same tool the emails written by the Apache Software Foundation developers.

Calefato et al. [15] investigated the relationship between project success and the propensity to trust, one of the *agreeableness* facets in the Five-Factor Model. They approximated the overall performance of two Apache Software Foundation projects with the history of successfully merged pull requests in GitHub. Using the LIWC-based version of IBM PI, they analyzed the word usage in pull request comments to extract the developers' agreeableness scores. The results suggested that the propensity to trust of code reviewers (integrators) is an antecedent of successful pull request integration.

To the best of our knowledge, this study is the first attempt at replicating results from previous work on computational personality detection in software engineering. However, one partial exception is the work of Bazelli et al. [11] who performed a quasi-replication of the study by Rigby and Hassan [97]. Specifically, they used LIWC to infer the personality of Stack Overflow users from Q&A posts. They found that the top reputed authors on Stack Overflow are more extroverted, as compared to medium- and low-reputed users. They argued that such a personality profile is consistent with the one observed by Rigby and Hassan regarding the two top Apache `httpd` developers.

Overall, the findings from these studies show the existence of different clusters of personalities among developers and that their traits vary with their degree of contribution and reputation, while also changing over short periods. Yet, the negative results of our replications suggest that these results should be also reassessed.

8 CONCLUSIONS

In this paper, we have studied the impact of the choice of a personality detection tool when conducting software engineering studies. We have observed a decrease in performance when general-purpose tools are used out of domain as neither they agree with each other nor with the self-reported personality scores. Also, we have observed that the disagreement among tool predictions can lead to diverging conclusions, making it impossible to replicate previously published results when different personality detection tools are used. Our results suggest a need for personality detection tools specially targeted for the software engineering domain. We hope that sharing the complete replication package—the technical corpus annotated with self-reported personality scores and the experimental workflow scripts—can accelerate the advancement in the field.

ACKNOWLEDGMENTS

We are grateful to Filippo Lorè, *Esq.* for his feedback on GDPR compliance. We also thank our CS students Saverio Telera and Marco Iannotta for their help with the replications. Part of the computational work has been executed on the IT resources of the ReCaS-Bari data center.

REFERENCES

[1] Neil R Abramson. 2010. Internal reliability of the Keirsey Temperament Sorter II: cross-national application to American, Canadian, and Korean samples. *Journal of Psychological Type* 70, 2 (2010), 19–30.

[2] Silvia T. Acuña, Marta Gómez, and Natalia Juristo. 2009. How do personality, team processes and task characteristics relate to job satisfaction and software quality? *Information and Software Technology* 51, 3 (mar 2009), 627–639. https://doi.org/10.1016/j.infsof.2008.08.006

[3] Silvia T Acuña, Marta N Gómez, Jo E Hannay, Natalia Juristo, and Dietmar Pfahl. 2015. Are team personality and climate related to satisfaction and software quality? Aggregating results from a twice replicated experiment. *Information and Software Technology* 57 (2015), 141–156.

[4] Zulal Akarsu, Pinar Orgun, Hakan Dinc, Bora Gunyel, and Murat Yilmaz. 2019. Assessing personality traits in a large scale software development company: exploratory industrial case study. In *European Conference on Software Process Improvement*. Springer, 192–206.

[5] Gordon W Allport and Henry S Odbert. 1936. Trait-names: A psycho-lexical study. *Psychological monographs* 47, 1 (1936), i.

[6] Shlomo Argamon, Sushant Dhawle, Moshe Koppel, and James W. Pennebaker. 2005. Lexical predictors of personality type. In *Proceedings of joint annual meeting of the interface and The Classification Society of North America*. 1–16. https://doi.org/10.2105/AJPH.50.1.21

[7] Pierre-Hadrien Arnoux, Anbang Xu, Neil Boyette, Jalal Mahmud, Rama Akkiraju, and Vibha Sinha. 2017. 25 tweets to know you: A new model to predict personality with social media. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 11.

[8] Danny Azucar, Davide Marengo, and Michele Settanni. 2018. Predicting the Big 5 personality traits from digital footprints on social media: A meta-analysis. *Personality and Individual Differences* 124, September 2017 (apr 2018), 150–159. https://doi.org/10.1016/j.paid.2017.12.018

[9] Sogol Balali, Igor Steinmacher, Umayal Annamalai, Anita Sarma, and Marco Aurelio Gerosa. 2018. Newcomers' Barriers. . . Is That All? An Analysis of Mentors' and Newcomers' Barriers in OSS Projects. *Comput. Support. Coop. Work* 27, 3-6 (dec 2018), 679–714. https://doi.org/10.1007/s10606-018-9310-8

[10] Anderson S Barroso, Jamille S Madureira, Michel S Soares, and Rogerio PC do Nascimento. 2017. Influence of human personality in software engineering-a systematic literature review. In *International Conference on Enterprise Information Systems*, Vol. 2. SCITEPRESS, 53–62.

[11] Blerina Bazelli, Abram Hindle, and Eleni Stroulia. 2013. On the Personality Traits of StackOverflow Users. In *2013 IEEE International Conference on Software Maintenance*. IEEE, 460–463. https://doi.org/10.1109/ICSM.2013.72

[12] Gregory J Boyle. 1995. Myers-Briggs type indicator (MBTI): some psychometric limitations. *Australian Psychologist* 30, 1 (1995), 71–74.

[13] Gregory J Boyle and Edward Helmes. 2009. Methods of personality assessment. *The Cambridge handbook of personality psychology* (2009), 110.

[14] Fabio Calefato, Filippo Lanubile, Federico Maiorano, and Nicole Novielli. 2018. Sentiment polarity detection for software development. *Empirical Software Engineering* 23, 3 (2018), 1352–1382.

[15] Fabio Calefato, Filippo Lanubile, and Nicole Novielli. 2017. A preliminary analysis on the effects of propensity to trust in distributed software development. In *2017 IEEE 12th international conference on global software engineering (ICGSE)*. IEEE, 56–60.

[16] Fabio Calefato, Filippo Lanubile, Nicole Novielli, and Luigi Quaranta. 2019. EMTk-the emotion mining toolkit. In *2019 IEEE/ACM 4th International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. IEEE, 34–37.

[17] Fabio Calefato, Filippo Lanubile, and Bogdan Vasilescu. 2019. A large-scale, in-depth analysis of developers' personalities in the Apache ecosystem. *Information and Software Technology* 114 (2019), 1–20.

[18] Erik Cambria, Bjorn Schuller, Yunqing Xia, and Catherine Havasi. 2013. New avenues in opinion mining and sentiment analysis. *IEEE Intell. Syst.* 28, 2 (mar 2013), 15–21. https://doi.org/10.1109/MIS.2013.30

[19] Gerardo Canfora, Massimiliano Di Penta, Rocco Oliveto, and Sebastiano Panichella. 2012. Who is going to mentor newcomers in open source projects?. In *Proc. ACM SIGSOFT 20th Int. Symp. Found. Softw. Eng. - FSE '12*. ACM Press, New York, New York, USA, 1. https://doi.org/10.1145/2393596.2393647

[20] Giulio Carducci, Giuseppe Rizzo, Diego Monti, Enrico Palumbo, and Maurizio Morisio. 2018. TwitPersonality: Computing Personality Traits from Tweets Using Word Embeddings and Supervised Learning. *Information* 9, 5 (2018). 10.3390/info9050127

[21] Fabio Celli, Fabio Pianesi, David Stillwell, and Michal Kosinski. 2013. Workshop on computational personality recognition: Shared task. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 7.

[22] Max Coltheart. 1981. The MRC psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A* 33, 4 (1981), 497–505.

[23] Paul T Costa and Robert R McCrae. 1985. *The NEO personality inventory.* Psychological assessment resources Odessa, FL.

[24] Paul T Costa Jr and Robert R McCrae. 2008. *The Revised NEO Personality Inventory (NEO-PI-R).* Sage Publications, Inc.

[25] Shirley Cruz, Fabio QB da Silva, and Luiz Fernando Capretz. 2015. Forty years of research on personality in software engineering: A mapping study. *Computers in Human Behavior* 46 (2015), 94–113.

[26] Andrea Di Sorbo, Sebastiano Panichella, Corrado A. Visaggio, Massimiliano Di Penta, Gerardo Canfora, and Harald C. Gall. 2015. Development Emails Content Analyzer: Intention Mining in Developer Discussions. In *2015 30th IEEE/ACM Int. Conf. Autom. Softw. Eng.* IEEE, 12–23. https://doi.org/10.1109/ASE.2015.12

[27] J M Digman. 1990. Personality Structure: Emergence of the Five-Factor Model. *Annual Review of Psychology* 41, 1 (jan 1990), 417–440. https://doi.org/10.1146/annurev.ps.41.020190.002221

[28] M Brent Donnellan, Frederick L Oswald, Brendan M Baird, and Richard E Lucas. 2006. The mini-IPIP scales: tiny-yet-effective measures of the Big Five factors of personality. *Psychological assessment* 18, 2 (2006), 192.

[29] Golnoosh Farnadi, Geetha Sitaraman, Shanu Sushmita, Fabio Celli, Michal Kosinski, David Stillwell, Sergio Davalos, Marie Francine Moens, and Martine De Cock. 2016. Computational personality recognition in social media. *User Modelling and User-Adapted Interaction* 26, 2-3 (2016). https://doi.org/10.1007/s11257-016-9171-0

[30] Christopher J Ferguson and Moritz Heene. 2012. A vast graveyard of undead theories: Publication bias and psychological science's aversion to the null. *Perspectives on Psychological Science* 7, 6 (2012), 555–561.

[31] Andy Field, Jeremy Miles, and Zoë Field. 2012. *Discovering statistics using R.* Sage publications.

[32] Robert L Fowler. 1987. Power and robustness in product-moment correlation. *Applied Psychological Measurement* 11, 4 (1987), 419–428.

[33] Daviti Gachechiladze, Filippo Lanubile, Nicole Novielli, and Alexander Serebrenik. 2017. Anger and its direction in collaborative software development. In *2017 IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Technologies Results Track (ICSE-NIER)*. IEEE, 11–14.

[34] Annie T. Ginty. 2013. *Psychometric Properties.* Springer New York, New York, NY, 1563–1564. https://doi.org/10.1007/978-1-4419-1005-9_480

[35] Jennifer Golbeck, Cristina Robles, Michon Edmondson, and Karen Turner. 2011. Predicting Personality from Twitter. In *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing*. IEEE, 149–156. https://doi.org/10.1109/PASSAT/SocialCom.2011.33 arXiv:EJ551000 - ERIC

[36] Jennifer Goldbeck, Cristina Robles, and Karen Turner. 2011. Predicting personality with social media. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA '11*. 253. https://doi.org/10.1145/1979742.1979614

[37] Lewis R Goldberg. 1981. Language and individual differences: The search for universals in personality lexicons. *Review of personality and social psychology* 2, 1 (1981), 141–165.

[38] Lewis R. Goldberg. 1993. The structure of phenotypic personality traits. *The American psychologist* 48, 1 (jan 1993), 26–34. https://doi.org/10.1037/0003-066X.48.1.26

[39] Lewis R Goldberg, John A Johnson, Herbert W Eber, Robert Hogan, Michael C Ashton, C Robert Cloninger, and Harrison G Gough. 2006. The international personality item pool and the future of public-domain personality measures. *Journal of Research in personality* 40, 1 (2006), 84–96.

[40] Margeret Hall and Simon Caton. 2017. Am I who I say I am? Unobtrusive self-representation and personality recognition on Facebook. *PloS one* 12, 9 (2017), e0184417.

[41] Qiao Huang, Xin Xia, David Lo, and Gail C. Murphy. 2020. Automating Intention Mining. *IEEE Trans. Softw. Eng.* 46, 10 (oct 2020), 1098–1119. https://doi.org/10.1109/TSE.2018.2876340

[42] Rahul N Iyer, S Alex Yun, Meiyappan Nagappan, and Jesse Hoey. 2019. Effects of personality traits on pull request acceptance. *IEEE Transactions on Software Engineering* (2019).

[43] Hang Jiang, Xianzhe Zhang, and Jinho D Choi. 2020. Automatic Text-Based Personality Recognition on Monologues and Multiparty Dialogues Using Attentive Networks and Contextual Embeddings (Student Abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 13821–13822.

[44] Oliver P John and Sanjay Srivastava. 1999. The Big Five Trait taxonomy: History, measurement, and theoretical perspectives. In *Handbook of personality: Theory and research, 2nd ed.*, L.A. Pervin and O.P. John (Eds.). Guilford Press, 102–138.

[45] Robbert Jongeling, Proshanta Sarkar, Subhajit Datta, and Alexander Serebrenik. 2017. On negative results when using sentiment analysis tools for software engineering research. *Empirical Software Engineering* 22, 5 (2017), 2543–2584.

[46] Tanjila Kanij, Robert Merkel, and John Grundy. 2015. An Empirical Investigation of Personality Traits of Software Testers. In *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*. IEEE, 1–7. https://doi.org/10.1109/CHASE.2015.7

[47] Vishal Kaushal and Manasi Patwardhan. 2018. Emerging trends in personality identification using online social networks—a literature survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12, 2 (2018), 1–30.

[48] Amirmohammad Kazameini, Samin Fatehi, Yash Mehta, Sauleh Eetemadi, and Erik Cambria. 2020. Personality Trait Detection Using Bagged SVM over BERT Word Embedding Ensembles. In *Proceedings of the The Fourth Widening Natural Language Processing Workshop*. Association for Computational Linguistics.

[49] David Keirsey. 1998. *Please understand me II: Temperament, character, intelligence.* Prometheus Nemesis Book Company.

[50] Michal Kosinski, David Stillwell, and Thore Graepel. 2013. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the national academy of sciences* 110, 15 (2013), 5802–5805.

[51] Makrina Viola Kosti, Robert Feldt, and Lefteris Angelis. 2016. Archetypal personalities of software engineers and their work preferences: a new perspective for empirical studies. *Empirical Software Engineering* 21, 4 (2016), 1509–1532.

[52] Asadzadeh Laleh and Rahimi Shahram. 2017. Analyzing Facebook activities for personality recognition. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 960–964.

[53] Per Lenberg, Robert Feldt, and Lars Göran Wallgren. 2015. Behavioral software engineering: A definition and systematic literature review. *Journal of Systems and software* 107 (2015), 15–37.

[54] Sherlock Licorish, Anne Philpott, and Stephen G MacDonell. 2009. Supporting agile team composition: A prototype tool for identifying personality (in) compatibilities. In *2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*. IEEE, 66–73.

[55] Sherlock A. Licorish and Stephen G. MacDonell. 2015. Communication and personality profiles of global software developers. *Information and Software Technology* 64 (2015), 113–131. https://doi.org/10.1016/j.infsof.2015.02.004

[56] Bin Lin, Fiorella Zampetti, Gabriele Bavota, Massimiliano Di Penta, and Michele Lanza. 2019. Pattern-based mining of opinions in q&a websites. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 548–559.

[57] Bin Lin, Fiorella Zampetti, Gabriele Bavota, Massimiliano Di Penta, Michele Lanza, and Rocco Oliveto. 2018. Sentiment Analysis for Software Engineering: How Far Can We Go?. In *Proceedings of the 40th International Conference on Software Engineering* (Gothenburg, Sweden) *(ICSE '18)*. Association for Computing Machinery, New York, NY, USA, 94–104. https://doi.org/10.1145/3180155.3180195

[58] Zachary C. Lipton. 2018. The Mythos of Model Interpretability. *Commun. ACM* 61, 10 (Sept. 2018), 36–43. https://doi.org/10.1145/3233231

[59] Bing Liu. 2012. Sentiment Analysis and Opinion Mining. In *Synth. Lect. Hum. Lang. Technol.* Vol. 5. 1–167.

[60] Fei Liu, Julien Perez, and Scott Nowson. 2017. A Language-independent and Compositional Model for Personality Trait Recognition from Short Texts. In *Proc. 15th Conference ofthe European Chapter of the Association for Computational Linguistics (EACL'17): Volume 1*. 754–764.

[61] Xiaoqian Liu and Tingshao Zhu. 2016. Deep learning for constructing microblog behavior representation to identify social media user's personality. *PeerJ Computer Science* 2 (2016), e81.

[62] Veronica Lynn, Niranjan Balasubramanian, and H Andrew Schwartz. 2020. Hierarchical modeling for user personality prediction: The role of message-level attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5306–5316.

[63] François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. 2007. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of artificial intelligence research* 30 (2007),

457–500.

[64] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria. 2017. Deep Learning-Based Document Modeling for Personality Detection from Text. *IEEE Intelligent Systems* 32, 2 (Mar 2017), 74–79. https://doi.org/10.1109/MIS.2017.23

[65] Mika V Mäntylä, Fabio Calefato, and Maelick Claes. 2018. Natural language or not (NLON) a package for software engineering text analysis pipeline. In *Proceedings of the 15th International Conference on Mining Software Repositories*. 387–391.

[66] Davide Marengo and Christian Montag. 2020. Digital Phenotyping of Big Five Personality via Facebook Data Mining: A Meta-Analysis. *Digital Psychology* 1, 1 (jun 2020), 52–64. https://doi.org/10.24989/dp.v1i1.1823

[67] Sandra C Matz, Ruth E Appel, and Michal Kosinski. 2020. Privacy in the age of psychological targeting. *Current Opinion in Psychology* 31 (2020), 116–121. https://doi.org/10.1016/j.copsyc.2019.08.010 Privacy and Disclosure, Online and in Social Interactions.

[68] Wolfgang Mauerer, Mitchell Joblin, Damian Andrew Andrew Tamburri, Carlos Paradis, Rick Kazman, and Sven Apel. 2021. In Search of Socio-Technical Congruence: A Large-Scale Longitudinal Study. *IEEE Trans. Softw. Eng.* (2021). https://doi.org/10.1109/TSE.2021.3082074

[69] Robert R McCrae. 2001. Trait psychology and culture: Exploring intercultural comparisons. *Journal of personality* 69, 6 (2001), 819–846.

[70] Robert R McCrae. 2002. NEO-PI-R data from 36 cultures. In *The five-factor model of personality across cultures*. Springer, 105–125.

[71] Robert R McCrae and Paul T Costa. 1987. Validation of the five-factor model of personality across instruments and observers. *Journal of personality and social psychology* 52, 1 (1987), 81.

[72] Yash Mehta, Samin Fatehi, Amirmohammad Kazameini, Clemens Stachl, Erik Cambria, and Sauleh Eetemadi. 2020. Bottom-up and top-down: Predicting personality with psycholinguistic and language model features. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1184–1189.

[73] Yash Mehta, Navonil Majumder, Alexander Gelbukh, and Erik Cambria. 2019. Recent trends in deep learning based personality detection. *Artificial Intelligence Review* (2019), 1–27.

[74] E. Mellblom, I. Arason, L. Gren, and R. Torkar. 2019. The Connection Between Burnout and Personality Types in Software Developers. *IEEE Software* 36, 5 (2019), 57–64. https://doi.org/10.1109/MS.2019.2924769

[75] Fabiana Mendes, Emília Mendes, Norsaremah Salleh, and Markku Oivo. 2021. Insights on the relationship between decision-making style and personality in Software Engineering. *Information and Software Technology* (apr 2021), 106586. https://doi.org/10.1016/j.infsof.2021.106586

[76] Gregory J Meyer, Stephen E Finn, Lorraine D Eyde, Gary G Kay, Kevin L Moreland, Robert R Dies, Elena J Eisman, Tom W Kubiszyn, and Geoffrey M Reed. 2001. Psychological testing and psychological assessment: A review of evidence and issues. *American psychologist* 56, 2 (2001), 128.

[77] Michael K Mount, Murray R Barrick, and J Perkins Strauss. 1994. Validity of observer ratings of the big five personality factors. *Journal of Applied Psychology* 79, 2 (1994), 272.

[78] Isabel Briggs Myers, Mary H McCaulley, Naomi L Quenk, and Allen L Hammer. 1998. *MBTI manual: A guide to the development and use of the Myers-Briggs Type Indicator*. Vol. 3. Consulting Psychologists Press Palo Alto, CA.

[79] Zhu Nanli, Zou Ping, LI Weiguo, and Cheng Meng. 2012. Sentiment analysis: A literature review. In *2012 International Symposium on Management of Technology (ISMOT)*. IEEE, 572–576.

[80] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. 2015. The challenges of sentiment detection in the social programmer ecosystem. In *Proceedings of the 7th International Workshop on Social Software Engineering*. 33–40.

[81] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. 2018. A gold standard for emotion annotation in stack overflow. In *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*. IEEE, 14–17.

[82] Nicole Novielli, Fabio Calefato, Filippo Lanubile, and Alexander Serebrenik. 2021. Assessment of Off-the-Shelf SE-specific Sentiment Analysis Tools: An Extended Replication Study. *Empirical Software Engineering* 26, 2 (2021).

[83] Pavel Novikov, Larisa Mararitsa, and Victor Nozdrachev. 2021. Inferred vs traditional personality assessment: are we predicting the same thing? (2021). arXiv:2103.09632 http://arxiv.org/abs/2103.09632

[84] Jon Oberlander and Scott Nowson. 2006. Whose thumb is it anyway? Classifying author personality from weblog text. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. 627–634.

[85] Richard F Paige, Jordi Cabot, and Neil A Ernst. 2017. Foreword to the special section on negative results in software engineering. *Empirical Software Engineering* 22, 5 (2017), 2453–2456. https://doi.org/10.1007/s10664-017-9498-0

[86] O. H. Paruma Pabón, F. A. González, J. Aponte, J. E. Camargo, and F. Restrepo-Calle. 2016. Finding Relationships between Socio-Technical Aspects and Personality Traits by Mining Developer E-mails. In *2016 IEEE/ACM Cooperative and Human Aspects of Software Engineering (CHASE)*. 8–14. https://doi.org/10.1109/CHASE.2016.010

[87] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. *The development and psychometric properties of LIWC2015*. Technical Report.

[88] James W Pennebaker and Laura A King. 1999. Linguistic styles: language use as an individual difference. *Journal of personality and social psychology* 77, 6 (1999), 1296.

[89] Rosalind W Picard. 2000. *Affective computing*. MIT press.

[90] Rod Plotnik and Haig Kouyoumdjian. 2013. *Introduction to psychology*. Cengage Learning.

[91] Daniele Quercia, Michal Kosinski, David Stillwell, and Jon Crowcroft. 2011. Our twitter profiles, our selves: Predicting personality with twitter. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*. IEEE, 180–185. https://doi.org/10.1109/PASSAT/SocialCom.2011.26

[92] Naveen Raman, Minxuan Cao, Yulia Tsvetkov, Christian Kästner, and Bogdan Vasilescu. 2020. Stress and burnout in open source: Toward finding, understanding, and mitigating unhealthy interactions. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results*. 57–60.

[93] Ricelli Ramos, Georges Neto, Barbara Silva, Danielle Monteiro, Ivandré Paraboni, and Rafael Dias. 2018. Building a corpus for personality-dependent natural language understanding and generation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

[94] Francisco Rangel, Paolo Rosso, Martin Potthast, Benno Stein, and Walter Daelemans. 2015. Overview of the 3rd Author Profiling Task at PAN 2015. In *CLEF*. sn, 2015.

[95] Ayushi Rastogi and Nachiappan Nagappan. 2016. On the Personality Traits of GitHub Contributors. In *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 77–86. https://doi.org/10.1109/ISSRE.2016.43

[96] Louis M Rea and Richard A Parker. 2014. *Designing and conducting survey research: A comprehensive guide*. John Wiley & Sons.

[97] Peter C. Rigby and Ahmed E. Hassan. 2007. What Can OSS Mailing Lists Tell Us? A Preliminary Psychometric Text Analysis of the Apache Developer Mailing List. In *Fourth International Workshop on Mining Software Repositories (MSR'07:ICSE Workshops 2007)*. IEEE, 23–23. https://doi.org/10.1109/MSR.2007.35

[98] Brent W Roberts, Nathan R Kuncel, Rebecca Shiner, Avshalom Caspi, and Lewis R Goldberg. 2007. The power of personality: The comparative validity of personality traits, socioeconomic status, and cognitive ability for predicting important life outcomes. *Perspectives on Psychological science* 2, 4 (2007), 313–345.

[99] Shade Ruangwan, Patanamon Thongtanunam, Akinori Ihara, and Kenichi Matsumoto. 2019. The impact of human factors on the participation decision of reviewers in modern code review. *Empir. Softw. Eng.* 24, 2 (apr 2019), 973–1016. https://doi.org/10.1007/s10664-018-9646-1

[100] Richard M Ryckman. 2012. *Theories of personality*. Cengage Learning.

[101] Norsaremah Salleh, Emilia Mendes, John Grundy, and Giles St J Burch. 2010. An empirical study of the effects of conscientiousness in pair programming using the five-factor personality model. In *2010 ACM/IEEE 32nd International Conference on Software Engineering*, Vol. 1. IEEE, 577–586.

[102] Norsaremah Salleh, Emilia Mendes, John Grundy, and Giles St. J. Burch. 2010. The effects of neuroticism on pair programming. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '10*. ACM Press, 1–10. https://doi.org/10.1145/1852786.1852816

[103] Joni Salminen, Rohan Gurunandan Rao, Soon-gyo Jung, Shammur A Chowdhury, and Bernard J Jansen. 2020. Enriching Social Media Personas with Personality Traits: A Deep Learning Approach Using the Big Five Classes. In *International Conference on Human-Computer Interaction*. Springer, 101–120.

[104] Anand Ashok Sawant and Premkumar Devanbu. 2021. Naturally!: How Breakthroughs in Natural Language Processing Can Dramatically Help Developers. *IEEE Software* 38, 5 (2021), 118–123. https://doi.org/10.1109/MS.2021.3086338

[105] David P. Schmitt, Jüri Allik, Robert R. McCrae, Verónica Benet-Martínez, and et al. 2007. The geographic distribution of Big Five personality traits: Patterns and profiles of human self-description across 56 nations. *Journal of Cross-Cultural Psychology* 38, 2 (mar 2007), 173–212. https://doi.org/10.1177/0022022106297299

[106] Patrick Schober, Christa Boer, and Lothar A Schwarte. 2018. Correlation coefficients: appropriate use and interpretation. *Anesthesia & Analgesia* 126, 5 (2018), 1763–1768.

[107] H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one* 8, 9 (2013), e73791.

[108] Ben Shneiderman. 1980. *Software psychology: Human factors in computer and information systems (Winthrop computer systems series)*. Winthrop Publishers.

[109] Forrest J Shull, Jeffrey C Carver, Sira Vegas, and Natalia Juristo. 2008. The role of replications in empirical software engineering. *Empirical software engineering* 13, 2 (2008), 211–218.

[110] E. Smith, R. Loftin, E. Murphy-Hill, C. Bird, and T. Zimmermann. 2013. Improving developer participation rates in surveys. In *2013 6th Int'l Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. 89–92. https://doi.org/10.1109/CHASE.2013.6614738

[111] Edward K Smith, Christian Bird, and Thomas Zimmermann. 2016. Beliefs, practices, and personalities of software engineers: a survey in a large software company. In *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering*. 15–18.

[112] Igor Steinmacher, Igor Scaliante Wiese, and Marco Aurelio Gerosa. 2012. Recommending mentors to software project newcomers. In *2012 Third Int. Work. Recomm. Syst. Softw. Eng.* IEEE, 63–67. https://doi.org/10.1109/RSSE.2012.6233413

[113] David J Stillwell and Michal Kosinski. 2004. myPersonality project: Example of successful utilization of online social networks for large-scale social research. *American Psychologist* 59, 2 (2004), 93–104.

[114] Chris Sumner, Alison Byers, Rachel Boochever, and Gregory J Park. 2012. Predicting dark triad personality traits from twitter usage and a linguistic analysis of tweets. In *2012 11th international conference on machine learning and applications*, Vol. 2. IEEE, 386–393.

[115] Yla R Tausczik and James W Pennebaker. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of language and social psychology* 29, 1 (2010), 24–54.

[116] Louis Tay, Sang Eun Woo, Louis Hickman, and Rachel M. Saef. 2020. Psychometric and Validity Issues in Machine Learning Approaches to Personality Assessment: A Focus on Social Media Text Mining. *European Journal of Personality* 34, 5 (sep 2020), 826–844. https://doi.org/10.1002/per.2290

[117] Walter F Tichy. 2000. Hints for reviewing empirical work in software engineering. *Empirical Software Engineering* 5, 4 (2000), 309–312.

[118] Marc Tomlinson, David Hinote, and David Bracewell. 2013. Predicting conscientiousness through semantic analysis of facebook posts. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 7.

[119] Gias Uddin and Foutse Khomh. 2019. Automatic mining of opinions expressed about apis in stack overflow. *IEEE Transactions on Software Engineering* (2019).

[120] Alessandro Vinciarelli and Gelareh Mohammadi. 2014. A survey of personality computing. *IEEE Transactions on Affective Computing* 5, 3 (2014), 273–291. https://doi.org/10.1109/TAFFC.2014.2330816

[121] Sai Datta Vishnubhotla, Emilia Mendes, and Lars Lundberg. 2020. Investigating the relationship between personalities and agile team climate of software professionals in a telecom company. *Information and Software Technology* 126 (oct 2020), 106335. https://doi.org/10.1016/j.infsof.2020.106335

[122] Gerald M Weinberg. 1971. The psychology of computer programming. *New York* (1971).

[123] Aidan GC Wright. 2014. Current directions in personality science and the potential for advances through computing. *IEEE Transactions on Affective Computing* 5, 3 (2014), 292–296.

[124] Tal Yarkoni. 2010. Personality in 100,000 words: A large-scale analysis of personality and word use among bloggers. *Journal of research in personality* 44, 3 (2010), 363–373.

[125] Murat Yilmaz, Rory V. O'Connor, Ricardo Colomo-Palacios, and Paul Clarke. 2017. An examination of personality traits and how they impact on software development teams. *Information and Software Technology* 86 (2017), 101–122. https://doi.org/10.1016/j.infsof.2017.01.005

A APPENDIX: REPLICATION PACKAGE

Data Collection and Protection

The complete replication package is available on Zenodo at https://zenodo.org/record/4679303.

We are aware of the sensitive nature of the data collected and the privacy risks that come from misusing them. As such, here we clarify all the measures taken to ensure data privacy and protection during our research work. Albeit the GDPR took place in May 2018 (i.e., after the data collection conducted in January 2018), we nonetheless made efforts to comply with the directives already approved by the EU in May 2016.

To ensure that we had control over the data and that they were stored in servers located in the EU, rather than administering the survey through platforms such as Google Forms, we opted for developing in-house an electronic version of the Mini-IPIP personality survey. As such, the application was hosted on our University cloud infrastructure in Italy. The application also handled the sending of invitations to participants via email and the automatic removal of such messages after one month. Both the invitation emails and the landing page of the application identified the research team, clarified the research purpose, and contained a link to the privacy statements briefly summarized next.

In particular, we clarified that: we had retrieved their email address from the public archives of the Apache Software Foundation; there was not going to be any other follow-up email; there was no monetary compensation and study participation was voluntary; the survey responses were needed for research purposes and were going to be stored anonymously; the data and analysis results would be only shared in scientific venues, such as conferences and journals, and presented in an aggregate form, thus making it impossible to identify who participated. We also clarified that the link to the survey in the invitation email contained a randomly generated id that would match their survey responses to a corpus consisting of some of their emails (i.e., the gold standard) publicly archived by the Apache Software Foundation. However, we highlight that (i) at the end of the data collection, we erased all the emails from those who did not answer the survey; (ii) regarding the respondents, by submitting the survey, the system replaced their plain-text email address with the hashed survey id, thus preventing us and anyone else to match an author in the email corpus with their survey responses.

Furthermore, because the Apache Software Foundation email archives are publicly available, to further ensure anonymity and prevent third parties to guess the identity of the survey respondents from the content of their emails, we point out that the gold standard shared in the replication package is built after scrubbing from the email bodies any potentially-sensitive information, such as email and mail addresses, URLs, names, pieces of code, numbers, and stop words (we used three Python libraries, `clean-text`, `scrubadub`, and `NLTK`).

In conclusion, we are confident that all the measures taken are effective in protecting the privacy and anonymity of the developers who agreed to participate in the study.

Download and Setup

First, clone the repository and its submodules from GitHub:

```
git clone --recursive https://github.com/collab-uniba/tosem2021-personality-rep-package.git
```

Then, before the first execution, run the `setup.sh` script. Also, make sure that the requirements are satisfied, in particular Python 3.8.3+, R 4.0.4+, and Java 1.8+.

```
bash setup.sh
```

Execution Instruction

It is possible to automatically execute the full experimental workflow by launching the `repro.sh` script as follows:

```
bash repro.sh --stage all --dataset full
```

For test purposes, instead of supplying the argument `full`, it is possible to use the argument `test` to work with a small, random subsample of the experimental dataset to shorten the execution time (see the next subsection for more):

```
bash repro.sh --stage all --dataset test
```

It is also possible to execute the two workflow stages independently, using either the full dataset or the test one:

```
bash repro.sh --stage phase1 --dataset full
```

```
bash repro.sh --stage phase2 --dataset test
```

Execution Times

The execution of the entire pipeline is quite time-consuming as, depending on the machine specifications, it takes hours—if not days—when working on the full dataset. Additional time is also necessary in case one wants to retrain the TwitPersonality models instead of using the pre-trained ones.

Table 22 compares the execution times between two machines with different hardware specifications, running the same OS (Ubuntu 20.04). We observe that the most recent machine's performance when re-executing the full pipeline on the smaller test dataset is somewhat close to the older machine's (5m17s and 7m35s, respectively). The newer machine is also faster when it comes to model retraining (6m27s vs. 10m18s). The largest difference, however, is observed when using the full dataset, which increases the execution time to ~2 days for the more recent configuration and ~3.9 days for the older machine.

Table 22. A comparison of execution times on two machines with different hardware specifications running the same OS (Ubuntu LTS 20.4.2).

Year	Hardware specifications	Pipeline test mode	Pipeline full mode	Model retraining
2018	CPU Intel i7-7700 (Kaby Lake), 8 cores @ 3.60GHz, 16GB RAM	5m17s	~2d	6m27s
2011	CPU Intel Xeon E312xx (Sandy Bridge), 8 cores @ 2.00GHz, 16GB RAM	7m35s	~3.9d	10m18s

B APPENDIX: PHASE 1 – ADDITIONAL MATERIAL

Table 23. Descriptive statistics per trait for the distributions of scores in the gold standard and inferred by the tools.

Gold standard	**Min**	**Max**	**Mean**	**Median**	**SD**
Openness	2.75	5.00	4.31	4.50	0.63
Conscientiousness	1.75	5.00	3.71	3.75	0.76
Extraversion	1.00	5.00	2.76	2.75	0.83
Agreeableness	1.25	5.00	3.73	4.00	0.81
Neuroticism	1.00	5.00	2.77	2.75	0.93
LIWC	**Min**	**Max**	**Mean**	**Median**	**SD**
Openness	0.97	2.35	1.67	1.69	0.26
Conscientiousness	1.27	2.83	2.52	2.51	0.24
Extraversion	2.12	3.46	3.09	3.09	0.17
Agreeableness	3.24	4.30	3.60	3.58	0.17
Neuroticism	3.34	5.00	3.81	3.80	0.24
IBM PI	**Min**	**Max**	**Mean**	**Median**	**SD**
Openness	4.04	4.55	4.23	4.24	0.09
Conscientiousness	3.10	3.72	3.47	3.48	0.13
Extraversion	2.70	3.30	2.98	2.96	0.11
Agreeableness	3.42	3.92	3.61	3.59	0.11
Neuroticism	2.52	3.24	3.03	3.05	0.13
Personality Recognizer	**Min**	**Max**	**Mean**	**Median**	**SD**
Openness	2.80	4.07	3.45	3.43	0.31
Conscientiousness	3.00	4.07	3.51	3.52	0.23
Extraversion	2.33	4.17	3.52	3.53	0.29
Agreeableness	2.93	4.02	3.43	3.44	0.20
Neuroticism	2.47	3.68	2.98	2.92	0.33
TwitPersonality	**Min**	**Max**	**Mean**	**Median**	**SD**
Openness	4.13	4.14	4.13	4.13	0.01
Conscientiousness	3.51	3.52	3.51	3.51	0.01
Extraversion	3.34	3.36	3.35	3.35	0.01
Agreeableness	3.62	3.63	3.63	3.63	0.01
Neuroticism	2.60	2.60	2.60	2.60	0.00

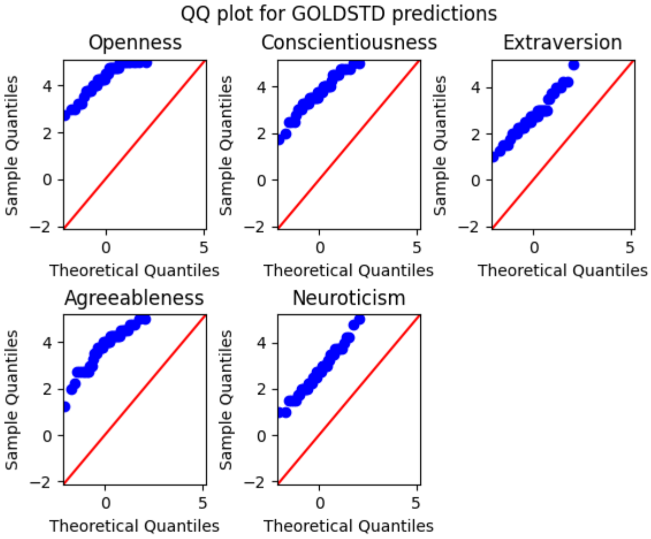


Fig. 5. Q-Q plot of the trait distributions from the gold standard.

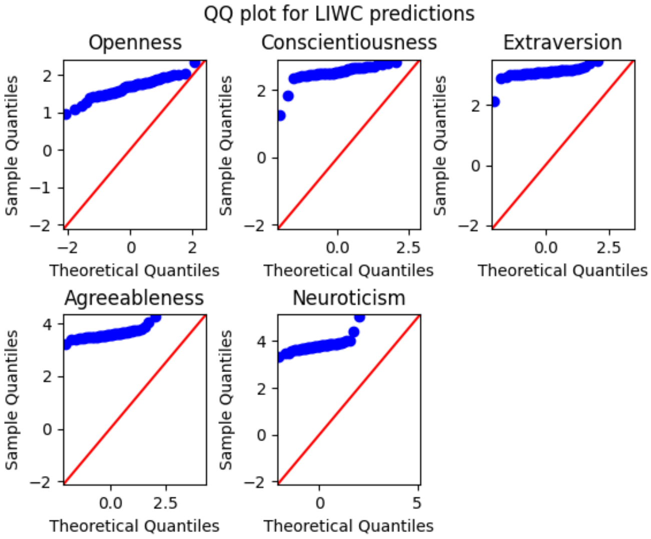


Fig. 6. Q-Q plot of the trait distributions from LIWC.

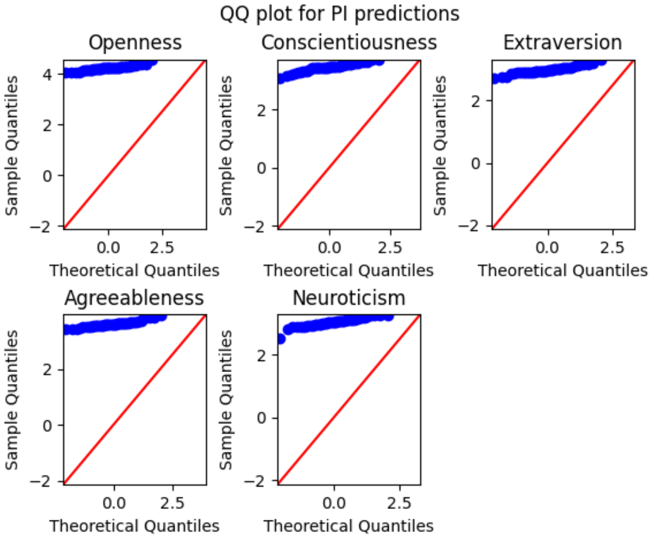


Fig. 7. Q-Q plot of the trait distributions from IBM PI.

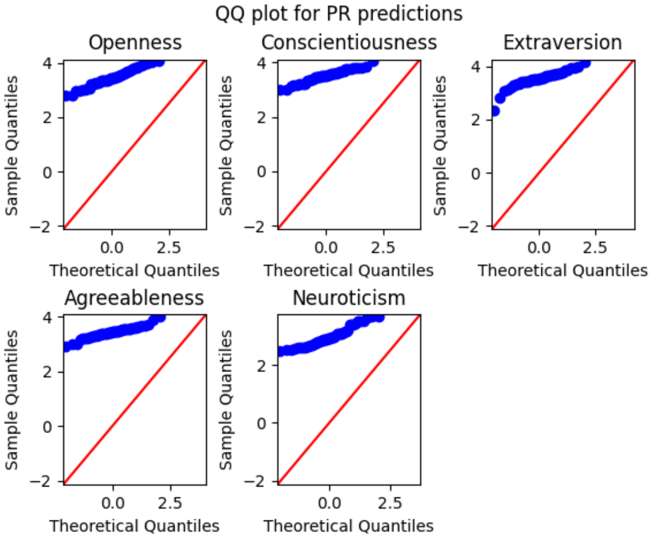


Fig. 8. Q-Q plot of the trait distributions from Personality Recognizer.

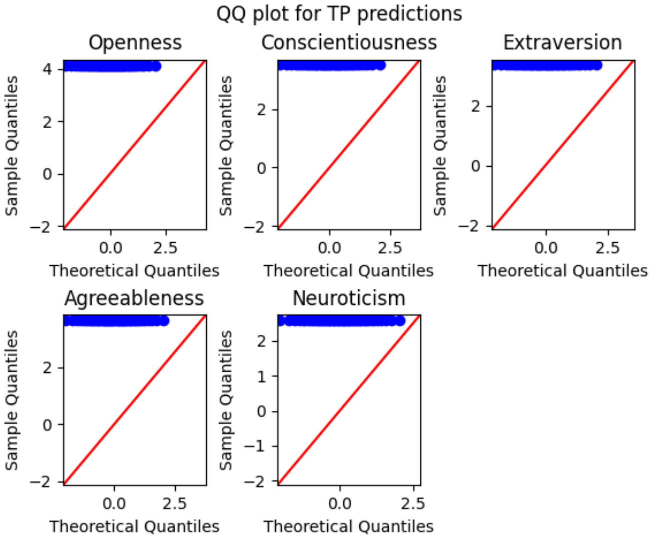
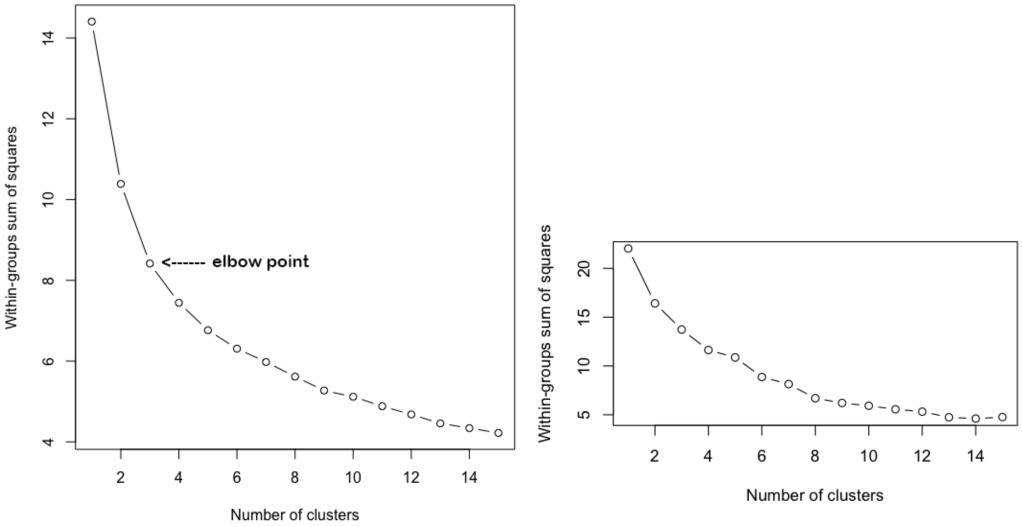


Fig. 9. Q-Q plot of the trait distributions from TwitPersonality.

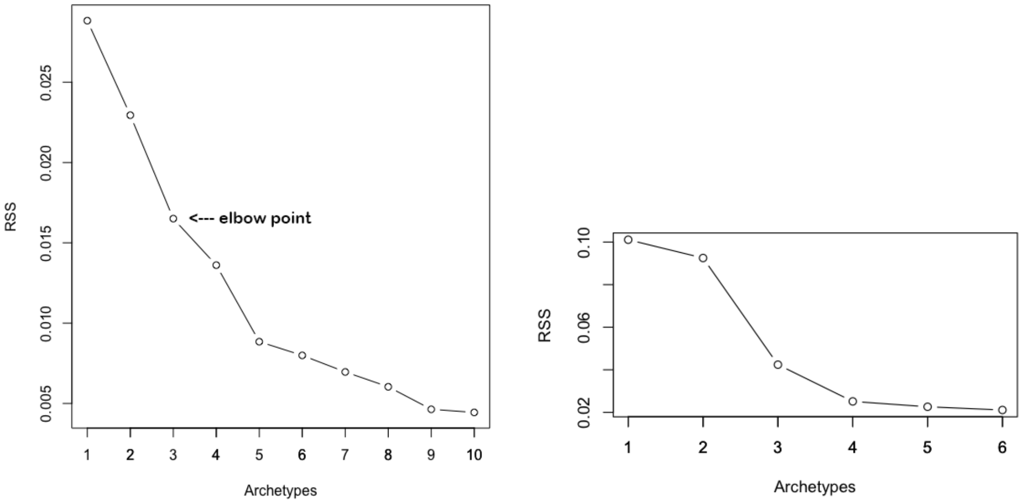
C APPENDIX: PHASE 2 – ADDITIONAL MATERIAL



(a) Original study

(b) Replication

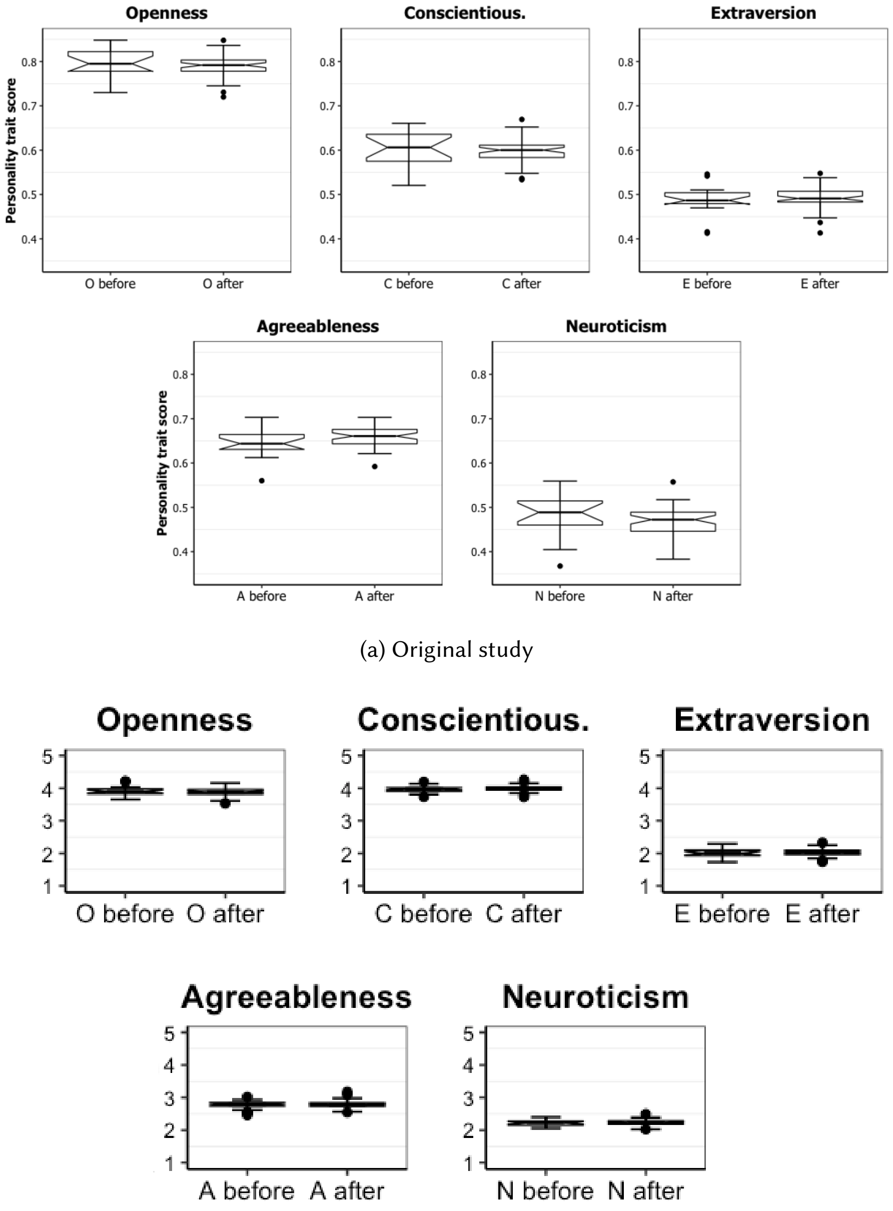
Fig. 10. Plot of within-group heterogeneity against the number of *k*-means clusters in the original study by Calefato et al. [17] (a) and the replication (b).



(a) Original study

(b) Replication

Fig. 11. Scree plot of the residual sum of squares against the number of archetypes in the original study by Calefato et al. [17] (a) and the replication (b).



(a) Original study

(b) Replication

Fig. 12. Differences in the personality traits of the developers before and after becoming core team members in the original study by Calefato et al. [17] (a) and the replication (b).