Outliers and Replication in Software Engineering

Henrik Larsson, Erik Lindqvist and Richard Torkar
Chalmers and the University of Gothenburg, Gothenburg, Sweden

***Abstract*—Empirical software engineering is a research field of growing interest. Studies within this field handles an increasing amount of data. In order to replicate a study the data needs to be accessible and all processing of this data needs to be reproducible. Specifically, the handling of deviating data points, also known as outliers, needs to be documented in order for a study to be replicated. This study investigated the data availability for recently published studies within empirical software engineering. Furthermore, it also investigated if outliers are documented in the same research field. Papers were reviewed using a literature review and the presence of outliers was investigated using an unsupervised outlier detection method. Only 37% of the papers reviewed had their data accessible. Furthermore, in many cases outliers were present in the reviewed studies but 63% of the papers studies did not mention how outliers were handled. The data availability within empirical software engineering research is low and is hindering replication of studies. Additionally, the lack of documentation regarding how outliers are handled is hindering replication.**

I. Introduction

Software is of importantance for both national and international infrastructure. Hence, it is increasingly important to produce software more cost-efficiently [1]. This has led to an increased interest, the last 30 years, in software engineering (SE). The field of SE does not just cover the technical aspects of creating software, it also attends to the aspects of managing software projects. Empirical studies plays an important role in order to study the effects of developed methods and tools within SE.

Data collected from empirical studies are used to draw conclusions. However, this data can contain outliers, values that deviates significantly from the rest, which may or may not impact the analysis of the study [2]. It is therefore important to understand the impact of outliers in order to interpret the results correctly and to draw valid conclusions [3]. Additionally, the task of identifying and removing outliers needs to be documented for the study to be more easily replicated. This is of importance since replication studies, which confirms earlier findings, helps build confidence in previously presented results. Consequently, replication is considered as one of the cornerstones in science and an indicator of how mature a scientific discipline is [4].

In this study the availability of data in research papers from empirical software engineering (ESE) will be investigated. Moreover, the presence of undocumented outliers, within the field of ESE, will be investigated using an unsupervised detection method. Ultimately, this study aims at providing guidelines regarding if/how outlier detection should be conducted and presented in empirical studies.

In order to be able to replicate an SE study the data has to be available. Therefore, our first research question is:

> **RQ1:** To what extent is data available in research papers from software engineering and in which form is the data made available?

The documentation of outliers in SE studies are of importance for the study's ability to be replicated. Therefore, our second research question aims at investigating to what extent the presence of outliers is documented:

> **RQ2:** Are undocumented outliers present and documented in software engineering studies?

In order to study the effects of removing outliers on the conclusions drawn in a paper the following research question was, in addition, investigated:

> **RQ3:** Does removing outliers change the conclusions of recently published studies?

II. Theoretical Background and Related Work

In this section two areas of importance for this study, outliers and replication will be presented. In Section II-A the concept of outliers will be discussed and issues regarding outliers as well as related work will be presented. Section II-B aims at presenting replication as a concept and its importance as well as some related work regarding replication within SE.

A. Outliers

Outliers are data points that differ significantly from other data points in a data set. A commonly seen definition for outliers is "an outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs" [2]. It is also stated by [2] that the effect of outliers can impair the statistical analysis. Therefore, determining if the collected data contains any data values that should be regarded as outliers is of importance.

There are several approaches for classifying outlier detection methods, three of them are mentioned by [5] as unsupervised, supervised and semi-supervised detection techniques. Unsupervised detection determines if a data point is an outlier with no prior knowledge of the data set and flags the data points most separated from the normal data as outliers. Supervised detection use data that is pre-labeled as normal or not normal in order to determine if the other data points are outliers or not. Finally, semi-supervised detection uses a small set of training data to detect outliers [5].

Seo and Bae have studied the effect of outliers in software effort estimation. This was done by investigating the effect outliers had on the estimation accuracy of commonly used software estimation methods [6]. The methods were evaluated on industrial data collected from publicly available repositories such as PROMISE and ISBSG. A Wilcoxon ranked sum test was used to see if removing outliers made a significant difference. The authors reported that there was a positive effect in estimation accuracy when removing outliers but not enough to say that it is *significantly* better. The study by [6] differs from our study by being focused on effort estimation and using publicly available data sets only. Our study relies on data retrieved from recently published papers within SE and takes a more automated approach to outlier detection. The focus in our study is on describing the state of practice when it comes to data availability and how research data is made available (**RQ1**).

In another study, Yuan and Bentler evaluated how outliers distorts the results in covariance structure analysis and the quantitative effect of outliers on statistical tests [7]. Covariance tests are used to determine how different variables affect each other. Since covariance tests are used within SE, see e.g. [8], it may be of additional importance for the SE community to understand what impact outliers have on the end result. The study by Yuan and Bentler reports that effects of a few outliers can discredit the value of using a model. They also report that outliers do not need to be very extreme to break down the covariance analysis.

B. Replication

As empirical studies have become more common within SE the importance of being able to replicate studies increases. Such replications are important since they help increase the body of knowledge around SE, which in turn leads to an increased maturity of the field. A replication of a study also comes with benefits for the original study in terms of increased confidence for the conducted experiment and the reported findings (e.g. tightened confidence intervals). Furthermore, the success of a conducted replication is not mainly depending on how well the replicated results conform to the original but on the contribution to the body of knowledge [4], [9], [10].

There are in general two forms of replication, internal and external. Internal replication is carried out by the original researcher or team while external replication is carried out by someone else than the original author [4]. Furthermore, the degree to which a replication is carried out can be divided into exact and conceptual replication. An exact replication is a replication which follows the original procedure as closely as possible, whereas a conceptual replication is a replication where the same hypothesis is validated through a different procedure [10]. However, Juristo and Vegas [11] state that exact replication within SE is close to non-existent due to the difficulties in recreating the exact conditions from the original experiment. Furthermore, Juristo and Vegas also propose that promoting non-exact experiments could encourage more researchers to perform replication experiments.

Sjøberg et al. [12] found in their survey that only 18% of the surveyed papers from SE were replications. This was a surprising finding considering that replication is seen as important in science [13]. On the other hand, the reason for this lack of replication studies might not be surprising since Lindsey and Ehrenberg [13] suggest that it might be due to that replicated experiments do not reward the researcher as much as an original experiment.

The previously mentioned papers regarding replication within SE are focusing mostly around replicating the experiment and do not regard data analysis to a great extent (a somewhat greater emphasize is devoted to this subject in this paper). Though, within the field of bioinformatics, where large data sets are common, more work has been carried out in order to promote a more reproducible way of presenting the analysis for a study [14]–[16]. The main purpose of these proposals is to make the analysis clear to the reader with the idea being that reproducing a study's report is merely executing an accompanying script in the report's repository. This allows the reader to easily reproduce artifacts such as plots and tables, and the reader can even perform changes to the analysis and study the result of them *in vivo*. If it is assumed that the amount of data used by empirical studies within SE is increasing, the need for standardized tools for handling data, such as those promoted for bioinformatics, increases as well.

Although it might sound simple in theory to ensure that a study is reproducible, the previously mentioned low outcome of replications could be an indicator that it is harder in practice.

In order to make this study reproducible all results are automatically generated and can be recreated at any time by downloading the study's resources[1] and executing the analysis script as described in the accompanying documentation (R and a number of packages are needed)

III. Study Execution

The study execution involved data collection and the application of an outlier detection algorithm using our developed 'pipeline'.

A. Data Collection

In the data collection phase the goal was to find suitable papers whose data could be used to run outlier detection on.

The process of searching and reviewing papers was carried out in a systematic way albeit not strictly following the guidelines for conducting systematic literature reviews in SE. Selection and quality assessment criteria were defined and used to determine which papers to include for the study. The process was meant to provide a sample of published papers to give an idea of the current state of the research field. Therefore, an exhaustive review over all papers published within the field was not carried out. Due to this limitation, this review cannot be considered a systematic literature review as defined in [17]. The methodology used for the review is further elaborated in the following paragraphs.

Papers used in this study were selected from the research field of SE. To refine the scope only papers from recognizable sources within SE were considered. More specifically the sources were the *Empirical Software Engineering Journal (ESEJ)*, *International Conference on Software Engineering (ICSE)* and *International Conference on Predictive Models in Software Engineering (PROMISE)*. ESEJ was chosen since it is a journal featuring articles on empirical research within SE. Proceedings from ICSE was chosen since the conference is considered to be among the leading conferences within

[1] https://github.com/linqcan/odser2014

SE. The PROMISE conference proceedings include empirical research and associated to this is an online repository[2] containing research data from papers. Therefore, PROMISE was deemed as suitable source for elicitation. Furthermore, only papers from 2013–2014 were considered, in order to get a current sample of the field, i.e. an exhaustive search was not conducted. As a final filter only papers regarding empirical studies and numerical data were chosen. The papers that passed the mentioned criteria were then considered for this study.

To answer **RQ1** we investigated how data used in the reviewed papers are made available to the public. This was carried out by trying to access the raw data from the papers selected previously. In the first step, a check was made to see if the data was available in the paper. If no data was found in the paper a search was made for references to webpages or online repositories in the paper to see if the data was made available online. The third, and last, step was contacting the corresponding author via email and asking for the data. The email sent was a short email acknowledging the author's paper and asking for access to the study's data without elaborating the intentions further. However, if an author asked about the intentions a description of this study was given. If a reply did not come within four weeks the data was regarded as 'not available'.

Papers that had data available, a described pre-process and analysis and were suitable to outlier removal were analyzed in order to answer **RQ3**.

B. *Outlier Detection Algorithm*

As the algorithm used to detect outliers we chose Modified Z Score (MZS). The MZS algorithm measures how much a particular data point differs from the rest of the data set, using a score calculated by Equations 1–2. MZS is applicable for one dimensional data and calculates the score, M_i, from the Median Absolute Deviation (MAD). The MAD is calculated by taking the median of the absolute value of the difference between each point and the median for the data set [18]. Hence, the algorithm is more robust than algorithms using the mean to score outliers [19]. In Equation 2 the M_i score for each data point is calculated by first taking the absolute value of the difference between the specific point and the median for the data set. As a second step the difference calculated in step one is multiplied with 0.6745 to make the calculation more robust [18]. As a third and last step the value from step two is divided with the MAD to get the M_i score. In the exceptional case where MAD equals zero the same alternative algorithm as used by [20] is implemented. Furthermore, to label outliers both the original and the alternative algorithm use the M_i score and compares it to the average M_i score of the data set. In [21] the authors suggests that if M_i is $> 3.5\sigma$ it should be considered as an outlier. Furthermore, the authors propose that using this cut-off value will make the method more robust. Since MZS is considered to be a robust algorithm, does not take any input parameters and does not require training data it was deemed as a suitable candidate for this study, i.e. where an unsupervised and automated process was a requirement.

$$\text{MAD} = \text{median}_i(|X_i - \text{median}_j(X_j)|) \tag{1}$$

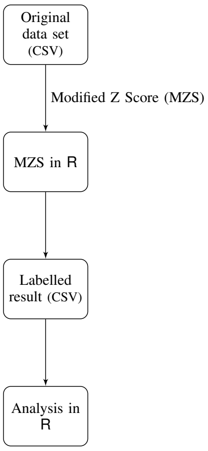

Fig. 1. Flowchart showing the steps involved in the pipeline. Analysis in R involves creating all the artifacts mentioned in Section III-C.

$$M_i = \frac{0.6745(|X_i - \text{median}_j(X_j)|)}{\text{MAD}} \tag{2}$$

C. *Pipeline*

In order to detect outliers unsupervised, in a large number of data sets, an automated process was created, referred to as the 'pipeline'. This pipeline takes a data set as input and applies the outlier detection algorithm on the set. A flowchart of the pipeline can be viewed in Fig. 1. The purpose of the pipeline was not only to facilitate automated detection but also to increase the reproducibility of this study. By downloading the published code a replicator can produce the same artifact for analysis as we used during our research.

The pipeline is a `Python` script utilizing a set of scripts, written in `Python` and `R`, that performs different tasks. The outlier detection algorithm, MZS, was implemented in `R` and produces a data file with the score for each value as well as a boolean indicating if it is an outlier depending on the set boundary. To facilitate testing of different boundaries, all configuration options are set in a configuration file separate from the pipeline. This lets the replicator test different boundary values in order to view the effect of them. Based on the information in the result file, a set of `R` scripts creates the following artifacts that help the replicator analyze the data set under investigation:

- The original data set with the identified outliers labeled
- Descriptive statistics for original and modified data sets:
    - Mean
    - Median
    - Standard deviation
    - Number of outliers
- Density and QQ-plots[3] for the original data set and the modified data set

[2] https://code.google.com/p/promisedata/

[3] The studied data set's distribution is compared with that of a normal distribution.

TABLE I. Descriptive statistics for the data collection.

Papers gathered	43
Papers missing contact information	2
Data requests made	30
Replies stating data is confidential	3
Replies stating ‘other reason’ to data not available	2
Replies with data	5
Papers with data available (online, paper, from author)	16
Papers with data online	4
Papers with data in the paper	7
Papers with identified outliers	8
Papers documenting outliers	16

TABLE II. Data availability, listed by source.

Available	ESEJ	ICSE	PROMISE
Yes	23%	33%	0%
No	65%	67%	40%
From author	11%	0%	60%

- Plot with outliers marked and the change of mean if they are removed
- Output of a Shapiro-Wilks normality test for both the original and modified data set
    - Output of a Welch's t-test. Significance testing for difference between the original and modified data set
    - Output of a Mann-Whitney U test. Significance testing for difference between the original and modified data set

In order to facilitate the viewing of results, the results can be browsed using the web browser. From the information available one could, for example, study the significant effect (on the original data set) of removing outliers. The normality tests help the replicator decide which significance test is applicable for the data set under investigation. A more detailed description of how the pipeline works can be found in the `README` file accompanying the repository. Our pipeline is open source and we encourage the community to try it out and provide feedback.

IV. Results

In Table I descriptive statistics for the paper search is presented. In the beginning, our paper search consisted of 187 papers (ESEJ 45, ICSE 130, PROMISE 12). 43 out of those papers matched the criteria stated in the data collection description. Out of these 43, 16 (37%) had data available after a request was sent to the authors. As additional information regarding data availability, the source with the highest rate of papers with data available was PROMISE (60%) followed by ESEJ (34%) and ICSE (33%). Additionally a more detailed description if data was made available or if it needed to be requested from the authors can be viewed in Table II. Furthermore, it is stated in the table that two replies were given as ‘other reason’, these reasons were that the data's size was too large for it to be handed over and that the data was not easily available to the author. Finally, Table III shows how the selected papers were divided by the three different sources.

TABLE III. Selected papers from source.

Papers gathered	43
From ESEJ	17
From ICSE	21
From PROMISE	5

TABLE IV. To what extent outliers are documented, listed by source.

	ESEJ	ICSE	PROMISE
Documents outliers	53%	14%	80%

Regarding **RQ2**, PROMISE (80%) had the largest percentage of papers documenting outliers and was followed by ESEJ (53%) and ICSE(14%) as listed in Table IV. After the application of the outlier algorithm, MZS, 24 out of 77 data sets (31%) were reported to have outliers.

In our data collection we collected 43 papers and 13 of those were investigated in more depth [22]–[34]. Finally, two papers were deemed fit for analyzing **RQ3**, they were analyzed as follows:

a) Do background colors improve program comprehension in the #ifdef hell?: In this analysis the focus is on on research hypotheses RH1, RH2 and RH4 from [29] since they all regard non-binary data and their data was made accessible by the authors.

For RH1 and RH2 post-processing was carried out on all the data sets regarding the maintenance tasks (Mx) as the authors had omitted response times for questions that were answered incorrectly. This omission was mentioned in the original paper and clarified by the authors via email correspondence.

The authors answer RH1 (“In static tasks, colors speed up program comprehension compared to ifdef directives”) and RH2 (“In maintenance tasks, there are no differences in response time between colors and ifdef directives”) by conducting a significance test and an effect size test for the static and maintenance tasks to compare the test using colors with the test using ifdef. To reproduce these tests a Mann-Whitney U test was used for non-parametric and Welch's t-test for parametric significance testing. Only the data sets with possible outliers, S1-ifdef, M1-ifdef and M2-ifdef was considered for the reproduction. The new significance tests carried out in this study, Mann-Whitney U for S1 and M2 and Welch's t-test for M1, indicated no difference in the conclusions compared to those carried out by the original authors. The effect size test, calculated using Cliff's δ, for S1 was slightly altered (-0.6417 compared to -0.61) but it resulted in no SSD.

RH4 was validated using significance tests in the original study. The data sets used in RH4 consisted of results from a survey using a 1–5 Lickert scale. After outlier detection was conducted on the data sets used for this research question, three data sets were reported to have potential outliers: M1-ifdef performance, M2-ifdef performance and M3-ifdef performance. Since these were the only modified data sets, significance tests were only reproduced for these three data sets. The reproduction of the statistical analysis, using a Mann-Whitney U test, showed no other results than those reported in the original study.

b) An empirical study on the developers' perception of software coupling: The results from the experiment were reported using a 1–5 Lickert scale and were not processed before being analyzed. In the original analysis the different coupling techniques' p-values are compared with each other. The p-values are calculated with a Mann-Whitney U test, this test is then used to determine if there is a perceived difference between the different coupling techniques. Out of the eight investigated data sets from `jEdit` only three contained outliers: Semantic-low, structural-low and logical-low. After removing all outliers from the three data sets the same Mann-Whitney U tests, as used in the original study, was executed. This led to six tests and in one of those tests the p-values changed noticeable, in the comparison between structural low and logical low. However, in all honesty, this change does not affect the overall conclusion of the original study.

V. Discussion

Based on the data gathered we answer our research questions as follows.

RQ1: 37% of the papers (16) collected had data available. In total, 16% of the papers in the sample offered data in the paper and 9% online. The result presented here is far from surprising for an, in our opinion, immature research field such as SE. There could be many reasons for the low outcome and we propose two reasons which we believe are more important:

First, there is a lack of consensus on how to treat and make raw data available within SE. This is, according to us, a major issue and something that the field of SE needs to address. In our guidelines we elaborate more on this.

The second reason, proposed by us, for the low outcome is that replication is not kept in mind by researchers while conducting their research. This might have to do with replication not being common within SE as it is a fairly immature research field (compared to physics and medicine). Also, we have observed that the replication mentioned in SE research literature concerns full experiment replication and does rarely mention replication of the data analysis, so called re-analysis. However, we believe that re-analysis is of importance as well since it can be used to validate if conclusions based on, for example, significance tests are correct. As an example of this, one of the papers included in this study was found by us to have incorrect calculations in the analysis. This error was later confirmed and corrected by the author who stated that it fortunately had no impact on the conclusions in the paper. In addition, being able to conduct re-analysis would further encourage meta-analysis in SE research, since access to data and statistical analysis procedures would be readily available.

RQ2: Out of the 16 papers having data available, 13 were used in this study. Our outlier detection algorithm identified outliers in data sets from 8 out of those 13 papers. In total, 24 of the 77 data sets that were analyzed contained outliers. We chose a robust method for detecting outliers since we wanted a method that fitted a wide range of data sets in order to implement an automatic process. However, this probably led to less outliers being found in comparison to if we would have chosen a suitable outlier detection method for each data set. This fact should be taken into account when interpreting our results and is mentioned in our threats to validity.

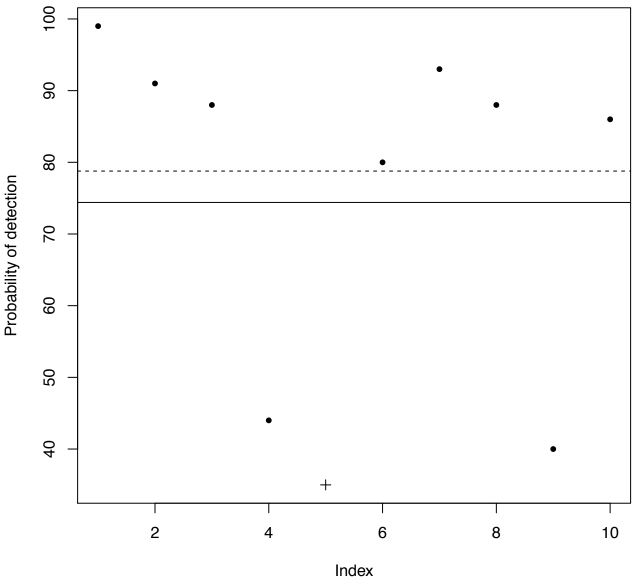

Fig. 2. Plot with with an obscure outlier. Data set from [24].

In this study, 27 of the 43 papers gathered do not document outliers at all. Furthermore, 3 out of the 8 papers identified to have data with outliers do not document the presence of them. This is troublesome as it hinders replication of SE studies in the long term as they do not mention how they handle outliers at all.

Figure 2 and Fig.3 provide examples of the output our pipeline creates. Furthermore, they also show how outliers can be more or less difficult to determine using visual inspection. Hence, it is imperative that researchers use these automated tools of analysis as input when analyzing possible outliers.

RQ3: None of the two papers further analyzed [22], [29] showed that removing the identified outliers changes the conclusion. However, one should remember that outliers are highly subjective but in our case treated 'delicately' by our algorithm. Furthermore, the final sample size (2) is very low which makes it hard to draw any conclusions.

During the course of data collection for this study we encountered some issues, except those mentioned earlier, regarding data handling and analysis worth mentioning. We will present our proposal for solutions for the following issues next:

- Some authors use data from repositories such as the PROMISE database. However, they do not clearly state how the data was extracted from the repository which makes replication tedious and in some cases impossible. For example, in one of the analyzed papers, the data sets were divided into subsets and it was not described how the division was done. This hindered us from recreating the post processing of the data and we had to exclude the paper.

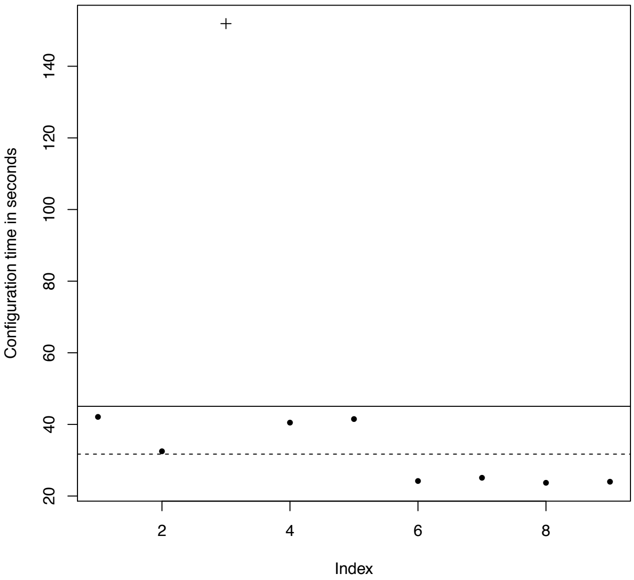

Fig. 3. Plot with a distinct outlier. Data set from [31].

- Even though we were given access to raw data at many times, this data was on the other hand, at times, 'too' raw. Meaning that to get the data needed to reproduce *their* analysis we also needed to perform some kind of data extraction. This is both a difficult and time consuming procedure (thus prone to errors) which further complicates the replication of a study.
- The connection between the raw data and the paper was not obvious for some of the studies we reviewed. For example, one data set used column names which were a combination of abbreviations and words in the authors' native language (not english). This led to some confusion and we needed to contact the author several times in order to understand the published data.
- For some of the papers analyzed we had to contact the authors to have them explain their analysis and motivation behind the choices they made. For example, some papers mentioned that they used a "Wilcoxon" significance test without specifying if it is one or two-sided and/or a paired test. This creates unnecessary uncertainty about the analysis conducted and makes replication more difficult.

When taking into account all the above items it is clear that journals and conferences should require authors to be more explicit in describing study execution and analysis.

Even though the removal of outliers did not affect any conclusions, the algorithm used (MZS) did identify outliers in some cases. As the algorithm is easy to use it could still be of interest for researchers to use this method to quickly identify outliers. However, when the algorithm proposes a data point to be an outlier, this alone is not enough to exclude it from a data set. Researchers are encouraged to use simple detection algorithms, such as MZS, but then use the results from these algorithms to discuss the inclusion or exclusion of data points in their study.

c) Guidelines for Outlier Detection:

- Outlier detection algorithms are only tools to help suggest what data points could be outliers. Researchers should view these suggestions critically and reflect over the results before removing data points.
- A motivation to why a data point is an outlier should always be provided.
- When conducting outlier detection, one should present which data points were regarded as outliers by the algorithm.
- Always document. It is important that no tacit knowledge is needed to replicate the outlier detection and removal conducted.

d) Guidelines for Facilitating Replication:

- When using already available data it is important to present how the information was extracted.
- Use online data storage solutions such as Figshare or Github instead of hosting data on personal university pages.
- The information presented in the paper should clearly correspond to the information in the raw data set. Preferably, the authors should provide a key stating the mapping between the paper and the data set.
- The type of significance test executed should be clearly stated together with a motivation of why this test was chosen.

VI. Threats to Validity

Our study relied on a several previously published studies, but any flaws in the execution of this study, or in the subsequent analysis that was performed, is solely ours. Below we list some of the more important threats to validity.

A. *Internal*

- For conducting outlier detection robust methods were used. These methods used parameters settings that were meant to fit a wide range of data sets and were not specialized. Having used specialized settings for each data set we might discover more/less outliers than we did in this study.
- Removing a full pair in data sets used for pairwise testing could be a validity threat as we could potentially remove non-outliers from one data set. This would then make it difficult to conclude anything about the effect of the outlier detection we did initially. However, we did not conduct this ourselves in this study but we like to underline the threat in conducting such elimination of data points.

B. External

- We only collected a current sample from the last one and a half year. This limited sample might not be representable for studies conducted earlier, but we deemed them to be a representative/stratified sample of current SE research.

C. Conclusion

- Some of the data sets gathered during the data collection did have a small sample size to begin with. The sizes became even smaller when you remove data points suggested as outliers. The initial small sample size is a validity threat to the original study, but the new smaller sample size is a validity threat to our study and in particular in regard to how we draw conclusions regarding the significant difference of data sets before and after outliers are removed.

D. Construct

- Only having one researcher reviewing each study might have caused a bias. To try and mitigate this risk we discussed issues regarding the studies among us.

VII. Conclusion

From the information we collected while preparing our replications we found that 63% of the investigated studies do not document outliers. Furthermore, 38% of the studies had outliers, according to our outlier detection, while not documenting any.

Regarding the data availability, we found that 26% of the studies had their data directly available either in the paper or online. Additionally, 12% of the studies' corresponding authors replied with data after we sent out an email request. In total, 37% of the studies investigated had data available. From this we conclude that the state of replication, in regards to replicating data analysis, is less than desirable within ESE and we believe it is in need of improvement.

In order to help the research field of SE to improve, our study provides the following contributions to the body of knowledge:

- Outliers exists within recently published ESE studies and can be found with robust methods.
- The extent to which recently published ESE studies document outliers.
- The extent to which recently published ESE studies make their data available and how it is made available.
- Guidelines for conducting and presenting outlier detection for ESE.
- Guidelines for how to improve the reproducibility of ESE studies.

VIII. Future Work

For future work we recommend not to conduct more studies regarding outliers and outlier detection on already published studies. Instead, we propose that this should be done on to-be-published studies by journal and conference authors and reviewers through the use of mandatory outlier detection. The reason we propose to not conduct more studies on already published studies is that our results shows that it is, to a certain extent, difficult to obtain data from recently published studies. Hence, we assume that trying to get hold of data from older studies could be even more difficult.

References

[1] I. Sommerville, *Software Engineering: (International Computer Science)*, 8th ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006.

[2] J. W. Osborne and A. Overbay, "The power of outliers (and why researchers should always check for them)," *Practical assessment, research & evaluation*, vol. 9, no. 6, pp. 1–12, 2004.

[3] H.-P. Kriegel, M. S hubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08. New York, NY, USA: ACM, 2008, pp. 444–452. [Online]. Available: http://doi.acm.org/10.1145/1401890.1401946

[4] A. Brooks, M. Roper, M. Wood, J. Daly, and J. Miller, "Replication's role in software engineering," in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. Sjøberg, Eds. Springer London, 2008, pp. 365–379. [Online]. Available: http://dx.doi.org/10.1007/978-1-84800-044-5_14

[5] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, Oct. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:AIRE.0000045502.10941.a9

[6] Y.-S. Seo and D.-H. Bae, "On the value of outlier elimination on software effort estimation research," *Empirical Software Engineering*, vol. 18, no. 4, pp. 659–698, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10664-012-9207-y

[7] K.-H. Yuan and P. M. Bentler, "Effect of outliers on estimators and tests in covariance structure analysis," *British Journal of Mathematical and Statistical Psychology*, vol. 54, no. 1, pp. 161–175, 2001. [Online]. Available: http://dx.doi.org/10.1348/000711001159366

[8] D. N. Card, F. E. McGarry, and G. T. Page, "Evaluating software engineering technologies," *IEEE Transactions on Software Engineering*, vol. 13, no. 7, pp. 845–851, Jul. 1987. [Online]. Available: http://dx.doi.org/10.1109/TSE.1987.233495

[9] V. R. Basili, F. Shull, and F. Lanubile, "Building knowledge through families of experiments," *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 456–473, Jul. 1999. [Online]. Available: http://dx.doi.org/10.1109/32.799939

[10] F. J. Shull, J. C. Carver, S. Vegas, and N. Juristo, "The role of replications in empirical software engineering," *Empirical Software Engineering*, vol. 13, no. 2, pp. 211–218, Apr. 2008. [Online]. Available: http://dx.doi.org/10.1007/s10664-008-9060-1

[11] N. Juristo and S. Vegas, "The role of non-exact replications in software engineering experiments," *Empirical Software Engineering*, vol. 16, no. 3, pp. 295–324, 2011. [Online]. Available: http://dx.doi.org/10.1007/s10664-010-9141-9

[12] D. I. Sjøberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N.-K. Liborg, and A. C. Rekdal, "A survey of controlled experiments in software engineering," *IEEE Transactions on Software Engineering*, vol. 31, no. 9, pp. 733–753, Sep. 2005. [Online]. Available: http://dx.doi.org/10.1109/TSE.2005.97

[13] R. M. Lindsay and A. S. Ehrenberg, "The design of replicated studies," *The American Statistician*, vol. 47, no. 3, pp. 217–228, 1993.

[14] T. W. Tan, J. C. Tong, A. M. Khan, M. de Silva, K. S. Lim, and S. Ranganathan, "Advancing standards for bioinformatics activities: Persistence, reproducibility, disambiguation and minimum information

about a bioinformatics investigation (MIABi)," *BMC genomics*, vol. 11, no. 4, p. S27, 2010.

[15] R. Gentleman, "Reproducible research: A bioinformatics case study," *Statistical applications in genetics and molecular biology*, vol. 4, no. 1, 2005.

[16] L. Preeyanon, A. Black Pyrkosz, and C. T. Brown, "Reproducible bioinformatics research for biologists," in *Implementing Reproducible Computational Research*, V. Stodden, F. Leisch, and R. D. Peng, Eds. Chapman and Hall/CRC, 2014, iSBN 978-1466561595. [Online]. Available: http://www.crcpress.com/product/isbn/9781466561595

[17] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele University and Durham University Joint Report, Tech. Rep. EBSE 2007-001, 2007.

[18] W. N. Venables and B. D. Ripley, *Modern applied statistics with S-PLUS*. New York, NY: Springer New York, 1999.

[19] F. A. A. Garcia, "Tests to identify outliers in data series," 2012. [Online]. Available: http://habcam.whoi.edu/HabCamData/HAB/processed/Outlier%20Methods_external.pdf

[20] IBM, "IBM SPSS modified z score," 2007. [Online]. Available: http://pic.dhe.ibm.com/infocenter/spssas/v1r0m0/index.jsp?topic=%2Fcom.ibm.spss.analyticcatalyst.help%2Fanalytic_catalyst%2Fmodified_z.html

[21] B. Iglewicz and D. C. Hoaglin, *How to detect and handle outliers*. ASQC Quality Press Milwaukee (Wisconsin), 1993, vol. 16.

[22] G. Bavota, B. Dit, R. Oliveto, M. Di Penta, D. Poshyvanyk, and A. De Lucia, "An empirical study on the developers' perception of software coupling," in *Proceedings of the 2013 International Conference on Software Engineering*, ser. ICSE'13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 692–701. [Online]. Available: http://dl.acm.org/citation.cfm?id=2486788.2486879

[23] J. Itkonen and M. Mäntylä, "Are test cases needed? Replicated comparison between exploratory and test-case-based software testing," *Empirical Software Engineering*, vol. 19, no. 2, pp. 303–342, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10664-013-9266-8

[24] L. K. Shar, H. B. K. Tan, and L. C. Briand, "Mining SQL injection and cross site scripting vulnerabilities using hybrid program analysis," in *Proceedings of the 2013 International Conference on Software Engineering*, ser. ICSE'13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 642–651. [Online]. Available: http://dl.acm.org/citation.cfm?id=2486788.2486873

[25] A. Arcuri and G. Fraser, "Parameter tuning or default values? An empirical investigation in search-based software engineering," *Empirical Software Engineering*, vol. 18, no. 3, pp. 594–623, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10664-013-9249-9

[26] L. L. Minku and X. Yao, "An analysis of multi-objective evolutionary algorithms for training ensemble models based on different performance measures in software effort estimation," in *Proceedings of the 9th International Conference on Predictive Models in Software Engineering*, ser. PROMISE '13. New York, NY, USA: ACM, 2013, pp. 8:1–8:10. [Online]. Available: http://doi.acm.org/10.1145/2499393.2499396

[27] J. Wang, X. Peng, Z. Xing, and W. Zhao, "Improving feature location practice with multi-faceted interactive exploration," in *Proceedings of the 2013 International Conference on Software Engineering*, ser. ICSE'13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 762–771. [Online]. Available: http://dl.acm.org/citation.cfm?id=2486788.2486888

[28] J. Nam, S. J. Pan, and S. Kim, "Transfer defect learning," in *Proceedings of the 2013 International Conference on Software Engineering*, ser. ICSE'13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 382–391. [Online]. Available: http://dl.acm.org/citation.cfm?id=2486788.2486839

[29] J. Feigenspan, C. Kästner, S. Apel, J. Liebig, M. Schulze, R. Dachselt, M. Papendieck, T. Leich, and G. Saake, "Do background colors improve program comprehension in the #ifdef hell?" *Empirical Software Engineering*, vol. 18, no. 4, pp. 699–745, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10664-012-9208-x

[30] N. Sawadsky, G. C. Murphy, and R. Jiresal, "Reverb: Recommending code-related web pages," in *Proceedings of the 2013 International Conference on Software Engineering*, ser. ICSE'13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 812–821. [Online]. Available: http://dl.acm.org/citation.cfm?id=2486788.2486895

[31] Y. Y. Lee, N. Chen, and R. E. Johnson, "Drag-and-drop refactoring: Intuitive and efficient program transformation," in *Proceedings of the 2013 International Conference on Software Engineering*, ser. ICSE'13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 23–32. [Online]. Available: http://dl.acm.org/citation.cfm?id=2486788.2486792

[32] L. Song, L. L. Minku, and X. Yao, "The impact of parameter tuning on software effort estimation using learning machines," in *Proceedings of the 9th International Conference on Predictive Models in Software Engineering*, ser. PROMISE'13. New York, NY, USA: ACM, 2013, pp. 9:1–9:10. [Online]. Available: http://doi.acm.org/10.1145/2499393.2499394

[33] C. Parnin, C. Bird, and E. Murphy-Hill, "Adoption and use of Java generics," *Empirical Software Engineering*, vol. 18, no. 6, pp. 1047–1089, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10664-012-9236-6

[34] S. Herbold, "Training data selection for cross-project defect prediction," in *Proceedings of the 9th International Conference on Predictive Models in Software Engineering*, ser. PROMISE'13. New York, NY, USA: ACM, 2013, pp. 6:1–6:10. [Online]. Available: http://doi.acm.org/10.1145/2499393.2499395