Using Blind Analysis for Software Engineering Experiments

Boyce Sigweni   Martin Shepperd
Department of Computer Science
Brunel University London
UB8 3PH,United Kingdom
{boyce.sigweni,martin.shepperd}@brunel.ac.uk

ABSTRACT

Context: In recent years there has been growing concern about conflicting experimental results in empirical software engineering. This has been paralleled by awareness of how bias can impact research results.
Objective: To explore the practicalities of blind analysis of experimental results to reduce bias.
Method: We apply blind analysis to a real software engineering experiment that compares three feature weighting approaches with a naïve benchmark (sample mean) to the Finnish software effort data set. We use this experiment as an example to explore blind analysis as a method to reduce researcher bias.
Results: Our experience shows that blinding can be a relatively straightforward procedure. We also highlight various statistical analysis decisions which ought *not* be guided by the hunt for statistical significance and show that results can be inverted merely through a seemingly inconsequential statistical nicety (i.e., the degree of trimming).
Conclusion: Whilst there are minor challenges and some limits to the degree of blinding possible, blind analysis is a very practical and easy to implement method that supports more objective analysis of experimental results. Therefore we argue that blind analysis should be the norm for analysing software engineering experiments.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management—*Cost estimation*; I.2.6 [**Learning**]: Analogies

Keywords

Researcher Bias, Blind analysis, Software engineering experimentation, Software effort estimation

1. INTRODUCTION

We seek to evaluate blinding techniques, specifically blind analysis, to reduce researcher bias within empirical software engineering. As a vehicle to explore this we use an experiment to compare various feature weighting techniques over a software project effort data set and report our experiences. Our goal is to make blind analysis a more widespread practice.

For some time commentators have been concerned about the lack of agreement amongst the many empirical studies conducted in the various branches of empirical software engineering including defect prediction [19] and software effort estimation (SEE) [15, 22]. Closer investigation suggests that a contributory reason — though only one of many — is selective reporting and partial analysis [21, 14]. One technique for reducing the propensity for bias is to conduct blind analysis [11]. This entails, as a minimum, the labels of the different treatments being anonymised such that the researchers performing the analysis of the results do not know which result data (i.e., the response variables) relate to which treatment. This renders "cherry picking" results more difficult.

In order to explore this technique in a practical setting we apply it to a real life empirical investigation of various feature weighting techniques applied to analogy-based SEE. This involves benchmarking various existing methods, naïve methods and a new method proposed by one of the authors [BS]. The techniques are evaluated on the Finnish software effort data set used in the study by [17].

The remainder of the paper is organised as follows. Next we review what is known about bias in scientific research in general and empirical software engineering in particular. This is followed by a description of the context of our SEE empirical study. Then we give a description of our experimental approach and the decision making involved, results and experimental conclusions. The final section discusses the conduct of, and issues relating to, blinding the analysis.

2. SOURCES OF BIAS IN RESEARCH AND BLINDING TECHNIQUES

> "[L]et us define bias as the combination of various design, data, analysis, and presentation factors that tend to produce research findings when they should not be produced." John Ioannidis [13]

Researchers have been concerned about the potential impact of unintentional bias upon the part of scientists for

EASE'15, April 27 - 29, 2015, Nanjing, China
Copyright 2015 ACM 978-1-4503-3350-4/15/04 ...$15.00.
http://dx.doi.org/10.1145/2745802.2745832

at least the past three decades. In considering this it is important to distinguish between bias where there are systematic underlying reasons and processes leading to wrong research findings and general randomness. Since confidence limits and null hypothesis testing typically set thresholds at 95% this implies an acceptance of 5% of Type I errors, i.e., wrongly rejecting the null hypothesis or where the true population statistic lies outside the estimated and reported sample confidence limits. Conversely, depending upon the power of the study there is also the random possibility of failing to reject the null hypothesis when we should i.e., a Type II error.

There have been concerns that many areas of research ranging from medicine to social policy and experimental psychology to genomics have been impacted by different sources of bias. Delgado-Rodríguez and Llorca [3] have published a catalogue of more than 70 different types of scientific bias. Moreover these exclude those specifically related to data analysis, reporting and citation behaviours. At a generic level these include:

- publication bias [4], which is the reduced likelihood of publishing certain types of study when the results are not perceived as 'interesting'. Generally results seen as not interesting are typically exemplified by the null hypothesis being retained. This may either be due to the peer review process (some results are seen more favourably by the referees than others) or the "file drawer problem" [20] (when researchers fail to complete or submit papers in a non-random way).

- selective reporting in that the study only reports a subset of results [12]. Again this process can lead to the over-reporting of 'interesting' results and the under reporting of non-significant results or results with small or no effect size.

- analysis bias where statistical procedures are selected according to their ability to yield 'interesting' results. In passing we note that null hypothesis significance testing (NHST) is particularly vulnerable since the logic of this approach leads the researcher to an all or nothing situation, of significance or no significance. More than twenty years ago Dickersin observed how significant results are substantially over-represented in the field of medical research [4].

Unfortunately software engineering does not seem to be immune from these biases. A major meta-analysis of 600 results derived from 42 primary studies of defect prediction algorithms found that the research team that conducted the work explained approximately 25 times more variance in the performance of the predictor as did the choice of algorithm [21]. Research group was also more important than the data set used to validate the predictor and considerably more so than choice of metrics or inputs to the predictor. Such biases also confound meta-analyses since the goal to uncover all relevant studies is thwarted by the systematic non-availability of certain types of result. Thus the entire research community is harmed along with our reduced ability to make reliable recommendations to practitioners.

Of course the question arises as to why scientists may exhibit bias. The first thing we wish to be absolutely clear about is that we are *not* suggesting that this bias is intentional or for morally questionable reasons. Possible explanations include the fact that expertise may not be evenly distributed, moreover some techniques are highly sophisticated and the parameter free-space extensive. As a result it is conceivable that a research group may be able to use Technique A more effectively than Technique B. Conversely a second group might behave in the opposite way. Another explanation is the majority of predictors exploit different machine learning techniques [26]. Such research generally proceeds experimentally and there is little theory to guide. Such research tends also to explore many variants of prediction systems often with many different parameter settings. The consequence is many results. This in itself is not necessarily problematic and there are various statistical procedures for adjusting significance thresholds accordingly. However what is less clear and therefore more difficult is the stopping criterion; at what stage should the researchers stop their experiments and report results? And a related problem is should all results be reported? There may be many intermediate results. These kind of problems mean that selective reporting can be difficult to address.

One approach to combat these biases is blinding the analysis [11]. The idea is that by relabeling the different treatments e.g., as predictor 1, 2, ... , n then the researcher conducting the analysis of the results is no longer aware of which is the new 'pet' technique nor which are the results from benchmarks. Searching for a test or procedure that yields statistical significance is less straightforward since it is more difficult for the analyst to have a view as to what results are 'interesting'. Note that only the response variables are blinded, therefore context descriptors will be unchanged. We are unaware of this approach being used in software engineering but there are examples in other disciplines such as physics [1]. Note also that the technique is not appropriate to other forms of empirical analysis such as case studies and focus groups.

Clearly for blind analysis to be effective it requires a minimum of two researchers. Figure 1 outlines the process which we describe in more detail in subsequent sections. Note that in our study Researcher 1 was BS and Researcher 2 was MS. Thus the blinding was achieved as follows. MS selected a data set. The application of the different prediction systems to the data set was performed by BS who then sanitised the treatment labels. Next the results files were passed to MS who performed the statistical analysis. Once this was complete BS revealed the actual treatments which are described in Section 4.

The remainder of this paper reports on our experiences of using blinding when experimentally evaluating a new algorithm for feature weighting when using case-based reasoning to predict software project effort.

3. EXPERIMENTAL DESIGN

The description follows the steps numbered within Figure 1.

Step 1: Researcher 1 determined four different treatments or methods for software effort estimation using various analogy or case-based reasoning (CBR) methods, specifically:

1. Forward sequential weighting (FSW) uses continuous, non-negative weights [24]

2. Forward sequential selection(FSS) uses binary weights, thus a feature is selected or excluded

3. Case-based reasoning (CBR) uses all features equally weighted

4. Naïve prediction uses the sample mean.

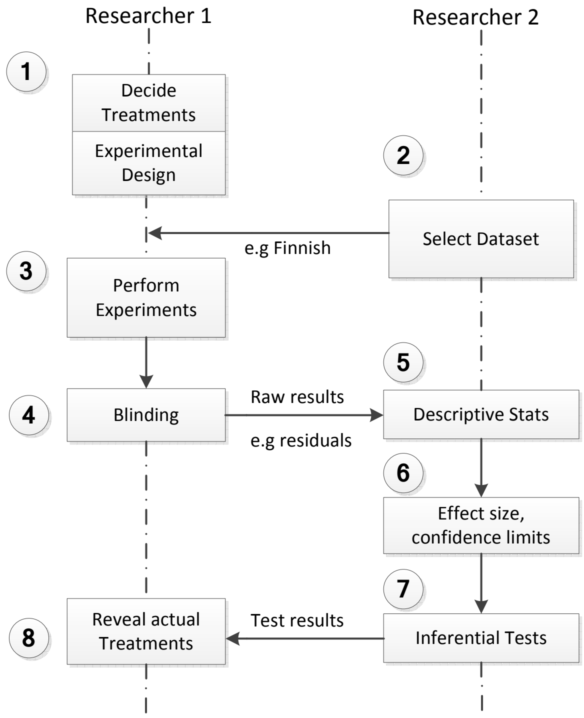

Figure 1: Experimental design and blind analysis process

These methods present a range of possible strategies for analogy based effort estimation. A trivial or naïve approach is included in order to determine the extent to which the more sophisticated techniques offer any value, in other words a baseline we expect to be able to improve upon.

In passing we note that elsewhere we have demonstrated the dangers of not using proper benchmarks and how researchers can be unaware that their methods in fact perform less well than guessing [22]. CBR might be thought of as the baseline well-established technique and has been applied since the mid 1990s [23]. Subsequently FSS has generally been found to be an effective improvement over CBR; our systematic review found 16 out of 17 relevant primary studies reported positive results [25]. Finally FSW is a recent improvement to FSS [24] that uses an efficient algorithm to search for individual feature weights.

The response variables are Standardised Accuracy (SA) [22] and the absolute residuals to measure predictive performance. MMRE is avoided due to its asymmetry and bias towards prediction models that under-estimate [18, 9].

For this experiment a leave-one-out cross-validation (LOOCV) procedure is employed [5]. Although computationally intensive for larger datasets when using a wrapper (because a new predictor has to be built for each case or project in the data set being held out) there is the advantage of the results being deterministic. By comparison, $m \times n$ fold cross validation will depend upon the random allocation of cases to the individual folds and so there is often some variability in the results.

Step 2: The choice of data set(s) is made independently by Researcher 2 without knowledge of the treatments. This is because it could be known that some data sets might particularly favour some SEE methods. It is a relatively complex data set so will challenge a feature weighting technique as there are a large number of cases and features. The data set characteristics are shown in Table 1. The Finnish data set has also been used for studies focusing on meta-heuristic search for FSS e.g.,[17, 16]. This data set is characterised by a skewed distribution of effort values as can be seen from the fact that the mean is considerably greater than the median. A redacted version of the data set is available from [8].

Step 3: The data set used in this study is the same data set used by [17] which removed some features due to missing values so as to ensure none of the projects had missing values. Researcher 1 did not remove any outliers. However, two projects had the actual effort being equal to zero which is hard to interpret as meaningful, therefore these were removed these two projects (so out of the total of 407 projects 405 remain). Researcher 1 then applied all the treatments (prediction methods) to the data set using the archANGEL software tool (adapted to also compute the FSW) and for each result computed the absolute residual i.e, $|y_i - \hat{y_i}|$.

4. EXPERIMENTAL RESULTS

Step 4: All statistical analysis of the results is based on absolute residuals. These were provided with anonymised treatment labels to Researcher 2. Note that for this experiment we were using a repeated measures design and there was no particular need to look at context variables or experimental moderators. In other settings this might be relevant, however, it is only the treatments labels that need blinding consequently blind analysis does not inhibit richer or more sophisticated analysis when appropriate.

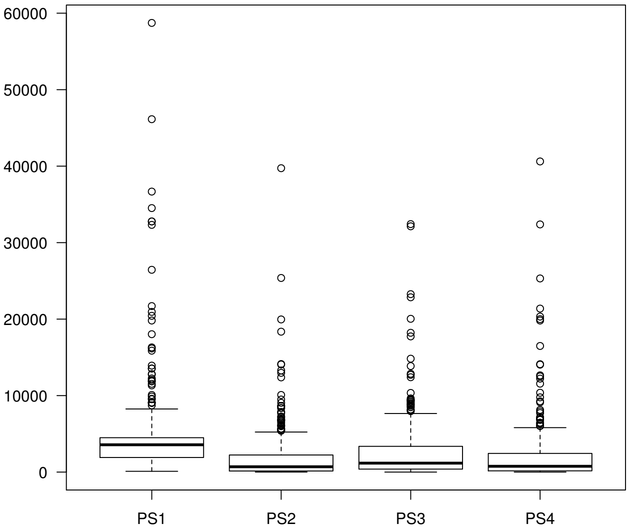

Figure 2: Boxplots showing residual distributions for Finnish Data Set

Table 1: Descriptive statistics for the 'cleaned' Finnish data set

Data set	No. of cases	No. of features	Min effort	Max effort	Mean effort	Median effort
Finnish	405	44	55	63694	5031	2500

Step 5: However, there were a number of challenges relating to the statistical analysis of the experimental results. First, the distributions of the residuals are extremely skewed and not amenable to simple transformations (see the box plots of the absolute residuals for the prediction systems PS1, . . . , PS4. in Figure 2). Second, there are many ties (depending upon the particular pairwise comparison this ranges between 114 and 168 out of 405 cases). Third the data are dependent since we are comparing the performance of four different predictors on the *same* data. Finally alpha needs correcting since multiple pairwise comparisons or tests are needed (in our case six, since there are four treatments).

Table 2: Comparing absolute residuals by prediction system

Pred system	Mean abs residual	Median abs residual
PS2 (FSW)	1911.5	701.9
PS4 (FSS)	2146.8	761.3
PS3 (CBR)	2645.7	1173.3
PS1 (Naïve)	4438.5	3580.0

Step 6: Next Researcher 2 considered the questions of confidence limits for the descriptive statistics such as medians and then measures of effect size. Non-parametric methods are required due to the non-normality of the distributions of the absolute residuals.

The Harrell-Davis percentile estimator [10] with bootstrap was used as an efficient, robust technique to estimate the 95% confidence limits for the median (i.e., the 50th quantile) value of the absolute residuals (see Table 3). If the intervals are compared it would seem some overlap, for example, PS2 and PS4 and others do not, for example, PS3 and PS1. Note that the treatments are listed in decreasing order of performance so that smallest residuals ,and therefore best predictive performance, occur first. The treatments are labelled for the reader's convenience only as this information was not available to Researcher 2 at the time of the analysis.

Table 3: Harrell-Davis 50th percentile estimators for prediction system absolute residuals

Pred system	Lower bound	Upper bound	Estimated median
PS2 (FSW)	581.8	848.1	711.4
PS4 (FSS)	676.6	913.76	772.7
PS3 (CBR)	1069.7	1712.9	1235.8
PS1 (Naïve)	3370.7	3787.2	3597.7

The next part of the analysis was to turn attention to effect size [6], in this case measured as Δ which is defined as the difference in the median absolute residuals for the two treatments being compared and normalised by the pooled standard deviation. This is reported in Table 4 along with the Standardised Accuracy (SA) of each approach relative to guessing based on permutation (see Shepperd and McDonnell [22] for details of the method). Note even using the sample mean is 11% better than guessing. To help interpret the effect sizes relative to guessing, these could be characterised as 'small' (~ 0.2) or 'medium' (~ 0.5) and although not obtained, ~ 0.8 might be regarded as a 'large' effect size [2]. Analysed in this fashion none of the SEE techniques can be seen as particularly successful and is a powerful reminder of how far we still have to go in pursuit of practical, effective SEE.

Table 4: SA and effect size Δ

Approach	Criteria	
	SA (%)	Δ
PS2 (FSW)	62.16	0.427
PS4 (FSS)	57.48	0.395
PS3 (CBR)	47.40	0.326
PS1 (Naïve)	11.42	0.078

Step 7: The basic descriptive analysis from Steps 5 and 6 suggests that the medians of the absolute residuals appear to differ by treatment but this difference needs to be tested using inferential statistics. Likewise Step 6 suggests that the 95% confidence limits from the medians do not all overlap implying significant differences.

Unfortunately traditional non-parametric tests such as Wilcoxon-Mann-Whitney can lack power [7], do not handle ties well and are unlikely to be satisfactory [28]. For this reason a robust test was used to compare differences in marginal medians using Wilcox's percentile bootstrap using the R function dmedpb from the WRS library. Family-wise errors arising from multiple testing were controlled using Rom's method since $k < 10$. For details see Wilcox [27].

The predictors are compared pairwise starting with the greatest median difference. The probability of the median difference = 0 is given by p. The upper and lower bounds give the 95% confidence limits for the median difference therefore for a significant difference one would not expect the limits to straddle zero.

The analysis is shown in Table 5 in which the pairwise comparisons between prediction systems are organised in decreasing order of difference which facilitates the application of Rom's method which is based on the idea of sequential rejection so that once a threshold has been exceeded there is no purpose in testing for smaller differences [28]. Again the results are presented unblinded for the convenience of the reader.

Step 8: The results of the analysis therefore show that whilst the new technique FSW outperforms the naïve

Table 5: Pairwise comparison of median absolute residual differences using Wilcox's percentile bootstrap

Test	p	Lower bound	Upper bound	Median difference
FSW v Naïve	~ 0	-2741.6	-2100.0	-2489.5
FSS v Naïve	~ 0	-2658.8	-1771.1	-2410.0
CBR v Naïve	~ 0	-2140.8	-1227.3	-1758.8
FSW v CBR	~ 0	-457.4	-146.0	-252.7
FSS v CBR	~ 0	-289.3	-58.9	-179.5
FSW v FSS	0.954	-0.5	0	0

sample mean and traditional CBR there is no significant difference with FSS for this particular data set despite a slightly superior effect size Δ and SA value (see Table 4). Thus we cannot argue the new feature weighting technique is superior for this particular data set.

5. DISCUSSION AND CONCLUSIONS

Although the previous section describes the procedure adopted by Researcher 2, in practice MS had a number of decisions to make and no *a priori* reason to consider one superior to another.

- The level of *trimming* to apply since trimming provides a continuum of approaches from including all observations in estimating population characteristics to the other extreme of excluding all but the central point, i.e., the median. Researcher 2 elected to use medians primarily because this is common practice but other decisions might easily be justified such as trimming 10% or 20% of each tail [28]. If we apply a 20% trim (see Table 6) then this yields a different set of results; specifically that there is a significant difference between the absolute residuals from FSW and FSS such that FSW would be reported as significantly superior.

- The choice between *Winsorized trimming* and trimming since Winsorizing involves the replacement of values with the trimmed minimum or maximum as opposed to discarding the values with trimming. The impact of such as choice is unclear.

- The type and direction of the *null hypothesis*, for example one could use one or two tailed tests. Researcher 2 chose to use 2-tailed tests.

- How to *correct alpha* since methods range from Bonferroni's correction which is a conservative method to methods such as Rom's method as adopted by Researcher 2.

- The choice of *inferential test* to compare medians is again somewhat open even if we correctly restrict ourselves to robust methods since these include Cliff's, Brunner-Munzel and Wilcox's methods.

- Lastly, a small but subtle difference is median difference between treatments or comparison of the medians of the treatments

Table 6: Pairwise comparison of <u>mean</u> absolute residual differences using Wilcox's percentile bootstrap (Trimmed means 0.2)

Test	p	Lower bound	Upper bound	Median difference
FSW v Naïve	~ 0	-2764.5	-2219.7	-2492.1
FSS v Naïve	~ 0	-2652.0	-2098.0	-2375.0
CBR v Naïve	~ 0	-2112.9	-1511.9	-1812.4
FSW v CBR	~ 0	-880.2	-479.3	-679.7
FSS v CBR	~ 0	-769.8	-355.5	-562.6
FSW v FSS	~ 0	-75.1	-59.0	-64.7

The decisions taken by Researcher 2, as previously mentioned can lead to a different conclusion. For example, Table 6 shows that using an analysis based on 20% trimmed means results in $p \sim 0$ for the pairwise comparison of FSW v FSS (see the highlighted cell). This strongly contrasts with Table 5 where the same test yields $p = 0.954$. The consequence is that a 'result' may be transformed from insignificant to significant by changing the choice of interferential test. Thus in evaluating FSW v FSS Researcher2 could easily and *'correctly'* employ trimmed means to evaluate FSW v FSS. Trimmed mean looks to reduce the effects of outliers but in a less conservative fashion than analysis based on medians which in a sense is the most extreme form of trimming possible since only the central observation is retained [28]. The choice results in different conclusions for the evaluation of FSW v FSS.

But our point is not which is the most appropriate statistical approach to make comparisons between experiment treatments but that if the analyst has *a priori* expectations, and it's difficult not to, then these can influence the choice of technique and in a highly non-random fashion. Blind analysis does not prevent inappropriate analysis, it does, however, militate against systematic use of statistical methods in order to yield 'positive' results.

So to summarise, it is relatively easy to change the results of a statistical analysis without resorting to scientific misconduct. This is particularly the case for null hypothesis significance testing. For example moving to trimmed means (0.2) has the impact on the results transforming a not significant result (Table 5) in terms of evaluating a new algorithm into a significant one (Table 6).

The basic principle of blind analysis was straightforward to implement. The analyst was only provided with residuals since actual predicted values could potentially jeopardise the blinding for techniques such as using a sample mean since all predicted values would be the same. One advantage of the relatively meaningless values was that the analyst (Researcher 2) could proceed in a somewhat detached fashion.

As a means of reducing systematic bias in terms of statistical and analysis decisions being made in order to achieve particular types of outcome we believe blind analysis has a great deal to commend it. However, it needs to be stressed that blind analysis will not eliminate statistical errors and poor practice but what it does address is statistical procedures being systematically selected on the basis of them yielding desired results.

In this paper we have described our experiences for a single experiment. There is no control and $n = 1$. All this demonstrates is that it is possible to manipulate results without recourse to poor practice or scientific misconduct and that it is straightforward to blind the analysis. Beyond this our argument rests upon advocacy. Nevertheless, we do argue that blind analysis should become normal practice within empirical software engineering when dealing with multiple treatments (and associated response variables) in some experimental or quasi-experimental setting.

Acknowledgements

The authors would like to thank Pekka Forselius for providing the Finnish data set.

6. REFERENCES

[1] E. Aprile and et al. Dark matter results from 225 live days of xenon100 data. *Phys. Rev. Lett.*, 109:181301, Nov 2012.

[2] J. Cohen. A power primer. *Psychological Bulletin*, 112(1):155–159, 1992.

[3] M. Delgado-Rodríguez and J. Llorca. Bias. *J. of Epidemiolgy and Community Health*, 58:635–641, 2004.

[4] K. Dickersin. The existence of publication bias and risk factors for its occurrence. *J. Am. Med. Assoc.*, 263:1385–1389, 1990.

[5] B. Efron and G. Gong. A leisurely look at the bootstrap, the jackknife and cross-validation. *The American Statistician*, 37(1):36–48, 1983.

[6] P. Ellis. *The Essential Guide to Effect Sizes: Statistical Power, Meta-Analysis, and the Interpretation of Research Results*. Cambridge University Press, 2010.

[7] M. W. Fagerland and L. Sandvik. The Wilcoxon–Mann–Whitney test under scrutiny. *Statistics in Medicine*, 28(10):1487–1497, 2009.

[8] Finnish Software Effort Dataset. http://dx.doi.org/10.6084/m9.figshare.1334271. 03 2015.

[9] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrtveit. A simulation study of the model evaluation criterion mmre. *IEEE Transactions on Software Engineering*, 29(11):985–995, 2003.

[10] F. Harrell and C. Davis. A new distribution-free quantile estimator. *Biometrika*, 69(3):635–640, 1982.

[11] J. Heinrich. Benefits of blind analysis techniques. Report CDF/MEMO/STATISTICS/PUBLIC/6576 Version 1, University of Pennsylvania, 2003.

[12] J. Hutton and P. Williamson. Bias in meta-analysis with variable selection within studies. *Applied Statistics*, 49(3):359–70, 2000.

[13] J. Ioannidis. Why most published research findings are false. *PLoS Medicine*, 2(8):e124, 2005.

[14] M. Jørgensen, T. Dybå, K. Liestøl, and D. Sjøberg. Incorrect results in software engineering experiments: How to improve research practices. *J. of Systems and Software*, under review, 2015.

[15] M. Jørgensen and M. Shepperd. A systematic review of software development cost estimation studies. *IEEE Transactions on Software Engineering*, 33(1):33–53, 2007.

[16] C. Kirsopp and M. Shepperd. Case and feature subset selection in case-based software project effort prediction. In *The 22nd BCS SGAI International Conference on Knowledge Based Systems and Applied Artificial Intelligence*, pages 61–74, 2003.

[17] C. Kirsopp, M. J. Shepperd, and J. Hart. Search heuristics, case-based reasoning and software project effort prediction. In *GECCO'02 Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1367–1374. Morgan Kaufmann Publishers Inc., 2002.

[18] B. Kitchenham, S. MacDonell, L. Pickard, and M. Shepperd. What accuracy statistics really measure. *IEE Proceedings - Software Engineering*, 148(3):81–85, 2001.

[19] T. Menzies and M. Shepperd. Editorial: Special issue on repeatable results in software engineering prediction. *Empirical Software Engineering*, 17(1-2):1–17, 2012.

[20] R. Rosenthal. The "file drawer problem" and tolerance for null results. *Psychological Bulletin*, 86(3):638–641, 1979.

[21] M. Shepperd, D. Bowes, and T. Hall. Researcher bias: The use of machine learning in software defect prediction. *IEEE Transactions on Software Engineering*, 40(6):603–616, 2014.

[22] M. Shepperd and S. MacDonell. Evaluating prediction systems in software project estimation. *Information and Software Technology*, 54(8):820–827, Jan. 2012.

[23] M. Shepperd and C. Schofield. Estimating software project effort using analogies. *IEEE Transactions on Software Engineering*, 23(11):736–743, 1997.

[24] B. Sigweni. Feature weighting for case-based reasoning software project effort estimation. In *The 18th International Conference on Evaluation and Assessment in Software Engineering*, page 54. ACM, 2014.

[25] B. Sigweni and M. Shepperd. Feature weighting techniques for CBR in software effort estimation studies: a review and empirical evaluation. In *The 10th International Conference on Predictive Models in Software Engineering*, pages 32–41. ACM, 2014.

[26] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang. Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology*, 54(1):41–59, 2012.

[27] R. Wilcox. Pairwise comparisons of dependent groups based on medians. *Computational Statistics and Data Analysis*, 50(10):2933–2941, 2006.

[28] R. Wilcox. *Introduction to robust estimation and hypothesis testing (3rd Edn)*. Academic Press, 3rd edition, 2012.