Revisiting the Relationship Between Fault Detection, Test Adequacy Criteria, and Test Set Size

Yiqun T. Chen
University of Washington
Seattle, WA, USA

Rahul Gopinath
CISPA Helmholtz-Zentrum
Saarbrücken, Germany

Anita Tadakamalla
George Mason University
Fairfax, VA, USA

Michael D. Ernst
University of Washington
Seattle, WA, USA

Reid Holmes
University of British Columbia
Vancouver, BC, Canada

Gordon Fraser
University of Passau
Passau, Germany

Paul Ammann
George Mason University
Fairfax, VA, USA

René Just
University of Washington
Seattle, WA, USA

ABSTRACT

The research community has long recognized a complex interrelationship between fault detection, test adequacy criteria, and test set size. However, there is substantial confusion about whether and how to experimentally control for test set size when assessing how well an adequacy criterion is correlated with fault detection and when comparing test adequacy criteria. Resolving the confusion, this paper makes the following contributions: (1) A review of contradictory analyses of the relationships between fault detection, test adequacy criteria, and test set size. Specifically, this paper addresses the supposed contradiction of prior work and explains why test set size is neither a confounding variable, as previously suggested, nor an independent variable that should be experimentally manipulated. (2) An explication and discussion of the experimental designs of prior work, together with a discussion of conceptual and statistical problems, as well as specific guidelines for future work. (3) A methodology for comparing test adequacy criteria on an equal basis, which accounts for test set size without directly manipulating it through unrealistic stratification. (4) An empirical evaluation that compares the effectiveness of coverage-based testing, mutation-based testing, and random testing. Additionally, this paper proposes probabilistic coupling, a methodology for assessing the representativeness of a set of test goals for a given fault and for approximating the fault-detection probability of adequate test sets.

CCS CONCEPTS

• **Software and its engineering** → **Software testing and debugging**; **Empirical software validation**.

KEYWORDS

Fault detection, Test set size, Mutation testing, Code coverage

ACM Reference Format:
Yiqun T. Chen, Rahul Gopinath, Anita Tadakamalla, Michael D. Ernst, Reid Holmes, Gordon Fraser, Paul Ammann, and René Just. 2020. Revisiting the Relationship Between Fault Detection, Test Adequacy Criteria, and Test Set Size. In *35th IEEE/ACM International Conference on Automated Software Engineering (ASE '20), September 21–25, 2020, Virtual Event, Australia.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3324884.3416667

ASE '20, September 21–25, 2020, Virtual Event, Australia

ACM ISBN 978-1-4503-6768-4/20/09…$15.00
https://doi.org/10.1145/3324884.3416667

1 INTRODUCTION

The software engineering research community has long recognized a complex interrelationship between three variables:

- *fault detection* (the degree to which a test set detects real faults),
- *test set adequacy* (the degree to which a test set satisfies a set of test goals, such as statements, branches, or mutants), and
- *test set size* (the cardinality of a test set).

Fault detection is the best estimate for a test set's efficacy, but fault detection is not directly measurable since the number of unfound faults in a program is unknowable. As a result, developers and researchers use test set adequacy criteria, such as code coverage or mutant detection, as a proxy measure.

There is a positive association between test set size and the other two variables. For example, adding a test to a given test set cannot decrease fault detection or test set adequacy. Similarly, reducing a given test set cannot increase fault detection or test set adequacy. Moreover, best practices (e.g., modularity and separation of concerns) result in a strong association between test set adequacy and test set size. For example, a developer may write one test per use case or function. Namin and Andrews [39] empirically showed a strong association between fault detection, test set adequacy, and test set size, and noted that large test sets with low adequacy and small test sets with high adequacy were unattainable in practice.

However, beyond the observation that the three variables are positively associated, the strength of the associations and their precise relationships are a matter of open debate and controversy in the research community.

Consider Fig. 1, which shows the relationship between the three variables: fault detection, test set adequacy (specifically, code coverage ratio and mutant detection ratio), and test set size. For each of 231[1] real faults from the Defects4J benchmark [29], we created 100 coverage-adequate test sets and 100 mutation-adequate test sets, greedily selecting tests from the pool of existing developer-written tests that accompany the fault. (At each selection step, tests are selected at random, and the first test that increases test set adequacy is added to the test set.) During test selection, we measured all three variables at each test selection step. Figure 1 plots the aggregated

[1]This paper uses a subset of Defects4J for consistency with prior work [43].

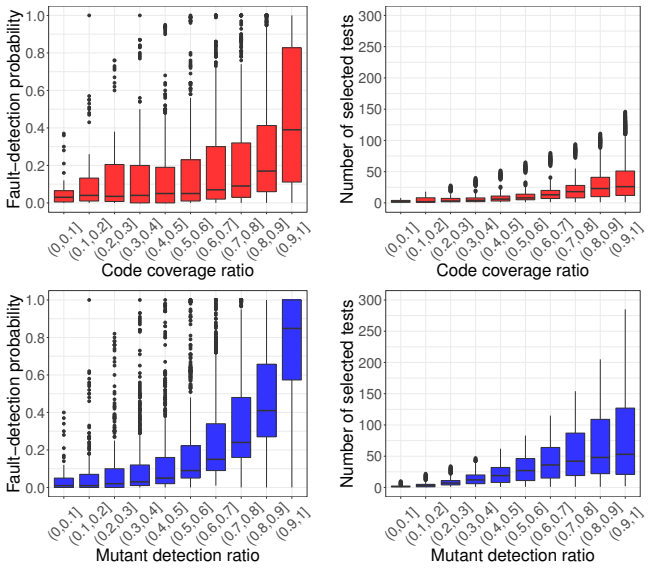


Figure 1: Test sets generated to achieve a mutant detection rate r exhibit higher fault-detection, but are also substantially larger, than those generated to achieve coverage rate r. These plots show the results of adequacy-based test selection for 231 faults (details in Section 6). Each adequacy bucket, i.e., all test sets that have a coverage ratio or mutation detection ratio in the indicated range, includes 231 data points.

results (fault-detection probability and average test set size) for each of the 231 faults and given adequacy threshold (shown in buckets for simplification). Each data point in an adequacy bucket corresponds to one of the 231 real faults.

Figure 1 illustrates the positive association between test set adequacy, fault detection, and test set size: test sets with higher adequacy (code coverage ratio or mutant detection ratio) have a higher fault detection probability and are larger in size. What Fig. 1 cannot answer, however, is which of the two adequacy criteria provides better selection goals at any point in the process, including the end point. More precisely, consider two test sets: T_m, generated to achieve 100% mutant detection, and T_c, generated to achieve 100% code coverage. According to Fig. 1, T_m has a greater fault-detection probability than T_c on average; however, T_m is also substantially larger than T_c on average. Does T_m have greater fault detection because mutation provides better test goals than coverage, or does it have greater fault detection just because satisfying mutation adequacy requires more tests? Likewise, does selecting tests based on an adequacy criterion achieve greater fault detection than randomly selecting the same number of tests? Consequently, should a developer select tests based on coverage, mutation, or just randomly?

One approach previously used to answer this question measures the correlation between test set adequacy and fault detection for fixed test set sizes. This stratification approach repeatedly draws test sets of the same size *independently and uniformly at random* from a test pool, and analyzes the results for each stratum [22, 43].

Prior work has also proposed two alternatives. Alternative 1 creates test sets based on a given test budget and objective, and then measures and correlates fault detection and test set adequacy [30]. Alternative 2 considers existing test sets, created based on some test objective, and assesses the importance of test set adequacy and test set size when modeling fault detection [13].

These three approaches to teasing out the role of test set size are all valid approaches in principle. However, the experiments in the literature adopting these approaches have resulted in contradictory conclusions. There are multiple reasons for this, including the interpretation of correlation values, noise in the data, and the applied models of (random) test selection.

This paper investigates and resolves the supposed contradiction of prior work; its contributions and organization are as follows:

- A review of four contradictory analyses of the relationship between fault detection, test set adequacy, and set size (Section 2), and an explication of their experimental designs (Section 3).
- A discussion of conceptual problems, explaining why test set size is neither a confounding variable, as previously suggested, nor an independent variable that should be experimentally manipulated (Section 4) and statistical pitfalls (Section 5).
- A methodology for comparing test-adequacy criteria that accounts for test set size without directly manipulating it through stratification (Section 6).
- An empirical evaluation that compares the effectiveness of coverage-based testing, mutation-based testing, and random testing (Sections 6.1 to 6.4). Additionally, this paper proposes probabilistic coupling, a methodology for assessing the representativeness of a set of test goals for a given fault and for approximating the fault-detection probability of adequate test sets (Section 6.5).

Consistent with prior work, this paper uses test set size as a proxy for the cost of creating or executing a set of tests; Section 7 discusses the validity of using test set size as a proxy for these variables.

2 TEST SET SIZE IN PRIOR WORK

Previous refereed papers that study the relationship between fault detection, test set adequacy, and test set size report on experiments with contradictory conclusions. This is a serious scientific problem: Which ones are trustworthy?

As examples, this section provides two pairs of papers with similar research questions but contradictory results: Gopinath et al. [13] and Inozemtseva and Holmes [22]; Just et al. [30] and Papadakis et al. [43]. These papers report contradictory conclusions about:

- whether and how test set size should be experimentally controlled when assessing the correlation between test set adequacy and fault detection, and
- whether the correlation between test set adequacy and fault detection is significant and strong.

While some previous work notes these conflicts without providing resolutions, of greater concern is the fact that many other papers simply cite the aforementioned papers without noting the contradictions. (As of August 2020, Google Scholar reports over 800 citations to these four papers.)

This section briefly reviews each of these four papers. Each review focuses on the aspects relevant to the apparent contradictions and does not necessarily provide a complete summary of contributions and findings. Sections 3 to 5 resolve the conflicts by showing which papers have flawed experimental or statistical methodology.

2.1 Gopinath et al. (ICSE'14)

Gopinath et al. [13] investigated whether code coverage is strongly correlated with fault detection for seeded faults (mutants). The study used 250 Java projects of different sizes and characteristics.

Test source Both developer-written tests (existing test sets) and automatically-generated tests (Randoop [42]) were separately analyzed. Automatically-generated tests were created based on a fixed time budget.

Test set sampling None.
Each test set (developer-written and automatically-generated) was used for analysis without sampling.

Test adequacy measures Statement coverage, block coverage, branch coverage, and path coverage.

Statistical methodology Considering all 250 Java projects, the study used regression analysis, with mutant detection as the dependent variable and test set adequacy, along with project size and cyclomatic complexity, as independent variables. Test set size was not included in the model to avoid multicollinearity (the study identified a strong correlation between test set size and project size); test set size was represented in the model via project size as a proxy.

Results and conclusions Statement coverage on its own was strongly correlated with mutant detection, and this correlation was stronger compared to those of all other studied code coverage criteria. The correlation between statement coverage and mutant detection was stronger for developer-written test sets than for automatically-generated test sets. Test set size did not increase model accuracy when test set adequacy was already included in that model.

2.2 Inozemtseva and Holmes (ICSE'14)

Inozemtseva and Holmes [22] investigated whether code coverage is strongly correlated with fault detection for seeded faults (mutants), when test set size is ignored and controlled for. The study used five Java projects of different sizes and characteristics.

Test source Developer-written tests (existing test sets).

Test set sampling Random sampling and stratification.
For each of the five Java projects and different test set sizes (3, 10, 30, 100, 300, 1,000, and 3,000 tests—up to the maximum size possible for that project), the study sampled 1,000 test sets of fixed size. In total, the study sampled 31,000 test sets across the five projects and different test set sizes. The study analyzed these test sets, both with and without controlling for test set size. Each test set was sampled uniformly at random without replacement.

Test adequacy measures Statement coverage, decision coverage, and modified condition coverage.

Statistical methodology For each of the five Java projects, the study measured and correlated code coverage and mutant detection of the randomly sampled test sets, both with and without controlling for test set size.

Results and conclusions The correlations between statement coverage and mutant detection were moderate to strong when test set size was ignored, with almost identical results for stronger code coverage criteria. The correlations became negligible to moderate when test set size was controlled.

2.3 Just et al. (FSE'14)

Just et al. [30] investigated whether mutant detection is strongly correlated with fault detection. The study used five Java programs with 357 real faults (Defects4J).

Test source Both developer-written tests (existing test sets) and automatically generated tests (EvoSuite [12], JCrasher [9], and Randoop [42]) were separately analyzed. Automatically-generated tests were created based on a fixed time budget.

Test set sampling None.
Each test set (developer-written and automatically-generated) was used for analysis without sampling.

Test adequacy measures Statement coverage and mutant detection.

Statistical methodology For each real fault, the study analyzed pairs of existing pre-fix and post-fix developer-written test sets and measured and correlated mutant detection and fault detection of automatically-generated test sets, both with and without controlling for code coverage. Test set size was ignored because it was not significantly associated with an increase in fault detection for pairs of pre-fix and post-fix developer-written test sets, and it was irrelevant for automatically-generated test sets, which were created based on a fixed time budget without further sampling.

Results and conclusions The correlation between mutant detection and fault detection was moderate to strong, and remained significant when code coverage was controlled for. The correlation between mutant detection and fault detection was stronger than the correlation between statement coverage and fault detection.

2.4 Papadakis et al. (ICSE'18)

Papadakis et al. [43] investigated whether mutant detection is strongly correlated with fault detection, when test size is ignored and controlled for. The study used five Java programs with 231 real faults (a subset of Defects4J)[2].

Test source Both developer-written tests (existing test sets) and automatically-generated tests (EvoSuite [12] and Randoop [42]). For each of the 231 real faults, all developer-written tests and automatically-generated tests were combined into a large test pool.

Test set sampling Random sampling and stratification.
For each real fault, the study sampled (1) 10,000 test sets of fixed size (in the range of 0-50% of the test pool size, with 2.5% increments) and (2) 10,000 test sets of random size (in the range of 0-20% of the test pool size). Each test set was sampled uniformly at random without replacement.

Test adequacy measures Mutant detection.

Statistical methodology For each real fault, the study measured and correlated mutant detection and fault detection of the randomly sampled test sets, both with and without controlling for test set size.

Results and conclusions The correlations between mutant detection and fault detection were moderate to strong when test set size was ignored. The correlations became negligible to weak when test set size was controlled.

[2]The study separately analyzed four C programs, reaching the same conclusions.

2.5 Apparent Contradictions

Code coverage vs. artificial fault detection Gopinath et al. [13] built a regression model that used code coverage to predict fault detection. The study concluded that code coverage is strongly correlated with fault detection and adding test set size to the regression model did not improve its accuracy. In apparent contradiction, Inozemtseva and Holmes [22] used stratification to control for test set size and computed correlations between code coverage and fault detection at each strata. Based on the small absolute values of the correlations, the study concluded that code coverage is not strongly correlated with fault detection, when test set size is taken into account. In other words, Gopinath et al. found test set size to play essentially *no role*, whereas Inozemtseva and Holmes found test set size to play a *dominant role*. (Note that Gopinath et al. performed a regression analysis across projects, whereas Inozemtseva and Holmes performed a correlation analysis for each project.)

Mutant detection vs. real fault detection Just et al. [30] correlated mutant detection and fault detection for pairs of pre-fix and post-fix developer-written test sets and, separately, automatically-generated test sets. The study concluded that mutant detection is strongly correlated with fault detection for most real faults, and the correlation between mutant detection and fault detection was stronger than the correlation between statement coverage and fault detection. Furthermore, test set size was an insignificant factor when comparing pre-fix and post-fix developer-written test sets.

In apparent contradiction, Papadakis et al. [43] expanded the approach used by Inozemtseva and Holmes, used stratification to control for test set size, and computed correlations between mutant detection and fault detection. Based on the small absolute values of the correlations, the study concluded that mutant detection is only weakly correlated with fault detection, when test set size is taken into account. Given the observed weak correlations, when test set size was controlled, the study concluded that *test set size is a confounding variable* that explains fault detection.

3 ABSTRACT EXPERIMENT DESIGN

The four papers described in Section 2 yielded seemingly contradictory results about the relationships between fault detection, test set adequacy, and test set size. This section describes (1) the experimental artifacts and design of these papers at an abstract level and (2) the constraints these artifacts imposed on the experimental results.

All experiments in Section 2 have the same overall structure. First, there is a universe of tests from which it is possible to sample a test set according to some distribution. Second, there is an oracle that determines, for each test, whether a test detects a fault—success or failure event. Note that a (failing) test that detects a fault corresponds to the *success event*, and a (passing) test that does not detect a fault corresponds to the *failure event*. Third, there is an adequacy criterion that imposes test goals on the system under test. For each test, it is possible to compute the set of test goals satisfied by that test, as well as a summary statistic that captures the overall criteria satisfaction. For example, for mutation-based testing, the former amounts to computing the set of mutants detected by a given test, and the latter amounts to computing the overall mutant detection ratio.

The crucial step is choosing the methodology for subsequent analysis, which we outlined in Section 1 and expand upon here:

(1) RandomSelection: Create (by various means) a large, fixed universe of tests (i.e., the *test pool*), sample test sets from this test pool *uniformly at random without replacement*, and measure and correlate fault detection and test set adequacy. This methodology has two variants:
   - SizeFixed: Fix the test set size via stratification and analyze the results independently for each stratum.
   - SizeRandom: Draw the test set size itself from a distribution, prior to sampling a test set.

(2) Alternative 1: Create test sets based on a given (time) budget and test objective, and measure and correlate fault detection and test set adequacy.

(3) Alternative 2: Consider existing test sets, created according to some test objective, measure fault detection, test set adequacy, and test set size, and statistically assess the contribution of test set adequacy and test set size.

We will expand on the RandomSelection methodology because it is popular, yet unrealistic and prone to wrong conclusions, and its underlying problems are easily neglected. The subsequent sections explicate and address its conceptual and statistical problems.

Section 4 first describes why the RandomSelection methodology is problematic at a conceptual level: Test set size is an unrealistic test objective (Section 4.1), and test generation in practice does not yield an independent random sample (Section 4.2).

Section 5 further provides a technical explanation of the statistical pitfalls of the RandomSelection methodology: Using highly correlated explanatory variables (Section 5.1), ignoring the bounds of the point biserial correlation (Section 5.2), and ignoring the class imbalance effect due to extremely low or high fault-detection probabilities (Section 5.3) produces misleading results.

4 CONCEPTUAL PROBLEMS

Prior work using the RandomSelection methodology performed a correlation analysis of test set adequacy and fault detection over randomly sampled test sets; in particular, it modeled test set creation as a (uniform) random selection process of tests from a finite test pool. There are two often neglected conceptual issues with this methodology: The first is concerned with the use of test set size as test objective and the second is concerned with how representative the random selection process is for test set creation in practice.

4.1 Test Set Size is an Unrealistic Test Objective

A test adequacy criterion can be used for either test set creation, where an adequacy criterion provides test goals that facilitate test selection under a given budget, or test set evaluation, where an adequacy criterion provides an adequacy score for an existing test set (e.g., generated by developer preference).

However, using the RandomSelection methodology to create a test set of a given size, measured as the number of tests, does not model developer behavior. *Developers do not use the number of tests as a test objective*—nor should they. An appropriate stopping criterion in terms of number of tests is unknowable—should a developer write 10 tests or 1,000 tests for a given program, and how? In practice, developers write tests using a mix of many different objectives, such as exercising a new feature, exposing a defect (i.e., regression test), or optimizing for an adequacy criterion [23, 44].

Furthermore, what exactly constitutes a single test is not well defined since real-world tests are decidedly non-uniform (unit vs. integration vs. system tests, one vs. many assertions per test, etc.) [54, 56]. For example, Just et al. [30] found that developers were equally likely to strengthen an existing test or adding a new test when exposing a defect. In other words, the number of tests was irrelevant in this case and did not accurately measure test set size.

Test adequacy criteria, such as code coverage and mutant detection, may be imperfect test objectives, but in contrast to test set size they are well defined and provide concrete test goals.

In conclusion, analyzing the correlation between test set adequacy and fault detection using the RandomSelection methodology does not provide actionable insights for real-world test creation.

4.2 Test Generation vs. Independent Random Sample

For an implementation of a simple function (e.g., $y = f(x)$, a test generation approach might sample inputs x uniformly at random without replacement from the universe of all inputs (e.g., all integers), and repeat this process independently for many iterations. However, automated test generators for, e.g., object-oriented programs, do not sample tests via independent random sampling. In fact, most automated test generators create subsequent tests based on the tests generated thus far. Hence, the RandomSelection methodology, while simple to execute, is not a realistic model of either developers or automated test generators. Note that the key conceptual issue is the non-guided random selection of n tests—selection based on a realistic test objective is reasonable.

The Eclat tool [41] introduced feedback-directed random test generation, which was popularized by Randoop [42]. When Randoop generates a valid test (a sequence of instructions), it uses that test to build subsequent, larger tests; in other words, it guides the search toward that valid test. When Randoop generates an invalid test, it prohibits creation of tests that build on it; in other words, it guides the search away from the invalid test. Additionally, Randoop eliminates redundant and subsumed tests from the final test set. Feedback-directed generation was motivated, in part, by the poor performance of independent random sampling.

Another example of a test generation tool is EvoSuite [12], which uses meta-heuristic search to generate test sets that maximize code coverage while minimizing size. Meta-heuristic search is essentially a sampling process of an implicit probability distribution in that search space using a combination of stochastic search operators. The main operators are mutation of tests, which may insert, remove, or replace arbitrary statements in a candidate sequence, and crossover of two tests, where subsequences of two parents are recombined to two new offspring sequences. The sampling process is guided by fitness functions that estimate how close tests are to satisfying test goals. The number of calls in a sequence is treated as a secondary objective, such that given two tests that are equal in terms of their fitness, the shorter one is preferred. The search uses a many-objective optimization algorithm, where a single population of individuals is evolved with respect to all test goals; thus, sampled tests are *not independent* and, in fact, *very much dependent*.

In conclusion, the RandomSelection methodology is not a good approximation of test creation processes in practice—neither for developers nor for automated test generators.

5 STATISTICAL PITFALLS

Measuring the efficacy of an adequacy criterion by correlating test set adequacy and fault detection, while controlling for test set size, has become increasingly popular in empirical studies [17, 22, 39, 43]: First, a large number of test sets is created based on random sampling from an existing test pool. Consistent with prior sections, we refer to the two variants of this RandomSelection methodology as SizeFixed if each test set has the same size (e.g., 30 tests or 10% of the test pool), and SizeRandom if the size of each test set follows a non-trivial probability distribution (e.g., 0–20% of the test pool, uniformly at random). Second, the adequacy criterion of interest (e.g., code coverage or mutant detection) and fault detection are computed for each test set. Finally, a higher empirical correlation between fault detection and test set adequacy corresponds to a more effective adequacy criterion.

Despite its popularity, the correlation analysis, based on the RandomSelection methodology, has three statistical pitfalls that are often neglected in practice.

First, correlation analysis does not allow causal conclusions without further assumptions on the underlying process. For instance, one cannot statistically distinguish a confounding variable (e.g., test set size impacts both fault detection and test set adequacy with no direct relationship between the two) from a mediating variable (e.g., test set size impacts fault detection by itself *and* through test set adequacy) [36, 37]. In particular, the notion of "confounding effects" [43] implies a causal relationship, which cannot be concluded from simply a change in correlations [20].

Second, correlation between a dichotomous variable (e.g., fault detection), which takes on either success or failure, and a continuous variable (e.g., mutant detection) is bounded, and the bound depends on the success probability of the dichotomous variable.

Third, correlating two variables based on stratification on a third is a crude approximation to what is known as the partial correlation [11, 49]. The stratification leads to a limited range of the resulting correlation and, in particular, often makes it incomparable to the non-stratified version [49].

The rest of this section explains why the RandomSelection methodology in general, and its SizeFixed variant in particular, is prone to misleading results and wrong conclusions. Section 5.1 argues that attempts to attribute the fault detection "contribution" to one of two highly correlated variables are ill-posed; the section proceeds to discuss some alternative data analysis methods. Section 5.2 explains that the absolute value of the point biserial correlation (a special case of the Pearson correlation) is bounded, with a maximum much smaller than 1 in many cases. Section 5.3 expands on the bounded-correlation observation and analyzes the RandomSelection methodology, used in previous studies; this section reveals the neglected flaws of this methodology, especially the correlations obtained with the SizeFixed variant.

5.1 Highly Correlated Explanatory Variables

In practice, test set adequacy and test set size are highly correlated, and both are correlated with fault detection. However, the question of which of the two "explains" fault detection is ill-posed and cannot be answered by measuring the correlation, e.g., between test set adequacy and fault detection, while controlling for test set size.

More precisely, when controlling for one of the two variables, the observed correlation of fault detection and the other variable will necessarily be weaker.[3] In particular, partial correlations may yield spurious results and do not necessarily give rise to an interpretable coefficient because their sign and range are quite sensitive to the pairwise correlations among the variables being studied [11, 49].

Therefore, a more principled data analysis method to investigate the *relative importance* of test set adequacy and test set size is to employ multiple regression and use effect size measures such as regression coefficients. Another possible approach is to use a standard variable selection method to see how much additional predictive power test set adequacy provides, in addition to test set size, when used to predict fault detection. We refer interested readers to more detailed expositions in Härdle and Simar [16] and James et al. [24].

5.2 Point Biserial Correlation is Bounded

Pearson correlation, with point biserial correlation as its special case, is widely used to characterize the relationship between two variables, and standard guidelines exist for interpreting the resulting coefficient (e.g., weak vs. strong correlation) [8, 34]. However, these guidelines are only appropriate when the Pearson correlation ranges from −1 to +1.

Consider a Pearson correlation between a continuous variable (e.g., test set adequacy) and a dichotomous variable (e.g., fault detection). This special case of the Pearson correlation is also known as the point biserial correlation[4]. The coefficient of this correlation is at most 0.8, and its range can be even smaller.

Following Gradstein [14] and Cheng and Liu [6], Eq. (1) expresses the maximal correlation between a normally distributed, continuous variable and a dichotomous variable, as a function of the latter's success probability p, where z_{1-p} is the $1-p$ quantile of a standard normal distribution:

$$r_{\max} = \frac{1}{\sqrt{2\pi p(1-p)}} \exp(-\frac{1}{2}(z_{1-p})^2), \tag{1}$$

We observe that (1) the maximal correlation, for $p = 0.5$, is close to 0.8 instead of 1, (2) the maximal correlation decreases monotonically as p moves away from 0.5, and (3) the maximal correlation is symmetric around $p = 0.5$. For example, the maximal correlation for $p = 0.1$ and $p = 0.9$ is 0.58, and the standard guidelines for interpreting the Pearson correlation would (incorrectly) conclude that two perfectly correlated variables are *not* strongly correlated. This is not a hypothetical problem. Consider the Defects4J dataset and a large test pool of automatically-generated test sets (say, from Randoop and EvoSuite). Some Defects4J faults are almost always detected with many fault-detecting tests per test set, whereas others are almost never detected with many non-fault-detecting tests per test set [43, 48, 51]. For these faults, the RandomSelection methodology results in very low or very high probabilities of success, and hence small maximal correlation coefficients.

Very low or very high probabilities of success can result in a significant underestimation of the true correlation [3, 50]. This problem is particularly pronounced for the SizeFixed variant. For example, 11 out of 20 test set sizes used in prior work [43] result in success probabilities of $p < 0.1$ or $p > 0.9$ for *most* faults. In fact, even the most balanced out of 20 test set sizes results in success probabilities of $p < 0.1$ or $p > 0.9$ for more than 20% of the faults.

A possible "fix" is to normalize the correlation by an upper bound such as the one computed from Eq. (1). However, not only does the maximal correlation depend on the exact distribution of the continuous variable (which makes the exact upper bound hard to compute), such procedures also lack statistical guarantees [52].

5.3 The RandomSelection Methodology is Flawed

This section shows that the correlation between test set adequacy and fault detection, computed with the RandomSelection methodology, is extremely sensitive to the distribution of fault-detecting tests in a test pool, which resulted in misleading interpretations of the SizeFixed results in previous work. To provide concrete examples, this section uses the 231 Defects4J faults and the corresponding, combined test pools, used by Papadakis et al. [43] (see Section 2.4).

We first make precise the mathematical relationships between test set size, fault detection, and test set adequacy under the SizeFixed variant. Let N denote the test pool size, $K \leq N$ the number of fault-detecting tests in the test pool, and $n \leq N$ the test set size; then the number of fault-detecting tests X (out of the n sampled tests) follows a hypergeometric distribution $HG(N, K, n)$, taking on integer values between 0 and $\min(K, n)$:

$$\mathbb{P}(X = x) = \frac{\binom{N-K}{n-x}\binom{K}{x}}{\binom{N}{n}}. \tag{2}$$

Given a test set size n, the probability of success p (i.e., the probability of sampling a fault-detecting test set), therefore, is equal to the probability of sampling at least one fault-detecting test:

$$p = \mathbb{P}(X \geq 1) = 1 - \mathbb{P}(X = 0) = \begin{cases} 1 - \frac{\binom{N-K}{n}}{\binom{N}{n}} & n \leq N - K \\ 1 & n > N - K \end{cases} \tag{3}$$

This analysis can be extended to the SizeRandom variant, where n itself is now drawn from a distribution, and the resulting probability of success p is a convolution of different distributions.

This analysis can also be extended to computing the probability of satisfying any single test goal for a given adequacy criterion (e.g., detecting a mutant). Combining these individual test goal distributions into a cumulative distribution for the corresponding test set adequacy measure (e.g., the mutant detection ratio) requires knowing the interdependencies between test goals, but is nonetheless computable. However, given the complexity of such an analysis, a common approach is to resort to simulations with sampled test sets to get a close approximation [46]. The key point is that test set size completely determines the distribution of test set adequacy and fault detection.

[3] Consider the extreme case in which two explanatory variables X_1 and X_2 are perfectly correlated (e.g., $X_1 = X_2$), and both are correlated with an outcome variable Y. Fixing the value of X_1 also fixes the value of X_2; hence, the conditional variance of $X_2|X_1$ is 0 and a partial correlation cannot even be computed. The general version of this problem is known as multicollinearity and results in unreliable estimates in regression analysis [10, 53].

[4] Prior work [43] missed the fact that the point biserial correlation is mathematically equivalent to the Pearson correlation, when arguing in favor of one over the other.

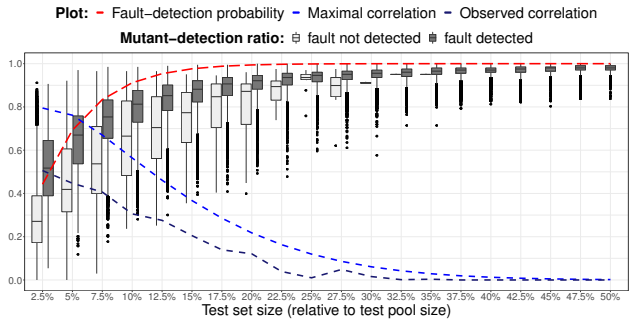


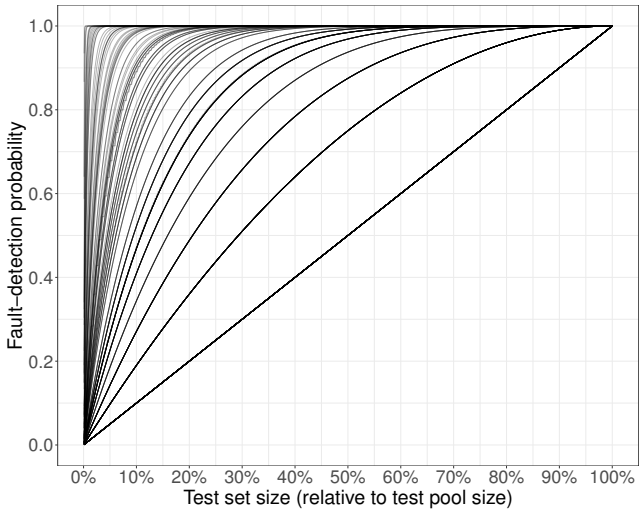
Figure 2: Relationship between mutant detection and fault detection when controlling for test set size (Closure-100).
The underlying test pool contains 6,068 tests, out of which 23 are fault-detecting. The red line gives the probability of sampling a fault-detecting test set, as a function of test set size (Eq. (3)). The blue lines give the theoretical maximal correlation (Eq. (1)) and the observed correlation between mutant detection and fault detection, respectively. (The observed correlation is set to 0 if all sampled test sets are fault-detecting.) For all but the smallest test set sizes, almost all sampled test sets are fault-detecting. This class imbalance leads to very small correlation coefficients, even if the underlying correlation is actually strong.

For a fixed test size n, the probability of success largely depends on the ratio K/N, which leads to the class imbalance effect, motivated in Section 5.2.

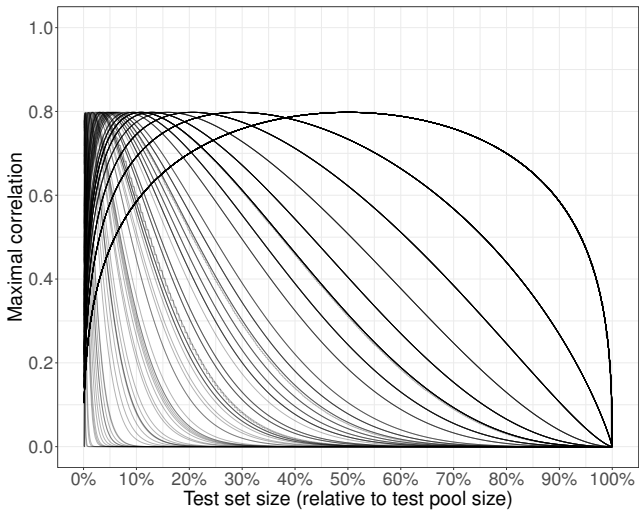
Figure 2 precisely demonstrates this class imbalance effect and its implication on the maximal and observed correlations. For a particular fault (Closure-100), we used the SizeFixed variant to sample 10,000 test sets for each of 20 distinct test sizes, following the methodology described in Section 2.4. For each test set size, Fig. 2 shows the results, in particular: (1) the mutant detection ratios of the sampled test sets (grouped by fault detection), (2) the probability of sampling a fault-detecting test set (Eq. (3)), and (3) the maximal correlation (Eq. (1)) and the observed correlation. The results show that the probability of sampling a fault-detecting test set is close to 1 for all but very small test set sizes. For example, at 10% test set size, that probability is 0.91 (in expectation, over 90% of the sampled test sets are fault-detecting), which leads to a maximal correlation of 0.56. A naive interpretation of the observed correlation coefficients would conclude that mutant detection and fault detection are weakly correlated at best, for most test set sizes.

We extended the analysis of the class imbalance effect to all 231 faults. For each fault, Fig. 3 plots the distribution of the fault-detection probabilities and the corresponding distribution of the maximal correlations, as a function of test set size. Figure 3a demonstrates that (1) the probability of detecting a fault monotonically increases as a function of test set size, and (2) different faults exhibit distinct fault-detection probability curves. More precisely, at 10% test set size, 231 faults take on 80 distinct probabilities. Note that the more upper-left a curve is, the more fault-detecting tests exist in that fault's test pool. Similarly, the diagonal corresponds to faults with only a single fault-detecting test. Connecting these distributions to the maximal correlation discussion in Section 5.2, Fig. 3b plots the maximal correlation[5] between mutant detection and fault detection for each fault, again as a function of test set size.

[5]Assuming that mutant detection is normally distributed.



(a) Fault-detection probability.



(b) Theoretical maximal correlation.

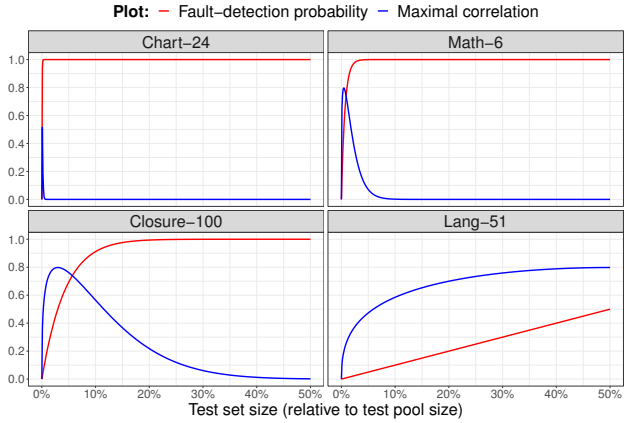
Figure 3: Fault-detection probability and theoretical maximal correlation as a function of test set size for all 231 faults.
Each line is a distinct fault (231 in total). **(a)** For faults with many fault-detecting tests in their test pool, the fault-detection probability (i.e., selecting at least one fault-detecting test) is nearly a step function; for faults with only one fault-detecting test in their test pool, the fault-detection probability is a straight line. **(b)** For faults with five or more fault-detecting tests in their test pool (50% of all faults), the maximal correlation peaks before 5% test set size; for faults with only one fault-detecting test in their test pool (30% of all faults), the maximal correlation peaks at 50% test set size.

The key takeaway from Fig. 3 is that the maximal correlation is a function of the test set size and varies drastically across faults and different test set sizes. For example, even at 10% test set size, 15% of all faults have a maximal correlation of less than 0.1. A naive interpretation of the observed correlation coefficients would always conclude that mutant detection and fault detection are weakly correlated or uncorrelated, even if they were perfectly correlated.

At a fixed test set size, the correlations between mutant detection and fault detection have drastically different maximums for different faults. Therefore, summary statistics (e.g., median) and boxplots can be misleading since each individual fault has a very different range of variation across different test set sizes. To illustrate this point, consider the four selected faults in Fig. 4, whose ratios of fault-detecting tests differ substantially. Figure 4a plots the fault detection probability and maximal correlation for these faults, as a function of test set size. The maximal correlations for Chart-24, Math-6, and Closure-100 peak very early, compared to Lang-51. Figure 4b reports the observed correlations between mutant detection and fault detection for SizeFixed sampling at 5%, 10%, 15%, and 20%, together with a SizeRandom sampling, where test set size is drawn uniformly at random from 0–20%. For SizeFixed, the median correlation across the four faults and all test set sizes is 0.065, and a naive interpretation of this correlation coefficient is that mutant detection and fault detection are uncorrelated. Note that extending the range of the SizeFixed sampling to 0–50% makes matters worse—the observed correlations become 0 for most test set sizes and faults, except for Lang-51. Further note that the class imbalance also affects the SizeRandom variant, though to a lesser extent. For example, 99% and 96% of the SizeRandom sampled test sets for Chart-24 and Math-6, respectively, are fault-detecting, whereas only 9% of the sampled test sets for Lang-51 are fault-detecting. Closure-100 is most balanced with 78% fault-detecting test sets.

While the correlation computed under the SizeRandom variant is larger than any of the SizeFixed ones for Chart-24, Math-6, and Closure-100, the opposite holds for Lang-51. This contradicts the "confounding effects" theory (mutant detection is weakly correlated with fault detection after controlling for test set size), put forward by Papadakis et al. [43], but is consistent with our analysis above: a relatively large ratio of fault-detecting tests for Chart-24 and Math-6 means that larger observed correlation coefficients occur below 2.5% test set size, which only the SizeRandom variant samples. In contrast, Lang-51 has only one fault-detecting test and the maximal correlation peaks at 50% test set size. This means the SizeRandom variant samples more of the low correlation region. If the "confounding effects" theory holds, one would expect the correlation *to be attenuated for all faults*. This, however, is not the case.

In conclusion, the RandomSelection methodology can be misleading and should be interpreted with care, if not avoided all together. In particular, the class imbalance problem has contributed to unsubstantiated claims and incorrect conclusions in previous work.



(a) Fault-detection probability and theoretical maximal correlation between fault detection and mutant detection.

Fault	Fault-detecting tests in test pool	Test set size (relative to test pool size)				
		SizeFixed				SizeRandom
		5%	10%	15%	20%	0–20%
Chart-24	1998 (42%)	0.00	0.00	0.00	0.00	0.38
Math-6	152 (2.0%)	0.01	0.00	0.00	0.00	0.42
Closure-100	23 (0.38%)	0.45	0.31	0.21	0.12	0.70
Lang-51	1 (0.01%)	0.57	0.62	0.66	0.70	0.39

(b) Observed correlations between fault detection and mutant detection for both variants of the RandomSelection methodology.

Figure 4: Case study for four selected faults.
Chart-24 and Lang-51 correspond to the two extremes in Fig. 3, and Closure-100 corresponds to the detailed example in Fig. 2. For Chart-24 and Math-6, the fault-detection probability is indistinguishable from 1 for all but the smallest test set sizes: it is highly unlikely to sample a non-fault-detecting test set, even when sampling 10,000 test sets. (The observed correlation is set to 0 if all sampled test sets are fault-detecting.) The median correlation across the four faults and all SizeFixed test set sizes is 0.065. The class imbalance problem also affects SizeRandom: out of the 10,000 sampled test sets for each fault, 99% are fault-detecting for Chart-24, 96% for Math-6, 78% for Closure-100, and 9% for Lang-51.

6 CONTROLLING FOR TEST SET SIZE

Section 1 motivated controlling for test set size when answering research questions about the effectiveness of test adequacy criteria. However, Sections 4 and 5 demonstrated the adverse effects of directly manipulating test set size as an independent variable in a random selection process. This section resolves this dilemma and describes a methodology that accounts for test set size, without directly manipulating it and without changing the test objective.

Recall Figure 1, which compared mutation-based with coverage-based testing. It is not surprising that mutation-adequate test sets achieve a higher fault detection probability than coverage adequate test sets—mutation adequacy requires satisfying more test goals. However, Figure 1 does not directly answer questions such as:

- *Q1: Is mutation-based test generation more effective than coverage-based test generation, for test sets of the same size?*
- *Q2: Does mutation provide better test goals than code coverage, or does it simply elicit more tests?*

The primary goal of this section is to describe a general methodology for evaluating testing approaches, taking test set size into account. Additionally, this section reports on comparisons between mutation-based testing, coverage-based testing, and random testing, using the described methodology.

This section consistently uses (1) the 231 Defects4J faults, used in prior work [43], together with all developer-written tests for each fault, and (2) mutation and coverage information obtained from the Major mutation framework [28] for each fault and test.

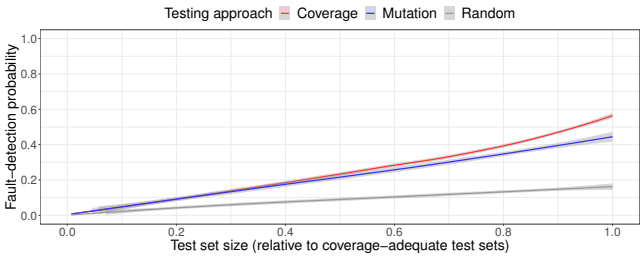


Figure 5: Fault-detection probability for coverage-based test selection, compared to equally sized baselines.
The trend lines are fitted over all 231 bugs and the test set size is normalized over the final number of tests in each coverage-adequate test set.

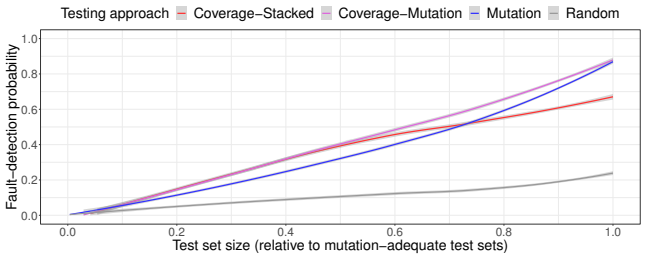


Figure 6: Fault-detection probability for mutation-based test selection, compared to equally sized baselines.
The trend lines are fitted over all 231 bugs and the test set size is normalized over the final number of tests in each mutation-adequate test set.

6.1 Adequacy-based test selection

Sometimes, developers need a subset of an existing test set that runs faster. Developers often do test selection manually, based on their intuition, but they may also use an adequacy-based test selection approach. There might also be a run-time budget (e.g., to enable a continuous integration server to run tests hourly, a selected test set must run in less than an hour). In particular, we note that developers rarely use random selection with test set size as the test objective.

We use an adequacy-based test selection approach and measure fault detection, test set adequacy, and test set size. Specifically, adequacy-based test selection performs the following steps:

(1) Greedily select one test at a time to incrementally achieve adequacy for a given adequacy criterion. (Tests are selected at random; the first one that satisfies at least one additional test goal is added to the test set.)
(2) As each test is added to the test set, record fault detection fd (0 or 1 for every ⟨test set, fault⟩ pair), test set adequacy a, and the number of tests n.
(3) Test selection stops as soon as all test goals are satisfied.

Our experimental methodology repeats the above adequacy-based test selection procedure 100 times for each fault, and we report averages over the 100 trials, which yield small error estimates.

Additionally, we compute a random baseline for each fault, each trial, and each test set size n. The fault-detection probability in this case can be computed from Eq. (3) in Section 5.3.

6.2 Comparing inadequate test sets

Comparing two adequacy criteria, accounting for test set size, is straightforward until the weaker criterion is satisfied. Figure 5 shows such a comparison for coverage-based and mutation-based testing. This plot shows the fault-detection probability of equally sized test sets, created by increasing coverage (red line) and increasing mutant detection (blue line), respectively. The trend lines are fitted over all 231 faults using local regression [7], and test set size is normalized over the final number of tests in each coverage-adequate test set. This plot sheds light on the first question (Q1) and shows that, on average, selecting tests based on coverage yields test sets that are as strong or stronger than those selected based on mutation (at least until coverage is satisfied). Moreover, both coverage and mutation are superior to random selection (gray line). The latter is consistent with the findings of Andrews et al. [2].

6.3 Comparing adequate test sets

Since coverage exhausts its usefulness much earlier than mutation, we need a different approach to further compare the two. Together, Figures 1 and 5 show that mutation-adequate test sets will eventually be more effective than coverage-adequate ones, but it is not clear whether mutants provide additional value beyond simply expanding the test set (e.g., with randomly selected tests).

To assess the independent contribution of mutants as test goals, we again need a baseline test set of equal size. One possibility to achieve this is to use a stacking approach [15]. This means that whenever a weaker test-adequacy criterion is satisfied, the resulting test set is preserved and the test selection process restarted—again optimizing for the same adequacy criterion. The final test set is then the union of all created test sets. Another possibility is a hybrid approach—switching to the stronger adequacy criterion as soon as the weaker one is satisfied.

Figure 6 shows the outcome of a testing simulation using both the stacking (*Coverage-Stacked*) and the hybrid (*Coverage-Mutation*) approach. Random is again included as a baseline. Overall, the *Coverage-Mutation* approach achieves the highest fault-detection probability at each step in the test-selection process.

The *Coverage-Mutation* approach is similar to the mutation testing set up at Google [44, 45]. While Google's decision to implement such an approach was driven by practicality and efficiency concerns, our results provide empirical evidence for the effectiveness of such an approach.

6.4 Adequacy-based test set reduction

In addition to adequacy-based test selection, focusing on adequacy-based test set reduction can provide useful insights into the sensitivity of an adequacy criterion and the expected loss of fault-detection capability. Figure 7 shows the outcome of coverage-based and mutation-based test set reduction, in comparison to an equally sized random baseline.

Figure 7 shows the loss in fault-detection probability when reducing a fault-detecting test set based on coverage or mutation—that is, when creating a smaller, yet adequate, test set. Since these criteria result in test sets of different size, the *Random* baseline differs for each criterion. The black dots indicate medians, showing that most test sets reduced based on mutants maintain their fault-detection capability, whereas test sets reduced based on coverage see a substantial loss of fault-detection capability.

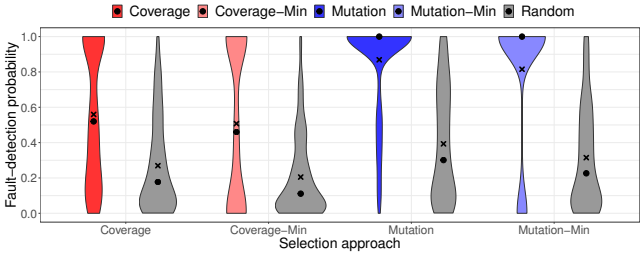
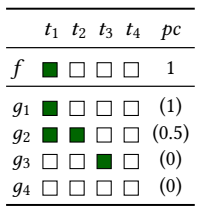
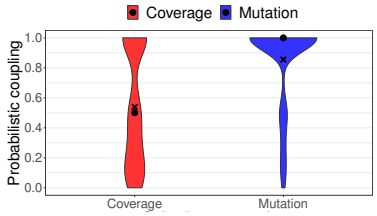


Figure 7: Fault-detection probability of coverage-adequate and mutation-adequate test sets for all 231 faults.

The left pair of violin plots (labeled "Coverage") correspond to two of the 1.0 points in Figure 5. This figure shows all 231 values (● indicates the median and × indicates the mean), whereas Figure 5 shows only one value—the fitted average across all 231 faults. Likewise, the third pair (labeled "Mutation") correspond to two of the 1.0 points in Figure 6. The text details the *Min* approaches.



(a) Test-goal matrix.



(b) Maximal pc for 231 faults.

Figure 8: Example test-goal matrix (left) and distribution of the maximal probabilistic coupling (pc) values for all 231 faults, for coverage and mutation (right).

The two violin plots correspond to the "Coverage" and "Mutation" plots in Fig. 7, showing that probabilistic coupling closely approximates the fault-detection probability of adequate test sets. ■ indicates that t_i detects f or satisfies g_j, ● indicates the median, and × indicates the mean.

Coverage-Min and *Mutation-Min*, refer to approximated minimum test sets, derived from greedily picking tests that maximize the corresponding criterion. Specifically, *Coverage-Min* is a test set generated in the same way as *Coverage* (described in Section 6.1), but always choosing the globally best test (the one in the test pool that maximizes coverage) rather than greedily choosing the first one that increases coverage at all. Next to it in Figure 7 is a randomly-generated test set of the same size. *Mutation-Min* is analogous.

6.5 Probabilistic coupling

The key takeaway from Section 5 is that even a well established and understood statistical measure, such as a correlation coefficient, may require a nuanced interpretation in software engineering research due to problems that arise from a limited set of known faults, class imbalance in fault detection, and noise (irrelevant test goals). Without expertise in statistics, and even with, this is difficult and prone to incorrect conclusions and seemingly contradictory results.

Section 6.4 described a general methodology for assessing and comparing adequacy criteria, accounting for test set size without experimentally manipulating it. However, estimating fault-detection probabilities and comparing adequacy criteria still requires costly simulations. Further, the limited set of known real faults and test goals unrelated to those faults introduce noise.

This section proposes a new measure, probabilistic coupling, for assessing the sensitivity of a set of test goals for a known real fault. Specifically, given a real fault f and a test goal g_i, probabilistic coupling provides an estimate for the conditional probability $p = \mathbb{P}(\text{detect } f \mid g_i \text{ is detected})$—that is, the probability of detecting the real fault when selecting a test that satisfies the test goal. If $p = 1$, we say that g_i is perfectly coupled to f. If $p = 0$, we say that g_i is perfectly decoupled from f. Otherwise, we say that g_i is probabilistically coupled to f.

Given a set of test goals, we compute the maximum probabilistic coupling between any of the test goals and the real fault. This is because we have incomplete knowledge about the set of all possible real faults. Computing the maximum allows reasoning about the sensitivity of the employed test goals for a known real fault and is agnostic to noise caused by unrelated test goals.

Figure 8a gives an example for a test-goal matrix with four test goals g_i, four tests t_j, and one real fault f. The symbol ■ indicates that t_i detects f or satisfies g_j. In this example, g_1 is perfectly coupled to f since every test that satisfies g_1 also detects f. Test goals g_3 and g_4 are perfectly decoupled—g_4 even unsatisfiable. The probabilistic coupling for g_2 is 0.5: it is satisfied by two tests, one of which detects f. The maximal probabilistic coupling is 1, which means that the set of test goals is highly sensitive to f.

Figure 8b shows the maximal probabilistic coupling for each of the 231 faults and their corresponding test goals (coverage and mutation) in the buggy code. Since probabilistic coupling does not capture the complex interdependencies between test goals, it is an approximation of the fault-detection probabilities in Figure 7.

For mutation-based testing, probabilistic coupling is related to the coupling effect and the notion of fault coupling, used in prior work [26, 30, 40]. For example, Just et al. focused on perfect fault coupling when studying the relationship between faults and mutants, using existing test sets [30]. If a fault-detecting test did not detect any additional mutants, compared to an existing non-fault-detecting test set, then the corresponding fault was considered not coupled to any of the mutants. This is a conservative approach for estimating how many real faults a mutation-based selection approach may have missed. We argue that in the context of adequacy-based testing, a probabilistic view on coupling is more appropriate. For example, a mutant may be detected by multiple tests. If only one of them is not a fault-detecting test, probabilistic coupling is still high and better approximates the probability of detecting the fault.

The notion of fault coupling is also closely related to mutant subsumption [1, 33, 38]. Indeed, when considering a real fault as just another mutant in the mutant-test matrix, then a mutant is perfectly coupled to a real fault if that mutant subsumes the real fault. We expect that incorporating subsumption information into the probabilistic coupling measure will further improve the estimates of fault-detection probabilities. We leave a deeper investigation as future work.

In summary, measuring probabilistic coupling has two key advantages. First, it does not require costly simulations to estimate fault-detection probabilities. Second, it is robust to noise, introduced by irrelevant test goals and tests.

7 DISCUSSION

Measuring test set size This paper measures test set size as the number of test methods—for consistency with previous work and to enable direct comparisons (e.g., [4, 5, 18, 21, 25, 27, 35, 47, 55]. For JUnit, this is the number of `@Test` annotations[6]. This measure assumes that all test methods are similar in terms of run time and number of exercised program behaviors. This assumption, however, rarely holds in practice [31]. For example, developer-written test methods range from unit tests to system tests, executing just a few instructions or millions of instructions. A single test method may exercise a single or multiple program behaviors—the latter could, in fact, be considered a set of distinct tests. Furthermore, some test methods validate the behavior of a program by simply ensuring that it does not crash, while others use a set of complex assertions. The number of test methods is only one possible measure of test set size. Alternative measures include number of lines of test code and number of assertions [32, 56].

Test set size as a proxy When constructing a test set, a developer's goal is not to "write *n* tests", but rather to exercise program behavior, with number of tests being a consequence of secondary concerns such as adhering to coding guidelines and best practices. Researchers also do not care about test set size per se, but rather use it as a proxy for a quantity of interest, such as test execution or construction cost. Test set size, however, is an imperfect and unreliable proxy for such quantities. For example, an entire set of tests can be combined into a single test method (e.g., table-driven testing), or split into an arbitrary number of test methods, without changing the overall test execution or construction costs.

- *Test execution cost* can and should be directly measured as run time of a test. (This assumes that the test can be run automatically, without human intervention or judgment.)
- *Test construction cost* can and should be directly measured for automatically-generated tests. For example, when comparing two test adequacy criteria, this comparison can be based on the same effort (i.e., the same test-generation budget for each of the criteria) rather than counting the number of generated tests.

Correlation does not imply causation The driving question for research about the interrelationship between fault detection, test adequacy criteria, and test set size is *causal* in nature. For example, that research aims to understand whether creating test sets based on an adequacy criterion yields a high degree of fault detection in practice, and if so, which adequacy criterion is most (cost) effective.

However, causal reasoning is a separate issue from statistical estimation, and different causal relationships can give rise to the same statistical observations. For example, the observation of a reduced correlation between test set adequacy and fault detection when controlling for test set size is compatible with multiple causal models—one of which is that *test set size is a confounder* [43]. An alternative causal model is that a developer writes tests to increase adequacy, which in turn results in a larger test set and higher fault detection (i.e., *test set adequacy is a confounder*). Yet another causal model is that *neither variable is a confounder* and the root cause of the observed correlations among all three variables is developers' desire to write effective tests, based on a variety of test objectives, while adhering to coding guidelines and best practices—thereby increasing test set size, test set adequacy, and fault detection.

We recommend that future studies should explicitly state their causal models and assumed underlying processes, which forces a clear statement of scientific questions and enables reasoning about whether the proposed experiments and analyses can answer that question [19]. Moreover, future studies should explicitly state their statistical quantities and justify why these answer the (causal) question of interest. For example, prior work operated under the assumption that test set adequacy has an impact on fault detection, if and only if the conditional and unconditional correlation between the two has the same distribution. However, the formal statistical quantity of interest—the conditional correlation (*Cor* (Fault detection, Test set adequacy | Test set size)) rarely makes an appearance, let alone an explanation as to why an equality in distributions translates into a causal impact or lack thereof.

When does correlation imply causation? As the research community continues to explore whether and how an adequacy-driven approach to test generation yields effective tests, experiments with clear causal frameworks can shed light on the practical impacts of such approaches, while preventing the aforementioned conflation between causal inference and statistical inference. While scale, content, and even realization of the actual experiments may vary (e.g., randomization may not be possible), we encourage these developer-centric experimentations because they best approximate the practical benefits of different testing approaches by avoiding unrealistic assumptions and test objectives such as test uniformity and test-set-size-driven development.

8 CONCLUSIONS

This paper addresses and resolves the contradictions in prior work that studied the interrelationship between fault detection, test adequacy criteria, and test set size. It explains why test set size is an unrealistic test objective and neither a confounding variable nor an independent variable that should be experimentally manipulated. Furthermore, it explains the conceptual and statistical issues that arise when controlling for test set size via random selection and stratification, concluding that the random-selection methodology is flawed.

Additionally, this paper proposes (1) a methodology for comparing test adequacy criteria on a fair basis, accounting for test set size without direct, unrealistic manipulation, and (2) probabilistic coupling, a methodology for approximating the fault-detection probability of adequate test sets. Using the proposed methodology, this paper concludes that adequacy-based test selection is superior to random selection and that mutation-based test selection is most effective when employed after coverage has exhausted its usefulness.

Finally, this paper argues that the number of test methods is not a reliable measure for test set size. Highlighting the non-uniformity of real-world test methods, it further discusses the validity of using this measure as a proxy for test-creation and test-execution cost.

ACKNOWLEDGMENTS

This paper is based upon work supported by the National Science Foundation under grants CNS-1823172 and CCF-1942055. We would like to thank Papadakis et al. [43] for discussions about their work and for sharing the corresponding data set.

[6] For JUnit 3, it is the number of test methods that follow JUnit's naming conventions.

REFERENCES

[1] Paul Ammann, Marcio Eduardo Delamaro, and Jeff Offutt. 2014. Establishing theoretical minimal sets of mutants. In *Proceedings of the International Conference on Software Testing, Verification and Validation (ICST)*. 21–30.

[2] J. H. Andrews, L. C. Briand, Y. Labiche, and A. S. Namin. 2006. Using Mutation Analysis for Assessing and Comparing Testing Coverage Criteria. *IEEE Transactions on Software Engineering (TSE)* 32, 8 (Aug 2006), 608–624.

[3] Gilbert Becker. 1986. Correcting the point-biserial correlation for attenuation owing to unequal sample size. *The Journal of Experimental Education* 55, 1 (1986), 5–8.

[4] Jennifer Black, Emanuel Melachrinoudis, and David Kaeli. 2004. Bi-criteria models for all-uses test suite reduction. In *Proceedings of the International Conference on Software Engineering (ICSE)*. 106–115.

[5] Xiang Chen, Lijiu Zhang, Qing Gu, Haigang Zhao, Ziyuan Wang, Xiaobing Sun, and Daoxu Chen. 2011. A test suite reduction approach based on pairwise interaction of requirements. In *ACM Symposium on Applied Computing (SAC)*. 1390–1397.

[6] Ying Cheng and Haiyan Liu. 2016. A short note on the maximal point-biserial correlation under non-normality. *Brit. J. Math. Statist. Psych.* 69, 3 (2016), 344–351.

[7] William S Cleveland. 1979. Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American statistical association* 74, 368 (Dec. 1979), 829–836.

[8] Jacob Cohen. 2013. *Statistical Power Analysis for the Behavioral Sciences*. Academic Press.

[9] Christoph Csallner and Yannis Smaragdakis. 2004. JCrasher: An Automatic Robustness Tester for Java. *Software: Practice and Experience* 34, 11 (Sept. 2004), 1025––1050.

[10] Donald E Farrar and Robert R Glauber. 1967. Multicollinearity in Regression Analysis: The Problem Revisited. *The Review of Economic and Statistics* 49, 1 (1967), 92–107.

[11] Joseph L Fleiss and Judith M Tanur. 1971. A Note on the Partial Correlation Coefficient. *The American Statistician* 25, 1 (1971), 43–45.

[12] Gordon Fraser and Andrea Arcuri. 2011. EvoSuite: Automatic Test Suite Generation for Object-Oriented Software. In *Proceedings of the Joint Meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering (ESEC/FSE)*. 416––419.

[13] Rahul Gopinath, Carlos Jensen, and Alex Groce. 2014. Code Coverage for Suite Evaluation by Developers. In *Proceedings of the International Conference on Software Engineering (ICSE)*.

[14] Mark Gradstein. 1986. Maximal Correlation Between Normal and Dichotomous Variables. *Journal of Educational Statistics* 11, 4 (Dec. 1986), 259–261.

[15] Michael Harder, Jeff Mellen, and Michael D Ernst. 2003. Improving test suites via operational abstraction. In *Proceedings of the International Conference on Software Engineering (ICSE)*. 60–71.

[16] Wolfgang Karl Härdle and Léopold Simar. 2015. *Applied Multivariate Statistical Analysis*. Springer, Berlin, Heidelberg.

[17] F. Hariri, A. Shi, V. Fernando, S. Mahmood, and D. Marinov. 2019. Comparing Mutation Testing at the Levels of Source Code and Compiler Intermediate Representation. In *Proceedings of the International Conference on Software Testing, Verification and Validation (ICST)*. 114–124.

[18] M Jean Harrold, Rajiv Gupta, and Mary Lou Soffa. 1993. A methodology for controlling the size of a test suite. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 2, 3 (1993), 270–285.

[19] Miguel A Hernán. 2018. The C-Word: Scientific Euphemisms Do Not Improve Causal Inference From Observational Data. *American Journal of Public Health* 108, 5 (May 2018), 616–619.

[20] Miguel A Hernán, Sonia Hernández-Díaz, Martha M Werler, and Allen A Mitchell. 2002. Causal knowledge as a prerequisite for confounding evaluation: an application to birth defects epidemiology. *American journal of epidemiology* 155, 2 (Jan. 2002), 176–184.

[21] Hwa-You Hsu and Alessandro Orso. 2009. MINTS: A general framework and tool for supporting test-suite minimization. In *Proceedings of the International Conference on Software Engineering (ICSE)*. 419–429.

[22] Laura Inozemtseva and Reid Holmes. 2014. Coverage Is Not Strongly Correlated With Test Suite Effectiveness. In *Proceedings of the International Conference on Software Engineering (ICSE)*. 435–445.

[23] Marko Ivanković, Goran Petrović, René Just, and Gordon Fraser. 2019. Code Coverage at Google. In *Proceedings of the Joint Meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering (ESEC/FSE)*. 955–963.

[24] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning: with Applications in R*. Springer, New York, NY.

[25] Dennis Jeffrey and Neelam Gupta. 2005. Test suite reduction with selective redundancy. In *Proceedings of the International Conference on Software Maintenance (ICSM)*. 549–558.

[26] Yue Jia and Mark Harman. 2011. An Analysis and Survey of the Development of Mutation Testing. *IEEE Transactions on Software Engineering (TSE)* 37, 5 (2011), 649–678.

[27] James A Jones and Mary Jean Harrold. 2003. Test-suite reduction and prioritization for modified condition/decision coverage. *IEEE Transactions on Software Engineering (TSE)* 29, 3 (2003), 195–209.

[28] René Just. 2014. The Major mutation framework: Efficient and scalable mutation analysis for Java. In *Proceedings of the International Symposium on Software Testing and Analysis (ISSTA)*. San Jose, CA, USA, 433–436.

[29] René Just, Darioush Jalali, and Michael D. Ernst. 2014. Defects4J: A database of existing faults to enable controlled testing studies for Java programs. In *Proceedings of the International Symposium on Software Testing and Analysis (ISSTA)*. 437–440.

[30] René Just, Darioush Jalali, Laura Inozemtseva, Michael D. Ernst, Reid Holmes, and Gordon Fraser. 2014. Are mutants a valid substitute for real faults in software testing?. In *Proceedings of the Symposium on the Foundations of Software Engineering (FSE)*. 654–665.

[31] René Just, Gregory M Kapfhammer, and Franz Schweiggert. 2012. Using non-redundant mutation operators and test suite prioritization to achieve efficient and scalable mutation analysis. In *Proceedings of the International Symposium on Software Reliability Engineering (ISSRE)*. 11–20.

[32] René Just, Chris Parnin, Ian Drosos, and Michael D. Ernst. 2018. Comparing Developer-Provided to User-Provided Tests for Fault Localization and Automated Program Repair. In *Proceedings of the International Symposium on Software Testing and Analysis (ISSTA)*. 287–297.

[33] Bob Kurtz, Paul Ammann, Jeff Offutt, Márcio E Delamaro, Mariet Kurtz, and Nida Gökçe. 2016. Analyzing the validity of selective mutation with dominator mutants. In *Proceedings of the Symposium on the Foundations of Software Engineering (FSE)*. 571–582.

[34] Joseph Lee Rodgers and W Alan Nicewander. 1988. Thirteen Ways to Look at the Correlation Coefficient. *The American Statistician* 42, 1 (Feb. 1988), 59–66.

[35] Jun-Wei Lin, Chin-Yu Huang, and Chu-Ti Lin. 2008. Test suite reduction analysis with enhanced tie-breaking techniques. In *Proceedings of the International Conference on Management of Innovation and Technology (ICMIT)*. 1228–1233.

[36] David P MacKinnon, Amanda J Fairchild, and Matthew S Fritz. 2007. Mediation analysis. *Annual Review of Psychology* 58 (2007), 593–614.

[37] D P MacKinnon, J L Krull, and C M Lockwood. 2000. Equivalence of the mediation, confounding and suppression effect. *Prevention Science* 1, 4 (Dec. 2000), 173–181.

[38] M. Marre and A. Bertolino. 2003. Using spanning sets for coverage testing. *IEEE Transactions on Software Engineering (TSE)* 29, 11 (2003), 974–984.

[39] Akbar Siami Namin and James H. Andrews. 2009. The influence of size and coverage on test suite effectiveness. In *Proceedings of the International Symposium on Software Testing and Analysis (ISSTA)*. 57–68.

[40] A Jefferson Offutt. 1992. Investigations of the software testing coupling effect. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 1, 1 (1992), 5–20.

[41] Carlos Pacheco and Michael D. Ernst. 2005. Eclat: Automatic generation and classification of test inputs. In *Proceedings of the European Conference on Object-Oriented Programming (ECOOP)*. 504–527.

[42] Carlos Pacheco, Shuvendu K. Lahiri, Michael D. Ernst, and Thomas Ball. 2007. Feedback-directed random test generation. In *Proceedings of the International Conference on Software Engineering (ICSE)*. 75–84.

[43] Mike Papadakis, Donghwan Shin, Shin Yoo, and Doo-Hwan Bae. 2018. Are Mutation Scores Correlated with Real Fault Detection? A Large Scale Empirical study on the Relationship Between Mutants and Real Faults. In *Proceedings of the International Conference on Software Engineering (ICSE)*. 537–548.

[44] Goran Petrović and Marko Ivanković. 2018. State of mutation testing at Google. In *Proceedings of the International Conference on Software Engineering (ICSE)*. 163–171.

[45] Goran Petrović, Marko Ivanković, Bob Kurtz, Paul Ammann, and René Just. 2018. An Industrial Application of Mutation Testing: Lessons, Challenges, and Research Directions. In *Proceedings of the International Workshop on Mutation Analysis (Mutation)*. 47–53.

[46] Christian P Robert and George Casella. 2004. *Monte Carlo Statistical Methods*. Springer, New York, NY.

[47] Greg Rothermel, Mary Jean Harrold, Jeffrey von Ronne, and Christie Hong. 2002. Empirical studies of test-suite reduction. *Software Testing, Verification and Reliability (JSTVR)* 12, 4 (2002), 219–249.

[48] Urko Rueda, René Just, Juan P Galeotti, and Tanja EJ Vos. 2016. Unit testing tool competition: round four. In *Proceedings of the International Workshop on Search-Based Software Testing (SBST)*. 19–28.

[49] Jane Sachar. 1980. Cautions in the Interpretation of the Partial Correlation Coefficient. *The Journal of Experimental Education* 48, 3 (March 1980), 209–216.

[50] Frank L Schmidt and John E Hunter. 2014. *Methods of meta-analysis: Correcting error and bias in research findings*.

[51] Sina Shamshiri, René Just, José M. Rojas, Gordon Fraser, Phil McMinn, and Andrea Arcuri. 2015. Do Automatically Generated Unit Tests Find Real Faults? An Empirical Study of Effectiveness and Challenges. In *Proceedings of the International Conference on Automated Software Engineering (ASE)*. 201–211.

[52] Weichung Joseph Shih and Wei-Min Huang. 1992. Evaluating Correlation with Proper Bounds. *Biometrics* 48, 4 (1992), 1207–1213.
[53] S D Silvey. 1969. Multicollinearity and Imprecise Estimation. *Journal of the Royal Statistical Society: Series B (Methodological)* 31, 3 (Sept. 1969), 539–552.
[54] Fabian Trautsch, Steffen Herbold, and Jens Grabowski. 2020. Are unit and integration test definitions still valid for modern Java projects? An empirical study on open-source projects. *Journal of Systems and Software* 159 (2020), 110421.
[55] Yanbing Yu, James Jones, and Mary Jean Harrold. 2008. An empirical study of the effects of test-suite reduction on fault localization. In *Proceedings of the International Conference on Software Engineering (ICSE)*. IEEE, 201–210.
[56] Yucheng Zhang and Ali Mesbah. 2015. Assertions Are Strongly Correlated with Test Suite Effectiveness. In *Proceedings of the Symposium on the Foundations of Software Engineering (FSE)*. 214–224.