# Paper Title: Dynamic Random Testing of Web Services: A Methodology and Evaluation

## Manuscript ID: TSC-2019-01-0031

Dear Editor-in-Chief,

Thank you for your email on May 8, 2019 regarding our paper titled “Dynamic Random Testing of Web Services: A Methodology and Evaluation” submitted to IEEE Transactions on Services Computing (Manuscript ID: TSC-2019-01-0031).

We are submitting a new version of the paper, in which we have made revisions to address and respond to each comment from the reviewers. Below are the detailed responses to the comments.

We look forward to hearing from you.

Yours sincerely,

Chang-ai Sun, Hepeng Dai, Guan Wang, Dave Towey, Kai-Yuan Cai, and Tsong Yueh Chen

# Response to comments of associate editor and reviewers

In the following, unless otherwise specified, all comments refer to the revised version of the paper.

### Associate editor’s comments

***E1C1: Having analyzed in detail the reviews and the manuscript, I have matured my*** ***recommendation to the EIC, which is that it undergoes a major revision.***

Response: Special thanks to you for your kind recommendation.

Action: None.

***E1C2: I believe the reviewers provided useful comments for you to improve the manuscript. Should you choose to revise your manuscript, pay attention to address all reviewers' concerns, especially those on: the novelty of this paper with respect to own previous publication; the improvements to the writing of various Sections; the repeatability of experiments; the definition of failure rate in the 2nd research question and the revision of the answer to the 3rd research question; more details about the mutants generated; insight as to why CP testing was able to uncover faults; the quality of the reported graphics.***

Response: Thank you for your reminding. You and the other three reviewers’ comments are all of great importance to our paper. All of these comments have contributed a lot to improve the quality of our paper.

Action: None.

### Reviewer 1’s comments

***R1C1: The paper is relevant to the services computing community. However, the specific representations for the SOA used in the paper (WSDL) etc sound a bit dated.***

Response: It is true that web services may sound a bit dated. However, there are still many companies or organizations around the world using web services to provide services such as W3C and Wikipedia (https://en.wikipedia.org/wiki/Web\_service). On the other hand, there are many researchers in the academic fields who are continuing to improve the performance of web services. Since 2019, TSC has published several articles related to web services, such as [1], [2].

[1] S. Wang, Y. Ma, B. Cheng, F. Yang, and R. N. Chang, Multi-Dimensional QoS Prediction for Service Recommendations, *IEEE Transactions on Service Computing,* vol.12, no.1, pp. 47-57, 2019.

[2] P. Wang, X. Du, QoS-Aware Service Selection Using an Incentive Mechanism, vol.12, no.2, pp. 262-275, 2019.

Action: None.

***R1C2: The probability distribution computation model described in section 3.2 is too detailed. Some of the proofs etc and moved into an appendix without taking away from the core message.***

Response: Thanks for the suggestion.

Action: In the revised version, we have moved the related proofs and other demonstration to the appendix without losing the core message.

***R1C3: The technique also requires a user to provide category partition details along with an initial test profile. This could be an unreasonable expectation from a test practitioner.***

Response: In this study, our approach is based on the partitions. We accordingly employed a method called decision table to construct partitions. There are many simple methods (such as equivalence class, category partition method) that are convenient for practitioners to obtain partitions. After constructing partitions, practitioners can use the uniform probability distribution (that is p\_1 = p\_2 = … = p\_m, where m is the number of partitions) as the initial test profile. In our previous paper titled Adaptive Partition Testing, we compared the equal and proportional (p\_i = k\_i / k, where k is the total number of test cases in the test suite and k\_i is number of test cases in the partition c\_i) initial test profile in terms of F, F-2, and T-measure. The results shown that there was no significant different between these two types of initial probability profiles, that means there is no a strong correlation between the probability profile type and DRT’s performance. The reason is that DRT can automatically update the test profile during the test, increase the selection probabilities of partitions with larger failure rates, and decrease the selection probabilities of partitions with smaller failure rates. In a word, practitioners can use any method that is easy to use to construct partitions, and then use the uniform probabilities distribution as the initial test profile.

Action: None.

***R1C4: The applications used in the study are fairly small (~100 SLOC). While the authors say it is not possible to gain access to service request implementations, they do seem to have access to the implementation of the web services used in the experiments in order to create the faulty mutants.***

Response: Sorry for the confusion. It’s true that users have no chance to get the source code. In this study, the applications used were developed by ourselves based on the real-life specifications. On the other hand, web service owners do not make source code of their web services accessible, which also limits our scales of studied applications.

Action: In the revised version, we have added a simple sentence to explain that the studied applications were implemented by ourselves based on the real-life specification in first paragraph of Section 4.2.

***R1C5: Authors should provide more details about the actual mutants generated and provide insight as to why the specification based testing (CP testing) was able to uncover those faults.***

Response: Thanks for the suggestion.

Action: In the revised version, we have followed the reviewer’s suggestion to provide more detailed discussions on the mutants generation, which mainly includes two aspects: (1) We used all provided mutation operators of Mujava to generate mutants for each application, and then randomly generated 50 deferent test suites through using 50 random seeds to execute all generated mutants, then calculated average number of test cases needed to kill each mutant; (2) we deleted those mutants that could be detected with less than 20 test cases, and we also detailed described the features of used mutants that include the specific mutation operators that can generate used mutants and the test cases that can detect the used mutants in the first paragraph of Section 4.2.

Furthermore, we have added detailed discussion on the reason that CP testing is able to reveal our used mutants. CP testing can provide the parameters information of interfaces. Practitioners can generate test cases and construct partitions based on the extractive information. If test cases execute the statements that include the detects, those test cases have probability to reveal the faults.

### Reviewer 2’s comments

***R2C1: The problem addressed in this manuscript is interesting. The proposed solution is appealing because of its simplicity and low applicability effort; moreover it improves significantly the performance of random testing and partition testing (commonly used in web services testing).***

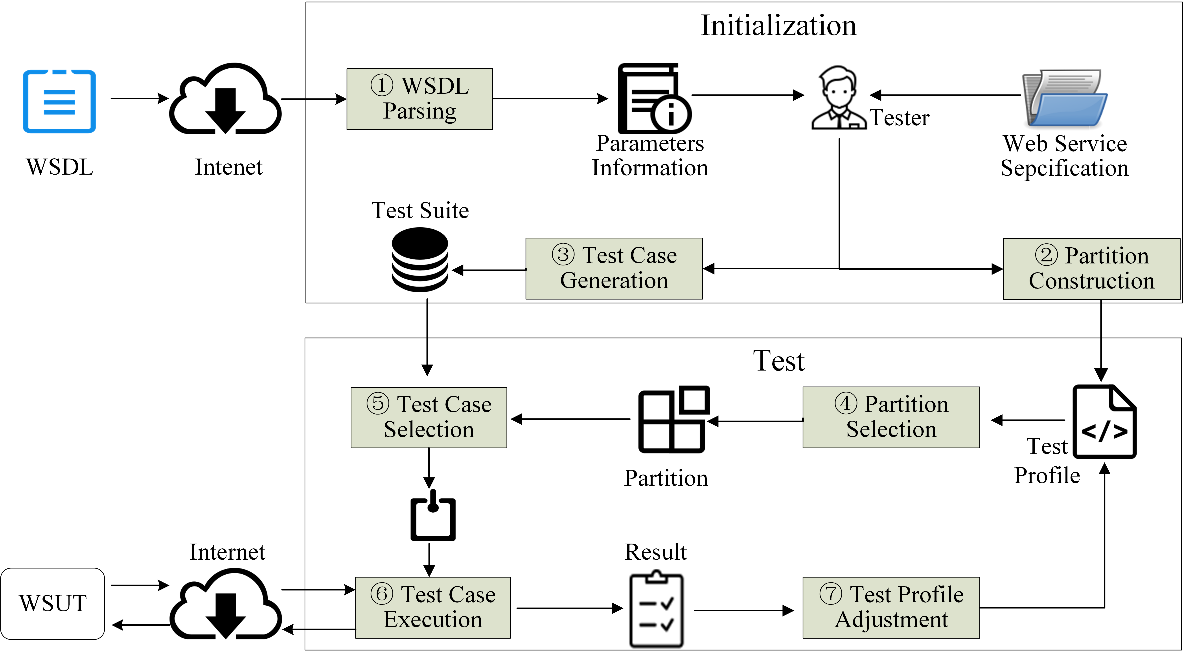
Response: Thank you for the endorsement.

Action: None.

***R2C2: In the model in Figure 1, the human interaction is not represented, although it is important not only for DRT parameters configuration, but also for partitions construction as specified in Section 3.3 “The tool provides two options for the partitions and test suites: either to manually specify the partitions (and test cases); or to upload the predefined partitions and test suites”.***

Response: Thanks for the suggestion.

Action: In the revised version, we have added the human interaction in Figure 1. The human interaction mainly includes partition construction and test cases generation. Accordingly, we have changed the part of description about our framework in the first paragraph of Section 3.1 and added a discussion about the component called “Test Case Generation”.



***R2C3: The first contribution reported by authors in Section 3.1 is “a DRT framework that addresses key issues for testing web services, and a prototype that partly automates the framework”. From the description in section 3.1, the focus is more on positive features than on issues, in fact the WSDL document allows to automate part of the testing process, that is a common practice in web services testing. The prototype description in Section 3.3 is very thin and too similar to the one in the conference paper. Further details are desirable.***

Response: Thanks for the comment.

Action: In revised version, we propose a totally different prototype that is easier to use and more flexible. Accordingly, we have added more details to describe our prototype.

***R2C4: The experimentation is conducted generating mutants of three web services. The authors remove equivalent mutants, and mutants that can be detected with less than 20 randomly generated test cases. This latter criterion is strongly influenced by luck: a mean of the number of test cases generated through multiple repetitions is more robust.***

Response: Sorry for this mistake. In fact, for each mutant we used 50 random seeds to generate different test suites, and calculated the average number of test cases needed to kill the mutant.

Action: In the revised version, we clearly explain the method of selecting used mutants in Section 4.2.

***R2C5: Some defections make the repeatability of experiments impossible, more details are required:***

***- The test profile initialization is not explicitly reported, although the authors, in the Section 4.4.2, indicate that “a feasible method is to use a uniform probability distribution as the initial testing profile. On the other hand, testers may also use past experience to guide a different probability distribution as the initial profile”.***

***- The authors set the partitions by making use of decision table, obtaining two partition schemes for each application. The decision tables used in the experiments for the partitioning are not reported.***

***- The applications are well described, but there is no reference to where they are taken (open source repository, private repository, …).***

Response: (1) If practitioners have no past experience to initialize test profile, they can use the uniform probability distribution (that is p\_1 = p\_2 = … = p\_m, where m is the number of partitions) as the initial test profile. In our other paper titled Adaptive Partition Testing, we compared the equal and proportional (p\_i = k\_i / k, where k is the total number of test cases in the test suite and k\_i is number of test cases in the partition c\_i) initial test profile in terms of F, F-2, and T-measure. The results shown that there was no significant different between these two types of initial probability profiles, that means there is no a strong correlation between the probability profile type and DRT’s performance. The reason is that DRT can automatically update the test profile during the test, increase the selection probabilities of partitions with larger failure rates, and decrease the selection probabilities of partitions with smaller failure rates. In a word, practitioners can use any method that is easy to use to construct partitions, and then use the uniform probabilities distribution as the initial test profile.

Otherwise, practitioners have the past experience, they can set larger selection probabilities to the partitions with higher failure rates and accordingly set a smaller selection probabilities to the partitions with lower failure rates.

(2) Thanks for the suggestion.

(3) Thanks for the suggestion.

Action (1) In the revised version, we have added a more detailed discussion on the method of setting initial test profile.

(2) In the revised version, we have added an example to present how we set the partitions using the decision table in the Section 4.4.1.

(3) In the revised version, we have added a sentence to explain where they are taken in Section 4.2.

***R2C6: In the first Research Question, DRT is evaluated by comparing the effectiveness (in terms of F-, F2- and T-measure) with that of RT and RPT, also if DRT is described in references as an improvement of RT and PT. An additional comparison with a more competitive technique is more*** ***significant to appreciate effectiveness of DRT. Some techniques that improve RT and PT are reported by authors in Section 6.2.***

Response: Thanks for the suggestion. As we description in Section 6.2, related work mainly includes two kind of techniques: adaptive random testing (ART) and adaptive testing (AT). ART is aiming to improve the fault-detecting efficiency of RT. AT takes advantage of RT and PT, but may require a very long execution time in practice, which is a driving force to develop DRT. There are some techniques that improve the performance of DRT depended on deferent intuition. Our method just employed original DRT technique, hence we have followed the reviewer’s suggestion and added AT as one of baselines.

Action: In the revised version, we have added AT as one of baselines.

***R2C7: In the second Research Question (Section 5.2), the definition of failure rate is not clear: it is defined as the ratio between k and ki, where k is the number of test cases until revealing a fault and ki is the total number of test cases in si, that could be infinite. For instance, if a parameter can take all real values between 1 and 10, the number of inputs is countable infinite. This formulation of failure rate is more proper of test case “selection” algorithm. In this case, a solution is to consider ki as the total number of test cases performed to reveal an error and k equal to 1. Failure rate should be defined as #number of failure/#number of executed tests.***

Response: Sorry for this mistake, and thank you for your carefully review.

Action: I am afraid I made a mistake, and I need to recalculate the failure rate and re-execute the test. Besides, the number of executed tests has a affect on the failure rate of a partition. How should I set the number of test cases to be executed in a partition?

***R2C8: The purpose of the Third Research Question is to validate that DRT requires linear time to generate test case through empirical examination of the actual test case generation and execution. Instead, in Section 5.3 the focus is on the comparison among the three techniques, without any reference to the temporal complexity. For this reason, the answer of the Research Question must be revised.***

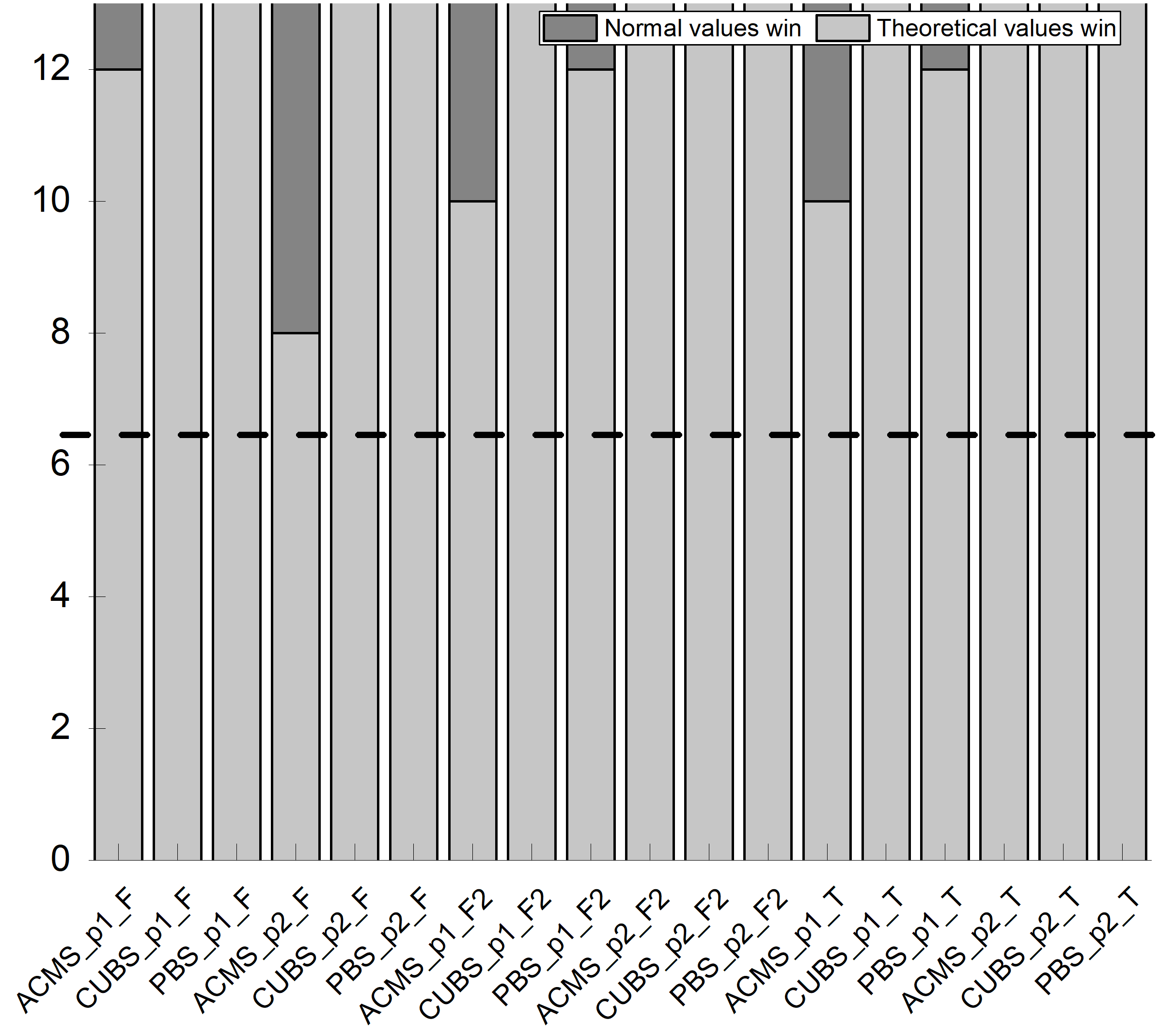
Response: Thanks for the suggestion. RT and PT have a linear time to generate test cases, and DRT theoretically has a linear time to generate test cases. In the most cases, the results of experiment show that DRT need less time in terms of F, F2, and T-measure compared to RT and PT by executing less test cases.

Action: In the revised version, we have added a paragraph to discuss the time complexity of DRT and its actual performance in Section 5.3.

***R2C9: In some cases, the results graphics could be scaled better, because the difference between the various elements is not very appreciable in the printing of paper.***

Response: Thanks for the suggestion. Figure 6 includes many elements, and some points are overlapping with those elements, which may make it difficult to distinguish in some cases.

Action: In the revised version, we have abandoned line chart and employed stacking bar chart to present the significant difference between DRT with theoretical parameters and DRT with normal parameters (i.e. non-theoretical parameters). Moreover, we checked all figures to make sure they are the suitable size.



***R2C10: The article is well written and it is easy to read. An incorrect tables layout is evident: Table 5 is printed after Table 2 and before Tables 3, 4 and 6.***

Response: Thanks for the careful review.

Action: In the revised version, we have made a revision as suggested and also checked all the tables carefully.

### Reviewer 3’s comments

***R3C1: The paper investigates the quality of service-based applications and proposes a dynamic random testing (DRT) technique for web services that improves existing methods. The authors present the technique, a framework for its usage and a prototype implementation. An empirical study including three different case studies is also included to show the effectiveness of the proposed approach.***

Response: Thanks for the endorsement.

Action: None.

***R3C2: The paper deals with an interesting topic and the proposed method extends an idea already proposed in [1]. I have a few comments about the current version of the paper. The authors should explicitly state in the Introduction the novelty of this paper with respect to their previous publication. The method and the prototype seem to be already present in [1] together with some empirical studies. Besides the original idea has already been presented in [7]. Even if a list is provided in the cover letter, a clear statement about the differences is necessary here.***

Response: Thanks for the suggestion.

Action: In the revised version, we have explicitly stated the deference between this study and previous in Section 1.

***R3C3: The cover letter describes a set of improvements concerning the writing (points (i), (ii) and (vi)) and some other major extensions concerning the presentation of the framework, the definition of guidelines about parameters settings in DRT and a more thorough empirical evaluation. From my point of view, the most consistent improvement is the evaluation part. The other ones need more clarification, as described below***

Response: Thanks for the kind advice.

Action: In the revised version, we have followed your suggestion to revise our paper.

***R3C4: Section 3 describes the application of DRT to web services. The novel part described in Section 3.2 needs some rewriting. I understand the importance of parameters setting and the need of mathematical treatment, but as it is now it is not easily understandable. A high-level description of the procedure and of the findings is necessary, together with a description of the followed procedure. The mathematical demonstrations and theorem should be moved to an Appendix. At present these details disrupt the reading flow and at the end of Section 3.2, it is not clear how to practically set the parameters.***

Response: Thanks for the suggestion.

Action: In the revised version, we have moved the proofs and other demonstration to the Appendix, and added detailed descriptions on the usage of our guidelines for setting parameter.

***R3C5: Section 3.3 describing the prototype should be expanded. Now it looks similar to the description in [1], while I would expect a more detailed description here together with the possibility to experiment with the tool for the sake of replicability.***

Response: Thanks for the suggestion.

Action: In the revised version, we have presented a new prototype.

***R3C6: Section 4 reports the empirical studies conducted to evaluate the performance of the proposed method. A set of research questions are described and then answers in the results section. The authors may consider the possibility to*** ***anticipate the research questions as a way to motivate the paper in the Introduction or in a section describing the research approach followed in this paper.***

Response: Thanks for the suggestion.

Action: In the revised version, we have followed the reviewer’s suggestion and anticipated the research questions in Section 1.

***R3C7: The empirical study itself is quite interesting. I would suggest adding a subsection summarising the results and possible limitations discovered during the experimentation. Besides, since the three different case studies have more or less the same dimensions in terms of LOC, it would be nice understanding the scalability of the proposed approach.***

Response: Thanks for the suggestion.

Action: In the revised version, we have added a new section to provide a summary of the results and the possible limitations.