Corrections to *Predicting SMT solver performance for software verification* based on examiners' comments

Andrew Healy

January 21, 2017

0 General comments

0.1 While the work focuses largely on machine learning, this is not reflected well in the overall research hypotheses, which feels both too general and difficult to evaluate (e.g. efficient with respect to what?). The overall hypothesis should be changed to better reflect the machine learning focus.

Re-worded the thesis statement to mention the use of machine learning. The new version is also more specific and verifiable: the vague term *"efficient"* has been removed and a comparison is made to *"any single solver"*.

0.2 The use of a pre-solver makes it hard to judge how much contribution the actual machine learning makes. A discussion on the impact of the pre-solver should be provided.

0.3 Further commentary on the selection and extraction of features is needed. Motivation for the approach taken should be clearly outlined and justified. Examples of alternative features that could have been used should be provided and the reasons for exclusion should be described.

0.4 Small numbers should be spelt in words. If the number is at the start of a sentence then spell as a word. Fix overhanging words throughout.

Fixed throughout.

1 Chapter 1

1.1 Sec. 1.1.3: last paragraph: rephrase (2$^{\text{nd}}$ sentence needs to be broken up).

Rephrased sentence beginning *"Using a specific driver file,..."* into four shorter sentences.

1.2 Sec. 1.2, I am not convinced by the hypothesis, as it is hard to measure. Can you say beyond a single prover or random prover maybe? I would also bring in machine learning as this is in my part the main contribution of this work.

Included in Comment 0.1 of this document.

1.3 Sec. 1.2, para 2: A PO my → A PO may

Typo fixed.

2 Chapter 2

2.1 Sec. 2.1, Remove we before stated (second last line)

Removed clause. Sentence now begins: *"It is Why3's driver-based..."*.

2.2 Sec. 2.1, Q1 (First sentence): starting with "integrating..": rephrase as sentence makes no sense. Fix overhang. Change "Why3 is clearly", to "Why3 is arguably"

Broke first sentence into two sentences. Beginning with the clause *"For users of SV tools,.."*.

2.3 Sec. 2.1.1, You could also mention TIP: Tons of Inductive Problems (which is used within the automated induction community). There was a benchmark paper at CICM 2015 that you may want to cite.

A discussion of TIP has been added to the end of Sec. 2.2.1. The paragraph beginning *"More recently, ..."*. A corresponding entry has been added to Table 2.1 and a note has been made in 3.1.1 in the paragraph beginning *"The TIP benchmarks..."*.

2.4 Sec. 2.1.1, change “a SMT-LIB” to “an SMT-LIB”. Insert “has” before “a wider scope”.

Fixed.

2.5 Sec. 2.1.1, You need to explain “function point” and not just cite it. It gets a bit hard to read as a standalone document if you only give citations.

Added explanatory sentence beginning *“The method of sizing software...”* and ISO citation.

2.6 Sec. 2.1.1. (first paragraph P13): You can also write specification in proof script: the procedural tactic applications are just part of it, so this needs rephrasing. Remove the word “being” line 2.

Rephrased as two sentences beginning *“Formal specifications give no...”* to account for specifications in proof scripts and to clarify the difference between the two approaches to verification. The word “being” removed.

2.7 Sec. 2.2: It wasn’t clear why you are discussing software metrics to this details, so a bit more motivation would be beneficial

Re-worded the sentence beginning *“This section gives...”* into two sentences which give more context to the following discussion of software metrics.

2.8 Sec. 2.2.1 P14: change “journals” to “journal”.

Fixed.

2.9 Sec. 2.3: Although not as close to your work as others, I believe that a literature review should be broad and shallow, so I would also expect other approaches to learning. One example is tactic learning, e.g. (i) Automatic Learning of Proof Methods in Proof Planning, (ii) Hazel Duncan's PhD thesis (Edinburgh Uni), (iii) Typed meta-interpretive learning for proof strategies (which I was involved in). You could also mention the Aris project in Maynooth, which uses case-based learning.

2.10 P15: fix overhang

Included in Comment 0.4 of this document.

2.11 Sec. 2.3.2: Here you discuss/compare concepts that haven't been introduced, with forward references to the place they are introduced (e.g. section 4.2.5). I'd suggest that you delay comparison to when you actually introduce the concept. Another example is that you talk about k-nearest neighbour without having introduced it. Use lowercase k for kNN.

Lowercase "k" used for "k-NN" throughout.

2.12 Sec. 2.4 P19: Rephrase second last sentence.

3 Chapter 3

3.1 P20: Sentence 1 needs closure or remove it and make the point as part of the next sentence.

3.2 Sec. 3.1.1 P22: I didn't understand why you don't see POs from Atelier-B to be software verification related? It is after all used to develop software. Do you only consider code-level verification? Please clarify/expand

3.3 Sec. 3.1.2: Correct spelling of university.

Typo fixed.

3.4 Sec. 3.1.2 P23: Insert a dash between non commercial.

Fixed.

3.5 Sec. 3.2 P24: I didn't understand why you used POs and lemmas. Why lemmas? Won't a PO be generated from it? It may just be a matter of clarifying terminology.

3.6 Sec. 3.2: Change "predicated" to "predicates"

Typo fixed.

3.7 Sec. 3.2.1 P25: In my experience feature selection is the key for successful learning so I was hoping for a bit more details of why you chose these features. Did you for example consider semantic properties (such as associativity of operators, inductively defined types, ...)?

3.8 Sec. 3.2.1: P25: You need to explain how to calculate McCabe's cyclomatic complexity, or at least reference the section where this is explained.

3.9 Sec. 3.2.1: P25: Note that a lemma asserts and does not check.

Terminology changed: *"...lemma checks that..."* → *"...lemma asserts that..."*.

3.10 Sec. 3.3.1 P27: Explain random error and student's t-distribution. How reasonable is the assumption? What are the implications?

3.11 Sec. 3.3.2 P30: Fix overhang.

Included in Comment 0.4 of this document.

3.12 Sec. 3.3.2 P31: How did you find the Peter's principle point for each prover? Are these numbers taken from literature or did you compute them (and in that case how)?

This is further explained by the sentence beginning *"To calculate this time.."*. See also lines 65-66 of `fig_3_7_linegraph.py`.

4 Chapter 4

4.1 Sec. 4.1 P33: Explain the relationship between file and theory; I guess a file can have multiple theories (e.g like a structure in ML) and not that a theory can cut across multiple file (e.g. like a module in C#)?

4.2 Sec. 4.1 P35, 4th para: I would suggest that you delay the comparison with 'best ranking' until you have defined it (or at least state what it is at the first point you mention it)

4.3 Sec. 4.1 P36: Hard file → hard files.

Typo fixed.

4.4 Sec. 4.1.1 P37 first line: (briefly) explain the scoring structure.

4.5 Sec. 4.2.5 P38-39: did you experiment with different cost functions? If so could you justify why this was chosen.

4.6 Sec. 4.3: Why did you use 4 folds? Did you stratify each fold?

The justification for 4 folds is given in the sentence beginning *"The number of folds used.."*. A version of stratification for these folds (modified due to the regression task – stratification is generally more common in classification tasks) has been added. The effect was not dramatic (Table 4.4 has been modified to record the new results) – k-NN algorithms improved as did weighted and discretised variants of most algorithms.

More details in `https://github.com/ahealy19/F-IDE-2016/commit/4b549bc62c2b6b67ef83198ed3879ddad1c9faba`.

4.7 Sec. 4.3.1 P40 (SVMs): Explain what you mean by the analogical family (or just omit it).

Deleted reference to analogical family when discussing SVMs, but kept the reference when comparing k-NN. The concept of an analogical family of learning algorithms is discussed in more detail here.

4.8 Sec. 4.3.1 P41 (Decision Trees): Put reference [90] at the end of the sentence. Explain choice of five for the number of training instances.

Fixed reference placement. Justification for the "choice of five" is given in the final two sentences added to the discussion of the Decision Tree algorithm. Beginning *"This relatively small number..."*.

4.9 Sec. 4.3.1. P42 (Random Forests): Explain choice of 100 trees.

4.10 Sec. 4.3.1 P42 (kNN) Did you normalise the features for kNN? If a feature has a big scale compared to another but this is not meaningful then you need to normalise when measuring distance.

Scaling of features had previously been implemented for SVMs only. I added the preprocessing step for k-NN also (see line 55 of `table_4_4_compare_regressors.py`). The new results have been documented in Table 4.4. Obviously, the main difference is seen in the k-NN results (which have improved) but the overall best result for each metric is unchanged. This preprocessing step has been documented in the introduction to the K-nn algorithm with the two sentences beginning *"In both the k-NN and SVM algorithms,.."*.

More details in `https://github.com/ahealy19/F-IDE-2016/commit/-f13ace247b2296a22c815c3c33a7791cf36c5a40`.

4.11 Sec. 4.4 P43 (last line). You may need a total order for reasons that are not clear to me, but Valid > Invalid seems unnecessary, as it should never happen: if one prover says invalid and one says valid then there is a soundness bug in a prover, so you really need to raise an exception.

4.12 Sec. 4.4.2: I found it hard to understand why Worst Ranking is interesting, and felt its inclusion requires more motivation.

4.13 Sec. 4.4.2 P44: will called → will be called.

Typo fixed.

4.14 Sec. 4.4.3 P45: I cannot understand how the random ranking works form this description. Is it deterministic? Does it try provers in a (pseudo-)random order? Some more details (or rephrasing) of the text is needed to clarify this.

4.15 Sec. 4.4.4: Define portfolio contributions how does this relate to marginal contribution which you define?

4.16 P45 (line 9): text is outside margin. Fix overhang.

Included in Comment 0.4 of this document.

4.17 Sec. 4.4.4 P46: Why do you multiply with 100/1 and not just 100? Fix overhang.

Changed 100/1 → 100. Fixed overhang.

4.18 Sec. 4.5.1: What if Valid/Invalid are not included in the set? Do you just ignore the time? Does that imply it is better to fail early than give a results (but require a long time)?

4.19 Sec. 4.5.2: It wasn't clear how R^2 was computed. Could you provide the equation?

4.20 Sec. 4.5.3 P49: $B > A > C$ should be $A > B > C$.

Typo fixed.

4.21 Sec. 4.5.3: I don't understand what nDCG tells me. Could you give a high-level description of what it really means to have a nDCG of say 0.5?

4.22 P50: Change “may be treated” to “should be treated”.

”may” → ”should”.

4.23 Sec. 4.5.6: What k value did you use and how was it selected?

4.24 P51, table 4.5: How did you compute the relative importance shown in the table? Please explain (or give the reference to where it is explained if you have explained it elsewhere [albeit I couldn't find it])

4.25 Fix overhang.

Included in Comment 0.4 of this document.

5 Chapter 5

5.1 P53: > is outside margin.

Included in Comment 0.4 of this document.

5.2 P53/54: I have a high-level (research methodology related) issue with the use of pre-solving, and this depends a bit on the overall aim/hypothesis of this work. As far as I am concerned, the main contribution of this work is the machine learning aspect. This is also reflected in the title, however the hypothesis is more general and only talk about portfolio solving. Now, the use of pre-solving from an engineering point of view makes sense to me, and would also be okay from the stated hypothesis. However, I feel the main contribution of this work is the machine learning aspect, and by using a pre-solver the analysis gets a bit distorted as it is not clear how much machine learning helps and how much pre-solving helps. I would have liked to see some analysis without the pre-solving to understand how much the machine learning by itself helps and how much of the results are due to the fast pre-solver. I don't expect you to redo experiments, but I would like to see this acknowledged and discussed at some point.

5.3 P56/57: Fix overhang

Included in Comment 0.4 of this document.

6 Chapter 6

6.1 Sec. 6.1 P61: Remind the reader what the best, worse etc. strategies does. Looking at table 6.1 I would expect that best is always best, but other seems to be faster for the theory/goal attribute. Could you expand on this?

6.2 Sec. 6.1 P61, para 4: it fair → it is fair.

Typo fixed.

6.3 Figure 6.1: what was the timeout limit?

6.4 Sec. 6.6.1 P63: You should be careful about changing algorithms during evaluation. It looks like you have used the test-data to tweak the algorithm and drawn a conclusion from it. Instead you should have used the development-data to find the optimal threshold and use the test-data to validate this in order to draw any conclusion. This needs to be discussed; if I have misinterpreted this then you need to update the text to clarify the approach you took. It would be useful to see if the differences are statistically significant?

6.5 Sec. 6.1.1 P65: The answer to EQ1 on defining a cut-off point is nice but does involve further work/cost.

6.6 P68: Resampling techniques for an unbalanced set would be useful here. For example, you can add copies of instances from the under-represented class (known as over-sampling or sampling with replacement), or you can remove instances from the over-represented class (under-sampling).

7 Chapter 7

7.1 Sec. 7.1 P70: What is the `split_goal_wp` transform? (Should it say transformation?)

7.2 Future work: You could also (i) Compare with just applying all in parallel and take the first valid/invalid answer, without any analysis (ii) Compare with and without pre-solving (iii) Compare with use of other features