Maynooth University
National University of Ireland Maynooth

Examiners Report Form – Research Masters

Candidate: Andrew Healy
Student No: 13250280 Qualification Code: MSCR
Thesis Title: *predicting SMT solver performance for software verification*

Detailed list of corrections

General comment: 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.
P2, 1.1.3: last paragraph: rephrase (2nd sentence needs to be broken up).
P4: 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.
P4, 1.2, para –2: A PO my -> A PO may
P7: Remove we before stated (second last line)
P8, Q1 (2nd sentence): starting with "integrating..": rephrase as sentence makes no sense. Fix overhang. Change “Why3 is clearly”, to “Why3 is arguably”
P10-11: 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.
P11: change “a SMT-LIB” to “an SMT-LIB”. Insert “has” before “a wider scope”
P12: 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.
Change i.e to i.e.
P13 (first paragraph): 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.
Section 2.2: It wasn't clear why you are discussing software metrics to this details, so a bit more motivation would be beneficial
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.

- Automatic Learning of Proof Methods in Proof Planning
- Hazel Duncan's PhD thesis (Edinburgh Uni)
- Typed meta-interpretive learning for proof strategies (which I was involved in: http://ceur-ws.org/Vol-1636/)

You could also mention the Aris project in Maynooth, which uses case-based learning.
P14: change “journals” to “journal”
P15: fix overhang
P16, 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
P19: Rephrase second last sentence.
P20: Sentence 1 needs closure or remove it and make the point as part of the next sentence.

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
Correct spelling of university.
P23: Insert a dash between non commercial
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.
Change “predicated” to “predicates”
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, ...)?
P25: You need to explain how to calculate McCabe's cyclomatic complexity, or at least reference the section where this is explained.
P25: Note that a lemma asserts and does not check
P27: Explain random error and student's t-distribution. How reasonable is the assumption? What are the implications?
P30: Fix overhang.
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)?
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#)?
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)
P36: Hard file -> hard files
P37: (briefly) explain the scoring structure
P38-39: did you experiment with different cost functions? If so could you justify why this was chosen. If not then there may be some interesting future work there (see below)
Why did you use 4 folds? Did you stratify each fold?
P40 (SVMs): Explain what you mean by the analogical family (or just omit it)
P41: Put reference [90] at the end of the sentence. Explain choice of five for the number of training instances.
P42: Explain choice of 100 trees. 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.
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.
4.4.2: I found it hard to understand why Worst Ranking is interesting, and felt its inclusion requires more motivation.
P44, 4.4.2: will called -> will be called
P45, 4.4.3: 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.
4.4.3: Define portfolio contributions – how does this relate to marginal contribution which you define?
(line –9): text is outside margin
Fix overhang.
P46: Why do you multiply with 100/1 and not just 100? Fix overhang.
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)?
4.5.2: It wasn't clear how R^2 was computed. Could you provide the equation?

P49: B > A > C should be A > B > C
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?
P50: Change “may be treated” to “should be treated”. What k value did you use and how was it selected?
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])
Fix overhang.
P53: > is outside margin
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.
P56/57: Fix overhang
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?
P61, para 4: it fair -> it is fair
Figure 6.1: what was the timeout limit?
P63, 6.6.1: 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?
P65: The answer to EQ1 on defining a cut-off point is nice but does involve further work/cost.
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).
P70: What is the split_goal_wp transform? (Should it say transformation?)
Future work: You could also

- Compare with just applying all in parallel and take the first valid/invalid answer, without any analysis
- Compare with and without pre-solving
- Compare with use of other features