**Editor Comments #a:**

1. **Comment:** “Both reviewer #1 and #2 suggest improving the early definition of aspects such as the latent parameter. I agree with these comments and recommend you do so.”
2. **Comment:** “While reviewer #1 has some issues with the delta from the previous FSE paper, I believe that NetML is a sensible step forward over AML and it is worthy of publication. As this will likely become the yardstick on spectra based + IR fault localization techniques, it makes sense to strengthen the evaluation further, which is the major change I am recommending”
3. **Comment:** “Reviewer #1 suggested a comparison with *Savant* and I second that. I know how hard it is to get the implementation of other techniques and make sure that you compare apples to apples, but you should have easy access to *Savant*”
4. **Comment:** “Reviewer #1 also mentions focusing on other technics that use dynamic spectrum information, and within reason, I think you should. I agree with the point that these techniques work best on different kinds of data and that type of analysis should be included here.”
5. **Comment:** “More so, I urge you to follow reviewer #2’s and #1’s recommendation and perform a deeper analysis of the results. Detailed analysis of the cases where your techniques achieves best performance and lowest performance is needed. The performance is likely data related, so the paper should highlight under what circumstances (that is, what kind of data) NetML should be the best choice and under what circumstances other technique may be better choices.”
6. **Comment:** “I agree with reviewer #1 that a replication package should be made available for the review and for the public.”
7. **Comment:** “All other comments relate to clarifications and structure improvements – please consider them carefully.”

**Reviewer 1 #1:**

1. **Comment:** “The approach description is not well described. It is *unclear how the latent parameters of bug reports and source-code methods are computed*, and why *they can make the bug localization results better*.  Since the current draft is still rather short, I encourage the authors to take the space to explain the approach in more details, and also include *easy-to-follow examples* to illustrate why it works.”
2. **Comment:** “The new proposed technique seems to have small delta with the prior AML work. It seems the main improvement is that the proposed work now additionally considers latent parameters for source-code methods.”
3. **Comment: “**Although the proposed technique has been compared against various existing baseline bug localization techniques, the used baseline techniques may be strawman, and state-of-the-art techniques have not been used in the comparison. For example, MULTRIC is the only evaluated technique that uses dynamic spectrum information. Actually, although MULTRIC can be effective on some subjects, itself may not always outperform even the existing *traditional spectrum-based techniques*, such as *Ochiai* and *Dstar*. Furthermore, recently, researchers have proposed more advanced fault localization techniques, such as the authors' own *Savant* work [Le2016]. I suggest the paper to include *more comparisons with existing traditional spectrum-based techniques*, and some recent advanced techniques such as *Savant*.”
4. **Comment:** “The evaluation benchmark suite is the same as prior work. This is good, since this enables direct comparison with the prior AML work. However, it is not clear whether the proposed technique works for other subjects, since only 157 bugs from 4 subject systems are used in this study.”
5. **Comment:** “The paper simply mentions that it performs 10-fold cross validation. It is not clear about the details. The paper can be much stronger if it can distinguish *cross-version* and *cross-project  
   predictions*. For example, the proposed latent parameters for source-code methods may be helpful if two similar bugs of the same system exist, and one is used for training while the other is used for prediction. However, *it is not clear how such proposed technique helps in the cases of cross-project prediction.*”
6. **Comment:** “The result analysis is also rather short. It would be better if the paper can list some qualitative examples to illustrate the poor/good performance of the proposed technique.”
7. **Comment: “**It seems the source code and reproduction steps for the newly proposed NetML is not available on the project webpage. I strongly encourage the authors to make them available for the ease of result reproduction.”

**Reviewer 2 #2:**

1. **Comment:** “The main downside of the paper is on the evaluation, which is described as "extensive" although it could only be considered as such from a longitudinal dimension. I elaborate on this and other issues next.”
2. **Comment:** “NetML works at method level. The paper motivates this choice by stating that methods are a good trade-off between retrieving a file and retrieving a line of code, as code files are too big, and lines of code do not provide enough information to understand the bug well enough. This is somewhat unfair, since a line of code is retrieved as a starting point to understand the bug, and assuming that developers have access to the context of that line (e.g., the line of code is retrieved in an IDE and the developer has the chance to go to that line and explore the surrounding code). In hindsight, retrieving a line of code is more precise than retrieving a whole method, if the method is too long.”
3. **Comment:** “The extensive evaluation listed as a contribution of the paper is a bit misleading. The evaluation is extensive in that it compares the performance of NetML to six bug localization techniques. However, it *falls short in the number of bug reports* used to run the techniques”
4. **Comment:** “Also, the evaluation is focused on quantitative analysis, which, yet important, limits the knowledge that can be gained from the study. A qualitative analysis of the results could help to understand when and why the approach works/fails. In this sense, I suggest to include a (manual) analysis of the outliers (i.e., cases where NetML works pretty well, and cases where it does not) and cases where the other techniques worked better than NetML.”
5. **Comment:** “Top N might be useful to compare performance between approaches, but it does not work to evaluate the performance of an approach by itself, if the total of bug reports in each system is not presented (at the same time). I suggest to add this number to Tables 4, 7 and 8, or even better, add the percentage of bug reports whose relevant methods are retrieved in Top 1, Top 5 and Top 10.”
6. **Comment:** “The concepts "latent parameter" and "features" are widely used in the paper but never defined or defined too late.”
7. **Comment:** “The structure of Section 3 (approach) is a bit odd. Since the integrator model depends on the outputs of the feature extraction and graph construction components (and their underlying concepts), this section would flow better if the latter two components are presented before the integrator.”

**Reviewer 3 #3:**

1. **Comment:** “At the point, my only comment is the authors to extend the related work section with recently published works.”