

Paper Review

Submitted to:  
  
Dr. Mahmoud Alfadel,  
Department of Computer Science,  
University of Calgary

Submitted by:

Mohammad Kaosain Akbar  
*Prospective PhD Applicant, Department of Computer Science,  
University of Calgary*

Contents:

1. Initial List of Papers

2. Final Choice of Selecting the Paper

3. Paper summary

4. Strength and Weakness

5. Recommendation

6. Personal Learning Feedback

6. Disclosure and Appendix.

**1. Initial List of Papers**

In order to select reviewing papers for this task, initially the following research articles were shortlisted.

1. BLAZE: Cross-Language and Cross-Project Bug Localization via Dynamic Chunking and Hard Example Learning.
2. Revisiting the Performance of Deep Learning-Based Vulnerability Detection on Realistic Datasets.
3. RLocator: Reinforcement Learning for Bug Localization

These three papers were chosen because they directly apply Machine and Deep Learning techniques to software engineering challenges. The first paper, BLAZE leverages large language models with dynamic chunking and hard example learning to improve cross-language and cross-project bug localization. Second paper, Revisiting the Performance of Deep Learning-Based Vulnerability Detection critically evaluates deep learning models on a proposed realistic dataset, highlighting their limitations and proposing improvements for practical use. Finally, RLocator, the third paper uses reinforcement learning to optimize bug localization and outperforms prior deep learning approaches.

**2. Final Choice of Selecting the Paper**

I selected this paper for review because it connects closely with my background in machine learning. Particularly, this work aligns with one of my prior research paper on Semi-supervised approach for Non-intrusive Load Monitoring (NILM), where my a semi supervised approach consisting of TCN-BiLSTM architecture was developed and the dataset used was slightly labelled and mostly unlabelled. Additionally, my own research deals with applying advanced learning methods to complex, noisy real-world data, and I found it interesting how this paper applies reinforcement learning to software engineering challenges in a similar way.

**3. Paper Summary**

The following is the summary of the paper.

1. The paper introduces RLocator, a reinforcement learning based framework for bug localization. The framework utilizes Markov Decision Process (MDP) and optimizes ranking quality of the bug containing file directly using the A2C algorithm with entropy regularization.
2. Initially, to manage scalability, ElasticSearch (ES) is used to retrieve the top-31 candidate files, after which an XGBoost classifier filters out bug reports to remove further irrelevant file which is termed as buggy by ES.
3. Within the architecture, the proposed model leverages CodeBERT embeddings for bug reports and source code, enhanced with a CNN to capture features and an LSTM to maintain state awareness during ranking.
4. The filtered candidates are then re-ranked by the RL model based on the MRR/MAP scores, which outputs prioritized source files for each bug report.
5. Evaluation on six Apache projects shows that RLocator surpasses baseline methods such as BugLocator, FLIM, and BL-GAN, highlighting its effectiveness in practical bug localization tasks.

**4. Strength and Weakness of the Paper**

***Strengths:***

i. The paper represents novelty in Bug localization approach where reinforcement learning is used combining MDP (having an internal architecture comprising of CNN-LSTM) with A2C and entropy policy.

ii. The paper enhances the initial ElasticSearch (ES) approach with an additional XGBoost filter, which predicts whether the top-31 candidate files which actually contain any relevant buggy files. This step reduces noise and prevents the model from being trained on cases where correct localization would not be possible.

iii. The paper maintained transparency especially showing its superior performance over the 91% of dataset while also informing that for the remaining 9% of the data, RLocator could not localize the bugs.

***Weakness:***

i. At the end of the introduction, the authors admit the difficulty in explaining why reinforcement learning performs better than supervised approaches. This highlights the proposed model’s limited explainability.

ii. Equation (1) introduces a hyperparameter *M*, whose optimal value was tuned empirically to 3. This is value is obtained based on the dataset which was utilized in this paper. So, this value may differ for other datasets, reducing the generalization of the RLocator.

iii. The candidate list is restricted to 31 files, but in realistic scenarios, a bug report may involve more than 31 files.

**5. Recommendation to improve the Paper**

In the following I have provided my personal recommendation for the paper.

i. The paper highlights challenges with “lower quality bug reports.” Future work could deal with developing a more robust reinforcement learning model that can handle noisy or incomplete bug reports.

ii. The current model is limited to 31 candidate files due to GPU/memory limitations. It is important to develop models that can handle larger number of candidate files while also being more efficient in using computational resources.

iii. At present, RLocator depends on developers to manually provide feedback by selecting buggy files. Future work could involving automating this process, reducing the need for human developer to further train the model.

**6. Personal Learning Feedback**

I really enjoyed reading this paper, as it helped me gain a lot of knowledge about both reinforcement learning and its application in bug localization. For the first time, I have seen researchers place the related works section at the end of the article, and still the overall reading flow was smooth. The inclusion of the ablation study clearly showed why such evaluations are essential for research of this kind that involves proposing various novel machine, deep and reinforcement learning models. I also found the inclusion of a Threats to Validity section particularly interesting, as it was the first time I encountered it in this type of paper, and I realized how important it is for ensuring the limitations of research are transparently discussed.

**Appendix.**

Disclosure: No online large language models were used in writing this report.

I have also attached the annotated version of the paper which I utilized to write the report