

Peer Review

Review of: "MVD: A Multi-Lingual Software Vulnerability Detection Framework"

Son Nguyen¹

1. Vietnam National University Hanoi, Viet Nam

This paper tackles a well-motivated and important problem—multi-lingual software vulnerability detection. The idea that vulnerabilities often manifest similarly across different programming languages is compelling, and I appreciate how MVD leverages this insight to improve detection performance. The potential of transferring vulnerability knowledge across languages is particularly exciting, as it could help in scenarios where labeled vulnerability data is scarce for certain languages. I also appreciate the extensive experiments, which demonstrate the performance of MVD compared to existing single-language baselines.

That being said, I have a few concerns and suggestions that I believe could strengthen the paper further:

1. While MVD achieves significant improvements over existing methods, it is trained on a larger, multi-language dataset compared to single-language baselines. This raises a natural question: how much of the improvement comes from MVD's architecture versus the additional training data? To validate the key observation that knowledge transfer across languages enhances performance, I strongly encourage the authors to conduct an experiment where MVD is trained on a multi-language dataset of size X and compared to single-language models trained on datasets of comparable size X . This would provide a clearer picture of the actual gains from multi-lingual learning rather than differences in dataset scale.
2. I noticed that MVD achieves different levels of improvement across languages. It would be valuable to analyze why certain languages benefit more than others. Does it depend on language syntax, dataset size, or the frequency of certain vulnerability patterns? These insights would help practitioners understand when MVD is most beneficial compared to single-language models.

3. One thing that remains unclear to me is how MVD handles language-specific vulnerabilities. If some vulnerabilities are unique to a particular language (e.g., memory safety issues in C/C++ but not in Python), how does MVD effectively detect them? A discussion or experiment addressing this question would strengthen the paper's claims about the model's adaptability.

4. I really like RQ3, which examines how MVD can learn from some languages to detect vulnerabilities in others. To further explore this capability, I suggest investigating a more realistic setting where MVD is trained on some languages plus a fraction of the target language's vulnerabilities, then tested on the remaining vulnerabilities in that target language. This could better reflect real-world deployment scenarios, where we may have partial data for a new language rather than none at all.

5. The ablation study in RQ2 is useful, but to further demonstrate the generality of MVD, I suggest the authors try alternative pretrained models beyond the one currently used. This would validate whether MVD's success is tied to a specific model or if it generalizes well across different pretrained embeddings.

Declarations

Potential competing interests: No potential competing interests to declare.