Testing Neural Program Analyzers

Md Rafiqul Islam Rabin[1], Ke Wang[2], Mohammad Amin Alipour[1]
University of Houston[1], Visa Research[2]

Problem Statement

- Neural program analyzers use (deep) neural networks to analyze programs in software engineering tasks. They take a program and make predictions about some characteristics of the program.

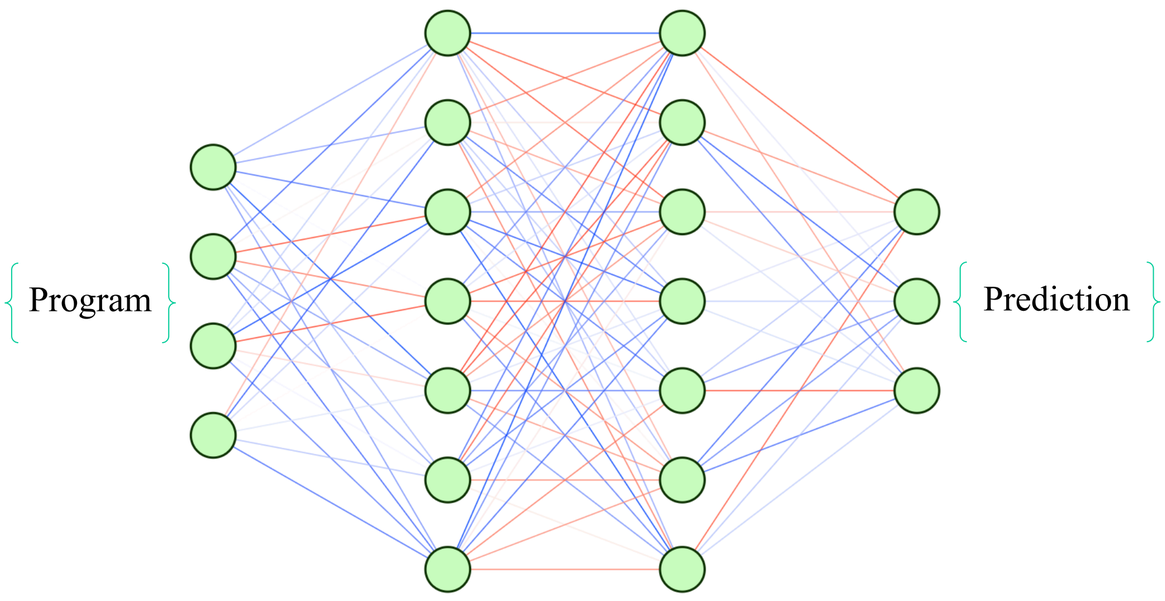

Fig. 1. Neural Program Analyzer.

- Problem Statement: How reliable are neural program analyzers?

Approach

- We propose a transformation-based testing framework to test the robustness of state-of-the-art neural models running on the programming task.

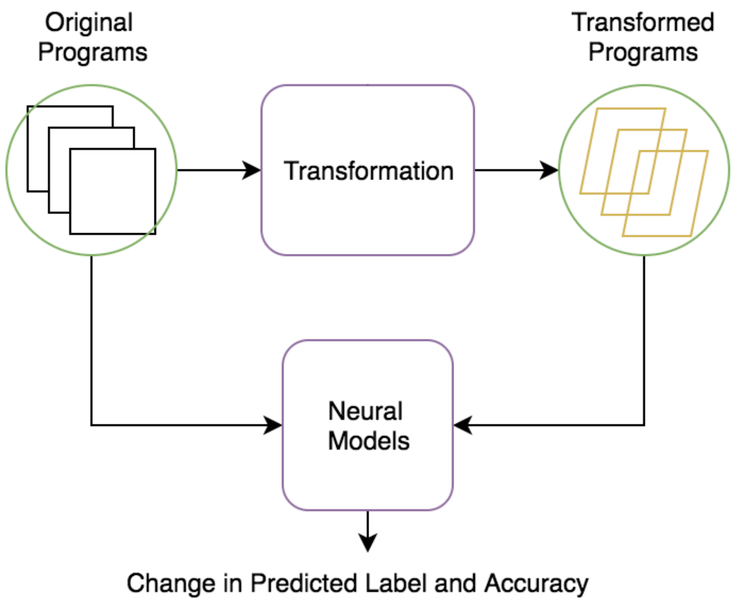

Fig. 3. An overview of our approach for testing the neural models.

Motivating Example

- Both functions check if their argument is a prime number. Their only difference is that one uses a '*for*' loop and the other uses a '*while*' loop.

- The code2vec model [2] predicts one of the methods as "***isPrime***" and the other as "***skip***".

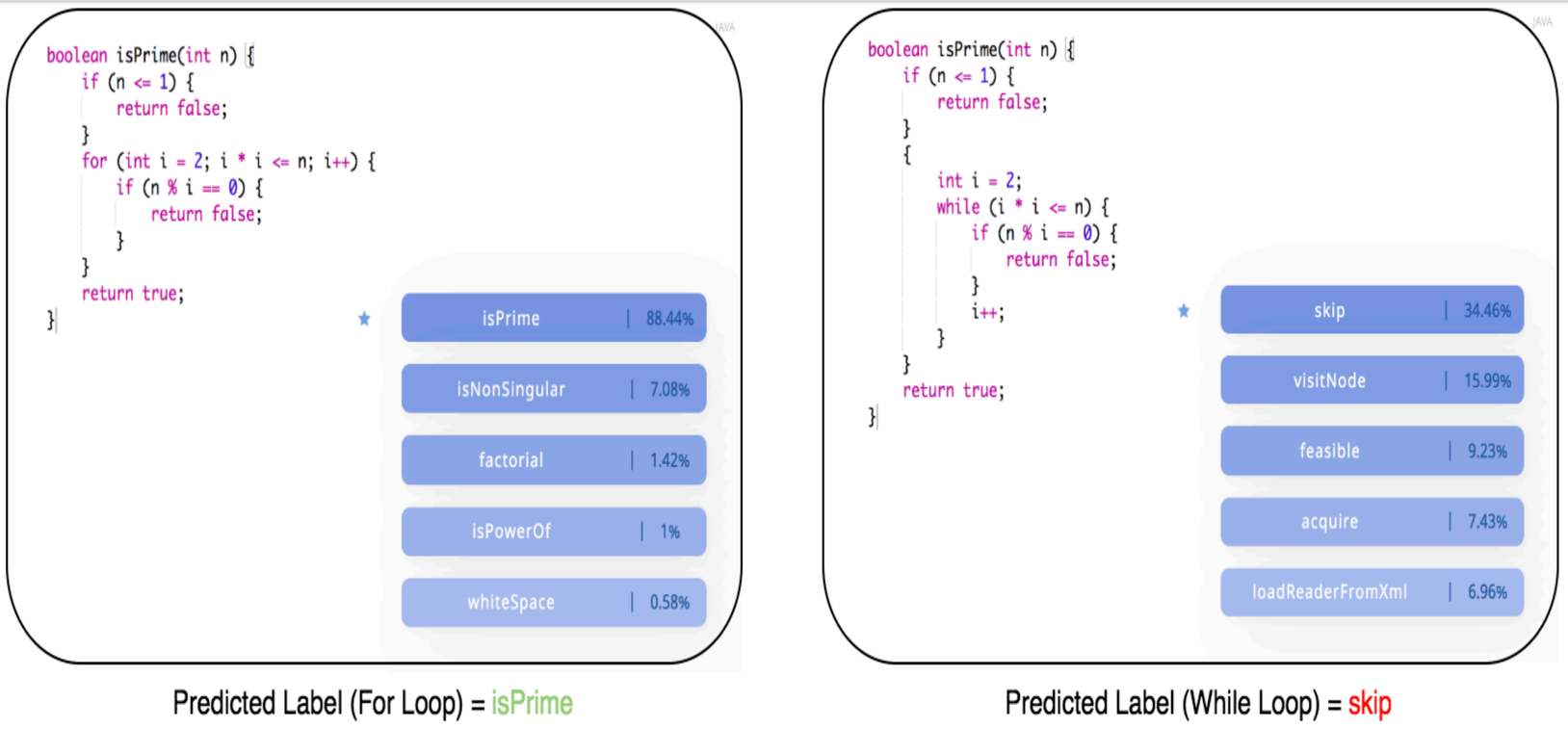

Fig. 2. A failure in code2vec revealed by a transformation.

Semantic-Preserving Transformation

- ***Insight***: The prediction of the neural program analyzer on a program and the semantically equivalent one should be similar.
- ***Approach*:** Perform sematic-preserving transformations on the programs and compare the prediction of the model.

Test Oracle: A neural program analyzer should behave similarly on both the original program and the transformed, yet semantically-equivalent program.

Example of Transformations:

- *Variable-Renaming*
- *Boolean-Swapping*
- *Loop-Exchanging*
- *Switch-to-If*
- *Permute-Statement*
- *Dead-Code Insertion*
- ...

#	Variable Renaming	Loop Exchanging	Boolean Swapping	Switch-to-If	Permute Statement	Dead-Code Insertion	Try-Catch Insertion
Original programs	13026	1441	100	131	4976	19627	19627
Transformed programs	45007	2689	146	154	23284	19627	19627
Misprediction	17087	705	10	61	2710	7685	9225
Misprediction Percentage	37.96%	26.21%	6.84%	39.61%	11.63%	39.15%	47.00%

Table 1. Results of evaluating code2vec neural program analyzer on java-small/validation/libgdx project.

Ongoing Efforts

- We are performing a large-scale study of the robustness of various neural program analyzers.
- We are investigating approaches to automatically identify the places in the program to transform, and the transformations to use.

References

[1] Allamanis et al., "Learning to represent programs with graphs," ICLR 2018.
[2] Alon et al., "code2vec: Learning distributed representations of code," POPL 2019.
[3] Tian et al., "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," ICSE 2018.
[4] Zhang et al., "Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems," ASE 2018.
[5] …