INTERPRETING DEEP LEARNING MODELS
FINE-TUNED FOR DETECTING VULNERABILITIES
RELATED TO MISSING CODE

Claudia Mamede\textsuperscript{1,2}, Claire Le Goues\textsuperscript{2}, José Campos\textsuperscript{1,3}, Rui Abreu\textsuperscript{1,4}
\textsuperscript{1}Faculty of Engineering, University of Porto, Portugal
\textsuperscript{2}Carnegie Mellon University, USA
\textsuperscript{3}LASIGE, Faculty of Sciences, University of Lisbon, Portugal
\textsuperscript{4}INESC-ID, Instituto Superior Técnico, University of Lisbon, Portugal

**MOTIVATION**

BACKGROUND. LEARNING-BASED VULNERABILITY DETECTION IS TYPICALLY TREATED AS A BINARY CLASSIFICATION PROBLEM.

PROBLEM 1. **MODEL'S HIGH ACCURACY DOES NOT ENSURE REAL-WORLD EFFECTIVENESS**

IF the code has > 500 LoC, how can we find and fix the vulnerability?

PROBLEM 2. **EXISTING xAI METHODS ONLY EXPLAIN TOKENS PRESENT IN THE INPUT**

How can we explain vulnerabilities arising from the absence of code?

**METHODOLOGY**

**OUR INTERPRETABILITY STRATEGY USING TAINT ANALYSIS**

VULNERABILITY DETECTION

CONVERT CODE TO

FIND TAINTED VARIABLES (A)

FIND CONTROL STRUCTURES (B)

COMPLEMENTING xAI, WE MAY HELP US LOCATE VULNERABILITIES.

**REPORT EVALUATION USING SECLINT**

**INTERPRETABILITY STANDARD**

We reviewed prior work to identify key properties in security reports that help developers understand and address vulnerabilities.

1. **VULN-DETET**: \textsc{<weakness name/id> at <location> (severity: <level>)}
2. **WHAT**: describe the weakness/problem
3. **WHY**: describe its impact
4. **HOW**: describe how the weakness can be triggered
5. **WHEN**: describe when the problem was found
6. **WHERE**: describe where the problem is located

(?) MANDATORY FIELDS

(?) OPTIONAL FIELDS

Acknowledgments: This work was partially funded by Fundação para a Ciência e a Tecnologia under the scope of the CMU/Portugal Program

Future Work

We are still refining the interpretability standard

Check out the QR code below for the updated version & give us feedback!