ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

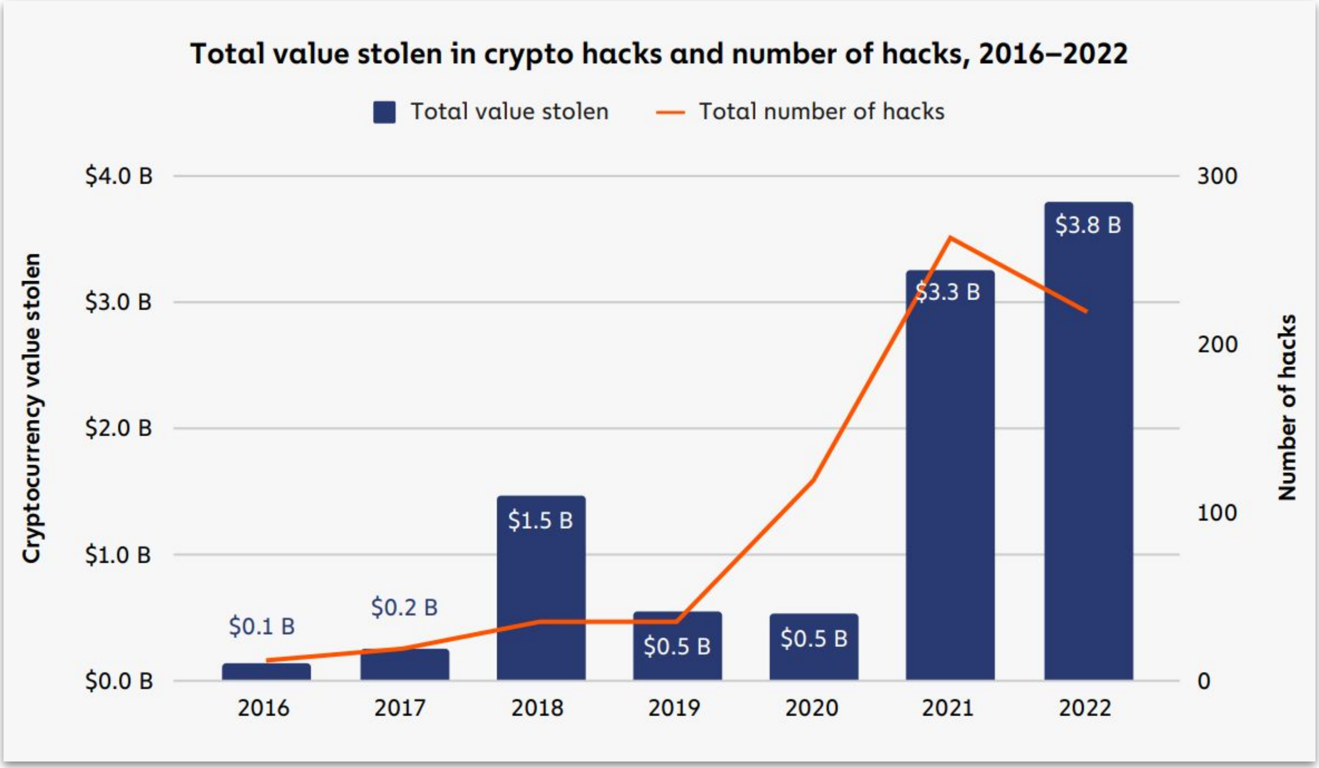
Smart Learning to Find Dumb Contracts

Tamer Abdelaziz and Aquinas Hobor. "Smart Learning to Find Dumb Contracts".
The 32nd USENIX Security Symposium (USENIX Security 23).

NUS National University of Singapore

UCL

Bugs in Cryptocurrency have been Expensive



Source: The Chainalysis 2023 Crypto Crime Report

Existing Smart Contract Vulnerability Detection Frameworks

Static Program Analysis

- \+ Impressive tools
- \- Difficult to maintain and extend
- \- Require expert rules
- \- Mostly need access to the source code
- \- Slow for large-scale analysis

Can deep learning help address these challenges?

DLVA: Deep Learning Vulnerability Analyzer

★ Detects 29 vulnerabilities in Ethereum smart contracts…

… without any predefined patterns or expert rules

★ Analyzes EVM bytecode (no source required)

★ High accuracy…

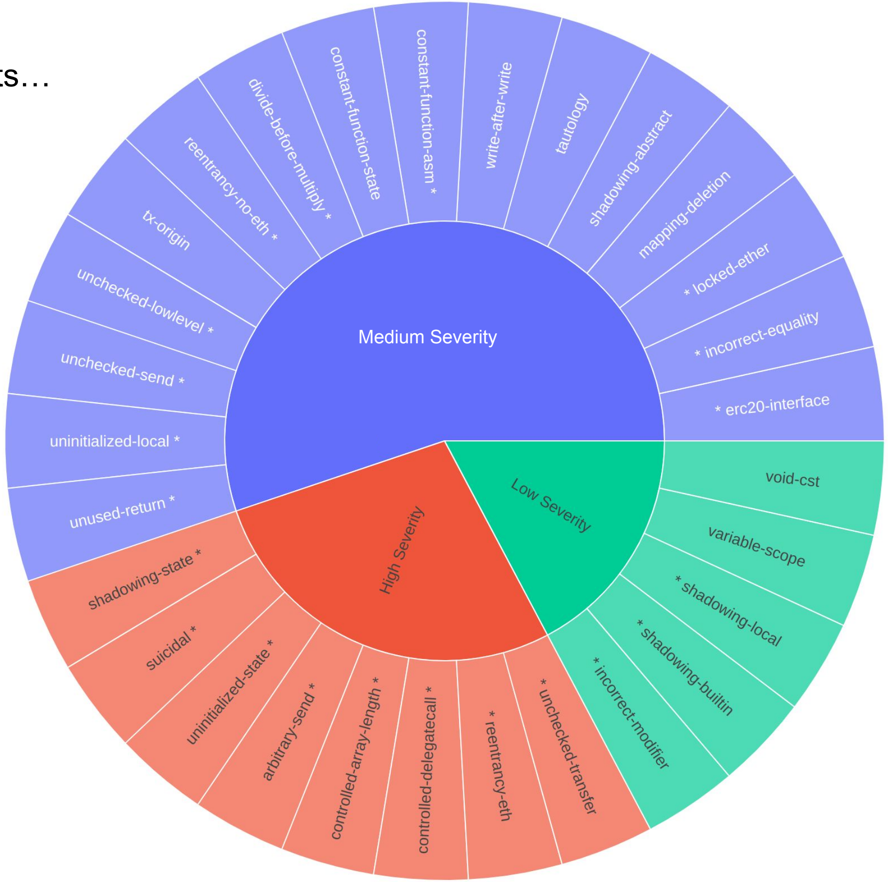
… and low false positive rate

★ Almost always answers…

… very quickly (10x-1,000x faster than competitors)

★ Extensively benchmarked

★ Available for immediate download and use



Reentrancy Attack Example

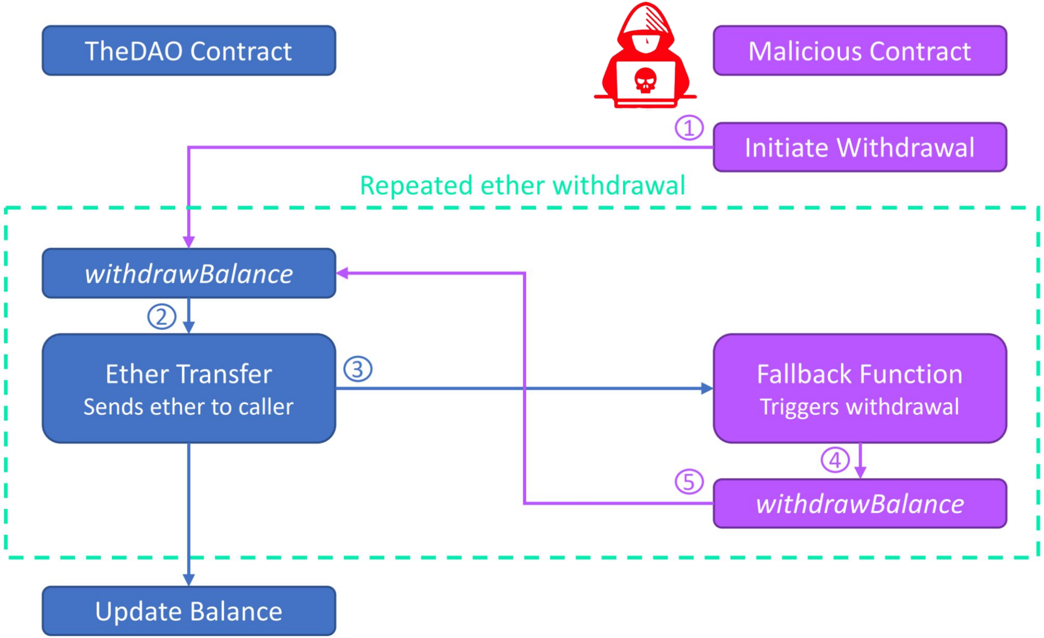
```
contract TheDAO {

mapping(address => uint) balances;

function deposit() public payable {
  uint amount = balances[msg.sender];
  balances[msg.sender] = amount + msg.value;
  }

function withdrawBalance() public {
  if (msg.sender.call.value(balances[msg.sender])())
     { balances[msg.sender] = 0; }
  }

}
```



In TheDAO hack, the attacker stole over 3.6 million Ether!

Next: Designing DLVA

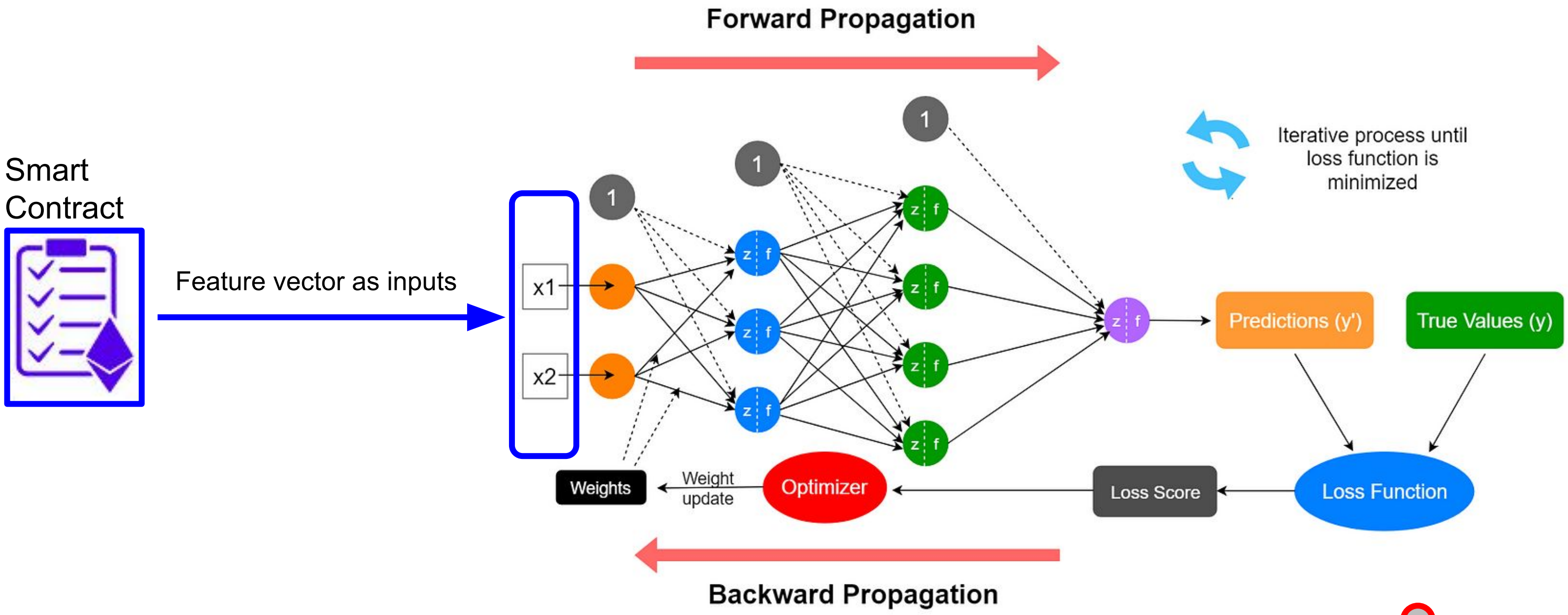
1. Motivation and Introduction ✅

2. Designing DLVA

3. Training DLVA

4. Benchmarking DLVA

Neural Network’s Training Process

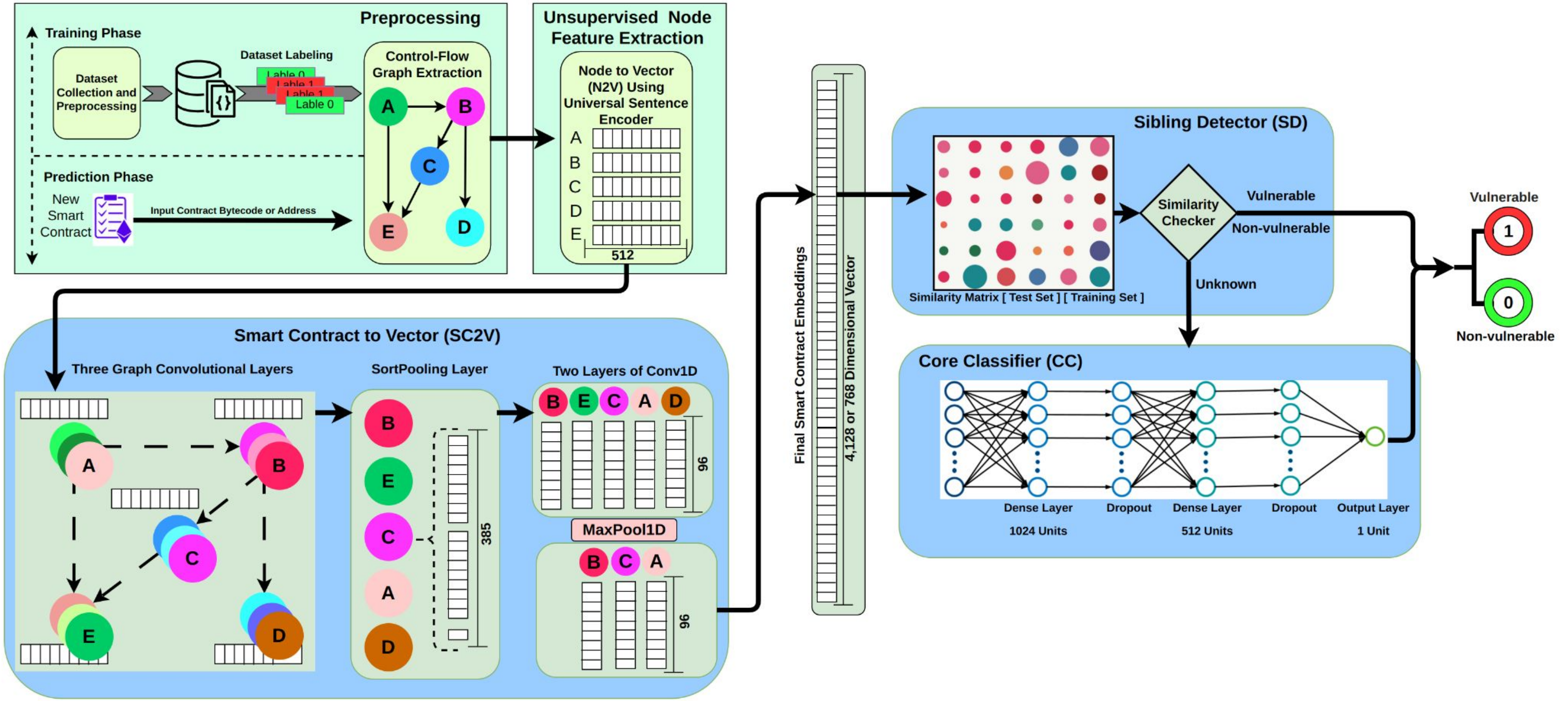


Main Challenge:
Representing smart contracts as feature vectors suitable for NN inputs

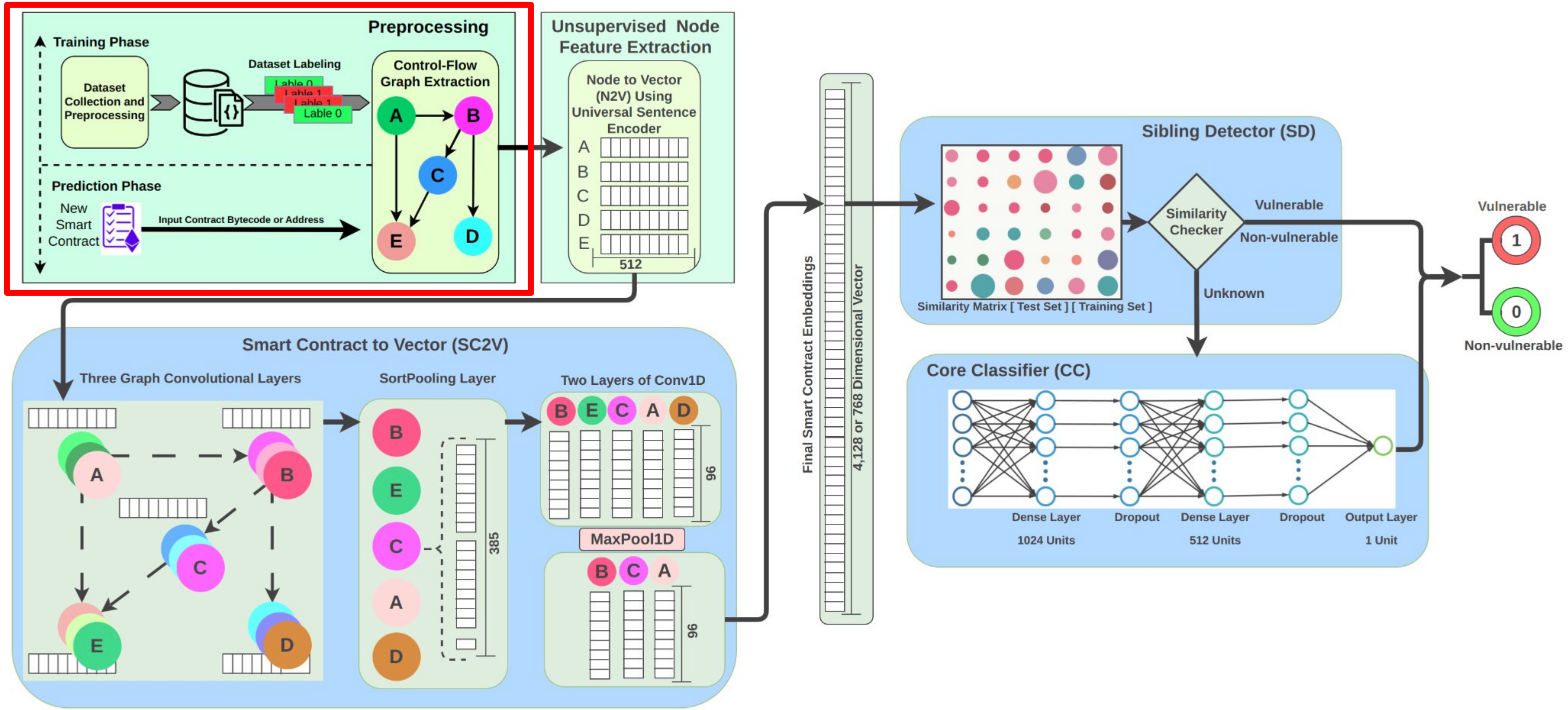
Our Solution: **DLVA’s Smart Contract to Vector (SC2V) engine**

Architecture of DLVA

- 🔑: Neural networks learn from feature vectors to classify contracts
- 🔑: Use graph convolution NN to extract semantic structure

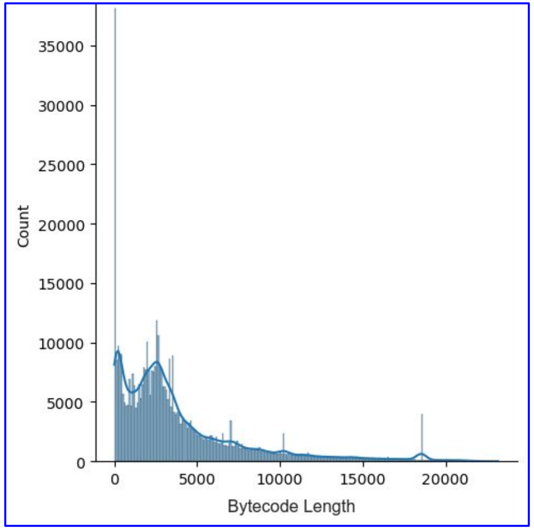


Preprocessing



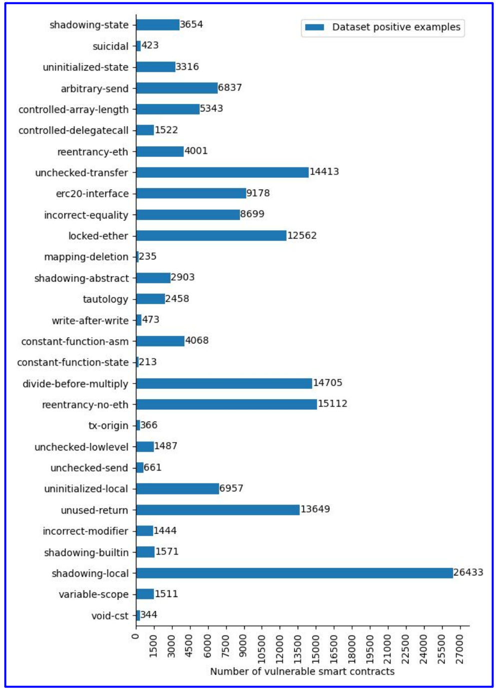
Preprocessing

1. Dataset Collection



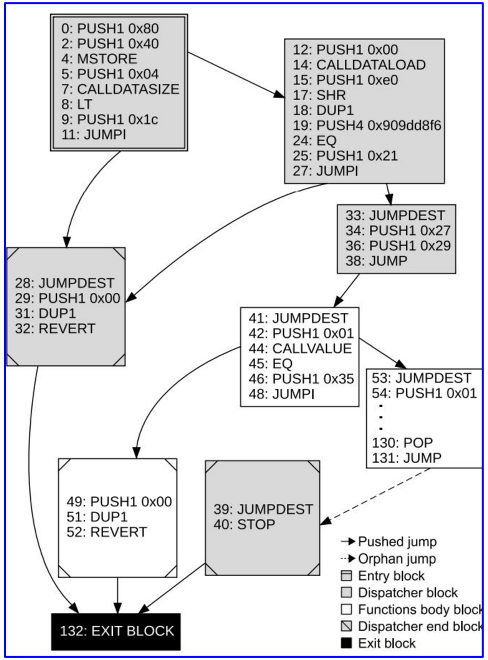
- **51,913,308** real-world contracts
- Remove redundant **99.3%**
- Only **368,438** are unique contracts

2. Dataset Labeling



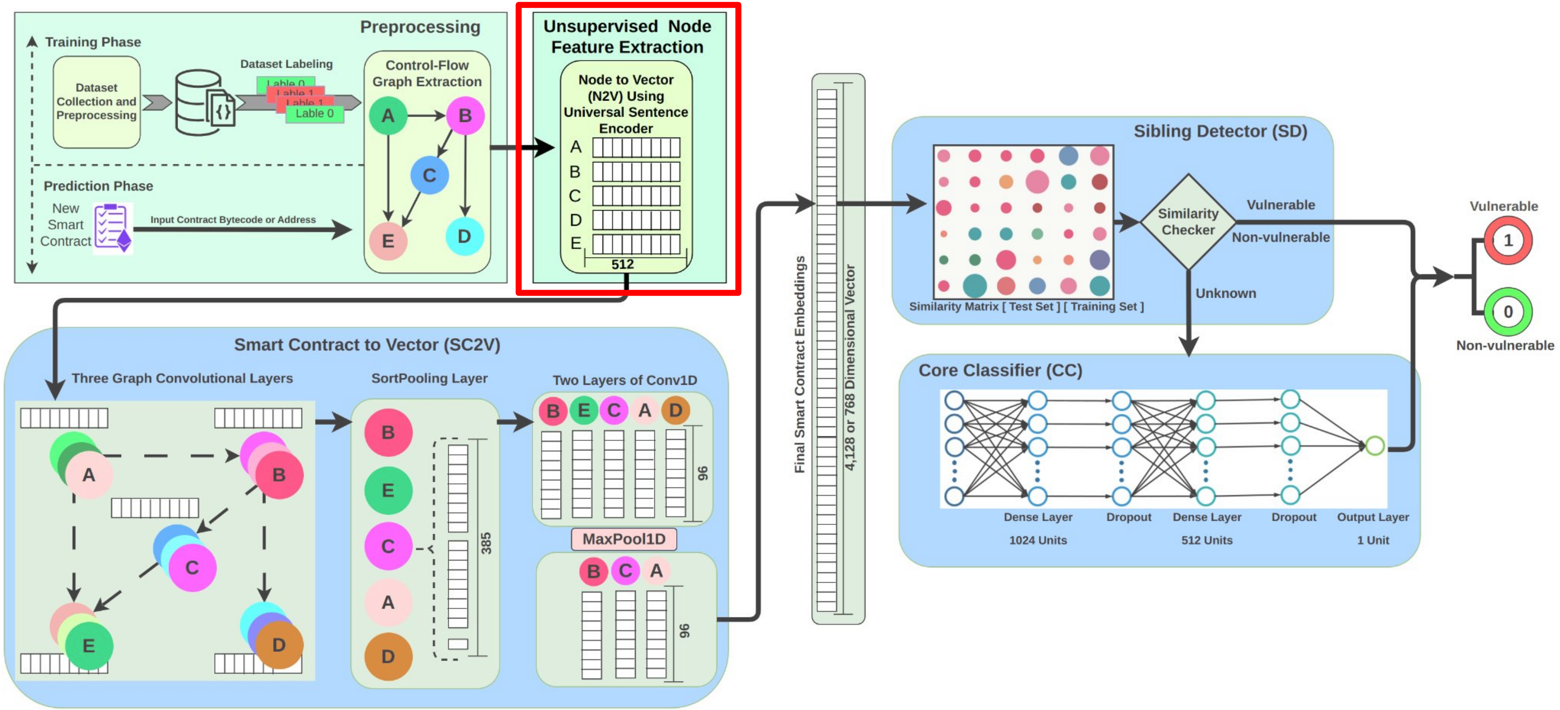
- Slither requires source code available
- Only ⅓ of unique contracts are labeled

3. CFG extraction



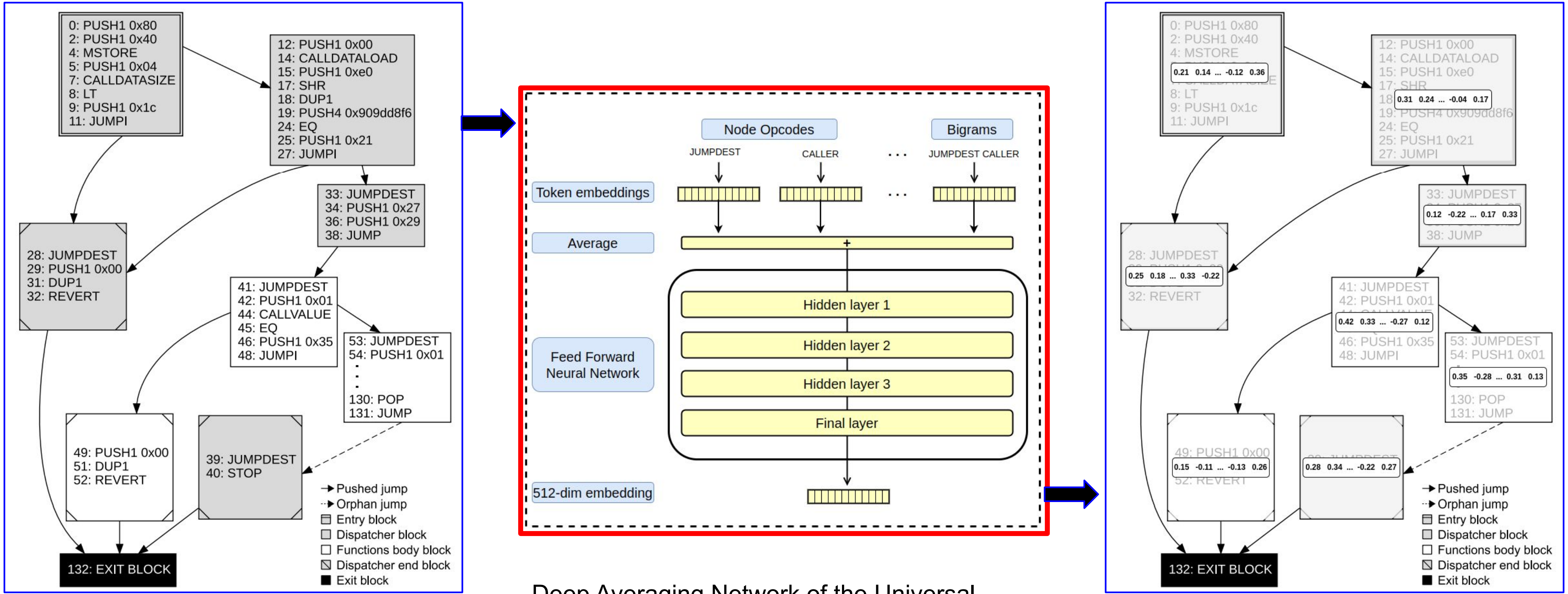
- CFG captures important semantics structures

Node to Vector (N2V)



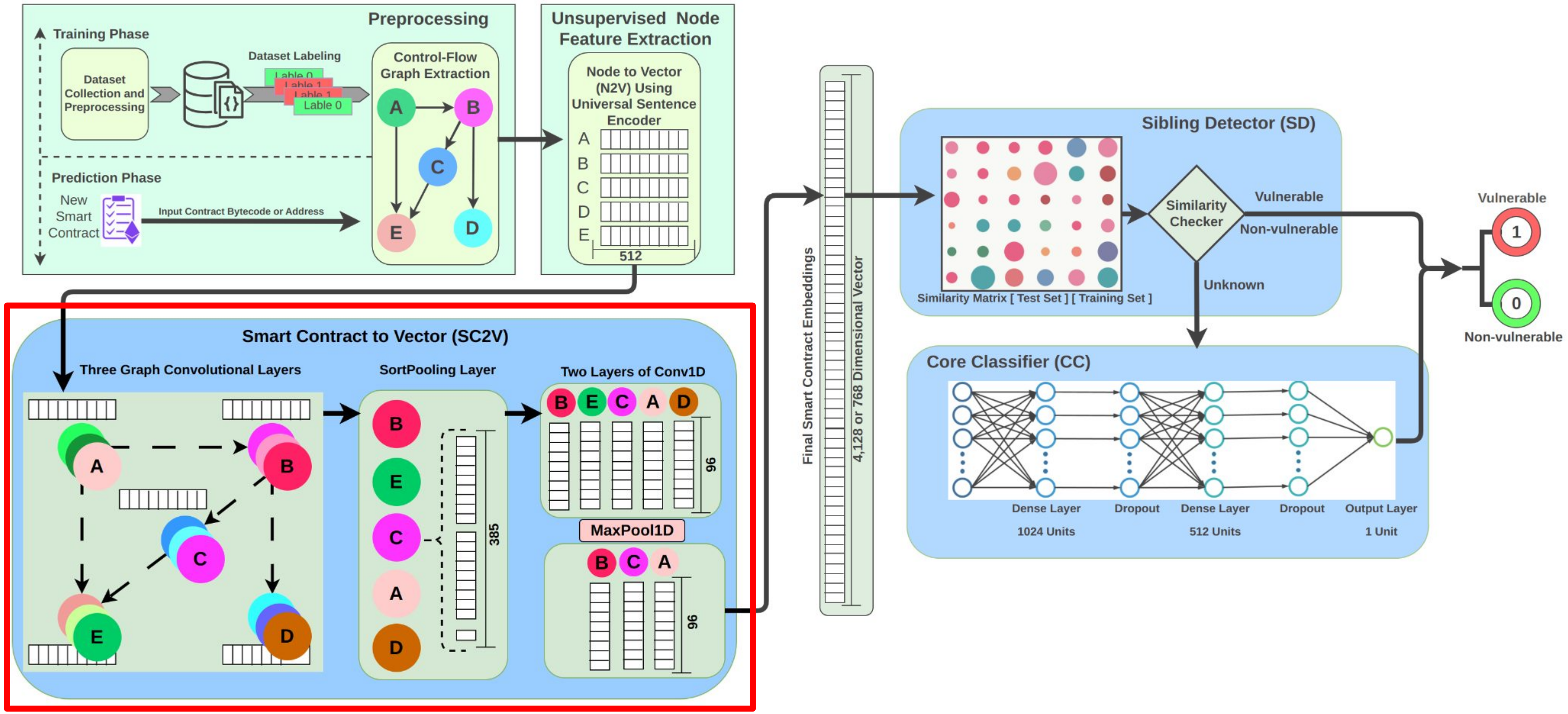
Node to Vector (N2V)

🔑: Similar basic blocks/nodes → Similar vectors in embedding space



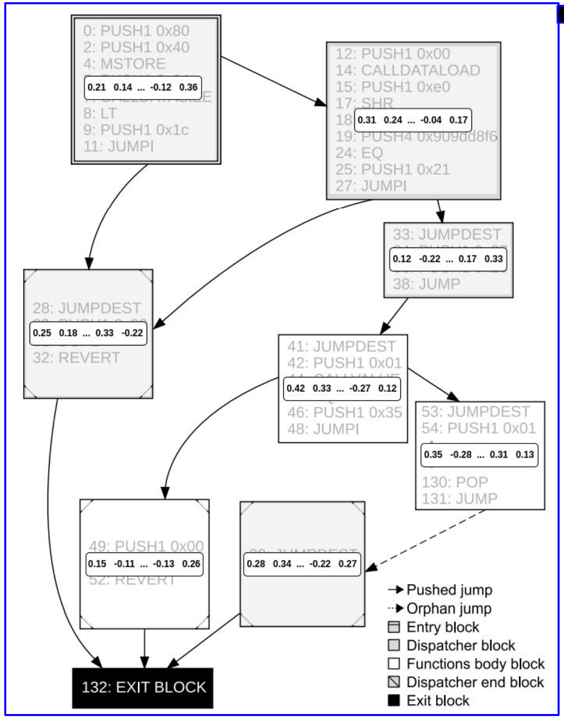
Deep Averaging Network of the Universal Sentence Encoder

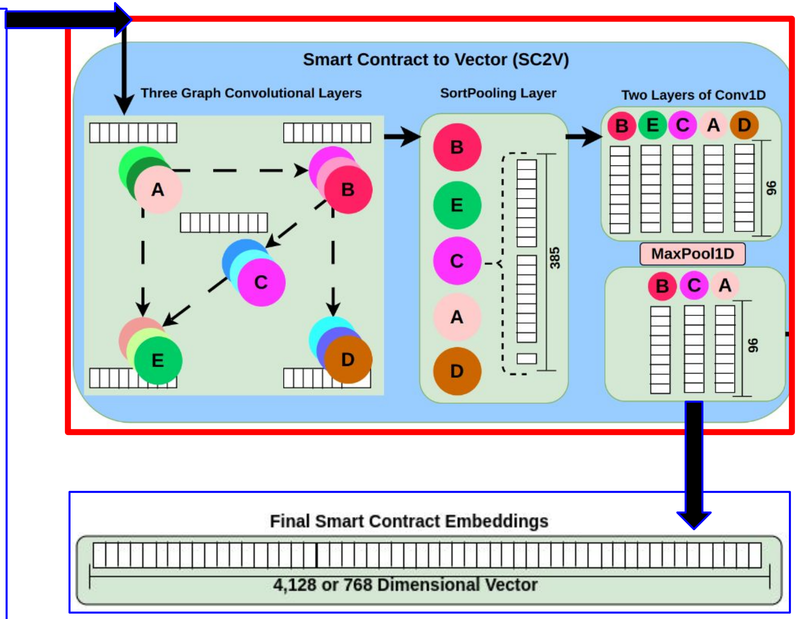
Smart Contract to Vector (SC2V)



Smart Contract to Vector (SC2V)

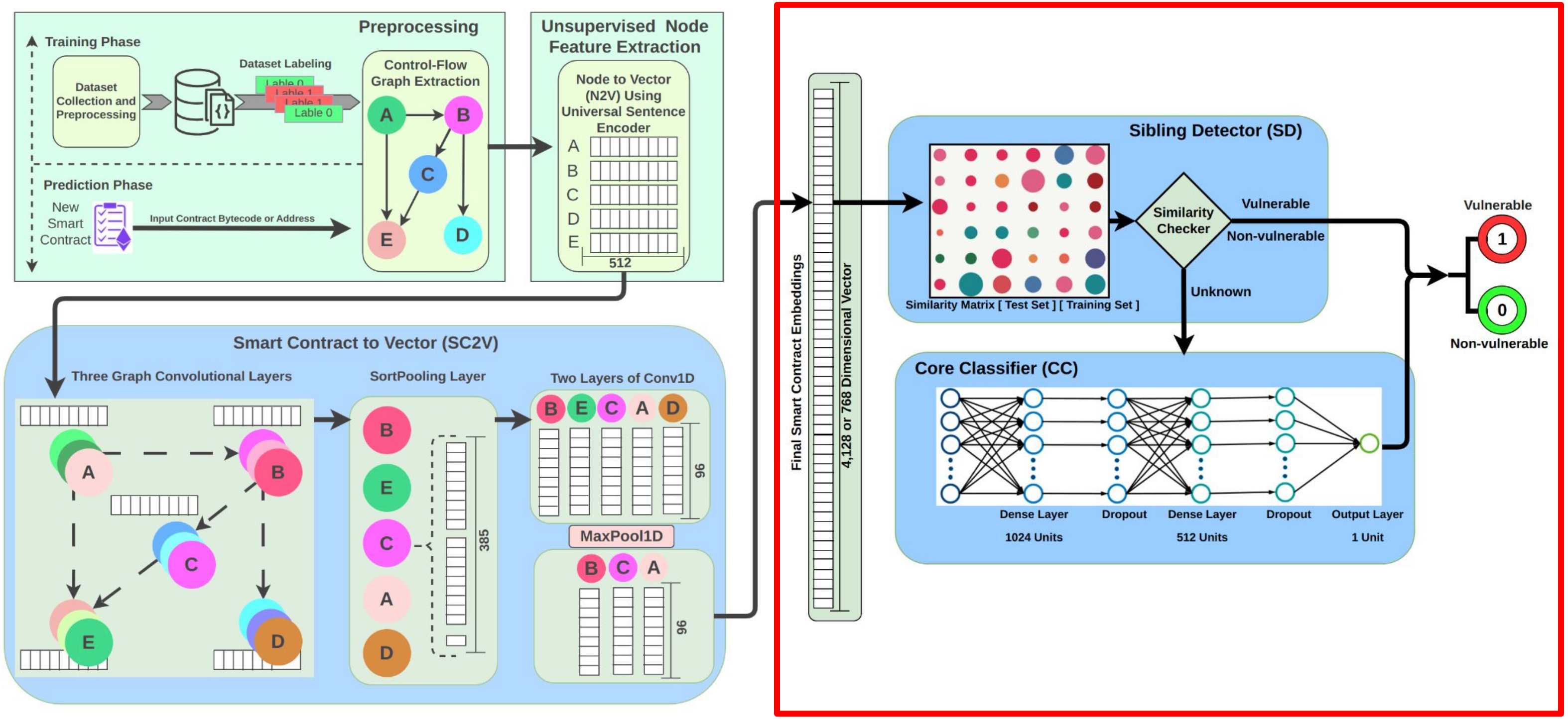
- SC2V uses node summaries to make a vector summary of the entire CFG
- 🔑: Use graph convolution NN to extract semantic structure from CFG





🔑: Similar contracts →
Similar vectors in embedding space

Sibling Detector (SD) and Core Classifier (CC)

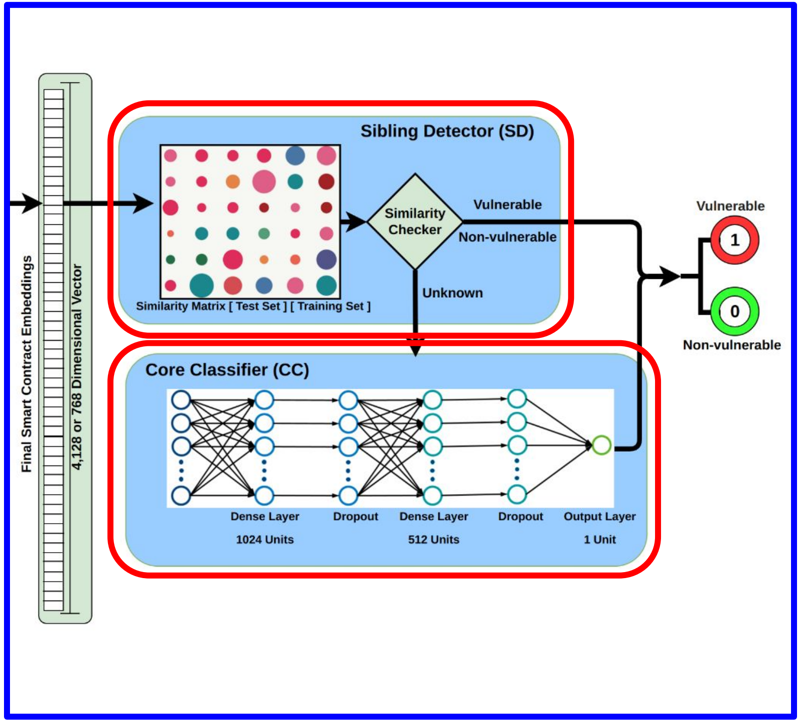


Sibling Detector (SD) and Core Classifier (CC)

★ SD tries to find a Euclidean-close neighbor to the target contract from the training data

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

★ If no close neighbor, SD reports "**unknown**"



★ CC only called when SD reports unknown

★ CC uses feedforward neural net to label vulnerable contracts regardless of vector distance.

Next: Training DLVA

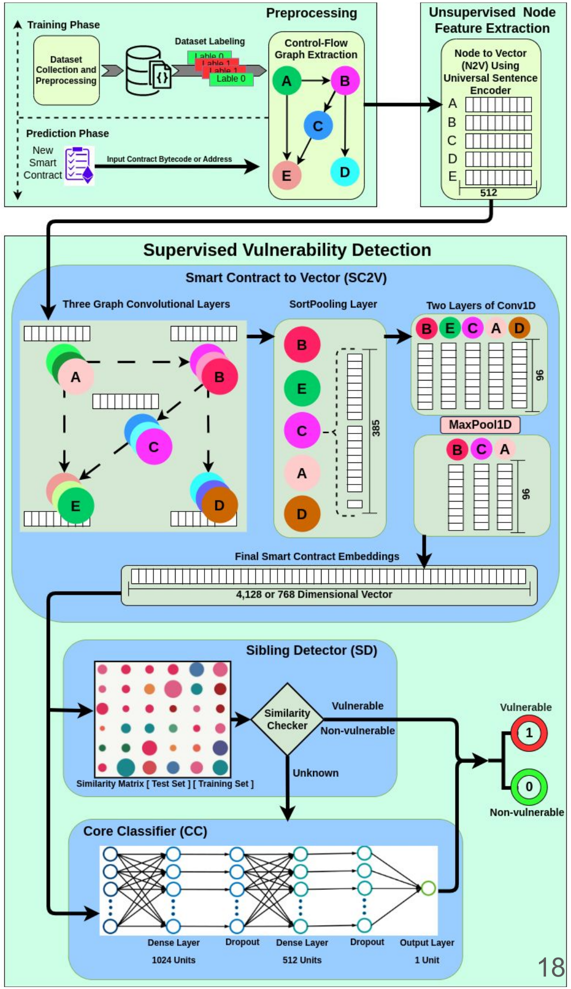
1. Motivation and Introduction ✅
2. Designing DLVA ✅
3. Training DLVA
4. Benchmarking DLVA

Training

★ N2V: 21.9 million basic blocks

★ 🔑 SC2V and CC trained together on 72 thousand contracts

🔑: Slither only labels source code, so we train the <u>bytecode</u> analyzer DLVA using a <u>source code</u> labeller as the oracle



Next: Benchmarking DLVA

1. Motivation and Introduction ✅

2. Designing DLVA ✅

3. Training DLVA ✅

4. Benchmarking DLVA

Datasets used to benchmark DLVA

Dataset	No. of contracts	No. of vulnerabilities	Contract size	Ground Truth
$EthereumSC_{large}$ [3]	22,634	29	Large	Slither
$EthereumSC_{small}$ [4]	1,381	21	Small	Slither
$Elysium_{benchmark}$ [2]	900 (57)	2	Small	Peer-reviewed
$Reentrancy_{benchmark}$ [6]	473 (472)	1	Small	GP: Manual and GN: 2 analyzers
$SolidiFI_{benchmark}$ [8]	444	4	Large	GP: Bug injection and GN: 5 analyzers

🔑: All datasets are disjoint from DLVA's training sets

🔑: All datasets are publicly available

Competitor summary

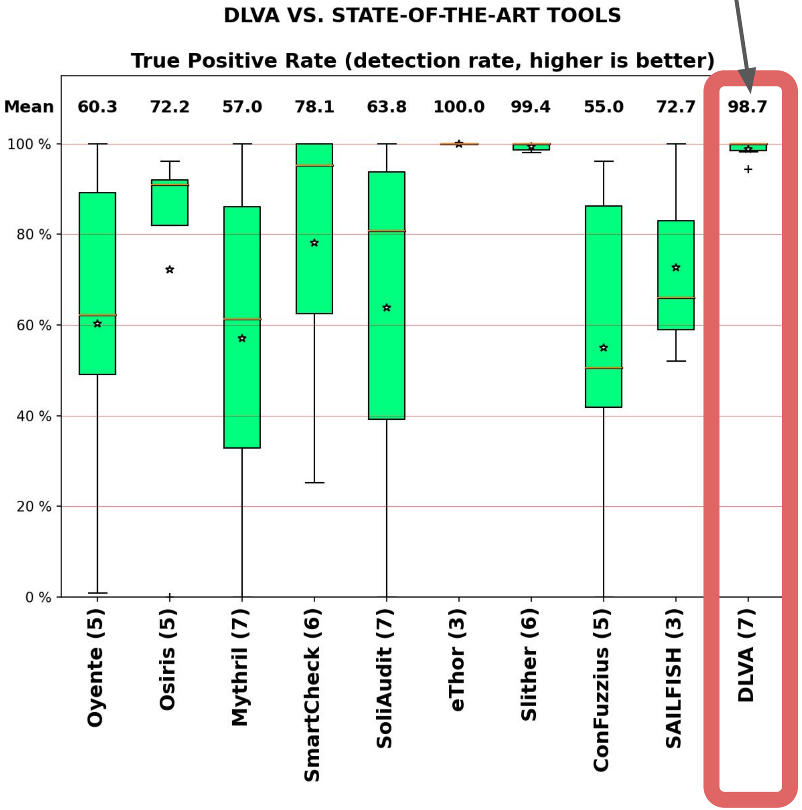
Analyzer	Input	Method	Vulnerabilities	Year	Citations
Oyente 0.2.7	source+ & binary-	static analysis	4	2017	1,996
Osiris	source+ & binary-	static analysis	5	2018	234
Mythril 0.21.20	source+ & binary-	static analysis	13	2019	127
SmartCheck 2.0	source	static analysis	43	2019	513
SoliAudit	source	machine learning & fuzzing	13	2019	76
eThor	binary	static analysis	1	2020	80
Slither 0.8.0	source	static analysis	74	2021	292
ConFuzzius	source	static analysis & fuzzing	10	2022	19
SAILFISH	source	static analysis	2	2022	24
DLVA	**binary**	**deep learning**	**29**	**2023**	**NA**

DLVA is highly sensitive

3rd best
with 98.7%

	Predicted Positive	Predicted Negative
Positive	True Positive ✅	False Negative ❌
Negative	False Positive ❌	True Negative ✅

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN}$$

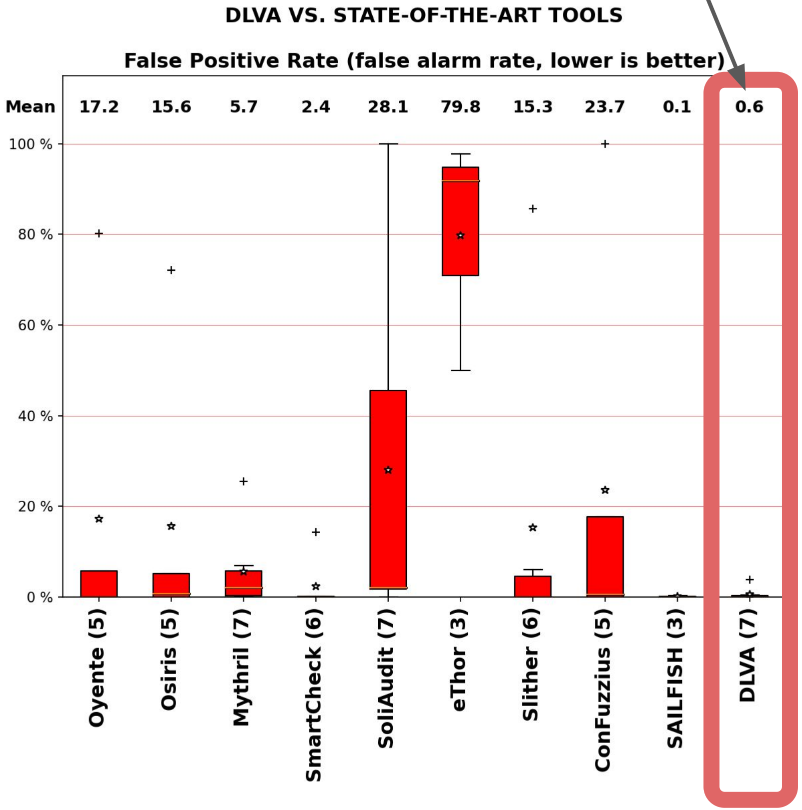


DLVA is highly selective

2nd best with 0.6%

	Predicted Positive	Predicted Negative
Positive	True Positive ✅	False Negative ❌
Negative	False Positive ❌	True Negative ✅

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP+TN}$$

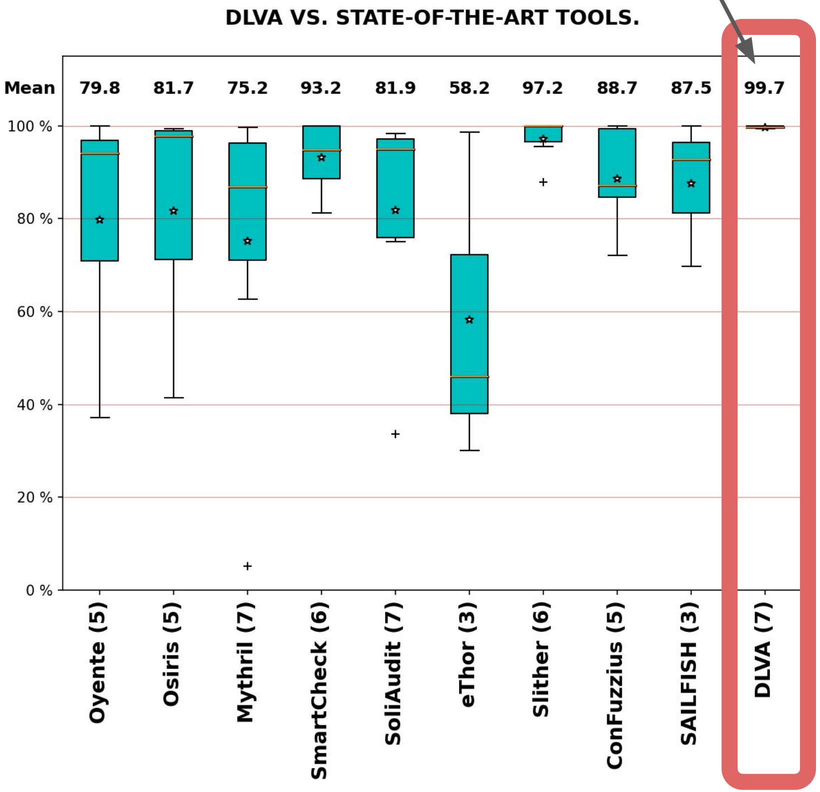


DLVA is highly accurate

Leads the pack with 99.7% accuracy

	Predicted Positive	Predicted Negative
Positive	True Positive ✅	False Negative ❌
Negative	False Positive ❌	True Negative ✅

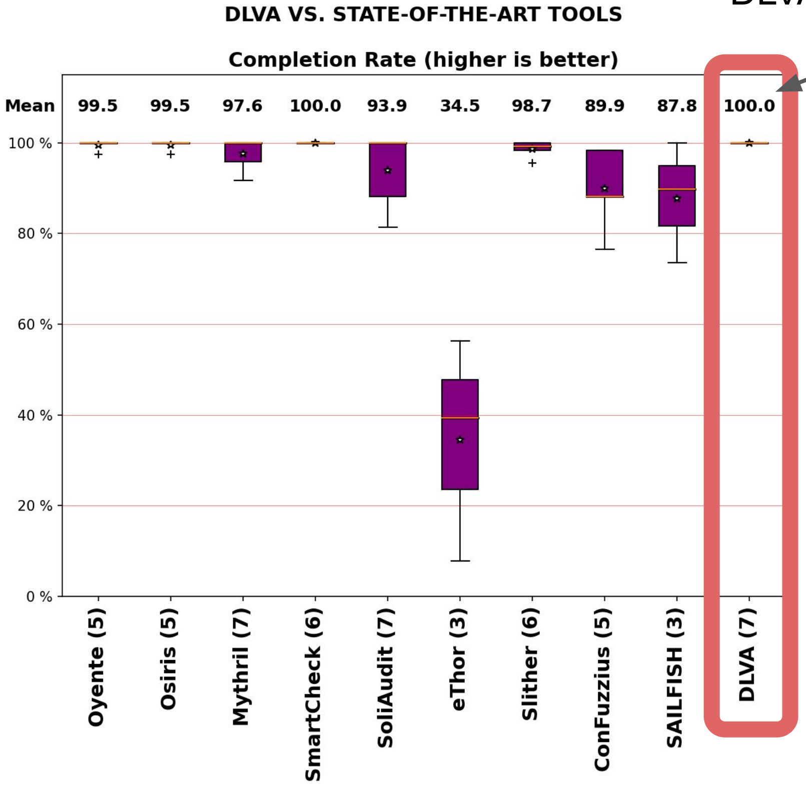
$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

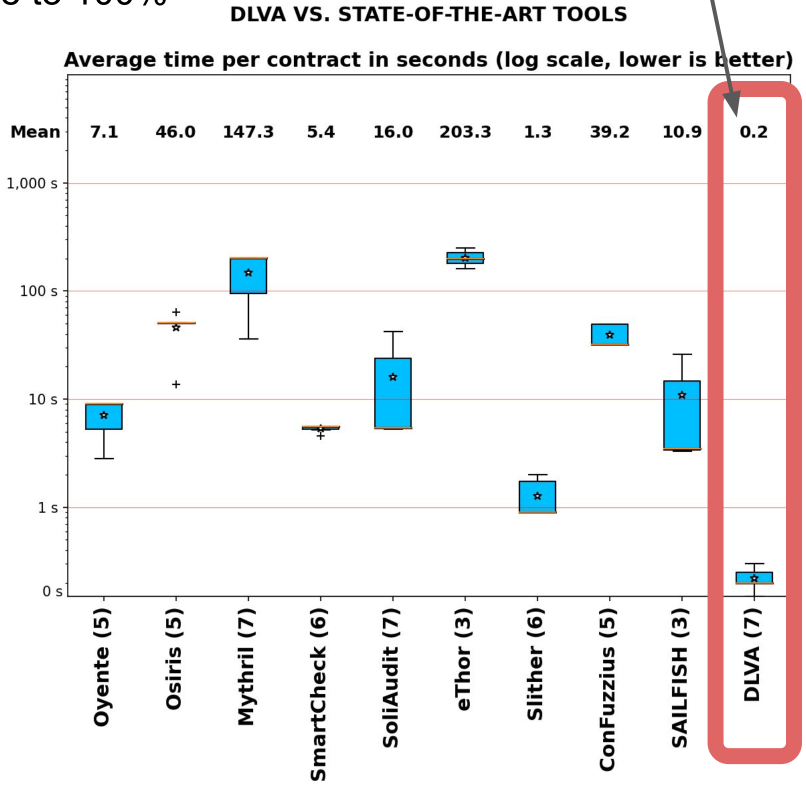


DLVA almost always answers… and very quickly

DLVA is 10x – 1000x faster than competitors

DLVA is close to 100%





Concluding thoughts

Thank you for your attention!

- ★ Neural nets are surprisingly good at understanding smart contracts
- ★ Incorporating some semantic understanding (CFGs) improved results
- ★ It is hard to pin down why something is flagged
- ★ Finding good datasets is tricky
- ★ Performance is essentially real-time
- ★ DLVA is available as a practical tool for immediate use: