On the Learnability, Robustness, and Adaptability of Deep Learning Models for Obfuscation-applied Code

Jiyong Uhm
jiyong423@g.skku.edu
Sungkyunkwan University
Suwon, South Korea

Yujeong Kwon
shr2008@g.skku.edu
Sungkyunkwan University
Suwon, South Korea

Hyungjoon Koo
kevin.koo@skku.edu
Sungkyunkwan University
Suwon, South Korea

2025.08.21

Deep Learning Models for Binary Analysis

- Numerous deep learning models for binary analysis!

Deep Learning Models for Binary Analysis

- Numerous deep learning models for binary analysis!

Binary obfuscation is common (e.g., malware)
How robust are DL models against obfuscation?

Overview

- Preliminary work

- Evaluating ML models against different obfuscation techniques
  - Learnability
  - Generalizability
  - Robustness
  - Adaptability

Public Obfuscation Tools

- IR-based
  - Obfuscator-LLVM

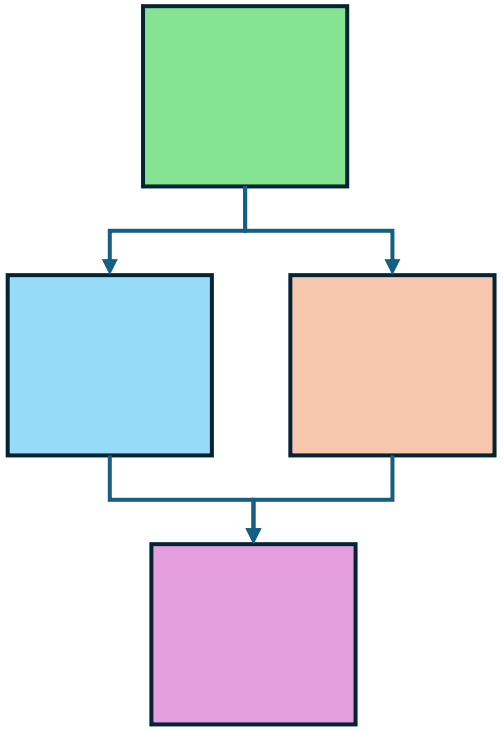
2015 IEEE/ACM 1st International Workshop on Software Protection

Obfuscator-LLVM — Software Protection for the Masses

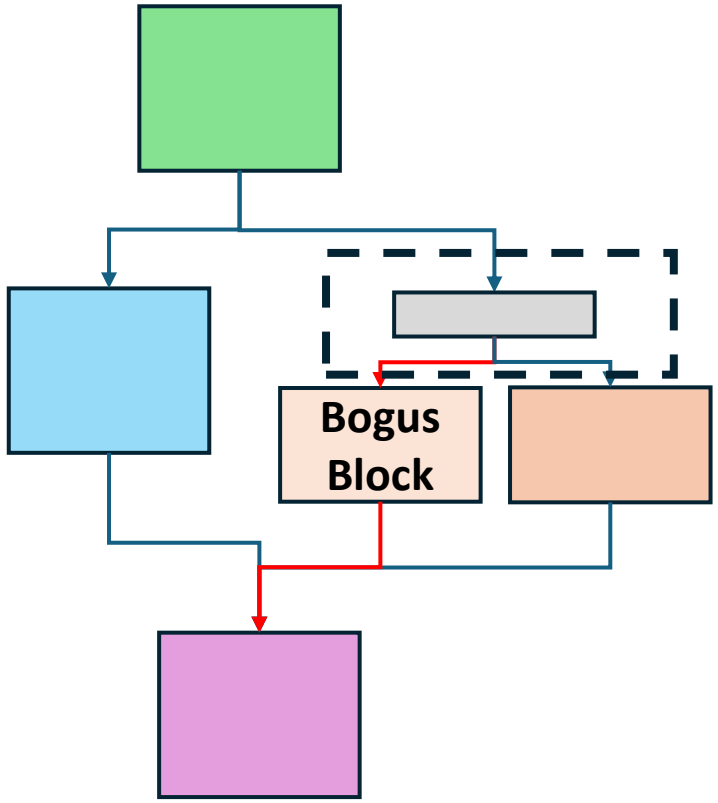
Pascal Junod*, Julien Rinaldini*, Johan Wehrli* and Julie Michielin†
*University of Applied Sciences and Arts Western Switzerland
HES-SO / HEIG-VD / IICT
Yverdon-les-Bains (Switzerland)
{pascal.junod, julien.rinaldini, johan.wehrli}@heig-vd.ch
†Kudelski Security - Nagravision SA
Cheseaux-sur-Lausanne (Switzerland)
julie.michielin@nagra.com

- Source-based
  - Tigress

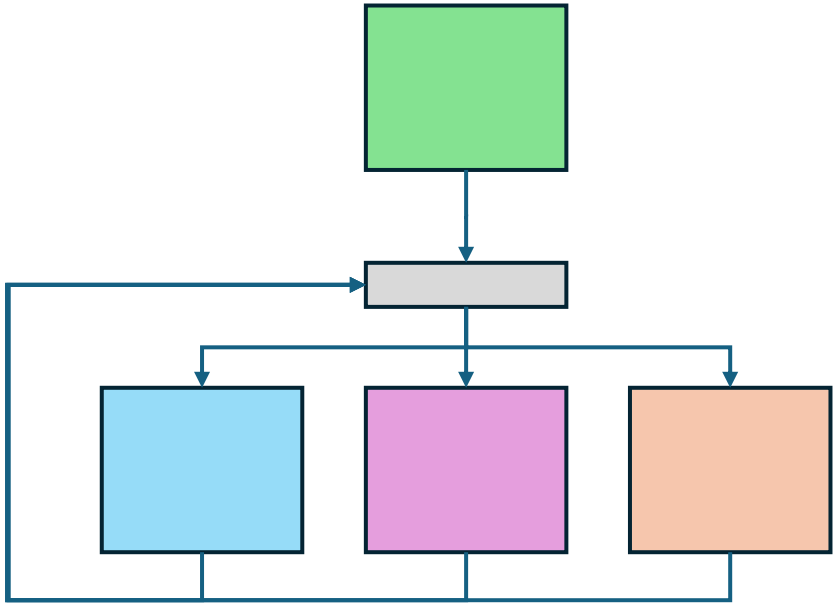
Code Obfuscation Techniques



Benign

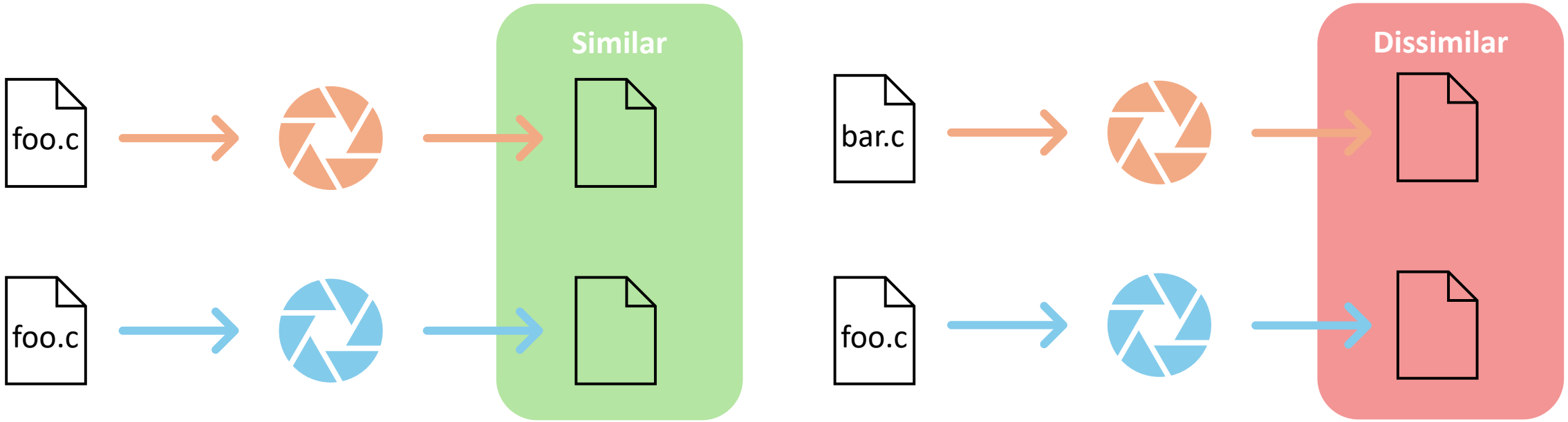


Opaque Predicate Insertion



Control Flow Flattening

Binary Code Similarity Detection Model



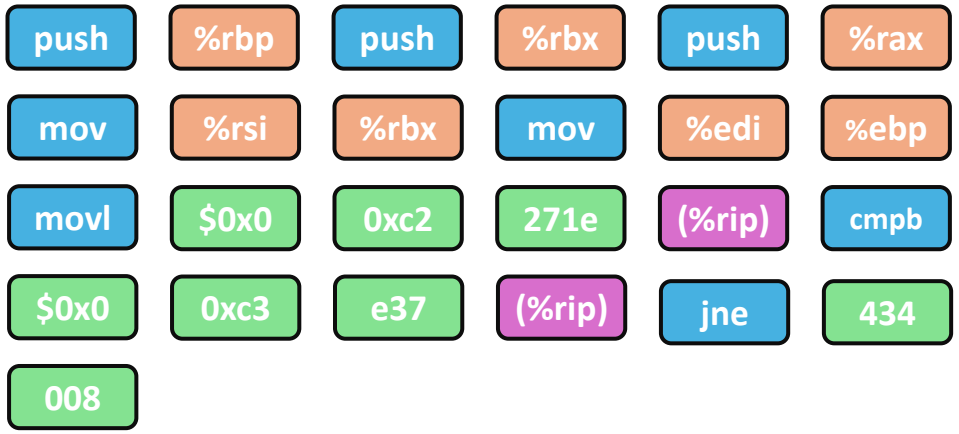
BinShot

- BinShot *(Practical Binary Code Similarity Detection with BERT-based Transferable Similarity Learning; ACSAC '22)*
  - BERT-based BCSD model
  - MLM Training Task

- Two models are trained on different obfuscation tools
  - Evaluate the transferability

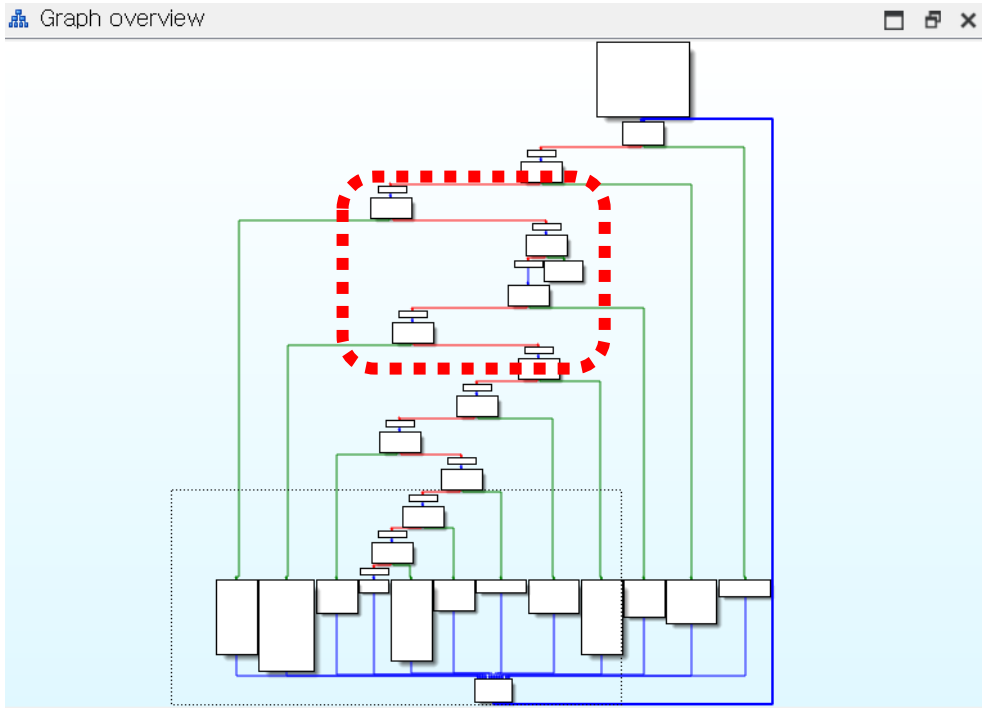
Learning Semantics of Code?

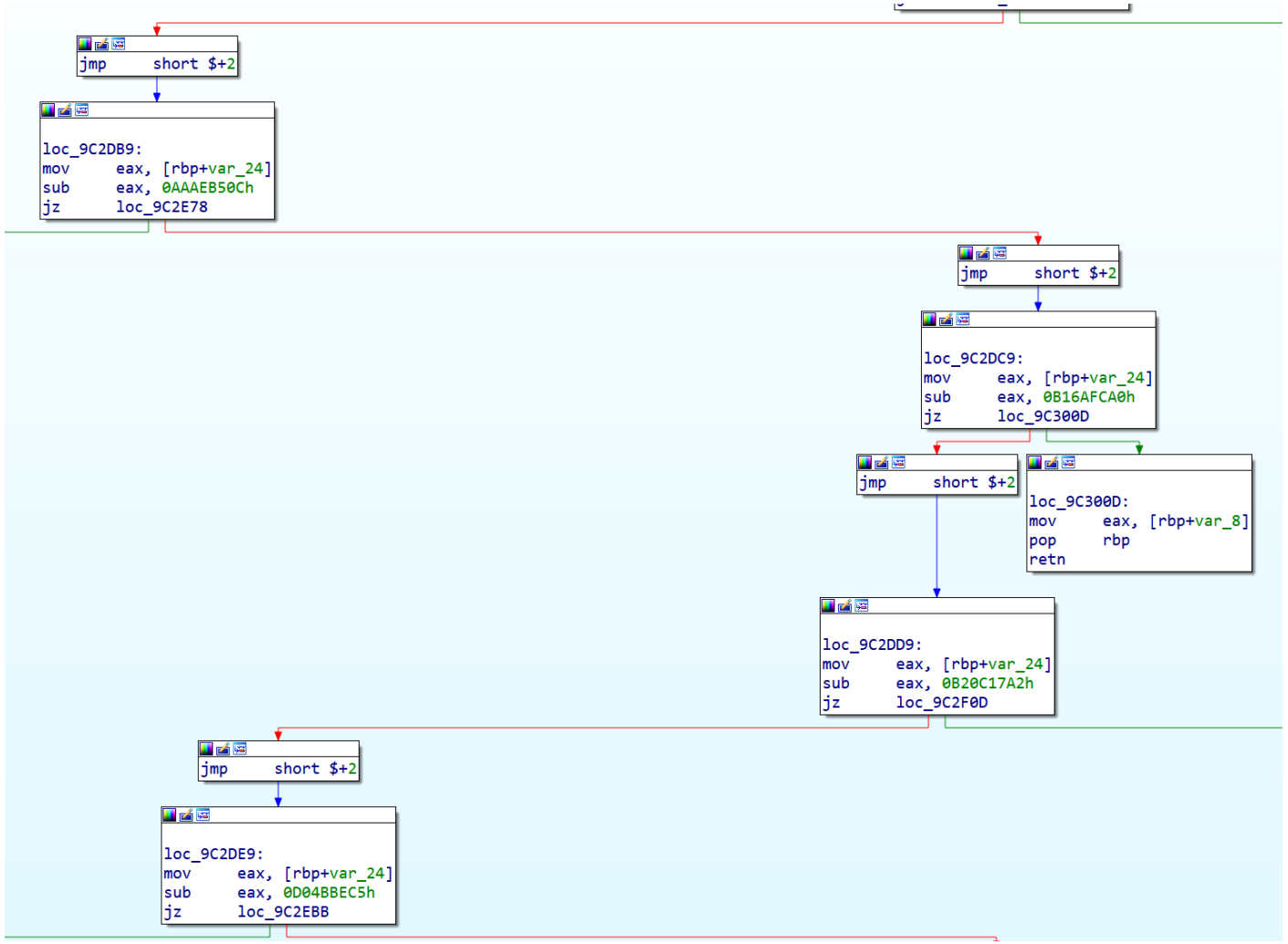
```
0000000000433fc0 <main>:
  433fc0:       55                      push   %rbp
  433fc1:       53                      push   %rbx
  433fc2:       50                      push   %rax
  433fc3:       48 89 f3                mov    %rsi,%rbx
  433fc6:       89 fd                   mov    %edi,%ebp
  433fc8:       c7 05 1e 28 0c 00 00    movl   $0x0,0xc281e(%rip)
  433fd2:       80 3d 37 3e 0c 00 00    cmpb   $0x0,0xc3e37(%rip)
  433fd9:       75 2d                   jne    434008 <main+0x48>
```



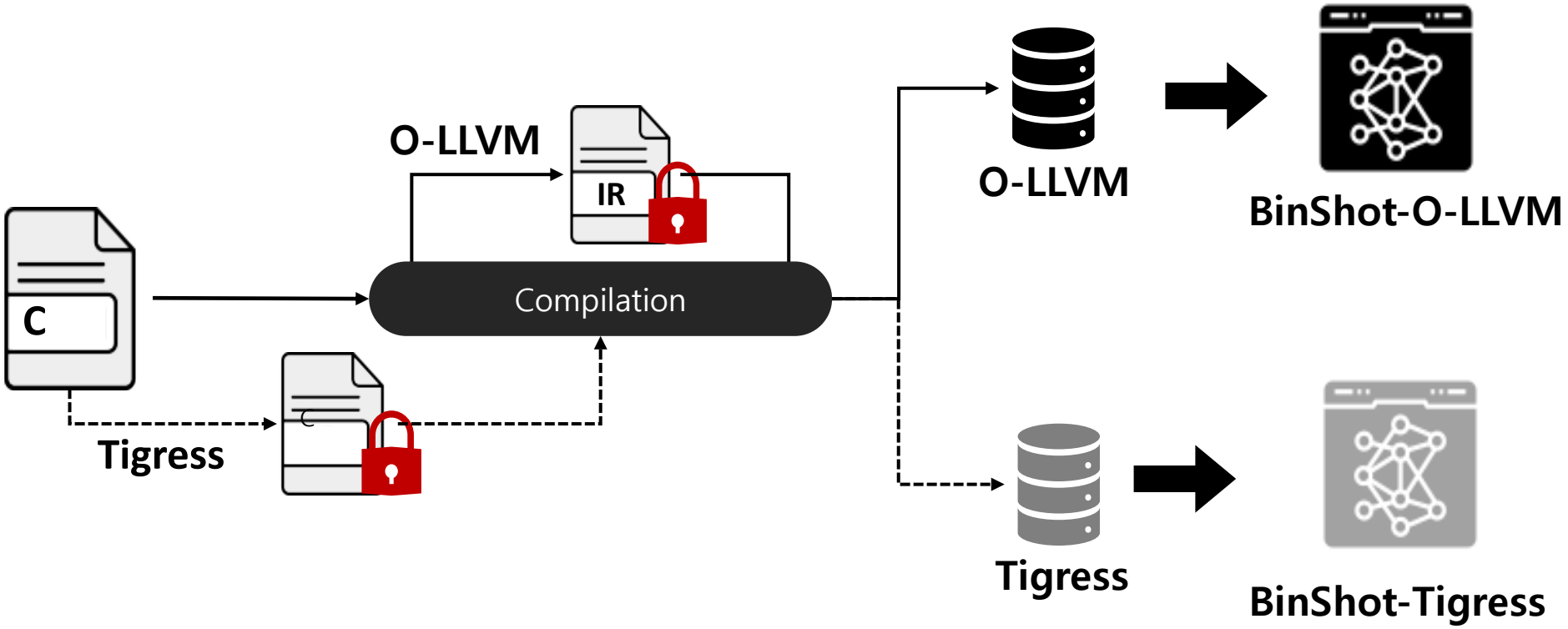
push = {Embedding}

Learning Semantics of Obfuscated Code?





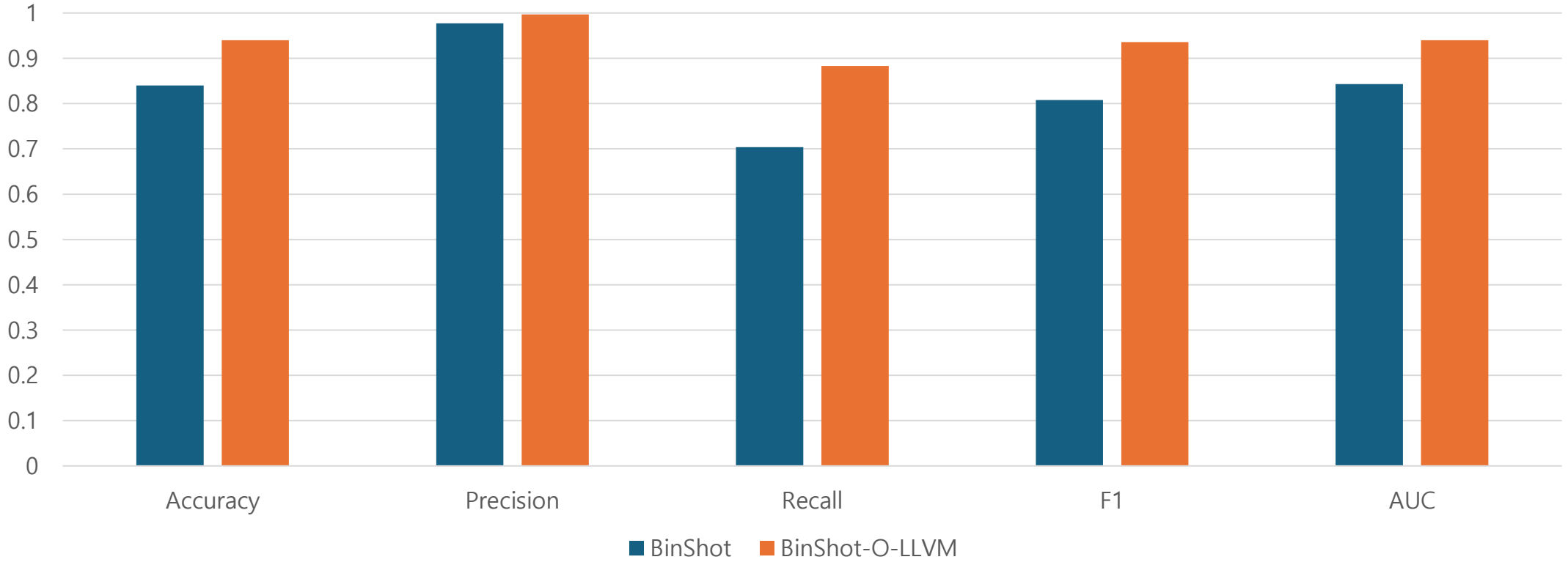
Overview


O-LLVM
IR
C
Compilation
Tigress
O-LLVM
BinShot-O-LLVM
Tigress
BinShot-Tigress

RQ1: Learnability

- How learnable is code obfuscation?
  - Original BinShot model vs. Obfuscation-aware BinShot model

RQ1: Learnability Results

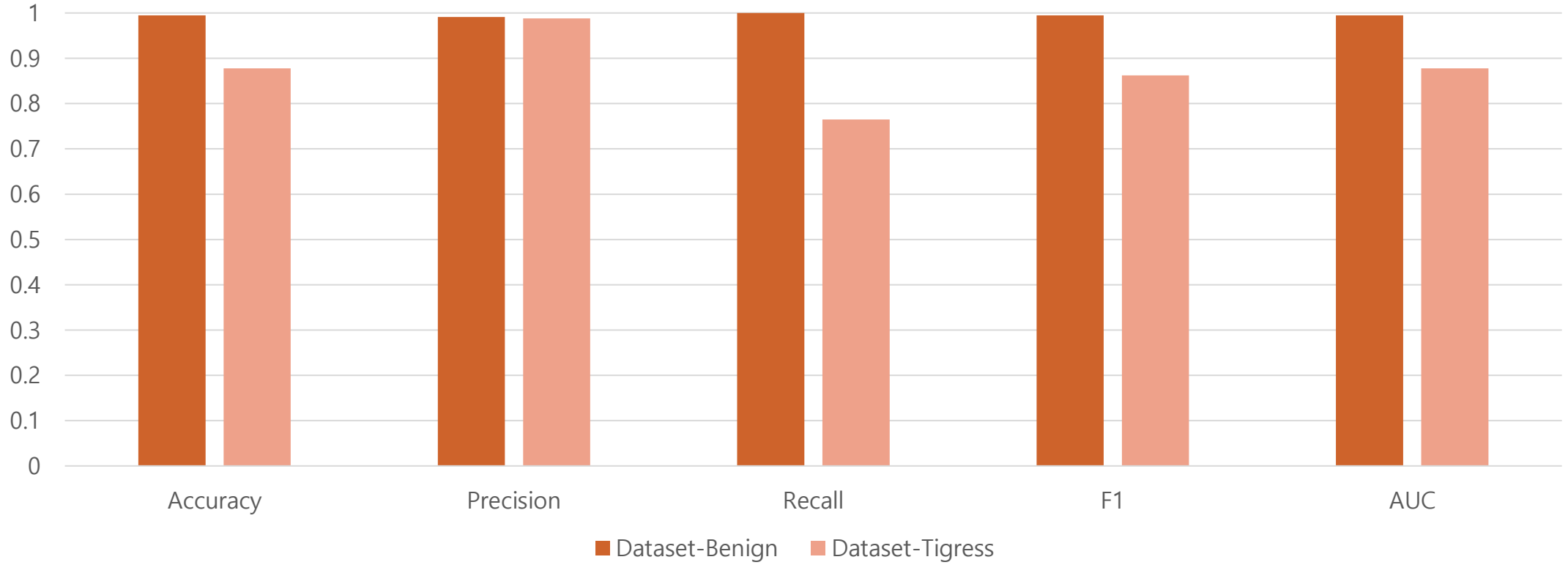


Models can learn (to an extent) by training directly on obfuscated code.

RQ2: Generalizability

- Obfuscation-aware model’s performance
  - Obfuscated vs. non-obfuscated

RQ2: Generalizability Results

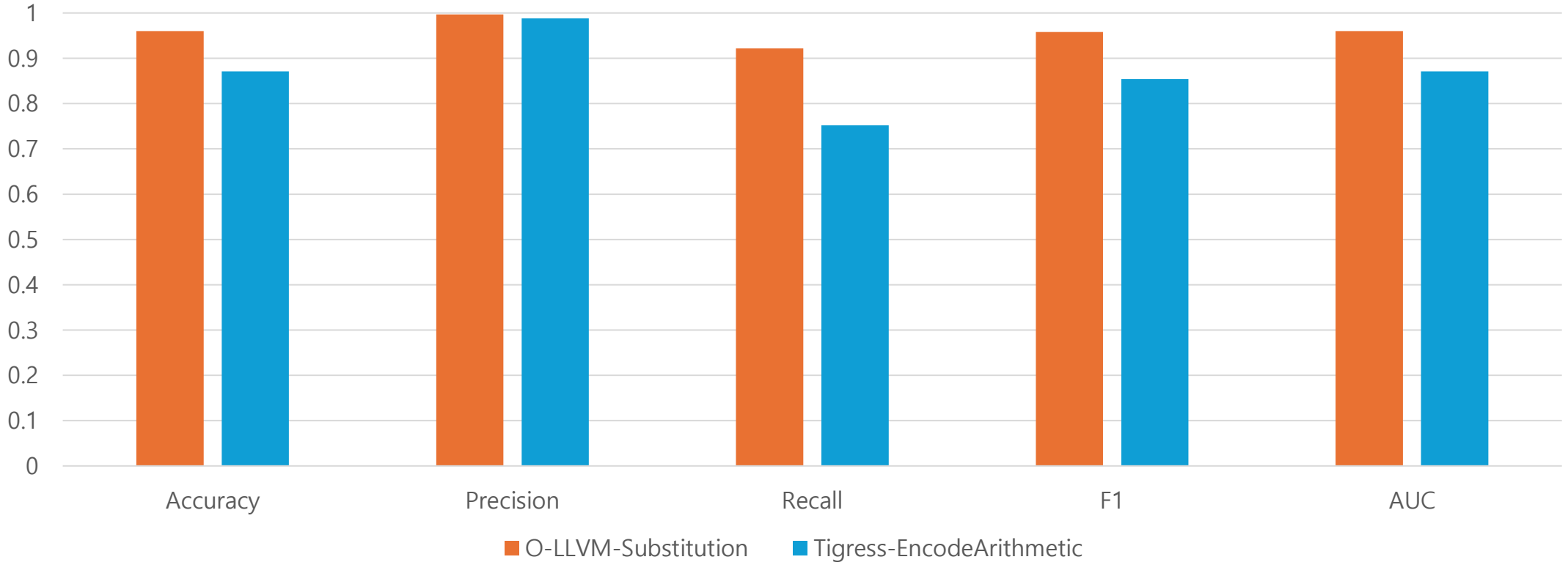


Achieving generalizability across both obfuscated and non-obfuscated code remains challenging

RQ3: Robustness

- Obfuscation-aware model’s performance
  - Known obfuscation techniques (seen during training)

RQ3: Robustness Results

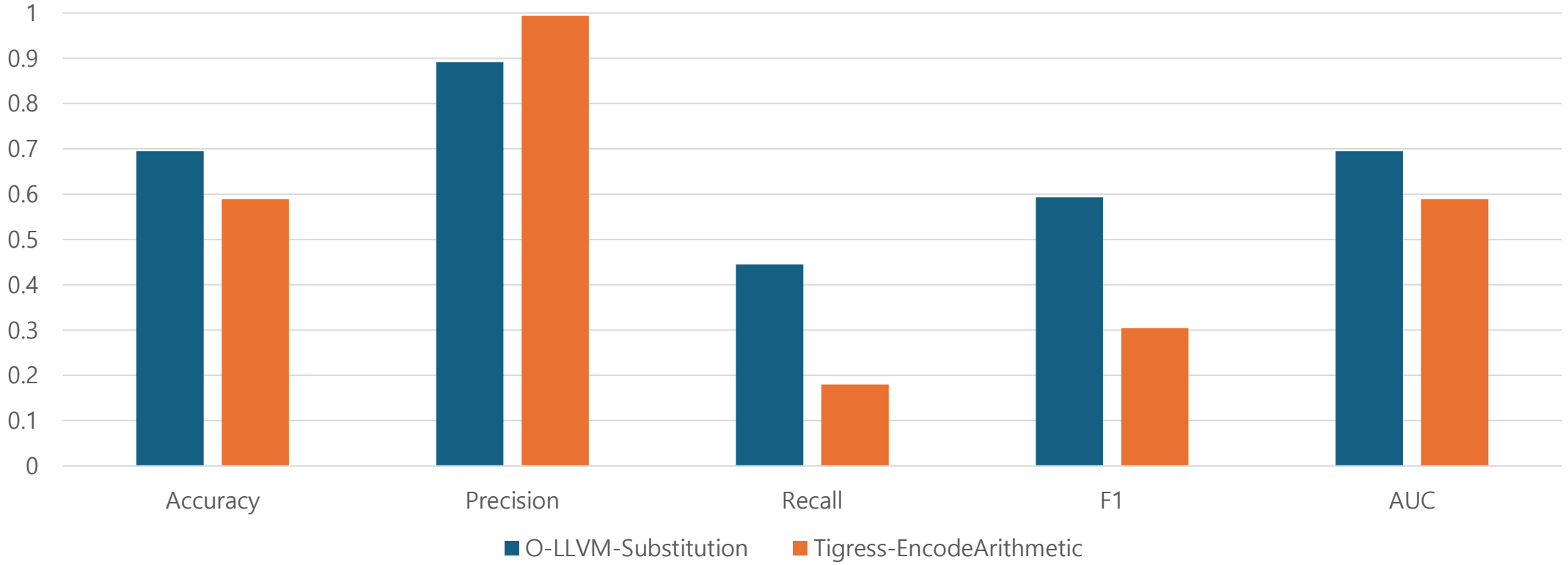


Models trained on obfuscated code demonstrate good performance for known obfuscation techniques

RQ4: Adaptation

- Obfuscation-aware model’s performance
  - Unknown obfuscation techniques (unseen during training)

RQ4: Adaptation Results



Model's ability to adapt to an unknown obfuscation technique depends varies depending on the obfuscation techniques it has encountered

Limitations

- Alternate models
  - How do other BCSD models react?
  - How does obfuscation affect other downstream tasks?

- Alternate obfuscation techniques: future work
  - Additional techniques in Tigress
  - Sophisticated obfuscation techniques from commercial tools

Thank You!

- Any Questions?

- We release the experiment code
  - https://github.com/SecAI-Lab/bcsd_obf_sure2025

Commercial Obfuscation Tools

Themida®
ADVANCED WINDOWS SOFTWARE PROTECTION

Advanced Windows software protection system

- Version: 3.2.4.0
- Date: 28-Jul-2025

ASPack

ASPack is an advanced EXE packer created to compress Win32 executable files an
protect them against non-professional reverse engineering.

The solution makes Windows programs and libraries smaller up to 70% (the compre
ratio is higher than the ZIP standard by 10-20%) what leads to a reduction in the dow
time of compressed applications in local networks and the Internet because of their
smaller size compared to uncompressed apps.

The ASPack exe compressor also provides protection to programs/applications from
unprofessional analysis, debuggers and decompilers. Programs compressed with AS
are self-contained and run exactly as before, with no runtime performance penalties.

Download | Buy now

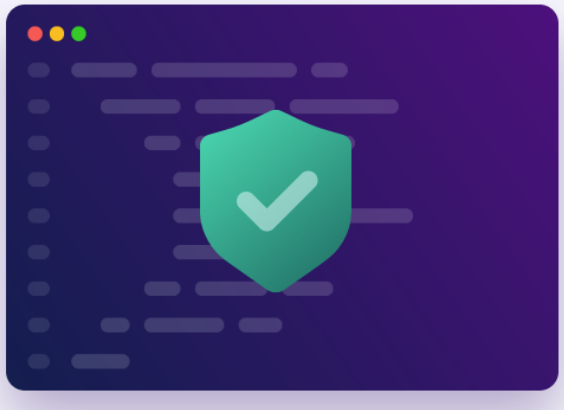
VMProtect Software

Products | News | Support

Advanced {code} security made simple and reliable

Protect your software against cracking, analyzing and reverse-engineering. Make your work safe and secure.

See Pricing → | Overview



Commercial Obfuscation Tools

- Why Not?
  - Packed/Encrypted Executables
  - Deep Learning Models require <u>successful</u> disassembly
- However (Future Work)
  - Real World Application
  - Sophisticated Obfuscation Techniques
  - Dynamic Analysis

Complications with Tigress

- Changes needed for Tigress
  - Generation of merged source
  - Alteration of build process (cannot simply use existing Makefile)

- Some compiler compatibility issues
  - Clang’s compatibility with Tigress’s custom CIL parser