SAMSUNG SDS

Foresee

Techtonic 2021

Partner

Disrupt

GNN을 이용한

악성코드 탐지

이상근 교수

Why do we need AI-based malware detection?

KISA challenge,
Kaggle 등
가용 데이터 증가

Malware Data

라벨링이 포함된 악성/양성 바이너리 코드
데이터 이용 및 ML 경쟁력 향상을 위한 추가
데이터 확보

기존 signature
기반 탐지방법의
한계 극복

Malware Detection & Automation

Zero-Day 공격 등에 대한 강건성

오탐 및 미탐 최소화

항상된 ML 모델
개발을 통한
연구 역량 증대

ML Research

다양한 형태의 데이터를 효과적으로 이용
가능한 ML 모델 연구

Sequence data (바이너리 코드 등)

Graph-structured data (AST, CFG 등)

네트워킹 증가
- Cloud, Edge IoT, 5G/6G 발전
- 세계 인구 77% 연결 예상 [화웨이 2025 Global Industry Vision]

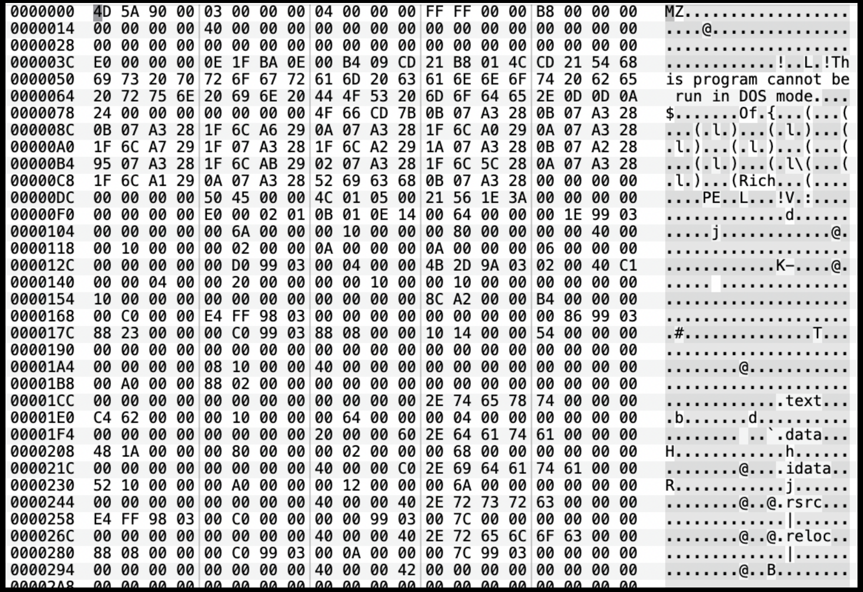
데이터 양, 속도, 종류 증가
- 약 23% 탐지된 공격 분석 미흡
- 약 15% 자동화 공격 증가
[Capgemini 보고서, 2019]

해커의 능력 증대
- 매일 약 200억건의 보안 위협 사례 보고 [Cisco, 2018]

AI 기반 해킹
- 사람보다 두배 이상 효과적 스피어 피싱 트위터 공격 수행 SNAP_R AI
[ZeroFox, 2019]

Feature Engineering / Embedding

Binary Code



```
0000000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00   MZ..................
0000014  00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ....@...............
0000028  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ....................
000003C  E0 00 00 00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68   ............!..L.!Th
0000050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F 74 20 62 65   is program cannot be
0000064  20 72 75 6E 20 69 6E 20 44 4F 53 20 6D 6F 64 65 2E 0D 0D 0A    run in DOS mode....
0000078  24 00 00 00 00 00 00 00 4F 66 CD 7B 0B 07 A3 28 0B 07 A3 28   $.......Of.{...(...(
000008C  0B 07 A3 28 1F 6C A6 29 0A 07 A3 28 1F 6C A0 29 0A 07 A3 28   ...(.l.)...(.l.)...(
00000A0  1F 6C A7 29 1F 07 A3 28 1F 6C A2 29 1A 07 A3 28 0B 07 A2 28   .l.)...(.l.)...(...(
00000B4  95 07 A3 28 1F 6C AB 29 02 07 A3 28 1F 6C 5C 28 0A 07 A3 28   ...(.l.)...(.l\(...(
00000C8  1F 6C A1 29 0A 07 A3 28 52 69 63 68 0B 07 A3 28 00 00 00 00   .l.)...(Rich...(....
00000DC  00 00 00 00 50 45 00 00 4C 01 05 00 21 56 1E 3A 00 00 00 00   ....PE..L...!V.:....
00000F0  00 00 00 00 E0 00 02 01 0B 01 0E 14 00 64 00 00 00 1E 99 03   .............d......
0000104  00 00 00 00 00 6A 00 00 00 10 00 00 00 80 00 00 00 00 40 00   .....j............@.
0000118  00 10 00 00 00 02 00 00 0A 00 00 00 0A 00 00 00 06 00 00 00   ....................
000012C  00 00 00 00 00 D0 99 03 00 04 00 00 4B 2D 9A 03 02 00 40 C1   ............K-....@.
0000140  00 00 04 00 00 20 00 00 00 00 10 00 00 10 00 00 00 00 00 00   ....................
0000154  10 00 00 00 00 00 00 00 00 00 00 00 8C A2 00 00 B4 00 00 00   ....................
0000168  00 C0 00 00 E4 FF 98 03 00 00 00 00 00 00 00 00 00 86 99 03   ....................
000017C  88 23 00 00 00 C0 99 03 88 08 00 00 10 14 00 00 54 00 00 00   .#..............T...
0000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ....................
00001A4  00 00 00 00 08 10 00 00 40 00 00 00 00 00 00 00 00 00 00 00   ........@...........
00001B8  00 A0 00 00 88 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ....................
00001CC  00 00 00 00 00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00   .............text...
00001E0  C4 62 00 00 00 10 00 00 00 64 00 00 00 04 00 00 00 00 00 00   .b.......d..........
00001F4  00 00 00 00 00 00 00 00 20 00 00 60 2E 64 61 74 61 00 00 00   ...........`.data...
0000208  48 1A 00 00 00 80 00 00 00 02 00 00 00 68 00 00 00 00 00 00   H............h......
000021C  00 00 00 00 00 00 00 00 40 00 00 C0 2E 69 64 61 74 61 00 00   ........@....idata..
0000230  52 10 00 00 00 A0 00 00 00 12 00 00 00 6A 00 00 00 00 00 00   R............j......
0000244  00 00 00 00 00 00 00 00 40 00 00 40 2E 72 73 72 63 00 00 00   ........@..@.rsrc...
0000258  E4 FF 98 03 00 C0 00 00 00 00 99 03 00 7C 00 00 00 00 00 00   .............|......
000026C  00 00 00 00 00 00 00 00 40 00 00 40 2E 72 65 6C 6F 63 00 00   ........@..@.reloc..
0000280  88 08 00 00 00 C0 99 03 00 0A 00 00 00 7C 99 03 00 00 00 00   .............|......
0000294  00 00 00 00 00 00 00 00 40 00 00 42 00 00 00 00 00 00 00 00   ........@..B........
```

Vector

1. Feature Engineering (Extraction)
   ex. Number/length/entropy of PE sections

2. Feature Embedding
   (Dis)similar objects → (dis)similar vectors
   Low-dimensional vectors are preferred

Problem Space | Feature Space

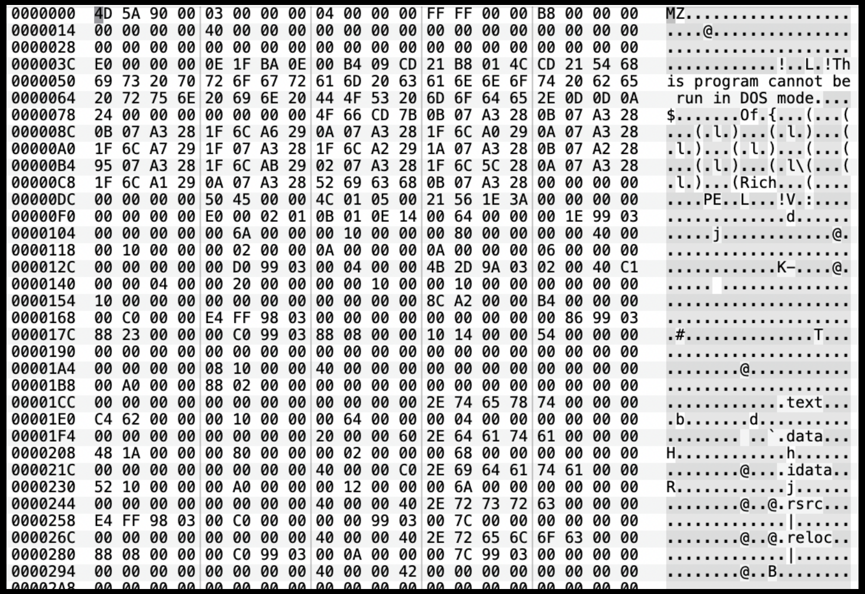
Graph-Structured Data

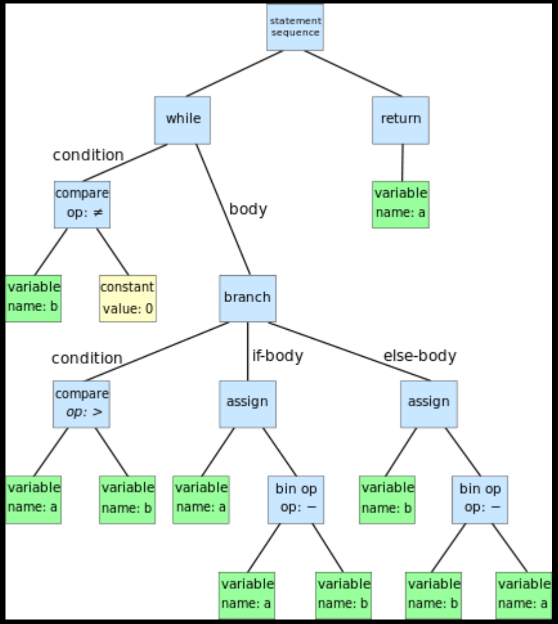
Robust to src changes & compilation options

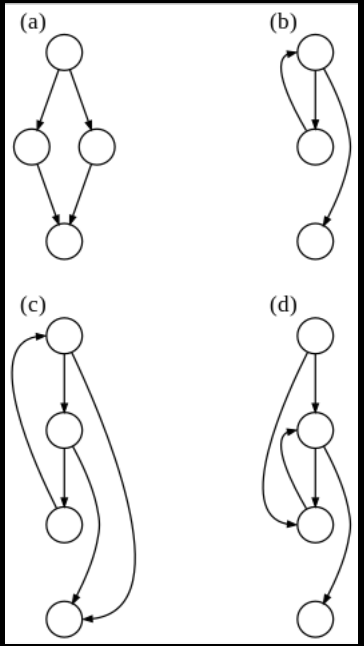
Binary Code

AST
(Abstract Syntax Tree)

CFG
(Control Flow Graph)







If-then-else

While loop

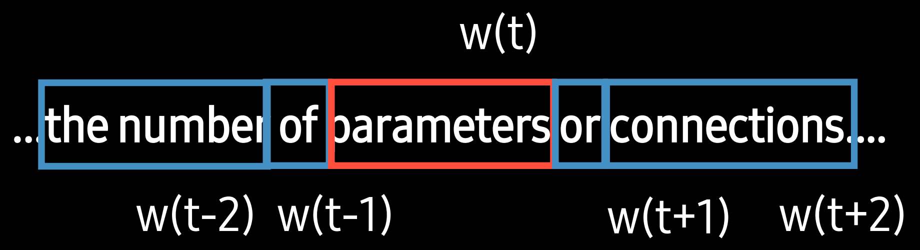
Two loops (reducible)

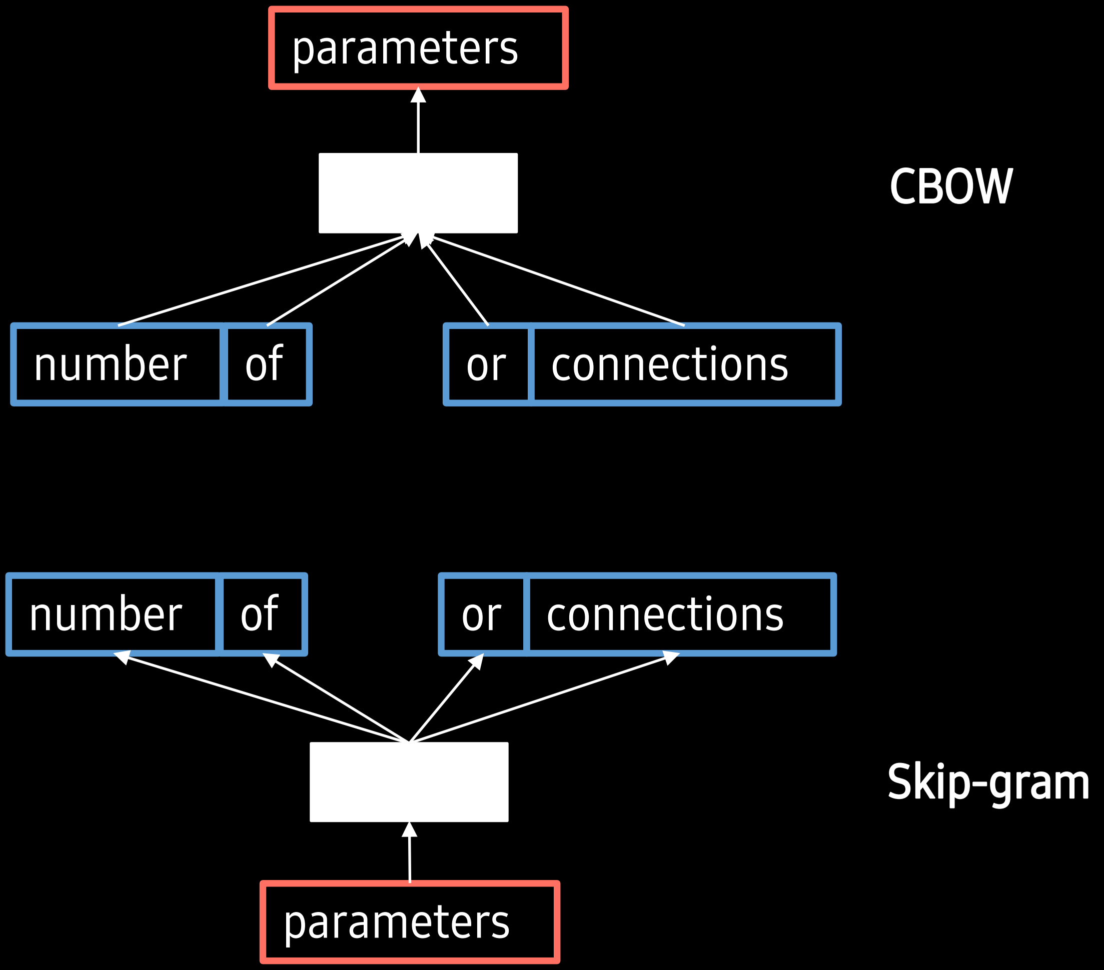
Two loops (irreducible)

Problem Space

https://en.wikipedia.org/wiki/Abstract_syntax_tree
https://en.wikipedia.org/wiki/Control-flow_graph

Embedding: Word2Vec



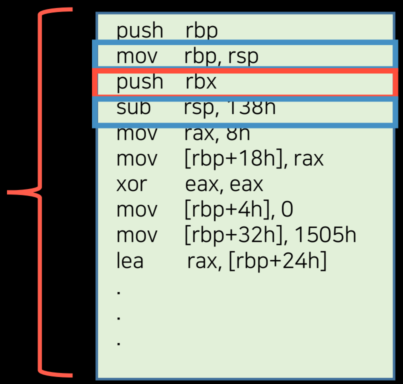


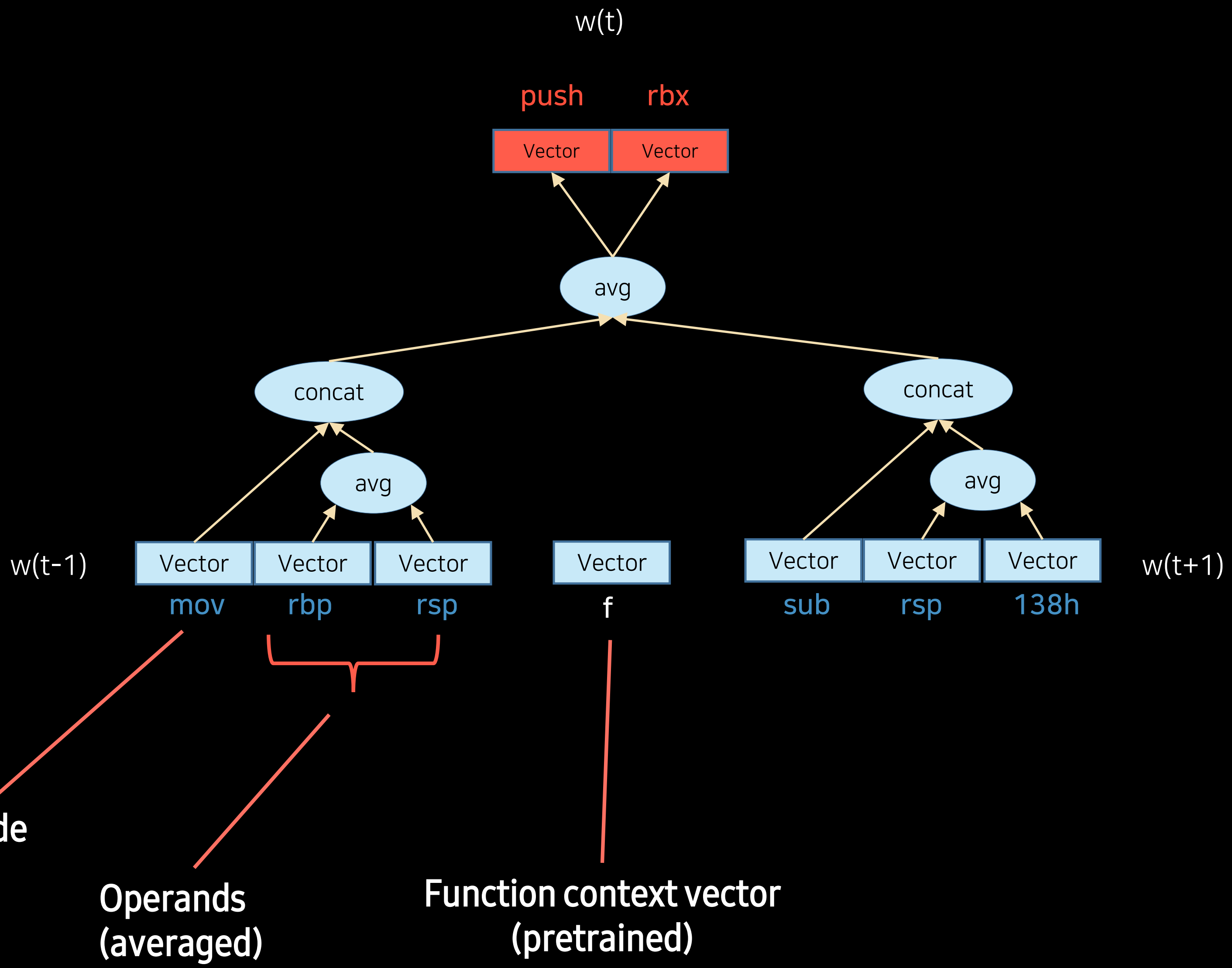
Mikolov et al., Efficient Estimation of Word Representations in Vector Space, 2013 (https://arxiv.org/pdf/1301.3781.pdf)

Embedding: Asm2Vec

Ding, Fung & Charland, Asm2Vec: Boosting Static Representation Robustness for Binary Clone Search against Code Obfuscation and Compiler Optimization, IEEE S&P, 2019

A function





An asm token → a vector (averaging)

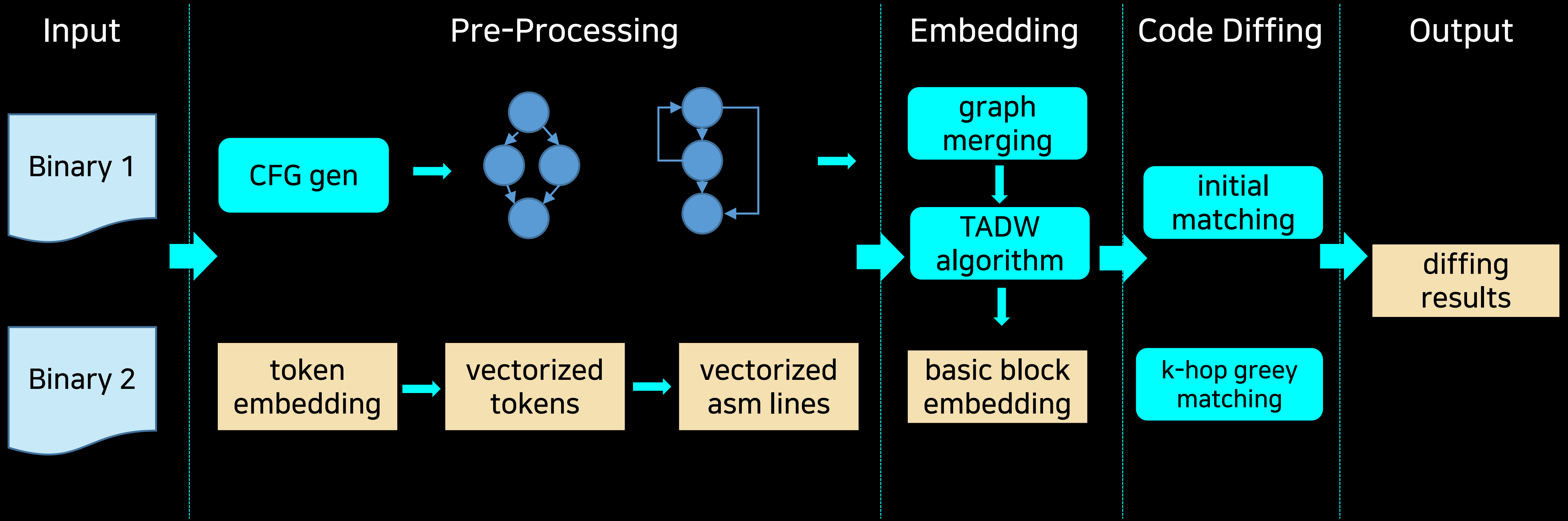
An asm line → a vector (averaging)

A function → a vector

Duan et al., DeepBinDiff: Learning Program-Wide Code Representations for Binary Diffing, NDSS, 2020

Embedding: DeepBinDiff

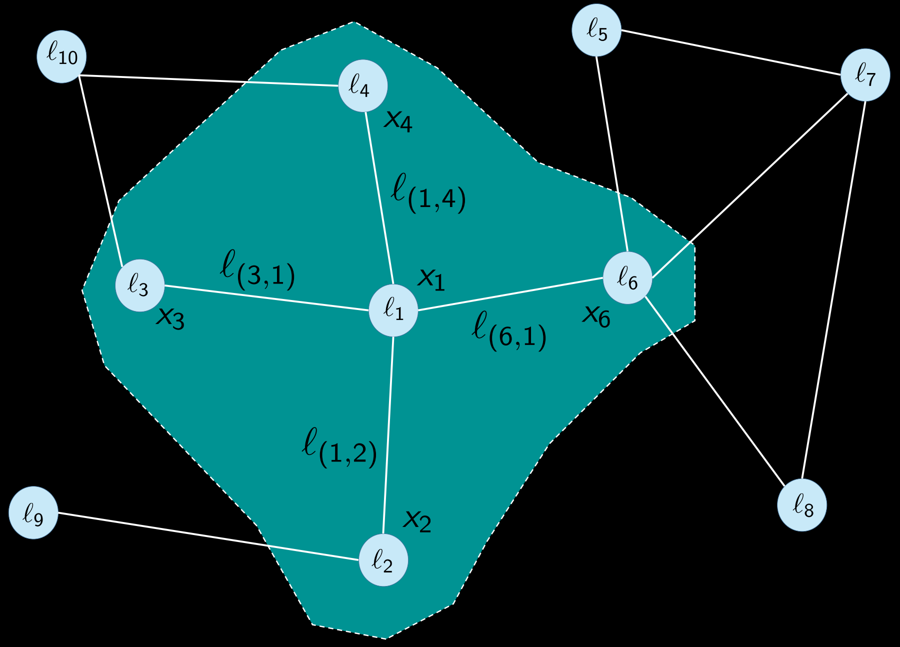
DeepBinDiff: an embedding learned from two binaries on a merged CFGs



An issue: graph information is implicit, being dissolved in the embedding vectors (Asm2Vec, DeepBinDiff)

Scarselli et al., The Graph Neural Network Model, IEEE Transactions on Neural Networks, 2009

Graph Neural Nets (GNNs)



$$x_1 = f_w(\ell_1, \underbrace{\ell_{(1,2)}, \ell_{(3,1)}, \ell_{(1,4)}, \ell_{(6,1)}}, \underbrace{x_2, x_3, x_4, x_6, \ell_2, \ell_3, \ell_4, \ell_6})$$

state node label, edge labels, neighboring node states/labels

Local transition function

$$\boldsymbol{x}_n = f_{\boldsymbol{w}}(\boldsymbol{l}_n, \boldsymbol{l}_{\mathrm{co}[n]}, \boldsymbol{x}_{\mathrm{ne}[n]}, \boldsymbol{l}_{\mathrm{ne}[n]})$$
$$\boldsymbol{o}_n = g_{\boldsymbol{w}}(\boldsymbol{x}_n, \boldsymbol{l}_n)$$

Local output function

Learning: fixed-point problem

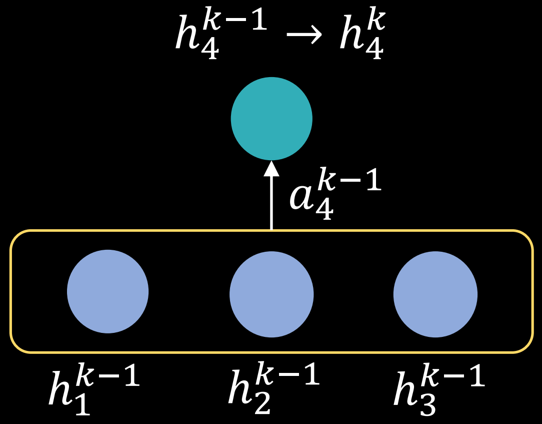
$$\boldsymbol{x}_n(t+1) = f_{\boldsymbol{w}}(\boldsymbol{l}_n, \boldsymbol{l}_{\mathrm{co}[n]}, \boldsymbol{x}_{\mathrm{ne}[n]}(t), \boldsymbol{l}_{\mathrm{ne}[n]})$$
$$\boldsymbol{o}_n(t) = g_{\boldsymbol{w}}(\boldsymbol{x}_n(t), \boldsymbol{l}_n), \qquad n \in \boldsymbol{N}.$$

Graph Isomorphism Issues in GNN

Major GNN operations:

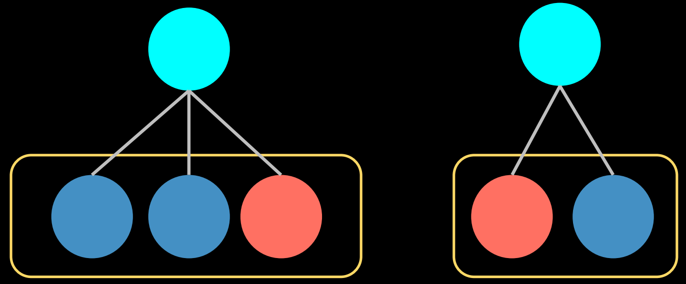
$$a_v^k = AGGREGATE^k(\{h_u^{k-1}: u \in N(v)\})$$

$$h^k = COMBINE^k(h_v^{k-1}, a^k)$$



GraphSage (NIPS 2017)

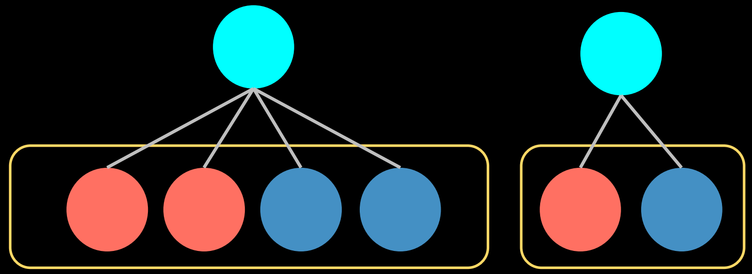
$$a_v^k = \mathrm{MAX}(\mathrm{ReLU}(W \cdot h_u^{k-1}, \forall u \in \mathcal{N}(v))$$



Fails to distinguish multi-sets with the same distinct elements

GCN (ICLR 2017)

$$h_v^k = \mathrm{ReLU}(W \cdot \mathrm{MEAN}\{h_u^{k-1}, \forall u \in \mathcal{N}(v) \cup \{v\}\})$$

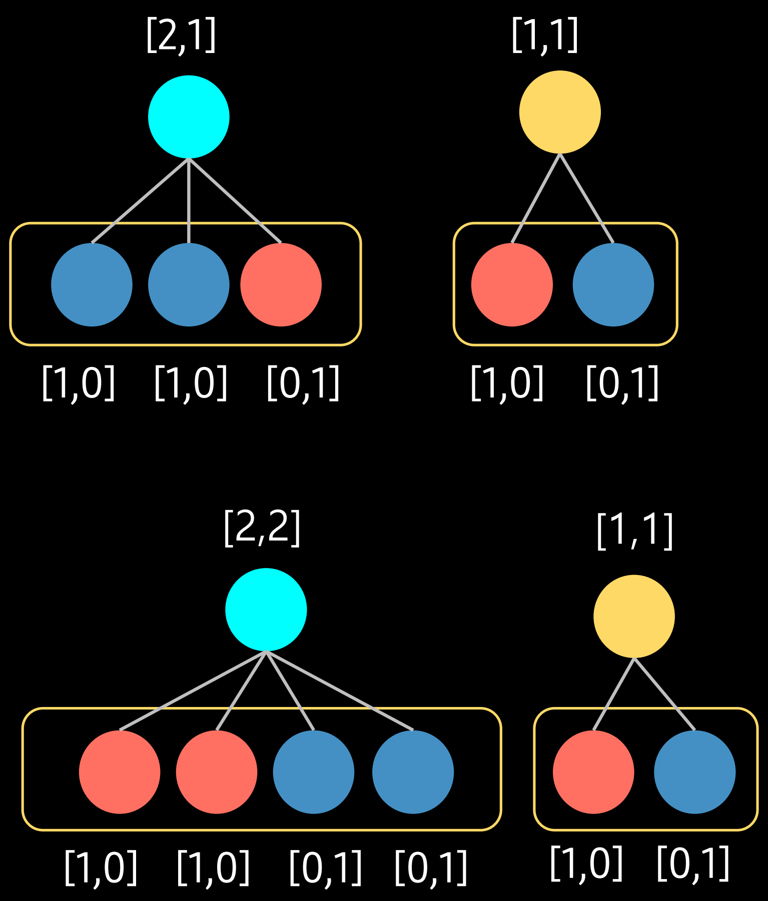


Fails to distinguish proportionally equivalent multi-sets

Graph Isomorphism Net (GIN)

GIN: use the summation as the aggregation function

$$h_v^k = \mathrm{MLP}(h_v^{k-1} + \sum_{v \in \mathcal{N}(v)} h_u^{k-1})$$



Theorem: GNN is as powerful as the Weisfeiler-Lehman test (test of graph isomorphism) if the combine and aggregate functions of GNN are injective in countable space

Challenges & Discussion

Binary packing and obfuscation

- 유효한 정보를 얻기 위한 unpacking 또는 비난독화 필요

CFG generation and cost

- 오픈소스 도구 사용시 오류 처리
- 노드 수가 많은 CFG 생성 시간 및 학습 시간 이슈

성능 향상을 위한 데이터 추가 확보 필요

- AI기반 악성코드 탐지기 학습을 위한 KISA 악성코드 탐지 challenge 데이터셋의 유효성 확인
- 성능 향상을 위한 추가 데이터 확보 필요

Thank you

SAMSUNG SDS