JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 31 No. 3, pp. 97-106, March 2026
https://doi.org/10.9708/jksci.2026.31.03.097

Android Malware Classification Using an Edge-Centric Graph Isomorphism Network

Yeeun Lee*, Hyeona Jang*, Hanseul Jung*, Eunjung Choi**

*Undergraduate Student, Department of Information Security, Seoul Women's University, Seoul, Korea
**Assistant Professor, Department of Information Security, Seoul Women's University, Seoul, Korea

[Abstract]

Function Call Graphs (FCGs) effectively represent execution flows and behavioral structures in static Android malware detection; however, existing studies have shown limitations in utilizing structural edge attribute design by treating function calls as simple connections. This study incorporates diverse edge attributes into Android malware FCGs and comparatively analyzes the performance of GINE (Graph Isomorphism Network with Edge Features)-based malware family classification. Experimental results show that the Baseline model achieved an accuracy of 0.8347 with high variance, whereas Betweenness, ID Distance, and Depth Difference exhibited consistent accuracy improvements of 6.4–7.4 percentage points, and Timestamp, Frequency, and Direction achieved gains of 4.9–5.6 percentage points. These results suggest that structural edge attribute design contributes to improved classification performance and stability in Android malware FCG analysis.

▸ **Key words**: GIN, android, malware, edge feature, cybersecurity, classification

[요 약]

Function Call Graph(FCG)는 정적 Android 악성코드 탐지에서 실행 흐름과 행위 구조를 효과적으로 표현하지만, 기존 연구는 함수 호출을 단순 연결로 취급하여 edge 수준 의미 정보 활용에 한계를 보였다. 본 연구는 다양한 edge attribute를 Android 악성코드 FCG에 통합하고, GINE(Graph Isomorphism Network with Edge Features) 기반 악성코드 패밀리 분류 성능을 비교·분석하였다. 실험 결과, Baseline 모델은 정확도 0.8347과 높은 분산을 기록한 반면 Betweenness, ID Distance, Depth Difference는 6.4–7.4%p, Timestamp, Frequency, Direction은 4.9–5.6%p의 일관된 성능 향상을 보였다. 이는 Android 악성코드 FCG 분석에서 구조적 엣지 속성 설계가 분류 성능과 안정성 향상에 기여함을 시사한다.

▸ **주제어**: 그래프 동형성 네트워크, 안드로이드, 악성코드, 엣지 특성, 사이버 보안, 분류

• First Author: Yeeun Lee, Corresponding Author: Eunjung Choi
*Yeeun Lee (leeyeeun0501@swu.ac.kr), Department of Information Security, Seoul Women's University
*Hyeona Jang (hyunnj17@swu.ac.kr), Department of Information Security, Seoul Women's University
*Hanseul Jung (right1magic@gmail.com), Department of Information Security, Seoul Women's University
**Eunjung Choi (chej@swu.ac.kr), Department of Information Security, Seoul Women's University
• Received: 2026. 01. 14, Revised: 2026. 02. 19, Accepted: 2026. 03. 02.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

모바일 악성코드의 규모와 복잡도는 지속적으로 증가하고 있으며[1], 정적 분석 기반의 악성코드 탐지 기법은 이에 대응하기 위해 더 정밀한 구조 분석 능력을 요구받고 있다. 안드로이드 악성코드 탐지 분야에서는 정적 분석 기반으로 애플리케이션의 실행 흐름과 행위 구조를 효과적으로 표현하기 위해 함수 호출 그래프(Function Call Graph, FCG)와 같은 그래프 기반 표현이 널리 활용되어 왔다 [2,4,15]. 기존 연구들은 FCG를 활용하여 안드로이드 악성코드의 구조적 특성을 효과적으로 분석해 왔으나, 호출 관계를 단순 연결로 취급하는 경향이 강해 edge 수준에서 의미적·구조적 정보를 명시적으로 모델링하는 연구는 상대적으로 제한적이었다.

특히 안드로이드 악성코드 분석 분야에서는 FCG를 활용한 대부분의 연구가 node 중심 표현에 기반하고 있으며, edge는 단순한 연결 정보 또는 방향성 정도로만 취급되는 경우가 많다 [2-5]. 이는 안드로이드 애플리케이션 구조상 API 호출 의미가 node 단위로 비교적 명확하게 정의될 수 있고, 정적 분석 환경에서 edge의 의미를 정량화하기가 상대적으로 어렵다는 점이 하나의 요인으로 작용했을 가능성이 있다. 그 결과, 안드로이드 플랫폼에서는 edge 수준의 의미적·구조적 특성을 독립적으로 설계하고 비교한 연구가 충분히 축적되지 않았다.

반면, 네트워크 침입 탐지 및 보안 로그 분석과 같은 다른 보안 플랫폼에서는 edge 정보를 명시적으로 활용하는 연구들이 보고되고 있다. 예를 들어, 대규모 네트워크 트래픽 데이터를 기반으로 한 침입 탐지 연구에서는 패킷 수, 전송량, 지속 시간과 같은 edge 통계 정보가 악성 행위 식별에 중요한 단서를 제공함이 실험적으로 입증되었다 [8]. 또한 계층적 GNN을 활용한 보안 인프라 로그 분석 연구에서는 edge 정보를 제외할 경우 탐지 성능이 유의미하게 저하됨이 보고되었다 [9]. 일반적인 그래프 신경망 연구에서도 edge attribute를 메시지 패싱 과정에 통합함으로써 그래프 표현력을 향상시킬 수 있음이 확인되었다 [10,14]. 이는 node 자체의 속성뿐 아니라, 관계(edge)에 부여된 속성이 탐지 성능에 직접적인 영향을 줄 수 있음을 의미하며, 관계 기반 정보가 그래프 구조 이해에 중요한 역할을 수행할 수 있음을 시사한다.

이처럼 edge 정보가 분류 성능에 유의미한 영향을 미친다는 기존 연구들은, 함수 간 호출 관계에 내재된 edge 수준 정보가 구조적 패턴 학습에 기여할 수 있음을 시사한다. 그러나 대부분의 연구는 노드 특성이나 단순 연결 구조에 초점을 두고 있으며, edge의 의미적·구조적 특성을 체계적으로 분석한 연구는 충분히 이루어지지 않았다. 특히 Android 악성코드 분류에서 edge attribute를 명시적으로 비교·분석한 사례는 상대적으로 제한적으로 보고되어 왔다. 이에 edge attribute의 중요성을 검증하기 위해 본 연구는 edge attribute를 메시지 패싱 과정에 직접 통합할 수 있는 GINE를 기반으로, 안드로이드 악성코드 FCG에서 다양한 edge 속성이 분류 성능에 미치는 영향을 체계적으로 분석한다.

본 연구에서는 MalNet 기반 FCG 데이터 셋 [13]을 활용하여 Distance, Direction, Timestamp, Frequency, Depth Difference, Betweenness의 총 6종 edge attribute를 설계하고, 이들이 악성코드 분류 성능에 미치는 상대적 영향을 정량적으로 비교·분석한다.

본 연구의 기여는 다음과 같이 요약할 수 있다. 첫째, 안드로이드 악성코드 FCG 분석에서 활용 가능한 다양한 edge attribute를 체계적으로 정리하고, 이를 GINE 모델에 적용할 수 있는 형태로 재구성하였다. 둘째, 동일한 실험 환경에서 총 6종 edge attribute를 개별 및 결합 방식으로 비교함으로써, edge attribute 기반 구조적 신호가 악성코드 분류 성능에 미치는 상대적 영향력을 정량적으로 분석하였다. 셋째, 실험 결과를 바탕으로 순서 기반 정보, 전역 중심성, 반복 호출 패턴 등 관계 기반 feature 설계 시 고려해야 할 요소를 도출하여, 향후 GNN 기반 안드로이드 악성코드 탐지 연구를 위한 실질적인 설계 지침을 제시한다. Android Malware Classification
Using an Edge-Centric Graph Isomorphism Network

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구를 정리하고, 3장에서는 본 연구에서 정의한 edge attribute의 설계 원칙과 계산 방법을 설명하고, 사용한 데이터셋, 모델 구조 및 실험 설정을 기술한다. 이어서 이러한 설정을 바탕으로 수행한 실험의 결과를 제시한다. 4장에서는 edge attribute별 실험 결과를 분석하고, 이를 종합적으로 논의한 후 향후 연구 방향을 제안한다.

II. Preliminaries

1. Related works

본 장에서는 그래프 기반 악성코드 탐지 연구를 세 범주로 나누어 정리한다.

먼저 Android 악성코드 분석에서 주로 사용된 node 중심 그래프 기반 접근을 살펴보고, 이어서 Android 외 보안

플랫폼에서의 edge 활용(edge-aware) 연구를 정리한다. 마지막으로 일반 그래프 신경망 연구에서의 edge attribute 모델링 흐름을 통해 본 연구의 위치를 명확히 한다.

1.1 Android Graph-based Malware Detection: Node-centric Approaches

Table 1. Android Malware Graph-Based Studies (Node-Centric Approaches)

Ref.	Graph type	Key feature design	Model
[2]	FCG	Graph-level Embedding Vector	SVM
[3]	FCG	Node-level Graph Centralities	GNN +JK
[4]	call graph	Permission, security-level, and code-based node features	GNN
[5]	FCG → BSS	Sensitive API-based node features	GNN
[15]	Weighted FCG	Community structure(node)	BayesNet (Best)

Android 악성코드 탐지 분야에서는 함수 호출 그래프(FCG)를 활용한 그래프 기반 정적 분석 기법이 다수 제안되어 왔다. 기존 연구들은 주로 API 또는 함수 단위를 node로 정의하고, node 수준의 의미적 특징이나 구조적 정보를 활용하여 악성코드를 분류하였다.

Ge 등[2]은 Android 애플리케이션의 함수 호출 관계를 그래프로 표현하고, 그래프 구조에서 추출한 통계적 특징을 이용한 악성코드 탐지 기법을 제안하였다.

Lo 등[3]은 Android FCG에 Jumping Knowledge(JK)를 적용한 GNN 기반 분류 모델을 제안하였으며, Drebin 및 MalNet-Tiny 데이터 셋에서 높은 분류 성능을 보고하였다.

Feng 등[4]은 호출 그래프에 권한, 보안 수준 등의 node 속성을 결합하여 GNN 기반 악성코드 탐지를 수행하였다.

He 등[5]은 민감 API 주변 서브그래프(BSS)를 구성하여 node 의미를 강화한 GNN 기반 탐지 기법을 제안하였다.

또한 Du 등[15]은 가중치가 부여된 함수 호출 그래프에서 커뮤니티 구조를 핵심 분석 단위로 활용한 Android 악성코드 탐지 기법을 제안하였다.

이들 연구는 그래프 구조를 활용한다는 공통점을 가지지만, edge 정보를 독립적인 비교 대상으로 설정하기보다는 node 표현을 보조적으로 강화하는 용도로 활용하였다.

1.2 Edge-aware Graph Learning in Other Security Domains

Table 2. Security-Platform Graph Studies Utilizing Edge Information

Ref.	Platform	Graph type	Key feature	Model
[8]	IoT	Flow Graph	Flow-level Feature	E-GraphSAGE
[9]	Enterprise Security Infra	Event Graph	Structural, temporal edge	Hierarchical GNN
[17]	Enterprise System	Knowledge Graph	Relation type	KG embedding +GNN
[18]	Host· Network	Provenance/ Network Graph	Relation type/Port-based attr.	MPNN
[19]	Enterprise Endpoint	Software Installation Graph	Dependency edge (audit event)	Graph LSTM + MLP (AE)

반면, Android 외 보안 플랫폼에서는 edge를 단순한 연결이 아닌, 시스템 행위나 인과 관계를 직접 표현하는 핵심적인 정보 단위로 활용하는 연구들이 지속적으로 보고되어 왔다. 특히 네트워크 침입 탐지, 시스템 로그 분석, IoT 보안과 같은 영역에서는 관계 자체가 공격 행위를 설명하는 주요 단서로 작용한다.

E-GraphSAGE[8]는 IoT 네트워크 플로우 그래프에서 패킷 수, 전송량, 연결 지속 시간과 같은 edge 통계 정보를 메시지 패싱 과정에 명시적으로 통합하여 악성 플로우를 탐지하였다. 실험 결과는 edge 기반 플로우 정보를 활용한 그래프 신경망 구조가 네트워크 플로우 기반 침입 탐지에 효과적임을 보여준다.

HiSec [9]은 사이버 레인지 기반 엔터프라이즈 보안 인프라에서 수집된 보안 로그를 바탕으로 자산 간 시스템 활동을 계층적 그래프로 모델링하는 GNN 구조를 제안하였다. 이는 edge 정보의 제거가 탐지 성능 저하로 이어짐을 통해 관계 기반 구조 표현의 중요성을 보였다.

SHADEWATCHER [17]는 Linux Audit 로그로부터 생성된 provenance 그래프를 지식 그래프로 확장하고, 시스템 엔터티 간 상호작용을 관계 단위의 edge로 모델링하였다. TransR 기반 관계 임베딩과 관계 조건부 attention 메시지 패싱을 통해 엔터티 간 상호작용을 학습하며, 탐지 단위를 엔터티 간 상호작용으로 설정하였다.

EdgeTorrent[18]는 provenance 및 네트워크 그래프에서 각 edge를 이벤트 단위의 상호작용으로 모델링하고, edge에 부여된 속성(edge-associated information)을 메시지 패싱 신경망(MPNN)의 입력으로 사용한다. edge feature는 데이터 셋에 따라 관계 타입이나 포트 조합으로 구성되며 node feature와 결합되어 node 임베딩에 반영되고, 이후에는 해당 임베딩의 시간적 변화를 기반으로

이상 행위를 탐지한다.

SIGL[19]은 엔터프라이즈 엔드포인트 환경에서 수집된 audit log를 기반으로 소프트웨어 설치 과정을 software installation graph로 모델링하고, 각 audit log event를 시스템 엔터티 간 dependency edge로 변환하여 그래프를 구성한다. 이러한 관계 구조를 바탕으로 SIGL은 graph LSTM을 encoder로 사용하는 그래프 오토인코더를 학습하여, 정상 설치 과정에서 나타나는 관계 및 순서 패턴을 구조적으로 모델링하고 이상 설치 행위를 탐지한다.

1.3 Edge Feature Modeling in General Graph Neural Networks

Table 3. Representative Edge Feature Types Used in Previous Studies

Ref.	Feature Type	Model	Learning Type
[8]	Weight	E-GraphSAGE	Message passing
[9]	System Activity	Hierarchical GNN	Hierarchical
[10], [11]	Direction	EGNN, HFGIN	Adaptive, Message passing
[11]	Temporal Dependency	HFGIN	Message passing
[12]	Position & Order	improved R-GCN	Order-aware
[16]	Frequency	DaDiDroid	Graph statistics

일반적인 그래프 신경망 연구에서도 edge feature의 중요성은 점차 강조되고 있다.

Gong과 Cheng[10]은 다차원 edge feature와 방향성을 메시지 패싱 과정에 통합하는 EGNN을 제안하여, edge 방향 정보가 그래프 표현력 향상에 기여함을 보였다.

Alzahrani 등[11]은 edge 정보를 구조적 관계로 활용한 GIN 기반 메시지 패싱을 통해 블록체인 트랜잭션 그래프에서 node와 edge 정보를 함께 반영하는 GIN 기반 모델을 제안하여 복잡한 관계 의존성을 효과적으로 표현하였다.

DaDiDroid[16]는 Android 플랫폼에서 edge 정보를 명시적으로 고려해 weighted directed call graph를 구성하고 API 호출 간의 방향성과 빈도를 edge 가중치로 표현한 뒤 이를 그래프 수준의 통계적 특징으로 요약하여 전통적인 분류기에 적용하였다. 해당 연구는 호출 빈도와 방향성이 악성코드 판별에 유의미함을 보였으나, edge 정보를 메시지 패싱 과정에서 직접 학습하는 GNN 기반 분석으로 확장되지는 않았다.

또한 API 호출 시퀀스를 기반으로 한 위협 탐지 연구[12]에서는 호출 순서와 위치 정보를 그래프 edge 수준에서 반영하여 애플리케이션 행위의 구조적 정보를 모델링하였다.

이러한 연구들은 edge feature가 그래프 학습에서 중요한 정보 단위로 기능할 수 있음을 반복적으로 입증하고 있다.

1.4 Summary and Positioning of This Study

기존 연구를 종합하면, Android 악성코드 그래프 분석에서는 node 중심 접근이 주류를 이루어 왔으며[2-5,15], edge 특성은 주로 보조적 구조 정보로 제한되어 왔다. 반면, 다른 보안 플랫폼 및 일반 GNN 연구에서는 edge feature의 중요성이 실험적으로 입증되었으나[8-12], 이러한 접근이 Android FCG 환경에 체계적으로 적용·비교된 사례는 부족하다.

본 연구는 이러한 연구 공백을 메우기 위해, Android FCG 환경에서 호출 관계의 구조적·순차적·통계적 특성을 다양한 edge attribute로 정의하고, 이를 동일한 실험 조건 하에서 비교·분석한다. 이는 Android 악성코드 그래프 분석에서 edge 특성의 기여도를 체계적으로 규명한 연구라는 점에서 기존 연구와 차별성을 가진다.

III. The Proposed Scheme

1. Experimental Procedure

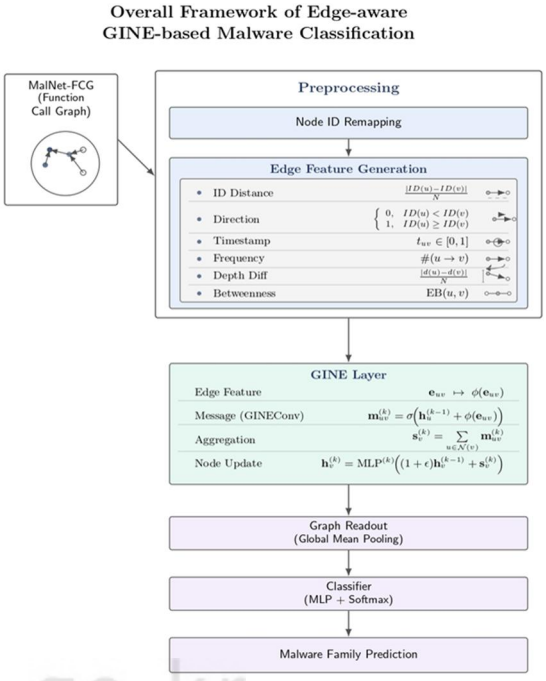

Fig. 1. Overview of the Experimental Procedure

본 연구의 전체 실험 과정은 Fig. 1과 같다. 먼저 MalNet-Tiny 데이터 셋으로부터 악성코드 및 양성 그래프를 로드한 뒤, 그래프 내 node 식별자를 일관성 있게 정렬하기 위해 node ID remapping 과정을 수행하였다. Node ID remapping은 각 그래프에서 노드 ID를 0부터 시작하는 연속 정수로 재할당하는 방식으로 수행하였다. 구체적으로, 그래프에 포함된 원본 노드 ID를 오름차순으로 정렬한 후, 해당 그래프의 노드 수를 N이라고 할 때 0 ~N−1 범위의 인덱스를 부여하였다. 이 과정은 노드 식별자만 변환하는 절차로, 그래프의 연결 구조는 변경되지 않는다. 이후, 그래프의 구조적·관계적 정보를 반영하기 위해 ID Distance, Frequency, Betweenness, Timestamp, Depth Difference, Direction 등을 포함한 다양한 edge attribute를 생성하였다.

전처리가 완료된 그래프는 GINE 모델에 입력되어 edge attribute 기반의 메시지 패싱과 node 업데이트를 통해 표현 학습을 수행한다. 학습된 모델의 성능은 분류 정확도를 기준으로 비교 및 분석하며, 최종적으로 악성 패밀리 분류 결과를 도출한다.

2. Dataset

본 연구에서는 Android 악성코드 그래프 분석 모델의 성능을 평가하기 위해 MalNet 데이터 셋[13]의 서브셋인 MalNet-Tiny를 실험 대상으로 선정하였다. 해당 데이터 셋에는 addisplay, adware, trojan, downloader 및 benign이라는 4개의 악성 패밀리와 1개의 양성 패밀리가 포함된다. 총 5개 클래스에 대해 균등하게 구성된 5,000개의 함수 호출 그래프(FCG)를 사용하였다. 그래프 규모는 평균 1,410.32개의 node와 2,859.95개의 edge를 포함하며, 최소 5개에서 최대 4,994개의 node로 구성된다. 또한 전체 그래프에 대해 14,299,744개의 함수 호출(call) 관계가 존재하여, 전처리 과정에서 그래프 정규화 및 구조적 일관성 확보가 필요하였다. 그래프 학습을 위한 node feature는 모든 실험에서 동일하게 적용하였으며, node의 degree와 32차원 node embedding을 결합하여 사용하였다. 전체 통계는 Table 4와 같다.

Table 4. Dataset and Graph Statistics

Category	Statistic	Value
Class Distribution	addisplay	1,000
	adware	1,000
	benign	1,000
	downloader	1,000
	trojan	1,000
Total Graphs	count	5,000
Nodes per Graph	Avg	1,410.32
	Min	5
	Max	4,994
Edges per Graph	Avg	2,859.95
	Min	4
	Max	20,096
Structural Stats	Total self-loops	10,173
	Isolated nodes	3,581,375
Relation Types	call edges	14,299,744

3. Edge Attribute Design

Table 5. Edge Attribute Design

Feature	Description	Purpose
ID Distance	ID Differences between Calling Nodes	Capture long-range call relations
Direction	Call Direction	Encode control -flow direction
Timestamp	Edge Occurrence Order	Reflect call-order patterns
Frequency	Frequency of Repeated invocations along the same edge	Detect repetitive/loop behaviors
Depth Difference	Depth Differences between Nodes	Model hierarchical transitions
Betweenness	Betweenness Centrality	Identify critical transit edges

본 연구에서는 함수 호출 그래프(FCG)의 구조적·동적 특성을 반영하기 위해 여섯 종류의 edge attribute를 설계하였다. 모든 속성은 단일 스칼라 값으로 인코딩되며, node 특성(degree)은 모든 실험에서 동일하게 유지하여 edge-level 정보가 분류 성능에 미치는 영향만을 분리하여 분석할 수 있도록 하였다. Table 5는 각 속성의 정의와 계산 방식을 요약한 것이다.

3.1 ID Distance

$$IDDist(e=(u,v)) = \frac{|ID(u)-ID(v)|}{N}$$

ID Distance는 노드의 상대적 위치 정보를 보조적으로 활용한 기존 연구의 관점[14]을 참고하여 정의한다. ID Distance는 재매핑된 노드 ID 간의 절대 차이에 기반한 구조적 지표로, 각 edge ($u \rightarrow v$)에 대해 두 node ID의 절대 차이 $|u-v|$를 계산한 뒤, 그래프의 node 수 N으로 정규화한 값을 사용한다. 이 정규화는 그래프 규모 차이에 따른 값의 편차를 보정하기 위한 것이다.

3.2 Direction

$$dir(e=(u,v)) = \begin{cases} 0, ID(u) < ID(v) \\ 1, ID(u) \geq ID(v) \end{cases}$$

Direction은 이전 연구[10]의 edge attribute로 취급하는 관점을 참고하여 정의하였다. Direction은 재매핑된 노드 ID의 대소 관계에 기반한 구조적 이진 지표로, e=(u,v)에 대해 ID(u)와 ID(v)의 크기 비교 결과에 따라 이진값을 부여한다. 이를 통해 ID가 증가하는 방향과 감소하는 방향의 분포를 단순화하여 표현한다.

3.3 Timestamp

$$ts(e_k) = \frac{k-1}{|E|-1}$$

Timestamp는 그래프 내 edge 목록에서 k번째 edge e_k의 상대적 위치를 전체 edge 수 |E|로 정규화한 값이다.

본 연구는 HiSec [9]의 edge 수준 시간 정보 활용 방식에 착안하여, 정적 edgelist 환경의 Android FCG에 적용 가능한 timestamp 정의를 설계하였다. 연구에 사용된 데이터 셋은 정적 edgelist 형태로 제공되며 edge의 실제 실행 시간 정보가 포함되지 않으므로 본 연구에서는 edge의 상대적 순서를 timestamp로 정의하였다. 해당 순서는 정적 FCG 환경에서 제한적인 순차 신호를 제공하기 위한 관계 속성으로 활용된다.

3.4 Frequency

Frequency는 동일 방향의 함수 호출 쌍 ($u \rightarrow v$)이 그래프 내에서 관찰되는 빈도를 나타내며, 호출 관계의 상대적 강도를 표현하는 edge attribute이다. 함수 호출 빈도의 차이는 edge 분포의 비균일성을 유발하여 node 집합 간 연결 밀도 차이를 형성할 수 있으며, 이러한 구조적 특성은 그래프 내 커뮤니티 구조의 존재를 시사한다 [15].

3.5 Depth Difference

$$depth\_diff(u,v) = \frac{|depth(u)-depth(v)|}{N}$$

Depth Difference는 각 그래프에서 진입 차수가 0인 노드를 루트로 설정하고 BFS 기반 계층 레벨을 계산한 뒤, 두 node 간 depth 차이의 절대값을 정규화하여 정의하였다. 진입 차수 0인 노드가 여러 개인 경우, 각 후보를 루트로 수행한 BFS 결과 중 최소 depth를 사용하였다. 진입 차수 0인 노드가 존재하지 않는 경우에는 임의의 노드(최소 ID)를 루트로 설정하였다.

최근 그래프 신경망 연구에서는 node의 구조적 위치나 계층적 레벨에서 유도된 구조적 정보가 그래프 표현 학습에 기여할 수 있음이 보고되었으며 [14], 이러한 depth 기반 정보는 본 연구와 같이 edge 단위의 관계 특성으로 활용될 경우 전역적 구조 차이를 반영하는 보조적 구조 신호로 작용할 수 있다.

3.6 Betweenness

$$bw(u,v) = edge\_betweeness\_centrality(u,v)$$

Betweenness는 해당 edge가 전체 그래프 내에서 경로의 "중간 경유 지점"으로 사용되는 비율을 나타낸다. 이전 관련 연구[15]에서는 함수 호출 그래프의 구조적 정보를 활용하여 기존 제어 흐름 그래프 기반 기법 대비 향상된 악성코드 분류 성능을 보고하였다. 해당 연구에서는 함수 호출 그래프를 커뮤니티 구조로 분할하는 과정에서 Betweenness와 같은 구조적 지표를 사용하였으며, 이를 통해 그래프의 구조적 특성이 분류에 유효함을 보였다. 이러한 선행 연구를 바탕으로 본 연구에서도 그래프의 구조적 정보를 반영할 수 있는 지표 중 하나로 Betweenness를 edge 속성으로 포함하였다.

4. Experimental Environment

실험은 PyTorch 2.2.0, PyG 2.5.3, Python 3.12 환경에서 수행되었다. 본 연구는 MalNet 기반 5개 클래스(addisplay, adware, benign, downloader, trojan)로 구성된 총 5,000개의 함수 호출 그래프(FCG)를 사용하였다. 원본 FCG에는 최대 4,994개의 node와 20,096개의 호출 관계가 포함되는 등 구조적 복잡성이 높기 때문에, 전처리 과정에서는 각 그래프의 node를 0부터 시작하는 연속된 정수로 재매핑하고, 모든 실험에서 공통적으로 degree를 기본 node 특성으로 사용하여 입력 표현을 정규화하였다. 본 실험에서는 전체 데이터셋을 클래스 비율을 유지한 상태로 훈련 80%, 테스트 20%로 분할하여 사용하였다.

MalNet-Tiny[13]는 Android 애플리케이션을 함수 호출 그래프(FCG) 형태로 제공하는 데이터셋이므로, 그래프 기반 분석 기법을 적용하였다. 모델은 GINE를 기반으로 설계하였다. 각 node는 degree(1차원)와 node ID 임베딩(32차원)을 결합하여 표현되며, 두 개의 GINEConv 레이어(hidden dimension = 64), Batch Normalization, ReLU 활성화 함수를 포함한 인코더를 통해 구조 정보를 학습한다. 이후 global mean pooling을 통해 그래프 수준 임베딩을 생성하고, 최종적으로 fully connected layer를 통해 5개 클래스에 대한 확률을 출력한다. 실험 전반에서 학습 설정은 동일하게 유지하였으며, Adam optimizer(learning rate = 0.001), batch size 16, epoch 10을 모든 조건에서 공통으로 적용하였다. 이를 통해 모델 구조나 학습 설정이 결과에 미치는 영향을 제거하고 edge attribute만의 효과를 순수하게 측정할 수 있도록 하였다.

변인 통제를 위해 다음과 같은 절차를 적용하였다. 첫째, 모든 실험에서 동일한 stratified 데이터 분할을 사용하여 클래스 비율을 일관되게 유지하였다. 둘째, 모델 아키텍처, 하이퍼파라미터를 고정함으로써 실험 간 성능 차이가 학습 설정의 변동에서 기인하지 않도록 하였다. 셋째, 비교 실험에서는 edge attribute를 제외한 입력 데이터 및 전처리 과정을 모두 동일하게 유지하고, 각 실험마다 하나의 attribute만을 교체하여 적용하였다. 이를 통해 개별 attribute가 GNN의 구조적 표현 학습에 미치는 기여도를 독립적이고 정량적으로 평가할 수 있었다.

본 연구의 목적은 다양한 GNN 아키텍처 간 성능 경쟁 비교가 아니라, 동일한 edge-aware 메시지 패싱 프레임워크 내에서 edge attribute 설계가 분류 성능에 미치는 영향을 분리하여 분석하는 데 있다. 따라서 모델 아키텍처는 고정하고 edge attribute만을 교체하여 실험하였다.

5. Performance Comparison Across Individual Edge Attributes

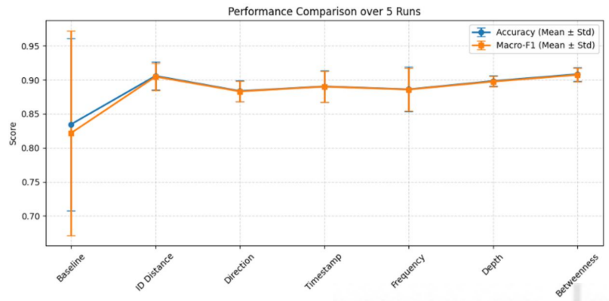

Fig. 2. Performance Comparison of Different Edge Attributes

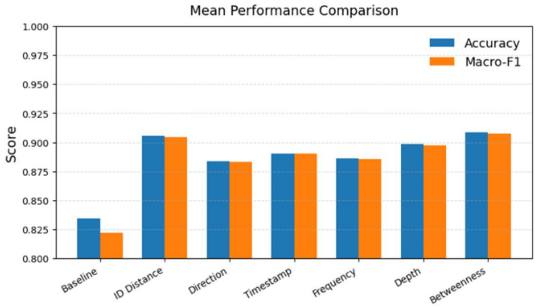

Fig. 3. Performance Comparison of Different Edge Attributes (Bar Chart)

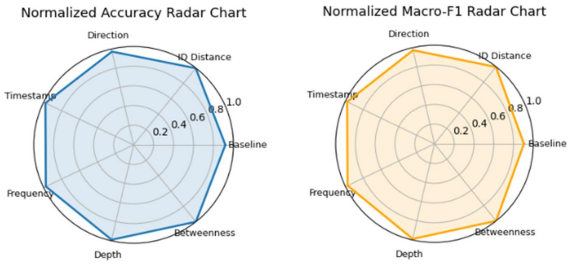

Fig. 4. Normalized Acc Radar Chart
Fig. 5. Normalized Macro-F1 Radar Chart

본 절에서는 Baseline(GINE + dummy edge attribute)을 기준으로, 총 6개의 edge attribute를 적용한 실험 결과를 비교한다. Fig. 2는 각 edge attribute의 평균 정확도를 비교한 결과를 제시하며, Fig. 3은 edge attribute별 Acc 및 F1을 시각화한 결과를 보여준다. 또한 Fig. 4와 Fig. 5는 이를 정규화하여 비교한 성능 결과를 나타낸다.

각 edge attribute에 대해 서로 다른 random seed를 적용하여 5회 반복 실험을 수행하고 Acc 및 Macro-F1의 평균과 표준편차를 산출하였다. Table 6에서 보이는 바와 같이 Baseline 모델은 평균 Acc 0.8347로 가장 낮은 성능을 기록하였으며, 실행 간 편차 역시 매우 커(Acc_Std = 0.1269) 일관성이 부족한 것으로 확인되었다. 반면, 제안된 edge attribute 기반 모델들은 전반적으로 Baseline 대비 유의미하게 향상된 정확도를 보였다. Fig. 6은 Baseline과 ID Distance 적용 모델의 클래스별 오분류 패턴 차이를 혼동행렬로 비교한 것이다.

ID Distance(Acc 0.9058, F1 0.9047), Depth Difference (Acc 0.8984, Acc_Std 0.0077), Betweenness(Acc 0.9084, F1 0.9074)는 높은 정확도와 함께 안정적인 편차를 나타냈으며, 특히 Betweenness가 가장 높은 성능을 기록하였다. Direction(Acc 0.8836), Timestamp(Acc 0.8903), Frequency(Acc 0.8861) 역시 중간 이상의 성능을 보였으나, Frequency는 비교적 편차가 큰 것으로 나타났다(Acc_Std = 0.0329).

Baseline과의 성능 향상폭을 정량적으로 비교하였을 때

제안된 edge attribute 모두 Baseline 대비 약 5~7%p 수준의 성능 개선을 달성하였다.

이러한 결과는 node 수준 통계만을 활용하는 Baseline과 달리, edge 관계 기반 정보가 악성코드 패밀리 분류 성능 개선에 기여할 수 있음을 시사한다. 특히 Betweenness 및 Depth Difference와 같은 구조 기반 지표의 성능 우세는 패밀리별 호출 구조 차이가 분류에 유효할 가능성을 뒷받침한다.

아울러 Baseline은 실행 간 성능 편차가 매우 컸던 반면, edge attribute 기반 모델들은 전반적으로 낮은 표준편차를 유지하여, edge 정보 활용이 분류 정확도뿐 아니라 모델의 신뢰성과 일관성 향상에도 기여함을 확인할 수 있었다.

Table 6. Performance Comparison Results by Edge Attribute

Feature	Acc (Mean)	Acc (Std)	F1 (Mean)	F1 (Std)
Baseline	0.8347	0.1269	0.8218	0.1504
ID Distance	0.9058	0.0202	0.9047	0.0200
Direction	0.8836	0.0152	0.8830	0.0153
Timestamp	0.8903	0.0233	0.8900	0.0226
Frequency	0.8861	0.0328	0.8857	0.0316
Depth Difference	0.8984	0.0077	0.8975	0.0078
Betweenness	0.9083	0.0096	0.9073	0.0099

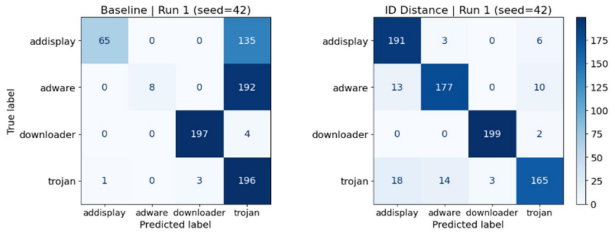

Fig. 6. Confusion Matrix Comparison: Baseline vs. ID Distance Methods

6. Performance Comparison of Top Edge Attributes and All-edge Combination

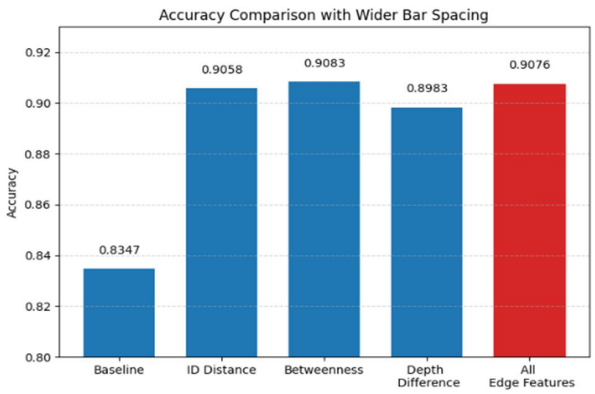

Fig. 7. Comparative Performance of All Edge Attributes, Baseline, and Top-3 Edge Attributes

앞선 실험에서 성능이 가장 우수한 상위 3개(Top-3) edge attribute를 선정하고, 이를 전체 edge를 입력으로 한 결과와 비교하는 실험을 수행하였다. 그 결과, 최종 정확도와 Macro-F1 점수 각각 0.9076, 0.9073으로 모두 0.9를 상회하는 우수한 성능을 보였다. 또한 해당 결과는 6종의 edge attribute를 결합하여 사용했을 때 중요도가 높은 attribute를 단독으로 사용했을 때의 성능과 유사한 수준임을 보여준다. Fig. 7은 Baseline, Top-3(ID Distance, Betweenness, Depth Difference), 그리고 All-edge 조합의 성능을 비교한 결과이다.

Ⅳ. Conclusion

종합적으로 본 연구는 edge attribute를 단일 부가 요소가 아닌 독립적인 설계 변수로 정의하고, 그 효과를 체계적으로 비교·분석하였다. 이를 통해 함수 호출 그래프에서 엣지 수준의 관계 정보를 명시적으로 활용할 경우 Android 악성코드 분류 성능과 모델 안정성이 유의미하게 향상될 수 있음을 실험적으로 확인하였다. 이러한 결과는 관계 기반 정보가 node 중심 표현을 보완하는 핵심 요소로 작용할 수 있음을 시사한다.

1. Discussion

실험 결과, Baseline은 평균 Acc 약 0.83과 높은 표준편차를 보여, 단순 degree 기반 node 특성만으로는 FCG의 구조적 차이를 안정적으로 설명하기 어렵다는 한계를 드러냈다. 반면 edge attribute 기반 모델들은 전반적으로 Baseline 대비 유의미한 성능 향상을 보였으며, 특히 ID Distance, Betweenness, Depth Difference가 가장 높은 성능을 기록하였다. 한편, Direction은 이진 값으로 정의되었지만 메시지 패싱 과정에서 노드 표현과 상호작용하며 고차 구조 패턴을 형성할 수 있다. 따라서 단순 이진 특성이라도 그래프 수준 표현 학습 과정에서 복합적인 구조 정보를 반영할 가능성이 있다. 이는 악성코드 동작의 핵심 정보가 개별 node보다 관계(edge)에 더 많이 내재되어 있고, 순서적·전역적·계층적 구조 신호가 분류 성능을 효과적으로 강화할 수 있음을 시사한다.

추가적으로, 개별 edge attribute의 기여도를 넘어, 서로 다른 edge attribute 간의 상호 보완 효과와 결합 사용의 실효성을 검증하기 위해 총 6종의 edge attribute를 모두 결합하여 모델 입력으로 적용한 추가 실험을 수행하였다. 총 10회의 학습(run) 결과 전체 edge attribute를

적용한 모델은 정확도와 Macro-F1 모두 0.9를 상회하는 성능을 보였으며, 이는 상위 3개의 핵심 edge attribute를 단독으로 사용했을 때와 유사한 수준이다. 이러한 결과는 단일 edge attribute 실험을 통해 확인된 주요 구조 신호가 전체 edge attribute 결합 환경에서도 효과적으로 보존되며 복수의 structural edge attribute design을 함께 활용하더라도 성능 저하 없이 안정적인 분류가 가능함을 시사한다.

2. Limitations

본 연구에서 사용한 데이터셋은 정적 edgelist 형식이며 node ID가 리인덱싱된 식별자일 가능성이 있어 ID Distance, Direction, Timestamp, Frequency와 같이 ID 또는 순서 정보에 기반한 일부 edge attribute는 의미론적 해석에 제약을 가지는 휴리스틱 특성으로 해석될 수 있다. 따라서 본 연구의 edge attribute는 호출 의미를 직접 모델링하기보다는, 정적 FCG 환경에서 관찰 가능한 구조적·순서 기반 신호를 반영한 설계 변수로 해석되어야 한다.

Timestamp는 파일 파싱 및 전처리 단계에서 부여되는 일관된 순번에 기반하며, 단일 정적 그래프 스냅샷 내에서 호출 관계의 순차적 패턴을 표현하는 order-based edge attribute로 해석된다.

Depth Difference의 정규화에는 최대 깊이($D_{\max}$)를 사용하는 대안도 고려할 수 있으나, 본 연구에서는 그래프 크기 변화에 따른 스케일 일관성을 유지하고 서로 다른 FCG 간 비교 가능성을 확보하기 위해 node 수 N을 정규화 상수로 선택하였다. 또한 동일 node쌍 edge의 중복 저장이 제한될 수 있어 Frequency의 정보성이 축소될 가능성이 있다. 향후 연구에서는 실행 로그 기반의 동적 호출 정보와 원래의 함수·패키지 메타데이터를 결합함으로써, 보다 의미론적으로 정합적인 edge attribute 설계 및 검증을 수행할 필요가 있다.

3. Future Work Directions

본 연구에서 확인된 structural edge attribute design의 효과와 한계를 바탕으로 향후 연구에서는 단일 edge attribute에 의존하는 방식에서 벗어날 필요가 있다. 나아가 거리·계층·중심성·순서 정보 등 다양한 structural edge attribute design을 결합하거나 멀티모달 방식으로 활용함으로써 악성 행위의 복합적 구조를 보다 정밀하게 포착할 수 있을 것이다.

REFERENCES

[1] Kaspersky, “Kaspersky report: Attacks on smartphones increased in the first half of 2025”, Kaspersky (Press Releases) https://www.kaspersky.com/about/press-releases/kaspersky-report-attacks-on-smartphones-increased-in-the-first-half-of-2025

[2] Xiuting Ge, Ya Pan, Yong Fan, Chunrong Fang, “AMDroid: Android Malware Detection Using Function Call Graphs,” 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp.71-77, Sofia, Bulgaria, July 2019. DOI: 10.1109/QRS-C.2019.00027.

[3] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, “Graph Neural Network-based Android Malware Classification with Jumping Knowledge,” 2022 IEEE Conference on Dependable and Secure Computing (DSC), pp.1-9 , Edinburgh, United Kingdom, Jun. 2022. DOI: 10.1109/DSC54232.2022.9888878

[4] P. Feng, J. Ma, T. Li, X. Ma, N. Xi, and D. Lu, “Android Malware Detection Based on Call Graph via Graph Neural Network,” 2020 International Conference on Networking and Network Applications (NaNA), pp. 368–374, Haikou City, China, Dec. 2020. DOI: 10.1109/NaNA51271.2020.00069

[5] Y. He, Y. Liu, L. Wu, Z. Yang, K. Ren, and Z. Qin, “MsDroid: Identifying Malicious Snippets for Android Malware Detection,” IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 3, pp. 2025–2039, May–Jun. 2023. DOI: 10.1109/TDSC.2022.3168285

[6] Y. Gao, H. Hasegawa, Y. Yamaguchi, and H. Shimada, “Malware Detection by Control-Flow Graph Level Representation Learning With Graph Isomorphism Network,” IEEE Access, vol. 10, pp. 111830–111841, Oct. 2022. DOI: 10.1109/ACCESS.2022.3215267

[7] B. Wu, Y. Xu, and F. Zou, “Malware Classification by Learning Semantic and Structural Features of Control Flow Graphs,” 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Shenyang, China, pp. 540-547, Oct. 2021. DOI: 10.1109/TrustCom53373.2021.00084

[8] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, “E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT,” NOMS 2022 – 2022 IEEE/IFIP Network Operations and Management Symposium, pp. 1-9 , Budapest, Hungary, Apr. 2022. DOI: 10.1109/NOMS54207.2022.9789878

[9] L. Xu, X. Lin, J. Li, M. Bai, and L. Wang, “HiSec: Towards Cyber Threat Correlation and Discovery Based on Hierarchical Graph Neural Networks,” 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 369–378, Exeter, United Kingdom, Nov. 2023. DOI: 10.1109/TrustCom60117.2023.00067

[10] L. Gong and Q. Cheng, “Exploiting Edge Features for Graph Neural Networks,” 2019 IEEE/CVF Conference on Computer

Vision and Pattern Recognition (CVPR), pp. 1-10, Long Beach, CA, USA, Jun. 2019. DOI: 10.1109/CVPR.2019.00943

[11] F. Alzahrani, M. Y. Idris, M. F. Rohani, and R. Budiarto, "Hyperledger Fabric Graph Isomorphism Network for Conflict Transactions Detection in Multi-Version Concurrency Control," IEEE Access, vol. 13, pp. 131314-131333, Jul. 2025. DOI: 10.1109/ACCESS.2025.3589165

[12] W. Sun, "Malicious Software Identification Based on Deep Learning Algorithms and API Feature Extraction," Journal of Cloud Computing, vol. 2025, art. no. 10, Mar. 2025. DOI: 10.1186/s13635-025-00197-4

[13] Scott Freitas, Yuxiao Dong, Joshua Neil, and Duen Horng Chau," A Large-Scale Database for Graph Representation Learning," 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Datasets and Benchmarks Track, pp. 1-13, Virtual Conference, December 2021. DOI: 10.48550/arXiv.2011.07682

[14] H. Cui, Z. Lu, P. Li, and C. Yang,"On Positional and Structural Node Features for Graph Neural Networks on Non-attributed Graphs," 31st ACM International Conference on Information and Knowledge Management (CIKM), pp. 1-10, Atlanta, GA, USA, Oct. 2022.DOI: 10.48550/arXiv.2107.01495.

[15] Y. Du, J. Wang, and Q. Li, "An Android Malware Detection Approach Using Community Structures of Weighted Function Call Graphs," IEEE Access, vol. 5, pp. 17478-17486, Jun. 2017. DOI: 10.1109/ACCESS.2017.2720160

[16] M. Ikram, P. Beaume, and M. A. Kaafar,"DaDiDroid: An Obfuscation-Resilient Tool for Detecting Android Malware via Weighted Directed Call Graph Modelling," 16th International Joint Conference on e-Business and Telecommunications (ICETE), pp. 211-219, Prague, Czech Republic, July 2019. DOI: 10.5220/0007834602110219

[17] Jun Zeng, Xiang Wang, Jiahao Liu, Yinfang Chen, Zhenkai Liang, Tat-Seng Chua, and Zheng Leong Chua,"SHADEWATCHER: Recommendation-guided Cyber Threat Analysis using System Audit Records," 2022 IEEE Symposium on Security and Privacy (SP 2022), pp. 489-506, May 2022.DOI: 10.1109/SP46214.2022.9833669

[18] I. J. King, X. Shu, J. Jang, K. Eykholt, T. Lee, and H. H. Huang, "EdgeTorrent: Real-time Temporal Graph Representations for Intrusion Detection," 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '23), pp. 77-91, Oct. 2023. DOI: 10.1145/3607199.3607201

[19] X. Han, T. Pasquier, X. Yu, D. Li, J. Rhee, J. Mickens, M. Seltzer, and H. Chen,"SIGL: Securing Software Installations through Deep Graph Learning," 30th USENIX Security Symposium (USENIX Security '21), pp. 2345-2362, Aug. 2021.

Authors

Yeeun Lee is currently pursuing the B.S. degree in Information Security at Seoul Women's University, Seoul, Republic of Korea. Her research interests include IoT security, malware analysis, and AI-based automation.

Hyeona Jang is an undergraduate student in the Department of Information Security at Seoul Women's University. Her research interests include privacy protection and malware analysis.

Hanseul Jung is an undergraduate student in the Department of Information Security at Seoul Women's University, Seoul, South Korea. Her research interests include malware analysis, IoT security, and cyber threat intelligence, with an emphasis on understanding adversarial behaviors and attack patterns.

Eunjung Choi received the B.S. degree in Computer Science from Seoul Women's University, Seoul, Republic of Korea, in February 1997, the M.S. degree in Computer Science from the Graduate School of Seoul Women's University in February 2000, and the Ph.D. degree in Computer Science from the Graduate School of Seoul Women's University in August 2005. She has been a Professor in the School of Intelligent Information Security at Seoul Women's University since March 2006. Her research interests include big data, artificial intelligence, and malware.