# GCN 과 자카드 유사도를 활용한 악성코드 탐지 기법 연구

박양훈*, 김봉현*, 최선오**
*호남대학교 컴퓨터공학과
**전북대학교 소프트웨어공학과
didgnsdl6784@naver.com, kibohy4853@gmail.com, suno7@jbnu.ac.kr

# A Study for Detecting Malware Using GCN

Yang-Hoon Park*, Bong-Hyun Kim*, Sunoh Choi**
* Dept. of Computer Engineering, Honam University
** Dept. of SoftwareEngineering, Jun-Book University

**요 약**

기술이 발전함에 따라 악성코드 또한 함께 발전하여 보안 위협이 증가되고 있다. 특히 PowerShell 과 같은 스크립트 언어를 사용하여 포렌식이 어려운 Fileless 악성코드가 지속적으로 증가하고 있다. 이에 본 논문에서 PowerShell 데이터셋을 활용하여 기존 패턴은 탐지할 수 없는 한계점을 가진 시그니처 또는 휴리스틱 기반 탐지 기법을 보완하여 기존의 악성코드들을 학습 및 새로운 악성코드를 추측하는 것이 가능한 심층 학습 기술, Graph Convolutional Networks 과 자카드 유사도를 활용하여 기존의 방식에 비해 더 효율적으로 탐지를 이루어 내는지를 판단해보려 한다.

## 1. 서론

최근에는 기존의 사용자가 직접 파일을 실행하거나 추출하여 시스템에 침투했던 공격방식과는 달리 공격자가 Fileless 형태로 공격 대상 단말의 조작 권한을 획득하는 등의 방법이 주로 사용되고 있다. 현재 백신 소프트웨어는 보고되었거나 파일 내의 문자열 패턴 등을 기준으로 하는 시그니처(Signature) 또는 휴리스틱(Heuristics) 기반으로 악성 여부를 검사하게 된다. 그러나 이 같은 탐지 방법은 변종 악성코드 또는 제로데이(ZeroDay) 같은 신종 공격에 대응하기 어렵다는 단점이 있다[1].

이러한 문제를 해결하기 위해 최근 활발하게 연구되고 있는 머신러닝(Machine Learning)과 딥러닝(Deep Learning)과 같은 인공지능 기법을 사용하여 악성코드를 탐지하고자 하는 노력이 이루어지고 있다. 이런 기법들은 컴퓨터 비전, 음성인식, 자연어 처리, 음성 및 신호처리 등에서 적용되어 좋은 결과를 보여주고 있다[2].

Fileless 형태의 공격 역시 나날이 증가하고 있으며, PowerShell 과 같은 스크립트 언어를 통해 주로 사용자가 실행을 시키는데 이는 실행 흔적을 거의 남기지 않아 포렌식 작업을 어렵게 하는 특징을 가지고 있다. 아래 그림과 같이 Fileless 공격 시장은 계속 커질 것으로 예상된다[3].


(그림 1) DATA BRIDGE Market Research 에서 조사한 Fileless Malware 전망

이에 본 논문에서는 마이크로소프트의 대표적인 스크립트 언어인 PowerShell 데이터셋을 활용하여 2 가지 파트로 실험을 진행한다.

첫번째는 정상과 악성 데이터셋을 이용하기전, 전처리 작업으로 악성 데이터셋에 사용되는 주 기법인 코드 난독화를 Revoke-Obfuscation 모듈을 통해 무력화 시킨 후 프리퀀시(Frequency) 데이터로 특징을 추출한다[4]. 그 다음 심층 학습 기법인 GCN(Graph Convolutional Network)과 자카드 유사도(Jaccard Similarity) 기법을 사용하되, 자카드 유사도의 Top-K 값이 탐지율에 영향을 미치는지 확인한다. 이어서 논문에서 제안하는 이 기법이 효율적으로 악성코드를 탐지 및 분석하는지 확인하려고 한다.

## 2. 사전지식 및 관련연구

### 2.1 파일리스 악성코드(Fileless Malware)

파일리스(Fileless)란 공격 대상의 단말에 악성 파일을 저장시키지 않고, 시스템 메모리에 바로 로드되어 실행 되는 것을 의미하며, 해당 방식을 이용하여 실행되는 악성코드를 파일리스 악성코드(Fileless Malware)라고 한다.

일반적으로 파일리스 악성코드는 메모리에 실행 데이터를 저장한 후 실행시키기 때문에 최초 실행 이후 메모리를 초기화하거나, 재부팅을 할 경우 이미 장악한 시스템 권한 등을 다시 얻기 위한 재공격이 필요하다.[3] 하지만, 최근 발견되고 있는 파일리스 악성코드는 공격대상 PC 에 최소한의 악성 파일을 남겨두어 지속적으로 제어권한을 획득할 수 있는 방식을 갈수록 파일리스 악성코드에 대한 위협이 높아지고 있다.

### 2.2 악성코드 탐지 기법

악성코드를 분석하는 기법은 크게 정적 분석과 동적 분석 두가지로 분류된다. 정적 분석은 악성코드를 실행하지 않고 가지고 있는 내용을 통해 악성 여부를 진단하는 것이며, 동적 분석은 해당 파일을 실행하므로서 나타나는 변화를 모니터링하여 어떠한 기능을 수행하는지 확인한다. 의심되는 파일이 실제 악성 행위를 할 수 있으므로 보통 가상의 환경에서 수행된다.

본 논문에서는 위 두 탐지 기법 중 정적 분석 기법에 해당하는 시그니처 기반 탐지와 이를 딥러닝 기술에 활용할 수 있도록 수정한 시퀀스기반 탐지를 활용하여 악성코드를 탐지한다.

#### 2.2.1 시그니처 기반 탐지

시그니처 기반 탐지 기법은 악성 파일을 식별하기 위해 사용되는 방법으로, 시그니처는 백신 프로그램이 파일을 스캔할 때 해당 파일을 유일하게 식별할 수 있도록 사용되는 데이터를 이야기한다.
시그니처 기반 탐지 기법은 백신프로그램을 통해 새로운 파일을 스캔할 때, 저장된 시그니처와 대조하여 일치 여부를 확인하는 기법이다.

대부분의 백신 프로그램들은 해당하는 벤더에 알려진 모든 악성코드들을 시그니처 형태로 가공한 데이터베이스를 포함하고 있으며, 이 데이터베이스는 바이러스 분석가 또는 시그니처 제작자들에 의해 새롭게 식별된 악성코드들의 시그니처가 정기적으로 업데이트된다. 이를 활용하여 백신 프로그램을 통해 파일을 스캔할 때, 시그니처와 파일 사이의 일치 여부를 확인하게 된다.

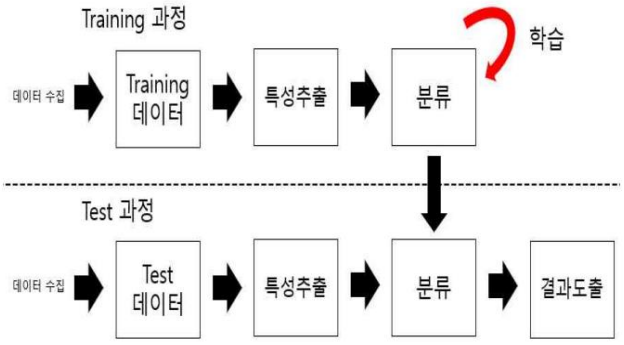
#### 2.2.2 시퀀스 기반 탐지

본 논문에서 서술할 시퀀스 기반 탐지 기법은 시퀀스 기반 탐지 기법은 기존 시그니처 기반 탐지 기법을 활용하여 사용 될 머신러닝 모델에 맞추기 위해 분석할 데이터를 파일 시퀀스 형태로 입력하여 분석하는 기법임을 의미한다.

### 2.3 악성코드 탐지와 딥러닝 기반 악성코드 탐지

딥러닝을 기반으로 하여 악성코드를 탐지하는 연구는 일부 존재하며, CNN, LSTM, GAN 등의 모델을 사용하여 탐지하는 연구 등이 있다.

그중 아주대학교에서 진행된 연구[5]에서 CNN 을 활용하여 악성코드 탐지를 한 사례가 있다.




(그림 2) 딥러닝을 적용한 악성코드 탐지 흐름도[5]

위는 딥러닝을 적용한 악성코드 탐지 알고리즘의 흐름도를 나타낸다. 해당 논문에서는 딥러닝 기술을 활용하여 악성코드를 탐지하기 위해 위 그림과 같은 방법을 사용한다. 먼저, 시그니쳐를 가진 가상의 악성코드를 만들어 Training data set 을 수집하는데 총 5 개의 악성코드 탐지를 위해 500 개의 시그니처를 직접 제작하여 수집한다.

수집 이후 악성코드의 특징을 추출하는 전처리과정을 통해 특징점을 추출해낸다. 이 방법은 본 논문에서 시퀀스 기반 탐지라고 정의한 기법과 유사하다고 할 수 있다.

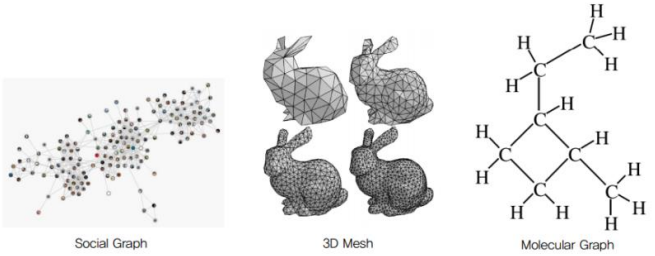
## 3. GCN 과 자카드 유사도

### 3.1 GCN(Graph Convolutional Networks)

Graph Convolutional Networks 즉, GCN은 그래프로 표현되는 데이터에 합성곱 연산을 수행하는 심층 학습 알고리즘이다[7].

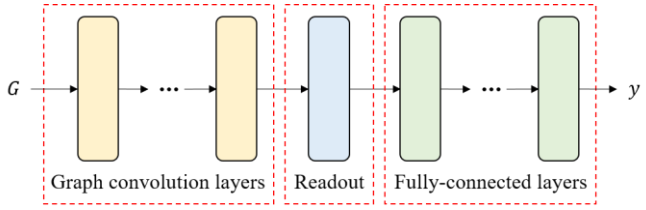
1)그래프 데이터

대부분의 머신 러닝 알고리즘은 입력 데이터가 유클리드 공간 (Euclidean space)에 존재함을 가정하고 있다. 즉, 통계 데이터나 이미지처럼 입력 데이터가 벡터의 형태로 표현될 수 있어야 한다. 그러나 소셜 네트워크, 관계형 데이터베이스, 분자 구조 등과 같이 객체들과 그 객체들 간의 관계로 표현되는 데이터는 기본적으로 그림 3과 같이 그래프로 표현된다[7].




(그림 3) 여러 유형의 그래프데이터[6]

그래프는 $G=(A,X)$와 같이 정의되며 $A \in R^{N\times N}$는 각 node 의 연결을 나타내는 인접 행렬이며, $X \in R^{N\times d}$는 node 특징 행렬이다. 여기서 N 은 node 의 개수, d 는 node 특징의 수이다. 이에 대한 연산 convolution $\psi$ 은 A 와 X 를 입력 받아 $H \in R^{N\times m}$라는 새로운 latent node 특성 행렬을 생성한다. m 은 latent 특성 벡터의 차원이다. 따라서 기본적인 convolution 은 $H=\psi(A,X)=\sigma(AXW)$ 이다. 이때, $W \in R^{d\times m}$ 은 학습이 가능한 가중치 행렬이며, GCN 을 학습시킨다는 것은 이 가중치 행렬을 조정하는 것과 같다[6].




(그림 4) GCN 의 구조[8]

GCN은 일반적으로 그림 11과 같이 Graph Convolution으로 정의되는 Graph Convolution 레이어와 Fully-Connected 레이어로 구성된다. 이 구성에서 가장 중요한 부분은 Graph Convolution 레이어이다. Graph Convolution 레이어를 통해 그래프 형태의 데이터가 행렬 형태의 데이터로 변환되기 때문에 Graph Convolution 레이어를 거친 데이터는 기존의 머신 러닝 알고리즘에 그대로 적용할 수 있게 된다. 또한, Readout은 Graph Convolution 레이어를 통해 생성된 latent feature 행렬을 그래프 전체를 표현하는 하나의 벡터로 변환하는 함수이다. 일반적으로 Readout은 전체 Node의 Latent Feature Vector를 평균을 내어 그래프 전체를 표현하는 하나의 벡터를 생성한다..

### 3.2 자카드 유사도(Jaccard Similarity)

본 논문에서는 실험의 정확도를 개선하기 위해 각 파일의 시퀀스 유사도를 계산하는 방법 중 하나인 자카드 유사도를 활용하였다[9].

자카드 유사도는 두 집합 사이의 유사도를 계산하는 방법 중 하나로, A 와 B 두 집합이 있다고 한다면 두 집합의 합집합에서 교집합의 비율을 구하여 유사도를 구하는 방법이며 이를 식으로 표현하면 다음과 같다.

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|}=\frac{SequenceA\cap SequenceB}{SequenceA\cup SequenceB}$$

(식 1) 자카드 유사도 식

### 3.3 GCN 모델과 자카드 유사도를 이용해서 악성파일 탐지방법

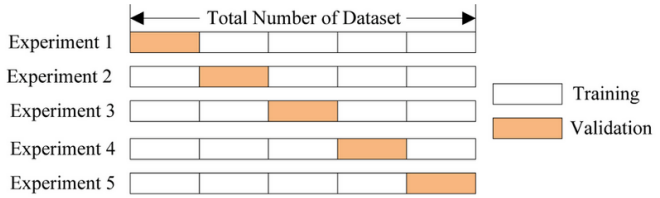
GCN(Graph Convolutional Networks) 에서는 CNN(Convolution Neural Networks)과 같이 인접행렬을 convolution 연산에 활용한다. 하지만 GCN 에서 인접 행렬 그대로 convolution 에 사용하는 것에는 한계점이 존재한다. 첫 번째로 인접행렬에는 인접한 노드와의 연결만 표현되어 있다. 따라서 graph convolution 과정에서 해당 노드 자체에 대한 정보는 latent vector 를 만들 때 고려되지 않는다. 두번째, 일반적으로 인접행렬은 정규화 되어 있지 않다. 그렇기 때문에 특성 벡터와 인접행렬을 곱할 경우 특성 벡터의 크기가 불안정하게 변할 수 있다.

보통 이 두 가지 문제를 해결하기위해 인접행렬에 self-loop 를 추가하고 인접행렬을 본 논문에서는 이러한 방법과 함께 각 시퀀스파일에 자카드 유사도를 적용하여 탐지율을 향상시키고자 한다.

## 4. 실험

### 4.1 실험 환경

이번 연구에서 진행되는 실험은 CNN(Convolutional Neural Network), 자카드 유사도를 적용한 GCN 딥러닝 기법을 사용하여 실험을 진행하며, 시퀀스 기반으로 추출된 PowerShell 데이터셋을 사용한다. 실험의 정확성을 위해 공격자가 주로 사용하는 난독화 기법을 Revoke-Obfuscation 모듈을 통해 무력화시킨 상태로 정상 1000 개, 악성 1000 개로 실험을 진행하며, 학습과 검증 데이터셋의 비율은 8:2 로 설정한다[4]. 최대 시퀀스 길이는 1000 개, Epochs 횟수는 50 으로 한다. 이외 정확도 향상을 위해 5-Fold 교차검증을 사용한다.




(그림 5) K-Fold Cross Validation Method 의 구조

### 4.2 실험 데이터

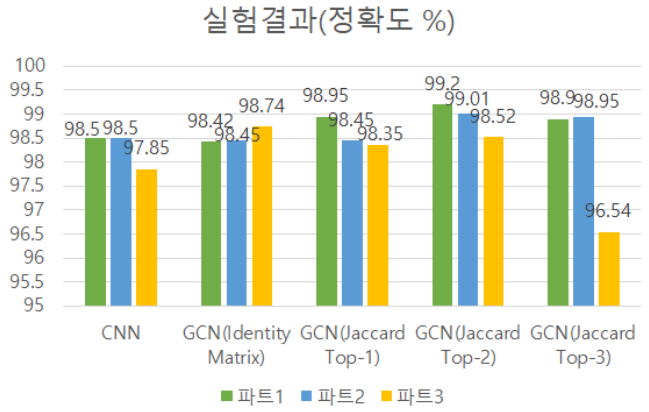
실험에 진행되는 데이터는 PowerShell 데이터셋을 시퀀스 데이터로 1 차적으로 변환하고, 2 차적으로

프리퀀시 데이터로 토큰화 하여 GCN 모델의 Jaccard 유사도 기법을 사용하여 인접행렬로 사용한다. 프리퀀시 데이터는 기존 시퀀스 데이터의 행위 간 중복을 막기 위해 분석할 모든 데이터셋의 행위 유형을 이진화 한 데이터이다.

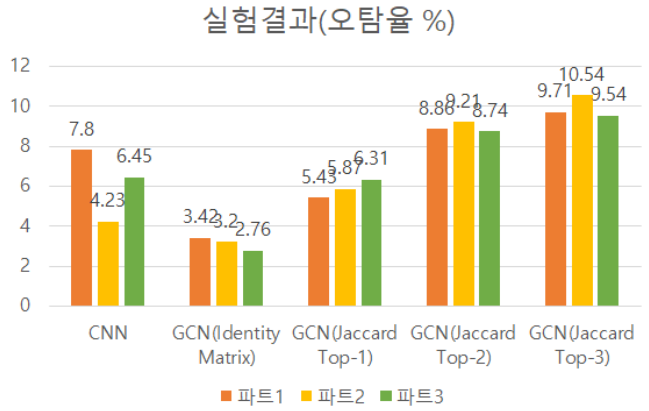
### 4.3 실험 결과

이번 연구에서 진행되는 자카드 유사도를 적용한 GCN 딥 러닝 결과는 다음과 같이 나왔으며, 논문에서는 대조군으로 CNN 과 Identity Matrix 를 인접행렬로 가진 GCN 딥러닝 결과로 설정하였다.

이번 실험에서는 대조군으로 CNN 과 Identity Matrix 를 인접행렬로 한 GCN, 실험군으로 자카드 유사도의 Top-K 값을 변경시킨 GCN 딥러닝 모델을 비교해 보았다. Identity Matrix 를 인접행렬로 가진 GCN 모델은 가장 낮은 정확도를 보여주었으나, 오탐율 또한 낮았다. 자카드 유사도를 적용한 모델중에서는 Top-2 가 가장 높은 정확도를 보여주었으나 높은 탐지율만큼 높은 오탐율을 보여주었다. 의외인 것은 인접행렬 정보가 많은 Top-3 가 가장 높을거라 기대하였으나 가장 높지 않은 정확도와 높은 오탐율을 보여주었는데 GCN-자카드 유사도 모델에서는 적정한 수의 Top-K 인접행렬을 가진 자카드 유사도 모델을 적용한 GCN 모델이 가장 우수한 모습을 보이고 있다는 것을 실험 결과를 통해 알 수 있다.




(그래프 1) 실험 정확도




(그래프 2) 실험 오탐율

### 5. 결론

본 논문에서는 GCN 모델과 자카드 유사도를 이용하여 대조군으로 사용하였던 CNN 모델과 Identity Matrix 을 인접행렬로 사용하는 GCN 모델 대비 정확도가 얼마나 향상 되었는지 확인하였다. GCN 모델의 핵심은 서로간의 인접행렬 정보가 중요한데, 이번 실험에서는 파일간의 유사도를 수치로 나타낸 자카드 유사도를 인접행렬로 하여 실험을 진행하였다.

GCN 모델은 기존의 CNN 모델 대비 성능이 소폭 우수하였고, 그 중 적절한 수의 Top-K 값을 가진 자카드 유사도 인접행렬이 가장 우수한 탐지율을 보였다.

GCN 모델을 효과적으로 활용하기 위해서는 노드간 이어주는 다양한 인접행렬 정보에 대해 연구한다면 더 우수한 결과를 보여줄 것으로 생각한다.

### 참고문헌

[1] 최선오, 김영수, 김종현, 김익균, 딥러닝을 이용한 악성코드탐지 연구동향, 정보보호학회지, 2017 년

[2] Deep Learning (Wikipidia), https://en.wikipedia.org/wiki/Deep_learning

[3] Fileless Threats Protection - kaspersky https://www.kaspersky.com/enterprise-security/wiki-section/products/fileless-threats-protection

[4] 조진호, 박양훈, 박광철, 파워쉘 공격 탐지방법에 대한 딥러닝 연구, 한국정보처리학회, 2020 년

[5] 우호성, 정건웅, 김재현, CNN 을 활용한 악성코드 탐지 모델 구현 한국통신학회, 2018 년

[6] [머신 러닝/딥 러닝] 그래프 합성곱 신경망 (Graph Convolutional Network, GCN) https://untitledtblog.tistory.com/152

[7] GCN(graph convolutional network) 기본이론1 https://m.blog.naver.com/demian7607/222090020257

[8] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. ICLR 2017

[9] Jaccard Index (Wikipedia), https://en.wikipedia.org/wiki/Jaccard_index