(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0024479
(43) 공개일자 2025년02월18일

(51) 국제특허분류(Int. Cl.)
H04L 9/40 (2022.01) *G06N 20/00* (2019.01)
G06N 3/092 (2023.01)
(52) CPC특허분류
H04L 63/1433 (2013.01)
G06N 20/00 (2021.08)
(21) 출원번호 10-2024-0093811
(22) 출원일자 2024년07월16일
심사청구일자 2024년07월16일
(30) 우선권주장
1020230105754 2023년08월11일 대한민국(KR)

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
김휘강
서울특별시 서초구 서운로 104, 201동 804호(서초동, 래미안서초에스티지에스)
최재웅
서울특별시 성북구 동소문로20나길 47 (동선동1가)
(뒷면에 계속)
(74) 대리인
이대호, 박건홍

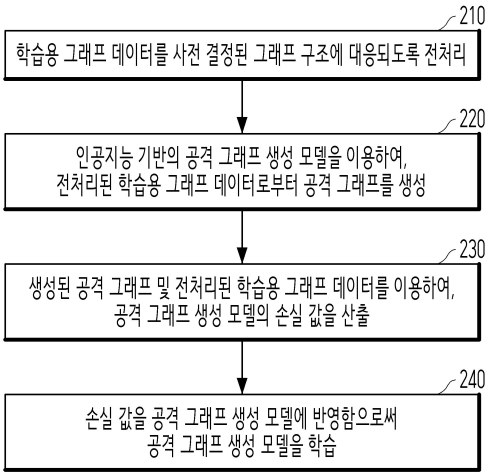
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 방법 및 장치**

(57) 요 약

본 개시의 일 실시예에 따라 컴퓨팅 장치에 의해 수행되는, 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 방법으로서, 상기 방법은: 학습용 그래프 데이터를 사전 결정된 그래프 구조에 대응되도록 전처리하는 단계; 인공지능 기반의 공격 그래프 생성 모델을 이용하여, 상기 전처리된 학습용 그래프 데이터로부터 공격 그래프를 생성하는 단계; 상기 생성된 공격 그래프 및 상기 전처리된 학습용 그래프 데이터를 이용하여, 상기 공격 그래프 생성 모델의 손실 값을 산출하는 단계; 및 상기 손실 값을 상기 공격 그래프 생성 모델에 반영함으로써 상기 공격 그래프 생성 모델을 학습시키는 단계;를 포함할 수 있다.

대 표 도 - 도2



(52) CPC특허분류
G06N 3/092 (2023.01)
H04L 63/1441 (2013.01)
(72) 발명자
유정도
서울특별시 성북구 화랑로5길 4-12 (하월곡동)
이효선
경기도 남양주시 와부읍 월문천로 95 (덕소아이파크아파트)
박상범
서울특별시 동대문구 제기로4다길 26, 102호 (제기동)
송민근
서울특별시 동대문구 무학로49길 22 (용두동)
이재성
서울특별시 성북구 고려대로27길 52-16 (안암동5가)
사공채연
서울특별시 동대문구 무학로45길 7 (용두동)

이 발명을 지원한 국가연구개발사업

과제고유번호	미부여
과제번호	2021-0-00624-003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보보호핵심원천기술개발(R&D,정보화)
연구과제명	C-ITS 인프라 보안성 강화를 위한 복합 위협헌팅 모델 기반 지능형 사이버 공격/방어 분석 프레임워크 기술 개발
과제수행기관명	(주)케이사인
연구기간	2024.01.01 ~ 2024.12.31

명 세 서

청구범위

청구항 1

컴퓨팅 장치에 의해 수행되는, 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 방법으로서, 상기 방법은:

학습용 그래프 데이터를 사전 결정된 그래프 구조에 대응되도록 전처리하는 단계;

인공지능 기반의 공격 그래프 생성 모델을 이용하여, 상기 전처리된 학습용 그래프 데이터로부터 공격 그래프를 생성하는 단계;

상기 생성된 공격 그래프 및 상기 전처리된 학습용 그래프 데이터를 이용하여, 상기 공격 그래프 생성 모델의 손실 값을 산출하는 단계; 및

상기 손실 값을 상기 공격 그래프 생성 모델에 반영함으로써 상기 공격 그래프 생성 모델을 학습시키는 단계;

를 포함하는,

방법.

청구항 2

제1항에 있어서,

상기 학습용 그래프 데이터는,

상기 공격 그래프 생성 모델의 학습에 사용되는 데이터로서, 그래프에 포함되는 복수의 노드들 각각 대한 CVE(common vulnerabilities and exposures) 데이터베이스 기반의 취약점 정보를 포함하는,

방법.

청구항 3

제1항에 있어서,

상기 공격 그래프는,

공격자가 이용가능한 적어도 하나의 공격 경로를 시각적으로 표시한 그래프인,

방법.

청구항 4

제1항에 있어서,

상기 공격 그래프 생성 모델은,

상기 전처리된 학습용 그래프 데이터를 이용하여, 상기 전처리된 학습용 그래프 데이터에 포함된 복수의 노드들 각각의 복수의 임베딩 값들을 출력하는 인코더(encoder) -하나의 노드는 하나의 임베딩 값에 대응됨-;

상기 인코더에서 출력된 상기 복수의 임베딩 값들을 이용하여, 상기 복수의 노드들의 순서를 결정하는 디코더(decoder); 및

상기 결정된 순서에 기초하여, 상기 복수의 노드들을 결합하여 상기 공격 그래프를 생성하는 공격 그래프 생성

부;

를 포함하는,

방법.

청구항 5

제4항에 있어서,

상기 인코더는,

상기 복수의 노드들 각각에 대한 CVE 데이터베이스 기반의 취약점 정보에 대응되는 CVSS(common vulnerability scoring system)의 점수(score)에 기초하여, 상기 복수의 노드들 각각에 대해서 적어도 하나의 이웃 노드와의 중요도를 반영한 상기 복수의 임베딩 값들을 출력하는,

방법.

청구항 6

제4항에 있어서,

상기 디코더는,

상기 복수의 임베딩 값들을 이용하여, 사전 선택된 기준 노드와 엣지를 통해 연결된 적어도 하나의 제1 이웃 노드 중에서 제1 노드를 선택하고,

상기 복수의 임베딩 값들을 이용하여, 상기 제1 노드와 엣지를 통해 연결된 적어도 하나의 제2 이웃 노드 중에서 상기 기준 노드와 상이한 제2 노드를 선택하고, 그리고

상기 복수의 노드들을 모두 선택할 때까지 노드를 선택하는 과정을 반복하는,

방법.

청구항 7

제4항에 있어서,

상기 공격 그래프 생성부는,

상기 복수의 노드들의 순서에 기초하여, 상기 복수의 노드들을 각각 연결하는 복수의 엣지들의 순서를 결정하고, 그리고

상기 복수의 노드들의 순서 및 상기 복수의 엣지들의 순서를 표시한 상기 공격 그래프를 생성하는,

방법.

청구항 8

제1항에 있어서,

상기 손실 값을 산출하는 단계는,

상기 전처리된 학습용 그래프 데이터에 기초하여, 상기 생성된 공격 그래프로부터 보상 값을 산출하는 단계;

베이스라인 모델을 이용하여, 상기 전처리된 그래프 데이터로부터 베이스라인 값을 산출하는 단계; 및

상기 보상 값 및 상기 베이스라인 값을 이용하여, 상기 손실 값을 산출하는 단계;

를 포함하는,

방법.

청구항 9

제8항에 있어서,

상기 보상 값은,

상기 생성된 공격 그래프에서 하나의 노드에서 다른 노드로 가는 전체 경로에 대한 평가를 나타내는 값으로, 상기 전처리된 학습용 그래프 데이터에 포함된 복수의 노드들 각각의 가중치에 기초하여 산출되는,

방법.

청구항 10

제8항에 있어서,

상기 베이스라인 모델의 구조는 상기 공격 그래프 생성 모델과 대응되고,

상기 베이스라인 모델의 신경망 가중치의 값은 상기 공격 그래프 생성 모델의 학습 단계 중 현재 단계의 이전 단계의 신경망 가중치의 값에 대응되는,

방법.

청구항 11

컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램으로서, 상기 컴퓨터 프로그램은 컴퓨팅 장치의 프로세서로 하여금 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 방법을 수행하도록 하며, 상기 방법은:

학습용 그래프 데이터를 사전 결정된 그래프 구조에 대응되도록 전처리하는 단계;

인공지능 기반의 공격 그래프 생성 모델을 이용하여, 상기 전처리된 학습용 그래프 데이터로부터 공격 그래프를 생성하는 단계;

상기 생성된 공격 그래프 및 상기 전처리된 학습용 그래프 데이터를 이용하여, 상기 공격 그래프 생성 모델의 손실 값을 산출하는 단계; 및

상기 손실 값을 상기 공격 그래프 생성 모델에 반영함으로써 상기 공격 그래프 생성 모델을 학습시키는 단계;

를 포함하는,

컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 12

공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 컴퓨팅 장치에 있어서,

프로세서;

메모리; 및

네트워크부;

를 포함하고,

상기 프로세서는:

학습용 그래프 데이터를 사전 결정된 그래프 구조에 대응되도록 전처리하는 동작;

인공지능 기반의 공격 그래프 생성 모델을 이용하여, 상기 전처리된 학습용 그래프 데이터로부터 공격 그래프를 생성하는 동작;

상기 생성된 공격 그래프 및 상기 전처리된 학습용 그래프 데이터를 이용하여, 상기 공격 그래프 생성 모델의 손실 값을 산출하는 동작; 및

상기 손실 값을 상기 공격 그래프 생성 모델에 반영함으로써 상기 공격 그래프 생성 모델을 학습시키는 동작;

을 수행하는,

컴퓨팅 장치.

발명의 설명

기 술 분 야

[0001] 본 개시는 공격 그래프의 생성에 관한 것으로, 보다 구체적으로, 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 현재 인터넷에 연결된 많은 네트워크는 토폴로지 구조가 변화하고 규모 확장이 쉬운 동적 네트워크 환경(dynamic network environment)으로 이루어져 있다. 사용자는 빠르게 변화하고 확장되는 동적 네트워크 환경에서 시스템의 취약한 부분을 찾기 쉽지 않으며, 공격자의 공격 경로를 예측하는 것 또한 어려울 수 있다.

[0003] 따라서, 공격 경로를 예측하기 위한 공격 그래프의 자동 생성은 빠르게 변화하고 확장되는 동적 네트워크 환경을 모델링할 수 있도록 한다. 생성된 공격 그래프의 노드와 엣지 정보는 시스템의 정보, 취약점 정보, 시스템 간의 연결성 등으로 설정되어 공격자의 공격 경로를 예측하는데 사용될 수 있다.

[0004] 하지만, 공격 그래프를 생성하는 과정은 공격 경로를 선택하고 노드와 엣지의 확률을 계산해야 하므로, 공격 그래프를 생성하는 데까지 오랜 시간이 걸릴 수 있다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 대한민국 공개특허 제10-2018-0113508호(2020.03.31. 공개)

발명의 내용

해결하려는 과제

[0006] 본 개시는 전술한 배경기술에 대응하여 안출된 것으로, 공격 그래프를 빠르고 정확하게 생성하기 위한 인공지능 모델을 학습시키는 방법 및 장치를 제공하고자 한다.

[0007] 본 개시의 기술적 과제들은 이상에서 언급한 기술적 과제로 제한되지 않으며, 언급되지 않은 또 다른 기술적 과제들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0008] 전술한 바와 같은 과제를 해결하기 위한 본 개시의 일 실시예에 따라, 컴퓨팅 장치에 의해 수행되는, 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 방법이 개시된다. 상기 방법은: 학습용 그래프 데이터를 사전 결정된 그래프 구조에 대응되도록 전처리하는 단계; 인공지능 기반의 공격 그래프 생성 모델을 이용하여, 상기 전처리된 학습용 그래프 데이터로부터 공격 그래프를 생성하는 단계; 상기 생성된 공격 그래프 및 상기 전처리된 학습용 그래프 데이터를 이용하여, 상기 공격 그래프 생성 모델의 손실 값을 산출하는 단계; 및 상기 손실 값을 상기 공격 그래프 생성 모델에 반영함으로써 상기 공격 그래프 생성 모델을 학습시키는 단계;를 포함할 수

있다.

[0009] 대안적으로, 상기 학습용 그래프 데이터는, 상기 공격 그래프 생성 모델의 학습에 사용되는 데이터로서, 그래프에 포함되는 복수의 노드들 각각 대한 CVE(common vulnerabilities and exposures) 데이터베이스 기반의 취약점 정보를 포함할 수 있다.

[0010] 대안적으로, 상기 공격 그래프는, 공격자가 이용가능한 적어도 하나의 공격 경로를 시각적으로 표시한 그래프일 수 있다.

[0011] 대안적으로, 상기 공격 그래프 생성 모델은, 상기 전처리된 학습용 그래프 데이터를 이용하여, 상기 전처리된 학습용 그래프 데이터에 포함된 복수의 노드들 각각의 복수의 임베딩 값들을 출력하는 인코더(encoder) -하나의 노드는 하나의 임베딩 값에 대응됨-; 상기 인코더에서 출력된 상기 복수의 임베딩 값들을 이용하여, 상기 복수의 노드들의 순서를 결정하는 디코더(decoder); 및 상기 결정된 순서에 기초하여, 상기 복수의 노드들을 결합하여 상기 공격 그래프를 생성하는 공격 그래프 생성부;를 포함할 수 있다.

[0012] 대안적으로, 상기 인코더는, 상기 복수의 노드들 각각에 대한 CVE 데이터베이스 기반의 취약점 정보에 대응되는 CVSS(common vulnerability scoring system)의 점수(score)에 기초하여, 상기 복수의 노드들 각각에 대해서 적어도 하나의 이웃 노드와의 중요도를 반영한 상기 복수의 임베딩 값들을 출력할 수 있다.

[0013] 대안적으로, 상기 디코더는, 상기 복수의 임베딩 값들을 이용하여, 사전 선택된 기준 노드와 엣지를 통해 연결된 적어도 하나의 제1 이웃 노드 중에서 제1 노드를 선택하고, 상기 복수의 임베딩 값들을 이용하여, 상기 제1 노드와 엣지를 통해 연결된 적어도 하나의 제2 이웃 노드 중에서 상기 기준 노드와 상이한 제2 노드를 선택하고, 그리고 상기 복수의 노드들을 모두 선택할 때까지 노드를 선택하는 과정을 반복할 수 있다.

[0014] 대안적으로, 상기 공격 그래프 생성부는, 상기 복수의 노드들의 순서에 기초하여, 상기 복수의 노드들을 각각 연결하는 복수의 엣지들의 순서를 결정하고, 그리고 상기 복수의 노드들의 순서 및 상기 복수의 엣지들의 순서를 표시한 상기 공격 그래프를 생성할 수 있다.

[0015] 대안적으로, 상기 손실 값을 산출하는 단계는, 상기 전처리된 학습용 그래프 데이터에 기초하여, 상기 생성된 공격 그래프로부터 보상 값을 산출하는 단계; 베이스라인 모델을 이용하여, 상기 전처리된 그래프 데이터로부터 베이스라인 값을 산출하는 단계; 및 상기 보상 값 및 상기 베이스라인 값을 이용하여, 상기 손실 값을 산출하는 단계;를 포함할 수 있다.

[0016] 대안적으로, 상기 보상 값은, 상기 생성된 공격 그래프에서 하나의 노드에서 다른 노드로 가는 전체 경로에 대한 평가를 나타내는 값으로, 상기 전처리된 학습용 그래프 데이터에 포함된 복수의 노드들 각각의 가중치에 기초하여 산출될 수 있다.

[0017] 대안적으로, 상기 베이스라인 모델의 구조는 상기 공격 그래프 생성 모델과 대응되고, 상기 베이스라인 모델의 신경망 가중치의 값은 상기 공격 그래프 생성 모델의 학습 단계 중 현재 단계의 이전 단계의 신경망 가중치의 값에 대응될 수 있다.

[0018] 본 개시의 일 실시예에 따른 컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램이 개시된다. 상기 컴퓨터 프로그램은 컴퓨팅 장치의 프로세서로 하여금 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 방법을 수행하도록 할 수 있다. 상기 방법은: 학습용 그래프 데이터를 사전 결정된 그래프 구조에 대응되도록 전처리하는 단계; 인공지능 기반의 공격 그래프 생성 모델을 이용하여, 상기 전처리된 학습용 그래프 데이터로부터 공격 그래프를 생성하는 단계; 상기 생성된 공격 그래프 및 상기 전처리된 학습용 그래프 데이터를 이용하여, 상기 공격 그래프 생성 모델의 손실 값을 산출하는 단계; 및 상기 손실 값을 상기 공격 그래프 생성 모델에 반영함으로써 상기 공격 그래프 생성 모델을 학습시키는 단계;를 포함할 수 있다.

[0019] 본 개시의 일 실시예에 따른 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 컴퓨팅 장치가 개시된다. 상기 컴퓨팅 장치는, 프로세서; 메모리; 및 네트워크부;를 포함할 수 있다. 상기 프로세서는: 학습용 그래프 데이터를 사전 결정된 그래프 구조에 대응되도록 전처리하는 동작; 인공지능 기반의 공격 그래프 생성 모델을 이용하여, 상기 전처리된 학습용 그래프 데이터로부터 공격 그래프를 생성하는 동작; 상기 생성된 공격 그래프 및 상기 전처리된 학습용 그래프 데이터를 이용하여, 상기 공격 그래프 생성 모델의 손실 값을 산출하는 동작; 및 상기 손실 값을 상기 공격 그래프 생성 모델에 반영함으로써 상기 공격 그래프 생성 모델을 학습시키는 동작;을 수행할 수 있다.

발명의 효과

[0020] 본 개시는 공격 그래프를 빠르고 정확하게 생성하기 위한 인공지능 모델을 학습시키는 방법 및 장치를 제공하고자 한다.

[0021] 본 개시는 Graph Neural Networks (GNN)를 그래프 임베딩에 활용하는 것으로 네트워크 그래프 자체를 학습하여 공격 그래프를 생성하고, 강화 학습을 사용하여 공격 그래프 생성 모델을 학습시킴으로써, 동적 네트워크에서 변화하는 네트워크 그래프에 맞춰 공격 경로를 선택하고 기계 학습 모델에서 그래프 재생성 과정 없이 공격 그래프를 도출할 수 있다. 이를 통해 본 개시의 공격 그래프 생성 과정은 동적 네트워크에서 빠르게 공격자의 공격 경로를 예측하고 확보하는 효과를 가질 수 있다.

[0022] 본 개시에서 얻을 수 있는 효과는 이상에서 언급한 효과로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 개시가 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0023] 다양한 양상들이 이제 도면들을 참조로 기재되며, 여기서 유사한 참조 번호들은 총괄적으로 유사한 구성요소들을 지칭하는데 이용된다. 이하의 실시예에서, 설명 목적을 위해, 다수의 특정 세부사항들이 하나 이상의 양상들의 총체적 이해를 제공하기 위해 제시된다. 그러나, 그러한 양상(들)이 이러한 구체적인 세부사항들 없이 실시될 수 있음은 명백할 것이다.

도 1은 본 개시의 일 실시예에 따른 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 컴퓨팅 장치를 설명하기 위한 도면이다.

도 2는 본 개시의 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 방법을 나타낸 도면이다.

도 3은 본 개시의 일 실시예에 따른 공격 그래프 생성 모델의 예시적인 구조를 간략하게 나타낸 도면이다.

도 4는 본 개시의 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 공격 그래프 생성 모델을 이용하여 공격 그래프를 생성하는 과정을 나타낸 도면이다.

도 5는 본 개시의 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 손실 값을 산출하는 과정을 나타낸 도면이다.

도 6a는 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 노드 선택 및 그래프 생성 과정의 제1 단계를 나타낸 도면이다.

도 6b는 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 노드 선택 및 그래프 생성 과정의 제2 단계를 나타낸 도면이다.

도 6c는 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 노드 선택 및 그래프 생성 과정의 제3 단계를 나타낸 도면이다.

도 6d는 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 노드 선택 및 그래프 생성 과정의 제4 단계를 나타낸 도면이다.

도 7은 본 개시의 일 실시예에 따른 인공지능 모델의 예시적인 구조를 나타낸 도면이다.

도 8은 본 개시의 실시예들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 간략하고 일반적인 개략도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0024] 다양한 실시예들이 이제 도면을 참조하여 설명된다. 본 명세서에서, 다양한 설명들이 본 개시의 이해를 제공하기 위해서 제시된다. 그러나, 이러한 실시예들은 이러한 구체적인 설명 없이도 실행될 수 있음이 명백하다.

[0025] 본 명세서에서 사용되는 용어 "컴포넌트", "모듈", "시스템" 등은 컴퓨터-관련 엔티티, 하드웨어, 펌웨어, 소프트웨어, 소프트웨어 및 하드웨어의 조합, 또는 소프트웨어의 실행을 지칭한다. 예를 들어, 컴포넌트는 프로세서 상에서 실행되는 처리과정(procedure), 프로세서, 객체, 실행 스레드, 프로그램, 및/또는 컴퓨터일 수 있지만, 이들로 제한되는 것은 아니다. 예를 들어, 컴퓨팅 장치에서 실행되는 애플리케이션 및 컴퓨팅 장치 모두 컴포넌

트일 수 있다. 하나 이상의 컴포넌트는 프로세서 및/또는 실행 스레드 내에 상주할 수 있다. 일 컴포넌트는 하나의 컴퓨터 내에 로컬화 될 수 있다. 일 컴포넌트는 2개 이상의 컴퓨터들 사이에 분배될 수 있다. 또한, 이러한 컴포넌트들은 그 내부에 저장된 다양한 데이터 구조들을 갖는 다양한 컴퓨터 판독가능한 매체로부터 실행할 수 있다. 컴포넌트들은 예를 들어 하나 이상의 데이터 패킷들을 갖는 신호(예를 들면, 로컬 시스템, 분산 시스템에서 다른 컴포넌트와 상호작용하는 하나의 컴포넌트로부터의 데이터 및/또는 신호를 통해 다른 시스템과 인터넷과 같은 네트워크를 통해 전송되는 데이터)에 따라 로컬 및/또는 원격 처리들을 통해 통신할 수 있다.

[0026] 더불어, 용어 "또는"은 배타적 "또는"이 아니라 내포적 "또는"을 의미하는 것으로 의도된다. 즉, 달리 특정되지 않거나 문맥상 명확하지 않은 경우에, "X는 A 또는 B를 이용한다"는 자연적인 내포적 치환 중 하나를 의미하는 것으로 의도된다. 즉, X가 A를 이용하거나; X가 B를 이용하거나; 또는 X가 A 및 B 모두를 이용하는 경우, "X는 A 또는 B를 이용한다"가 이들 경우들 어느 것으로도 적용될 수 있다. 또한, 본 명세서에 사용된 "또는", "및/또는"이라는 용어는 열거된 관련 아이템들 중 하나 이상의 아이템의 가능한 모든 조합을 지칭하고 포함하는 것으로 이해되어야 한다.

[0027] 또한, "포함한다" 및/또는 "포함하는"이라는 용어는, 해당 특징 및/또는 구성요소가 존재함을 의미하는 것으로 이해되어야 한다. 다만, "포함한다" 및/또는 "포함하는"이라는 용어는, 하나 이상의 다른 특징, 구성요소 및/또는 이들의 그룹의 존재 또는 추가를 배제하지 않는 것으로 이해되어야 한다. 또한, 달리 특정되지 않거나 단수 형태를 지시하는 것으로 문맥상 명확하지 않은 경우에, 본 명세서와 청구범위에서 단수는 일반적으로 "하나 또는 그 이상"을 의미하는 것으로 해석되어야 한다.

[0028] 그리고, "A 또는 B 중 적어도 하나"이라는 용어는, "A만을 포함하는 경우", "B 만을 포함하는 경우", "A와 B의 구성으로 조합된 경우"를 의미하는 것으로 해석되어야 한다.

[0029] 당업자들은 추가적으로 여기서 개시된 실시예들과 관련되어 설명된 다양한 예시적 논리적 블록들, 구성들, 모듈들, 회로들, 수단들, 로직들, 및 알고리즘 단계들이 전자 하드웨어, 컴퓨터 소프트웨어, 또는 양쪽 모두의 조합들로 구현될 수 있음을 인식해야 한다. 하드웨어 및 소프트웨어의 상호교환성을 명백하게 예시하기 위해, 다양한 예시적 컴포넌트들, 블록들, 구성들, 수단들, 로직들, 모듈들, 회로들, 및 단계들은 그들의 기능성 측면에서 일반적으로 위에서 설명되었다. 그러한 기능성이 하드웨어로 또는 소프트웨어로서 구현되는지 여부는 전반적인 시스템에 부과된 특정 어플리케이션(application) 및 설계 제한들에 달려 있다. 숙련된 기술자들은 각각의 특정 어플리케이션들을 위해 다양한 방법들로 설명된 기능성을 구현할 수 있다. 다만, 그러한 구현의 결정들이 본 개시내용의 영역을 벗어나게 하는 것으로 해석되어서는 안 된다.

[0030] 제시된 실시예들에 대한 설명은 본 개시의 기술 분야에서 통상의 지식을 가진 자가 본 발명을 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 개시의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이다. 여기에 정의된 일반적인 원리들은 본 개시의 범위를 벗어남이 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 개시는 여기에 제시된 실시예 들로 한정되는 것이 아니다. 본 개시는 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의의 범위에서 해석되어야 할 것이다.

[0031] 본 개시내용에서의 제1, 제2 , 또는 제3 과 같이 제 N으로 표현되는 용어들은 적어도 하나의 엔티티들을 구분하기 위해 사용된다. 예를 들어, 제1 과 제2 로 표현된 엔티티들은 서로 동일하거나 또는 상이할 수 있다.

[0032] 그리고, "A, B 등"과 같이 "~ 등"이라는 용어는, "A만을 포함하는 경우", "B 만을 포함하는 경우", "A와 B의 구성으로 조합된 경우"를 의미하는 것으로 해석되어야 한다.

[0033] 도 1은 본 개시의 일 실시예에 따른 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 컴퓨팅 장치를 설명하기 위한 도면이다.

[0034] 도 1을 참조하면, 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 컴퓨팅 장치(100)는 프로세서(110), 메모리(130), 네트워크부(150)를 포함할 수 있다. 상술한 컴퓨팅 장치(100)의 구성은 간략화 하여 나타낸 예시일 뿐이다. 일 실시예에서, 컴퓨팅 장치(100)는 임의의 형태의 서버, 사용자 디바이스 등을 포함할 수 있다.

[0035] 프로세서(110)는 하나 이상의 코어로 구성될 수 있으며, 컴퓨팅 장치(100)의 중앙 처리 장치(CPU: central processing unit), 범용 그래픽 처리 장치(GPGPU: general purpose graphics processing unit), 텐서 처리 장치(TPU: tensor processing unit)등의 데이터의 처리에 관련된 동작을 수행하기 위한 프로세서를 포함할 수 있다.

[0036] 본 개시의 일 실시예에 따라 프로세서(110)는 신경망의 학습을 위한 연산을 수행할 수도 있다. 예를 들어, 프로세서(110)는 딥러닝(DL: deep learning)에서 학습을 위한 입력 데이터의 처리, 입력 데이터에서의 피처 추출, 오차 계산, 역전파(backpropagation)를 이용한 신경망의 가중치 업데이트 등의 신경망의 학습을 위한 계산을 수행할 수 있다. 프로세서(110)의 CPU, GPGPU, 및 TPU 중 적어도 하나가 네트워크 함수의 학습을 처리할 수 있다. 예를 들어, CPU 와 GPGPU가 함께 네트워크 함수의 학습, 네트워크 함수를 이용한 데이터 분류를 처리할 수 있다. 또한, 본 개시의 일 실시예에서 복수의 컴퓨팅 장치들의 프로세서들을 함께 사용하여 네트워크 함수의 학습, 네트워크 함수를 이용한 데이터 분류를 처리할 수도 있다. 예를 들어, 프로세서(110)는 공격 그래프를 생성하기 위한 인공지능 기반의 공격 그래프 생성 모델을 학습시킬 수 있다. 프로세서(110)는 학습된 공격 그래프 생성 모델에 특정 그래프 데이터를 입력함으로써, 학습된 공격 그래프 생성 모델에서 공격 그래프가 출력되도록 할 수 있다.

[0037] 프로세서(110)는 통상적으로 컴퓨팅 장치(100)의 전반적인 동작을 제어할 수 있다. 프로세서(110)는 컴퓨팅 장치(100)에 포함된 구성요소들을 통해 입력 또는 출력되는 신호, 데이터, 정보 등을 처리하거나 메모리(130)에 저장된 응용 프로그램을 구동함으로써, 사용자에게 적절한 정보 또는 기능을 제공 또는 처리할 수 있다.

[0038] 본 개시의 일 실시예에서, 메모리(130)는 프로세서(110)가 생성하거나 결정한 임의의 형태의 정보 및/또는 네트워크부(150)가 수신한 임의의 형태의 정보를 저장할 수 있다. 일 실시예에서, 메모리(130)는 데이터베이스를 저장하고 있을 수 있다. 데이터베이스는 컴퓨팅 장치(100)가 처리할 수 있는 형태로 저장된 데이터의 집합일 수 있다.

[0039] 본 개시의 일 실시예에서, 메모리(130)는 플래시 메모리 타입(flash memory type), 하드디스크 타입(hard disk type), 멀티미디어 카드 마이크로 타입(multimedia card micro type), 카드 타입의 메모리(예를 들어 SD 또는 XD 메모리 등), 램(Random Access Memory, RAM), SRAM(Static Random Access Memory), 롬(Read-Only Memory, ROM), EEPROM(Electrically Erasable Programmable Read-Only Memory), PROM(Programmable Read-Only Memory), 자기 메모리, 자기 디스크 및/또는 광디스크 중 적어도 하나의 타입의 저장매체를 포함할 수 있다. 컴퓨팅 장치(100)는 인터넷(internet) 상에서 메모리(130)의 저장 기능을 수행하는 웹 스토리지(web storage)와 관련되어 동작할 수도 있다. 전술한 메모리에 대한 기재는 예시일 뿐, 본 개시는 이에 제한되지 않는다. 메모리(130)는 프로세서(110)에 의하여 동작 될 수 있다.

[0040] 본 개시의 일 실시예에 따른 네트워크부(150)는 임의의 형태의 데이터 및 신호 등을 송신 및 수신할 수 있는 임의의 유무선 통신 네트워크가 본 개시 내용에서 표현되는 네트워크에 포함될 수 있다. 본 명세서에서 설명된 기술들은 위에서 언급된 네트워크들뿐만 아니라, 다른 네트워크들에서도 사용될 수 있다.

[0041] 이하에서, 컴퓨팅 장치에서 수행되는, 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 방법을 후술하도록 한다.

[0042] 도 2는 본 개시의 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 공격 그래프를 생성하기 위한 인공지능 모델을 학습시키는 방법을 나타낸 도면이다. 도 3은 본 개시의 일 실시예에 따른 공격 그래프 생성 모델의 예시적인 구조를 간략하게 나타낸 도면이다. 일 실시예에서, 도 2에서 도시되는 단계들은 컴퓨팅 장치(100)에 의해 수행될 수 있다. 추가적인 실시예에서, 도 2에서 도시되는 단계들 중 일부는 서버(100)에서 수행되고 다른 일부는 다른 서버에서 수행되는 방식과 같이 복수의 엔티티(entity)들에 의해 구현될 수도 있다. 도 2에서 도시되는 단계들은 본 개시의 일 실시예에 따른 예시적인 단계로, 일부 단계를 추가되거나 생략할 수 있다. 도 2에서 도시되는 단계들의 순서는 상황에 따라 달라질 수 있다.

[0043] 일 실시예에서, 컴퓨팅 장치(100)의 프로세서(110)는 학습용 그래프 데이터를 사전 결정된 그래프 구조에 대응되도록 전처리할 수 있다(210).

[0044] 학습용 그래프 데이터는 공격 그래프 생성 모델의 학습에 사용되는 데이터로서, 그래프에 포함되는 복수의 노드들 각각 대한 CVE(common vulnerabilities and exposures) 데이터베이스 기반의 취약점 정보를 포함할 수 있다. 학습용 그래프 데이터는 초기 네트워크 그래프 데이터, 랜덤 그래프 등을 포함할 수 있다.

[0045] CVE는 보안 취약점을 식별하기 위한 식별자일 수 있다. CVE는 소프트웨어, 하드웨어, 네트워크 장비, 프로토콜 등에서 발견된 보안 취약점에 대한 통일된 언어와 식별 방법을 제공할 수 있다. CVE는 각 보안 취약점에 고유한 식별 번호를 할당하여 이러한 취약점을 추적하고 공유하는 데 사용될 수 있다. 일 실시예에서, 식별 번호는 "CVE-YYYY-NNNN" 형식일 수 있다. "YYYY"는 해당 년도를 나타낼 수 있다. "NNNN"은 해당 년도에 발견된 취약점 중에서 일련 번호를 의미할 수 있다. 프로세서(110)는 CVE를 이용함으로써, 취약점 정보를 효과적으로 공유하고

추적할 수 있다.

[0046] CVE 데이터베이스는 CVE와 관련된 정보가 저장되어 있는 데이터베이스로서, 소프트웨어, 하드웨어, 네트워크 장비, 프로토콜 등을 대상으로 공개된 보안 취약점과 관련된 정보를 포함할 수 있다. CVE 데이터베이스는 메모리(130)에 사전 저장되어 있을 수 있다. 일 실시예에서, CVE 데이터베이스는 메모리(130)가 아닌 외부의 저장소에 저장되어 있을 수 있다. 이 경우 프로세서(110)는 네트워크부(150)를 통해 외부의 저장소에 접근하여 CVE 데이터베이스 기반의 취약점 정보를 획득할 수 있다. 또는, 프로세서(110)는 네트워크부(150)를 통해 외부의 저장소로부터 CVE 데이터베이스 기반의 취약점 정보를 수신할 수 있다.

[0047] 그래프 구조는 복수의 노드들 및 복수의 노드들을 연결하는 엣지(edge)로 이루어진 집합을 의미할 수 있다. 일 실시예에서, 그래프 구조는 트리(tree) 구조를 포함할 수 있다. 트리 구조는 트리에 포함된 복수의 노드들 중에서 서로 다른 2개의 노드를 연결시키는 경로가 하나인 구조일 수 있다. 일 실시예에서, 사전 결정된 그래프 구조는 트리 구조일 수 있다. 다만, 그래프 구조의 종류는 이에 한정되지 않으며, 다양한 종류의 그래프 구조를 포함할 수 있다.

[0048] 공격 그래프는 공격자가 이용가능한 적어도 하나의 공격 경로를 시각적으로 표시한 그래프일 수 있다. 공격 그래프는 시스템의 보안 취약점과 공격 경로를 시각적으로 표현한 도구로, 시스템 보안을 강화하고 보안 정책을 최적화하기 위해 사용될 수 있다. 공격 그래프는 컴퓨터 시스템에서 발생할 수 있는 다양한 보안 위협과 공격 경로(attack path)를 시각적으로 나타낸 그래픽 모델일 수 있다.

[0049] 공격 경로는 공격자가 시스템에 침투하고 공격을 수행하는 데 사용할 수 있는 특정한 경로나 순서를 나타낼 수 있다. 공격 경로는 여러 단계로 나뉘어져 있을 수 있다. 공격자는 각 단계에서 시스템의 취약점을 이용하여 시스템에 침투하거나 공격을 수행할 수 있다. 공격자는 공격 경로를 이용하여 시스템 내의 중요한 데이터나 리소스에 접근하거나 시스템을 제어할 수 있다.

[0050] 일 실시예에서, 공격 그래프는 네트워크 환경을 기반으로 작성될 수 있다. 공격 그래프에서 각 노드는 네트워크에 존재하는 시스템 또는 서비스를 나타낼 수 있다. 각 노드 간의 연결을 나타내는 엣지는 공격자가 시스템을 침투하기 위해 이용할 수 있는 특정 경로를 나타낼 수 있다. 특정 경로는 시스템의 취약점, 서비스 구성 및 보안 정책 등을 기반으로 생성될 수 있다.

[0051] 공격 그래프는 로지컬(logical) 공격 그래프, 베이지안(bayesian) 공격 그래프 등을 포함할 수 있다. 로지컬 공격 그래프는 보안 전문가들이 자신의 경험을 바탕으로 보안 위협과 공격 패턴을 이해하고 시스템을 분석하기 위한 공격 그래프이며, 그래프 생성자인 보안 전문가의 주관의 영향을 받을 수 있다. 베이지안 공격 그래프는 확률론적인 모델을 기반으로 공격 그래프를 생성하는 방식이며, 그래프의 각 노드와 엣지의 확률을 사용하기에 보다 정교한 정량적인 분석을 제공할 수 있다.

[0052] 공격 그래프는 보안 전문가들이 시스템의 취약점을 이용하여 시스템을 침투하고 공격할 수 있는 방법을 분석하고 이해하는 데 사용될 수 있다. 보안 전문가들은 공격 그래프를 사용하여 시스템 내의 취약점을 식별하고 공격 경로를 시각적으로 파악할 수 있다. 이를 통해 보안 전문가들은 시스템 보안을 강화하고 취약점을 해결함으로써 공격 경로를 차단할 수 있다.

[0053] 프로세서(110)는 인공지능 기반의 공격 그래프 생성 모델을 이용하여, 상기 전처리된 학습용 그래프 데이터로부터 공격 그래프를 생성할 수 있다(220).

[0054] 도 3을 참조하면, 공격 그래프 생성 모델(300)은 입력되는 그래프 데이터로부터 공격 그래프를 출력하는 인공지능 기반의 모델을 의미할 수 있다. 일 실시예에서, 공격 그래프 생성 모델(300)은 인코더(encoder)(310), 디코더(decoder)(320), 공격 그래프 생성부(330) 등을 포함할 수 있다. 다만, 공격 그래프 생성 모델(300)의 구성은 이에 한정되지 않는다.

[0055] 일 실시예에서, 공격 그래프 생성 모델(300)은 노드 임베딩을 위한 그래프 뉴럴 네트워크(graph neural networks) 계열의 인공지능 모델을 포함할 수 있다. 일 실시예에서, 공격 그래프 생성 모델(300)은 그래프 뉴럴 네트워크 계열의 인공지능 모델 중 그래프 어텐션 네트워크(graph attention networks) 모델을 포함할 수 있다.

[0056] 그래프 뉴럴 네트워크는 그래프 구조 데이터를 다루기 위한 인공지능 모델을 의미할 수 있다. 그래프는 노드와 엣지로 이루어진 네트워크 구조를 의미할 수 있다. 그래프 데이터는 그래프에 대한 정보로서, 예를 들어, 소셜 네트워크, 분자 구조, 지식 그래프, 웹 페이지 간의 링크 구조 등을 포함할 수 있다.

[0057] 그래프 뉴럴 네트워크는 구조적인 그래프 데이터의 특징을 고려하여 노드 간의 관계를 학습할 수 있다. 그래프

뉴럴 네트워크는 이웃 노드들과의 상호작용을 통해 노드의 임베딩(embedding)을 학습하고, 이를 기반으로 그래프 전체의 특징을 추출할 수 있다. 그래프 뉴럴 네트워크는 각 노드에서 노드 자신과 적어도 하나의 이웃 노드의 정보를 종합하여 표현할 수 있다. 이러한 종합된 정보는 그래프 내의 구조와 상호 작용을 고려한 특정 형태의 임베딩으로 표현될 수 있다. 이웃 노드는 사전 결정된 기준에 따라 특정 노드와 이웃한 것으로 결정된 노드일 수 있다. 예를 들어, 이웃 노드는 특정 노드와 엣지로 직접적으로 연결된 노드를 의미할 수 있다. 다른 예를 들어, 이웃 노드는 특정 노드와 엣지로 직접적으로 연결된 노드가 없는 경우, 특정 노드와 엣지로 간접적으로 연결된 노드들 중에서 특정 노드와 연결된 엣지의 개수가 가장 적은 노드를 의미할 수 있다.

[0058] 그래프 어텐션 네트워크 모델은 어텐션 매커니즘(attention mechanism)을 적용하여 각 노드의 가중치(weight)를 다르게 설정하여 노드 간의 관계를 학습하는 인공지능 모델을 의미할 수 있다.

[0059] 어텐션 매커니즘은 입력 데이터(input data)의 모든 부분을 동일하게 다루는 것이 아닌, 각 부분들의 중요도에 따라서 어떤 부분에 집중할 것인지 가중치를 다르게 설정하는 것을 의미할 수 있다. 그래프에 적용하였을 때 일반적인 그래프 뉴럴 네트워크의 구조는 모든 이웃 노드를 동일하게 취급해 이들의 평균을 통해 현재 노드의 히든 벡터(hidden vector)를 업데이트하지만, 그래프 어텐션 네트워크 모델은 이웃 노드 간의 중요도가 다르다고 가정하고 이들에게 각각 다른 가중치를 부여하는 것으로 현재 노드의 히든 상태(hidden state)를 업데이트할 수 있다.

[0060] 일 실시예에서, 인코더(310)는 주어진 입력 데이터를 저차원의 잠재 공간(latent space)으로 변환하는 네트워크 함수일 수 있다. 인코더(310)는 전처리된 학습용 그래프 데이터를 이용하여, 전처리된 학습용 그래프 데이터에 포함된 복수의 노드들 각각의 복수의 임베딩 값들을 출력할 수 있다. 여기서, 하나의 노드는 하나의 임베딩 값에 대응될 수 있다.

[0061] 인코더(310)는 복수의 노드들 각각에 대한 CVE 데이터베이스 기반의 취약점 정보에 대응되는 CVSS(common vulnerability scoring system)의 점수(score)에 기초하여, 복수의 노드들 각각에 대해서 적어도 하나의 이웃 노드와의 중요도를 반영한 복수의 임베딩 값들을 출력할 수 있다. CVSS는 컴퓨터 시스템 및 네트워크 보안 취약점에 대한 심각성을 측정하고 평가하는 표준화된 방법론일 수 있다. 점수는 사전 결정된 범위(예를 들어, 범위 0~10, 최소단위 0.1 등) 내에 존재할 수 있다. 복수의 임베딩 값들은 각각 노드 간의 중요도에 대한 정보를 포함하는 N차원의 벡터일 수 있다. 여기서, N은 자연수일 수 있다.

[0062] 공격 그래프의 생성을 위해 각 노드는 CVE 취약점을 가지고 있으며, 공격자의 입장에서 취약점의 영향 또는 심각성에 따라 공격 경로가 달라질 수 있다. 따라서, 인코더(310)는 그래프 어텐션 네트워크 모델을 사용하여 노드에 존재하는 CVE 취약점 정보에 대응되는 CVSS의 점수를 가중치 계산에 활용함으로써, 노드마다 서로 다른 가중치를 가지도록 할 수 있다.

[0063] 인코더(310)는 임베딩 과정에서 어텐션 계수(attention coefficient)를 계산하여 그래프의 노드 간의 중요도를 생성할 수 있다.

[0064] 어텐션 계수는 아래의 수학식 1을 통해 산출될 수 있다.

수학식 1

[0065] $$e_{ij} = a(W\vec{h}_i, W\vec{h}_j)$$

[0066] 어텐션 계수(e_{ij})는 i노드에 있어서 j노드의 중요도를 의미할 수 있다. a는 어텐션 매커니즘을 의미하며, 활성화 함수를 활용하여 두 노드 간의 유사도를 설명할 수 있다. a는 공유된 어텐션 매커니즘(shared attention mechanism)으로 공유된 셀프 어텐션(self-attention)의 파라미터들을 의미할 수 있다. (Wh_i, Wh_j)에서 W는 가중치 행렬을 의미할 수 있다. h는 노드의 중요도의 집합일 수 있다.

[0067] 상술한 바와 같이, 인코더(310)는 그래프 뉴럴 네트워크 계열의 인공지능 모델 중 그래프 어텐션 네트워크 모델을 사용하여 현재 노드에서 이웃 노드에 대해 각기 다른 가중치를 부여하기에, 공격 경로를 발견하는데 의미 있는 영향을 미칠 수 있다.

[0068] 일 실시예에서, 디코더(320)는 잠재 공간에서의 표현을 원래의 입력 데이터 형식으로 복원 또는 생성하는 네트워크 함수일 수 있다. 디코더(320)는 인코더에서 출력된 복수의 임베딩 값들을 이용하여, 복수의 노드들의 순서를 결정할 수 있다.

[0069] 디코더(320)는 마스크된 어텐션(masked attention)을 활용하여 특정 노드의 이웃 노드에 대해서만 중요도를 계산할 수 있다.

[0070] 따라서, 디코더(320)는 특정 노드와 모든 노드 간에 중요도를 전부 계산하지 않고, 특정 노드의 이웃 노드에 대해서만 중요도를 계산함으로써, 불필요한 계산을 줄여 효율적일 수 있다. 디코더(320)는 이전에 선택한 노드와 이웃한 노드들에 대해서 높은 점수를 부여하여 선택되도록 할 수 있다. 그리고 디코더(320)는 이웃하지 않는 노드들에 낮은 점수를 부여하여 선택되지 않도록 할 수 있다.

[0071] 특정 노드의 이웃 노드에 대한 중요도는 아래의 수학식 2를 통해 산출될 수 있다.

수학식 2

[0072]
$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}$$

[0073] α_{ij}는 노드 i와 노드 j 사이의 연결에 할당된 어텐션을 의미할 수 있다. $softmax_j(e_{ij})$는 e_{ij}에 대해 j를 기준으로 소프트맥스 함수를 적용하는 것을 의미할 수 있다. $exp(e_{ij})$는 e_{ij}에 대해 지수 함수를 적용한 값일 수 있다. $\Sigma_{k \in N_i} exp(e_{ik})$는 노드 i와 이웃 N_i에 있는 모든 노드 k에 대해 지수화된 점수의 합일 수 있다.

[0074] 디코더(320)는 복수의 임베딩 값들을 이용하여 특정 노드를 선택할 수 있다.

[0075] 일 실시예에서, 디코더(320)는 복수의 임베딩 값들을 이용하여, 사전 선택된 기준 노드와 엣지를 통해 연결된 적어도 하나의 제1 이웃 노드 중에서 제1 노드를 선택할 수 있다. 예를 들어, 디코더(320)는 복수의 임베딩 값들 중에서 점수가 가장 높은 임베딩 값에 대응되는 제1 노드를 선택할 수 있다.

[0076] 디코더(320)는 복수의 임베딩 값들을 이용하여, 제1 노드와 엣지를 통해 연결된 적어도 하나의 제2 이웃 노드 중에서 기준 노드와 상이한 제2 노드를 선택할 수 있다. 예를 들어, 디코더(320)는 선택된 노드에 대응되는 임베딩 값을 제외하고, 나머지 복수의 임베딩 값들 중에서 점수가 가장 높은 임베딩 값에 대응되는 제 2 노드를 선택할 수 있다.

[0077] 디코더(320)는 복수의 노드들을 모두 선택할 때까지 노드를 선택하는 과정을 반복할 수 있다. 예를 들어, 디코더(320)는 이전에 선택된 제1 노드를 제외하고 나머지 노드들 중에서 새로운 제2 노드를 선택하는 과정을 반복함으로써, 복수의 노드들을 모두 선택할 수 있다. 컴퓨팅 장치(100) 및/또는 디코더(320)는 복수의 노드들을 선택하는 순서를 복수의 노드들의 순서로 결정할 수 있다.

[0078] 일 실시예에서, 공격 그래프 생성부(330)는 그래프 데이터로부터 공격 그래프를 생성하는 모듈을 의미할 수 있다. 공격 그래프 생성부(330)는 결정된 순서에 기초하여, 복수의 노드들을 결합하여 공격 그래프를 생성할 수 있다.

[0079] 공격 그래프 생성부(330)는 복수의 노드들의 순서에 기초하여, 복수의 노드들을 각각 연결하는 복수의 엣지들의 순서를 결정하고, 그리고 복수의 노드들의 순서 및 복수의 엣지들의 순서를 표시한 공격 그래프를 생성할 수 있다.

[0080] 다시 도 2를 참조하면, 프로세서(110)는 생성된 공격 그래프 및 전처리된 학습용 그래프 데이터를 이용하여, 공격 그래프 생성 모델의 손실 값을 산출할 수 있다(230).

[0081] 일 실시예에서, 프로세서(110)는 전처리된 학습용 그래프 데이터에 기초하여, 생성된 공격 그래프로부터 보상 값을 산출할 수 있다.

[0082] 보상 값은 생성된 공격 그래프에서 하나의 노드에서 다른 노드로 가는 전체 경로에 대한 평가를 나타내는 값일 수 있다. 보상 값은 전처리된 학습용 그래프 데이터에 포함된 복수의 노드들 각각의 가중치에 기초하여 산출될

수 있다.

[0083] 일 실시예에서, 프로세서(110)는 생성된 공격 그래프에서 한 노드로부터 다른 모든 노드로 가는 경로(예를 들어, 공격 경로)를 계산할 수 있다. 그리고 프로세서(110)는 전처리된 학습용 그래프 데이터에 포함된 복수의 노드들 각각의 가중치를 이용하여 각 경로에 대한 비용을 계산할 수 있다. 프로세서(110)는 모든 경로에 대한 비용을 계산하고, 비용에 기초하여 공격 경로별로 보상 값을 결정할 수 있다. 예를 들어, 프로세서(110)는 비용이 높을수록 경로에 낮은 보상 값을 부여하고, 비용이 낮을수록 경로에 높은 보상 값을 부여할 수 있다. 경로는 노드 간에 연결되는 엣지를 포함할 수 있다.

[0084] 일 실시예에서, 보상 값은 그래프 어텐션 네트워크 모델이 생성하는 그래프의 정밀도를 향상시키기 위한 강화학습 과정에서 사용되는 값일 수 있다. 보상 값은 단일 출발지 최단 경로(Single-source Shortest Path, SSP) 알고리즘을 사용하여 산출될 수 있다. 즉, 프로세서(110)는 단일 출발지 최단 경로 알고리즘을 사용하여 보상 값을 산출할 수 있다. 프로세서(110)는 단일 출발지 최단 경로 알고리즘을 이용함으로써, 공격 경로가 포함된 공격 그래프를 생성하는 과정에서 공격자가 목표로 하는 타겟 노드에 도달하는 경로 중 가장 효율적인 경로를 찾을 수 있다.

[0085] 단일 출발지 최단 경로 알고리즘은 아래의 수학식 3에 대응될 수 있다.

수학식 3

[0086]
$$r = -\sum_{i=1}^{|V|} \sum_{e \in Q_i} W(e)$$

[0087] r은 보상 값일 수 있다. r은 각 경로의 가중치 합을 나타내며, 부호를 반대로 취한 값일 수 있다. $\sum_{i=1}^{|V|}$은 그래프 내 모든 노드에 대해 반복하는 합일 수 있다. i는 노드의 인덱스일 수 있다. $|V|$는 노드의 총 개수일 수 있다. $\sum_{e \in Q_i}$는 노드 i에서 특정 경로 Q_i에 포함된 모든 엣지 e에 대한 합일 수 있다. 즉, 노드 i에서 다른 노드로 가는 경로를 구성하는 모든 엣지의 가중치의 합일 수 있다. $W(e)$는 엣지 e의 가중치일 수 있다.

[0088] 보상 값은 노드의 가중치의 총합의 음수 값일 수 있다. 예를 들어, 프로세서(110)는 모든 노드를 선택하는 단계를 $i = 1$부터 노드 개수만큼 진행할 수 있다. 프로세서(110)는 각 단계마다 선택된 노드의 가중치 총합을 계산할 수 있다. 프로세서(110)는 최종적으로 계산된 모든 가중치 총합을 한 번 더 총합하여 음수 값으로 표현한 값을 보상 값으로 사용할 수 있다. 프로세서(110)는 보상 값을 음수 값으로 사용함으로써, 노드의 가중치가 작을수록 임베딩 값이 커지도록 설정할 수 있다. 따라서, 프로세서(110)는 노드를 선택하는 단계에서 특정 노드와 인접한 복수의 이웃 노드들 중에서 임베딩 값이 가장 큰 노드(노드의 가중치가 가장 작은 노드)를 선택할 수 있다.

[0089] 일 실시예에서, 프로세서(110)는 베이스라인 모델을 이용하여, 전처리된 그래프 데이터로부터 베이스라인 값을 산출할 수 있다.

[0090] 베이스라인 모델은 공격 그래프 생성 모델의 강화 학습을 위해 기준이 되는 성능을 나타내는 모델일 수 있다. 베이스라인 모델의 구조는 공격 그래프 생성 모델과 대응될 수 있다. 다만, 베이스라인 모델의 구조 및/또는 종류는 이에 한정되지 않는다. 베이스라인 모델의 신경망 가중치의 값은 공격 그래프 생성 모델의 학습 단계 중 현재 단계의 이전 단계의 신경망 가중치의 값에 대응될 수 있다. 일 실시예에서, 베이스라인 모델의 신경망 가중치의 값은 사전 결정된 시간(예를 들어, 1분, 10분 등)마다 공격 그래프 생성 모델의 현재 단계의 이전 단계의 신경망 가중치의 값으로 업데이트될 수 있다. 일 실시예에서, 공격 그래프 생성 모델의 학습 과정에서 공격 그래프 생성 모델의 가중치가 업데이트되는 경우, 베이스라인 모델의 신경망 가중치의 값은 공격 그래프 생성 모델의 현재 단계의 이전 단계의 신경망 가중치의 값으로 업데이트될 수 있다.

[0091] 프로세서(110)는 베이스라인 모델을 이용하여 공격 그래프 생성 모델에서 보상을 출력하는 과정과 동일한 과정을 수행함으로써, 베이스라인 모델에서 출력되는 보상 값인 베이스라인 값을 출력할 수 있다. 프로세서(110)는

베이스라인 값을 이용하여 보상 값의 분산을 줄일 수 있다.

[0092] 프로세서(110)는 보상 값 및 베이스라인 값을 이용하여, 손실 값을 산출할 수 있다.

[0093] 손실 값은 모델의 예측 값과 실제 타겟 값 사이의 차이를 나타내는 값일 수 있다. 프로세서(110)는 REINFORCE 알고리즘을 이용하여 손실 값을 산출할 수 있다. 다만, 손실 값을 산출하는 알고리즘은 이에 한정되지 않는다.

[0094] REINFORCE 알고리즘은 아래의 수학식 4에 대응될 수 있다.

수학식 4

[0095]
$$loss = \sum_{t=1}^{N} (r_t - b) \log p(a_t)$$

[0096] loss는 손실 값을 의미할 수 있다. N은 디코더를 동작시키는 프로세스를 수식화한 것일 수 있다. N은 노드 개수에서 1을 뺀 자연수일 수 있다. t는 디코딩(decoding)의 단계를 나타낼 수 있다. t가 N까지 증가하며 각 단계를 수행하면 그 단계에서 계산되는 보상 값과 베이스라인 값을 연산할 수 있다. 예를 들어, t=3인 경우, 디코더가 3번째로 디코딩을 수행했을 때의 단계를 나타낼 수 있다. r_t는 t번째 디코딩을 했을 때 계산된 보상 값일 수 있다. b는 베이스라인 값일 수 있다. $p(a_t)$는 t번째 디코딩을 했을 때 선택했던 노드의 선택 확률 값일 수 있다.

[0097] 프로세서(110)는 손실 값을 공격 그래프 생성 모델에 반영함으로써 공격 그래프 생성 모델을 학습시킬 수 있다(240). 예를 들어, 프로세서(110)는 손실 값을 공격 그래프 생성 모델에 반영함으로써, 공격 그래프 생성 모델의 신경망 가중치를 업데이트할 수 있다.

[0098] 도 4 내지 도 6d을 참조하여 후술될 각 구성들(예를 들어, 컴퓨팅 장치(100), 공격 그래프 생성 모델(300) 등)에 대한 구체적인 설명은 도 1 내지 도 3을 참조하여 상술한 내용으로 대체될 수 있으며, 중복되는 내용은 생략될 수 있다.

[0099] 도 4는 본 개시의 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 공격 그래프 생성 모델을 이용하여 공격 그래프를 생성하는 과정을 나타낸 도면이다.

[0100] 도 4를 참조하면, 인코더(420)는 전처리된 학습용 그래프 데이터(410)를 입력 받고, 노드의 임베딩을 수행할 수 있다(421).

[0101] 인코더(420)는 노드의 임베딩을 통해 그래프의 각각 노드 간의 중요도에 대한 정보를 포함하는 복수의 임베딩 값들을 출력할 수 있다(422).

[0102] 디코더(430)는 출력된 복수의 임베딩 값들(422)을 입력 받고, 노드별로 점수를 산출할 수 있다(431). 노드의 점수는 노드의 중요도(또는 가중치)에 따라 결정되는 점수를 의미할 수 있다.

[0103] 디코더(430)는 노드별로 산출된 점수에 기초하여 노드를 선택할 수 있다(432). 예를 들어, 디코더(430)는 노드별로 산출된 점수 중에서 가장 높은 점수에 대응되는 노드를 선택할 수 있다.

[0104] 디코더(430)는 노드를 선택하는 과정(432) 이후, 선택되지 않은 노드가 존재하는 경우, 노드 점수를 산출하는 과정(431) 및 노드를 선택하는 과정(432)을 다시 반복할 수 있다. 디코더(430)는 노드 점수를 산출하는 과정(431) 및 노드를 선택하는 과정(432)을 반복함으로써, 복수의 노드들을 모두 선택할 수 있다. 디코더(430)는 복수의 노드들을 선택하는 순서를 복수의 노드들의 순서로 결정할 수 있다.

[0105] 공격 그래프 생성부(440)는 디코더(430)에서 결정된 복수의 노드들의 순서에 기초하여, 복수의 노드들을 각각 연결하는 복수의 엣지들의 순서를 결정하고, 그리고 복수의 노드들의 순서 및 복수의 엣지들의 순서를 표시한 공격 그래프를 생성할 수 있다(441).

[0106] 도 5는 본 개시의 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 손실 값을 산출하는 과정을 나타낸 도면이다.

[0107] 도 5를 참조하면, 프로세서(110)는 전처리된 학습용 그래프 데이터(510) 및 공격 그래프(520)를 이용하여, 보상 값을 산출할 수 있다(530). 일 실시예에서, 프로세서(110)는 생성된 공격 그래프에서 한 노드로부터 다른 모든

노드로 가는 경로(예를 들어, 공격 경로)를 계산할 수 있다. 그리고 프로세서(110)는 전처리된 학습용 그래프 데이터에 포함된 복수의 노드들 각각의 가중치를 이용하여 각 경로에 대한 비용을 계산할 수 있다. 프로세서(110)는 모든 경로에 대한 비용을 계산하고, 비용에 기초하여 공격 경로별로 보상 값을 결정할 수 있다.

[0108] 프로세서(110)는 전처리된 학습용 그래프 데이터(510)를 이용하여, 베이스라인 값을 산출할 수 있다(540). 일 실시예에서, 프로세서(110)는 전처리된 학습용 그래프 데이터(510) 및 공격 그래프(520)를 이용하여, 베이스라인 값을 산출할 수 있다. 예를 들어, 프로세서(110)는 베이스라인 모델을 이용하여 공격 그래프 생성 모델에서 보상을 출력하는 과정과 동일한 과정을 수행함으로써, 베이스라인 모델에서 출력되는 보상 값인 베이스라인 값을 출력할 수 있다.

[0109] 프로세서(110)는 보상 값 및 베이스라인 값을 이용하여, 손실 값을 산출할 수 있다(550). 프로세서(110)는 REINFORCE 알고리즘을 이용하여 손실 값을 산출할 수 있다.

[0110] 도 6a는 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 노드 선택 및 그래프 생성 과정의 제1 단계를 나타낸 도면이다.

[0111] 도 6a를 참조하면, 그래프(예를 들어, 공격 그래프)(600)의 노드는 CVE 데이터베이스 기반의 취약점 정보를 가지고 있으며, CVSS 점수를 기반으로 노드 가중치를 가질 수 있다. 각 노드와 노드는 무방향으로 연결될 수 있다. 노드 n_i은 m개가 존재할 수 있으며 이는 n_i으로 표현될 수 있다. 각 노드가 가지는 가중치는 인코더를 거쳐 노드 임베딩 값(e_i)으로 변환될 수 있다.

[0112] 그래프(600)에서 점선 테두리의 노드 및 엣지는 본래 그래프로, 아직 선택되지 않는 노드 및 엣지를 의미할 수 있다. 또한, 그래프(600)에서 실선 테두리의 노드 및 엣지는 선택된 노드 및 엣지를 의미할 수 있다.

[0113] 그래프(600)에서 사전 선택된 기준 노드는 제1 노드(610)일 수 있다. 따라서, 프로세서(110)는 제1 노드(610)를 선택할 수 있다.

[0114] 도 6b는 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 노드 선택 및 그래프 생성 과정의 제2 단계를 나타낸 도면이다.

[0115] 도 6b를 참조하면, 프로세서(110)는 그래프(600)에서 기존에 선택된 제1 노드(610)를 제외하고 나머지 인접한 노드들 중에서 임베딩 값이 가장 작은 제2 노드(620)를 선택할 수 있다. 그리고, 프로세서(110)는 제1 노드(610)와 제2 노드(620) 간에 연결된 엣지를 선택할 수 있다.

[0116] 도 6c는 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 노드 선택 및 그래프 생성 과정의 제3 단계를 나타낸 도면이다.

[0117] 도 6c를 참조하면, 프로세서(110)는 그래프(600)에서 기존에 선택된 제1 노드(610) 및 제2 노드(620)를 제외하고 나머지 인접한 노드들 중에서 임베딩 값이 가장 작은 제3 노드(630)를 선택할 수 있다. 그리고, 프로세서(110)는 제1 노드(610)와 제3 노드(630) 간에 연결된 엣지를 선택할 수 있다.

[0118] 도 6d는 일 실시예에 따른 컴퓨팅 장치에서 수행되는, 노드 선택 및 그래프 생성 과정의 제4 단계를 나타낸 도면이다.

[0119] 도 6d를 참조하면, 프로세서(110)는 그래프(600)에서 기존에 선택된 제1 노드(610), 제2 노드(620) 및 제3 노드(630)를 제외하고 나머지 인접한 노드들 중에서 임베딩 값이 가장 작은 제4 노드(640)를 선택할 수 있다. 그리고, 프로세서(110)는 제1 노드(610)와 제4 노드(640) 간에 직접적으로 연결된 엣지가 아닌, 제2 노드(620)와 제4 노드(640) 간에 연결된 엣지를 선택할 수 있다.

[0120] 도 7은 본 개시의 일 실시예에 따른 인공지능 모델의 예시적인 구조를 나타낸 도면이다.

[0121] 본 명세서에 걸쳐, 인공지능 모델, 인공지능 기반 모델, 연산 모델, 신경망, 네트워크 함수, 뉴럴 네트워크(neural network)는 동일한 의미로 사용될 수 있다.

[0122] 신경망은 일반적으로 노드라 지칭될 수 있는 상호 연결된 계산 단위들의 집합으로 구성될 수 있다. 이러한 노드들은 뉴런(neuron)들로 지칭될 수도 있다. 신경망은 적어도 하나 이상의 노드들을 포함하여 구성된다. 신경망들을 구성하는 노드(또는 뉴런)들은 하나 이상의 링크에 의해 상호 연결될 수 있다.

[0123] 신경망 내에서, 링크를 통해 연결된 하나 이상의 노드들은 상대적으로 입력 노드 및 출력 노드의 관계를 형성할 수 있다. 입력 노드 및 출력 노드의 개념은 상대적인 것으로서, 하나의 노드에 대하여 출력 노드 관계에 있는

임의의 노드는 다른 노드와의 관계에서 입력 노드 관계에 있을 수 있으며, 그 역도 성립할 수 있다. 상술한 바와 같이, 입력 노드 대 출력 노드 관계는 링크를 중심으로 생성될 수 있다. 하나의 입력 노드에 하나 이상의 출력 노드가 링크를 통해 연결될 수 있으며, 그 역도 성립할 수 있다.

[0124] 하나의 링크를 통해 연결된 입력 노드 및 출력 노드 관계에서, 출력 노드의 데이터는 입력 노드에 입력된 데이터에 기초하여 그 값이 결정될 수 있다. 여기서 입력 노드와 출력 노드를 상호 연결하는 링크는 가중치(weight)를 가질 수 있다. 가중치는 가변적일 수 있으며, 신경망이 원하는 기능을 수행하기 위해, 사용자 또는 알고리즘에 의해 가변 될 수 있다. 예를 들어, 하나의 출력 노드에 하나 이상의 입력 노드가 각각의 링크에 의해 상호 연결된 경우, 출력 노드는 상기 출력 노드와 연결된 입력 노드들에 입력된 값들 및 각각의 입력 노드들에 대응하는 링크에 설정된 가중치에 기초하여 출력 노드 값을 결정할 수 있다.

[0125] 상술한 바와 같이, 신경망은 하나 이상의 노드들이 하나 이상의 링크를 통해 상호 연결되어 신경망 내에서 입력 노드 및 출력 노드 관계를 형성한다. 신경망 내에서 노드들과 링크들의 개수 및 노드들과 링크들 사이의 연관관계, 링크들 각각에 부여된 가중치의 값에 따라, 신경망의 특성이 결정될 수 있다. 예를 들어, 동일한 개수의 노드 및 링크들이 존재하고, 링크들의 가중치 값이 상이한 두 신경망이 존재하는 경우, 두 개의 신경망들은 서로 상이한 것으로 인식될 수 있다.

[0126] 신경망은 하나 이상의 노드들의 집합으로 구성될 수 있다. 신경망을 구성하는 노드들의 부분 집합은 레이어(layer)를 구성할 수 있다. 신경망을 구성하는 노드들 중 일부는, 최초 입력 노드로부터의 거리들에 기초하여, 하나의 레이어(layer)를 구성할 수 있다. 예를 들어, 최초 입력 노드로부터 거리가 n인 노드들의 집합은, n 레이어를 구성할 수 있다. 최초 입력 노드로부터 거리는, 최초 입력 노드로부터 해당 노드까지 도달하기 위해 거쳐야 하는 링크들의 최소 개수에 의해 정의될 수 있다. 그러나, 이러한 레이어의 정의는 설명을 위한 임의적인 것으로서, 신경망 내에서 레이어의 차수는 상술한 것과 상이한 방법으로 정의될 수 있다. 예를 들어, 노드들의 레이어는 최종 출력 노드로부터 거리에 의해 정의될 수도 있다.

[0127] 본 개시내용의 일 실시예에서, 뉴런들 또는 노드들의 집합은 레이어라는 표현으로 정의될 수 있다.

[0128] 최초 입력 노드는 신경망 내의 노드들 중 다른 노드들과의 관계에서 링크를 거치지 않고 데이터가 직접 입력되는 하나 이상의 노드들을 의미할 수 있다. 또는, 신경망 네트워크 내에서, 링크를 기준으로 한 노드 간의 관계에 있어서, 링크로 연결된 다른 입력 노드들을 가지지 않는 노드들을 의미할 수 있다. 이와 유사하게, 최종 출력 노드는 신경망 내의 노드들 중 다른 노드들과의 관계에서, 출력 노드를 가지지 않는 하나 이상의 노드들을 의미할 수 있다. 또한, 히든 노드는 최초 입력 노드 및 최후 출력 노드가 아닌 신경망을 구성하는 노드들을 의미할 수 있다.

[0129] 본 개시의 일 실시예에 따른 신경망은 입력 레이어의 노드의 개수가 출력 레이어의 노드의 개수와 동일할 수 있으며, 입력 레이어에서 히든 레이어로 진행됨에 따라 노드의 수가 감소하다가 다시 증가하는 형태의 신경망일 수 있다. 또한, 본 개시의 다른 일 실시예에 따른 신경망은 입력 레이어의 노드의 개수가 출력 레이어의 노드의 개수 보다 적을 수 있으며, 입력 레이어에서 히든 레이어로 진행됨에 따라 노드의 수가 감소하는 형태의 신경망일 수 있다. 또한, 본 개시의 또 다른 일 실시예에 따른 신경망은 입력 레이어의 노드의 개수가 출력 레이어의 노드의 개수보다 많을 수 있으며, 입력 레이어에서 히든 레이어로 진행됨에 따라 노드의 수가 증가하는 형태의 신경망일 수 있다. 본 개시의 또 다른 일 실시예에 따른 신경망은 상술한 신경망들의 조합된 형태의 신경망일 수 있다.

[0130] 딥 뉴럴 네트워크(DNN: deep neural network, 심층신경망)는 입력 레이어와 출력 레이어 외에 복수의 히든 레이어를 포함하는 신경망을 의미할 수 있다. 딥 뉴럴 네트워크를 이용하면 데이터의 잠재적인 구조(latent structures)를 파악할 수 있다. 즉, 사진, 글, 비디오, 음성, 단백질 시퀀스 구조, 유전자 시퀀스 구조, 펩타이드 서열의 구조, 음악의 잠재적인 구조(예를 들어, 어떤 물체가 사진에 있는지, 글의 내용과 감정이 무엇인지, 음성의 내용과 감정이 무엇인지 등), 및/또는 펩타이드와 MHC 간의 결합 친화도를 파악할 수 있다. 딥 뉴럴 네트워크는 컨볼루션 뉴럴 네트워크(CNN: convolutional neural network), 리커런트 뉴럴 네트워크(RNN: recurrent neural network), 오토 인코더(auto encoder), 제한 볼츠만 머신(RBM: restricted boltzmann machine), 심층 신뢰 네트워크(DBN: deep belief network), Q 네트워크, U 네트워크, 샴 네트워크, 적대적 생성 네트워크(GAN: Generative Adversarial Network), 트랜스포머 등을 포함할 수 있다. 전술한 딥 뉴럴 네트워크의 기재는 예시일 뿐이며 본 개시는 이에 제한되지 않는다.

[0131] 본 개시내용의 인공지능 기반 모델은 입력 레이어, 히든 레이어 및 출력 레이어를 포함하는 전술한 임의의 구조

의 네트워크 구조에 의해 표현될 수 있다.

[0132] 본 개시내용의 인공지능 기반 모델에서 사용될 수 있는 뉴럴 네트워크는 지도 학습(supervised learning), 비지도 학습(unsupervised learning), 반지도학습(semi supervised learning), 전이 학습(transfer learning), 능동 학습(active learning) 또는 강화학습(reinforcement learning) 중 적어도 하나의 방식으로 학습될 수 있다. 뉴럴 네트워크의 학습은 뉴럴 네트워크가 특정한 동작을 수행하기 위한 지식을 뉴럴 네트워크에 적용하는 과정일 수 있다.

[0133] 뉴럴 네트워크는 출력의 오류를 최소화하는 방향으로 학습될 수 있다. 뉴럴 네트워크의 학습에서 반복적으로 학습 데이터를 뉴럴 네트워크에 입력시키고 학습 데이터에 대한 뉴럴 네트워크의 출력과 타겟의 에러를 계산하고, 에러를 줄이기 위한 방향으로 뉴럴 네트워크의 에러를 뉴럴 네트워크의 출력 레이어에서부터 입력 레이어 방향으로 역전파(backpropagation)하여 뉴럴 네트워크의 각 노드의 가중치를 업데이트 하는 과정이다. 지도 학습의 경우 각각의 학습 데이터에 정답이 라벨링되어있는 학습 데이터를 사용하며(즉, 라벨링된 학습 데이터), 비지도 학습의 경우는 각각의 학습 데이터에 정답이 라벨링되어 있지 않을 수 있다. 즉, 예를 들어 데이터 분류에 관한 지도 학습의 경우의 학습 데이터는 학습 데이터 각각에 카테고리가 라벨링 된 데이터 일 수 있다. 라벨링된 학습 데이터가 뉴럴 네트워크에 입력되고, 뉴럴 네트워크의 출력(카테고리)과 학습 데이터의 라벨을 비교함으로써 오류(error)가 계산될 수 있다. 다른 예로, 데이터 분류에 관한 비지도 학습의 경우 입력인 학습 데이터가 뉴럴 네트워크 출력과 비교됨으로써 오류가 계산될 수 있다. 계산된 오류는 뉴럴 네트워크에서 역방향(즉, 출력 레이어에서 입력 레이어 방향)으로 역전파 되며, 역전파에 따라 뉴럴 네트워크의 각 레이어의 각 노드들의 연결 가중치가 업데이트 될 수 있다. 업데이트 되는 각 노드의 연결 가중치는 학습률(learning rate)에 따라 변화량이 결정될 수 있다. 입력 데이터에 대한 뉴럴 네트워크의 계산과 에러의 역전파는 학습 사이클(epoch)을 구성할 수 있다. 학습률은 뉴럴 네트워크의 학습 사이클의 반복 횟수에 따라 상이하게 적용될 수 있다. 예를 들어, 뉴럴 네트워크의 학습 초기에는 높은 학습률을 사용하여 뉴럴 네트워크가 빠르게 일정 수준의 성능을 확보하도록 하여 효율성을 높이고, 학습 후기에는 낮은 학습률을 사용하여 정확도를 높일 수 있다.

[0134] 뉴럴 네트워크의 학습에서 일반적으로 학습 데이터는 실제 데이터(즉, 학습된 뉴럴 네트워크를 이용하여 처리하고자 하는 데이터)의 부분집합일 수 있으며, 따라서, 학습 데이터에 대한 오류는 감소하나 실제 데이터에 대해서는 오류가 증가하는 학습 사이클이 존재할 수 있다. 과적합(overfitting)은 이와 같이 학습 데이터에 과하게 학습하여 실제 데이터에 대한 오류가 증가하는 현상이다. 예를 들어, 노란색 고양이를 보여 고양이를 학습한 뉴럴 네트워크가 노란색 이외의 고양이를 보고는 고양이임을 인식하지 못하는 현상이 과적합의 일종일 수 있다. 과적합은 머신러닝 알고리즘의 오류를 증가시키는 원인으로 작용할 수 있다. 이러한 과적합을 막기 위하여 다양한 최적화 방법이 사용될 수 있다. 과적합을 막기 위해서는 학습 데이터를 증가시키거나, 레귤라리제이션(regularization), 학습의 과정에서 네트워크의 노드 일부를 비활성화하는 드롭아웃(dropout), 배치 정규화 레이어(batch normalization layer)의 활용 등의 방법이 적용될 수 있다.

[0135] 본 개시의 일 실시예에 따른 데이터 구조를 저장한 컴퓨터 판독가능 매체가 개시된다. 전술한 데이터 구조는 본 개시내용에서의 저장부에 저장될 수 있으며, 프로세서에 의해 실행될 수 있으며 그리고 통신부에 의해 송수신될 수 있다.

[0136] 데이터 구조는 데이터에 효율적인 접근 및 수정을 가능하게 하는 데이터의 조직, 관리, 저장을 의미할 수 있다. 데이터 구조는 특정 문제(예를 들어, 데이터 분석, 데이터 검색, 데이터 저장, 데이터 수정) 해결을 위한 데이터의 조직을 의미할 수 있다. 데이터 구조는 특정한 데이터 처리 기능을 지원하도록 설계된, 데이터 요소들 간의 물리적이거나 논리적인 관계로 정의될 수도 있다. 데이터 요소들 간의 논리적인 관계는 사용자 정의 데이터 요소들 간의 연결관계를 포함할 수 있다. 데이터 요소들 간의 물리적인 관계는 컴퓨터 판독가능 저장매체(예를 들어, 영구 저장 장치)에 물리적으로 저장되어 있는 데이터 요소들 간의 실제 관계를 포함할 수 있다. 데이터 구조는 구체적으로 데이터의 집합, 데이터 간의 관계, 데이터에 적용할 수 있는 함수 또는 명령어를 포함할 수 있다. 효과적으로 설계된 데이터 구조를 통해 컴퓨팅 장치는 컴퓨팅 장치의 자원을 최소한으로 사용하면서 연산을 수행할 수 있다. 구체적으로 컴퓨팅 장치는 효과적으로 설계된 데이터 구조를 통해 연산, 읽기, 삽입, 삭제, 비교, 교환 및 검색의 효율성을 높일 수 있다.

[0137] 데이터 구조는 데이터 구조의 형태에 따라 선형 데이터 구조와 비선형 데이터 구조로 구분될 수 있다. 선형 데이터 구조는 하나의 데이터 뒤에 하나의 데이터만이 연결되는 구조일 수 있다. 선형 데이터 구조는 리스트(List), 스택(Stack), 큐(Queue), 데크(Deque)를 포함할 수 있다. 리스트는 내부적으로 순서가 존재하는 일련의 데이터 집합을 의미할 수 있다. 리스트는 연결 리스트(Linked List)를 포함할 수 있다. 연결 리스트는 각각의

데이터가 포인터를 가지고 한 줄로 연결되어 있는 방식으로 데이터가 연결된 데이터 구조일 수 있다. 연결 리스트에서 포인터는 다음이나 이전 데이터와의 연결 정보를 포함할 수 있다. 연결 리스트는 형태에 따라 단일 연결 리스트, 이중 연결 리스트, 원형 연결 리스트로 표현될 수 있다. 스택은 제한적으로 데이터에 접근할 수 있는 데이터 나열 구조일 수 있다. 스택은 데이터 구조의 한 쪽 끝에서만 데이터를 처리(예를 들어, 삽입 또는 삭제)할 수 있는 선형 데이터 구조일 수 있다. 스택에 저장된 데이터는 늦게 들어갈수록 빨리 나오는 데이터 구조(LIFO-Last in First Out)일 수 있다. 큐는 제한적으로 데이터에 접근할 수 있는 데이터 나열 구조로서, 스택과 달리 늦게 저장된 데이터일수록 늦게 나오는 데이터 구조(FIFO-First in First Out)일 수 있다. 데크는 데이터 구조의 양 쪽 끝에서 데이터를 처리할 수 있는 데이터 구조일 수 있다.

[0138] 비선형 데이터 구조는 하나의 데이터 뒤에 복수개의 데이터가 연결되는 구조일 수 있다. 비선형 데이터 구조는 그래프(Graph) 데이터 구조를 포함할 수 있다. 그래프 데이터 구조는 정점(Vertex)과 간선(Edge)으로 정의될 수 있으며 간선은 서로 다른 두개의 정점을 연결하는 선을 포함할 수 있다. 그래프 데이터 구조 트리(Tree) 데이터 구조를 포함할 수 있다. 트리 데이터 구조는 트리에 포함된 복수개의 정점 중에서 서로 다른 두개의 정점을 연결시키는 경로가 하나인 데이터 구조일 수 있다. 즉 그래프 데이터 구조에서 루프(loop)를 형성하지 않는 데이터 구조일 수 있다.

[0139] 본 명세서에 걸쳐, 인공지능 기반 모델, 연산 모델, 신경망, 네트워크 함수, 뉴럴 네트워크(neural network)는 상호 교환 가능한 의미로 사용될 수 있다. 이하에서는 신경망으로 통일하여 기술한다. 데이터 구조는 신경망을 포함할 수 있다. 그리고 신경망을 포함한 데이터 구조는 컴퓨터 판독가능 매체에 저장될 수 있다. 신경망을 포함한 데이터 구조는 또한 신경망에 의한 처리를 위하여 전처리된 데이터, 신경망에 입력되는 데이터, 신경망의 가중치, 신경망의 하이퍼 파라미터, 신경망으로부터 획득한 데이터, 신경망의 각 노드 또는 레이어와 연관된 활성 함수, 신경망의 학습을 위한 손실 함수 등을 포함할 수 있다. 신경망을 포함한 데이터 구조는 상기 개시된 구성들 중 임의의 구성 요소들을 포함할 수 있다. 즉 신경망을 포함한 데이터 구조는 신경망에 의한 처리를 위하여 전처리된 데이터, 신경망에 입력되는 데이터, 신경망의 가중치, 신경망의 하이퍼 파라미터, 신경망으로부터 획득한 데이터, 신경망의 각 노드 또는 레이어와 연관된 활성 함수, 신경망의 학습을 위한 손실 함수 등 전부 또는 이들의 임의의 조합을 포함하여 구성될 수 있다. 전술한 구성들 이외에도, 신경망을 포함한 데이터 구조는 신경망의 특성을 결정하는 임의의 다른 정보를 포함할 수 있다. 또한, 데이터 구조는 신경망의 연산 과정에 사용되거나 발생되는 모든 형태의 데이터를 포함할 수 있으며 전술한 사항에 제한되는 것은 아니다. 컴퓨터 판독가능 매체는 컴퓨터 판독가능 기록 매체 및/또는 컴퓨터 판독가능 전송 매체를 포함할 수 있다. 신경망은 일반적으로 노드라 지칭될 수 있는 상호 연결된 계산 단위들의 집합으로 구성될 수 있다. 이러한 노드들은 뉴런(neuron)들로 지칭될 수도 있다. 신경망은 적어도 하나 이상의 노드들을 포함하여 구성된다.

[0140] 데이터 구조는 신경망에 입력되는 데이터를 포함할 수 있다. 신경망에 입력되는 데이터를 포함하는 데이터 구조는 컴퓨터 판독가능 매체에 저장될 수 있다. 신경망에 입력되는 데이터는 신경망 학습 과정에서 입력되는 학습 데이터 및/또는 학습이 완료된 신경망에 입력되는 입력 데이터를 포함할 수 있다. 신경망에 입력되는 데이터는 전처리(pre-processing)를 거친 데이터 및/또는 전처리 대상이 되는 데이터를 포함할 수 있다. 전처리는 데이터를 신경망에 입력시키기 위한 데이터 처리 과정을 포함할 수 있다. 따라서 데이터 구조는 전처리 대상이 되는 데이터 및 전처리로 발생되는 데이터를 포함할 수 있다. 전술한 데이터 구조는 예시일 뿐 본 개시는 이에 제한되지 않는다.

[0141] 데이터 구조는 신경망의 가중치를 포함할 수 있다. (본 명세서에서 가중치, 파라미터는 상호 교환가능한 의미로 사용될 수 있다.) 그리고 신경망의 가중치를 포함한 데이터 구조는 컴퓨터 판독가능 매체에 저장될 수 있다. 신경망은 복수개의 가중치를 포함할 수 있다. 가중치는 가변적일 수 있으며, 신경망이 원하는 기능을 수행하기 위해, 사용자 또는 알고리즘에 의해 가변 될 수 있다. 예를 들어, 하나의 출력 노드에 하나 이상의 입력 노드가 각각의 링크에 의해 상호 연결된 경우, 출력 노드는 상기 출력 노드와 연결된 입력 노드들에 입력된 값들 및 각각의 입력 노드들에 대응하는 링크에 설정된 가중치에 기초하여 출력 노드에서 출력되는 데이터 값을 결정할 수 있다. 전술한 데이터 구조는 예시일 뿐 본 개시는 이에 제한되지 않는다.

[0142] 제한이 아닌 예로서, 가중치는 신경망 학습 과정에서 가변되는 가중치 및/또는 신경망 학습이 완료된 가중치를 포함할 수 있다. 신경망 학습 과정에서 가변되는 가중치는 학습 사이클이 시작되는 시점의 가중치 및/또는 학습 사이클 동안 가변되는 가중치를 포함할 수 있다. 신경망 학습이 완료된 가중치는 학습 사이클이 완료된 가중치를 포함할 수 있다. 따라서 신경망의 가중치를 포함한 데이터 구조는 신경망 학습 과정에서 가변되는 가중치 및/또는 신경망 학습이 완료된 가중치를 포함한 데이터 구조를 포함할 수 있다. 그러므로 상술한 가중치 및/또는 각 가중치의 조합은 신경망의 가중치를 포함한 데이터 구조에 포함되는 것으로 한다. 전술한 데이터 구조는 예

시일 뿐 본 개시는 이에 제한되지 않는다.

[0143] 신경망의 가중치를 포함한 데이터 구조는 직렬화(serialization) 과정을 거친 후 컴퓨터 판독가능 저장 매체(예를 들어, 메모리, 하드 디스크)에 저장될 수 있다. 직렬화는 데이터 구조를 동일하거나 다른 컴퓨팅 장치에 저장하고 나중에 다시 재구성하여 사용할 수 있는 형태로 변환하는 과정일 수 있다. 컴퓨팅 장치는 데이터 구조를 직렬화하여 네트워크를 통해 데이터를 송수신할 수 있다. 직렬화된 신경망의 가중치를 포함한 데이터 구조는 역직렬화(deserialization)를 통해 동일한 컴퓨팅 장치 또는 다른 컴퓨팅 장치에서 재구성될 수 있다. 신경망의 가중치를 포함한 데이터 구조는 직렬화에 한정되는 것은 아니다. 나아가 신경망의 가중치를 포함한 데이터 구조는 컴퓨팅 장치의 자원을 최소한으로 사용하면서 연산의 효율을 높이기 위한 데이터 구조(예를 들어, 비선형 데이터 구조에서 B-Tree, R-Tree, Trie, m-way search tree, AVL tree, Red-Black Tree)를 포함할 수 있다. 전술한 사항은 예시일 뿐 본 개시는 이에 제한되지 않는다.

[0144] 데이터 구조는 신경망의 하이퍼 파라미터(Hyper-parameter)를 포함할 수 있다. 그리고 신경망의 하이퍼 파라미터를 포함한 데이터 구조는 컴퓨터 판독가능 매체에 저장될 수 있다. 하이퍼 파라미터는 사용자에 의해 가변되는 변수일 수 있다. 하이퍼 파라미터는 예를 들어, 학습률(learning rate), 비용 함수(cost function), 학습 사이클 반복 횟수, 가중치 초기화(Weight initialization)(예를 들어, 가중치 초기화 대상이 되는 가중치 값의 범위 설정), Hidden Unit 개수(예를 들어, 히든 레이어의 개수, 히든 레이어의 노드 수)를 포함할 수 있다. 전술한 데이터 구조는 예시일 뿐 본 개시는 이에 제한되지 않는다.

[0145] 본 개시의 일 실시예에 따른 공격 그래프 생성 모델에 대한 네트워크 함수로서 트랜스포머(transformer)가 고려될 수도 있다. 일례로, 공격 그래프 생성 모델은 어텐션 알고리즘이 적용된 순환 신경망 또는 어텐션 알고리즘이 적용된 트랜스포머를 사용하여 동작될 수 있다.

[0146] 일 실시예에서, 트랜스포머는 임베딩된 데이터들을 인코딩하는 인코더 및 인코딩된 데이터들을 디코딩하는 디코더로 구성될 수 있다. 트랜스포머는 일련의 데이터(a series of data)들을 수신하여, 인코딩 및 디코딩 단계를 거쳐 상이한 타입의 일련의 데이터들을 출력하는 구조를 지닐 수 있다. 일 실시예에서, 일련의 데이터들은 트랜스포머가 연산가능한 형태로 가공될 수 있다. 일련의 데이터들을 트랜스포머가 연산가능한 형태로 가공하는 과정은 임베딩 과정을 포함할 수 있다. 데이터 토큰, 임베딩 벡터, 임베딩 토큰 등과 같은 표현들은, 트랜스포머가 처리할 수 있는 형태로 임베딩된 데이터들을 지칭하는 것일 수 있다.

[0147] 트랜스포머가 일련의 데이터들을 인코딩 및 디코딩하기 위하여, 트랜스포머 내의 인코더 및 디코더들을 어텐션(attention) 알고리즘을 활용하여 처리할 수 있다. 어텐션 알고리즘이란 주어진 쿼리(Query)에 대해, 하나 이상의 키(Key)에 대한 유사도를 구하고, 이렇게 주어진 유사도를, 각각의 키(Key)와 대응하는 값(Value)에 반영한 후, 유사도가 반영된 값(Value)들을 가중합하여 어텐션(attention) 값을 계산하는 알고리즘을 의미할 수 있다.

[0148] 쿼리(Query), 키(Key) 및 값(Value)를 어떻게 설정하느냐에 따라, 다양한 종류의 어텐션 알고리즘이 분류될 수 있다. 예를 들어, 쿼리, 키 및 값을 모두 동일하게 설정하여 어텐션을 구하는 경우, 이는 셀프-어텐션 알고리즘을 의미할 수 있다. 입력된 일련의 데이터들을 병렬로 처리하기 위해, 임베딩 벡터를 차원을 축소하여, 각 분할된 임베딩 벡터에 대해 개별적인 어텐션 헤드를 구하여 어텐션을 구하는 경우, 이는 멀티-헤드(multi-head) 어텐션 알고리즘을 의미할 수 있다.

[0149] 일 실시예에서, 트랜스포머는 복수의 멀티-헤드 셀프 어텐션 알고리즘 또는 멀티-헤드 인코더-디코더 알고리즘을 수행하는 모듈들로 구성될 수 있다. 일 실시예에서, 트랜스포머는 임베딩 레이어, 정규화 레이어, 소프트맥스(softmax) 층 등 어텐션 알고리즘이 아닌 부가적인 구성요소들 또한 포함할 수 있다. 어텐션 알고리즘을 이용하여 트랜스포머를 구성하는 방법은 Vaswani et al., Attention Is All You Need, 2017 NIPS에 개시된 방법을 포함할 수 있으며, 이는 여기에 참조로서 통합된다.

[0150] 트랜스포머는 임베딩된 자연어, 임베딩된 시퀀스 정보, 분할된 이미지 데이터, 오디오 파형 등 다양한 데이터 도메인에 적용하여, 일련의 입력 데이터를 일련의 출력 데이터로 변환할 수 있다. 다양한 데이터 도메인을 가진 데이터들을 트랜스포머에 입력가능한 일련의 데이터들로 변환하기 위해, 트랜스포머는 데이터들을 임베딩할 수 있다. 트랜스포머는 일련의 입력 데이터 사이의 상대적 위치관계 또는 위상관계를 표현하는 추가적인 데이터를 처리할 수 있다. 또는 일련의 입력 데이터에 입력 데이터들 사이의 상대적인 위치관계 또는 위상관계를 표현하는 벡터들이 추가적으로 반영되어 일련의 입력 데이터가 임베딩될 수 있다. 일 예에서, 일련의 입력 데이터 사이의 상대적 위치관계는, 자연어 문장 내에서의 어순, 각각의 분할된 이미지의 상대적 위치 관계, 분할된 오디오 파형의 시간 순서 등을 포함할 수 있으나, 이에 제한되지 않는다. 일련의 입력 데이터들 사이의 상대적인 위

치관계 또는 위상관계를 표현하는 정보를 추가하는 과정은 위치 인코딩(positional encoding)으로 지칭될 수 있다.

[0151] 이미지 데이터를 임베딩하여 트랜스포머로 변환하는 방법의 일 예는 Dosovitskiy, et al., AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE에 개시되어 있으며, 해당 문서는 여기에 참조로서 통합된다.

[0152] 도 8은 본 개시의 실시예들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 간략하고 일반적인 개략도를 도시한다.

[0153] 본 개시가 일반적으로 컴퓨팅 장치에 의해 구현될 수 있는 것으로 전술되었지만, 당업자라면 본 개시가 하나 이상의 컴퓨터 상에서 실행될 수 있는 컴퓨터 실행가능 명령어 및/또는 기타 프로그램 모듈들과 결합되어 및/또는 하드웨어와 소프트웨어의 조합으로써 구현될 수 있다는 것을 잘 알 것이다.

[0154] 일반적으로, 프로그램 모듈은 특정의 태스크를 수행하거나 특정의 추상 데이터 유형을 구현하는 루틴, 프로그램, 컴포넌트, 데이터 구조, 기타 등등을 포함한다. 또한, 당업자라면 본 개시의 방법이 단일-프로세서 또는 멀티프로세서 컴퓨터 시스템, 미니컴퓨터, 메인프레임 컴퓨터는 물론 퍼스널 컴퓨터, 핸드헬드(handheld) 컴퓨팅 장치, 마이크로프로세서-기반 또는 프로그램가능 가전 제품, 기타 등등(이들 각각은 하나 이상의 연관된 장치와 연결되어 동작할 수 있음)을 비롯한 다른 컴퓨터 시스템 구성으로 실시될 수 있다는 것을 잘 알 것이다.

[0155] 본 개시의 설명된 실시예들은 또한 어떤 태스크들이 통신 네트워크를 통해 연결되어 있는 원격 처리 장치들에 의해 수행되는 분산 컴퓨팅 환경에서 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치 둘 다에 위치할 수 있다.

[0156] 컴퓨터는 통상적으로 다양한 컴퓨터 판독가능 매체를 포함한다. 컴퓨터에 의해 액세스 가능한 매체는 그 어떤 것이든지 컴퓨터 판독가능 매체가 될 수 있고, 이러한 컴퓨터 판독가능 매체는 휘발성 및 비휘발성 매체, 일시적(transitory) 및 비일시적(non-transitory) 매체, 이동식 및 비-이동식 매체를 포함한다. 제한이 아닌 예로서, 컴퓨터 판독가능 매체는 컴퓨터 판독가능 저장 매체 및 컴퓨터 판독가능 전송 매체를 포함할 수 있다. 컴퓨터 판독가능 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보를 저장하는 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성 매체, 일시적 및 비-일시적 매체, 이동식 및 비이동식 매체를 포함한다. 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital video disk) 또는 기타 광 디스크 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 컴퓨터에 의해 액세스될 수 있고 원하는 정보를 저장하는 데 사용될 수 있는 임의의 기타 매체를 포함하지만, 이에 한정되지 않는다.

[0157] 컴퓨터 판독가능 전송 매체는 통상적으로 반송파(carrier wave) 또는 기타 전송 메커니즘(transport mechanism)과 같은 피변조 데이터 신호(modulated data signal)에 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터 등을 구현하고 모든 정보 전달 매체를 포함한다. 피변조 데이터 신호라는 용어는 신호 내에 정보를 인코딩하도록 그 신호의 특성들 중 하나 이상을 설정 또는 변경시킨 신호를 의미한다. 제한이 아닌 예로서, 컴퓨터 판독가능 전송 매체는 유선 네트워크 또는 직접 배선 접속(direct-wired connection)과 같은 유선 매체, 그리고 음향, RF, 적외선, 기타 무선 매체와 같은 무선 매체를 포함한다. 상술된 매체들 중 임의의 것의 조합도 역시 컴퓨터 판독가능 전송 매체의 범위 안에 포함되는 것으로 한다.

[0158] 컴퓨터(1102)를 포함하는 본 개시의 여러가지 측면들을 구현하는 예시적인 환경이 나타내어져 있으며, 컴퓨터(1102)는 처리 장치(1104), 시스템 메모리(1106) 및 시스템 버스(1108)를 포함한다. 시스템 버스(1108)는 시스템 메모리(1106)(이에 한정되지 않음)를 비롯한 시스템 컴포넌트들을 처리 장치(1104)에 연결시킨다. 처리 장치(1104)는 다양한 상용 프로세서들 중 임의의 프로세서일 수 있다. 듀얼 프로세서 및 기타 멀티프로세서 아키텍처도 역시 처리 장치(1104)로서 이용될 수 있다.

[0159] 시스템 버스(1108)는 메모리 버스, 주변장치 버스, 및 다양한 상용 버스 아키텍처 중 임의의 것을 사용하는 로컬 버스에 추가적으로 상호 연결될 수 있는 몇 가지 유형의 버스 구조 중 임의의 것일 수 있다. 시스템 메모리(1106)는 판독 전용 메모리(ROM)(1110) 및 랜덤 액세스 메모리(RAM)(1112)를 포함한다. 기본 입/출력 시스템(BIOS)은 ROM, EPROM, EEPROM 등의 비휘발성 메모리(1110)에 저장되며, 이 BIOS는 시동 중과 같은 때에 컴퓨터(1102) 내의 구성요소들 간에 정보를 전송하는 일을 돕는 기본적인 루틴을 포함한다. RAM(1112)은 또한 데이터를 캐싱하기 위한 정적 RAM 등의 고속 RAM을 포함할 수 있다.

[0160] 컴퓨터(1102)는 또한 내장형 하드 디스크 드라이브(HDD)(1114)(예를 들어, EIDE, SATA) -이 내장형 하드 디스크

드라이브(1114)는 또한 적당한 섀시(도시 생략) 내에서 외장형 용도로 구성될 수 있음-, 자기 플로피 디스크 드라이브(FDD)(1116)(예를 들어, 이동식 디스켓(1118)으로부터 판독을 하거나 그에 기록을 하기 위한 것임), 및 광 디스크 드라이브(1120)(예를 들어, CD-ROM 디스크(1122)를 판독하거나 DVD 등의 기타 고용량 광 매체로부터 판독을 하거나 그에 기록을 하기 위한 것임)를 포함한다. 하드 디스크 드라이브(1114), 자기 디스크 드라이브(1116) 및 광 디스크 드라이브(1120)는 각각 하드 디스크 드라이브 인터페이스(1124), 자기 디스크 드라이브 인터페이스(1126) 및 광 드라이브 인터페이스(1128)에 의해 시스템 버스(1108)에 연결될 수 있다. 외장형 드라이브 구현을 위한 인터페이스(1124)는 USB(Universal Serial Bus) 및 IEEE 1394 인터페이스 기술 중 적어도 하나 또는 그 둘 다를 포함한다.

[0161] 이들 드라이브 및 그와 연관된 컴퓨터 판독가능 매체는 데이터, 데이터 구조, 컴퓨터 실행가능 명령어, 기타 등등의 비휘발성 저장을 제공한다. 컴퓨터(1102)의 경우, 드라이브 및 매체는 임의의 데이터를 적당한 디지털 형식으로 저장하는 것에 대응한다. 상기에서의 컴퓨터 판독가능 매체에 대한 설명이 HDD, 이동식 자기 디스크, 및 CD 또는 DVD 등의 이동식 광 매체를 언급하고 있지만, 당업자라면 집 드라이브(zip drive), 자기 카세트, 플래쉬 메모리 카드, 카트리지, 기타 등등의 컴퓨터에 의해 판독가능한 다른 유형의 매체도 역시 예시적인 운영 환경에서 사용될 수 있으며 또 임의의 이러한 매체가 본 개시의 방법들을 수행하기 위한 컴퓨터 실행가능 명령어를 포함할 수 있다는 것을 잘 알 것이다.

[0162] 운영 체제(1130), 하나 이상의 애플리케이션 프로그램(1132), 기타 프로그램 모듈(1134) 및 프로그램 데이터(1136)를 비롯한 다수의 프로그램 모듈이 드라이브 및 RAM(1112)에 저장될 수 있다. 운영 체제, 애플리케이션, 모듈 및/또는 데이터의 전부 또는 그 일부분이 또한 RAM(1112)에 캐싱될 수 있다. 본 개시가 여러가지 상업적으로 이용가능한 운영 체제 또는 운영 체제들의 조합에서 구현될 수 있다는 것을 잘 알 것이다.

[0163] 사용자는 하나 이상의 유선/무선 입력 장치, 예를 들어, 키보드(1138) 및 마우스(1140) 등의 포인팅 장치를 통해 컴퓨터(1102)에 명령 및 정보를 입력할 수 있다. 기타 입력 장치(도시 생략)로는 마이크, IR 리모콘, 조이스틱, 게임 패드, 스타일러스 펜, 터치 스크린, 기타 등등이 있을 수 있다. 이들 및 기타 입력 장치가 종종 시스템 버스(1108)에 연결되어 있는 입력 장치 인터페이스(1142)를 통해 처리 장치(1104)에 연결되지만, 병렬 포트, IEEE 1394 직렬 포트, 게임 포트, USB 포트, IR 인터페이스, 기타 등등의 기타 인터페이스에 의해 연결될 수 있다.

[0164] 모니터(1144) 또는 다른 유형의 디스플레이 장치도 역시 비디오 어댑터(1146) 등의 인터페이스를 통해 시스템 버스(1108)에 연결된다. 모니터(1144)에 부가하여, 컴퓨터는 일반적으로 스피커, 프린터, 기타 등등의 기타 주변 출력 장치(도시 생략)를 포함한다.

[0165] 컴퓨터(1102)는 유선 및/또는 무선 통신을 통한 원격 컴퓨터(들)(1148) 등의 하나 이상의 원격 컴퓨터로의 논리적 연결을 사용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(들)(1148)는 워크스테이션, 컴퓨팅 디바이스 컴퓨터, 라우터, 퍼스널 컴퓨터, 휴대용 컴퓨터, 마이크로프로세서-기반 오락 기기, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있으며, 일반적으로 컴퓨터(1102)에 대해 기술된 구성요소들 중 다수 또는 그 전부를 포함하지만, 간략함을 위해, 메모리 저장 장치(1150)만이 도시되어 있다. 도시되어 있는 논리적 연결은 근거리 통신망(LAN)(1152) 및/또는 더 큰 네트워크, 예를 들어, 원거리 통신망(WAN)(1154)에의 유선/무선 연결을 포함한다. 이러한 LAN 및 WAN 네트워킹 환경은 사무실 및 회사에서 일반적인 것이며, 인트라넷 등의 전사적 컴퓨터 네트워크(enterprise-wide computer network)를 용이하게 해주며, 이들 모두는 전세계 컴퓨터 네트워크, 예를 들어, 인터넷에 연결될 수 있다.

[0166] LAN 네트워킹 환경에서 사용될 때, 컴퓨터(1102)는 유선 및/또는 무선 통신 네트워크 인터페이스 또는 어댑터(1156)를 통해 로컬 네트워크(1152)에 연결된다. 어댑터(1156)는 LAN(1152)에의 유선 또는 무선 통신을 용이하게 해줄 수 있으며, 이 LAN(1152)은 또한 무선 어댑터(1156)와 통신하기 위해 그에 설치되어 있는 무선 액세스 포인트를 포함하고 있다. WAN 네트워킹 환경에서 사용될 때, 컴퓨터(1102)는 모뎀(1158)을 포함할 수 있거나, WAN(1154) 상의 통신 컴퓨팅 디바이스에 연결되거나, 또는 인터넷을 통하는 등, WAN(1154)을 통해 통신을 설정하는 기타 수단을 갖는다. 내장형 또는 외장형 및 유선 또는 무선 장치일 수 있는 모뎀(1158)은 직렬 포트 인터페이스(1142)를 통해 시스템 버스(1108)에 연결된다. 네트워크화된 환경에서, 컴퓨터(1102)에 대해 설명된 프로그램 모듈들 또는 그의 일부분이 원격 메모리/저장 장치(1150)에 저장될 수 있다. 도시된 네트워크 연결이 예시적인 것이며 컴퓨터들 사이에 통신 링크를 설정하는 기타 수단이 사용될 수 있다는 것을 잘 알 것이다.

[0167] 컴퓨터(1102)는 무선 통신으로 배치되어 동작하는 임의의 무선 장치 또는 개체, 예를 들어, 프린터, 스캐너, 데스크톱 및/또는 휴대용 컴퓨터, PDA(portable data assistant), 통신 위성, 무선 검출가능 태그와 연관된 임의

의 장비 또는 장소, 및 전화와 통신을 하는 동작을 한다. 이것은 적어도 Wi-Fi 및 블루투스 무선 기술을 포함한다. 따라서, 통신은 종래의 네트워크에서와 같이 미리 정의된 구조이거나 단순하게 적어도 2개의 장치 사이의 애드혹 통신(ad hoc communication)일 수 있다.

[0168] Wi-Fi(Wireless Fidelity)는 유선 없이도 인터넷 등으로의 연결을 가능하게 해준다. Wi-Fi는 이러한 장치, 예를 들어, 컴퓨터가 실내에서 및 실외에서, 즉 기지국의 통화권 내의 아무 곳에서나 데이터를 전송 및 수신할 수 있게 해주는 셀 전화와 같은 무선 기술이다. Wi-Fi 네트워크는 안전하고 신뢰성 있으며 고속인 무선 연결을 제공하기 위해 IEEE 802.11(a, b, g, 기타)이라고 하는 무선 기술을 사용한다. 컴퓨터를 서로에, 인터넷에 및 유선 네트워크(IEEE 802.3 또는 이더넷을 사용함)에 연결시키기 위해 Wi-Fi가 사용될 수 있다. Wi-Fi 네트워크는 비인가 2.4 및 5GHz 무선 대역에서, 예를 들어, 11Mbps(802.11a) 또는 54 Mbps(802.11b) 데이터 레이트로 동작하거나, 양 대역(듀얼 대역)을 포함하는 제품에서 동작할 수 있다.

[0169] 본 개시의 기술 분야에서 통상의 지식을 가진 자는 정보 및 신호들이 임의의 다양한 상이한 기술들 및 기법들을 이용하여 표현될 수 있다는 것을 이해할 것이다. 예를 들어, 위의 설명에서 참조될 수 있는 데이터, 지시들, 명령들, 정보, 신호들, 비트들, 심볼들 및 칩들은 전압들, 전류들, 전자기파들, 자기장들 또는 입자들, 광학장들 또는 입자들, 또는 이들의 임의의 결합에 의해 표현될 수 있다.

[0170] 본 개시의 기술 분야에서 통상의 지식을 가진 자는 여기에 개시된 실시예들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 프로세서들, 수단들, 회로들 및 알고리즘 단계들이 전자 하드웨어, (편의를 위해, 여기에서 소프트웨어로 지칭되는) 다양한 형태들의 프로그램 또는 설계 코드 또는 이들 모두의 결합에 의해 구현될 수 있다는 것을 이해할 것이다. 하드웨어 및 소프트웨어의 이러한 상호 호환성을 명확하게 설명하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들 및 단계들이 이들의 기능과 관련하여 위에서 일반적으로 설명되었다. 이러한 기능이 하드웨어 또는 소프트웨어로서 구현되는지 여부는 특정한 애플리케이션 및 전체 시스템에 대하여 부과되는 설계 제약들에 따라 좌우된다. 본 개시의 기술 분야에서 통상의 지식을 가진 자는 각각의 특정한 애플리케이션에 대하여 다양한 방식들로 설명된 기능을 구현할 수 있으나, 이러한 구현 결정들은 본 개시의 범위를 벗어나는 것으로 해석되어서는 안 될 것이다.

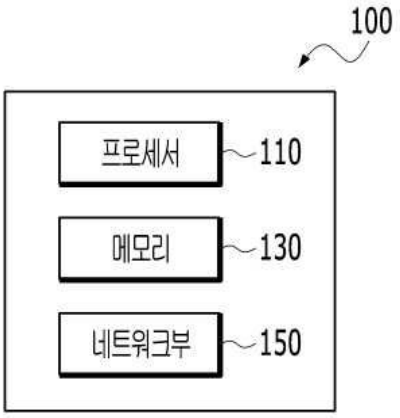
[0171] 여기서 제시된 다양한 실시예들은 방법, 장치, 또는 표준 프로그래밍 및/또는 엔지니어링 기술을 사용한 제조 물품(article)으로 구현될 수 있다. 용어 제조 물품은 임의의 컴퓨터-판독가능 저장장치로부터 액세스 가능한 컴퓨터 프로그램, 캐리어, 또는 매체(media)를 포함한다. 예를 들어, 컴퓨터-판독가능 저장매체는 자기 저장 장치(예를 들면, 하드 디스크, 플로피 디스크, 자기 스트립, 등), 광학 디스크(예를 들면, CD, DVD, 등), 스마트 카드, 및 플래쉬 메모리 장치(예를 들면, EEPROM, 카드, 스틱, 키 드라이브, 등)를 포함하지만, 이들로 제한되는 것은 아니다. 또한, 여기서 제시되는 다양한 저장 매체는 정보를 저장하기 위한 하나 이상의 장치 및/또는 다른 기계-판독가능한 매체를 포함한다.

[0172] 제시된 프로세스들에 있는 단계들의 특정한 순서 또는 계층 구조는 예시적인 접근들의 일례임을 이해하도록 한다. 설계 우선순위들에 기반하여, 본 개시의 범위 내에서 프로세스들에 있는 단계들의 특정한 순서 또는 계층 구조가 재배열될 수 있다는 것을 이해하도록 한다. 첨부된 방법 청구항들은 샘플 순서로 다양한 단계들의 엘리먼트들을 제공하지만 제시된 특정한 순서 또는 계층 구조에 한정되는 것을 의미하지는 않는다.

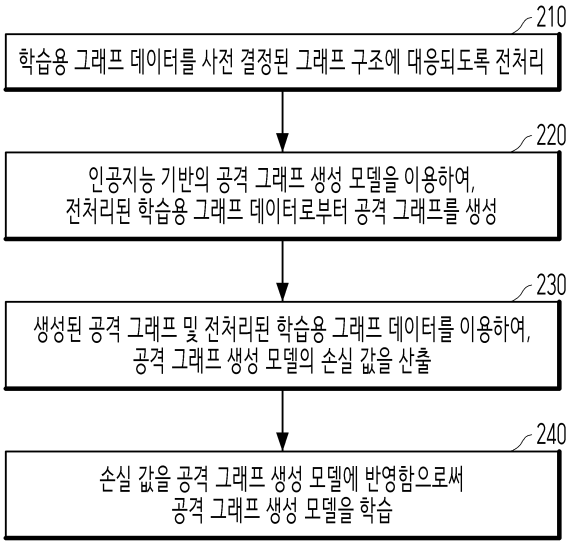
[0173] 제시된 실시예들에 대한 설명은 임의의 본 개시의 기술 분야에서 통상의 지식을 가진 자가 본 개시를 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 개시의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이며, 여기에 정의된 일반적인 원리들은 본 개시의 범위를 벗어남이 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 개시는 여기에 제시된 실시예들로 한정되는 것이 아니라, 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의의 범위에서 해석되어야 할 것이다.

도면

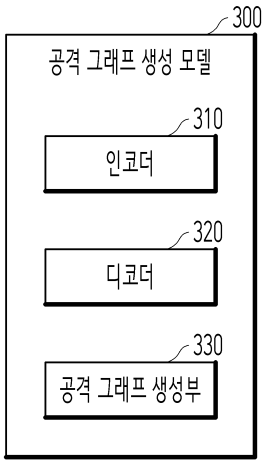
도면1


100
프로세서
110
메모리
130
네트워크부
150

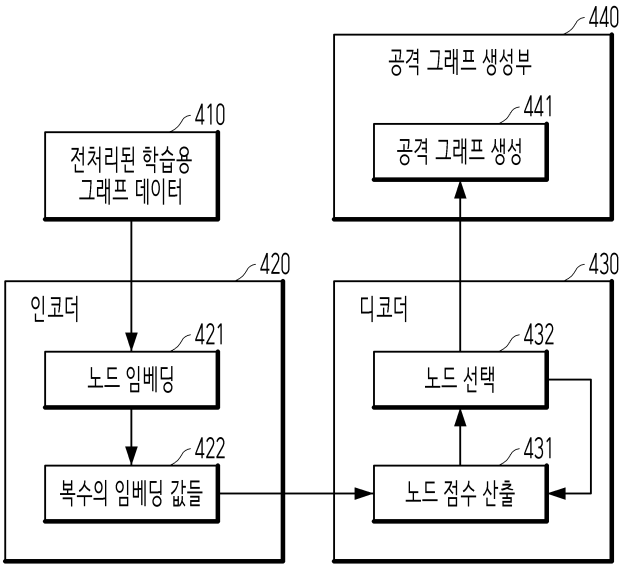
도면2


210
학습용 그래프 데이터를 사전 결정된 그래프 구조에 대응되도록 전처리
220
인공지능 기반의 공격 그래프 생성 모델을 이용하여,
전처리된 학습용 그래프 데이터로부터 공격 그래프를 생성
230
생성된 공격 그래프 및 전처리된 학습용 그래프 데이터를 이용하여,
공격 그래프 생성 모델의 손실 값을 산출
240
손실 값을 공격 그래프 생성 모델에 반영함으로써
공격 그래프 생성 모델을 학습

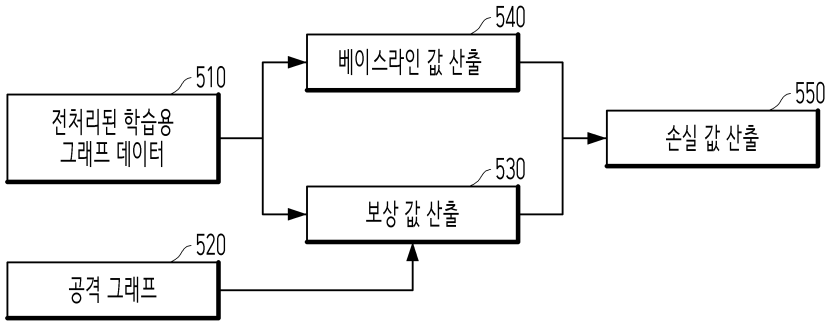
도면3


300
공격 그래프 생성 모델
310
인코더
320
디코더
330
공격 그래프 생성부

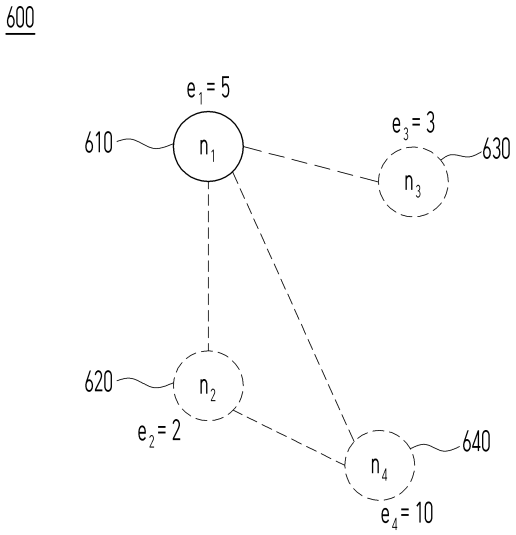
도면4


440
공격 그래프 생성부
441
공격 그래프 생성
410
전처리된 학습용
그래프 데이터
420
인코더
421
노드 임베딩
422
복수의 임베딩 값들
430
디코더
432
노드 선택
431
노드 점수 산출

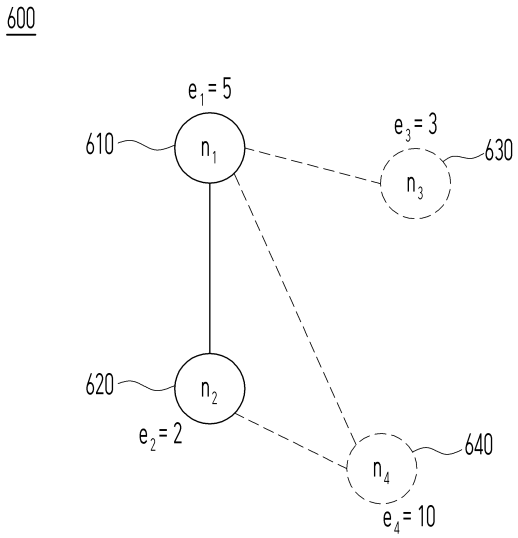
도면5


540
베이스라인 값 산출
510
전처리된 학습용
그래프 데이터
550
손실 값 산출
530
보상 값 산출
520
공격 그래프

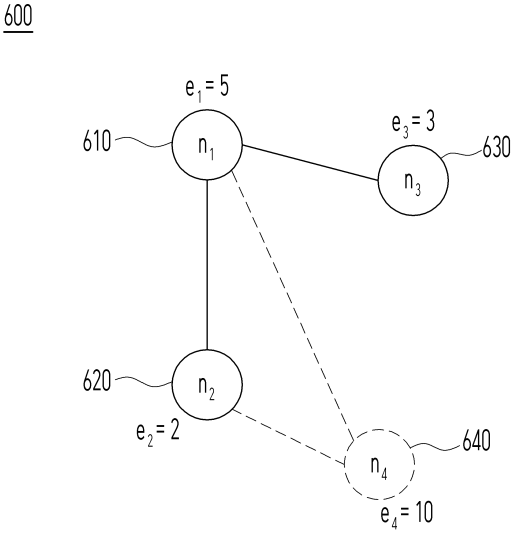
도면6a


600
e1 = 5
610
n1
e3 = 3
630
n3
620
n2
e2 = 2
640
n4
e4 = 10

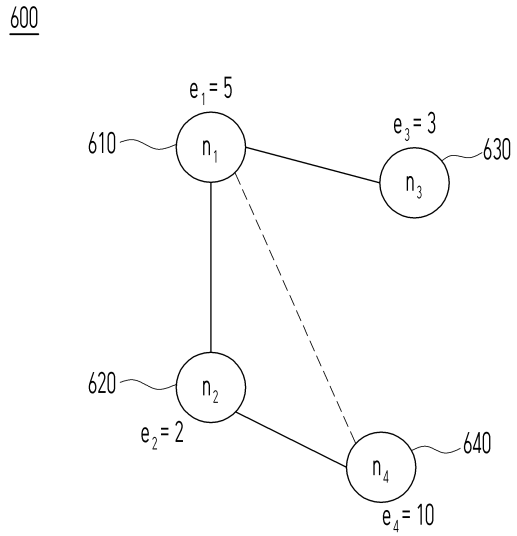
도면6b


600
e1 = 5
610
n1
e3 = 3
630
n3
620
n2
e2 = 2
640
n4
e4 = 10

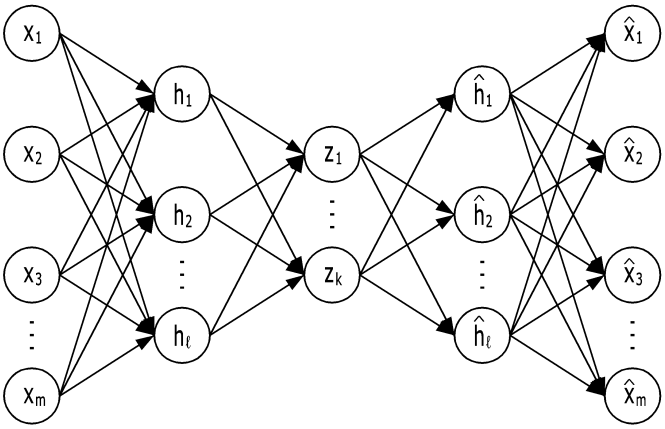
도면6c


600
e1 = 5
610
n1
e3 = 3
630
n3
620
n2
e2 = 2
640
n4
e4 = 10

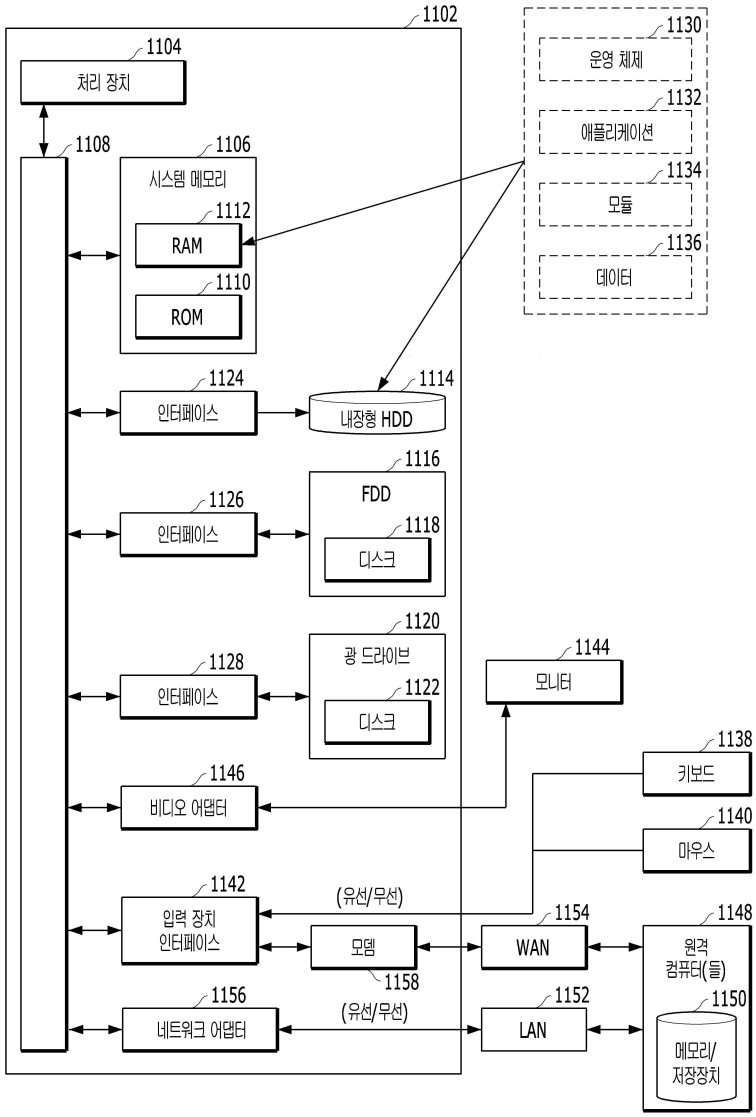
도면6d


600
e1 = 5
610
n1
e3 = 3
630
n3
620
n2
e2 = 2
640
n4
e4 = 10

도면7


x1
x2
x3
xm
h1
h2
hℓ
z1
zk
ĥ1
ĥ2
ĥℓ
x̂1
x̂2
x̂3
x̂m

도면8


1102
1104
처리 장치
1108
1106
시스템 메모리
1112
RAM
1110
ROM
1130
운영 체제
1132
애플리케이션
1134
모듈
1136
데이터
1124
인터페이스
1114
내장형 HDD
1116
FDD
1126
인터페이스
1118
디스크
1120
광 드라이브
1128
인터페이스
1122
디스크
1144
모니터
1138
키보드
1146
비디오 어댑터
1140
마우스
1142
입력 장치
인터페이스
(유선/무선)
모뎀
1158
1154
WAN
1148
원격
컴퓨터(들)
1156
네트워크 어댑터
(유선/무선)
1152
LAN
1150
메모리/
저장장치