



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0087232
(43) 공개일자 2023년06월16일

(51) 국제특허분류(Int. Cl.)
G06N 20/00 (2019.01) H04L 29/06 (2006.01)
(52) CPC특허분류
G06N 20/00 (2021.08)
G06F 21/55 (2013.01)
(21) 출원번호 10-2021-0175995
(22) 출원일자 2021년12월09일
심사청구일자 2021년12월09일

(71) 출원인
국방과학연구소
대전광역시 유성구 북유성대로488번길 160 (수남동)
세종대학교산학협력단
서울특별시 광진구 능동로 209 (군자동, 세종대학교)
(72) 발명자
김진국
대전광역시 유성구 북유성대로488번길 160 (수남동)
박정찬
대전광역시 유성구 북유성대로488번길 160 (수남동)
(뒷면에 계속)
(74) 대리인
제일특허법인(유)

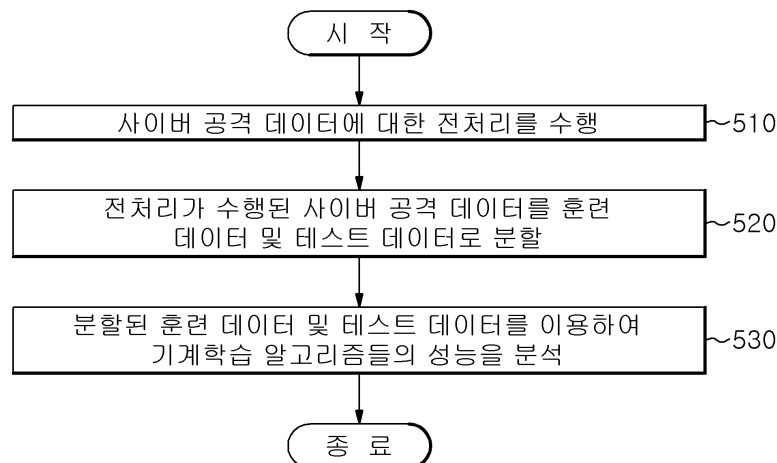
전체 청구항 수 : 총 16 항

(54) 발명의 명칭 LID-DS 데이터 세트를 이용한 기계학습 알고리즘 성능 분석 방법 및 시스템

(57) 요약

본 발명에 따른 기계학습 알고리즘 성능 분석 방법은, 알고리즘 비교 시스템에 의해 수행되는 기계학습 알고리즘 성능 분석 방법에 있어서, 사이버 공격 데이터에 대한 전처리를 수행하는 단계; 전처리가 수행된 상기 사이버 공격 데이터를 훈련 데이터 및 테스트 데이터로 분할하는 단계; 및 분할된 상기 훈련 데이터 및 상기 테스트 데이터를 이용하여 기계학습 알고리즘들의 성능을 분석하는 단계를 포함할 수 있다.

대표도 - 도5



(52) CPC특허분류

H04L 63/1416 (2013.01)

(72) 발명자

신동일

서울특별시 광진구 능동로 209, 대양 AI센터 825
호(군자동)

신동규

서울특별시 광진구 능동로 209, 대양 AI센터 826
호(군자동)

박대경

서울특별시 광진구 능동로 209, 대양 AI센터 834
호(군자동)

명세서

청구범위

청구항 1

알고리즘 비교 시스템에 의해 수행되는 기계학습 알고리즘 성능 분석 방법에 있어서,
사이버 공격 데이터에 대한 전처리를 수행하는 단계;
전처리가 수행된 상기 사이버 공격 데이터를 훈련 데이터 및 테스트 데이터로 분할하는 단계; 및
분할된 상기 훈련 데이터 및 상기 테스트 데이터를 이용하여 기계학습 알고리즘들의 성능을 분석하는 단계
를 포함하는 기계학습 알고리즘 성능 분석 방법.

청구항 2

제 1 항에 있어서,
상기 사이버 공격 데이터는,
LID-DS(Leipzig Intrusion Detection-Data Set)이고,
상기 LID-DS 데이터는,
복수 개의 사이버 공격 방법에 따라 분류되고, 각 사이버 공격 방법에 따라 복수 개의 특징으로 구성된 것을 포함하며,
상기 전처리를 수행하는 단계는,
상기 LID-DS 데이터에 대해 아규먼트 특징(Argument Feature)과 결측값을 삭제하고, event_time Feature에 대해 콜론을 제거하며, event_direction과 event_type에 대해 LabelEncoder를 사용하는 단계
를 포함하는 기계학습 알고리즘 성능 분석 방법.

청구항 3

제 2 항에 있어서,
상기 전처리를 수행하는 단계는,
각 사이버 공격에 사용된 프로세스들을 하나의 프로세스로 통합하고, 상기 통합된 프로세스로 구성된 라벨들을 라벨인코더(LabelEncoder)를 사용하여 각 프로세스에 라벨을 부착하는 단계
를 포함하는 기계학습 알고리즘 성능 분석 방법.

청구항 4

제 1 항에 있어서,
상기 전처리를 수행하는 단계는,
최소 최대 정규화(Min-Max Normalization)를 사용하여 상기 LID-DS 데이터가 가진 특징에 대해 각각의 최소값을 0, 최대값을 1, 나머지 값들을 0과 1사이의 값으로 변환하는 단계
를 포함하는 기계학습 알고리즘 성능 분석 방법.

청구항 5

제 1 항에 있어서,
상기 분할하는 단계는,

기계학습 모델에 사용할 상기 훈련 데이터와 상기 테스트 데이터를 기 설정된 비율로 분할하는 단계를 포함하는 기계학습 알고리즘 성능 분석 방법.

청구항 6

제 1 항에 있어서,

상기 분석하는 단계는,

분할된 상기 훈련 데이터를 각각의 기계학습 모델에 입력받고, 분할된 상기 훈련 데이터를 이용하여 상기 각각의 기계학습 모델을 학습시키고, 상기 학습된 각각의 기계학습 모델에 분할된 상기 테스트 데이터를 입력받고, 상기 학습된 각각의 기계학습 모델을 이용하여 분할된 상기 테스트 데이터에 대한 침입 탐지 정확성을 도출하는 단계

를 포함하는 기계학습 알고리즘 성능 분석 방법.

청구항 7

제 6 항에 있어서,

상기 분석하는 단계는,

정밀도(Precision), 재현율(Recall), F1 스코어(F1 Score)와 오류율을 사용하여 상기 학습된 각각의 기계학습 모델의 성능을 평가하는 단계

를 포함하는 기계학습 알고리즘 성능 분석 방법.

청구항 8

컴퓨터 프로그램을 저장하고 있는 컴퓨터 판독 가능 기록매체로서,

상기 컴퓨터 프로그램은, 프로세서에 의해 실행되면,

사이버 공격 데이터에 대한 전처리를 수행하는 단계;

전처리가 수행된 상기 사이버 공격 데이터를 훈련 데이터 및 테스트 데이터로 분할하는 단계; 및

분할된 상기 훈련 데이터 및 상기 테스트 데이터를 이용하여 기계학습 알고리즘들의 성능을 분석하는 단계를

를 포함하는 동작을 상기 프로세서가 수행하도록 하기 위한 명령어를 포함하는 컴퓨터 판독 가능한 기록매체.

청구항 9

컴퓨터 판독 가능한 기록매체에 저장되어 있는 컴퓨터 프로그램으로서,

상기 컴퓨터 프로그램은, 프로세서에 의해 실행되면,

사이버 공격 데이터에 대한 전처리를 수행하는 단계;

전처리가 수행된 상기 사이버 공격 데이터를 훈련 데이터 및 테스트 데이터로 분할하는 단계; 및

분할된 상기 훈련 데이터 및 상기 테스트 데이터를 이용하여 기계학습 알고리즘들의 성능을 분석하는 단계를

포함하는 동작을 상기 프로세서가 수행하도록 하기 위한 명령어를 포함하는 컴퓨터 프로그램.

청구항 10

기계학습 알고리즘 성능 분석 시스템에 있어서,

사이버 공격 데이터에 대한 전처리를 수행하는 전처리부;

전처리가 수행된 상기 사이버 공격 데이터를 훈련 데이터 및 테스트 데이터로 분할하는 데이터 분할부; 및

분할된 상기 훈련 데이터 및 상기 테스트 데이터를 이용하여 기계학습 알고리즘들의 성능을 분석하는 성능 분석부

를 포함하는 알고리즘 성능 분석 시스템.

청구항 11

제 10 항에 있어서,

상기 사이버 공격 데이터는,

LID-DS(Leipzig Intrusion Detection-Data Set)이고,

상기 LID-DS 데이터는,

복수 개의 사이버 공격 방법에 따라 분류되고, 각 사이버 공격 방법에 따라 복수 개의 특징으로 구성된 것을 포함하며,

상기 전처리부는,

상기 LID-DS 데이터에 대해 아규먼트 특징(Argument Feature)과 결측값을 삭제하고, event_time Feature에 대해 콜론을 제거하며, event_direction과 event_type에 대해 LabelEncoder를 사용하는

것을 특징으로 하는 알고리즘 성능 분석 시스템.

청구항 12

제 11 항에 있어서,

상기 전처리부는,

각 사이버 공격에 사용된 프로세스들을 하나의 프로세스로 통합하고, 상기 통합된 프로세스로 구성된 라벨들을 라벨인코더(LabelEncoder)를 사용하여 각 프로세스에 라벨을 부착하는

것을 특징으로 하는 알고리즘 성능 분석 시스템.

청구항 13

제 10 항에 있어서,

상기 전처리부는,

최소 최대 정규화(Min-Max Normalization)를 사용하여 상기 LID-DS 데이터가 가진 특징에 대해 각각의 최소값을 0, 최대값을 1, 나머지 값들을 0과 1사이의 값으로 변환하는

것을 특징으로 하는 알고리즘 성능 분석 시스템.

청구항 14

제 10 항에 있어서,

상기 데이터 분할부는,

기계학습 모델에 사용할 상기 훈련 데이터와 상기 테스트 데이터를 기 설정된 비율로 분할하는

것을 특징으로 하는 알고리즘 성능 분석 시스템.

청구항 15

제 10 항에 있어서,

상기 성능 분석부는,

분할된 상기 훈련 데이터를 각각의 기계학습 모델에 입력받고, 분할된 상기 훈련 데이터를 이용하여 상기 각각의 기계학습 모델을 학습시키고, 상기 학습된 각각의 기계학습 모델에 분할된 상기 테스트 데이터를 입력받고, 상기 학습된 각각의 기계학습 모델을 이용하여 분할된 상기 테스트 데이터에 대한 침입 탐지 정확성을 도출하는

것을 특징으로 하는 알고리즘 성능 분석 시스템.

청구항 16

제 15 항에 있어서,

상기 성능 분석부는,

정밀도(Precision), 재현율(Recall), F1 스코어(F1 Score)와 오류율을 사용하여 상기 학습된 각각의 기계학습 모델의 성능을 평가하는

것을 특징으로 하는 알고리즘 성능 분석 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 사이버 공격의 탐지 정확성을 분석하는 기술에 관한 것으로, 더욱 상세하게는 LID-DS 데이터 세트를 이용하여 기계학습 알고리즘의 성능을 분석하는 방법 및 시스템에 관한 것이다.

배경 기술

[0002] 오늘날 정보통신 기술이 급격하게 발달하면서 IT 인프라에서 보안의 중요성이 점진적으로 높아지고 있고, 이와 동시에 사이버 공격은 지능형 지속 공격처럼 고도화되고 지능적으로 다양해지고 있다.

[0003] 점점 더 고도화되는 사이버 공격을 방어하는 것은 매우 중요한 사안인데, IDS(Intrusion Detection System) 기술의 발달 속도가 빠르게 변형되는 사이버 공격을 완벽하게 탐지하지 못하고 있는 실정이다. 따라서 현재는 HIDS 데이터 분석을 통해서 위와 같은 사이버 공격을 방어하는데 침입 탐지 시스템에서 생성된 데이터를 이용하고 있다.

[0004] 침입 탐지 시스템은 네트워크 기반인 NIDS, 호스트 기반인 HIDS 두 가지 방식으로 나눌 수 있다. 네트워크 기반 침입 탐지 시스템과 달리 호스트 기반 침입 탐지 시스템은 시스템 내부와 외부로 전체적으로 모니터링을 해야 하는 어려움이 있다.

[0005] 그에 따라, 연구가 많이 부족하고 침입 탐지 시스템은 새로운 사이버 공격 및 내부 사이버 공격에 대한 방어 대책이 미흡하며 오경보가 증가하는 문제점이 있다.

[0006] 호스트 기반 침입 탐지 시스템 방식은 오용 탐지와 이상 탐지 2가지 방법으로 나눌 수 있다. 오용 탐지 방법은 시그니처 기반으로 사이버 공격을 탐지하기 때문에 기존의 사이버 공격을 탐지하는 것은 효과적이지만 반면에 새로운 사이버 공격에 대한 탐지는 부적합하다.

[0007] 이상 탐지 방법은 오용 탐지 방법과 반대로 정상적인 동작 및 행위로 정의된 상태가 아닌 것에 대한 모든 상황을 이상 행위로 판단하여 탐지하게 된다. 즉, 오용탐지 방법과 달리 이상 탐지 방법은 제로 데이터 공격에 대한 탐지에는 적합하지만, 정상 동작 및 이상 행위를 판단할 수 있는 많은 데이터가 요구되거나 데이터가 너무 부족하여 기계학습에 적용하기에는 어려움이 있다.

[0008] 시그니처 기반 탐지 기술에서 IDS는 공격 시그니처가 포함된 데이터베이스를 사용하여 데이터의 침입을 탐지하므로 탐지율이 상대적으로 우수하다. 하지만 탐지 체계의 문제점은 새로운 사이버 공격에 대한 데이터가 없어서 새로운 사이버 공격을 탐지할 수 없다는 문제가 있다.

[0009] 최근 들어, 호스트 기반 침입 탐지 연구는 기계학습 기술을 사용하여 변칙에 기반한 탐지 기술은 비정상적인 행동을 찾아 새로운 공격을 탐지하고 이상을 탐지하기 위해 기계학습 알고리즘이 사용되고 있다.

선행기술문헌

특허문헌

[0010] (특허문헌 0001) 한국등록특허 제10-2292968호(공고일 : 2021. 08. 25.)

(특허문헌 0002) 한국공개특허 제10-2018-0085157호(공개일 : 2018. 07. 26.)

발명의 내용

해결하려는 과제

[0011] 본 발명은 LID-DS 데이터 세트를 사용하여 기계학습 알고리즘의 성능을 분석하는 방법 및 시스템을 제공하고자 한다.

과제의 해결 수단

[0012] 본 발명은, 일 관점에 따라, 알고리즘 비교 시스템에 의해 수행되는 기계학습 알고리즘 성능 분석 방법에 있어서, 사이버 공격 데이터에 대한 전처리를 수행하는 단계; 전처리가 수행된 상기 사이버 공격 데이터를 훈련 데이터 및 테스트 데이터로 분할하는 단계; 및 분할된 상기 훈련 데이터 및 상기 테스트 데이터를 이용하여 기계학습 알고리즘들의 성능을 분석하는 단계를 포함하는 기계학습 알고리즘 성능 분석 방법을 제공할 수 있다.

[0013] 본 발명의 상기 사이버 공격 데이터는, LID-DS(Leipzig Intrusion Detection-Data Set)이고, 상기 LID-DS 데이터는, 복수 개의 사이버 공격 방법에 따라 분류되고, 각 사이버 공격 방법에 따라 복수 개의 특징으로 구성된 것을 포함하며, 상기 전처리를 수행하는 단계는, 상기 LID-DS 데이터에 대해 아규먼트 특징(Argument Feature)과 결측값을 삭제하고, event_time Feature에 대해 콜론을 제거하며, event_direction과 event_type에 대해 LabelEncoder를 사용하는 단계를 포함할 수 있다.

[0014] 본 발명의 상기 전처리를 수행하는 단계는, 각 사이버 공격에 사용된 프로세스들을 하나의 프로세스로 통합하고, 상기 통합된 프로세스로 구성된 라벨들을 라벨인코더(LabelEncoder)를 사용하여 각 프로세스에 라벨을 부착하는 단계를 포함할 수 있다.

[0015] 본 발명의 상기 전처리를 수행하는 단계는, 최소 최대 정규화(Min-Max Normalization)를 사용하여 상기 LID-DS 데이터가 가진 특징에 대해 각각의 최소값을 0, 최대값을 1, 나머지 값들을 0과 1사이의 값으로 변환하는 단계를 포함할 수 있다.

[0016] 본 발명의 상기 분할하는 단계는, 기계학습 모델에 사용할 상기 훈련 데이터와 상기 테스트 데이터를 기 설정된 비율로 분할하는 단계를 포함할 수 있다.

[0017] 본 발명의 상기 분석하는 단계는, 분할된 상기 훈련 데이터를 각각의 기계학습 모델에 입력받고, 분할된 상기 훈련 데이터를 이용하여 상기 각각의 기계학습 모델을 학습시키고, 상기 학습된 각각의 기계학습 모델에 분할된 상기 테스트 데이터를 입력받고, 상기 학습된 각각의 기계학습 모델을 이용하여 분할된 상기 테스트 데이터에 대한 침입 탐지 정확성을 도출하는 단계를 포함할 수 있다.

[0018] 본 발명의 상기 분석하는 단계는, 정밀도(Precision), 재현율(Recall), F1 스코어(F1 Score)와 오류율을 사용하여 상기 학습된 각각의 기계학습 모델의 성능을 평가하는 단계를 포함할 수 있다.

[0019] 본 발명은, 다른 관점에 따라, 컴퓨터 프로그램을 저장하고 있는 컴퓨터 판독 가능 기록매체로서, 상기 컴퓨터 프로그램은, 프로세서에 의해 실행되면, 사이버 공격 데이터에 대한 전처리를 수행하는 단계; 전처리가 수행된 상기 사이버 공격 데이터를 훈련 데이터 및 테스트 데이터로 분할하는 단계; 및 분할된 상기 훈련 데이터 및 상기 테스트 데이터를 이용하여 기계학습 알고리즘들의 성능을 분석하는 단계를 포함하는 컴퓨터 판독 가능한 기록매체를 제공할 수 있다.

[0020] 본 발명은, 또 다른 관점에 따라, 컴퓨터 판독 가능한 기록매체에 저장되어 있는 컴퓨터 프로그램으로서, 상기 컴퓨터 프로그램은, 프로세서에 의해 실행되면, 사이버 공격 데이터에 대한 전처리를 수행하는 단계; 전처리가 수행된 상기 사이버 공격 데이터를 훈련 데이터 및 테스트 데이터로 분할하는 단계; 및 분할된 상기 훈련 데이터 및 상기 테스트 데이터를 이용하여 기계학습 알고리즘들의 성능을 분석하는 단계를 포함하는 동작을 상기 프로세서가 수행하도록 하기 위한 명령어를 포함하는 컴퓨터 프로그램을 제공할 수 있다.

[0021] 본 발명은, 또 다른 관점에 따라, 기계학습 알고리즘 성능 분석 시스템에 있어서, 사이버 공격 데이터에 대한 전처리를 수행하는 전처리부; 전처리가 수행된 상기 사이버 공격 데이터를 훈련 데이터 및 테스트 데이터로 분할하는 데이터 분할부; 및 분할된 상기 훈련 데이터 및 상기 테스트 데이터를 이용하여 기계학습 알고리즘들의 성능을 분석하는 성능 분석부를 포함하는 알고리즘 성능 분석 시스템을 제공할 수 있다.

[0022] 본 발명의 상기 사이버 공격 데이터는, LID-DS(Leipzig Intrusion Detection-Data Set)이고, 상기 LID-DS 데이터는, 복수 개의 사이버 공격 방법에 따라 분류되고, 각 사이버 공격 방법에 따라 복수 개의 특징으로 구성된

것을 포함하며, 상기 전처리부는, 상기 LID-DS 데이터에 대해 아규먼트 특징(Argument Feature)과 결측값을 삭제하고, event_time Feature에 대해 컬론을 제거하며, event_direction과 event_type에 대해 LabelEncoder를 사용할 수 있다.

[0023] 본 발명의 상기 전처리부는, 각 사이버 공격에 사용된 프로세스들을 하나의 프로세스로 통합하고, 상기 통합된 프로세스로 구성된 라벨들을 라벨인코더(LabelEncoder)를 사용하여 각 프로세스에 라벨을 부착할 수 있다.

[0024] 본 발명의 상기 전처리부는, 최소 최대 정규화(Min-Max Normalization)를 사용하여 상기 LID-DS 데이터가 가진 특징에 대해 각각의 최소값을 0, 최대값을 1, 나머지 값들을 0과 1사이의 값으로 변환할 수 있다.

[0025] 본 발명의 상기 데이터 분할부는, 기계학습 모델에 사용할 상기 훈련 데이터와 상기 테스트 데이터를 기 설정된 비율로 분할할 수 있다.

[0026] 본 발명의 상기 성능 분석부는, 분할된 상기 훈련 데이터를 각각의 기계학습 모델에 입력받고, 분할된 상기 훈련 데이터를 이용하여 상기 각각의 기계학습 모델을 학습시키고, 상기 학습된 각각의 기계학습 모델에 분할된 상기 테스트 데이터를 입력받고, 상기 학습된 각각의 기계학습 모델을 이용하여 분할된 상기 테스트 데이터에 대한 침입 탐지 정확성을 도출할 수 있다.

[0027] 본 발명의 상기 성능 분석부는, 정밀도(Precision), 재현율(Recall), F1 스코어(F1 Score)와 오류율을 사용하여 상기 학습된 각각의 기계학습 모델의 성능을 평가할 수 있다.

발명의 효과

[0028] 본 발명의 실시예에 따르면, LID-DS 데이터 세트에서 공격 유형에 따른 기계학습 알고리즘의 성능을 효과적으로 평가할 수 있다.

도면의 간단한 설명

[0029] 도 1은 본 발명의 실시예에 있어서, 데이터 세트를 설명하기 위한 표이다.

도 2는 본 발명의 실시예에 있어서, LID-DS 데이터 세트의 공격 시뮬레이션 절차를 나타낸 도면이다.

도 3은 본 발명의 실시예에 있어서, LID-DS 데이터 세트 성능 비교 모델의 구조를 설명하기 위한 도면이다.

도 4는 본 발명의 실시예에 따른 알고리즘 비교 시스템의 구성을 설명하기 위한 블록도이다.

도 5는 본 발명의 실시예에 따른 알고리즘 비교 시스템에서 기계학습 알고리즘을 비교하는 방법을 설명하기 위한 흐름도이다.

도 6은 본 발명의 실시예에 있어서, LID-DS에 저장된 데이터 형식을 설명하기 위한 도면이다.

도 7은 본 발명의 실시예에 있어서, 데이터 타입과 공격 타입을 설명하기 위한 표이다.

발명을 실시하기 위한 구체적인 내용

[0030] 먼저, 본 발명의 장점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되는 실시 예들을 참조하면 명확해질 것이다. 여기에서, 본 발명은 이하에서 개시되는 실시 예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시 예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 발명의 범주를 명확하게 이해할 수 있도록 하기 위해 예시적으로 제공되는 것이므로, 본 발명의 기술적 범위는 청구항들에 의해 정의되어야 할 것이다.

[0031] 아울러, 아래의 본 발명을 설명함에 있어서 공지 기능 또는 구성 등에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다. 그리고, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들인 것으로, 이는 사용자, 운용자 등의 의도 또는 관례 등에 따라 달라질 수 있음은 물론이다. 그러므로, 그 정의는 본 명세서의 전반에 걸쳐 기술되는 기술사상을 토대로 이루어져야 할 것이다.

[0032] 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시 예에 대하여 상세하게 설명한다.

[0033] 도 1은 본 발명의 실시예에 있어서, 데이터 세트를 설명하기 위한 표이다.

[0034] 2018년 Leipzig University에서 호스트 기반 침입 탐지 시스템의 이상 탐지 연구를 위한 LID-DS 데이터 세트가

공개되었는데, LID-DS 데이터는 연속적인 시스템 호출을 공격 유형과 연관시킨 데이터이다.

- [0035] LID-DS 데이터 세트는 기존에 공개되었던 데이터들과 다르게 현재 공개된 데이터 세트들보다 최신 컴퓨터 시스템의 다양한 특징들과 사이버 공격 방법 및 시나리오로 구성되어 있다.
- [0036] 이러한 LID-DS 데이터를 통해 기존에 데이터 세트들의 데이터가 부족하여 기계학습에 적용하기 어려웠던 부분을 해결하고 기계학습 방법을 이용하여 새로운 이상 행동들을 더 정확하게 탐지하여 차단할 수 있으며, 이를 통해 침입 탐지 시스템의 문제점인 오경보율을 줄일 수 있다.
- [0037] LID-DS 데이터 세트는 시스템 호출과 관련된 다양한 데이터가 포함되어 있으며, 소프트웨어와 다양한 사이버 공격이 기록된다. LID-DS 데이터 세트는, 일례로서 도 1의 표에 도시된 바와 같이 사이버 공격 방법과 여러 시나리오로 구성되며, 시나리오를 통해 정상적인 데이터, 비정상적인 데이터를 생성하고 기록하는 프로세스를 구성할 수 있다.
- [0038] 도 2는 본 발명의 실시예에 있어서, LID-DS 데이터 세트의 공격 시뮬레이션 절차를 나타낸 도면이다.
- [0039] 도 2는 도1의 사이버 공격 시나리오를 이용하여 데이터를 생성하는 과정을 나타낸 것으로, LID-DS 데이터 세트의 결과 시스템 호출 추적을 기록하기 위해 공격 대상은 초기 상태를 정의하고, 각 공격 후 초기 상태로 되돌리기 위해 도커(Docker) 10 컨테이너 가상화 소프트웨어 내에서 실행될 수 있다.
- [0040] 기록을 위해 LID-DS 프레임워크를 이용하여 먼저 공격 대상을 호스팅 하는 Docker 컨테이너를 시작한다. 그 다음 시나리오에 따라 초기화 작업이 실행되며 정상 동작의 시뮬레이션이 시작된다.
- [0041] 그 후, 공격 대상 소프트웨어의 시작 효과를 기록하지 않기 위해서 Sysdig가 활성화되기 전에 짧은 시간 동안 기다린다. 공격 동작을 기록하는 경우 임의의 시간이 지나면 공격이 시작되는데 원하는 시간 동안 녹화가 실행된 후 제어 스크립트에 의해 녹화가 중지된다. 또한, 정상적인 동작 및 사용된 Docker 컨테이너의 시뮬레이션을 중지하고 제거한다.
- [0042] ADFA 데이터 세트는 일련의 시스템 호출 ID만 포함하고 현대의 사이버 공격 패턴을 포함하지 않기 때문에 ADFA 데이터 세트를 이용하여 이상 탐지 테스트를 하기에는 적절하지 않다.
- [0043] 실시예에서는 기존에 사용되었던 데이터 세트에서 결여된 스레드 정보, 메타 데이터 및 버퍼 데이터를 포함하고, 데이터가 부족하여 기계학습에 적용하지 못했던 문제를 해결하기 위해서 LID-DS(Leipzig Intrusion Detection-Data Set) 데이터를 사용한다. LID-DS 데이터를 사용하여 기계학습 알고리즘에 관한 비교 연구를 통해 호스트 기반 침입 탐지 시스템이 나아갈 방향을 제시할 수 있다.
- [0044] 도 6을 참조하면, LID-DS의 데이터 파일 자체의 형식을 나타낸 표이다. ADFA와 다르게 LID-DS 데이터에는 시스템 호출의 인수, 반환 값, 고정밀 타임 스탬프, 해당 프로세스 이름 및 데이터 버퍼의 내용이 포함될 수 있다.
- [0045] 도 3은 본 발명의 실시예에 있어서, LID-DS 데이터 세트 성능 비교 모델의 구조를 설명하기 위한 도면이다.
- [0046] 알고리즘 비교 시스템은 LID-DS 데이터 셋에 대한 전처리를 수행할 수 있다. LID-DS 데이터 세트는 복수 개의 사이버 공격 방법에 따라 데이터가 분류될 수 있으며, 각 사이버 공격 방법은 복수 개의 특징으로 구성될 수 있다.
- [0047] 도 7을 참조하면, 데이터 타입과 공격 타입을 설명하기 위한 표이다. 도 7은 LID-DS 데이터 세트에 대한 10개의 사이버 공격 방법에 따라 데이터가 분류되고, 각 사이버 공격 방법은 7개의 특징으로 구성된 것을 나타낸 표이다. 데이터를 기계학습 알고리즘에 사용하기 위해 원본 크기가 큰 일부 데이터를 제외하고 CSV 형식으로 구성될 수 있다.
- [0048] 알고리즘 비교 시스템은 모든 데이터에 대해 Argument Feature와 결측값을 삭제하고, event_time Feature에서 콜론(:)을 제거할 수 있다. event_direction과 event_type은 LabelEncoder를 사용할 수 있다.
- [0049] 프로세스 카테고리는 총 16개로 구성되어 있으며 각 공격 방법마다 프로세스의 개수는 다르다. 그에 따라, 각 사이버 공격 방법에 사용된 프로세스들은 하나의 프로세스로 통합될 수 있다. 그 결과, 총 10개의 프로세스로 구성된 라벨들은 라벨인코더(LabelEncoder)를 사용하여 각 프로세스에 라벨을 붙여 사용할 수 있다.
- [0050] 이때, 원 핫 인코더(OneHotEncoder)를 사용하지 않은 이유는 event_type의 카테고리가 99개로 구성되어 있기 때문이다. 즉, 차원의 저주(Curse of Dimensionality)가 발생할 수 있는 위험이 있으므로 LabelEncoder만이 사용될 수 있다. 정규화(Normalization)는 데이터가 가진 특징의 스케일이 심하게 차이가 나는 문제를 해결하기 위

해서 최소-최대 정규화(Min-Max Normalization)를 사용하여 특징에 대해 각각의 최솟값 0, 최댓값 1로 다른 값들은 0과 1 사이의 값으로 변환될 수 있다.

[0051] 알고리즘 비교 시스템은 전처리가 수행된 LID-DS 데이터를 훈련 데이터 및 테스트 데이터로 분할할 수 있다. 훈련 데이터는 10개의 사이버 공격 방법 데이터를 하나의 CSV 파일로 합쳐서 사용될 수 있다. 기계학습에 사용할 훈련 데이터와 테스트 데이터는 train_test_split 모듈을 사용하여 일반적으로 많이 사용하는 8:2 비율로 나누어 사용될 수 있다.

[0052] 하지만 훈련 데이터를 학습시킨 후 테스트 데이터에 모델을 적용함에도, 과적합(Overfitting) 현상이 발견될 수 있다. 과적합이란 모델이 너무 과적합 되도록 학습한 나머지, 예측물이 현저히 떨어지는 현상을 말한다. 과적합을 방지하기 위하여 다른 비율로 나누어진 데이터를 포함하여 교차 검증(Cross-validation)으로 모델 평가가 진행될 수 있다.

[0053] 그 결과, 특정 비율(예컨대, 7:3 비율)로 나누어진 데이터 세트의 모델 성능이 가장 높게 나올 수 있다. 본 실시예에서는 훈련 데이터와 테스트 데이터를 7:3 비율로 나누어 사용하는 것을 예를 들어 설명한다.

[0054] 알고리즘 비교 시스템은 분할된 훈련 데이터 및 테스트 데이터를 이용하여 기계학습 알고리즘들의 성능을 분석할 수 있다. 알고리즘 비교 시스템은 Decision Tree, Naive Bayes, MLP, Logistic Regression, LSTM, RNN를 포함하는 6가지의 기계학습 알고리즘들의 성능을 분석할 수 있다. 예컨대, 기계학습 알고리즘들의 성능을 분석하기 위하여 실험이 수행될 수 있다. 이 실험에 사용된 알고리즘의 하이퍼 파라미터는 아래의 표 1과 같다.

[0055] [표 1]

Algorithm	Hyper Parameter
Decision Tree	criterion='entropy', max_depth=3, random_state=0
Naive Bayes	default
MLP	max_iter=1000, alpha=1, random_state=0
Logistic Regression	max_iter=500, solver='liblinear'
LSTM	Activation=' softmax' , optimizer=adam,
RNN	loss= 'categorical_crossentropy'

[0056]

[0057] 일례로, LID-DS의 앞서 설명 방법으로 전처리 과정이 수행된 사이버 공격 데이터가 사용될 수 있다. LabelEncoder를 사용하여 라벨을 부여한 Process 카테고리 10개의 데이터를 통하여 LID-DS 데이터 세트의 침입 탐지 정확성을 비교하는 실험이 진행될 수 있다.

[0058] 학습된 모델의 성능 평가는 정밀도(Precision), 재현율(Recall), F1 스코어(F1 Score)를 사용할 수 있다. 그 이유는 데이터에 대해서 정확도만을 가지고 평가하는 것은 부적합하기 때문이다. 또한, 보안에서 중요한 문제점인 오류율을 확인하기 위해 FAR(False Alarm Rate) 및 ERR(Error Rate)의 수치가 확인될 수 있다.

[0059] 일례로, 실험에 사용한 알고리즘은 크게 데이터의 시퀀스(sequence)를 고려하지 아니하고 학습하는 알고리즘 4개와 시퀀스를 고려하여 학습하는 알고리즘 2개로 실험될 수 있다. LID-DS 데이터 세트에서 가장 특징적인 부분이라고 할 수 있는 공격 유형에 따른 연속적인 시스템 콜을 어떠한 기계학습 알고리즘이 가장 잘 찾아낼 수 있는지 평가할 수 있다.

[0060] 실시예에 따르면, LID-DS 데이터의 경우 사이버 공격 데이터를 직접 만들 수 있어서 데이터의 양이 많으므로 기존에 데이터가 부족하여 기계학습을 진행하지 못하는 부분을 해결할 수 있다.

[0061] 도 4는 일 실시예에 따른 알고리즘 비교 시스템의 구성을 설명하기 위한 블록도이고, 도 5는 일 실시예에 따른 알고리즘 비교 시스템에서 기계학습 알고리즘을 비교하는 방법을 설명하기 위한 흐름도이다.

[0062] 알고리즘 비교 시스템(100)의 프로세서는 전처리부(410), 데이터 분할부(420) 및 성능 분석부(430) 등을 포함할 수 있다. 이러한 프로세서의 구성요소들은 알고리즘 비교 시스템에 저장된 프로그램 코드가 제공하는 제어 명령에 따라 프로세서에 의해 수행되는 서로 다른 기능들(different functions)의 표현들일 수 있다. 프로세서 및

프로세서의 구성요소들은 도 5의 기계학습 알고리즘 비교 방법이 포함하는 단계들(510 내지 530)을 수행하도록 알고리즘 비교 시스템을 제어할 수 있다. 이때, 프로세서 및 프로세서의 구성요소들은 메모리가 포함하는 운영체제의 코드와 적어도 하나의 프로그램의 코드에 따른 명령(instruction)을 실행하도록 구현될 수 있다.

- [0063] 프로세서는 기계학습 알고리즘 비교 방법을 위한 프로그램의 파일에 저장된 프로그램 코드를 메모리에 로딩할 수 있다. 예컨대, 알고리즘 비교 시스템에서 프로그램이 실행되면, 프로세서는 운영체제의 제어에 따라 프로그램의 파일로부터 프로그램 코드를 메모리에 로딩하도록 알고리즘 비교 시스템을 제어할 수 있다.
- [0064] 이때, 프로세서 및 프로세서가 포함하는 전처리부(410), 데이터 분할부(420) 및 성능 분석부(430) 각각은 메모리에 로딩된 프로그램 코드 중 대응하는 부분의 명령을 실행하여 이후 단계들(510 내지 530)을 실행하기 위한 프로세서의 서로 다른 기능적 표현들일 수 있다.
- [0065] 단계(510)에서 전처리부(410)는 사이버 공격 데이터에 대한 전처리를 수행할 수 있다. 전처리부(410)는 LID-DS 데이터에 대해 아규먼트 특징(Argument Feature)과 결측값을 삭제하고, event_time Feature에 대해 콜론을 제거하고, event_direction과 event_type에 대해 LabelEncoder를 사용할 수 있다.
- [0066] 전처리부(410)는 각 사이버 공격에 사용된 프로세스들을 하나의 프로세스로 통합하고, 통합된 프로세스로 구성된 라벨들을 라벨인코더(LabelEncoder)를 사용하여 각 프로세스에 라벨을 부착할 수 있다.
- [0067] 전처리부(410)는 최소 최대 정규화(Min-Max Normalization)를 사용하여 LID-DS 데이터가 가진 특징에 대해 각각의 최소값을 0, 최대값을 1, 나머지 값들을 0과 1사이의 값으로 변환할 수 있다.
- [0068] 단계(520)에서 데이터 분할부(420)는 전처리가 수행된 사이버 공격 데이터를 훈련 데이터 및 테스트 데이터로 분할할 수 있다. 데이터 분할부(420)는 기계학습에 사용할 훈련 데이터와 테스트 데이터를 기 설정된 비율로 분할할 수 있다.
- [0069] 단계(530)에서 성능 분석부(430)는 분할된 훈련 데이터 및 테스트 데이터를 이용하여 기계학습 알고리즘들의 성능을 분석할 수 있다. 성능 분석부(430)는 분할된 훈련 데이터를 각각의 기계학습 모델에 입력받고, 각각의 기계학습 모델을 상기 분할된 훈련 데이터를 이용하여 학습시키고, 학습된 각각의 기계학습 모델에 분할된 테스트 데이터를 입력받고, 학습된 각각의 기계학습 모델을 이용하여 분할된 테스트 데이터에 대한 침입 탐지 정확성을 도출할 수 있다.
- [0070] 성능 분석부(430)는 정밀도(Precision), 재현율(Recall), F1 스코어(F1 Score)와 오류율을 사용하여 학습된 각각의 기계학습 모델의 성능을 평가할 수 있다.
- [0071] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.
- [0072] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.
- [0073] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판

독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0074] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

[0075] 그러므로, 다른 구현들, 다른 실시예들 및 청구범위와 균등한 것들도 후술하는 청구범위의 범위에 속한다.

부호의 설명

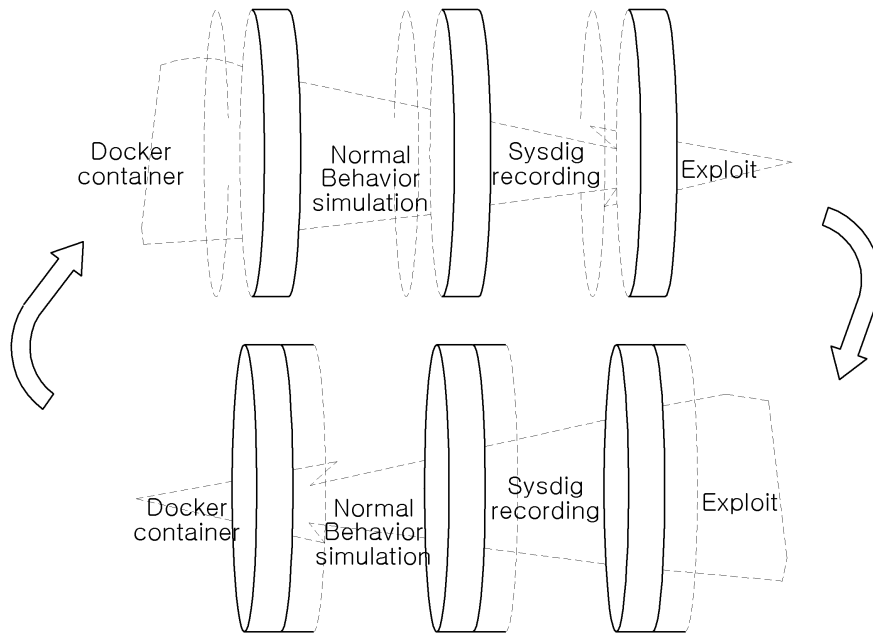
[0076] 410 : 전처리부
420 : 데이터 분할부
430 : 성능 분석부

도면

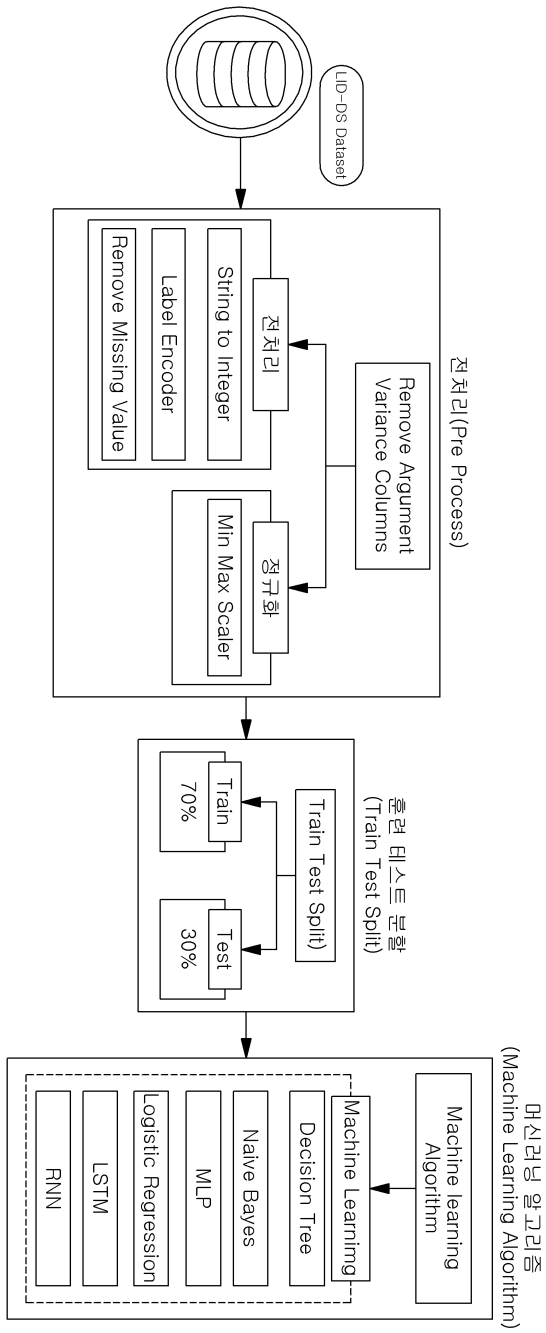
도면1

Cyber attack types	Explanation
CVE-2012-2122	Allows remote attackers to bypass authentication by repeatedly authenticating with the same incorrect password.
CVE-2014-0160	Allows remote attackers to obtain sensitive information from process memory by reading private keys.
Heartbleed	
CWE-307	Improper Restriction of Excessive Authentication Attempts.
CWE-89 SQL Injection	Improper Neutralization fo Special Elements used in an SQL Command.
CWE-434 (PHP, EPS)	Unrestricted Upload of File with Dangerous Type.
CVE-2014-3120	Allows remote attackers to execute arbitrary MVEL expressions and Java code via the source parameter to search.
CVE-2015-1427	Allows remote attackers to bypass the sandbox protection mechanism and execute arbitrary shell commands via a crafted script.
CVE-2017-7529	Vulnerable to integer overflow vulnerability range filter module resulting into leak of potentially sensitive information triggered by specially crafted request.
CVE-2018-3760	Specially crafted requests can be used to access files that exists on the filesystem that is outside an application's root directory.
CVE-2019-5418	Specially crafted accept headers can cause contents of arbitrary files on the target system's filesystem to be exposed.
Zip slip	Arbitrary file overwrite vulnerability that triggers from a directory traversal attack.

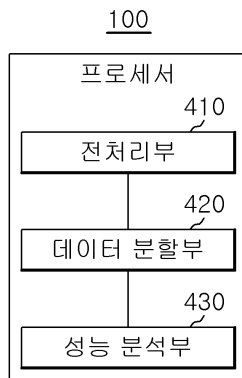
도면2



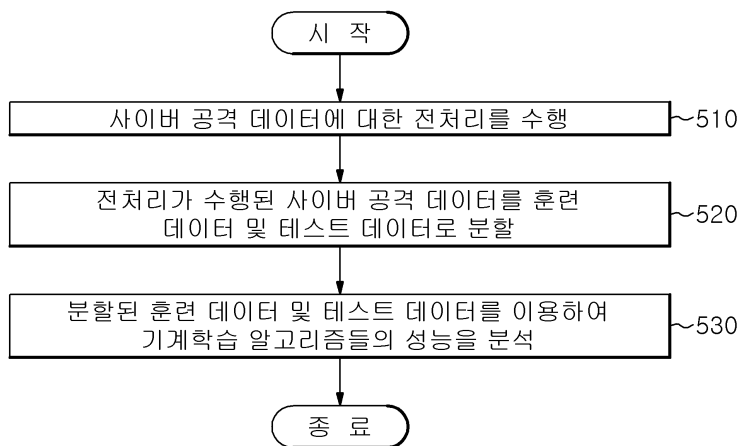
도면3



도면4



도면5



도면6

8	13:16:35.219910910	1	33	apache2	21486	<	epoll_wait
							res=1
9	13:16:35.219919842	1	33	apache2	21486	>	accept flage=0

도면7

Feature	event_num
	event_time
	cpu
	user_uid
	process
	process_id
	event_direction
	event_type
Attack Type	Bruteforce
	CVE-2012-2122
	CVE-2014-0160
	CVE-2017-7529
	CVE-2018-3760
	CVE-2019-5418
	EPS_CWE-434
	PHP_CWE-434
	SQL_Injection_CWE-89
	ZipSlip