



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2023년01월09일

(11) 등록번호 10-2485570

(24) 등록일자 2023년01월03일

(51) 국제특허분류(Int. Cl.)
H04L 9/40 (2022.01) G06N 20/00 (2019.01)
G06N 3/08 (2023.01) H04L 12/40 (2006.01)

(52) CPC특허분류
H04L 63/1416 (2013.01)
G06N 20/00 (2021.08)

(21) 출원번호 10-2022-0093333

(22) 출원일자 2022년07월27일

심사청구일자 2022년07월27일

(56) 선행기술조사문헌

KR1020220046408 A

KR1020180105850 A

KR1020200098904 A

KR1020210043053 A

(73) 특허권자

고려대학교 산학협력단

서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)

(72) 발명자

김휘강

서울특별시 서초구 서운로 104, 201동 804호(서초동, 래미안서초에스티지에스)

이휘재

서울특별시 동대문구 무학로26길 30, 5동 1506호 (용두동, 신동아아파트)

(뒷면에 계속)

(74) 대리인

이대호, 박건홍

전체 청구항 수 : 총 11 항

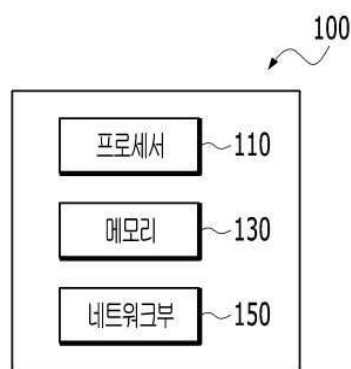
심사관 : 천대식

(54) 발명의 명칭 DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 방법

(57) 요약

본 개시의 몇몇 실시예에 따른 프로세서를 포함하는 컴퓨팅 장치에 의해 수행되는 DBC(database CAN) 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 방법으로서, CAN(Controller Area Network)에서 발생하는 제 1 CAN 메시지를 획득하는 단계; 사전 저장된 DBC 파일에 기초하여 상기 제 1 CAN 메시지를 복수의 카테고리들 중에서 제 1 카테고리로 결정하는 단계; 사전 학습된 제 1 신경망 모델을 이용하여 상기 제 1 CAN 메시지에서부터 제 1 예측 데이터를 획득하는 단계 -상기 사전 학습된 제 1 신경망 모델은 상기 제 1 카테고리에 대응되고 그리고 제 1 히든 레이어(hidden layer)를 포함함-; 및 상기 제 1 CAN 메시지에 기초하여 획득되는 제 1 실제 데이터와 상기 제 1 예측 데이터를 비교하여 상기 제 1 CAN 메시지의 이상 유무를 결정하는 단계;를 포함할 수 있다.

대표도 - 도1



(52) CPC특허분류

G06N 3/08 (2023.01)

H04L 12/40 (2013.01)

H04L 2012/40267 (2013.01)

(72) 발명자

이상호

서울특별시 서대문구 포방터8길 30, 303동 401호(홍은동, 북한산 삼부르네상스아파트)

강연재

경기도 남양주시 별내중앙로 83, 3803동 904호(별내동, 동익미라벨)

피대권

경기도 고양시 일산서구 탄현로 144-87(탄현동)

최재웅

서울특별시 성북구 고려대로 45, 과인빌 401호(안암동2가)

이희주

서울특별시 동대문구 약령시로2길 44, 303호(제기동)

이 발명을 지원한 국가연구개발사업

과제고유번호 1711152707

과제번호 2021-0-01348-002

부처명 과학기술정보통신부

과제관리(전문)기관명 정보통신기획평가원

연구사업명 자율주행기술개발혁신사업(R&D)

연구과제명 자율주행차량의 차세대 내부 네트워크의 보안 및 초고속 무결성 부여 기술 개발

기 여 율 1/1

과제수행기관명 한국전자기술연구원

연구기간 2022.01.01 ~ 2022.12.31

명세서

청구범위

청구항 1

프로세서를 포함하는 컴퓨팅 장치에 의해 수행되는 DBC(database CAN) 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 방법으로서,

CAN(Controller Area Network)에서 발생하는 제 1 CAN 메시지를 획득하는 단계;

사전 저장된 DBC 파일에 기초하여 상기 제 1 CAN 메시지를 복수의 카테고리들 중에서 제 1 카테고리로 결정하는 단계;

사전 학습된 제 1 신경망 모델을 이용하여 상기 제 1 CAN 메시지로부터 제 1 예측 데이터를 획득하는 단계 -상기 사전 학습된 제 1 신경망 모델은 상기 제 1 카테고리에 대응되고 그리고 제 1 히든 레이어(hidden layer)를 포함함-; 및

상기 제 1 CAN 메시지에 기초하여 획득되는 제 1 실제 데이터와 상기 제 1 예측 데이터를 비교하여 상기 제 1 CAN 메시지의 이상 유무를 결정하는 단계;

를 포함하는,

방법.

청구항 2

제 1 항에 있어서,

상기 CAN에서 발생하는 제 2 CAN 메시지를 획득하는 단계;

상기 사전 저장된 DBC 파일에 기초하여 상기 제 2 CAN 메시지를 상기 복수의 카테고리들 중에서 제 2 카테고리로 결정하는 단계;

사전 학습된 제 2 신경망 모델을 이용하여 상기 제 2 CAN 메시지로부터 제 2 예측 데이터를 획득하는 단계 -상기 사전 학습된 제 2 신경망 모델은 상기 제 2 카테고리에 대응되고 그리고 상기 제 1 히든 레이어와 대응되는 제 2 히든 레이어를 포함함-; 및

상기 제 2 CAN 메시지에 기초하여 획득되는 제 2 실제 데이터와 상기 제 2 예측 데이터를 비교하여 상기 제 2 CAN 메시지의 이상 유무를 결정하는 단계;

를 더 포함하는,

방법.

청구항 3

제 1 항에 있어서,

상기 제 1 CAN 메시지에 기초하여 획득되는 제 1 실제 데이터와 상기 제 1 예측 데이터를 비교하여 상기 제 1 CAN 메시지의 이상 유무를 결정하는 단계는,

상기 제 1 실제 데이터와 상기 제 1 예측 데이터의 차이에 기초하여 상기 사전 학습된 제 1 신경망 모델의 제 1 손실(loss) 값을 산출하는 단계; 및

상기 제 1 손실 값과 사전 설정된 제 1 임계 값을 비교하여 상기 제 1 CAN 메시지의 이상 유무를 결정하는 단계;

를 포함하는,
방법.

청구항 4

제 2 항에 있어서,

상기 제 1 히든 레이어로부터 획득되는 제 1 히든 상태(hidden state)의 값과 상기 제 2 히든 레이어로부터 획득되는 제 2 히든 상태의 값을 연결(concatenation)한 제 1 연결 값을 생성하는 단계;

사전 학습된 제 3 신경망 모델을 이용하여, 상기 제 1 연결 값으로부터 제 1 출력 값을 획득하는 단계; 및

상기 제 1 연결 값과 상기 제 1 출력 값을 비교하여 상기 제 1 CAN 메시지 및 상기 제 2 CAN 메시지를 포함하는 전체 CAN 메시지의 이상 유무를 결정하는 단계;

를 더 포함하는,

방법.

청구항 5

제 2 항에 있어서,

상기 제 1 신경망 모델은 상기 제 1 히든 레이어와 상이한 제 3 히든 레이어를 더 포함하고, 상기 제 2 신경망 모델은 상기 제 3 히든 레이어와 대응되는 제 4 히든 레이어를 더 포함하며,

상기 제 2 CAN 메시지에 기초하여 획득되는 제 2 실제 데이터와 상기 제 2 예측 데이터를 비교하여 상기 제 2 CAN 메시지의 이상 유무를 결정하는 단계 이후, 상기 제 3 히든 레이어로부터 획득되는 제 3 히든 상태의 값과 상기 제 4 히든 레이어로부터 획득되는 제 4 히든 상태의 값을 연결(concatenation)한 제 2 연결 값을 생성하는 단계;

사전 학습된 제 3 신경망 모델을 이용하여, 상기 제 2 연결 값으로부터 제 2 출력 값을 획득하는 단계; 및

상기 제 2 연결 값과 상기 제 2 출력 값을 비교하여 상기 제 1 CAN 메시지 및 상기 제 2 CAN 메시지를 포함하는 전체 CAN 메시지의 이상 유무를 결정하는 단계;

를 더 포함하는,

방법.

청구항 6

제 4 항에 있어서,

상기 제 1 연결 값과 상기 제 1 출력 값을 비교하여 상기 제 1 CAN 메시지 및 상기 제 2 CAN 메시지를 포함하는 전체 CAN 메시지의 이상 유무를 결정하는 단계는,

상기 제 1 연결 값과 상기 제 1 출력 값의 차이에 기초하여 상기 사전 학습된 제 3 신경망 모델의 제 2 손실 값을 산출하는 단계; 및

상기 제 2 손실 값과 사전 설정된 제 2 임계 값을 비교하여 상기 전체 CAN 메시지의 이상 유무를 결정하는 단계;

를 포함하는,

방법.

청구항 7

제 1 항에 있어서,

상기 사전 학습된 제 1 신경망 모델은,

상기 제 1 카테고리에 대응되는 CAN 메시지에 기초하여 생성된 학습 데이터를 이용하고, 그리고 출력된 데이터와 실제 데이터 간의 값의 차이가 감소되도록 사전 학습되는,

방법.

청구항 8

제 2 항에 있어서,

상기 사전 학습된 제 2 신경망 모델은,

상기 제 2 카테고리에 대응되는 CAN 메시지에 기초하여 생성된 학습 데이터를 이용하고, 그리고 출력된 데이터와 실제 데이터 간의 값의 차이가 감소되도록 사전 학습되는,

방법.

청구항 9

제 4 항에 있어서,

상기 사전 학습된 제 3 신경망 모델은,

서로 상이한 학습 데이터를 통해 CAN 메시지의 카테고리별로 사전 학습된 복수의 신경망 모델들의 서로 대응되는 히든 레이어에서 각각 획득되는 복수의 히든 상태들의 값을 연결한 연결 데이터를 입력 데이터로 이용하고, 그리고 상기 입력 데이터에 대응되는 출력 데이터를 출력하도록 사전 학습되는,

방법.

청구항 10

컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램으로서, 상기 컴퓨터 프로그램은 DBC(database CAN) 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 컴퓨팅 장치의 프로세서로 하여금 이하의 단계들을 수행하기 위한 명령들을 포함하며, 상기 단계들은:

CAN(Controller Area Network)에서 발생하는 제 1 CAN 메시지를 획득하는 단계;

사전 저장된 DBC 파일에 기초하여 상기 제 1 CAN 메시지를 복수의 카테고리들 중에서 제 1 카테고리로 결정하는 단계;

사전 학습된 제 1 신경망 모델을 이용하여 상기 제 1 CAN 메시지에서부터 제 1 예측 데이터를 획득하는 단계 -상기 사전 학습된 제 1 신경망 모델은 상기 제 1 카테고리에 대응되고 그리고 제 1 히든 레이어(hidden layer)를 포함함-; 및

상기 제 1 CAN 메시지에 기초하여 획득되는 제 1 실제 데이터와 상기 제 1 예측 데이터를 비교하여 상기 제 1 CAN 메시지의 이상 유무를 결정하는 단계;

를 포함하는,

컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 11

DBC(database CAN) 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 컴퓨팅 장치에 있어서,
적어도 하나의 코어를 포함하는 프로세서;
상기 프로세서에 의해 실행가능한 컴퓨터 프로그램을 저장하는 메모리; 및
네트워크부;
를 포함하고,
상기 프로세서는,
CAN(Controller Area Network)에서 발생하는 제 1 CAN 메시지를 획득하고,
사전 저장된 DBC 파일에 기초하여 상기 제 1 CAN 메시지를 복수의 카테고리들 중에서 제 1 카테고리로 결정하고,
사전 학습된 제 1 신경망 모델을 이용하여 상기 제 1 CAN 메시지에서부터 제 1 예측 데이터를 획득하고 -상기 사전 학습된 제 1 신경망 모델은 상기 제 1 카테고리에 대응되고 그리고 제 1 히든 레이어(hidden layer)를 포함함-, 그리고
상기 제 1 CAN 메시지에 기초하여 획득되는 제 1 실제 데이터와 상기 제 1 예측 데이터를 비교하여 상기 제 1 CAN 메시지의 이상 유무를 결정하는,
컴퓨팅 장치.

발명의 설명

기술 분야

[0001] 본 개시는 침입을 탐지하기 위한 방법에 관한 것으로, 보다 구체적으로, DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 방법에 관한 것이다.

배경 기술

[0002] CAN(Controller Area Network)은 1983년에 개발된 통신 규격으로 현재 대부분의 차량 내부 통신에서 사용되고 있으며, 안정성과 저비용의 장점을 기반으로 다양한 가상 물리 시스템(Cyber Physical System, CPS)에서도 내부 통신의 용도로 사용되고 있다.

[0003] 과거 차량에서의 CAN 시스템은 외부와 연결되지 않은 폐쇄적인 시스템이었기에, 프로토콜 설계상 보안이 고려되어 있지 않다. 최근 V2V(vehicle-to-vehicle), V2X(vehicle-to-everything)와 같은 새로운 환경을 통해 차량들이 외부와 연결되며, 보안상의 문제가 발생하게 되었다.

[0004] 기존의 기계학습(machine learning) 기반의 이상징후 탐지 기술은 특정 기능부에 대한 고려 없이 전체 데이터를 한 번에 이용한 기계학습 모델을 생성한다. 따라서, 기존의 기계학습 기반의 이상징후 탐지 기술은 특정 기능부에 대한 탐지가 어렵고, 기능부들 간에 유사성을 학습하기 어렵다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 대한민국 등록특허 제10-1754951호(2017.06.30. 등록)

발명의 내용

해결하려는 과제

[0006] 본 개시는 전술한 배경기술에 대응하여 안출된 것으로, DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 방법을 제공하고자 한다.

[0007] 본 개시의 기술적 과제들은 이상에서 언급한 기술적 과제로 제한되지 않으며, 언급되지 않은 또 다른 기술적 과

제들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

- [0008] 전술한 바와 같은 과제를 해결하기 위한 본 개시의 몇몇 실시예에 따라, 프로세서를 포함하는 컴퓨팅 장치에 의해 수행되는 DBC(database CAN) 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 방법으로서, CAN(Controller Area Network)에서 발생하는 제 1 CAN 메시지를 획득하는 단계; 사전 저장된 DBC 파일에 기초하여 상기 제 1 CAN 메시지를 복수의 카테고리들 중에서 제 1 카테고리로 결정하는 단계; 사전 학습된 제 1 신경망 모델을 이용하여 상기 제 1 CAN 메시지로부터 제 1 예측 데이터를 획득하는 단계 -상기 사전 학습된 제 1 신경망 모델은 상기 제 1 카테고리에 대응되고 그리고 제 1 히든 레이어(hidden layer)를 포함함-; 및 상기 제 1 CAN 메시지에 기초하여 획득되는 제 1 실제 데이터와 상기 제 1 예측 데이터를 비교하여 상기 제 1 CAN 메시지의 이상 유무를 결정하는 단계;를 포함할 수 있다.
- [0009] 대안적으로, 상기 CAN에서 발생하는 제 2 CAN 메시지를 획득하는 단계; 상기 사전 저장된 DBC 파일에 기초하여 상기 제 2 CAN 메시지를 상기 복수의 카테고리들 중에서 제 2 카테고리로 결정하는 단계; 사전 학습된 제 2 신경망 모델을 이용하여 상기 제 2 CAN 메시지로부터 제 2 예측 데이터를 획득하는 단계 -상기 사전 학습된 제 2 신경망 모델은 상기 제 2 카테고리에 대응되고 그리고 상기 제 1 히든 레이어와 대응되는 제 2 히든 레이어를 포함함-; 및 상기 제 2 CAN 메시지에 기초하여 획득되는 제 2 실제 데이터와 상기 제 2 예측 데이터를 비교하여 상기 제 2 CAN 메시지의 이상 유무를 결정하는 단계;를 더 포함할 수 있다.
- [0010] 대안적으로, 상기 제 1 CAN 메시지에 기초하여 획득되는 제 1 실제 데이터와 상기 제 1 예측 데이터를 비교하여 상기 제 1 CAN 메시지의 이상 유무를 결정하는 단계는, 상기 제 1 실제 데이터와 상기 제 1 예측 데이터의 차이에 기초하여 상기 사전 학습된 제 1 신경망 모델의 제 1 손실(loss) 값을 산출하는 단계; 및 상기 제 1 손실 값과 사전 설정된 제 1 임계 값을 비교하여 상기 제 1 CAN 메시지의 이상 유무를 결정하는 단계;를 포함할 수 있다.
- [0011] 대안적으로, 상기 제 1 히든 레이어로부터 획득되는 제 1 히든 상태(hidden state)의 값과 상기 제 2 히든 레이어로부터 획득되는 제 2 히든 상태의 값을 연결(concatenation)한 제 1 연결 값을 생성하는 단계; 사전 학습된 제 3 신경망 모델을 이용하여, 상기 제 1 연결 값으로부터 제 1 출력 값을 획득하는 단계; 및 상기 제 1 연결 값과 상기 제 1 출력 값을 비교하여 상기 제 1 CAN 메시지 및 상기 제 2 CAN 메시지를 포함하는 전체 CAN 메시지의 이상 유무를 결정하는 단계;를 더 포함할 수 있다.
- [0012] 대안적으로, 상기 제 1 신경망 모델은 상기 제 1 히든 레이어와 상이한 제 3 히든 레이어를 더 포함하고, 상기 제 2 신경망 모델은 상기 제 3 히든 레이어와 대응되는 제 4 히든 레이어를 더 포함하며, 상기 제 2 CAN 메시지에 기초하여 획득되는 제 2 실제 데이터와 상기 제 2 예측 데이터를 비교하여 상기 제 2 CAN 메시지의 이상 유무를 결정하는 단계 이후, 상기 제 3 히든 레이어로부터 획득되는 제 3 히든 상태의 값과 상기 제 4 히든 레이어로부터 획득되는 제 4 히든 상태의 값을 연결(concatenation)한 제 2 연결 값을 생성하는 단계; 사전 학습된 제 3 신경망 모델을 이용하여, 상기 제 2 연결 값으로부터 제 2 출력 값을 획득하는 단계; 및 상기 제 2 연결 값과 상기 제 2 출력 값을 비교하여 상기 제 1 CAN 메시지 및 상기 제 2 CAN 메시지를 포함하는 전체 CAN 메시지의 이상 유무를 결정하는 단계;를 더 포함할 수 있다.
- [0013] 대안적으로, 상기 제 1 연결 값과 상기 제 1 출력 값을 비교하여 상기 제 1 CAN 메시지 및 상기 제 2 CAN 메시지를 포함하는 전체 CAN 메시지의 이상 유무를 결정하는 단계는, 상기 제 1 연결 값과 상기 제 1 출력 값의 차이에 기초하여 상기 사전 학습된 제 3 신경망 모델의 제 2 손실 값을 산출하는 단계; 및 상기 제 2 손실 값과 사전 설정된 제 2 임계 값을 비교하여 상기 전체 CAN 메시지의 이상 유무를 결정하는 단계;를 포함할 수 있다.
- [0014] 대안적으로, 상기 사전 학습된 제 1 신경망 모델은, 상기 제 1 카테고리에 대응되는 CAN 메시지에 기초하여 생성된 학습 데이터를 이용하고, 그리고 출력된 데이터와 실제 데이터 간의 값의 차이가 감소되도록 사전 학습될 수 있다.
- [0015] 대안적으로, 상기 사전 학습된 제 2 신경망 모델은, 상기 제 2 카테고리에 대응되는 CAN 메시지에 기초하여 생성된 학습 데이터를 이용하고, 그리고 출력된 데이터와 실제 데이터 간의 값의 차이가 감소되도록 사전 학습될 수 있다.
- [0016] 대안적으로, 상기 사전 학습된 제 3 신경망 모델은, 서로 상이한 학습 데이터를 통해 CAN 메시지의 카테고리별로 사전 학습된 복수의 신경망 모델들의 서로 대응되는 히든 레이어에서 각각 획득되는 복수의 히든 상태들의

값을 연결한 연결 데이터를 입력 데이터로 이용하고, 그리고 상기 입력 데이터에 대응되는 출력 데이터를 출력하도록 사전 학습될 수 있다.

[0017] 전술한 바와 같은 과제를 해결하기 위한 본 개시의 몇몇 다른 실시예에 따라, 컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램으로서, 상기 컴퓨터 프로그램은 DBC(database CAN) 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 컴퓨팅 장치의 프로세서로 하여금 이하의 단계들을 수행하기 위한 명령들을 포함하며, 상기 단계들은: CAN(Controller Area Network)에서 발생하는 제 1 CAN 메시지를 획득하는 단계; 사전 저장된 DBC 파일에 기초하여 상기 제 1 CAN 메시지를 복수의 카테고리들 중에서 제 1 카테고리로 결정하는 단계; 사전 학습된 제 1 신경망 모델을 이용하여 상기 제 1 CAN 메시지로부터 제 1 예측 데이터를 획득하는 단계 -상기 사전 학습된 제 1 신경망 모델은 상기 제 1 카테고리에 대응되고 그리고 제 1 히든 레이어(hidden layer)를 포함함-; 및 상기 제 1 CAN 메시지에 기초하여 획득되는 제 1 실제 데이터와 상기 제 1 예측 데이터를 비교하여 상기 제 1 CAN 메시지의 이상 유무를 결정하는 단계;를 포함할 수 있다.

[0018] 전술한 바와 같은 과제를 해결하기 위한 본 개시의 몇몇 또 다른 실시예에 따라, DBC(database CAN) 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 컴퓨팅 장치에 있어서, 적어도 하나의 코어를 포함하는 프로세서; 상기 프로세서에 의해 실행가능한 컴퓨터 프로그램을 저장하는 메모리; 및 네트워크부;를 포함하고, 상기 프로세서는, CAN(Controller Area Network)에서 발생하는 제 1 CAN 메시지를 획득하고, 사전 저장된 DBC 파일에 기초하여 상기 제 1 CAN 메시지를 복수의 카테고리들 중에서 제 1 카테고리로 결정하고, 사전 학습된 제 1 신경망 모델을 이용하여 상기 제 1 CAN 메시지로 부터 제 1 예측 데이터를 획득하고 -상기 사전 학습된 제 1 신경망 모델은 상기 제 1 카테고리에 대응되고 그리고 제 1 히든 레이어(hidden layer)를 포함함-, 그리고 상기 제 1 CAN 메시지에 기초하여 획득되는 제 1 실제 데이터와 상기 제 1 예측 데이터를 비교하여 상기 제 1 CAN 메시지의 이상 유무를 결정할 수 있다.

발명의 효과

[0019] 본 개시는 DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지할 수 있다.

[0020] 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

[0021] 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 몇몇 실시예에 따른 DBC(database CAN) 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 방법을 제공하기 위한 동작을 수행하는 컴퓨팅 장치의 블록 구성도이다.

도 2는 본 개시의 몇몇 실시예에 따른 신경망 모델을 나타낸 개략도이다.

도 3은 본 개시의 몇몇 실시예에 따른 DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 컴퓨팅 장치의 프로세서의 블록 구성도이다.

도 4는 본 개시의 몇 실시예에 따른 CAN 메시지의 구성을 설명하기 위한 도면이다.

도 5는 본 개시의 몇몇 실시예에 따른 DBC 파일의 구성을 설명하기 위한 블록 구성도이다.

도 6은 본 개시의 몇몇 실시예에 따른 제 1 신경망 모델을 설명하기 위한 블록 구성도이다.

도 7은 본 개시의 몇몇 실시예에 따른 제 3 신경망 모델을 설명하기 위한 블록 구성도이다.

도 8은 본 개시의 몇 실시예에 따른 컴퓨팅 장치에 의해 수행되는 DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 방법을 설명하기 위한 순서도이다.

도 9는 본 개시의 실시예들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 간략하고 일반적인 개략도를 도시한다.

발명을 실시하기 위한 구체적인 내용

- [0022] 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.
- [0023] 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 컴포넌트일 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.
- [0024] 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 "및/또는"이라는 용어는 열거된 관련 아이템들 중 하나 이상의 아이템의 가능한 모든 조합을 지칭하고 포함하는 것으로 이해되어야 한다.
- [0025] 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.
- [0026] 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.
- [0027] 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 방법으로 설명된 기능성을 구현할 수 있다. 다만, 그러한 구현의 결정들이 본 개시내용의 영역을 벗어나게 하는 것으로 해석되어서는 안된다.
- [0028] 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 벗어나지 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 발명은 여기에 제시된 실시예 들로 한정되는 것이 아니다. 본 발명은 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의의 범위에서 해석되어야 할 것이다.
- [0030] 도 1은 본 개시의 몇몇 실시예에 따른 이상을 탐지하는 방법을 제공하기 위한 동작을 수행하는 컴퓨팅 장치의 블록 구성도이다.
- [0031] 도 1에 도시된 컴퓨팅 장치(100)의 구성은 간략화 하여 나타난 예시일 뿐이다. 본 개시의 일 실시예에서 컴퓨팅 장치(100)는 컴퓨팅 장치(100)의 컴퓨팅 환경을 수행하기 위한 다른 구성들이 포함될 수 있고, 개시된 구성들 중 일부만이 컴퓨팅 장치(100)를 구성할 수도 있다.
- [0032] 본 개시의 몇몇 실시예에 따른 컴퓨팅 장치(100)는 CAN(Controller Area Network)에서 발생하는 CAN 메시지의 이상 유무를 판단하기 위한 장치일 수 있다. 컴퓨팅 장치(100)는 신경망 모델들을 이용하여 CAN 메시지의 이상

유무를 판단하기 위한 장치일 수 있다. 예를 들어, 컴퓨팅 장치(100)는 DBC(database CAN) 파일 및 신경망 모델을 이용하여 시퀀스 기반의 침입을 탐지하여, CAN 메시지의 이상 유무를 판단하기 위한 장치일 수 있다.

[0033] 그리고, CAN 메시지에 이상이 존재하는 경우, 컴퓨팅 장치(100)는 CAN 메시지를 수집한 해당 CAN이 사용된 장치(예를 들어, 차량 등)를 위험 상태로 인식하고, 외부 기기(예를 들어, 차량 사용자의 단말 등)에 위험 알람 신호를 전송할 수도 있다.

[0034] 컴퓨팅 장치(100)는 프로세서(110), 메모리(130), 네트워크부(150)를 포함할 수 있다.

[0035] 프로세서(110)는 하나 이상의 코어로 구성될 수 있으며, 컴퓨팅 장치의 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 데이터 분석, 딥러닝을 위한 프로세서를 포함할 수 있다. 프로세서(110)는 메모리(130)에 저장된 컴퓨터 프로그램을 판독하여 본 개시의 일 실시예에 따른 기계 학습을 위한 데이터 처리를 수행할 수 있다. 본 개시의 일 실시예에 따라 프로세서(110)는 신경망의 학습을 위한 연산을 수행할 수 있다. 프로세서(110)는 딥러닝(DL: deep learning)에서 학습을 위한 입력 데이터의 처리, 입력 데이터에서의 피쳐 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 장치의 프로세서를 함께 사용하여 네트워크 함수의 학습, 네트워크 함수를 이용한 데이터 분류를 처리할 수 있다. 또한, 본 개시의 일 실시예에 따른 컴퓨팅 장치에서 수행되는 컴퓨터 프로그램은 CPU, GPGPU 또는 TPU 실행가능 프로그램일 수 있다.

[0036] 본 개시의 몇몇 실시예에 따르면, 메모리(130)는 프로세서(110)가 생성하거나 결정한 임의의 형태의 정보 및 네트워크부(150)가 수신한 임의의 형태의 정보를 저장할 수 있다.

[0037] 본 개시의 몇몇 실시예에 따르면, 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, 광디스크 중 적어도 하나의 타입의 저장매체를 포함할 수 있다. 컴퓨팅 장치(100)는 인터넷(internet) 상에서 상기 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.

[0038] 본 개시의 몇몇 실시예에 따른 네트워크부(150)는 임의의 형태의 데이터 및 신호 등을 송수신할 수 있는 임의의 유무선 통신 네트워크가 본 개시 내용에서 표현되는 네트워크에 포함될 수 있다.

[0039] 본 명세서에서 설명된 기술들은 위에서 언급된 네트워크들뿐만 아니라, 다른 네트워크들에서도 사용될 수 있다.

[0041] 도 2는 본 개시의 몇몇 실시예에 따른 신경망 모델을 나타낸 개략도이다.

[0042] 본 명세서에 걸쳐, 연산 모델, 신경망, 신경망 모델, 네트워크 함수, 뉴럴 네트워크(neural network)는 동일한 의미로 사용될 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. 신경망들을 구성하는 노드(또는 뉴런)들은 하나 이상의 링크에 의해 상호 연결될 수 있다.

[0043] 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.

[0044] 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 가변될 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 대응

하는 링크에 설정된 가중치에 기초하여 출력 노드 값을 결정할 수 있다.

- [0045] 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.
- [0046] 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 거리에 기초하여, 하나의 레이어(layer)를 구성할 수 있다. 예를 들어, 최초 입력 노드로부터 거리가 n 인 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.
- [0047] 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 최종 출력 노드가 아닌 신경망을 구성하는 노드들을 의미할 수 있다.
- [0048] 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.
- [0049] 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, 음악의 잠재적인 구조(예를 들어, 어떤 물체가 사진에 있는지, 글의 내용과 감정이 무엇인지, 음성의 내용과 감정이 무엇인지 등)를 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, 삼 네트워크, 적대적 생성 네트워크(GAN: Generative Adversarial Network) 등을 포함할 수 있다. 전술한 딥 뉴럴 네트워크의 기재는 예시일 뿐이며 본 개시는 이에 제한되지 않는다.
- [0050] 본 개시의 일 실시예에서 네트워크 함수는 오토 인코더(autoencoder)를 포함할 수도 있다. 오토 인코더는 입력 데이터와 유사한 출력 데이터를 출력하기 위한 인공 신경망의 일종일 수 있다. 오토 인코더는 적어도 하나의 히든 레이어를 포함할 수 있으며, 홀수 개의 히든 레이어가 입출력 레이어 사이에 배치될 수 있다. 각각의 레이어의 노드의 수는 입력 레이어의 노드의 수에서 병목 레이어(인코딩)라는 중간 레이어로 축소되었다가, 병목 레이어에서 출력 레이어(입력 레이어와 대칭)로 축소와 대칭되어 확장될 수도 있다. 오토 인코더는 비선형 차원 감소를 수행할 수 있다. 입력 레이어 및 출력 레이어의 수는 입력 데이터의 전처리 이후에 차원과 대응될 수 있다. 오토 인코더 구조에서 인코더에 포함된 히든 레이어의 노드의 수는 입력 레이어에서 멀어질수록 감소하는 구조를 가질 수 있다. 병목 레이어(인코더와 디코더 사이에 위치하는 가장 적은 노드를 가진 레이어)의 노드의 수는 너무 작은 경우 충분한 양의 정보가 전달되지 않을 수 있으므로, 특정 수 이상(예를 들어, 입력 레이어의 절반 이상 등)으로 유지될 수도 있다.
- [0051] 뉴럴 네트워크는 교사 학습(supervised learning), 비교사 학습(unsupervised learning), 반교사학습(semi supervised 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.

[0052] 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. 교사 학습의 경우 각각의 학습 데이터에 정답이 라벨링되어있는 학습 데이터를 사용하며(즉, 라벨링된 학습 데이터), 비교사 학습의 경우는 각각의 학습 데이터에 정답이 라벨링되어 있지 않을 수 있다. 즉, 예를 들어 데이터 분류에 관한 교사 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 비교사 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.

[0053] 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, 레귤라이제이션(regularization), 학습의 과정에서 네트워크의 노드 일부를 비활성화하는 드롭아웃(dropout), 배치 정규화 레이어(batch normalization layer)의 활용 등의 방법이 적용될 수 있다.

[0055] 본 개시의 일 실시예에 따라 데이터 구조를 저장한 컴퓨터 판독가능 매체가 개시된다.

[0056] 데이터 구조는 데이터에 효율적인 접근 및 수정을 가능하게 하는 데이터의 조직, 관리, 저장을 의미할 수 있다. 데이터 구조는 특정 문제(예를 들어, 최단 시간으로 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, 검색의 효율성을 높일 수 있다.

[0057] 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.

- [0058] 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.
- [0059] 본 명세서에 걸쳐, 연산 모델, 신경망, 신경망 모델, 네트워크 함수, 뉴럴 네트워크(neural network)는 동일한 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 활성화 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 활성화 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 발생하는 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.
- [0060] 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 발생하는 데이터를 포함할 수 있다. 전술한 데이터 구조는 예시일 뿐 본 개시는 이에 제한되지 않는다.
- [0061] 데이터 구조는 신경망의 가중치를 포함할 수 있다. (본 명세서에서 가중치, 파라미터는 동일한 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 가변될 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.
- [0062] 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 예시일 뿐 본 개시는 이에 제한되지 않는다.
- [0063] 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, Trie, m-way search tree, AVL tree, Red-Black Tree)를 포함할 수 있다. 전술한 사항은 예시일 뿐 본 개시는 이에 제한되지 않는다.

- [0064] 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 사용자에게 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.
- [0066] 앞서 도 1 및 2를 통해 서술한 신경망 모델을 이용하여 CAN 메시지의 이상 유무를 판단하기 위한 프로세서(110)의 구성에 대해서 도 3을 참조하여 후술한다.
- [0067] 도 3은 본 개시의 몇몇 실시예에 따른 DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 컴퓨팅 장치의 프로세서의 블록 구성도이다.
- [0068] 도 3을 참조하면, 컴퓨팅 장치(100)의 프로세서(110)는 메시지 수집부(200), 전처리부(300), 제 1 신경망 모델(400), 제 2 신경망 모델(500) 및 제 3 신경망 모델(600)를 포함할 수 있다. 다만, 상술한 구성 요소들은 프로세서(110)를 구현하는데 있어서, 필수적인 것은 아니며, 프로세서(110)는 위에 열거된 구성 요소들보다 많거나, 또는 적은 구성 요소들을 가질 수 있다. 또한, 프로세서(110)는 상술한 구성들을 소프트웨어로 구현할 수 있다.
- [0070] 메시지 수집부(200)는 CAN에서 발생하는 적어도 하나의 CAN 메시지(예를 들어, 제 1 CAN 메시지, 제 2 CAN 메시지 등)를 획득할 수 있다. 예를 들어, 메시지 수집부(200)는 CAN의 버스에서 발생하는 트래픽(traffic)을 스니핑(sniffing)하여 적어도 하나의 CAN 메시지를 수집할 수 있다. 다만, 이에 한정되지 않으며, 메시지 수집부(200)는 상술한 방법 이외에 다양한 방법을 통해 적어도 하나의 CAN 메시지를 획득할 수 있다.
- [0071] CAN은 버스를 통해 연결된 시스템들 간에 데이터 전송 및 수신을 포함하는 통신을 수행하기 위해 개발된 통신 규격일 수 있다.
- [0072] CAN의 구조는 특정 시스템(예를 들어, 차량 시스템, 자동화 기기 시스템, 의료용 장비 시스템 등)을 구성하는 노드들(예를 들어, 차량 안전 시스템, 차량 편의사양 시스템 등)이 버스(bus)를 통해 연결된 구조일 수 있다.
- [0073] CAN에서 노드들은 각각 마스터로서 병렬로 연결되고, 모든 노드에서 송수신 되는 데이터에 접근이 가능한 멀티 마스터(multi-master) 방식으로 동작할 수 있다.
- [0074] CAN에서 버스는 노드들이 서로 데이터를 전송하는 경우, 공용으로 사용되는 전기적 통로를 의미할 수 있다.
- [0075] CAN 메시지는 CAN에서 발생하는 메시지로, CAN에 포함된 시스템들 각각에서 데이터를 전송하기 위해 생성되는 메시지 일 수 있다. CAN 메시지는 중재 필드 및 페이로드 필드를 포함할 수 있다. 중재 필드는 Arbitration ID를 포함할 수 있다. Arbitration ID는 11비트 또는 29비트로 구성되고, CAN 메시지의 식별자일 수 있다. 페이로드 필드는 0 내지 64비트로 구성되고, 전송되는 실제 데이터의 값인 페이로드를 포함할 수 있다. 페이로드는 특정 노드에서 다른 노드로 전송하려는 정보를 포함할 수 있다. 정보는 DBC(Database CAN) 파일에 기초하여 해석될 수 있다. DBC(Database CAN) 파일은 Arbitration ID 및 Arbitration ID에 대응되는 페이로드 필드별 의미와 기능에 대한 정보를 포함하는 데이터베이스 파일일 수 있다. 따라서, 프로세서(110)는 DBC 파일을 통해 페이로드 필드에 기재된 페이로드를 해석하여 특정 노드에서 다른 노드로 전송하려는 정보를 획득할 수 있다.
- [0077] 전처리부(300)는 사전 저장된 DBC 파일에 기초하여 CAN 메시지를 복수의 카테고리들 중에서 하나의 카테고리로 결정할 수 있다. 예를 들어, 전처리부(300)는 사전 저장된 DBC 파일에 기초하여 제 1 CAN 메시지를 복수의 카테고리들 중에서 제 1 카테고리로 결정할 수 있다. 다른 예를 들어, 전처리부(300)는 사전 저장된 DBC 파일에 기초하여 제 2 CAN 메시지를 복수의 카테고리들 중에서 제 2 카테고리로 결정할 수 있다.
- [0078] 복수의 카테고리들은 기능을 기준으로 분류된 카테고리일 수 있다. 예를 들어, 복수의 카테고리들은 계기판과 관련된 기능인 클러스터(cluster, CLU) 관련 기능, 스마트 크루즈 컨트롤(smart cruise control, SCC) 관련 기능, 에어백 제어 유닛(airbag control units, ACU) 관련 기능 등을 포함할 수 있다. 예를 들어, 클러스터 관련 기능은 자동차 속도에 영향을 주는 시그널(signal)인 바퀴의 속도를 나타내는 전자 제어 유닛(electronic control unit, ECU)을 포함할 수 있다. 다른 예를 들어, 스마트 크루즈 컨트롤과 관련된 기능은 자율 주행에 필

요한 핸들 조향 및 차선 인식과 관련된 전자 제어 유닛을 포함할 수 있다. 다만, 복수의 카테고리들은 이에 한정되지 않고, 다양한 기능과 관련된 카테고리를 포함할 수 있다.

- [0079] 전처리부(300)는 사전 저장된 DBC 파일에 기초하여 복수의 CAN 메시지들을 카테고리별로 분류하여 복수의 신경망 모델들의 학습 데이터로 각각 생성할 수 있다. 예를 들어, 전처리부(300)는 클러스터 관련 기능의 카테고리에 해당되는 CAN 메시지를 이용하여 복수의 신경망 모델들 중에서 제 1 신경망 모델의 학습 데이터를 생성할 수 있다. 다른 예를 들어, 전처리부(300)는 스마트 크루즈 컨트롤 관련 기능의 카테고리에 해당되는 CAN 메시지를 이용하여 복수의 신경망 모델들 중에서 제 2 신경망 모델의 학습 데이터를 생성할 수 있다.
- [0080] 전처리부(300)는 사전 결정된 시간대(time window)에 대응되도록 CAN 메시지의 크기를 조정할 수 있다. 예를 들어, 전처리부(300)의 사전 결정된 시간대가 10초인 경우, CAN 메시지의 크기를 10초에 대응되도록 조정할 수 있다. 시간대는 전체 시간 중에서 노이즈로 인한 영향을 줄이기 위해 초기 시간(예를 들어, 0~5초) 및 종료 시간(예를 들어, 종료되기 0~5초 전)을 제외한 나머지 시간에서 사전에 결정될 수 있다. 다만, 이에 한정되지 않으며, 시간대는 다양한 방법을 통해 사전에 결정될 수 있다.
- [0081] 전처리부(300)는 CAN 메시지에서 각 시간에 따라 학습 데이터를 생성할 수 있다. 전처리부(300)는 CAN 메시지에서 제 1 시간에 대응되는 CAN 메시지의 데이터를 입력 데이터로 결정하고, 제 1 시간의 다음 시간인 제 2 시간에 대응되는 CAN 메시지의 데이터를 정답 데이터로 결정하여 입력 데이터에 정답 데이터가 라벨링되어 있는 학습 데이터를 생성할 수 있다. 여기서, CAN 메시지는 사전 결정된 시간대에 대응되도록 조정된 CAN 메시지, 또는 조정이 이루어지지 않은 원본 CAN 메시지일 수 있다.
- [0083] 제 1 신경망 모델(400)은 시퀀스 데이터(sequence data)의 처리가 가능한 모델로, 히든 레이어를 포함하는 모델일 수 있다. 예를 들어, 제 1 신경망 모델(400)은 리커런트 뉴럴 네트워크(recurrent neural network, RNN), 장단기 메모리 알고리즘(long short-term memory, LSTM), 게이트 순환 유닛(gated recurrent unit, GRU) 등을 포함할 수 있다. 제 1 신경망 모델(400)은 주로 장단기 메모리 알고리즘을 사용할 수 있다. 다만, 이에 한정되지 않으며, 제 1 신경망 모델(400)은 다양한 시퀀스 데이터 처리가 가능한 모델을 포함할 수 있다.
- [0084] 제 1 신경망 모델(400)은 복수의 입력 레이어(input layer)들 중 적어도 하나, 복수의 히든 레이어(hidden layer)들 중 적어도 하나, 및/또는 복수의 출력 레이어(output layer)들 중 적어도 하나를 포함할 수 있다.
- [0085] 복수의 입력 레이어들은 각각 입력 데이터가 직접 입력되는 최초 레이어를 의미할 수 있다.
- [0086] 복수의 히든 레이어들은 복수의 입력 레이어들 각각에 대응되어 구성되고, 복수의 입력 레이어들 각각으로부터 데이터를 입력 받는 레이어를 의미할 수 있다. 복수의 히든 레이어들 각각은 히든 상태(hidden state)의 값을 포함할 수 있다. 히든 상태는 히든 레이어의 내부 상태를 의미할 수 있다. 예를 들어, 히든 상태는 히든 레이어를 구성하는 뉴런(neuron)들의 내부 상태를 의미할 수 있다. 히든 상태의 값은 히든 레이어의 내부 상태를 나타내는 값을 의미할 수 있다. 예를 들어, 히든 상태의 값은 히든 상태의 값은 피드백 구조에 의해 이전 시점에서 처리된 정보들을 반영한 히든 레이어의 내부 상태를 나타내는 값을 의미할 수 있다. 히든 상태의 값은 벡터 형태의 값일 수 있다.
- [0087] 복수의 출력 레이어들은 복수의 히든 레이어들 각각에 대응되어 구성되고, 복수의 히든 레이어들 각각으로부터 데이터를 입력 받는 최종 레이어를 의미할 수 있다. 예를 들어, 제 1 신경망 모델(400)은 제 1 입력 데이터를 제 1 입력 레이어에 입력시키고, 제 1 입력 레이어에서 생성된 데이터를 제 1 히든 레이어에 입력시키고, 제 1 히든 레이어에서 생성된 데이터를 제 1 출력 레이어에 입력시킴으로써, 제 1 출력 레이어를 통해 제 1 입력 데이터에 대한 제 1 출력 데이터를 출력할 수 있다. 다른 예를 들어, 제 1 신경망 모델(400)은 제 2 입력 데이터를 제 2 입력 레이어에 입력시키고, 제 2 입력 레이어에서 생성된 데이터를 제 2 히든 레이어에 입력시키고, 제 2 히든 레이어에서 생성된 데이터를 제 2 출력 레이어에 입력시킴으로써, 제 2 출력 레이어를 통해 제 2 입력 데이터에 대한 제 2 출력 데이터를 출력할 수 있다.
- [0088] 제 1 신경망 모델(400)은 전처리부(300)에서 생성된 특정 카테고리에 대응되는 학습 데이터를 이용하여 사전 학습될 수 있다. 예를 들어, 제 1 신경망 모델(400)은 복수의 카테고리들 중에서 제 1 카테고리에 대응되는 CAN 메시지에 기초하여 생성된 학습 데이터를 이용하여 사전 학습될 수 있다. 제 1 신경망 모델(400)은 출력된 데이터와 실제 데이터 간의 값의 차이가 감소되도록 사전 학습될 수 있다. 예를 들어, 제 1 신경망 모델(400)은 학습 데이터에 포함된 입력 데이터를 입력하여 출력된 데이터와 입력 데이터에 라벨링된 정답 데이터(즉, 실제 데이터) 간의 값의 차이가 감소되도록 학습될 수 있다. 정답 데이터는 학습 데이터를 생성할 때 사용된 CAN 메시지 중에서 실제로 획득된 값으로, 실제 데이터일 수 있다.

- [0090] 프로세서(110)는 사전 학습된 제 1 신경망 모델(400)을 이용하여 CAN 메시지에서부터 예측 데이터를 획득할 수 있다.
- [0091] 예를 들어, 프로세서(110)는 사전 학습된 제 1 신경망 모델(400)을 이용하여 제 1 CAN 메시지에서부터 제 1 예측 데이터를 획득할 수 있다. 여기서, 사전 학습된 제 1 신경망 모델(400)은 제 1 카테고리에 대응되고, 그리고 제 1 히든 레이어를 포함할 수 있다. 여기서, 사전 학습된 제 1 신경망 모델(400)은 제 1 카테고리에 해당되는 학습 데이터를 이용하여 사전에 학습된 모델일 수 있다.
- [0092] 다른 예를 들어, 프로세서(110)는 사전 학습된 제 2 신경망 모델(500)을 이용하여 제 2 CAN 메시지에서부터 제 2 예측 데이터를 획득할 수 있다. 여기서, 사전 학습된 제 2 신경망 모델(500)은 제 2 카테고리에 대응되고, 그리고 사전 학습된 제 1 신경망 모델(400)의 제 1 히든 레이어와 대응되는 제 2 히든 레이어를 포함할 수 있다. 여기서, 사전 학습된 제 2 신경망 모델(500)은 제 2 카테고리에 해당되는 학습 데이터를 이용하여 사전에 학습된 모델일 수 있다.
- [0094] 프로세서(110)는 CAN 메시지에 기초하여 획득되는 실제 데이터와 예측 데이터를 비교하여 CAN 메시지의 이상 유무를 결정할 수 있다. 예측 데이터는 CAN 메시지의 제 1 시간에 대응되는 CAN 메시지의 데이터를 신경망 모델(예를 들어, 제 1 신경망 모델, 제 2 신경망 모델 등)에 입력하여 제 1 시간의 다음 시간인 제 2 시간에 대응되는 CAN 메시지의 데이터를 예측하여 출력된 데이터일 수 있다. 실제 데이터는 CAN 메시지의 제 1 시간의 다음 시간인 제 2 시간에 대응되는 CAN 메시지의 데이터일 수 있다. 예를 들어, 프로세서(110)는 제 1 CAN 메시지에 기초하여 획득되는 제 1 실제 데이터와 제 1 예측 데이터를 비교하여 제 1 CAN 메시지의 이상 유무를 결정할 수 있다. 다른 예를 들어, 프로세서(110)는 제 2 CAN 메시지에 기초하여 획득되는 제 2 실제 데이터와 제 2 예측 데이터를 비교하여 제 2 CAN 메시지의 이상 유무를 결정할 수 있다.
- [0095] 프로세서(110)는 실제 데이터와 예측 데이터의 차이에 기초하여 사전 학습된 신경망 모델의 손실 값을 산출할 수 있다. 프로세서(110)는 평균 절대 오차 및/또는 평균 제곱 오차 중 적어도 하나를 이용하여 손실 값을 산출할 수 있다. 평균 절대 오차는 실제 데이터와 예측 데이터의 차이를 절대값으로 변환하여 평균화를 하는 방식일 수 있다. 평균 제곱 오차는 실제 데이터와 예측 데이터의 차이를 제곱하여 평균화를 하는 방식일 수 있다.
- [0096] 그리고, 프로세서(110)는 손실 값과 사전 설정된 임계 값을 비교하여 CAN 메시지의 이상 유무를 결정할 수 있다. 사전 설정된 임계 값은 반복적인 실험을 통해 산출된 값일 수 있다. 예를 들어, 프로세서(110)는 제 1 신경망 모델의 제 1 손실 값과 사전 설정된 제 1 임계 값을 비교하여 제 1 CAN 메시지의 이상 유무를 결정할 수 있다. 다른 예를 들어, 프로세서(110)는 제 2 신경망 모델의 손실 값과 사전 설정된 임계 값을 비교하여 제 2 CAN 메시지의 이상 유무를 결정할 수 있다.
- [0098] 제 2 신경망 모델(500)은 제 1 신경망 모델(400)과 대응되는 구성을 가질 수 있다. 따라서, 제 2 신경망 모델(500)의 구성에 관한 내용은 전술한 제 1 신경망 모델(400)의 구성에 관한 내용으로 대체될 수 있으며, 1 신경망 모델(400)과 다른 부분에 대해서만 후술한다.
- [0099] 제 2 신경망 모델(500)은 제 1 신경망 모델(400)의 학습에 사용되는 학습 데이터와 상이한 학습 데이터를 이용하여 사전 학습될 수 있다. 예를 들어, 제 2 신경망 모델(500)은 제 2 카테고리에 대응되는 CAN 메시지에 기초하여 생성된 학습 데이터를 이용하여 사전 학습될 수 있다. 제 2 신경망 모델(500)은 출력된 데이터와 실제 데이터 간의 값의 차이가 감소되도록 사전 학습될 수 있다.
- [0100] 프로세서(110)는 복수의 카테고리에 각각 대응되는 복수의 신경망 모델들을 생성하고, 각각 카테고리별로 학습시킬 수 있다. 예를 들어, 프로세서(110)는 복수의 카테고리가 2개인 경우, 제 1 카테고리에 대응되는 제 1 신경망 모델을 생성하고, 제 2 카테고리에 대응되는 제 2 신경망 모델을 생성할 수 있다. 그리고, 프로세서(110)는 제 1 신경망 모델을 제 1 카테고리에 대응되는 학습 데이터를 이용하여 사전 학습시키고, 제 2 신경망 모델을 제 2 카테고리에 대응되는 학습데이터를 이용하여 사전 학습시킬 수 있다.
- [0102] 프로세서(110)는 제 1 신경망 모델(400)의 제 1 히든 레이어로부터 획득되는 제 1 히든 상태의 값과 제 2 신경망 모델(500)의 제 2 히든 레이어로부터 획득되는 제 2 히든 상태의 값을 연결(concatenation)한 제 1 연결 값을 생성할 수 있다. 여기서, 제 1 히든 레이어와 제 2 히든 레이어는 대응될 수 있다. 프로세서(110)는 사전 학습된 제 3 신경망 모델(600)을 이용하여 제 1 연결 값으로부터 제 1 출력 값을 획득할 수 있다.
- [0103] 제 3 신경망 모델(600)은 서로 상이한 학습 데이터를 통해 CAN 메시지의 카테고리별로 사전 학습된 복수의 신경망 모델들의 서로 대응되는 히든 레이어에서 각각 획득되는 복수의 히든 상태들의 값을 연결한 연결 데이터를

입력 데이터로 이용하여 사전 학습될 수 있다. 제 3 신경망 모델(600)은 입력 데이터에 대응되는 출력 데이터를 출력하도록 사전 학습될 수 있다. 연결 데이터는 복수의 히든 상태들의 값을 연결한 연결 값을 포함할 수 있다. 출력 데이터는 입력 데이터인 복수의 히든 상태들의 값을 연결한 연결 값에 대응되어 출력되는 출력 값을 포함할 수 있다.

- [0104] 제 3 신경망 모델(600)은 도 2를 참조하여 상술한 바와 같이 다양한 신경망 모델을 포함할 수 있다. 예를 들어, 제 3 신경망 모델(600)은 오토 인코더를 포함할 수 있다. 제 3 신경망 모델(600)은 인코더부, 잠재 벡터 및 디코더부를 포함할 수 있다.
- [0105] 인코더부는 복수의 히든 레이어들을 포함할 수 있고, 연결 데이터가 입력될 수 있다. 인코더부는 입력된 연결 데이터의 차원을 축소시킨 잠재 벡터를 출력할 수 있다. 잠재 벡터는 연결 데이터가 인코더부를 통해 사전 결정된 차원(예를 들어, 1차원, 2차원 등)으로 축소된 벡터일 수 있다. 디코더부는 복수의 히든 레이어들을 포함할 수 있고, 잠재 벡터가 입력될 수 있다. 디코더부는 입력된 잠재 벡터의 차원을 확장하여 연결 데이터와 대응되는 차원의 출력 값을 출력할 수 있다.
- [0106] 프로세서(110)는 제 3 신경망 모델(600)에 입력한 제 1 연결 값과 제 3 신경망 모델(600)을 이용하여 획득된 제 1 출력 값을 비교하여 제 1 CAN 메시지 및 제 2 CAN 메시지를 포함하는 전체 CAN 메시지의 이상 유무를 결정할 수 있다.
- [0107] 예를 들어, 프로세서(110)는 제 1 연결 값과 제 1 출력 값의 차이에 기초하여 사전 학습된 제 3 신경망 모델(600)의 제 2 손실 값을 산출할 수 있다. 프로세서(110)는 평균 절대 오차 및/또는 평균 제곱 오차 중 적어도 하나를 이용하여 제 2 손실 값을 산출할 수 있다. 그리고, 프로세서(110)는 제 3 신경망 모델(600)의 제 2 손실 값과 사전 설정된 제 2 임계 값을 비교하여 전체 CAN 메시지의 이상 유무를 결정할 수 있다. 예를 들어, 프로세서(110)는 제 2 손실 값이 사전 설정된 제 2 임계 값보다 큰 경우 전체 CAN 메시지에 이상이 있는 것으로 결정할 수 있다.
- [0108] 본 개시의 몇몇 다른 실시예에 따르면, 제 1 신경망 모델(400)은 제 1 히든 레이어와 상이한 제 3 히든 레이어를 더 포함할 수 있다. 제 2 신경망 모델(500)은 제 3 히든 레이어와 대응되는 제 4 히든 레이어를 더 포함할 수 있다. 프로세서(110)는 제 2 CAN 메시지에 기초하여 획득되는 제 2 실제 데이터와 제 2 예측 데이터를 비교하여 제 2 CAN 메시지의 이상 유무를 결정하고 난 후, 제 3 히든 레이어로부터 획득되는 제 3 히든 상태의 값과 제 4 히든 레이어로부터 획득되는 제 4 히든 상태의 값을 연결한 제 2 연결 값을 생성할 수 있다.
- [0109] 프로세서(110)는 사전 학습된 제 3 신경망 모델(600)을 이용하여 제 2 연결 값으로부터 제 2 출력 값을 획득할 수 있다. 사전 학습된 제 3 신경망 모델(600)은 제 2 연결 값을 입력 받고, 제 2 출력 값을 출력할 수 있다. 프로세서(110)는 제 2 연결 값과 제 2 출력 값을 비교하여 제 1 CAN 메시지 및 제 2 CAN 메시지를 포함하는 전체 CAN 메시지의 이상 유무를 결정할 수 있다.
- [0111] 도 4는 본 개시의 몇 실시예에 따른 CAN 메시지의 구성을 설명하기 위한 도면이다.
- [0112] 도 4를 참조하면, CAN 메시지는 복수의 필드(field)들이 분할되어 연결된 형태일 수 있다. 복수의 필드들은 시작(start of frame, SOF) 필드, 중재(arbitration) 필드, 제어(control) 필드, 페이로드(payload) 필드, 순환 중복 검사(cyclic redundancy check, CRC) 필드, 긍정응답(acknowledgement, ACK) 필드, 프레임 엔드(end of frame, EOF) 필드 및/또는 인터미션(intermision, ITM) 필드 중 적어도 하나를 포함할 수 있다. 다만, 상술한 구성 요소들은 CAN 메시지를 구현하는데 있어서, 필수적인 것은 아니며, CAN 메시지는 위에 열거된 구성 요소들보다 많거나, 또는 적은 구성 요소들을 가질 수 있다.
- [0113] 시작 필드는 1비트로 구성되고, CAN 메시지의 시작을 표시할 수 있다. 따라서, 시작 필드는 메시지 전송이 시작되었음을 버스에 연결된 모든 노드에게 알림으로써, 버스에 연결된 모든 노드의 동기화를 시킬 수 있다.
- [0114] 중재 필드는 Arbitration ID 및 RTR(remote transmission request)을 포함할 수 있다.
- [0115] Arbitration ID는 11비트 또는 29비트로 구성되고, CAN 메시지의 식별자일 수 있다. Arbitration ID는 우선순위를 결정하기 위한 값일 수 있다. 예를 들어, CAN 메시지는 Arbitration ID의 값이 작을수록 우선순위를 가질 수 있다. 다만, CAN 메시지는 이에 한정되지 않으며, Arbitration ID의 값이 클수록 우선순위를 가질 수도 있다.
- [0116] RTR은 1비트로 구성되고, 동시에 전송된 데이터 프레임(data frame)인 제 1 CAN 메시지와 원격 프레임(remote frame)인 제 2 CAN 메시지가 서로 동일한 Arbitration ID를 가지는 경우, 우선순위를 결정하기 위한 값일 수 있다.

다. 데이터 프레임은 페이로드 필드를 포함하는 구조를 의미할 수 있다. 원격 프레임은 페이로드 필드를 포함하지 않는 구조를 의미할 수 있다. 예를 들어, RTR의 값이 '0'인 경우, 해당 CAN 메시지는 데이터 프레임으로 인식될 수 있다. RTR의 값이 '1'인 경우, 해당 CAN 메시지는 원격 프레임으로 인식될 수 있다.

- [0117] 따라서, 동시에 전송된 제 1 CAN 메시지와 제 2 CAN 메시지 중에서, RTR의 값이 '0'인 경우를 우선순위로 판단하여 제 1 CAN 메시지를 우선순위로 인식할 수 있다.
- [0118] 제어 필드는 IDE(identifier extension), Reserved 및 DLC(data length code)를 포함할 수 있다.
- [0119] IDE는 1비트로 구성되고, 해당 CAN 메시지의 종류인 표준 CAN 메시지와 확장 CAN 메시지를 구분하기 위한 값일 수 있다. 표준 CAN 메시지는 Arbitration ID가 11비트인 CAN 메시지일 수 있다. 확장 CAN 메시지는 Arbitration ID가 29비트인 메시지일 수 있다. 예를 들어, IDE의 값이 0인 경우, 표준 CAN 메시지로 판단할 수 있다. 그리고, IDE의 값이 1인 경우, 확장 CAN 메시지로 판단할 수 있다.
- [0120] Reserved는 2개의 1비트들(예를 들어, r0, r1)로 구성되고, 추후에 Arbitration ID를 확장(예를 들어, 29비트 등)하는 경우에 사용되는 값일 수 있다. 예를 들어, Reserved는 CAN이 아닌 다른 통신 네트워크에서 사용하는 경우, 다른 통신 네트워크에 대응되도록 Arbitration ID를 확장시키기 위해 사용될 수 있다.
- [0121] DLC(data length code)는 4비트로 구성되고, 페이로드 필드에 포함된 데이터의 길이를 나타내는 값일 수 있다.
- [0122] 페이로드 필드는 0 내지 64비트로 구성되고, 전송되는 실제 데이터의 값인 페이로드를 포함할 수 있다. 페이로드는 특정 노드에서 다른 노드로 전송하려는 정보를 포함할 수 있다. 여기서, 정보는 DBC(Database CAN) 파일에 기초하여 해석될 수 있다. DBC(Database CAN) 파일은 페이로드 필드별 의미와 기능에 대한 정보를 포함하는 데이터베이스 파일일 수 있다. 따라서, 프로세서(110)는 DBC 파일을 통해 페이로드 필드에 기재된 페이로드를 해석하여 특정 노드에서 다른 노드로 전송하려는 정보를 획득할 수 있다.
- [0123] 순환 중복 검사 필드는 순환 중복 검사 시퀀스(CRC sequence) 및 순환 중복 검사 구획 문자(CRC delimiter)를 포함할 수 있다.
- [0124] 순환 중복 검사 시퀀스는 15비트로 구성되고, 전송 노드에서 특정 알고리즘에 기초하여 산출된 값일 수 있다. 따라서, 수신 노드는 순환 중복 검사 시퀀스의 값을 상술한 특정 알고리즘에 기초하여 계산하여 CAN 메시지 안에 비트 에러가 존재하는지 여부를 판단할 수 있다.
- [0125] 순환 중복 검사 구획 문자는 1비트로 구성되고, 순환 중복 검사 필드의 끝을 표시할 수 있다.
- [0126] 긍정응답 필드는 긍정응답 슬롯(ACK slot) 및 긍정응답 구획 문자(ACK delimiter)를 포함할 수 있다.
- [0127] 긍정응답 슬롯은 1비트로 구성되고, CAN 메시지의 정상적인 수신이 완료되었는지 여부를 판단하기 위한 값일 수 있다. 예를 들어, 순환 중복 검사 시퀀스의 검사 결과, 정상이라고 판단된 경우, 긍정응답 슬롯은 '0'의 값을 포함할 수 있다. 그리고, 순환 중복 검사 시퀀스의 검사 결과, 비정상이라고 판단된 경우, 긍정응답 슬롯은 '1'의 값을 포함할 수 있다.
- [0128] 긍정응답 구획 문자는 1비트로 구성되고, 긍정응답 필드의 끝을 표시할 수 있다.
- [0129] 프레임 엔드 필드는 7비트로 구성되고, CAN 메시지의 프레임의 끝을 표시할 수 있다. 예를 들어, 프레임 엔드 필드는 7비트인 '1111111'로 구성될 수 있다.
- [0130] 인터미션 필드는 3비트로 구성되고, 노드 간에 CAN 메시지의 전송 및 수신을 원활하게 수행하기 위한 버퍼 영역(buffer area)일 수 있다.
- [0132] 도 5는 본 개시의 몇몇 실시예에 따른 DBC 파일의 구성을 설명하기 위한 블록 구성도이다.
- [0133] 도 5를 참조하면, DBC(Database CAN) 파일은 Arbitration ID 및 Arbitration ID에 대응되는 페이로드 필드별 의미와 기능에 대한 정보를 포함하는 데이터베이스 파일일 수 있다. DBC 파일은 신호 이름(signal name), 신호의 시작 비트(start bit), 사용하는 비트의 길이(bit length), 증감 단위(increment), 값의 범위(range) 및/또는 영향을 미치는 기능(affecting function)에 대한 정보 중 적어도 하나를 포함할 수 있다. 예를 들면, 신호 이름은 전방의 좌측 바퀴 속도일 수 있다. 신호의 시작 비트는 0일 수 있다. 사용하는 비트의 길이는 8일 수 있다. 증감 단위는 0.0625일 수 있다. 값의 범위는 0-255일 수 있다. 영향을 미치는 기능은 스마트 크루즈 컨트롤일 수 있다.

- [0135] 도 6은 본 개시의 몇몇 실시예에 따른 제 1 신경망 모델을 설명하기 위한 블록 구성도이다.
- [0136] 도 6을 참조하면, 제 1 신경망 모델(400)은 복수의 입력 레이어들(예를 들어, 제 1 입력 레이어(411), 제 3 입력 레이어(412), 제 N-1 입력 레이어(413), 제 N 입력 레이어(414) 등이며, 여기서 N은 자연수일 수 있다.) 중 적어도 하나, 복수의 히든 레이어들(예를 들어, 제 1 히든 레이어(421), 제 3 히든 레이어(422), 제 N-1 히든 레이어(423), 제 N 히든 레이어(424) 등이며, 여기서 N은 자연수일 수 있다.) 중 적어도 하나 및/또는 복수의 출력 레이어들(예를 들어, 제 1 출력 레이어(431), 제 3 출력 레이어(432), 제 N-1 출력 레이어(433), 제 N 출력 레이어(434) 등이며, 여기서 N은 자연수일 수 있다.) 중 적어도 하나를 포함할 수 있다.
- [0137] 복수의 입력 레이어들은 각각 입력 데이터가 직접 입력되는 최초 레이어를 의미할 수 있다. 예를 들어, 제 1 입력 레이어(411)는 CAN 메시지의 제 1 시간에 대응되는 CAN 메시지의 데이터가 입력될 수 있다. 다른 예를 들어, 제 3 입력 레이어(412)는 CAN 메시지의 제 2 시간에 대응되는 CAN 메시지의 데이터가 입력될 수 있다. 또 다른 예를 들어, 제 N 입력 레이어(414)는 CAN 메시지의 제 N 시간에 대응되는 CAN 메시지의 데이터가 입력될 수 있다.
- [0138] 복수의 히든 레이어들은 복수의 입력 레이어들 각각에 대응되어 구성되고, 복수의 입력 레이어들 각각으로부터 데이터를 입력 받는 레이어를 의미할 수 있다. 예를 들어, 제 1 히든 레이어(421)는 제 1 입력 레이어(411)에 대응되어 제 1 입력 레이어(411)로부터 데이터를 입력 받도록 구성될 수 있다. 다른 예를 들어, 제 3 히든 레이어(422)는 제 3 입력 레이어(412)에 대응되어 제 3 입력 레이어(412)로부터 데이터를 입력 받도록 구성될 수 있다. 또 다른 예를 들어, 제 N 히든 레이어(424)는 제 N 입력 레이어(414)에 대응되어 제 N 입력 레이어(414)로부터 데이터를 입력 받도록 구성될 수 있다.
- [0139] 복수의 출력 레이어들은 복수의 히든 레이어들 각각에 대응되어 구성되고, 복수의 히든 레이어들 각각으로부터 데이터를 입력 받는 최종 레이어를 의미할 수 있다. 예를 들어, 제 1 출력 레이어(431)는 제 1 히든 레이어(421)에 대응되어 제 1 히든 레이어(421)로부터 데이터를 입력 받아 제 1 출력 데이터를 출력할 수 있다. 다른 예를 들어, 제 3 출력 레이어(432)는 제 3 히든 레이어(422)에 대응되어 제 3 히든 레이어(422)로부터 데이터를 입력 받아 제 3 출력 데이터를 출력할 수 있다. 또 다른 예를 들어, 제 N 출력 레이어(434)는 제 N 히든 레이어(424)에 대응되어 제 N 히든 레이어(424)로부터 데이터를 입력 받아 제 N 출력 데이터를 출력할 수 있다.
- [0140] 제 2 신경망 모델(500)은 제 1 신경망 모델(400)에 대응되는 구성을 포함할 수 있다. 예를 들어, 제 2 신경망 모델(500)은 제 1 신경망 모델(400)의 제 1 입력 레이어(411)와 대응되는 제 2 입력 레이어, 제 1 히든 레이어(421)와 대응되는 제 2 히든 레이어, 및 제 1 출력 레이어(431)와 대응되는 제 2 출력 레이어를 포함할 수 있다. 다른 예를 들어, 제 2 신경망 모델(500)은 제 1 신경망 모델(400)의 제 3 입력 레이어(412)와 대응되는 제 4 입력 레이어, 제 3 히든 레이어(422)와 대응되는 제 4 히든 레이어 및 제 3 출력 레이어(432)와 대응되는 제 4 출력 레이어를 포함할 수 있다.
- [0141] 전술한 바와 같이, 제 2 신경망 모델(500)은 제 1 신경망 모델(400)에 대응되는 구성을 포함할 수 있다. 따라서, 제 2 신경망 모델(500)의 구성에 관한 설명은 전술한 제 1 신경망 모델(400)의 구성에 관한 설명으로 대체될 수 있다.
- [0143] 도 7은 본 개시의 몇몇 실시예에 따른 제 3 신경망 모델을 설명하기 위한 블록 구성도이다.
- [0144] 도 7을 참조하면, 제 3 신경망 모델(600)은 인코더부(610), 잠재 벡터(620) 및 디코더부(630)를 포함할 수 있다.
- [0145] 인코더부(610)는 복수의 히든 레이어들을 포함할 수 있고, 연결 데이터가 입력될 수 있다. 예를 들어, 인코더부(610)는 제 1 신경망 모델(400)의 제 1 히든 레이어(421)로부터 획득되는 제 1 히든 상태의 값과 제 2 신경망 모델(500)의 제 2 히든 레이어로부터 획득되는 제 2 히든 상태의 값을 연결한 제 1 연결 값이 입력될 수 있다. 다른 예를 들어, 인코더부(610)는 제 1 신경망 모델(400)의 제 3 히든 레이어(422)로부터 획득되는 제 3 히든 상태의 값과 제 2 신경망 모델(500)의 제 4 히든 레이어로부터 획득되는 제 4 히든 상태의 값을 연결한 제 2 연결 값이 입력될 수 있다. 인코더부(610)는 입력된 연결 데이터(예를 들어, 제 1 연결 값, 제 2 연결 값 등)의 차원을 축소시킨 잠재 벡터(620)를 출력할 수 있다.
- [0146] 잠재 벡터(620)는 연결 데이터가 인코더부(610)를 통해 사전 결정된 차원(예를 들어, 1차원, 2차원 등)으로 축소된 벡터일 수 있다.
- [0147] 디코더부(630)는 복수의 히든 레이어들을 포함할 수 있고, 잠재 벡터(620)가 입력될 수 있다. 디코더부(630)는

입력된 잠재 벡터(620)의 차원을 확장하여 연결 데이터와 대응되는 차원의 출력 데이터(예를 들어, 제 1 출력 값, 제 2 출력 값 등)을 출력할 수 있다.

- [0149] 도 8은 본 개시의 몇 실시예에 따른 컴퓨팅 장치에 의해 수행되는 DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 방법을 설명하기 위한 순서도이다.
- [0150] 도 8을 참조하면, 컴퓨팅 장치(100)의 프로세서(110)는 CAN(Controller Area Network)에서 발생하는 제 1 CAN 메시지를 획득할 수 있다(S110).
- [0151] 프로세서(110)는 사전 저장된 DBC 파일에 기초하여 제 1 CAN 메시지를 복수의 카테고리들 중에서 제 1 카테고리로 결정할 수 있다(S120).
- [0152] 프로세서(110)는 사전 학습된 제 1 신경망 모델을 이용하여 제 1 CAN 메시지로부터 제 1 예측 데이터를 획득할 수 있다(S130). 사전 학습된 제 1 신경망 모델은 제 1 카테고리에 대응되고 그리고 제 1 히든 레이어(hidden layer)를 포함할 수 있다.
- [0153] 프로세서(110)는 제 1 CAN 메시지에 기초하여 획득되는 제 1 실제 데이터와 상기 제 1 예측 데이터를 비교하여 제 1 CAN 메시지의 이상 유무를 결정할 수 있다(S140). 예를 들어, 프로세서(110)는 제 1 실제 데이터와 제 1 예측 데이터의 차이에 기초하여 사전 학습된 제 1 신경망 모델의 제 1 손실(loss) 값을 산출할 수 있다. 프로세서(110)는 제 1 손실 값과 사전 설정된 제 1 임계 값을 비교하여 제 1 CAN 메시지의 이상 유무를 결정할 수 있다.
- [0155] 프로세서(110)는 CAN에서 발생하는 제 2 CAN 메시지를 획득할 수 있다.
- [0156] 프로세서(110)는 사전 저장된 DBC 파일에 기초하여 제 2 CAN 메시지를 상기 복수의 카테고리들 중에서 제 2 카테고리로 결정할 수 있다.
- [0157] 프로세서(110)는 사전 학습된 제 2 신경망 모델을 이용하여 상기 제 2 CAN 메시지로부터 제 2 예측 데이터를 획득할 수 있다. 사전 학습된 제 2 신경망 모델은 제 2 카테고리에 대응되고 그리고 제 1 신경망 모델의 제 1 히든 레이어와 대응되는 제 2 히든 레이어를 포함할 수 있다.
- [0158] 프로세서(110)는 제 2 CAN 메시지에 기초하여 획득되는 제 2 실제 데이터와 제 2 예측 데이터를 비교하여 제 2 CAN 메시지의 이상 유무를 결정할 수 있다.
- [0160] 프로세서(110)는 제 1 히든 레이어로부터 획득되는 제 1 히든 상태(hidden state)의 값과 제 2 히든 레이어로부터 획득되는 제 2 히든 상태의 값을 연결(concatenation)한 제 1 연결 값을 생성할 수 있다.
- [0161] 프로세서(110)는 사전 학습된 제 3 신경망 모델을 이용하여, 제 1 연결 값으로부터 제 1 출력 값을 획득할 수 있다.
- [0162] 프로세서(110)는 제 1 연결 값과 제 1 출력 값을 비교하여 제 1 CAN 메시지 및 제 2 CAN 메시지를 포함하는 전체 CAN 메시지의 이상 유무를 결정할 수 있다. 예를 들어, 프로세서(110)는 상기 제 1 연결 값과 상기 제 1 출력 값의 차이에 기초하여 사전 학습된 제 3 신경망 모델의 제 2 손실 값을 산출할 수 있다. 프로세서(110)는 제 2 손실 값과 사전 설정된 제 2 임계 값을 비교하여 전체 CAN 메시지의 이상 유무를 결정할 수 있다.
- [0164] 제 1 신경망 모델은 상기 제 1 히든 레이어와 상이한 제 3 히든 레이어를 더 포함할 수 있다. 제 2 신경망 모델은 상기 제 3 히든 레이어와 대응되는 제 4 히든 레이어를 더 포함할 수 있다.
- [0165] 프로세서(110)는 제 2 CAN 메시지에 기초하여 획득되는 제 2 실제 데이터와 제 2 예측 데이터를 비교하여 제 2 CAN 메시지의 이상 유무를 결정된 이후, 제 3 히든 레이어로부터 획득되는 제 3 히든 상태의 값과 제 4 히든 레이어로부터 획득되는 제 4 히든 상태의 값을 연결(concatenation)한 제 2 연결 값을 생성할 수 있다.
- [0166] 프로세서(110)는 사전 학습된 제 3 신경망 모델을 이용하여, 제 2 연결 값으로부터 제 2 출력 값을 획득할 수 있다.
- [0167] 프로세서(110)는 제 2 연결 값과 제 2 출력 값을 비교하여 제 1 CAN 메시지 및 제 2 CAN 메시지를 포함하는 전체 CAN 메시지의 이상 유무를 결정할 수 있다.
- [0168] 사전 학습된 제 1 신경망 모델은 제 1 카테고리에 대응되는 CAN 메시지에 기초하여 생성된 학습 데이터를 이용하고, 그리고 출력된 데이터와 실제 데이터 간의 값의 차이가 감소되도록 사전 학습될 수 있다.

- [0169] 사전 학습된 제 2 신경망 모델은 제 2 카테고리에 대응되는 CAN 메시지에 기초하여 생성된 학습 데이터를 이용하고, 그리고 출력된 데이터와 실제 데이터 간의 값의 차이가 감소되도록 사전 학습될 수 있다.
- [0170] 사전 학습된 제 3 신경망 모델은 서로 상이한 학습 데이터를 통해 CAN 메시지의 카테고리별로 사전 학습된 복수의 신경망 모델들의 서로 대응되는 히든 레이어에서 각각 획득되는 복수의 히든 상태들의 값을 연결한 연결 데이터를 입력 데이터로 이용하고, 그리고 입력 데이터에 대응되는 출력 데이터를 출력하도록 사전 학습될 수 있다.
- [0171] 도 8에 도시되는 단계들은 예시적인 단계들이다. 따라서, 본 개시내용의 사상의 범위를 벗어나지 않는 한도에서도 8의 단계들 중 일부가 생략되거나 추가적인 단계들이 존재할 수 있다는 점 또한 당업자에게 명백할 것이다.
- [0172] 또한, 도 8에 기재된 구성들(100 내지 600)에 관한 구체적인 내용은 앞서 도 1 내지 도 7을 통해 설명한 내용으로 대체될 수 있다.
- [0173] 도 1 내지 도 8을 참조하여 상술한 바와 같이, 본 개시의 몇몇 실시예에 따른 DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 컴퓨팅 장치(100)는 DBC 파일을 이용하여 CAN 메시지를 기능을 기준으로 분류하고, 해당 기능별로 나누어진 CAN 메시지를 이용하여 복수의 신경망 모델들(예를 들어, 제 1 신경망 모델, 제 2 신경망 모델) 각각을 학습시킬 수 있다. 예를 들어, 제 1 신경망 모델은 CLU 관련 CAN 메시지를 이용하여 학습될 수 있다. 제 2 신경망 모델은 SCC 관련 CAN 메시지를 이용하여 학습될 수 있다. 따라서, 본 개시의 몇몇 실시예에 따른 컴퓨팅 장치(100)는 기능별로 침입(즉, 공격)을 탐지하여 마이크로(micro) 공격에 대한 탐지를 수행할 수 있다.
- [0174] 본 개시의 몇몇 실시예에 따른 DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 컴퓨팅 장치(100)는 각 기능별로 나누어진 CAN 메시지를 이용하여 각각 학습된 복수의 신경망 모델들(예를 들어, 제 1 신경망 모델, 제 2 신경망 모델)로부터 각각 획득되는 히든 상태의 값을 연결한 연결 값을 제 3 신경망 모델에 입력하여 출력 값을 획득할 수 있다. 컴퓨팅 장치(100)는 연결 값과 출력 값을 비교하여 전체 CAN 메시지의 이상 유무를 결정할 수 있다. 따라서, 본 개시의 몇몇 실시예에 따른 컴퓨팅 장치(100)는 전체 CAN 메시지를 이용하여 전체적인 침입(즉, 공격)을 탐지하여 매크로(macro) 공격에 대한 탐지를 수행할 수 있다.
- [0175] 본 개시의 몇몇 실시예에 따른 DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 컴퓨팅 장치(100)는 네트워크 환경을 고려하여 필요한 주요 기능부에 해당되는 신경망 모델만을 탑재하여 실시간으로 운용이 가능할 수 있다.
- [0176] 본 개시의 몇몇 실시예에 따른 DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 컴퓨팅 장치(100)는 DBC 파일에 기반하여 각 기능별로 정리 및 학습하기 때문에 이상 값을 탐지할 수 있는 높은 정확도와 각 기능별 상호 연관성을 학습하여 높은 탐지율을 가질 수 있다.
- [0177] 또한, 본 개시의 몇몇 실시예에 따른 DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 컴퓨팅 장치(100)는 외부 연구자와 공격자보다 제조사에서 더 정확한 DBC 파일을 가지고 있으므로, 외부에서 IDS 제작 후 탐지 못하는 공격을 생성하여도 실제 제조사의 IDS는 더 정교하여 탐지가 가능할 수 있다.
- [0178] 또한, 본 개시의 몇몇 실시예에 따른 DBC 파일을 이용하여 시퀀스 기반의 침입을 탐지하기 위한 컴퓨팅 장치(100)는 DBC 파일에 기반하여 각 기능별로 정리 및 학습하기 때문에 공격의 타겟이 되는 위치를 확인할 수 있다.
- [0180] 도 9는 본 개시의 실시예들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 간략하고 일반적인 개략도를 도시한다.
- [0181] 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.
- [0182] 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.

- [0183] 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.
- [0184] 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.
- [0185] 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.
- [0186] 컴퓨터(1102)를 포함하는 본 개시의 여러가지 측면들을 구현하는 예시적인 환경(1100)이 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.
- [0187] 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.
- [0188] 컴퓨터(1102)는 또한 내장형 하드 디스크 드라이브(HDD)(1114)(예를 들어, EIDE, SATA)-이 내장형 하드 디스크 드라이브(1114)는 또한 적당한 새시(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.
- [0189] 이들 드라이브 및 그와 연관된 컴퓨터 판독가능 매체는 데이터, 데이터 구조, 컴퓨터 실행가능 명령어, 기타 등의 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 zip 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.
- [0190] 운영 체제(1130), 하나 이상의 애플리케이션 프로그램(1132), 기타 프로그램 모듈(1134) 및 프로그램 데이터를

(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.

[0191] 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.

[0192] 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.

[0193] 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.

[0194] 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. 네트워킹된 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.

[0195] 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 임의의 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 단순히 적어도 2개의 장치 사이의 애드혹 통신(ad hoc communication)일 수 있다.

[0196] 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.

[0197] 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.

[0198] 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 방식으로 설명된 기능을 구현할 수 있으나, 이러한 구현 결정들은 본 개시의 범위를 벗어나는 것으로 해석되어서는 안 될 것이다.

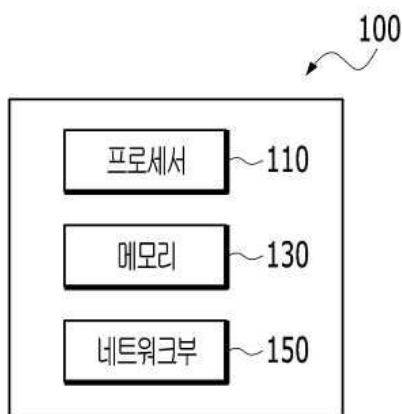
[0199] 여기서 제시된 다양한 실시예들은 방법, 장치, 또는 표준 프로그래밍 및/또는 엔지니어링 기술을 사용한 제조 물품(article)으로 구현될 수 있다. 용어 제조 물품은 임의의 컴퓨터-관독가능 저장장치로부터 액세스 가능한 컴퓨터 프로그램, 캐리어, 또는 매체(media)를 포함한다. 예를 들어, 컴퓨터-관독가능 저장매체는 자기 저장 장치(예를 들면, 하드 디스크, 플로피 디스크, 자기 스트립, 등), 광학 디스크(예를 들면, CD, DVD, 등), 스마트 카드, 및 플래쉬 메모리 장치(예를 들면, EEPROM, 카드, 스틱, 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-관독가능한 매체를 포함한다.

[0200] 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.

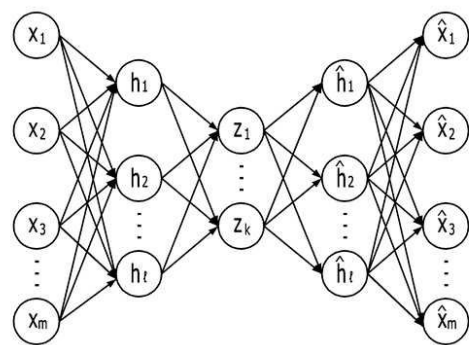
[0201] 제시된 실시예들에 대한 설명은 임의의 본 개시의 기술 분야에서 통상의 지식을 가진 자가 본 개시를 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 개시의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이며, 여기에 정의된 일반적인 원리들은 본 개시의 범위를 벗어나지 않고 다른 실시예들에 적용될 수 있다. 그리하여, 본 개시는 여기에 제시된 실시예들로 한정되는 것이 아니라, 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의의 범위에서 해석되어야 할 것이다.

도면

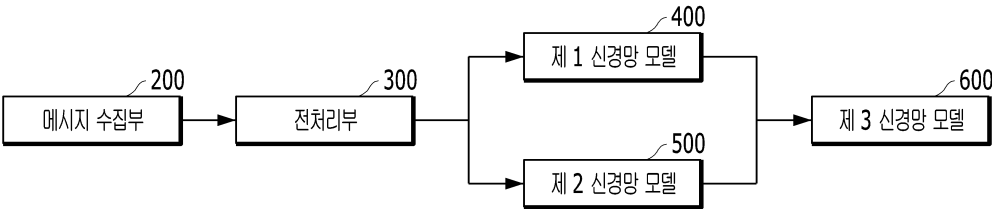
도면1



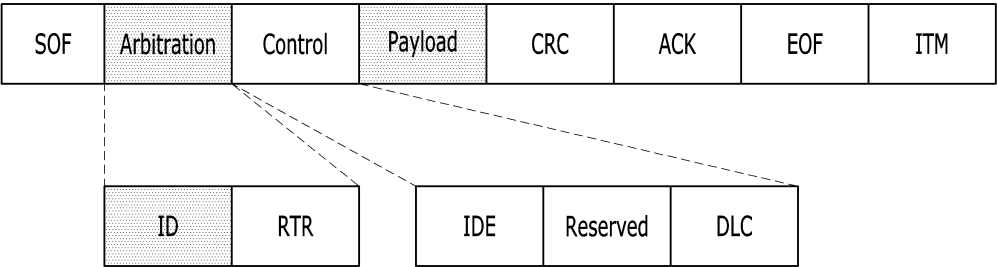
도면2



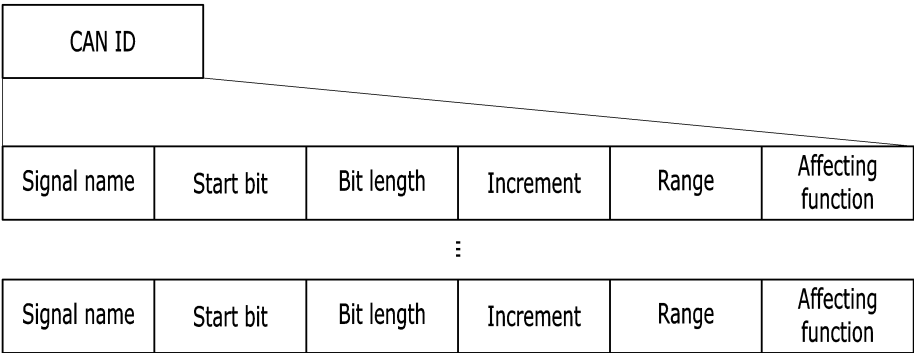
도면3



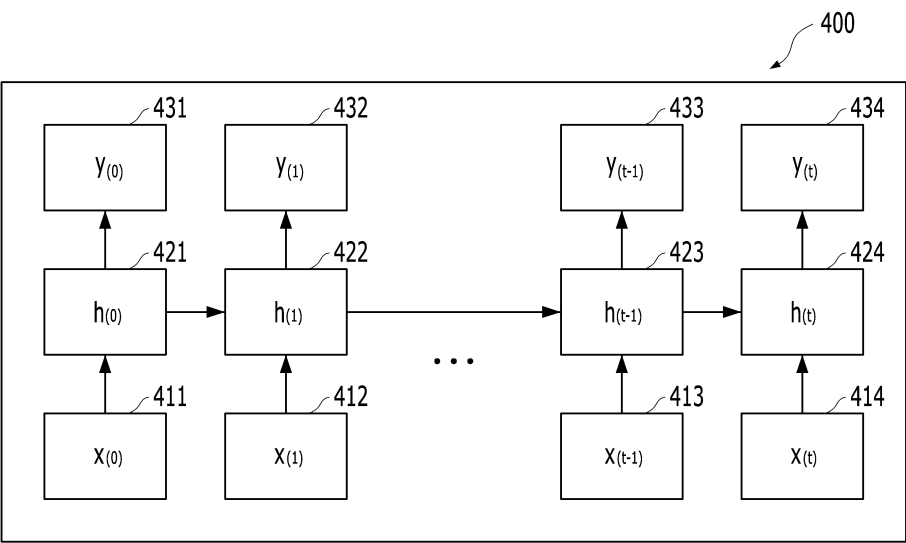
도면4



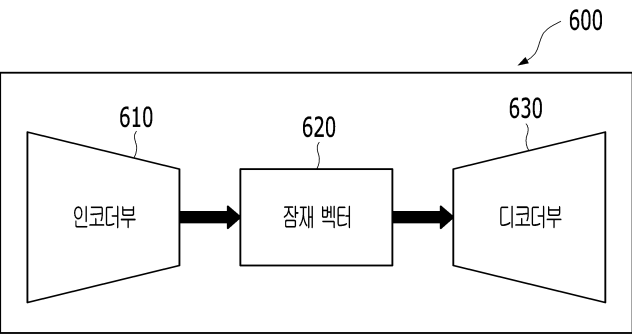
도면5



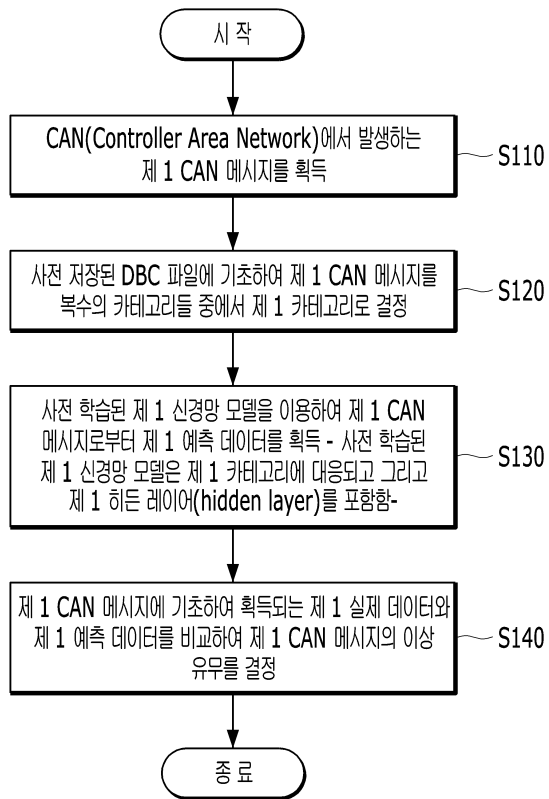
도면6



도면7



도면8



도면9

