정 보 보 호 학 회 지
제 32 권 제4호, 2022. 8

# AI를 이용한 차량용 침입 탐지 시스템에 대한 평가 프레임워크

김 형 훈*, 정 연 선**, 최 원 석***, 조 효 진*

요 약

운전자 보조 시스템을 통한 차량의 전자적인 제어를 위하여, 최근 차량에 탑재된 전자 제어 장치 (ECU; Electronic Control Unit)의 개수가 급증하고 있다. ECU는 효율적인 통신을 위해서 차량용 내부 네트워크인 CAN(Controller Area Network)을 이용한다. 하지만 CAN은 기밀성, 무결성, 접근 제어, 인증과 같은 보안 메커니즘이 고려되지 않은 상태로 설계되었기 때문에, 공격자가 네트워크에 쉽게 접근하여 메시지를 도청하거나 주입할 수 있다. 악의적인 메시지 주입은 차량 운전자 및 동승자의 안전에 심각한 피해를 안길 수 있기에, 최근에는 주입된 메시지를 식별하기 위한 침입 탐지 시스템 (IDS; Intrusion Detection System)에 대한 연구가 발전해왔다. 특히 최근에는 AI(Artificial Intelligence) 기술을 이용한 IDS가 다수 제안되었다. 그러나 제안되는 기법들은 특정 공격 데이터셋에 한하여 평가되며, 각 기법에 대한 탐지 성능이 공정하게 평가되었는지를 확인하기 위한 평가 프레임워크가 부족한 상황이다. 따라서 본 논문에서는 machine learning/deep learning에 기반하여 제안된 차량용 IDS 5가지를 선정하고, 기존에 공개된 데이터셋을 이용하여 제안된 기법들에 대한 비교 및 평가를 진행한다. 공격 데이터셋에는 CAN의 대표적인 4가지 공격 유형이 포함되어 있으며, 추가적으로 본 논문에서는 메시지 주기 유형을 활용한 공격 유형을 제안하고 해당 공격에 대한 탐지 성능을 평가한다.

## Ⅰ. 서 론

최근 자동차 산업계에서는 운전자에게 편리함을 주기 위하여 차선 유지 보조 시스템, 긴급 제동, 차간 거리 유지 등의 운전자 보조 시스템을 탑재한 차량을 판매한다. 이러한 운전 보조 시스템을 제공하기 위해서는 차량을 전자적으로 제어하기 위한 ECU (Electronic Control Units)의 개수가 증가하고 있으며, 최근 고급 차량의 경우 약 70개의 ECU가 탑재되어 있는 것으로 알려져 있다. 수많은 ECU가 효율적으로 통신하기 위해서는 1993년 ISO 11898 표준으로 정의된 차량용 내부 네트워크 중 한 종류인 CAN (Controller Area Network)을 사용한다[1]. 하지만, CAN은 1986년 bosch 사에 의해 개발될 때 보안 메커니즘이 고려되지 않았다. 따라서 기밀성, 무결성, 인증 및 접근 제어에 대한 통제가 부족하기 때문에 악의적인 공격자가 네트워크에 접근하여 CAN 메시지를 도청하거나 주입할 수 있다. 대표적인 공격 사례로는 2015년 C.Miller 등의 공격으로 Jeep Cherokee 차량에 대해 디스플레이, 브레이크, 핸들 등이 원격으로 제어되었다[2]. 가장 최근에는 Tencent의 Keen security lab에서 테슬라 모델의 autopilot 기능을 포함한 여러 기능의 취약점을 발견하고 차량을 제어하였다[3,4].

이러한 악의적인 패킷 주입을 탐지하기 위해 다양한 보안 솔루션이 개발되고 있으며, 그중 CAN 메시지 기반 차량용 침입 탐지 시스템인 Automotive IDS (Intrusion Detection System)에 대한 연구가 활발히 진행되고 있다[5]. 특히 최근에는 machine learning과 deep learning 알고리즘을 활용한 IDS가 다수 제안되고 있다. 하지만, 제안되는 기법들은 특정 공격 데이터셋에 한하여 평가되고 있으며, 공개된 데이터셋을 이용하지 않고 자체 제작한 공격 데이터셋을 이용하는 경우도 많

본 연구는 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-00111, AI 기술을 활용한 자율주행 자동차 사이버 공격 및 방어 기술 연구)
* 숭실대학교 소프트웨어학부 (대학원생, axolotl0210@gmail.com, 조교수, hyojin.jo@ssu.ac.kr)
** 고려대학교 정보보호대학원 (대학원생, ys_jeong@korea.ac.kr)
*** 한성대학교 IT융합공학부 (조교수, wonsuk@hansung.ac.kr)

다. 따라서 제안되는 기법들에 대하여 공정하게 비교 평가할 수 있는 평가 프레임워크가 필요하며, 본 논문에서는 공개된 데이터셋에 기반한 IDS 평가 프레임워크에 대해 제안한다.

본 논문의 기여도는 다음과 같다.

- machine learning 및 deep learning에 기반한 5가지의 Automotive IDS를 선정하고, 해당 IDS에서 제안한 모델들을 직접 구축하여 공격에 대한 탐지율을 검증한다. 또한, 각 IDS에 대해 binary classification과 multi-class classification을 수행한다.
- 공개된 데이터셋을 이용한 평가 프레임워크를 제안하였으며, 기존 4가지의 공격뿐만 아니라 CAN 메시지 주기 유형을 활용한 공격에 대해 평가함으로써 기존에 연구되었던 Automotive IDS에 대한 한계점을 보여준다.

본 논문의 구성은 다음과 같다. 2장에서는 배경지식 및 관련 연구에 대해 설명하고, 3장에서는 CAN 공격 시나리오 및 선정된 machine learning과 deep learning 기반의 Automotive IDS에 대해 평가 및 비교하는 평가 프레임워크에 대해 소개한다. 4장에서는 공개된 데이터셋과 추가적으로 제작한 공격 데이터셋을 이용하여 3장에서 소개된 IDS에 대하여 평가한다. 마지막으로 5장에서는 결론을 맺는다.

## Ⅱ. 배경지식 및 관련 연구

### 2.1. CAN 프로토콜

CAN 프로토콜은 1986년 Robert Bosch 사에 의해 개발되었으며[6], bus 형태의 통신 구조로 ECU 간의 효율적인 데이터 송·수신을 지원한다. CAN bus는 두 개의 꼬임선 구조를 이용하는데 이는 CAN-H(high)와 CAN-L(low)로 정의되며, 두 선의 전압 차이를 이용하여 비트 0과 1을 표현할 수 있다. CAN-H와 CAN-L가 각 3.5V와 1.5V의 전압을 가지면 전압의 차이는 2.0V 이상이며 이는 비트 0을 나타낸다. 반면, CAN-H와 CAN-L 모두 2.5V 전압을 가지면 전압의 차이는 없게 되고, 이는 비트 1을 나타낸다. 전압차를 이용한 비트 통신은 CAN bus가 잡음에 견고한 특성을 가지게 한다. 또한, CAN은 동시에 전송되는 CAN 데이터 프레임에 대해 arbitration로 정의된 과정을 통해 메시지의 우선순위를 할당하며, 낮은 arbitration 필드를 가지는 메시지가 더 높은 우선순위를 가진다.

| SOF | ID | RTR | IDE | R | DLC | Data | CRC | ACK | EOF |
|---|---|---|---|---|---|---|---|---|---|
| 1 bit | 11 bits | 1 bit | 1 bit | 1 bit | 4 bits | 0-8 bytes | 16 bits | 2 bits | 7 bits |

〔그림 1〕 CAN data frame의 구조

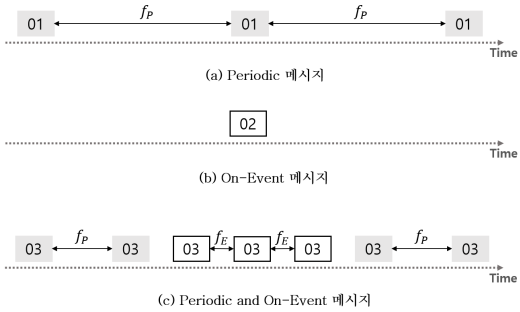
### 2.2. CAN 데이터 프레임

CAN 2.0 프로토콜은 CAN ID의 길이에 따라 11bits ID를 가지는 CAN 2.0 A, 29bits의 확장된 ID를 가지는 CAN 2.0 B로 나누어 정의된다. CAN data frame의 형태는 그림 1과 같으며, 각 필드에 대한 설명은 다음과 같다.

- SOF(Start Of Frame): CAN 프레임의 시작을 나타냄
- ID(Identifier): CAN 메시지 우선순위를 정하기 위한 arbitration 필드로 사용됨
- RTR(Remote Transmission Request): 4개의 CAN 프레임 종류 (data frame, remote frame, error frame, overload frame) 중, 어떤 프레임을 사용하는지 나타냄
- IDE(Identifier Extension): CAN 2.0 A/B에 대한 정보를 나타냄
- R(Reserved): R은 예약 비트로 CAN 2.0 B을 위해 사용됨
- DLC(Data Length Code): 데이터 필드의 가변 길이를 나타냄
- Data: 전송하고자 하는 실제 데이터 값을 가지고 있으며, 최대 64bits(8bytes)를 가짐
- CRC(Cyclic Redundancy Check): CAN 데이터 프레임에 대한 순환 중복 검사 값을 계산함
- ACK(Acknowledgement): 데이터 프레임의 전송 성공 여부를 나타냄
- EOF(End Of Frame): CAN 프레임의 마지막을 나타냄

### 2.3. CAN 메시지 전송 유형

ECU에서 전송되는 CAN 메시지 유형은 메시지의 주기성에 따라 3가지로 분류된다. 그림 2의 (a)는 항상




〔그림 2〕 CAN 메시지 전송 유형

동일한 시간 차이를 갖고 주기적으로 전송되는 주기 메시지(i.e., P(Periodic) 메시지)를 보여준다. 그림 2의 (b)는 차량에서 특정 기능을 활성화했을 시에 이벤트성으로 나타나는 비주기적 메시지(i.e., E(On-Event) 메시지)를 의미한다. 마지막으로, 주기 메시지와 비주기적 메시지가 혼합된 PE(Periodic and On-Event) 메시지가 존재한다. 이는 그림 2의 (c)와 같이 주기적으로 전송되던 메시지가 특정 이벤트 발생 시에 추가적으로 메시지가 전송되다 다시 주기적으로 전송되는 특징을 가진다. 일반적으로 이벤트가 발생하는 경우, 연속된 3개의 이벤트 메시지가 전송되며, 이러한 패턴은 차량 제조업체마다 조금씩 차이가 있는 것으로 알려져 있다[7].

### 2.4. 기존 Automotive IDS 평가 프레임워크

본 단락에서는 기존 제안된 Automotive IDS에 대한 평가 프레임워크에 대해 소개한다. 기존의 평가 프레임워크들은 machine learning 및 deep learning에 기반한 IDS에 중점을 맞춘 것이 아닌 규칙(e.g., ID Sequence, Timing Interval, Hamming Distance, etc.)에 기반한 IDS들을 위주로 평가하고 있다.

P. Agbaje et al.[8]은 침입 탐지 시스템을 위한 알고리즘 종류를 크게 Timing, Statistical, ML (Machine Learning)으로 나누고, 세부적인 15개의 알고리즘에 대해 평가한다. 모든 알고리즘에 대한 일관적인 평가를 위해 DoS(Denial-of-Service), Fuzzy, Gear Spoofing, RPM Spoofing과 같은 4개의 종류의 공격이 포함된 Car-hacking 데이터셋[1]을 사용한다. 알고리즘에 따라 각 공격에 대한 탐지 정확도의 차이를 관찰하였으며, 결과를 알고리즘의 특성에 따라 분석하였다. 최종적으로, ML과 Statistical 종류 중 하나인 Graph-based 모델과 같이 메시지 간의 관계를 고려하는 알고리즘이 CAN bus 상에서 적합하다고 판단하였다. 해당 연구에서는 새로운 침입 탐지 시스템에 대한 알고리즘이 개발될 때, Car-hacking 데이터셋을 이용하여 성능을 평가한다면 기존 평가된 알고리즘들과의 성능 비교를 수행할 수 있다고 서술하였다.

D. Stabili et al.[9]은 Message Sequence, Bus Entropy, Hamming Distance, Missing Message 총 4개의 알고리즘에 대한 평가를 진행한다. 평가를 위한 데이터셋은 Replay, Fuzzing, Disruption으로 총 3가지 공격에 대해 Volvo 차량에서 수집하였다. 해당 데이터셋에 대한 평가 수행 결과, Message Sequence 알고리즘이 모든 공격에 대해 일관된 성능을 가지고 탐지하고 있다고 평가하였다.

## Ⅲ. 평가 프레임워크

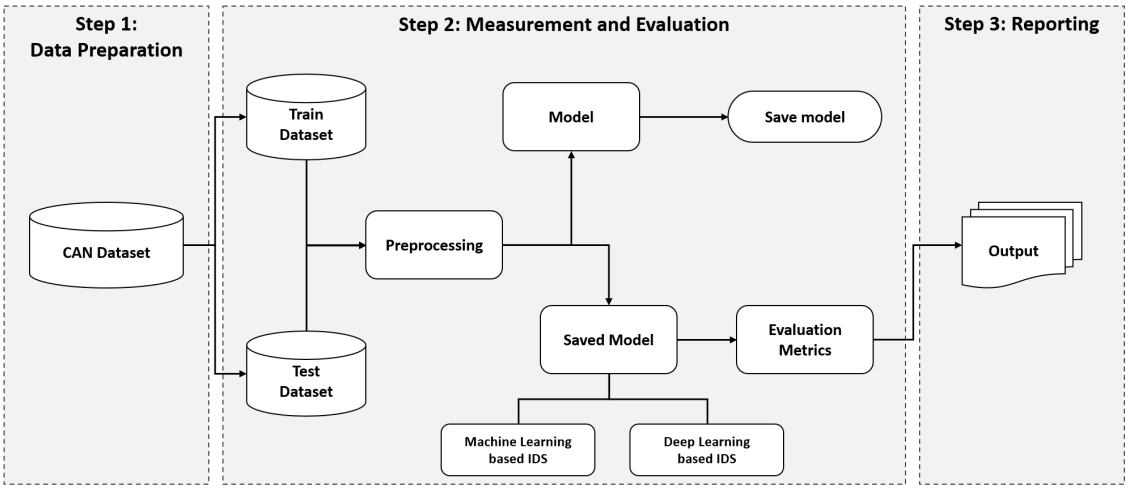
이번 장에서는 본 논문에서 제안하는 Automotive IDS에 대한 평가 프레임워크에 대해 설명하도록 한다. 평가 프레임워크에 대해 알아보기 전에 공격 시나리오를 먼저 설명하도록 한다.

### 3.1. 공격 시나리오

CAN 프로토콜에 대해서 기존에 알려진 공격 기법은 Flooding, Spoofing, Replay, Fuzzing으로 총 4가지로 구성되어 있다. 본 논문에서는 PE 공격을 추가하였으며, 각 공격 기법은 아래와 같다.

- Flooding 공격: 차량에서 사용되는 CAN ID보다 더 높은 우선순위인 CAN 메시지(e.g., CAN ID가 0x000인 메시지)를 대량으로 주입하는 공격 방법.
- Spoofing 공격: 차량의 특정 기능을 오작동 시키기 위해, 해당 기능을 제어하는 CAN 메시지를 파악한 후 생성하여 주입하는 공격 방법.
- Replay 공격: 차량에서 정상적으로 통신하고 있는 CAN 트래픽을 수집한 후, 그대로 다시 주입하는 공격 방법.
- Fuzzing 공격: 무작위 값으로 생성한 CAN 메시지를 차량에 주입하는 공격 방법이며, 예상치 못한 차

1) https://ocslab.hksecurity.net/Datasets/car-hacking-dataset




〔그림 3〕 Automotive IDS에 대한 평가 프레임워크 시스템 구성도

량의 이상 작동 현상이 발생할 수 있음.

- PE 공격: 해당 공격은 Spoofing 공격의 일종으로, 주기와 비주기 메시지가 혼합된 CAN 메시지를 이용하여 차량에 주입하는 공격 방법.

### 3.2. 시스템 구성도

Automotive IDS에 대한 평가 프레임워크의 시스템 구성도는 그림 3과 같으며, Data Preparation, Measurement and Evaluation, Reporting으로 총 3가지 단계로 설계되어 있다. 첫 번째 단계는 Data Preparation으로, IDS를 평가할 때 사용할 데이터를 수집하고 평가 형식에 맞게 데이터셋을 변환한다. 두 번째 단계인 Measurement and Evaluation은 첫 번째 단계에서 가공된 데이터셋을 이용하여 모델을 학습시키고 성능을 평가하며, train과 test 과정으로 구분된다. train 과정은 IDS를 구축할 때 사용될 모델을 선정한 후, 각 모델에 맞게 전처리 과정을 진행하며 학습이 완료된 모델은 저장하게 된다. test 과정은 train 과정에서 사용되지 않은 test 데이터셋을 이용하고 평가 지표를 통해 학습된 모델의 성능을 평가한다. 마지막 세 번째 단계인 Reporting은 두 번째 단계에서 평가 지표를 이용하여 모델의 성능을 평가한 결과를 도출하게 된다.

### 3.3. 1 단계: Data Preparation

Automotive IDS를 평가하기 위해서 데이터를 수집하고 데이터셋으로 변환하는 과정은 필수적인 요소이며, 차량의 OBD-II 포트를 통해 CAN 트래픽을 송·수신할 수 있다. 데이터는 공격 시나리오가 반영되지 않은 정상 데이터와 공격 시나리오가 반영된 공격 데이터에 대해 수집하며, CAN 메시지마다 정상인지 공격인지 레이블 정보가 기입되어 있어야 한다. 추가적으로, 평가가 진행될 때 편의성을 위해, 수집한 데이터를 지정한 평가 형식에 맞게 데이터셋으로 변환하는 과정을 진행한다.

만약 차량을 소유할 수 없는 환경이라면, 공개된 데이터셋을 활용하도록 한다. 최근 차량으로부터 CAN 트래픽을 직접 수집하고 데이터셋으로 변환하여 공개하는 연구들이 많이 진행되고 있으며, 공격 시나리오가 반영된 데이터셋도 포함되어 있다[10,11].

### 3.4. 2 단계: Measurement and Evaluation

이번 단계에서는 Automotive IDS에 사용될 모델을 학습시키고 성능을 평가하며, train과 test 과정으로 이루어져 있다. Data Preparation 단계에서 수집 및 가공한 데이터셋을 train과 test 과정에 맞게 데이터셋을 구성한다. 구성된 데이터셋을 이용하여 모델 학습 및 성능을 평가하기 위해, 본 논문에서는 5가지의 IDS 모델(i.e., 1개의 machine learning 기반 IDS와 4개의 deep learning 기반 IDS)과 분류 평가 지표를 선정하였다. 이번 절에서는 선정한 AI 모델과 평가 지표에 대해서 설명하도록 한다.

#### 3.4.1. AI 기반 Automotive IDS

##### 3.4.1.1. Machine Learning 기반 IDS

O. Minawi et al.[12]은 machine learning을 이용한 Automotive IDS를 제안하였으며, 3가지의 계층으로 구성되어 있다. 첫 번째는 CAN Message Input Layer이며, CAN bus로부터 CAN 메시지를 수신하고, feature를 생성한다. feature를 생성하기 위해, CAN data frame에서 hexadecimal인 ID와 Data를 decimal로 변환하는 전처리 과정을 수행하였다. 두 번째는 Threat Detection Layer로, 이전 단계에서 생성한 feature와 machine learning 알고리즘을 이용하여 수신한 메시지가 정상인지 공격인지 분류하는 과정을 수행한다. 해당 연구는 4가지의 machine learning 알고리즘(i.e., Random Tree(RT), Random Forest(RF), Stochastic Gradient Descent(SGD), Naive Bayes (NB))을 사용하였다. 세 번째는 Alert Layer이며, 수신한 메시지가 공격으로 판단될 경우, 경고 알람을 생성하여 운전자에게 알려주는 작업을 수행한다.

##### 3.4.1.2. GAN 기반 IDS

E. Seo et al.[13]은 anomaly detection 분야에서 주로 사용되는 deep learning 알고리즘 중 하나인 GAN(Generative Adversarial Network)[14]을 이용하여 GIDS로 불리는 Automotive IDS를 제안하였다. GIDS는 2가지의 탐지 모델이 존재하며, 첫 번째 모델은 기존에 알려진 공격을 탐지하는 discriminator이고, 두 번째는 알려지지 않은 새로운 공격을 탐지하기 위한 모델로 구성되어 있다. 해당 연구에서는 두 모델에 사용될 input을 생성하기 위해 CAN data frame의 ID를 이용하였다. ID마다 one hot vector encoding 방식을 적용한 후, 2D grid 형식으로 표현함으로써 image로 변환하는 전처리 과정을 수행한다. 첫 번째 모델에서는 수집한 데이터셋에 대해 전처리 과정을 수행하여 real image를 생성한 후, 해당 image의 spatial correlation을 파악하도록 학습하였다. 두 번째 탐지 모델은 무작위로 생성한 noise를 real image와 매우 비슷한 fake image로 생성할 수 있도록 generator를 이용하며, real image와 fake image를 구분할 수 있도록 discriminator를 학습시킨다. 최종적으로 학습된 두 개의 discriminator에 대해 특정 임계값을 조절하면서 공격 메시지를 탐지하게 된다.

##### 3.4.1.3. DCNN 기반 IDS

H. M. Song et al.[15]은 이미지 분류에 주로 사용되는 deep learning 알고리즘 중 하나인 DCNN (Deep Convolutional Neural Network)[16]을 이용하여 Automotive IDS를 제안하였다. 해당 연구에서도 모델의 input을 생성하기 위해, CAN data frame에서 ID를 사용하였다. Frame Builder로 불리는 과정을 통해서 input을 생성하게 되는데, 해당 연구에서는 29개의 CAN 메시지를 기준으로 hexadecimal인 ID를 29bits 크기의 binary로 변환함으로써 29 × 29 형태인 2D grid frame을 생성하였다. 모델의 경우, Inception-ResNet 모델[17]의 아키텍처를 수정 및 제거함으로써 input의 크기를 줄이고 모델의 파라미터 수를 줄인 경량화된 Reduced Inception-ResNet이라는 모델을 생성하였다, 해당 모델은 레이블이 지정된 데이터셋에 대해 Frame Builder로 2D grid frame을 생성한 후 해당 frame의 spatial correlation 특징을 학습하며, 정상 또는 공격 메시지를 분류하게 된다.

##### 3.4.1.4. LSTM 기반 IDS

M. D. Hossain et al.[18]은 시계열 데이터 예측에 주로 사용되는 deep learning 알고리즘 중 하나인 LSTM(Long Short-Term Memory)[19]을 이용하여 Automotive IDS를 제안하였으며, 2가지의 시스템으로 구성되어 있다. 첫 번째는 Attack Verification System으로, 공격 생성 알고리즘을 통해 공격 데이터셋을 생성한다. 두 번째는 IDS이며, CAN bus로부터 CAN 메시지를 송·수신하며, 해당 메시지가 공격으로 판단될 경우, 경고 알람을 생성하게 된다. IDS의 학습을 위해, CAN data frame에서 ID, DLC, Data를 추출하고 Data를 byte 단위로 나눔으로써 총 10개의 feature를 생성하였다. hexadecimal인 ID와 Data는 decimal로 변환하고, DLC가 8bytes 보다 작은 경우에는 Data를 -1 값으로 패딩하였다. 해당 연구에서는 2가지의 LSTM 모델을 이용하여 CAN 메시지를 분류하였으며, 여러 가지 hyper-parameter를 설정함으로써 최적의 탐지 모델을 생성하였다.

3.4.1.5. MLP 기반 IDS

F. Amato et al.[20]은 deep learning 알고리즘인 NN(Neural Network)와 MLP(Multi-Layer Perceptron)을 이용하여 Automotive IDS를 제안하였으며, 3가지 단계로 구성되어 있다. 첫 번째 단계는 Descriptive Statistics로, CAN data frame의 Data를 hexadecimal인 값에서 decimal로 변환한 후, 정상과 공격 메시지 간의 분포 차이를 계산함으로써 공격 메시지의 특성을 파악하였다. 두 번째 단계는 Hypotheses Testing이며, 정상과 공격 메시지 간의 분포에 대해 가설 검정 기법(i.e., Mann-Whitney, Kolmogorov-Smirnov, Wald-Wolfowitz)을 이용하여 가설의 합당성을 검증한다. 세 번째 단계는 Classification Analysis로, NN과 여러 개의 MLP 모델을 생성하고 학습한 후, 각 모델이 공격 메시지를 얼마나 효율적으로 탐지하는지 평가함으로써 최적의 모델을 도출하게 된다.

3.4.2. 평가 지표

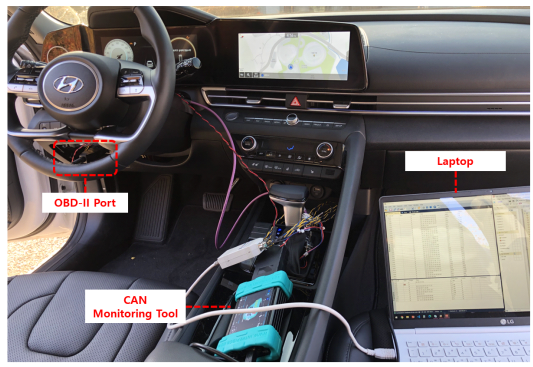
앞서 설명한 Automotive IDS들의 성능을 평가하기 위해 정확도 (Accuracy), 정밀도 (Precision), 재현율 (Recall), F1-score을 측정하였다. 정확도는 전체 메시지에서 정상 메시지인지 공격 메시지인지 정확하게 예측한 비율을 의미한다. 정밀도는 공격으로 예측한 메시지에서 실제 공격 메시지의 비율을 의미한다. 재현율은 실제 공격 메시지에서 정확하게 공격 메시지로 예측한 비율을 의미한다. F1-score는 정밀도와 재현율 간의 조화 평균을 의미하며, 해당 지표들을 계산하는 수식은 다음과 같다.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$




〔그림 4〕 Avante CN7 차량 데이터 수집 환경

여기서 TP (True Positive)는 공격 메시지를 공격 메시지로, TN (True Negative)는 정상 메시지를 정상 메시지로 분류한 것을 의미한다. FP (False Positive)는 정상 메시지를 공격 메시지로, FN (False Negative)는 공격 메시지를 정상 메시지로 분류한 것을 의미한다.

### 3.5. 3 단계: Reporting

마지막으로 이번 단계에서는 Measurement and Evaluation 단계에서 설명한 Automotive IDS들에 대해 평가 지표를 이용하여 각 모델의 성능을 평가한 결과를 도출하게 된다. 도출된 결과를 통해, 각 모델 간의 성능 비교를 함으로써 최적의 IDS를 파악할 수 있다.

## Ⅳ. 실험 및 결과

이번 장에서는 공개된 데이터셋과 PE 공격 데이터셋을 이용하여 5가지의 AI 기반 Automotive IDS에 대해 탐지 성능을 평가한다.

### 4.1. 실험 환경

본 논문에서 제안한 Automotive IDS에 대한 평가 프레임워크를 Intel(R) Core(TM) i7-10750H CPU @ 2.60Hz, NVIDIA GeForce GTX 1650 Ti, RAM 16GB, 운영 체제 Window 10 Pro으로 구성된 PC에서 실험을 진행하였다. machine learning과 deep learning 기반 IDS를 개발하기 위해, Python 3.8을 사용하였고, Scikit-learn, Tensorflow[21], Keras 등의 라이브러리를 사용하였다.

### 4.2. 데이터셋에 대한 설명

우리는 Hacking and Countermeasure Research Lab (HCRL)에서 제공하는 “Car Hacking: Attack & Defense Challenge 2020” 데이터셋[11]을 사용하였다. 해당 데이터셋은 Hyundai Avante CN7 차량에서 제작되었으며, 3.1절에서 설명한 것처럼 4가지 공격 상황(Flooding, Spoofing, Replay, Fuzzing 공격)을 포함하고 있다. 특히 해당 데이터셋은 preliminary와 final round로 구성되어 있는데, 그중 preliminary의 training을 훈련 데이터셋으로 사용하였고, submission을 테스트 데이터셋으로 사용하였다.

추가적으로 PE 공격에 대한 평가를 진행하기 위해 데이터셋을 제작하였다. 제작된 데이터셋에 대한 정보는 표 1과 같다. PE 공격이 수집된 환경 그림 4와 같으며, 이는 Car Hacking 데이터셋이 수집된 차량 환경과 동일한 환경이다. 따라서 Car Hacking 데이터셋으로 학습된 모델을 이용하여 PE 공격에 대해서 탐지한다.

〔표 1〕 PE 공격 데이터셋

| Dataset | The number of CAN messages | | |
|---|---|---|---|
| | Normal | Attack | Total |
| PE Attack 1 | 168,314 | 249 | 168,563 |
| PE Attack 2 | 201,670 | 477 | 202,147 |
| PE Attack 3 | 236,841 | 258 | 237,099 |

### 4.3. 실험 결과

Automotive IDS에 대한 정확한 평가를 위하여, 각 모델에 대해 binary와 multi-class classification을 수행하도록 2가지 모델을 구축하여 실험을 진행하였다. binary classification은 normal 또는 attack에 대하여 2가지 분류를 진행하는 반면, multi-class classification은 5가지 class(normal과 4개의 공격 시나리오)에 대한 분류를 진행한다. 또한, 평가되는 IDS들의 소스 코드는 공개되어 있지 않기 때문에 저자들이 직접 구현하고 기존 모델들이 평가할 때 사용한 데이터셋과 파라미터 값을 이용하여 구축한 모델에 대한 성능을 검증하였다. 특히 machine learning[12]을 구현할 때는 Random Tree 대신 Decision Tree (DT)를 사용하였다.

#### 4.3.1. 전체적인 Automotive IDS 성능 비교

표 2는 선별된 5가지 Automotive IDS에 대한 평가 결과이다. Machine learning[12]에는 4개(DT, RF, SGD, NB)의 알고리즘이 포함되어 있기 때문에 이를 포함한 총 8개의 machine learning 및 deep learning 모델에 대한 평가 결과를 보여준다. 전체적인 모델 binary 와 multi-class classificaiton 모두에서 Random Forest 모델이 각 0.99, 0.96의 F1-score로 가장 높은 탐지율을 보여주었다. 반면, Naive Bayes 알고리즘이 binary classification에서 약 0.66, multi-class

〔표 2〕 Car Hacking 데이터셋에 대한 각 모델 탐지 결과

| | Model | | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|
| Binary Classification | Machine Learning | DT[12] | 0.9689 | 0.9278 | 0.7629 | 0.8373 |
| | | RF[12] | 0.9983 | 0.9938 | 0.9909 | 0.9924 |
| | | SGD[12] | 0.9473 | 0.9929 | 0.501 | 0.666 |
| | | NB[12] | 0.9464 | 0.985 | 0.4668 | 0.6605 |
| | Deep Learning | GAN[13] | 0.757 | 0.7773 | 0.756 | 0.7665 |
| | | DCNN[15] | 0.7433 | 0.7156 | 0.7629 | 0.7385 |
| | | LSTM[18] | 0.9712 | 0.9976 | 0.7269 | 0.841 |
| | | MLP[20] | 0.9494 | 0.7681 | 0.7419 | 0.7548 |
| Multi-class Classification | Machine Learning | DT[12] | 0.9685 | 0.9542 | 0.9685 | 0.959 |
| | | RF[12] | 0.9759 | 0.9587 | 0.9729 | 0.9632 |
| | | SGD[12] | 0.9549 | 0.9267 | 0.9549 | 0.9402 |
| | | NB[12] | 0.8195 | 0.9254 | 0.8195 | 0.8673 |
| | Deep Learning | GAN[13] | 0.7116 | 0.7033 | 0.7166 | 0.6102 |
| | | DCNN[15] | 0.6409 | 0.6176 | 0.6409 | 0.6209 |
| | | LSTM[18] | 0.9718 | 0.9593 | 0.9718 | 0.9588 |
| | | MLP[20] | 0.9488 | 0.9413 | 0.9488 | 0.933 |

classification에 대해서는 약 0.61의 F1-score로 낮은 탐지율을 보여주었다.

#### 4.3.2. 공격 시나리오 별 Automotive IDS 평가

표 3은 공격 시나리오 별로 Automotive IDS들의 탐지 성능에 대해 측정한 결과를 보여준다. Flooding 공격의 경우, MLP[20]를 제외한 대부분의 모델들이 F1-score가 약 0.99 이상으로 좋은 탐지 성능을 보여준다, 하지만, Spoofing 공격에 대해서는 모든 IDS가 거의 탐지하지 못하는 것을 볼 수 있다. 탐지한 결과를 확인해보았을 때, 해당 공격을 Normal이나 Replay 및 Fuzzing 공격으로 오탐하였다. Spoofing 공격은 특정 기능을 제어하기 위해서 기존에 흐르던 정상 메시지보다 더 많이 공격 메시지를 주입해야 한다. 각 CAN ID의 주기가 영향을 받는 것을 고려하였을 때, 우리는 Spoofing 공격이 Replay나 Fuzzing 공격과 비슷한 특성을 가지기 때문에 위와 같은 현상이 발생된다고 간주하였다. 또한, 대부분의 IDS들이 Replay 공격도 탐지하기 어려운 것으로 보이며, Fuzzing 공격의 경우, DCNN[15]이 가장 약한 탐지율을 보여주었다.

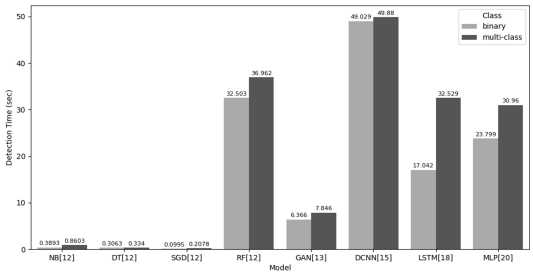
〔표 4〕 PE 공격에 대한 모델 별 탐지 결과

| | PE Attack | | |
|---|---|---|---|
| | Precision | Recall | F1-score |
| DT〔12〕 | 0 | 0 | 0 |
| RF〔12〕 | 0 | 0 | 0 |
| SGD〔12〕 | 0 | 0 | 0 |
| NB〔12〕 | 0 | 0 | 0 |
| GAN〔13〕 | 0.2539 | 0.478 | 0.3316 |
| DCNN〔15〕 | 0.0461 | 0.7679 | 0.087 |
| LSTM〔18〕 | 0 | 0 | 0 |
| MLP〔20〕 | 0 | 0 | 0 |

PE 공격에 대한 모델들의 탐지 결과는 표 4와 같으며, 표 1에 표기된 3개의 데이터셋에 대한 결과를 평균 낸 것이다. GAN[13]과 DCNN[15]을 제외한 대부분의 모델들은 PE 공격을 탐지 못하는 것을 볼 수 있다.

〔표 3〕 Car Hacking 데이터셋의 공격 시나리오에 대한 모델 별 탐지 결과

| | DT〔12〕 | | | RF〔12〕 | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| Normal | 0.9727 | 0.9927 | 0.9826 | 0.9729 | 0.9976 | 0.9851 |
| Flooding | 1 | 1 | 1 | 1 | 1 | 1 |
| Spoofing | 0 | 0 | 0 | 0 | 0 | 0 |
| Replay | 0.6477 | 0.193 | 0.2974 | 0.7201 | 0.193 | 0.3044 |
| Fuzzing | 0.8416 | 0.9951 | 0.912 | 0.9668 | 0.9993 | 0.9828 |
| | SGD〔12〕 | | | NB〔12〕 | | |
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| Normal | 0.9576 | 0.9934 | 0.9751 | 0.9563 | 0.8397 | 0.8942 |
| Flooding | 1 | 1 | 1 | 1 | 1 | 1 |
| Spoofing | 0 | 0 | 0 | 0.0248 | 0.3144 | 0.0459 |
| Replay | 0 | 0 | 0 | 0 | 0 | 0 |
| Fuzzing | 0.701 | 0.5524 | 0.6179 | 0.7014 | 0.5174 | 0.5955 |
| | GAN〔13〕 | | | DCNN〔15〕 | | |
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| Normal | 0.6244 | 0.9991 | 0.7685 | 0.7569 | 0.7341 | 0.7453 |
| Flooding | 1 | 0.9998 | 0.9999 | 1 | 0.9997 | 0.9999 |
| Spoofing | 0.9846 | 0.0069 | 0.0137 | 0.0592 | 0.0027 | 0.0051 |
| Replay | 0 | 0 | 0 | 0.2529 | 0.3504 | 0.2938 |
| Fuzzing | 0.9538 | 0.8277 | 0.8863 | 0.3725 | 0.6696 | 0.4787 |
| | LSTM〔18〕 | | | MLP〔20〕 | | |
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| Normal | 0.9696 | 0.9999 | 0.9845 | 0.9699 | 0.9739 | 0.9719 |
| Flooding | 1 | 1 | 1 | 0.6892 | 1 | 0.816 |
| Spoofing | 0 | 0 | 0 | 0 | 0 | 0 |
| Replay | 0.8745 | 0.0318 | 0.0614 | 0.6585 | 0.0676 | 0.1226 |
| Fuzzing | 0.9972 | 0.9808 | 0.989 | 0.9924 | 0.9811 | 0.9867 |




〔그림 5〕 모델 별 탐지 소요 시간

#### 4.3.3. Automotive IDS 탐지 시간

다음으로 각 Automotive IDS가 test 데이터셋에 대해 탐지하는 시간을 측정하였다. 그림 5는 총 8개의 모델에 대해서 binary와 multi-class classification을 수행할 때 측정한 탐지 시간을 나타낸다. 결과적으로 Random Forest 모델을 제외한 machine learning 기반 모델들이 가장 빠른 탐지 시간을 가진다. DCNN[15]의 경우 binary와 multi-class classification에서 모두 가장 많은 소요 시간을 보여주고 있다. LSTM[18]은 binary와 multi-class classification 간의 탐지 시간 차이가 큰 것을 볼 수 있다.

## Ⅴ. 결 론

운전자에게 편리한 주행 및 보조 기능을 제공하는 CAV(Connected and Automated Vehicle)의 거듭된 발전으로 인해, 차량의 안전을 보호하기 위한 차량 보안에 대한 인식도 더욱 중요해지고 있다. 그러나 차량 내부 네트워크인 CAN에는 무결성 및 인증 등과 같은 보안 기법이 적용되어 있지 않기 때문에, 해당 취약점을 이용하여 공격자가 가속 또는 긴급 제어 등 차량의 안전에 중요한 기능을 제어하도록 악의적인 메시지를 주입할 수 있다. 이로 인해, 차량 보안에 대한 연구가 이루어지고 있으며, 특히 Automotive IDS에 대한 연구가 활발히 진행되고 있다. 본 논문에서는 기존에 연구된 5가지의 AI 기반 IDS를 선정한 후, 해당 IDS에 대해 성능을 평가 및 비교하는 평가 프레임워크를 제안하였다. 추가적으로 CAN 메시지 주기 유형을 활용한 PE 공격을 제안하고 해당 공격에 대한 IDS 모델의 성능을 평가하였다.

향후 연구로는 향후 연구로는 Automotive IDS의 공정한 평가를 위한 추가적인 공격 종류를 포함한 데이터셋 공개하고, 본 논문에서 제안한 PE 공격을 높은 정확도로 탐지하기 위한 IDS 연구가 필요하다.

## 참 고 문 헌

[1] ISO, ISO. “11898-1: 2003-Road vehicles–Controller area network.” *International Organization for Standardization*, Geneva, Switzerland, 2003.

[2] Miller, Charlie, and Chris Valasek. “Remote exploitation of an unaltered passenger vehicle.” *Black Hat USA 2015*, no. S 91, 2015.

[3] Nie, Sen, Ling Liu, and Yuefeng Du. “Free-fall: Hacking tesla from wireless to can bus.” *Briefing, Black Hat USA*, 25, p.1-16, 2017.

[4] Nie, Sen, et al., “Over-the-air: How we remotely compromised the gateway, BCM, and autopilot ECUs of Tesla cars.”, *Briefing, Black Hat USA*, 2018.

[5] Kim, Kyounggon, et al., “Cybersecurity for autonomous vehicles: Review of attacks and defense.”, *Computers & Security*, 103, p.102150, 2021.

[6] Bosch, R. G., “CAN specification version 2.0: Robert Bosch GmbH.”, *Systems und Products for Car Manufacturer*, 1991.

[7] Lee, Seyoung, and Wonsuk Choi., “Periodic-and-on-Event Message-Aware Automotive Intrusion Detection System.”, *Journal of the Korea Institute of Information Security & Cryptology*, 31(3), pp.373-385, 2021.

[8] Agbaje, Paul, et al., “A Framework for Consistent and Repeatable Controller Area Network IDS Evaluation.”

[9] Stabili, Dario, Francesco Pollicino, and Alessio Rota., “A Benchmark Framework for CAN IDS.”, In *ITASEC*, pp. 233-245, 2021.

[10] Verma, Miki E., et al., “ROAD: the real ORNL automotive dynamometer controller area network intrusion detection dataset (with a comprehensive CAN IDS dataset survey & guide).”, *arXiv preprint arXiv:2012.14600*, 2020.

[11] Kang, Hyunjae, et al., “Car hacking and defense

competition on in-vehicle network." In *Workshop on automotive and autonomous vehicle security (AutoSec),* vol, p. 25, 2021.

[12] Minawi, Omar, et al., "Machine learning-based intrusion detection system for controller area networks.", In *Proceedings of the 10th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, pp. 41-47, 2020.

[13] Seo, Eunbi, Hyun Min Song, and Huy Kang Kim., "GIDS: GAN based intrusion detection system for in-vehicle network.", In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pp. 1-6. IEEE, Aug 2018.

[14] Goodfellow, Ian, et al., "Generative adversarial nets.", *Advances in neural information processing systems*, 27, 2014.

[15] Song, Hyun Min, Jiyoung Woo, and Huy Kang Kim., "In-vehicle network intrusion detection using deep convolutional neural network.", *Vehicular Communications*, 21, p.100198, 2020.

[16] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton., "Imagenet classification with deep convolutional neural networks.", *Advances in neural information processing systems*, 25, 2012.

[17] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun., "Deep residual learning for image recognition.", In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016.

[18] Hossain, Md Delwar, et al., "LSTM-based intrusion detection system for in-vehicle can bus communications.", *IEEE Access*, 8, pp.185489-185502, 2020.

[19] Hochreiter, Sepp, and Jürgen Schmidhuber., "Long short-term memory.", *Neural computation*, 9(8), pp.1735-1780, 1997.

[20] Amato, Flora, et al., "CAN-bus attack detection with deep learning.", *IEEE Transactions on Intelligent Transportation Systems*, 22(8), pp.5081-5090, Jan 2021.

[21] Abadi, Martín, et al., "{TensorFlow}: a system for {Large-Scale} machine learning.", *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016.

## 〈저자소개〉

**김 형 훈 (Hyunghoon Kim)**

학생회원

2019년 8월 : 한림대학교 컴퓨터공학과 졸업

2021년 8월 : 숭실대학교 융합소프트웨어학과 석사

2021년 9월~현재 : 숭실대학교 소프트웨어학과 박사 과정

<관심분야> 자동차 보안, IoT/CPS 보안, 암호학

**정 연 선 (Yeonseon Jeong)**

학생회원

2021년 2월 : 한림대학교 융합소프트웨어학과 졸업

2021년 3월~현재 : 고려대학교 정보보호대학원 정보보호학과 석사 과정

<관심분야> 자동차 보안, IoT 보안, 무선 네트워크 보안

**최 원 석 (Wonsuk Choi)**

종신회원

2008년 2월 : 서울시립대학교 수학과 졸업

2013년 2월 : 고려대학교 정보보호대학원 정보보호학과 석사

2018년 8월 : 고려대학교 정보보호대학원 정보보호학과 박사

2018년 9월~2020년 2월 : 고려대학교 정보보호연구원 연구교수

2020년 3월~현재 : 한성대학교 IT융합공학부 조교수

<관심분야> 센서 보안, 자동차 보안, 암호 프로토콜

**조 효 진 (Hyo Jin jo)**

종신회원

2009년 2월 : 고려대학교 산업공학과 졸업

2016년 2월 : 고려대학교 정보보호대학원 정보보호학과 박사

2016년 6월~2018 8월 : University of Pennsylvania 박사 후 연구원

2018년 9월~2020년 8월 : 한림대학교 소프트웨어융합대학 조교수

2020년 9월~현재 : 숭실대학교 소프트웨어학과 조교수

<관심분야> 자동차 보안, IoT 보안, 프라이버시