The Transactions of the Korea Information Processing Society
TKIPS Vol.13, No.9, pp.395~403
eISSN: 3022-7011
https://doi.org/10.3745/TKIPS.2024.13.9.395

Detection of Abnormal CAN Messages Using Periodicity and Time Series Analysis

Se-Rin Kim† · Ji-Hyun Sung†† · Beom-Heon Youn† · Harksu Cho†††

ABSTRACT

Recently, with the advancement of technology, the automotive industry has seen an increase in network connectivity. CAN (Controller Area Network) bus technology enables fast and efficient data communication between various electronic devices and systems within a vehicle, providing a platform that integrates and manages a wide range of functions, from core systems to auxiliary features. However, this increased connectivity raises concerns about network security, as external attackers could potentially gain access to the automotive network, taking control of the vehicle or stealing personal information. This paper analyzed abnormal messages occurring in CAN and confirmed that message occurrence periodicity, frequency, and data changes are important factors in the detection of abnormal messages. Through DBC decoding, the specific meanings of CAN messages were interpreted. Based on this, a model for classifying abnormalities was proposed using the GRU model to analyze the periodicity and trend of message occurrences by measuring the difference (residual) between the predicted and actual messages occurring within a certain period as an abnormality metric. Additionally, for multi-class classification of attack techniques on abnormal messages, a Random Forest model was introduced as a multi-classifier using message occurrence frequency, periodicity, and residuals, achieving improved performance. This model achieved a high accuracy of over 99% in detecting abnormal messages and demonstrated superior performance compared to other existing models.

Keywords : CAN, GRU, Anomaly Detection, Time Series, Machine Learning

CAN 메시지의 주기성과 시계열 분석을 활용한 비정상 탐지 방법

김 세 린† · 성 지 현†† · 윤 범 헌† · 조 학 수†††

요 약

최근 자동차 산업의 기술 발전과 함께 네트워크 연결성이 증대되고 있다. CAN(Controller Area Network) 버스 기술은 차량 내 다양한 전자 기기와 시스템 간의 신속하고 효율적인 데이터 통신을 가능하게 하여, 핵심 시스템부터 다양한 기능을 통합 관리할 수 있는 플랫폼을 제공한다. 그러나 이러한 연결성 증가는 외부 공격자가 자동차 네트워크에 접근하여 차량 제어를 장악하거나, 개인 정보를 탈취하는 등 네트워크 보안 우려를 초래할 수 있다. 본 논문은 CAN에서 발생하는 비정상 메시지를 분석하여, 메시지 발생 주기성 또는 빈도와 데이터 변화량이 비정상 메시지의 탐지에 중요한 요소임을 확인하였다. DBC 디코딩을 통해 CAN 메시지의 구체적인 의미를 해석하였다. 이를 바탕으로 메시지 발생의 주기성과 추이 분석을 위해 GRU 모델을 활용하여 일정 주기 이내에 발생한 메시지에 대해 예측 메시지와 발생한 메시지의 차이(잔차)를 비정상 측도로 이용한 비정상 분류 모델을 제안하고 비정상 메시지의 공격 기법에 대한 다중 분류에는 메시지와 발생 주기, 잔차를 이용한 랜덤 포레스트 모델을 도입하여 다중 분류기로 활용하여 성능 향상을 이루었다. 이 모델은 비정상 메시지 탐지에서 99% 이상의 높은 정확도를 달성하며 기존의 다른 모델보다 우수한 성능을 보여주었다.

키워드 : CAN, GRU, 비정상 탐지, 시계열, 머신러닝

1. 서 론

최근 몇 년간 자동차 산업은 급격한 기술적 발전을 이루며 지능형 자동차의 성장이 두드러지고 있다. 이와 함께 차량 내 다양한 시스템과 전자 장치들이 서로 긴밀하게 통신하고 협력하는 방식도 한층 정교해졌다. 특히, 차량을 전자적으로 제어하는 데 필수적인 ECU(Electronic Control Unit)의 수가 꾸준

※ 본 연구는 과학기술정보통신부와 정보통신기획평가원의 SW중심대사업의 연구결과로 수행되었음(2019-0-01834).
※ 이 논문은 2024년 ACK 2024의 우수논문으로 “휴리스틱과 XGBoost를 활용한 비정상 CAN 메시지 탐지”의 제목으로 발표된 논문을 확장한 것임.
† 비 회 원 : 호서대학교 컴퓨터공학부 학사과정
†† 준 회 원 : 호서대학교 컴퓨터공학부 학사과정
††† 정 회 원 : 호서대학교 컴퓨터공학부 교수
Manuscript Received : July 9, 2024
First Revision : August 16, 2024
Accepted : August 28, 2024
* Corresponding Author : Harksu Cho(marius1406@gmail.com)

히 증가하고 있다. ECU는 차량의 특정 기능을 제어하는 전자 장치로, 과거에는 그 수가 제한적이었으나 최근 들어 고급 차량에서는 70개 이상의 ECU가 탑재되는 것으로 알려져 있다. 이렇게 많은 ECU가 차량에 장착되면서, 이들 간의 원활하고 신속한 통신은 차량의 성능과 안전성에 중요한 역할을 하게 되었다. 이러한 변화는 자동차 기술의 발전이 단순한 하드웨어적 성능 향상을 넘어, 전자적 제어와 통합 시스템의 효율성에 얼마나 크게 의존하게 되었는지를 잘 보여준다.

이러한 다수의 ECU가 효율적으로 통신하기 위해 사용하는 주요 네트워크 기술 중 하나가 바로 CAN(Controller Area Network)이다[1]. CAN은 1993년 ISO 11898 표준으로 정의된 차량용 내부 네트워크로, 차량 내 ECU들이 서로 데이터를 주고받는 데 있어 핵심적인 역할을 한다. CAN은 차량의 엔진, 브레이크, 에어백 등 주요 시스템부터 엔터테인먼트 시스템, 네비게이션, 스마트폰 연결 등 다양한 기능을 담당하는 ECU 간의 데이터를 빠르고 안정적으로 전달할 수 있도록 하는 중요한 통신망이다. CAN은 멀티플렉싱 방식을 사용하여 여러 ECU가 동일한 통신 버스를 공유하면서도, 충돌 없이 데이터를 주고받을 수 있도록 한다. 이를 통해, 각 ECU는 필요에 따라 실시간으로 데이터를 교환하고 차량의 각 기능이 원활하게 동작하도록 지원한다. 예를 들어, 차량의 엔진 제어 ECU가 차량 속도 데이터를 CAN 버스를 통해 브레이크 시스템 ECU에 전달하면, 이 데이터는 즉시 차량의 제동 성능에 반영되어 안전한 운행을 가능하게 한다.

CAN의 활용이 확대됨에 따라 ECU 시스템 간의 연결과 상호작용이 더욱 복잡해지고 있으며, 이는 차량에 더욱 정교하고 다양한 기능 구현의 기반을 제공한다. 이 기술을 통해 현대의 차량에서는 차량 내부의 네트워크를 통해 자동 주행, 차량 상태 모니터링, 실시간 교통 정보 제공 등 다양한 첨단 기능이 구현되고 있다.

그러나 이러한 연결성의 증가는 동시에 네트워크 보안에 대한 새로운 우려를 일으키고 있다. 자동차 네트워크가 더욱 복잡해짐에 따라, 외부 공격자들이 이 네트워크에 접근하여 차량 제어를 장악하거나, 차량 내의 중요한 데이터를 탈취할 위험이 증가하고 있다[2]. 이러한 보안 위협은 자동차의 안전성과 사용자 개인 정보 보호에 심각한 문제를 초래할 수 있다. 실제로 2015년에 C. Miller와 C. Valasek은 Jeep Cherokee 차량을 원격으로 해킹하여 디스플레이, 브레이크, 핸들 제어 등 차량의 여러 기능을 공격자가 임의로 조작할 수 있음을 입증하였다[3]. 또한 2016년에는 텐센트의 Keen Security Lab에서 테슬라 차량의 네트워크 취약점을 이용해 차량 제어가 가능함을 발표하였다[4]. 이들은 차량의 주요 시스템에 접근하여 차량을 원격으로 조작할 수 있었고, 이로 인해 자동차 네트워크의 보안 문제가 더욱 부각되었다.

이러한 사례와 더불어 최근에도 여러 공격 사례들이 보고되고 있으며, 이는 CAN 버스를 포함한 자동차 네트워크의 보안 취약성을 드러내고 있다. 이에 따라 제조업체들은 보안 취약성을 보완하기 위해 다양한 침입 탐지 시스템 연구를 진행하고 있다[5]. 특히, ECU의 고유한 하드웨어적 특징과 이를 수집 및 활용하는 방법에 따라 개발된 다양한 식별 기술들이 주목받고 있다. 이러한 기술들은 ECU 간의 통신에서 비정상적인 활동을 감지하여 잠재적인 공격을 방지하는 데 사용된다.

현재까지 개발된 식별 기술들은 각기 고유한 장점을 가지고 있지만, 실제 차량에 적용하기에는 여러 한계점이 존재한다. 예를 들어, 전압 신호를 활용한 기술은 공격자가 모방하기 어렵다는 점에서 높은 보안성을 제공하지만, 외부 환경의 변화에 매우 민감하다는 단점이 있다. 또한, 시간차(Clock-skew) 기반의 식별 방법은 ECU의 주기적 메시지에 대해서는 효과적이지만, 비주기적인 메시지에 대해서는 식별 능력이 떨어진다는 한계가 있다. ECU의 오류 상태를 기반으로 한 식별 연구는 시스템에 직접적인 영향을 미칠 수 있어 추가적인 검증이 필요하다[6].

최근에는 인공지능(AI)을 활용한 다양한 연구가 이루어지고 있다[7-18]. AI 기반 기술은 기존의 한계를 넘어, 보다 정교하고 정확한 침입 탐지 시스템을 개발하는 데 도움을 줄 수 있다. AI는 대량의 데이터를 분석하고 패턴을 학습하여 비정상적인 활동을 보다 신속하게 감지할 수 있는 능력을 가지고 있다. 특히, CAN 프로토콜에서의 메시지 특성과 시계열 데이터의 변화를 분석하는 과정에서, AI 모델은 시간에 따른 데이터의 미묘한 변화를 인식하고 이를 바탕으로 비정상적인 메시지를 탐지하는 데 큰 도움을 줄 수 있다[7-12]. 이러한 기술들은 앞으로의 자동차 네트워크 보안에서 중요한 역할을 할 것으로 기대된다.

본 논문에서는 CAN 프로토콜의 메시지 특성을 분석하고, 이를 바탕으로 비정상적인 메시지를 탐지하는 방법을 연구하였다. 먼저, 메시지의 주기성과 데이터의 시계열 변화를 조사하고, CAN 메시지를 디코딩하여 데이터 변화량을 학습하였다. 이를 통해 비정상적인 메시지를 효과적으로 탐지하는 데 어떤 영향을 미치는지 평가하였다. 또한 본 논문은 위 연구를 토대로 시계열 데이터 처리에 강점을 가진 GRU를 이용해 예측되는 정상 메시지와 실제 발생한 메시지의 편차를 분석하여 정상과 비정상으로 이진 분류한다. 이때 피쳐의 특성으로 인해 오분류를 유발하는 피쳐를 식별하고 해당 피쳐를 자동으로 제어하기 위한 방법론을 제안하였으며, 메시지의 주기성을 활용하여 정상과 비정상 메시지를 분류하여 성능을 대폭 향상하였다. 또한 공격 메시지에 대해 세부 공격 분류를 위하여 랜덤 포레스트 모델을 활용하여 다중 분류를 수행하여 공격 메시지에 대한 정확한 분류가 가능한 방법을 제안한다.

본 논문은 보다 안전하고 신뢰할 수 있는 자동차 네트워크 구축에 기여할 수 있기를 기대한다.

2. 관련 연구

Min의 논문에서는 CAN 메시지의 시간 간격 분석을 기반으로 한 경량 침입 탐지 알고리즘을 제안한다. 유명 제조업체 차량의 CAN 메시지를 캡처해 세 가지 메시지 주입을 수행한

결과, 시간 간격이 공격 탐지에 중요한 기능임을 발견하였다. 정상 상태에서 특정 CAN ID의 시간 간격은 약 0.1초였으나, 공격 시에는 약 10%로 짧아졌다. 제안된 IDS는 메시지 주입을 밀리초 만에 탐지하며, 오탐 오류 없이 100%의 탐지 정확도를 보인다[7].

Koltai의 논문에서는 CAN 버스 통신에서 비정상적인 메시지를 탐지하기 위해 상관관계와 시계열 예측을 결합한 방법을 제안한다. 상관 기반의 시간적 합성곱 네트워크(TCN) 모델을 사용하여 차량 신호의 후속 값을 예측하고, 이를 통해 이상을 감지한다. 이 방법으로 탐지 성능이 68%에서 95%로 향상되어 제안된 방법이 대부분의 공격 시나리오를 효과적으로 탐지할 수 있음을 보여준다[8].

Tariq의 논문에서는 차량의 CAN 버스에서 이상 및 공격을 감지하기 위한 포괄적인 시스템인 CAN-ADF (Controller Area Network Attack Detection Framework)을 소개한다. 이 시스템은 규칙 기반과 RNN (Recurrent Neural Networks)을 결합하여 공격을 탐지한다. 기아 소울과 현대 소나타에서 수집된 7,875,791개의 차량 CAN 패킷을 사용하여 알고리즘의 정확도를 평가한 결과, 평균 99.45%의 정확도를 달성하였다[9].

Khan의 논문에서는 통계 분석 방법을 사용하여 침입 탐지를 위한 매개변수 기반 최적화 임계값 슬라이딩 윈도우 접근 방식을 설계, 개발 및 구현했다. CAN 버스 네트워크의 고유한 특징을 활용하여 최소 MR(Miss Rate)을 달성하였다. 여러 윈도우 크기에 대한 임계값을 최적화하여 최소 MR을 얻었으며, 이는 브루트 포스 최적화를 사용하여 달성되었다[10].

Wang의 논문에서는 bzip2 압축 알고리즘을 기반으로 한 CAN 버스 트래픽 이상 탐지 방법인 SIDuBzip2를 제안한다. CAN ID의 시계열 유사도를 계산하여 비정상 트래픽을 식별하며, 0.02의 전송 주기를 선택해 탐지를 수행한다. 실험 결과, SIDuBzip2는 Replay 공격을 제외한 모든 공격에서 99.2% 이상의 F1-score를 기록하였다[11].

Hossain의 논문에서는 CAN 버스 네트워크 공격을 탐지하고 완화하기 위해 LSTM기반의 IDS을 제안한다. 실험용 자동차에서 공격이 없는 데이터를 추출하고, 여기에 공격을 주입하여 자체 데이터셋을 생성하였다. 이 데이터셋을 모델의 학습과 테스트에 사용하였으며, 설정된 하이퍼파라미터 값으로 99.995%의 높은 탐지 정확도를 달성하였다[12].

Serag의 논문에서는 CAN 버스의 일반적인 공격인 Message injection, impersonation, flooding으로부터 보호하는 ZbCAN 시스템을 제안한다. CAN 프로토콜의 사용되지 않는 비트에 인증 정보를 삽입함으로써, ZbCAN은 효율성을 유지하면서 보안을 강화한다. 최악의 응답 시간 분석과 다양한 공격 유형에 대한 확률적 보안 분석을 통해 유효성을 입증하였으며, 방지율은 targeted-injection 98.5%, replay 98.4%, 탐지율은 각각 100%를 보여주었다[13].

Wang의 논문에서는 GAN 모델을 기반으로 하는 침입탐지 시스템을 제안한다. CAN-FD의 ID 이미지에서 관련된 특징을 추출하기 위해 합성곱의 깊이를 향상시켜 이미지 특징을 보다 효율적으로 추출하였다. 또한 알려지지 않은 공격으로 효과적으로 처리하기 위해 이중 판별기 개념을 도입한다. 메시지 탐지는 0.15ms 이내에 이루어지며 fuzzy, DoS, RPM, Gear 공격 각각의 탐지율은 99.7% 이상의 수치를 기록하여 평균 99.93%의 탐지율을 기록하였다[14].

Hoang의 논문에서는 해킹 및 대책 연구실(HCRL)에서 제작한 두 가지 인기 있는 데이터셋을 사용하였다. 제안된 시스템은 지도 대조 학습과 전이 학습을 결합하였다. CAN ID 시퀀스를 이진 형태로 조작하여 지도대조손실(supervised contrastive loss)로 학습된 SupCon ResNet을 도입하였다. 결과적으로 지도 대조 손실은 퍼지 공격의 거짓 부정률을 크게 줄였으며, Normal, DoS, Fuzzy, Malfunction, Overall의 각 F1 값이 99.2% 이상을 보였다. 그러나 알려지지 않은 공격은 탐지가 불가능하다는 한계가 있다[15].

Kim의 논문은 CAN 버스를 보호하기 위해 GAN을 통해 생성한 CAN 데이터와 실제 데이터를 사용한 트리 기반의 침입 탐지 시스템을 연구하였다. 트리 기반의 IDS는 GAN을 통해 생성된 가짜 공격 데이터와 실제 데이터를 학습하여, 기존 판별되지 않은 공격을 탐지하는 것이 가능해진다. 최종적으로 의사결정트리의 F1값 0.993 과 랜덤 포레스트 F1값 0.996의 결과를 얻었다[16].

Hoang의 논문은 CAN 버스 침입 탐지 시스템(IDS)을 위한 반지도 학습 기반의 컨볼루셔널 적대적 오토인코더 모델을 제안하였다. 이 모델은 오토인코더와 생성적 적대 신경망(GAN)을 결합한 적대적 오토인코더(AAE)를 사용한다. 비라벨링 데이터로 정상 및 공격 패턴을 학습한 후 소량의 라벨링 데이터로 감독 학습을 수행한다. 실험 결과, 제안된 모델은 F1 점수 0.9984와 0.1%의 낮은 오류율을 달성하였으며, 모델 파라미터 수와 추론 시간을 각각 5배, 8배 감소시켰다[17].

Shafique의 논문은 차량 내 네트워크 보안을 강화하기 위해 AI 기반 생성 공격과 기존 공격 모두를 방어하는 방법을 제안한다. CTGAN (Conditional Tabular Generative Adversarial Network)를 사용하여 차량 내 네트워크 트래픽을 생성하고, 이를 벤치마크 트래픽과 결합하여 다양한 시나리오를 만든다. 랜덤 포레스트 모델은 벤치마크 트래픽과 AI 기반 트래픽에 대해 각각 0.93과 0.89의 정확도를 달성하였다[18].

본 연구는 위 기존의 CAN 메시지 비정상 탐지 방법론과 비교하여 몇 가지 차별화된 접근을 제안한다. 첫째, GRU 모델을 활용하여 CAN 메시지의 발생 주기를 분석하고, 예측된 메시지와 실제 발생한 메시지 간의 차이를 이용한 비정상 메시지 탐지 기법을 제시하였다. 이는 메시지의 주기성과 변동성을 효과적으로 포착함으로써 높은 정확도와 계산 효율성을 동시에 달성할 수 있었다. 둘째, DBC 디코딩을 통해 CAN 메시지의 구체적인 의미를 해석하고, 이를 비정상 탐지에 활용하였다. 이러한 접근은 메시지의 시계열 특성뿐만 아니라 내용까지 고려하여 보다 정교한 탐지를 가능하게 하였다.

3. 방법론

3.1 데이터 분석

본 논문에서는 CAN 메시지 아이디 값의 130 ACU 메시지를 예제로 사용하여 CAN 메시지의 주기성과 시계열 변화량을 분석하였다.

1) 데이터 주기성 분석

Fig. 1은 130 ACU 메시지의 발생 주기를 시각적으로 보여준다. 이를 통해 130 ACU 메시지가 10ms 주기로 발생되며, 주기 오차가 10% 내외임을 알 수 있다. 이는 메시지가 규칙적으로 발생하고 있음을 나타낸다.

그러나 일부 데이터에서 이 주기성을 벗어난 경우를 발견하였다. 분석 결과 비정상 데이터로 식별되었다. 이로써 주기성을 통해 정상 데이터와 비정상 데이터를 구분할 수 있다는 사실을 확인할 수 있었다.

Table 1은 각 CAN 메시지를 ID별로 평균 주기를 정리한 것이다.

Tabel 1. CAN Message ID Periods and Error Ranges Summary

Time Delta(ms)	CAN ID
10	130, 140, 153, 164, 220, 251, 260, 2B0, 329, 340, 356, 366, 367, 368
20	381, 386, 387, 389, 38D, 391, 394, 420, 421, 453, 470, 47F
50	436, 485, 48A, 490, 492, 58B
70	484
100	42D, 479, 495, 500, 507, 50C, 520, 53E, 541, 568
200	410, 412, 44E, 483, 48C, 49F, 4A9, 4C9, 4CB, 50A, 50E, 52A, 53B, 53F, 544, 553, 559, 572, 593, 5A6, 5CD
500	4A2, 4A7, 563
1,000	043, 07F, 5B0, 5BE
1,500	57F
2,000	000, 4A4, 7C4, 7CC, 7D0, 7D4, 7D8, 7DC,

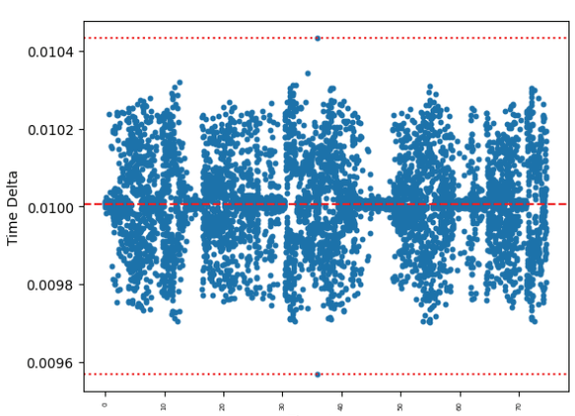

Fig. 1. 130 ACU Message Cycle

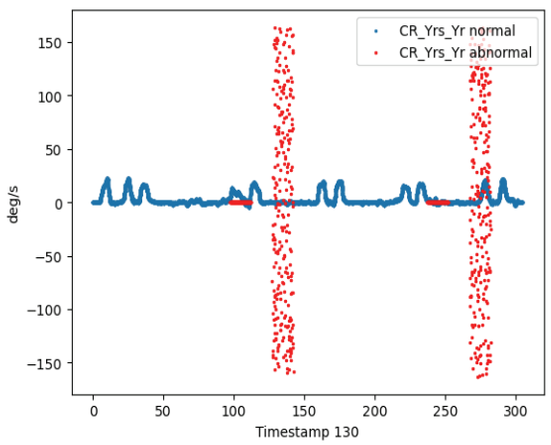

Fig. 2. 130 ACU Message Angular Velocity(deg/s).

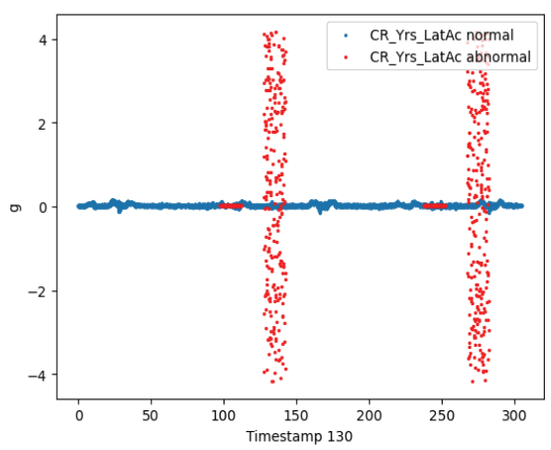

Fig. 3. 130 ACU Message Angular Acceleration(g).

2) 시계열 분석

일례로 메시지 ID 130인 에어백 컨트롤 유닛(ACU) 메시지의 각속도(deg/s)와 각가속도(g) 변화를 시계열 데이터로 분석한 결과이다. 아래 Fig. 2, 3은 130 메시지의 각속도와 각가속도가 시간에 따라 어떻게 변화하는지를 보여준다. 파란색 선의 정상적인 메시지는 이전의 값에서 각속도와 각가속도가 일정한 연속성과 추세 패턴을 따르지만, 붉은색 선의 비정상 메시지는 이전의 패턴에서 벗어난 예측 불가능한 변동이 나타난다. 이는 ID 366의 엔진 컨트롤 유닛(ECU) 메시지에서도 동일한 경향이 나타난다.

3.2 데이터 전처리

1) OG (원본데이터:Original Data)

피쳐 중 'Data'의 경우 16진수로 이루어져 있으며 '20 A1 10 FF 00 FF 50 1F' 와 같이 값 사이에 공백이 포함되어 있기 때문에 학습에 방해가 될 수 있다. 따라서 공백을 기준으로 'D0'~'D7'까지 피쳐를 구분하였다.

학습 데이터와 테스트 데이터 모두 각 ID들이 함께 결합되어 있는 형태이나, TF+DF+DBC+BDF를 진행하기 위해서는

Table 2. Performance Comparison by Timediff

Timediff	Accuracy	F1-score	Precision
Timediff1	0.991	0.873	0.887
Timediff1,2	0.997	0.964	0.978
Timediff1,2,3	0.998	0.968	0.981
Timediff1,2,3,4	0.998	0.969	0.983
Timediff1,2,3,4,5	0.998	0.971	0.983
Timediff1,2,3,4,5,6	0.998	0.973	0.985
Timediff1,2,3,4,5,6,7	0.998	0.973	0.985

각 ID 별로 구분해서 학습해야 한다. 이로 인해 모든 데이터셋을 각 ID 별로 구분하여 학습하였다.

2) TF (발생 주기 데이터:Time-differential value)

TF는 OG 데이터에 'Timediff'를 결합한 형태이다. 정상 데이터와 공격 데이터는 주기성에 따라 다르게 나타난다. 이를 학습에 반영하기 위해 'Timediff'값을 피쳐로 추가하였다. Table 2는 Pre_train_D_1과 Pre_train_D_2를 사용하여 학습 및 테스트한 결과를 'Timediff'별로 나타낸 것이다. 'Timediff'에 과거 7개의 값을 추가한 이후에는 성능이 더 이상 개선되지 않았으므로, Timediff 6까지만 추가하여 학습을 수행하였다.

Timediff은 'Timestamp'를 기준으로 발생 간격을 계산한 것이다. 'Timediff1'은 바로 이전 행과의 발생 간격을, 'Timediff2'는 2번째 전 행과의 발생 간격을 계산하였다. 이와 같은 패턴을 사용하여 'Timediff6'까지 값을 계산하였다. 하지만 각 ID별로 나뉜 데이터들의 1-6번째 행은 값의 부재로 인해 'Timediff' 값을 구하지 못한다. 따라서 그 경우 ID별로 'Timediff'값의 평균값을 구하여 대체하였다.

3) TF+DF(차분데이터:Data-differential value)

TF+DF는 TF에 'Datadiff'가 결합된 형태이다. 'Datadiff'의 경우 D2의 차분값을 구한 값으로, 'Timediff'와 같은 방법으로 총 3개의 'Datadiff'값을 계산하였다.

4) TF+DF+DBC[1)]+BDF(DBC의 차분값)

TF+DF+DBC+BDF는 TF+DF에 디코딩 값과 디코딩 데이터의 'Datadiff'값을 결합한 형태이다. 피쳐 중 'Data'의 경우 16진수로 표현되어 있으며, 이를 10진수로 변환한다고 하더라도 값의 의미를 파악하기 어렵다. 따라서 학습 데이터의 자동차 회사인 현대의 hyundai_kia_generic.dbc파일[19]을 이용하여 CAN메시지의 'Data'를 디코딩을 하였다. DBC 파일에서 동일한 ID를 찾은 후, 'Data'의 의미를 디코딩하여 값을 해석하였다.

각 ID 별로 DBC ID 구조를 가지고 있으며, Python의 'Cantools' 라이브러리를 활용하면 디코딩을 진행하였다.

학습과 테스트에 사용된 'Pre_train_D_1~2', 'Pre_Submit_D_0~3' 데이터셋에는 모든 ID가 포함되어 있다. 각 ID는 서로 다른 피쳐를 가지고 있는 DBC 데이터를 디코딩하면서, 중복되지 않는 피쳐들로 인해 다수의 'NaN' 값이 발생하였다. 디코딩 후 피쳐의 수는 총 651개로 증가하였다. 이에 따라 ID별로 데이터를 분리하여 총 56개의 데이터를 생성한 후, 각 ID 별로 학습을 진행하였다.

디코딩 전 데이터의 'Datadiff'를 구하여 학습에 활용했듯이, 디코딩된 데이터들의 'Datadiff'도 추가로 계산하여 학습에 피쳐로 활용하였다. 그러나 문자열로 구성된 피쳐들과 모든 값이 동일하여 차분값을 구하는 의미가 없는 피쳐들의 경우 제외하였다.

5) GRU+ 비정상 분류

3.1절에서 언급한 'Timediff' 값을 기반으로, 정상 메시지를 GRU(Gated Recurrent Unit) 모델을 이용하여 학습하고, 시간 윈도우 내에 발생한 메시지를 통해 차기 메시지를 예측하였다. 예측값과 실제 발생값 간의 차이 잔차(L1 Loss)를 계산하여, 동일한 발생 주기 내에서 다수의 메시지가 발생할 경우 잔차가 가장 작은 메시지를 정상 메시지로 분류하였다. 즉, 과거 시간 윈도우 내에 발생한 정상 메시지들의 집합을 '메시지 윈도우'로 정의하고, 이를 GRU 모델의 입력 시계열 데이터로 활용하였다. 마지막으로 발생한 정상 메시지로부터 메시지 발생 간격 T_{gap}과 그 오차 T_{err}을 합산한 기간을 T_{span}으로 정의하였으며, 이 기간 동안 발생한 메시지를 정상 또는 비정상으로 분류하는 문제로 설정하였다.

CAN 메시지를 GRU를 활용한 비정상 탐지를 학습할 때 메시지 피쳐 중에 CRC와 같이 시간적 연관성이 없는 피쳐가 존재하며 이 데이터로 인해 시계열 학습과 예측에 오류를 만들어 내어 정상과 비정상 분류의 성능을 저하됨을 확인하였다.

시계열 성격을 가지지 않는 피쳐를 배제하고 훈련하기 위하여 정상 메시지로 학습한 이후에 다시 정상 메시지로 예측을 진행하여 각 피쳐의 L1 차이를 계산하여 일정 임계값을 초과하는 피쳐를 학습에서 배제하였다.

6) 정규화

모든 특성을 동등하게 고려할 수 있도록 최소-최대 정규화를 사용하였다. 이 방법을 통해 각 특성의 값 범위를 [0, 1] 사이로 조정하여, 모델이 데이터를 더 잘 학습하고 일반화할 수 있도록 하였다. 또한, 데이터의 스케일을 일정하게 맞추어 모델이 학습할 때 발생할 수 있는 수치적 불안정성을 줄이는 데 기여하였다.

3.3 모델

1) 다중분류 모델

본 논문에서는 다양한 머신 러닝 모델과 딥러닝 모델을 활용하여 CAN 비정상 메시지 분석을 수행하였다. 다중 분류만 수행하는 머신 러닝 모델로는 XGBoost, 랜덤 포레스트, SVC (Support Vector Classifier)를 사용하였으며, 각각의 모델이

1) CAN Database

비정상 메시지 탐지에서 어떻게 성능을 발휘하는지 비교 분석하였다. 딥러닝 모델로는 인공 신경망(ANN, Artificial Neural Network)을 사용하여, 복잡한 패턴 인식과 비정상 메시지 탐지에서의 성능을 평가하였다.

2) GRU+RF(예측기반 공격 메시지 분류 모델)

정상 데이터와 비정상 데이터를 분류하는 이진 분류기는 3.2의 5)에서 제시한 GRU(Gated Recurrent Unit)를 사용하였다. GRU는 LSTM과 유사한 구조를 이루고 있다. GRU는 RNN에서 발생할 수 있는 기울기 소실(Gradient Vanishing)문제를 해결할 수 있으며, LSTM보다 구조가 간결하기 때문에 더 효율적으로 계산할 수 있다. 또한 시계열 데이터나 연속된 데이터의 종속성을 잘 처리할 수 있도록 설계되어 있다. GRU에는 갱신 게이트(Update gate), 리셋 게이트(Reset gate)가 존재한다. 갱신 게이트는 LSTM의 망각 게이트(forget gate)와 입력 게이트(input gate) 역할을 하며, 과거의 정보와 새로운 정보를 얼마나 가져와야 할지 결정한다. 리셋 게이트는 현재의 상태에서 이전 정보를 얼마나 유지할지를 결정한다. LSTM은 셀 상태(Cell state)와 히든 상태(Hidden state)가 모두 존재하지만, GRU는 셀 상태와 히든 상태를 하나의 히든 상태로 합쳤기 때문에 LSTM에 비해 빠른 연산속도를 수행할 수 있다.[20]

LSTM에만 존재하는 셀 상태의 경우 장기적인 정보를 유지하기 위한 경로이다. 셀 상태는 장기기억을 저장하고, 게이트를 통한 정보를 제어한다. 정보가 각 시간 단계를 거쳐도 비교적 변하지 않도록 설계되어 있기 때문에 LSTM이 장기 의존성(Long-term dependency)를 학습할 수 있도록 한다. 또한 LSTM의 게이트(망각 게이트, 입력 게이트, 출력 게이트)를 통해 셀 상태는 어떤 정보를 기억할지, 갱신해야 할지, 잊을지를 결정한다. 이로 인해 셀 상태는 시계열 데이터에서 중요한 정보를 장기간 유지할 수 있다[21]. 본 논문에서 연구하는 CAN 메시지의 경우 장기기억보다는 최근 정상 메시지의 추세가 더 중요하다. 예를 들어 차량이 이동할 때, 속도가 급격하기 변하기보다는 최근에 변화된 속도를 바탕으로 오르거나 내려간다. 가속도나 RPM도 비슷한 양상을 보인다. 따라서 CAN 메시지를 연구에서는 장기기억을 학습할 수 있는 주요 요소인 셀 상태가 포함된 LSTM보다는 셀 상태가 제외된 GRU를 비정상 메시지를 탐지에 활용하였다.

하지만 GRU는 시계열 분석에 유용할 뿐 다중 분류기로서의 역할을 하기에는 한계가 존재한다. 따라서 GRU를 활용하여 예측값과 발생값의 잔차를 바탕으로 정상과 비정상 메시지를 분류하는 데 활용하였으며, 본 연구의 목표인 비정상 메시지를 세부 공격 기법(Flooding, Fuzzing, Replay, Spoofing)으로 추가 분류하기 위해 다중 분류를 수행할 필요가 있다. 다중 분류 모델은 정규화한 메시지와 L1 오차와 정상/비정상을 분류한 이진 분류 결과를 활용하였으며 입력 피쳐의 수가 최대 50개 이내로 많지 않아 랜덤 포레스트 모델을 활용하였다.

Fig. 4는 제안된 GRU+RF 모델의 전반적인 네트워크 아키텍처이다. 각 모듈의 세부 사항은 다음과 같다.

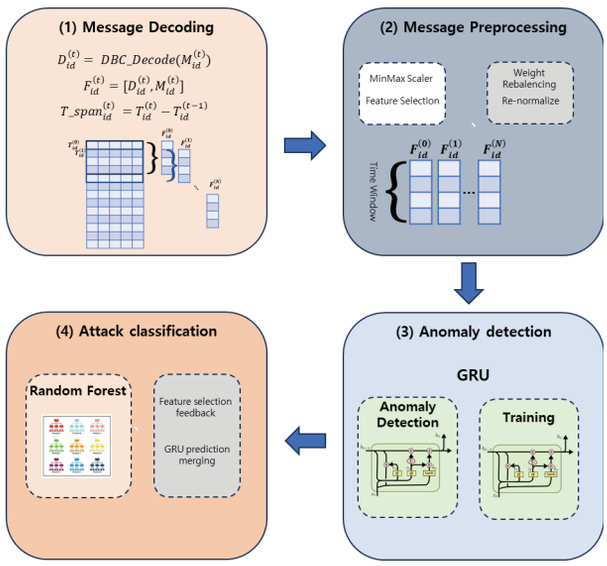

Fig. 4. The Overall Framework of GRU+RF

4. 실험 환경

4.1 데이터셋

본 연구에서는 고려대학교 해킹 대응 기술 연구실에서 개발한 주행 및 주차 정보로 이루어진 데이터셋을 활용하였다. 이 데이터셋은 현대 아반떼 CN7의 CAN Message를 기반으로 하며, 정상, Flooding, Fuzzing, Replay, Spoofing으로 분류되어 있다[23].

Train Set으로 Pre_train_D_1(806,390개)과 Pre_train_ D_2 (889,395개)로 사용했으며, Test Set으로는 Pre_submit _D_0 (333,456개), Pre_submit_D_1(333,456개), Pre_sub mit_D_2 (333,456개), Pre_submit_D_3(333,456개)를 사용하였다.

4.2 실험 환경

실험은 파이썬을 사용하였다. Table 3에서와 같이 디코딩, 데이터 전처리, 모델 학습, 성능 평가, 시각화를 위하여 파이썬 라이브러리인 cantools, scikit-learn, pandas, numpy, matplotlib, tensorflow, keras를 활용하였다. 실험은 Google Colab을 통해 진행하였으며, 디코딩과 GRU 학습에는 Colab에서 제공되는 GPU를 사용하였고 비정상 탐지와 랜덤 포레스트를 활용한 다중 분류는 시스템의 CPU를 사용하였다.

Table 3. Experimental Environment

Environment	Name
Language	Python
Library	cantools, scikit-learn, pandas, numpy, matplotlib, tensorflow, keras
CPU	AMD Ryzen 7 4700U with Radeon Graphics 2.00GHz
RAM	16GB

4.3 평가 지표

본 논문은 다양한 분류 모델의 성능을 측정하기 위해 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1-Score을 사용하였다.

정확도는 모델이 올바르게 분류한 샘플의 비율을 나타낸다. 전체 예측 중에서 올바르게 예측한 비율을 측정한다.

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

정밀도는 모델이 양성으로 예측한 샘플 중에서 실제로 양성인 샘플의 비율을 나타낸다. 모델이 양성이라고 예측한 것 중에서 실제로 양성인 비율을 측정한다.

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

재현율은 실제 양성인 샘플 중에서 모델이 양성으로 올바르게 예측한 비율을 나타낸다. 실제 양성인 샘플들 중에서 모델이 얼마나 많은 샘플을 식별할 수 있는지를 측정한다.

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

F1-Score는 Precision과 Recall의 조화 평균이다. 모델의 정밀도와 재현율 사이의 균형을 평가하는 지표로, 불균형한 클래스 분포에서 유용하게 사용된다.

$$F_1 = 2 * \frac{\mathrm{Precision} * Recall}{\mathrm{Precision} + Recall} \quad (4)$$

5. 연구결과

5.1 모델별 연구결과

1) 다중 분류 모델

실험은 3절에서 소개한 총 5개의 데이터 중 4가지 데이터를 총 4가지 모델로 테스트하여 성능을 비교하였다. Table 4는 랜덤 포레스트, XGBoost, SVM, ANN이 산출한 F1값이다.

Table 4를 보면 TF를 추가했을 때 랜덤 포레스트와 XGBoost의 F1값이 약 20%가량 향상되었다. 또한 DF를 추가했을 때는 0.7% 향상되었으며, DBC와 'DBC diff' 추가로 0.7%가 향상되었다.

TF+DF까지는 랜덤 포레스트가 높은 성능을 보였지만 DBC 디코딩 피쳐를 추가한 4번째 데이터셋부터는 XGBoost가 우세한 성능을 보였다. Table 5은 XGBoost를 사용하였을 때 정확도, F1값, 정밀도, 재현율이다.

랜덤 포레스트는 랜덤으로 샘플을 선정하여 나무를 만드는데, 만약 피쳐가 너무 많아지면 주요 피쳐가 반영되지 않을 확률이 올라간다. Table 6을 보면, ID 251의 경우, TF+DF+DBC+BDF 데이터셋에서 디코딩된 피쳐와 해당 피쳐의 'Datadiff' 값이 추가되면서 피쳐의 수가 거의 두 배로 증가한 것을 확인할 수 있다. 따라서 피쳐가 다수 증가된 TF, DF, DBC, BDF의 결합된 경우, 부스팅 기법을 적용한 XGBoost가 랜덤 포레스트보다 우수한 성능을 나타내는 결과를 보였다.

Table 4. Model-specific F1-scores

	Data	RF	XGBoost	SVM	ANN
1	OG	0.719	0.715	0.684	0.646
2	TF	0.946	0.925	0.683	0.687
3	TF+DF	0.953	0.935	0.702	0.681
4	TF+DF+DBC+BDF	0.955	0.961	0.713	0.708

Table 5. Performance Evaluation of XGB using TF+DF+DBC+BDF

Model	Accuracy	F1-score	Precision	Recall
XGBoost	0.997	0.961	0.989	0.943

Table 6. Comparison of Column Counts for ID 251

Data	Counts	Columns
OG	12	'Timestamp', 'Arbitration_ID', 'D0', 'D1', 'D2', 'D3', 'D4', 'D5', 'D6', 'D7', 'DLC', 'SubClass'
TF	18	'Timestamp', 'Arbitration_ID', 'D0', 'D1', 'D2', 'D3', 'D4', 'D5', 'D6', 'D7', 'DLC', 'SubClass', 'TimeDiff1', 'TimeDiff2', 'TimeDiff3', 'TimeDiff4', 'TimeDiff5', 'TimeDiff6'
TF+DF	21	'Timestamp', 'Arbitration_ID', 'D0', 'D1', 'D2', 'D3', 'D4', 'D5', 'D6', 'D7', 'DLC', 'SubClass', 'TimeDiff1', 'TimeDiff2', 'TimeDiff3', 'TimeDiff4', 'TimeDiff5', 'TimeDiff6', 'diff_1', 'diff_2', 'diff_3'
TF+DF+DBC +BDF	46	'Timestamp', 'Arbitration_ID', 'SubClass', 'DLC', 'D0', 'D1', 'D2', 'D3', 'D4', 'D5', 'D6', 'D7', 'DBC_F1', 'DBC_F2', 'DBC_F3', 'DBC_F4', 'DBC_F5', 'DBC_F6', 'DBC_F7', 'DBC_F8', 'DBC_F9', 'DBC_F10', 'DBC_F11', 'TimeDiff1', 'TimeDiff2', 'TimeDiff3', 'TimeDiff4', 'TimeDiff5', 'TimeDiff6', 'diff_1', 'diff_2', 'diff_3', 'DBC_F1_diff', 'DBC_F2_diff', 'DBC_F3_diff', 'DBC_F4_diff', 'DBC_F5_diff', 'DBC_F6_diff', 'DBC_F7_diff', 'DBC_F8_diff', 'DBC_F9_diff', 'DBC_F10_diff', 'DBC_F11_diff', 'diff_1_diff', 'diff_2_diff', 'diff_3_diff', 'Sum_diff'

2) GRU+RF 모델

실험은 3절의 GRU+비정상 분류에서 소개한 메시지의 L1 잔차 데이터를 활용한다. 또한 예측 기반 공격 메시지 분류 모델에서 소개한 GRU+를 통해 정상과 비정상을 분류하고 랜덤 포레스트를 공격에 대한 다중 분류기로 활용한 결과이다.

Table 7은 GRU+RF모델로 학습했을 때 Accuracy, F1 값,

Table 7. Performance Evaluation of a Model Combining Binary Classification and Multiclassification

Model	Accuracy	F1-score	Precision	Recall
GRU+RF	0.993	0.992	0.992	0.993

Precision의 평균값이다. 다중 분류 모델만 사용해서 비정상 탐지를 했을 때보다 F1-score의 값이 약 3% 향상된 결과를 볼 수 있다. 전체적으로 99%의 성능 결과를 기록하였으며, 다중 분류 모델만 사용했을 때와 달리 F1값, 정밀도, 재현율이 대체적으로 안정적인 결과를 보인다.

5.2 최종 모델 성능 결과

최종적으로 L1 잔차 데이터를 활용한 데이터셋을 이용하여 학습한 GRU+RF모델이 가장 좋은 성능을 보였다. 모든 성능 평가에서 약 99%이상의 성능을 기록했으며 이는 모델이 비정상 메시지 탐지를 효과적으로 수행하고 있다는 의미이다. 본 실험에서 분석한 오탐 사례는 앞서 다른 실험에서와 같이 'Replay'공격에 대해 정상메시지로 탐지하는 문제와 'Replay' 공격이 발생하는 시점에 정상 메시지가 오히려 비정상 메시지로 분류되고 'Replay'메시지가 정상메시지로 분류되는 문제가 발견되었다. 이는 발생 시간 간격을 비정상 메시지 탐지에 중요 요소로 활용한 측면에서 피하기 어려운 문제로 보인다. 본 실험에서는 정상 메시지만 시계열로 분석하였기 때문에 한계가 있었지만 공격메시지에 대한 시계열 분석 방안을 마련하면 'Replay' 공격과 같이 일회성이 아닌 주기성을 가지는 정상과 유사한 메시지에 대해서도 공격을 식별할 수 있을 것이다.

6. 결론 및 향후 개선점

본 연구에서는 CAN(Controller Area Network) 비정상 메시지 분석에서 발생 주기성이 중요한 요소임을 입증하였다. 특히, 혼동 행렬을 사용한 실험에서 플러딩과 정상 메시지를 분류할 때 주기성 데이터가 핵심 역할을 하는 것을 확인하였다. 추가적으로, 데이터 변화량 분석이 비정상 여부 판단에 보조적인 역할을 수행함을 밝혀냈다. 또한, 데이터 차분값과 DBC 디코딩이 CAN 비정상 메시지 분석 성능 향상에 얼마나 기여하는지 평가한 결과, DBC 디코딩은 CAN 메시지의 구체적 의미를 해석하는 데 필수적이었다. 이를 통해 데이터 주기성을 활용한 효과적인 비정상 메시지 분석 방법을 도출할 수 있었다. 이러한 분석을 바탕으로 모델에 학습을 진행하였다. 학습 결과, 랜덤 포레스트와 XGBoost가 뛰어난 성능을 보였다. 다른 모델에 비해 우세한 성능을 보이던 랜덤 포레스트는 DBC 디코딩 과정에서 피쳐 수가 확장되자 XGBoost보다 낮은 성능을 보였다. 최종적으로 XGBoost 모델이 96%의 F1값으로 가장 우수한 성능을 보여주었다.

본 논문에서는 이를 바탕으로 새로운 모델을 제안한다. 제안된 모델은 시계열 데이터 처리에 강점을 가진 GRU를 사용하였으며, 특히 정상 메시지에 섞여 있는 비정상 메시지를 식별하기 위해 메시지 발생의 주기성을 활용한 GRU+ 모델을 도입하였다. 기본적으로 시계열 분석을 통해 비정상 메시지를 탐지하므로 알려지지 않은 공격에 대해서도 그 공격 분류가 부정확할 뿐 공격 자체를 탐지하는 능력에는 문제가 발생하지 않는 점이 이 모델의 장점이다. 이 모델은 L1 잔차를 이용해 비정상 메시지를 식별하고, L1 잔차, 발생 주기, 메시지 정보를 활용한 다중 분류기를 통해 세부적인 공격 기법을 효과적으로 분류하여 F1 값을 향상시켰다. 그 결과, 각 메시지 ID에 대해 평균 99%의 높은 F1값을 달성했으며, 다른 모델들보다 뛰어난 성능을 보여주었다.

그러나 일부 ID의 메시지에서는 낮은 탐지 성능이 나타났는데, 이는 적절한 메시지 발생 주기의 편차가 큰 메시지에서 나타나는 특징으로 분석하였다. 따라서 메시지의 발생 주기에 대한 편차가 큰 환경에서도 정상과 비정상 메시지를 보다 안정적으로 분류할 수 있는 방법에 대한 연구가 필요하다. 또한, 장기적인 데이터 패턴을 학습할 수 있는 모델을 개발하고, 다양한 공격 시나리오를 통해 모델의 범용성과 강인성을 평가할 필요가 있다. 본 연구는 GRU+ 모델 탐지를 위한 최소의 시간 범위 이내의 데이터만을 활용하여 실시간으로 발생하는 메시지를 분석하고, 정상과 비정상을 즉시 탐지가 가능하므로 실제 운영하는 차량에 적용을 검토가 가능하다는 장점이 있다.

References

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

[2] Upstream Security, 2024, "2024 Global Automotive Cybersecurity Report," https://upstream.auto/reports/global-automotive-cybersecurity-report/

[3] M. Charlie and V. Chris, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA 2015*, No.S 91, 2015.

[4] S. Nie, L. Liu, and Y. Du, "Free-fall: Hacking tesla from wireless to can bus," *Briefing, Black Hat USA*, Vol.25, pp.1-16, 2017.

[5] K. Kim, J. S. Kim, S. Jeong, J. H. Park, and H. K. Kim, "Cybersecurity for autonomous vehicles: Review of attacks and defense," *Computers & Security*, Vol.103, pp.102150, 2021.

[6] S. Lee, W. Choi, and D. H. Lee, "Research trends on techniques for identifying malicious ECUs in CAN networks," *Review of KIISC*, Vol.33, No.4, pp.47-55, 2023.

[7] S. H. Min, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network," *2016 International Conference on Information Networking (ICOIN)*. IEEE, pp.63-68, 2016.

[8] B. Koltai, A. Gazdag, and G. Acs, "Supporting CAN bus anomaly detection with correlation data," *ICISSP*, pp. 285-296, 2024.
[9] S. Tariq, S. Lee, H. K. Kim, and S. S. Woo, "CAN-ADF: The controller area network attack detection framework," *Computers & Security*, Vol.94, pp.101857, 2020.
[10] J. Khan, D. W. Lim, and Y. S. Kim, "Intrusion detection system can-bus in-vehicle networks based on the statistical characteristics of attacks," *Sensors*, Vol.23, No.7, pp.3554, 2023.
[11] C. Wang, X. Xu, K. Xiao, Y. He, and G. Yang, "Traffic anomaly detection algorithm for CAN bus using similarity analysis," *High-Confidence Computing*, 100207, 2024.
[12] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "LSTM-based intrusion detection system for in-vehicle can bus communications," *IEEE Access*, Vol.8, pp.185489-185502, 2020.
[13] K. Serag et al., "ZBCAN: A Zero-Byte CAN Defense System," *32nd USENIX Security Symposium*, Anaheim, 2023.
[14] X. Wang, Y. Xu, Y. Xu, Z. Wang, and Y. Wu, "Intrusion Detection System for In-Vehicle CAN-FD Bus ID Based on GAN Model," *IEEE Access*, 2024.
[15] T. N. Hoang and D. Kim. "Supervised contrastive ResNet and transfer learning for the in-vehicle intrusion detection system," *Expert Systems with Applications*, Vol.238, pp.122181, 2024.
[16] S. O. Kim and S. Y. Lee, "A study on the GAN-TREE Based vehicle network intrusion detection system," *Proceedings of the Korean Institute of Communications and Information Sciences Conference(KICS)*, pp.1444-1445, 2024.
[17] T. N. Hoang and D. Kim, "Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders," *Vehicular Communications*, Vol.38, pp.100520, 2022.
[18] R. Shafique, F. Rustam, G. S. Choi, and A. D. Jurcut, "Enhancing in-vehicle network security against ai-generated cyberattacks using machine learning," *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2024.
[19] Comma.ai, 2020, opendbc [Github], https://github.com/commaai/opendbc/blob/master/hyundai_kia_generic.dbc (2024)
[20] Y. E. Seo, C. H. Son, and H. Y. Lee, "*Method for Improving QQ-based Oxygen Extraction Fraction Estimation Accuracy through GRU Model*," *Journal of Korean Institute of Information Technology*, Vol.22, No.7, pp.131-139, 2024.
[21] Olah, Christopher, "Understanding lstm networks," 2015.
[22] H. Kang, B. Kwak, Y. H. Lee, H. Lee, H. Lee, and H. K. Kim, "Car Hacking: Attack & Defense Challenge 2020 Dataset," IEEE, 2021.
[23] S. R. Kim, B. H. Youn, and H. S. Cho, "Anomaly CAN message detection using heuristics and XGBoost," *Proceedings of the Annual Symposium of Korea Information Processing Society Conference (KIPS)*, Vol.31, No.1, pp.362-363, 2024.

김 세 린
https://orcid.org/0009-0009-6124-2664
e-mail : kimserin48@gmail.com
2020년 ~ 현 재 호서대학교 컴퓨터공학부 학사과정
관심분야 : 컴퓨터공학, 인공지능, 정보보호

성 지 현
https://orcid.org/0009-0008-2749-0843
e-mail : tjdwlgus0204@naver.com
2020년 ~ 현 재 호서대학교 컴퓨터공학부 학사과정
관심분야 : 컴퓨터공학, 네트워크보안, 정보보호

윤 범 헌
https://orcid.org/0009-0001-0905-2849
e-mail : ybh7159@gmail.com
2021년 ~ 현 재 호서대학교 컴퓨터공학부 학사과정
관심분야 : 컴퓨터공학, 정보보호

조 학 수
https://orcid.org/0009-0000-2411-043X
e-mail : marius1406@gmail.com
1997년 서울대학교 계산통계학과(학사)
1999년 서울대학교 전산과학과(석사)
2018년 고려대학교 컴퓨터전파통신공학과 (박사수료)
2024년 ~ 현 재 호서대학교 컴퓨터공학부 교수
관심분야 : 네트워크보안, 클라우드보안, 인공지능