논문 2024-2-7 http://dx.doi.org/10.29056/jsav.2024.06.07

ST-AAE: 시공간 적대적 오토인코더를 사용한 차량 내부 네트워크 침입 탐지

강효은*†

ST-AAE: Intrusion Detection in In-Vehicle Networks Using Spatio-Temporal Adversarial Autoencoder

Hyo-Eun Kang*†

요 약

차량 내 네트워크인 컨트롤러 영역 네트워크(CAN)는 악의적인 물리적 및 사이버 공격에 매우 취약하다. 사용자와 차량 제공자의 안전을 보장하기 위해 적절한 보안 조치가 필요하다. 본 논문에서는 차량 내 네트워크를 위한 새로운 침입 탐지 시스템(IDS)인 시공간 적대적 오토인코더(ST-AAE)를 제안한다. ST-AAE는 차량 내 IDS 분야에서 새로운 접근법으로, CAN 트래픽의 시공간 특성을 활용한다. 이 프레임워크는 비정상 트래픽을 탐지하고 공격 유형을 분류한다. 실제 운전 데이터에 대한 실험 결과, ST-AAE는 다양한 공격 유형에서 높은 정확도와 낮은 오탐률로 기존 모델보다 뛰어난 탐지 성능을 보였다. 이러한 성능은 시공간 특징을 효과적으로 학습하고, 적대적 학습을 통해 일반화를 향상시키며, CAN ID와 페이로드 정보를 통합하는 능력에 기인한다. 이러한 결과는 ST-AAE가 자동차 네트워크 보안 분야에서 효과적인 솔루션이 될 수 있음을 시사한다.

Abstract

The Controller Area Network (CAN), the in-vehicle network, is highly vulnerable to malicious physical and cyber attacks. To ensure the safety of both users and vehicle providers, appropriate security measures are necessary. This paper proposes a novel intrusion detection system (IDS) called Spatio-Temporal Adversarial Autoencoder (ST-AAE) for in-vehicle networks. ST-AAE utilizes the spatio-temporal characteristics of CAN traffic, a new approach in the field of in-vehicle IDS. The framework detects abnormal traffic and classifies attack types. Experimental results on real-world driving data show that ST-AAE outperforms existing models with high accuracy and low false positive rates across various attack types. Its performance is due to effective spatio-temporal feature learning, enhanced generalization through adversarial learning, and integration of CAN ID and payload information. These findings suggest that ST-AAE is an effective solution for automotive network security.

한글키워드 : 침입 탐지 시스템, 차량 네트워크 보안, CAN 버스, 오토인코더, 딥러닝

keywords : intrusion detection system, in-vehicle network security, control area network bus, autoencoder, deep learning

* 스마트엠투엠
† 교신저자: 강효은(email: hyoeun405@gmail.com)

접수일자: 2024.06.01. 심사완료: 2024.06.08.
게재확정: 2024.06.20.

1. 서 론

공유 모빌리티 산업의 최근 발전과 함께 카셰어링 시장은 빠른 성장세를 유지하고 있다[1]. 글로벌 시장조사기관인 맥킨지글로벌연구소에 따르면 2030년까지 카셰어링의 확산으로 인해 일반 소비자의 자동차 구매량은 연간 최대 400만 대 감소하고, 카셰어링 판매량은 200만 대 증가할 것으로 예측된다. 지속 가능한 도시 건설의 관점에서 볼 때, 자동차 공유에 사용되는 차량은 일반적으로 연료 효율이 높으며 도시 배출 감소 및 도시 혼잡에 긍정적인 효과를 가져온다[2].

그러나 이와 관련하여 자동차 산업의 보안 위협도 크게 증가할 것으로 예상된다[3][4]. 특히, 카셰어링 서비스를 통해 많은 사용자가 동일한 차량에 접근할 수 있어 로컬 공격자의 위협 시나리오가 증가한다. 이러한 공격은 장비와 시스템 간의 연결과 통신을 허용하는 CAN 프로토콜을 악용하여 이루어질 수 있다[5].

컨트롤러 영역 네트워크(CAN)는 차량 내 장치 통신을 위해 설계된 표준 차량 네트워크 프로토콜이다. CAN의 모든 전자 제어 장치(ECU)는 브로드캐스트 방식으로 메시지를 수신하며, 필요한 경우 메시지를 수신하고 그렇지 않은 경우 무시한다. CAN 통신 프로토콜에는 별도의 인증 또는 접근 제어 시스템이 없기 때문에 각 수신자는 발신자를 식별할 수 없으며, 수신된 패킷이 합법적인지 여부를 판단할 수 없다. CAN 버스에 대한 로컬 액세스 공격은 이러한 CAN 설계 취약점을 이용한다. 이로 인해 시스템의 CAN 버스에 있는 모든 데이터는 신뢰할 수 있는 것으로 간주되어 CAN의 공격자가 생성한 실제 오류 메시지와 가짜 오류 메시지를 구별할 수 없게 된다. 따라서 위조된 패킷이 CAN에 주입되거나 패킷 조작과 같은 악의적인 공격이 발생하는 경우 운전자의 생명에 치명적인 위협이 될 수 있다.

2015년 지프 체로키에서 자동차가 해킹된 사례가 있다. 해커들은 물리적 접근 없이 인터넷을 통해서만 원격으로 차량을 제어했다. 이 취약점이 알려진 후 지프는 140만 대의 체로키 차량을 리콜했다[6]. 또한, CAN 버스 프로토콜은 긴급 메시지(예: ABS)가 최단 시간 내에 처리될 수 있도록 메시지의 우선순위를 설정하여 특정 메시지가 다른 메시지보다 우선하도록 허용한다. 이 설계는 차량 애플리케이션에 매우 적합하지만, 동시에 DoS 공격에 취약점을 제공하기도 한다. 악의적인 공격자는 우선순위 기반 DoS 공격으로 정상적인 CAN 통신을 차단할 수 있다[7].

ECU 상태를 이용한 버스 오프(bus-off) 공격을 통해 ECU가 비정상적인 CAN 메시지를 감지할 수 없는 상태로 전환하는 스푸핑 공격 가능성도 있는 것으로 알려졌다[8]. 이러한 보안 문제를 최소화하기 위해 차량 내 침입 탐지 메커니즘을 반드시 적용해야 한다.

본 논문은 차량 내 침입 탐지 메커니즘으로서 딥러닝 기반의 CAN 버스 네트워크 침입 탐지 방법을 제안한다. 적대적 오토인코더(Adversarial Autoencoder)를 사용하여 차량 내 CAN 버스의 정상 상태를 학습하고, CAN ID와 CAN 페이로드를 동시에 분석 및 탐지한다. 오토인코더의 데이터 분포 학습 특성을 활용하여 CAN 버스 패킷의 비정상적인 패킷을 찾아내어 공격 유형을 분류한다. 실험은 정보보호 R&D 데이터 챌린지 데이터세트를 사용하여 성능을 입증한다.

2. 관련 연구

2.1 딥러닝 기반 CAN 네트워크 이상 탐지

차량 해킹에 대한 우려가 증가함에 따라 많은 연구자들이 차량 내 컨트롤러 영역 네트워크 위협 탐지에 주력하고 있다. 특히, 침입 탐지 시스

템(IDS)을 활용하여 위협을 탐지하는 것은 공격 탐지의 효율성과 단순성으로 인해 많은 관심을 받고 있다. 일반적으로 IDS를 사용한 CAN 위협 탐지에 관한 많은 연구 논문에서는 모델이 CAN을 통과하는 데이터 흐름을 통계적으로 학습하고, 임계값을 벗어난 신호를 통해 공격을 추론한다고 제안했다. 이러한 시스템은 CAN 특성(예: 트래픽의 시간 간격, 패킷의 빈도)을 활용하여 이상 징후를 식별한다. 예를 들어, 고정된 시간 간격으로 모든 ECU에 브로드캐스트되는 CAN 데이터가 정상 상태와 다른 경우 공격으로 분류될 수 있다.

Taylor 등은 LSTM(Long Short-Term Memory)을 사용한 이상 탐지 방법을 제안했다[9]. 이 방법은 CAN 버스에서 각 송신자가 보낸 다음 데이터를 예측하기 위해 정상 패킷 시퀀스를 학습하며, 실제 다음 데이터가 예측 값과 크게 차이가 날 때 이상을 감지한다. 이 기술은 오탐률이 낮다는 장점이 있지만, 단일 CAN ID에서만 작동한다는 단점이 있다.

GIDS는 Generative Adversarial Networks (GAN)을 기반으로 제안된 IDS 모델이다[10]. GIDS는 정상 데이터의 분포를 학습하고 분포에서 벗어나는 데이터를 공격으로 분류한다. 이 기술은 알 수 없는 공격을 높은 정확도로 탐지할 수 있다는 장점이 있다.

Deep Convolutional Neural Network(DCNN)을 기반으로 한 IDS도 있다[11]. 이 모델은 CAN 트래픽의 패킷 시퀀스 패턴을 학습하여 순차 패턴 변화를 식별함으로써 공격을 탐지한다. 구체적으로, DCNN 기반 IDS는 다양한 차량 모델에서 생성된 CAN 데이터세트를 사용하여 정상적인 패킷 시퀀스와 이상 패킷 시퀀스를 학습한다. 이 과정에서 모델은 패킷 간의 연관성과 시계열 패턴을 포착하여 정상적인 통신과 악의적인 활동을 구분할 수 있다.

3. 학습 데이터세트

본 논문에서는 국내에서 개최된 정보보호 R&D 데이터 챌린지 중 하나인 차량 내부 네트워크 침입 탐지 챌린지에서 제공하는 데이터세트를 활용한다[12][13]. 해당 챌린지는 차량 내 네트워크 통신의 표준으로 널리 사용되는 CAN 네트워크에 대한 공격 및 탐지 기술 개발을 목표로 진행되었다. 데이터세트는 현대 쏘나타, 기아 쏘울, 쉐보레 스파크 등 실제 차량에서 수집된 주

표 1. 데이터 필드 설명
Table 1. Data field description

필드명	설명
Timestamp	UNIX 타임스탬프로 기록된 시간 (단위: 초)
Arbitration_ID	16진수로 표현된 CAN 식별자
DLC	데이터 길이 코드 (1-8)
Data	최대 8바이트의 데이터를 포함하는 CAN 데이터 필드 (16진수로 표현되며, 각 바이트는 공백으로 구분됨)
Class	"Normal" 또는 "Attack"
SubClass	공격 유형 (Class가 "Attack"인 경우, "Flooding", "Spoofing", "Replay", "Fuzzing" 중 하나; Class가 "Normal"인 경우, "Normal")

행 데이터로 구성되어 있으며, 정상 상태의 데이터와 다양한 유형의 공격 데이터를 포함하고 있다. 공격 유형으로는 플러딩(Flooding), 퍼지(Fuzzy), 오작동(Malfunction), 리플레이(Replay) 공격 등이 있다.

데이터세트의 각 레코드는 표 1과 같이 타임스탬프(Timestamp), CAN ID(Arbitration ID), 데이터 길이 코드(DLC, Data Length Code), 페이로드(Data, CAN 데이터 필드) 등의 정보를 담고 있다.

차량 내부 네트워크에서의 플러딩 공격(Flooding attack)은 ECU가 CAN에서 데이터를 수신할 때, CAN ID의 우선순위에 따라 메시지를 먼저 수신한다는 점을 악용한다. CAN ID는 낮을수록 우선순위가 높은 특징을 지니고 있다. 플러딩 공격은 높은 우선순위의 메시지를 ECU에 지속적으로 전송하여 정상 메시지의 지연을 유발하거나, 더 나아가 전송 실패를 야기할 수 있다. 이는 정상적인 주행을 방해할 수 있다.

퍼지 공격(Fuzzy attack)은 무작위로 CAN ID를 선택하고 메시지를 전송한다. 선택된 ID가 존재하지 않는 경우, 즉 해당 CAN ID를 가진 차량이 없는 경우에는 네트워크에 큰 위험을 야기하지 않는다. 하지만 CAN 버스에 연결된 ECU의 ID가 선택된 경우, 차량의 장치가 비정상적으로 작동할 수 있다. 이 데이터세트에는 0x000부터 0x7FF까지의 값이 포함되어 있으며, 여기에는 차량에 실제로 존재하는 ECU의 CAN ID와 존재하지 않는 ID가 모두 포함된다.

오작동 공격(Malfunction attack)은 특정 ECU 값을 조작하여 차량이 비정상적으로 작동하도록 한다. 데이터는 차량에 존재하는 ECU의 CAN ID를 무작위로 선택하고, 차량이 오작동하도록 8바이트 범위 내에서 임의의 값을 입력하도록 구성된다.

4. 침입 탐지 모델 설계

4.1 적대적 오토인코더

적대적 오토인코더(Adversarial Autoencoder, AAE)는 생성 모델의 일종으로, 오토인코더(Autoencoder)와 적대적 생성 신경망(Generative Adversarial Network, GAN)의 개념을 결합한 모델이다[14]. 다음 그림 1은 일반적인 적대적 오토인코더 모델의 구조이다.

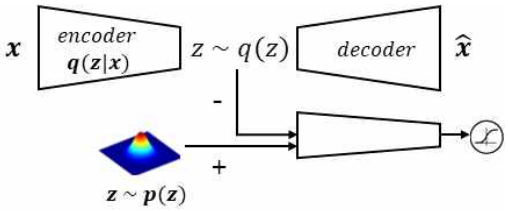

그림 1. 적대적 오토인코더 구조
Fig. 1. The structure of unsupervised AAE

그림 1과 같이 적대적 오토인코더는 인코더(Encoder), 디코더(Decoder), 그리고 판별자(Discriminator)의 세 가지 주요 구성요소로 이루어져 있으며, 이들 간의 상호작용을 통해 데이터의 잠재 표현(latent representation)을 학습한다.

인코더는 입력 데이터 x를 잠재 공간(latent space)의 저차원 벡터 z로 매핑하는 함수 $q(z|x)$로 정의된다. 이 과정은 다음 식 (1)과 같이 표현할 수 있다.

$$z = q(z|x) = f_{\phi}(x) \quad (1)$$

식 (1)에서 f_{ϕ}는 인코더 신경망을 나타내며, ϕ는 인코더의 파라미터를 의미한다.

디코더는 잠재 벡터 z를 입력으로 받아 원래의 입력과 유사한 데이터 x′를 재구성하는 함수 $p(x|z)$로 정의된다.

$$x' = p(x|z) = g_\theta(z) \quad (2)$$

식 (2)에서 g_θ는 디코더 신경망을 나타내며, θ는 디코더의 파라미터를 의미한다.

적대적 오토인코더의 목표는 인코더에서 생성된 잠재 벡터의 분포 $q(z)$를 사용자가 지정한 사전 분포 $p(z)$에 근사시키는 것이다. 이를 위해 판별자 D는 다음 식 (3)에 따라 이진 분류 문제를 수행한다.

$$D(z) = \sigma(d_\psi(z)) \quad (3)$$

식 (3)에서 d_ψ는 판별자 신경망, ψ는 판별자의 파라미터, 그리고 σ는 sigmoid 활성화 함수를 나타낸다.

적대적 오토인코더의 학습은 다음의 세 가지 손실 함수를 최적화하는 과정으로 이루어진다.

1) 재구성 손실 (Reconstruction Loss):

$$L_{rec} = E_{x' \sim p_{data(x)}}[\log p(x|z)] \approx \frac{1}{N}\sum_{i=1}^{N} \log p(x^i|z^i) \quad (4)$$

2) 인코더 손실 (Encoder Loss):

$$L_{enc} = -E_{z \sim q_{(z)}}[\log D(z)] \approx -\frac{1}{M}\sum_{j=1}^{M} \log D(z^j) \quad (5)$$

3) 판별자 손실 (Discriminator Loss):

$$L_{dis} = -E_{z \sim p_{(z)}}[\log D(z)] - E_{z \sim q_{(z)}}[\log 1 - D(z)] \approx -\frac{1}{M}\sum_{j=1}^{M}[\log D(z_p^j) + \log(1 - D(z_q^j))] \quad (6)$$

식 (4)~(6)의 $z^j{}_p$는 사전 분포 $p(z)$에서, $z^j{}_q$는 인코더 분포 $q(z)$에서 샘플링된 잠재 벡터를 의미한다.

재구성 손실 L_{rec}은 원래 데이터 x와 잠재 변수 z로부터 재구성된 데이터 x' 사이의 차이를 측정하는데 사용된다.

인코더 손실 L_{enc}은 인코더가 생성한 잠재 벡터 z를 판별자 D가 진짜로 인식하도록 하는데 사용된다.

인코더와 디코더는 재구성 손실 L_{rec}와 인코더 손실 L_{enc}을 최소화하도록 학습되며, 판별자는 판별자 손실 L_{dis}를 최소화하도록 학습된다. 이러한 적대적 학습을 통해 인코더는 데이터의 중요 특징을 포착하면서도 사전 정의된 분포를 따르는 잠재 표현을 학습하게 된다. 따라서 적대적 오토인코더의 최적화 목적 함수는 식 (7)과 같다.

$$\min_{\phi,\theta} L_{rec} + \lambda \cdot L_{enc} \quad \min_{\psi} L_{dis} \quad (7)$$

식 (7)에서 λ는 인코더 손실의 가중치를 조절하는 하이퍼파라미터이다. 학습 과정에서는 이 두 부분을 번갈아가며 최적화를 수행한다. 이를 통해 인코더는 입력 데이터를 잠재 공간으로 매핑하고, 디코더는 잠재 공간에서 원본 데이터를 재구성하며, 판별자는 인코더가 생성한 잠재 벡터와 사전 정의된 분포에서 샘플링된 벡터를 구분하도록 학습된다.

적대적 오토인코더는 오토인코더의 잠재 공간에 명시적인 분포 가정을 도입함으로써, 해석 가능하고 제어 가능한 표현 학습을 가능케 한다. 이는 생성 모델링, 이상 탐지, 차원 축소 등 다양한 응용 분야에서 활용될 수 있으며, 오토인코더 기반 표현 학습 연구의 새로운 방향을 제시하고 있다.

4.2 이상 탐지에서의 적대적 오토인코더

이상 탐지 관점에서 적대적 오토인코더의 잠재 벡터는 이상 징후를 식별하는 데 유용한 정보를 제공한다. 적대적 오토인코더는 정상 데이터의 압축된 표현을 학습하기 때문에, 정상 데이터는 높은 확률의 잠재 벡터로 변환되어 디코더에 의해 원래 데이터와 유사하게 재구성된다. 반면에 이상 데이터는 학습된 정상 데이터의 패턴에서 벗어나기 때문에 낮은 확률의 잠재 벡터로 변환되거나 높은 재구성 오차를 보일 가능성이 크다.

수식적으로 표현하면, 인코더 $q(z|x)$는 입력 데이터 x를 잠재 벡터 z로 매핑하고, 디코더 $p(x|z)$는 잠재 벡터 z로부터 입력 데이터를 재구성한다. 정상 데이터 x_{normal}의 경우 인코더-디코더 네트워크를 통과할 때 낮은 재구성 오차를 보이며, 이는 다음 식(8)과 같이 표현할 수 있다.

$$x_{normal} \approx p(x|q(z|x_{normal})) \quad (8)$$

반면, 이상 데이터 $x_{anomaly}$는 인코더-디코더 네트워크에서 높은 재구성 오차를 나타내며, 다음과 같이 표현할 수 있다.

$$x_{anomaly} \neq p(x|q(z|x_{anomaly})) \quad (9)$$

또한, 정상 데이터는 학습 과정에서 인코더에 의해 높은 확률을 갖는 잠재 공간 영역으로 매핑되는 반면, 이상 데이터는 학습된 패턴에서 벗어나 낮은 확률의 영역으로 매핑된다.

적대적 오토인코더를 사용하여 정상 데이터의 압축된 표현을 학습한 후, 테스트 데이터를 인코더-디코더에 통과시켜 재구성 오차를 계산하거나 잠재 공간에서의 밀도를 추정하여 이상 탐지를 수행할 수 있다. 정상 데이터와 유사한 패턴을 가진 데이터는 낮은 재구성 오차와 높은 잠재 공간 밀도를 보이지만, 이상 데이터는 높은 재구성 오차와 낮은 잠재 공간 밀도를 나타낸다.

$$L_{anomaly} = \|x - p(x|q(z|x))\|^2 + \lambda \cdot [-\log q(z|x)] \quad (10)$$

위 식 (10)에서 첫 번째 항은 재구성 오차를, 두 번째 항은 잠재 공간에서의 밀도를 추정하는 항이며, λ는 두 항의 상대적 중요도를 조절하는 하이퍼파라미터이다. 재구성 오차가 크거나 잠재 공간에서의 밀도가 낮은 경우, 해당 데이터는 이상 데이터로 간주될 수 있다.

4.3 시공간 적대적 오토인코더 설계

본 연구에서는 자동차의 CAN 버스 패킷 데이터를 분석하여 비정상적인 공격을 탐지하기 위한 새로운 방법인 시공간 적대적 오토인코더(Spatio-Temporal Adversarial Autoencoder, ST-AAE)을 제안한다. ST-AAE는 자동차가 정상적인 주행 상태일 때의 CAN 패킷 데이터를 사용하여 시간과 공간 정보를 함께 고려한 데이터 표현을 학습한다.

ST-AAE 모델은 그림 2와 같이 인코더와 디코더, 그리고 두 개의 판별자(Discriminator)으로 구성된다.

입력되는 CAN 페이로드 시퀀스를 $\{x_t\}$라고 하면, 인코더의 목표는 각 시간 단계 t에서의 CAN 페이로드 x_t를 시공간적 특성을 반영한 잠재 표현 z_t로 매핑한다. 이 과정을 통해 인코더는 입력 CAN 페이로드의 공간적, 시간적 특성을 학습한다. 인코더는 다음 식 (11)과 같이 표현할 수 있다.

$$z_t = Enc(x_t, h_{t-1}) \quad (11)$$

여기서 h_{t-1}은 이전 시간 단계의 인코더 hidden state를 나타낸다.

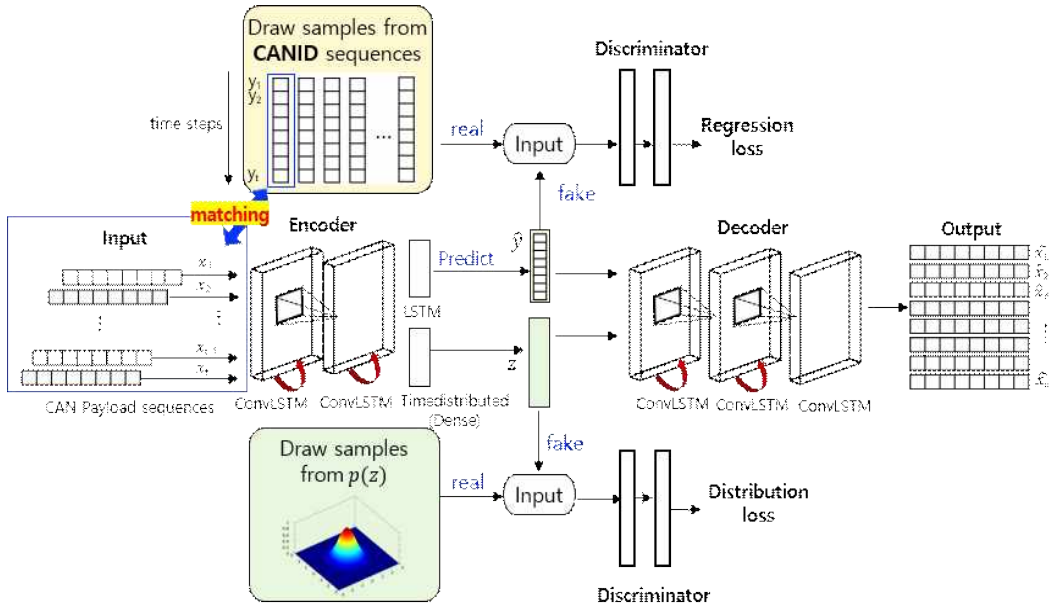

그림 2. ST-AAE 네트워크 구조
Fig. 2. Network architecture of the ST-AAE

인코더에서 생성된 잠재 표현 z_t는 두 개의 잠재 벡터로 분리된다. 이는 다음과 같이 나타낼 수 있다.

$$z_t = f(x_t)$$
$$\hat{y_t} = f(g(z_t)) \quad (12)$$

첫 번째 잠재 벡터 z_t는 인코더에 의해 입력 데이터에서 추출된 특징을 압축적으로 표현한다. 두 번째 잠재 벡터 $\hat{y_t} = (\hat{y_1}, \hat{y_2}, ..., \hat{y_3})$는 시간 단계 t에 해당하는 CAN ID를 나타낸다.

디코더는 이 두 잠재 벡터를 이용해 원래의 CAN 페이로드를 재구성한다.

$$x'_t = Dec(\hat{y_t}, z, h'_{t-1}) \quad (13)$$

여기서 h'_{t-1}은 이전 시간 단계의 디코더 hidden state이다.

ST-AAE의 손실 함수는 크게 세 부분으로 구성된다.

1) 재구성 손실(Reconstruction Loss):

$$L_{DATA_t} = \|x'_t - x_t\|_{l_2}^2 \quad (14)$$

ST-AAE의 재구성 손실은 입력 CAN 페이로드와 재구성된 페이로드 간의 차이를 평균 제곱 오차(MSE)로 측정한다.

2) CAN ID 예측 손실(ID Prediction Loss):

$$L_{ID_t} = \|\hat{y_t} - y_t\|_{l_2}^2 \quad (15)$$

ST-AAE의 CAN ID 예측 손실 함수는 모델이 예측한 CAN ID와 실제 CAN ID 사이의 차이를 측정한다.

3) 적대적 손실(Adversarial Losses):

$$L_{adv_y} = E[\log Dis_y y_{real}] + E[\log(1 - Dis_y y_{fake})] \quad (16)$$

$$L_{adv_s} = E[\log Dis_s s_{real}] + E[\log(1 - Dis_s s_t)] \quad (17)$$

ST-AAE의 적대적 손실 함수는 모델이 생성한 CAN ID와 스타일 정보가 각각의 사전 정의된 분포를 따르도록 유도한다.

최종 손실 함수는 위의 세 손실 함수를 조합하여 다음과 같이 나타낼 수 있다.

$$L_{total} = \Sigma_t (L_{DATA_t} + \alpha^* L_{ID_t}) + \beta^* (L_{adv_y} + L_{adv_s}) \quad (18)$$

식 (18)에서 α와 β는 각 손실 항의 중요도를 조절하는 하이퍼파라미터이다.

학습이 완료된 후, ST-AAE는 재구성 손실과 CAN ID 예측 손실의 합을 이상 점수로 사용하여 새로운 CAN 데이터의 이상 여부를 판단한다. 이상 점수가 높을수록 해당 데이터가 정상 패턴에서 벗어났을 가능성이 큼을 나타낸다.

5. 실 험

5.1 실험 방법

본 연구에서 제안하는 ST-AAE 모델은 CAN 패킷의 시계열 데이터를 입력으로 받아 공격을 탐지한다. 각 CAN 패킷은 CAN ID와 최대 8바이트의 페이로드 데이터를 포함한다.

아래 표 2는 Replay Attack 레이블이 적용된 CAN 패킷을 2개의 연속적인 time step으로 표현한 예시이다.

CAN ID는 메시지의 우선순위와 송신자를 식별하는 역할을 하며, 16진수의 기본형 식별자로 구성된다. ST-AAE 모델에서는 CAN ID를 one-hot encoding하여 사용한다.

CAN Data 필드는 최대 8바이트의 페이로드를 포함하며, 각 바이트는 0x00부터 0xFF까지의 값을 가질 수 있다. 모델에서는 이를 16진수 형태로 입력받아 처리한다.

데이터 길이 코드(DLC)는 CAN 데이터 필드의 바이트 수를 나타내는 필드로, 0부터 8까지의 값을 가진다.

표 2. Replay Attack CAN 패킷 예시
Table 2. Example of replay attack CAN packet

Timestamp	CAN ID	DLC	Data	Attack
1513925348.249034	0690	8	00 00 01 00 80 22 00 00	R
1513925348.250425	05F0	2	00 00	R

ST-AAE 모델의 입력은 아래 그림 3과 같이 16개의 연속된 time step에 해당하는 CAN 패킷들로 구성된다. 따라서 ST-AAE 모델의 입력 데이터는 (16, 8, 1)의 형태를 가지며, 16은 time step의 수, 8은 각 CAN 패킷의 최대 데이터 길

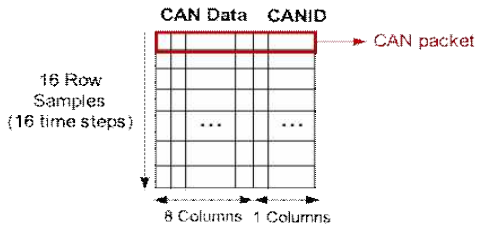

그림 3. 입력 데이터 구조
Fig. 3. Architecture of inputs

이, 1은 데이터의 차원을 의미한다. 이러한 입력 데이터를 기반으로 ST-AAE 모델은 각 time step에서의 CAN 패킷을 재구성하고, 정상 패킷과 공격 패킷을 구분할 수 있는 잠재 공간 표현을 학습하게 된다.

5.2 실험 환경

본 연구에서 제안하는 ST-AAE 모델의 성능을 평가하기 위해 다음과 같은 실험 환경을 구성하였다. 모델 학습을 위해 NVIDIA V100 GPU (32GB) 2대를 사용하였으며, 256GB의 메모리를 할당하였다. 딥러닝 프레임워크로는 파이토치(PyTorch)[15]를 사용하였다.

ST-AAE 모델 구현을 위해 ConvLSTM (Convolutional Long Short-Term Memory) 구조를 활용하였다[16]. ConvLSTM은 연속적인 이미지 프레임과 같은 2차원 데이터의 시공간적 특징을 효과적으로 학습할 수 있는 구조로, 시간에 따라 변하는 패턴을 포착하고 예측해야 하는 ST-AAE 모델에 적합한 선택이라 할 수 있다.

ST-AAE 모델 학습 시 배치 크기(batch size)는 16으로 설정하였으며, 총 50 에폭(epoch)동안 학습을 진행하였다. 최적화 알고리즘으로는 Adam optimizer를 사용하였다. Adam optimizer는 적응적 학습률(adaptive learning rate)을 적용하여 효과적으로 모델을 최적화할 수 있는 알고리즘으로 알려져 있다[17].

5.3 실험 결과

본 논문에서는 ST-AAE 모델의 성능을 평가하기 위해 두 가지 기준 모델을 선정하였다.

1) LSTM-VAE (Long Short-Term Memory Variational Autoencoder) [18]: LSTM-VAE 네트워크는 일정 시간 동안의 CAN ID 시퀀스를 슬라이딩 윈도우 방식으로 입력받아 재구성하는 작업을 수행한다.

2) CNN-VAE (Convolutional Neural Network Variational Autoencoder) [19]: CNN-VAE는 일정 시간 동안의 CAN 페이로드를 이어 붙여 입력으로 사용하여 페이로드의 분포를 파악한다. 구체적으로는 16×8×1 크기의 데이터(시간 단계, 페이로드 속성(8바이트), 채널)를 입력으로 제공한다.

아래 그림 4는 세 가지 모델(ST-AAE, LSTM-VAE, CNN-VAE)의 정상 패킷 분류 정확도를 비교한 결과를 보여준다. ST-AAE 모델의 정확도는 0.995로, LSTM-VAE (0.990)와 CNN-VAE (0.981)보다 높은 성능을 보인다. 이 결과는 ST-AAE 모델이 CAN 버스 네트워크의 정상 패킷을 더욱 효과적으로 분류할 수 있음을 나타낸다.

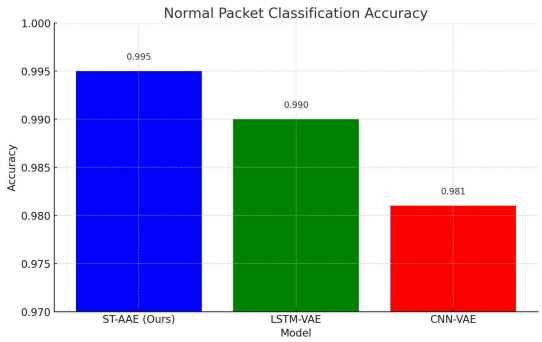

그림 4. 정상 패킷 분류 정확도

Fig. 4. Normal packet classification accuracy

아래 그림 5는 4가지 공격 유형에 대해 ST-AAE, LSTM-VAE, CNN-VAE 모델의 True Positive Rate (TPR)과 True Negative Rate (TNR)을 비교하였다.

ST-AAE 모델은 Flooding Attack에 대해 0.981의 TPR과 0.997의 TNR을 기록하여, 다른 두 모델에 비해 매우 높은 성능을 보였다. 이는 ST-AAE 모델이 Flooding Attack을 보다 효과

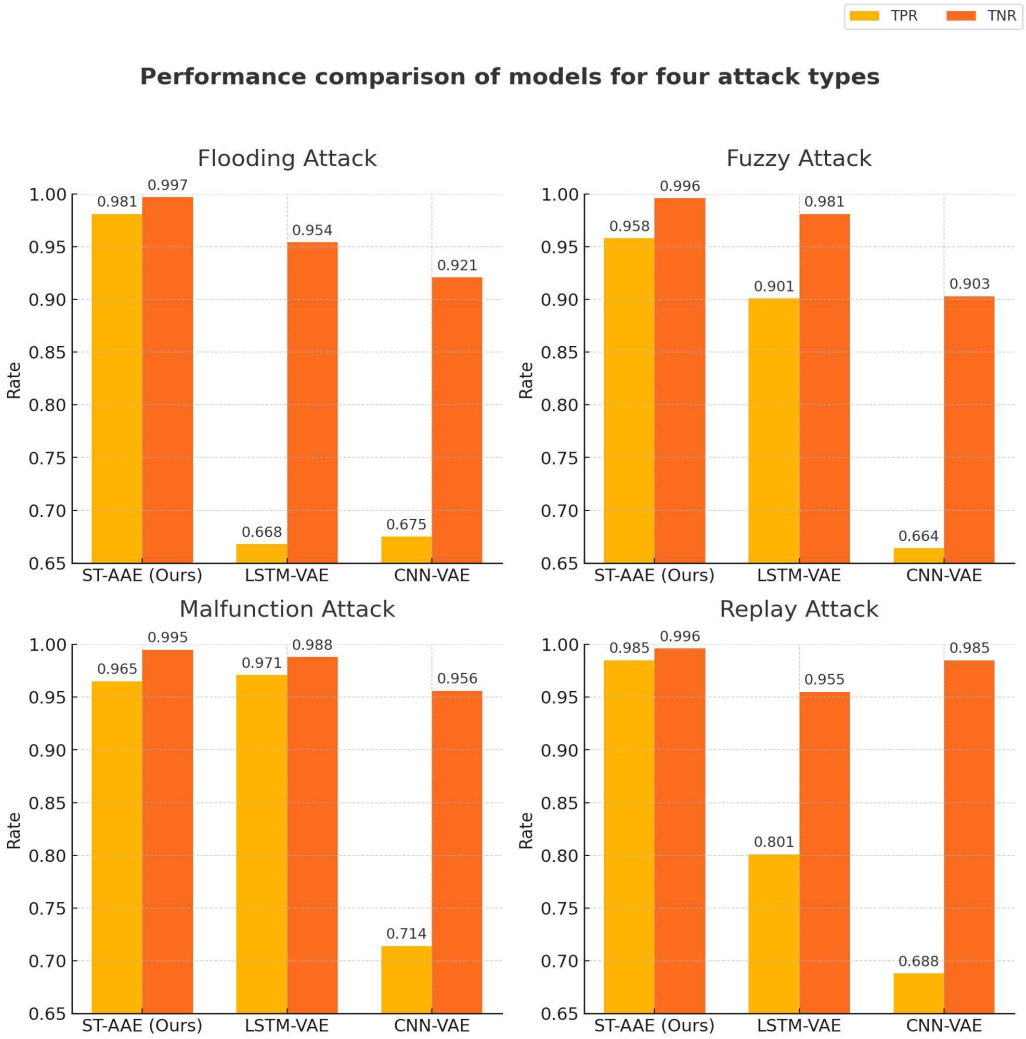

그림 5. 4가지 공격 유형에 대한 모델 성능 비교

Fig. 5. Performance comparison of models for four attack types

적으로 탐지하고, 정상 상태를 더 잘 유지함을 시사한다.

Fuzzy Attack의 경우에도 ST-AAE 모델은 0.958의 TPR과 0.996의 TNR을 기록하여, LSTM-VAE와 CNN-VAE 모델보다 우수한 성능을 보였다. 특히, CNN-VAE 모델은 Fuzzy Attack에 대해 상대적으로 낮은 탐지 성능을 보였다.

Malfunction Attack의 경우, ST-AAE 모델은 0.965의 TPR과 0.995의 TNR로 우수한 성능을 보여주었으나, TPR에서는 LSTM-VAE 모델에 비해 상대적으로 낮은 성능을 보인다.

마지막으로, Replay Attack의 경우, ST-AAE 모델은 0.985 TPR과 0.996 TNR로 최고의 성능을 기록하였다. 반면, LSTM-VAE 모델과 CNN-VAE 모델은 상대적으로 낮은 성능을 보

였으며, 특히 CNN-VAE 모델은 매우 낮은 TPR을 나타냈다.

본 논문의 실험 결과는 ST-AAE (Spatio-Temporal Adversarial Autoencoder) 모델이 다양한 공격 유형에 대해 일관되게 높은 탐지 성능(TPR)과 낮은 오탐률(TNR)을 유지함으로써, 다른 모델들에 비해 우수한 보안 성능을 제공함을 입증한다. 구체적으로, ST-AAE 모델은 Flooding Attack, Fuzzy Attack, Malfunction Attack, Replay Attack 등 여러 유형의 공격에 대해 높은 TPR과 TNR을 기록했다. 이러한 결과는 제안된 ST-AAE 모델이 차량 내부 네트워크 침입에 대한 효과적인 방어 수단으로서의 가능성을 제시한다.

5. 결 론

본 논문에서는 자동차 내부 네트워크인 CAN 버스에서 발생하는 다양한 유형의 공격을 탐지하기 위한 새로운 방법으로 ST-AAE (Spatio-Temporal Adversarial Autoencoder)를 제안하였다. ST-AAE는 ConvLSTM 구조를 활용하여 CAN 데이터의 시공간적 특징을 학습하고, 적대적 학습을 통해 모델의 일반화 능력을 향상시킨다.

ST-AAE의 인코더는 CAN 패킷의 잠재 표현을 추출하여 시공간 특성을 포착하고, 이를 기반으로 디코더는 원본 데이터를 재구성한다. 이 과정에서 발생하는 재구성 손실과 회귀 손실을 통해 모델은 정상 패킷과 비정상 패킷을 구분할 수 있는 능력을 학습한다. 또한, ST-AAE는 CAN ID와 페이로드 정보를 모두 활용하여 데이터의 특징을 통합적으로 고려할 수 있다.

ST-AAE 모델의 성능을 평가하기 위해, 한국정보보호 R&D 데이터 챌린지의 차량 내부 네트워크 침입 탐지 대회에서 제공된 실제 주행 차량 데이터를 사용하였다. 실험 결과, ST-AAE는 기존의 LSTM-VAE와 CNN-VAE 모델에 비해 우수한 탐지 성능을 보였으며, 다양한 유형의 공격에 대해 높은 정확도와 낮은 오탐율을 달성하였다.

ST-AAE의 우수한 성능은 다음과 같은 요인에 기인한다. 첫째, ConvLSTM 구조를 통해 시공간 특징을 효과적으로 학습함으로써 시간에 따른 패턴 변화와 공간적 상관관계를 동시에 고려할 수 있다. 둘째, 적대적 학습을 통해 모델의 일반화 능력을 향상시켜 새로운 유형의 공격에 대해서도 강인한 탐지 성능을 보인다. 셋째, CAN ID와 페이로드 정보를 통합적으로 활용함으로써 단일 정보만을 사용하는 기존 모델에 비해 더 풍부한 정보를 활용할 수 있다.

종합하면, 본 논문에서 제안한 ST-AAE 모델은 자동차 네트워크 보안 분야에서 효과적인 이상 탐지 솔루션이 될 수 있음을 보여준다. 향후 연구에서는 ST-AAE 모델을 실제 자동차 환경에 적용하고, 다양한 공격 시나리오에 대한 검증을 수행할 예정이다. 또한, 모델의 경량화 및 최적화를 통해 실시간 탐지 성능을 향상시키는 방안을 모색할 계획이다.

참 고 문 헌

[1] F. Ferrero, G. Perboli, M. Rosano, and A. Vesco, “Car-sharing services: An annotated review”, Sustainable Cities and Society, Vol. 37, pp. 501-518, 2018, DOI : 10.1016/j.scs.2017.09.020.

[2] E. W. Martin and S. A. Shaheen, “Greenhouse Gas Emission Impacts of Carsharing in North America”, IEEE Transactions on Intelligent Transportation

Systems, Vol. 12, No. 4, pp. 1074-1086, Dec. 2011, DOI : 10.1109/tits.2011.2158539.

[3] S. Woo, H. J. Jo, and D. H. Lee, “A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN”, IEEE Transactions on Intelligent Transportation Systems, Vol. 15, No. 1, pp. 1-14, 2014, DOI : 10.1109/tits.2014.2351612.

[4] F. Fakhfakh, M. Tounsi, and M. Mosbah, “Cybersecurity attacks on CAN bus based vehicles: a review and open challenges”, Library Hi Tech, Vol. 40, No. 5, pp. 1179-1203, 2021, DOI : 10.1108/lht-01-2021-0013.

[5] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, “Evaluation of CAN Bus Security Challenges”, Sensors, Vol. 20, No. 8, pp. 2364, Apr. 2020, DOI : 10.3390/s20082364.

[6] A. Greenberg, “Hackers remotely kill a Jeep on the highway－with me in it”, Wired, Jul. 21, 2015. https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[7] S. Chen, C. H. R. Lin, “Evaluation of DoS Attacks on Vehicle CAN Bus System, Smart Innovation, Systems and Technologies,” Springer International Publishing, pp.308-314, Nov. 2018, DOI : 10.1007/978-3-030-03748-2_38

[8] K. Iehira, H. Inoue and K. Ishida, “Spoofing attack using bus-off attacks against a specific ECU of the CAN bus”, Proceedings of the 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), pp. 1-4, Las Vegas, NV, USA, Jan. 2018, DOI : 10.1109/CCNC.2018.8319180.

[9] A. Taylor, S. Leblanc, and N. Japkowicz, “Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks”, Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 130-139, Montreal, QC, Canada, Oct. 2016, DOI : 10.1109/DSAA.2016.20.

[10] E. Seo, H. M. Song, and H. K. Kim, “GIDS: GAN based Intrusion Detection System for In-Vehicle Network”, Proceedings of the 2018 16th Annual Conference on Privacy, Security and Trust (PST), pp. 1-6, Belfast, Ireland, Aug. 2018, DOI : 10.1109/PST.2018.851.

[11] M. Delwar Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, “An Effective In-Vehicle CAN Bus Intrusion Detection System Using CNN Deep Learning Approach”, Proceedings of the GLOBECOM 2020 - 2020 IEEE Global Communications Conference, pp. 1-6, Taipei, Taiwan, Dec. 2020, DOI : 10.1109/GLOBECOM42002.2020.9322395.

[12] H. Kang, B. I. Kwak, Y. H. Lee, H. Lee, H. Lee, H. K. Kim, Car Hacking and Defense Competition on In-Vehicle Network, Proceedings Third International Workshop on Automotive and Autonomous Vehicle Security, Internet Society, 2021, DOI : 10.14722/autosec.2021.23035

[13] H. K. Kim, “Car Hacking: Attack & Defense Challenge 2020 Dataset”, IEEE, Feb. 2021, https://dx.doi.org/10.21227/qvr7-n418

[14] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow, “Adversarial Autoencoders,” Proceedings of the ICLR Workshop, San Juan, Puerto Rico, May 2016., https://doi.org/10.48550/arXiv.1511.05644

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., “Pytorch: An imperative style, high-performance deep learning library”, Proceedings of the 33rd International Conference on Neural Information Processing Systems, No. 721, pp. 8026 - 8037, Vancouver, BC, Canada,

Dec. 2019, https://api.semanticscholar.org/CorpusID:202786778
[16] X. Shi, Z. Chen, H. Wang, D. Yeung, W. K. Wong, and W. C. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting”, Advances in Neural Information Processing Systems, vol. 28, pp. 802-810, Dec. 2015., DOI : 10.5555/2969239.2969329
[17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, Proceedings of the International Conference on Learning Representations (ICLR), Poster, 2015, http://arxiv.org/abs/1412.6980
[18] D. Park, Y. Hoshi, and C. C. Kemp, “A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder”, IEEE Robotics and Automation Letters, Vol. 3, No. 3, pp. 1544-1551, Jul. 2018, DOI : 10.1109/lra.2018.2801475.
[19] D. P. Kingma, M. Welling, “Auto-encoding variational bayes”, Proceedings of the 2nd International Conference on Learning Representations (ICLR), Conference Track Proceedings, Banff, AB, Canada, Apr. 14-16, 2014, https://arxiv.org/abs/1312.6114v10

저 자 소 개

강효은(Hyo-Eun Kang)

2017.2 부산대학교 IT응용공학과 졸업
2022.8 부산대학교 정보융합공학과 석사
2024.2 부산대학교 정보융합공학과 박사
2023.6-현재 : SmartM2M 선임 연구원
<주관심분야> 초거대 언어 모델, 딥러닝, 머신러닝, AI 보안