JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 24, No. 11, pp. 2871-2879, Nov. 2023

Check for updates

젯슨 나노를 이용한 양방향 GPT 기반 CAN IDS 실시간 구현

김 송 목[1] · 박 승 영[2*]
[1]강원대학교 전기전자공학과 학사과정
[2*]강원대학교 전기전자공학과 교수

Real-Time Implementation of Bi-directional GPT-based CAN IDS Using Jetson Nano

Song-Mok Kim[1] · Seungyoung Park[2*]

[1]Bachelor's Course, Department of Electrical and Electronics Engineering, Kangwon National University, Chuncheon 24341, Korea
[2*]Professor, Department of Electrical and Electronics Engineering, Kangwon National University, Chuncheon 24341, Korea

[요 약]

V2X (vehicle-to-everything) 통신 기술의 발전은 생활의 편의성을 향상시켰지만, 동시에 해킹 위협은 더욱 대두되고 있다. 차량은 V2X 통신을 통해 외부와 내부 디바이스 간 정보를 주고받는데, 이 과정에서 보안 취약점을 가지고 있는 CAN (controller area network) bus를 사용하게 된다. 본 논문에서는 CAN bus 통신의 보안 취약성을 해결하기 위해 제안되었던 양방향 GPT (generative pre-trained transformer) 기반의 CAN ID (identifier) 시퀀스 이상 탐지 기법을 저사양 하드웨어인 젯슨 나노 보드를 이용하여 구현하였다. 구체적으로, 탐지 대상 시퀀스를 생성할 때, 이동 윈도우의 보폭을 적절히 조정함으로써, 저사양 하드웨어인 젯슨 나노에서도 실시간 이상 탐지가 가능함을 보였다. 이러한 저사양 하드웨어를 이용한 실시간 구현은 차량 내 통신 보안 강화를 위한 효율적이고 경제적인 방법을 제시하며, 다양한 차량 환경에서의 적용 가능성을 제시한다.

[Abstract]

The emergence of connected vehicular technologies via vehicle-to-everything (V2X) communication has significantly enhanced user convenience. However, these technologies are susceptible to critical security vulnerabilities. Notably, data transmitted via the controller area network (CAN) bus, which was not designed with security in focus, is particularly susceptible to threats. The CAN bus's absence of standard security measures permits unauthorized devices to transmit malicious data. A distinctive approach using a bi-directional Generative Pre-trained Transformer (GPT) has been introduced for anomaly detection in CAN identifier (ID) sequences, demonstrating superior performance over conventional methods. This study delves into the real-time deployment of this GPT-based detection system on the Jetson Nano board. By adjusting the moving window's stride, the system exhibited stable operation, even on low-spec hardware such as Jetson Nano. This approach offers cost-effective solutions for embedded systems and holds potential for broader vehicular communication protocols, thus ensuring reliable anomaly detection across various vehicular environments.

색인어 : 컨트롤러영역네트워크, 침입탐지시스템, 양방향GPT, 실시간처리, 젯슨 나노
Keyword : Controller Area Network, Intrusion Detection System, Bi-directional GPT, Real Time Processing, Jetson Nano

http://dx.doi.org/10.9728/dcs.2023.24.11.2871

Received 25 September 2023; **Revised** 19 October 2023
Accepted 19 October 2023

***Corresponding Author; Seungyoung Park**

Tel: +82-33-250-6326
E-mail: s.young.park@kangwon.ac.kr

Ⅰ. 서 론

V2X (vehicle-to-everything) 통신을 통한 자동차의 커넥티드 기술의 등장으로 인해 생활의 편리함은 증대되었다. 하지만, 이 기술의 취약점을 공격하는 해킹의 위협은 사람들의 안전과 생명에 직결되기에 이에 대한 보안의 중요성은 더욱 대두되고 있다[1]-[4]. 특히, V2X 통신은 차량의 통신 장치를 통해 외부로부터 정보를 수신할 뿐만 아니라 내부의 정보를 전송하기도 한다. 이때, 외부 통신 장치와 차량 내 전자 장치들이 CAN (controller area network) bus로 연결되어 있기 때문에 차량 외부로부터의 공격은 차량 내부에 치명적인 악영향을 끼칠 수 있다.

차량 내 통신을 위해 설계된 CAN bus 통신에서는 CAN 데이터 프레임을 이용하여 메시지가 전송된다. 발신 및 수신 주소가 메시지와 함께 구성된 이더넷 패킷과는 달리 CAN 데이터 프레임은 CAN ID (identifier)와 메시지만으로 구성되어 있으며, 임의의 ECU (electrical control unit) 혹은 컨트롤러는 자신이 전송하려는 메시지를 방송하므로 bus에 연결된 모든 장치에서 수신할 수 있다. 따라서, 각 ECU는 방송된 CAN 데이터 프레임들 중 특정 CAN ID를 가진 것만을 수신한다. CAN bus 통신이 개발될 당시에는 보안에 대한 고려 없이 설계되었기에 심각한 보안 취약점이 존재한다[4]. 구체적으로, CAN bus 통신에서는 접근 제어 혹은 인증 등의 메커니즘이 없기 때문에 CAN bus에 허가받지 않은 장치가 접근하여 악의적인 데이터를 전송할 수 있다. 예를 들어, 공격자는 CAN ID를 조정하여 특정 공격 의도를 가진 악의적인 메시지를 특정 ECU로 손쉽게 전송할 수 있다.

정상적인 차량 운행 상황에서 CAN bus에 연결된 ECU들은 주기적인 CAN 신호와 비주기적인 CAN 신호를 전송한다. 이때, 전송되는 CAN 신호의 ID를 순차적으로 수집하면 CAN ID 시퀀스를 구성할 수 있다. 이 시퀀스는 정상적인 운행 상황에서 수집되었으므로 일정한 패턴을 갖고 있을 것이다. 딥러닝을 이용하여 이러한 CAN ID 시퀀스의 일반적인 패턴을 학습한다면, 공격이 발생으로 인한 패턴의 변화를 탐지할 수 있을 것이다. 이러한 공격 탐지 기법 중 하나로, GPT (generative pre-trained transformer) 네트워크 구조를 양방향으로 결합한 구조를 이용하여 CAN ID 시퀀스의 이상을 탐지하는 비지도 기반 기법이 제안되었다[5]. 이 기법은 주어진 CAN ID에 대하여 정방향 및 역방향 예측을 결합하여 예측함으로써, 기존의 비지도 기반 이상 탐지 기법들에 비해 성능이 향상됨을 보여주었다.

본 논문에서는 저사양 하드웨어인 젯슨 나노 보드를 활용하여 기존에 제안된 양방향 GPT 기반의 CAN ID 시퀀스 이상탐지 시스템을 실시간으로 구현하였다. 구현된 시스템은 차량에서 발생한 CAN 신호를 보드에 입력하고, CAN ID 시퀀스를 구성하여 이에 대한 이상을 탐지할 수 있도록 구성하였다. 이때, 젯슨 나노 보드는 CAN 통신을 지원하지 않기 때문에 보드의 USB 포트에 CANable Pro를 연결하여 CAN 신호를 수신하였다[6]. 실시간 이상 탐지를 수행하기 위해 i) CAN 신호를 수신하고 CAN ID를 추출하는 프로세스, ii) CAN ID를 저장하는 프로세스, iii) 그리고 이를 일정한 길이로 읽어 들여 이상 여부를 추론하는 프로세스들이 병렬로 동작하도록 구성하였다. 구체적인 동작 과정은 다음과 같다.

보드로 수신된 CAN 신호는 CAN ID만을 추출하여 인메모리 데이터베이스인 Redis 서버에 저장하였고[7], 이를 순차적으로 읽어 들여 길이가 K인 CAN ID 시퀀스를 생성하였다. 양방향 GPT 네트워크는 해당 시퀀스에 대한 NLL (negative log likelihood) 값을 계산하였고, 이 값을 문턱치와 비교하여 이상 여부를 판정하였다. 이때, CAN ID 시퀀스는 이동 윈도우 (sliding window) 와 보폭 (stride) 개념을 활용하여 생성하고 이상 여부를 판정하였다. 이동 윈도우의 크기가 K이고, 보폭이 k라 가정했을 때, 보폭 k가 작을 경우 CAN ID 당 탐지 시도 횟수 (= $\lfloor K/k \rfloor$) 가 증가하여 처리 지연이 증가하는 단점이 있으나 CAN ID 당 탐지 횟수가 증가하면 미탐 확률이 감소하는 장점이 있다. 반면, 보폭 k가 클 경우 처리 지연 시간이 감소하지만 미탐 확률이 증가하는 단점이 있다. 이러한 상충 관계를 고려하여 실험을 통해 적절한 보폭 k를 결정하였다.

본 논문의 구성은 다음과 같다. 2장은 CAN bus 프로토콜의 구성과 보안 취약점에 대해 논하며 이상 탐지에 사용할 CAN ID를 포함한 전체적인 CAN 데이터 프레임 구조를 설명한다. 3장은 이상 탐지 기법에 사용될 양방향 GPT의 구조에 관하여 설명한다. 4장은 CAN ID 시퀀스를 생성하기 위한 이동 윈도우 및 보폭에 대해 설명하고, 젯슨 나노 보드를 이용해 실시간으로 구현한 CAN IDS의 구조를 설명한다. 5장은 실험 구성 및 조건을 설명하고 양방향 GPT 기법 및 다른 기법의 검출 성능과 종단 간 지연 성능을 평가한다. 마지막으로 6장에서 결론을 제시한다.

Ⅱ. CAN Bus 프로토콜 개요

1983년 Bosch에 의해 개발된 CAN bus 통신은 차량 내의 ECU 간 통신을 위한 표준 통신 규격이다. CAN bus 통신은 단순하면서 효율적인 구조를 갖고 있기 때문에 1993년에 이를 ISO 11898 표준으로 제정하였다 [1]. CAN bus는 한 장치에서 메시지를 전송할 경우 버스에 연결되어 있는 모든 장치가 수신할 수 있는 방송 방식으로 동작한다. 이때, CAN ID와 메시지로 구성된 CAN 데이터 프레임이 전송되며, CAN bus에 연결된 임의의 장치는 자신에게 필요한 특정 CAN ID에 해당되는 CAN 데이터 프레임만을 수신한다.

그림 1은 CAN 데이터 프레임과 CAN ID 시퀀스의 구조를 나타낸다. CAN의 통신 프로토콜에서 사용되는 프레임 구조는 SOF (start of frame), ID field, (RTR) remote

transmission request, control field, data field, CRC (cyclic redundancy code), ACK (acknowledgement), EOF (end of frame) 로 구성된다. CAN ID는 데이터 프레임 내부의 ID field에서 11bits를 사용하므로, 0x000부터 0x7FF까지 최대 2,048개의 CAN ID를 지원할 수 있다. 또한, 모든 데이터 프레임의 CAN ID 값에 따라 전송의 우선순위가 정해지는데 ID 값이 낮을수록 전송 우선권을 갖는다.

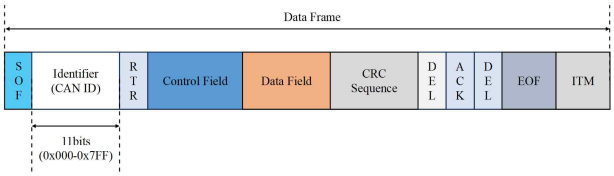

그림 1. CAN 데이터 프레임 구조
Fig. 1. Structure of CAN data frame

구체적으로, 서로 다른 CAN ID를 사용하는 임의의 ECU들이 메시지를 동시에 전송되는 경우, CAN ID에 대하여 자신이 전송하는 CAN ID와 다른 ECU가 전송하는 CAN ID를 비트 단위로 관찰하여 상대방이 자신보다 우선순위가 높은 메시지를 전송하는지를 확인할 수 있다. 이를 통해, 우선순위가 낮은 CAN 데이터 프레임을 전송하는 ECU는 CAN ID가 전송되는 구간 동안 우선순위가 높은 CAN 데이터 프레임이 전송되고 있음을 감지할 수 있으며, 해당 CAN 데이터 프레임 전송이 완료될 때까지 대기한 후 자신의 메시지를 전송함으로써 충돌 없이 전송할 수 있다.

불행히도, CAN bus 프로토콜이 개발될 당시에는 보안에 대한 고려가 전혀 이루어지지 않았기 때문에 암호화나 인증과 같은 기능이 전혀 포함되지 않았다. CAN bus에는 ECU들뿐만 아니라, Bluetooth, 3G/4G, Wi-Fi, 무선 통신 센서, GPS (global positioning system), 차량 제어 장치 등 많은 장치가 병렬로 연결된다. 즉, 안전과는 상관없는 외부 장치들과 안전에 직결된 ECU들이 보안이 전혀 고려되지 않은 하나의 물리적인 bus에 연결되므로, 외부로부터의 CAN bus 공격에 매우 취약하다[4].

이와 관련하여, 그림 2는 속도와 방향을 제어하는 안전과 직결된 ECU들이 USB와 Wi-Fi를 제어하는 장치들과 하나의 CAN bus를 통해 연결된 상황을 보여준다. 만약 외부 통신을 담당하는 장치를 공격하여 CAN bus에 접근할 수 있다면, 이를 통해 bus에 연결된 다른 장치에 악의적인 메시지를 전송하거나, 정상적인 프레임을 저장한 다음 이를 다시 재전송하여 다른 장치들의 정상적인 동작을 방해할 수 있을 것이다. 또한, 반복적인 오류 메시지를 보내어 CAN bus 시스템을 종료시키거나, 각각의 ECU에 접근해 차량의 운전 방향, 속도 등에 대한 정보를 바꾸어 고의로 사고를 유발할 수도 있을 것이다. 이러한 공격이 가능한 이유는 CAN 데이터 프레임에 발신자 정보가 포함되지 않기 때문에 승인받은 장치인 것처럼 가장하여 시스템에 접근하기 쉽기 때문이다.

Ⅲ. CAN IDS를 위한 양방향 GPT

CAN bus 통신은 차량 내에서 호스트 컴퓨터 없이 ECU 간에 서로 효율적으로 통신하기 위해 1983년 Bosch에 의해 개발된 표준 통신 규격이다. CAN bus 통신은 단순하면서 효율적인 구조로 되어 있어서 1993년에 이를 ISO 11898 표준으로 제정하였다[1]. CAN bus는 한 장치에서 메시지를 전송하면 버스에 연결된 모든 장치가 수신할 수 있는 방송 방식으로 동작한다. 이때, CAN ID와 메시지로 구성된 CAN 데이터 프레임이 전송되며, CAN bus에 연결된 임의의 장치는 자신에게 필요한 특정 CAN ID에 해당하는 CAN 데이터 프레임만을 수신한다.

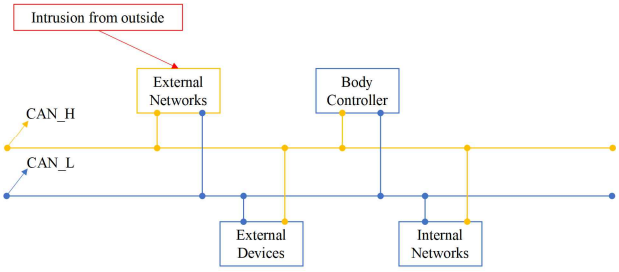

그림 2. CAN bus 연결 예시
Fig. 2. Example of CAN bus connection

본 장에서는 CAN IDS를 위해 제안되었던 양방향 GPT를 소개한다[5]. 기존의 기계번역은 순환신경망 기반의 인코더와 디코더로 구성된 시퀀스-to-시퀀스 model을 사용하였다. 구체적으로, 번역할 문장을 단어 단위로 나누어 word embedding 벡터로 변환하고 이를 순차적으로 인코더에 입력하여 하나의 벡터 표현(representation)으로 압축한다. 디코더는 이 벡터 표현을 이용하여 번역 문장을 생성한다. 이러한 구조는 번역할 문장을 하나의 벡터 표현으로 압축하는 과정에서 일부 정보가 손실된다는 문제가 있다. 이를 보완하기 위해 디코더에서 출력 단어를 예측하는 매 시점 인코더의 전체 입력 문장을 다시 한번 참고하는 attention 메커니즘이 제안되었다[8].

Attention 메커니즘은 디코더의 각 출력 단어를 예측할 시점에서, 해당 단어와 연관이 있는 인코더의 입력 단어에 좀 더 큰 가중치를 할당할 수 있는 attention 함수를 사용하여 번역 성능을 개선하였다. 그러나, 순환신경망 기반으로 인코더와 디코더를 구성하게 되면 attention 메커니즘을 사용한다고 하더라도 여전히 긴 문장에 대한 번역 성능에 한계가 있다. 이를 개선하기 위해 순환신경망을 사용하지 않고 attention 구조만을 이용한 transformer 기법이 제안되었다[9]. OpenAI는 이러한 transformer를 구성하는 구성요소 중 디코더만을 이용하여 문장 생성에 적용한 GPT 기법을 제안하였다 [10].

3-1 GPT의 구조

GPT는 이전의 출력이 다음의 입력이 되는 masked self-attention 구조를 활용한 자기 회귀 모델(auto-regressive model)로서, 입력된 단어들에 대한 다음 단어의 예측 능력이 우수하여 언어모델 기반의 문장 생성에 활용된다. 그림 3은 GPT 네트워크를 활용한 문장 생성 과정을 구체적으로 보여준다.

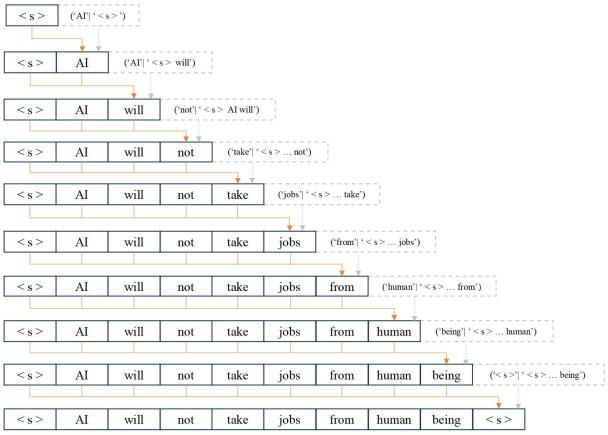

그림 3. GPT 네트워크를 활용한 문장 생성 과정 예시
Fig. 3. Example of sentence generation using GPT network

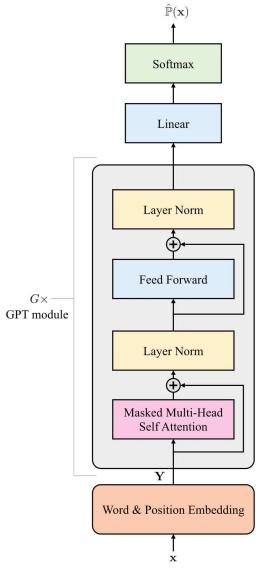

그림 4. GPT 네트워크 블록도
Fig. 4. Block diagram of GPT network

문장의 시작과 끝을 의미하는 단어 '<s>' 가 입력되면 그 다음에 등장할 단어를 추정하기 위해 발생할 수 있는 모든 단어에 대한 확률 $\mathbb{P}(\cdot|\text{'<s>'})$을 추정한다. 만약 추정된 확률들로부터 $\mathbb{P}(\text{'AI'}|\text{'<s>'})$ 값이 가장 크다면 단어 'AI'를 선택하고, 이 단어를 다시 GPT에 입력한다. '<s> AI'가 입력된 다음에 등장할 단어를 추정하기 위해 발생할 수 있는 모든 단어에 대한 확률 $\mathbb{P}(\cdot|\text{'<s> AI'})$을 계산한다. 계산된 확률들로부터 $\mathbb{P}(\text{'will'}|\text{'<s> AI'})$ 값이 가장 크다면 단어 'will'을 선택하고, 이 단어를 다시 GPT에 입력한다. 이 과정을 문장의 종료를 의미하는 단어 '<s>'가 선택될 때까지 반복하여 문장을 생성한다.

그림 4는 GPT 네트워크의 구체적인 구조를 보여준다. 이 그림으로부터 GPT 네트워크는 G개의 GPT 모듈이 적층되어 구성되어 있으며, 각 GPT 모듈은 masked multi-head 모듈, self-attention 모듈, layer normalization 모듈, feed-forward 모듈, layer normalization 모듈이 적층되어 있음을 알 수 있다.

우선, GPT 네트워크는 L개의 단어로 구성된 문장

$$\mathbf{x} = [\mathbf{x}_0 \cdots \mathbf{x}_{(\mathrm{L}-1)}]^\top \tag{1}$$

을 생성하며, 발생할 수 있는 총 단어 개수는 M이라 가정하자. 여기서, x_l은 $0 \le x_l \le (M-1)$을 만족하는 정수이다. 이러한 구성에서, 각 단어 x_l은 E차원의 word embedding 벡터로 변환된 후, 문장 내의 해당 단어의 위치에 따라 E차원 positional encoding 벡터가 더해져 열벡터 $\mathbf{y}_l$로 변환된다. 이때, positional encoding 벡터는 문장의 길이인 L개만큼 서로 다른 벡터들로 구성된 벡터 집합으로부터 해당 단어의 위치에 따라 그에 해당하는 벡터를 선택하여 사용한다.

결과적으로, 시퀀스 $\mathbf{x}$는

$$\mathrm{Y} = [\mathrm{y}_0 \cdots \mathrm{y}_{(\mathrm{L}-1)}] \tag{2}$$

로 변환되고, 이는 G개의 GPT 모듈로 구성된 GPT 네트워크에 입력된다. GPT 네트워크는 최종적으로 $L \times M$ 크기의 행렬을 출력하며, l번째 출력인 열벡터

$$u_l = [u_{0,l} \cdots u_{(M-1),l}]^\top \tag{3}$$

을 이용하여 시점 l직후에 등장할 단어 $x_{(l+1)}$에 대한 확률 추정치

$$\begin{aligned} &\hat{\mathbb{P}}(x_{(l+1)} | \{x_{l'}\}_{l'=0}^{l}) \\ &= \mathrm{softmax}(u_1) \\ &= \left[\frac{\exp(u_{0,l})}{\sum_{m=0}^{M-1} \exp(u_{m,l})} \cdots \frac{\exp(u_{(M-1),l})}{\sum_{m=0}^{M-1} \exp(u_{m,l})} \right]^\top \end{aligned} \tag{4}$$

계산한다. 최종적으로 시퀀스 $\mathbf{x}$에 대한 발생 확률 추정값

$$\hat{\mathbb{P}}(\mathrm{x}) = \prod_{1=0}^{(\mathrm{L}-1)} \hat{\mathbb{P}}(\mathrm{x}_{(1+1)} | \mathrm{x}_0, \cdots \mathrm{x}_1) \tag{5}$$

을 계산할 수 있다.

3-2 양방향 GPT 구조

양방향 GPT 기반의 CAN bus 공격 탐지 기법은 CAN ID 시퀀스를 정수로 변환하여 양방향 GPT 네트워크에 입력하고 해당 시퀀스에 대한 NLL 값을 계산하여, 문턱치 값과 비교하여 공격 여부를 탐지한다. 구체적인 과정은 다음과 같다. 탐지해야 할 CAN ID 시퀀스는 길이가 L이며, 이를 구성하는 유효한 CAN ID는 $(M-1)$개가 존재한다고 가정하자. 또한, 탐지하려는 CAN ID 시퀀스를 구성하는 각각의 CAN ID에 대하여, CAN ID 값을 오름차순으로 0부터 $(M-2)$까지의 정수로 대치하고 정상 CAN 신호에는 존재하지 않은 CAN ID는 모두 정수 $(M-1)$로 변환한다고 가정하자. 결과적으로, CAN ID 시퀀스는

$$\mathrm{x} = [x_0 \cdots x_{(L-1)}]^\top \tag{6}$$

로 표현된다. 여기서, $0 \le x_l \le (M-1)$을 만족한다.

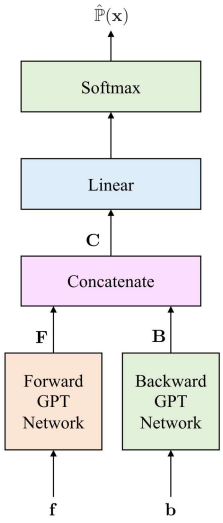

그림 5. 양방향 GPT 네트워크 블록도
Fig. 5. Block diagram of bi-directional GPT network

그림 5는 정방향 및 역방향 GPT 모듈이 결합된 양방향 GPT 네트워크 구조를 보여준다. 양방향 GPT 네트워크 구조를 이용한 탐지하려는 CAN ID 시퀀스에 대한 공격 탐지 과정은 다음과 같이 설명된다. 탐지하려는 CAN ID 시퀀스 $\mathbf{x}$는 각각

$$\mathrm{f} = [\mathrm{x}_0 \cdots \mathrm{x}_{(\mathrm{L}-2)}]^\top \tag{7}$$

과

$$\mathrm{b} = [\mathrm{x}_{(\mathrm{L}-1)} \cdots \mathrm{x}_1]^\top \tag{8}$$

로 변환되어 정방향 및 역방향 GPT 모듈에 입력된다. 이때, GPT 모듈은 입력된 단어들을 이용하여 다음 단어를 예측한다는 사실을 상기하면, 정방향 GPT 모듈은 $\mathbf{f}$를 입력받아

$$\mathrm{F} = \left[\tilde{\mathrm{f}}_0 \cdots \tilde{\mathrm{f}}_{(\mathrm{L}-2)}\right] \tag{9}$$

를 출력한다. 여기서, $\tilde{\mathbf{f}}_l$은 E차원 벡터로서

$$\mathbb{P}(x_{(l+1)} | x_0, \cdots x_l) \tag{10}$$

과 관련된 정보를 가지고 있다. 또한, 역방향 GPT 모듈은 $\mathbf{b}$를 입력받아

$$\mathrm{B} = \left[\tilde{\mathrm{b}}_0 \cdots \tilde{\mathrm{b}}_{(\mathrm{L}-2)}\right] \tag{11}$$

를 출력한다. 여기서, $\tilde{\mathbf{b}}_l$은 E차원 벡터로서

$$\mathbb{P}(x_{(L-l-2)} | x_{(L-l-1)}, \cdots x_{(L-1)}) \tag{12}$$

과 관련된 정보를 가지고 있다.

이때, 초기 예측 시점에서는 이용할 수 있는 관측 CAN ID 개수가 적어서 예측 성능이 감소하는 문제가 발생한다. 이러한 문제를 완화하기 위해 $\mathbf{F}$와 $\mathbf{B}$를 결합하여 $2E \times L$ 크기의 행렬

$$\begin{aligned} \mathrm{C} &= \begin{bmatrix} 0_{\mathrm{E}} & \mathrm{F} \\ \mathrm{BP} & 0_{\mathrm{E}} \end{bmatrix} \\ &= [\mathrm{c}_0 \cdots \mathrm{c}_{(\mathrm{L}-1)}] \end{aligned} \tag{13}$$

를 생성한다. 여기서, $\mathbf{P}$는 $\mathbf{B}$를 구성하는 각 열벡터의 순서를 $\mathbf{F}$의 열벡터 순서와 일치시키기 위한 exchange 행렬이고 [11], $\mathbf{0}_E$은 E차원 0 벡터이다. $\mathbf{C}$를 입력받은 linear layer는 $M \times L$ 크기의 행렬

$$\begin{aligned} \mathrm{U} &= \mathrm{W}^\top \mathrm{C} \\ &= [\mathrm{u}_0 \cdots \mathrm{u}_{\mathrm{L}-1}] \end{aligned} \tag{14}$$

를 출력한다. 여기서, $\mathbf{W}$는 linear layer를 구성하는 $2E \times M$ 크기의 행렬이다. 최종적으로, x_l에 대한 예측을 위해 softmax layer는 $\mathbf{u}_l$를 입력받아

$$\hat{\mathbb{P}}\left(x_l | \{x_{l'}\}_{l'=0, l' \neq l}^{(L-1)}\right) = \mathrm{softmax}(u_l) \tag{15}$$

를 출력한다. 최종적으로 탐지하려는 CAN ID 시퀀스 $\mathbf{x}$에 대한 확률 예측값은

$$\hat{\mathbb{P}}(\mathrm{x}) = \prod_{\mathrm{l}=0}^{(\mathrm{L}-1)} \hat{\mathbb{P}}\left(x_l = y_l | \{x_{l'}\}_{l'=0, l' \neq l}^{(L-1)}\right) \tag{16}$$

이 된다. 여기서, y_l은 x_l에 대한 정답 (ground truth) 값이다.

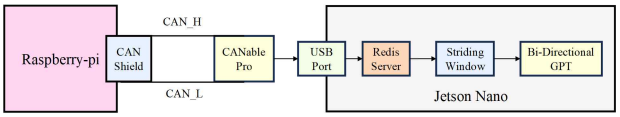

그림 6. 실시간 CAN IDS 실험 블록도
Fig. 6. Block diagram of real-time CAN IDS experiment

3-3 공격 탐지 방법

훈련을 위해 N개의 정상 CAN ID 시퀀스를 수집한다고 가정하자. 수집된 훈련 데이터를 이용하여 양방향 GPT 네트워크의 출력에 대해 다음과 같이 정의되는 NLL 손실 함수

$$\mathrm{NLL} = -\frac{1}{\mathrm{NL}}\sum_{\mathrm{n}=01=0}^{\mathrm{N}-1\mathrm{L}-1}\log\hat{\mathbb{P}}\left(\mathrm{x_l}=\mathrm{y_l}|\left\{\mathrm{x_{l'}}\right\}_{\mathrm{l'}=0,\mathrm{l'}\neq\mathrm{l}}^{(\mathrm{L}-1)}\right) \tag{17}$$

를 최소화하도록 훈련한다. 여기서, $x_l^{(n)}$은 훈련에 사용되는 n번째 정상 CAN ID 시퀀스의 l번째 변수이며, $y_l^{(n)}$은 $x_l^{(n)}$에 대한 정답 값이다.

훈련이 완료된 후, 주어진 CAN ID 시퀀스 $\mathbf{x}$의 공격 여부를 탐지하기 위해 양방향 GPT 네트워크에 $\mathbf{x}$를 입력하여

$$\mathrm{NLL_x} = -\frac{1}{\mathrm{L}}\sum_{\mathrm{l}=0}^{\mathrm{L}-1}\log\hat{\mathbb{P}}\left(\mathrm{x_l}=\mathrm{y_l}|\left\{\mathrm{x_{l'}}\right\}_{\mathrm{l'}=0,\mathrm{l'}\neq\mathrm{l}}^{(\mathrm{L}-1)}\right) \tag{18}$$

를 계산한다. 최종적으로, $\mathrm{NLL_x}$의 값이 문턱치 Γ보다 더 크다면, 즉,

$$\mathrm{NLL_x} > \Gamma \tag{19}$$

를 만족하면 공격이라 판정한다.

Ⅳ. CAN IDS 구성

젯슨 나노는 저사양 하드웨어로 작은 크기와 낮은 소비 전력량을 가지고 있으며, 이러한 특성이 차량 환경에 적합하다고 판단하여 양방향 GPT 기반 CAN IDS의 실시간 구현에 사용하였다. 기본적으로 젯슨 나노는 CAN 통신을 지원하지 않기 때문에, CANable Pro라는 외부 장치를 추가로 활용하였다[6]. 그림 6은 실시간 구현 구조를 보여준다.

본 연구에서 구현한 시스템은 크게 세 가지 병렬 프로세스로 구성되어 있다. 첫 번째 프로세스는 CAN 신호를 수신하고 CAN ID를 추출하는 과정을 담당한다. 두 번째 프로세스는 추출된 CAN ID를 인메모리 데이터베이스인 Redis 서버에 저장하는 역할을 한다. 이는 신호 분석에 필요한 데이터를 빠르게 조회하고 활용할 수 있게 해준다. 세 번째 프로세스는 저장된 CAN ID를 일정 길이만큼 읽어 들여, 양방향 GPT 네트워크를 이용하여 이상 여부를 추론한다.

수신된 CAN 신호에서 CAN ID만을 추출하여 Redis 서버에 저장하며, 이를 순차적으로 읽어 들여 길이가 K인 CAN ID 시퀀스를 생성한다. 이 시퀀스는 PyTorch를 이용하여 구현된 양방향 GPT 네트워크에 입력되어, NLL 값을 계산한다. 이 NLL 값은 문턱치와 비교되며, 이를 통해 시스템은 CAN 신호의 이상 여부를 판정하게 된다.

서론에서 언급했듯, 본 연구에서는 이동 윈도우와 보폭 개념을 도입하여 CAN ID 시퀀스를 생성하였다. 이동 윈도우의 크기는 K로, 보폭은 k로 설정하였다. 길이가 K인 CAN ID 시퀀스가 양방향 GPT 네트워크를 통해 이상 여부를 판정받은 후, CAN ID 시퀀스의 앞부분에서 k개의 CAN ID가 제거되고, Redis 서버로부터 k개의 새로운 CAN ID를 순차적으로 꺼내어 시퀀스의 뒷부분에 연속적으로 추가한다. 이를 통해 시퀀스는 지속적으로 업데이트되며, 탐지가 수행된다. 이 과정에서 보폭 k의 크기 설정은 매우 중요한 역할을 한다. 보폭 k가 작은 경우, CAN ID 당 탐지 시도 횟수

$$\Psi \triangleq \left\lfloor \frac{\mathrm{K}}{\mathrm{k}} \right\rfloor \tag{20}$$

가 증가하여 각 CAN ID를 여러 번 검사하게 된다. 따라서, 처리 지연이 발생하지만 이를 통해 미탐 확률을 줄일 수 있다. 반면, 보폭 k가 큰 경우, 처리 지연은 줄어들지만 미탐 확률이 증가하게 된다. 이러한 상충 관계를 고려하여 적절한 보폭 k를 결정하는 것이 중요하다.

Ⅴ. 실험 결과

5-1 실험 환경 구성 및 성능 평가 지표

차량에 직접 CAN IDS를 설치하고 실험하는 것은 현실적으로 쉽지 않아 다음과 같은 방식으로 진행하였다. 2020년식 Avante CN7 차량에서 발생하는 CAN 신호를 실시간으로 수집하여 blf 파일 형식으로 저장하였다. 저장된 blf 파일을, 라즈베리파이를 활용하여 실시간으로 재생하여 젯슨 나노 보드로 전송하였다[12]. 수집된 정상적인 CAN 신호는 총 90개의 유효한 CAN ID로 구성되어 있었으며, 이들 ID는 순차적으로 0부터 89까지 번호를 할당하였다. 유효하지 않은 CAN ID의 경우에는 모두 90번을 할당하였다.

Spoofing 공격은 일반적으로 알려진 다른 공격 유형인 Flooding, Reply, Fuzzing에 비해 정상 CAN ID를 사용하여 상대적으로 큰 주기로 주입하므로 검출 성능이 낮은 경향이 있다[5]. 그에 비해 다른 공격 유형은 공격 주입 주기가 매우 짧아서 서로 다른 검출 기법 간의 검출 성능에 큰 차이가 없으므로 본 논문에서는 spoofing 공격만을 고려하였다. 본 논문에서 수행한 spoofing 공격은 정상적인 CAN ID인 0x366과 0x553을 주입하였으며, 이는 CAN bus 네트워크에 존재

하는 ECU가 정상적인 CAN 신호를 전송하는 것으로 가장하여 시스템에 혼란을 주는 공격 방식이다.

양방향 GPT 네트워크의 입력으로 사용되는 CAN ID 시퀀스의 길이 K를 256으로 설정하였다. 양방향 GPT 네트워크는 정방향 및 역방향 GPT 모듈은 각각 6개의 계층으로 구성되어 있으며, 각 계층의 multi-head 개수는 8, 임베딩 벡터의 차원은 128로 설정하였다. 추론 과정에서는 GPU (graphic processing unit)의 병렬 연산 능력을 활용하여 처리 지연 시간을 줄이기 위해, 4개의 CAN ID 시퀀스를 mini-batch로 구성하여 탐지를 수행하였다.

성능 평가를 위한 성능 지표로서, FNR (false negative rate) 과 FPR (false positive rate)을 사용한다. FNR은 모든 공격 CAN ID 시퀀스들 중에 정상 (즉, false negative)로 잘못 판정한 비율이며, FPR은 모든 정상 CAN ID 시퀀스들 중에 공격 (즉, false positive)로 잘못 판정한 비율이다. 덧붙여, 다음과 같이 정의되는 정밀도 (precision)와 재현율 (recall)의 조화 평균인 F-measure

$$F = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \tag{21}$$

를 성능 평가 지표로 사용한다. 여기서, 정밀도는 공격이라 판정된 CAN ID 시퀀스들 중에 실제 공격의 비율이며, 재현율은 실제 공격인 CAN ID 시퀀스들 중에 공격이라 판정한 비율이다. F-measure는 그 값이 클수록 탐지 성능이 높다고 평가한다.

본 연구에서는 양방향 GPT 기법과의 성능을 비교하기 위해 LSTM (long short-term memory), 양방향 LSTM, 그리고 단방향 GPT를 적용하여 같은 실험을 진행하였다. 성능 비교가 공정하게 이루어질 수 있도록 네트워크 구조를 조정하였는데, 단방향 검출 기법인 LSTM과 GPT는 12개의 layer로 구성된 네트워크를 사용하였다. 반면, 양방향 검출 기법인 양방향 LSTM과 양방향 GPT는 방향별로 6개의 layer를 가진 네트워크로 설정하였다. 표 1은 기법별 신경망 구조를 나타낸다.

5-2 공격 검출 성능

본 절에서는 spoofing 공격에 대한 검출 성능에 대하여 논한다. 탐지하려는 대상인 CAN ID 시퀀스가 하나 이상의 공격 CAN ID가 포함되는 경우 이를 공격 시퀀스로 정의한다. 표 2는 FPR = 0.5%에서 공격 탐지 기법에 따른 FNR 및 F-measure 성능을 비교하였다. 여기서, 이동 윈도우를 사용하여 생성되는 CAN ID 시퀀스를 각각 독립적인 시퀀스로 간주하고 성능을 평가하였다. 예상대로 양방향 GPT의 성능이 가장 우수하나 보폭의 변화에 따른 성능변화가 없음을 확인할 수 있다. 그 이유는 독립 시퀀스 가정을 기반으로 성능을 측정하였기 때문이다.

구체적으로, 임의의 공격 CAN ID가 있을 때, 해당 CAN ID가 포함된 공격 시퀀스들은 이동 윈도우에 의해 Ψ개가 생성될 수 있다. 즉, 보폭 값 k가 작으면 임의의 공격 CAN ID가 포함된 시퀀스에 대한 탐지 시도 횟수가 증가한다. 그러나, 이동 윈도우에 의해 생성되는 시퀀스를 독립적으로 보고 성능을 측정하였기에 보폭 혹은 탐지 시도 회수에 따른 탐지 성능의 변화를 비교해 볼 수 없다.

이러한 한계를 해결하기 위해, 임의의 공격 CAN ID가 포함된 CAN ID 시퀀스들 중에서 한 번이라도 성공적으로 검출된다면 해당 CAN ID에 대한 검출에 성공하였다고 정의하였다. 이 정의에 의한다면 false negative는 임의의 공격 CAN ID가 이동 윈도우에 의해 생성된 여러 CAN ID 시퀀스에 포함되었음에도 한 번도 검출되지 않는 경우를 의미한다. 따라서, 탐지 시도 횟수 Ψ가 증가할수록 FNR은 감소하는 경향을 보일 것이다.

표 1. 기법별 신경망 구조
Table 1. Summary of neural network structures

	LSTM	Bi-LSTM		GPT	Bi-GPT	
Direction	Forward	Forward	Backward	Forward	Forward	Backward
No. of Layers	12	6	6	12	6	6
Seq. len.	256					
Dim. of embed	128					
Dropout prob.	0.1					

표 2. 공격 탐지 기법에 따른 FNR 및 F-measure 성능 비교 (FPR = 0.5%)
Table 2. Performance comparison of FNR and F-measure with stride length (FPR=0.5%)

	LSTM		Bi-LSTM		GPT		Bi-GPT	
Stride(k)	FNR	F-measure	FNR	F-measure	FNR	F-measure	FNR	F-measure
1	2.59e-1	8.37e-1	1.57e-1	9.00e-1	3.75e-1	7.55e-1	6.28e-2	9.52e-1
32	2.66e-1	8.32e-1	1.60e-1	8.98e-1	3.72e-1	7.57e-1	6.37e-2	9.52e-1
128	2.75e-1	8.26e-1	1.56e-1	9.01e-1	3.77e-1	7.54e-1	6.31e-2	9.52e-1
256	2.83e-1	8.21e-1	1.68e-1	8.93e-1	3.84e-1	7.48e-1	6.39e-2	9.54e-1

표 3은 이와 같은 검출 정의에서 FPR = 0.5%일 때, 보폭 값에 따른 FNR 성능변화 비교를 한 것이다. Spoofing 공격에 대하여 보폭 값이 32 이하인 경우, 양방향 GPT는 false negative가 전혀 발생하지 않았음을 확인하였다. 또한, 보폭 값 128과 256에 대해 다른 비교 기법에 비해 FNR이 최소 약 9%에서 최대 약 18%까지 감소함을 알 수 있다. F-measure의 경우, 최소 약 23%에서 최대 약 109%까지 성능이 향상됨을 알 수 있다.

표 3. 보폭 값에 따른 주어진 CAN ID에 대한 FNR 성능 비교 (FPR = 0.5%)
Table 3. Performance comparison of FNR for a given attack CAN ID with stride length (FPR = 0.5%)

Stride (k)	LSTM	Bi-LSTM	GPT	Bi-GPT
1	4.20e-2	0	8.17e-2	0
32	5.93e-2	1.50e-2	1.11e-1	0
128	1.04e-1	2.75e-2	1.65e-1	5.40e-3
256	1.44e-1	7.49e-2	1.97e-1	2.20e-2

5-3 종단 간 지연 성능

젯슨 나노 보드를 이용한 실시간 탐지 기법의 성능을 평가하기 위해 다양한 보폭 값 k를 설정하여 실험을 진행하였다. 그림 7은 K = 256일 때, 보폭 k 값에 따른 양방향 GPT 기법의 종단 간 지연 성능의 변화를 보여준다. 이 그래프로부터, k = 15일 때 종단 간 지연이 처음으로 수렴하며 최소 지연 시간이 1.43초로 측정됨을 알 수 있다.

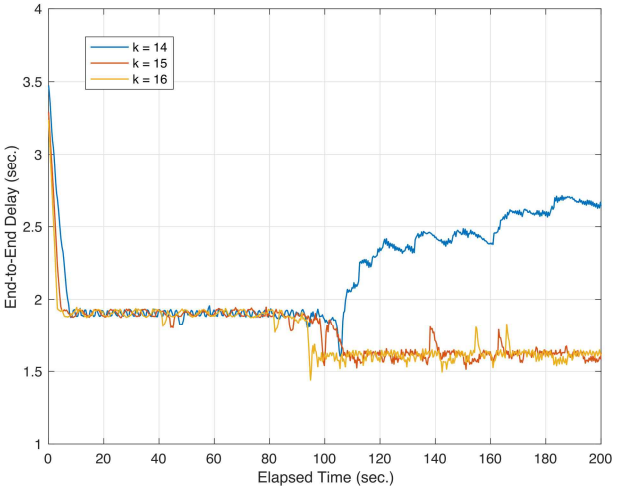

그림 7. 보폭 값에 따른 종단 간 지연 성능 (K = 256)
Fig. 7. End-to-end delay performance with stride length (K = 256)

그림 8은 K = 256, k = 15에서 기법별 종단 간 지연 성능을 보여준다. 기존의 다양한 공격 탐지 기법들과 양방향 GPT 기법의 종단 간 지연 시간을 비교하고 분석한 결과, 양방향 GPT 기법이 가장 우수함을 확인할 수 있다.

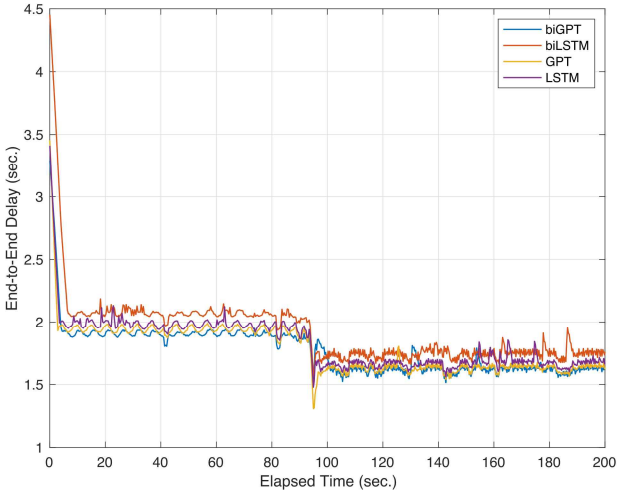

그림 8. 기법별 종단 간 지연 성능 (K = 256, k = 15)
Fig. 8. Performance comparison of end-to-end delay (K = 256, k = 15)

Ⅵ. 결 론

본 논문에서는 보안에 취약한 CAN bus 프로토콜에 대한 공격을 탐지하기 위해 양방향 GPT 기반의 CAN ID 시퀀스 이상 탐지 기법을 젯슨 나노 보드를 이용하여 실시간 구현을 검증하였다. 이 과정에서 실시간 구현에 따른 제약 조건과 그에 따른 최적화를 통해 최적의 탐지 보폭 값을 결정하였다. 특히 저사양의 하드웨어에서도 본 기법의 효율적인 구현이 가능하다는 점은, 임베디드 시스템에서의 실시간 CAN 이상 탐지에 있어서 비용 효율적인 해결책을 제안한다는 의미가 있다.

본 연구의 방법론은 새로운 차량 내부 통신용 프로토콜에도 적용할 수 있을 것으로 기대된다. 이는 차량 내부의 다양한 통신 환경에서의 이상 탐지 기법의 효율성을 보장할 것으로 기대된다.

감사의 글

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원 (No. RS-2023-00221494)과, 농림축산식품부의 재원으로 농림식품기술기획평가원의 스마트팜다부처패키기혁신기술개발 사업 (421040-04)의 지원과, 아우토크립트㈜의 지원을 받아 연구되었음.

참고문헌

[1] H. Chen and J. Tian, “Research on the Controller Area

Network," in *Proceedings of International Conference on Networking and Digital Society*, Guiyang, China, pp. 251-254, May 2009. https://doi.org/10.1109/ICNDS.2009.142

[2] M. Marchetti and D. Stabili, "Anomaly Detection of CAN Bus Messages Through Analysis of ID Sequences," in *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles: CA, pp. 1577-1583, June 2017. https://doi.org/10.1109/IVS.2017.7995934

[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. 16, No. 2, pp. 993-1006, April 2015. https://doi.org/10.1109/TITS.2014.2351612

[4] H. M. Song, J. Woo, and H. K. Kim, "In-Vehicle Network Intrusion Detection Using Deep Convolutional Neural Network," *Vehicular Communications*, Vol. 21, 100198, January 2020. https://doi.org/10.1016/j.vehcom.2019.100198

[5] M. Nam, S. Park, and D. S. Kim, "Intrusion Detection Method Using Bi-Directional GPT for In-Vehicle Controller Area Networks," *IEEE Access*, Vol. 9, pp. 124931-124944, 2021. https://doi.org/10.1109/ACCESS.2021.3110524

[6] Openlight Labs. CANable Pro [Internet]. Available: https://canable.io/getting-started.html.

[7] Redis Ltd. Redis [Internet]. Available: https://redis.io/.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, ... and Y. Bengio, "Generative Adversarial Nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS '14)*, Montreal, Canada, pp. 2672-2680, December 2014.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, ... and I. Polosukhin, "Attention is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS '17)*, Long Beach: CA, pp. 6000-6010, December 2017. https://dl.acm.org/doi/10.5555/3295222.3295349

[10] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language Models are Unsupervised Multitask Learners [Internet]. Available: https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf.

[11] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. New York, NY: Cambridge University Press, 2013.

[12] Korea Internet & Security Agency. K-Cyber Security Challenge 2020 [Internet]. http://datachallenge.kr/challenge20/car/rules/.

김송목(Song-Mok Kim)

2017년～현　재:강원대학교 전기전자공학과 학사과정
※관심분야 : 기계학습, 차량시스템, 임베디드 시스템 등

박승영(Seungyoung Park)

1999년 : 고려대학교 통신시스템학과 (공학석사)
2002년 : 고려대학교 전파통신공학과 (공학박사)
2003년～2005년: 삼성종합기술원 책임연구원
2006년～2007년: 퍼듀대학교 박사후연구원
2007년～현　재: 강원대학교 전기전자공학과 교수
2023년～현　재: 아우토크립트㈜ 연구자문
※관심분야 : 기계학습, V2X보안, 디지털통신 등