ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 23-04-22
j.inst.Korean.electr.electron.eng.Vol.27,No.4,524~530,December 2023
https : //doi.org/10.7471/ikeee.2023.27.4.524
157

초경량 Convolutional Neural Network를 이용한 차량용 Intrusion Detection System의 설계 및 구현

Design and Implementation of Automotive Intrusion Detection System Using Ultra-Lightweight Convolutional Neural Network

이 명 진*, 임 형 철*, 최 민 석*, 차 민 재*, 이 성 수*★

Myeongjin Lee*, Hyungchul Im*, Minseok Choi*, Minjae Cha*, and Seongsoo Lee*★

Abstract

This paper proposes an efficient algorithm to detect CAN (Controller Area Network) bus attack based on a lightweight CNN (Convolutional Neural Network), and an IDS(Intrusion Detection System) was designed, implemented, and verified with FPGA. Compared to conventional CNN-based IDS, the proposed IDS detects CAN bus attack on a frame-by-frame basis, enabling accurate and rapid response. Furthermore, the proposed IDS can significantly reduce hardware since it exploits only one convolutional layer, compared to conventional CNN-based IDS. Simulation and implementation results show that the proposed IDS effectively detects various attacks on the CAN bus.

요 약

본 논문에서는 경량화된 CNN(Convolutional Neural Network)을 사용하여 CAN(Controller Area Network) 버스 상의 공격을 탐지하는 효율적인 알고리즘을 제안하고, 이를 기반으로 하는 IDS(Intrusion Detection System)를 FPGA로 설계, 구현 및 검증하였다. 제안한 IDS는 기존의 CNN 기반 IDS에 비해 CAN 버스 상의 공격을 프레임 단위로 탐지할 수 있어서 정확하고 신속한 대응이 가능하다. 또한 제안한 IDS는 기존의 CNN 기반 IDS에 비해 컨볼루션 레이어를 하나만 사용하기 때문에 하드웨어를 크게 줄일 수 있다. 시뮬레이션 및 구현 결과는 제안된 IDS가 CAN 버스 상의 다양한 공격을 효과적으로 탐지한다는 것을 보여준다.

Key words : Convolutional Neural Network, Intrusion Detection System, CAN Bus, Lightweight, Automotive Security

* School of Electronic Engineering and Department of Intelligent Semiconductor, Soongsil University (Student, Student, Student, Student, Professor)
★ Corresponding author
E-mail : sslee@ssu.ac.kr, Tel : +82-2-820-0692
※ Acknowledgment
This work was supported by the R&D Program of the Ministry of Trade, Industry, and Energy (MOTIE) and Korea Evaluation Institute of Industrial Technology (KEIT). (20023805, RS-2022-00155731, RS-2022-00232192)
Manuscript received Dec. 14, 2023; revised Dec. 21, 2023; accepted Dec. 22, 2023.

Ⅰ. 서론

CAN(Controller Area Network)[1] 버스는 자동차와 산업 제어 시스템에서 사용되는 고성능 통신 프로토콜로, 안정적인 데이터 교환을 지원한다. 주요 특징으로는 주기적 통신으로 버스 모니터링, 멀티마스터 네트워크로 다중 노드 간의 동시 통신, 높은 대역폭을 통한 데이터 전송을 가능하게 한다. 또한 노이즈와 간섭에 강한 차동 꼬임선(Twisted Differential Pair) 구조로 되어있어 안정적으로 작동할 수 있다.

하지만 CAN 버스는 메시지 암호화 또는 인증과 같은 보안 메커니즘을 제공하지 않기 때문에 공격자가 CAN 버스에 쉽게 접근할 수 있다. 예를 들어, OBD-II[2] 포트를 통한 CAN 프레임 주입 공격은 차량 주행 중 브레이크를 비활성화하거나 즉각적인 회전수 서지를 유도할 수 있다[3]. 또한 동글을 사용하여 OBD-II에 접근하고 원격으로 공격을 수행할 수 있다[4].

최근의 증가하는 사이버 공격과 침입으로부터 CAN 버스를 안전하게 유지하기 위한 기술의 필요성이 더욱 커지고 있다[5]. 따라서 인공지능을 이용한 다양한 침임 탐지 시스템(IDS: Intrusion Detection system)이 제안되고 있다[6]-[9]. 특히 [9]에서는 합성곱 신경망(CNN: Convolutional Neural Network)을 이용한 딥러닝 방식의 침입 탐지 시스템을 제안하였다. CAN 버스에서 등장하는 ID의 순차적 패턴을 학습하였고 우수한 성능을 보였다. 하지만 이 연구에서는 CAN 프레임을 29 프레임씩 묶어서 학습을 하였기 때문에 29 프레임 묶음 안에 공격이 존재한다는 것은 탐지하지만 어떠한 프레임이 공격인지 정확히 알 수 없다는 단점이 존재한다. 또한 컨볼루션에 필요한 연산량이 매우 높아서 차량에 탑재하기에는 하드웨어가 커져서 적합하지 않다.

본 논문에서는 CAN ID의 순차적 패턴을 이용하여 실시간으로 공격 여부를 탐지할 수 있는 CNN 기반 침임 탐지 시스템을 제안한다. 또한 기존에 제안되어 있는 CNN 기반 침입 탐지 시스템들과는 다르게 컨볼루션 레이어를 하나만 이용하여 경량화된 모델을 구성하였다. 이를 Verilog HDL로 설계하고 FPGA로 검증하였다.

Ⅱ. 초경량 CNN 기반 침입 탐지 모델

1. CAN 버스 공격

본 논문에서는 공개되어 있는 차량 해킹 데이터세트[10]를 사용하였다. 데이터 세트는 공격이 존재하지 않는 정상 데이터와 DoS 공격(Denial of Service Attack) 공격, 스푸핑 공격(Spoofing Attack), 퍼지 공격(Fuzzy Attack)의 공격 데이터로 구성되어 있다. 해당 데이터는 실제 차량에서 메시지 주입 공격이 수행되는 동안 OBD-II 포트를 통해 CAN 트래픽을 추출하여 생성되었다.

DoS 공격은 0x00과 같은 우선 순위가 높은 ID를 가진 CAN 프레임을 주입하는 것을 특징으로 한다. 이러한 유형의 공격은 공격자가 CAN 버스를 계속 점유하고 정상적인 통신을 차단하여 다른 노드가 메시지를 전송하지 못하게 유도한다. 스푸핑 공격은 실제 노드 간 통신에 사용되는 CAN 프레임의 ID와 데이터를 수집했다가 나중에 이 ID와 데이터를 가진 CAN 프레임을 내보내어 해당 프레임을 정상 프레임인 것처럼 오인하여 특정 장치를 조작할 수 있게 한다. 본 논문에서 사용된 데이터 세트에는 기어(Gear) 또는 회전수(RPM) 데이터를 주입하는 두 가지 공격 데이터가 들어 있다. 마지막으로 퍼지 공격은 랜덤하게 생성된 ID와 데이터를 가진 CAN 프레임을 전송하여 다른 노드의 오동작을 유도한다.

2. 데이터 전처리

본 논문에서 제안하는 CNN 기반 침임 탐지 시스템은 11개의 순차적 CAN ID로 학습하여 CAN 버스에 나타나는 CAN ID의 패턴을 분석한 후 공격 여부를 탐지할 수 있다. 데이터 전처리는 슬라이딩 윈도우 기법으로 다음과 같이 진행하였다. 먼저 주어진 데이터 세트에서 CAN ID 영역만을 11개 추출하고, 윈도우 크기는 1로 설정하여 학습 데이터를 구성하였다. 또한 마지막 CAN ID의 공격 여부로 라벨링을 진행하였다. 예를 들어, 첫 번째 학습 데이터가 ID1, ID2, ID3, ..., ID11로 구성되어 있다면 두 번째 학습 데이터는 ID2, ID3, ID4, ..., ID12로 구성되며 각각의 라벨링은 ID11과 ID12의 공격 여부로 결정한다. 이와 같은 방법은 CAN 버스에 나타나는 CAN ID만을 수신하고, 현재 수신된 ID의 공격 여부를 즉각적으로 판단할 수 있다.

3. 경량화 CNN 모델

본 논문에서는 차량용 침임 탐지 시스템을 하드웨어로 구현하기 위해 그림 1과 같이 경량화된 CNN 모델을 제안한다. 제안하는 모델은 단 하나의 컨볼루션 레이어(Convolution Layer)만을 사용하여 경량화하였다. 또한 컨볼루션 대상인 CAN ID가 11개의 비트로 이루어졌기 때문에 곱셈기 대신에 AND 게이트로 구현하여 일반적

인 CNN 모델에 비해 하드웨어 크기를 크게 줄였다

학습에 사용한 공격 데이터[10]는 DoS 공격, 스푸핑 공격, RPM 공격의 세 가지 시나리오가 있고 이 중에서 스푸핑 공격이 Gear 데이터 공격과 RPM 데이터 공격으로 나뉘어져 있다. 따라서 DoS 공격, Gear 공격, RPM 공격, 퍼지 공격의 4 종류에 대해 데이터 세트를 각각 15만개씩 추출하여 합쳐서 총 60만개의 데이터를 사용하여 훈련을 진행하였다. 훈련에 사용된 하이퍼 파라미터(Hyper-Parameter)는 32 배치 크기(batch size), 10 에포크(epochs), 옵티마이저(optimizer)는 Adam을 사용하였으며, 학습률(learning rate)은 0.001로 설정하였다. 또한 손실 함수로는 이진 교차 엔트로피(Binary Cross-Entropy)를 사용하여 4가지 공격에 대해 공격 또는 정상으로 이진 분류를 수행하도록 설정하였다.

Ⅲ. 하드웨어 설계

1. 컨볼루션 레이어 구현

본 논문에서 설계한 CNN 가속기의 컨볼루션 레이어는 11×11 비트를 인풋 컨트롤러로 입력받아 제로 패딩 처리 후 3×3 가중치와 컨볼루션 연산을 한다. 결과로는 11×11의 출력 16채널을 출력한다.

그림 2는 인풋 컨트롤러의 구조이다. 인풋 컨트롤러는 제로 패딩과 커널에 들어갈 데이터를 업데이트하는 역할을 한다. 저장된 11×11 입력을 13개의 13비트 버퍼에 저장한다. 0번째와 12번째 버퍼를 0으로 채우고 나머지 버퍼의 0번째와 12번째 비트도 0으로 채워 제로 패딩을 표현했다. 커널로의 출력은 0, 1, 2번 버퍼의 오른쪽 3 비트를 사용하며 해당 버퍼를 그림 2의 녹색 화살표와 같이 오른쪽으로 시프트시켜 데이터를 업데이트한다. 선택된 버퍼의 출력이 끝나면 그림 2의 적색 화살표와 같이 버퍼 하나를 내려서 1, 2, 3번 버퍼를 선택하여 출력 데이터를 업데이트한다. 이러한 방법으로 10, 11, 12번 버퍼까지 출력한다.

그림 3은 커널의 구조이다. 커널마다 9개의 고정된 가중치가 하드와이어로 내장되어 있으며 가중치 세트는 커널마다 다른 세트를 사용한다. 컨볼루션 연산은 한쪽은 고정된 가중치, 다른 한쪽은 인풋 컨트롤러로부터의 입력을 받아 수행한다. 각 입력 피처 한 칸의 값들은 1과 0으로 이루어져 있어 컨볼루션 연산의 곱셈 연산을 AND 연산으로 대체할 수 있다. 입력이 1일 때 가중치 값이 통과하고, 입력이 0일 때는 0을 출력한다. 이와 같은 방법을 통해 큰 리소스를 갖는 곱셈 연산기를 대체하고 경량화된 컨볼루션 레이어를 구현하였다.

2. 맥스 풀링 레이어 구현

그림 4는 맥스 풀링(Max Pooling) 레이어로 2×2 필터를 사용하여 11×11 입력을 5x5로 축소한다. 2x2 필터를 사용하기 때문에 마지막 열과 행들은 그림 3과 같이 생략하여 5×5로 출력한다. 마지막 열과 행의 생략은 제로 패딩으로 인해 결과값에 영향을 주지 않는다.

맥스 풀링의 설계는 플립플롭과 타이밍을 최적화하기 위하여 그림 5와 같은 방법을 사용하였다. 먼저 컨볼루

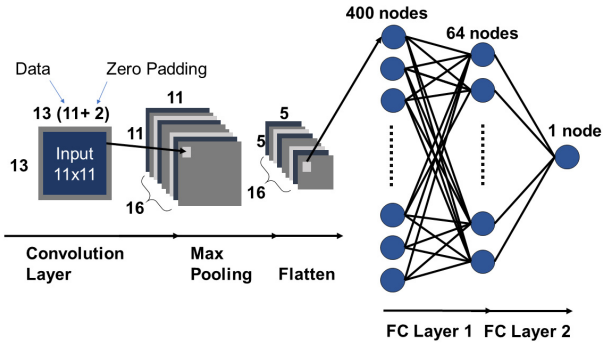


Fig. 1. Lightweight CNN architecture.
그림 1. 경량화된 CNN의 구조

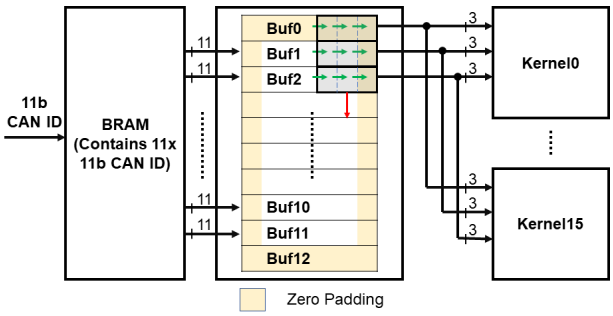


Fig. 2. Input controller architecture.
그림 2. 인풋 컨트롤러의 구조

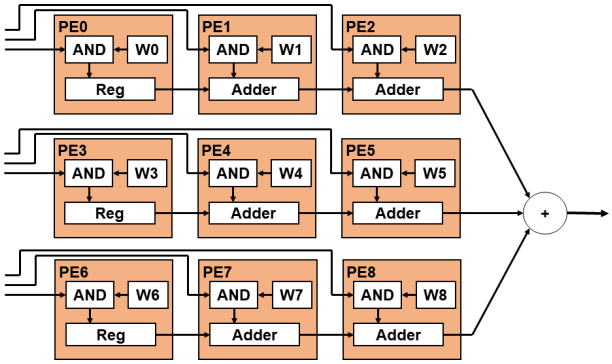


Fig. 3. Kernel architecture.
그림 3. 커널의 구조

션 레이어의 출력을 순차적으로 2개씩 입력받아 비교한다. 10개의 데이터를 5번에 받아 비교하여 큰 값 5개를

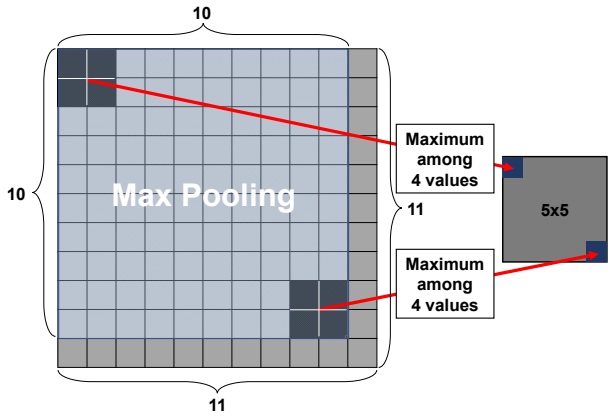


Fig. 4. Max pooling architecture.
그림 4. 맥스 풀링의 구조

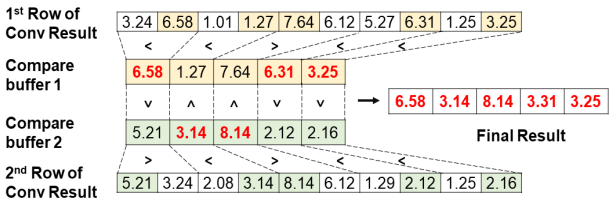


Fig. 5. Comparison method of max pooling.
그림 5. 맥스 풀링의 비교 방식

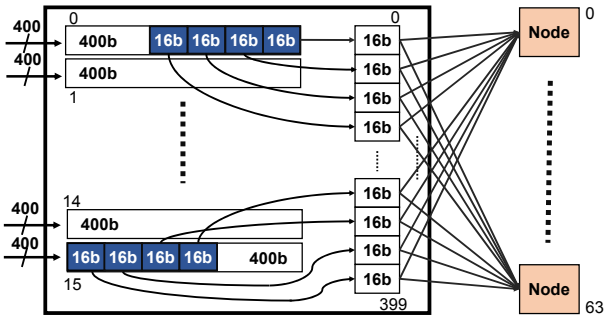


Fig. 6. Flatten 1 architecture.
그림 6. Flatten 1의 구조

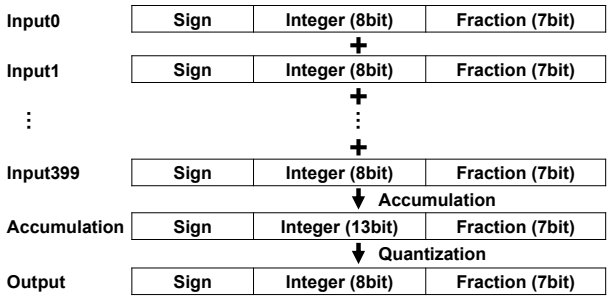


Fig. 7. Bit resolution expansion of FC layer 1.
그림 7. FC 레이어 1의 비트 해상도 확장

첫 번째 버퍼에 채운 후, 같은 방식으로 두 번째 버퍼에 순차적으로 채운다. 두 개의 버퍼가 채워지면 두 개의 버퍼의 순서에 맞게 크기를 비교하여 큰 값들을 결과로 출력한다. 이렇게 함으로서 20개의 컨볼루션 레이어 결과값이 전달되면 5개의 맥스 풀링 결과가 출력된다. 위의 방식을 이용하여 2×2 커널을 가진 맥스 풀링 레이어를 하드웨어로 구현하였다.

3. FC 레이어 구현

FC 레이어(Fully connected layer) 1은 400개의 입력과 64개의 출력 노드로 구성되어 있다. 각각의 노드는 400개의 컨볼루션 결과를 입력으로 받아 Flatten 1을 통과하여 각 노드에 입력한다. Flatten 1의 구조는 그림 6와 같다. 16비트 단위로 쪼개어 가장 위의 0번째 16비트부터 순차적으로 각 노드에 입력한다. 입력된 값은 가중치와의 곱셈 연산의 결과를 모두 더한 값으로 출력된다. 64개의 노드의 출력은 16비트를 순차적으로 나열하여 1024비트로 출력된다.

FC 레이어 2는 1개의 노드로 구성된다. FC 레이어 1과 마찬가지로 FC 레이어 1의 출력을 Flatten 2를 통하여 16비트씩 순차적으로 하나의 노드에 입력한다. 가중치를 곱셈 연산 결과를 모두 더해 출력한다. 출력된 노드의 값이 0보다 작다면 공격으로 판단하고, 0보다 크다면 정상 메시지로 판단한다. 이와 같은 방법은 시그모이드(Sigmoid) 함수를 구현하지 않아도 되기 때문에 연산량을 줄일 수 있다.

FC 레이어 1과 2에서는 덧셈과 곱셈을 연산 시에 1비트의 부호, 7비트의 정수, 8비트의 소수로 구성된 16비트 고정소수점을 사용한다. 부동소수점 연산기는 고정소수점보다 큰 리소스를 차지하기 때문에 고정소수점 연산기를 사용한다. 덧셈 연산 시에 최대 400회의 덧셈이 하나의 결과로 출력되기 때문에 8비트의 정수를 초과하여 오버플로우가 발생하게 된다. 따라서 FC 레이어 1과 2에서는 오차를 최소화하고 오버플로우 문제를 해결하기 위하여 그림 7과 같이 덧셈 연산 중에 24비트로 확장하는 방법[11]을 이용하였다.

4. 전체 아키텍처

그림 8은 본 논문에서 설계한 IDS CNN 가속기의 전체 구조이다. 전체 구조는 크게 컨볼루션 레이어와 FC 레이어 1, FC 레이어 2로 구성되어 있다. 컨볼루션 레이어의 가중치는 고정값을 하드와이어로 내장하였지만 FC 레이어 1, FC 레이어 2의 가중치는 알고리즘 고도화 시

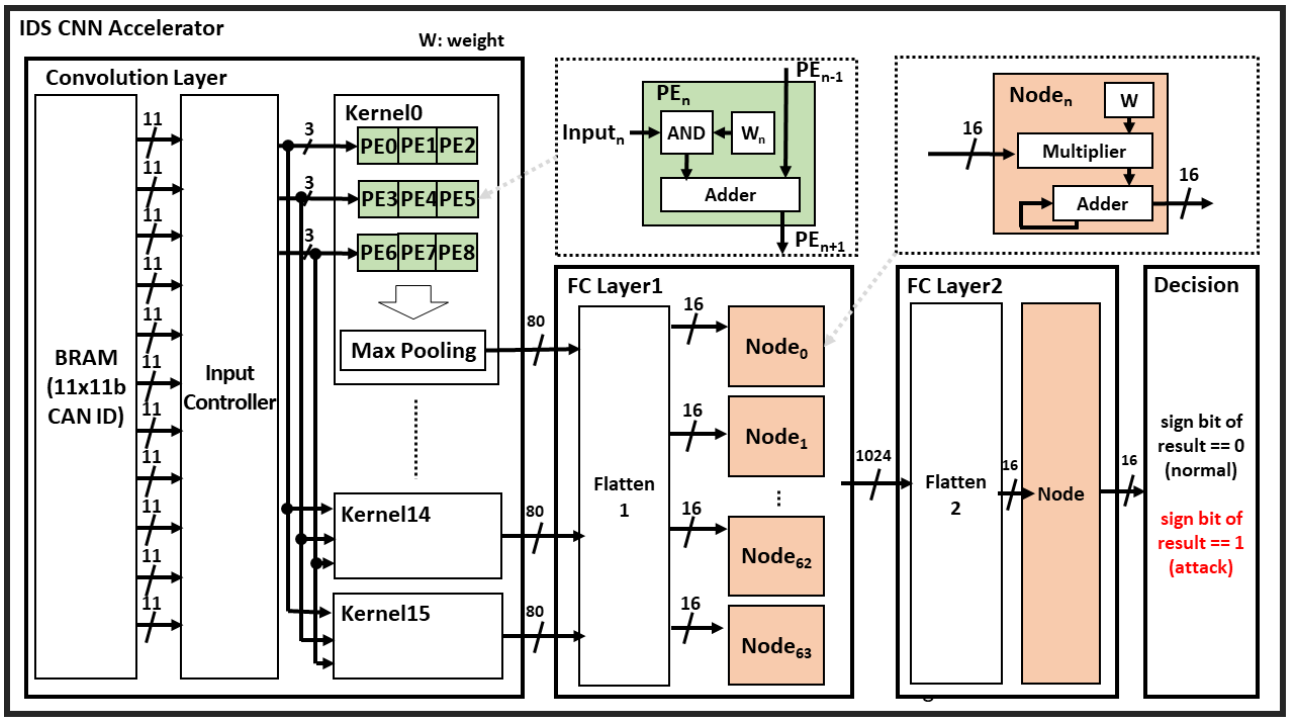


Fig. 8. IDS CNN architecture.
그림 8. IDS CNN 가속기의 구조

에 업데이트를 하기 위해서 FPGA에서 제공하는 BRAM에 저장하였다. 이 BRAM은 ASIC 칩 제작에서는 Embedded SRAM으로 대체할 계획이다.

Ⅳ. 검증 및 성능평가

1. FPGA 검증

본 논문에서 CNN 기반 IDS를 FPGA로 구현하기 위해 Arty-a7 보드를 사용하였으며 IDEC에서 지원한 Xilinx Vivado 툴로 100MHz 클록에 합성하였다. 검증 환경은 그림 9와 같이 구현하였다. FPGA 내부에는 IDS CNN 코어 이외에 CAN ID로 구성된 입력 피처를 저장해놓은 BRAM과 출력값을 PC로 전송하기 위한 UART TX 모듈을 추가하였다. 표 1은 FPGA 구현 리포트인데 Arty-a7 리소스의 약 10% 정도 차지함을 알 수 있다.

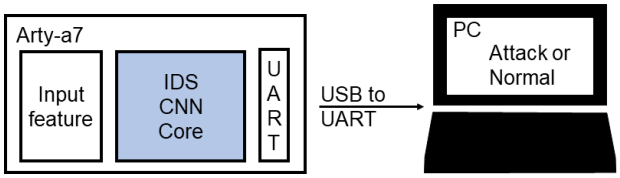


Fig. 9. FPGA implementation and verification environment.
그림 9. FPGA 구현 및 검증 환경

Table 1. FPGA Implementation report.
표 1. FPGA 구현 결과

Parameters	Utilization
LUT	10949
LUTRAM	0
FF	15350
BRAM	32.5
DSP	65
BUF	1

2. 성능 평가

성능평가에서는 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1점수(F1-Score)의 4가지 평가 지표를 사용하였다. Accuracy는 라벨링된 데이터를 정확하게 측정한 비율이다. Precision은 모델이 참으로 분류한 것 중에 실제 참인 것을 나타낸 비율이다. Recall은 실제 참인 것 중에서 모델이 참이라고 예측한 것의 비율이다. F1-Score는 Precision과 Recall의 조화평균으로 불균형한 데이터에서 잘 동작하는 지를 평가하는 지표이다.

성능평가를 위한 입력 데이터는 학습에 사용되지 않은 각 공격 데이터 세트에서 1만개씩 추출하여 진행하였다. 여기에서 부동소수점을 사용한 TensorFlow와 고정소수점을 사용한 FPGA의 성능이 다르게 나올 가능성이 있어

서 본 논문에서는 TensorFlow와 FPGA 모두에서 성능을 측정하였다. TensorFlow로 시뮬레이션하여 얻은 성능지표는 표 2에, FPGA를 수행하여 얻은 성능지표는 표 3에 나타내었다. 이를 통해 본 논문에서 사용한 경량화된 CNN 모델의 성능이 비교적 우수함을 알 수 있고, FPGA 수행 결과가 TensorFlow 수행 결과와 거의 유사하다는 점을 알 수 있다.

Table 2. Performance of TensorFlow simulation.
표 2. TensorFlow 시뮬레이션 성능

Attack	Accuraccy	Precision	Recall	F1-score
Dos	99.98999	99.95196	100	99.97597
Fuzzy	99.27942	97.44172	98.44916	97.94285
Gear	97.15772	90.06326	99.30242	94.45745
RPM	98.31865	92.42658	98.84297	95.52715

Table 3. Performance of FPGA execution.
표 3. FPGA 동작 성능

Attack	Accuraccy	Precision	Recall	F1-score
Dos	99.96998	99.85605	100	99.92797
Fuzzy	99.02922	96.86431	97.58759	97.22461
Gear	97.07766	89.88472	99.17932	94.30355
RPM	98.27862	92.14982	98.95317	95.43039

Ⅴ. 결론

본 논문에서는 합성곱 신경망과 FPGA를 활용하여 CAN 버스에서의 공격을 효과적으로 탐지하는 CNN 기반 경량 실시간 IDS 시스템을 설계하고 구현하였다. 제안하는 IDS는 컨볼루션 레이어를 하나만 사용하고 이진 입력값의 사용을 통해 곱셈기 대신에 AND 게이트를 사용하여 하드웨어를 크게 경량화하였다. 또한 TensorFlow 시뮬레이션과 FPGA 수행 결과를 비교하였을 때 제안하는 알고리즘의 성능이 비교적 우수하였으며 고정소수점을 사용한 FPGA 구현이 부동소수점을 사용한 TensorFlow 시뮬레이션과 거의 동일하다는 것을 검증하였다.

References

[1] ISO 11898-1:2015, “Road Vehicles-Controller Area Network (CAN)-Part 1: Data Link Layer and Physical Signaling,” https://www.iso.org/standard/63648.html

[2] ISO 9141-3:1998, “Road Vehicles-Diagnostic Systems-Part 3: Verification of the Communication between Vehicle and OBD II Scan Tool,” https://www.iso.org/standard/28621.html

[3] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, and S. Savage, “Experimental Security Analysis of a Modern Automobile,” *Proceedings of IEEE Symposium on Security and Privacy,* pp.447-462, 2010. DOI: 10.1109/SP.2010.34

[4] S. Woo, H. Jo, and D. 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, 2015. DOI: 10.1109/TITS.2014.2351612

[5] C. Lin and A. Sangiovanni-Vincentelli, “Cyber-Security for the Controller Area Network (CAN) Communication Protocol,” *Proceedings of International Conference on Cyber Security*, pp.1-7, 2012. DOI: 10.1109/CyberSecurity.2012.7

[6] S. Khandelwal and S. Shreejith, “A Lightweight FPGA-based IDS-ECU Architecture for Automotive CAN,” *Proceedings of International Conference on Field Programmable Technology,* pp.1-6, 2019.

[7] E. Seo, H. Song, and H. Kim, “GIDS: GAN Based Intrusion Detection System for In-Vehicle Network,” *Proceedings of Annual Conference on Privacy, Security, and Trust,* pp.1-6, 2018. DOI: 10.1109/PST.2018.8514157

[8] H. Im, D. Lee, and S. Lee, “Intrusion Detection System for In-Vehicle Network to Improve Detection Performance Considering Attack Counts and Attack Types,” *Korean.electr.elctron.eng.,* vol.26, no.4, pp.622-627, 2022.

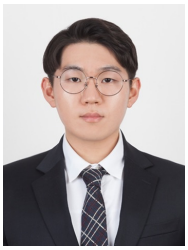
[9] H. Son, J. Woo, and H. Kim. “In-Vehicle Network Intrusion Detection Using Deep Convolutional Neural Network,” *Vehicular Communications,* vol.21, pp.100198, 2020. DOI: 10.1016/j.vehcom.2019.100198

[10] “Hacking and Countermeasure Research Lab. Car-Hacking Dataset”. https://ocslab.hksecurity.

net/Datasets/car-hackingdataset
[11] Y. Wu and C. Huang, "Efficient Dynamic Fixed-Point Quantization of CNN Inference Accelerators for Edge Devices," Proceedings of International Symposium on VLSI Design, Automation and Test, pp.1-4, 2019. DOI: 10.1109/VLSI-DAT.2019.8742040

BIOGRAPHY

Myeongjin Lee (Member)



2019 : BS degree in Electronic Engineering, Inje University.
2022~: Candidate for MS degree in the Department of Intelligent Semiconductors, Soongsil University.
〈Main Interest〉 Vehicle Security, Accelerator, Automotive SoC

Hyungchul Im (Member)

2021 : BS degree in Mechanical Engineering, Soongsil University.
2021~: Candidate for Ph.D degree in the Department of Intelligent Semiconductors, Soongsil University.
〈Main Interest〉 Vehicle Security, Artificial Intelligence, Automotive SoC

Minseok Choi (Member)

2023 : BS degree in Electronic Engineering, Soongsil University.
2023~: Candidate for MS degree in the Department of Intelligent Semiconductors, Soongsil University.
〈Main Interest〉 Vehicle Security, Accelerator, Automotive SoC

Minjae Cha (Member)

2018~ : Candidate for BS degree in Electronic Engineering, Soongsil University
〈Main Interest〉 Al SoC, NPU, Processor

Seongsoo Lee (Life Member)

1991 : BS degree in Electronic Engineering, Seoul National University.
1993 : MS degree in Electronic Engineering, Seoul National University.
1998 : PhD degree in Electrical Engineering, Seoul National University.
1998~2000 : Research Associate, University of Tokyo
2000~2002 : Research Professor, Ewha Womans University
2002~Now : Professor in School of Electronic Engineering, Soongsil University
〈Main Interest〉 AI SoC, Automotive SoC, Security SoC, Processor SoC, Power Management SoC, Battery Management SoC, Reliability and Safety