Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.6, Dec. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.6.919

부채널 분석을 이용한 DNN 기반 MNIST 분류기 가중치 복구 공격 및 대응책 구현*

이 영 주,[1†] 이 승 열,[1] 하 재 철[2‡]
[1,2]호서대학교 (학생, 교수)

Weight Recovery Attacks for DNN-Based MNIST Classifier Using Side Channel Analysis and Implementation of Countermeasures*

Youngju Lee,[1†] Seungyeol Lee,[1] Jeacheol Ha[2‡]
[1,2]Hoseo University (Student, Professor)

요 약

딥러닝 기술은 자율 주행 자동차, 이미지 생성, 가상 음성 구현 등 다양한 분야에서 활용되고 있으며 하드웨어 장치에서 고속 동작을 위해 딥러닝 가속기가 등장하게 되었다. 그러나 최근에는 딥러닝 가속기에서 발생하는 부채널 정보를 이용한 내부 비밀 정보를 복구하는 공격이 연구되고 있다. 본 논문에서는 DNN(Deep Neural Network) 기반 MNIST 숫자 분류기를 마이크로 컨트롤러에서 구현한 후 상관 전력 분석(Correlation Power Analysis) 공격을 시도하여 딥러닝 가속기의 가중치(weight)를 충분히 복구할 수 있음을 확인하였다. 또한, 이러한 전력 분석 공격에 대응하기 위해 전력 측정 시점의 정렬 혼돈(misalignment) 원리를 적용한 Node-CUT 셔플링 방법을 제안하였다. 제안하는 대응책은 부채널 공격을 효과적으로 방어할 수 있으며, Fisher-Yates 셔플링 기법을 사용하는 것보다 추가 계산량이 1/3보다 더 줄어듦을 실험을 통해 확인하였다.

ABSTRACT

Deep learning technology is used in various fields such as self-driving cars, image creation, and virtual voice implementation, and deep learning accelerators have been developed for high-speed operation in hardware devices. However, several side channel attacks that recover secret information inside the accelerator using side-channel information generated when the deep learning accelerator operates have been recently researched. In this paper, we implemented a DNN(Deep Neural Network)-based MNIST digit classifier on a microprocessor and attempted a correlation power analysis attack to confirm that the weights of deep learning accelerator could be sufficiently recovered. In addition, to counter these power analysis attacks, we proposed a Node-CUT shuffling method that applies the principle of misalignment at the time of power measurement. It was confirmed through experiments that the proposed countermeasure can effectively defend against side-channel attacks, and that the additional calculation amount is reduced by more than 1/3 compared to using the Fisher-Yates shuffling method.

Keywords: Deep Neural Network, Side Channel Attack, Correlation Power Analysis, Node-CUT Shuffling

Received(10. 11. 2023), Modified(11. 14. 2023), Accepted(11. 15. 2023)

* 본 성과물은 중소벤처기업부에서 지원하는 2022년도 스마트 제조 혁신 기술개발(R&D) (No. RS-2022-00140535)의 연구수행으로 인한 결과물임을 밝힙니다.

* 본 논문은 2023년도 한국정보보호학회 충청지부 학술대회에 발표한 우수논문을 개선 및 확장한 것임.

† 주저자, lyj04281@gmail.com

‡ 교신저자, jcha@hoseo.edu(Corresponding author)

I. 서 론

최근 4차 산업 혁명을 선도하는 기술인 인공 지능에 대한 관심이 급증하고 있으며, 특히 자율 주행 자동차, 이미지 생성, 가상 음성 생성 등에 활용되는 딥러닝(deep learning)의 연구가 활발하다. 성능이 우수한 딥러닝 모델에 대한 개발과 더불어 저전력, 저사양 하드웨어 환경에서 동작시키기 위한 딥러닝 가속기에 관한 연구가 진행 중이다.

그러나, 공격자가 딥러닝 가속기에 쉽게 접근하거나 이를 탈취하는 경우, 공격자는 가속기의 동작 과정에서 발생하는 부채널 신호를 이용하여 가속기 내부의 비밀 정보인 가중치(weight)나 편향(bias) 값을 복구할 수 있다.

먼저, 딥러닝 네트워크의 내부 비밀 정보 복구와 관련된 연구는 Batina 등[1]의 연구가 있었다. 이 연구에서는 MNIST 데이터셋으로 학습한 MLP(Multi Layer Perceptron) 모델과 CNN(Convolutional Neural Network) 모델을 공격 대상으로 하여 가중치, 활성화 함수(activation function), 레이어(layer)와 노드 수 등을 복구하였다.

또한, Park 등[2]은 Batina 등의 연구에서 가중치 범위를 제한하는 문제를 해결하기 위해 분할 정복(divide and conquer) 방식의 새로운 가중치 복구 방법을 제시하였다. 이 방법은 입력 데이터와 가중치가 곱해지는 부분을 공격 지점으로 설정하고, 가중치 값을 부호(sign), 지수부(exponent), 가수부(mantissa)로 분할하여 상관 전력 분석(Correlation Power Analysis) 공격을 수행하였다. 이 연구에서 저자들은 분할 정복 방법을 통해 가중치의 범위를 제한하지 않고 비밀 정보들을 99.99%까지 복구할 수 있음을 보였다.

전력 소비 신호와 같은 부채널 정보를 이용한 딥러닝 네트워크의 내부 비밀 정보 복구 공격이 증가하고 있어, 이에 대응하기 위한 연구도 활발히 진행되고 있다. 일반적으로 부채널 공격(side channel attack)에 대한 알고리즘적인 대응책으로 셔플링(shuffling) 기법과 마스킹(masking) 방법을 많이 사용하고 있다.

DNN(Deep Neural Network) 모델 복구에 대한 대응책으로 Liu 등[3]은 셔플링 기술을 사용하는 것을 제안하기도 하였다. 또한, Fang 등[4]은 다중 레벨 셔플링을 도입하여 성능 저하를 최소화하는 방법을 제안하였다. 이외에도 Athanasiou 등[5]은 신경망(neural network) 복구에 대한 대응책으로 마스킹 기법을 도입하여 부채널 공격에 대응할 수 있음을 확인하였다.

본 논문에서는 먼저 DNN 기반 MNIST (Modified National Institute of Standards and Technology) 숫자 분류기를 마이크로 컨트롤러에 구현한 후 상관 전력 분석 공격을 수행하여 딥러닝 가속기의 가중치가 복구됨을 확인하였다. 또한, 이러한 전력 분석 공격에 대응하기 위해 Node-CUT 셔플링 방법을 제안하였으며, 이는 전력 측정 시점의 정렬을 혼돈(misalignment)시키는 원리를 적용한 것이다. 제안하는 대응책은 전력 분석 공격을 충분히 방어할 수 있을 뿐만 아니라, Fisher-Yates 셔플링 기법을 사용하는 것보다 추가 연산량이 크게 감소함을 실험을 통해 확인하였다.

II. 배경 지식

본 장에서는 부채널 공격 실험에 사용된 MNIST 숫자 분류기 구현을 위해 필요한 딥러닝 설계와 관련한 주요 요소를 기술한다. 또한, 가중치 복구 실험을 위해 필요한 전력 신호를 기반으로 하는 부채널 공격 및 대응책에 관한 내용과 가중치를 표현하는 실수의 구성에 관해 기술한다.

2.1 딥러닝

딥러닝은 인공 지능의 한 분야로, 인공 신경망을 기반으로 한 머신러닝(machine learning) 알고리즘이다. 딥러닝은 다층 인공 신경망을 사용하여 입력 데이터로부터 복잡한 패턴을 학습하고, 이를 통해 데이터를 분석하고 예측한다. 딥러닝 알고리즘에는 다양한 구조가 존재하지만, 대표적으로 DNN이 있다.

DNN은 Fig. 1.과 같이 입력층(input layer)과 출력층(output layer) 그리고 그 사이의 여러 개의 은닉층(hidden layer)을 가진 신경망을 의미한다[6]. 이와 같이 신경이 깊게 연결된 네트워크 구조를 통해 복잡한 문제를 해결하고 다양한 특징을 추출할 수 있다. 즉, 은닉층과 출력층에서는 다음과 같은 연산을 수행한다.

$$h_j = a(w_{i,j} \cdot x_i + b_j)$$

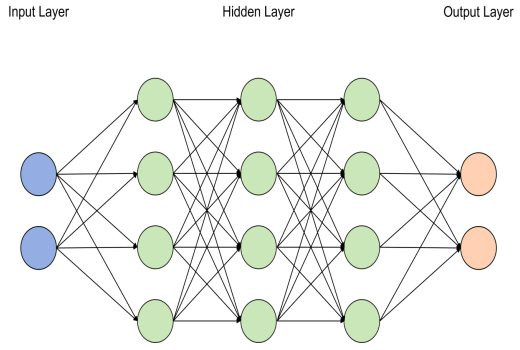

Fig. 1. DNN architecture

여기서 x_i는 입력 노드, $w_{i,j}$는 가중치, b_j는 편향 그리고 h_j는 출력 노드 그리고 a는 활성화 함수를 의미한다.

DNN은 역전파 알고리즘을 사용하여 가중치를 조정하면서 데이터를 학습한다. 역전파는 출력과 실제 값의 오차를 역방향으로 전파하여 각 가중치의 기여도를 계산하고, 경사 하강법을 통해 가중치를 업데이트하는 방식으로 학습한다. 이 과정을 반복하여 오차를 최소화하는 방법으로 입력과 출력 간의 관계를 학습하게 된다.

2.2 부채널 공격

부채널 공격은 디바이스가 동작할 때 발생하는 부채널 정보를 분석해 비밀 정보를 추출하는 공격이다. 부채널 정보에는 소비 전력, 전자기파, 프로그램 동작 시간 등이 있다. 부채널 정보 중 소비 전력을 분석해 공격하는 방법은 크게 단순 전력 분석(Simple Power Analysis), 차분 전력 분석(Differential Power Analysis), 상관 전력 분석으로 분류된다[7-9].

단순 전력 분석은 수집된 하나의 전력 소비 파형의 관찰을 통해 딥러닝 모델의 네트워크 구조, 사용된 노드 수를 복구하는 공격을 말한다. 차분 전력 분석은 수집된 다량의 전력 소비 파형을 비밀 정보와 연관된 그룹으로 분류한 후, 이 두 그룹 신호 평균의 차를 이용하여 가중치와 같은 디바이스 내부에 있는 비밀 정보를 복구하는 공격이다. 상관 전력 분석은 수집된 다량의 전력 소비 파형과 비밀 정보와의 상관도를 분석하여 비밀 정보를 복구하는 공격 방법이다.

2.3 딥러닝 네트워크 부채널 공격 대응책 관련 연구

최근 딥러닝 네트워크를 대상으로 부채널 공격을 진행해 내부 가중치, 파라미터, 모델 구조 등을 복원하기 위한 연구가 활발히 진행중이다. 이러한 부채널 공격에 대응하기 위해 셔플링과 마스킹을 이용한 연구도 진행되고 있다.

셔플링 대응책과 관련 대표 연구로는 Liu 등[3]의 연구가 있다. 해당 연구에서는 DNN 모델의 내부 비밀정보가 부채널 공격을 통해 복원 되는 것을 방지하기 위해 셔플링 기법을 사용하였다. 셔플링을 통해 패턴을 혼동시켜 내부 비밀 정보가 복원되는 것을 방지하였고, 공격 복잡성을 효과적으로 높이면서도 낮은 오버헤드를 유지하여 대규모 DNN 모델에 확장 가능함을 확인하였다.

셔플링을 이용한 다른 연구로는 Fang 등[4]이 있다. 해당 연구에서는 부채널 공격을 이용한 CNN 모델의 내부 비밀정보 복원 대응책으로 다중 레벨 셔플링을 사용한다. 다중 레벨 셔플링은 MAC 레벨과 PE 타일 레벨을 결합하여 CNN의 전력 사용 패턴과 데이터 처리 방식에 대한 부채널 공격에 대응하고, 내부 비밀정보를 추출하는 것을 어렵게 만든다.

마스킹 기법을 이용한 대표적인 연구는 Athanasiou 등[5]의 연구이다. 해당 연구에서는 임베디드 장치에서 동작하는 MLP, CNN, BNN 등 다양한 신경망이 전력 분석 공격을 통해 내부 비밀 정보를 복원할 수 있음을 보이고, 이에 대응하기 위해 마스킹 기법을 사용하였다. 마스킹 기법은 신경망의 파라미터와 중간 계산 결과에 임의성을 추가하여 공격자가 내부 비밀 정보를 추출하기 어렵게 만드는 기법이다. 이 기법이 성능 저하가 있지만 정확도에는 큰 영향을 미치지 않는 것을 확인하였다.

2.4 IEEE 754 32비트 단정밀도 표준

MNIST 분류기에서 사용되는 가중치는 실수 값으로 IEEE 754 32비트 단정밀도 표준으로 표현된다. IEEE 754는 컴퓨터에서 부동 소수점을 표현하는 가장 널리 쓰이는 표준으로, 크게 부호, 지수부, 가수부로 구성되고, 각각 1, 8, 23 비트로 구성된다.

MNIST 분류기에서 사용되는 가중치 -0.025550을 예로 들어 32비트 단정밀도 표준으로 작성하면 Fig. 2.의 형태가 된다. 해당 실수가 음수

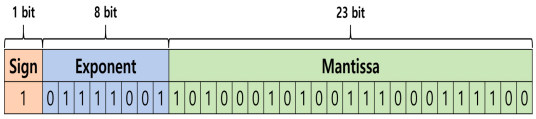

Fig. 2. IEEE 754 32bit floating point expression of an example -0.025550

이기 때문에 부호 비트는 1이 되고, 실수를 2진수로 표현하게 되면 $0.00000110\cdots_{(2)}$가 된다. 부동 소수점 표현에서는 정규화를 이용하기 때문에 $1.xx\cdots_{(2)}$의 형태로 변환해야 한다. 해당 값에 정규화를 적용하게 되면 $1.0111\cdots_{(2)} \times 2^{-6}$의 형태로 표현되고 부족한 비트 수는 0으로 채워진다. 이후 지수부에서는 편향이 더해지기 때문에 $127+(-6)=121$가 된다. 지수부 121을 2진수로 표현하게 되면 $01111001_{(2)}$이 된다.

III. 내부 가중치 복구 실험

본 장에서는 MNIST 숫자 분류기를 마이크로 컨트롤러에 구현한 후 딥러닝 가속기의 내부 정보인 가중치를 복구하는 방법을 기술한다. 또한, 상관 전력 분석 공격 실험을 통해 가중치가 복구 가능함을 보여 MNIST 분류기가 부채널 공격에 취약함을 증명한다.

3.1 가중치 복구 방법론

본 논문에서 공격 목표가 되는 가중치를 가수부, 지수부, 부호 비트 순서로 복구하였다. Park 등이 제시한 분할 정복 방식을 적용하였는데, 분할 정복 방식이란 긴 길이의 값을 중요한 부분부터 일정 크기로 나누어서 점차적으로 복구하여 전체 정보를 확인하는 과정을 의미한다.

전체 길이가 23비트인 가수부 복구는 Fig. 3.과 같이 복구할 상위 8비트를 공격 범위로 설정 후, 입력 데이터와 예측한 가중치의 가수부를 곱한 값의 해밍웨이트(Hamming weight)와 수집한 전력 소비 파형 간의 상관 계수를 계산한다. 여기서 상관 계수가 가장 높게 나온 예측 값의 최상위 5비트를 복구 값으로 사용하며, 동일한 방법으로 5번의 상관 전력 분석을 진행한다. 단, 마지막 5번째 상관 전력 분석에서는 최상위 3비트만 복구하면 전제 가수부를 복구할 수 있다.

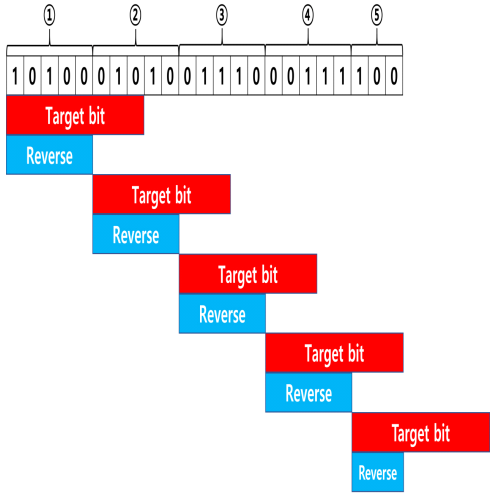

Fig. 3. Recovery order of the 23bit mantissa

먼저 23비트 가수부를 복구하고, 8비트 지수부를 복구한다. 지수부에 대한 부채널 정보 분석 공격은 이미 가수부 값을 알고 있고, 8비트 지수부는 공격자가 임의로 입력할 수 있다고 가정한다. 또한, 지수부 8비트가 사용되는 지점을 전력 신호 분석 영역으로 설정한다. 여기에서 주의할 점은 입력 데이터와 가중치의 곱해진 값의 지수부를 연산하기 위해서는 두 값의 지수부를 더하고, 곱해진 값의 가수부를 정규화할 때 발생하는 보정 값을 필요로 한다는 것이다.

가중치의 지수부와 가수부를 복구했다면 부호를 제외한 실수 값 구성이 가능하다. 부호 비트 복구 방법은 다음과 같다. 실수 값이 음수일 경우와 양수일 경우로 나누어 입력 데이터와 복구한 실수를 곱한 후 나온 중간 값과 전력 소비 파형의 상관 계수를 비교해 상관 계수가 높게 나온 값의 부호 비트를 가중치의 부호 비트로 결정한다.

3.2 네트워크 내부 가중치 복구 실험 환경

본 논문에서는 네트워크 내부 가중치 복구 실험을 위해 MNIST 숫자 분류기를 구현하였다. 이를 위해 DNN 구조를 사용하였고, 모델의 구조는 각 1개의 입력층, 은닉층, 출력층으로 구성하였다. 각 층의 노드 수는 입력층 784개, 은닉층 13개, 출력층은 10개로 구성하였다. 은닉층의 활성화 함수는 ReLU 함수, 출력층에는 Softmax 함수를 사용하였다. 또한 2.10.0 버전의 Keras 프레임워크를 사용해 DNN 모델을 구현하였고, 학습을 위해 MNIST 숫

자 데이터셋을 사용하였다.

MNIST 데이터셋은 총 70,000개의 데이터로 구성되어 있으며, 이 중 60,000개는 학습 데이터 10,000개는 테스트 데이터로 사용하였다. Batch 크기는 128로 설정하고, epoch를 20으로 설정하여 학습을 진행하였다. 학습 결과, 약 95.07%의 정확도로 이미지를 분류하는 것을 확인하였고, 테스트 이미지를 통해 모델 평가 진행 결과 약 94.19%의 정확도로 분류하는 것을 확인하였다.

학습된 모델에서 가중치와 편향 값을 확정하고, 이 값들을 사용하여 ARM-Cortex-M4 32bit STM32F MCU에 실제로 MNIST 숫자 분류기를 구현하였다. 실험에서 분류기의 동작을 확인하기 위해 Fig. 4.와 같이 7, 2, 1, 0, 4의 테스트 이미지를 사용하였다. 출력을 확인한 결과, 학습된 모델의 숫자 이미지와 MNIST 분류기의 이미지가 동일함을 확인하였다.

구현된 MNIST 분류기의 내부 가중치 복구를 위해, 은닉층의 첫 번째 노드 연산 중에서 첫 번째 가중치가 사용되는 지점을 공격 지점으로 선택하였다. 본 논문에서는 ChipWhisperer Lite를 사용해 분

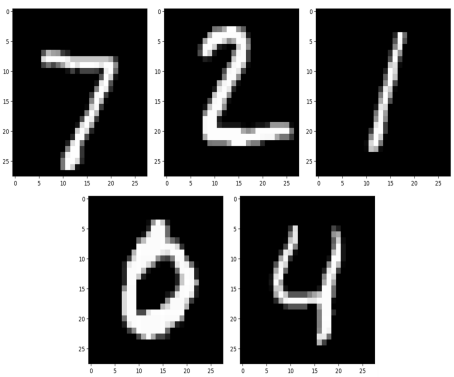

Fig. 4. Test image examples used in MNIST classifier

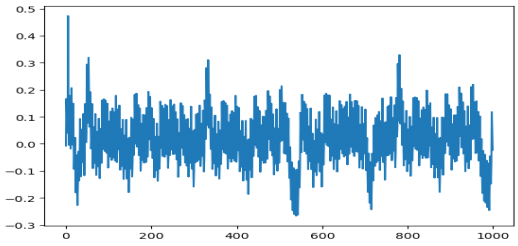

Fig. 5. Target points for power analysis attack

류기가 동작할 때 29.5MSamples/sec의 속도로 500개의 파형을 측정하였다. 공격 대상이 되는 POI(Points Of Interest)는 1,000개의 샘플을 사용하였으며, 이 영역의 소비 전력 파형을 나타낸 것이 Fig. 5.이다.

3.3 가중치 복구 실험 결과

MNIST 숫자 분류기를 대상으로 분할 정복 방식을 사용해 가중치 복구 실험을 진행하였다. 가수부 23비트 복구 단계에서 예측한 8비트 중 전력 소비 파형과의 상관 계수가 가장 높은 값 5비트를 복구하는 과정을 반복하여 가수부를 모두 복구하였다. 가수부에 대한 자세한 복구 실험 결과를 나타낸 것이 Fig. 6.이다.

지수부 8비트는 복구된 가수부와 입력 데이터의 가수부를 곱한 후, 정규화 과정에서 발생하는 보정

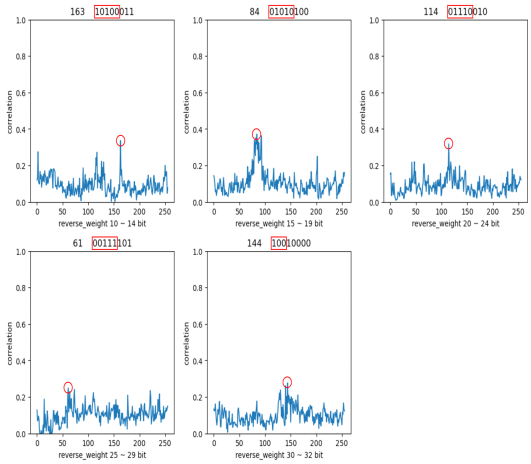

Fig. 6. Recovery of 23-bits mantissa

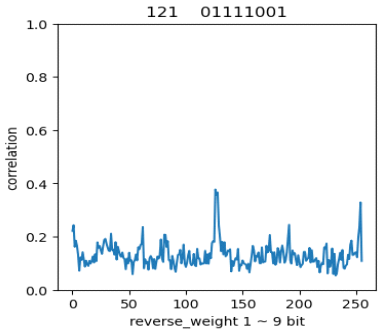

Fig. 7. Recovery of 8-bits exponent

값과 예측한 지수부를 연산한 결과를 얻는다. 이후, 이 결과와 전력 소비 파형과의 상관 계수를 구하여 가장 높게 나온 예측 값으로 결정한다. 다음 Fig. 7.과 같이 실제 가중치의 지수부와 동일한 값이 복구되는 것을 실험적으로 확인하였다.

IV. 전력 분석 공격에 대한 셔플링 대응책

본 장에서는 MNIST 숫자 분류기에 대한 전력 분석 공격에 대한 대응책으로 Node-CUT 셔플링을 제안한다. 데이터 셔플링을 위해 일반적으로 사용되는 Fisher-Yates 기법을 이용하는 것과 비교하여 공격에 대한 방어 능력과 동작 성능을 마이크로 컨트롤러 보드 실험과 컴퓨터 시뮬레이션을 통해 확인해 본다. 참고로 딥러닝 가속기에 사용되는 값이 부동소수점인 것을 고려하면 마스킹을 이용한 대응 기법 적용하기는 어렵다.

4.1 Fisher-Yates 셔플링

셔플링 기법은 나열된 일련의 데이터를 무작위로 섞는 방법을 말한다. 딥러닝 가속기의 내부 가중치 복구의 경우, 딥러닝 가속기가 동작할 때 발생하는 전력 소비와 예측한 중간 값 사이의 상관도를 구하고 복구한다. 따라서 공격자가 중간 값을 예측할 수 없게 만드는 셔플링 과정을 통해 공격에 대응한다[10-11].

Fisher-Yates 셔플링 알고리즘[12]은 배열 혹은 리스트의 요소들을 무작위로 섞는 데 사용되는 효율적이고 널리 사용되는 알고리즘 중 하나인데, 동작 과정을 간단히 도시한 것이 Fig. 8.이다.

먼저 필요한 랜덤 수만큼의 배열을 생성한다. 그 다음 배열의 마지막 인덱스를 선택하고, 나머지 인덱스 중에서 랜덤하게 하나를 선택하여 두 값을 교환하고, 마지막 인덱스의 값을 결정한다. 이 과정을 배열의 끝에서부터 시작하여 처음까지 반복한다. 예를 들어, 필요한 랜덤 수가 5개라면 다음과 같이 진행된다. 초기 배열은 {0, 1, 2, 3, 4}이다. 먼저 배열의 마지막 인데스 4를 선택하고, 나머지 인덱스 중 랜덤하게 선택한 인덱스와 값을 교환한다. 예를 들어, 2번 인덱스를 선택하여 2와 4를 교환하고, 배열은 {0, 1, 4, 3, 2}가 된다. 그 다음, 인덱스를 하나 줄여 3을 선택하고, 나머지 인덱스 중에서 다시 랜덤하게 선택한 인덱스 0과 값을 교환하고. 배열은 {3, 1, 4, 0, 2}가 된다. 이러한 과정을 배열의 끝에서부터 처음까지 반복하여 전체 배열을 무작위로 섞게 되면 Fig. 8.과 같은 {4, 3, 1, 0, 2} 배열을 얻게 된다.

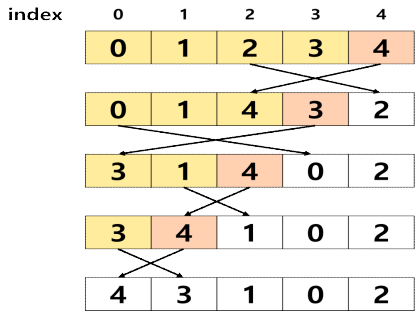

Fig. 8. Fisher-Yates shuffling

Fisher-Yates 셔플링을 MNIST 분류기에 적용하는 과정은 다음과 같다. 먼저, 각 층별 노드 수에 해당하는 배열을 생성하고, Fisher-Yates 셔플링을 통해 무작위로 섞는다. 그 후, 무작위로 섞인 순서에 따라 입력 데이터와 가중치를 가져와 연산을 수행하여 연산 순서를 랜덤화 시킨다.

Fisher-Yates 셔플링이 적용된 MNIST 분류기가 정상적으로 동작하는지 확인을 위해 Fig. 4.의 이미지로 검증을 진행하였고, 동일한 출력이 나타남을 확인하였다. 또한, 이전에 수행한 동일한 상관 전력 분석을 통해 가중치 복구 공격을 수행해 본 결과, Fig. 9.에서 보는 바와 같이 가수부 복구가 불가능한 것을 확인하였다.

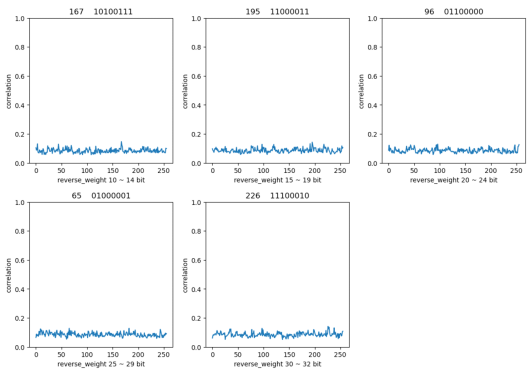

Fig. 9. Weight recovery attack on MNIST classifier applied with Fisher-Yates shuffling countermeasure

4.2 Node-CUT 셔플링

상기한 Fisher-Yates 셔플링 기법은 부채널 분석 공격을 방어할 수 있지만, 요구되는 연산량이 많아진다. 따라서, 본 논문에서는 Node-CUT 셔플링 방식을 제안하고자 한다. 이 대응 방법의 원리는 공격자가 가정한 중간 값을 계산하는 위치를 랜덤하게 결정함으로써 일종의 전력 측정 시점을 정렬(alignment)하기 어렵게 하는 것이다.

Node-CUT 셔플링은 각 층별로 하나의 랜덤 수를 사용해 연산의 시작 지점을 랜덤화하는 방법을 말한다. 다음 Fig. 10.은 계산하는 노드를 랜덤 수가 2일 때 Node-CUT 셔플링의 동작 방식을 나타낸 것인데, 한 개의 랜덤 수만으로 셔플링이 가능하다. 이와 비교하여 상기한 Fisher-Yates 셔플링은 모든 노드의 개수 만큼의 랜덤 수와 저장 공간을 필요로 하고 있어 그만큼 성능 저하가 예상된다.

다음 Fig. 11.은 Node-CUT 셔플링을 MNIST 숫자 분류기 모델에 적용한 알고리즘이다. 이 알고리즘에서 가장 먼저 수행되는 단계는 입력층과 은닉층 노드의 연산 시작 지점 랜덤화를 위해 랜덤 수를 생성하는 것이다. 여기서는 입력층은 784개, 은닉층은 13개, 출력층은 10개의 노드를 가정하였다. 생성된 랜덤 수를 반복문의 초기 값으로 사용하고, 입력 데이터의 해당 인덱스 값을 가져와 은닉층의 노드 값을 계산한다. 위 과정과 동일하게 은닉층과 출력층 사이의 연산 시작 지점도 랜덤화할 수 있다. 이러한 과정을 통해 출력층 노드의 값을 계산한 후 Softmax 함수를 통과하여 MNIST 숫자 이미지를 가장 확률이 높은 클래스로 분류한다.

Node-CUT 셔플링이 적용된 MNIST 숫자 분류기를 Fig. 4.의 이미지로 검증을 진행한 결과, 분류기는 정상적으로 동작하였다. 또한, 동일한 전력 분석 환경에서 상관 전력 분석을 이용하여 가중치 복구 공격을 수행해 본 결과, Fig. 12.에서 볼 수 있듯이

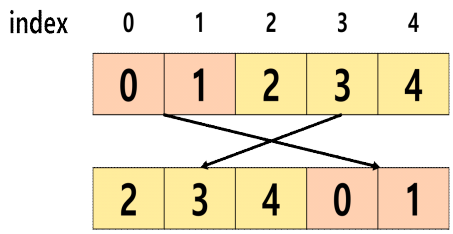

Fig. 10. Node-CUT shuffling

Algorithm 1: Node-CUT shuffling forward

Input: *input* I, *weight* w_1, w_2, *bias* b_1, b_2

Output: ouput O

```
hiddenStart ← rand(0, 13)
for i = hiddenStart to (hiddenStart+13) do
    tmp1 ← i % 13
    inputStart ← rand(0, 784)
    for j = inputStart to (inputStart+784) do
        tmp2 ← j % 784
        hiddenNode[tmp1] += i[tmp2] × w1[tmp1][tmp2]
    hiddenNode[tmp1] ← relu(hiddenNode[tmp1]+b1[tmp1])
outputStart ← rand(0, 10)
for i = outputStart to (outputStart+10) do
    tmp1 ← i % 10
    hiddenStart ← rand(0, 13)
    for j = hiddenStart to (hiddenStart+13) do
        tmp2 ← j % 784
        outputNode[tmp1] += hiddenNode [tmp2] × w1[tmp1][tmp2]
    outputNode [tmp1] ← relu(outputNode[tmp1]+b1[tmp1])
O = softmax(outputNode)
```

Fig. 11. MNIST classifier model applied with Node-CUT shuffling

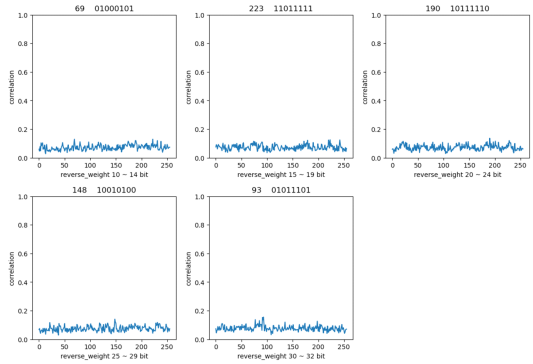

Fig. 12. Weight recovery attack on MNIST classifier applied with Node-CUT shuffling countermeasure

가수부 복구가 불가능한 것을 확인하였다.

4.3 컴퓨터 시뮬레이션 및 효율성

본 절에서는 일반적인 DNN 네트워크 구조에서 부채널 공격에 대응하기 위한 방법의 성능을 컴퓨터 시뮬레이션을 통해 살펴보고자 한다. 은닉층에서 볼

때 이전 입력층의 노드 수가 m이고, 은닉층의 노드 수가 n이라면, 은닉층에서 계산하는 각 노드를 랜덤하게 정할 수 있고, 입력층에서도 가중치 값을 랜덤하게 곱할 수 있기 때문에 이중으로 노드의 계산 위치를 변경할 수 있다.

Fisher-Yates 셔플링 기법에서는 각각 $m+n$개의 셔플링 위치를 결정하는 랜덤한 정수 값을 생성하고 이를 저장하고 있어야 한다. 이 경우 공격자가 특정한 공격 지점을 맞출 수 있는 확률은 $1/(2^m \cdot 2^n)$이 된다.

반면, Node-CUT 셔플링 기법에서는 2개의 셔플링 위치를 결정하는 랜덤한 정수 값만 저장하고 있으면 된다. 이 경우 공격자는 특정한 공격 지점을 맞출 수 있는 확률은 $1/(m \cdot n)$이 된다. 공격자가 상관 전력 분석을 통해 고정된 노드의 가중치 값을 복구하는데 500개의 파형이 필요하다면 Node-CUT 셔플링 기법은 최소한 그보다 $(m \cdot n)$배의 파형이 필요하게 된다.

다음 Table 1과 같은 DNN 네트워크 구조를 가정했을 경우, Fisher-Yates 셔플링과 Node-CUT 셔플링 방법의 성능을 비교하였다. 모델의 크기에 따른 셔플링 성능 평가를 위해 테스트 데이터 10,000개를 입력 데이터로 사용하고, 은닉층의 노드 수를 각각 100개, 500개, 1,000개로 설정하여 MNIST 숫자 분류기가 동작하는 시간을 측정 후 평균을 구해 비교하였다.

다음 Table 2는 은닉층의 노드 수별로 MNIST 숫자 분류기의 동작 시간을 나타낸 것이다. 단, 최종적인 숫자 이미지 분류 성능은 대응책의 적용 여부와 관계없이 94.19%로 동일하였다. 표에서 보는 바와 같이 은닉층 노드 수 1,000개를 기준으로 Fisher-Yates 셔플링은 대응책이 적용되지 않은 분류기와의 성능을 비교해 보면 약 32.8%의 추가 연산이 필요함을 확인할 수 있었다.

Table 1. Simulation environment for performance evaluation of MNIST classifier

CPU	AMD Ryzen 5 3500 6-Core
RAM	16GB
GPU	NVIDIA GeForce RTX 2060
OS	Windows 10
Language	Python 3.9.15
IDE	JupyterNotebook
Library	Keras 2.10.0 numpy 1.21.5 matplotlib 3.6.2

Table 2. Comparison of operational time applied with shuffling countermeasures

Countermeasures	Hidden Layer Node	Time(s)
None	100	0.0499
	500	0.2493
	1000	0.4973
Fihser-Yates Shuffling	100	0.0643
	500	0.3334
	1000	0.6602
Node-CUT Shuffling	100	0.0544
	500	0.2736
	1000	0.5430

이에 비해 Node-CUT 셔플링은 대응책이 적용되지 않은 분류기에 비해 약 9.18%의 추가 연산이 필요하였다. 따라서 본 논문에서 제안하는 Node-CUT 셔플링이 성능 저하가 거의 없이 부채널 공격에 효율적으로 대응 가능함을 확인하였다.

V. 결 론

본 논문에서는 MNIST 데이터셋으로 학습한 숫자 분류기 모델을 구현한 딥러닝 가속기는 부채널 공격에 취약하여 내부 비밀 정보를 노출할 수 있음을 확인하였다. ARM-Cortex-M4 32bit STM32F 마이크로 컨트롤러에 구현된 딥러닝 가속기가 동작할 때 발생하는 전력 소비 파형을 이용한 상관 전력 분석 공격을 이용하여 가중치 정보를 복구할 수 있었다.

또한, 부채널 공격에 대응하기 위해 Node-CUT 셔플링을 제안하고 이 대응책이 동일 환경하에서 상관 전력 분석 공격에 대응할 수 있음을 실험적으로 확인하였다. Node-CUT 셔플링 대응 기법은 기존 랜덤화 시스템에 많이 사용하던 Fisher-Yates 셔플링 기법과 비교하여 적은 랜덤 수 발생과 저장 공간을 사용함으로써 딥러닝 가속기에 대한 실용적인 공격 대응책으로 활용 가능하다.

추후 연구로 제안한 대응책을 CNN, BNN 등 다양한 모델에 적용하고, 해당 기술이 적용된 모델들의 성능과 정확도의 영향이 없는 것을 확인한다. 또한 모델의 크기별 발생할 수 있는 취약점에 대한 연구 진행이 필수적이다.

References

[1] L. Batina, S. Bhasin, D. Jap, and S. Picek, "CSI NN: Reverse engineering of neural network architectures through Electromagnetic side channel," Proceedings of the 28th USENIX Conference on Security Symposium, pp. 515-532, Aug. 2019.

[2] S. Park, J. Lee and H. Kim, "Reverse engineering of eeep learning network secret information through side channel," Korea Institute of Information Security & Cryptology, Vol. 32, No 5, pp. 855-867, Oct. 2022.

[3] Y. Liu, D. Dachman-Soled and A. Srivastava, "Mitigating reverse engineering attacks on deep neural networks," 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 657-662, Sep. 2019.

[4] Q. Fang, L. Lin, H. Zhang, T. Wang and M. Alioto, "Voltage scaling-agnostic counteraction of side-channel neural net reverse engineering via machine learning compensation and multi-level shuffling," 2023 IEEE Symposium on VLSI Technology and Circuits, pp. 1-2, Jun. 2023.

[5] K. Athanasiou, T. Wahl, A. Ding and Y. Fei, "Masking feedforward neural networks against power analysis attacks," proceedings on privacy enhancing technologies, pp. 501-521, Jan. 2022.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature 521, pp 436-444. May. 2015.

[7] S. Mangard, E Oswald, and T. Popp, "Simple power analysis - power analysis attacks revealing the secrets of smart cards," pp. 101-118, Springer-Verlag, Jan. 2008

[8] P. Kocher, J. Jaffe and B. Jun, "Differential power analysis," Advances in Cryptology, CRYPTO' 99, LNCS 1666, pp. 388-397, Dec. 1999.

[9] E. Brier, C. Clavier and F. Olivier, "Correlation power analysis with a leakage model," CHES'04, LNCS 3156, pp. 16-29, Aug. 2004.

[10] A. G. Bayrak, N. Velickovic and P. Ienne, "An architecture-independent instruction shuffler to protect against side-channel attacks," ACM Transactions on Architecture and Code Optimization (TACO), pp. 1-19, Jan. 2012.

[11] N. Veyrat-Charvillon, M. Medwed, S. Kerckhof, and F. Standaert, "Shuffling against side-channel attacks: a comprehensive study with cautionary note," Advances in Cryptology - ASIACRYPT, pp. 740-757, Dec. 2012.

[12] R. A. Fisher and F. Yates, "Statistical table for bio, agriculture, and medical research," 6th Ed., Oliver & Boyd, 1963.

〈저자소개〉

이 영 주 (Youngju Lee) 학생회원
2018년 3월~현재: 호서대학교 컴퓨터공학부 학사과정
〈관심분야〉 부채널 공격, 양자내성 암호, 인공지능 보안

이 승 열 (Seungyeol Lee) 학생회원
2018년 3월~현재: 호서대학교 컴퓨터공학부 학사과정
〈관심분야〉 자동차 보안, 양자내성 암호, 인공지능 보안

하 재 철 (Jaecheol Ha) 종신회원
1989년 2월: 경북대학교 전자공학과 학사
1993년 8월: 경북대학교 전자공학과 석사
1998년 2월: 경북대학교 전자공학과 박사
1998년 3월~2007년 2월: 나사렛대학교 정보통신학과 교수
2007년 3월~현재: 호서대학교 컴퓨터공학부 교수
2009년 1월~현재: 한국산학기술학회 이사
1993년 1월~현재: 한국정보보호학회 수석부회장
〈관심분야〉 암호학, 부채널 공격, 네트워크 보안, 정보보호