Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.5, Oct. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.5.855

부채널 분석을 이용한 딥러닝 네트워크 신규 내부 비밀정보 복원 방법 연구*

박 수 진,[1†] 이 주 헌,[1] 김 희 석[2‡]
[1,2]고려대학교 (학생, 교수)

Reverse Engineering of Deep Learning Network Secret Information Through Side Channel Attack*

Sujin Park,[1†] Juheon Lee,[1] HeeSeok Kim[2‡]
[1,2]Korea University (Student, Professor)

요 약

IoT 장비의 발달로 딥러닝 가속기의 필요성이 증대됨에 따라 이에 탑재되는 딥러닝 가속기의 구현 및 안전성 검증에 대한 연구가 활발히 진행 중이다. 본 논문에서는 Usenix 2019에 발표된 딥러닝 네트워크 복원 논문의 한계점을 극복한 내부 비밀정보 신규 부채널 분석 방법론에 대해 제안한다. 기존 연구에서 네트워크 내부 가중치의 범위를 제한하며 32비트 가중치의 16비트만 복원한 단점이 있다. 제안하는 신규 가중치 복원 방법으로 상관전력분석을 이용하여 IEEE754 32비트 단정밀도 가중치를 99% 정확도로 복원할 수 있음을 보인다. 또한 특정 입력값에 대해서만 활성함수 복원이 가능한 기존 연구의 제약을 극복하고, 딥러닝을 이용한 신규 활성함수 복원 방법으로 입력값에 대한 조건 없이 99% 정확도로 활성함수를 복원한다. 이를 통해 기존 연구가 가지는 한계점들을 극복했을 뿐만 아니라 제안하는 신규 방법론이 효과적이라는 것을 입증한다.

ABSTRACT

As the need for a deep learning accelerator increases with the development of IoT equipment, research on the implementation and safety verification of the deep learning accelerator is actively. In this paper, we propose a new side channel analysis methodology for secret information that overcomes the limitations of the previous study in Usenix 2019. We overcome the disadvantage of limiting the range of weights and restoring only a portion of the weights in the previous work, and restore the IEEE754 32bit single-precision with 99% accuracy with a new method using CPA. In addition, it overcomes the limitations of existing studies that can reverse activation functions only for specific inputs. Using deep learning, we reverse activation functions with 99% accuracy without conditions for input values with a new method. This paper not only overcomes the limitations of previous studies, but also proves that the proposed new methodology is effective.

Keywords: Power Analysis Attack, Side-Channel Analysis, CPA, Deep Learning, MLP

Received(08. 04. 2022), Modified(09. 30. 2022), Accepted(10. 04. 2022)

* 본 연구는 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행되었습니다. (No.2021-0-00903, 고신뢰 온-디바이스 딥러닝 가속기 설계를 위한 물리채널 기반 취약점 검증 및 대응기술 개발)

† 주저자, lemontrees33@korea.ac.kr

‡ 교신저자, 80khs@korea.ac.kr(Corresponding author)

I. 서 론

4차 산업혁명 시대를 대표하는 머신러닝(machine learning) 기술은 대량의 데이터를 학습하여 결과를 추론하는 방법이다. 머신러닝 기술 중 하나인 딥러닝(deep learning)은 인간의 두뇌 구조를 모방한 인공 신경망 네트워크를 기반으로 연산을 수행한다[1]. 이러한 딥러닝 기술은 음성 인식, 이미지 인식 등의 분야에서 뛰어난 성능을 보이고 있으며 다양한 분야에 접목하려는 연구가 활발히 진행중이다.

특히 IoT(Internet of Things) 장비 및 서버에서 연산을 고속으로 처리하고 저전력 환경에서 동작할 수 있는 하드웨어가 요구됨에 따라 딥러닝 가속기가 등장하였다. 딥러닝 가속기는 딥러닝 네트워크 구현에 필요한 대규모 연산을 효율적으로 처리할 수 있는 반도체로 기존의 범용적인 하드웨어의 1000배 이상의 효율을 보여주고 있다. 이러한 획기적인 성능 때문에 자율주행 자동차, 스마트폰, 클라우드 등 다양한 분야에서 딥러닝 가속기를 필수적으로 탑재하는 추세이다.

이에 따라 딥러닝 가속기의 안전성에 대한 연구 필요성 또한 증대되었다. 전력 및 전자파 분석과 같은 부채널 분석으로 딥러닝 가속기의 내부 비밀 정보를 탈취하거나, 오류 주입과 같은 공격 기법을 이용하여 암호가 구현된 FPGA의 비밀키를 찾는 것이 가능하기 때문이다[18]. 학습된 네트워크 및 내부 비밀정보 탈취 시 임의의 입력을 사용하여 얼굴인식 보안시스템에서 사용자 안면인식 인증 우회와 같은 공격이 이루어질 수 있다. 딥러닝 가속기의 안전성에 대해 Weizhe Hua 등[19]이 최초로 언급한 이후 이에 관하여 많은 연구가 진행되었으며, 그 중 부채널 분석을 이용한 공격이 활발히 연구되고 있다[17].

딥러닝 네트워크 부채널 분석은 타이밍 기반과 전력 및 전자기파 분석을 바탕으로 이루어진다. 최근 연구 동향은 MLP(Multi Layer Perceptron), CNN(Convolutional Neural Network), BNN(Binarized Neural Network) 모델을 공격 대상으로 설정하며, MNIST/CIFAR-10/Image Net과 같은 이미지 데이터 셋(data set)을 사용한다. 주로 네트워크 내부 구조 및 파라미터(parameter) 복원과 관련한 연구가 활발하게 진행중이다[8-14]. 대표적으로 Batina 등[8]은 MLP와 CNN 모델을 공격 대상으로 MNIST 데이터 셋을 사용하여 모델의 가중치, 활성함수 및 레이어(layer)와 뉴런(neuron)의 수 등을 높은 정확도로 복원하였다.

그러나 내부 가중치 복원에서 추측하는 가중치의 범위를 제한하여 실험을 수행하였으며, 활성함수 복원에서 입력값의 범위를 제한한 점에서 비현실적인 상황에서의 실험이라고 볼 수 있다. 본 논문에서는 기존 연구의 비현실적인 가정 없이 최소한의 부채널 정보를 이용하여 네트워크 내부 가중치 및 활성함수를 복원하는데 성공했음을 증명한다.

제안하는 신규 가중치 복원에서 가중치에 대한 범위 설정 없이 모든 경우에 대해 실수 가중치를 복원한다. 기존 가중치 복원 방법은 32비트 가중치의 최상위 16비트만 복원하며 정밀도를 0.01로 설정하여 실제 상황과는 맞지 않았으나, 제안하는 방법에서는 32비트 실수 전체를 정밀도 설정없이 분할정복(divide-and-conquer)방식으로 복구하여 더 높은 정확도로 정교하게 값을 복원할 수 있음을 보인다. 제안하는 신규 활성함수 복원에서 입력값에 대한 조건 및 활성함수 연산시간 분포 템플릿(template) 없이 딥러닝 모델을 이용하여 활성함수를 복원한다. 기존 활성함수 복원 방법은 연산시간 분포 템플릿이 필요하며 템플릿과 동일한 범위의 입력값에 대해서만 복원이 가능했으나, 제안하는 방법은 이러한 한계점을 극복하고 활성함수 파형 하나만으로 해당 활성함수의 종류를 복원할 수 있음을 증명한다.

본 논문의 2장에서는 관련 배경지식에 대해 서술하고, 3장에서는 상관전력분석(Correlation Power Analysis, CPA)을 이용하여 분할정복 방식으로 내부 실수 가중치를 복원한다. 4장에서는 학습된 CNN 모델을 이용하여 활성함수를 복원한다. 5장에서 결론 및 추후 연구 방향에 대해 서술한다.

II. 배경 지식

본 절에서는 딥러닝과 MLP, CNN 모델에 대한 소개 후, 제안하는 방법론의 이해를 돕기 위해 부채널 분석에 대해 설명한다. 이후 부채널 분석을 이용한 딥러닝 네트워크의 복원 대상에 따른 연구 최신 동향을 서술한다.

2.1 딥러닝(Deep Learning)

딥러닝은 인공신경망을 이용하여 대량의 데이터를 학습하는 알고리즘으로 기존 머신러닝과 달리 학습 데이터에 대한 특징을 자동으로 추출한다는 장점이 있다. 인공 신경망의 가장 기본 단위인 퍼셉트론(perceptron)은 입력층과 출력층을 가지고 있어 하나의 퍼셉트론 노드에 입력값이 들어오면 출력값을 생성한다. 출력값은 입력층에서 받은 입력값을 가중치와 곱하고 편향을 더한 후 활성함수의 입력으로 넣은 결과이다.

딥러닝에는 MLP, CNN 등 다양한 형태의 모델이 존재하며 학습 데이터에 의존하여 딥러닝 모델을 선택한다. Fig. 1.과 같이 MLP는 퍼셉트론이 여러 층으로 구성된 가장 기본적인 딥러닝 네트워크이며 입력층, 은닉층, 출력층으로 구성된다. 다층 퍼셉트론은 기존 단층 퍼셉트론이 해결하지 못한 비선형성 문제를 은닉층을 이용하여 해결하였다. MLP는 구조가 단순하여 처리 속도가 빨라 1차원 데이터 처리에 용이하다.

CNN은 MLP 구조에서 2차원 이상의 데이터를 처리하기 위한 컨볼루션 레이어(convolution layer)와 풀링 레이어(pooling layer)를 추가한 구조의 딥러닝 네트워크 모델이다. Fig. 2.와 같이 컨볼루션 레이어와 풀링 레이어의 반복 이후 MLP 구조가 나오는 것을 볼 수 있다. 컨볼루션 레이어는 커널(kernel)을 이용하여 입력 데이터의 특징을 추출한다. 풀링 레이어는 데이터 압축을 수행하여 대푯값을 추출한다. 이처럼 CNN은 특수한 2개의 레이어를 사용하여 2차원 이상의 데이터를 가공하여 효율적으로 분류할 수 있는 특징이 있다.

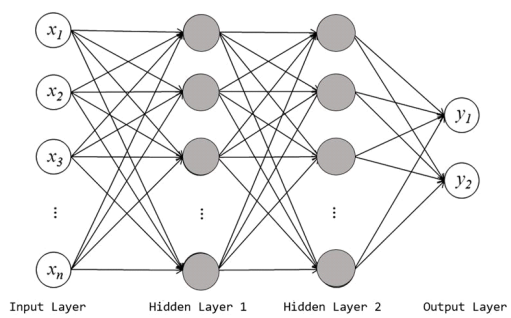


Fig. 1. MLP architecture

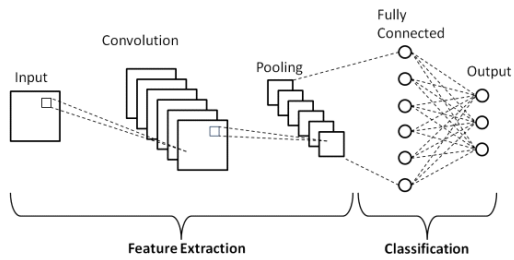


Fig. 2. CNN architecture

2.2 부채널 분석(Side Channel Analysis)

부채널 분석은 전자기기가 동작할 때 발생하는 소비전력, 전자파, 시간 등의 부채널 정보를 이용하여 내부 비밀 정보를 복원하는 기술이다. 복원하는 방법에 따라 단순전력분석, 차분전력분석, 상관전력분석 등으로 나뉘는데, 그 중 가장 대표적인 방법이 상관전력분석이다[3,4,5]. 상관전력분석은 다수의 파형을 이용하는 통계적 분석을 통해 내부 비밀 정보를 복원하는 방법이다. 전자기기에서 수집한 부채널 정보와 공격자가 추측한 중간값과의 상관계수를 계산하여 가장 유의미한 결과를 도출한 중간값을 내부 비밀 정보로 결정한다.

최근에는 딥러닝 기술의 발전으로 다양한 분야에 접목하려는 연구가 활발히 진행 중인데 부채널 분석 분야에서는 딥러닝 기반 프로파일링(profiling) 부채널 분석 기법이 등장했다[6,7]. 프로파일링 부채널 분석이란 공격자가 공격 대상 장비와 동일한 프로파일링 장비를 소유한 상태에서 원하는 입력값에 대한 중간값, 출력값, 부채널 정보를 관찰할 수 있는 환경에서의 공격을 의미한다. 공격자는 수집한 중간값과 부채널 정보를 대응시키는 프로파일링 과정을 거쳐 실제 공격 대상 장비의 내부 비밀정보를 소수의 부채널 정보만으로 복원할 수 있다.

딥러닝을 이용한 프로파일링 부채널 분석은 인공신경망을 이용하여 프로파일링 과정을 수행하며 학습 단계와 추론 단계로 나눠진다. 학습 단계에서는 프로파일링 장비에서 발생한 부채널 정보에 라벨(label)을 붙여 인공 신경망을 학습시킨다. 추론 단계에서는 공격 대상 장비에서 발생한 부채널 정보를 학습된 딥러닝 모델에 입력으로 넣어 계산된 출력값을 통하여 내부 비밀정보를 알아낸다.

2.3 딥러닝 네트워크에 대한 부채널 공격

딥러닝 네트워크에 대한 부채널 공격은 복원 대상을 기준으로 크게 내부 파라미터 및 내부 구조 복원으로 나눌 수 있다. 내부 파라미터의 경우 내부 가중치, 입력값, 활성함수 등 네트워크를 구성하는 다양한 요소가 복원 대상이 될 수 있다. 내부 구조의 경우 네트워크 층의 수나 뉴런의 수가 복원 대상이다.

딥러닝 네트워크 내부 파라미터 복구와 관련된 주요 최신 연구 동향을 소개하면 다음과 같다. Yoshida 등[10]은 네트워크의 내부 구조를 알고 있는 환경에서 Chain CPA를 제안하여 가중치 정보를 복구하였다. 동일 저자의 추후 연구에서 3차 상관전력분석 및 단순전력분석을 수행하였으나, 한계점으로 단순전력분석의 정확도가 낮다는 것을 언급하였다[11]. Maji 등[12]은 그레이 박스(gray-box) 환경에서 CNN과 BNN 모델의 가중치, 편향, 입력값을 복구하며 제안하는 기술은 측정된 파형의 SNR(Signal-to-Noise Ratio) 및 모델의 복잡성을 최소화할 수 있음을 입증하였다. 그러나 그레이 박스 환경은 공격 대상 네트워크의 구조를 공격자가 미리 알고 있다는 점에서 제안하는 블랙 박스(black-box) 환경에서의 실험에 비해 단점으로 작용할 수 있다.

딥러닝 네트워크의 활성함수 복원에 대한 최근 연구에서는 블랙 박스 환경에서 단일 전자기파 부채널 분석을 이용하여 MLP의 활성함수를 복원하였다[13]. 기존 타이밍 기반 활성함수 복원 방법과 비교하여 제안하는 방법은 입력값에 영향을 받지 않으며, 더 적은 측정으로 분석이 가능하다. 또한 활성 함수의 일정하지 않은 타이밍 기반 특징에 덜 의존한다는 특성을 가진다.

기존 딥러닝 네트워크 가중치 복원 연구는 (-5, -5)의 가중치 범위를 설정하고 실험을 수행하였다. 이는 실제 가중치 범위의 매우 한정적인 영역이므로 현실적이지 않은 가정이다. 더불어 실수 가중치 32비트 중 16비트만 복원하여 복원 정확도를 낮추었다. 활성함수 복원에서도 (-2, 2)라는 매우 제한적인 입력값의 범위를 설정하였으며 복원 방법에도 한계점이 있다. 입력에 따른 활성함수 연산시간 분포를 이용하게 되면 충분한 입력이 필요할 뿐 아니라 고른 입력과 제한된 범위의 입력만 사용해야 한다는 점이다. 마지막으로 활성함수의 입력값을 구하기 위해서는 가중치 복원이 반드시 선행되어야 한다는 단점이 있다. 이는 독립적인 복원이 아닌 가중치 복원 결과에 따라서 활성함수 복원 결과에 영향을 미친다는 뜻이다.

이외에도 딥러닝 네트워크의 입력값 복원에 대해 화이트 박스(white-box)환경에서 CNN과 BNN 모델의 입력값 및 모델 파라미터 정보를 복원하는 타이밍 기반 SPA(Simple Power Analysis) 방법론이 발표되었다[12]. 또한 기존 연구에서 딥러닝 네트워크의 구조는 공격을 성공하는데 매우 중요하다는 점이 입증됨에 따라 DNN(Deep Neural Network) 모델의 특징과 계층 간 시간 연관 가능성을 모두 고려한 공격이 제안되었다[14].

III. 딥러닝 네트워크 신규 내부 가중치 복원기술

본 장에서는 부채널 분석을 이용하여 딥러닝 네트워크의 내부 가중치를 복원하기 위한 신규 방법론에 대해 설명한다. 먼저 제안하는 알고리즘의 이해를 돕기 위해 기호를 정의하고, 신규 복원 방법론을 제안한다. 이후 설정한 실험 환경에 대해 간략히 소개하고, 실험 결과를 분석한다.

3.1 기호 정의

제안하는 신규 내부 가중치 복원 방법론의 알고리즘 이해를 돕기위한 기호를 Table 1.에 정의한다.

Table 1. Notation

Notation	Description
N	Number of traces
I	32bit float input value $I((i_0 i_1 \dots i_{31})_{32})$
W	32bit weight value $W((w_0 w_1 \dots w_{31})_{32})$
HW	Hamming weight
S	traces data
key	key with absolute maximum correlation coefficient
I_m	Single input mantissa bits of I $I_m((i_9 i_{10} \dots i_{31})_{23})$
W_m	Single weight mantissa bits of W $W_m((w_9 w_{10} \dots w_{31})_{23})$

W_e	Single weight exponent bits of W $W_e((w_1w_2 \ldots w_8)_8)$
W_s	Single weight sign bit of W $W_s((w_0)_1)$
$W_s^\star$	Several weights sign k-bit of W $W_s((w_0w_1 \ldots w_{k-1})_k)$
O	Result of multiplying input mantissa 23bit by target weight mantissa k-bit $O((o_9o_{10} \ldots o_{31})_{23})$
ard	Add result digit
mrd	Multiply result digit
I_md	Input mantissa digit
W_md	Weight mantissa digit
c	Correction value
re	Result exponent value

3.2 내부 가중치 신규 복원 방법론

3.2.1 IEEE 754 32비트 단정밀도 표준

IEEE 754는 컴퓨터에서 부동소수점을 표현하기 위해 가장 널리 쓰이는 표준으로, 32비트 단정밀도나 64비트 배정밀도 등에 대한 형식을 정의한다. IEEE 754의 32비트 단정밀도 표현은 세 부분으로 구성되며 최상위 비트부터 순서대로 부호, 지수부, 가수부이다. Fig. 3.은 IEEE 754의 32비트 단정밀도 실수 표현을 나타낸 것이다.

10진법으로 표현된 실수 $(+4.84)$를 예로 들어 부호, 지수부, 가수부 표현방식에 대해 서술한다. 실수 $(+4.84)$는 양수이므로 부호 비트는 0이다. 10진법으로 표현된 실수를 이진법으로 나타내면 $100.1001101_{(2)}$가 된다. 부동소수점 단정밀도 표현에서는 고정소수점보다 적은 비트를 이용하여 큰 수를 표현하기 위해 정규화가 필요하다. 정규화를 이용하여 소수점을 왼쪽으로 이동시켜 $1.xx\ldots$와 같은 형태로 만들 수 있고, $100.1001101 = 1.001^* 2^2$과 같이 표현 가능하다. 이후 지수부에 편향을 더하는 것이 필요한데, IEEE 754 32비트 단정밀도 형식에서 편향은 127이다. 정규화 후 1.001^*2^2 이므로 지수 2에 127을 더하면 129와 같다. 129를 이진법으로 변환하면 지수부 8비트가 결정된다. 가수부는 전체 32비트 중 23비트를 차지하며, 정규화한 부동소수점 수에 부족한 비트 수만큼 0으로 채워 23비트를 만든다.

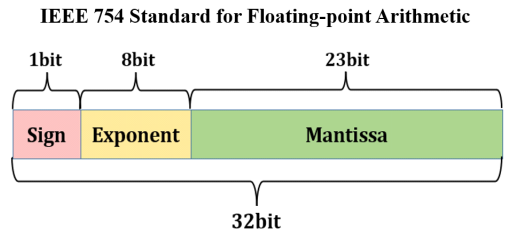


Fig. 3. IEEE 754 32bit floating point arithmetic

3.2.2 가중치 신규 복원 방법론

제안하는 내부 가중치 복원 방법은 총 10회의 CPA를 수행한다. 가수부 23비트 복원을 위해 8회의 CPA가 필요하고, 지수부 8비트와 부호 1비트 복원은 각각 1회의 CPA를 요구한다. Fig. 4.는 가중치가 $(+4.84)$일 때 복원하는 순서를 나타낸 것이다.

Fig. 4.와 같이 가중치의 복원 순서는 가수부, 지수부, 부호 비트 순이다. 가중치의 가수부와 지수부가 결정되면 부호를 제외한 하나의 실수값을 구성할 수 있다. 이를 바탕으로 부호 1비트만 추측하면 가중치 전체 복원이 가능하다.

가수부 23비트 복원의 경우 가수부 최상위 7비트

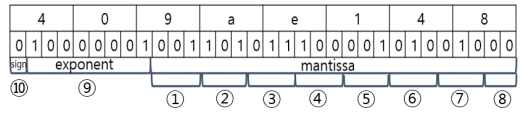


Fig. 4. Weight cpa sequence

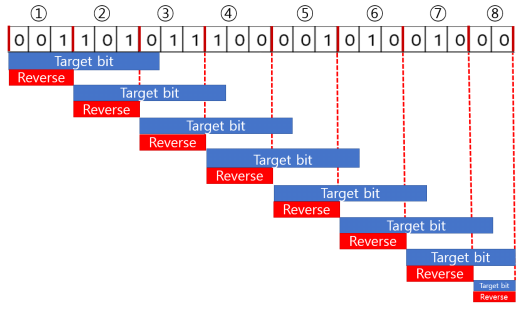


Fig. 5. Mantissa 23bit reversing sequence

를 공격 범위로 설정한 후 네트워크의 입력과 가중치를 곱한 결과 값의 해밍웨이트(hamming weight)와 파형 간의 상관계수를 계산한다. FIg. 5.와 같이 설정한 7비트 중 상위 3비트를 복원하고, 복원된 3비트 이후로 다시 7비트를 설정하고 7비트의 상위 3비트를 복원하는 과정을 반복한다. 가수부 23비트의 상위 3비트를 복원하는 방법을 알고리즘화 하면 Fig. 6.과 같다. 상위 3비트 이후의 가수부 비트들도 Fig. 6.을 이용하여 모두 복원할 수 있다. 지수부 8비트 복원의 경우 복원에 앞서 입력값과 가중치의 가수부를 곱한 결과로부터 지수부 보정값을 추출하는 것이 필수적이다. 보정값은 먼저 곱하는 두 실수의 지수들을 더한다. 입력값의 가수부와 가중치의 가수부를 곱하고 결과값을 정규화한 후 지수의 합의 오버플로우(overflow) 혹은 언더플로우(underflow)를 계산한다. 가중치 지수부 8비트와 보정값을 반영하여 얻어낸 중간 연산값과 전력 파형 간의 상관계수 계산을 통해 지수부 8비트를 복원할 수 있으며 이를 알고리즘으로 나타내면 Fig. 7.과 같다. Fig. 7.에서 ard는 입력값 가수부의 자릿수 $I_m d$와 가중치 가수부의 자릿수 $W_m d$의 단순 덧셈한 자릿수를 의미한다. mrd는 I_m과 W_m을 곱한 결과의 자릿수를 의미한다. mrd에서 ard의 차이를 계산하면 보정값 c를 얻을 수 있다.

Reverse engineering MSB 3bit of target weight mantissa 23bit

Input: Power consumption traces S, 32bit input value I of input layer node
Output: Weight mantissa MSB 3bit $W_m((w_9 w_{10} w_{11}))$

1. for $k=0$ to 127 do
2. for $j=1$ to N do
3. $I \leftarrow I[j]$
4. $I_m \leftarrow I_m$ + 0x800000 // get input mantissa
5. $W_m \leftarrow k+128$ // get weight mantissa
6. $O \leftarrow I_m \times W_m$ // get multiply of input mantissa by weight mantissa
7. $HW[j] \leftarrow HW((o_9 o_{10} \ldots o_{15}))$ // get 8-bit hamming weight
8. end
9. $cT[k] \leftarrow \rho(\{HW[1],\ldots,HW[N]\}, S)$
10. if($|cT[k]| > \max(|cT[k]|)$)
11. $key \leftarrow k$
12. $\max(|cT[k]|) \leftarrow |cT[k]|$
13. end
14. $W_m((w_9 w_{10} w_{11})) \leftarrow key$ MSB 3-bit
15. return $W_m((w_9 w_{10} w_{11}))$

Fig. 6. Algorithm of reverse engineering MSB 3bit of target weight mantissa 23bit

Reverse engineering exponent 8bit of target weight

Input: Power consumption traces S, 32bit input value I of input layer node, weight mantissa 23bit $W_m((w_9 w_{10} \ldots w_{31}))$
Output: Weight exponent 8bit W_e

1. for $k=0$ to 256 do
2. $W_e \leftarrow k$
3. for $j=1$ to N do
4. $I \leftarrow I[j]$
5. $I_m \leftarrow I_m$ + 0x800000 // get input mantissa
6. $ard \leftarrow (23-I_m d)+(23-W_m d)$
7. $mrd \leftarrow digit\ of\ (I_m \times W_m)$
8. $c \leftarrow mrd-ard-1$ // get correction value
9. $re \leftarrow (I_e-127)+(k-127)+c+127$
10. $HW[j] \leftarrow HW(re)$ // get result 8-bit exponent value hamming weight
11. end
12. $cT[k] \leftarrow \rho(\{HW[1],\ldots,HW[N]\}, S)$
13. if($|cT[k]| > \max(|cT[k]|)$
14. $key \leftarrow k$
15. $\max(|cT[k]|) \leftarrow |cT[k]|$
16. end
17. $W_e \leftarrow key$
18. return W_e

Fig. 7. Algorithm of reverse engineering exponent 8bit of target weight

Reverse engineering sign 1bit of target weight

Input: Power consumption traces S, 32bit input value I of input layer node, weight mantissa 23bit W_m, weight exponent 8bit W_e

Output: Weight sign 1bit W_s

1. // Suppose W is negative
2. for $j=1$ to N do
3. $I \leftarrow I[j]$
4. $W_s \leftarrow 1$
5. $HW[j] \leftarrow HW(I \times W)$ // get 32-bit hamming weight
6. end
7. $cT_n \leftarrow \rho(\{HW[1],...,HW[N]\},S)$
8. // Suppose W is positive
9. for $j=1$ to N do
10. $I \leftarrow I[j]$
11. $W_s \leftarrow 0$
12. $HW[j] \leftarrow HW(I \times W)$ // get 32-bit hamming weight
13. end
14. $cT_p \leftarrow \rho(\{HW[1],...,HW[N]\},S)$
15. if $cT_n > cT_p$
16. $W_s \leftarrow 1$
17. if $cT_p > cT_n$
18. $W_s \leftarrow 0$
19. return W_s

Fig. 8. Algorithm of reverse engineering sign 1bit of target weight

가중치의 가수부와 지수부를 모두 복원하였다면 부호를 제외한 하나의 실수값을 구성하는 것이 가능하다. 부호 비트 복원 시 방법은 2가지로 나눌 수 있다. 첫 번째로, 단일 가중치의 부호 1비트 복원의 경우 부호 비트가 0일 때 양수, 1일 때 음수를 의미한다. 가중치의 부호 비트가 0일 때 입력과 가중치를 곱한 값과 파형 간의 상관계수를 구한다. 같은 방법으로 부호 비트가 1일 때의 상관계수를 구하여 0일 때의 상관계수와 1일 때의 상관계수 값을 비교한다. 두 상관계수 중 ± 1에 더 가까운 값의 비트를 가중치의 부호 비트로 결정한다. 이를 알고리즘으로 표현하면 Fig. 8.과 같다.

두 번째로, 여러 가중치의 부호 비트를 동시에 복원하는 것이 가능하다. 과정을 간략히 설명하면 다음과 같다. 은닉층 노드의 결과값은 수식(1)과 같은 연산을 통해 결정된다. Fig. 6.과 Fig. 7.을 이용하여 가중치 가수부와 지수부 복원을 자동화하고 부호를 제외한 은닉층의 노드의 입력값에 사용되는 모든 가중치를 복원한다. 여러 가중치의 부호 비트들을 추측하는 키로 설정하여 수식(1)과 같이 은닉층 노드의 입력값을 계산한다. 해당 값과 파형 간의 상관계수를 이용하여 상관전력분석을 수행하면 여러 가중치의 부호 비트들을 동시에 복원할 수 있다.

$$h_k = \sum_{i=1}^{n} x_i W_{ik} \tag{1}$$

제안하는 신규 가중치 복원 방법론을 이용하여 실수 가중치 32비트 전체를 높은 정확도로 복원할 수 있다. 이를 자동화하면 다른 노드에 연결된 가중치도 순차적으로 복원하는 것이 가능하다.

3.3 내부 가중치 복원 실험결과

3.3.1 실험 환경

공격 대상은 MLP 모델이며, 입력층 노드 3개, 첫 번째 은닉층 노드 6개, 두 번째 은닉층 노드 5개, 출력층 노드 2개로 구성되어 있다. 실험 환경 구성은 Table 2.와 같이 설정하였다.

내부 가중치 복원에서는 공격모델의 첫 번째 은닉층의 첫 번째 노드의 가중치를 공격 대상으로 정하여 실험을 수행하였다. 실험 가정은 내부 중간값에 대한 정보가 없는 공격자를 설정하고, 내부 가중치 및 입력값의 범위에 제한을 두지 않았다. 입력층의 입력은 임의의 부호가 있는 32비트 실수값이고, 공격자가 원하는 값을 설정할 수 있다. 공격 위치는

Table 2. Experimental setup for weight recovery

Processor	STM32F303RCT7
Measurement	Chipwhisperer lite
Operating frequency	7.37 MHz
Trace	1000

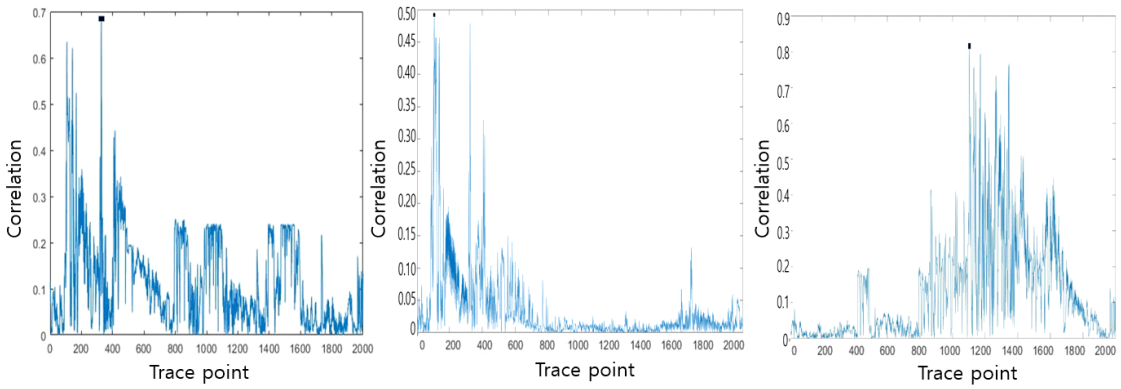


Fig. 9. Correlation corrtrace of target weight 32bit

첫 번째 은닉층의 첫 번째 노드 결과값이다. 각 입력층 노드의 입력값과 가중치가 곱해지고 이를 모두 더해 하나의 결과값을 생성하게 되는데, 이 덧셈 연산을 타겟으로 해밍웨이트를 설정하여 가중치 정보를 하나씩 획득하게 된다[15]. 분석 대상이 되는 수식은 (1)과 같다. 네트워크 내부의 활성함수로 ReLU를 사용하였다.

3.3.2 실험 결과

앞서 제안한 방법을 이용하여 가중치에 대한 범위 제한이 없는 환경에서 가중치의 모든 비트를 복원할 수 있다. 이는 범위 제한이 있었던 기존 가중치 복원 기술의 한계점을 극복한 것이다. 연산 복잡도를 줄이기 위해 가수부 23비트를 3비트씩 7회, 2비트 1회로 나눠 복원하고 지수부 8비트, 부호 비트를 순차적으로 복원하였다.

가수부 23비트 복원에서 7비트 추측 후 전력 파형과 가장 높은 상관계수를 갖는 상위 3비트를 복원하고, 이 과정을 반복하여 가수부 23비트 전체를 얻을 수 있다. Fig. 9(a).는 추측한 가수부 최상위 3비트와 전력 파형 간의 상관계수 파형을 나타낸 것이다. 예를 들어 가중치의 값이 (+4.84)일 때 가수부 23비트는 1.00110101110000101001000이고 복원한 가수부 23비트는 1.00110101110000000000001으로, 가수부의 비트 일치율은 99.99%이다. Fig. 9(b).는 공격 대상 가중치의 추측한 지수부와 전력 파형 간의 상관계수 파형이다. Fig. 9(c).은 공격 대상 가중치의 추측한 단일 부호 비트와 전력 파형 간의 상관계수 파형을 나타낸 것이다. 실험 결과 지수부와 부호 비트의 일치율은 각각 100%를 보였다. Fig. 10.은 실제 공격 대상 가중치 32비트와 복원한 가중치 32비트를 비교한 것이다. 실제 가중치가 (+4.84)일 때 복원한 가중치는 (+4.83984)이고, 전체 32비트 중 4비트 차이를 보였으며 하위 비트 차이는 최종 복원값에는 큰 영향을 미치지 않기 때문에 높은 정확도로 복원이 가능함을 알 수 있다.

실험을 통해 가수부 8개, 지수부 1개, 부호 비트 1개의 상관계수 파형을 얻을 수 있다. 수집한 파형과 가장 높은 상관계수를 가지는 비트를 옳은 비트라고 판단하여 결과적으로 하나의 실수를 완성할 수 있다. 이를 통해 딥러닝 네트워크 내부 비밀 가중치를 높은 정확도로 복원할 수 있음을 증명하였다.

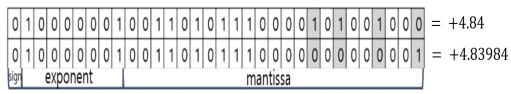


Fig. 10. Result of target weight 32bit

IV. 딥러닝 네트워크 신규 내부 활성함수 복원 기술

본 절에서는 딥러닝 네트워크의 활성함수를 복원하는 방법에 대해 소개한다. 새롭게 제안하는 딥러닝

기반 복원 방법은 기존 방법의 한계점을 극복하였으며 실험을 통해 99% 정확도를 증명하였다.

4.1 활성함수 신규 복원 방법론

기존 연구에서는 제한된 범위 내의 임의의 값을 활성함수의 입력으로 넣어 연산 시간 분포를 프로파일링하여 활성함수를 복원하였다. 이 방법은 공격 과정에서 활성함수의 연산 시간 분포를 그릴 수 있는 충분한 입력이 필요하며 입력값의 범위 또한 프로파일링 단계에서 설정한 (-2, 2)와 일치해야 한다는 단점이 있었다. 하지만 제안하는 신규 복원 방법론은 입력값에 대한 정보 없이 부채널 정보만으로 활성 함수 복원을 진행할 수 있으며 딥러닝 기반 프로파일링 공격 특성상 소수의 전력 파형으로도 복원할 수 있는 장점이 있다. 또한 딥러닝 모델 중 하나인 CNN 모델을 사용하게 되면 부채널 분석 분야의 중요한 부분 중 하나인 파형 정렬 문제를 해결할 수 있는 장점이 있다. 딥러닝 연산인 가중합과 활성함수 연산은 입력값에 따라 연산시간 차이가 발생하여 정렬이 맞지 않지만 CNN 모델을 사용하면 정렬을 맞추지 않고 활성함수를 복원할 수 있다.

제안하는 딥러닝 기반 활성함수 복원은 프로파일링 환경에서 수행한다. Fig. 11.은 활성함수 복원 전체 과정을 나타낸 것이다. 복원 대상이 되는 활성함수는 기존 논문과 동일하게 ReLU, Sigmoid, Tanh, Softmax로 설정하였다. 학습 단계에서 MLP 네트워크에 임의의 실수 입력을 넣어 소비전력을 수집한다. 수집한 전력 파형에서 복원 대상인 활성함수 연산 부분을 추출한다. 원-핫 인코딩(one-hot encoding)을 이용하여 라벨링(labeling) 작업을 수행한다. 라벨링 작업이 이루어진 학습용 전력 파형 데이터 셋으로 활성함수를 분류할 수 있도록 CNN 모델을 학습시킨다. 추론 단계에서 학습시킨 CNN 모델에 복원 대상 활성함수 소비 전력파형을 입력으로 넣는다. CNN 모델은 추측한 활성함수의 라벨을 결과로 반환한다.

4.2 활성함수 복원 실험 및 결과

4.2.1 실험 환경

프로파일링 부채널 분석을 수행하기 위해 딥러닝 네트워크가 동작할 때의 소비전력을 측정하였다. 공격 대상 MLP 네트워크는 입력층 노드 3개, 은닉층 노드 6개, 출력층 노드 2개로 구성하였다. 32비트 MCU인 STM32F303RCT7에서 7.37MHz로 동작할 때의 소비전력을 오실로스코프(HDO6104A)를 이용하여 측정하였다. 샘플링 레이트(sampling rate)는 학습 속도와 정확도 향상을 위해 10GS/s로 높게 설정하여 오버 샘플링 후 1:50 비율로 압축하였다. 수집한 전력 파형을 분류하는 CNN 모델을

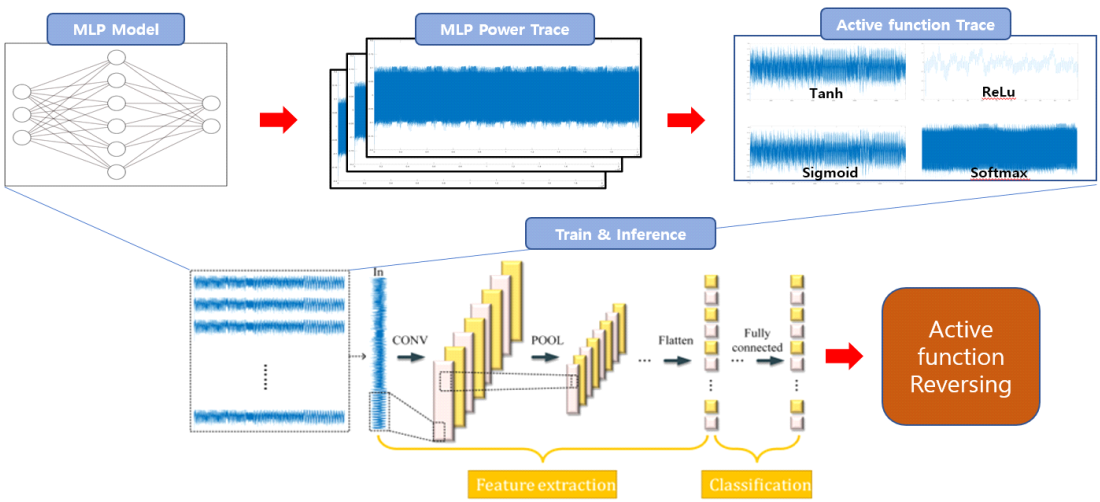


Fig. 11. Activation function reversing process

만들기 위해 TensorFlow와 Keras 라이브러리를 사용하였다.

4.2.2 실험 방법 및 결과

프로파일링 단계에서 MLP 네트워크에 임의의 실수 입력을 넣었을 때의 소비전력 파형을 4종류의 활성함수 각각 850개씩 총 3400개를 수집하였다. 수집된 전력 파형 중 2200개는 학습 데이터, 520개는 검증 데이터, 나머지 680개는 테스트 데이터로 사용하였다. 이를 정리하면 Table 3.과 같다.

수집된 데이터는 딥러닝 전체 네트워크의 소비전력이므로 활성함수를 분류하는 모델을 설계하기 위해 활성함수 연산을 여러 번 포함하도록 학습 데이터를 전처리하였다. 이렇게 구성된 학습 데이터는 파형 모양의 차이를 더 많이 포함하는 것과 같기 때문에 높은 성능의 분류 모델을 만들 수 있다. 원-핫 인코딩으로 라벨링하여 4개의 활성함수 파형을 구분하도록 하였다. Fig. 12.와 같이 파형의 일부분을 학습 데이터로 설정하여 학습시킨다.

4개의 활성함수 각각에 대해서 전체 전력 파형 중 220000의 길이를 갖도록 학습 파형을 추출하였다. Softmax 함수는 출력층에 사용되는 함수로 전력 파형의 뒷부분에서 추출하였고, ReLU 함수는 다른 활성함수에 비해 상대적으로 연산이 짧아 전체 파형 중 대부분을 학습 데이터로 사용하였다.

활성함수 파형을 분류하는 CNN 모델은 Table 4.와 같이 구성하였다. 컨볼루션 레이어와 풀링 레이어의 반복으로 전력 파형의 특징을 뽑아냈으며 파형의 크기 또한 줄일 수 있었다. 또한 마지막 출력층의 노드의 개수를 4로 설정하여 4개의 활성함수 중 하나로 추론할 수 있게 하였으며, 성능 향상을 위해 배치 정규화 층도 추가하였다.

Table 3. Experimental setup for activation function recovery

Processor	STM32F303RCT7	
Measurement	Oscilloscope(HDO6104A)	
Operating frequency	7.37 MHz	
Trace	Simgoid	850
	Tanh	850
	Softmax	850
	ReLU	850

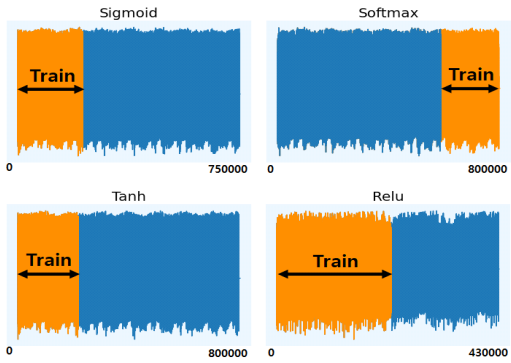


Fig. 12. Whole power trace and train data

학습 단계에서 손실 함수는 범주형 크로스 엔트로피(categorical cross-entropy), 최적화는 Adam으로 설정하였다. 학습률은 0.001로 에포크는 4로 학습하였다. Fig. 13.은 학습 과정에서 모델의 정확도와 손실값의 추이를 나타낸 것이다. 결과적으로 4번의 학습만으로도 검증 데이터에서 높은 정확도를 갖는 모델을 만드는 것이 가능하다.

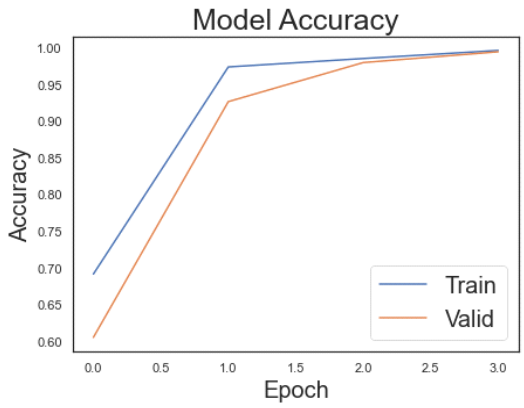
추론 단계에서 복원하려는 활성함수 전력 파형을 학습된 CNN 모델의 입력으로 넣어 추측하는 활성함수에 해당하는 라벨값을 출력으로 받는다. 학습에 이용되지 않은 680개의 테스트 데이터 셋을 이용하여 모델을 평가한 결과 99%의 정확도를 입증하였다.

Fig. 14.는 혼동행렬(confusion matrix)을 이용하여 활성함수의 분류 결과를 보여준다. 680개 중 5개의 활성함수 파형을 제외한 모든 입력을 정확하게 복원하는 것을 알 수 있다.

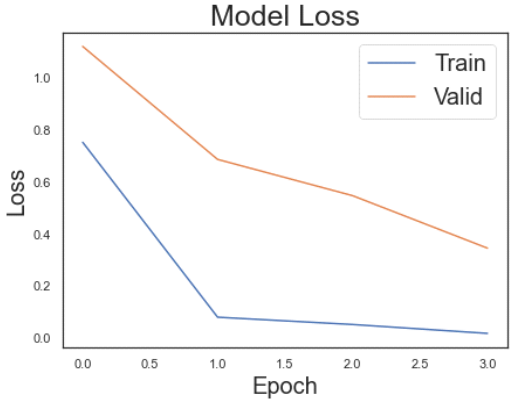
Table 4. CNN model specification

Layer Type	Output Shape	Parameter
Conv1D	(None, 219969, 8)	264
Batch Normalization	(None, 219969, 8)	32
MaxPooling	(None, 12220, 8)	0
Conv2D	(None, 12189, 8)	2056
MaxPooling	(None, 677, 8)	0
Flatten	(None, 5416)	0
Dense	(None, 50)	270850
Dense	(None, 4)	204

Total Parameters : 273,406

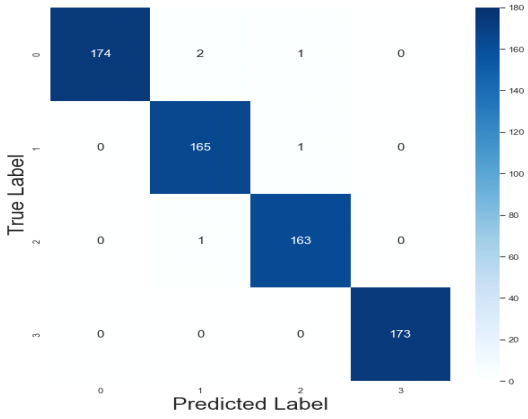


(a) Traning Accuracy



(b) Loss Value

Fig. 13. Activation function classification model training accuracy(a) and loss value(b)



Flg. 14. Activation function classification accuracy

V. 결 론

본 논문에서는 딥러닝 네트워크 모델의 내부 비밀정보인 가중치와 활성함수를 복원하는 효과적인 방법을 제시하였다. 기존 논문의 한계점을 극복한 환경에서 실험을 진행하여 새롭게 제안한 방법론을 검증하였다. 가중치 복원 실험에서는 가중치의 범위에 제한을 두지 않고 IEEE754 32비트 단정밀도 실수 가중치를 99.99% 정확도로 모두 복원하였다. 활성함수 복원 실험에서는 특정 입력값에 대한 제약 없이 프로파일링 환경에서 CNN 모델을 학습시켜 99% 정확도로 활성함수를 복원할 수 있음을 증명하였다.

추후 연구 방안으로 제안한 기술의 최적화 구현 및 16비트 실수 데이터를 대상으로 한 가중치 복원기술 구현, 하드웨어 및 사용자 인터페이스에서 동작 가능한 딥러닝 가속기 구현 등이 필요하다. 또한 취약점에 대한 대응기술 및 하드웨어 구현에 최적화된 딥러닝 설계기법 연구가 필수적이다.

References

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

[2] I.J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," International Conference on Learning Representations, Poster, Mar. 2015.

[3] S. Mangard, E. Oswald, and T. Popp, "Simple Power Analysis," Power Analysis Attacks: Revealing the Secrets of Smart Cards, pp.101-118, Jan. 2008.

[4] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," Advances in Cryptology, CRYPTO' 99, LNCS 1666, pp.388-397. 999.

[5] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," Cryptographic Hardware and Embedded Systems, LNCS 3156, pp.16-29, 2004.

[6] H. Maghrebi and T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," International Conference on Security, Privacy, and Applied Cryptography Engineering, LNCS 10076, pp. 3-26, 2016.

[7] E. Cagli, C. Dumas, and E. Prouff, "Convolutional neural networks with data augmentation against jitter-based countermeasures," International Conference on Cryptographic Hardware and Embedded Systems, LNCS 10529, pp. 45-68, 2017.

[8] 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

[9] H. Yu, H. Ma, K. Yang, Y. Zhao, and Y. Jin, "DeepEM: Deep Neural Networks Model Recovery through EM Side-Channel Information Leakage," In Proceedings of the 2020 IEEE International Symposium on Hardware Oriented Security and Trust, pp. 209 - 218, Dec. 2020.

[10] K. Yoshida, T. Kubota, S.Okura, M. Shiozaki, and T. Fujino, "Model Reverse-Engineering Attack using Correlation Power Analysis against Systolic Array Based Neural Network Accelerator," In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems, pp. 1 - 5, Oct. 2020.

[11] K. Yoshida, M. Shiozaki, S. Okura, T. Kubota, and T. Fujino, "Model Reverse-Engineering Attack against Systolic-Array-Based DNN Accelerator Using Correlation Power Analysis" IEICE Trans. Fundam. Electron. Commun. Computer Science, pp. 152-161, Jan. 2021.

[12] S. Maji, U. Banerjee, and A.P. Chandrakasan, "Leaky Nets: Recovering Embedded Neural Network Models and Inputs through Simple Power and Timing Side-Channels - Attacks and Defenses," IEEE Internet Things Journal, pp. 12079-12092, Feb. 2021.

[13] G. Takatoi, T. Sugawara, K. Sakiyama, Y. Li, "Simple Electromagnetic Analysis Against Activation Functions of Deep Neural Networks," Applied Cryptography and Network Security Workshops, LNCS 12418, pp. 181-197, Oct. 2020

[14] X. Hu, L. Liang, S. Li, L. Deng, P. Zuo, Y. Ji, X. Xie, Y. Ding, C. Liu, T. Sherwood, and Y. Xie, "DeepSniffer: A DNN Model Extraction Framework Based on Learning Architectural Hints," Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 385-399, Mar. 2020.

[15] Francois-Xavier Standaert, "Introduction to side-channel attacks," Secure integrated circuits and systems, pp. 27-42, Dec. 2010.

[16] M. MéndezReal, and R. Salvador, "Physical Side - Channel Attacks on Embedded Neural Networks: A Survey," Applied Sciences, pp. 6790, July. 2021.

[17] Sung-hyun Jin, Su-hri Kim, Hee-seok Kim, and Seok-hie Hong, "Recent advances in deep learning based side channel analysis," ETRI Journal, 42(2), pp. 292-304, April. 2020.

[18] F. Zhang, X. Lou, X. Zhao, W. He, R. Ding, S. Qureshi, and K. Ren, "Persistent fault analysis on block ciphers," IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 150-172, Aug. 2018.

[19] W. Hua, Z. Zhang, and G.E. Suh, "Reverse engineering convolutional neural networks through side-channel information leaks," In Proceedings of the 55th Annual Design Automation Conference, pp. 1-6, June. 2018

〈저자소개〉

박 수 진 (Sujin Park) 학생회원
2019년 3월~현재: 고려대학교 과학기술대학 인공지능사이버보안학과 학사과정
〈관심분야〉 부채널 공격

이 주 헌 (Juheon Lee) 학생회원
2020년 3월~현재: 고려대학교 과학기술대학 인공지능사이버보안학과 학사과정
〈관심분야〉 부채널 공격

김 희 석 (HeeSeok Kim) 종신회원
2006년: 연세대학교수학과 학사
2008년: 고려대학교정보보호대학원 석사
2011년: 고려대학교정보보호대학원 박사
2011년 9월~2012년 12월: Bristol University 박사후연구원
2013년~2016년 8월: 한국과학기술정보연구원(KISTI) 선임연구원
2015년~2016년 8월: 과학기술연합대학원대학교(UST) 조교수
2016년 9월~현재: 고려대학교 과학기술대학 인공지능사이버보안학과 부교수
〈관심분야〉 부채널 공격, 암호시스템 안전성 분석 및 고속구현, 암호칩 설계 기술, 보안관제, 네트워크 보안