HD-NIDSBench: 고차원 컴퓨팅을 활용한 효율적인 침입탐지 시스템에 대한 벤치마킹 프레임워크

강윤의 김지승** 김예성*** 김현주* 이현우*

KENTECH (학부생), *KENTECH(교수), **DGIST (대학원생), ***DGIST(교수)

HD-NIDSBench: A Benchmarking Framework for Efficient Intrusion Detection Systems Using High-Dimensional Computing

Yunui Kang, Jiseung Kim*, Yeseong Kim*, Hyunju Kim, Hyunwoo Lee

KENTECH, *DGIST

요 약

사물인터넷(IoT) 환경 확산에 따라 자원이 제약적인 IoT 게이트웨이/네트워크 엣지에 적합한 효율적인 네트워크 침입 탐지 시스템(Network Intrusion Detection System, NIDS)의 필요성이 증대되고 있다. 이에 따라, 연산 비용이 높은 기존 딥러닝을 활용한 NIDS의 대안으로 효율적 알고리즘인 고차원 컴퓨팅(Hyperdimensional Computing, HDC)을 활용한 NIDS를 고려할 수 있다. 우리는 HDC를 NIDS에 적용했을 때의 성능을 체계적으로 검증하기 위해, HDC를 구성하는 다양한 요소에 따라 NIDS의 성능을 평가하는 HD-NIDSBench 프레임워크를 제안한다. 우리는 본 프레임워크를 활용하여 3가지 인코딩 방식, 2가지 학습 방법과 5가지 유사도 척도를 변경하며 3개 표준 데이터셋에 대한 성능 평가를 수행하였다. 우리는 재학습 과정이 모델의 정확도를 평균 약 17% 향상시키는 핵심적인 역할을 하며, 유사도 척도의 영향은 미미함을 확인하였다. 또한, HDC는 유사한 침입 탐지 성능의 딥러닝 모델 대비 약 80배 높은 메모리 효율성을 보여, 자원 제약적 환경에서의 NIDS를 위해 효과적임을 입증하였다.

I. 서론

사물인터넷(IoT) 기술의 보편화는 가정과 공공 인프라 등 다양한 영역에 걸쳐 연결성을 확장시키며 공격 표면을 급격히 넓히고 있다 [1]. 수많은 스마트 기기가 상호 연결된 환경은 단일 기기의 취약점이 전체 네트워크의 안정을 위협하는 구조적 위험을 가진다 [2]. 이에 따라 네트워크 엣지 단에서 트래픽을 실시간으로 감시하고 악의적인 활동을 탐지하는 네트워크 침입 탐지 시스템(NIDS)의 중요성이 부각되고 있다 [3]. 그러나 IoT 게이트웨이와 같은 엣지 디바이스는 일반적으로 메모리 및 연산 능력이 제한적인 자원 제약적 환경에서 동작하므로 [4], 이러한 환경에 적용하기 위한 효율적인 NIDS의 개발이 시급한 과제로 떠오르고 있다.

최근 인공지능 기술의 발전과 함께 딥러닝(Deep Learning)에 기반한 NIDS가 다양하게 제안되었다. 이러한 딥러닝 모델은 높은 탐지 정확도를 달성하지만, 수많은 파라미터를 반복적으로 갱신하는 역전파(Backpropagation) 알고리즘으로 인해 학습 과정에서 막대한 연산 자원을 필요로 한다 [5]. 이는 저사양의 엣지 디바이스에 딥러닝 기반 NIDS를 활용하는 것은 현실적으로 쉽지 않다. 본 연구에서는 이에 대한 대안으로, 메모리 사용량이 적고 연산 속도가 빠른 경량화된 학습 알고리즘으로 주목받는 HDC를 NIDS에 적용하고자 한다 [6]. HDC는 데이터를 수천 차원의 고차원 벡터로 표현하고, 복잡한 역전파 과정 없이 간단하고 병렬화가 용이한 연산을 통해 학습을 수행한다. 또한 정보가 벡터 공간 전체에 분산되어 저장되므로 하드웨어 오류나 노이즈에 강건한 특성을 보여, 자원이 제한된 IoT 플랫폼을 위한 유망한 솔루션으로 평가받는다 [7].

HDC를 활용한 효율적인 NIDS를 개발하기 위해서는 HDC 모델의 성능을 다각적으로 평가하고 최적의 구성 요소를 탐색할 수 있어야 한다. 이를 위해 HD-NIDSBench라는 성능 분석 프레임워크를 제안한다. HDC의 침입 탐지 성능은 인코딩 방식, 유사도 측정 방법, 학습 방식 등 핵심 파라미터의 조합에 따라 크게 달라질 수 있다. HD-NIDSBench는 이러한 핵심 파라미터를 변경해가며 HDC 기반의 NIDS를 체계적으로 분석한다. 우리는 널리 알려진 NSL-KDD [8], UNSW-NB15 [9], Mirai [10] 벤치마킹 데이터셋에 대한 HDC 기반의 NIDS의 성능 분석 실험을 수행하였고, 기존 머신러닝 및 딥러닝 모델과의 비교를 통해 HDC 기반 NIDS의 효율성과 실용성을 입증하였다.

II. 배경 이론

HDC는 정보를 처리하기 위해 고차원 벡터를

활용하는 계산 프레임워크이다. HDC는 모든 정보를 1,000에서 10,000 사이의 매우 큰 차원(D)을 갖는 고차원 벡터로 표현하며, 이들 벡터들은 고차원에서 무작위로 생성되어 상호 독립적인 특성을 갖는다. 이러한 특성은 정보가 벡터 공간 전체에 분산되게 함으로써 일부 데이터에 오류나 노이즈가 발생하더라도 전체 정보 손실이 적은 높은 강건성을 보장한다. 이는 안정적인 탐지가 요구되는 NIDS에 매우 적합하다.

HDC 기반 분류 모델은 인코딩(Encoding), 학습(Training), 추론(Inference)의 세 단계로 구성된다. 인코딩 단계를 통해 학습 데이터셋의 모든 샘플은 고차원 하이퍼벡터로 변환한다. 학습 단계에서는 공격 샘플과 정상 샘플 별로 관련 고차원 벡터들을 통합하여 공격과 정상 각각에 대한 '클래스 하이퍼 벡터'를 생성한다. 마지막으로 추론 단계에서는 새로운 샘플이 입력되면 동일한 방식으로 인코딩한 후, 생성된 고차원 벡터에 대해 공격과 정상 클래스 하이퍼벡터 각각과 유사도를 계산한 후 가까운 클래스로 최종 분류한다.

주요 **인코딩 방식**으로는 데이터의 각 피처와 값을 분리하여 표현하는 ID-level, 데이터를 무작위 행렬 곱으로 고차원에 투영하는 Random Projection, 그리고 여기에 비선형 변환을 추가한 Nonlinear 방식이 있다. **학습 방식**으로는 모든 데이터를 한 번에 묶어 클래스 하이퍼벡터를 생성하는 단일 패스 방식과, 예측이 틀린 샘플을 기반으로 모델을 점진적으로 최적화하는 재학습 방식이 있다. 추론 단계에서 사용하는 **유사도 척도**로는 내적(Dot) 기반 유사도, 코사인(Cosine) 기반 유사도, Geodesic 거리, Fisher-Rao 거리, Tanimoto 계수 등이 있다.

III. HD-NIDSBench 설계

HD-NIDSBench는 HDC 기반 네트워크 침입 탐지 모델의 성능을 체계적으로 평가하고 분석하기 위해 설계된 벤치마킹 프레임워크이다. 이는 데이터셋을 입력받아 HDC 모델의 학습 및 추론 과정을 수행하며, 최종적으로 분류 성능을 출력한다. 이를 위해 우리는 데이터를 고차원으로 변환하는 인코더(Encoder) 모듈, 클래스 하이퍼벡터를 생성하는 학습(Training) 모듈, 그리고 데이터의 클래스를 판별하는 추론(Inference) 모듈로 구성된다. 이를 통해 모델 성능에 영향을 미치는 핵심 구성 요소를 변수화하여 다양한 조건에서의 평가를 가능하게 한다.

IV. HD-NIDSBench 평가

우리는 HD-NIDSBench를 활용하여, 다양한 구성 요소와 환경 변화에 따른 HDC 기반 NIDS의 성능을 분석하였다.

4.1 실험 세팅

데이터셋 본 연구에서는 NIDS 분야에서 널리 사용되는 NSL-KDD, UNSW-NB15, Mirai 데이터셋을 사용하여 HDC 모델의 성능을 다각적으로 평가한다.

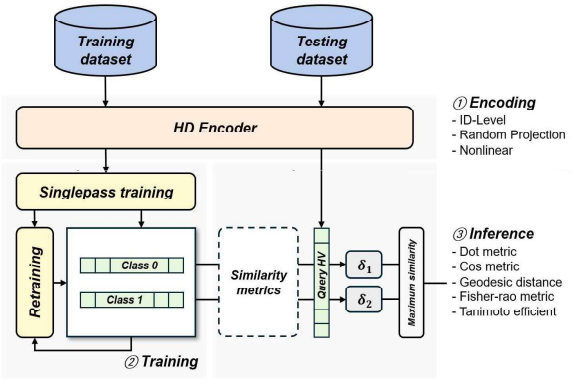

그림 1 HD-NIDSBench Workflow

실험 환경 Python 라이브러리 hdzoo[11]를 기반으로 HD-NIDSBench를 구현하였으며, 다양한 모델과의 성능 비교 및 분석을 위해 AI Analyzer[12]를 활용하였다. 우리는 HDC 기반 NIDS를 높은 성능이 입증된 LightGBM, Random Forest와 같은 앙상블 모델, 딥러닝 모델인 Feed-forward, 그리고 전통적인 확률 기반 알고리즘인 Naïve Bayes와 비교하였다 [13].

모든 실험은 AMD EPYC 7742 64코어 프로세서 기반 서버에서 6개의 CPU 코어와 50GB의 메모리, 그리고 GPU 자원을 할당받아 수행되었다. 모델의 성능은 F1-Score를 핵심 평가 지표로 사용하여 측정하였다.

4.2 차원 크기에 따른 성능 평가

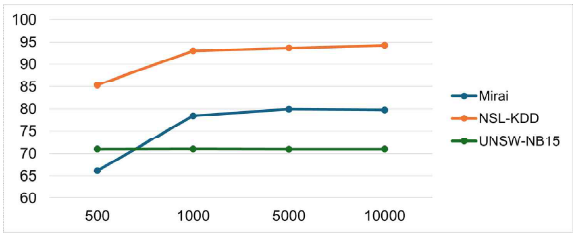

그림 2. 차원에 따른 F1-Score

Nonlinear 인코딩, 코사인 유사도, 그리고 재학습 방법을 모두 동일하게 고정한 상태에서, 고차원 벡터의 차원을 500, 1000, 5000, 10000으로 증가시키며 성능 변화를 측정했다.

실험 결과, 차원 수를 높일수록 모델의 F1 Score가 점진적으로 향상되는 뚜렷한 경향성을 확인할 수 있었다. 이는 고차원 벡터의 차원이 증가함에 따라 데이터의 복잡한 특징을 더 잘 표현할 수 있게 되어, 결과적으로 모델의 분류 정확도가 개선된 것으로 분석된다.

4.3 데이터별 성능 평가

우리는 데이터셋 별로 인코딩 방식, 학습 방식, 유사도 척도를 바꿔가며 HDC 기반 NIDS의 성능을 측정하였다. 차원 수의 영향을 배제하고 모델의 성능을 일관성 있게 비교하기 위해, 차원은 10,000으로 고정하였다.

		Mirai	NSL-KDD	UNSW-NB15
	Retrain	F1score	F1score	F1score
ID-level	False	61.92	73.82	69.69
	True	79.76	90.74	71.02
Random Projection	False	61.92	79.89	73.19
	True	81.58	93.73	79.57
Nonlinear	False	69.74	76.09	**79.92**
	True	**89.97**	**93.82**	71.0
LightGBM		92.53	82.57	88.06
Random Forest		83.12	77.98	88.78
Feed Forward		98.11	85.82	87.15
Naïve Bayes		72.6	81.5	78.76

표 2. 인코더별 성능 비교

Mirai와 NSL-KDD에서는 Nonlinear 인코딩과 재학습을 함께 적용했을 때 가장 높은 성능을 보였고, UNSW-NB15의 경우 Nonlinear 인코딩을 사용하되 재학습을 적용하지 않은 단일 패스 방식에서 가장 우수한 성능을 보였다. 이는 곧, 환경에 따라 HDC를 위한 최적의 조합은 달라질 수 있다는 것을 보여 준다.

전체 실험 결과를 종합했을 때, 재학습을 적용한 모델은 단일 패스 방식만을 사용한 모델에 비해 F1 Score가 평균적으로 약 16.95% 향상되어, 재학습 과정이 모델의 분류 정확도를 개선하는 데 매우 효과적임을 확인하였다.

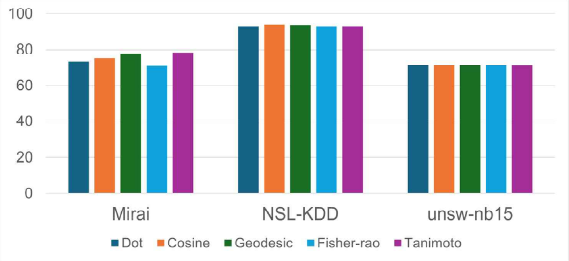

그림 3. 유사도 매트릭스별 성능 비교

유사도 척도가 HDC 성능에 미치는 영향을 분석하기 위해, 우리는 우수한 성능을 보인 Nonlinear 인코딩 및 재학습 방식에 대해 5가지 척도를 비교하였다. 실험 결과, 유사도 척도에 따른 F1-Score 성능 차이는 거의 발생하지 않았다. 이는 모델이 반복적인 재학습을 통해 오분류 샘플을 지속적으로 교정하며 최적의 분류 경계를 찾아가므로, 특정 유사도 계산 방식에 대한 의존도가 낮아지기 때문으로 분석된다. 따라서 재학습을 적용할 경우, 유사도 척도의 선택은 모델의 최종 성능에 결정적인 영향을 미치지 않는다.

4.4 메모리 효율성 평가

HDC	2.44KB
Feed Forward	193KB

표 4. 알고리즘 별 메모리 소모량

본 연구에서는 모델의 실용성을 평가하기 위해, 유사한 F1-Score 성능을 보이는 조건에서 제안하는 HDC 모델과 Feed Forward 신경망의 메모리 소모량을 비교 분석하였다.

계산 결과, HDC 모델은 약 2.44KB의 메모리를 필요로 하는 반면, Feed Forward 모델은 약 193KB를 소모하여 HDC의 메모리 효율성이 약 80배 가까이 뛰어난 것을 확인하였다.

이러한 결과는 HDC가 딥러닝 모델과 대등한 분류 성능을 유지하면서도, 연산에 필요한 자원을 획기적으로 절감할 수 있음을 의미한다. 따라서, HDC는 IoT 엣지 디바이스나 경량화된 시스템과 같이 자원이 제약적인 환경에서 네트워크 침입 탐지를 수행하기 위한 매우 효과적이고 실용적인 대안이 될 수 있음을 시사한다.

V. 결론

본 연구는 자원 제약적 환경을 위한 HDC 기반 침입 탐지 평가 프레임워크 HD-NIDSBench를 제안하였다. 실험 결과, 재학습 과정이 모델 성능을 평균 17% 향상시키고, 유사 성능의 신경망 모델 대비 약 80배 높은 메모리 효율성을 달성함을 입증하였다. 이는 HDC가 IoT 엣지 디바이스와 같은 실제 환경에서 매우 실용적인 대안임을 시사한다. 향후 연구로는 최적화 기법을 재학습에 통합하고, 프레임워크를 이상 탐지 문제로 확장하고자 한다.

Acknowledgements

이 연구는 2025년도 산업통상자원부 및 한국산업기술기획평가원(KEIT) 연구비 지원에 의한 연구임(과제번호: RS-2025-02653102)

[참고문헌]

[1] A. Al-Fuqaha et. al., "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Commun. Surv. Tutor., vol. 17, no. 4, 2015.
[2] N. Neshenko et. al., "Demystifying IoT security: An exhaustive survey of security vulnerabilities and defense mechanisms," IEEE Commun. Surv. Tutor., 2019.
[3] B. B. Zarpel et. al., "A survey of intrusion detection in Internet of Things," J. Netw. Comput. Appl., 2017.
[4] W. Shi et. al., "Edge computing: Vision and challenges," IEEE Internet Things J., 2016.
[5] Y. LeCun et. al., "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, May 2015.
[6] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," 2009.
[7] J. Wang et. al.,"HyperDetect: A real-time hyperdimensional solution for intrusion detection in IoT networks," IEEE Internet Things J., Apr. 2024.
[8] M. Tavallaee et. al., "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symp. Comput. Intell. Security Defense Appl., 2009.
[9] N. Moustafa et. al., "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in Proc. Mil. Commun. Inf. Syst. Conf. 2015.
[10] Lee, H et. al., An infection-identifying and self-evolving system for IoT early defense from multi-step attacks., ESORICS
[11] https://github.com/CELL-DGIST/HDZoo-official.git.
[12] https://github.com/comsyssec/ai-ids-analyzer.git
[13] Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on tabular data?. In Advances in Neural Information Processing Systems