패킷 표현 학습에서 플로우 별 분할 기반 평가: 암호화 트래픽 분류에서 데이터 스누핑과 페이로드 의존성*

김강욱[0], 노희준*
인하대학교 컴퓨터공학과
ices17@inha.edu, hjroh@inha.ac.kr

Per-Flow Split Evaluation for Packet Representation Learning: Data Snooping and Payload Dependency in Encrypted Traffic Classification

Kangwook Kim[0], Heejun Roh*
Department of Computer Engineering, Inha University

요 약

현대의 인터넷에서 대부분의 패킷의 페이로드가 암호화되어 교환되면서, 기존 심층 패킷 검사 기반의 트래픽 분류 방식의 한계를 극복하고자 딥러닝을 활용한 트래픽 분류 연구가 활발히 진행되고 있다. 특히 트랜스포머 구조의 등장 이후, 암호화된 트래픽 분류 문제에 이를 적용한 분류 프레임워크가 제안되고 있다. 이중 사전학습된 인코더를 활용한 모델들은 여러 태스크에서 높은 분류 성능을 보인다고 알려져 있었으나, 최근 연구에서 ET-BERT의 성능 평가 과정에 문제가 있음이 보고되었다. 본고에서는 사전학습된 인코더를 활용하면서도 ET-BERT와는 다른 방식으로 동작하는 PacRep 또한 성능 평가 과정에서 데이터 스누핑 오류를 범했음을 간접적으로 확인하는 실험을 설계, 과적합 가능성이 있음을 확인한다. 또한 암호화된 페이로드를 입력으로 활용하는 PacRep의 분류 성능이 페이로드에 얼마나 의존하는지 실험을 통해 살펴보고, 과적합 가능성에 대해 논의한다.

1. 서 론

네트워크 트래픽 분류 기술은 트래픽 데이터로부터 유발 응용 정보나 침입 탐지 및 차단, 서비스 품질(QoS) 제어를 위한 정보를 추출하는 등의 다양한 목적을 달성하는데 활용할 수 있는 기술이다. 현대의 인터넷에서는 엔드포인트에서 (또는 몇몇 상황에서는 미들박스에서) 대부분 패킷을 암호화하여 교환하기 때문에 전통적인 네트워크 트래픽 분류 기술인 심층 패킷 검사(deep packet inspection, DPI) 방식으로 네트워크 트래픽을 분류하는 것은 한계가 있다. 이러한 한계와 함께 컴퓨터 비전, 자연어 처리 분야에서 딥러닝 기술의 성공적인 도입에 힘입어, 네트워크 트래픽 분석 분야에서도 딥러닝을 활용한 네트워크 트래픽 분류 기술 연구가 진행되기 시작하였다 [1].

특히, 시퀀스 입력을 효율적으로 처리하는 것으로 알려진 트랜스포머(transformer) [2] 구조는 (종래에 Recurrent Neural Network(RNN)를 사용하여 학습하던) 트래픽 플로우의 순차적 특징[3]이나 (Convolutional Neural Network(CNN)을 사용하여 학습하던) 패킷의 바이트 시퀀스 [4]를 효과적으로 학습할 수 있을 것으로 기대된다. 이에 따라 암호화된 트래픽 분류 문제에 트랜스포머 기반 모델을 적용하려는 시도가 늘어나고 있다 [5]. 이중, PacRep [6]은 패킷 헤더 등에서 디코딩한 텍스트 데이터와 암호화된 바이트 시퀀스를 입력으로 받은 뒤, BERT (Bidirectional Encoder Representations from Transformers)와 같은 사전학습된 트랜스포머 인코더를 활용해 대조 표현 학습(contrastive representation learning) [7] 프레임워크를 제안하여 높은 분류 성능을 가짐을 보인 바 있다.

하지만, PacRep은 성능 평가 과정에서 훈련 및 테스트셋을 패킷 단위로 단순 분할한 것으로 추정되는데, 이러한 분할 방식은 성능을 과대평가할 가능성이 존재한다. 예를 들어, 패킷으로부터 해당 패킷을 생성한 애플리케이션을 알아내는 문제를 고려해 보자. 이때, 대부분의 애플리케이션은 클라이언트-서버 모델에 따라 두 엔드포인트 간에 많은 수의 패킷을 교환하게 된다. 컴퓨터 네트워크 분야에서는 이를 추상화하여 (네트워크) 플로우라고 칭한다. 이때, 동일한 플로우에 속한 패킷들은 유사한 값 또는 패킷 간 상관관계를 가진 헤더 필드(header field)를 가질 수 있으며, TCP의 시퀀스 번호(sequence number)가 대표적이다.

문제는, 동일한 플로우에 속한 패킷이 이러한 상관관계를 가지고 있음에도 불구하고 이에 대한 고려 없이 단순 분할하게 되면, 훈련, 검증, 테스트 데이터셋 모두에 같은 플로우에 속한 패킷이 존재할 수 있게 된다는 것이다. 이러한 패킷 단위 분할 방식은 테스트셋의 정보가 훈련 과정에서 노출되는 데이터 스누핑(data snooping) [8] 오류가 발생할 수 있으며, 이는 모델의 실제 성능 평가를 왜곡할 가능성이 있다. 특히 PacRep은 디코딩한 텍스트 데이터를 학습하기 때문에, 같은 플로우에 속한 패킷의 TCP 시퀀스 번호의 일부를 보고 특정 애플리케이션이라고 판단하는 것과 같은 오류를 범할 가능성이 있다.

이러한 문제에 대해서 [5]에서는 바이트 시퀀스 기반 트래픽 분류 모델로 알려진 ET-BERT [4]의 성능 평가에 문제가 있음을 보인 바 있다. [9]에서는 ET-BERT를 포함한 여러 바이트 시퀀스 트랜스포머 기반 분류기의 성능을 재평가하였으나, 바이트 시퀀스와 디코딩한 텍스트 데이터를 모두 사용하는 PacRep에 대해서는 여러 문제점을 지적하면서도 실제 성능 평가를 수행하지 않은 한계가 있다.

따라서 본 논문에서는 기존 PacRep 실험에서 발생할 수 있는 데이터 스누핑 오류 여부를 간접적으로 검증하고자, 데이터셋을 구성할 때 플로우로 인한 데이터 스누핑 오류를 피하도록 실험을 설계, 수행한다. 또한, PacRep은 기존 패킷 표현 학습 기반 분류기에서 주로 활용하는 바이트 시퀀스도 텍스트화 하여 같이 특징으로 활용하기에, 이로 인해 발생하는 과적합 가능성에 대해서도 같이 검토하기 위한 실험을 수행한다.

* 이 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. RS-2023-00235509) (* 교신저자)

2. 관련 연구

본 절에서는 논문에서 진행한 실험에 대한 이해를 돕고자 먼저 딥러닝 기반 암호화된 트래픽 분류와 관련된 기존 연구를 간략히 조망하고, 서론에서 언급한 두 연구인 ET-BERT와 PacRep의 동작 방식에 대해 간단히 설명한다.

최근 몇 년간 발표된 딥러닝 기반 암호화된 트래픽 분류 연구는, '트래픽에 대한 전처리를 일절 또는 복잡하게 하지 않더라도 트래픽의 표현을 자동적으로 학습할 수 있는가'라는 질문에 다양한 답을 도출해왔다. 특히 몇몇 연구는 전문가가 프로토콜에 대한 지식을 바탕으로 세심하게 특징을 설계, 선택하는 기존의 통계적 특징 기반 접근법을 대체할 수 있다고 생각될 정도로 높은 정확도를 보여주었다. 초기에는 지도학습 기반으로 CNN, RNN, LSTM 등을 활용해 특정 분류 태스크를 수행하는 것에 초점을 맞추는 경우가 많았으나, 이내 여러 공개 데이터셋에서 높은 성능을 달성하면서 알려지지 않은 트래픽에 대한 분류, 전이 학습(transfer learning), 다중 태스크 학습(multi-task learning)과 같이 복잡하고 어려운 미해결 문제(open problem)을 풀고자 하는 연구가 관심을 받기 시작하였다 [10].

서론에서 언급한 ET-BERT와 본 논문에서 다루는 PacRep은 모두 트래픽의 표현을 사전학습(pre-training)한 뒤 (넓은 의미에서) 다중 태스크 학습 문제를 풀고 있으나, 상당히 다른 방식의 해법을 제안한다.

ET-BERT는 레이블이 없는 대규모 암호화 트래픽에서 트랜스포머 구조로 자기 지도학습을 하여 패킷에 대한 일반적인 표현을 찾는다. 세부적으로는 네트워크 플로우를 대표하는 연속된 패킷 집합인 버스트(BURST)를 bi-gram으로 토큰화한 뒤, 일부를 마스킹해 재구성하는 문제와 서로 다른 두 버스트의 조각이 같은 출처에서 비롯되었는지 여부를 맞추는 문제를 학습한다. 이후 미세 조정(fine tuning) 단계에서는 주어진 태스크에 대해 사전학습과 동일한 구조로 처리한 뒤, 예측을 위한 다중 클래스 분류기에 넣어 최종 분류 결과를 도출하는 전이 학습(transfer learning) 방식을 따른다. 이중 마스킹 기반 사전학습과 전이 학습은 여러 후속 연구에서 공통적으로 관찰된다.

반면, PacRep은 1절에서 언급했듯이 패킷에서 디코딩한 텍스트 데이터와 암호화된 바이트 시퀀스를 입력으로 받은 뒤 이를 BERT와 같은 트랜스포머 기반 인코더로 초기 잠재 표현(latent representation)을 생성한다. 이후 태스크 별로 랜덤하게 샘플을 선택하고, 대조 표현 학습에 기반하여 라벨 없이 샘플 간의 유사 관계를 학습하여 구별력이 있는(discriminative) 패킷 표현을 학습한다. 이때 PacRep은 대조 표현 학습을 위한 손실과 함께 각 태스크의 레이블에 대해 파라미터 조정을 하기 위한 손실을 가중 합산하여 한 번에 학습하는 다중 태스크 학습 방식을 사용한다. 이러한 접근 방법은 동일한 입력으로 여러 태스크에 대한 출력을 한 번에 효율적으로 얻을 수 있다는 점에서 전이 학습 기반의 연구와는 다른 장점을 가지고 있다.

3. 실험 설계

실험에 사용한 데이터셋으로는 PacRep 논문 [6]에서의 성능 평가 결과와 본 실험의 결과를 비교할 수 있도록 해당 논문에서 사용한 세 가지 공개 암호화 트래픽 데이터셋인 ISCXVPN2016, DoHBrw2020, USTCTFC2016의 PCAP 파일을 사용하였다.

- ISCXVPN2016: 첫 번째 태스크(task 1)는 VPN과 NonVPN을 구분하는 이진 클래스로, 두 번째 태스크(task 2)는 Facebook, YouTube 등 총 16개의 다중 클래스로 구성된다.
- DoHBrw2020: 첫 번째 태스크(task 3)는 정상과 공격을 분류하는 이진 클래스로, 두 번째 태스크(task 4)는 공격 트래픽을 발생시킨 도구에 따라 DNScat2, Iodine 등 총 5개의 다중 클래스로 구성된다.
- USTCTFC2016: 첫 번째 태스크(task 5)는 정상과 공격을 분류하는 이진 클래스로, 두 번째 태스크(task 6)는 Gmail, Skype, MySQL 등 총 20개의 다중 클래스로 구성된다.

PCAP 파일의 전처리를 위해 SplitCap [11]을 사용하였으며, 송수신 IP 주소, 송수신 포트 번호, 프로토콜 번호의 5-tuple로 식별되는 단방향 플로우(unidirectional flow) 단위로 분할하였다. 이후, 모델이 특정 네트워크 환경과 플로우 식별자에 과적합되는 것을 방지하기 위한 기본적인 전처리로서, 분할된 PCAP 파일을 구성하고 있는 각 패킷마다 IP/MAC 주소, 포트 번호는 제거하였다.

모델 성능을 측정하는 지표로 각 클래스의 F1 점수를 평균을 내어 계산한 F1 macro 점수와 전체 클래스의 정답률을 계산한 정확도(accuracy)를 사용하였다.

4. 실험 결과

표 1과 표 2는 PacRep 논문 [6]의 방법을 기반으로 재현 실험한 데이터셋 전처리 및 실험 결과를 나타낸다. 처리 과정의 차이로 인해 훈련에 사용한 패킷 수의 경우 [6]보다 감소하였으나, 검증, 테스트에 사용한 패킷 수는 동일하게 맞추었다.

표 1. 재현 실험 데이터셋 전처리 결과 (단위: 패킷수)

datasets	train	valid	test
ISCXVPN	181,416	600	600
DoHBrw	99,413	125	125
USTCTFC	241,365	475	475
Total	522,194	1,200	1,200

표 2. 재현 실험 결과

	ISCXVPN		DoHBrw		USTCTFC	
	task 1	task 2	task 3	task 4	task 5	task 6
정확도	1.0000	0.9833	1.0000	0.9680	1.0000	0.9633
F1 (macro)	1.0000	0.9828	1.0000	0.9744	1.0000	0.9744

표2는 PacRep 논문에서 보고된 F1 Macro와 유사한 수준(task 1, 3, 5의 경우 1% 이내, task 2, 4, 6의 경우 각각 1%, 3%, 8% 이내 차이)의 성능을 보이고 있다. 그러나 위 실험 결과는 데이터셋을 학습, 검증, 테스트 데이터셋으로 분할하는 과정에서 학습 및 테스트셋에 동일 플로우에 속한 패킷 여부를 고려하지 않았다. 이러한 결과는 데이터 스누핑 오류를 범한 결과일 수 있으므로 표 3과 같이 데이터셋을 플로우 단위로 재구성하여 실험을 진행하였으며, 표 4는 실험 수행 결과를 나타낸 것이다.

표 3. 플로우 단위 실험 데이터셋 전처리 결과

dataset	packets	flows
train	171,740	19,631
valid	20,614	2,460
test	21,552	2,453

표 4. 플로우 단위 실험 결과

	ISCXVPN		DoHBrw		USTCTFC	
	task 1	task 2	task 3	task 4	task 5	task 6
정확도	0.9779	0.7747	1.0000	0.8588	1.0000	0.8215
F1 (macro)	0.9774	0.7191	1.0000	0.8622	1.0000	0.8468

표 4의 실험 결과에서 앞선 실험에 비해 모델의 분류 성능이 하락하는 현상이 관찰되었으며, 특히 트래픽을 세밀하게 분류하는 task 2, task 4, task 6에서 상당한 성능 하락이 관찰되었다.

표 2와 표 4의 결과를 비교하였을 때, PacRep의 기존 성능 평가 결과는 데이터 스누핑 오류를 범한 것으로 보아야 할 것이다. 특히, 동일 암호화 플로우에서 비롯된 두 패킷이 우연히 암호화된 페이로드에서 유사성을 보일 가능성은 높지 않고, 따라서 플로우 단위 분할을 하지 않더라도 암호화된 페이로드로 인해 데이터 스누핑 오류가 발생할 가능성 또한 높지 않으므로, 표 4의 실험에서 PacRep의 성능 감소는 디코딩한 텍스트 데이터로 인한 결과일 가능성이 높다고 판단된다.

한편, [9]에서는 ET-BERT를 비롯한 바이트 시퀀스 기반 트래픽 분류 모델에서 주로 사용하는 암호화된 페이로드의 학습이 암호화 알고리즘이 강건하다는 전제 하에 거의 또는 전혀 타당하지 않다고 주장한다. PacRep은 분류 모델의 입력으로 패킷 헤더 등에서 디코딩한 텍스트 데이터 뿐만 아니라 암호화된 페이로드도 사용하고 있으므로, 위 주장에 따르면 과적합 가능성이 있으므로 암호화된 페이로드가 입력으로 주어지지 않는 상황에서 패킷에 대한 표현 학습의 성능을 알아볼 가치가 있다. 이에 따라 페이로드를 제거한 조건에서 추가 실험을 진행하였다.

표 5는 페이로드를 제거한 후, 기존 플로우 단위 실험과의 비교를 위해 플로우 개수를 유지하도록 전처리한 결과로, 표 3과 플로우 수는 동일하지만 샘플링을 다시 했기 때문에 패킷 수가 미세하게 차이가 난다. 표 6과 그림 1의 녹색 막대는 해당 데이터셋으로 수행한 페이로드 제거 실험의 결과를 나타낸 것이다.

표 5. 페이로드 제거 실험 데이터셋 전처리 결과

dataset	packets	flows
train	172,641	19,631
valid	21,389	2,460
test	20,531	2,453

표 6. 페이로드 제거 실험 결과

	ISCXVPN		DoHBrw		USTCTFC	
	task 1	task 2	task 3	task 4	task 5	task 6
정확도	0.9731	0.6173	1.0000	0.8342	0.9964	0.6634
F1 (macro)	0.9726	0.5470	1.0000	0.8384	0.9963	0.5457

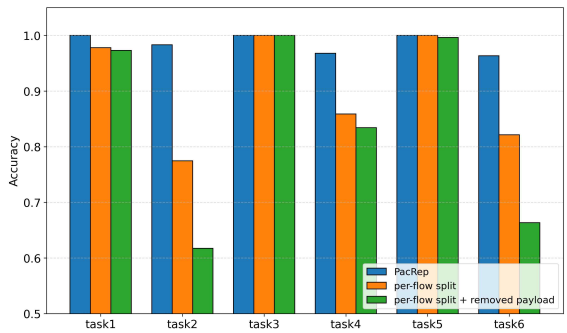

그림 1. PacRep, 플로우 단위, 페이로드 제거 실험의 태스크별 분류 정확도 비교

이진 분류(task 1, 3, 5)의 경우, 페이로드 제거 전후의 분류 성능이 차이가 크지 않아, 디코딩한 텍스트 데이터만으로도 비교적 정확하게 이진 분류가 가능함을 보여준다. 반면, 다중 클래스 분류(task 2, 4, 6)에서는 성능이 크게 저하된 것을 관찰하였다. 이러한 결과는 기존 PacRep 모델이 암호화된 페이로드로부터 특정 패턴을 학습하며 과적합이 발생하여 모델 학습에 영향을 미쳤을 가능성을 시사한다. 실제로, task 2와 task 6의 데이터셋인 ISCXVPN2016 및 USTCTFC2016에는 암호화되지 않은 트래픽이나 구식 암호화 알고리즘을 사용한 트래픽이 다수 포함되어 있음이 알려져 있다 [12].

5. 결론

본 연구에서는 패킷 표현 학습 기반 트래픽 분류 프레임워크인 PacRep의 기존 성능 평가가 데이터 스누핑 오류에 의한 것임을 간접적으로 확인하고자, 데이터셋을 플로우 단위 기반으로 데이터를 재구성하여 실험을 수행하였다. 그 결과, 모델의 분류 성능이 하락하는 현상이 관찰되었으며, 이는 기존 PacRep 실험에서 데이터 스누핑으로 인한 과적합이 발생했을 가능성을 시사한다. 또한 재구성된 데이터셋에서 PacRep이 암호화된 페이로드에 대한 의존성을 확인하여, 이 결과가 데이터셋의 암호화된 페이로드에 내포된 구조적 패턴을 학습했을 가능성을 논의하였다. 이를 통해 후속 연구에서는 보다 통제된 데이터셋을 통해 분류 모델에 대한 성능 평가가 이루어질 필요가 있음을 알 수 있다.

[참고 문헌]

[1] O. Barut, Y. Luo, T. Zhang, W. Li, and P. Li, "Multi-task Hierarchical Learning Based Network Traffic Analytics," in *Proc. of IEEE ICC*, 2021.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Proc. of NeurIPS*, 2017.

[3] J. Ha and H. Roh, "Experimental Evaluation of Malware Family Classification Methods from Sequential Information of TLS-Encrypted Traffic," *Electronics*, 2021.

[4] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu, "ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification," in *Proc. of ACM Web Conference*, 2022.

[5] H. Shon, J. Im, S. Eom, J. Oh, and M. Yoon, "Packet-Length is All You Need: Application Classification from Encrypted Traffic," in *Proc. of KIISC CISC-S*, 2025.

[6] X. Meng, Y. Wang, R. Ma, H. Luo, X. Li, and Y. Zhang, "Packet Representation Learning for Traffic Classification," in *Proc. of ACM KDD*, 2022.

[7] T. Wang and P. Isola, "Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere," in *Proc. of ICML*, 2020.

[8] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and Don'ts of Machine Learning in Computer Security," in *Proc. of USENIX Security*, 2022.

[9] Y. Zhao, G. Dettori, M. Boffa, L. Vassio, and M. Mellia, "The Sweet Danger of Sugar: Debunking Representation Learning for Encrypted Traffic Classification," in *Proc. of ACM SIGCOMM*, September 2025.

[10] S. Rezaei and X. Liu, "Deep Learning for Encrypted Traffic Classification: An Overview," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 76-81, May 2019.

[11] Netresec, Splitcap, https://www.netresec.com/?page=SplitCap

[12] N. Wickramasinghe, A. Shaghaghi, G. Tsudik, and S. Jha, "SoK: Decoding the Enigma of Encrypted Network Traffic Classifiers," in *Proc. of IEEE S&P*, 2025.