태스크-관계적 네트워크 트래픽 분류 연구의 필드 특징 중요도 분석

장윤성[1], 박은혁[1], 심규상[1], 정의종[1], 유경민[1], 최양서[2], 김명섭[1]
[1]고려대학교 컴퓨터 정보학과, [2]한국전자통신연구원 지능형네트워크보안연구실
e-mail : brave1094@korea.ac.kr

Task-Relational Field-Level Feature Importance Analysis of Deep Learning-Based Network Traffic Classification

Yun-Seong Jang[1], Eun-Hyeok Park[1], Gyu-Sang Shim[1], Ui-Jong Jeong[1], Gyeong-Min Yu[1], Yang-Seo Choi[2], Myung-Sup Kim[1]
[1]Department of Computer and Information Science, Korea University,
[2]Department of Cyber Security Research Division, ETRI

요 약

최근 암호화 트래픽의 확산으로 바이트열을 직접 학습하는 딥러닝(DL) 기반 네트워크 트래픽 분류(NTC)가 표준이 되면서, 모델이 어떤 필드를 근거로 분류하는지 규명하는 특징 중요도 분석이 이루어지고 있다. 그러나 기존 분석은 모델이 중요하게 본 필드를 보고하는 수준에 머물러, 태스크에 따른 필드 중요도 차이나 네트워크 관점에서 그 필드가 정당한 특징인지를 판정하지 못한다. 본 논문은 RFC 표준에 근거해 IPv4·TCP·UDP 필드를 선험 판단군(제거·조건부·사용)으로 정의하고, 11종 데이터셋과 5종 DL 모델의 SHAP 중요도를 태스크(NTC·응용·공격)별로 대비하며 필드 크기를 통제한 Lift와 데이터셋별 편차·클래스 분리도로 분석한다. 그 결과 ip.src·ip.dst·포트·timestamp가 제 크기의 1.6~3.1배로 과주목되고 정당한 콘텐츠 신호인 tcp.payload는 0.61배로 미달함을 확인하였으며, 값 하나로 클래스가 갈리는(cic18의 ip.src 97%) 과적합 의심 필드와 태스크-관계적 필드 사용 권고를 제시한다.

Ⅰ. 서 론

네트워크 트래픽 분류(Network Traffic Classification, NTC)는 트래픽을 응용·서비스·공격 등의 클래스로 식별하는 기술로 QoS 보장·과금·침입 탐지(IDS)의 기반이며, 암호화 확산으로 페이로드 관측이 어려워지면서 최근에는 원시 바이트 시퀀스를 직접 학습하는 DL 기반 접근이 표준이 되었다[1-6]. 이에 모델이 어떤 필드를 근거로 분류하는지를 규명하는 특징 중요도 분석이 요구되나[7,8], 모델이 크게 의존하는 필드의 상당수는 수집 환경에서 비롯된 지름길(shortcut)이어서[9] 중요도만으로 입력을 줄이면 과적합을 고착시킬 수 있다.

따라서 필요한 기준은 '중요하면서도 네트워크 관점에서 정당한' 필드이나, 기존 분석은 대부분 정당성 판정 체계를 제공하지 못한다. 본 논문은 IPv4·TCP·UDP 필드를 태스크-관계적으로 분석하여, (1) RFC 값-생성 원리로 사용가능성(제거·조건부·사용)을 선험 판정해 실측과 대조하고, (2) 5종 DL 모델과 11종 데이터셋의 SHAP 중요도를 NTC·응용·공격으로 대비하며 Lift로 과주목 필드를 분리하고, (3) 데이터셋별 중요도 편차와 클래스 분리도로 과적합 의심 필드를 식별한다.

Ⅱ. 관련 연구

본 연구는 11종 데이터셋을 응용 분류 5종, 공격 탐지 5종, 둘을 합친 NTC 전체의 세 태스크로 구성해 사용한다. 응용 분류는 ISCX VPN-nonVPN[10]·Tor-nonTor[11], CipherSpectrum[12]와 자체 수집 private#1(대전, 2023)·private#2(조치원, 2024)를, 공격 탐지는 USTC-TFC[2], CIC-IDS2017[13]·2018[14], CICIoT2023[15]와 자체 수집 private#3(대전, 2024)를 사용하며, CSTNET-TLS 1.3[3]은 NTC 전체에 포함한다(클래스 수 8~120종). 모델은 원시 바이트 시퀀스를 입력하는 대표 구조 5종, 즉 2D-CNN[2], ET-BERT[3], YaTC[4], NetMamba[5], TrafficFormer[6]를 사용한다.

바이트·필드 입력 DL-NTC 논문 54편(2020 - 2025) 중 네트워크 관점에서 특징의 의미를 해석한 논문(A등급)은 7편(13%)에 불과하다. Akbari 등[7]은 모델이 서버명을 라벨 룩업처럼 이용함을 밝혔고, TRUSTEE[9]와 SoK[8]는 IP·포트·순서번호·timestamp 등 플로우 식별자가 배포 환경에서 붕괴하고 다수 필드의 정보가 누출됨을 보였다. 그러나 이들은 개별 사례에 집중하거나 대리모델·occlusion에 의존해 태스크별 대비와 값-생성 원리 기반의 일관된 판정 틀이 없다. 본 논문은 5종 모델의 SHAP[16]로 실제 부여 중요도를 직접 측정하고 세 태스크로 대비하며 RFC 원리로 정당성을 일관 판정한다.

Ⅲ. 본 론

제안 방법은 그림 1과 같이 (1) RFC 선험 판단군 정의, (2) 11종 데이터셋 × 5종 모델의 실험 모델 기준 성능 학습, (3) 바이트 SHAP의 필드 집계와 태스크·Lift·분리도 분석의 3단계로 구성된다.

3.1 RFC 선험 판단군

분석 대상은 IPv4 11개, TCP 14개, UDP 4개의 헤더 필드에 tcp.payload를 더한 30개 필드이며, 1바이트를 공유하는 (ip.version + ip.hdr_len)은 하나로 병합해 측정한다. 각 필드의 생성·결정 규칙은 RFC 표준을 근거로 한다. IPv4는 RFC 791·2474·6864, TCP는 RFC 9293·7323·2018, UDP는 RFC 768을 참고하며, 무결성 필드 계산은 RFC 1071을 따른다[17]. 각 필드가 RFC상 어떻게 결정되는지에 따라 실측 전에 세 판단군으로 분류한다. (1) 제거: 호스트에 할당되거나 수신 시 재계산되거나 스택이 임의로 부여하는 값으로 클래스와 무관하다. (2) 조건부: 연결 상태·경로·정책·협상에 따라 정해지는 필드로 태스크 관련성이 데이터에 의존한다. (3) 사용: 페이로드 크기·상위 프로토콜·내용에서 산출되는 필드로 정당한 분류 신호가 될 수 있다.

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구이며(00235509, ICT융합 공공 서비스●인프라의 암호화 사이버위협에 대한 네트워크 행위기반 보안관제 기술 개발), 2026년도 정부(산업통상부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임(P0031136, 2026년 중부권 지역혁신클러스터육성)

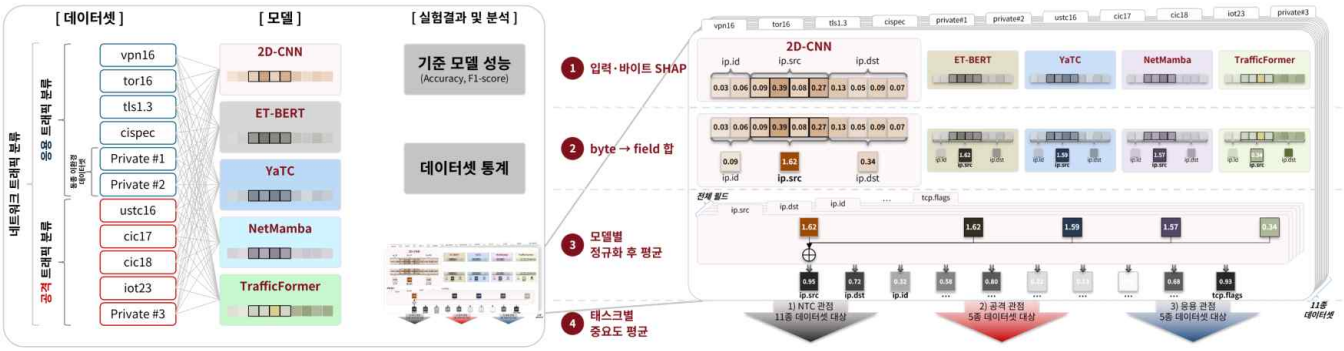


그림 1. 특징 중요도 분석에 사용된 프레임워크

모델	vpn16	tor16	tls1.3	cispec	private#1	private#2	ustc16	cic17	cic18	iot23	private#3
2D-CNN [2]	82.2	95.6	58.9	76.8	66.1	58.2	93.9	95.2	99.8	87.4	97.2
ET-BERT [3]	92.4	92.2	90.4	94.7	92.9	86.2	98.7	99.7	97.5	97.2	99.9
YaTC [4]	91.5	94.7	95.6	97.8	91.1	88.8	71.7	99.7	99.7	96.4	99.1
NetMamba [5]	89.8	95.3	94.8	95.2	89.4	88.2	98.3	99.6	99.8	93.3	99.7
TrafficFormer [6]	86.4	93.7	84.8	28.2	78.0	75.7	97.6	99.7	99.5	90.0	99.7

표 1. 실험 모델 기준 성능 정확도(%)

특히 조건부 필드 중 tcp.options.timestamp 옵션(tcp.options.timestamp.tsval, tcp.options.timestamp.tsecr)은 송신 호스트의 부팅 시점을 기준으로 증가하여 결정되는 값으로, 응용·공격 트래픽의 내용과 무관하게 시각의 흐름과 송신 호스트만을 반영한다. 따라서 실측에서 이 필드가 중요한 특징으로 사용된다면 정당한 근거가 아니라 수집 시간대·호스트를 드러내는 환경 지문일 가능성이 높다.

3.2 실험 모델 기준 성능

11종 데이터셋과 5종 모델의 55개 조합을 각 모델의 공개 사전학습 파라미터와 기준 에폭으로 독립 학습하여 기준 성능을 확보하고, 이 학습된 모델에서 특징 중요도를 추출한다. 입력·전처리는 모델별 원본 설정을 따른다.

표 1과 같이 대부분의 조합이 90% 이상의 정확도를 보였고 모델 평균은 82.8 ~ 94.9%였다. tls1.3의 2D-CNN(58.9%)과 cispec의 TrafficFormer(28.2%)처럼 낮은 조합도 있으나(모델·데이터셋 상성), 중요도 분석은 각 조합에서 실제 학습된 모델을 대상으로 하므로 기준 성능으로 충분하다.

3.3 필드 중요도 집계·Lift·분리도

각 학습 모델에 SHAP를 적용하여 입력 바이트별 기여도(절대값)를 구하고, 바이트 위치를 프로토콜 필드에 역매핑하여 필드 단위로 합산한다. 패딩·구분 토큰과 마스킹 바이트를 제외한 뒤 모델별 합이 1이 되도록 정규화하고, 5개 모델 → 데이터셋 → 태스크(NTC 11종·응용 5종·공격 5종) 순으로 평균한다. 원시 중요도는 필드의 바이트 수에 비례해 커 보이므로, 중요도 비율을 그 필드가 입력 창에서 차지한 바이트 위치 비율(점유율)로 나눈 Lift = 중요도 ÷ 점유율을 정의한다. 또한 '필드 값만으로 클래스가 갈리는가'를 보기 위해, 테스트 세션 첫 패킷의 필드 값을 복원하고 값의 순위 백분위를 20분위로 나눈 뒤 각 분위의 다수 클래스로 예측한 정확도 acc(1-ft)(우연 1/K)를 측정한다. 분리가 잘 됨이 곧 정당함은 아니며, RFC상 태스크와 무관한 필드가 선명히 갈린다면 그것이 곧 수집 환경 지문의 증거이다.

Ⅳ. 실험 평가

4.1 필드별 기여도와 Lift

Lift = 1.0은 필드가 제 크기만큼 주목받았음(중립)을 뜻하며, 1보다 크면 과주목, 작으면 미달이다. 본 논문은 Lift를 표 2의 5단계로 해석한다. 경계값은 기준선 1.0의 배수(×0.5, ×2, ×4) 체계이면서, 전체 필드-데이터셋 Lift의 상위 약 5%가 2.0 부근, 최댓값이 4.0 부근에 분포하는 실측 분위점과 일치한다.

Lift 범위	해석	예시 필드(실측Lift)
≥ 4.0	극단 과주목	일부 데이터셋의 tcp.options.timestamp.tsval (최댓값≈ 4.0)
2.0 - 4.0	강한 과주목	tcp.options.timestamp.tsval(3.14), tcp.options.timestamp.tsecr(2.51)
1.0 - 2.0	과주목	ip.dst(1.82), tcp.srcport(1.78), ip.src(1.65)
0.5 - 1.0	미달	ip.len(0.88), tcp.seq(0.71), tcp.payload(0.61)
< 0.5	강한 미달	tcp.options.wscale.shift(0.48), (ip.version + ip.hdr_len)(0.48)

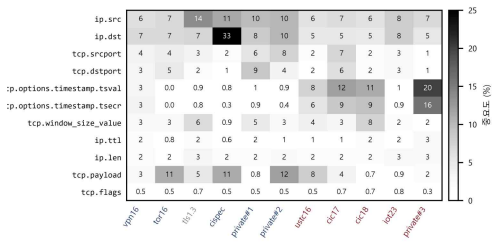
표 2. Lift 5단계 해석 기준

표 3은 30개 필드의 태스크별 중요도(%)와 Lift이다. 선험 '제거'로 둔 ip.src·ip.dst 및 포트 필드는 Lift 1.3~1.8로 과주목이 확인되고, '조건부'인 tcp.options.timestamp.tsval·tcp.options.timestamp.tsecr는 Lift 2.5~3.1로 가장 강한 과주목을 보여 실측상 수집 환경 지문에 가깝다. 반면 '사용'으로 둔 tcp.payload·ip.len은 Lift 0.6~0.9로 미달이다. 같은 '제거'군이라도 ip.id·ip.checksum·tcp.seq·tcp.ack는 Lift 0.7~1.2로 과주목이 아니라 정보 없는 바이트를 크기만큼 흡수할 뿐이며, tcp.flags는 0.56%(Lift 0.55)로 방치되어 있다.

4.2 데이터셋에 따른 특징 중요도 차이

그림 2는 주요 필드의 중요도를 11종 데이터셋으로 분해한 히트맵이다. 같은 필드의 중요도가 데이터셋마다 크게 흔들리며, 그 편차의 상당수는 수집 설계로 설명된다. ip.dst의 응용 평균 13.0%는 앱마다 서버가 고정된 cispec의 33.0%가 끌어올린 값이고, tcp.options.timestamp.tsval의 공격 평균 10.4%는 페이로드 없는 클래스가 많은 private#3의 20.0%가 주도한다. 반대로 timestamp 옵션이 없는 tor16에서는 그 중요도가 0이다. 이러

그림 2. 데이터셋별 필드 중요도(%)



프로토콜	필드	선험	NTC	응용	공격	Lift
IPv4	(ip.version + ip.hdr_len)	조건부	0.63	0.56	0.76	0.48
IPv4	ip.dsfield	조건부	0.96	0.63	1.16	0.74
IPv4	ip.len	사용	2.28	2.11	2.27	0.88
IPv4	ip.id	제거	2.96	2.59	3.47	1.14
IPv4	ip.flags	조건부	1.37	1.54	1.15	1.05
IPv4	ip.frag_offset	조건부	0.89	0.72	1.14	0.70
IPv4	ip.ttl	조건부	1.56	1.28	1.78	1.23
IPv4	ip.proto	사용	1.06	0.85	1.36	0.84
IPv4	ip.checksum	제거	2.90	2.73	3.21	1.15
IPv4	ip.src	제거	8.34	8.91	6.69	1.65
IPv4	ip.dst	제거	9.20	13.03	5.74	1.82
TCP	tcp.srcport	제거	3.86	4.70	3.09	1.78
TCP	tcp.dstport	제거	3.46	4.38	2.77	1.64
TCP	tcp.seq	조건부	2.90	3.32	2.72	0.71
TCP	tcp.ack	조건부	2.71	2.94	2.55	0.67
TCP	tcp.hdr_len	조건부	0.76	0.60	0.89	0.75
TCP	tcp.flags	조건부	0.56	0.48	0.60	0.55
TCP	tcp.window_size_value	조건부	3.46	2.74	3.58	1.70
TCP	tcp.checksum	제거	1.72	1.68	1.71	0.85
TCP	tcp.options.mss_val	조건부	0.82	0.69	1.11	0.84
TCP	tcp.options.wscale.shift	조건부	0.17	0.18	0.20	0.48
TCP	tcp.options.sack_le	조건부	0.01	0.00	0.00	1.17
TCP	tcp.options.sack_re	조건부	0.01	0.00	0.00	1.36
TCP	tcp.options.timestamp.tsval	조건부	5.37	1.21	10.44	3.14
TCP	tcp.options.timestamp.tsecr	조건부	4.20	1.01	8.07	2.51
UDP	udp.srcport	제거	0.45	0.56	0.43	1.33
UDP	udp.dstport	제거	0.41	0.54	0.35	1.21
UDP	udp.length	사용	0.23	0.22	0.28	0.69
UDP	udp.checksum	제거	0.21	0.20	0.27	0.62
L7	tcp.payload	사용	5.30	7.54	3.09	0.61

표 3. L3/L4 필드별 중요도(%)와 Lift (선험 = RFC 판단군)

구분	vpn16	tor16	tls1.3	cispec	private#1	private#2	ustc16	cic17	cic18	iot23	private#3
암호화	32	14	100	100	65	75	18	2	3	1	47
3Way - HS 없음	0	0	100	0	0	0	50	0	39	40	8
페이로드 없음	49	33	23	49	69	32	69	47	26	81	84

표 4. 데이터셋 수집 통계 (%)

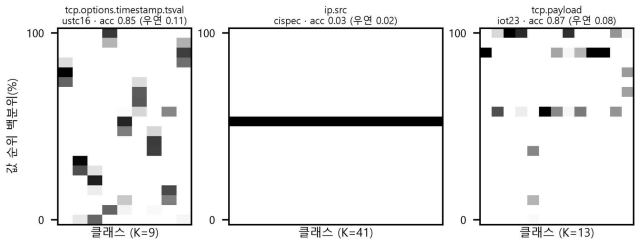
한 편차는 표 4의 수집 통계로 설명된다.

나아가 클래스 분리도를 그림 3으로 확인한다. ustc16에서는 악성 클래스가 서로 다른 시간대에 수집되어 tcp.options.timestamp.tsval 값 하나로 세션의 85%를 맞출 수 있고(우연 11%), 반면 41개 앱이 단일 IP로 수집된 cispec에서는 같은 ip.src로 구분이 불가능하다(3%). 정당 신호의 예로 iot23의 tcp.payload는 길이만으로 87%를 분리한다. 분리가 잘 됨이 곧 정당함은 아니며, 태스크와 무관한 ip.src·timestamp 필드가 선명히 갈린다면 그 분리는 수집 흔적에서 온 것이고 환경이 바뀌면 붕괴할 과적합 의심의 근거가 된다.

4.3 태스크-관계적 필드 중요도와 Lift

응용 5종과 공격 5종의 평균 중요도(표 3의 응용·공격 열)를 그림 4의 덤벨로 대비하면, 응용은 ip.dst(13.0%)와 tcp.payload(7.5%)에, 공격은 tcp.options.timestamp.tsval(10.4%)·tcp.options.timestamp.tsecr(8.1%)에 의존이 집중된다. 크기를 통제한 Lift로 보아도 지름길의 과주목은 유지되어(ip.src·ip.dst·포트 1.6~1.8, timestamp 2.5~3.1) tcp.payload의 미달(0.61)과 대비되며, 선험 '제거'군(ip.src·ip.dst·포트)의 중요도 합은 응용 43.5%, 공격 33.0%에 이른다.

그림 3. 클래스 분리도 (값 순위 20분위 × 클래스)



태스크 축 분류(큰 쪽이 1.5% 이상이고 1.5배 이상이면 태스크-특유)에 따르면 공격-특유는 timestamp 2개뿐이고, 응용-특유는 ip.dst·tcp.payload·tcp.srcport·tcp.dstport 4개, 공통 10개, 저관여 14개다. 태스크-특유가 곧 정당은 아니다. 응용은 앱을 정의하는 값(서버·콘텐츠)이라는 정당 축으로, 공격은 송신 호스트의 부팅 이후 시간을 반영하는 timestamp라는 지문 축으로 갈렸다. 어느 패킷을 보는지도 갈린다(표 5). 공격은 첫 패킷(핸드셰이크)에 31.0%, 응용은 첫 데이터 패킷인 4번째에 30.1%가 집중된다.

그림 4. 태스크 축 대비 (● 응용, ■ 공격)

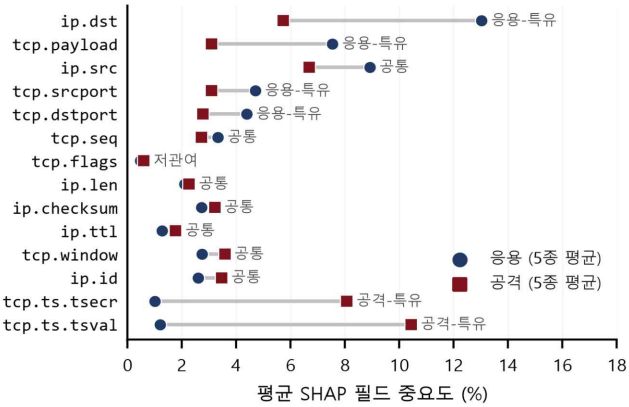


표 5. 패킷 위치별 중요도(%)

관점	1번째	2번째	3번째	4번째	5번째
NTC	29.3	20.6	10.0	22.3	15.1
응용	22.1	19.8	9.3	30.1	16.6
공격	31.0	23.6	10.5	16.6	14.5

태스크	권고	필드 리스트	근거
NTC	제거	ip.src, tcp.srcport, tcp.options.timestamp.tsval, tcp.options.timestamp.tsecr, ip.ttl (과주목 지문) · ip.id, ip.checksum, tcp.checksum, udp.checksum, tcp.seq, tcp.ack, 상수·희소 필드 (정보 없음)	값이 수집 환경·호스트·재계산으로 결정. ip.src·ip.dst·포트·timestamp는 Lift 1.6~3.1로 과주목, 나머지는 분류 정보 없이 입력만 차지
NTC	주의	ip.dst, tcp.dstport, udp.srcport, udp.dstport, tcp.window_size_value	서버·서비스 식별에 정당할 수 있으나 고정 서버·수집 설계 시 지문으로 전락(cispec의 ip.dst 33%)
NTC	사용	tcp.payload, ip.len, udp.length	페이로드 내용·크기에서 산출되는 정당한 콘텐츠 신호 (tcp.payload 단독으로 iot23 87% 분리)
응용	제거	ip.src(8.9%), tcp.srcport(4.7%), ip.ttl, tcp.options.timestamp.tsval·tcp.options.timestamp.tsecr (응용 1.2%) · ip.id, ip.checksum, tcp.seq, tcp.ack, 상수 필드	ip.src·포트가 과주목되나 앱 본질과 무관. timestamp는 응용에서 거의 미사용(1.2%)
응용	주의	ip.dst(13.0%), tcp.dstport(4.4%), udp.srcport, udp.dstport, tcp.window_size_value	앱이 접속하는 서버·서비스를 지시해 정당하나 앱별 고정 서버 수집이면 지문(cispec 33%)
응용	사용	tcp.payload(7.5%), ip.len, udp.length	응용의 콘텐츠·크기 신호로 가장 정당
공격	제거	tcp.options.timestamp.tsval(10.4%), tcp.options.timestamp.tsecr(8.1%) [최우선], ip.src(6.7%), tcp.srcport, ip.ttl · ip.id, ip.checksum, tcp.seq, tcp.ack, 상수 필드	최상위 의존이 timestamp – 시간대별 순차 수집 흔적(ustc16 tcp.options.timestamp.tsval 85% 분리). ip.src도 대역 종속(cic18 97%)
공격	주의	ip.dst, tcp.dstport, tcp.window_size_value	C&C 서버·포트 식별의 단서가 될 수 있으나 데이터셋 IP 대역·수집 시점에 종속
공격	사용	tcp.payload(3.1%), ip.len, udp.length, tcp.flags	페이로드·크기·연결 상태는 정당하나 공격셋은 페이로드 부재 잦아 (private#3 84%) 활용 제한

표 6 태스크별 필드 사용 권고 (제거 / 주의 / 사용)

4.4 종합 분석

RFC 값-생성 원리(선험), 실측 중요도·Lift, 클래스 분리도를 결합하여 각 태스크별 필드 사용 권고를 표 6으로 정리한다. 권고는 '제거 / 주의(사용 시 주의) / 사용'의 3단계이며 선험 판단군을 실측으로 재판정한 결과이다(예: ip.dst는 선험상 '제거'군이나 응용에서 서버 식별의 정당한 단서가 되어 '주의'로 상향).

연결 상태를 담는 tcp.flags는 특히 공격 탐지에서 정당한 신호이나 0.56%(Lift 0.55)로 방치되어 '사용'으로 권고하되 활용 확대가 필요하다. 종합하면 필드 중요도 보고는 클래스 구성·수집 방식·분리도의 실측을 병기할 때에만 올바르게 해석될 수 있다.

Ⅳ. 결 론

본 논문은 딥러닝 기반 네트워크 트래픽 분류(NTC)의 필드 특징 중요도를 RFC 값-생성 원리(선험), 태스크-관계적 대비, 필드 크기를 통제한 Lift, 데이터셋 통계·클래스 분리도로 판정하는 네트워크 관점의 분석을 제안하고 5종 모델 × 11종 데이터셋에서 실측하였다. 응용은 ip.dst·tcp.payload, 공격은 tcp.options.timestamp를 주로 사용하는 상반된 의존을 확인하였고, ip.src·ip.dst·포트·timestamp의 과주목(Lift 1.6~3.1)과 tcp.payload의 미달(0.61)을 보였으며, 데이터셋 의존성과 클래스 분리도(cic18의 ip.src 97%)를 근거로 과적합 의심 필드를 식별하였다. 또한 공격 분류는 페이로드가 담긴 뒤쪽 패킷보다 초기 핸드셰이크 패킷에 의존하여 내용이 아닌 헤더 지문에 기대는 경향을 보였다. 이로써 본 연구는 중요도 해석에 부재했던 정당성 판정의 기준을 RFC 선험 판단군과 태스크-관계적 Lift·분리도를 결합해 제시하고, 같은 필드가 태스크에 따라 정당한 신호와 환경 지문으로 갈림을 정량적으로 규명하였다. 나아가 과적합 의심 필드를 태스크별 '제거·주의·사용' 필드 사용 권고(표 6)로 정리하여 후속 모델 설계에 활용할 수 있는 실무 지침을 제공하였다. 그러나 본 연구는 과적합을 '의심' 수준에서 규명한 한계가 있다. 이러한 Shortcut Learning이 NTC 도메인에 나타나는 효과에 대해 향후 같은 태스크·다른 환경 쌍(private#1 ↔ private#2)의 동종이환경 교차검증으로 일반화를 직접 검증하고(예비 실측 교차 정확도 88~94%로 하락), ip.src·ip.dst·포트·timestamp를 마스킹한 재학습으로 중요도가 tcp.payload·L7로 재배분되는지, 모델이 페이로드를 어떤 패킷·바이트 위치에서 보는지를 정밀 분석할 계획이다.

참 고 문 헌

[1] M. Shen, K. Ye, X. Liu, et al., "Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive Survey," IEEE Commun. Surveys Tuts., 2023.

[2] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware Traffic Classification Using Convolutional Neural Network for Representation Learning," Proc. ICOIN, pp. 712-717, 2017.

[3] X. Lin et al., "ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification," Proc. ACM Web Conf. (WWW), pp. 633-642, 2022.

[4] R. Zhao et al., "Yet Another Traffic Classifier: A Masked Autoencoder Based Traffic Transformer with Multi-Level Flow Representation," Proc. AAAI, pp. 5420-5427, 2023.

[5] T. Wang et al., "NetMamba: Efficient Network Traffic Classification via Pre-training Unidirectional Mamba," Proc. IEEE ICNP, 2024.

[6] G. Zhou et al., "TrafficFormer: An Efficient Pre-trained Model for Traffic Data," Proc. IEEE Symp. Security and Privacy (S&P), 2025.

[7] I. Akbari et al., "A Look Behind the Curtain: Traffic Classification in an Increasingly Encrypted Web," Proc. ACM Meas. Anal. Comput. Syst. (SIGMETRICS), Vol. 5, No. 1, pp. 1-26, 2021.

[8] N. Wickramasinghe, A. Shaghaghi, G. Tsudik, and S. Jha, "SoK: Decoding the Enigma of Encrypted Network Traffic Classifiers," Proc. IEEE Symp. Security and Privacy (S&P), 2025.

[9] A. S. Jacobs et al., "AI/ML for Network Security: The Emperor Has No Clothes," Proc. ACM CCS, pp. 1537-1551, 2022.

[10] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Encrypted and VPN Traffic Using Time-related Features," Proc. ICISSP, pp. 407-414, 2016.

[11] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Tor Traffic Using Time Based Features," Proc. ICISSP, pp. 253-262, 2017.

[12] CipherSpectrum: Encrypted Application Traffic Dataset (TLS 1.3), 2024. (서지 확정 필요.)

[13] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," Proc. ICISSP, pp. 108-116, 2018.

[14] Canadian Institute for Cybersecurity, "CSE-CIC-IDS2018 Dataset," unb.ca/cic/datasets/ids-2018.html, 2018.

[15] E. C. P. Neto et al., "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," Sensors, 23(13), 2023.

[16] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," Proc. NeurIPS, pp. 4765-4774, 2017.

[17] J. Postel, W. Eddy, and D. Borman (Eds.), "Internet Protocol (RFC 791), User Datagram Protocol (RFC 768), Transmission Control Protocol (RFC 9293), and TCP Extensions for High Performance (RFC 7323)," IETF.