딥러닝을 활용한 네트워크 이상 탐지

Network Intrusion Detection Using Deep Learning Technique

[TEAM SAMCM] 이민호 | 이서영 | 강승완

Contents

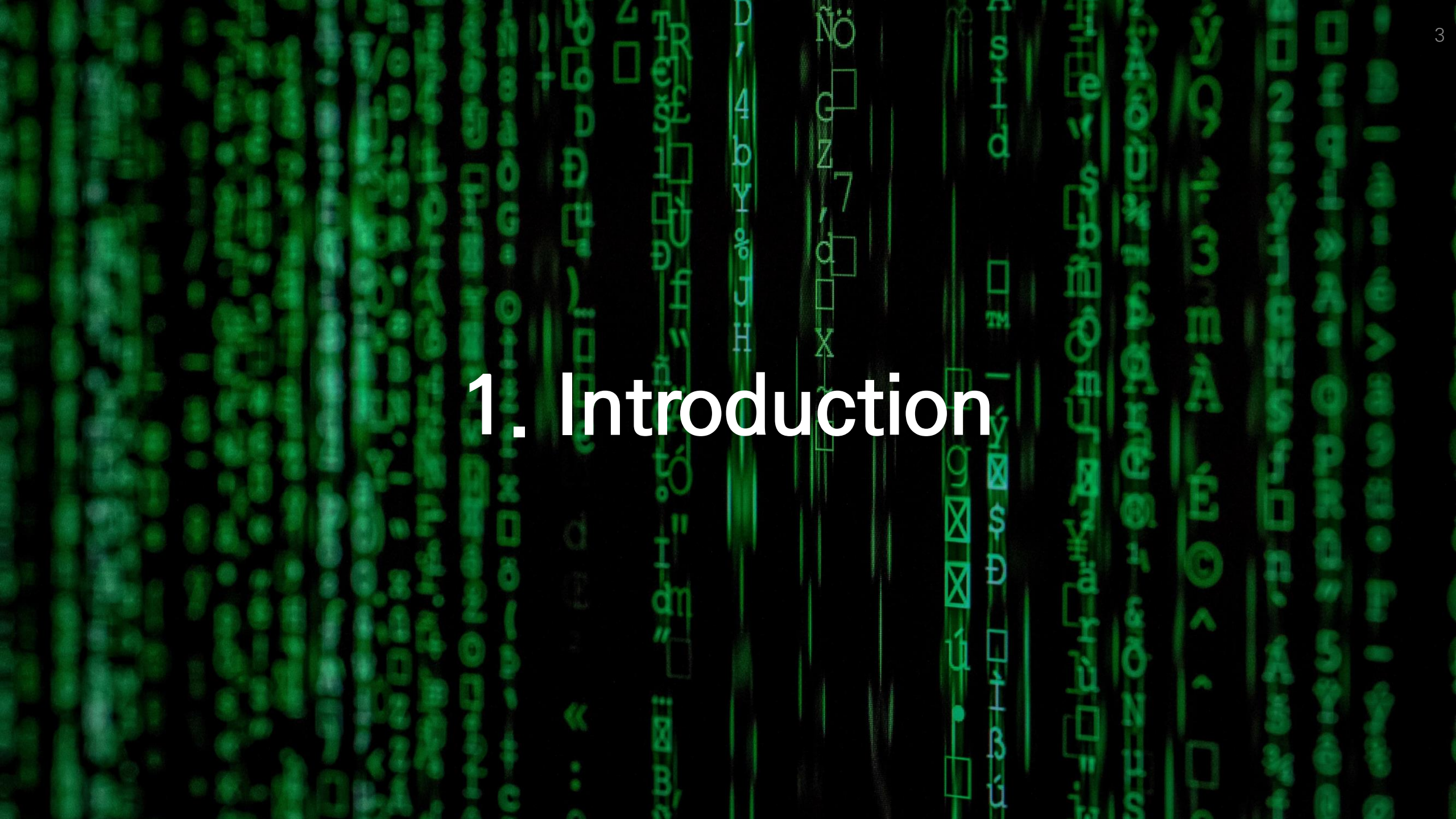
Introduction

Literature Review

Exploratory Data Analysis

Modeling

Conclusion



1. Introduction

Network Intrusion?

- **고려대학교, 중앙대학교 수강신청 서버가 사이버 공격을 받아 마비된 사례 (2020. 08.)**

고려대·중앙대 수강신청 서버 '디도스 공격'...경찰 내사 착수

| 뉴스1 제공 2020.08.20 21:06

https://news.mt.co.kr/mtview.php?no=2020082021008292455&type=1 기사주소 복사

(서울=뉴스1) 이승환 기자 = 고려대학교와 중앙대학교 수강신청 서버가 사이버 공격을 받아 한때 마비된 사건과 관련해 경찰이 수사에 나섰다. 고려대와 중앙대는 수강신청을 중단하고 조만간 재실시하기로 했다.

20일 경찰에 따르면 서울지방경찰청 사이버수사대는 이날 고려대 수강신청 서버가 디도스(DDOS) 공격을 받은 것과 관련해 내사에 착수했다. 디도스는 특정 사이트를 집중 공격해 접속 불능 상태로 만드는 사이버 테러를 말한다.

고려대에 따르면 이날 오전 10시쯤 학교 2학기 수강신청에 대한 대규모 디도스 공격이 감지됐다. 이후 같은 날 오후 2시로 수강신청 시간을 연기했지만 서버는 재차 사이버 테러를 당했다.

학교 측은 정상적인 수강신청이 불가능하다고 판단해 이를 연기했다. 향후 수강신청은 오는 24~25일 진행된다.

다양한 형태의 Network Intrusion가 존재하고, 이를 탐지 · 예방할 필요가 있음

✓ 기사 원문 : https://news.mt.co.kr/mtview.php?no=2020082021008292455

Network Intrusion? (계속)

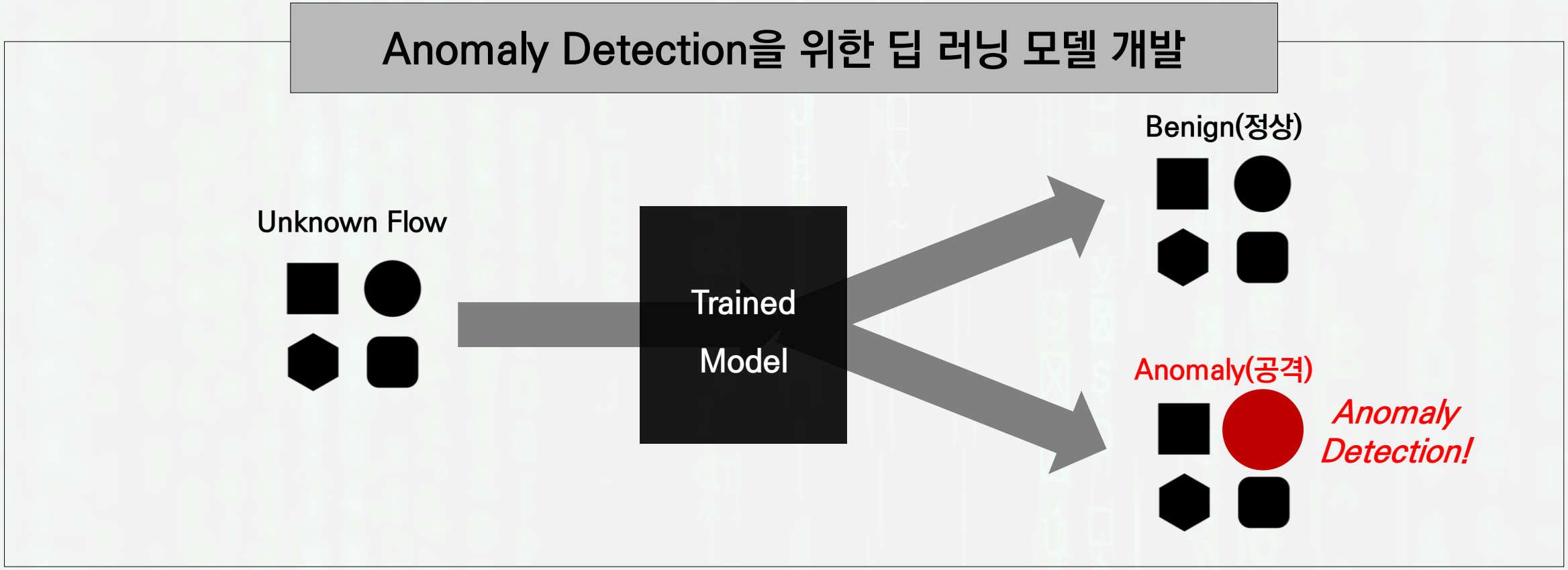
- IDS(Intrusion Detection System)

  - 패킷* 사이의 연관성을 통해 네트워크 침입을 탐지하는 시스템
  - 최근 다양한 형태의 공격이 발생하면서, 그 중요성이 부각되고 있음
  - 대표적인 IDS 방법 : Anomaly-based Detection

Cf. Packet(패킷)
- 데이터 전송에 사용되는 데이터 묶음
- 정상 상태의 패킷의 모양과 비정상 상태의 패킷의 모양이 다르게 나타남

✓ **Photo:** https://fedtechmagazine.com/article/2020/04/intrusion-detection-systems-what-are-they-and-how-do-they-work-perfcon

프로젝트 목표



새로운 유형의 Network Intrusion도 탐지할 수 있는, 이상 탐지 기반 딥 러닝 모델을 개발하는 것이 최종 목표

2. Literature Review

Anomaly

- Anomaly : 정상 데이터와 본질적으로 다른 데이터
- Ex) 호랑이가 정상 데이터일 때, 백호는 Anomaly가 아니지만, 말, 치타 등은 Anomaly에 해당함

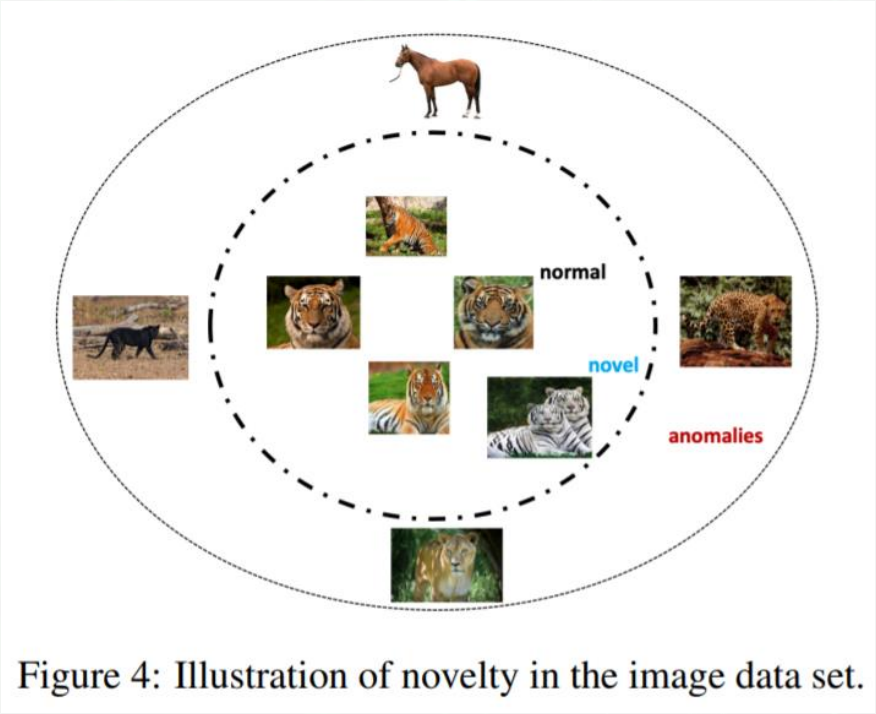


Figure 4: Illustration of novelty in the image data set.

✓ Paper: Deep Learning Model for Anomaly Detection: Survey

Label of Anomalies

- AD(Anomaly Detection) 학습에 있어서 가장 중요한 요소는 Label 활용 가능 여부
- 최근 산업 및 학계에서 가장 많이 활용되는 방식은 Semi-supervised Learning 모델

Supervised Learning	Unsupervised Learning	Semi-supervised Learning
– 라벨이 있어서, **높은 탐지 성능**을 보임 – 사실상 Classification 문제와 동일하기 때문에 **다양한 분류 모델 활용 가능** – 하지만, 라벨 정보가 존재한다는 것은 전문 인력의 확보 등 **많은 기회비용**이 존재 – **새로운 유형에 대처하기가 쉽지 않음** – AD 특성상, Class **불균형** 문제 발생	– **어떠한 경우라도 사용**할 수 있는 방법 – 전체 데이터 셋의 대부분이 정상 데이터라는 가정이 필요함 – 앞선 두 경우와 비교하여 일반적으로 가장 성능이 떨어지고, noise에 민감함	– **가장 비용 현실적인 방법** – 정상 데이터로 학습하고, Anomaly **데이터와의 차이**를 바탕으로 구분 – 정상 데이터로만 학습하기 때문에, Representation Feature**가 과적합될 수 있음** – 지도 학습과 비교하여 모델 성능이 낮음

✓ Paper: Deep Learning Model for Anomaly Detection: Survey

Auto-Encoder

- 입력 벡터 X 의 차원을 Z-dim으로 축소하고, 다시 원래의 차원으로 복원하여 Y를 재생성하는 모델
- X와 Y 사이의 Loss를 줄이는 방식으로 학습 = 나 자신으로 최대한 잘 복원할 수 있도록 설계
- Z를 Hidden representation, Latent vector, Code라고 부름

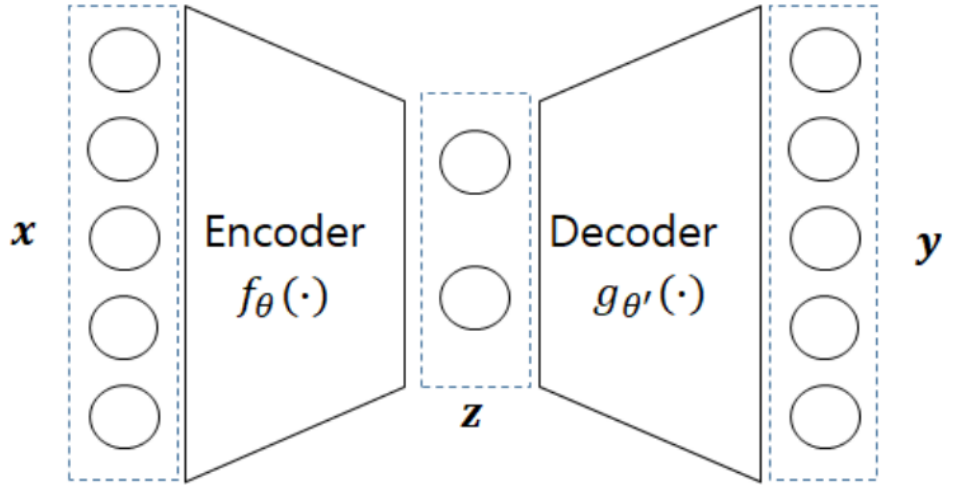


Fig. 1. Basic AutoEncoder

✓ Paper: Network Anomaly Detection Technologies Using Unsupervised Learning AutoEncoders

Auto-Encoder(계속)

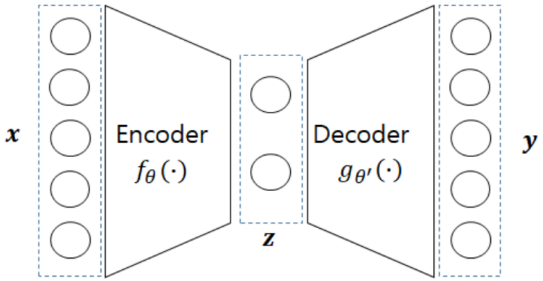
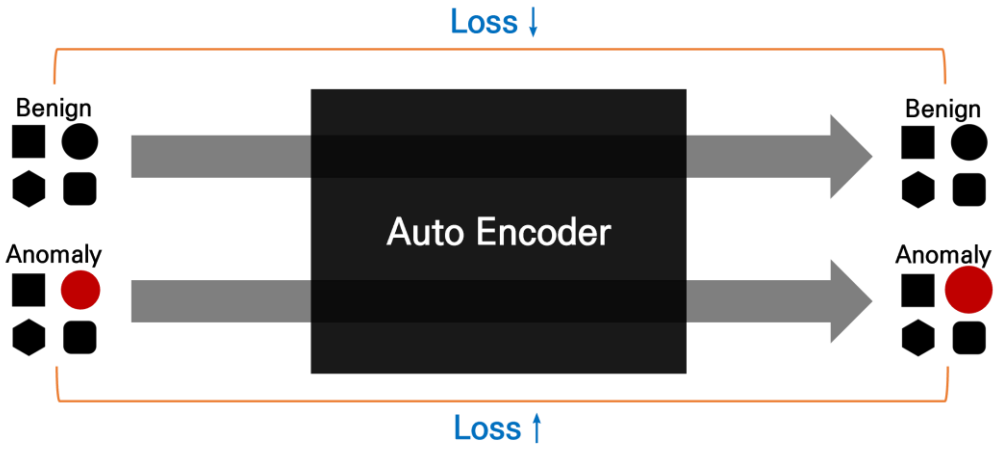


Fig. 1. Basic AutoEncoder

- 손실 함수 L : RE(Reconstruction Error; 재복원 오류)

$$L(x^{(i)}, y^{(i)}) = \sum_{j=1}^{d} \left(x_j^{(i)} - y_j^{(i)}\right)^2$$

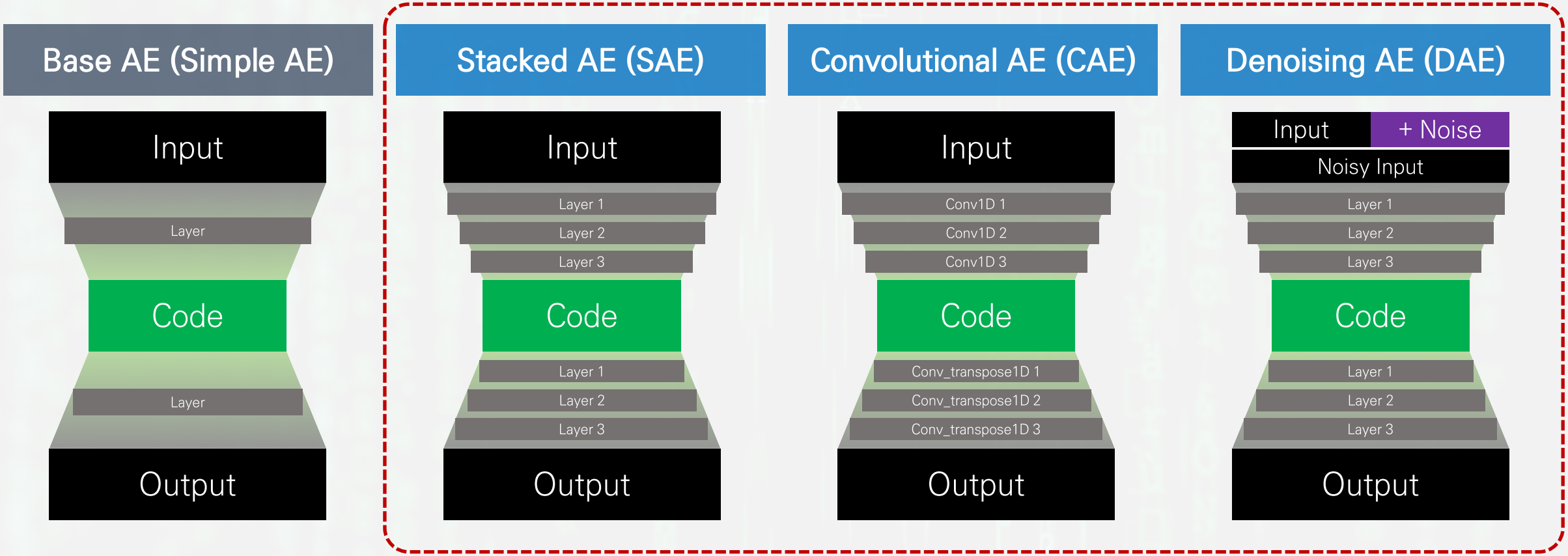
- AD 전제 : Anomaly Data의 RE는 Normal Data의 RE보다 훨씬 클 것이다.



✓ Paper: Network Anomaly Detection Technologies Using Unsupervised Learning AutoEncoders

AE Variation

- Anomaly-based AE 모델은 아래 3가지로 분화하여 학습 가능



✓ Paper: Deep Learning Model for Anomaly Detection: Survey

Threshold

- Anomaly라고 판단할 Loss Threshold를 설정하는 것이 중요함
- Train data의 Loss 값의 백분위수를 Threshold로 정하고 Test 수행

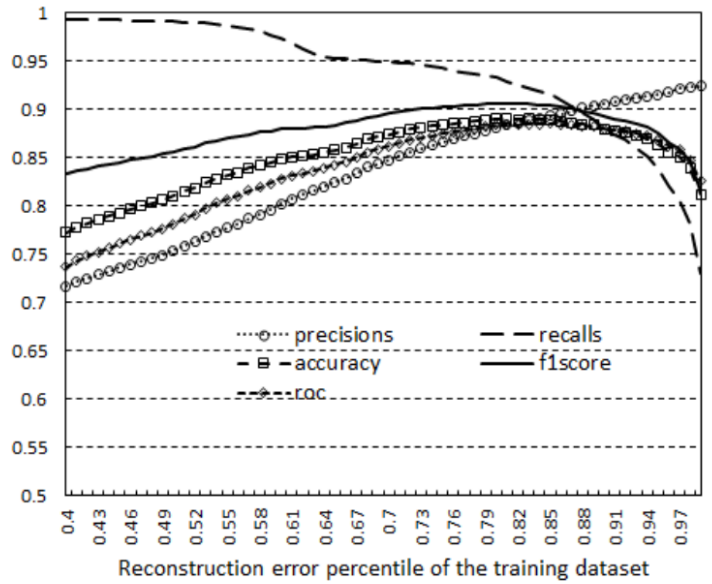


Fig. 8. Precisions, recalls, accuracies, f1-scores, and ROC AUCs on KDDTest+ with the RE percentiles of the training dataset

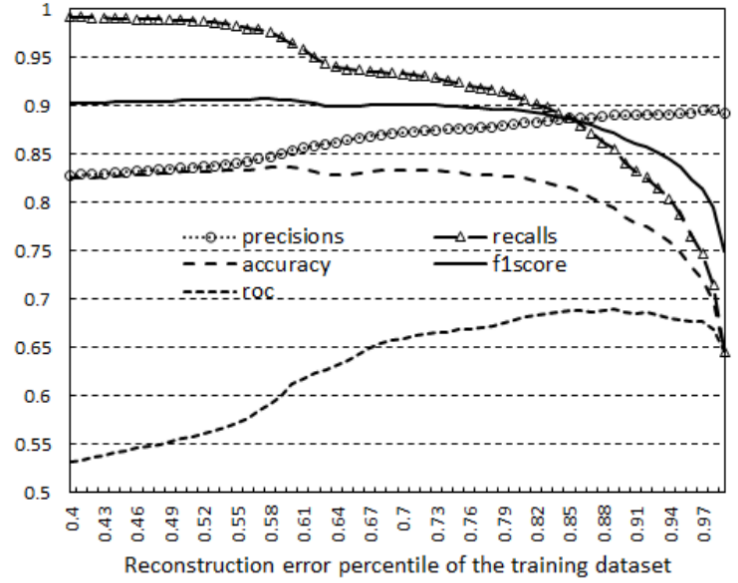


Fig. 9. Precisions, recalls, accuracies, f1-scores, and ROC AUCs on KDDTest-21 with the RE percentiles of the training dataset

✓ Paper: Network Anomaly Detection Technologies Using Unsupervised Learning AutoEncoders

3. EDA

Dataset

- **Logs of the University of New Brunswick's servers**
  - 2018. 02. 14 – 03. 02, Wed, Thu, Fri(9일) 데이터
  - 총 8,284,254건의 Flow Data, 80개의 Features

2018-02-14	2018-02-15	2018-02-16	2018-02-21	2018-02-22	2018-02-23	2018-02-28	2018-03-01	2018-03-02
1,048,575	1,048,575	1,048,575	1,048,575	1,048,575	1,048,575	613,104	331,125	1,048,575

데이터 출처 : https://www.kaggle.com/solarmainframe/ids-intrusion-csv

Dataset (계속)

- **Features**
  - Flow 단위의 데이터가 80개의 Feature를 기준으로 추출되어 있음 (시계열 데이터가 아님)

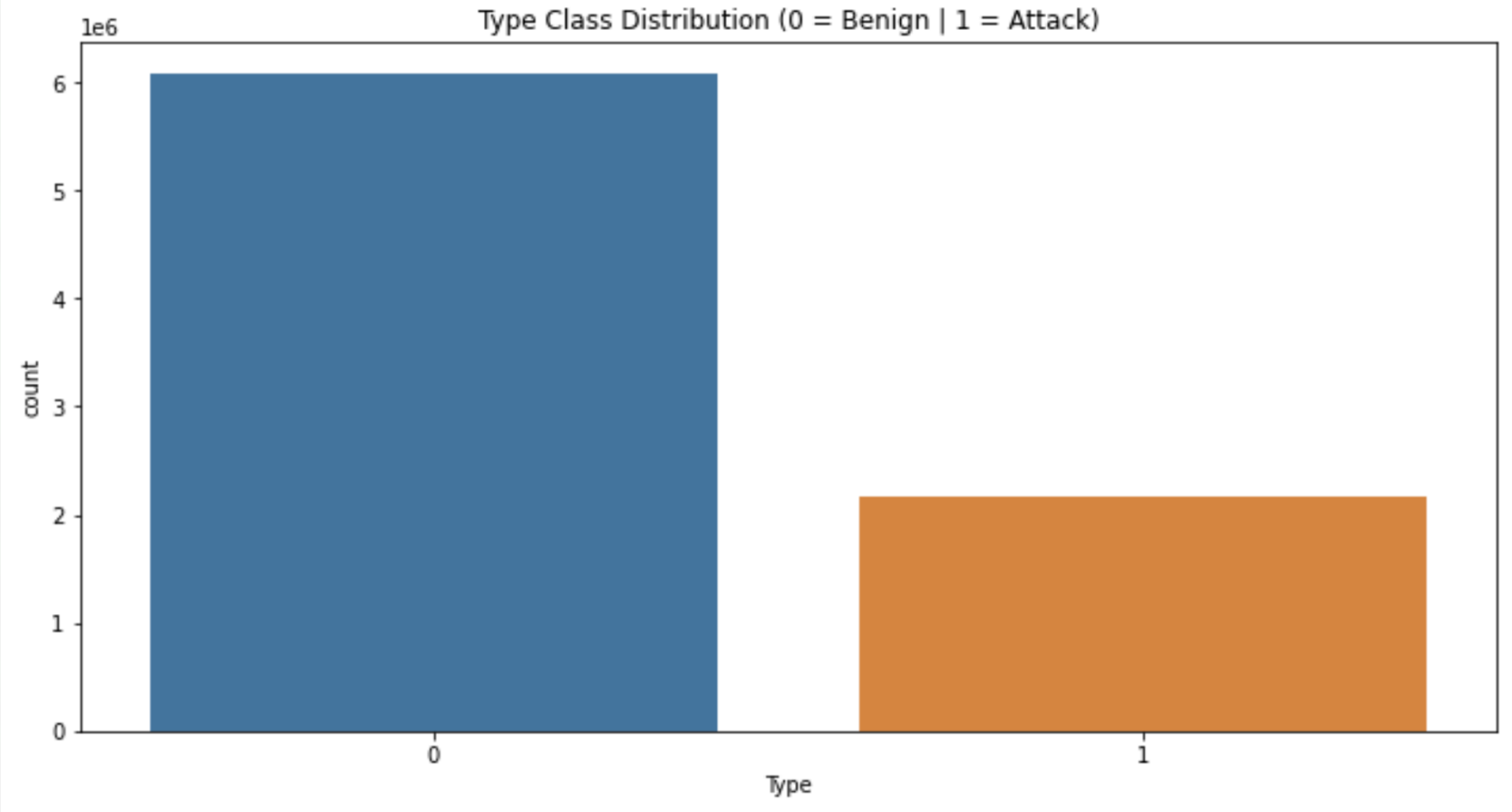
    *Network Traffic Flow Generator(CICFlowMeter)에 의해서 추출된 80개의 Feature
  - 일정 시간 동안의 Log Flow의 **총계(Total), 평균(Mean), 표준편차(Std), 최댓값(Max), 최솟값(Min)** 등이 모두 표시
  - Flow, Packet, IAT, Flag, Header, Segment, Subflow, Window, Active, Idle 등의 특성으로 구분됨
  - Flow Duration은 해당 Flow 동안의 시간으로 마이크로초(Microsecond) 단위로 표현되어 있음

	Dst Port	Protocol	Timestamp	Flow Duration	Tot Fwd Pkts	Tot Bwd Pkts	TotLen Fwd Pkts	TotLen Bwd Pkts	Fwd Pkt Len Max	Fwd Pkt Len Min	Fwd Pkt Len Mean	Fwd Pkt Len Std	Bwd Pkt Len Max	Bwd Pkt Len Min	Bwd Pkt Len Mean	Bwd Pkt Len Std	Flow Byts/s	Flow Pkts/s	Flow IAT Mean	Flow IAT Std	Flow IAT Max	Flow IAT Min	Fwd IAT Tot	Fwd I… Me…
0	0	0	16/02/2018 08:27:23	112640768	3	0	0	0	0	0	0	0	0	0	0	0	0	0.0266333	5.63e+07	138.593	56300000	56300000	113000000	5.63e+…
1	0	0	16/02/2018 08:30:12	112641773	3	0	0	0	0	0	0	0	0	0	0	0	0	0.0266331	5.63e+07	263.751	56300000	56300000	113000000	5.63e+…
2	35605	6	16/02/2018 08:26:55	20784143	23	44	2416	1344	240	64	105.043	54.5423	64	0	30.5455	32.3365	180.907	3.22361	314911	1.14595e+06	9058214	66	20700000	9404…
3	0	0	16/02/2018 08:33:01	112640836	3	0	0	0	0	0	0	0	0	0	0	0	0	0.0266333	5.63e+07	82.0244	56300000	56300000	113000000	5.63e+…
4	23	6	16/02/2018 08:27:59	20	1	1	0	0	0	0	0	0	0	0	0	0	0	100000	20	0	20	20	0	

데이터셋 예시

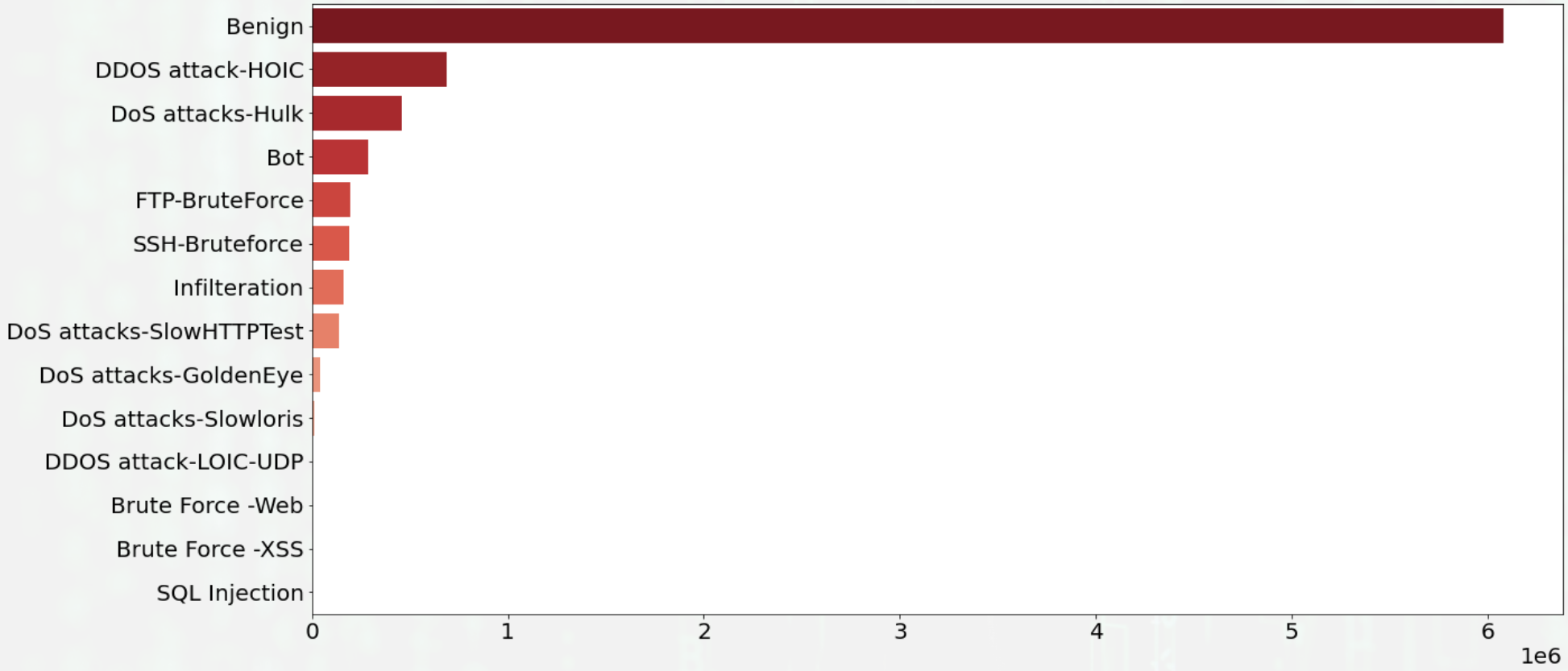
Label

- Dataset의 정상(Benign)과 비정상(Attack)의 비율이 약 3:1로 나타남



Label

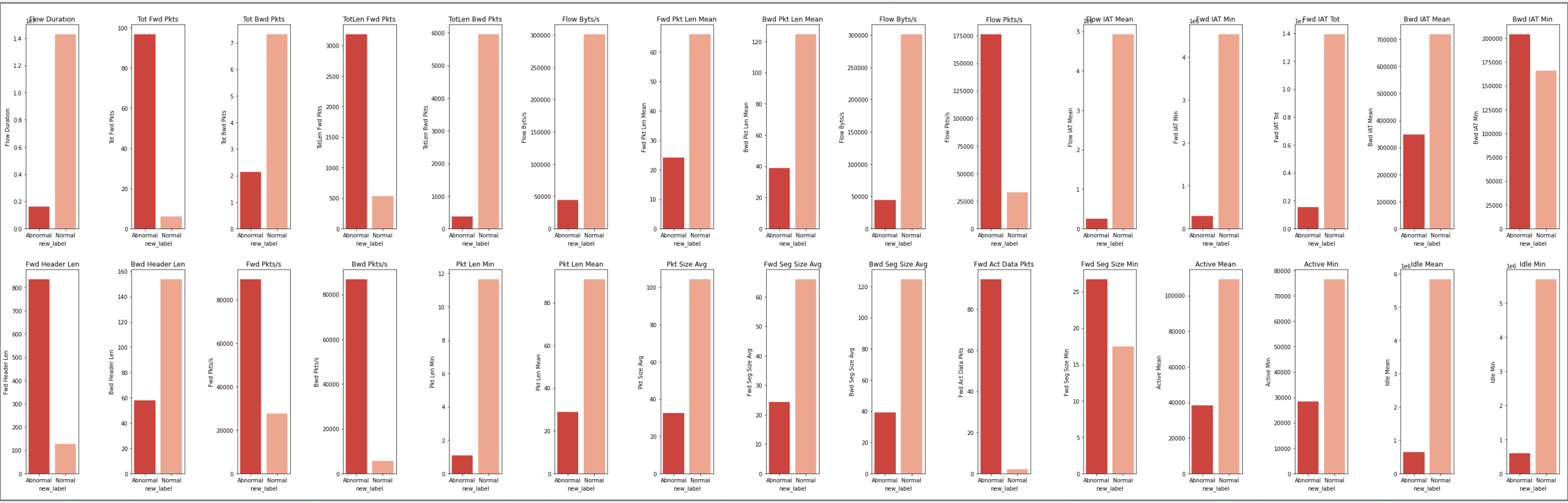
- **총 13개(Benign 포함, 12개의 공격 유형)의 Label로 나누어 제시**
  - Infilteration Label은 별도로 두어, 구분되지 않는 공격 유형은 따로 분류함



Benign Vs. Attack 비교

- **대부분의 특성에서 Benign과 Attack 사이의 차이가 크게 나타남 (Mean)**

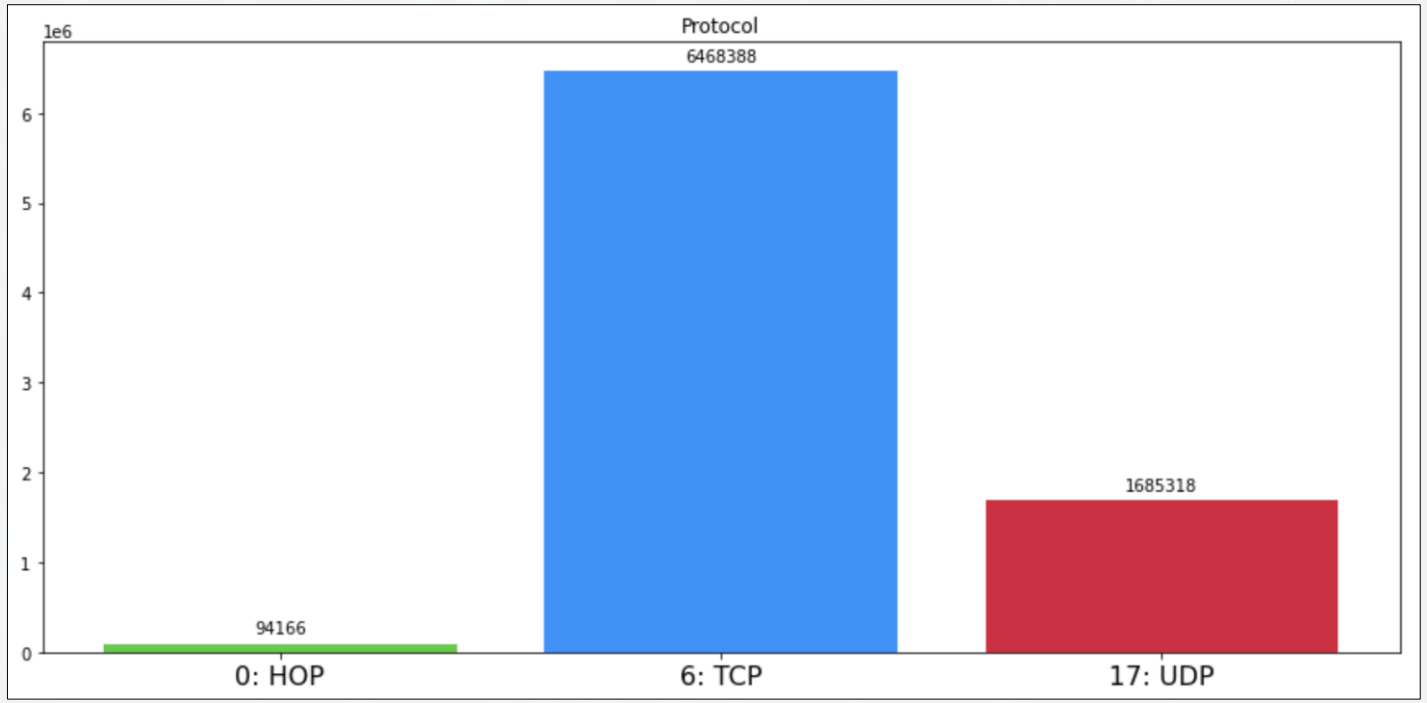
Benign, Attack의 주요 30개 특성 평균(Mean)값 비교



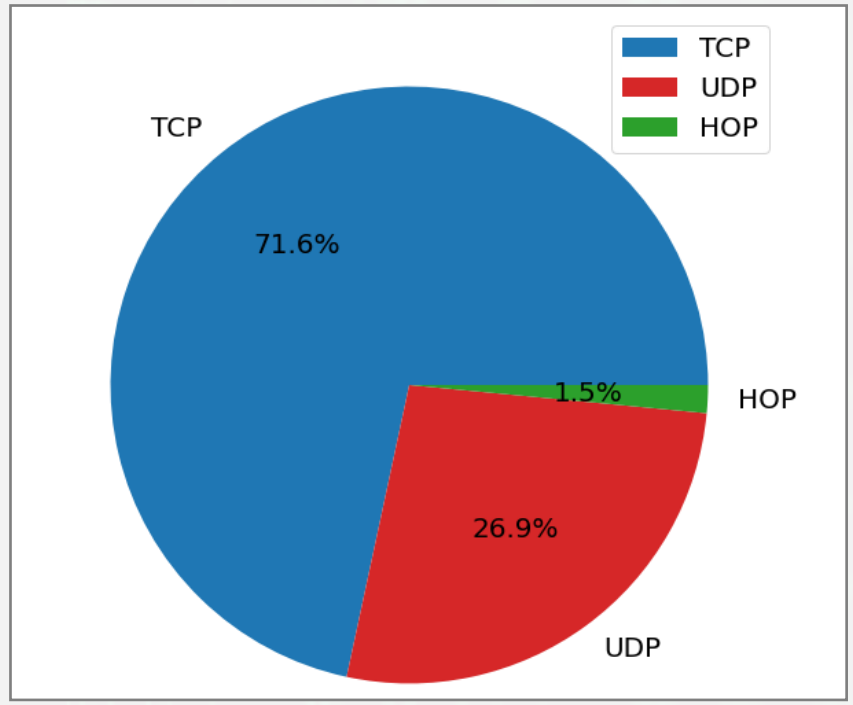
Protocol

- 총 3개의 Protocol이 나타나고 있으며, TCP가 가장 많음
- TCP에서 대부분의 공격이 나타나고 있음

Protocol별 전체 데이터 Count (0 : HOP / 6 : TCP / 17 : UDP)

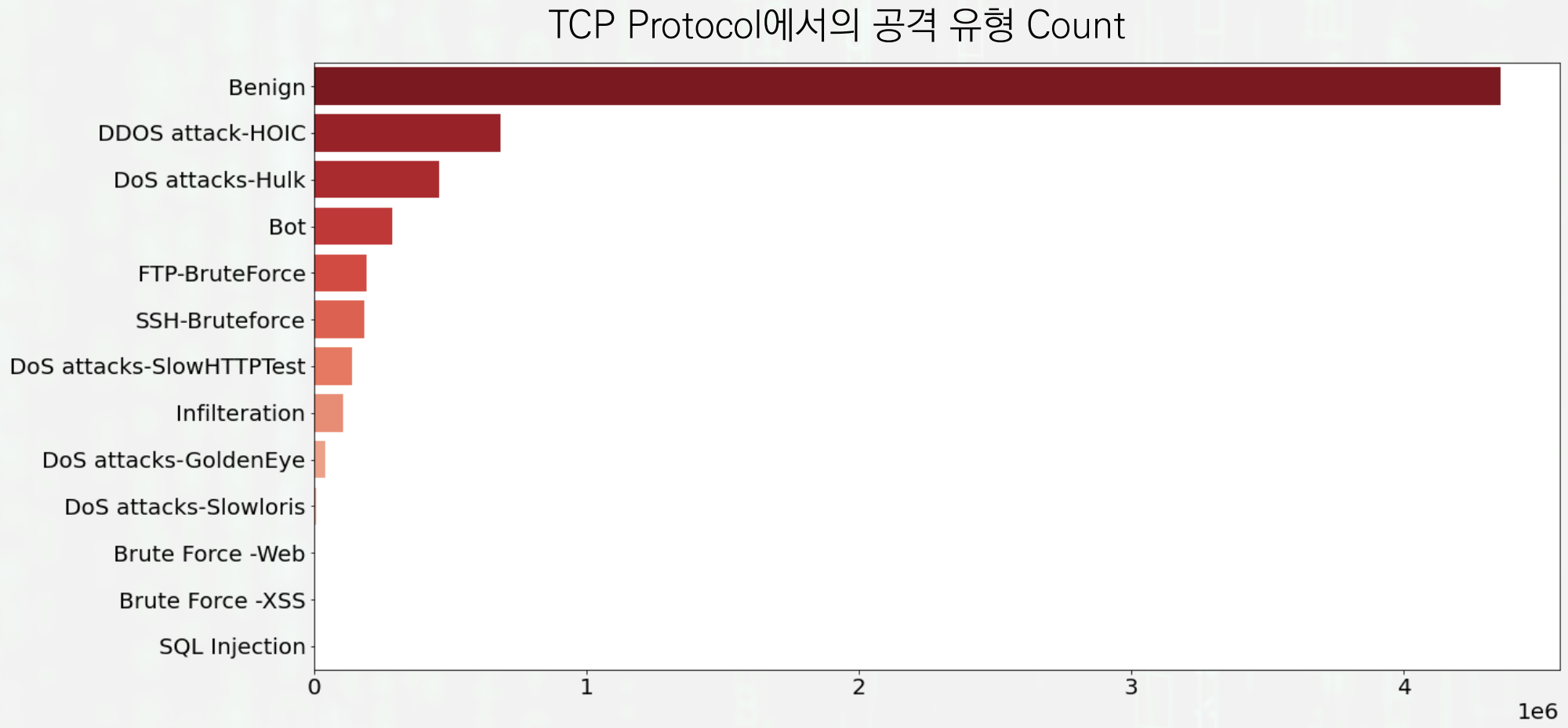


Protocol별 공격 Count



Protocol별 공격 형태 – TCP

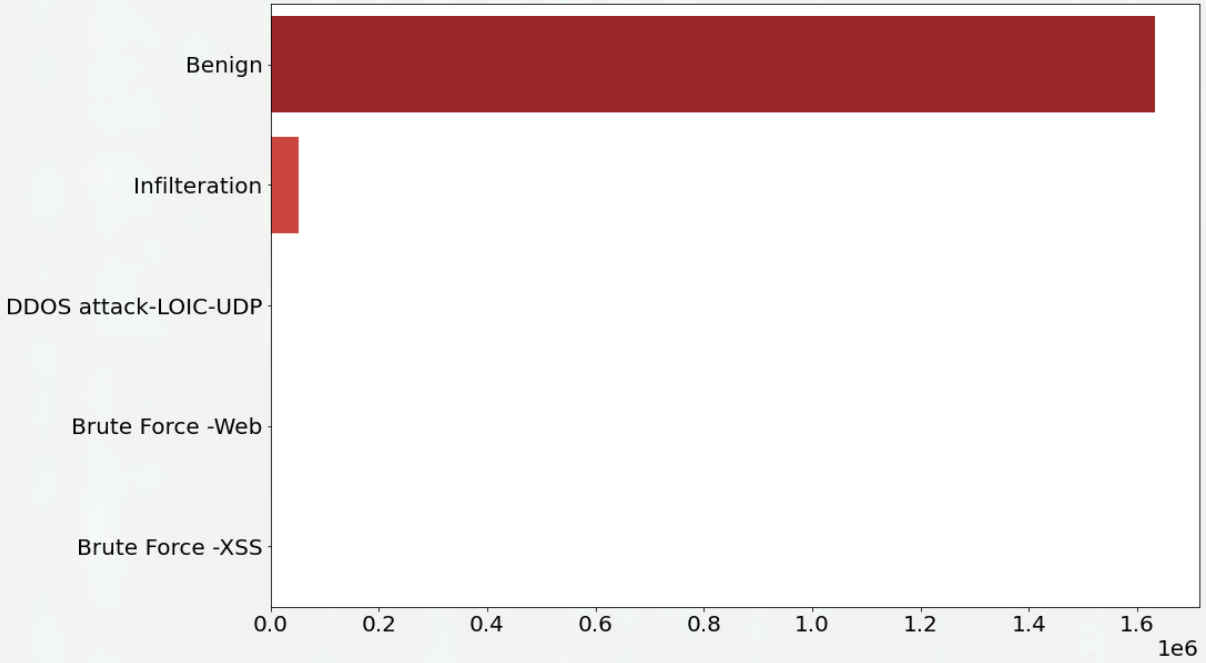
- TCP에서는 거의 모든 공격 형태가 나타나고 있음



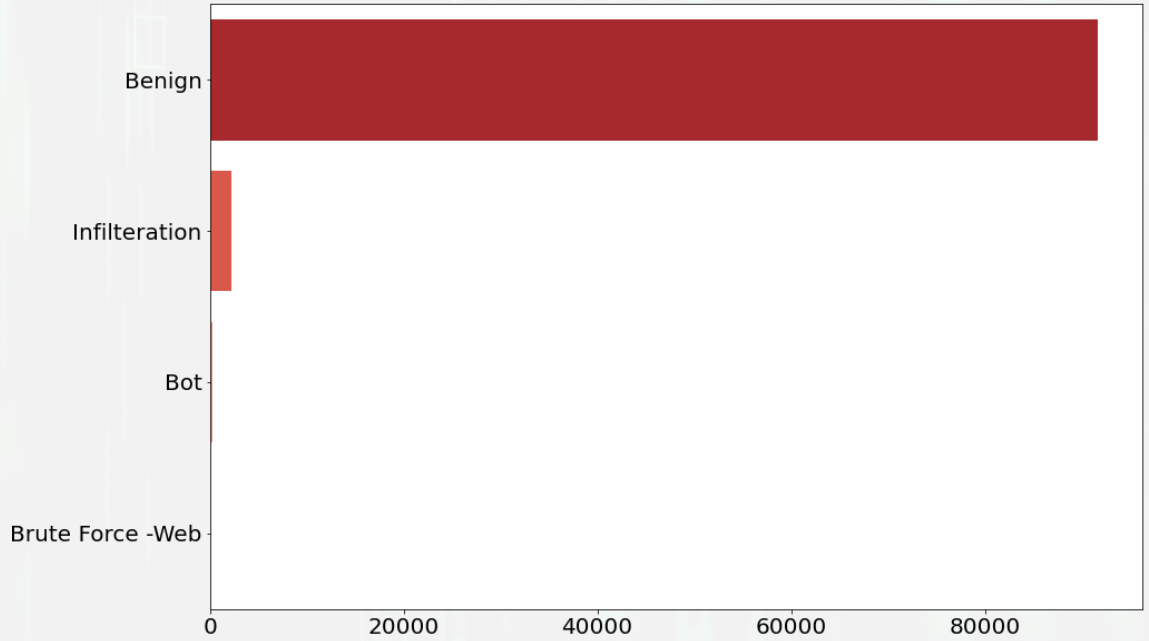
Protocol별 공격 형태 – UDP / HOP

- **반면 UDP, HOP에서는 극히 일부의 공격만 나타나고 있음**
  - UDP(4) : DDOS attack-LOIC-UDP, Brute Force –Web, Brute Force –XSS, Infilteration
  - HOP(3) : Bot, Brute Force –Web, Infilteration

UDP Protocol에서의 공격 유형 Count



HOP Protocol에서의 공격 유형 Count

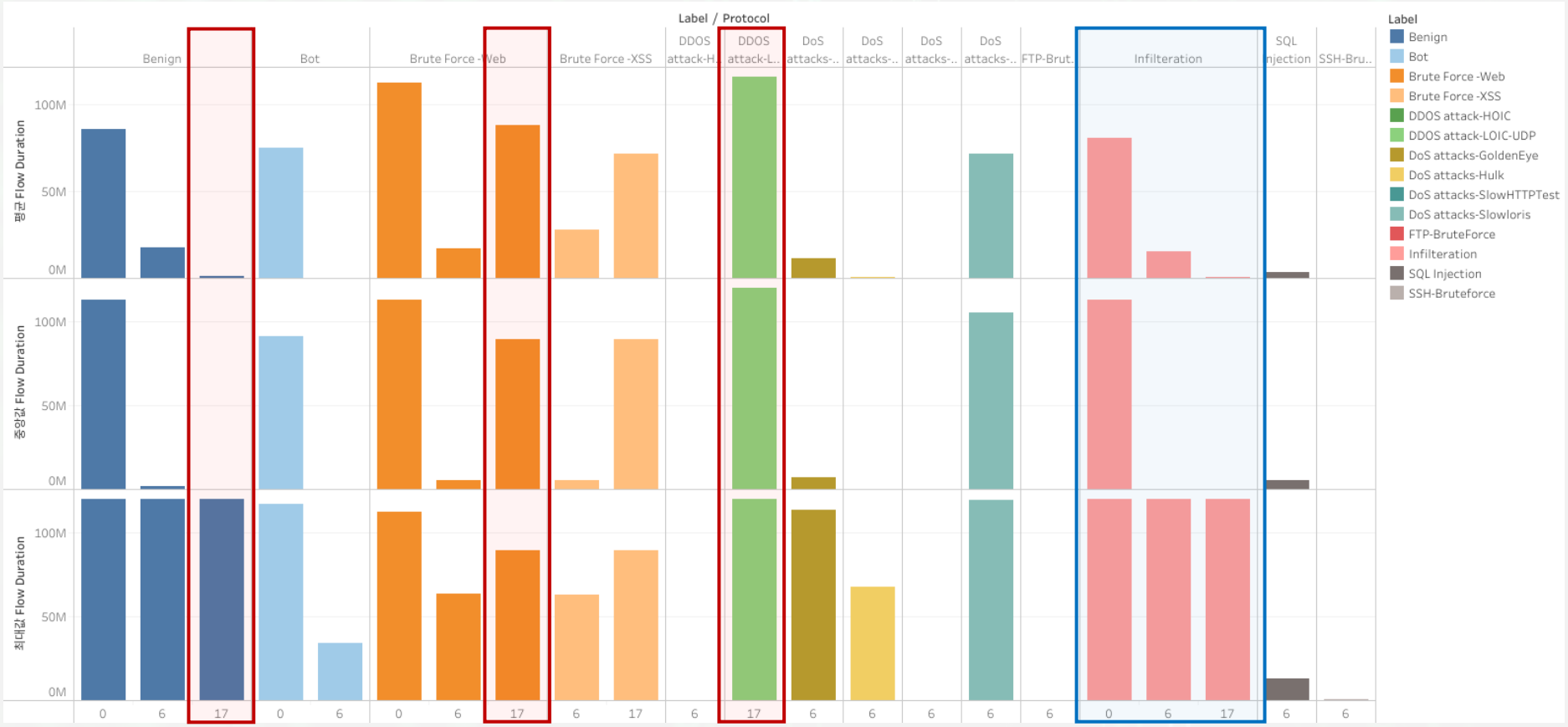


Protocol별 주요 Feature 비교 – Init Fwd Win Byts

- TCP(6)에서 Label 간 특성 차이가 크게 나타나지만, HOP(0), UDP(17)에서는 거의 변화가 없음

Protocol별 주요 Feature 비교 – Flow Duration

- UDP(17)에서 Benign과 특정 Attack을 구분할 수 있는 유의미한 특성으로서 역할을 함
- Benign과 Infilteration의 형태가 매우 유사하게 나타나고 있음

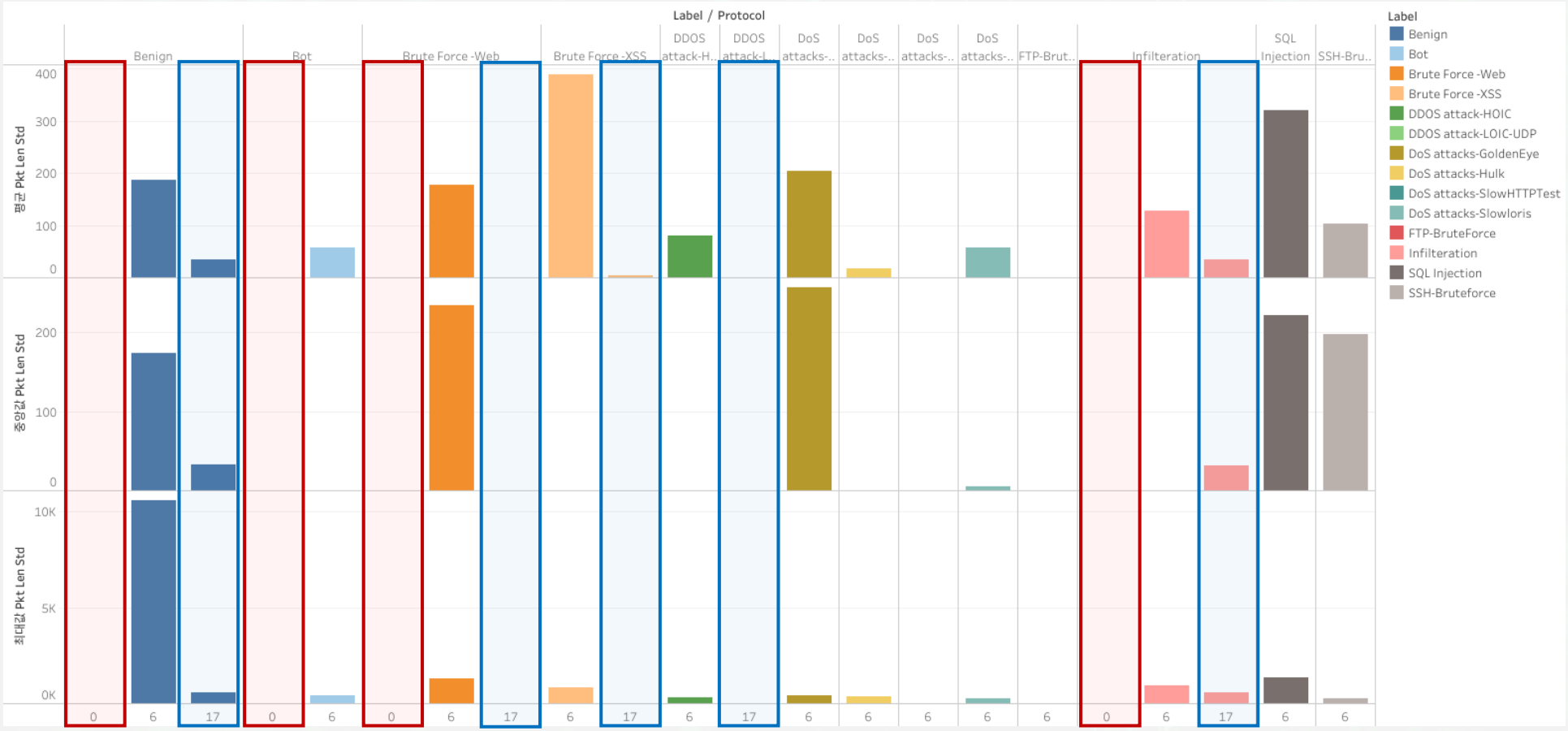


Protocol별 주요 Feature 비교 – Flow IAT Min

- TCP(6)에서 Benign과 일부 Attack 간의 차이가 거의 없음

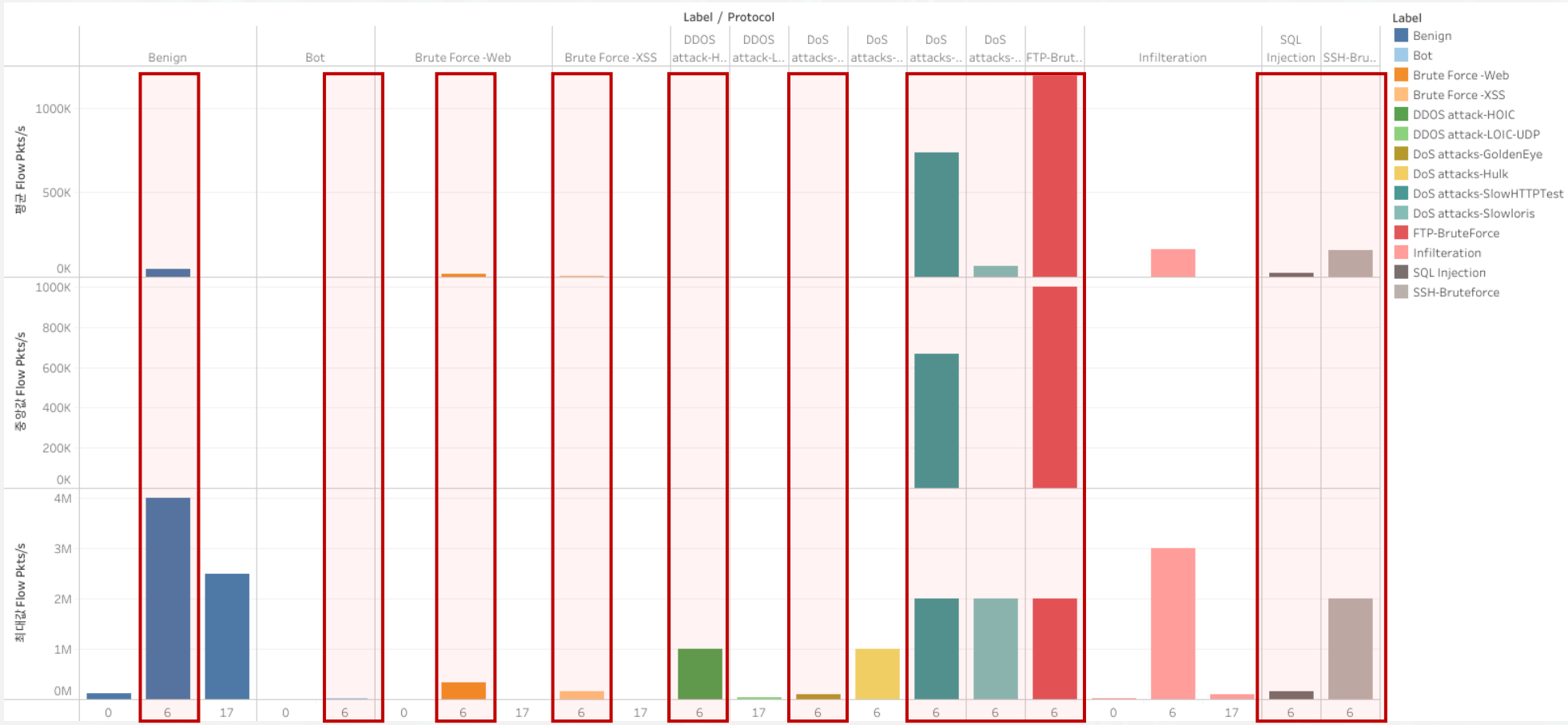
Protocol별 주요 Feature 비교 – Pkt Len Std

- HOP(0번), UDP(17번)에서 Benign과 Attack 사이의 차이가 크지 않음



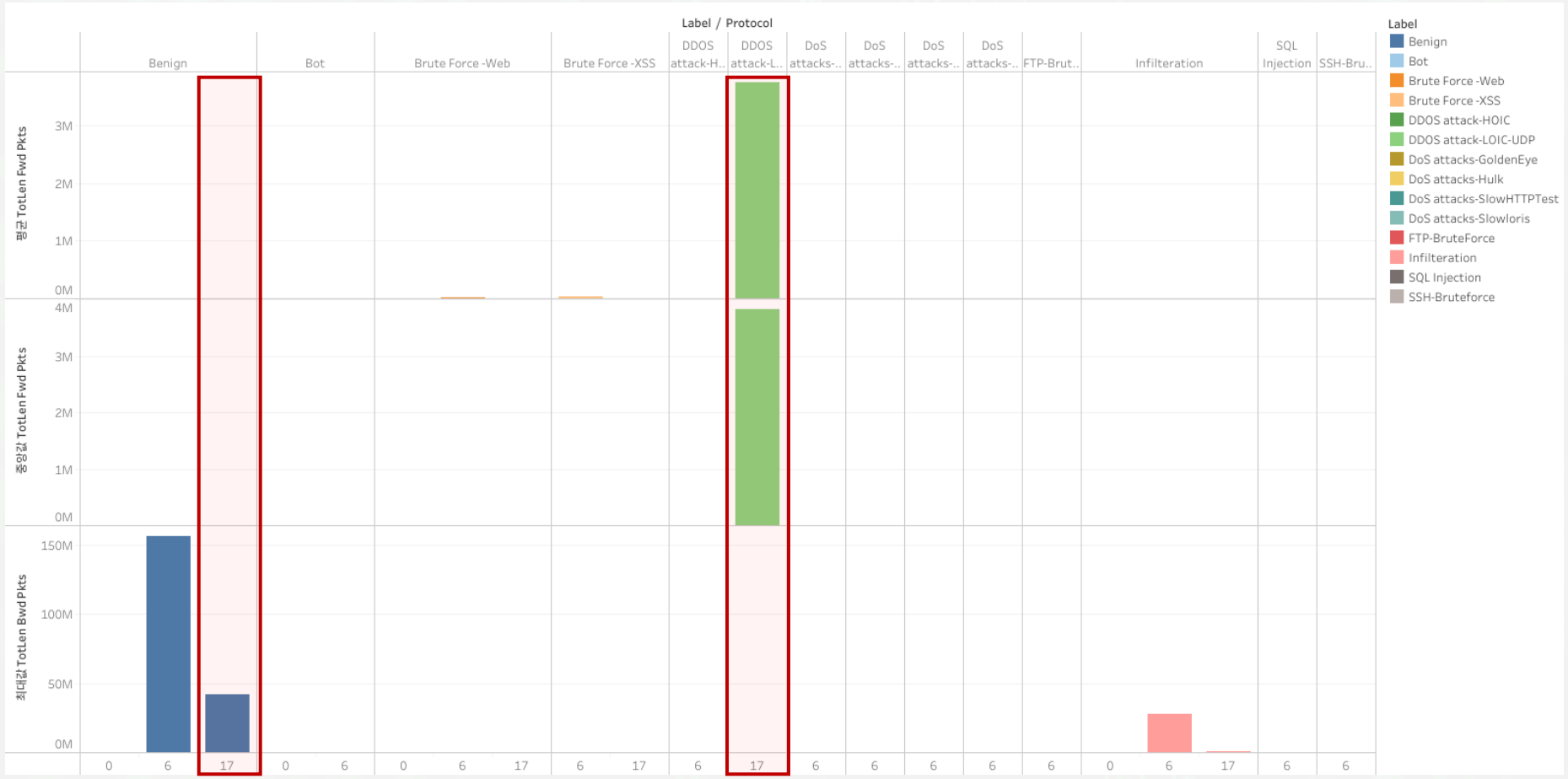
Protocol별 주요 Feature 비교 – Flow Pkts/s

- TCP(6)에서 Benign과 특정 Attack을 구분할 수 있는 유의미한 특성으로서 역할을 함



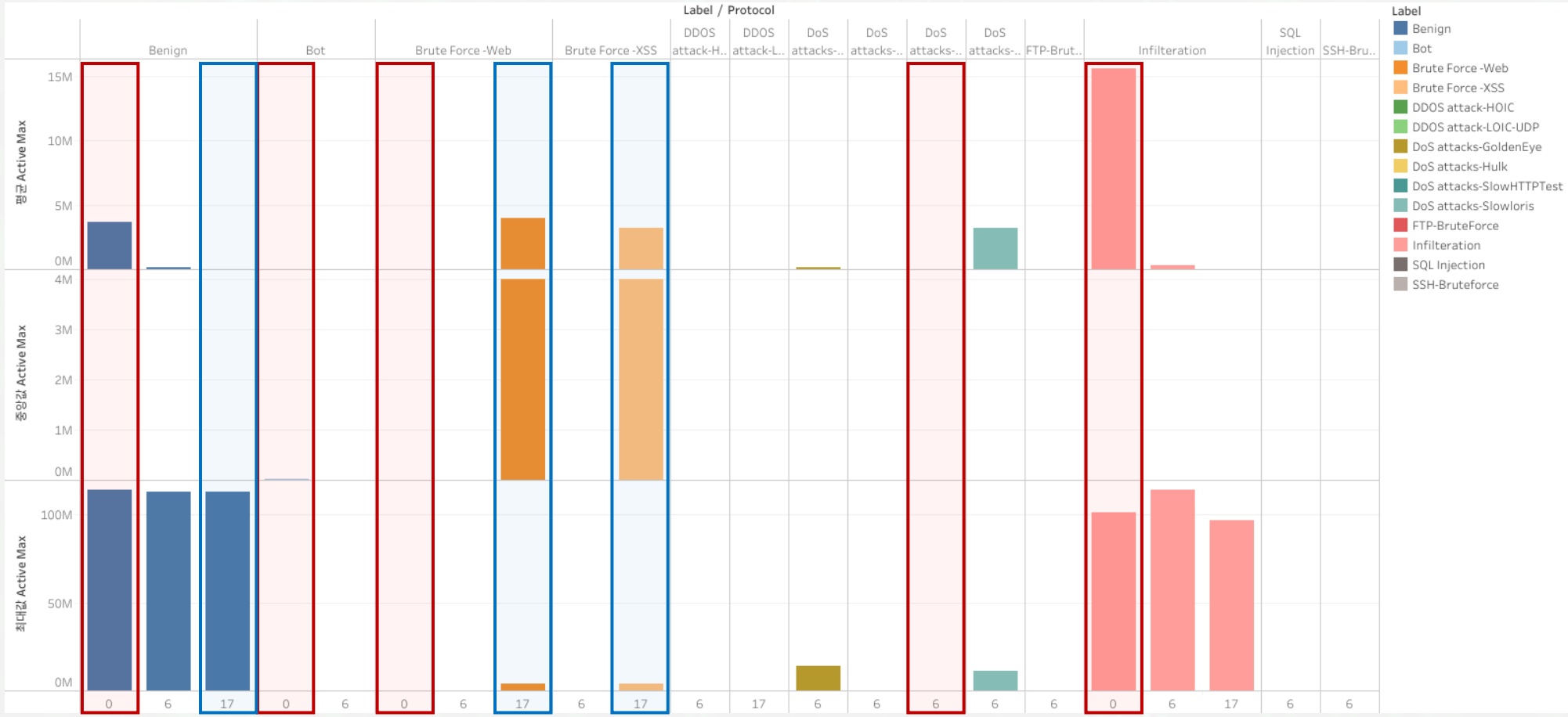
Protocol별 주요 Feature 비교 – TotLen Fwd Pkts

- UDP(17)에서 Benign과 특정 Attack을 구분할 수 있는 유의미한 특성으로서 역할을 함



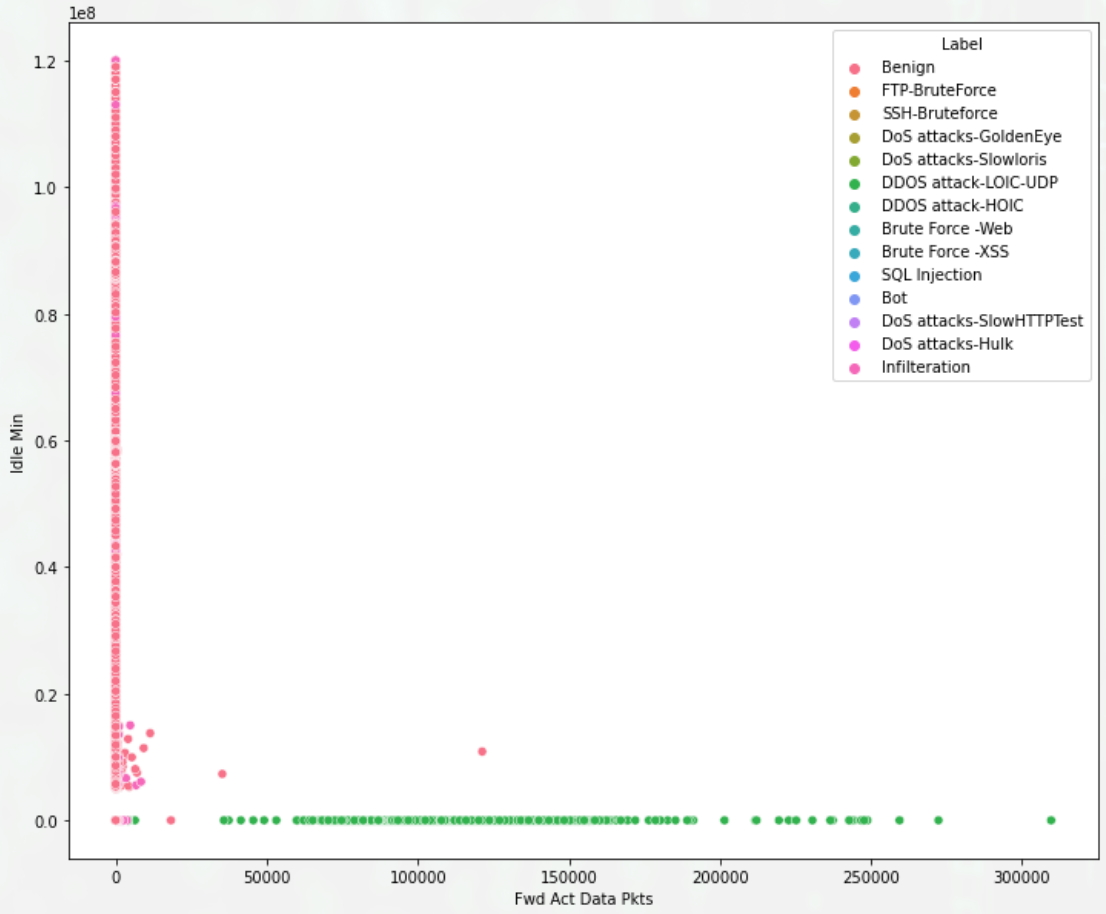
Protocol별 주요 Feature 비교 – Active Max

- HOP(0), UDP(17)에서 Benign과 특정 Attack을 구분할 수 있는 유의미한 특성으로서 역할을 함



2-Feature Correlation

- Fwd Act Data Pkts - Idle Min

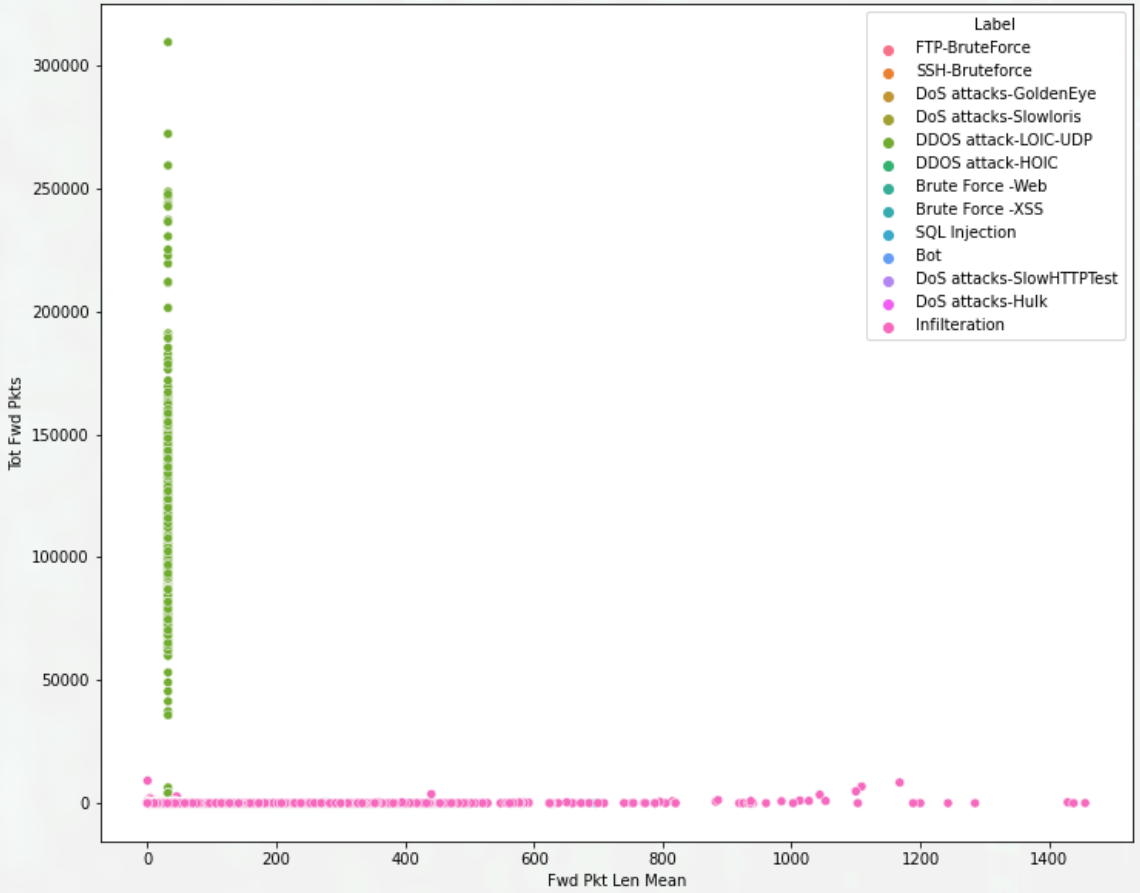


✓ Benign 데이터는 Idle Min 값이 다양하고, Fwd Act Data Pkts 값이 0 부근에서 나타나고 있음

✓ DDoS attack-LOIC-UDP 공격의 Idle Min 값이 0으로 나타남

2-Feature Correlation (계속)

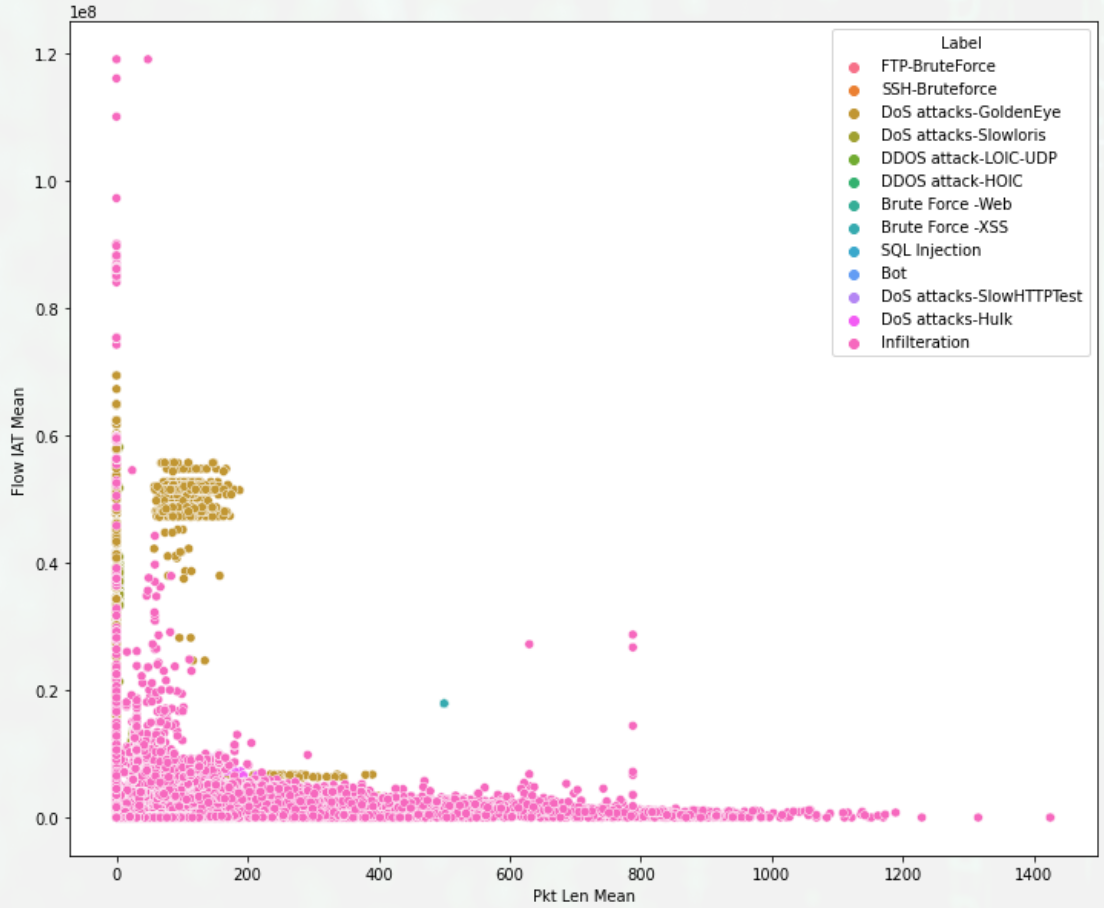
- Fwd Pkt Len Mean - Tot Fwd Pkts



- ✓ DDoS attack-LOIC-UDP 공격은 Fwd Pkt Len Mean 값이 매우 작은 값으로 나타남
- ✓ Infilteration(기타 분류되지 않은 공격)의 Tot Fwd Pkts는 대체로 0에서 발생함

2-Feature Correlation (계속)

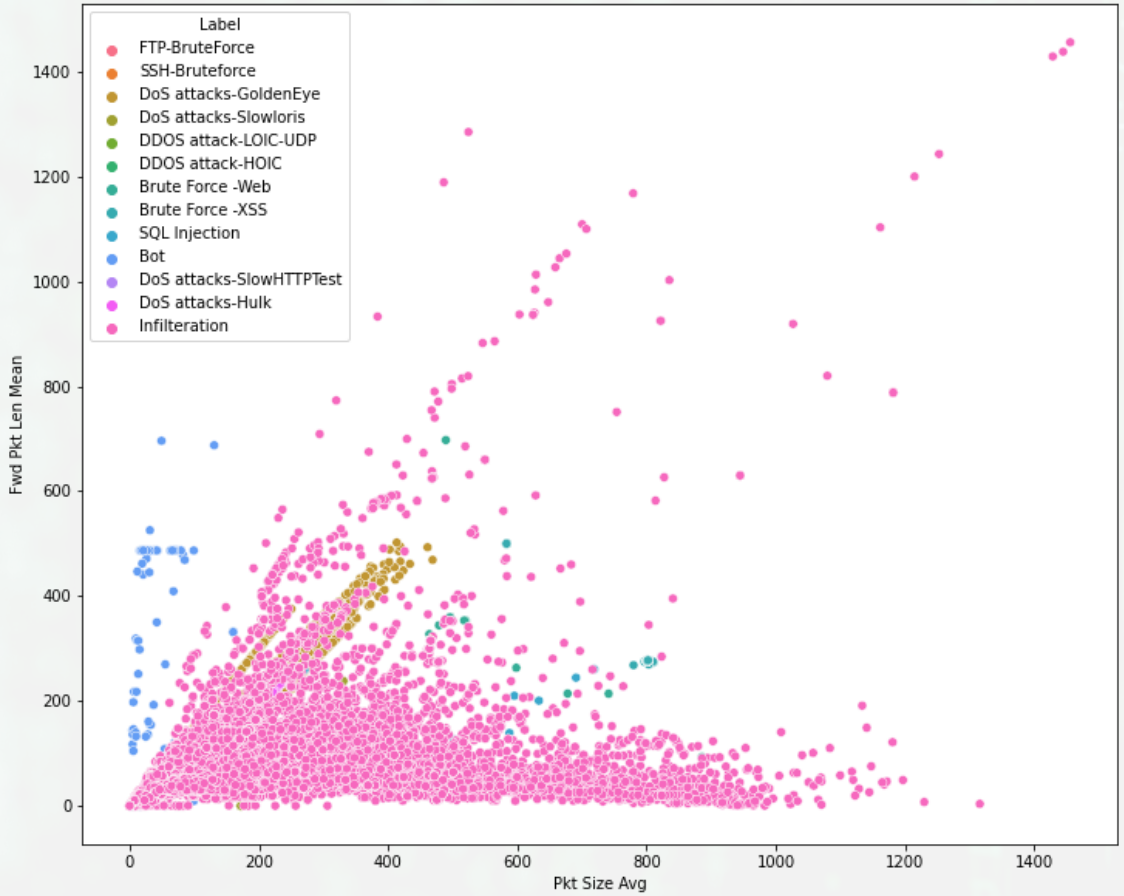
- Pkt Len Mean - Flow IAT Mean



- ✓ DoS attacks-GoldenEye 공격이 다수 발생하는 클러스터 구간이 형성됨
- ✓ Infilteration은 두 값 모두 다양하게 나타남

2-Feature Correlation (계속)

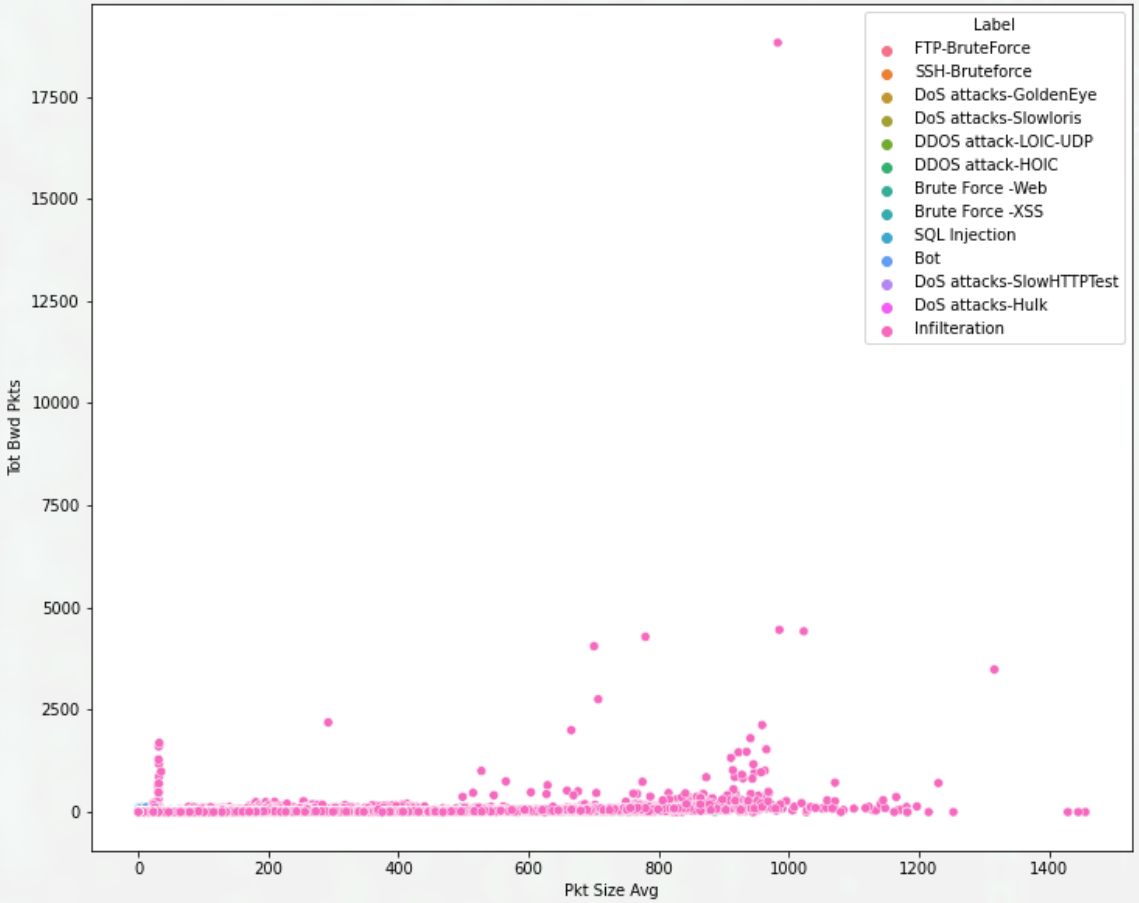
- Pkt Size Avg - Fwd Pkt Len Mean



- ✓ Bot 공격은 Pkt Size Avg 값이 0~ 200, Fwd Pkt Len Mean 값이 0 ~ 600에서 발생함
- ✓ Brute Force 공격 유형은 두 값 모두 크게 나타나는 경향이 있음
- ✓ Infilteration은 구분이 쉽지 않음

2-Feature Correlation (계속)

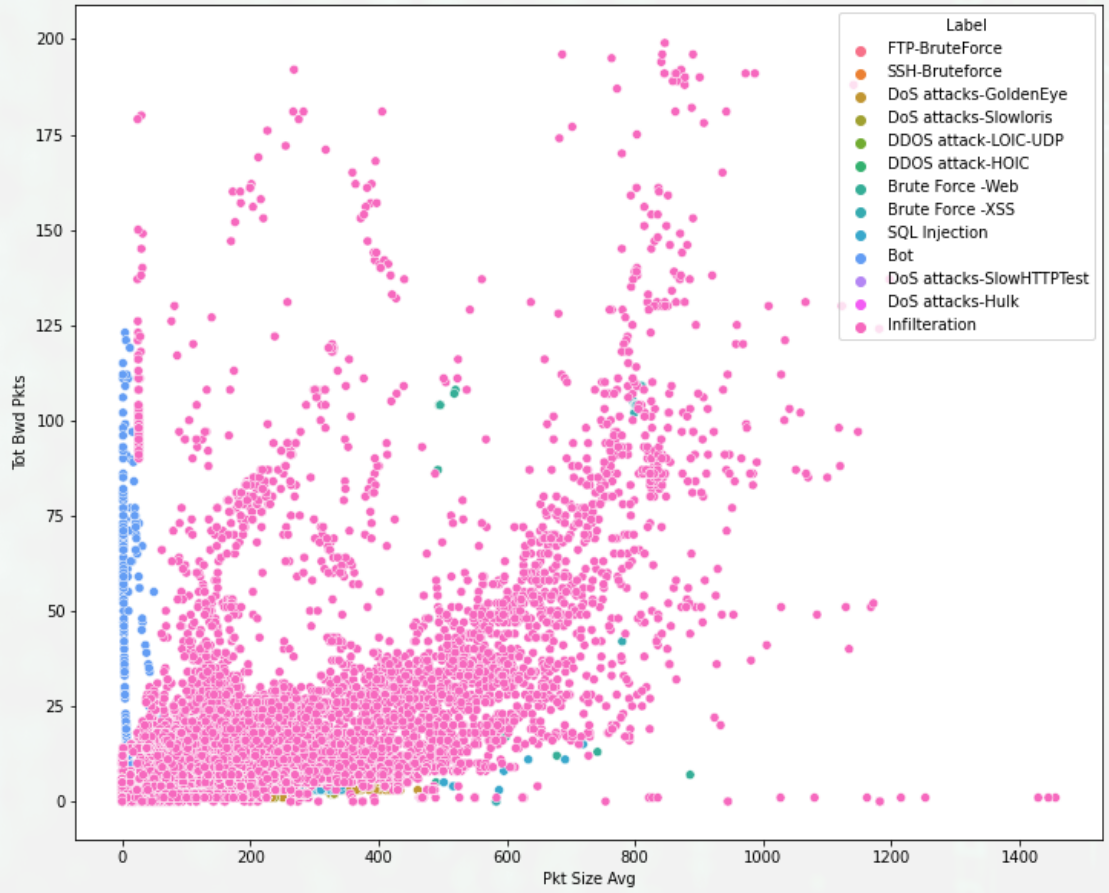
- Pkt Size Avg (전체 범위) – Tot Bwd Pkts



✓ Infiteration에 Tot Bwd Pkts 값이 매우 크게 나타나는 Outlier가 있음

2-Feature Correlation (계속)

- Pkt Size Avg (200 이하) – Tot Bwd Pkts

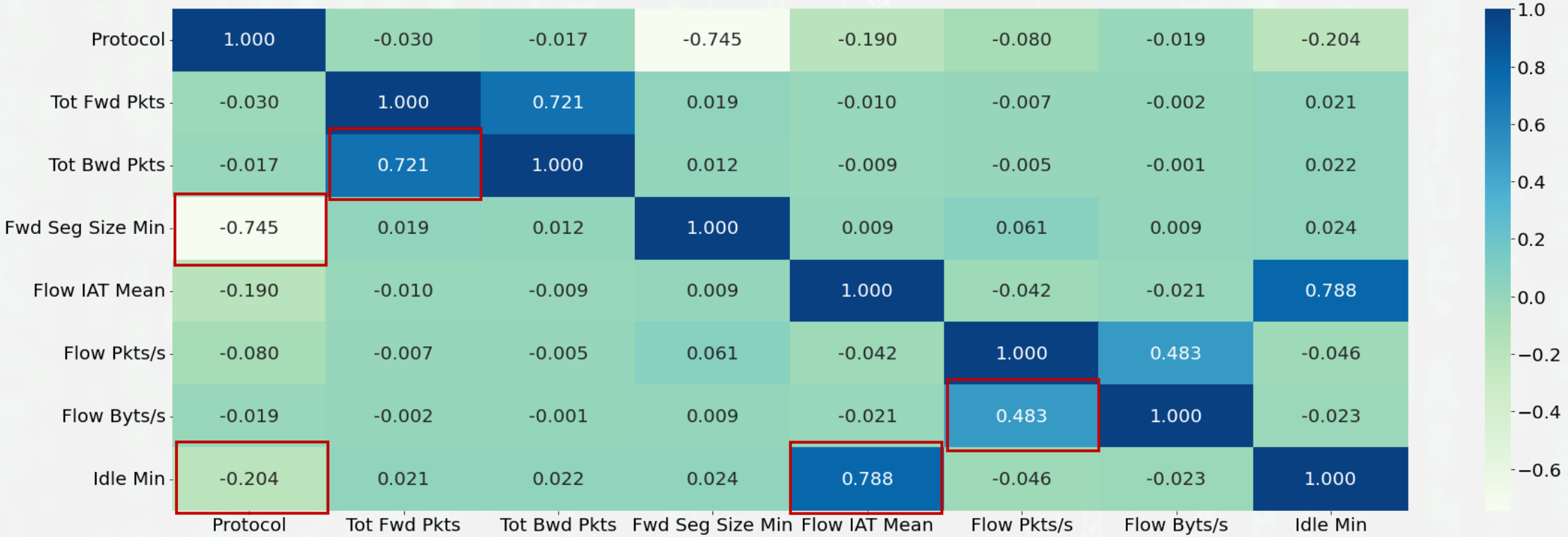


✓ Bot 공격은 **특정 Pkt Size Avg**에서 **다양한** Tot Bwd Pkts 값을 보이고 있음

✓ Infiteration은 대체적으로 Pkt Size Avg 값과 Tot Bwd Pkt 값이 비례하는 것으로 보임

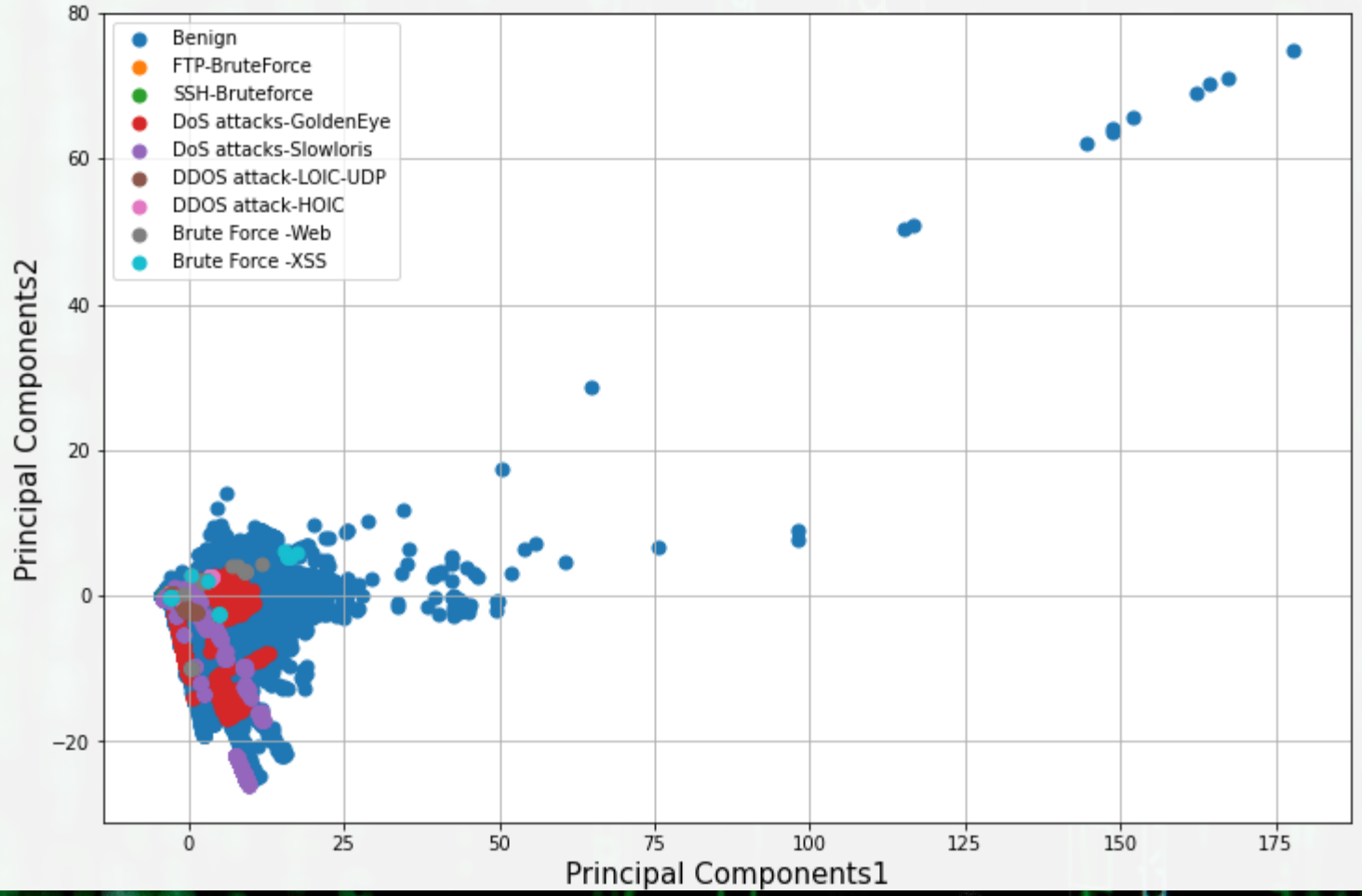
Benign Heatmap

- **Important Feature Correlation**
  - (Protocol – Fwd Seg Size Min, Idle Min), (Tot Fwd Pkts – Tot Bwd Pkts), (Flow IAT Mean – Idle Min), (Flow Bytes – Flow Pkts/s)이 상관관계가 크게 나타나고 있음



Visualization with Dimension Reduction

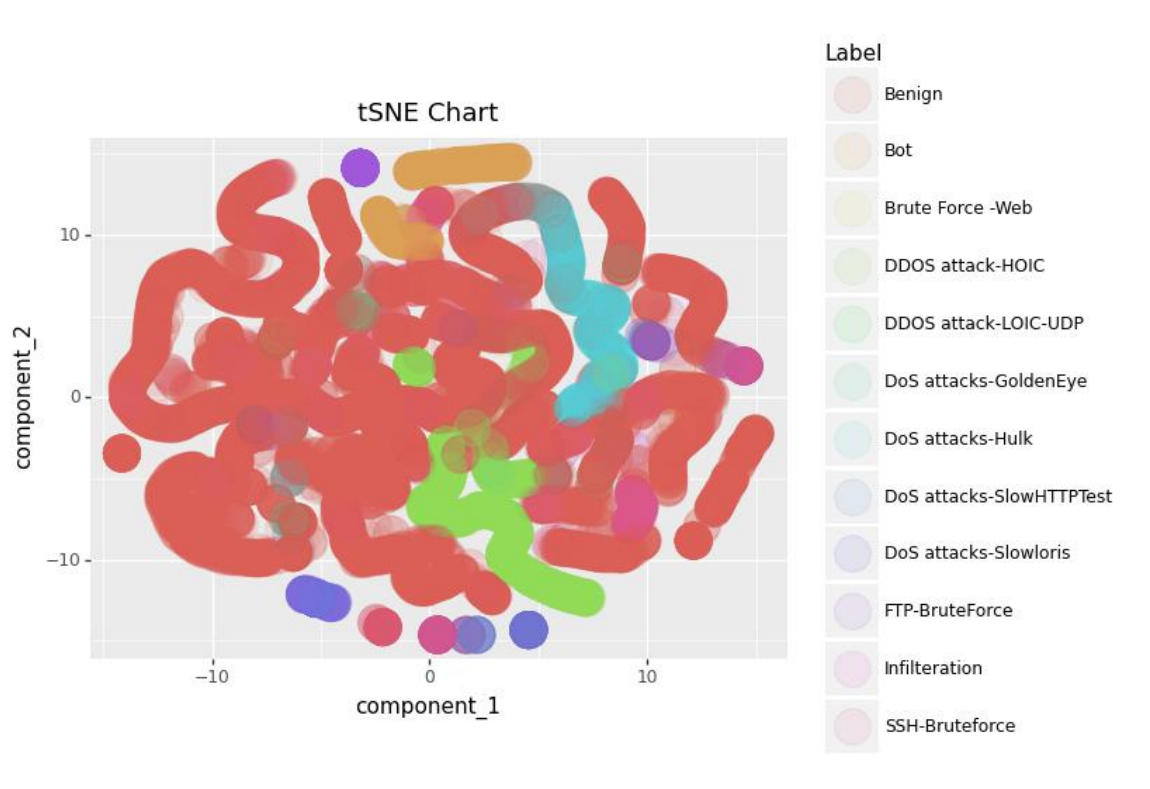
- PCA : 50,000 Sample, n_components=2



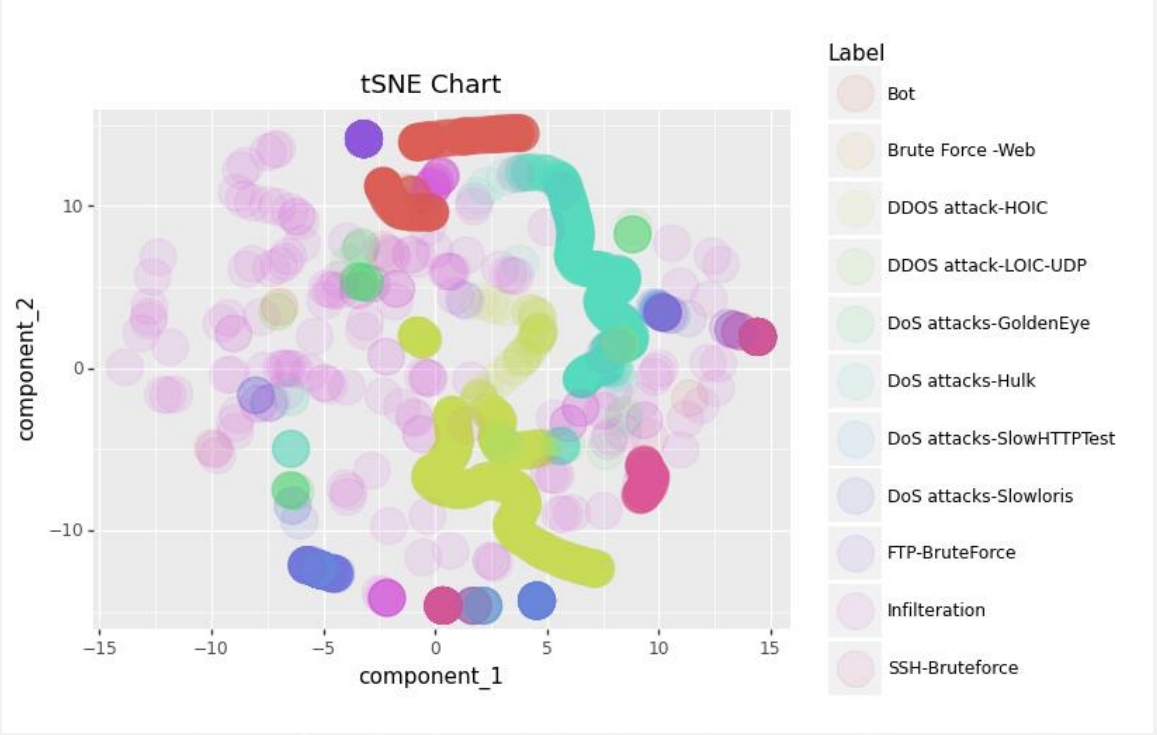
Visualization with Dimension Reduction (계속)

- t-SNE : 10,000 Sample, n_components=2

Benign 포함



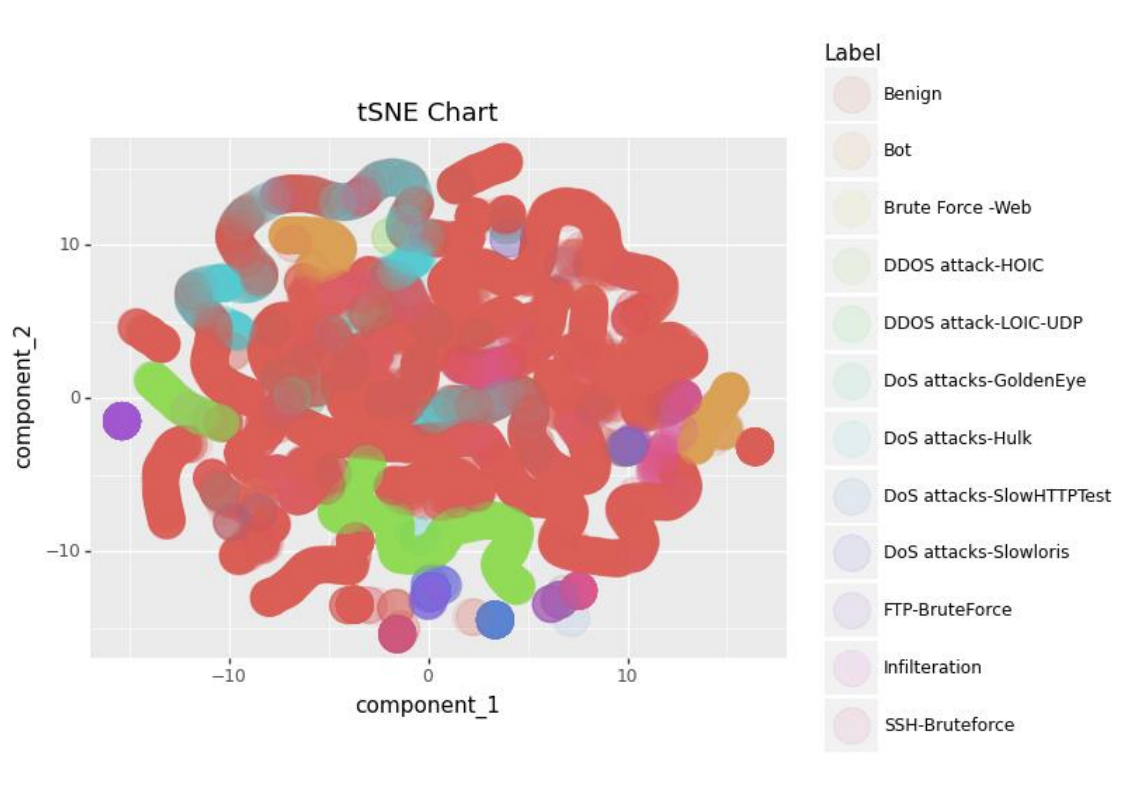
Benign 제외



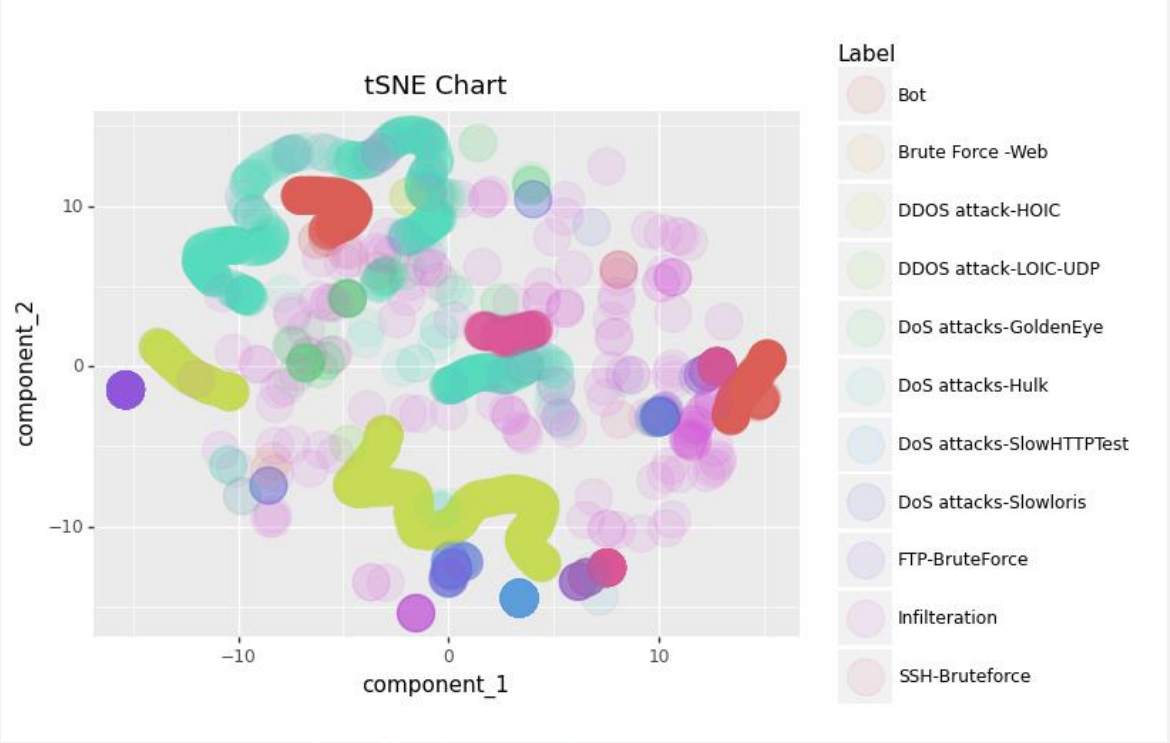
Visualization with Dimension Reduction (계속)

- t-SNE : 10,000 Sample, n_components=2, Feature Selection by SelectFromModel(RF)

Benign 포함

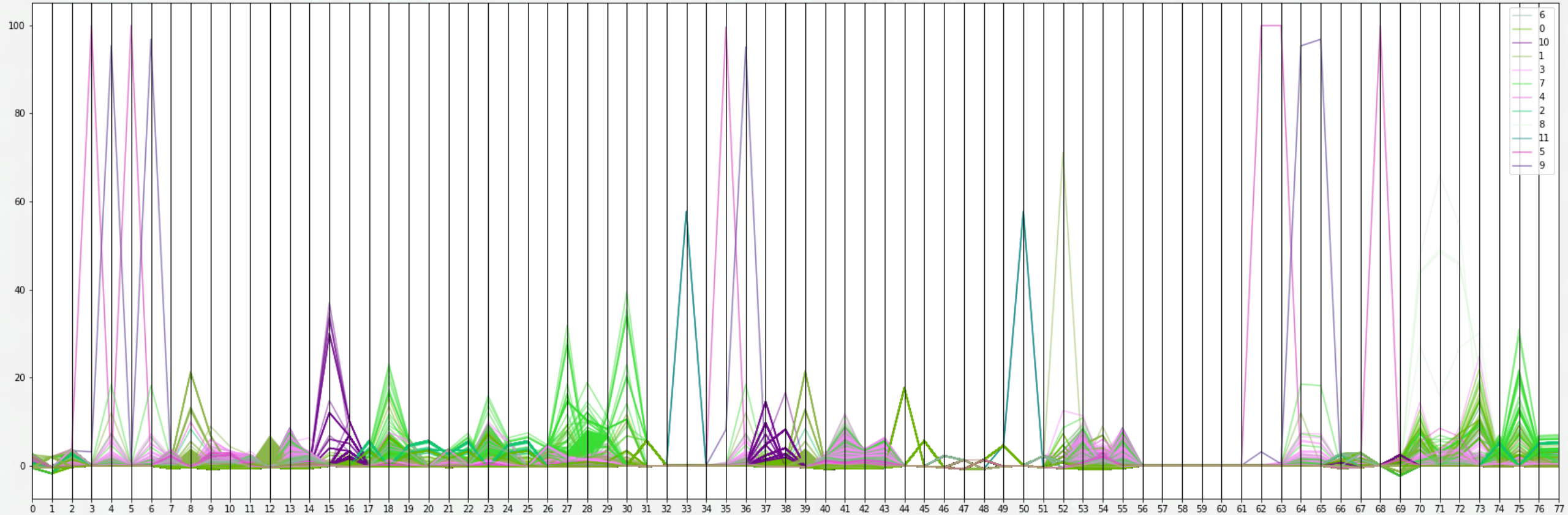


Benign 제외



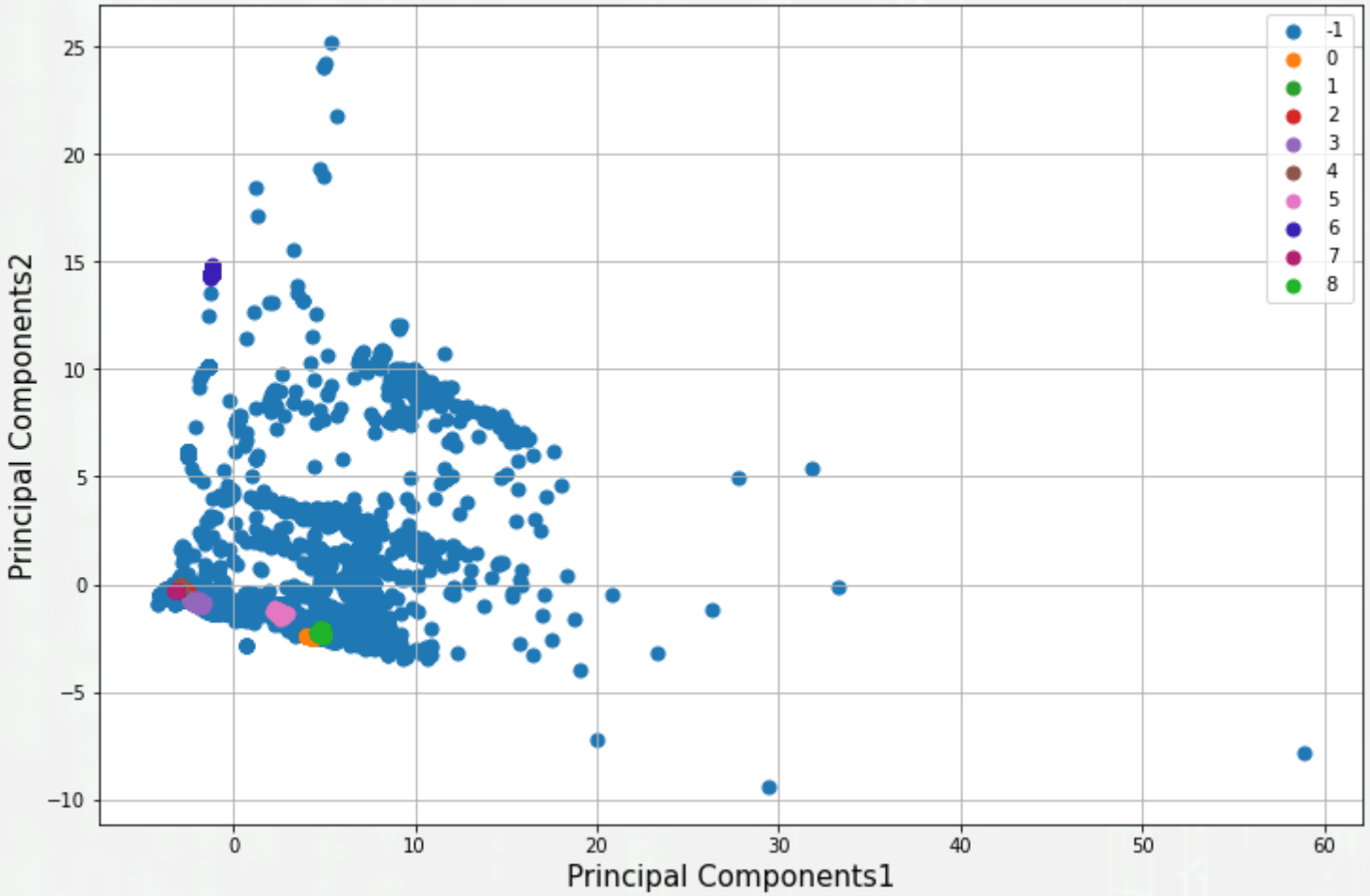
Visualization with Clustering

- K-means : 10,000 Sample, n_clusters=12, silhouette_score=0.409363



Visualization with Clustering

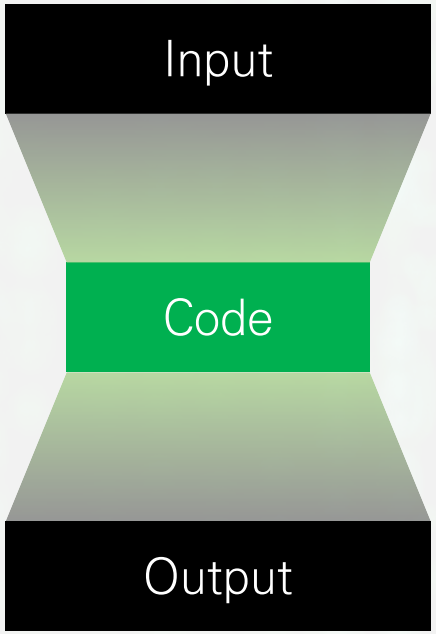
- DBSCAN : 10,000 Sample, min_samples=200, silhouette_score=0.231785



4. Modeling

모델 제안

- AutoEncoder



Type	Base AE (layer=1) / Stacked AE (layer=4) / Denoising AE (layer=4)
Optimizer	Adam
Loss	Mean Squared Error(MSE)
Activation	Relu
Output Activation	Sigmoid
Learning Rate	0.0008

모델 제안 (계속)

- TCP Model
  - Protocol별 데이터 특성 차이가 큼 → TCP Protocol 데이터 학습 모델 제시

T C P

Base AE	Stacked AE	Denoising AE
(layer=1)	(layer=4)	(Noise, layer=4)

모델 평가 결과

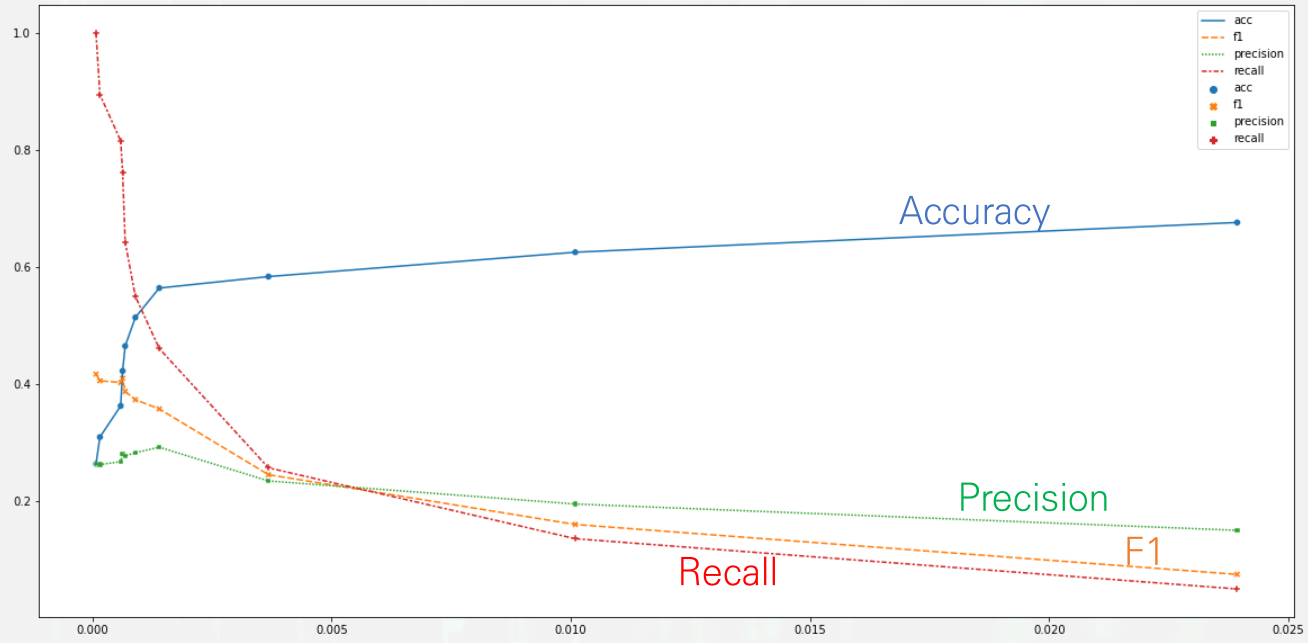
- TCP Model Best Perfomance

구분	TCP		
Model	Base	Stacked	Denoising
Threshold	0.001335	0.006295	0.000674
Accuracy	*0.795849*	*0.636601*	*0.735556*
F1 Score	*0.862727*	*0.752353*	*0.842316*
Precision	0.823718	0.727252	0.729147
Recall	0.905614	0.77925	0.997067

Initial model

- **Protocol 구분 없이 전체 데이터 학습 모델**
  - Base AE Model, BATCH_SIZE = 512, EPOCH=5
  - Train Data의 Loss 값의 백분위수 (0~100, 10단위)를 Threshold로 설정

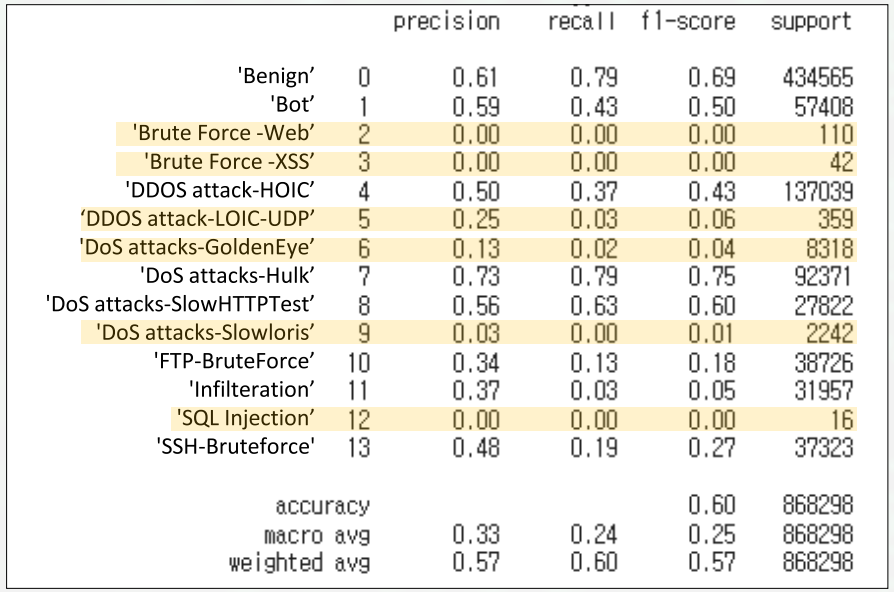
	tp	fp	tn	fn	acc	f1	precision	recall
0.000076	434149	1215426	0	0	0.263188	0.416705	0.263188	1.0
0.000158	388323	1093971	121455	45826	0.309036	0.405254	0.261974	0.894446
0.000591	354038	971810	243616	80111	0.362308	0.402317	0.267028	0.815476
0.000631	330820	850023	365403	103329	0.422062	0.409686	0.280156	0.761996
0.000898	238632	606900	608526	195517	0.513561	0.372955	0.282227	0.549655
0.003681	111351	364400	851026	322798	0.583409	0.244754	0.234053	0.256481
0.010093	58733	243053	972373	375416	0.625074	0.159615	0.194618	0.135283
0.023930	21437	121881	1093545	412712	0.675921	0.074245	0.149576	0.049377
0.178395	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
0.001397	199889	485240	730186	234260	0.563827	0.357175	0.291754	0.460416
0.000685	278824	727744	487682	155325	0.464669	0.387063	0.277005	0.642231



ML Test

- **RandomForestClassifier 등 5개 Classifier 지도 학습, 성능 비교**
  - ML 지도 학습에서도 성능이 매우 떨어지는 것을 확인함

Model	RF	DT	LR	LGBM	CB
F1 Score	0.60268	0.565153	0.542362	0.574432	0.6111



		precision	recall	f1-score	support
'Benign'	0	0.61	0.79	0.69	434565
'Bot'	1	0.59	0.43	0.50	57408
'Brute Force -Web'	2	0.00	0.00	0.00	110
'Brute Force -XSS'	3	0.00	0.00	0.00	42
'DDOS attack-HOIC'	4	0.50	0.37	0.43	137039
'DDOS attack-LOIC-UDP'	5	0.25	0.03	0.06	359
'DoS attacks-GoldenEye'	6	0.13	0.02	0.04	8318
'DoS attacks-Hulk'	7	0.73	0.79	0.75	92371
'DoS attacks-SlowHTTPTest'	8	0.56	0.63	0.60	27822
'DoS attacks-Slowloris'	9	0.03	0.00	0.01	2242
'FTP-BruteForce'	10	0.34	0.13	0.18	38726
'Infilteration'	11	0.37	0.03	0.05	31957
'SQL Injection'	12	0.00	0.00	0.00	16
'SSH-Bruteforce'	13	0.48	0.19	0.27	37323
accuracy				0.60	868298
macro avg		0.33	0.24	0.25	868298
weighted avg		0.57	0.60	0.57	868298

▶ Classification Report (RF)

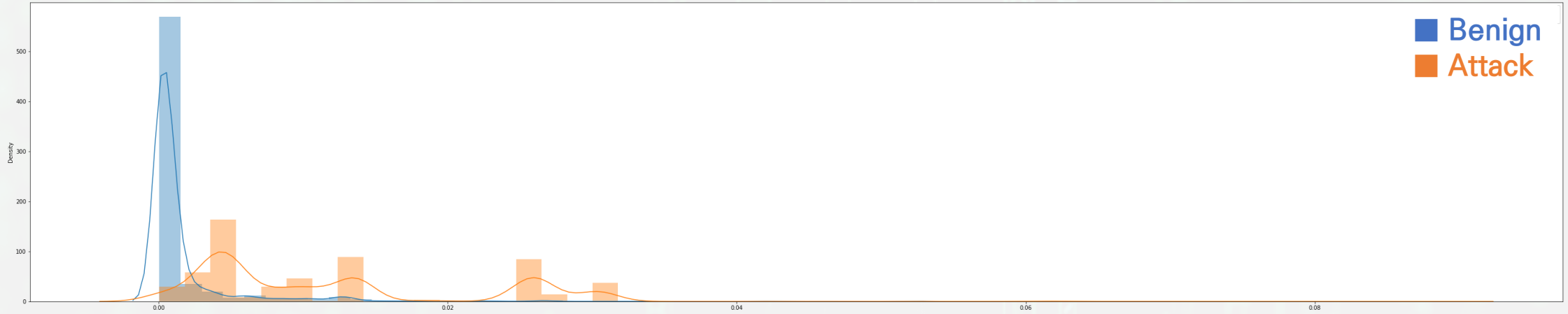
- 일부 공격 유형은 거의 예측을 못하고 있음

Protocol별 모델링

- **TCP 데이터만으로 Model Evaluate 진행**
  - 500,000 Sample 중 TCP에 해당하는 데이터만 Train / Test
  - 성능이 대폭 향상되는 것을 확인함 → Protocol별로 모델 분리 결정

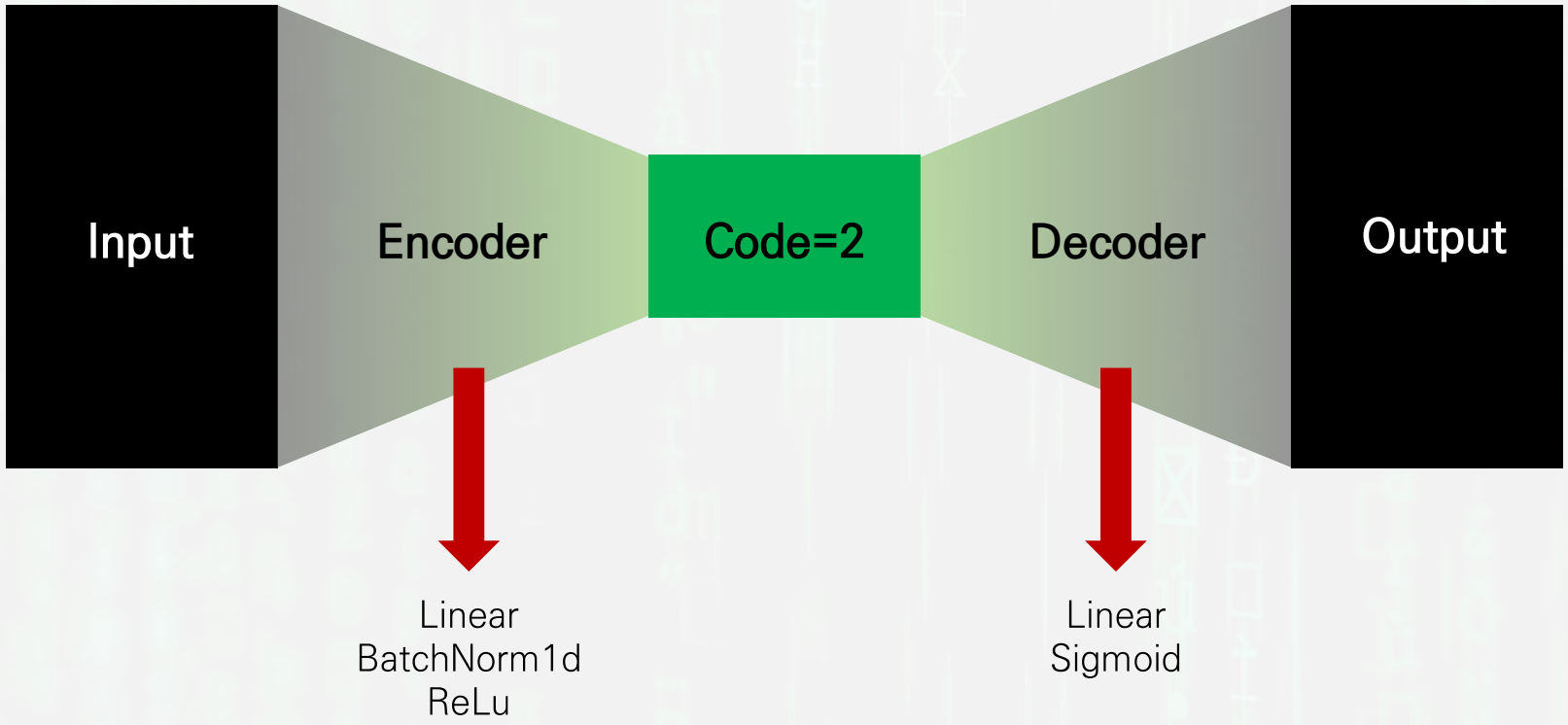
Threshold = 0.0015	Accuracy	F1 Score	Precision	Recall
Score	0.892255	0.897986	0.852419	0.9487

  - Benign과 Attack의 Loss(MSE) 분포



TCP Model – 1) Base AutoEncoder

- Base AE 모델은 layer=1로 구성한 가장 기초적인 모델



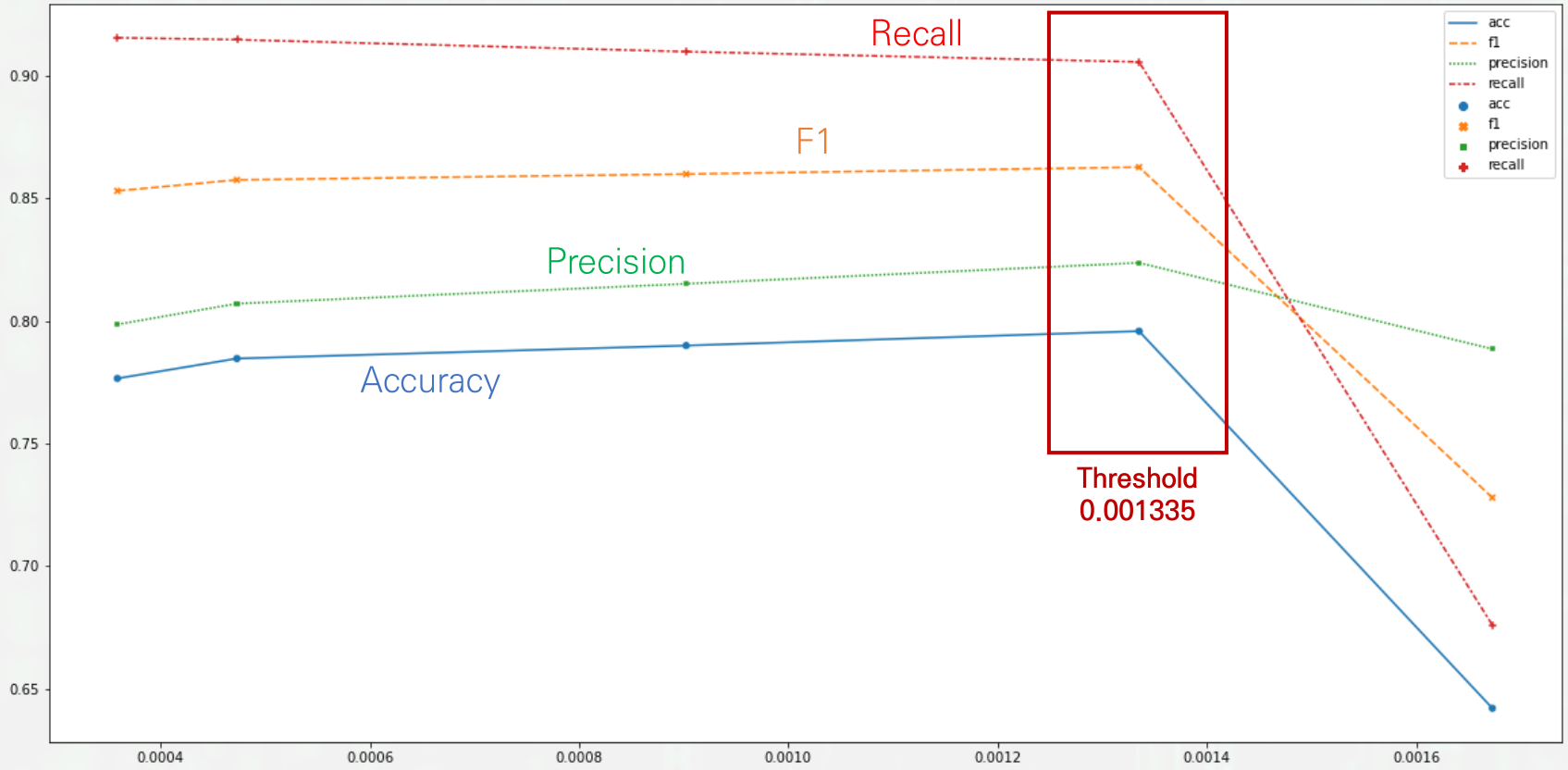
TCP Model – 1) Base AutoEncoder (계속)

- **Threshold 0.001335에서 Accuracy 0.7958, F1 Score 0.8627로 가장 좋은 성능을 보임**
  - Train Loss 기준 50번째 백분위수에 해당

Train Loss Percentile	Threshold	Accuracy	F1	Precision	Recall	TP	FP	TN	FN
44	0.000473	0.784691	0.857526	0.807079	0.914699	1934576	462432	408249	180410
47	0.000902	0.789975	0.859892	0.81516	0.909818	1924252	436331	434350	190734
50	**0.001335**	**0.795849**	**0.862727**	**0.823718**	**0.905614**	**1915361**	**409902**	**460779**	**199625**
53	0.000359	0.776572	0.853051	0.79859	0.915484	1936236	488333	382348	178750
56	0.001673	0.642091	0.727958	0.788568	0.676	1429731	383341	487340	685255

TCP Model – 1) Base AutoEncoder (계속)

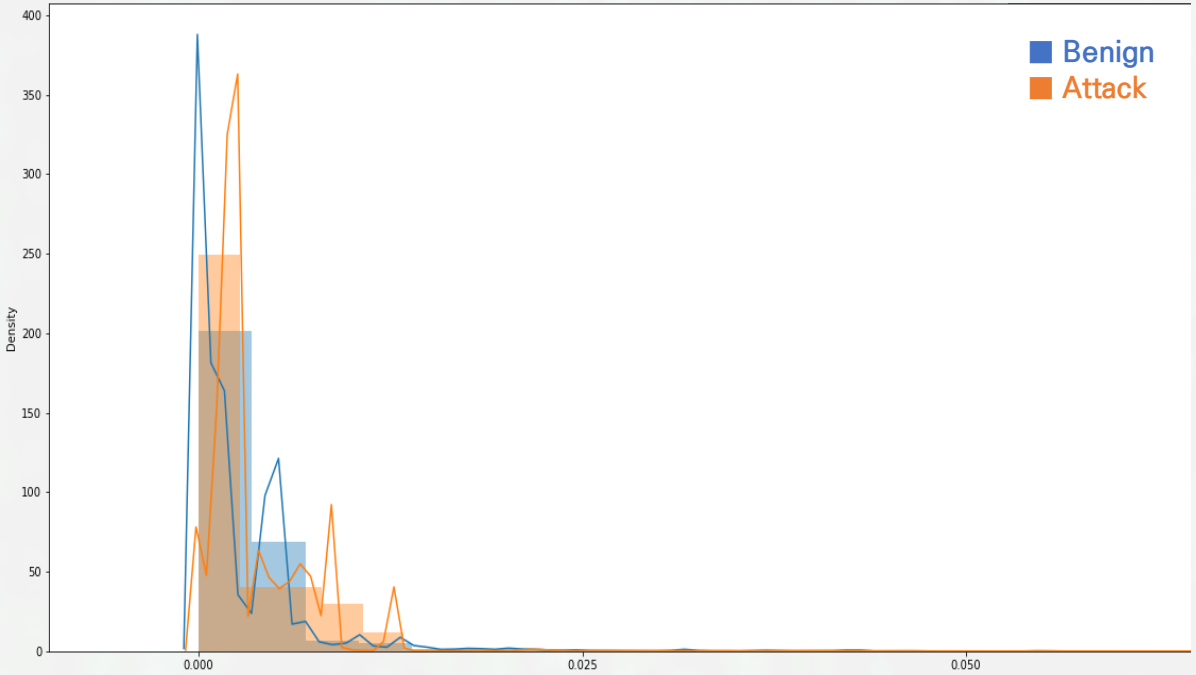
- **Threshold 0.001335에서 Accuracy 0.7958, F1 Score 0.8627로 가장 좋은 성능을 보임**
  - Train Loss 기준 50번째 백분위수에 해당



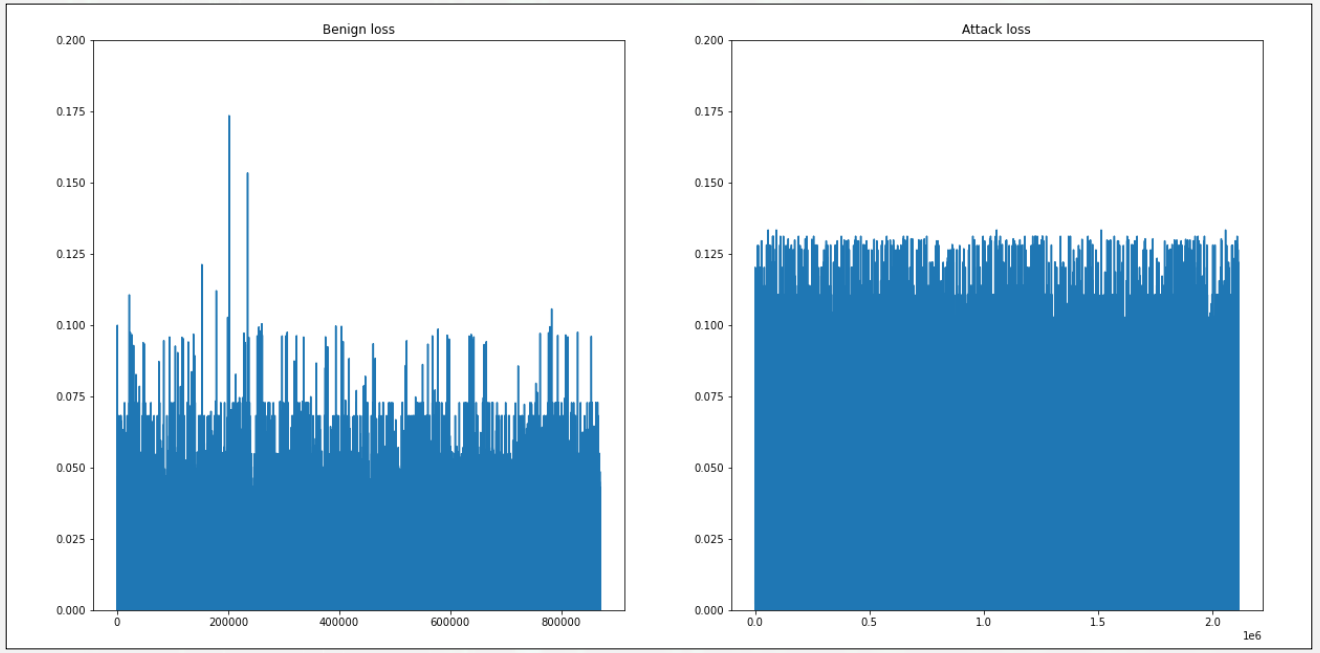
TCP Model – 1) Base AutoEncoder (계속)

- **Loss Visualization**

TCP Benign / Attack Loss 분포

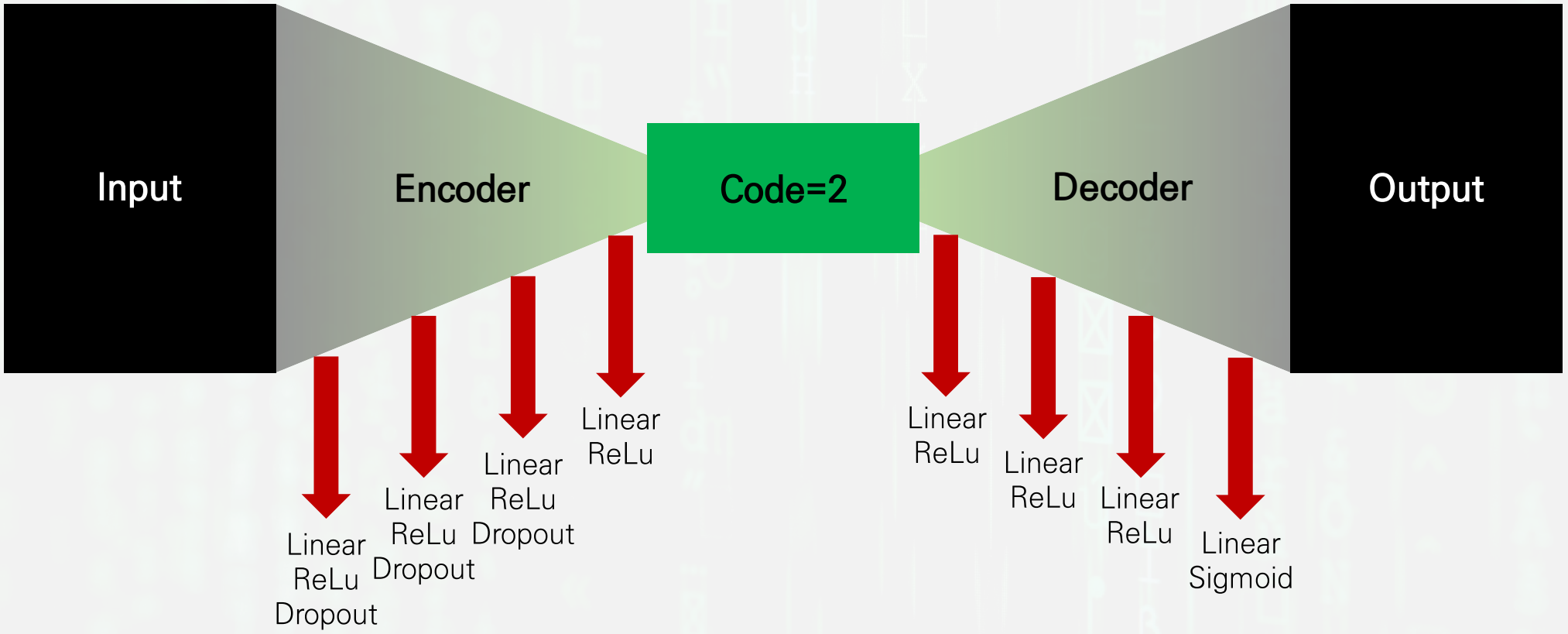


TCP Benign / Attack Loss 비교



TCP Model – 2) Stacked AutoEncoder

- Stacked AE 모델은 layer=4로 구성한 모델로, Dropout 등을 포함함



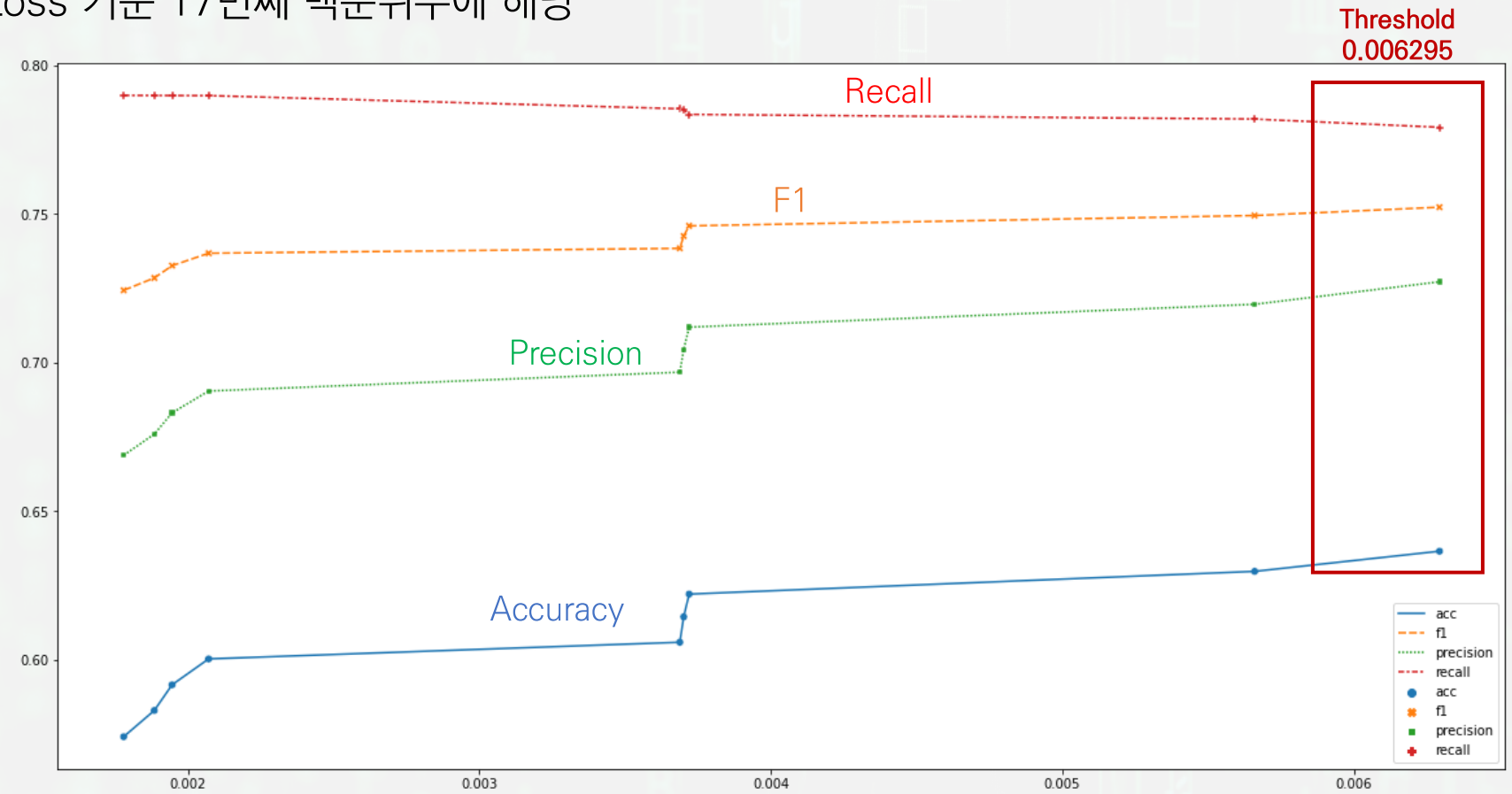
TCP Model – 2) Stacked AutoEncoder (계속)

- **Threshold 0.006295에서 Accuracy 0.6366, F1 Score 0.7793로 가장 좋은 성능을 보임**
  - Train Loss 기준 17번째 백분위수에 해당

Train Loss Percentile	Threshold	Accuracy	F1	Precision	Recall	TP	FP	TN	FN
11	0.003702	0.614527	0.742659	0.704493	0.785198	1660683	696592	174089	454303
14	0.00372	0.622170	0.746079	0.711998	0.783586	1657274	670364	200317	457712
17	**0.006295**	**0.636601**	**0.752353**	**0.727252**	**0.77925**	**1648102**	**618103**	**252578**	**466884**
20	0.002072	0.600368	0.736866	0.690501	0.789907	1670642	748823	121858	444344
23	0.001885	0.58301	0.728552	0.676004	0.78995	1670753	800761	69920	444233

TCP Model – 2) Stacked AutoEncoder (계속)

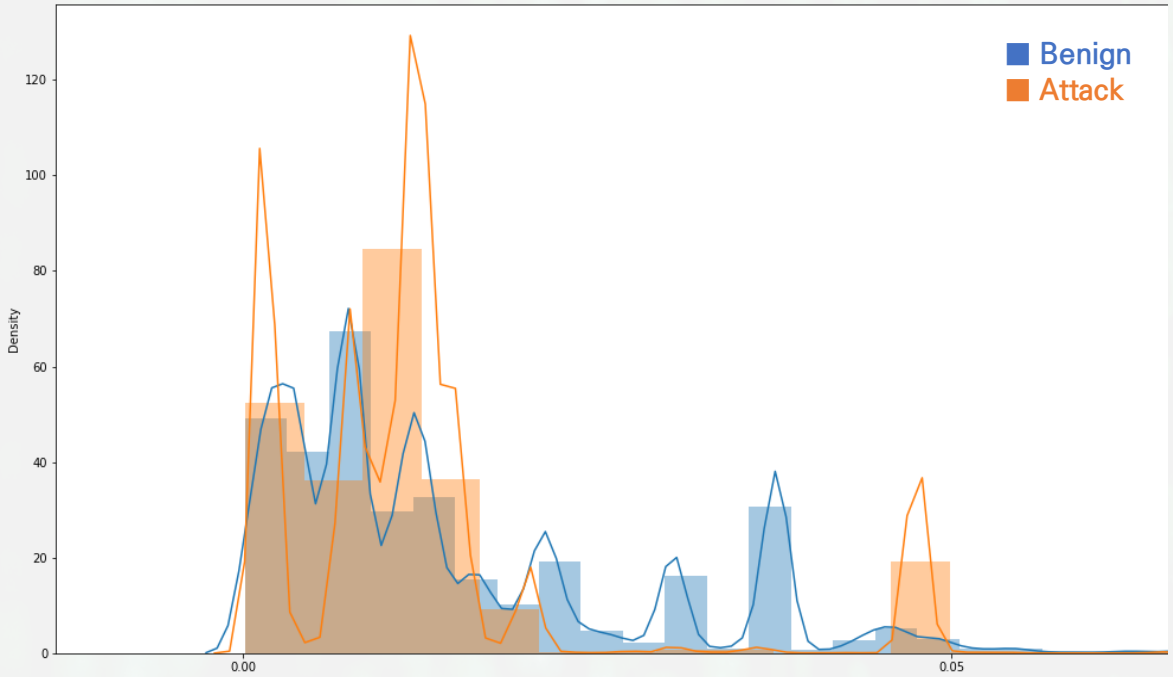
- **Threshold 0.006295에서 Accuracy 0.6366, F1 Score 0.7793로 가장 좋은 성능을 보임**
  - Train Loss 기준 17번째 백분위수에 해당



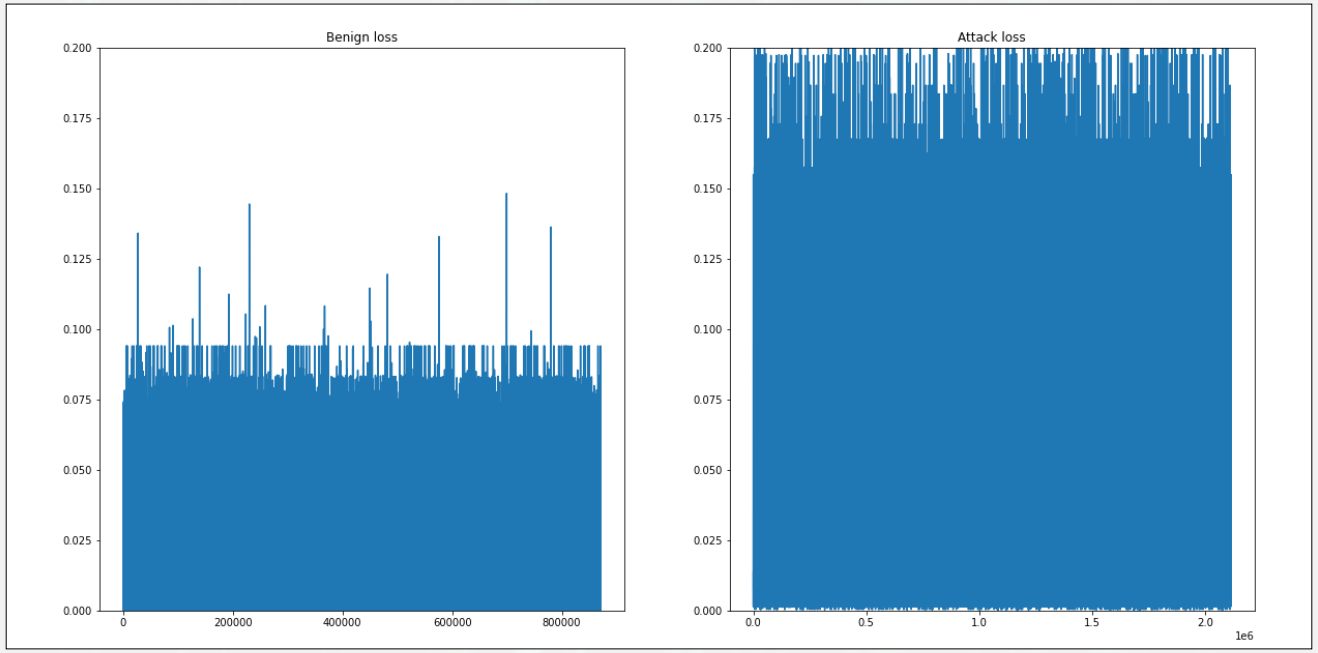
TCP Model – 2) Stacked AutoEncoder (계속)

- **Loss Visualization**

TCP Benign / Attack Loss 분포

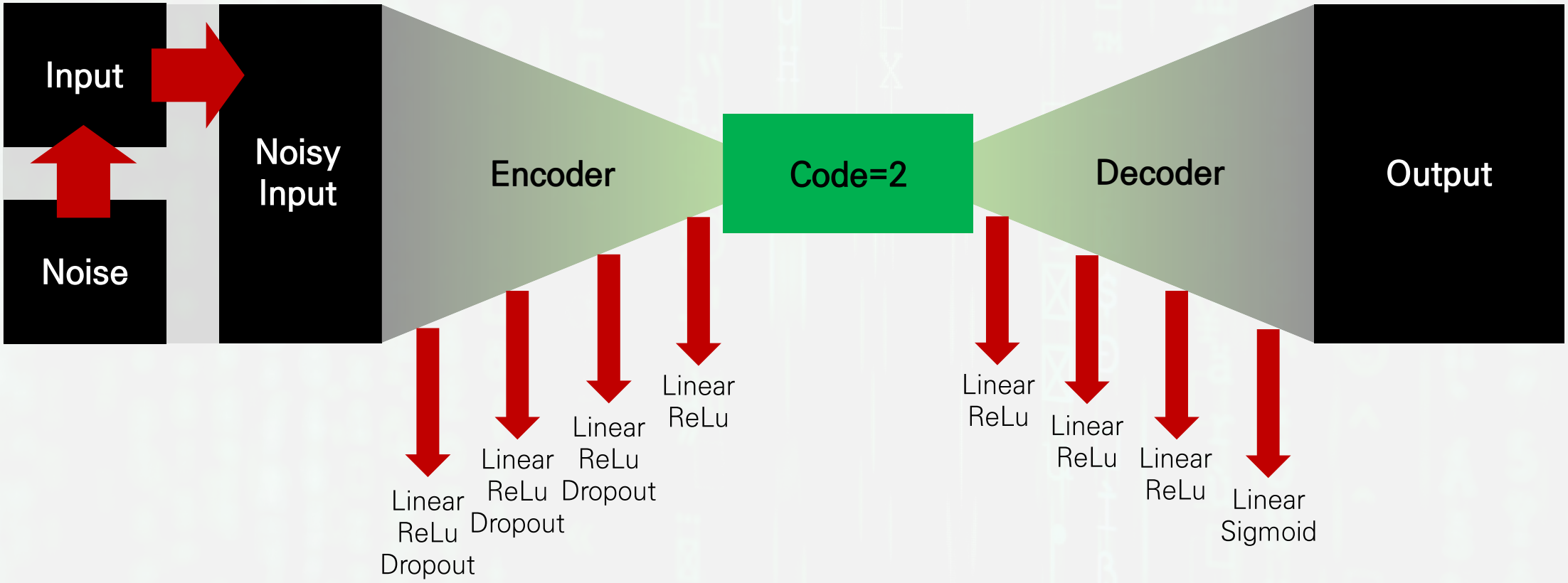


TCP Benign / Attack Loss 비교



TCP Model – 3) Denoising AutoEncoder

- Denoising AE 모델은 기존 Stacked AE를 그대로 활용하고, Noise를 추가한 Input으로 Train 진행



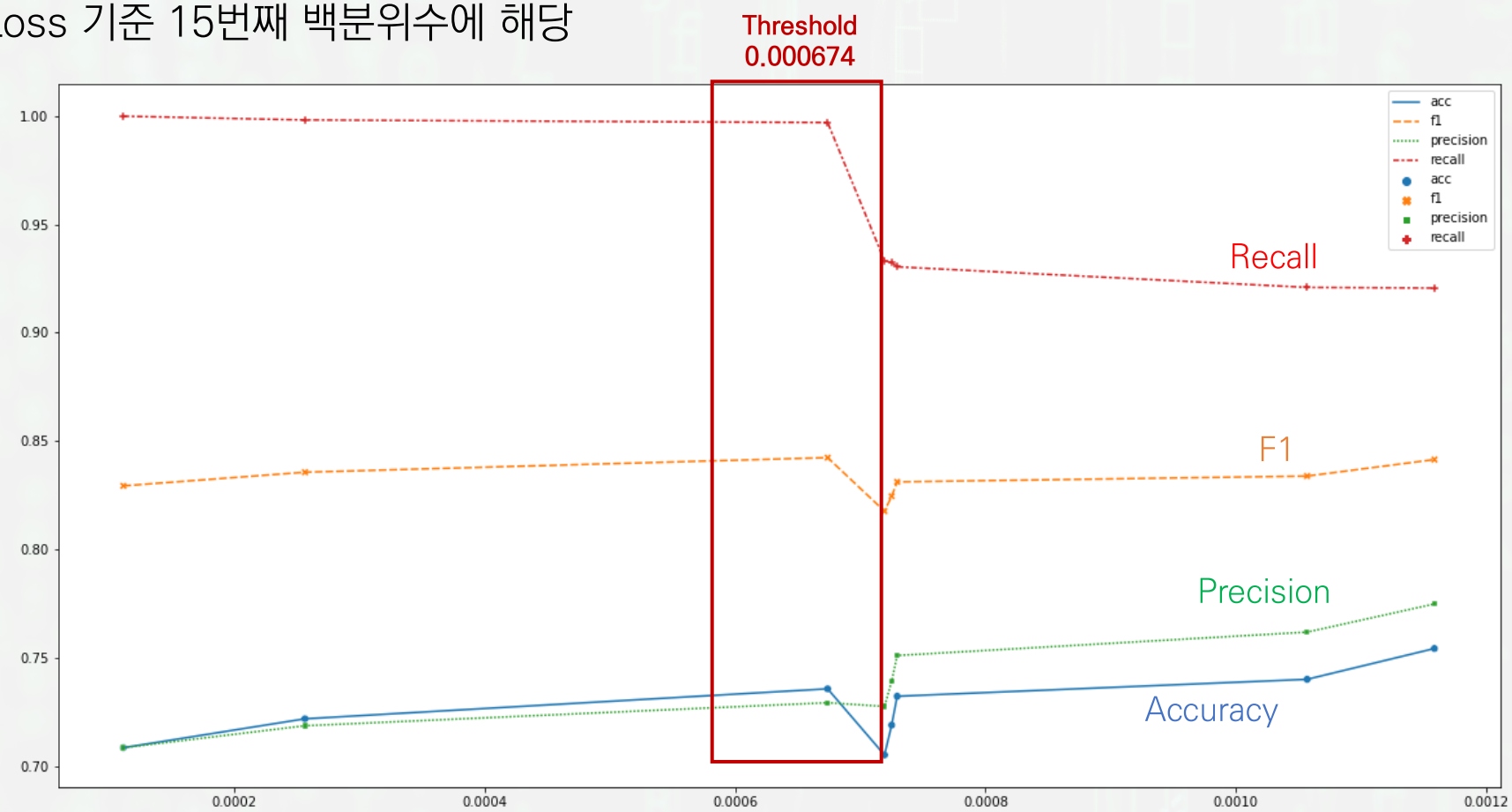
TCP Model – 3) Denoising AutoEncoder (계속)

- **Threshold 0.000674에서 Accuracy 0.7356, F1 Score 0.8423로 가장 좋은 성능을 보임**
  - Train Loss 기준 15번째 백분위수에 해당

Train Loss Percentile	Threshold	Accuracy	F1	Precision	Recall	TP	FP	TN	FN
5	0.000256	0.721717	0.835585	0.718506	0.998246	2111277	827150	43531	3709
10	0.000725	0.718816	0.824496	0.738983	0.932391	1971993	696530	174151	142993
15	**0.000674**	**0.735556**	**0.842316**	**0.729147**	**0.997067**	**2108782**	**783338**	**87343**	**6204**
20	0.000719	0.705024	0.817609	0.727423	0.933321	1973961	739674	131007	141025
25	0.000729	0.732136	0.831114	0.750954	0.930433	1967852	652618	218063	147134

TCP Model – 3) Denoising AutoEncoder (계속)

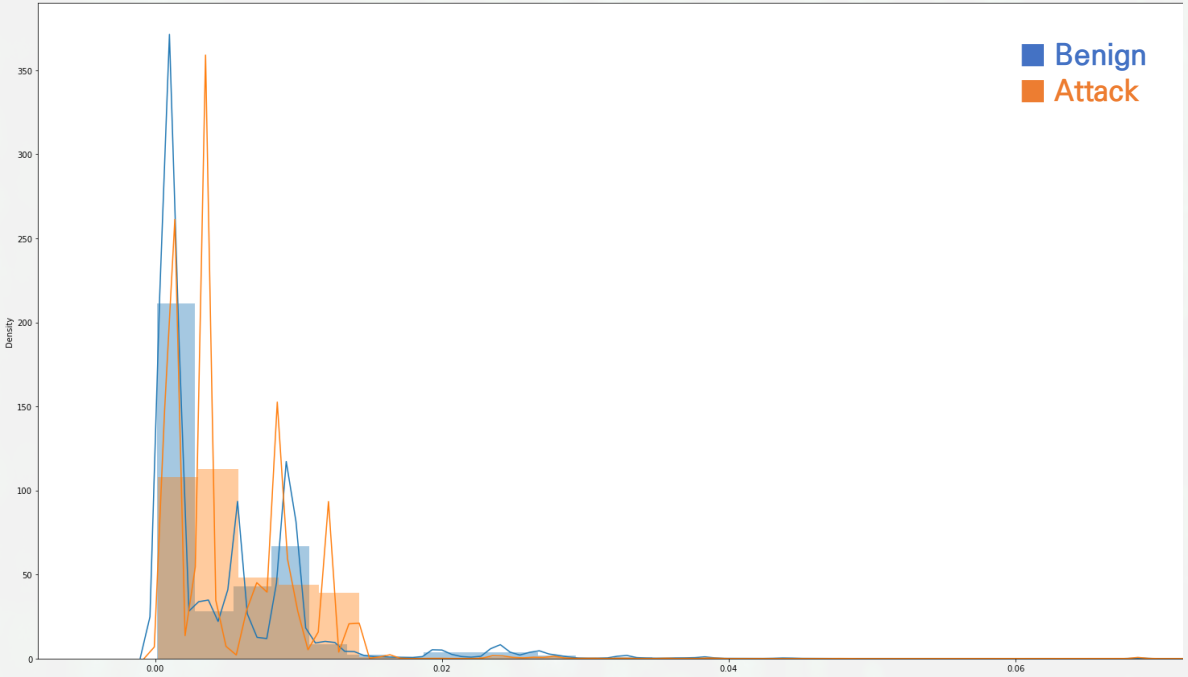
- **Threshold 0.000674에서 Accuracy 0.7356, F1 Score 0.8423로 가장 좋은 성능을 보임**
  - Train Loss 기준 15번째 백분위수에 해당



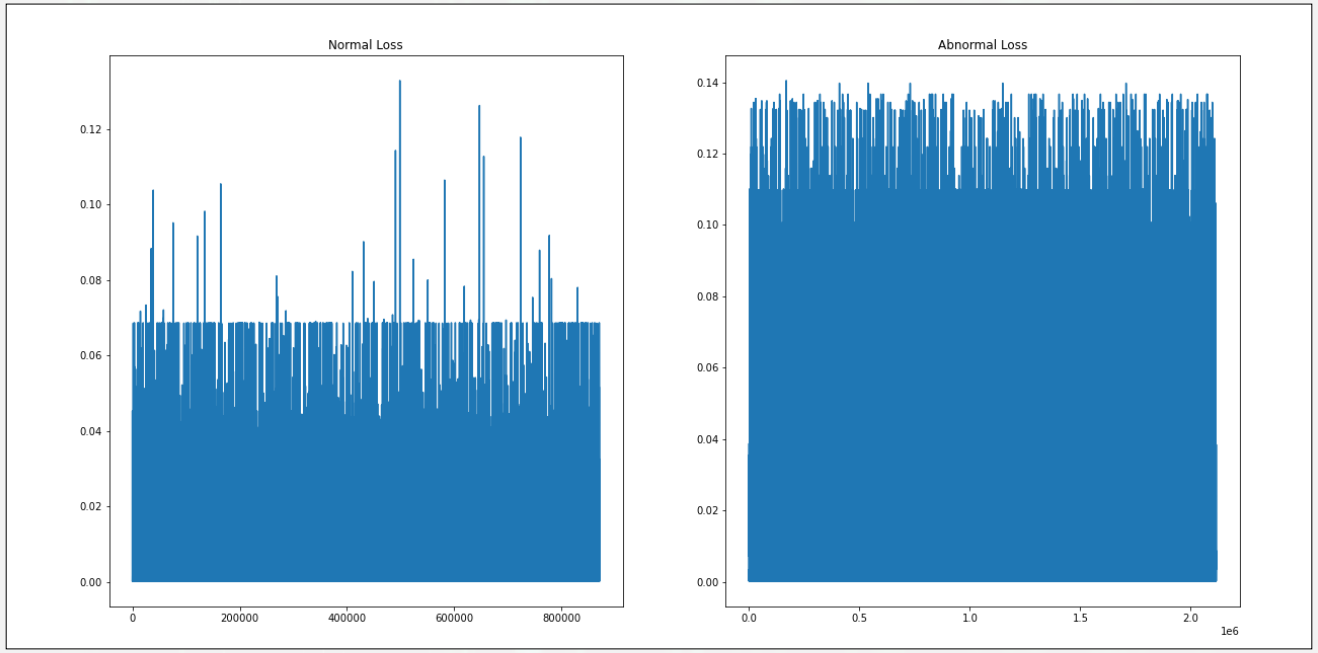
TCP Model – 3) Denoising AutoEncoder (계속)

- **Loss Visualization**

TCP Benign / Attack Loss 분포



TCP Benign / Attack Loss 비교



5. Conclusion

모델 평가 결과 (Again)

- TCP Model Best Performance

구분	TCP		
Model	Base	Stacked	Denoising
Threshold	0.001335	0.006295	0.000674
Accuracy	*0.795849*	*0.636601*	*0.735556*
F1 Score	*0.862727*	*0.752353*	*0.842316*
Precision	0.823718	0.727252	0.729147
Recall	0.905614	0.77925	0.997067

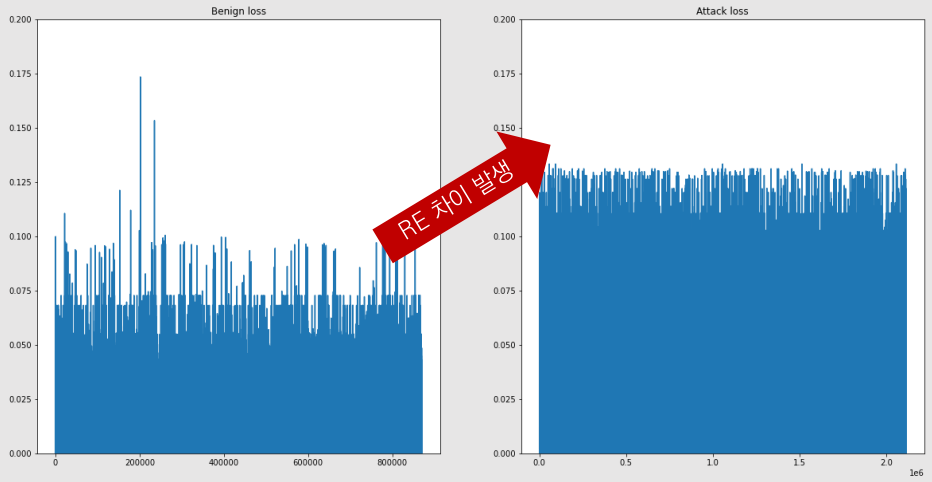
✓ UDP, HOP 모델 성능 업데이트 예정

Project Review

79.6%

TCP Base Model 기준

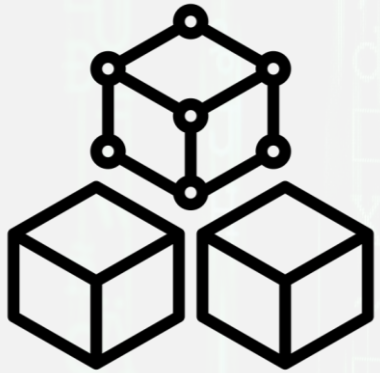
- Accuracy : 0.7958
- F1 Score : 0.8627

연구 목표	새로운 유형의 Network Intrusion도 탐지할 수 있는, 이상 탐지 기반 딥 러닝 모델을 개발하는 것이 최종 목표
모델링 결과	Model을 통해 Normal와 Anomaly의 Reconstruction Error 차이가 발생 
연구 성과	- AutoEncoder Reconstruction Error를 활용한 이상 탐지 성과 (Acc 79.6%) - 준지도학습을 통한 새로운 공격 유형에 대한 탐지 가능성 확인

연구의 한계점

Dataset (UDP, HOP)

- ✓ 9일치의 데이터만을 분석
- ✓ 정상/비정상 데이터가 매우 유사
- ✓ Network Domain 지식 부족
- ✓ 정상 데이터의 Outlier 분석 필요



Model(AutoEncoder)

- ✓ Noise에 매우 민감함
- ✓ 데이터에 따라 Anomaly 구분 X
- ✓ GAN, ML 등 Hybrid Model 필요
- ✓ AE와는 다른 심층방법론 적용 필요

Development Environment

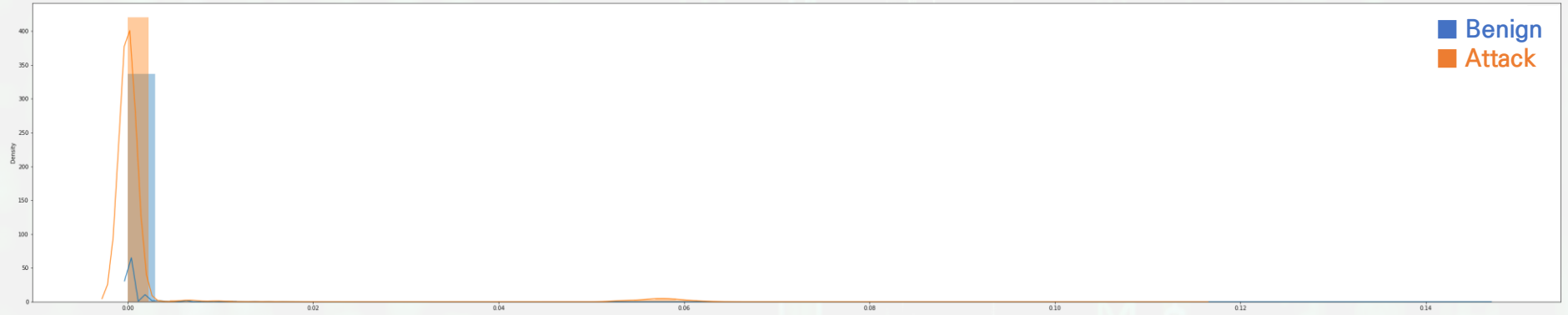
- ✓ 대용량 데이터 처리 환경 부족
- ✓ 3개 모델을 모두 학습할 기기 부족
- ✓ 비용·시간적 한계

Appendix

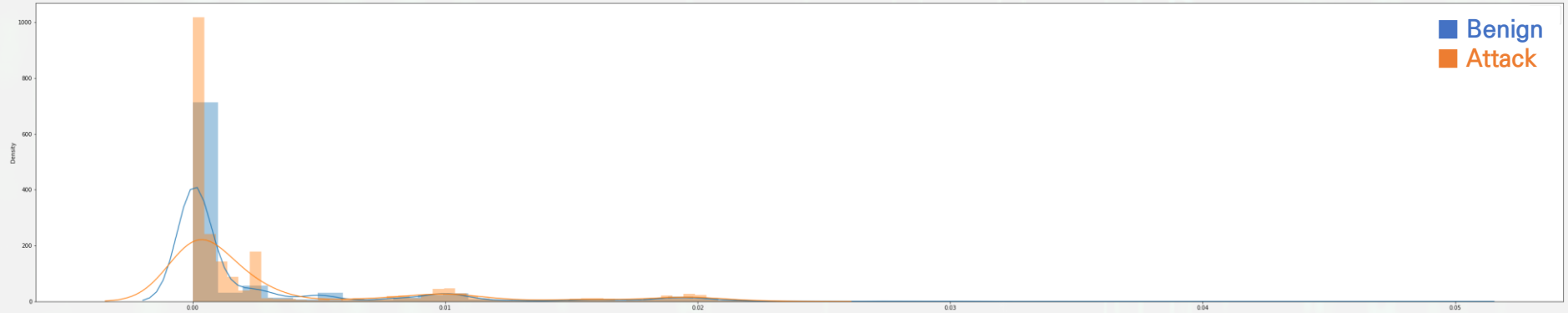
Challenge – UDP, HOP

- **예상과는 달리 UDP, HOP에서 Anomaly를 구별해내지 못하는 상황**

UDP(Base AE) Loss Distribution



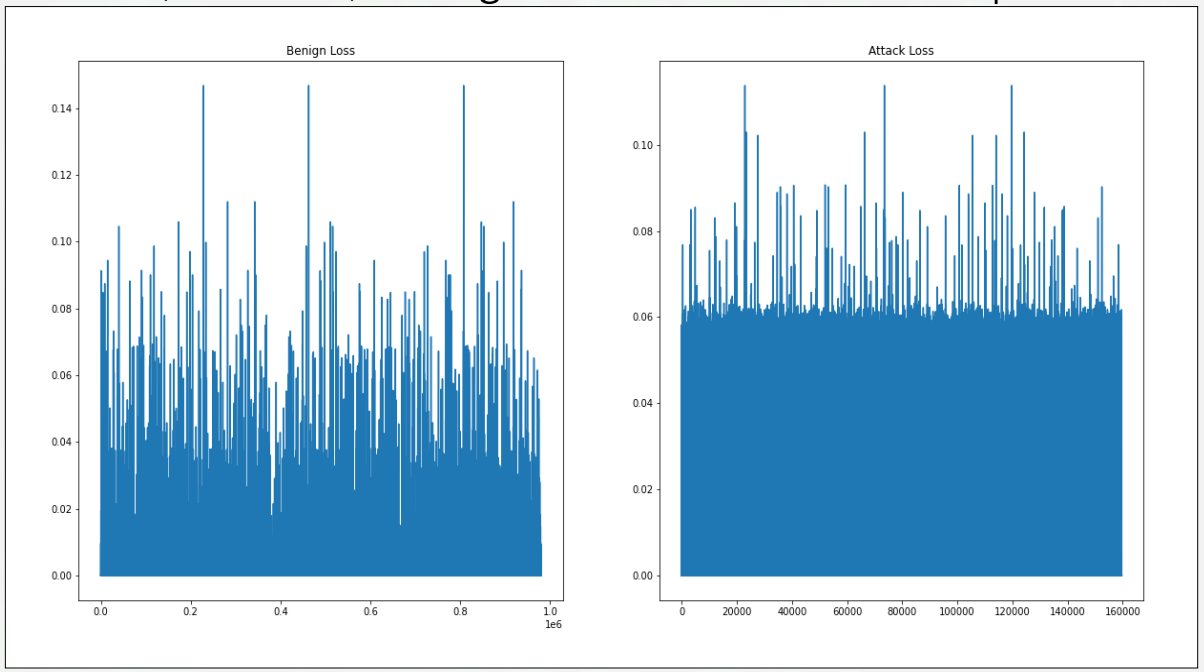
HOP(Base AE) Loss Distribution



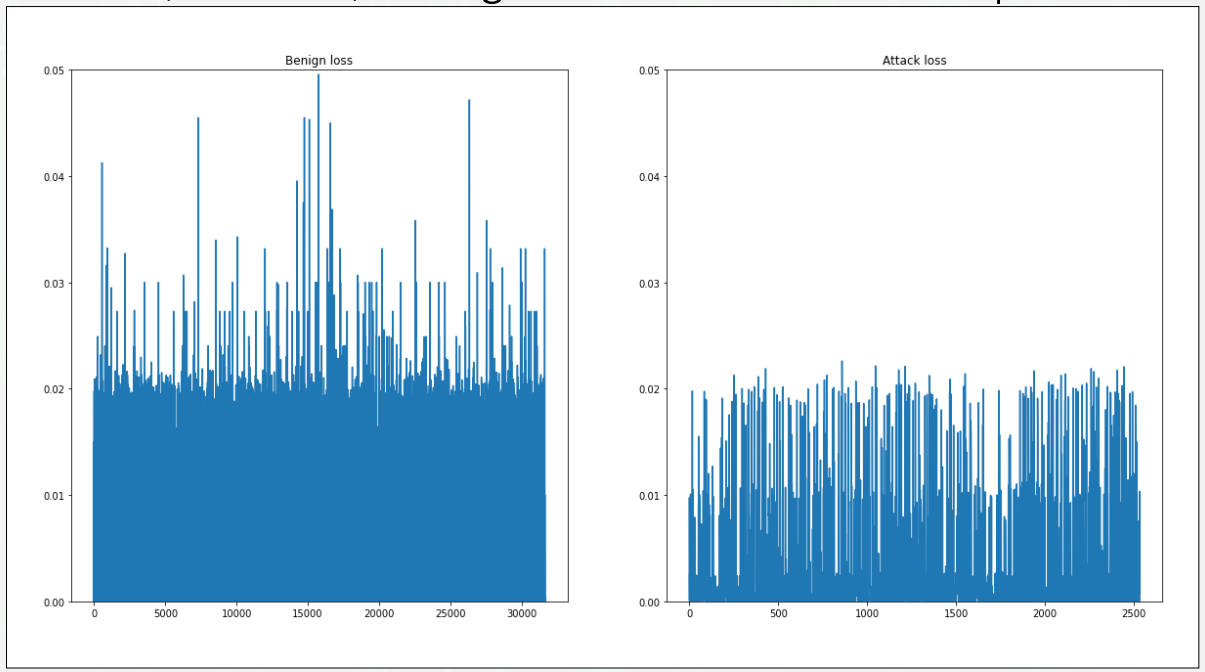
Challenge – UDP, HOP (계속)

- **UDP에서 Benign vs Attack Loss 차이는 크지 않으며, HOP는 Benign Loss가 오히려 크게 나타남**

UDP(Base AE) Benign Vs. Attack Loss Comparison



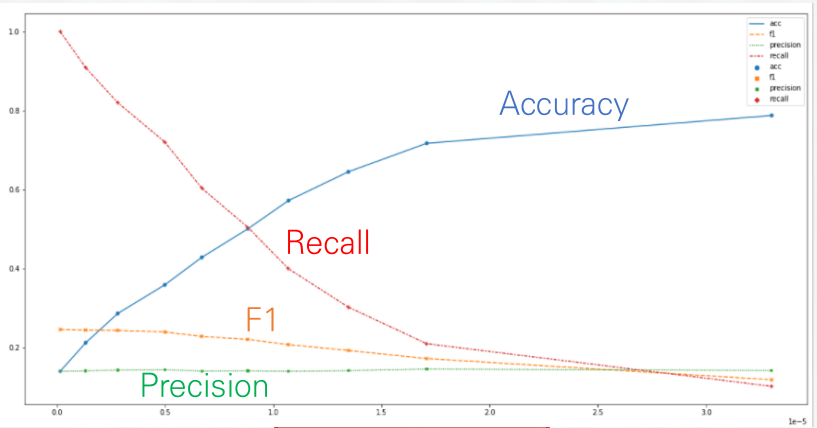
HOP(Base AE) Benign Vs. Attack Loss Comparison



Challenge – UDP, HOP (계속)

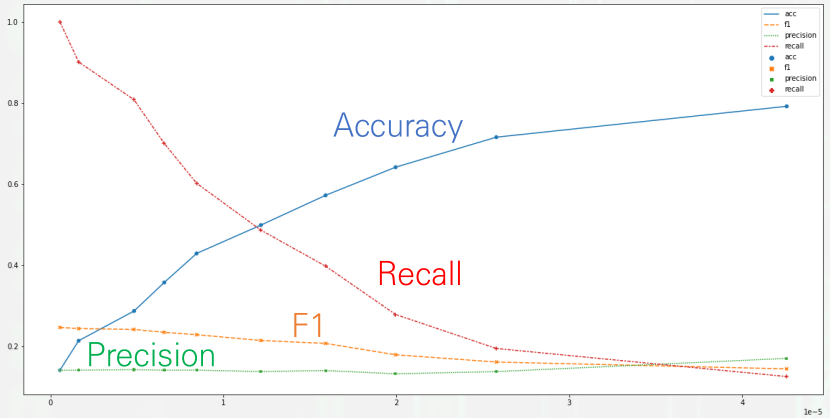
- UDP's 3 Models Performance

Base AE



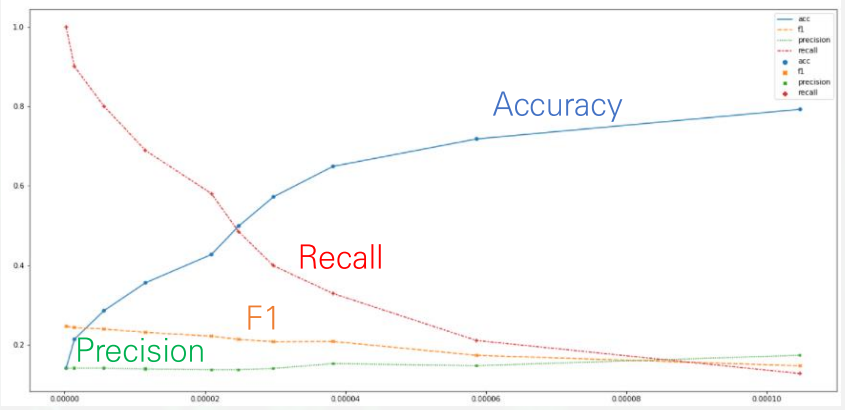
Train Loss Percentile	Threshold	Accuracy	F1	Precision	Recall
0	0.000000	0.140193	0.245910	0.140193	1.000000
10	0.000001	0.212155	0.244427	0.141197	0.908987
20	0.000003	0.286412	0.243713	0.143122	0.820134
30	0.000005	0.358901	0.239650	0.143722	0.720666
40	0.000007	0.428546	0.228454	0.140896	0.603480
50	0.000009	0.500806	0.220831	0.141344	0.504594
60	0.000011	0.572297	0.207599	0.140219	0.399639
70	0.000013	0.645163	0.192917	0.141616	0.302501
80	0.000017	0.717234	0.172334	0.146132	0.209984
90	0.000033	0.787658	0.118742	0.141976	0.102042

Stacked AE



Train Loss Percentile	Threshold	Accuracy	F1	Precision	Recall
0	0.000001	0.140193	0.24591	0.140193	1
10	0.000002	0.213082	0.243052	0.140469	0.901171
20	0.000005	0.286378	0.24095	0.141588	0.807921
30	0.000007	0.356826	0.233837	0.140359	0.700111
40	0.000008	0.428491	0.227926	0.140589	0.601732
50	0.000012	0.498462	0.213674	0.136935	0.486068
60	0.000016	0.571939	0.206448	0.139472	0.397178
70	0.00002	0.641341	0.17846	0.131438	0.277869
80	0.000026	0.715495	0.160487	0.13686	0.193976
90	0.000043	0.791664	0.143617	0.169469	0.124608

Denoising AE

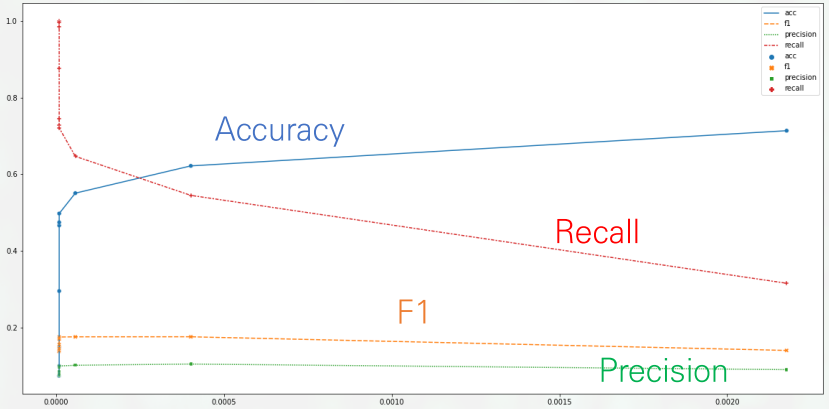


Train Loss Percentile	Threshold	Accuracy	F1	Precision	Recall
0	0.000000	0.140193	0.24591	0.140193	1.000000
10	0.000001	0.213146	0.242905	0.14039	0.900381
20	0.000006	0.426731	0.22102	0.136516	0.580106
30	0.000012	0.285116	0.238953	0.140436	0.800537
40	0.000021	0.355509	0.230612	0.138483	0.688969
50	0.000025	0.498401	0.212958	0.136506	0.484058
60	0.00003	0.571955	0.206996	0.13981	0.398493
70	0.000038	0.64873	0.207779	0.151924	0.32858
80	0.000059	0.717818	0.172744	0.14664	0.210154
90	0.000105	0.792523	0.14637	0.172932	0.126881

Challenge – UDP, HOP (계속)

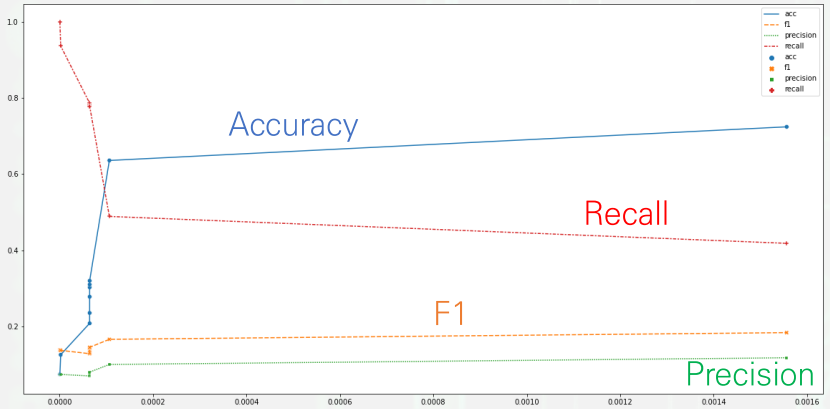
- HOP's 3 Models Performance

Base AE



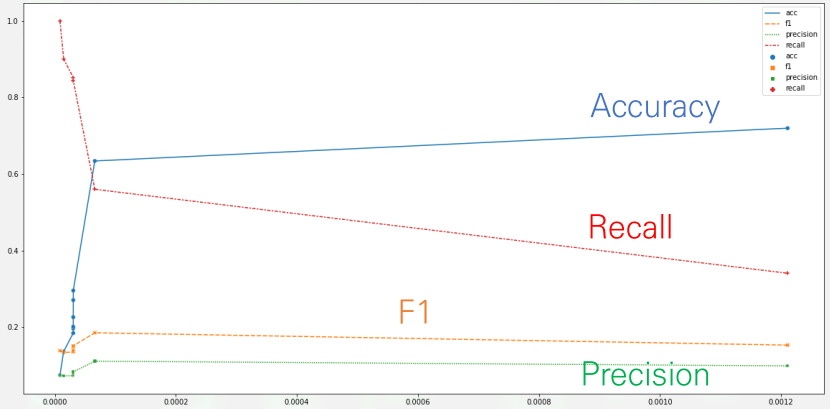
Train Loss Percentile	Threshold	Accuracy	F1	Precision	Recall
0	0.0000084	0.0741966	0.1380964	0.0741695	1.0000000
10	0.0000091	0.0999239	0.1410056	0.0758733	0.9960537
20	0.0000091	0.1493297	0.1464819	0.0791294	0.9842147
30	0.0000091	0.2953814	0.1556538	0.0854184	0.8756906
40	0.0000091	0.4666628	0.1717273	0.0970410	0.7454617
50	0.0000091	0.4745361	0.1704173	0.0965091	0.7277032
60	0.0000092	0.4976585	0.1755296	0.0999289	0.7209945
70	0.0000571	0.5507522	0.1759811	0.1018455	0.6468035
80	0.0004014	0.6218463	0.1761255	0.1050350	0.5449882
90	0.0021773	0.7136627	0.1405605	0.0904057	0.3157064

Stacked AE



Train Loss Percentile	Threshold	Accuracy	F1	Precision	Recall
0	0.0000007	0.0741673	0.1380926	0.0741673	1.0000000
10	0.0000024	0.1255342	0.1371264	0.0739771	0.9368587
20	0.0000644	0.2081894	0.1284191	0.0699176	0.7865036
30	0.0000644	0.2354680	0.1315868	0.0718461	0.7809787
40	0.0000644	0.2779957	0.1379045	0.0756518	0.7786109
50	0.0000644	0.3027864	0.1421111	0.0781913	0.7786109
60	0.0000644	0.3101914	0.1434179	0.0789832	0.7786109
70	0.0000645	0.3199672	0.1451803	0.0800536	0.7786109
80	0.0001066	0.6359539	0.1661303	0.1000646	0.4889503
90	0.0015562	0.7243166	0.1837248	0.1177124	0.4183110

Denoising AE



Train Loss Percentile	Threshold	Accuracy	F1	Precision	Recall
0	0.0000086	0.0741966	0.1380964	0.0741695	1.0000000
10	0.0000147	0.1361295	0.1339241	0.0723411	0.9005525
20	0.0000305	0.1841597	0.1341865	0.0728254	0.8524073
30	0.0000305	0.1962477	0.1359303	0.0738537	0.8524073
40	0.0000305	0.2007259	0.1365878	0.0742421	0.8524073
50	0.0000306	0.2258386	0.1403406	0.0764681	0.8520126
60	0.0000306	0.2702980	0.1476290	0.0808160	0.8520126
70	0.0000306	0.2953521	0.1508835	0.0828460	0.8441200
80	0.0000662	0.6339636	0.1849583	0.1107728	0.5599842
90	0.0012110	0.7197506	0.1527298	0.0984373	0.3405683

Challenge – UDP, HOP (계속)

- **Feature Selection – Permutation Importance**
  - RandomForest, DecisionTree, LogisticRegression, LGBM, Catboost 5개 ML Classifier 성능 비교
  - Importance 〉 0인 Feature만 선택

UDP ML Classifier

Classifier	F1 score	Importance 〉 0
RF	0.631799	30개
DT	0.595021	41개
LR	0.539408	36개
LGBM	0.583419	45개
CB	0.653828	49개

HOP ML Classifier

Classifier	F1 score	Importance 〉 0
RF	0.573325	1개
DT	0.512053	15개
LR	0.134544	7개
LGBM	0.839015	7개
CB	0.856908	2개

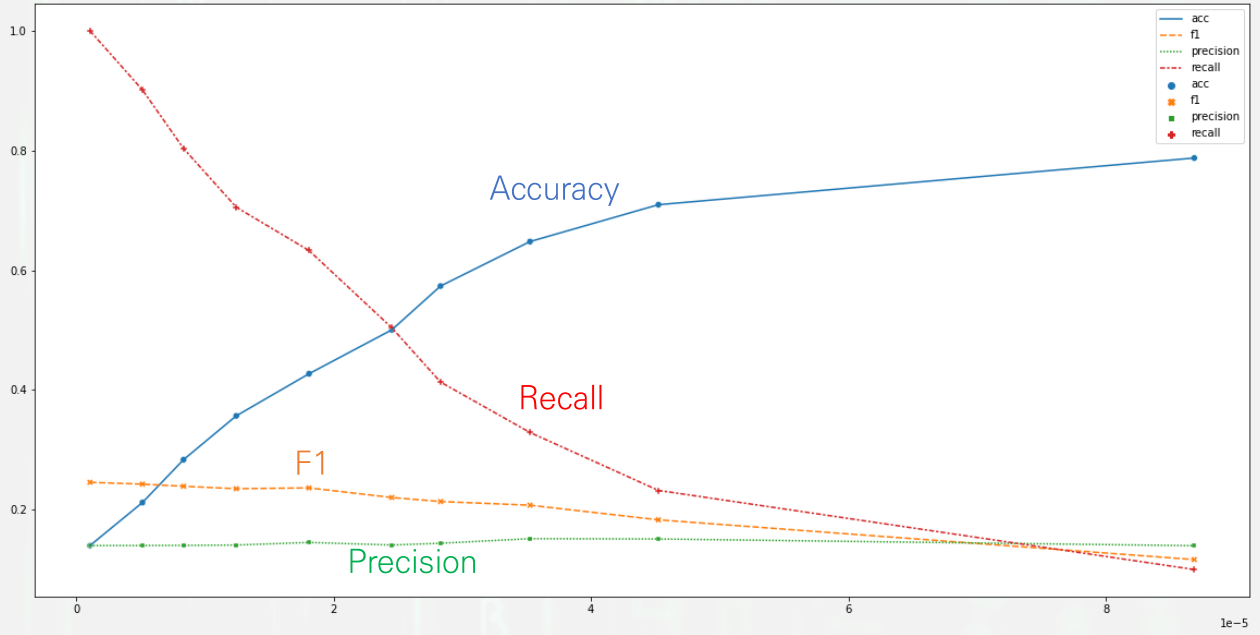
Challenge – UDP, HOP (계속)

- **Feature Selection – Permutation Importance (계속)**
  - UDP RF Columns (30개)
  - Base AE 성능 개선 미미

Threshold별 Score 표

	tp	fp	tn	fn	acc	f1	precision	recall
0.000001	53223	326419	0	0	0.140193	0.24591	0.140193	1
0.000005	47971	293872	32547	5252	0.212089	0.242851	0.140331	0.901321
0.000018	33691	197916	128503	19532	0.427229	0.236569	0.145466	0.633016
0.000008	42762	261464	64955	10461	0.283733	0.239262	0.14056	0.80345
0.000012	37543	228472	97947	15680	0.356889	0.235204	0.141131	0.705391
0.000025	26822	163222	163197	26401	0.500522	0.220515	0.141136	0.503955
0.000028	21997	130622	195797	31226	0.573683	0.213727	0.14413	0.413299
0.000035	17517	97949	228470	35706	0.647945	0.207684	0.151707	0.329125
0.000045	12368	69406	257013	40855	0.709566	0.183234	0.151246	0.232381
0.000087	5336	32781	293638	47887	0.787516	0.116838	0.13999	0.100257

Threshold별 Score 그래프



✓ 전체 Feature 리스트는 Appendix 첨부

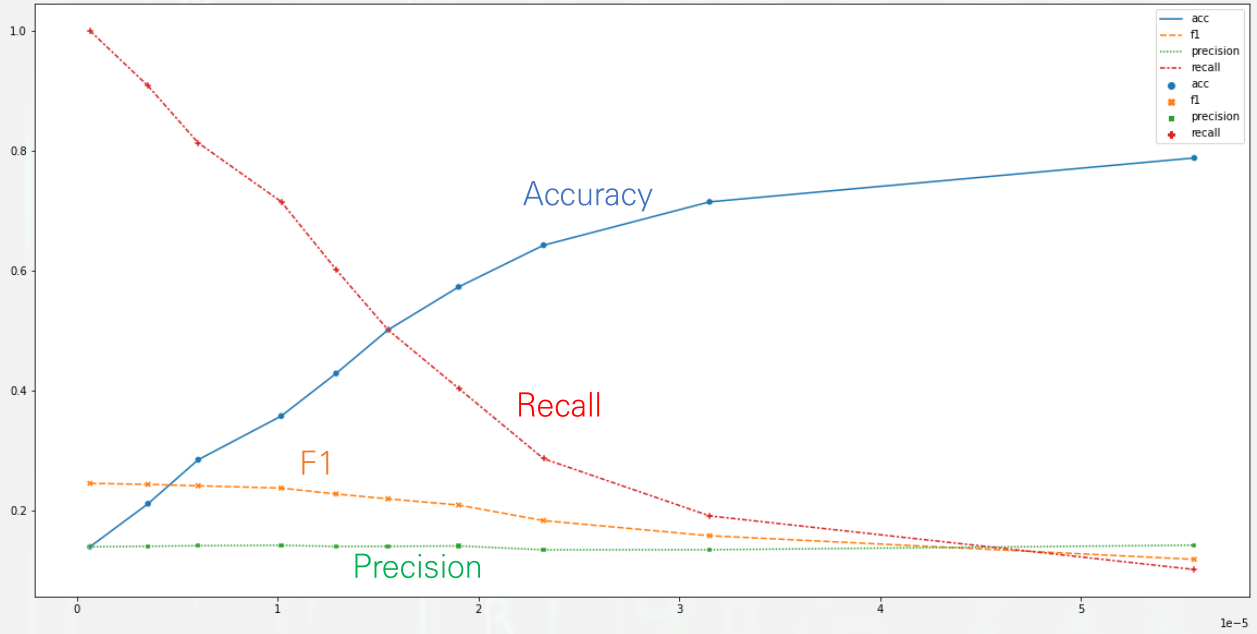
Challenge – UDP, HOP (계속)

- **Feature Selection – Permutation Importance (계속)**
  - UDP CB Columns (49개)
  - Base AE 성능 개선 미미

Threshold별 Score 표

	tp	fp	tn	fn	acc	f1	precision	recall
6.629792e-07	53223	326418	1	0	0.140195	0.245911	0.140193	1
3.551242e-06	48342	294316	32103	4881	0.211897	0.244225	0.141079	0.908292
1.291943e-05	32045	195553	130866	21178	0.429117	0.228224	0.140796	0.602089
6.051201e-06	43290	261420	64999	9933	0.28524	0.241889	0.14207	0.81337
1.018575e-05	38068	228554	97865	15155	0.358056	0.23804	0.142779	0.715255
1.550116e-05	26708	162715	163704	26515	0.501557	0.22014	0.140997	0.501813
1.901794e-05	21516	130330	196089	31707	0.573185	0.209842	0.141696	0.404261
2.324707e-05	15280	97712	228707	37943	0.642677	0.183858	0.135231	0.287094
3.151037e-05	10205	65202	261217	43018	0.714942	0.158672	0.135332	0.19174
5.564205e-05	5450	32666	293753	47773	0.788119	0.119336	0.142985	0.102399

Threshold별 Score 그래프



✓ 전체 Feature 리스트는 Appendix 첨부

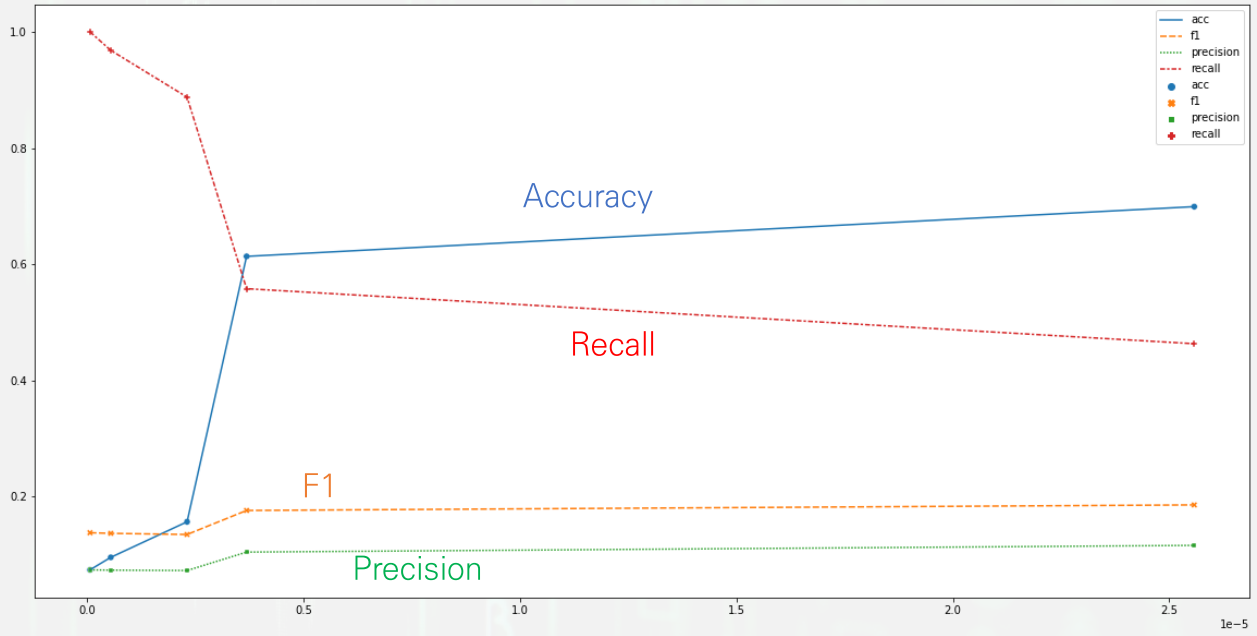
Challenge – UDP, HOP (계속)

- **Feature Selection – Permutation Importance (계속)**
  - HOP CB Columns (2개)
  - Base AE 성능 개선 미미

Threshold별 Score 표

	tp	fp	tn	fn	acc	f1	precision	recall
0.0000001	2534	31631	1	0	0.0741966	0.1380964	0.0741695	1.0000000
0.0000005	2453	30814	818	81	0.0957385	0.1370353	0.0737367	0.9680347
0.0000256	1174	8918	22714	1360	0.6991746	0.1859655	0.1163298	0.4632991
0.0000023	2249	28520	3112	285	0.1569104	0.1350629	0.0730930	0.8875296
0.0000037	1414	12082	19550	1120	0.6135925	0.1764192	0.1047718	0.5580110

Threshold별 Score 그래프



✓ 전체 Feature 리스트는 Appendix 첨부

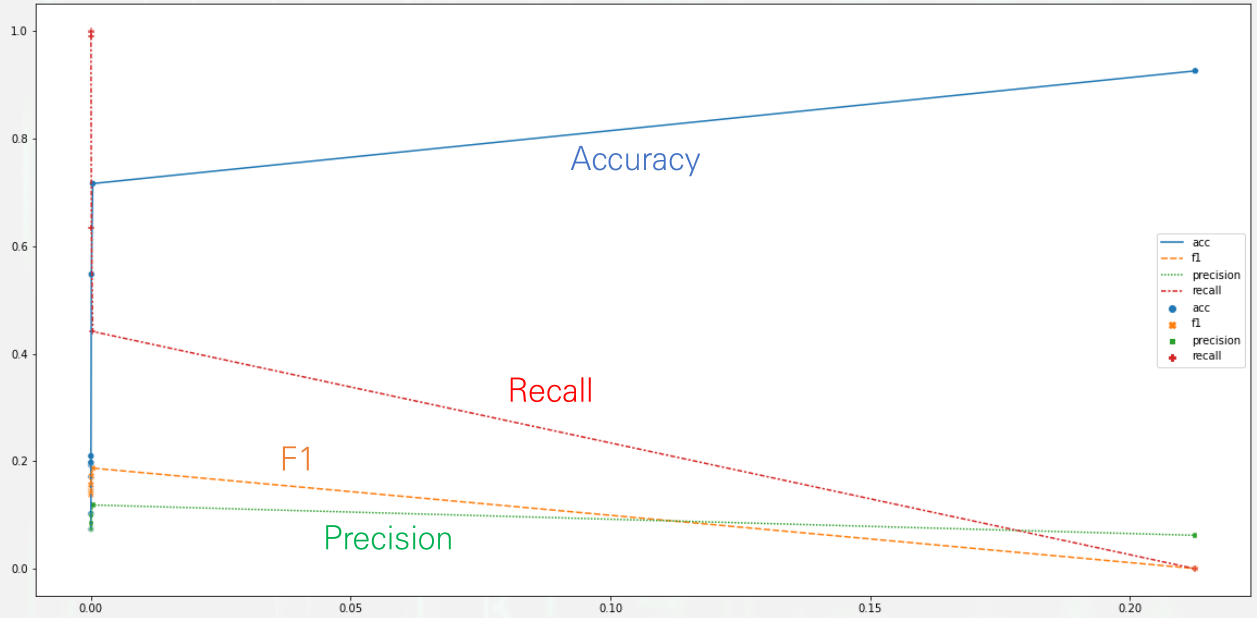
Challenge – UDP, HOP (계속)

- **Feature Selection – Permutation Importance (계속)**
  - HOP LGBM Columns (7개)
  - Base AE 성능 개선 미미

Threshold별 Score 표

	tp	fp	tn	fn	acc	f1	precision	recall
0.0000000	2534	31631	1	0	0.0741966	0.1380964	0.0741695	1.0000000
0.0000066	2527	30666	966	7	0.1022361	0.1414616	0.0761305	0.9972376
0.0000066	2506	29148	2484	28	0.1460516	0.1466011	0.0791685	0.9889503
0.0000066	2506	28293	3339	28	0.1710765	0.1503615	0.0813663	0.9889503
0.0000066	2506	27543	4089	28	0.1930282	0.1538225	0.0833971	0.9889503
0.0000066	2506	27383	4249	28	0.1977112	0.1545816	0.0838436	0.9889503
0.0000066	2506	26969	4663	28	0.2098285	0.1565810	0.0850212	0.9889503
0.0000843	1607	14536	17096	927	0.5474156	0.1720833	0.0995478	0.6341752
0.0003437	1118	8284	23348	1416	0.7160920	0.1873324	0.1189109	0.4411997
0.2126032	1	15	31617	2533	0.9254229	0.0007843	0.0625000	0.0003946

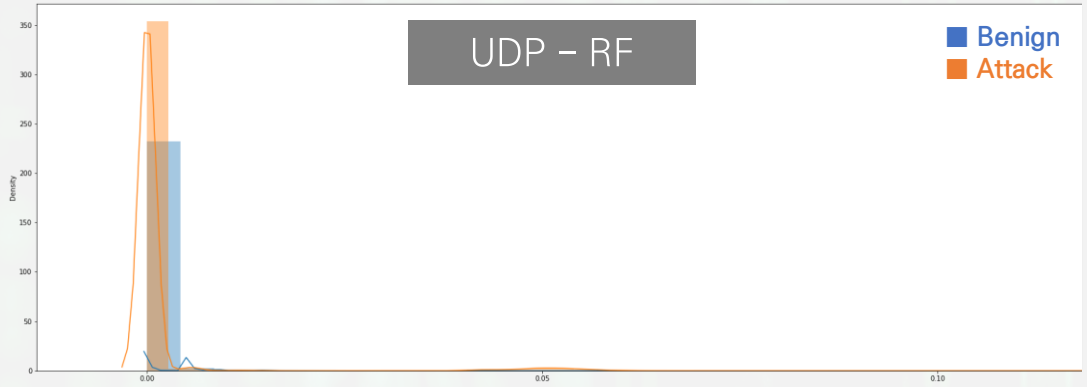
Threshold별 Score 그래프

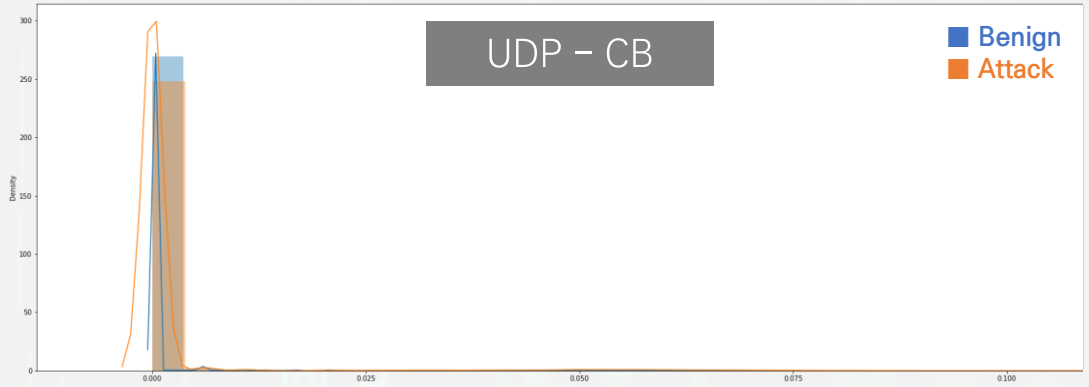


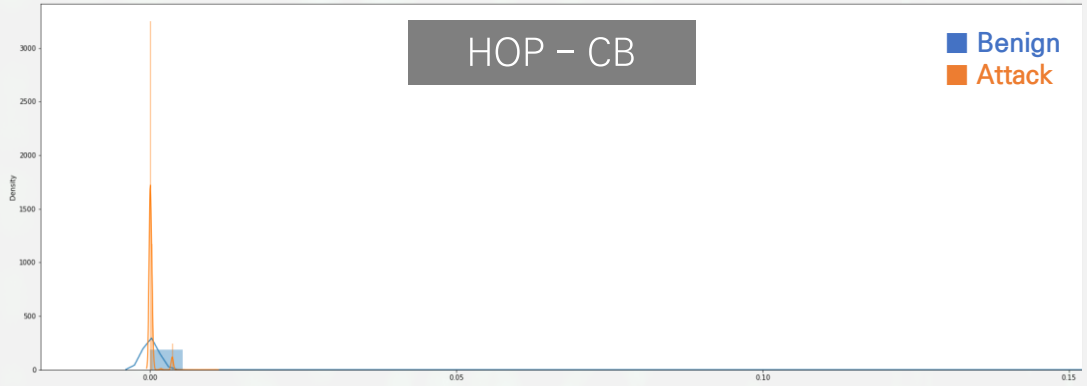
✓ 전체 Feature 리스트는 Appendix 첨부

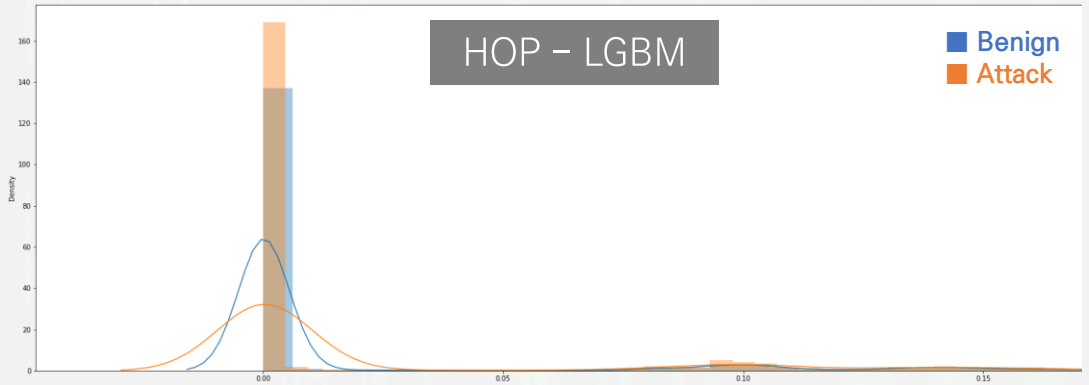
Challenge – UDP, HOP (계속)

- **Feature Selection – Permutation Importance (계속)**
  - Benign과 Attack의 모양이 매우 흡사해서, 모델이 구분 자체를 못하고 있다고 판단됨



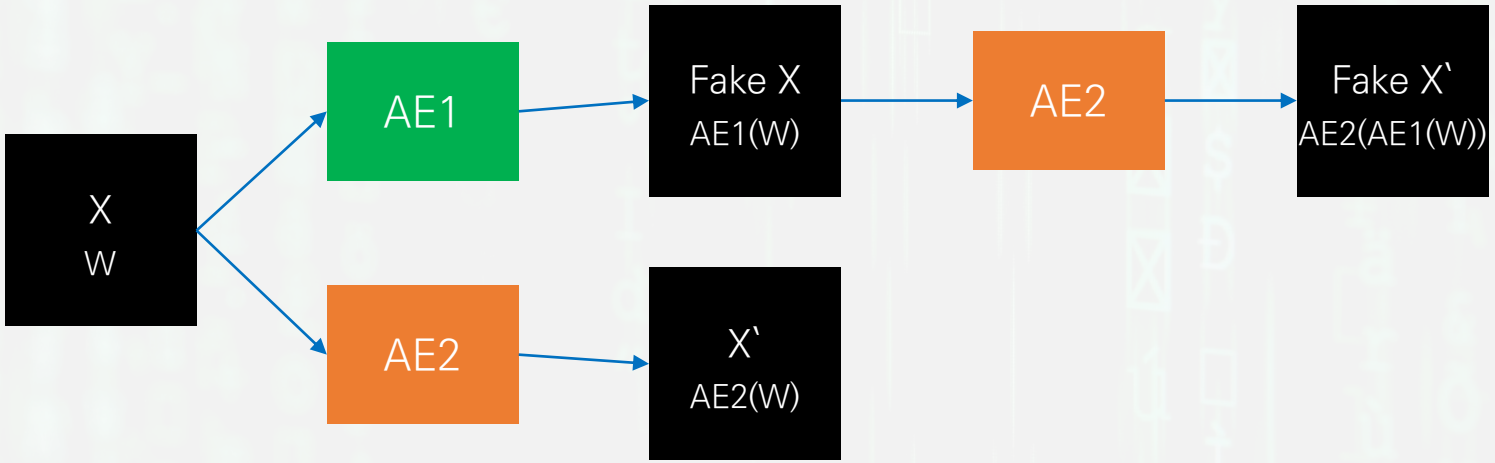






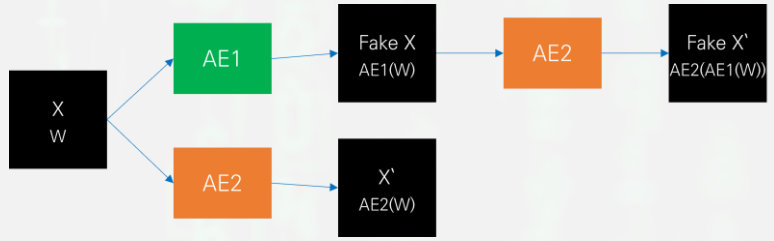
Challenge – UDP, HOP (계속)

- **USAD Model 적용**
  - Real Data의 Abnormal이 **Normal과 유사한 형태를 띄는 경우, AE의 성능이 매우 저하됨**
  - 이를 개선하기 위해서, GAN 모델을 결합하여 Fake Normal 데이터를 생성하여 실제 Normal과 구별하도록 학습
  - **Normal과 유사하지만 실제로는 Abnormal인 데이터를 Normal과 구별**할 수 있도록 고안한 USAD Model 제안



✓ Paper: USAD: UnSupervised Anomaly Detection on Multivariate Time Series

Challenge – UDP, HOP (계속)



- **USAD Model 적용 (계속)**
  - Input을 잘 복원하면서 Fake의 특성과 유사하도록 학습하는 AE1
  - Input을 잘 복원하면서 Fake가 복원한 데이터와 Input을 잘 구분하도록 학습하는 AE2
  - AE1과 AE2는 하나의 Encoder를 공유함

$$\min_{AE_1} \frac{1}{n}\|W_t - AE_1(W_t)\|_2 + \left(1 - \frac{1}{n}\right)\|W_t - AE_2(AE_1(W_t))\|_2$$

$$\min_{AE_2} \frac{1}{n}\|W_t - AE_2(W_t)\|_2 - \left(1 - \frac{1}{n}\right)\|W_t - AE_2(AE_1(W_t))\|_2$$

$n = epoch$

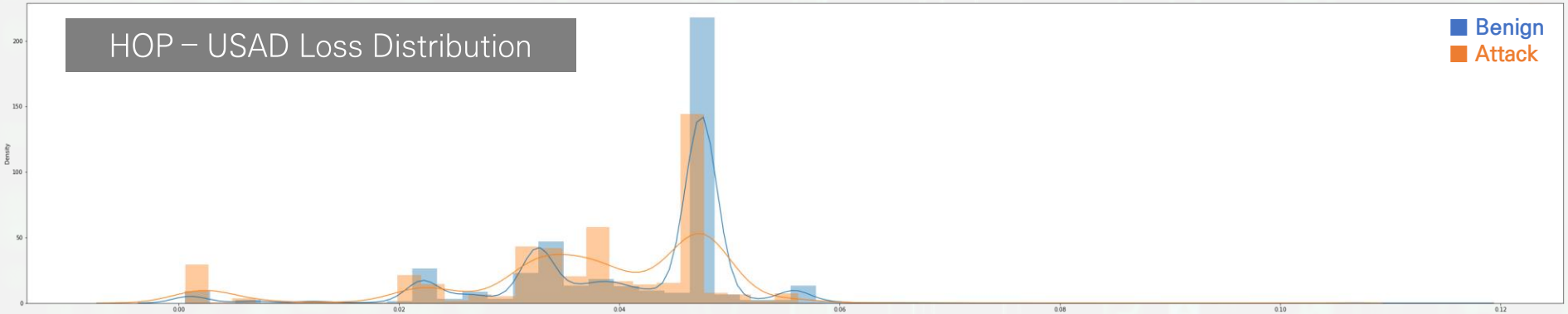
안정적인 학습을 위해 초반에는 reconstruction에 가중치를 주고
후반에는 adversarial training에 가중치를 부여함

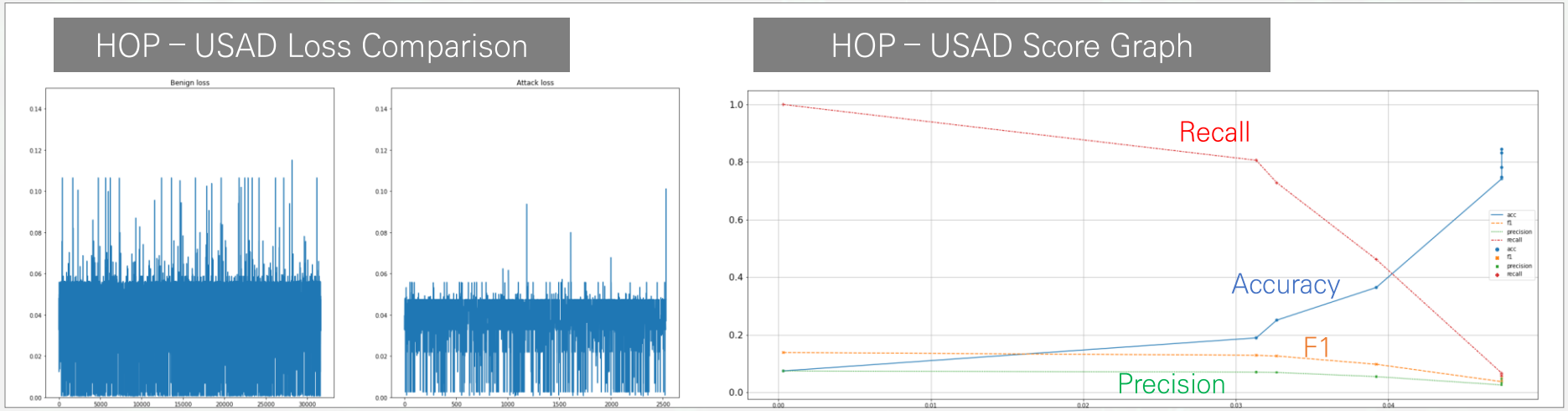
출처: 고려대학교 DSBA

✓ Paper: USAD: UnSupervised Anomaly Detection on Multivariate Time Series

Challenge – UDP, HOP (계속)

- USAD Model 적용 결과 (HOP) : 정상 데이터와 공격 데이터를 구분해내지 못하고 있음





모델 평가 결과 (종합)

- 각 프로토콜별 모델의 Best Performance

구분	TCP			UDP					HOP				
Model	Base	Stacked	Denoising	Base	Base + RF selected	Base + CB selected	Stacked	Denoising	Base	Base + CB selected	Base + LGBM selected	Stacked	Denoising
Threshold	0.0013	0.0063	0.0007	0.00002	0.00005	0.00002	0.00003	0.00003	0.0004	0.00003	0.0003	0.0016	0.00007
Accuracy	*0.7958*	*0.6366*	*0.7356*	*0.7172*	*0.7096*	*0.6427*	*0.7155*	*0.5720*	*0.6218*	*0.6992*	*0.7161*	*0.7243*	*0.6340*
F1 Score	*0.8627*	*0.7524*	*0.8423*	*0.1723*	*0.1832*	*0.1839*	*0.1605*	*0.2070*	*0.1761*	*0.1860*	*0.1873*	*0.1837*	*0.1850*
Precision	0.8237	0.7273	0.7291	0.1461	0.1512	0.1352	0.1369	0.1398	0.1050	0.1163	0.1189	0.1177	0.1108
Recall	0.9056	0.7793	0.9971	0.2100	0.2324	0.2871	0.1940	0.3985	0.5450	0.4633	0.4412	0.4183	0.5600

✓ UDP, HOP 모델 성능 업데이트 예정

Features of Dataset

Flow	'Dst Port', 'Protocol', 'Timestamp', 'Flow Duration', 'Down/Up Ratio', 'Flow Byts/s', 'Flow Pkts/s', 'Fwd Blk Rate Avg', 'Bwd Byts/b Avg', 'Bwd Blk Rate Avg', 'Fwd Act Data Pkts'
Packet	'Tot Fwd Pkts', 'Tot Bwd Pkts', 'TotLen Fwd Pkts', 'TotLen Bwd Pkts', 'Fwd Pkt Len Max', 'Fwd Pkt Len Min', 'Fwd Pkt Len Mean', 'Fwd Pkt Len Std', 'Bwd Pkt Len Max', 'Bwd Pkt Len Min', 'Bwd Pkt Len Mean', 'Bwd Pkt Len Std', 'Fwd Pkts/s', 'Bwd Pkts/s', 'Pkt Len Min', 'Pkt Len Max', 'Pkt Len Mean', 'Pkt Len Std', 'Pkt Len Var', 'Pkt Size Avg', 'Fwd Byts/b Avg', 'Fwd Pkts/b Avg', 'Bwd Pkts/b Avg'
IAT	'Flow IAT Mean', 'Flow IAT Std', 'Flow IAT Max', 'Flow IAT Min', 'Fwd IAT Tot', 'Fwd IAT Mean', 'Fwd IAT Std', 'Fwd IAT Max', 'Fwd IAT Min', 'Bwd IAT Tot', 'Bwd IAT Mean', 'Bwd IAT Std', 'Bwd IAT Max', 'Bwd IAT Min'
Flag	'Fwd PSH Flags', 'Bwd PSH Flags', 'Fwd URG Flags', 'Bwd URG Flags', 'FIN Flag Cnt', 'SYN Flag Cnt', 'RST Flag Cnt', 'PSH Flag Cnt', 'ACK Flag Cnt', 'URG Flag Cnt', 'CWE Flag Count', 'ECE Flag Cnt'
Header	'Fwd Header Len', 'Bwd Header Len'
Segment	'Fwd Seg Size Avg', 'Bwd Seg Size Avg', 'Fwd Seg Size Min'
Subflow	'Subflow Fwd Pkts', 'Subflow Fwd Byts', 'Subflow Bwd Pkts', 'Subflow Bwd Byts'
Window	'Init Fwd Win Byts', 'Init Bwd Win Byts'
Active	'Active Mean', 'Active Std', 'Active Max', 'Active Min'
Idle	'Idle Mean', 'Idle Std', 'Idle Max', 'Idle Min'

UDP, HOP Feature Selection

- **UDP – RandomForestClassifier (30개)**

  ['Pkt Len Mean', 'Pkt Len Var' ,'Pkt Len Std' ,'Bwd Pkt Len Min' ,'Pkt Len Max' ,'Fwd Pkt Len Min', 'Bwd Pkt Len Max' , 'Pkt Size Avg', 'Bwd Seg Size Avg', 'Bwd Pkt Len Mean', 'Pkt Len Min', 'Bwd IAT Mean' ,'Fwd IAT Min', 'Subflow Bwd Byts', 'TotLen Bwd Pkts', 'Fwd IAT Tot', 'Subflow Bwd Pkts' ,'Fwd IAT Mean', 'Fwd Pkt Len Max' ,'Flow IAT Std' ,'Fwd Pkt Len Mean' ,'Subflow Fwd Pkts' ,'Bwd Header Len' ,'Fwd Header Len', 'Fwd Pkt Len Std' ,'Down/Up Ratio' ,'Tot Bwd Pkts', 'Fwd Act Data Pkts', 'Active Min' ,'Bwd IAT Std']

- **UDP – CatBoostClassifier (49개)**

  ['Pkt Len Std', 'Pkt Len Mean', 'Pkt Size Avg', 'Pkt Len Var', 'Bwd Pkt Len Min', 'Pkt Len Max', 'Bwd Pkt Len Max' ,'Flow IAT Min', 'TotLen Bwd Pkts' ,'Fwd Pkt Len Max' ,'Fwd Pkt Len Min' ,'Bwd Pkt Len Mean', 'TotLen Fwd Pkts', 'Fwd Pkt Len Mean', 'Flow Byts/s', 'Flow IAT Mean', 'Bwd Pkts/s', 'Bwd Seg Size Avg' ,'Fwd Seg Size Avg', 'Subflow Bwd Byts', 'Flow Duration', 'Fwd IAT Min', 'Pkt Len Min', 'Fwd IAT Mean', 'Flow IAT Std', 'Fwd IAT Std', 'Bwd IAT Max', 'Subflow Fwd Byts', 'Fwd IAT Tot', 'Fwd Pkts/s', 'Bwd IAT Min', 'Idle Mean', 'Fwd IAT Max', 'Flow Pkts/s', 'Active Mean', 'Flow IAT Max', 'Idle Max', 'Bwd IAT Tot', 'Active Max', 'Active Min', 'Bwd IAT Mean', 'Idle Std', 'Idle Min', 'Tot Fwd Pkts', 'Fwd Act Data Pkts', 'Down/Up Ratio', 'Subflow Fwd Pkts', 'Bwd IAT Std', 'Fwd Header Len']

UDP, HOP Feature Selection

- **HOP – LGBMClassifier (7개)**

  ['Active Max', 'Active Std', 'Active Mean', 'Bwd IAT Std', 'Idle Max', 'Bwd IAT Max', 'Bwd IAT Mean']

- **HOP – CatBoostClassifier (2개)**

  ['Subflow Fwd Pkts', 'Tot Bwd Pkts']

Reference

- Amer Abdulmajeed Abdulrahman & Mahmood Khalel Ibrahem. (2020). Toward Constructing a Balanced Intrusion Detection Dataset
- Arash Habibi Lashkari & Andi Fitriah A.Kadir & Hugo Gonzalez & Kenneth Fon Mbah & Ali A. Ghorbani. (2017). Towards a Network Based Framework for Android Malware Detection and Characterization. Canadian Institute for Cybersecurity (CIC), UNB, Canada
- Byeoungjun Min & Jihoon Yoo & Sangsoo Kim & Dongil Shin & Dongkyoo Shin. (2021). Network Intrusion Detection with One Class Anomaly Detection Model based on Auto Encoder. Journal of Internet Computing and Services(JICS) 2021. Feb.: 22(1): 13-22
- Chaitanya Buragohain & Manash Jyoti Kalita & Santosh Singh & Dhruba K. Bhattacharyya. (2015). Anomaly based DDoS Attack Detection. International Journal of Computer Applications (0975 – 8887) Volume 123 – No.17, August 2015

Reference (계속)

- Chongzhen Zhang & Yanli Chen & YangMeng & Fangming Ruan & Runze Chen & Yidan Li & Yaru Yang. (2021). A Novel Framework Design of Network Intrusion Detection. Hindawi Security and Communication Networks Volume 2021, p15
- Giovanni Apruzzese & Mauro Andreolini & Michele Colajanni & Mirco Marchetti. (2019). Hardening Random Forest Cyber Detectors Against Adversarial Attacks. IEEE
- Guansong Pang & Longbing Cao & Charu Aggarwal. (2021). Deep Learning for Anomaly Detection_ Challenges, Methods, and Opportunities
- Koohong Kang. (2020). Network Anomaly Detection Technologies Using Unsupervised Learning AutoEncoders. Journal of The Korea Institute of Information Security & Cryptology

Reference (계속)

- M Odusami & S Misra & E Adetiba & O Abayomi-Alli & R Damasevicius & R Ahuja. (2019). An Improved Model for Alleviating Layer Seven Distributed Denial of Service Intrusion On Webserver. The 3rd International Conference on Computing and Applied Informatics 2018
- Matthew Yung & Eli T. Brown & Alexander Rasin & Jacob D. Furst & Daniela S. Raicu. (2018). Synthetic Sampling for Multi-Class Malignancy Prediction. KDD MLMH'18, August 2018, London, United Kingdom
- Mohammed Gharib & Bahram Mohammadi & Shadi Hejareh Dastgerdi & Mohammad Sabokrou. (2019). AutoIDS_ Auto-encoder Based Method for Instrusion Detection System
- Nitin Mathur. (2020). Application of Autoencoder Ensembles in Anomaly and Intrusion Detection using Time-Based Analysis

Reference (계속)

- Nicholas Bashour. (2021). Identifying Network Intrusions Using Deep Learning
- Raghavendra Chalapathy & Sanjay Chawla. (2019). DEEP LEARNING FOR ANOMALY DETECTION: A SURVEY
- Razan Abdulhammed & Hassan Musafer & Ali Alessa & Miad Faezipour & Abdelshakour Abuzneid. (2019). Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection. Electronics 2019. MPDI Journal
- Saman Sarraf. (2020). Analysis and Detection of DDoS Attacks Using Machine Learning Techniques. American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS) (2020) Volume 66, No 1, pp 95-104
- Taher Al-Shehari & Rakan A. Alsowail. (2021). An Insider data Leakage Detection Using One-Hot Encoding, Synthetic Minority Oversampling and Machine. Entropy 2021. MPDI Journal


끝.