Intrusion Detection Project

2021.04.05 ~ 05.15

송주환 (Joowhan Song)
juansong.77@gmail.com
82-10-6256-7540

Project overview

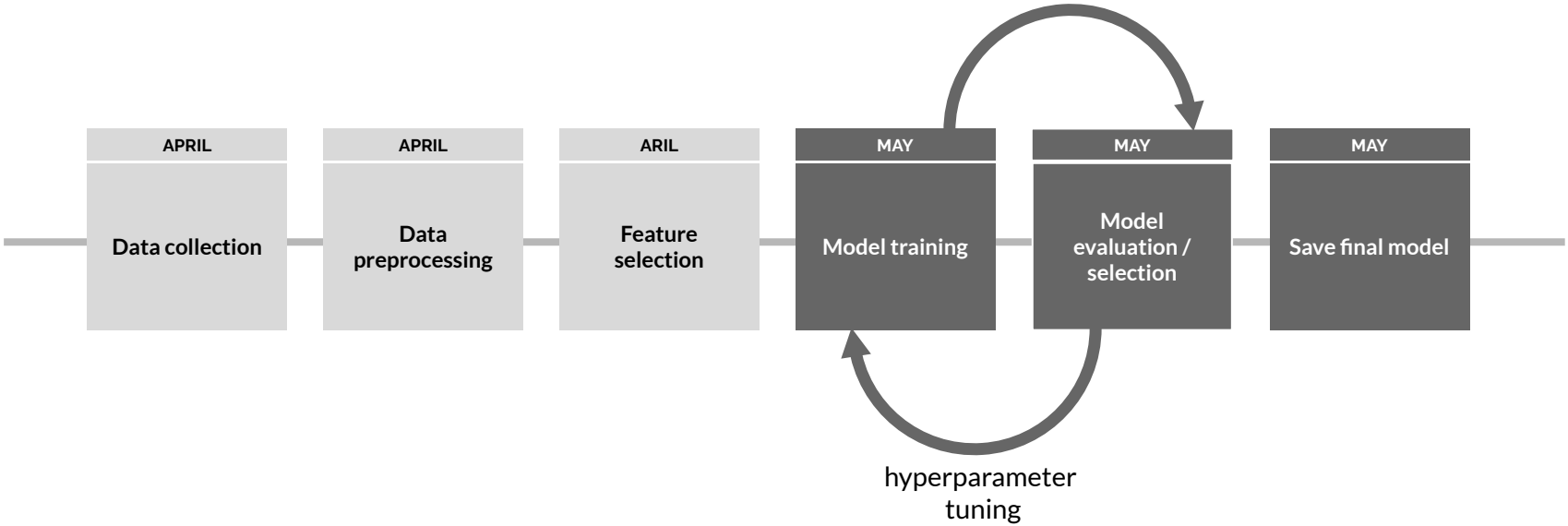
Objective

네트워크 패킷데이터 분석부터 공격 탐지 모델링까지의 Pipeline 구현

Development

- Source code: https://github.com/juansong/intrusion_detection.git
- Written in Python (Jupyter Notebook)
- Data preprocessing (pandas, Numpy, matplotlib, seaborn)
- Feature selection (scikit-learn)
- Model training (scikit-learn)
- Evaluation (scikit-learn, matplotlib)

Timeline


APRIL
Data collection
APRIL
Data preprocessing
ARIL
Feature selection
MAY
Model training
MAY
Model evaluation / selection
MAY
Save final model
hyperparameter tuning

Data Preprocessing

- Dataset : CICIDS 2017 (2,830,743 x 79)
- Missing value (NaN, Inf, -Inf), duplicate 제거
- BENIGN(정상) 상태로만 측정된 요일 제거
- 필요없는 column 제거
- 데이터 용량 간소화 & 연산속도 증가를 위해 데이터 타입 축소
- SMOTE를 이용한 undersampling (handling imbalanced data)
- StandardScaler()를 이용한 standardization

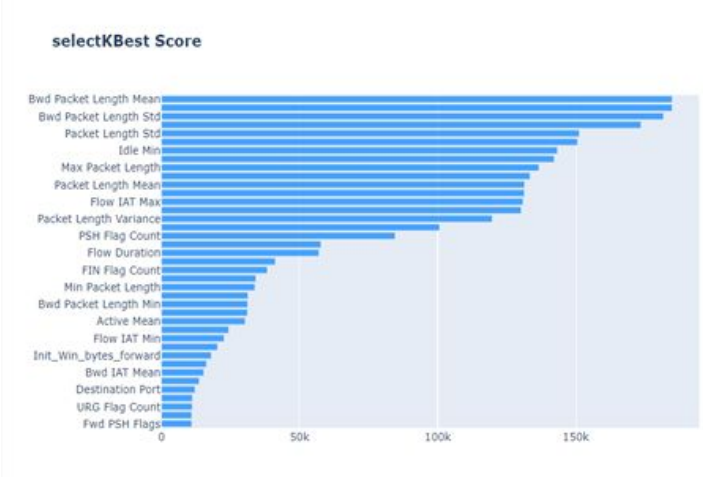
Raw 데이터의 종속변수(Y) 빈도수 분포

BENIGN	2273097
DoS Hulk	231073
PortScan	158930
DDoS	128027
DoS GoldenEye	10293
FTP-Patator	7938
SSH-Patator	5897
DoS slowloris	5796
DoS Slowhttptest	5499
Bot	1966
Web Attack Brute Force	1507
Web Attack XSS	652
Infiltration	36
Web Attack Sql Injection	21
Heartbleed	11

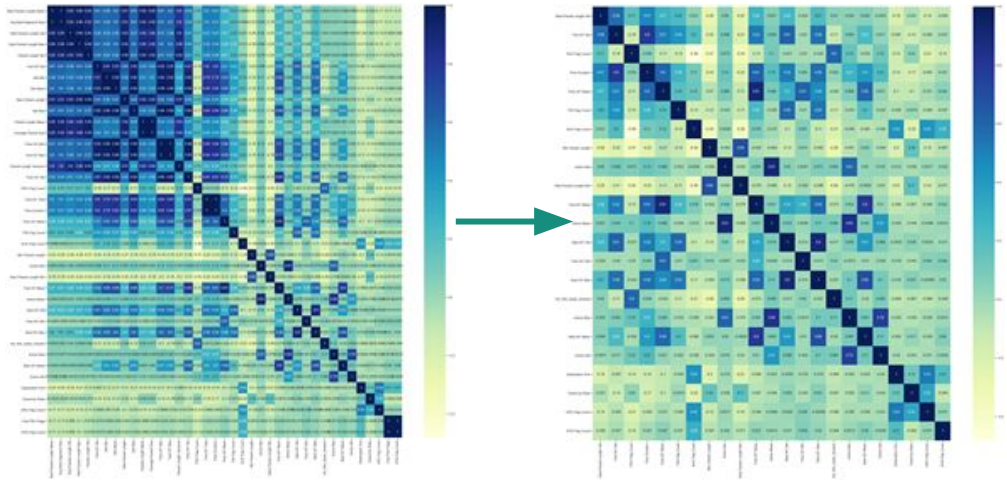
Name: Label, dtype: int64

Feature Selection

- SelectKBest로 상위 39개 f-value score 확인

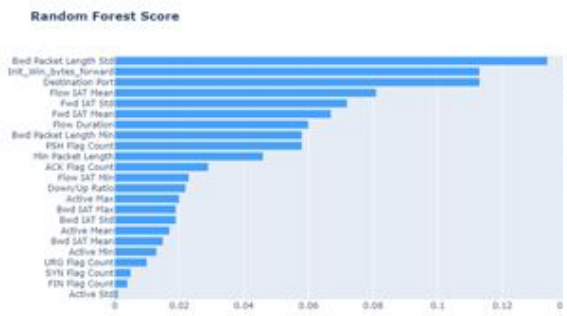


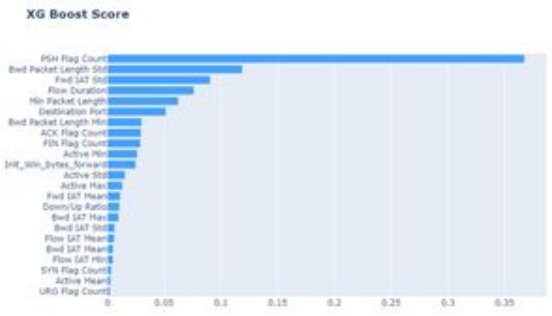
- Correlation Coefficient 비교를 위한 heatmap (16개 제거)



Feature Selection

- 상관계수로 feature 제거 이후, 23개 Feature만을 사용하여 model별 feature importance plot
- 최종으로 12개 feature 선별





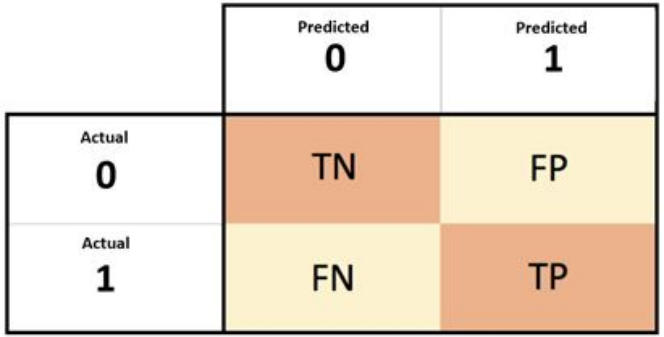
Model Training

머신러닝 모델 학습 (SVM, Decision Tree, Random Forest, Gradient Boosting, XGBoost)

모델 평가 기준

정상상태(BENIGN)가 아닌 것을 맞다고 판단할 수 있는가?

- 데이터의 특성 고려 (imbalanced data)
- FN(공격을 정상상태로 판단) 여부가 중요
- Precision / Recall(정밀도 / 재현율)이 중요
- Precision-recall tradeoff 발생하므로 f1-score로 모델 평가



Evaluation

SVM

Classification report:

	precision	recall	f1-score	support
0	0.86	0.92	0.89	60
1	1.00	1.00	1.00	641
2	0.99	1.00	1.00	1768
3	1.00	0.83	0.91	103
4	1.00	1.00	1.00	2966
5	0.67	1.00	0.80	495
6	0.81	0.60	0.69	58
7	0.89	0.07	0.13	240
accuracy			0.96	6331
macro avg	0.90	0.80	0.80	6331
weighted avg	0.96	0.96	0.94	6331

Random Forest

Classification report:

	precision	recall	f1-score	support
0	1.00	0.98	0.99	60
1	1.00	1.00	1.00	641
2	1.00	1.00	1.00	1768
3	0.99	0.99	0.99	103
4	1.00	1.00	1.00	2966
5	0.73	0.83	0.78	495
6	0.97	0.98	0.97	58
7	0.51	0.36	0.42	240
accuracy			0.96	6331
macro avg	0.90	0.89	0.89	6331
weighted avg	0.96	0.96	0.96	6331

XGBoost

Classification report:

	precision	recall	f1-score	support
0	1.00	0.98	0.99	60
1	1.00	1.00	1.00	641
2	1.00	1.00	1.00	1768
3	1.00	0.99	1.00	103
4	1.00	1.00	1.00	2966
5	0.74	0.81	0.78	495
6	0.97	0.97	0.97	58
7	0.51	0.41	0.45	240
accuracy			0.96	6331
macro avg	0.90	0.89	0.90	6331
weighted avg	0.96	0.96	0.96	6331

Evaluation

Ensemble model (5개의 classifier를 hard vote)

Random Forest Classifier
(max_depth = 40)

96% Accuracy 89% f1-score

```
Model Accuracy:
 0.9624072026536092

Confusion matrix:
 [[  59    1    0    0    0    0    0    0]
 [   0  641    0    0    0    0    0    0]
 [   0    0 1767    1    0    0    0    0]
 [   0    1    0  102    0    0    0    0]
 [   0    0    0    0 2966    0    0    0]
 [   0    0    0    0    0  414    1   80]
 [   0    0    0    0    0    1   57    0]
 [   0    0    0    0    1  149    3   87]]
```

Classification report:

	precision	recall	f1-score	support
0	1.00	0.98	0.99	60
1	1.00	1.00	1.00	641
2	1.00	1.00	1.00	1768
3	0.99	0.99	0.99	103
4	1.00	1.00	1.00	2966
5	0.73	0.84	0.78	495
6	0.93	0.98	0.96	58
7	0.52	0.36	0.43	240
accuracy			0.96	6331
macro avg	0.90	0.89	0.89	6331
weighted avg	0.96	0.96	0.96	6331

The End