2021.03.04~03.24

KISA DATA CHALLENGE

손수민
강주희

DATA PREPROCESSING

Time 속성 전처리

```
import datetime
```

각 csv 파일마다 time의 단위를 utc time을 밀리초 단위로 바꿈.

```
i=9
normal[i]['_ws.col.UTCtime']=pd.to_datetime(normal[i]['_ws.col.UTCtime'])
normal[i]['_ws.col.UTCtime']=normal[i]['_ws.col.UTCtime'].astype(np.int64)// 10**9
normal[i]['_ws.col.UTCtime']=normal[i]['_ws.col.UTCtime']-normal[i].iloc[0,0]
```

한 엑셀파일당 처음 밀리초를 0부터 시작하게 만듦.

```
normal[i]=normal[i].loc[:,"_ws.col.UTCtime": "tcp.ack"] #열자르기
normal[i] = normal[i].dropna(axis=0)
```

```
normal[i]
```

```
normal[i].to_csv("normal"+str(i)+".csv", mode='w',index = False)
```

DATA PREPROCESSING

Groupset 생성

In [7]:
```
#normal data를 그룹핑하고 100개가 넘는 그룹을 골라내기
groupset=[]
over_100_group=[]
for i in range(len(readdata.normal)):
    gb = readdata.normal[i].groupby(['_ws.col.Protocol','ip.src','ip.dst','tcp.srcport'])
    for key, group in gb:
        group = np.asarray(group)
        if len(group)>100:
            over_100_group.append(group)
        else :
            groupset.append(group)
```

In [8]:
```
for i in range(len(readdata.normal)):
    gb = readdata.normal[i].groupby(['_ws.col.Protocol','ip.src','ip.dst','tcp.dstport'])
    for key, group in gb:
        group = np.asarray(group)
        if len(group)>100:
            over_100_group.append(group)
        else :
            groupset.append(group)
```

In [9]:
```
#normal data 100개가 넘는 그룹 100개씩 잘라서 그룹셋에 넣기
for i in range(len(over_100_group)):
    for j in range(0,len(over_100_group[i]),100):
        groupset.append(over_100_group[i][j:j+100])
```

In [10]:
```
len(groupset)
```

Out[10]: 1339219

1. Protocol, ip.src/dst, tcp.srcport
2. Protocol, ip.src/dst, dstport

데이터의 연속성을 반영한 그룹화 방식 적용.

DATA PREPROCESSING

배열 X 생성.

```
X=[]
for i in range(len(groupset)):
    temp=np.delete(groupset[i],[1,2,3,4,5],1)
    num=100-len(temp)
    X.append(np.pad(temp,((0,num),(0,0)),'constant', constant_values=-1))
```

각 groupset들의 헤더 제거

100-해당 그룹의 패킷 개수

-1로 패딩

X_data 로 변환.

```
X_data=np.asarray(X)
```

```
len(X_data)
```

```
1339219
```

Y_data (라벨)생성.

```
Y_data=[]
for i in range(len(X_data)):
    Y_data.append(0)
```

Normal이므로 '0'

```
len(Y_data)
```

```
1339219
```

DATA PREPROCESSING


```
groupset_a
```

```
[array([[3, 'FTP', '172.16.0.1', '192.168.10.50', 52108, 21, 14, 1, 21],
        [4, 'FTP', '172.16.0.1', '192.168.10.50', 52108, 21, 11, 15, 55],
        [4, 'FTP', '172.16.0.1', '192.168.10.50', 52108, 21, 6, 26, 78]],
       dtype=object),
 array([[87, 'FTP', '172.16.0.1', '192.168.10.50', 52112, 21, 14, 1, 21],
        [87, 'FTP', '172.16.0.1', '192.168.10.50', 52112, 21, 17, 15, 55],
        [91, 'FTP', '172.16.0.1', '192.168.10.50', 52112, 21, 14, 32, 77],
        [91, 'FTP', '172.16.0.1', '192.168.10.50', 52112, 21, 26, 46, 111],
        [94, 'FTP', '172.16.0.1', '192.168.10.50', 52112, 21, 14, 72, 133],
        [94, 'FTP', '172.16.0.1', '192.168.10.50', 52112, 21, 20, 86, 167],
        [97, 'FTP', '172.16.0.1', '192.168.10.50', 52112, 21, 14, 106,
         190]], dtype=object),
 array([[87, 'FTP', '172.16.0.1', '192.168.10.50', 52114, 21, 14, 1, 21],
        [87, 'FTP', '172.16.0.1', '192.168.10.50', 52114, 21, 20, 15, 55],
        [91, 'FTP', '172.16.0.1', '192.168.10.50', 52114, 21, 14, 35, 77],
        [91, 'FTP', '172.16.0.1', '192.168.10.50', 52114, 21, 17, 49, 111],
        [94, 'FTP', '172.16.0.1', '192.168.10.50', 52114, 21, 14, 66, 133],
        [94, 'FTP', '172.16.0.1', '192.168.10.50', 52114, 21, 17, 80, 167],
        [97, 'FTP', '172.16.0.1', '192.168.10.50', 52114, 21, 14, 97, 190]],
```

DATA PREPROCESSING

X_a 배열 생성

```
X_a=[]
for i in range(len(groupset_a)):
    temp=np.delete(groupset_a[i],[1,2,3,4,5],1)
    num=100-len(temp)
    X_a.append(np.pad(temp,((0,num),(0,0)),'constant', constant_values=-1))
```

Y_a 배열 생성

```
Y_a=[]
for i in range(len(X_a)):
    Y_a.append(1)
```

X_attack_data, Y_attack_data 생성

```
X_attack_data=np.asarray(X_a)
Y_attack_data=np.asarray(Y_a)
```

X_data

Y_data

DATA PREPROCESSING

X_total 생성

```
X_total=np.concatenate((X_data,X_attack_data), axis=0)
#index : 0~891936 까지 normal, 총 891937
```

```
X_total.shape
```

```
(1339578, 100, 4)
```

Y_total 생성

```
Y_total=np.concatenate((Y_data,Y_attack_data), axis=0)
```

```
Y_total.shape
```

```
(1339578,)
```

MODEL

개선한 점

1. 교차검증 사용
2. Train_test_split 함수대신 StratifiedKFold를 사용하여 데이터를 k개로 분할
3. Threshold Moving 삭제
4. 미리 smote를 적용하는 대신 fold당 smote를 새로 적용함
5. Epoch을 늘리고 lr을 줄이는 방식으로 최적의 모델을 찾아감

MODEL

개선한 점

```
X_train, X_test, Y_train, Y_test = train_test_split(X_total,Y_total, test_size=0.2, shuffle=True, stratify=Y_total, ran
X_val, X_test,Y_val,Y_test = train_test_split(X_test,Y_test, test_size=0.5, shuffle=True, stratify=Y_test, random_state
```

```
print(X_train.shape,Y_train.shape,X_test.shape,Y_test.shape)
```

```
(1071643, 100, 4) (1071643,) (133956, 100, 4) (133956,)
```

```
skf = StratifiedKFold(n_splits=10)
k=10
```

```
for train_index, test_index in skf.split(X_total,Y_total):
    #print('train_index : ', train_index,'₩ntest_index : ',test_index)
    X_train = X_total[train_index]
    y_train = Y_total[train_index]
    X_test = X_total[test_index]
    y_test = Y_total[test_index]
```

Train_test_split 함수대신 StratifiedKFold를 사용하여 데이터를 k개로 분할

MODEL


```
for train_index, test_index in skf.split(X_total,Y_total):
    #print('train_index : ', train_index,'₩ntest_index : ',test_index)
    X_train = X_total[train_index]
    y_train = Y_total[train_index]
```

K번 반복되는 반복문

Skf.split으로 각 fold마다 분류형한 데이터를

```
def build_model():
    learning_rate = 0.00001
    seq_length = 100
    data_dim = 4
    METRICS = [
      tf.keras.metrics.BinaryAccuracy(name='accuracy')
    ]
    model = Sequential()
    model.add(Masking(mask_value=-1.,input_shape=(100, 4)))
    model.add(Bidirectional(LSTM(128, kernel_regularizer='l2', input_shape=(100,4))))

    model.add(Dense(128, activation='relu', kernel_regularizer='l2'))
    model.add(Dense(1, activation='sigmoid', kernel_regularizer='l2'))

    model.compile(loss='binary_crossentropy', optimizer=tf.keras.optimizers.Adam(lr=learning_rate), metrics=METRICS)


    return model
```

있게 만들어줌

으

```
    matrix.append(confusion_matrix(y_test, y_pred_binary))

    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_binary).ravel()
    print(f'Accuracy: {acc}')
    print(f'f-score: {f1}')
    print(f'tn fp fn tp :{tn, fp, fn, tp}')
```

Aa 이름	k	epoch	lr	acc	f1	tn,fp,fn,,tp	맞춘개수 (88개중)
test_cross_new_10_25_00001 열기	10	25	0.00001	0.999236327731792	0.5531475704859814	133462 460 0 36 133857 65 0 36 133863 59 1 35 133771 151 0 36 133792 130 0 36 133842 80 0 36 133890 32 3 33 133893 29 0 36 133920 2 0 35 133917 4 7 29	204→192개 58/88
test_cross_new_10_40_00001	10	40	0.00001	0.9992602158049715	0.5455613296375394	133515 407 0 36 133830 92 0 36 133862 60 1 35 133777 145 0 36 133802 120 7 29 133832 90 0 36 133896 26 0 36 133894 28 0 36 133919 3 0 35 133916 5 7 29	145개→129개 54/88
test_cross_new_10_60_00001	10	60	0.00001	0.9994020512945054	0.5775120893108479	133581 341 0 36 133855 67 0 36 133875 47 1 35 133792 130 0 36 133836 86 7 29 133874 48 0 36 133887 35 7 29 133907 15 0 36 133920 2 0 35 133917 4 11 25	150개—>146개 64/88
test_cross_new_10_80_00001	10	80	0.00001	0.9994438554309287	0.5941669329991256	133635 287 0 36 133855 67 0 36 133873 49 1 35 133815 107 0 36 133800 122 7 29 133873 49 0 36 133900 22 4 32 133911 11 0 36 133919 3 0 35 133910 11 5 31	205개→196개 53/88

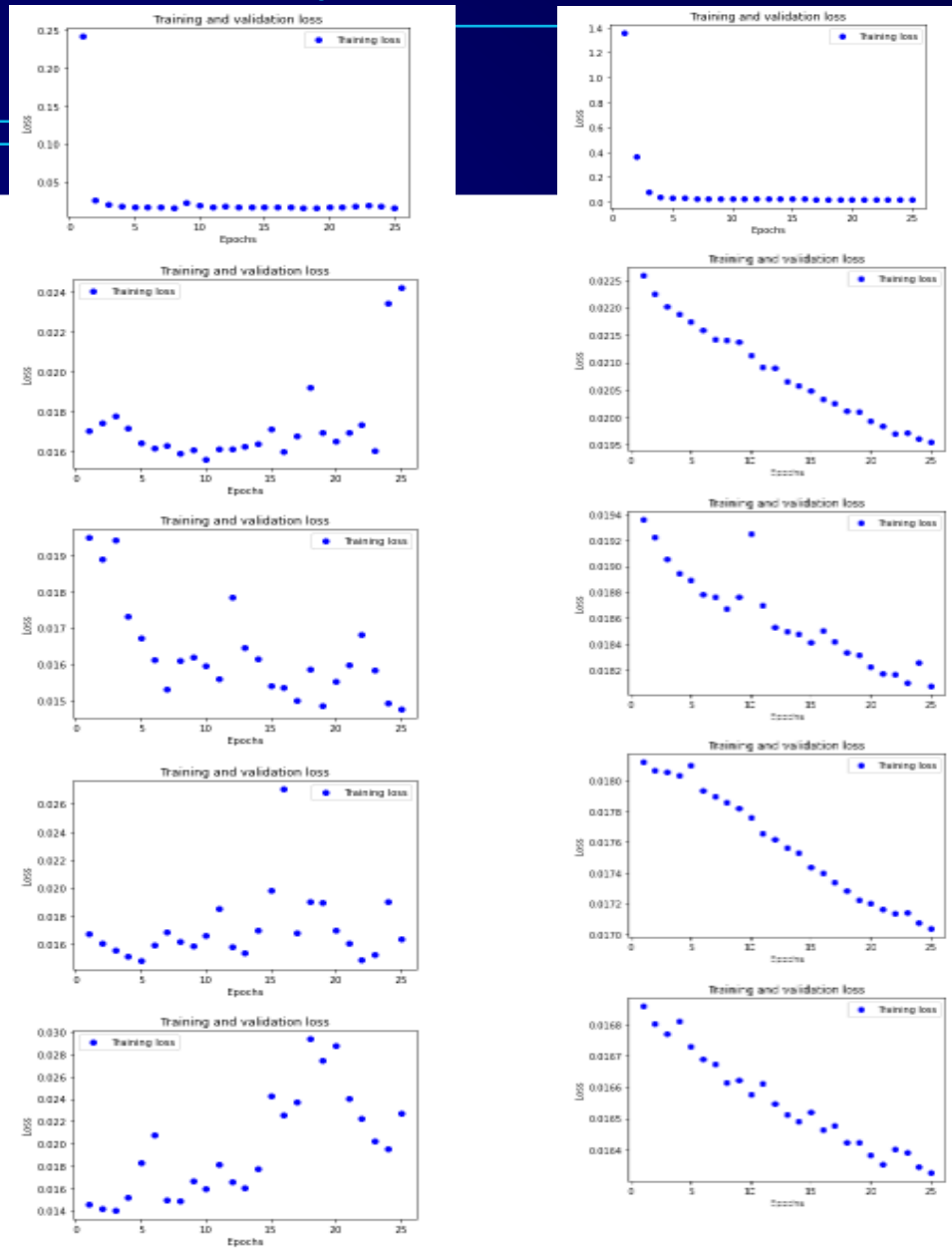
MODEL

Learning rate : 0.0001 vs 0.00001

test_cross_new_10_25와
test_cross_new_10_25_00001

⇒ **0.00001로 lr을 줄이기로 결정**

⇒ **0.000001로 더 줄여 시도해봤지만 더 결과가 나빠져 시도X**



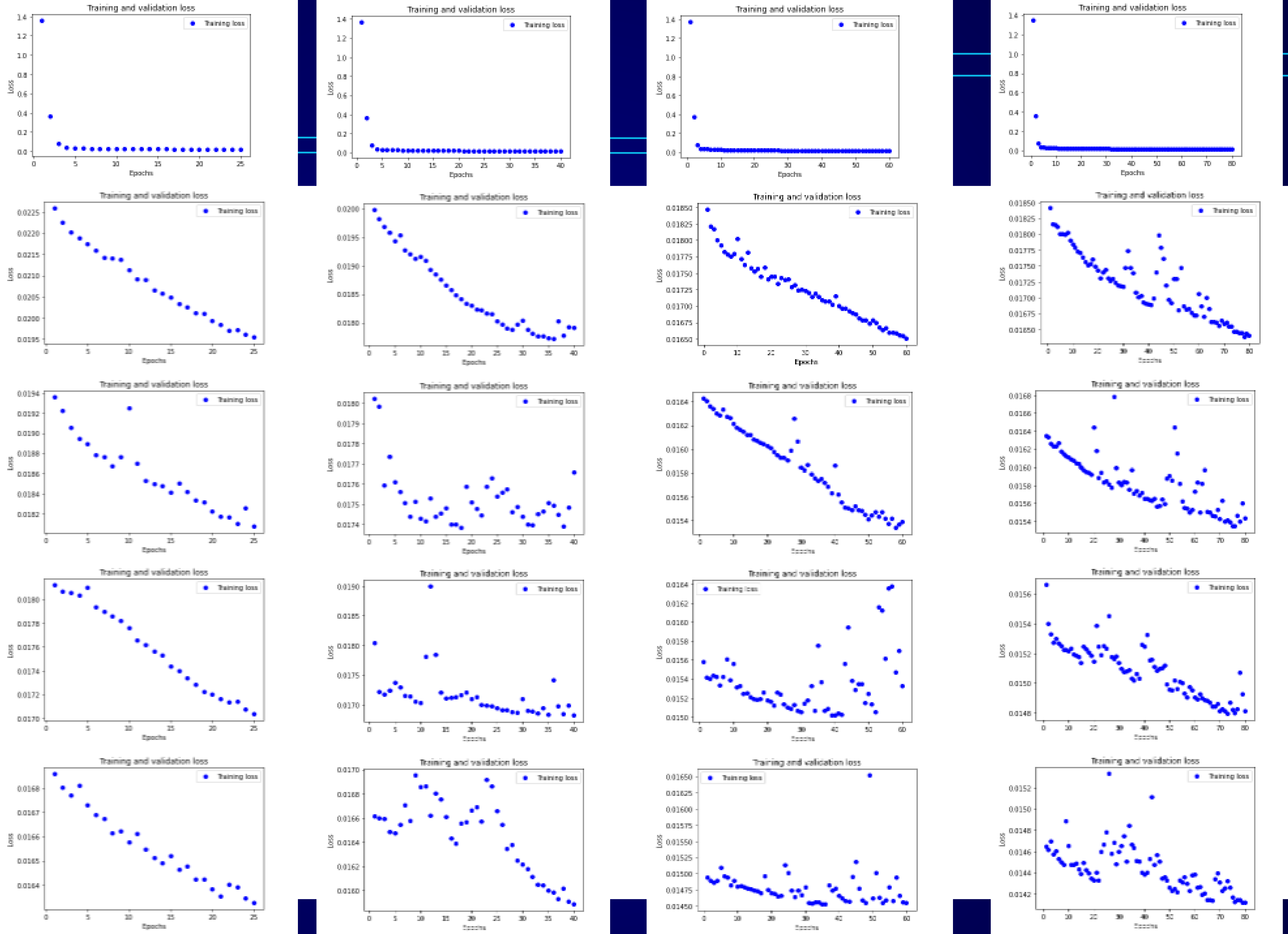
Page
13

DATA PREPROCESSING

MODEL

RESULT

Need to improvement



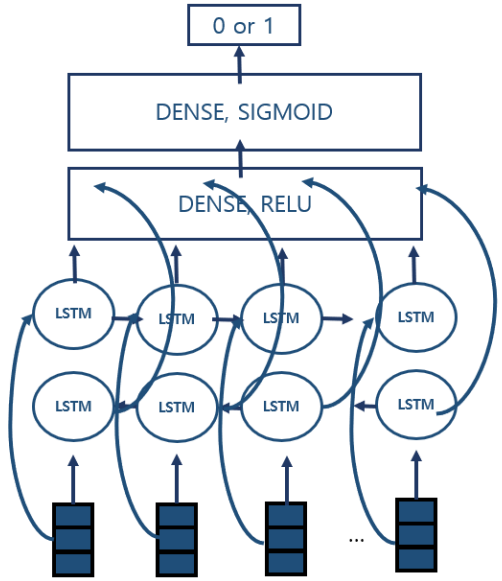
이름	k	epoch	lr	acc	f1	tn,fp,fn,,tp	맞춘개수 (88개중)
test_cross_new_10_25_00001 열기	10	25	0.00001	0.999236327731792	0.5531475704859814	133462 460 0 36 133857 65 0 36 133863 59 1 35 133771 151 0 36 133792 130 0 36 133842 80 0 36 133890 32 3 33 133893 29 0 36 133920 2 0 35 133917 4 7 29	204→192개 58/88
test_cross_new_10_40_00001	10	40	0.00001	0.9992602158049715	0.5455613296375394	133515 407 0 36 133830 92 0 36 133862 60 1 35 133777 145 0 36 133802 120 7 29 133832 90 0 36 133896 26 0 36 133894 28 0 36 133919 3 0 35 133916 5 7 29	145개→129개 54/88
test_cross_new_10_60_00001	10	60	0.00001	0.9994020512945054	0.5775120893108479	133581 341 0 36 133855 67 0 36 133875 47 1 35 133792 130 0 36 133836 86 7 29 133874 48 0 36 133887 35 7 29 133907 15 0 36 133920 2 0 35 133917 4 11 25	150개—>146개 64/88
test_cross_new_10_80_00001	10	80	0.00001	0.9994438554309287	0.5941669329991256	133635 287 0 36 133855 67 0 36 133873 49 1 35 133815 107 0 36 133800 122 7 29 133873 49 0 36 133900 22 4 32 133911 11 0 36 133919 3 0 35 133910 11 5 31	205개→196개 53/88

MODEL

최종 모델

평균 F1점수(0.5941669329991256)가 제일 높

은test_cross_new_10_80_00001로 선택



총196개

88개중에 53개를 맞춤

```
correct
```

```
[array(['172.16.0.1', '192.168.10.50', 52278, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52278, 21, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52280, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52280, 21, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52282, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52282, 21, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52258, 6], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52278, 6], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52280, 6], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52282, 6], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52278, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52280, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52282, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52284, 6], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52284, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52284, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52284, 21, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52286, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52286, 21, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52288, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52288, 21, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52290, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52290, 21, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52286, 6], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52286, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52288, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52290, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52292, 6], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52292, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52294, 6], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52294, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52296, 6], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52296, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52292, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52292, 21, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52294, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52294, 21, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52296, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52296, 21, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52298, 6], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52298, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52300, 6], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52300, 20], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52302, 6], dtype=object),
 array(['192.168.10.50', '172.16.0.1', 21, 52302, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52298, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52298, 21, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52300, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52300, 21, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52302, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52302, 21, 20], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52304, 21, 6], dtype=object),
 array(['172.16.0.1', '192.168.10.50', 52304, 21, 20], dtype=object)]
```

```
len(correct)
```

```
53
```

Need to improvement

- test_cross_new_10_100_00001을 돌리던 중 계속 중간에 멈추는 현상 발생으로 끝까지 돌리지 못한 점이 아쉬움
- 더 높은 정확도를 가지기 위해서는 새로운 방법이 필요

2021.03.04~24

Thank you

손수민
강주희