

In [1]:

```
!pip install liac-arff
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
import arff as arf
```

Requirement already satisfied: liac-arff in c:\users\ajay\anaconda3\envs\tf\lib\site-packages (2.5.0)

In [2]:

```
from json import decoder

file=open("C:/Ajay/documents/Ddos/final-dataset.arff")
decoder=arf.ArffDecoder()
data=decoder.decode(file, encode_nominal=True)
```

In [3]:

```
vals=[val[0:-1] for val in data['data']]
labels=[lab[-1] for lab in data['data']]
```

In [4]:

```
da=set(labels)
brac=600
temp1=[]
tempd=[]
for i in da:
    coun=0
    while coun<brac:
        for j in range(len(labels)):
            if labels[j]:
                temp1.append(labels[j])
                tempd.append(vals[j])
                coun+=1
            if coun==brac:
                break
vals=tempd
labels=temp1
```

In [5]:

```
l=len(vals)
print(l)
```

3000

In [6]:

```
X_train,X_test,Y_train,Y_test=train_test_split(vals,labels,stratify=labels,test_size=0.2,ra
```

In [7]:

```
scaler=StandardScaler()  
x_train=scaler.fit_transform(X_train)  
x_test=scaler.transform(X_test)  
y_train=np.array(Y_train)  
y_test=np.array(Y_test)
```

In [8]:

```
model=SVC(kernel='sigmoid',gamma='auto')  
model.fit(x_train,y_train)
```

Out[8]:

```
▼ SVC  
SVC(gamma='auto', kernel='sigmoid')
```

In [9]:

```
y_pred=model.predict(x_test)
```

In [10]:

```
print((accuracy_score(y_pred,y_test))*100,"%")
```

88.66666666666667 %

In [11]:

```
model1= KNeighborsClassifier(n_neighbors=5)  
model1.fit(x_train,y_train)
```

Out[11]:

```
▼ KNeighborsClassifier  
KNeighborsClassifier()
```

In [12]:

```
y_pred1=model1.predict(x_test)
```

In [13]:

```
print((accuracy_score(y_pred1,y_test)*100,"%")
```

98.16666666666667 %

In [14]:

```
model2= GaussianNB()
model2.fit(x_train,y_train)
```

Out[14]:

```
▼ GaussianNB
GaussianNB()
```

In [15]:

```
y_pred2=model2.predict(x_test)
```

In [16]:

```
print((accuracy_score(y_pred2,y_test)*100),"%")
```

```
90.16666666666666 %
```

In [17]:

```
train_x,val_x,train_y,val_y=train_test_split(x_train,y_train,stratify=y_train,test_size=0.2)
```

In [18]:

```
print(x_train.shape,x_test.shape)
```

```
(2400, 27) (600, 27)
```

In [19]:

```
columns = ['SRC_ADD', 'DES_ADD', 'PKT_ID', 'FROM_NODE', 'TO_NODE', 'PKT_TYPE',
'PKT_SIZE', 'FLAGS', ' FID', 'SEQ_NUMBER', 'NUMBER_OF_PKT', 'NUMBER_OF_BYTE', 'NODE_NAME_FR',
'NODE_NAME_TO', 'PKT_IN', 'PKT_OUT', 'PKT_R', 'PKT_DELAY_NODE', 'PKT_RATE', 'BYTE_RATE', '
'UTILIZATION', 'PKT_DELAY', 'PKT_SEND_TIME', 'PKT RESEVED_TIME', 'FIRST_PKT SENT', 'LAST P
```

In [20]:

```
model1=SVC(kernel='sigmoid',gamma='auto')
model1.fit(train_x,train_y)
y_val_pred1=model1.predict(val_x)
y_val_pred1=pd.DataFrame(y_val_pred1)
y_test_pred1=model1.predict(x_test)
y_test_pred1=pd.DataFrame(y_test_pred1)
```

In [21]:

```
model2=KNeighborsClassifier(n_neighbors=5)
model2.fit(train_x,train_y)
y_val_pred2=model2.predict(val_x)
y_val_pred2=pd.DataFrame(y_val_pred2)
y_test_pred2=model2.predict(x_test)
y_test_pred2=pd.DataFrame(y_test_pred2)
```

In [22]:

```

model3=GaussianNB()
model3.fit(train_x,train_y)
y_val_pred3=model3.predict(val_x)
y_val_pred3=pd.DataFrame(y_val_pred3)
y_test_pred3=model3.predict(x_test)
y_test_pred3=pd.DataFrame(y_test_pred3)

```

In [23]:

```

val_input = pd.concat([pd.DataFrame(val_x,columns=columns),y_val_pred1,y_val_pred2,y_val_pr
test_input = pd.concat([pd.DataFrame(x_test,columns=columns),y_test_pred1,y_test_pred2,y_te

```

In [24]:

```

model=RandomForestClassifier(n_estimators=200)
model.fit(val_input,val_y)

```

C:\Users\AJAY\anaconda3\envs\tf\lib\site-packages\sklearn\utils\validation.p  
y:1858: FutureWarning: Feature names only support names that are all string  
s. Got feature names with dtypes: ['int', 'str']. An error will be raised in  
1.2.  
warnings.warn(

Out[24]:

▼	RandomForestClassifier
RandomForestClassifier(n_estimators=200)	

In [25]:

```

print((model.score(test_input,y_test))*100,"%")

```

97.83333333333334 %

C:\Users\AJAY\anaconda3\envs\tf\lib\site-packages\sklearn\utils\validation.p  
y:1858: FutureWarning: Feature names only support names that are all string  
s. Got feature names with dtypes: ['int', 'str']. An error will be raised in  
1.2.  
warnings.warn(

In [ ]: