NAME: AVIREDDY NVSRK ROHAN

REG.NO: 19BCE1180

DATASET USED: KDD CUP

In [1]:
```python
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
from matplotlib import pyplot as plt
```

In [2]:
```python
from keras.layers import Dense
from keras.models import Sequential
```

In [3]:
```python
data=pd.read_csv("/content/drive/MyDrive/ML_LAB/corrected",header=None)
data.columns=['duration','protocol_type','service','flag','src_bytes','dst_bytes','land
```

In [4]:
```python
data.head()
```

Out[4]:

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	nur
0	0	udp	private	SF	105	146	0	0	0	0	
1	0	udp	private	SF	105	146	0	0	0	0	
2	0	udp	private	SF	105	146	0	0	0	0	
3	0	udp	private	SF	105	146	0	0	0	0	
4	0	udp	private	SF	105	146	0	0	0	0	

In [ ]:
```python
data["class"].value_counts()
```

Out[ ]:
```
smurf.            164091
normal.            60593
neptune.           58001
snmpgetattack.      7741
mailbomb.           5000
guess_passwd.       4367
snmpguess.          2406
satan.              1633
warezmaster.        1602
back.               1098
mscan.              1053
apache2.             794
```

```
processtable.          759
saint.                 736
portsweep.             354
ipsweep.               306
httptunnel.            158
pod.                    87
nmap.                   84
buffer_overflow.        22
multihop.               18
sendmail.               17
named.                  17
ps.                     16
rootkit.                13
xterm.                  13
teardrop.               12
land.                    9
xlock.                   9
xsnoop.                  4
ftp_write.               3
udpstorm.                2
perl.                    2
sqlattack.               2
loadmodule.              2
worm.                    2
phf.                     2
imap.                    1
Name: class, dtype: int64
```

In [5]:
```python
le=LabelEncoder()
data["protocol_type"]=le.fit_transform(data["protocol_type"])
data["service"]=le.fit_transform(data["service"])
data["flag"]=le.fit_transform(data["flag"])
data["protocol_type"]=data["protocol_type"].astype("object")
data["service"]=data["service"].astype("object")
data["flag"]=data["flag"].astype("object")
data["is_host_login"]=data["is_host_login"].astype("object")
data["is_guest_login"]=data["is_guest_login"].astype("object")
data["logged_in"]=data["logged_in"].astype("object")
```

In [6]:
```python
cond=((data['class']=="smurf.") | (data['class']=="normal.") | (data['class']=="satan."
```

In [7]:
```python
new_data=data[cond]
```

In [8]:
```python
multi_label=new_data["class"]
new_data.drop(["class"],inplace=True,axis=1)
multi_label=le.fit_transform(multi_label)
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/frame.py:4174: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  errors=errors,
```

In [ ]:
```python
new_data.columns.get_loc("root_shell")
```

```
Out[ ]:  13
```

```
In [9]:   new_data.shape
```

```
Out[9]:  (227053, 41)
```

```
In [ ]:   temp=pd.DataFrame()
          for i in new_data.columns:
              if new_data.dtypes[i] is not 'O':
                  temp.insert(new_data.columns.get_loc(i),i,new_data[i])
```

```
In [10]:  st_obj=StandardScaler()
          data_std=st_obj.fit_transform(new_data)
          data_std=pd.DataFrame(data_std)
```

```
In [11]:  import imblearn
          from imblearn.over_sampling import SMOTE
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/externals/six.py:31: FutureWarning: The m
odule is deprecated in version 0.21 and will be removed in version 0.23 since we've drop
ped support for Python 2.7. Please rely on the official version of six (https://pypi.or
g/project/six/).
  "(https://pypi.org/project/six/).", FutureWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:144: FutureWarning:
The sklearn.neighbors.base module is  deprecated in version 0.22 and will be removed in
version 0.24. The corresponding classes / functions should instead be imported from skle
arn.neighbors. Anything that cannot be imported from sklearn.neighbors is now part of th
e private API.
  warnings.warn(message, FutureWarning)
```

class distributions before using SMOTE for 80:20 train test split

```
In [13]:  train_data,test_data,train_labels,test_labels=train_test_split(data_std,multi_label,tes
```

```
In [ ]:   g=pd.Series(train_labels)
```

```
In [ ]:   g.value_counts()*100/train_labels.shape[0]
```

```
Out[ ]:  3    72.292752
         0    26.671144
         2     0.718997
         1     0.317107
         dtype: float64
```

Multilayer perceptron model

```
In [33]:  mlp=Sequential()
          mlp.add(Dense(32,input_dim=41,activation="relu"))
```

```python
mlp.add(Dense(16,activation="relu"))
mlp.add(Dense(8,activation="relu"))
mlp.add(Dense(1,activation="sigmoid"))
mlp.compile(optimizer="Adam",loss="categorical_crossentropy",metrics=['accuracy'])
```

Performance before SMOTE

In [44]:
```python
mlp.fit(train_data,train_labels)
```

5677/5677 [==============================] - 12s 2ms/step - loss: 0.0000e+00 - accuracy: 0.2606

Out[44]: <keras.callbacks.History at 0x7f29f898b810>

on training data

In [45]:
```python
uns_train_pred=mlp.predict(train_data)
```

In [46]:
```python
print(classification_report(train_labels,uns_train_pred))
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272: Undefine
dMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels wit
h no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       1.00      0.97      0.99     48446
           1       0.00      0.86      0.01       576
           2       0.00      0.00      0.00      1306
           3       0.00      0.00      0.00    131314

    accuracy                           0.26    181642
   macro avg       0.25      0.46      0.25    181642
weighted avg       0.27      0.26      0.26    181642
```

In [54]:
```python
cm=confusion_matrix(train_labels,uns_train_pred)
disp=ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=["0","1","2","3"])
disp.plot()
plt.show()
```
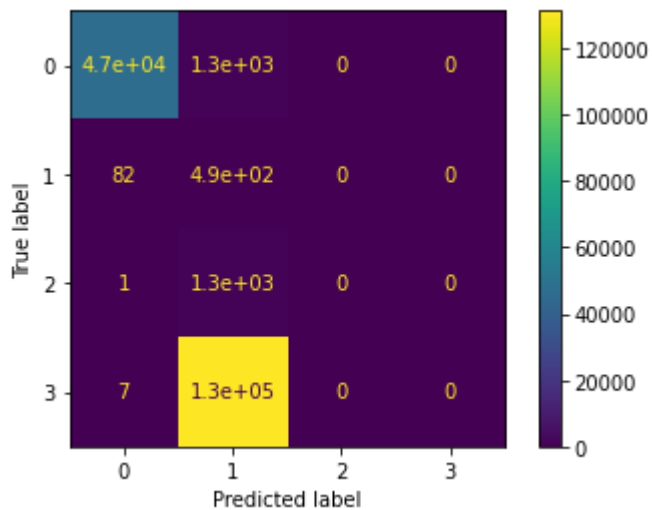
on testing data

In [48]:
```python
uns_test_pred=mlp.predict(test_data)
```

In [49]:
```python
print(classification_report(test_labels,uns_test_pred))
```

	precision	recall	f1-score	support
0	1.00	0.97	0.98	12147
1	0.00	0.82	0.01	160
2	0.00	0.00	0.00	327
3	0.00	0.00	0.00	32777
accuracy			0.26	45411
macro avg	0.25	0.45	0.25	45411
weighted avg	0.27	0.26	0.26	45411

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272: Undefine
dMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels wit
h no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [55]:
```python
cm=confusion_matrix(test_labels,uns_test_pred)
disp=ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=["0","1","2","3"])
disp.plot()
plt.show()
```
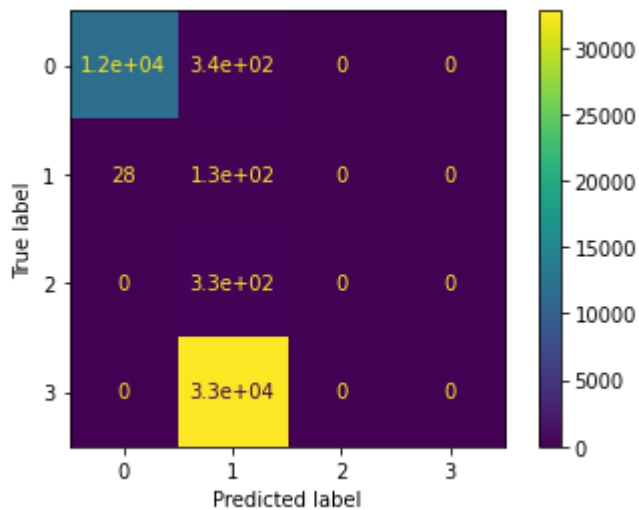
PERFORMANCE AFTER SMOTE

In [12]:
```python
smote=SMOTE()
X_smote,y_smote=smote.fit_resample(data_std,multi_label)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: F
unction safe_indexing is deprecated; safe_indexing is deprecated in version 0.22 and wil
l be removed in version 0.24.
  warnings.warn(msg, category=FutureWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: F
unction safe_indexing is deprecated; safe_indexing is deprecated in version 0.22 and wil
l be removed in version 0.24.
  warnings.warn(msg, category=FutureWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: F
unction safe_indexing is deprecated; safe_indexing is deprecated in version 0.22 and wil
l be removed in version 0.24.
  warnings.warn(msg, category=FutureWarning)
```

In [50]:
```python
train_smote,test_smote,train_smote_labs,test_smote_labs=train_test_split(X_smote,y_smot
```

In [51]:
```python
mlp.fit(train_smote,train_smote_labs)
```

```
18461/18461 [==============================] - 28s 2ms/step - loss: 0.0000e+00 - accurac
y: 0.4544
```

Out[51]: <keras.callbacks.History at 0x7f29f0833690>

on smote train data

In [52]:
```python
mlp_train_pred=mlp.predict(train_smote)
```

In [53]:
```python
print(classification_report(train_smote_labs,mlp_train_pred))
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272: Undefine
dMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels wit
h no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

	precision	recall	f1-score	support
0	0.87	0.96	0.91	147712
1	0.30	0.86	0.44	147373
2	0.00	0.00	0.00	147757
3	0.00	0.00	0.00	147885
accuracy			0.45	590727
macro avg	0.29	0.45	0.34	590727
weighted avg	0.29	0.45	0.34	590727

In [56]:
```python
cm=confusion_matrix(train_smote_labs,mlp_train_pred)
disp=ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=["0","1","2","3"])
disp.plot()
plt.show()
```
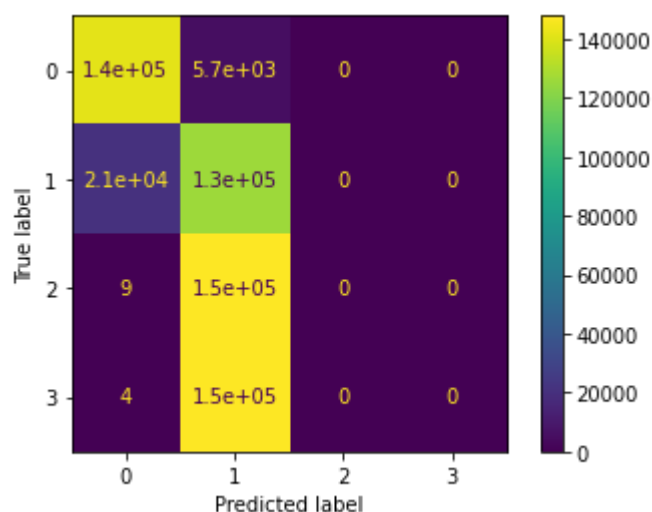

on smote test data

In [58]:
```python
mlp_test_pred=mlp.predict(test_smote)
```

In [59]:
```python
print(classification_report(test_smote_labs,mlp_test_pred))
```

	precision	recall	f1-score	support
0	0.87	0.96	0.91	16379
1	0.30	0.86	0.45	16718
2	0.00	0.00	0.00	16334
3	0.00	0.00	0.00	16206
accuracy			0.46	65637
macro avg	0.29	0.45	0.34	65637
weighted avg	0.29	0.46	0.34	65637

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272: Undefine
dMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels wit
h no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [60]:
```python
cm=confusion_matrix(test_smote_labs,mlp_test_pred)
```

```python
disp=ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=["0","1","2","3"])
disp.plot()
plt.show()
```