PRACTICAL NO: 6

AIM: Anomalies detection in Network Security.

---CODE---

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import IsolationForest

url = "http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data.gz"
data = pd.read_csv(url, header=None)

data.columns = ['duration', 'protocol_type', 'service', 'flag', 'src_bytes', 'dst_bytes',
                'land', 'wrong_fragment', 'urgent', 'hot', 'num_failed_logins', 'logged_in',
                'num_compromised', 'root_shell', 'su_attempted', 'num_root', 'num_file_creations',
                'num_shells', 'num_access_files', 'num_outbound_cmds', 'is_host_login',
'is_guest_login',
                'count', 'srv_count', 'serror_rate', 'srv_serror_rate', 'rerror_rate',
'srv_rerror_rate',
                'same_srv_rate', 'diff_srv_rate', 'srv_diff_host_rate', 'dst_host_count',
'dst_host_srv_count',
                'dst_host_same_srv_rate', 'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate',
                'dst_host_srv_diff_host_rate', 'dst_host_serror_rate', 'dst_host_srv_serror_rate',
                'dst_host_rerror_rate', 'dst_host_srv_rerror_rate', 'label']

data['label'] = data['label'].apply(lambda x: 1 if x != 'normal.' else 0)

X = data.drop('label', axis=1)
y = data['label']

le = LabelEncoder()
X['protocol_type'] = le.fit_transform(X['protocol_type'])
X['service'] = le.fit_transform(X['service'])
X['flag'] = le.fit_transform(X['flag'])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = IsolationForest(contamination=0.1, random_state=42)
model.fit(X_train)

y_pred = model.predict(X_test)
y_pred = [1 if x == -1 else 0 for x in y_pred]

accuracy = (y_pred == y_test).mean()
print(accuracy)
```

---OUTPUT---

```
0.20603927580951875
```