In [22]:
```
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
import tensorflow as tf
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix
from keras.models import Model, load_model, Sequential
from keras.layers import Input, Dense
from keras.callbacks import ModelCheckpoint, TensorBoard
```

In [23]:
```
%matplotlib inline
sns.set(style='whitegrid')
```

In [24]:
```
df = pd.read_csv("C:/Users/Suraj/Downloads/creditcard.csv")
```

In [25]:
```
df = df.drop(['Time'], axis=1)
```

In [26]:
```
df['Amount'] = StandardScaler().fit_transform(df['Amount'].values.reshape(-1, 1))
```

In [27]:
```
df_fraud = df[df['Class']==1]
df_normal = df[df['Class']==0]
df_normal = df_normal.sample(frac = 1.0).reset_index(drop = True) #Just shuffling
df_normal_1 = df_normal.iloc[:int(df_normal.shape[0]*0.8),:] #80% of normal data for training
df_normal_2 = df_normal.iloc[int(df_normal.shape[0]*0.8):,:] #20% of normal data to merge with f
```

In [28]:
```
X_test = pd.concat([df_fraud,df_normal_2], axis = 0)
X_test = X_test.sample(frac = 1.0).reset_index(drop = True) #Just shuffling

#Separate in input and target variables
X_train = df_normal_1[df_normal_1['Class'] == 0]
X_train = X_train.drop(['Class'], axis=1)

y_test = X_test['Class']
X_test = X_test.drop(['Class'], axis=1)
```

In [29]:
```
#Build the Neural Network
input_dim = X_train.shape[1]
encoding_dim = 14

model = Sequential()
model.add(Dense(29,input_dim = input_dim, activation="relu"))
model.add(Dense(14, activation="relu"))
model.add(Dense(7, activation="relu"))
model.add(Dense(14, activation="relu"))
model.add(Dense(input_dim, activation="sigmoid"))
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['accuracy'])
```

In [30]:
```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 dense_5 (Dense)             (None, 29)                870       
                                                                 
 dense_6 (Dense)             (None, 14)                420       
                                                                 
 dense_7 (Dense)             (None, 7)                 105       
                                                                 
 dense_8 (Dense)             (None, 14)                112       
                                                                 
 dense_9 (Dense)             (None, 29)                435       
                                                                 
=================================================================
Total params: 1942 (7.59 KB)
Trainable params: 1942 (7.59 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [31]:

```
#Fit the autoencoder and check loss for train and test
checkpointer = ModelCheckpoint(filepath="nae.h5", verbose=0, save_best_only=True)
```

In [33]:

```
#Save history to plot learning curves
history = model.fit(X_train, X_train,
epochs=100,
batch_size=32,
shuffle=True,
validation_data=(X_test, X_test),
verbose=1,
callbacks=[checkpointer]).history

autoencoder = load_model('nae.h5')
```

```
Epoch 1/100
7108/7108 [==============================] - 22s 3ms/step - loss: 0.8727 - accuracy: 0.38
94 - val_loss: 1.0968 - val_accuracy: 0.3821
Epoch 2/100
  52/7108 [..............................] - ETA: 21s - loss: 0.8115 - accuracy: 0.3918
```

```
C:\Users\Suraj\anaconda3\lib\site-packages\keras\src\engine\training.py:3079: UserWarning: You a
re saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We reco
mmend using instead the native Keras format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8642 - accuracy: 0.39
69 - val_loss: 1.0884 - val_accuracy: 0.4041
Epoch 3/100
7108/7108 [==============================] - 27s 4ms/step - loss: 0.8604 - accuracy: 0.40
88 - val_loss: 1.0855 - val_accuracy: 0.4173
Epoch 4/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8572 - accuracy: 0.41
46 - val_loss: 1.0836 - val_accuracy: 0.4179
Epoch 5/100
7108/7108 [==============================] - 26s 4ms/step - loss: 0.8553 - accuracy: 0.41
29 - val_loss: 1.0832 - val_accuracy: 0.4183
Epoch 6/100
7108/7108 [==============================] - 27s 4ms/step - loss: 0.8533 - accuracy: 0.40
96 - val_loss: 1.0801 - val_accuracy: 0.4074
Epoch 7/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8515 - accuracy: 0.40
25 - val_loss: 1.0792 - val_accuracy: 0.3995
Epoch 8/100
7108/7108 [==============================] - 27s 4ms/step - loss: 0.8477 - accuracy: 0.39
92 - val_loss: 1.0749 - val_accuracy: 0.4031
Epoch 9/100
7108/7108 [==============================] - 26s 4ms/step - loss: 0.8460 - accuracy: 0.40
25 - val_loss: 1.0746 - val_accuracy: 0.3981
Epoch 10/100
7108/7108 [==============================] - 26s 4ms/step - loss: 0.8450 - accuracy: 0.39
66 - val_loss: 1.0730 - val_accuracy: 0.3935
Epoch 11/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8440 - accuracy: 0.40
01 - val_loss: 1.0729 - val_accuracy: 0.3957
Epoch 12/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8430 - accuracy: 0.39
98 - val_loss: 1.0740 - val_accuracy: 0.3970
Epoch 13/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8419 - accuracy: 0.40
10 - val_loss: 1.0711 - val_accuracy: 0.4017
Epoch 14/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8412 - accuracy: 0.40
17 - val_loss: 1.0700 - val_accuracy: 0.4006
Epoch 15/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8407 - accuracy: 0.40
03 - val_loss: 1.0704 - val_accuracy: 0.4025
Epoch 16/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8402 - accuracy: 0.39
81 - val_loss: 1.0702 - val_accuracy: 0.3922
Epoch 17/100
7108/7108 [==============================] - 25s 3ms/step - loss: 0.8400 - accuracy: 0.39
74 - val_loss: 1.0698 - val_accuracy: 0.3993
Epoch 18/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8395 - accuracy: 0.39
75 - val_loss: 1.0673 - val_accuracy: 0.3912
Epoch 19/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8362 - accuracy: 0.39
76 - val_loss: 1.0650 - val_accuracy: 0.3973
Epoch 20/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8354 - accuracy: 0.39
68 - val_loss: 1.0636 - val_accuracy: 0.3852
Epoch 21/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8350 - accuracy: 0.39
34 - val_loss: 1.0633 - val_accuracy: 0.3939
Epoch 22/100
7108/7108 [==============================] - 25s 4ms/step - loss: 0.8347 - accuracy: 0.39
24 - val_loss: 1.0627 - val_accuracy: 0.3870
Epoch 23/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8344 - accuracy: 0.39
```

```
12 - val_loss: 1.0624 - val_accuracy: 0.3912
Epoch 24/100
7108/7108 [==============================] - 27s 4ms/step - loss: 0.8341 - accuracy: 0.39
16 - val_loss: 1.0626 - val_accuracy: 0.3856
Epoch 25/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8339 - accuracy: 0.39
31 - val_loss: 1.0618 - val_accuracy: 0.3888
Epoch 26/100
7108/7108 [==============================] - 25s 3ms/step - loss: 0.8337 - accuracy: 0.39
03 - val_loss: 1.0615 - val_accuracy: 0.3892
Epoch 27/100
7108/7108 [==============================] - 25s 4ms/step - loss: 0.8336 - accuracy: 0.39
08 - val_loss: 1.0614 - val_accuracy: 0.3862
Epoch 28/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8334 - accuracy: 0.39
03 - val_loss: 1.0619 - val_accuracy: 0.3813
Epoch 29/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8332 - accuracy: 0.38
83 - val_loss: 1.0618 - val_accuracy: 0.3906
Epoch 30/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8333 - accuracy: 0.38
85 - val_loss: 1.0612 - val_accuracy: 0.3831
Epoch 31/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8331 - accuracy: 0.38
75 - val_loss: 1.0599 - val_accuracy: 0.3877
Epoch 32/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8329 - accuracy: 0.38
93 - val_loss: 1.0603 - val_accuracy: 0.3842
Epoch 33/100
7108/7108 [==============================] - 22s 3ms/step - loss: 0.8327 - accuracy: 0.38
80 - val_loss: 1.0602 - val_accuracy: 0.3810
Epoch 34/100
7108/7108 [==============================] - 20s 3ms/step - loss: 0.8327 - accuracy: 0.38
64 - val_loss: 1.0599 - val_accuracy: 0.3850
Epoch 35/100
7108/7108 [==============================] - 19s 3ms/step - loss: 0.8324 - accuracy: 0.38
75 - val_loss: 1.0709 - val_accuracy: 0.3485
Epoch 36/100
7108/7108 [==============================] - 19s 3ms/step - loss: 0.8325 - accuracy: 0.38
88 - val_loss: 1.0581 - val_accuracy: 0.3886
Epoch 37/100
7108/7108 [==============================] - 20s 3ms/step - loss: 0.8322 - accuracy: 0.39
01 - val_loss: 1.0580 - val_accuracy: 0.3882
Epoch 38/100
7108/7108 [==============================] - 20s 3ms/step - loss: 0.8320 - accuracy: 0.39
13 - val_loss: 1.0583 - val_accuracy: 0.3897
Epoch 39/100
7108/7108 [==============================] - 20s 3ms/step - loss: 0.8316 - accuracy: 0.39
27 - val_loss: 1.0575 - val_accuracy: 0.3882
Epoch 40/100
7108/7108 [==============================] - 20s 3ms/step - loss: 0.8317 - accuracy: 0.39
65 - val_loss: 1.0577 - val_accuracy: 0.3942
Epoch 41/100
7108/7108 [==============================] - 20s 3ms/step - loss: 0.8314 - accuracy: 0.40
03 - val_loss: 1.0590 - val_accuracy: 0.4057
Epoch 42/100
7108/7108 [==============================] - 21s 3ms/step - loss: 0.8311 - accuracy: 0.40
93 - val_loss: 1.0567 - val_accuracy: 0.4042
Epoch 43/100
7108/7108 [==============================] - 20s 3ms/step - loss: 0.8311 - accuracy: 0.41
09 - val_loss: 1.0568 - val_accuracy: 0.4051
Epoch 44/100
7108/7108 [==============================] - 20s 3ms/step - loss: 0.8310 - accuracy: 0.41
24 - val_loss: 1.0569 - val_accuracy: 0.4207
```

```
Epoch 45/100
7108/7108 [==============================] - 22s 3ms/step - loss: 0.8307 - accuracy: 0.41
14 - val_loss: 1.0563 - val_accuracy: 0.4145
Epoch 46/100
7108/7108 [==============================] - 22s 3ms/step - loss: 0.8306 - accuracy: 0.41
38 - val_loss: 1.0573 - val_accuracy: 0.4006
Epoch 47/100
7108/7108 [==============================] - 22s 3ms/step - loss: 0.8304 - accuracy: 0.40
89 - val_loss: 1.0558 - val_accuracy: 0.4277
Epoch 48/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8303 - accuracy: 0.41
28 - val_loss: 1.0592 - val_accuracy: 0.4220
Epoch 49/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8301 - accuracy: 0.41
36 - val_loss: 1.0603 - val_accuracy: 0.3988
Epoch 50/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8301 - accuracy: 0.41
05 - val_loss: 1.0589 - val_accuracy: 0.4198
Epoch 51/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8298 - accuracy: 0.40
68 - val_loss: 1.0550 - val_accuracy: 0.3939
Epoch 52/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8295 - accuracy: 0.40
51 - val_loss: 1.0542 - val_accuracy: 0.4176
Epoch 53/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8288 - accuracy: 0.41
15 - val_loss: 1.0536 - val_accuracy: 0.4144
Epoch 54/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8285 - accuracy: 0.40
78 - val_loss: 1.0541 - val_accuracy: 0.3902
Epoch 55/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8286 - accuracy: 0.40
02 - val_loss: 1.0535 - val_accuracy: 0.4212
Epoch 56/100
7108/7108 [==============================] - 26s 4ms/step - loss: 0.8283 - accuracy: 0.41
64 - val_loss: 1.0532 - val_accuracy: 0.4178
Epoch 57/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8286 - accuracy: 0.41
54 - val_loss: 1.0549 - val_accuracy: 0.4284
Epoch 58/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8282 - accuracy: 0.41
71 - val_loss: 1.0532 - val_accuracy: 0.4166
Epoch 59/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8283 - accuracy: 0.41
25 - val_loss: 1.0534 - val_accuracy: 0.4153
Epoch 60/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8282 - accuracy: 0.41
78 - val_loss: 1.0541 - val_accuracy: 0.4174
Epoch 61/100
7108/7108 [==============================] - 27s 4ms/step - loss: 0.8281 - accuracy: 0.42
09 - val_loss: 1.0532 - val_accuracy: 0.4234
Epoch 62/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8281 - accuracy: 0.41
45 - val_loss: 1.0539 - val_accuracy: 0.4230
Epoch 63/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8280 - accuracy: 0.40
99 - val_loss: 1.0534 - val_accuracy: 0.3901
Epoch 64/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8278 - accuracy: 0.40
32 - val_loss: 1.0528 - val_accuracy: 0.3967
Epoch 65/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8278 - accuracy: 0.40
75 - val_loss: 1.0529 - val_accuracy: 0.4176
Epoch 66/100
```

```
7108/7108 [==============================] - 25s 3ms/step - loss: 0.8279 - accuracy: 0.41
02 - val_loss: 1.0528 - val_accuracy: 0.4131
Epoch 67/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8280 - accuracy: 0.41
02 - val_loss: 1.0528 - val_accuracy: 0.4091
Epoch 68/100
7108/7108 [==============================] - 25s 3ms/step - loss: 0.8278 - accuracy: 0.40
57 - val_loss: 1.0526 - val_accuracy: 0.4037
Epoch 69/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8277 - accuracy: 0.40
97 - val_loss: 1.0541 - val_accuracy: 0.4056
Epoch 70/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8279 - accuracy: 0.40
90 - val_loss: 1.0540 - val_accuracy: 0.3887
Epoch 71/100
7108/7108 [==============================] - 26s 4ms/step - loss: 0.8276 - accuracy: 0.40
94 - val_loss: 1.0528 - val_accuracy: 0.3997
Epoch 72/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8276 - accuracy: 0.41
52 - val_loss: 1.0537 - val_accuracy: 0.4147
Epoch 73/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8272 - accuracy: 0.41
85 - val_loss: 1.0531 - val_accuracy: 0.4070
Epoch 74/100
7108/7108 [==============================] - 25s 4ms/step - loss: 0.8271 - accuracy: 0.41
41 - val_loss: 1.0519 - val_accuracy: 0.4180
Epoch 75/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8273 - accuracy: 0.41
38 - val_loss: 1.0544 - val_accuracy: 0.3918
Epoch 76/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8272 - accuracy: 0.40
99 - val_loss: 1.0534 - val_accuracy: 0.4193
Epoch 77/100
7108/7108 [==============================] - 26s 4ms/step - loss: 0.8271 - accuracy: 0.41
45 - val_loss: 1.0531 - val_accuracy: 0.4162
Epoch 78/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8270 - accuracy: 0.41
58 - val_loss: 1.0518 - val_accuracy: 0.4185
Epoch 79/100
7108/7108 [==============================] - 26s 4ms/step - loss: 0.8268 - accuracy: 0.41
65 - val_loss: 1.0524 - val_accuracy: 0.4167
Epoch 80/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8268 - accuracy: 0.41
76 - val_loss: 1.0523 - val_accuracy: 0.4139
Epoch 81/100
7108/7108 [==============================] - 27s 4ms/step - loss: 0.8268 - accuracy: 0.41
70 - val_loss: 1.0517 - val_accuracy: 0.4071
Epoch 82/100
7108/7108 [==============================] - 25s 4ms/step - loss: 0.8267 - accuracy: 0.41
68 - val_loss: 1.0518 - val_accuracy: 0.4187
Epoch 83/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8268 - accuracy: 0.41
65 - val_loss: 1.0515 - val_accuracy: 0.4231
Epoch 84/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8266 - accuracy: 0.41
70 - val_loss: 1.0537 - val_accuracy: 0.4097
Epoch 85/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8267 - accuracy: 0.41
53 - val_loss: 1.0517 - val_accuracy: 0.4167
Epoch 86/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8265 - accuracy: 0.41
59 - val_loss: 1.0512 - val_accuracy: 0.4144
Epoch 87/100
7108/7108 [==============================] - 25s 4ms/step - loss: 0.8265 - accuracy: 0.41
```

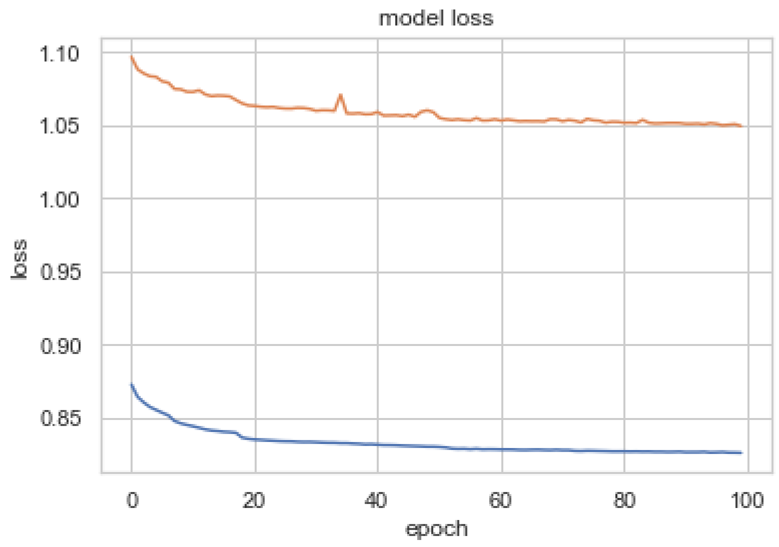
```
63 - val_loss: 1.0514 - val_accuracy: 0.4159
Epoch 88/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8264 - accuracy: 0.41
90 - val_loss: 1.0516 - val_accuracy: 0.4065
Epoch 89/100
7108/7108 [==============================] - 25s 4ms/step - loss: 0.8264 - accuracy: 0.41
74 - val_loss: 1.0516 - val_accuracy: 0.4159
Epoch 90/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8266 - accuracy: 0.41
50 - val_loss: 1.0515 - val_accuracy: 0.4079
Epoch 91/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8263 - accuracy: 0.41
62 - val_loss: 1.0510 - val_accuracy: 0.4170
Epoch 92/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8263 - accuracy: 0.41
61 - val_loss: 1.0510 - val_accuracy: 0.4133
Epoch 93/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8263 - accuracy: 0.41
49 - val_loss: 1.0511 - val_accuracy: 0.4211
Epoch 94/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8265 - accuracy: 0.41
34 - val_loss: 1.0506 - val_accuracy: 0.4065
Epoch 95/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8261 - accuracy: 0.41
52 - val_loss: 1.0514 - val_accuracy: 0.4143
Epoch 96/100
7108/7108 [==============================] - 24s 3ms/step - loss: 0.8262 - accuracy: 0.41
34 - val_loss: 1.0510 - val_accuracy: 0.4094
Epoch 97/100
7108/7108 [==============================] - 25s 3ms/step - loss: 0.8263 - accuracy: 0.41
65 - val_loss: 1.0499 - val_accuracy: 0.4174
Epoch 98/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8260 - accuracy: 0.41
56 - val_loss: 1.0504 - val_accuracy: 0.4220
Epoch 99/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8259 - accuracy: 0.41
59 - val_loss: 1.0507 - val_accuracy: 0.4176
Epoch 100/100
7108/7108 [==============================] - 23s 3ms/step - loss: 0.8258 - accuracy: 0.41
50 - val_loss: 1.0495 - val_accuracy: 0.4055
```

In [34]:

```
#Plot losses
plt.plot(history['loss'])
plt.plot(history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
```

Out[34]:

```
Text(0.5, 0, 'epoch')
```



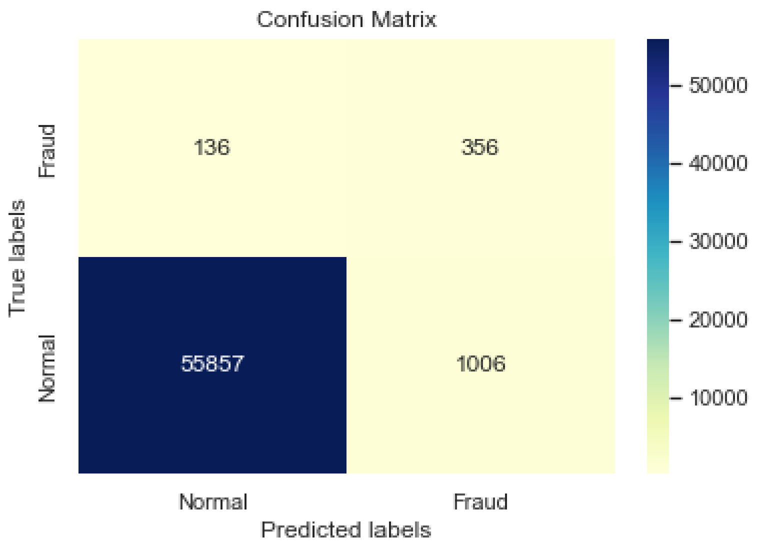
In [35]:
```
#Predict on test set
predictions = model.predict(X_test)

mse = np.mean(np.power(X_test - predictions, 2), axis=1)
error_df = pd.DataFrame({'mse': mse,'fraud': y_test})
```

```
1793/1793 [==============================] - 4s 2ms/step
```

In [36]:
```
#Set an error threshold above which a transaction is considered fraud
threshold = 4.5
error_df['pred_01'] = [1 if e > threshold else 0 for e in error_df['mse'].values]
conf_mat = confusion_matrix(error_df['fraud'], error_df['pred_01'])
```

In [37]:
```
#Print confusion matrix for the given threshold
ax= plt.subplot()
sns.heatmap(conf_mat, annot=True, fmt="g", cmap="YlGnBu")
# labels, title and ticks
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
ax.set_ylim([0,2])
ax.xaxis.set_ticklabels(["Normal", "Fraud"]); ax.yaxis.set_ticklabels(["Normal", "Fraud"])
```

Out[37]:
```
[Text(0, 0.5, 'Normal'), Text(0, 1.5, 'Fraud')]
```



In [ ]: