```
import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from scipy import stats
import seaborn as sns
import tensorflow as tf
from tensorflow import keras
import sklearn
import sys
import io
def plot learning curves(loss, val loss):
    plt.figure()
    plt.plot(np.arange(len(loss)), loss, "b.-", label="Training loss")
    plt.plot(np.arange(len(val_loss)), val_loss, "r.-", label="Validation loss")
    plt.gca().xaxis.set_major_locator(mpl.ticker.MaxNLocator(integer=True))
   plt.legend(fontsize=14)
   plt.xlabel("Epochs")
   plt.ylabel("Loss")
    plt.grid(True)
n \text{ steps} = 50
forecast = 10
#EXTRACT FLOW, Z SCORE, OUTLIERS
data west = pd.read csv('denoised data 1015 south.csv')
data west o = np.array(data west.FLOW)
print(data west o.shape)
#EXRTEND DATA
array to concatinate = data west o[177:]
for iter in range (35):
    data west o = np.concatenate([data west o,array to concatinate])
#SCALE AND RESHAPE DATA
scaler = MinMaxScaler()
array = data_west_o.reshape(-1, 1)
array scaled = scaler.fit transform(array)
flow_reshaped = array_scaled[:(len(array_scaled) - (len(array_scaled) % (n_steps+forecast
print(flow reshaped.shape)
#TRAIN SET, VALIDATION SET, TEST SET
test = int(0.7 * flow reshaped.shape[0])
valid = int(0.9 * flow_reshaped.shape[0])
X train = flow reshaped[:test, :n steps]
X valid = flow reshaped[test:valid, :n steps]
X_test = flow_reshaped[valid:, :n_steps]
print(X test.shape)
print(X test[-1:].shape)
#prepare targets
Y = np.empty((flow_reshaped.shape[0], n_steps, forecast))
for step ahead in range(1, forecast + 1):
    Y[:, :, step ahead - 1] = flow reshaped[:, step ahead:step ahead + n steps, 0]
y train = Y[:test]
y valid = Y[test:valid]
y test = Y[valid:]
#DEFINE AND COMPILE MODEL
model = keras.models.Sequential([
   keras.layers.LSTM(20, return sequences=True, input shape=[None, 1]),
    keras.layers.LSTM(20, return sequences=True),
    keras.layers.TimeDistributed(keras.layers.Dense(forecast))
])
```

```
def last_time_step_mse(Y_true, Y_pred):
    return keras.metrics.mean squared error(Y true[:, -1], Y pred[:, -1])
model.compile(loss="mse", optimizer="adam", metrics=[last_time_step_mse])
early stopping cb = keras.callbacks.EarlyStopping(patience=10, restore best weights=True)
history = model.fit(X_train, y_train, epochs=700,
                     validation_data=(X_valid, y_valid), callbacks=[early_stopping_cb])
model.save("lstm 1014 south.h5")
plot learning curves(history.history["loss"], history.history["val loss"])
#50 MINS PREDICTION
flow unscaled = array[:(len(array) - (len(array) % (n steps + forecast)))].reshape(-1, (n
y test unscaled = flow unscaled[valid:, n steps:, 0]
y_real_rescaled = y_test_unscaled[-1, :].reshape(-1, 1)
print(y_real_rescaled.shape)
flow_not_reshaped = array[:(len(array) - (len(array) % (n_steps+forecast)))]
#flow prediction
y_pred = model.predict(X_test[-1, :].reshape(-1, n_steps, 1)) #shape (1, 50, 10)
y_pred = y_pred[-1,-1,:].reshape(-1,1)
y_pred_rescaled = scaler.inverse_transform(y_pred).reshape(-1, 1) #shape (10, 1)
print(y pred rescaled.shape)
#t.ime
time not reshaped = np.array(data west['TIME'][:(len(data west['TIME']) - (len(data west[
time reshaped = np.array(data west['TIME'][:(len(data west['TIME']) - (len(data west['TIME'))
    reshape(-1, (n steps+forecast), 1)
valid_time = int(0.9 * time_reshaped.shape[0])
y time test = time reshaped[valid time:, n steps:, 0]
print(y time test[-1, :].shape)
def plot prediction(y real resacled, y pred rescaled, flow not reshaped, time not reshape
    plt.figure()
    plt.title("50 minutes prediction", fontsize=14)
    plt.plot(time not reshaped[-300:-forecast], flow not reshaped[-300:-forecast], 'b-')
    plt.plot(y_time_test[-1, :], y_real_resacled, 'ro-', label = 'Real values')
plt.plot(y_time_test[-1, :], y_pred_rescaled, 'gx-', label = 'Predicted values')
    plt.legend(loc="upper left")
    plt.xlabel("Time (in 5 minutes intervals)")
    plt.ylabel('Volume (veh/h)')
plot prediction(y real rescaled, y pred rescaled, flow not reshaped, time not reshaped, y
plt.show()
```

C→

```
(17168,)
(10197, 60, 1)
(1020, 50, 1)
(1, 50, 1)
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/pytho
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the
Train on 7137 samples, validate on 2040 samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/pytho
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Epoch 1/700
Epoch 2/700
Epoch 3/700
Epoch 4/700
Epoch 5/700
Epoch 6/700
Epoch 7/700
Epoch 8/700
Epoch 9/700
Epoch 10/700
Epoch 11/700
Epoch 12/700
Epoch 13/700
Epoch 14/700
Epoch 15/700
Epoch 16/700
Epoch 17/700
Epoch 18/700
Epoch 19/700
Epoch 20/700
Epoch 21/700
Epoch 22/700
Epoch 23/700
Epoch 24/700
```

```
, + > , , , + > , |
       110 2mb, bampic
Epoch 25/700
Epoch 26/700
Epoch 27/700
Epoch 28/700
Epoch 29/700
Epoch 30/700
Epoch 31/700
Epoch 32/700
Epoch 33/700
Epoch 34/700
Epoch 35/700
Epoch 36/700
Epoch 37/700
Epoch 38/700
Epoch 39/700
Epoch 40/700
Epoch 41/700
Epoch 42/700
Epoch 43/700
Epoch 44/700
Epoch 45/700
Epoch 46/700
Epoch 47/700
Epoch 48/700
Epoch 49/700
Epoch 50/700
Epoch 51/700
Epoch 52/700
Epoch 53/700
Epoch 54/700
```

```
Epoch 55/700
Epoch 56/700
Epoch 57/700
Epoch 58/700
Epoch 59/700
Epoch 60/700
Epoch 61/700
Epoch 62/700
Epoch 63/700
Epoch 64/700
Epoch 65/700
Epoch 66/700
Epoch 67/700
Epoch 68/700
Epoch 69/700
Epoch 70/700
Epoch 71/700
Epoch 72/700
Epoch 73/700
Epoch 74/700
Epoch 75/700
Epoch 76/700
Epoch 77/700
Epoch 78/700
Epoch 79/700
Epoch 80/700
Epoch 81/700
Epoch 82/700
Epoch 83/700
```

```
Epoch 84/700
Epoch 85/700
Epoch 86/700
Epoch 87/700
Epoch 88/700
Epoch 89/700
Epoch 90/700
Epoch 91/700
Epoch 92/700
Epoch 93/700
Epoch 94/700
Epoch 95/700
Epoch 96/700
Epoch 97/700
Epoch 98/700
Epoch 99/700
Epoch 100/700
Epoch 101/700
Epoch 102/700
Epoch 103/700
Epoch 104/700
Epoch 105/700
Epoch 106/700
Epoch 107/700
Epoch 108/700
Epoch 109/700
Epoch 110/700
Epoch 111/700
Epoch 112/700
Epoch 113/700
```

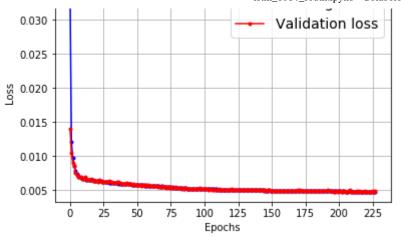
```
Epoch 114/700
Epoch 115/700
Epoch 116/700
Epoch 117/700
Epoch 118/700
Epoch 119/700
Epoch 120/700
Epoch 121/700
Epoch 122/700
Epoch 123/700
Epoch 124/700
Epoch 125/700
Epoch 126/700
Epoch 127/700
Epoch 128/700
Epoch 129/700
Epoch 130/700
Epoch 131/700
Epoch 132/700
Epoch 133/700
Epoch 134/700
Epoch 135/700
Epoch 136/700
Epoch 137/700
Epoch 138/700
Epoch 139/700
Epoch 140/700
Epoch 141/700
Epoch 142/700
```

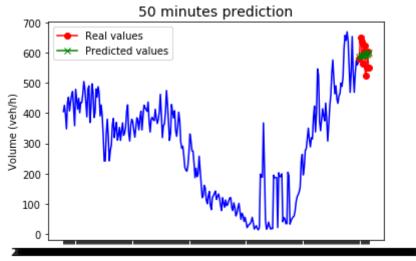
```
Epoch 143/700
Epoch 144/700
Epoch 145/700
Epoch 146/700
Epoch 147/700
Epoch 148/700
Epoch 149/700
Epoch 150/700
Epoch 151/700
Epoch 152/700
Epoch 153/700
Epoch 154/700
Epoch 155/700
Epoch 156/700
Epoch 157/700
Epoch 158/700
Epoch 159/700
Epoch 160/700
Epoch 161/700
Epoch 162/700
Epoch 163/700
Epoch 164/700
Epoch 165/700
Epoch 166/700
Epoch 167/700
Epoch 168/700
Epoch 169/700
Epoch 170/700
Epoch 171/700
Epoch 172/700
```

```
Epoch 173/700
Epoch 174/700
Epoch 175/700
Epoch 176/700
Epoch 177/700
Epoch 178/700
Epoch 179/700
Epoch 180/700
Epoch 181/700
Epoch 182/700
Epoch 183/700
Epoch 184/700
Epoch 185/700
Epoch 186/700
Epoch 187/700
Epoch 188/700
Epoch 189/700
Epoch 190/700
Epoch 191/700
Epoch 192/700
Epoch 193/700
Epoch 194/700
Epoch 195/700
Epoch 196/700
Epoch 197/700
Epoch 198/700
Epoch 199/700
Epoch 200/700
Epoch 201/700
```

```
Epoch 202/700
Epoch 203/700
Epoch 204/700
Epoch 205/700
Epoch 206/700
Epoch 207/700
Epoch 208/700
Epoch 209/700
Epoch 210/700
Epoch 211/700
Epoch 212/700
Epoch 213/700
Epoch 214/700
Epoch 215/700
Epoch 216/700
Epoch 217/700
Epoch 218/700
Epoch 219/700
Epoch 220/700
Epoch 221/700
Epoch 222/700
Epoch 223/700
Epoch 224/700
Epoch 225/700
Epoch 226/700
Epoch 227/700
Epoch 228/700
(10, 1)
(10, 1)
(10,)
0.035

    Training loss
```





Time (in 5 minutes intervals)