```
import numpy as np
import pandas as pd
import tensorflow as tf
from datetime import datetime,timedelta


pd.set_option("max_columns", 100)


def fecha_MDA(value):
    date=[]
    for i in value:
#         print(i)
        try:
            date.append(datetime.strptime(str(i), "%Y-%d-%m %H:%M:%S").strftime('%d-%m-%Y'))
        except:
            date.append(i.replace('/','-'))
#         print(type(date[-1]),date[-1])
    return date


def Upload_Dicc(name):
    with open(name,'r') as archivo:
        separador,dicc = '\t',{}
        for linea in archivo:
            key, value = linea.split(separador)
            dicc[key.strip()] = value.strip()
    return dicc


#


Data=pd.read_csv('BDHumidity.csv')
# Data['Date']=fecha_MDA(Data['Date'])
# Data['Time']=Data['Time'].astype(str)
# Data.insert(0,'Date_Time',pd.to_datetime(Data['Date']+'-'+Data['Time'],format='%m-%d-%Y-%H:
# Data=Data.drop(columns=['Date','Time'])
Data.head()
```

	Date	Time	Millitm	{[SUPREMA]MIC_202.Val_PV}	{[SUPREMA]MIC_202.Val_SP}	{[S
0	2022-02-02 00:00:00	23:30:59	124	NaN	NaN	
1	2022-02-02 00:00:00	23:31:59	125	90.931252	89.699997	
	2022					

```
Data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41667 entries, 0 to 41666
Data columns (total 27 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Date                          41667 non-null  object 
 1   Time                          41667 non-null  object 
 2   Millitm                       41667 non-null  int64  
 3   {[SUPREMA]MIC_202.Val_PV}     41666 non-null  float64
 4   {[SUPREMA]MIC_202.Val_SP}     41666 non-null  float64
 5   {[SUPREMA]MIC_203.Val_PV}     41666 non-null  float64
 6   {[SUPREMA]MIC_203.Val_SP}     41666 non-null  float64
 7   {[SUPREMA]MIC_302.Val_PV}     41666 non-null  float64
 8   {[SUPREMA]MIC_302.Val_SP}     41666 non-null  float64
 9   {[SUPREMA]MIC_304.Val_PV}     41666 non-null  float64
 10  {[SUPREMA]MIC_304.Val_SP}     41666 non-null  float64
 11  {[SUPREMA]TE_121.VALOR}       41667 non-null  int64  
 12  {[SUPREMA]TE_302.VALOR}       41667 non-null  float64
 13  {[SUPREMA]TE_305A.VALOR}      41667 non-null  float64
 14  {[SUPREMA]TE_401.VALOR}       41667 non-null  float64
 15  {[SUPREMA]TIC_121.OSet_SP}    41667 non-null  int64  
 16  {[SUPREMA]TIC_201.Val_PV}     41667 non-null  float64
 17  {[SUPREMA]TIC_201.Val_SP}     41667 non-null  float64
 18  {[SUPREMA]TIC_202.Val_PV}     41667 non-null  float64
 19  {[SUPREMA]TIC_202.Val_SP}     41667 non-null  float64
 20  {[SUPREMA]TIC_203.Val_PV}     41667 non-null  float64
 21  {[SUPREMA]TIC_203.Val_SP}     41667 non-null  float64
 22  {[SUPREMA]TIC_302.OSet_SP}    41667 non-null  float64
 23  {[SUPREMA]TIC_305.Val_PV}     41667 non-null  float64
 24  {[SUPREMA]TIC_305.Val_SP}     41667 non-null  float64
 25  {[SUPREMA]TIC_305A.OSet_SP}   41667 non-null  float64
 26  {[SUPREMA]TIC_401.OSet_SP}    41667 non-null  int64  
dtypes: float64(21), int64(4), object(2)
memory usage: 8.6+ MB
```

```
Data.shape
```

```
(41667, 27)
```

```
Dicc_est=Upload_Dicc('Diccionario Estaciones.txt')
Dicc_deltat=Upload_Dicc('Diccionario Deltat.txt')
```

```
Data_filter=(Data
             .rename(columns=Dicc_est)
             .filter(regex=(r'\A\w{2}\W?\w{3}|\A\w{4,7}'))
             .drop(index=0)
             .reset_index(drop=True)
            )
Data_filter.head()
```

	Date	Time	Millitm	ME-202	ME-203	ME-302	ME-304	TE-302	TE
0	2022-02-02 00:00:00	23:31:59	125	90.931252	83.087502	69.606255	77.006248	78.658386	64.73
1	2022-02-02 00:00:00	23:32:59	126	90.706253	82.700005	69.731255	77.012505	79.811981	61.98
2	2022-02-02	23:33:59	127	91.206253	82.849998	68.962502	77.087502	80.020264	64.36

```
Data_filter.shape
```

```
(41666, 11)
```

```
time_ini,time_end,TE_201,ME_202,TE_202,ME_203,TE_203,TE_302,ME_302,ME_304=[],[],[],[],[],[],[],[],[],[]
rango=[TE_201,ME_202,TE_202,ME_203,TE_203,TE_302,ME_302,ME_304]
apuntador=0
# for i in range(1000):
for i in Data_filter.index:
    try:
        time_ini.append(Data_filter.loc[i,'Date'])
        time_end.append(Data_filter.loc[i,'Time'])
        for j,k in enumerate(Dicc_deltat.keys()):
            apuntador=i+int(Dicc_deltat[k])
#             print(j,k,apuntador)
            rango[j].append(Data_filter.loc[apuntador,k])
    except:
        break;
time_ini.pop(-1)
time_end.pop(-1)
```

```
datetime.time(6, 35, 43)
```

```
for i in range(len(rango)-1):
    rango[i].pop(-1)
#     print(len(rango[i]))
```

```
Data_Final=pd.DataFrame({'Date':time_ini,'Time':time_end,'TE-201':TE_201,'ME-202':ME_202,'TE-
```

```
                          'TE-302':TE_302,'ME-302':ME_302,'ME-304':ME_304})
Data_Final=Data_Final.set_index(['Date','Time'])

Data_Final.head()
```

Date	Time	TE-201	ME-202	TE-202	ME-203	TE-203	TE-302	ME
2022-02-02 00:00:00	23:31:59	64.736267	90.818748	75.437927	82.206253	77.408661	77.328552	69.71
	23:32:59	61.988647	90.656250	72.553925	81.956253	76.960022	79.683807	69.16
	23:33:59	64.366730	90.387505	75.245666	81.818748	76.863892	77.056183	69.33
	23:34:59	63.659729	88.781250	74.460571	82.837502	77.600922	79.587677	69.05
	23:35:59	62.727783	88.781250	72.537903	82.868752	76.399261	77.536835	68.90

```
Data_Final.shape
```

```
(41463, 8)
```

```
Data_Final.to_csv('BDHumidity-processed.csv')


def preprocess(BD):
    data = list(tf.constant(BD, dtype="float32"))
    return tf.data.Dataset.from_tensor_slices(data).map(lambda seq: (tf.reshape(seq[:-1], (-1

dataset1 = preprocess( Data_Final[['TE-201', 'ME-202', 'TE-202', 'ME-203', 'TE-203', 'TE-302'


dataset1
```

```
<BatchDataset element_spec=(TensorSpec(shape=(64, 7, 1), dtype=tf.float32, name=None), T
```

```
for X, y in dataset1.take(1):
    print("X")
    print(X)
    print("y")
    print(y)
```

```
X
tf.Tensor(
[[[64.73627 ]
  [90.81875 ]
  [75.43793 ]
  [82.20625 ]
  [77.40866 ]
```

```
         [77.32855 ]
         [69.71875 ]]

        [[61.988647]
         [90.65625 ]
         [72.553925]
         [81.95625 ]
         [76.96002 ]
         [79.68381 ]
         [69.168755]]

        [[64.36673 ]
         [90.387505]
         [75.24567 ]
         [81.81875 ]
         [76.86389 ]
         [77.05618 ]
         [69.33125 ]]

        [[63.65973 ]
         [88.78125 ]
         [74.46057 ]
         [82.8375  ]
         [77.60092 ]
         [79.58768 ]
         [69.05    ]]

        [[62.727783]
         [88.78125 ]
         [72.5379  ]
         [82.86875 ]
         [76.39926 ]
         [77.536835]
         [68.90625 ]]

        [[64.89697 ]
         [90.525   ]
         [75.30975 ]
         [83.44375 ]
         [77.84125 ]
         [80.6131  ]
         [69.4875  ]]

        [[64.591675]
         [88.5375  ]
         [72.98651 ]
         [81.5625  ]
         [76.09485 ]
         [78.86667 ]
         [69.3125  ]]
```

```
Data_Final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 41463 entries, (datetime.datetime(2022, 2, 2, 0, 0), datetime.time(23, 31, 5
```

```
Data columns (total 8 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   TE-201  41463 non-null  float64
 1   ME-202  41463 non-null  float64
 2   TE-202  41463 non-null  float64
 3   ME-203  41463 non-null  float64
 4   TE-203  41463 non-null  float64
 5   TE-302  41463 non-null  float64
 6   ME-302  41463 non-null  float64
 7   ME-304  41463 non-null  float64
dtypes: float64(8)
memory usage: 4.0+ MB
```