The package used below will have to be explicitly imported

Package name: TSForecasting

All the packages mentioned below will be installed automatically installed in case they are missing :

- pandas
- numpy
- matplotlib
- dateutil
- re

The pandas package that is explicitly imported will be used to read the data supplied by you

```
In [1]: from TSForecasting.TsForecasting import TimeSeriesForecast
        import pandas as pd
```

```
Using TensorFlow backend.
```

===

To read the documentation of the class, just run the following code

```
In [2]: TimeSeriesForecast?
```

Now, we create an object of the TimeSeriesForecast class

This object contains the model and methods to predict the imputed values

The object has a pretrained model on the data described below. The model is capable of performing imputations

```
In [3]: obj = TimeSeriesForecast()
```

```
Epoch 1/1
212852/212852 [==============================] - 436s 2ms/step - loss:
5.9258e-04
```

The following method describes data that has been consolidaed from the CGMAnalyzer, Gluvarpro and the Ohio datasets

In [4]:

```
obj.dataDescribe()
```

```
Here is a glimpse of the data:

         Display Time subjectId  GlucoseValue
0 2016-06-21 00:00:00     GVP01         197.0
1 2016-06-21 00:15:00     GVP01         208.0
2 2016-06-21 00:30:00     GVP01         226.0
3 2016-06-21 00:45:00     GVP01         241.0
4 2016-06-21 01:00:00     GVP01         250.0
```

	Length of readings	Max. Glucose Value	Min. Glucose Value	MAGE Score	Days
count	28.000000	28.000000	28.000000	28.000000	28
mean	7601.892857	317.106071	49.691429	108.196373	42 days 14:19:02.678571
std	8762.341947	98.519774	22.217624	45.395263	104 days 12:44:49.738624
min	576.000000	163.800000	0.000000	33.992727	3 days 00:06:00
25%	2181.500000	211.150000	40.000000	65.397143	5 days 12:03:00
50%	3178.500000	346.000000	42.500000	120.877433	13 days 18:54:00
75%	10903.250000	400.000000	62.070000	139.814124	45 days 11:30:00
max	45696.000000	508.710000	95.400000	189.593750	566 days 23:45:00

```
Here is the statistical analysis of the data:
```

	Subject ID	Length of readings	Max. Glucose Value	Min. Glucose Value	MAGE Score	Days	Start	End
0	OD596	10877	367.00	40.00	121.875622	47 days 07:44:00	2027-04-09 16:15:00	2027-05-26 23:59:00
1	OD591	10847	397.00	40.00	133.250923	44 days 06:52:00	2021-11-30 17:06:00	2022-01-13 23:58:00
2	OD588	12640	400.00	40.00	112.819398	45 days 12:02:00	2021-08-30 11:53:00	2021-10-14 23:55:00
3	OD584	12150	400.00	40.00	143.566308	45 days 23:53:00	2025-05-14 00:03:00	2025-06-28 23:56:00
4	OD575	11866	400.00	40.00	140.910156	45 days 11:51:00	2021-11-17 12:04:00	2022-01-01 23:55:00
5	OD570	10982	377.00	46.00	145.633333	40 days 07:30:00	2021-12-07 16:29:00	2022-01-16 23:59:00
6	OD567	10858	400.00	40.00	145.016461	46 days 23:54:51	2026-12-28 00:04:30	2027-02-12 23:59:21
7	OD563	12124	400.00	40.00	119.387931	45 days 11:23:00	2021-09-13 12:33:00	2021-10-28 23:56:00
8	OD559	10796	400.00	40.00	179.030303	41 days 22:39:00	2021-12-07 01:17:00	2022-01-17 23:56:00
9	OD552	9080	345.00	45.00	125.322917	38 days 12:42:09	2025-04-16 11:17:05	2025-05-24 23:59:14
10	OD544	10623	400.00	48.00	139.448780	43 days 23:53:00	2027-05-11 00:02:00	2027-06-23 23:55:00
11	OD540	11947	369.00	40.00	125.281250	45 days 12:20:15	2027-05-19 11:36:29	2027-07-03 23:56:44
12	ID33	1265	163.80	39.60	33.992727	13 days 15:49:00	2019-02-21 14:59:00	2019-03-07 06:48:00
13	ID32	1337	202.00	47.00	55.223881	13 days 21:59:00	2018-08-01 12:00:00	2018-08-15 09:59:00
14	ID31	1443	182.19	66.47	48.568387	3 days 00:06:00	2016-06-06 16:55:00	2016-06-09 17:01:00
15	ID30	2472	187.20	59.40	39.564706	5 days 03:33:00	2016-12-21 07:29:00	2016-12-26 11:02:00

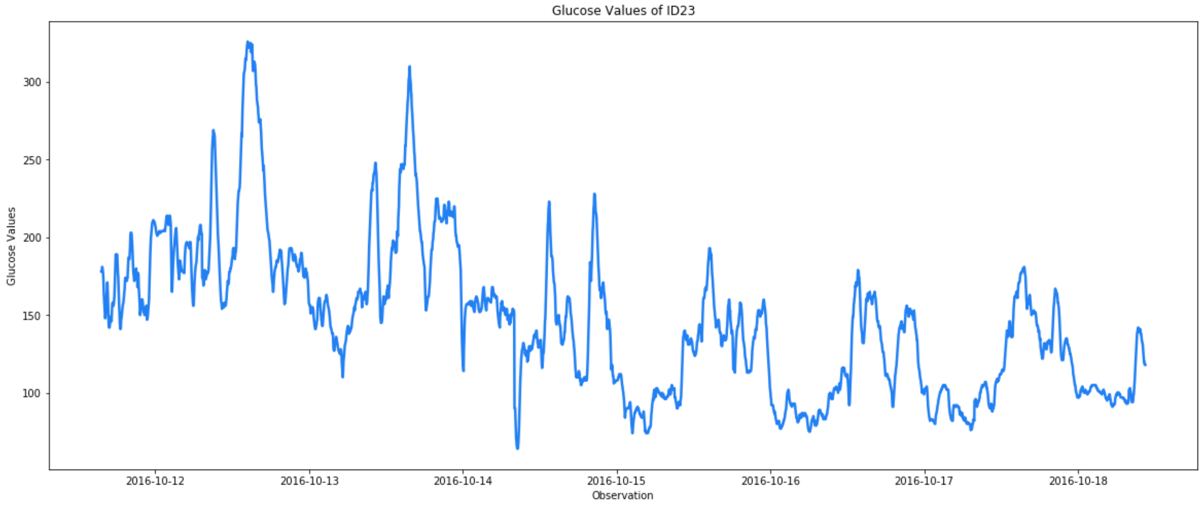
	Subject ID	Length of readings	Max. Glucose Value	Min. Glucose Value	MAGE Score	Days	Start	End
16	ID29	1728	195.64	75.15	57.160000	3 days 14:21:00	2016-12-19 22:42:00	2016-12-23 13:03:00
17	ID23	3260	326.83	64.53	87.135429	6 days 18:57:00	2016-10-11 15:34:00	2016-10-18 10:31:00
18	ID22	2111	170.39	34.20	56.753500	4 days 09:30:00	2016-12-29 07:35:00	2017-01-02 17:05:00
19	ID21	2686	214.20	66.60	68.142857	5 days 14:15:00	2016-11-22 07:26:00	2016-11-27 21:41:00
20	ID13	1681	246.06	94.75	81.946286	3 days 12:00:00	2016-12-01 07:44:00	2016-12-04 19:44:00
21	ID12	2980	271.80	93.60	78.646154	6 days 04:57:00	2016-08-02 08:19:00	2016-08-08 13:16:00
22	ID11	2205	316.80	95.40	124.305882	4 days 14:12:00	2017-01-07 08:52:00	2017-01-11 23:04:00
23	ID03	3097	508.71	61.25	179.607679	6 days 10:48:00	2016-05-26 06:09:00	2016-06-01 16:57:00
24	ID02	3016	170.98	14.02	40.234656	6 days 06:45:00	2016-05-26 07:50:00	2016-06-01 14:35:00
25	ID01	2510	291.37	55.39	137.199913	5 days 05:27:00	2016-06-25 09:14:00	2016-06-30 14:41:00
26	GVP03	45696	429.00	25.00	119.879244	566 days 23:45:00	2016-04-24 00:00:00	2017-11-11 23:45:00
27	GVP01	576	347.00	0.00	189.593750	5 days 23:45:00	2016-06-21 00:00:00	2016-06-26 23:45:00

To see what a specific individual's glucose values look like, run the following line.

In between the quotes, pass the Subject ID of that individual

The different subject ID's can be found in the description table above

In [5]:
```
obj.plotSpecific('ID23')
```



To test the pretrained model, write the path of the testing dataset below:

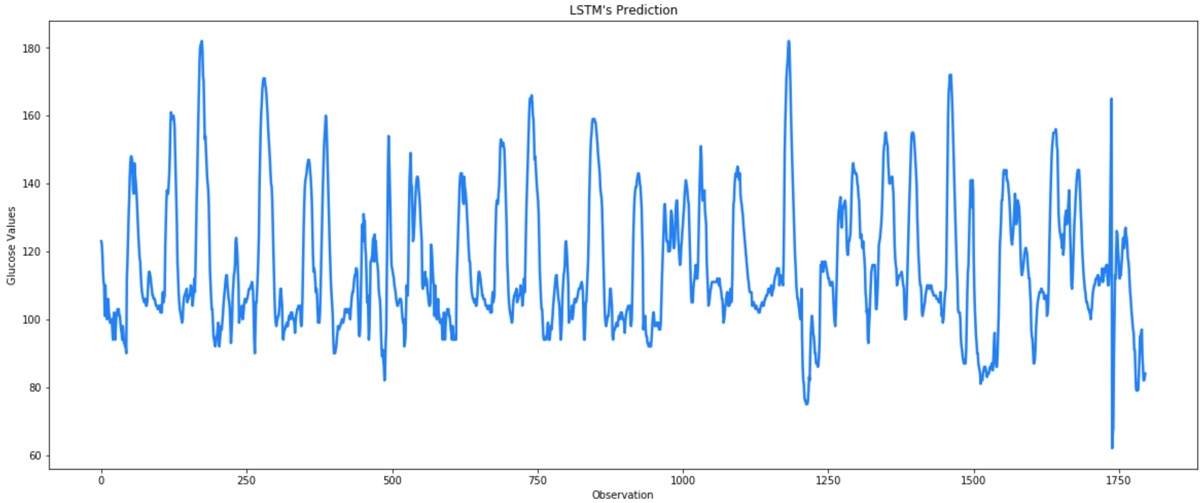
In [6]:
```
testing_data = pd.read_csv("~/Desktop/NCSA_genomics/Python - notebooks/T
SForecasting/Data/test/generated_test.txt", sep=",")
```

Now, we run the imputation model on the data you supplied above

The graph you see is the imputed glucose values

```
In [7]: obj.impute(testing_data)
```

```
Gap detected!
566/566 [==============================] - 0s 860us/step
Imputations performed!
File saved!
Location:/Users/snehgajiwala/Desktop/NCSA_genomics/Python - notebooks/T
SForecasting/Data/Output/ImputedValues.csv
```



==

In case you wish to train your own model, use the following code

Write the path to your file for the training and test set

Be careful of the following:

1. Enter the path of your file
2. The file should have only two columns in the following order with the exact names: Timestamp, GlucoseValue
3. The date of the timestamp should have the entire year (all 4 digits). Make sure the data is numeric (irrespective of the delimiter)

```
In [8]: training_data = pd.read_csv("~/Desktop/NCSA_genomics/Python - notebooks/
        TSForecasting/Data/test/Training-TestFile.csv")
```

Use the following function to train the model on the data you just supplied

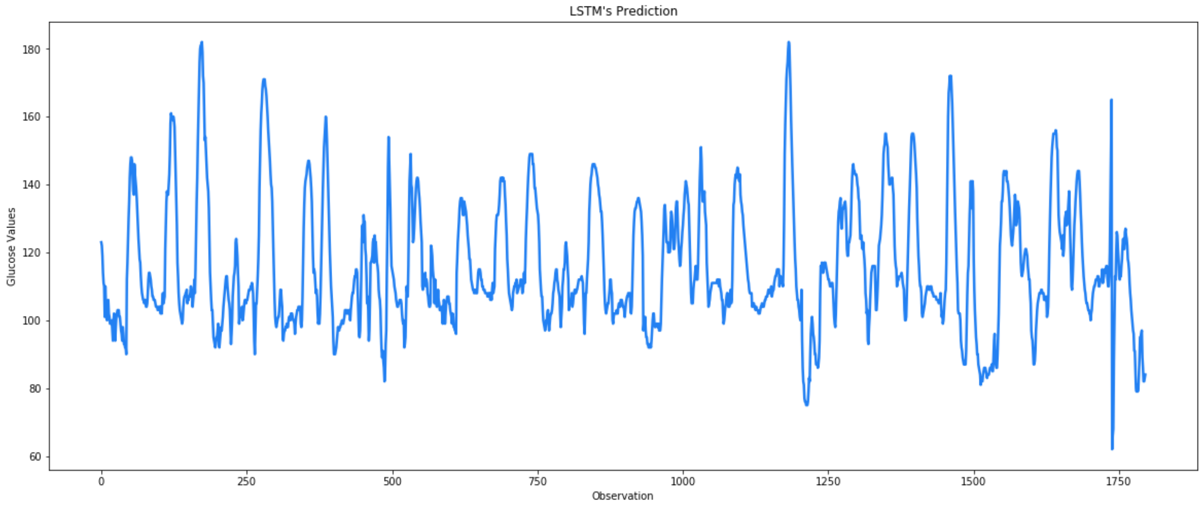
```
In [9]: obj.train(training_data)
```

```
Train shape:  (1795, 1, 1)
Epoch 1/1
1795/1795 [==============================] - 4s 2ms/step - loss: 0.0188
```

To test the model you just trained, just run it on the training set we imported earlier:

```
In [10]: obj.impute(testing_data)
```

```
Gap detected!
566/566 [==============================] - 1s 887us/step
Imputations performed!
File saved!
Location:/Users/snehgajiwala/Desktop/NCSA_genomics/Python - notebooks/T
SForecasting/Data/Output/ImputedValues.csv
```



Let's look at the various data we have used in our consolidated dataset:

OHIO Dataset

In [11]:

```
obj.ohio_full
```

Out[11]:

	Display Time	subjectId	GlucoseValue
0	2021-12-07 16:29:00	OD570	101
1	2021-12-07 16:34:00	OD570	100
2	2021-12-07 16:39:00	OD570	100
3	2021-12-07 16:44:00	OD570	99
4	2021-12-07 16:49:00	OD570	98
...	...	...	...
166528	2025-06-07 16:49:51	OD552	238
166529	2025-06-07 16:54:51	OD552	233
166530	2025-06-07 16:59:51	OD552	229
166531	2025-06-07 17:04:51	OD552	224
166532	2025-06-07 17:09:51	OD552	215

166533 rows × 3 columns

CGM Analyzer and CGM Analysis Dataset

In [12]:

```
obj.cgm_appended
```

Out[12]:

	Display Time	subjectId	GlucoseValue
0	2016-06-25 09:14:00	ID01	108.000
1	2016-06-25 09:17:00	ID01	108.000
2	2016-06-25 09:20:00	ID01	108.000
3	2016-06-25 09:23:00	ID01	107.658
4	2016-06-25 09:26:00	ID01	107.496
...	...	...	...
31786	2019-03-07 05:48:00	ID33	79.200
31787	2019-03-07 06:03:00	ID33	82.800
31788	2019-03-07 06:18:00	ID33	84.600
31789	2019-03-07 06:33:00	ID33	82.800
31790	2019-03-07 06:48:00	ID33	79.200

31791 rows × 3 columns

GluVarPro Dataset

```
In [13]: obj.gluvarpro
```

```
Out[13]:
                Display Time  subjectId  GlucoseValue
      0  2016-06-21 00:00:00      GVP01         197.0
      1  2016-06-21 00:15:00      GVP01         208.0
      2  2016-06-21 00:30:00      GVP01         226.0
      3  2016-06-21 00:45:00      GVP01         241.0
      4  2016-06-21 01:00:00      GVP01         250.0
    ...                  ...        ...           ...
  46267  2017-11-11 22:45:00      GVP03          86.0
  46268  2017-11-11 23:00:00      GVP03         149.0
  46269  2017-11-11 23:15:00      GVP03         178.0
  46270  2017-11-11 23:30:00      GVP03         174.0
  46271  2017-11-11 23:45:00      GVP03         174.0

46272 rows × 3 columns
```

Compiled Dataset:

In [14]:

```
obj.consolidated_paper
```

Out[14]:

	Display Time	subjectId	GlucoseValue
0	2016-06-21 00:00:00	GVP01	197.0
1	2016-06-21 00:15:00	GVP01	208.0
2	2016-06-21 00:30:00	GVP01	226.0
3	2016-06-21 00:45:00	GVP01	241.0
4	2016-06-21 01:00:00	GVP01	250.0
...	...	...	...
212848	2025-06-28 23:36:00	OD584	206.0
212849	2025-06-28 23:41:00	OD584	208.0
212850	2025-06-28 23:46:00	OD584	213.0
212851	2025-06-28 23:51:00	OD584	224.0
212852	2025-06-28 23:56:00	OD584	235.0

212853 rows × 3 columns