به نام خدا

گزارش پروژه درس داده کاوی

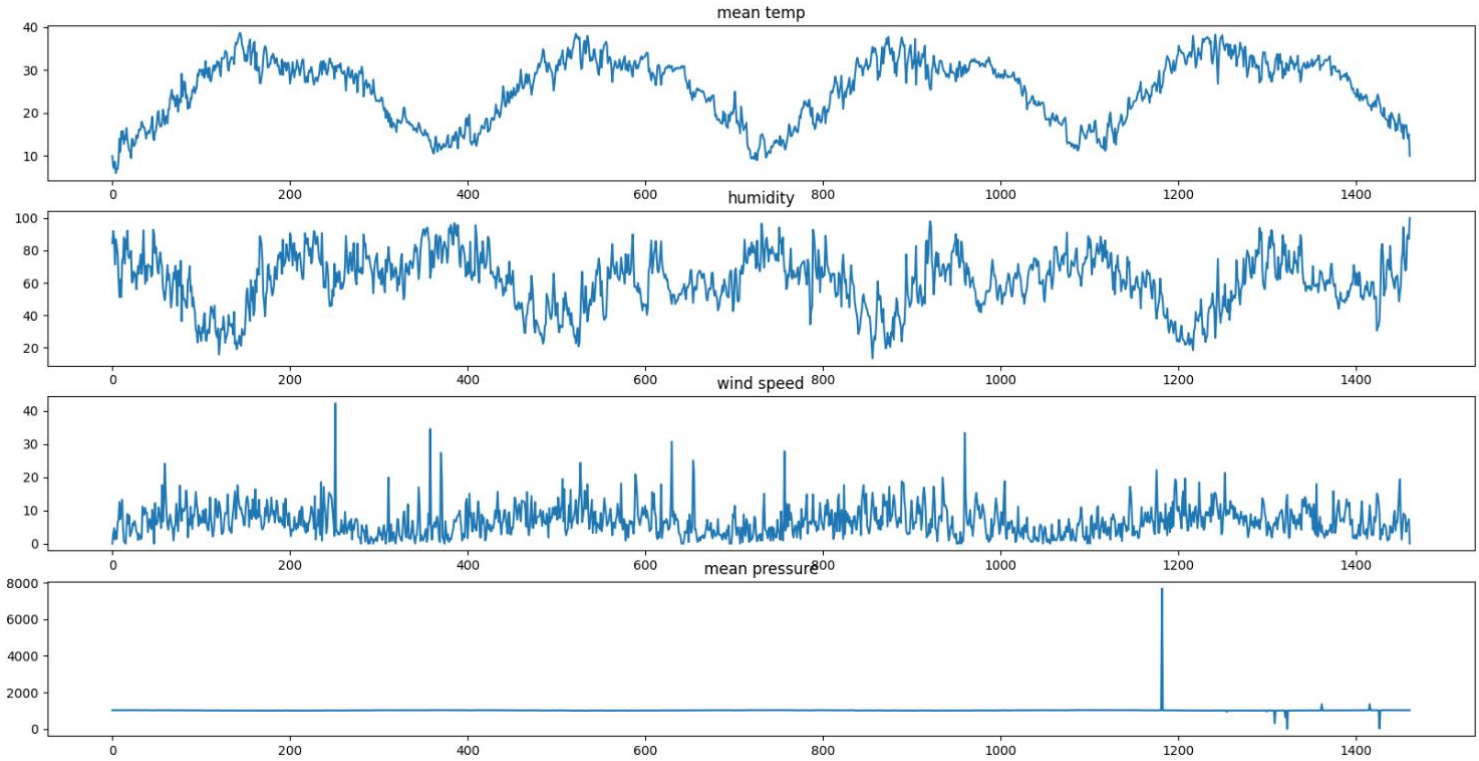
نویسنده: رضا دلیر

استاد: دکتر هدیه ساجدی

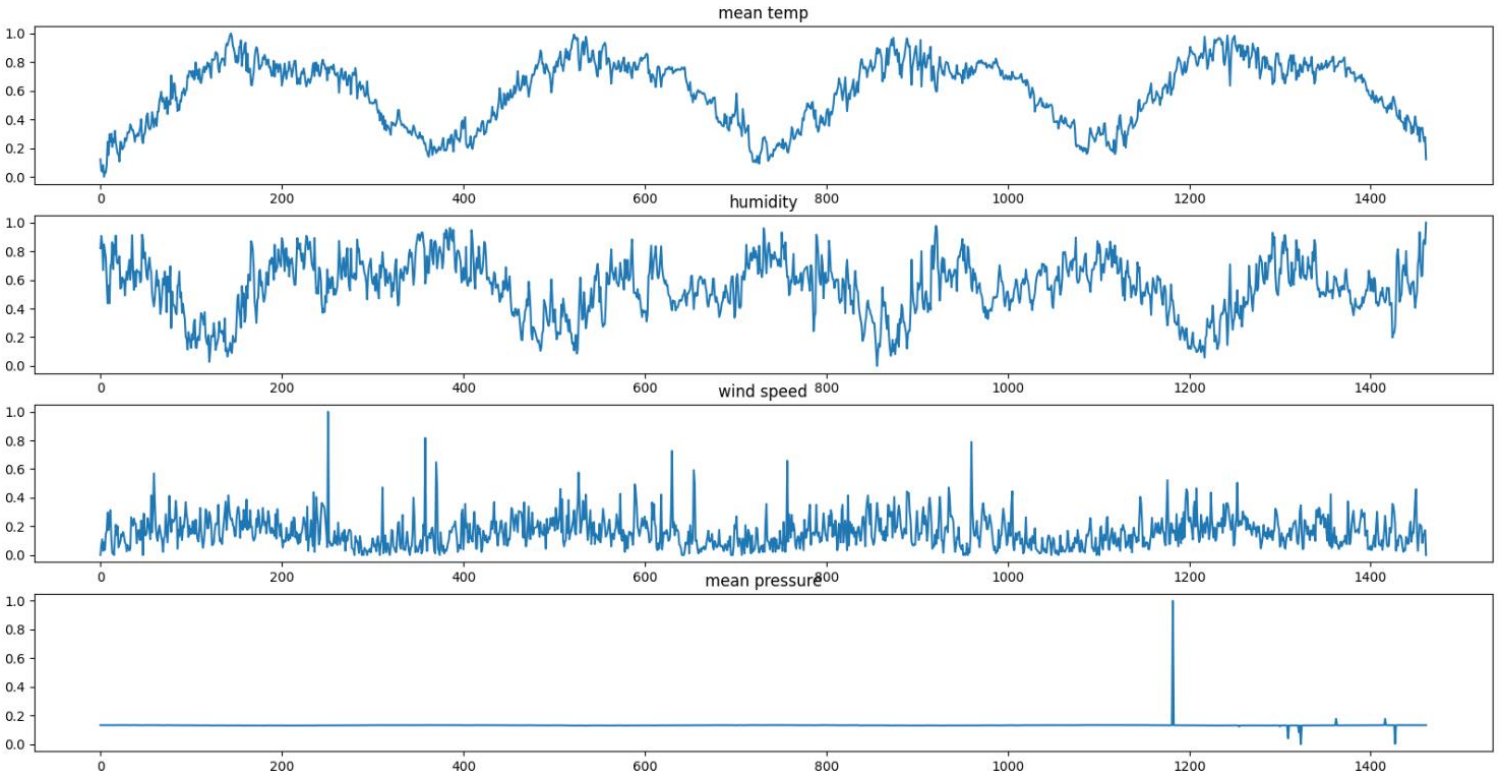
در این پروژه هدف این است که داده های آب و هوایی شهر دهلی هند را که به صورت sequential است را با مدلی ترین کنیم و سپس بر روی داده های تست فشار هوای میانگین در 124 روز آینده آن را پیشبینی کنیم.

در ابتدا نیاز است که بر روی داده ها پیش پردازش انجام شود. اما برای اینکه ببینیم چه پیش پردازش هایی مناسب تر است نیاز است که داده ها را رسم کنیم و ببینیم که به چه چیزی نیاز است.
همچنین پیش از آن داده ها را طبق تاریخ آنها مرتب میکنیم و ستون تاریخ را از داده ها حذف میکنم زیرا در ترین شدن و پیشبینی به آن نیاز نیست.

نتیجه نمایان کردن فیچر ها یک به یک به صورت زیر است:



همانطور که در نمودار مشخص است محور عمودی نمودار ها از رنج متفاوتی از اعداد تشکیل شده اند و بنابراین اعداد بزرگتر تاثیر بیشتری در مدل سازی ما خواهند داشت. پس برای اینکه در این مورد به مشکل نخوریم نیاز است که فیچر ها را اسکیل کنیم. بدین منظور از minmaxscaler استفاده میکنیم که رنج هر کدام از فیچر ها را بین صفر تا یک قرار دهیم. پس از انجام این کار نمودار مشابه قسمت قبل بدست می آید که به صورت زیر است:

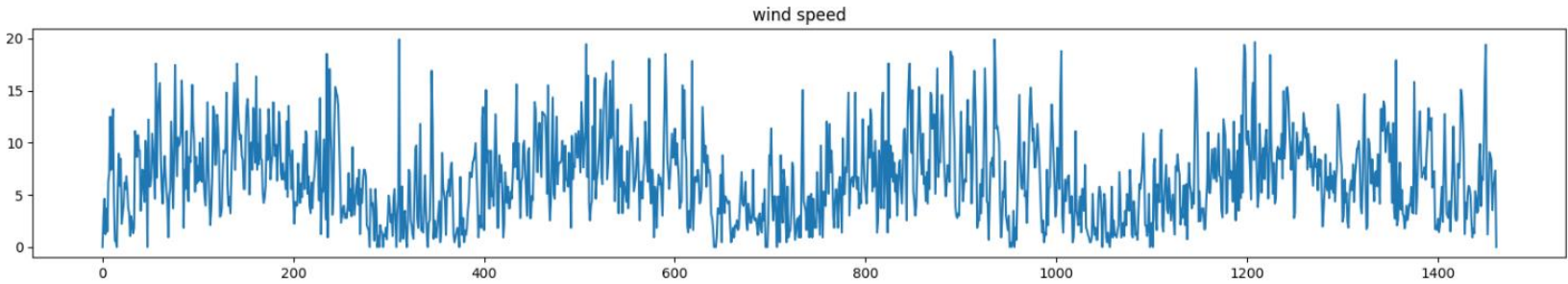


همانطور که مشاهده میشود محور عمودی این نمودار ها به بازه ای بین صفر تا یک اسکیل شده اند و در ترین کردن از این بابت به مشکلی نمیخوریم.

اما مشکل اساسی دیگری که وجود دارد این است که در این نمودار ها در بخش سرعت باد و مخصوصا در بخش میانگین فشار هوا داده هایی پرت وجود دارد که نمودار را خراب میکند و در فرایند ترین کردن مدل برای ما مشکل به وجود می آورد و دقت ما را کاهش میدهد.

برای جلوگیری از این مورد نیاز است که داده های پرت را حذف کنیم.

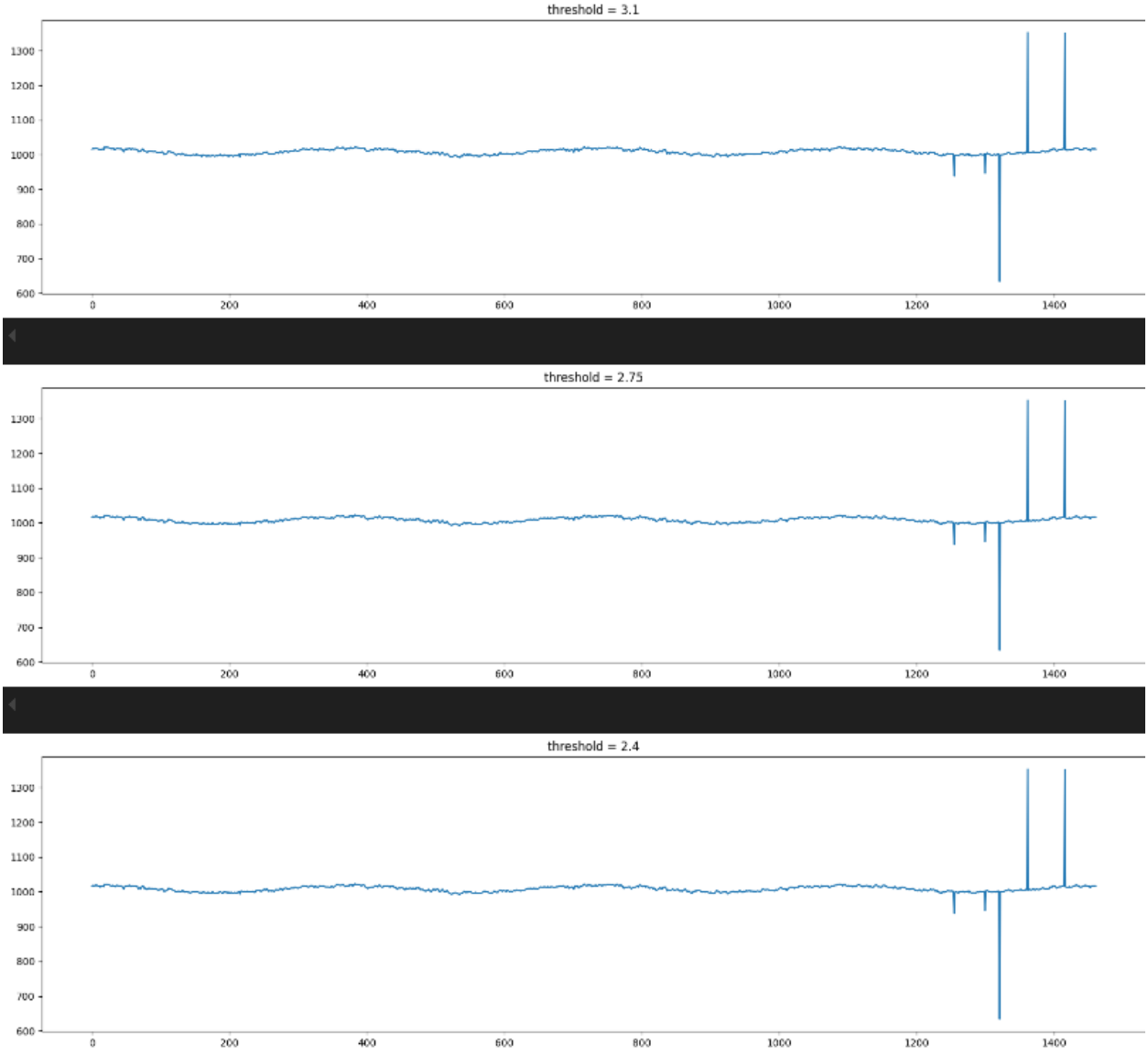
به منظور حذف داده های پرت در سرعت باد از معیار z-score استفاده میکنیم.

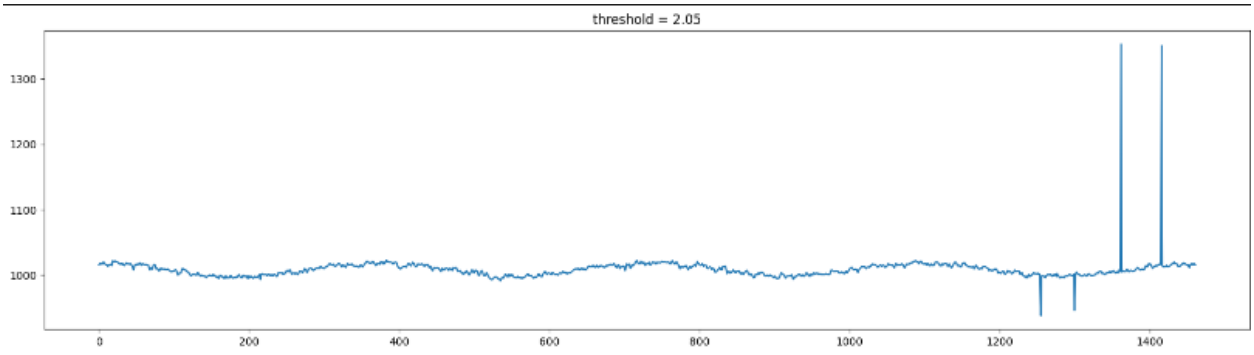


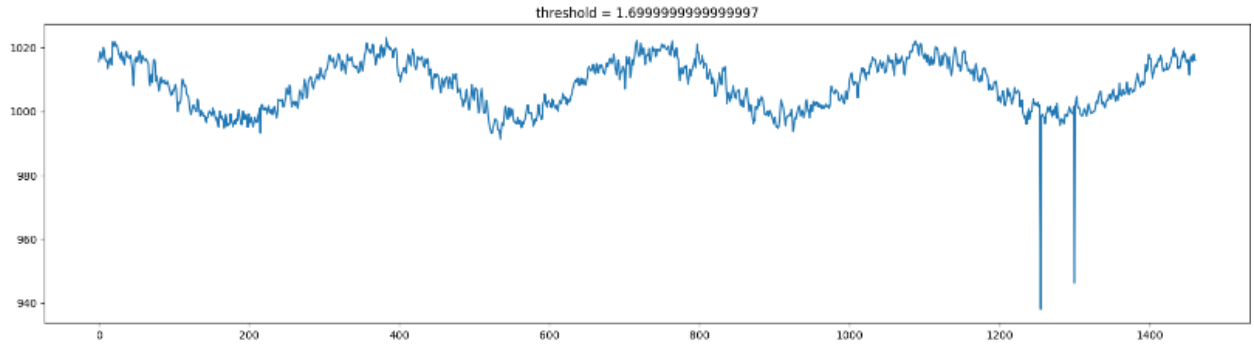
همانطور که مشاهده میشود نسبت به نمودار بالا تعدادی از داده های پرت که باعث پرش های شدید در نمودار شده بودند حذف شد. در این مورد ترشهولد برای حذف داده ها را عدد 3 در نظر گرفتیم یعنی داده هایی که z-score آنها از 3 بیشتر یا یا از 3- کمتر است حذف شدند.

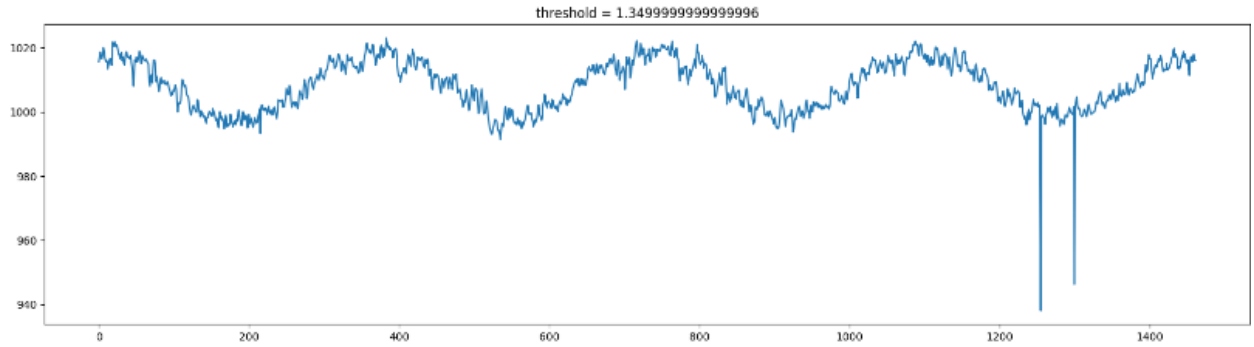
بخش اصلی و مهم تر در حذف داده های پرت حذف داده های پرت mean pressure است. بدین منظور همچنان از z-score استفاده میکنیم با این تفاوت که چون واریانس داده های اصلی در این فیچر بسیار کم است پس ترشهولد کوچکتری باید انتخاب شود. برای پیدا کردن ترشهولد مناسب در یک حلقه ترشهولد های مختلف را در نظر میگیریم و میبینیم که کدام یک داده های پرت را به خوبی حذف میکند.

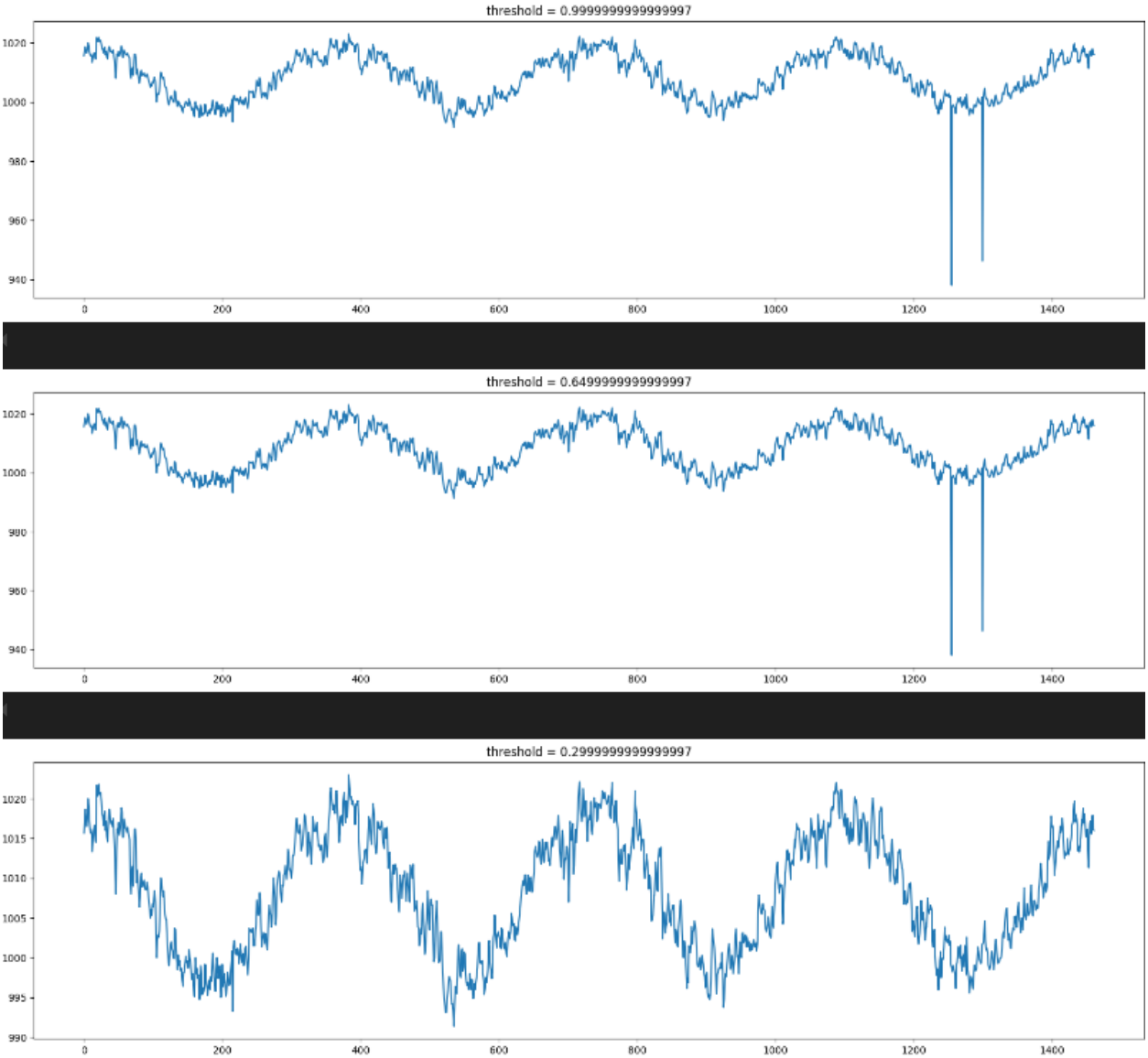
بنابراین نمودار ها به صورت زیر است:




threshold = 2.05
1300
1200
1100
1000
0
200
400
600
800
1000
1200
1400

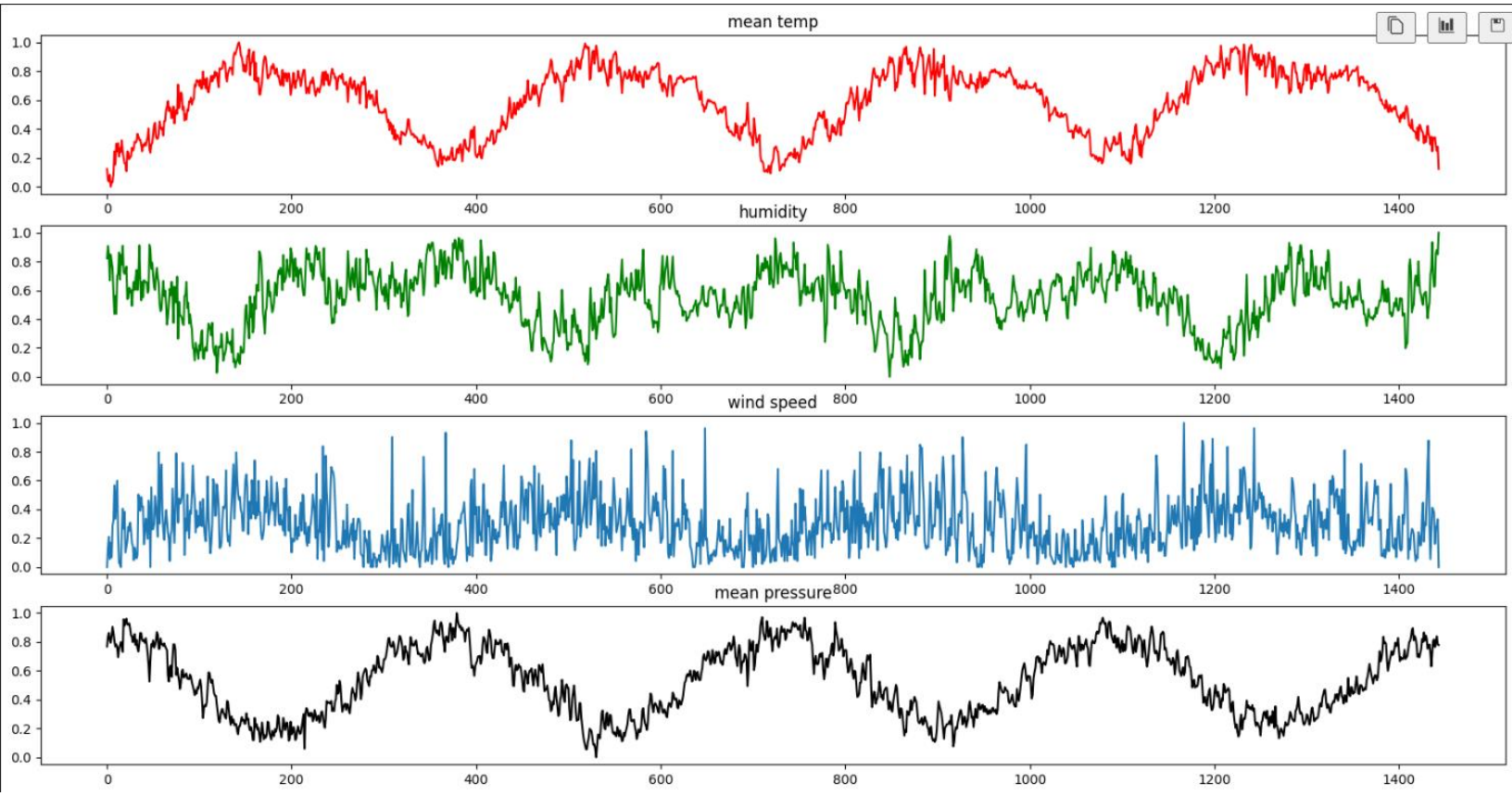

threshold = 1.6999999999999997
1020
1000
980
960
940
0
200
400
600
800
1000
1200
1400


threshold = 1.3499999999999996
1020
1000
980
960
940
0
200
400
600
800
1000
1200
1400



مشاهده میشود که با انتخاب ترشهولد های کوچکتر داده های پرت بیشتری حذف میشود و با انتخاب ترشهود 0.3 دیگر داده پرتی نداریم رنج میانگین فشار هوا بین 990 تا 1030 است.

بنابراین در مرحله بعد هر دو این موارد که در پیش پردازش نیاز بود را استفاده میکنیم و داده ها به صورت زیر در می آید:



همانطور که مشاهده میشود دیگر در این نمودار نه داده پرتی وجود دارد و همچنین محور عمودی نیز اسکیل شده است و میتواند در ترین مدل به خوبی استفاده شود.

نکته جالبی که در این نمودار به چشم میخورد رابطه کاملا معکوسی است که بین میانگین دما و لیبل ما که همان میانگین فشار است. البته با اندکی اغماض و ریزبینی بیشتر مشابه همین رابطه با سرعت باد نیز به چشم میخورد.

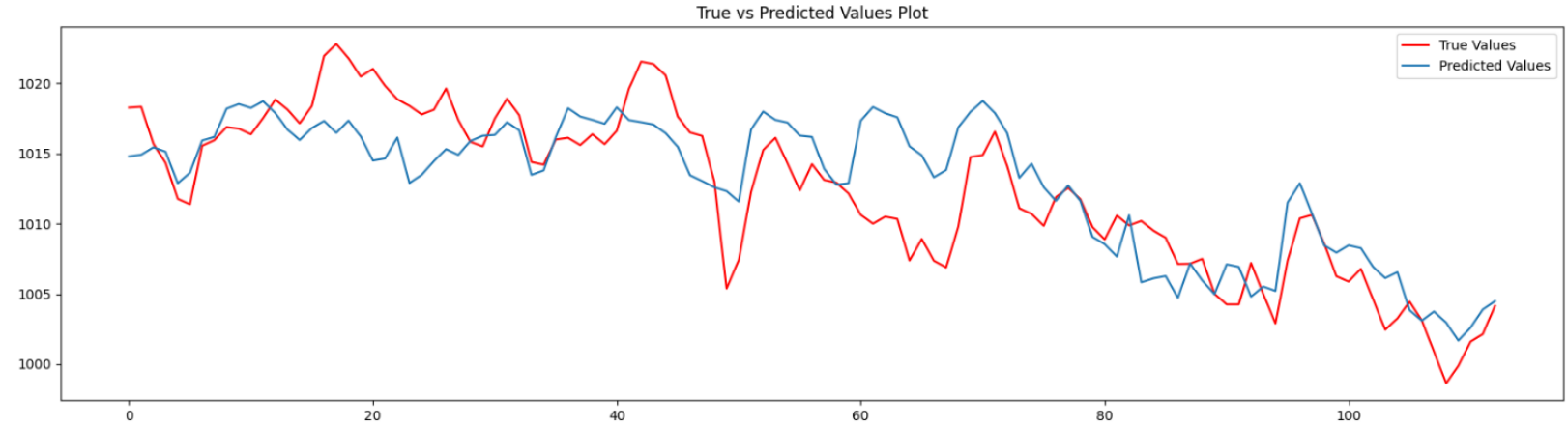
بنابراین مرحله پیش پردازش تمام شده و حال وارد فاز ترین کردن مدل میشویم.

از آنجا که داده ها به صورت سری است پس مدل های مناسبی که برای ترین کردن این داده ها میتوان استفاده کرد مدل هایی مثل RNN و یا LSTM است.

در ابتدا ساده ترین مدل LSTM را ترین میکنیم:

در این مدل تنها یک لایه پنهان داریم که شامل 4 نود است و همچنین از اکتیویشن فانکشن سیگموید استفاده کرده ایم. با در نظر گرفتن تعداد ایپاک 100 و بچ سایز برابر 5 این مدل با سرعت نسبتا بالایی ترین میشود.

سپس برای بررسی دقت این مدل داده های تست را وارد میکنیم. در داده های تست لیبل یکی از داده ها که فشار هوا است برابر با 50 ثبت شده است که این مورد در کره زمین در هیچ جایی امکان پذیر نمیباشد و به وضوح داده پرت است.
پس این داده را حذف میکنیم و بقیه ی داده ها را مطابق با اسکیلری که برای داده های ترین استفاده کرده بودیم، اسکیل میکنیم و سپس لیبل های داده های تست را به وسیله مدلی که ترین کرده بودیم پیشبینی میکنیم. نمودار مقادیر پیشبینی شده در مقابل مقادیر واقعی فشار هوا در داده تست در این مدل به صورت زیر است:



همانطور که مشاهده میشود تقریبا این مدل هم راستا در جهت مقادیر واقعی عمل میکند و بنابراین مدل درست ترین شده است اما با انتخاب پارامتر های مناسب تر میتوان دقت بیشتری را در این مدل کسب کرد. برای مقایسه مدل های مختلف نیاز به معیارهایی برای سنجش دقت داریم. بدین منظور از معیار های mae و mse و rmse و r2 استفاده میکنیم.

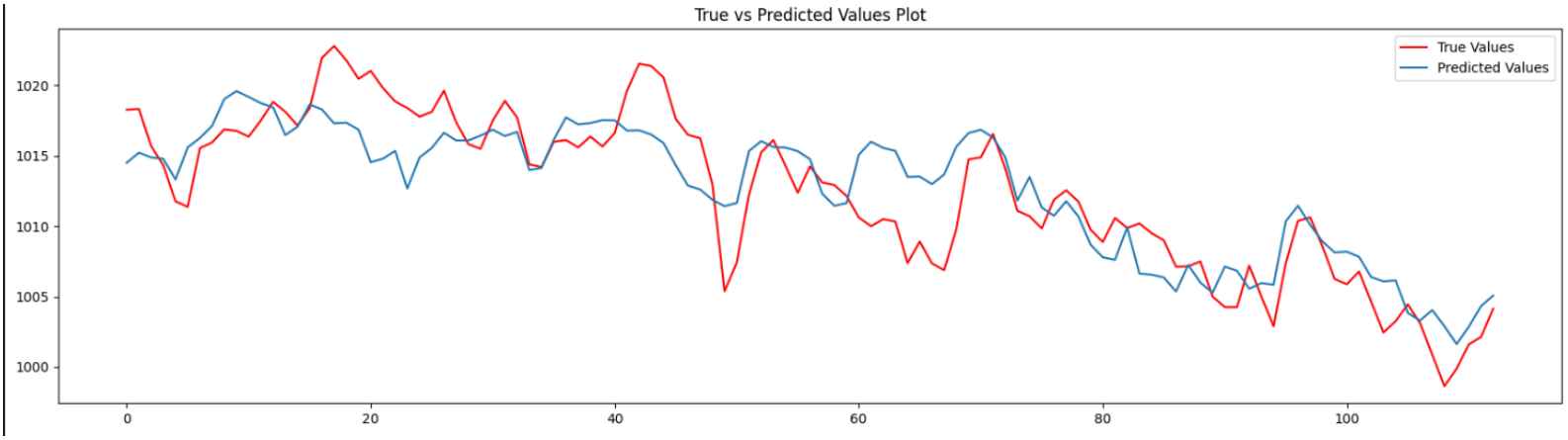
در این حالت مقادیر زیر برای این مدل بدست آمده است:

```
Mean Absolute Error: 2.30060958393844
Mean Squared Error: 8.725687443980767
Root Mean Squared Error: 2.953927460852884
R2 Error: 0.7299832070213325
```

که به نسبت مقادیر مناسب و خوبی هستند.

حال شبکه RNN دیگری که کمی پیچیده تر میباشد را امتحان میکنیم.
در این شبکه دو لایه عصبی و در هر لایه 4 نود قرار میدهیم و بقیه موارد را مانند دفعه قبل نگه میداریم.

نمودار و دقت به صورت زیر خواهد بود:



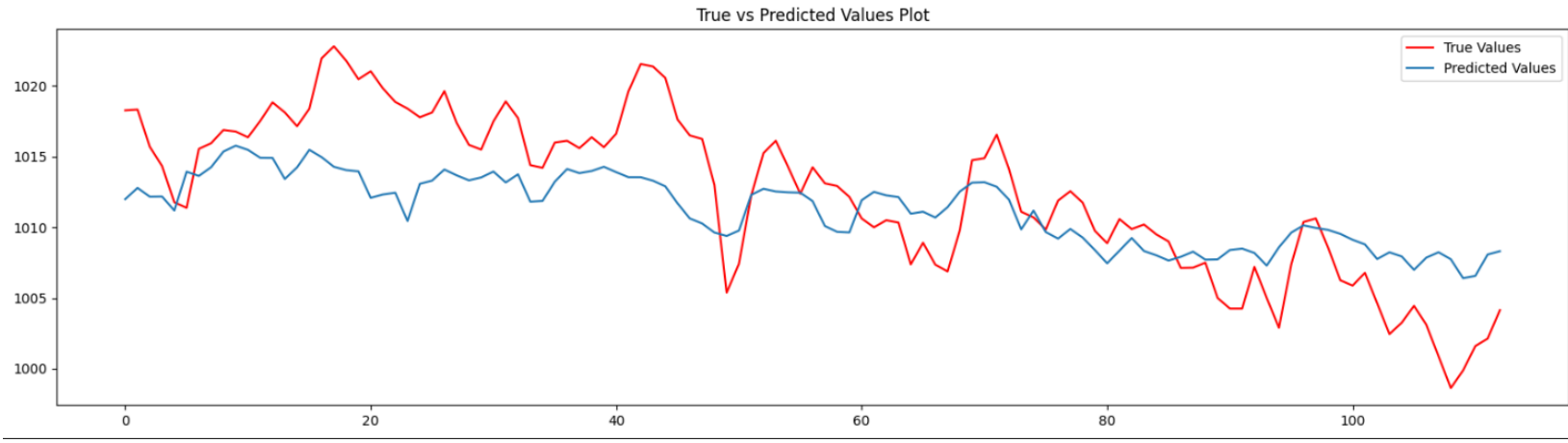

```
Mean Absolute Error: 2.328506396846301
Mean Squared Error: 8.450345940814072
Root Mean Squared Error: 2.9069478737696817
R2 Error: 0.7385036623019401
```

همانطور که مشاهده میشود در این مورد تفاوت چشمگیری با مورد قبل مشاهده نمیشود زیرا هنوز شبکه از پیچدگی مناسبی برخوردار نمیباشد.

دوباره مدلی جدید را امتحان میکنیم:

در این شبکه از dropout استفاده میکنیم و همچنین لایه دیگری به مدل قبلی نیز اضافه میکنیم و بقیه موارد را بدون تغییر نگه میداریم.

نمودار و دقت به صورت زیر خواهد بود:

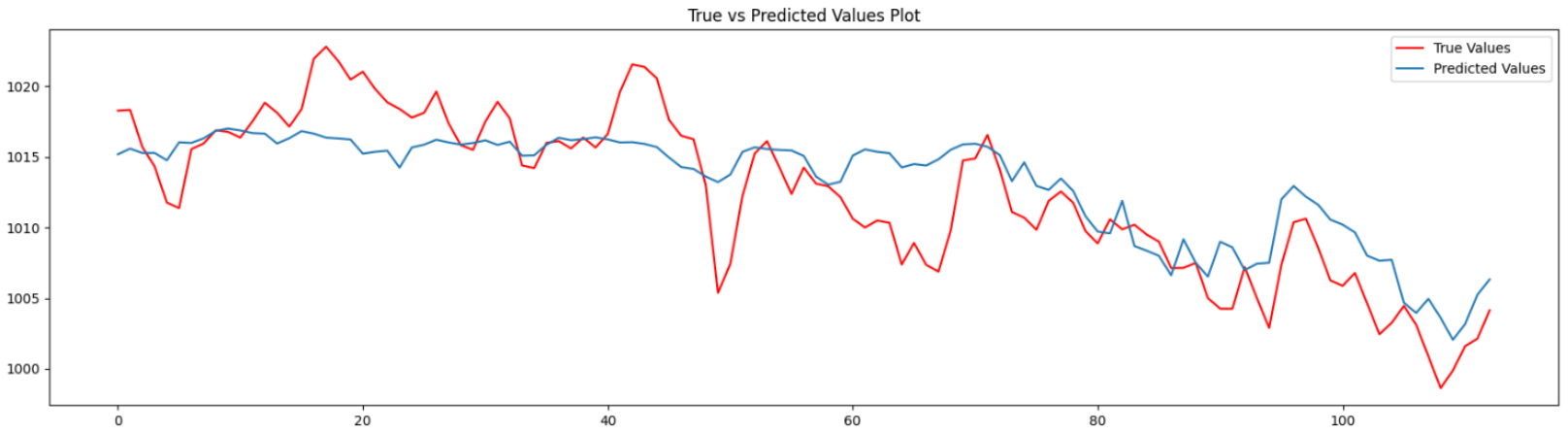



```
Mean Absolute Error: 3.394763274608716
Mean Squared Error: 16.408978386517372
Root Mean Squared Error: 4.050799721847202
R2 Error: 0.49222342097067473
```

همانطور که مشاهده میشود دقت کاهش یافته است و این میتواند به این دلیل باشد که مدل به سمت اورفیت شدن میرود.

اینبار تغییر دیگری در مدل میدهیم و از اکتیویشن فانکشن tanh استفاده میکنیم
همچنین تعداد ایپاک ها را به 150 افزایش میدهیم و بچ سایز را نیز به 20 میرسانیم

نتایج به صورت زیر خواهد بود:



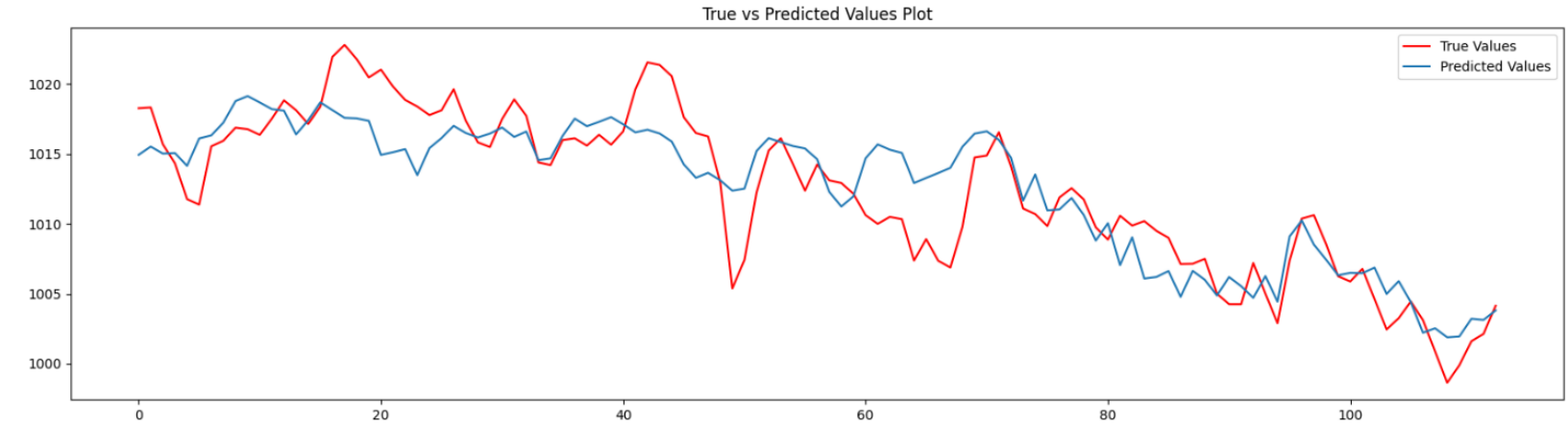

```
Mean Absolute Error: 2.550851825388708
Mean Squared Error: 10.568706280973647
Root Mean Squared Error: 3.2509546722422393
R2 Error: 0.672950905674419
```

همانطور که مشاهده میشود دقت افزایش یافته است و از نمودار میتوان دید که تقریبا به درستی در حال پیشبینی هستیم اما نمودار نسبیت به تغییرات مقاوم است و این میتواند نشانه از underfit بودن مدل باشد.

در نهایت از تمام تغییرات بالا نتیجه میگیریم که بهترین مدل مدل زیر خواهد بود:

```
regressor = Sequential()
regressor.add(LSTM(units=50, return_sequences=True, activation='tanh', input_shape=(None, 1)))
regressor.add(LSTM(units=50, return_sequences=True, activation='tanh'))
regressor.add(LSTM(units=50, activation='tanh'))
regressor.add(Dense(units=1))
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')
regressor.fit(x_train, y_train, batch_size = 20, epochs = 150)
```

که نمودار و دقت ان به شرح زیر میباشد:




```
Mean Absolute Error: 2.1910668153422246
Mean Squared Error: 7.838228594406248
Root Mean Squared Error: 2.79968365970269
R2 Error: 0.7574456612979812
```

در نمودار مدل در اکثر جاها به درستی پردیکت کرده است و به خوبی منطبق هستند همچنین در دقت به اعداد نسبتا بهتری از مقادیر اولیه رسیدیم و مقادیر کاملا قابل قبول هستند و این مدل را به عنوان مدل پیشنهادی ارائه میکنیم.

به عنوان مدلی دیگر که زیر مدلی از RNN است از GRU استفاده میکنیم.

در این مدل تمامی کد ها مشابه همان قبل است.

مدل زیر را ترین میکنیم:

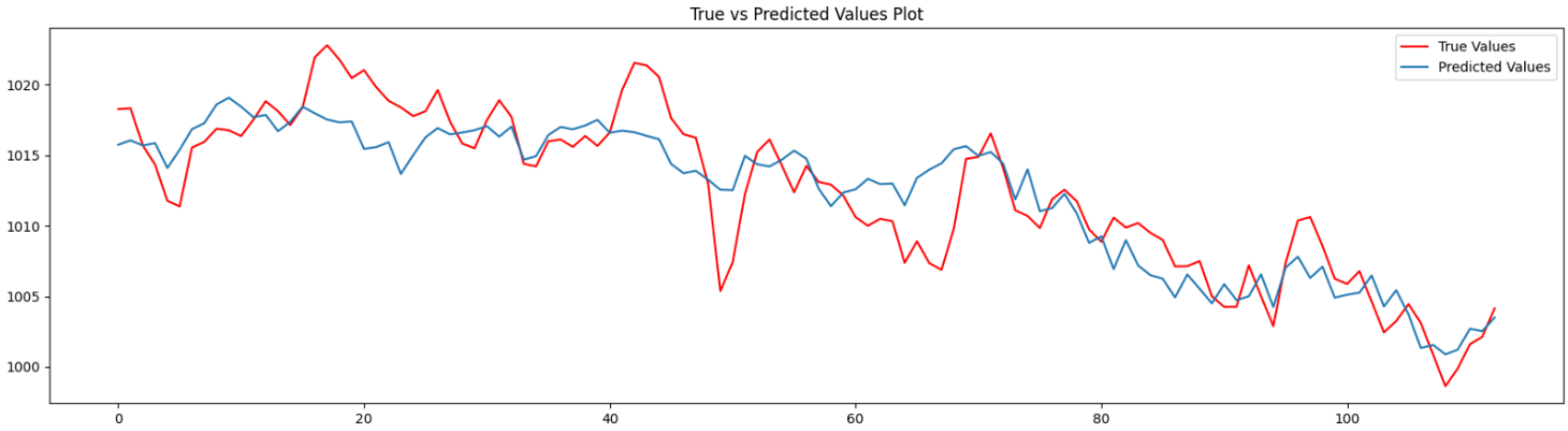
```
from tensorflow.keras.layers import GRU
from tensorflow.keras.optimizers import Adam

model = Sequential()
model.add(GRU(units=50, input_shape=(None, 1), return_sequences=True))
model.add(GRU(units=50, input_shape=(None, 1), return_sequences=True))
model.add(GRU(units=50, input_shape=(None, 1), return_sequences=False))
model.add(Dense(1))

model.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')
model.fit(x_train, y_train, epochs=150, batch_size=32, validation_split=0.2)

✓ 34.4s
```

این مدل بسیار سریع تر است. حال دقت و نمودار آن را بررسی میکنیم:




```
Mean Absolute Error: 2.0453102345561267
Mean Squared Error: 6.89169972682083
Root Mean Squared Error: 2.6252047018891367
R2 Error: 0.7867360399561636
```

همانطور که مشاهده میشود نمودار این مدل بسیار بهتر از مدل قبلی فیت شده است و همچنین خطا ها از مدل قبلی بسیار کمتر هستند.

در نهایت با توجه به بررسی مدل های مختلف LSTM و GRU به این نتیجه رسیدیم که GRU بهتر عمل میکند و به داده های تست بیشتر فیت میشود و همچنین دقت آن بیشتر و خطای آن در داده های تست کمتر میباشد.

به عنوان مدل پیشنهادی نهایی برای این دیتاسیت مدل GRU با پارامتر های زیر پیشنهاد میشود.

```
from tensorflow.keras.layers import GRU
from tensorflow.keras.optimizers import Adam

model = Sequential()
model.add(GRU(units=50, input_shape=(None, 1), return_sequences=True))
model.add(GRU(units=50, input_shape=(None, 1), return_sequences=True))
model.add(GRU(units=50, input_shape=(None, 1), return_sequences=False))
model.add(Dense(1))

model.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')
model.fit(x_train, y_train, epochs=150, batch_size=32, validation_split=0.2)

✓ 34.4s
```

همچنین سرعت این مدل بالاتر بوده و دقت نسبتا بهتری دارد.