به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

شبکه های عصبی و یادگیری عمیق

مینی پروژه سری اول

نام و نام خانوادگی	سینا شریفی	صدف صادقیان
شماره دانشجویی	810195412	810195419
تاریخ ارسال گزارش	1399/3/13	

فهرست گزارش سوالات

سوال ۱ – طراحی شبکه های عصبی

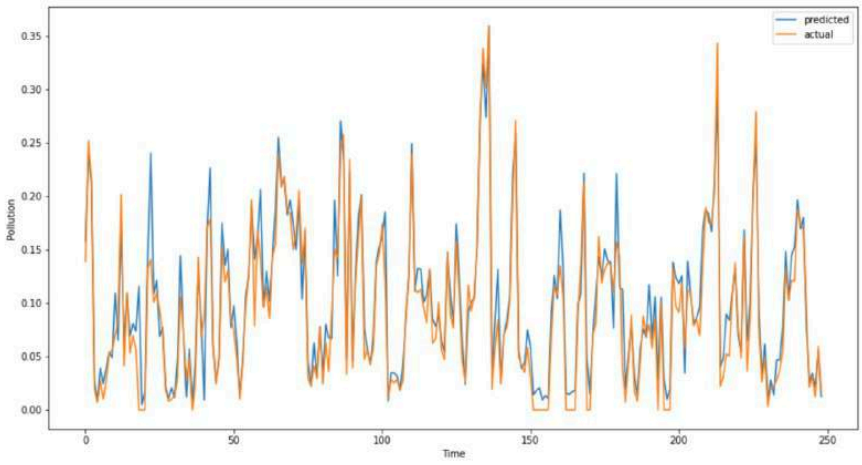
۱) برای هر کدام از شبکه هایی که طراحی می کنید نمودار train, test و همچنین نمودار مقدار حقیقی و پیش بینی را رسم کنید(۱۲۰۰۰رکورد اول را به عنوان داده trainو validationو ۳۰۰۰داده بعدی را به عنوان داده testاستفاده کنید)

ابتدا ۱۲۰۰۰ داده اولی را برای trainو ۳۰۰۰ داده بعدی که تا داده ۱۵۰۰۰ می‌شد را برای تست جدا کردیم. سپس دیتا ست‌مان را به نحوی ساختیم که آلودگی هوا هر ساعت خاص به عنوان label یا خروجی باشد و اطلاعات ۱۱ ساعت گذشته (هر ۸ ستون) آن به عنوان ورودی کنار هم قرار بگیرند. در نتیجه ورودی شبکه مان ۸*۱۱ شد. همچنین در هر ایپاک ۲۰ درصد داده‌ها را برای validation استفاده می کنیم.

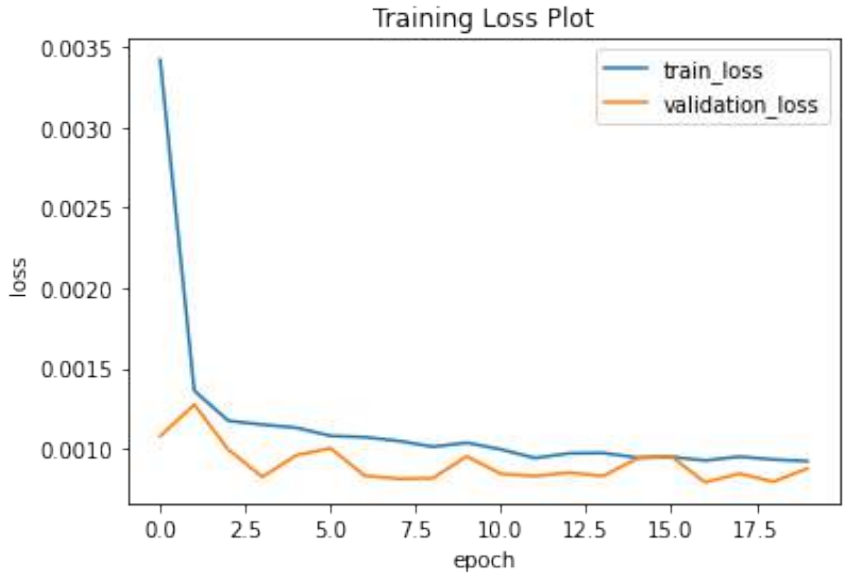
در ادامه با استفاده از keras یک شبکه دولایه‌ی ساده شامل یک لایه RNN و یک لایه‌ی dense طراحی می‌کنیم.

شبکه simpleRNN را برای ۱۰ ایپاک آموزش میدهیم.

از نظر زمانی هر ایپاک حدودا ۳ ثانیه طول میکشد و با معیار MSE، در نهایت به loss 0.00045 میرسیم و در ادامه نمودارهای حاصل را مشاهده می‌کنیم.



شکل ۱- نمودار مقدار حقیقی و پیش‌بینی برای شبکه RNN



شکل ۲- نمودار loss برای train و validation برای شبکه RNN

۲) شبکه را با RNN،LSTM،GRU طراحی کنید و سرعت و دقت هر کدام را مقایسه کنید) زمان آموزش برای یک تعداد epoch مشخص اندازه بگیرید) تفاوت ها را تحلیل کنید.

لایه RNN را به ترتیب Simple RNN، LSTM و GRU قرار دادیم و ۲۰ ایپاک هر کدام را train کردیم.

از نظر زمان آموزش برای ۲۰ ایپاک به نتیجه زیر می‌رسیم:

RNN = ۶۵ sec GRU = ۱۰۹ sec LSTM = ۱۲۰ sec

RNN < GRU < LSTM

سرعت زیاد RNN با توجه به معماری ساده‌ی آن قابل پیش‌بینی بوده و همانطور که انتظار میرفت،GRU از LSTM سریع تر است که دوباره علت آن تعداد گیت‌های کمتر آن است.

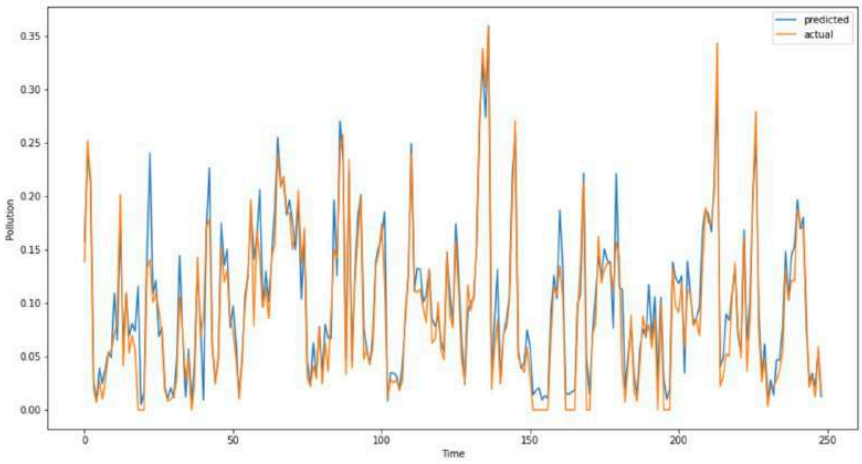
از نظر MSE LOSS هم به نتایج زیر می‌رسیم:

RNN = ۰.۰۰۰۴۵ GRU =۰.۰۰۰۴۲ LSTM = ۰.۰۰۰۴۱

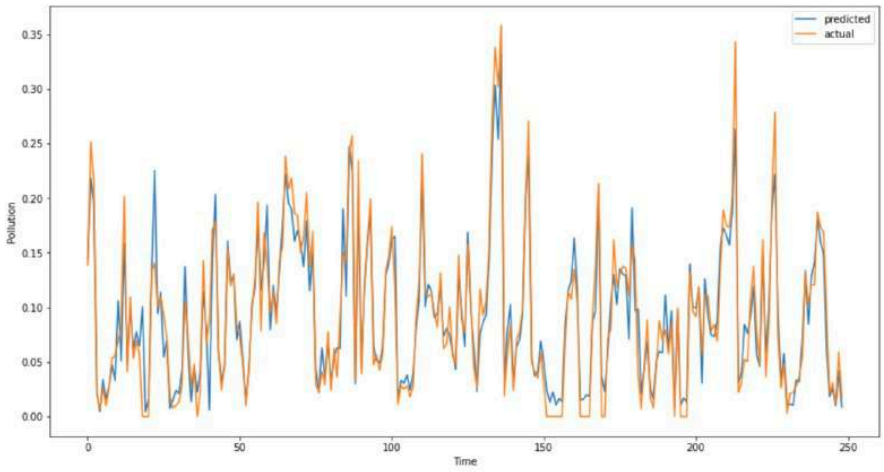
RNN > GRU > LSTM

با توجه به سادگی شبکه‌ی RNN، این شبکه دارای کمترین دقت است و دو شبکه دیگر چون اطلاعات را از گذشته دورتری به یاد دقت بیشتر آن‌ها نسبت به Simple RNN قابل پیش‌بینی بود. همچنین با توجه به طراحی پیچیده‌تر شبکه LSTM لاس این شبکه کم‌تر از دو شبکه دیگر است و دقت بهتری دارد.

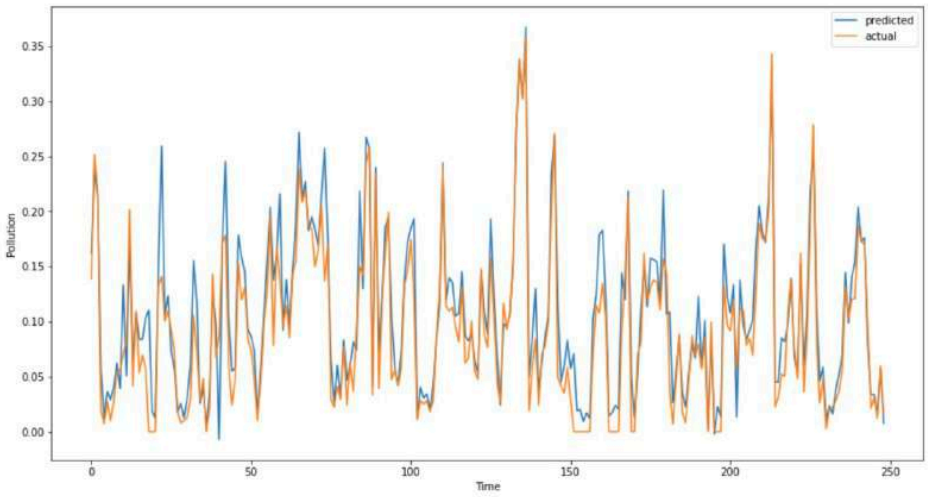
همچنین برای مقایسه‌ی بهتر، نمودارهای مقایسه‌ی مقادیر حقیقی و پیش‌بینی شده و نمودارهای train and validation loss در ادامه آمده است.



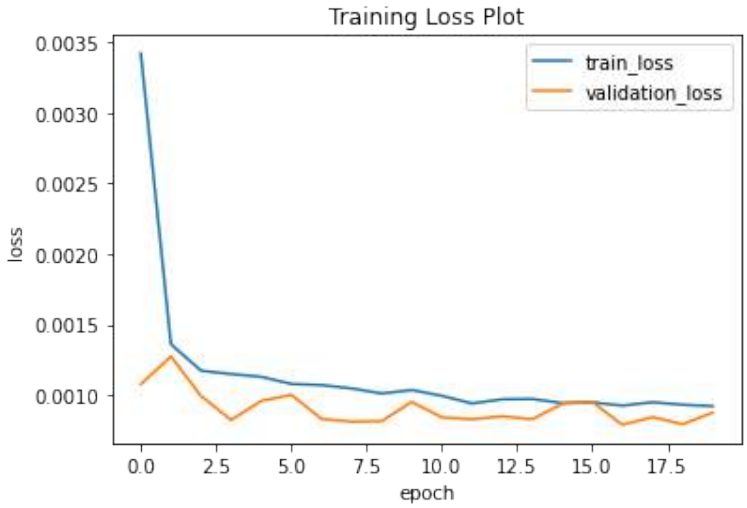
شکل ۳- نمودار مقدار حقیقی و پیش‌بینی برای شبکه RNN



شکل ۴- نمودار مقدار حقیقی و پیش‌بینی برای شبکه LSTM

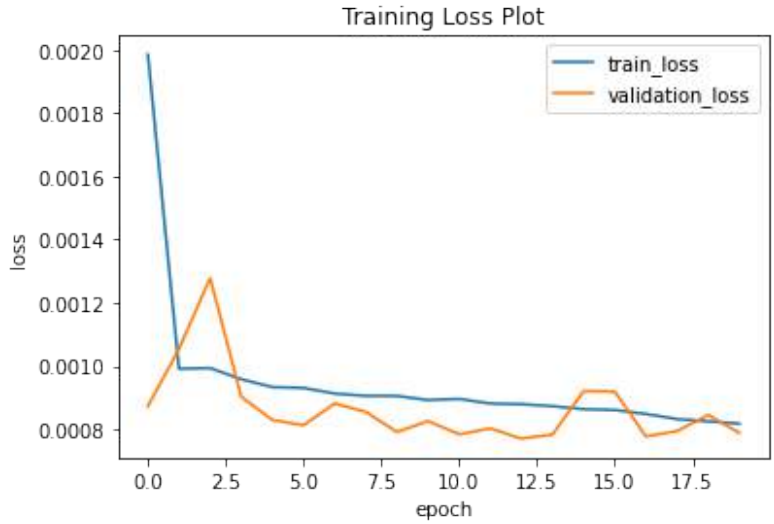


شکل ۵- نمودار مقدار حقیقی و پیش‌بینی برای شبکه GRU



شکل ۶- نمودار loss برای train و validation برای شبکه RNN

شکل ۷- نمودار loss برای train و validation برای شبکه LSTM



شکل ۸- نمودار loss برای train و validation برای شبکه GRU

۳) نحوه‌ی عملکرد شبکه برای تابع‌های هزینه متفاوت و روش‌های بهینه‌سازی متفاوت(MSE وMAE) و همچنین توابع خطای متفاوت (ADAgrad , RMSProp ,Adam) بررسی کنید .

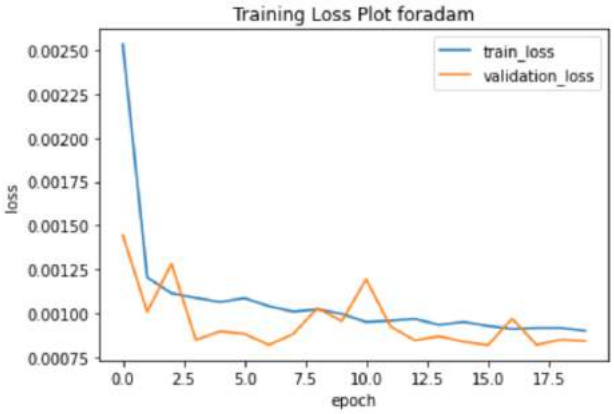
در ادامه نمودار مقدار حقیقی و پیش‌بینی برای شبکه ها برای ۲۰ ایپاک را به ازای روش های بهینه سازی مختلف و توابع خطای مختلف داریم:

روش بهینه ساز = MSE

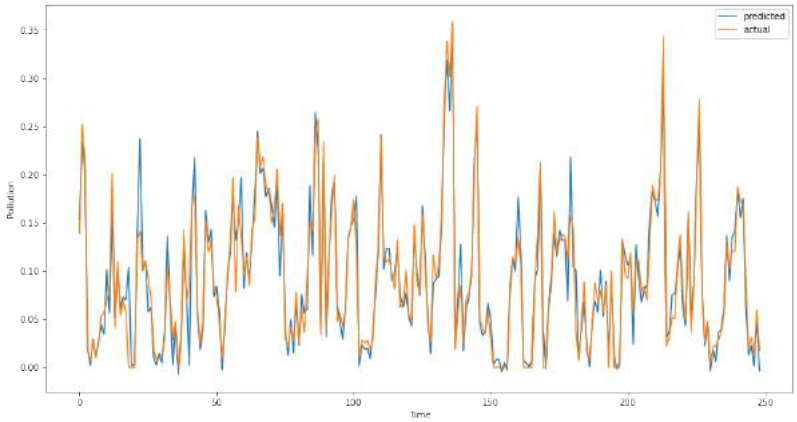
- ابتدا با simpleRNN را در نظر می‌گیریم:

۱) تابع خطا = Adam

MSE Loss = ۰.۰۰۰۴۵



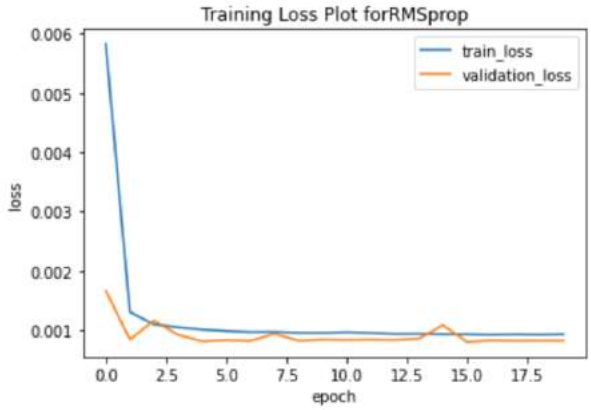
شکل ۹- نمودار مقدار loss برای شبکه RNN با تابع خطای MSE و بهینه‌ساز adam



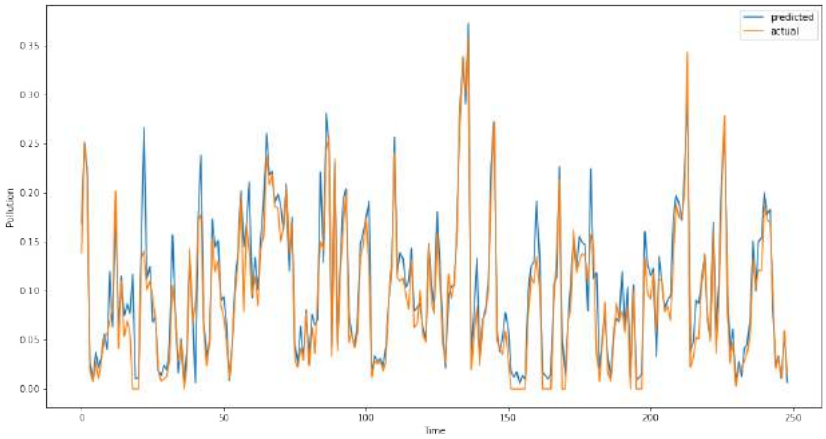
شکل ۱۰- نمودار مقدار حقیقی و پیش‌بینی برای شبکه RNN با تابع خطای MSE و بهینه‌ساز adam

۲) تابع خطا = RMSprop

MSE Loss = ۰.۰۰۰۴۳



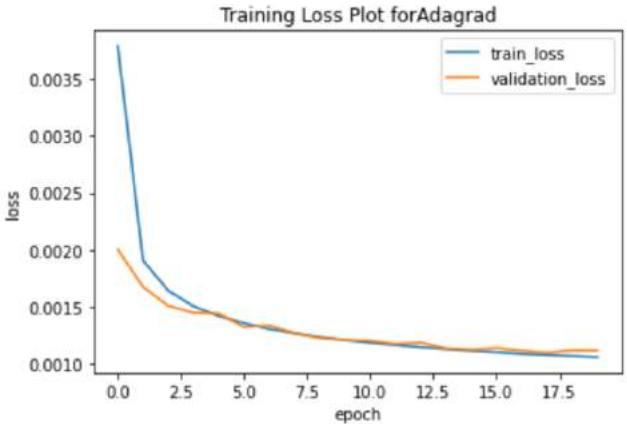
شکل ۱۱- نمودار مقدار loss برای شبکه RNN با تابع خطای MSE و بهینه‌ساز RMSprop



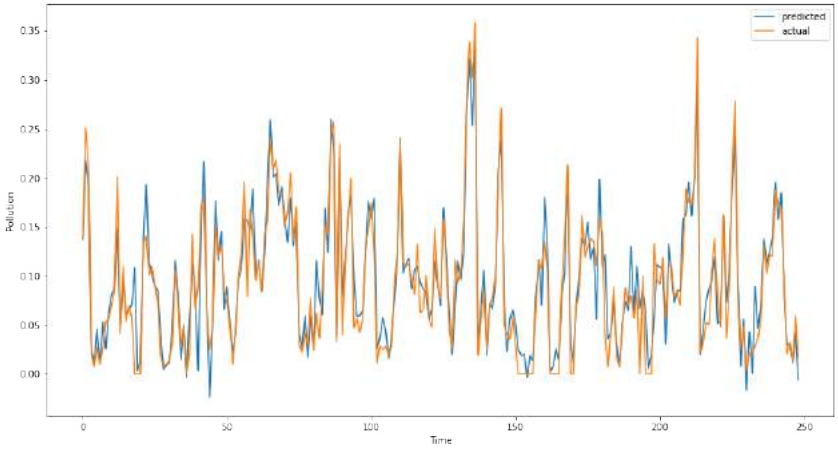
شکل ۱۲- نمودار مقدار حقیقی و پیش‌بینی برای شبکه RNN با تابع خطای MSE و بهینه‌ساز RMSprop

۳) تابع خطا = ADAgrad

MSE Loss = ۰.۰۰۰۶۲



شکل ۱۳- نمودار مقدار loss برای شبکه RNN با تابع خطای MSE و بهینه‌ساز Adagrad

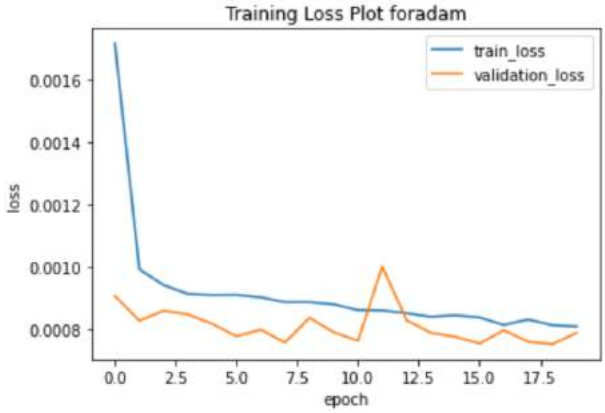


شکل ۱۴- نمودار مقدار حقیقی و پیش‌بینی برای شبکه RNN با تابع خطای MSE و بهینه‌ساز Adagrad

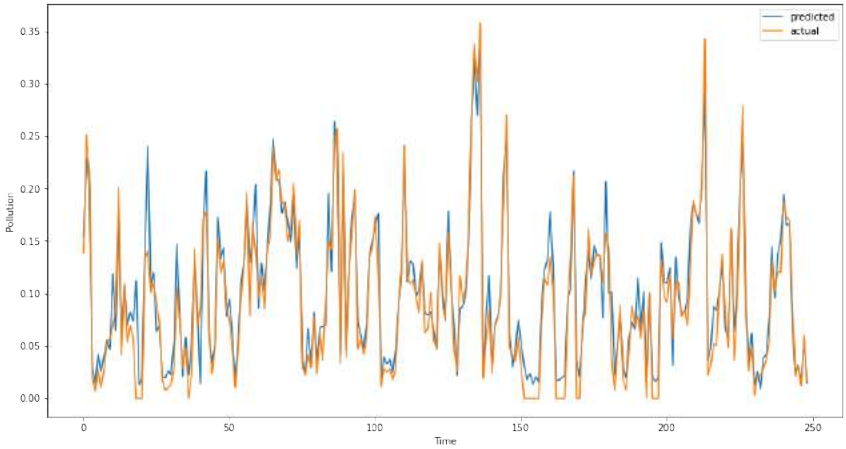
- حال شبکه‌ی LSTM را بررسی می‌کنیم:

۱) تابع خطا = Adam

MSE Loss = ۰.۰۰۰۴۷



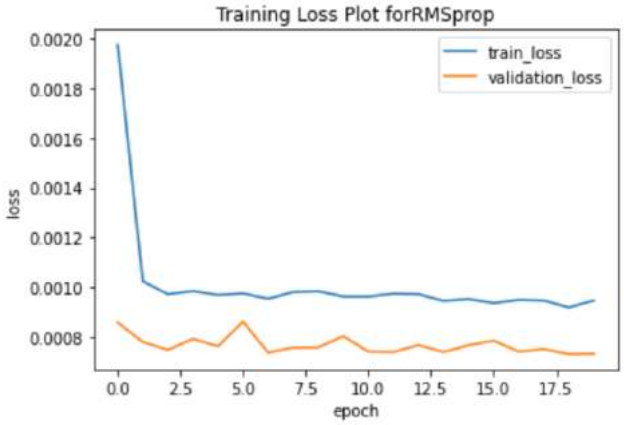
شکل ۱۵- نمودار مقدار loss برای شبکه LSTM با تابع خطای MSE و بهینه‌ساز adam



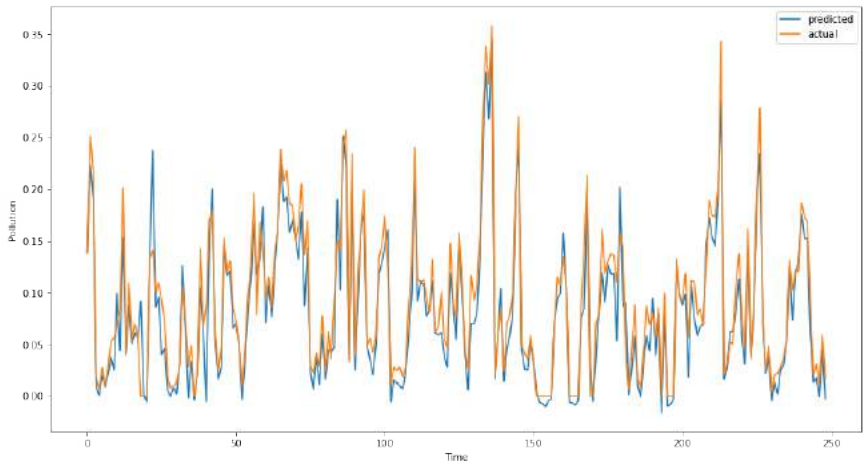
شکل ۱۶- نمودار مقدار حقیقی و پیش‌بینی برای شبکه LSTM با تابع خطای MSE و بهینه‌ساز adam

۲) تابع خطا = RMSprop

MSE Loss = ۰.۰۰۰۴۱



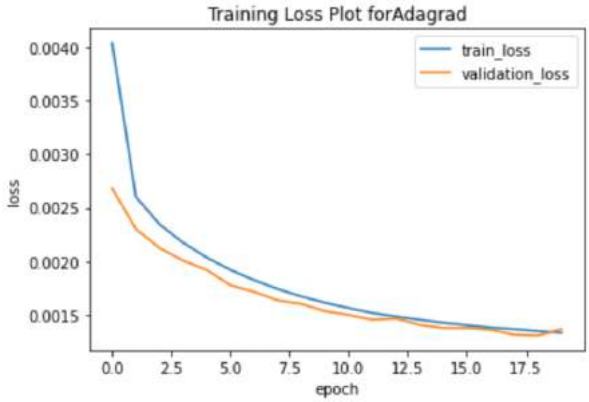
شکل ۱۷- نمودار مقدار loss برای شبکه LSTM با تابع خطای MSE و بهینه‌ساز RMSprop



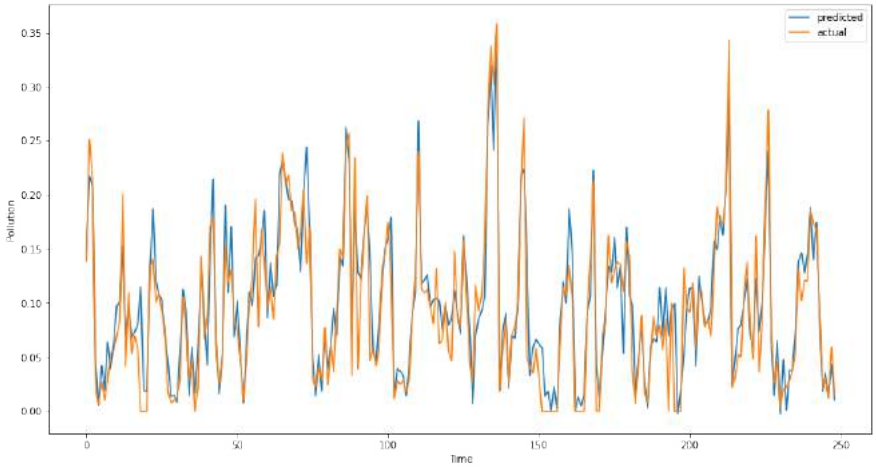
شکل ۱۸- نمودار مقدار حقیقی و پیش‌بینی برای شبکه LSTM با تابع خطای MSE و بهینه‌ساز RMSprop

۳) تابع خطا = ADAgrad

MSE Loss = ۰.۰۰۰۸۳



شکل ۱۹- نمودار مقدار loss برای شبکه LSTM با تابع خطای MSE و بهینه‌ساز Adagrad

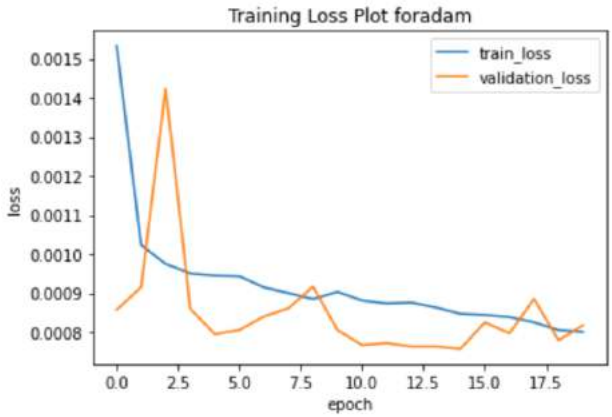


شکل ۲۰- نمودار مقدار حقیقی و پیش‌بینی برای شبکه LSTM با تابع خطای MSE و بهینه‌ساز Adagrad

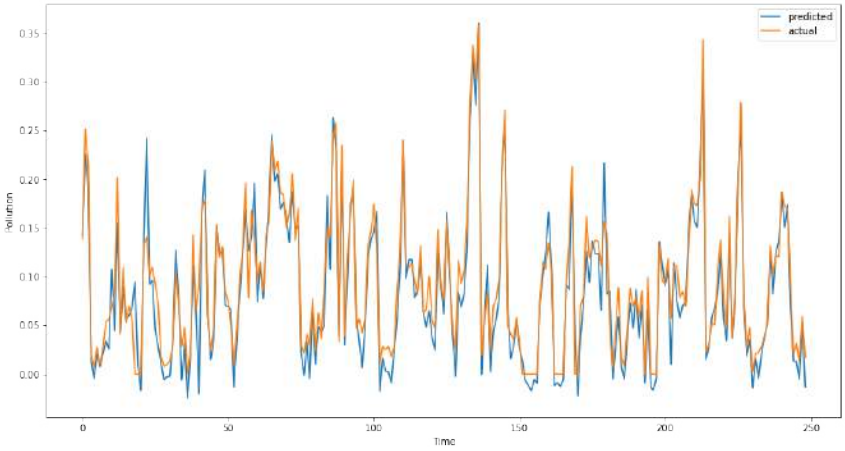
- در انتها شبکه‌ی GRU را بررسی می‌کنیم:

۱) تابع خطا = Adam

MSE Loss = ۰.۰۰۰۵۸



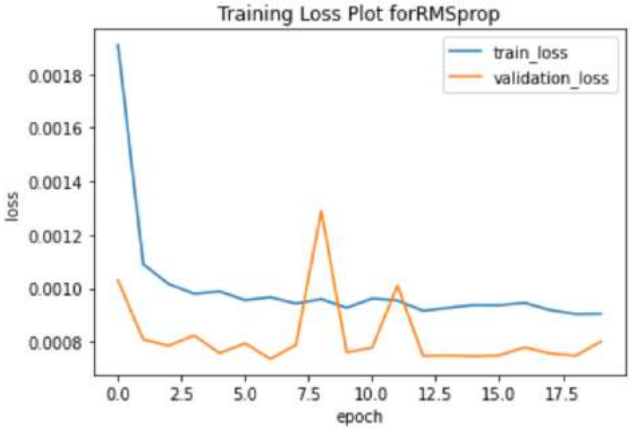
شکل ۲۱- نمودار مقدار loss برای شبکه GRU با تابع خطای MSE و بهینه‌ساز Adam



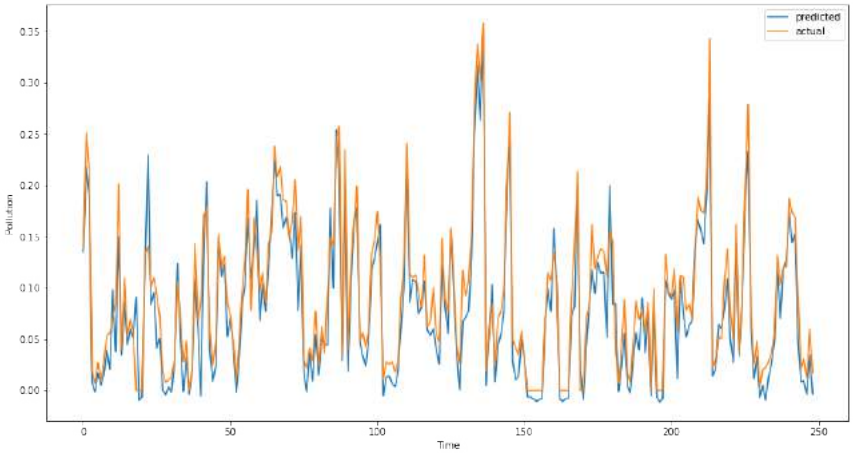
شکل ۲۲- نمودار مقدار حقیقی و پیش‌بینی برای شبکه GRU با تابع خطای MSE و بهینه‌ساز Adam

۲) تابع خطا = RMSprop

MSE Loss = ۰.۰۰۰۴۱



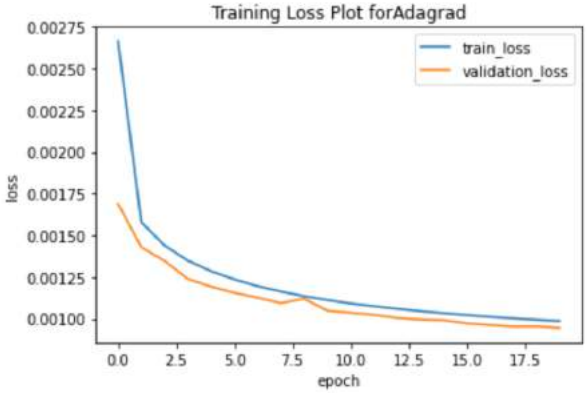
شکل ۲۳- نمودار مقدار loss برای شبکه GRU با تابع خطای MSE و بهینه‌ساز RMSprop



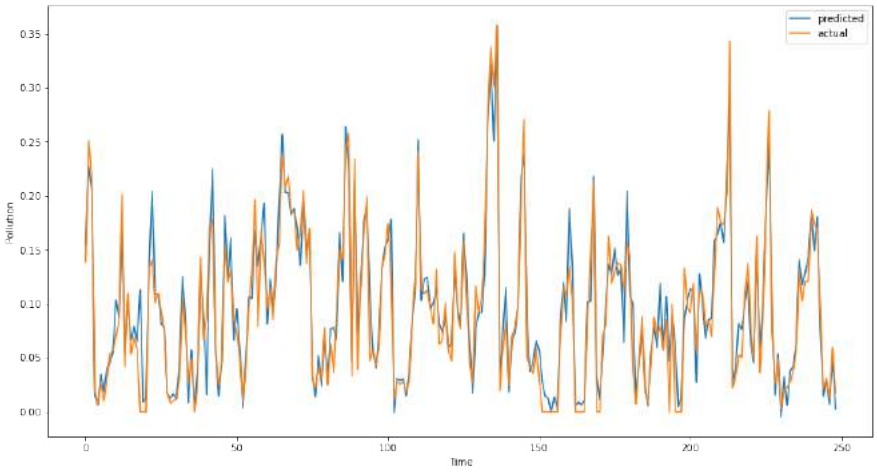
شکل ۲۴- نمودار مقدار حقیقی و پیش‌بینی برای شبکه GRU با تابع خطای MSE و بهینه‌ساز RMSprop

۳) تابع خطا = ADAgrad

MSE Loss = ۰.۰۰۰۴۷



شکل ۲۵- نمودار مقدار loss برای شبکه GRU با تابع خطای MSE و بهینه‌ساز Adagrad



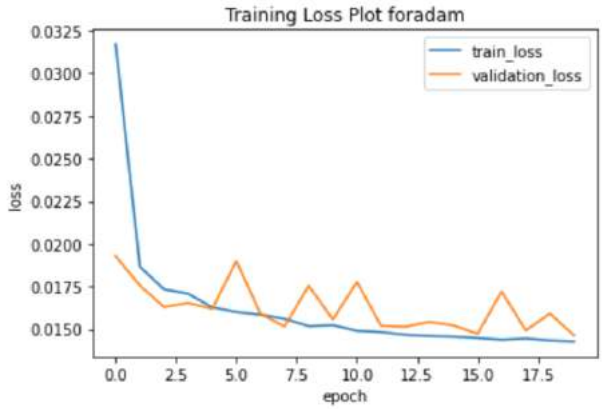
شکل ۲۶- نمودار مقدار حقیقی و پیش‌بینی برای شبکه GRU با تابع خطای MSE و بهینه‌ساز Adagrad

روش بهینه ساز = MAE

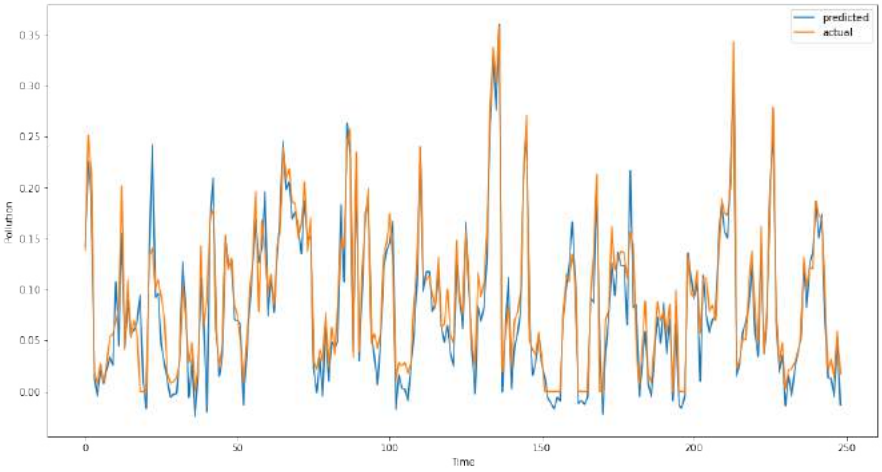
- از شبکه‌ی RNN شروع میکنیم.

۱) تابع خطا = Adam

MSE Loss = ۰.۰۰۰۴۵



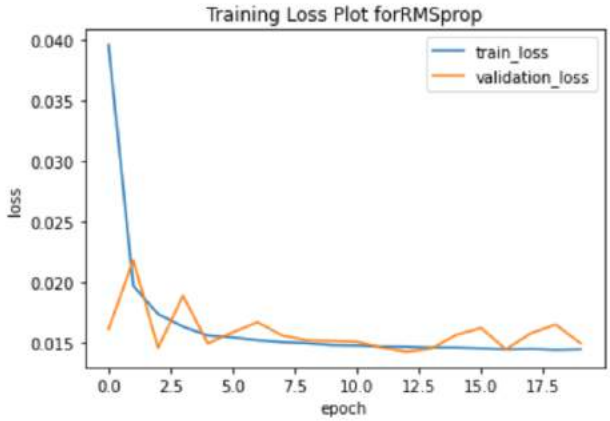
شکل ۲۷- نمودار مقدار loss برای شبکه RNN با تابع خطای MAE و بهینه‌ساز Adam



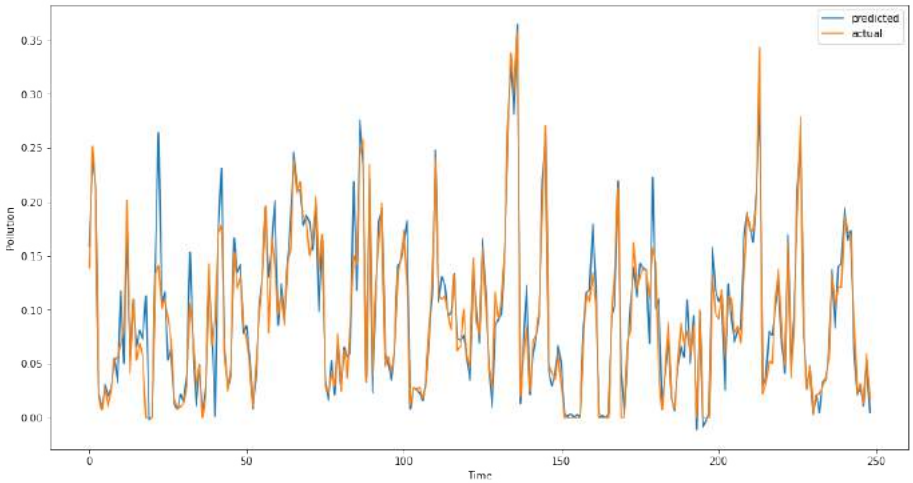
شکل ۲۸- نمودار مقدار حقیقی و پیش‌بینی برای شبکه RNN با تابع خطای MAE و بهینه‌ساز Adam

۲) تابع خطا = RMSprop

MSE Loss = ۰.۰۰۰۴۵



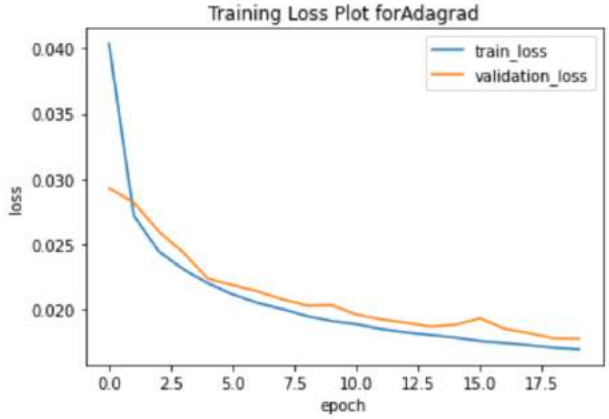
شکل ۲۹- نمودار مقدار loss برای شبکه RNN با تابع خطای MAE و بهینه‌ساز RMSprop



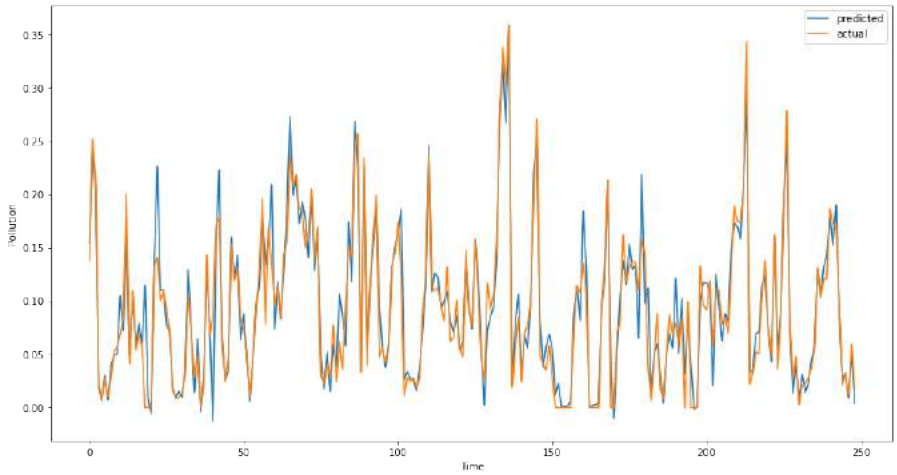
شکل ۳۰ - نمودار مقدار حقیقی و پیش‌بینی برای شبکه RNN با تابع خطای MAE و بهینه‌ساز RMSprop

۳) تابع خطا = ADAgrad

MSE Loss = ۰.۰۰۰۵۱



شکل ۳۱- نمودار مقدار loss برای شبکه RNN با تابع خطای MAE و بهینه‌ساز Adagrad

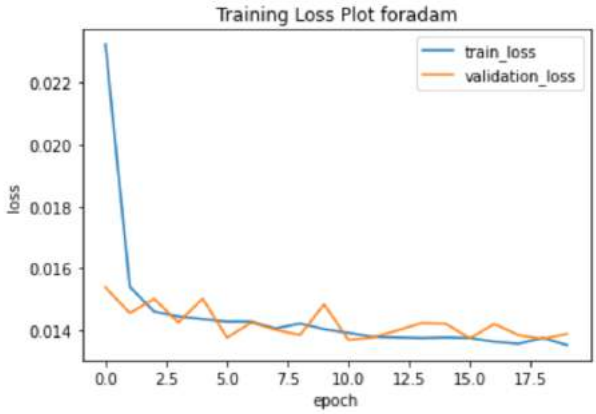


شکل ۳۲ - نمودار مقدار حقیقی و پیش‌بینی برای شبکه RNN با تابع خطای MAE و بهینه‌ساز Adagrad

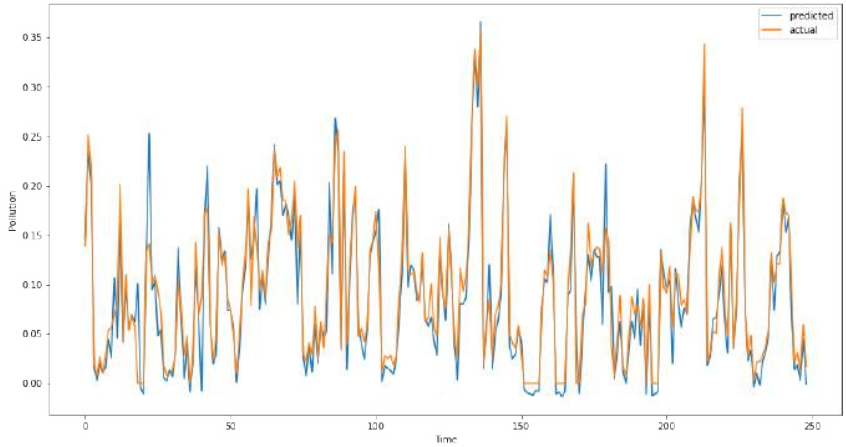
- سپس شبکه‌ی LSTM:

۱) تابع خطا = Adam

MSE Loss = ۰.۰۰۰۴۳



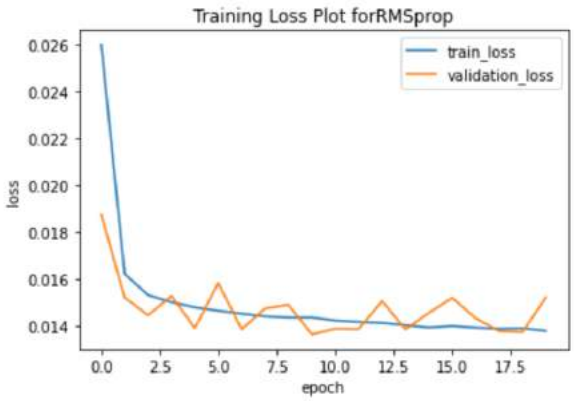
شکل ۳۳- نمودار مقدار loss برای شبکه LSTM با تابع خطای MAE و بهینه‌ساز Adam



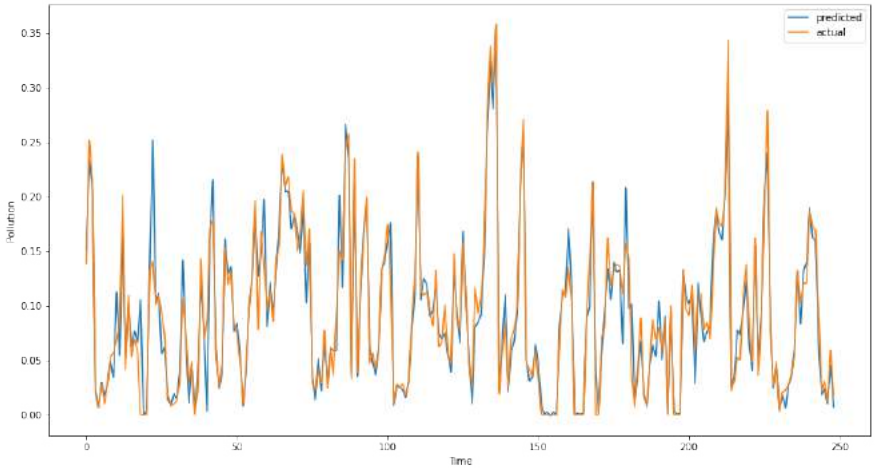
شکل ۳۴ - نمودار مقدار حقیقی و پیش‌بینی برای شبکه LSTM با تابع خطای MAE و بهینه‌ساز Adam

۲) تابع خطا = RMSprop

MSE Loss = ۰.۰۰۰۵۰



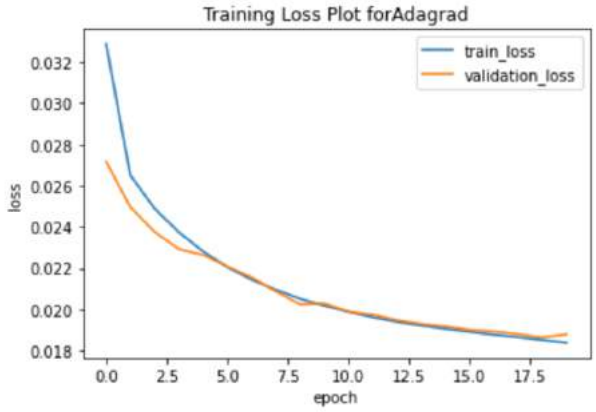
شکل ۳۵- نمودار مقدار loss برای شبکه LSTM با تابع خطای MAE و بهینه‌ساز RMSprop



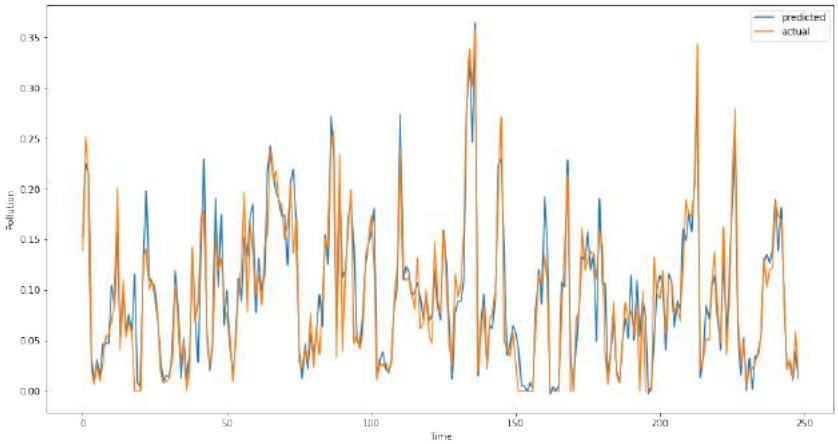
شکل ۳۶ - نمودار مقدار حقیقی و پیش‌بینی برای شبکه LSTM با تابع خطای MAE و بهینه‌ساز RMSprop

۳) تابع خطا = ADAgrad

MSE Loss = ۰.۰۰۰۵۹



شکل ۳۷- نمودار مقدار loss برای شبکه LSTM با تابع خطای MAE و بهینه‌ساز Adagrad

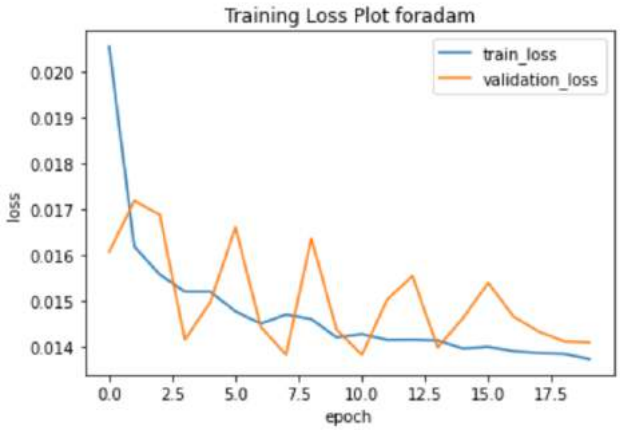


شکل ۳۸- نمودار مقدار حقیقی و پیش‌بینی برای شبکه LSTM با تابع خطای MAE و بهینه‌ساز Adagrad

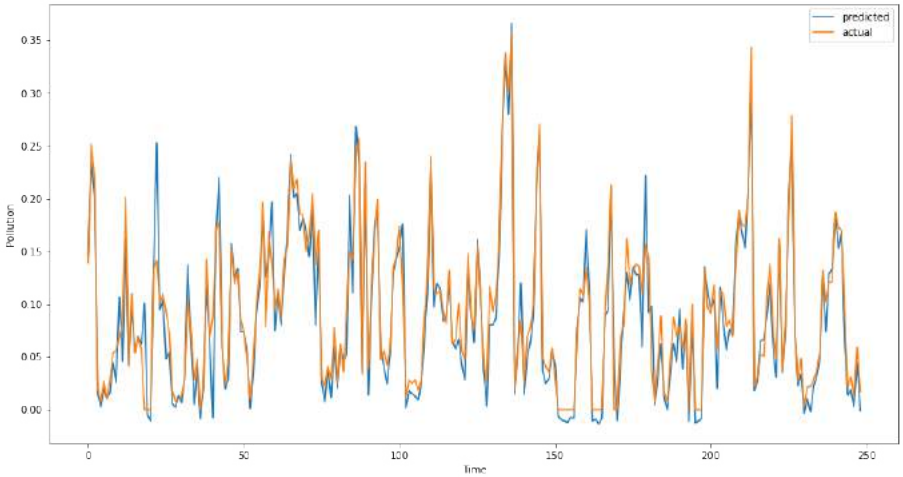
- شبکه‌ی GRU:

۱) تابع خطا = Adam

MSE Loss = ۰.۰۰۰۴۲



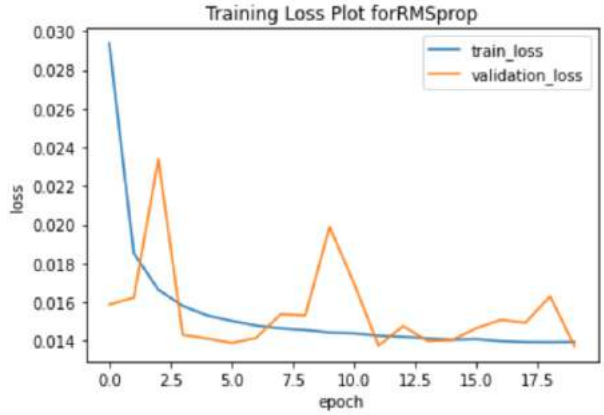
شکل ۳۹- نمودار مقدار loss برای شبکه GRU با تابع خطای MAE و بهینه‌ساز Adam



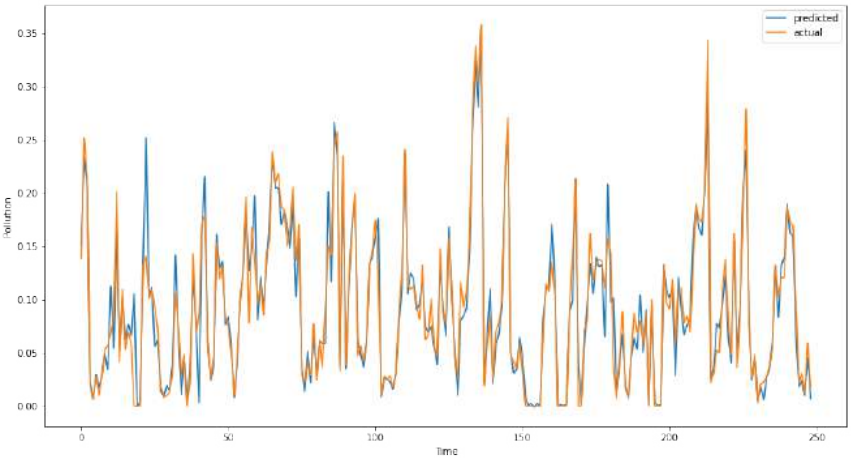
شکل ۴۰- نمودار مقدار حقیقی و پیش‌بینی برای شبکه GRU با تابع خطای MAE و بهینه‌ساز Adam

۲) تابع خطا = RMSprop

MSE Loss = ۰.۰۰۰۴۱



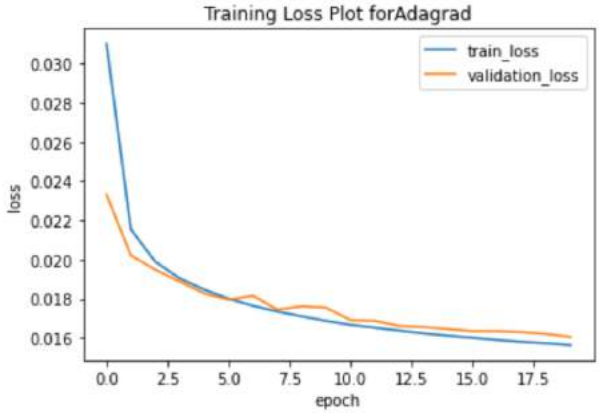
شکل ۴۱- نمودار مقدار loss برای شبکه GRU با تابع خطای MAE و بهینه‌ساز RMSprop



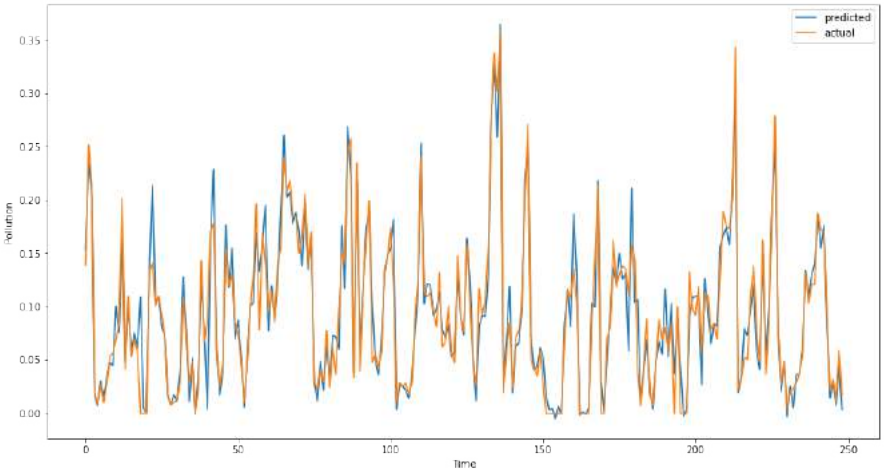
شکل ۴۲- نمودار مقدار حقیقی و پیش‌بینی برای شبکه GRU با تابع خطای MAE و بهینه‌ساز RMSprop

۳) تابع خطا = ADAgrad

MSE Loss = ۰.۰۰۰۴۴



شکل ۴۳- نمودار مقدار loss برای شبکه GRU با تابع خطای MAE و بهینه‌ساز Adagrad



شکل ۴۴- نمودار مقدار حقیقی و پیش‌بینی برای شبکه GRU با تابع خطای MAE و بهینه‌ساز Adagrad

از نمودارها و دقت‌های گزارش شده در این قسمت میتوانیم نتیجه بگیریم که با این تعداد داده عملکرد اکثر این شبکه‌ها خوب است ولی شبکه LSTM با تابع خطای MSE و بهینه‌ساز RMSprop بهترین عملکرد را بین این ۱۸ شبکه داشته است.

۴) عملکرد شبکه را برای سری زمانی های هفتگی(با استفاده از تابع رندم، یک ساعت رندم را انتخاب کنید و از داده ۶ روز پیاپی برای پیش بینی آلودگی در همان ساعت از روز ۷ ام استفاده کنید) و ماهانه (با استفاده از تابع رندم، یک روز رندم و یک ساعت رندم انتخاب کنید و با داده ۳ هفته پیاپی همان روز و همان ساعت، آلودگی را در همان روز و همان ساعت برای هفته ۴ ام پیش بینی کنید) نیز بررسی کنید .

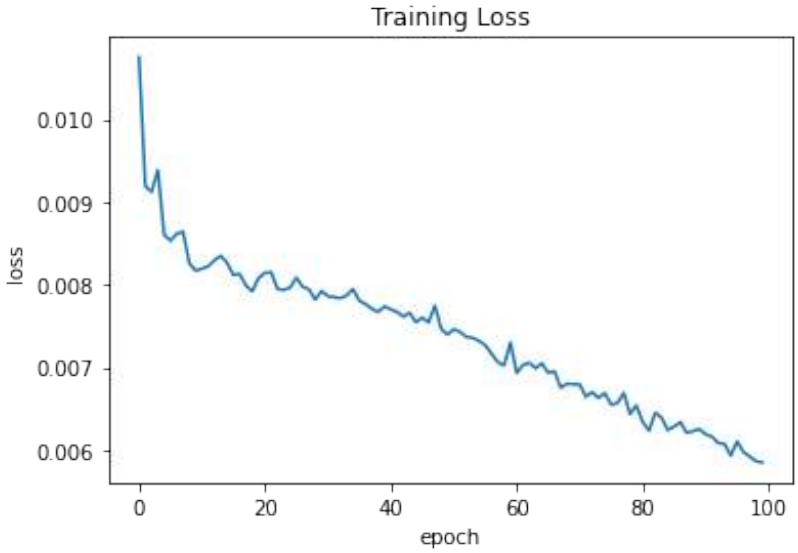
در هر دو بخش این سوال به علت کمبودن داده‌ها از 18000 داده اول برای آموزش و 4800 داده بعدی برای تست استفاده شد.

همچنین در این بخش از شبکه LSTM با تابع خطای mse و روش بهینه سازی adam استفاده شده است.

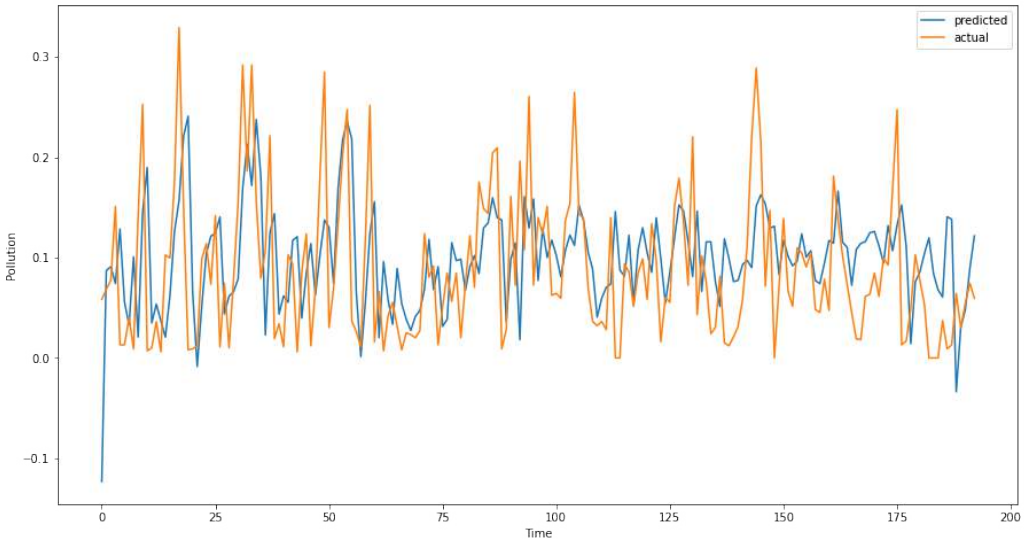
سری زمانی هفتگی

ابتدا برای حالت هفتگی، دیتاست را تشکیل میدهیم و شبکه را به اندازه ۱۰۰ ایپاک ترین میکنیم. روش کار به این صورت است که داده‌های ۶ روز و ساعت خاص را به شبکه میدهیم تا آلودگی روز هفتم همان ساعت را پیش‌بینی کند.

MSE = ۰.۰۰۵۲



شکل ۴۵- نمودار مقدار loss برای شبکه LSTM با برای حالت داده‌ی ۶ روز متوالی



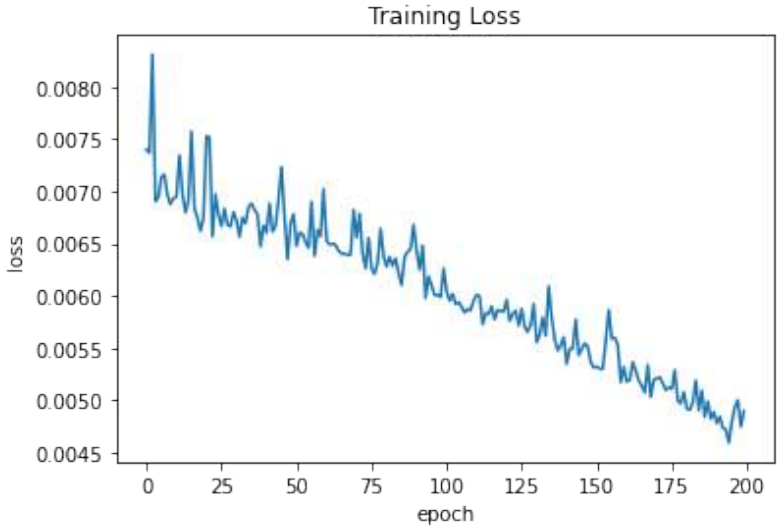
شکل ۴۶- نمودار مقدار حقیقی و پیش‌بینی برای شبکه LSTM با برای حالت داده‌ی ۶ روز متوالی

همانطورکه مشخص است، با توجه به دو نمودار فوق، شبکه به خوبی train شده و نتایج آن قابل قبول است.

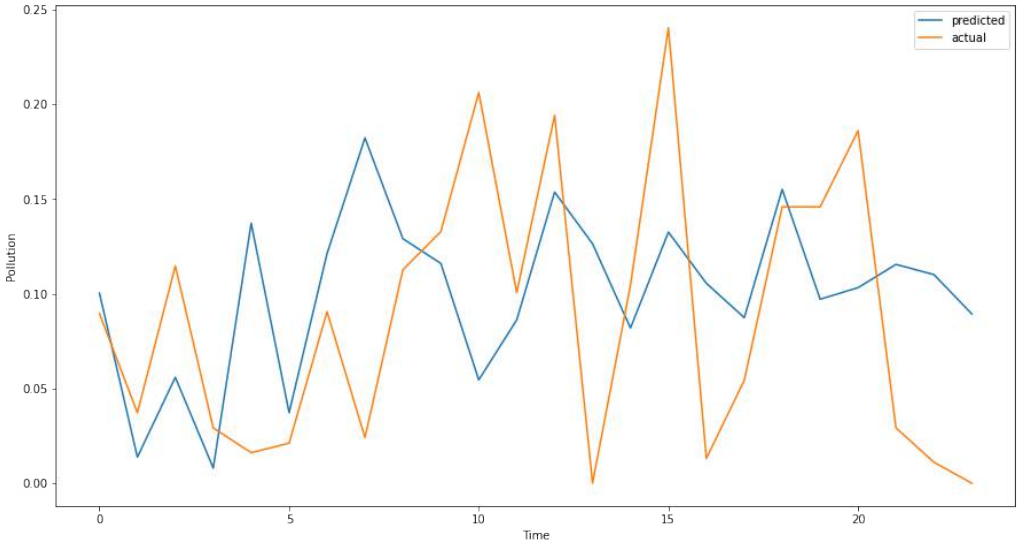
سری زمانی ماهانه

سپس برای حالت ماهانه، دیتاست را تشکیل میدهیم و شبکه را به مدت ۲۰۰ ایپاک ترین میکنیم. روش کار به این صورت است که داده‌های یک روز و ساعت خاص از آن روز را برای ۳ هفته به شبکه میدهیم تا آلودگی آن روز و ساعت خاص هفته‌ی چهارم را پیش‌بینی کند.

MSE = ۰.۰۰۵۹



شکل ۴۷- نمودار مقدار loss برای شبکه LSTM با برای حالت داده‌ی ۳ هفته متوالی

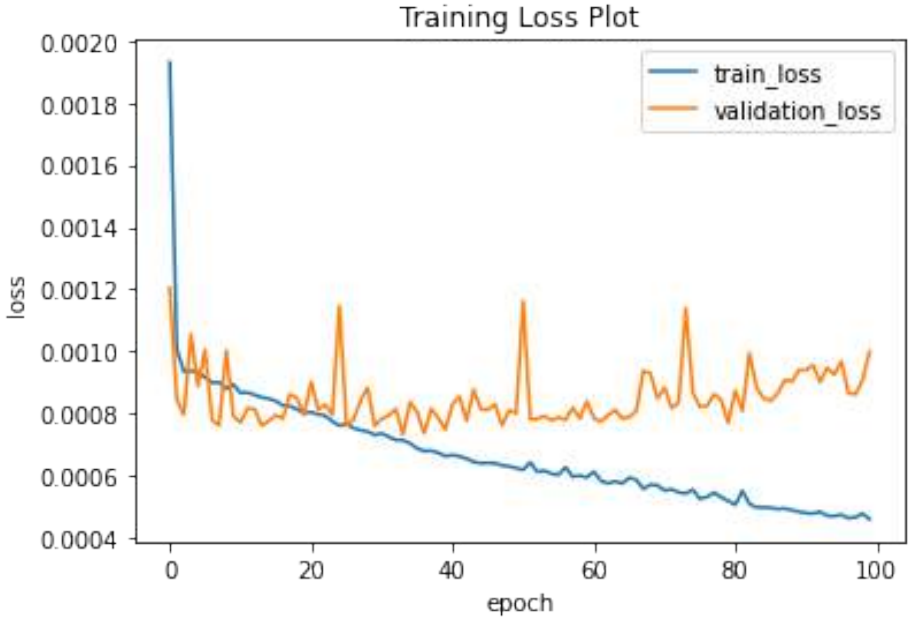


شکل ۴۸- نمودار مقدار حقیقی و پیش‌بینی برای شبکه LSTM با برای حالت داده‌ی ۳ هفته متوالی

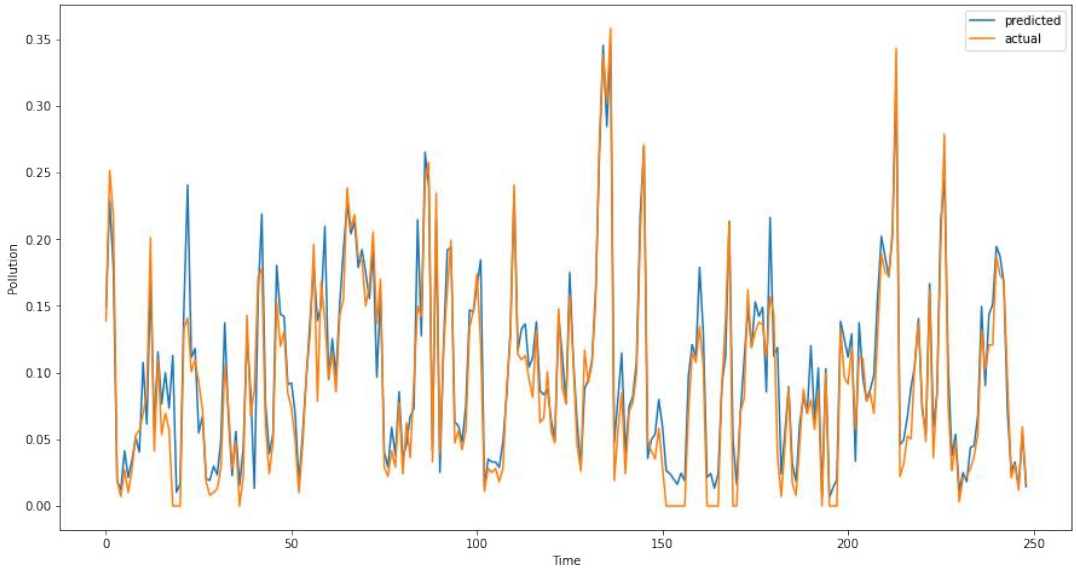
همانطورکه مشخص است، با توجه به کم بودن داده‌های دیتاست طبق نمودار فوق نتایج آن قابل قبول است و مشخصا نمودار loss روند نزولی دارد.

۵) تاثیر لایه dropout را بروی یک شبکه ی طراحی شده (به دلخواه) بررسی کنید .

شبکه LSTM با تابع خطای mse و روش بهینه سازی adam را که آلودگی روزانه(ساعت های هر روز) را پیش بینی می‌کرد را انتخاب میکنیم. ابتدا یک بار بدونdropout آن را برای ۱۰۰ ایپاک train میکنیم که نتایج در زیر آمده است.

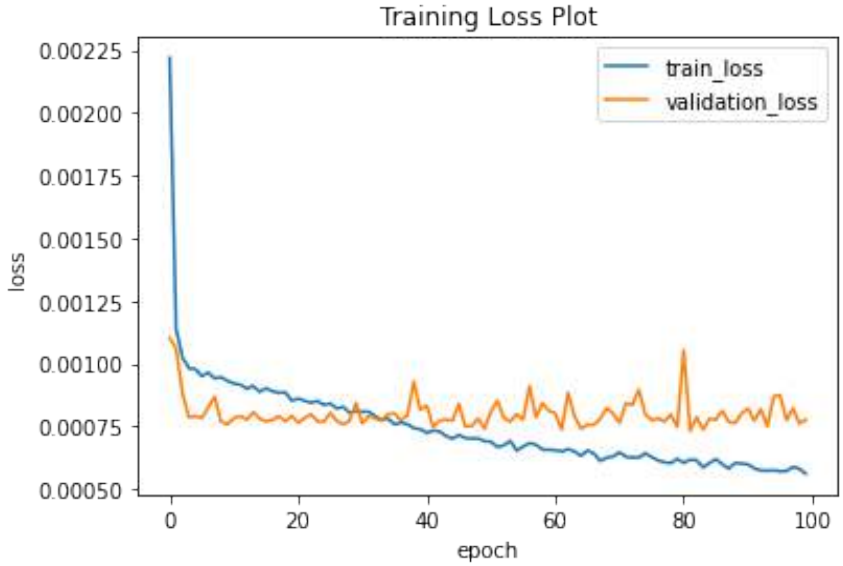


شکل ۴۹- نمودار مقدار loss برای شبکه LSTM با برای حالت داده‌ی ۱۱ ساعت متوالی بدون dropout

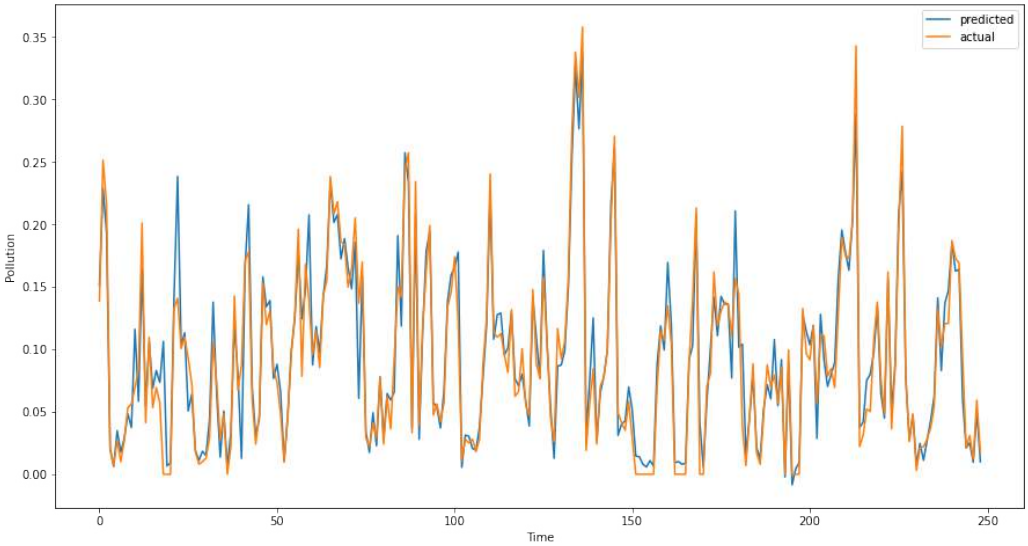


شکل ۵۰- نمودار مقدار حقیقی و پیش‌بینی برای شبکه LSTM با برای حالت داده‌ی ۱۱ ساعت متوالی بدون dropout

سپس با اضافه کردن یک لایه‌ی dropout با اندازه‌ی ۰.۱ بعد از سلول‌های LSTM دوباره نتایج را گزارش میکنیم.



شکل ۵۱- نمودار مقدار loss برای شبکه LSTM با برای حالت داده‌ی ۱۱ ساعت متوالی با dropout



شکل ۵۲- نمودار مقدار حقیقی و پیش‌بینی برای شبکه LSTM با برای حالت داده‌ی ۱۱ ساعت متوالی با dropout

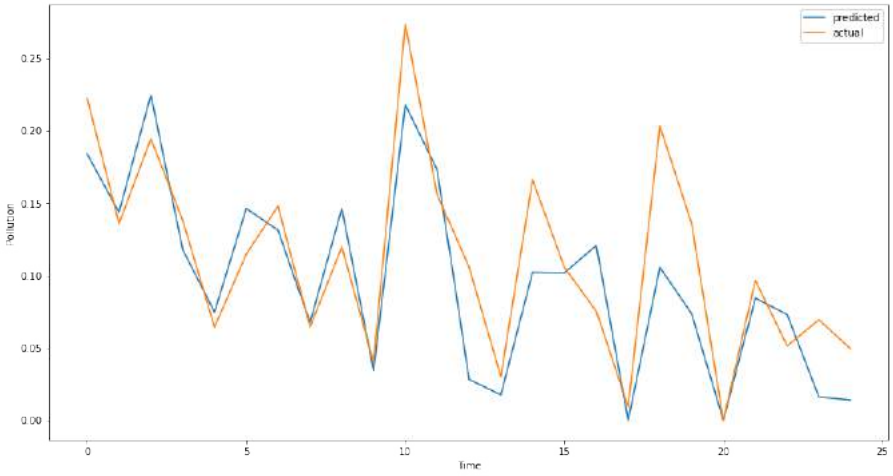
از ۴ نمودار فوق میتوانیم نتیجه بگیریم، که وجود Dropout نه تنها باعث نزدیک شدن نمودار ترین و ولیدیشن شده(که نشان دهنده‌ی جلوگیری از overfit شدن است) باعث کاهش خطای MSE هم شده است. در مجموع Dropout مشخصا باعث بهبود عملکرد شبکه شده است.

۶) بهترین شبکه بازگشتی در مراحل قبل را انتخاب کنید و دو شبکه‌ی بازگشتی دیگر با همان ساختار نیز به موازات آن بسازید .سپس سه نوع سری زمانی توضیح داده شده را برای پیش بینی مقدار آلودگی در یک ساعت مشخص به هر کدام از آن ها اعمال کنید .سپس به کمک یک لایه ی fusionخروجی سه شبکه ی recurrent را با هم ترکیب کنید. نتیجه را بررسی کنید .لزومی ندارد که سایز سری زمانی های سه شبکه ی موازی یکسان باشد .

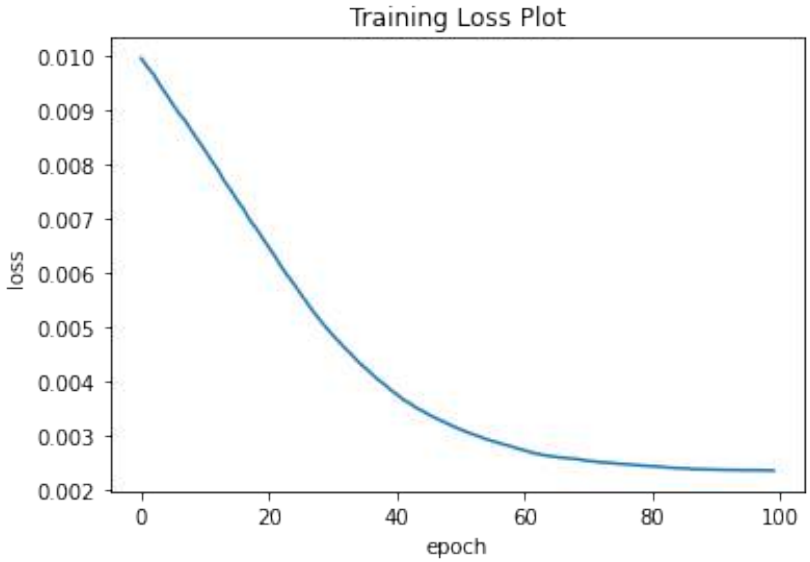
پس از تشکیل دیتاست‌های جدید از بهترین شبکه‌ی قسمت قبل (LSTM) استفاده میکنیم.

ابتدا ۳ شبکه‌ی قبل را اجرا میکنیم و سپس توسط یک لایه‌ی Dense با activation function تابع relu این ۳ شبکه را به هم وصل کرده و یک شبکه‌ی جدید میسازیم. (حالت میانگین‌گیری پاسخ ایده‌آلی نداد در نتیجه از یک لایه Dense استفاده کردیم.)

که حاصل این شبکه که از روی ترکیب سه شبکه پیش بینی می‌کند در ادامه آمده است:



شکل ۵۳- نمودار مقدار حقیقی و پیش‌بینی برای شبکه‌ی حاصل از ۳ شبکه‌ی موازی



شکل ۵۴- نمودار مقدار loss برای شبکه‌ی حاصل از ۳ شبکه‌ی موازی

مقادیر loss روی دیتای تست در ادامه گزارش شده:

MSE loss ۱۱ hours model = ۰.۰۰۱

MSE loss ۶ days model = ۰.۰۰۷

MSE loss ۳ weeks model = ۰.۰۰۵

MSE loss parallel model = ۰.۰۰۱

وزن‌های شبکه‌ی موازی به صورت زیر درآمد:

[۰.۵, ۰.۳, ۰,۴]

همانطور که مشخص است، شبکه از داده‌های هر سه شبکه قبلی استفاده کرده و لاس خود را کاهش میدهد و به نتایج قابل قبولی میرسد.

۷) اکنون فرض کنید برای پیش بینی آلودگی، فقط میتوانید از دو ستون دیگر (به جز آلودگی) کمک بگیرید (یعنی در مجموع ۳ ستون از ۸ ستون داده). برای اینکه میزان دقت پیش بینی شما بالاتر رود باید سعی کنید ۲ ستونی را انتخاب کنید که بیش ترین تاثیر را در پیش بینی درست آلودگی داشته باشد. روش شما برای انجام اینکار چیست؟

برای پیدا کردن دو ویژگی که باعث عملکرد بهتر شبکه شوند، از correlation ۷ بردار ویژگی با pollution استفاده میکنیم. بدین صورت که قدر مطلق تک تک عناصر بردار correlation را محاسبه کرده و دو ویژگی با اندازه correlation بزرگتر را انتخاب میکنیم.

در اصل correlation با استفاده از رابطه‌ی زیر، شباهت تغییرات دو سیگنال یا نمونه‌های متغیر تصادفی را بررسی میکند.

$$r_{xy} = \frac{\sum_{i=0}^{N}(x_i - x_{avg})(y_i - y_{avg})}{\sqrt{\sum_{i=0}^{N}(x_i - x_{avg})^2 \sum_{i=0}^{N}(y_i - y_{avg})^2}}$$

پس در صورتی که قدر مطلق r_{xy} برای دو ویژگی زیاد باشند یعنی به هم مربوط هستند. پس correlation ویژگی ها با pollution را حساب میکنیم.

نتایج به صورت زیر بدست آمد در نتیجه dew و wind speed به عنوان ویژگی‌های مناسب انتخاب شدند.

```
pollution           1.0
dew                 0.2698188525163449
temp                0.03491193627587944
pressure            -0.2434192350246169
wind_dir            0.20066129707031094
wind_spd            -0.24598641458243298
snow                -0.015577513565734448
rain                -0.0340748817112134
```

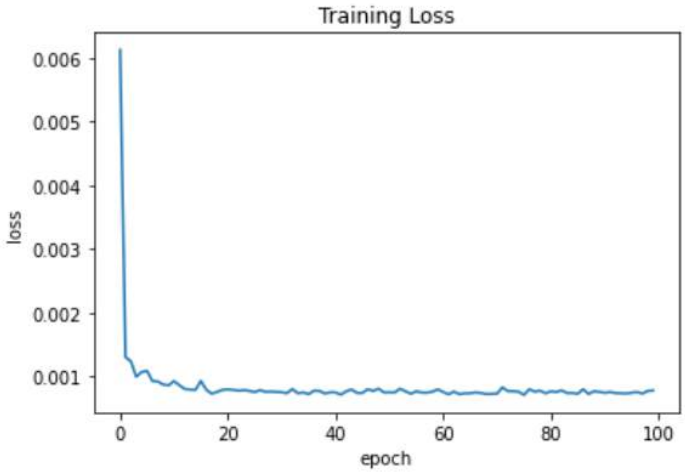
شکل ۵۵- مقادیر correlation هر ویژگی با pollution

۸) روش خود را برای قسمت قبل پیاده سازی کرده و با استفاده از این ۳ ستون(آلودگی و ۲ستونی که یافته اید) میزان آلودگی روزانه را برای هر سه سلول LSTMوRNNو GRUبررسی کنید.

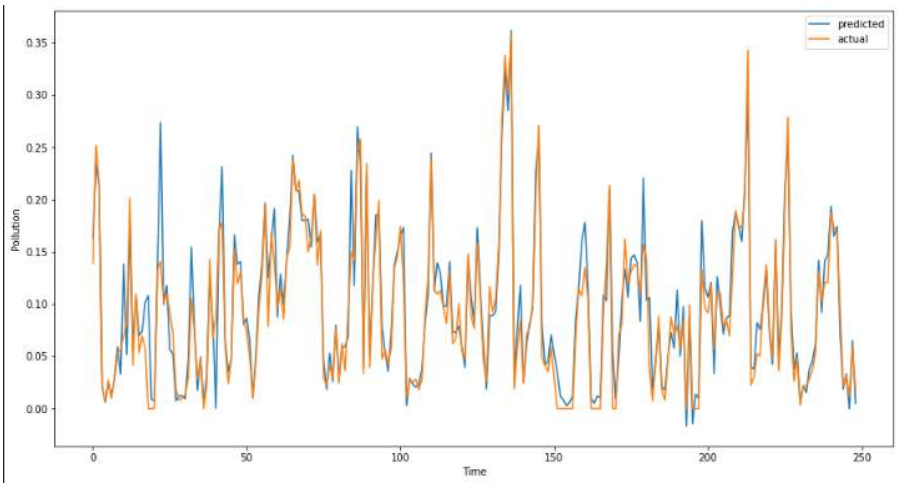
پس از پیدا کردن دو بردار ویژگی مورد نظر سه شبکه را مجددا برای ۱۰۰ ایپاک ترین میکنیم.
با توجه به نتایج زیر و نتیجه‌ی حاصل از قسمت ۴، مشخص است که دقت شبکه افزایش یافته است.

شبکه Simple RNN

MSE = ۰.۰۰۵۲



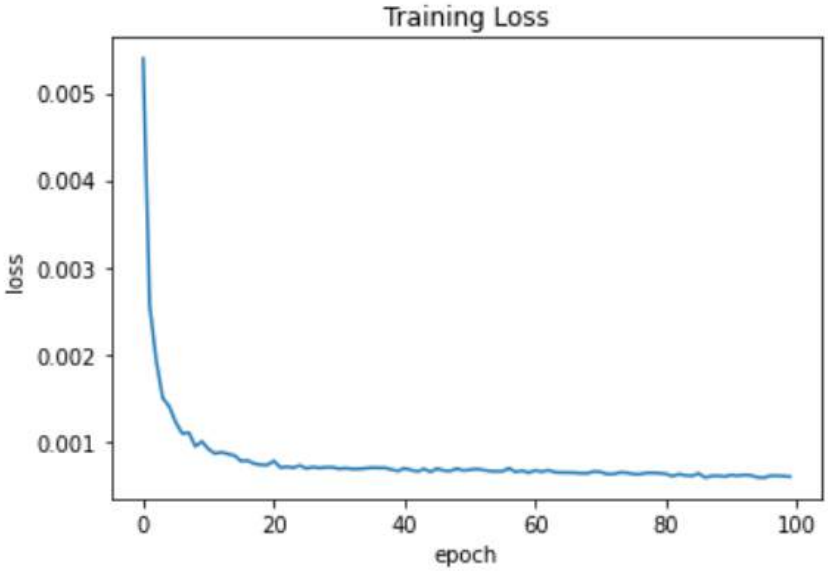
شکل ۵۶- نمودار مقدار loss برای شبکه‌ی RNN با ۲ ویژگی



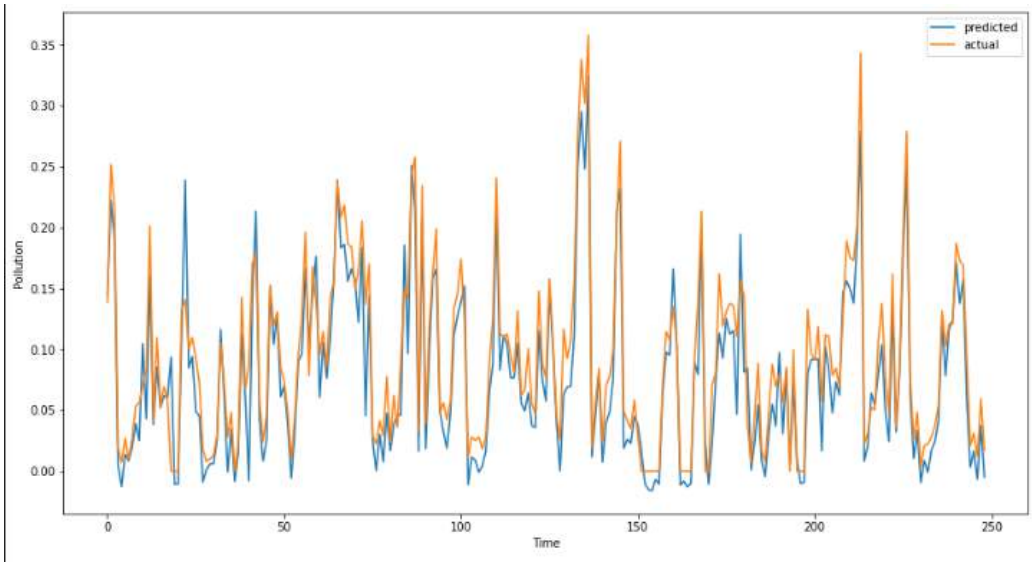
شکل ۵۷- نمودار مقدار حقیقی و پیش‌بینی برای شبکه‌ی RNN با ۲ ویژگی

شبکه LSTM

MSE = ۰.۰۰۷۰

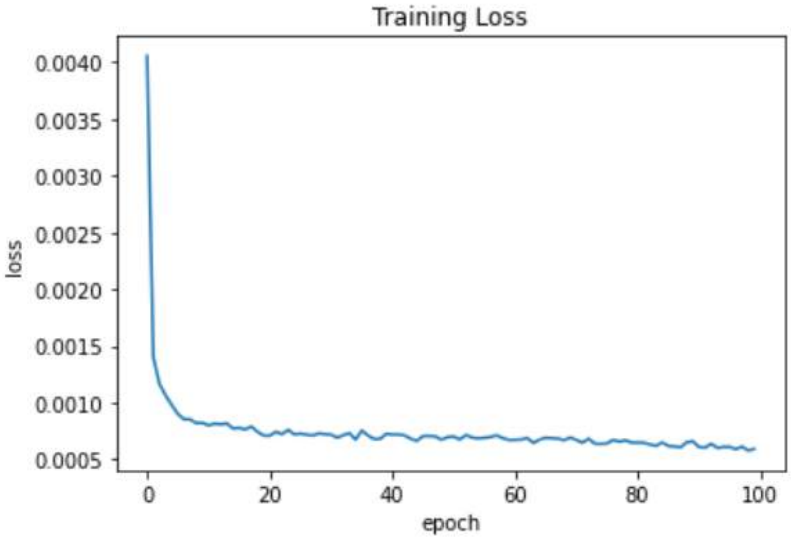


شکل ۵۸- نمودار مقدار loss برای شبکه‌ی LSTM با ۲ ویژگی

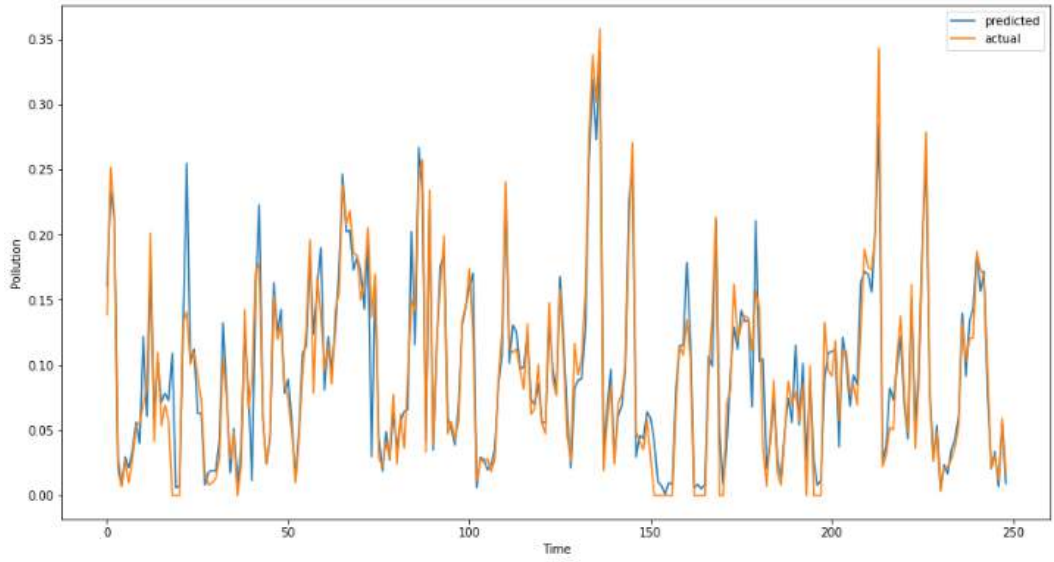


شکل ۵۹- نمودار مقدار حقیقی و پیش‌بینی برای شبکه‌ی LSTM با ۲ ویژگی

MSE = ۰.۰۰۴۹



شکل ۶۰- نمودار مقدار loss برای شبکه‌ی GRU با ۲ ویژگی



شکل ۶۱- نمودار مقدار حقیقی و پیش‌بینی برای شبکه‌ی GRU با ۲ ویژگی

سوال ۲ – نقصان دادگان

۱) برای هر ستون از قسمت دادگان آموزش، به صورت رندم ۲۰ درصد از دادگان را حذف کنید .

همانطور که در فایل نوتبوک مشخص است، به طور تصادفی ۲۰٪ از هر ویژگی داده‌ها را حذف کرده و دیتاست جدید را به اسم missing_train_data تشکیل میدهیم.

۲) با تحقیق در منابع ۳ روش بر طرف کردن نقصان دادگان را بیابید و به صورت کامل شرح دهید .

در حالت کلی هدف ما از بین بردن داده‌هایی است که که دارای missing value هستند در نتیجه یکی از راه‌های معمول در صورت داشتن دیتاست بزرگ، حذف این داده‌هاست. اما در این سوال فرض ما این است که این کار عملی نیست و راه هایی برای پرکردن این داده ها پیشنهاد می‌دهیم. برای مثال روش هایی مثل صفر جایگزین کردن، جایگزین کردن عنصر با بیشترین تکرار، جایگزینی میانگین یا میانه و ... . ادامه ۳ روش برای پر کردن این اطلاعات از دست رفته پیشنهاد و توضیح میدهیم.

روش اول) میانگین

میتوان بجای داده‌ی از دست رفته، میانگین آن همان ویژگی را جایگزین کرد بدین صورت که برای هر فیچر، میانگین محاسبه شده و missing value ها را پر میکنیم و در نهایت یک دیتاست بدون missing value داریم.

روش دوم) KNN imputation

در این روش، برای هر بردار داده که دارای missing value است، در فضا، با توجه سایر ویژگی‌هایش، به دنبال k تا از نزدیک ترین همسایگانش میگردیم. برای محاسبه‌ی فاصله میتوان از فاصله‌ی اقلیدسی یا سایر فاصله‌ها استفاده کرد و برای هر ویژگی میتوان وزن دلخواهی متصور شد.

پس از پیدا شدن این k همسایه، برای پرکردن آن missing value، از این فیچر خاص این k همسایه میانگین گرفته میشود.

روش سوم) Iterative Imputer

در این روش برای یافتن feature ای که مقدارش مشخص نیست از بقیه feature های داده استفاده می‌شود که به این روش multivariate feature imputation گفته می‌شود. در IterativeImputer هر feature با نقصان داده را تابعی از بقیه feature ها در نظر می‌گیرد و از این مقدار تخمین زده شده برای پر کردن نقصان داده‌ها استفاده می‌کند. این کار به صورت iterative انجام می‌شود به این شکل که هر بار یکی از ستون از ویژگی‌ها y یا خروجی در نظر گرفته می‌شود و بقیه ویژگی‌ها x یا ورودی در نظر گرفته

می‌شوند سپس یک regressor روی این داده‌ها برای y های معلوم fit می‌شود و سپس از این regressor برای پیش بینی مقدارهای نامعلوم استفاده می‌شود.

۴)یک روش را به دلخواه انتخاب کنید و با استفاده از آن دادگان از بین رفته را پیش بینی کنید.

این بخش هم به کمک کتابخانه‌ی sklearn و هم به صورت دستی پیاده سازی شده و نتایج هر دو روش در ادامه آورده شده است.

روش های مختلفی با استفاده از sklearn داخل فایل نوتبوک بررسی شد از جمله knn و median و mean و ... ولی بهترین نتیجه از Iterative Imputer با تخمین‌گر ExtraTreesRegressor بدست آمد.

از آنجا که با استفاده از knn، نتایج بهتری نسبت به mean و median بدست آمد، برای پیاده‌سازی دستی، از knn استفاده میکنیم.

برای پیدا کردن نزدیک‌ترین همسایه‌های یک بردار ویژگی، فیچری که مقدار آن np.nan است را در نظر نمیگیریم و در فضای n – 1 بعدی دنبال این همسایه‌ها میگردیم. در صورتی که یکی از فیچرهای بردار دوم هم missing بود، از میانگین فعلی آن ویژگی برای پر کردن آن استفاده میکنیم.

به همین دلیل در قسمت‌های 5 و 6 از این روش استفاده شده است.

۵)با استفاده از روش خطای MSE ، میزان خطای موارد پیش بینی شده برای دادگان از دست رفته را برای هر ستون را گزارش دهید.

خطا را برای هر ستون محاسبه کرده و در نهایت هم میانگین آن ها را برای هر دو حالت حساب میکنیم.

حالت اول) روش iterative با کتابخانه‌ی sklearn

Mean MSE = 0.002

```
Iterative
   MSE for column 0  :  0.0013300512
   MSE for column 1  :  0.001958104
   MSE for column 2  :  0.0018683493
   MSE for column 3  :  0.0020026197
   MSE for column 4  :  0.014336703
   MSE for column 5  :  0.0014095402
   MSE for column 6  :  0.0001805558
   MSE for column 7  :  0.0003326872
 mean :  0.0029273261
```

شکل 62- محاسبه‌ی خطا برای هر ستون با روش Iterative با sklearn

حالت دوم) روش KNN بدون استفاده از کتابخانه

Mean MSE = 0.003

```
  MSE for column 0  :  0.0015277078
  MSE for column 1  :  0.003069666
  MSE for column 2  :  0.002864277
  MSE for column 3  :  0.0028878823
  MSE for column 4  :  0.016698748
  MSE for column 5  :  0.001507774
  MSE for column 6  :  0.00026550636
  MSE for column 7  :  0.00039839643
mean :  0.0036524946
```

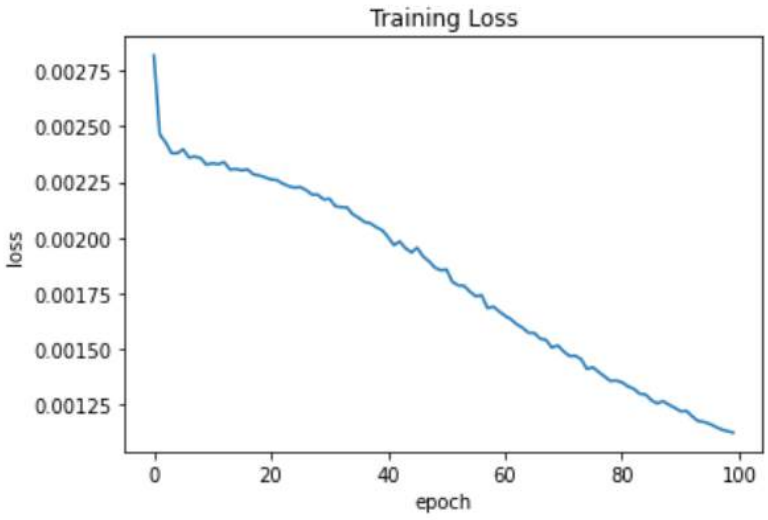
شکل 63- محاسبه‌ی خطا برای هر ستون با روش KNN

برای پیاده‌سازی این بخش، برای هر داده(سطر) هر کدام از ستون‌هایی که مقدار نداشتند را به صورت جدا در نظر می‌گرفتیم و اگر سطر ستون‌های نامشخص دیگری نیز داشت ابتدا آن‌ها را با میانگین همان ستون پر می‌کردیم و همین کار را برای تمامی داده‌های دیگر نیز انجام دادیم و سپس فاصله اقلیدسی تمامی داده‌ها را از داده موردنظر پیدا می‌کردیم و k تا (این‌جا ۵۴۰ عدد) از نزدیکترین همسایه هایش را انتخاب کرده و میانگین مقادیر ستون مورد نظر را در آن‌ها برای داده مورد نظرمان قرار می‌دادیم و این کار را برای تمامی داده‌های نامشخص تکرار کردیم.

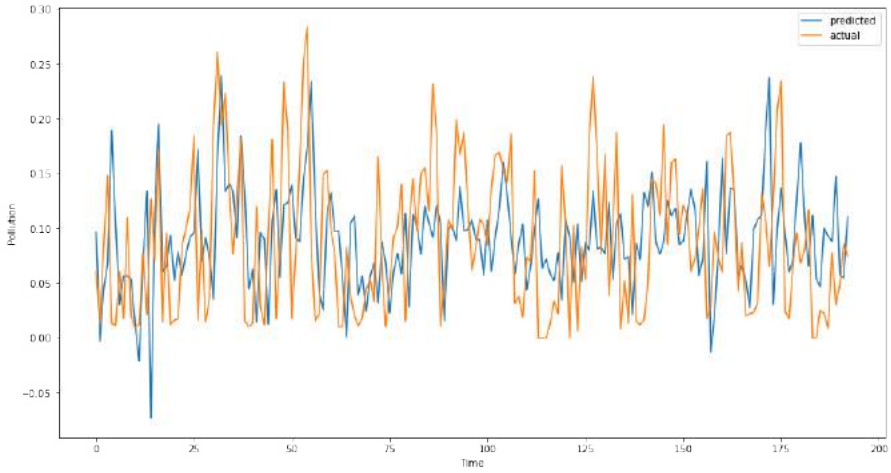
۶) اکنون با استفاده از دادگان پیش بینی شده، برای سلول های GRU ، LSTM و یکی از توابع هزینه به دلخواه، میزان آلودگی روزانه را پیش بینی کنید.

با توجه به دیتاست جدید، دو شبکه زیر را آموزش میدهیم.

MSE = ۰.۰۰۱

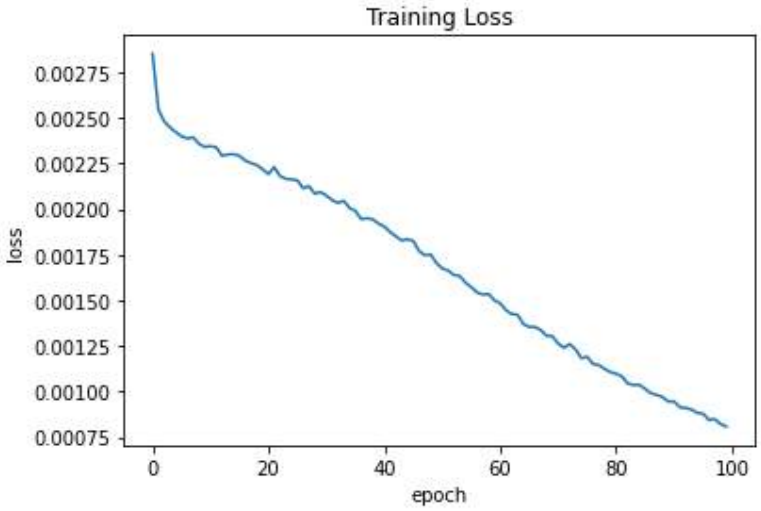


شکل 64- نمودار مقدار loss برای شبکه‌ی LSTM با missing value

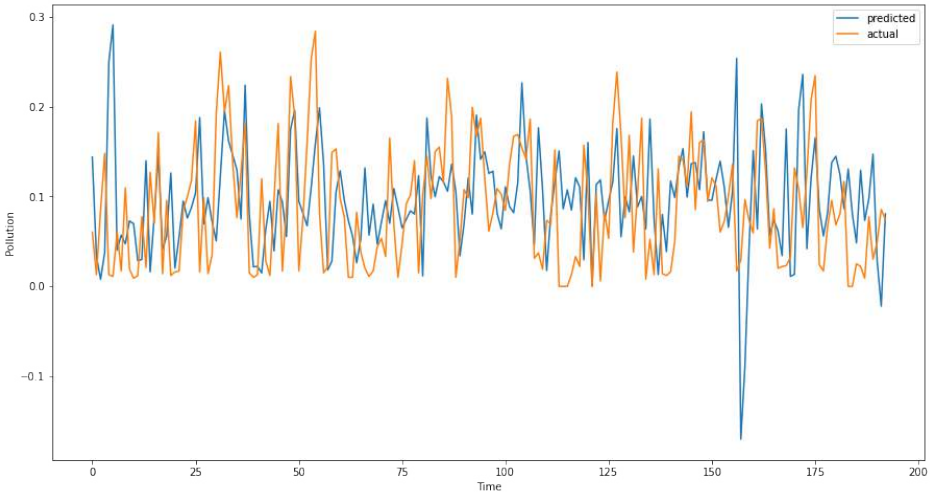


شکل 65- نمودار مقدار حقیقی و پیش‌بینی برای شبکه‌ی LSTM

MSE = ۰.۰۰۱



شکل ۶۶- نمودار مقدار loss برای شبکه‌ی GRUبا missing value



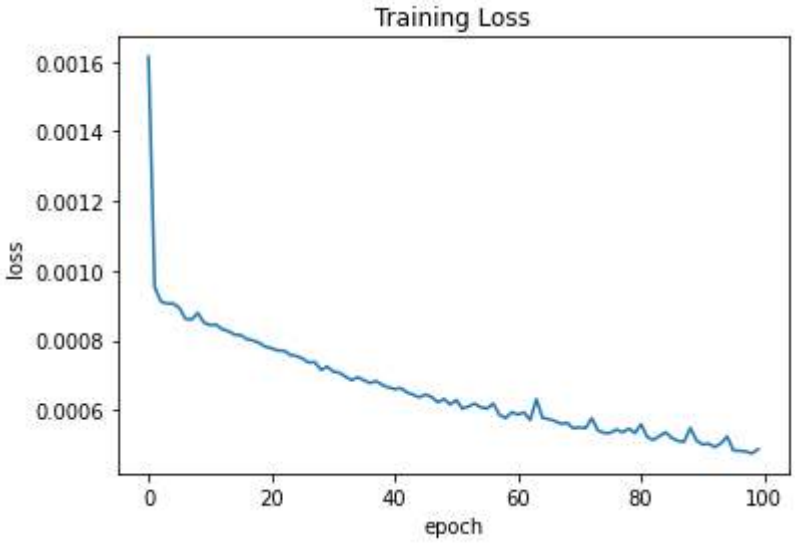
شکل ۶۷- نمودار مقدار حقیقی و پیش‌بینی برای شبکه‌ی GRU

با مقایسه‌ی این دو شبکه با حالت‌های با دیتاست کامل میتوانیم به نتایج زیر برسیم.

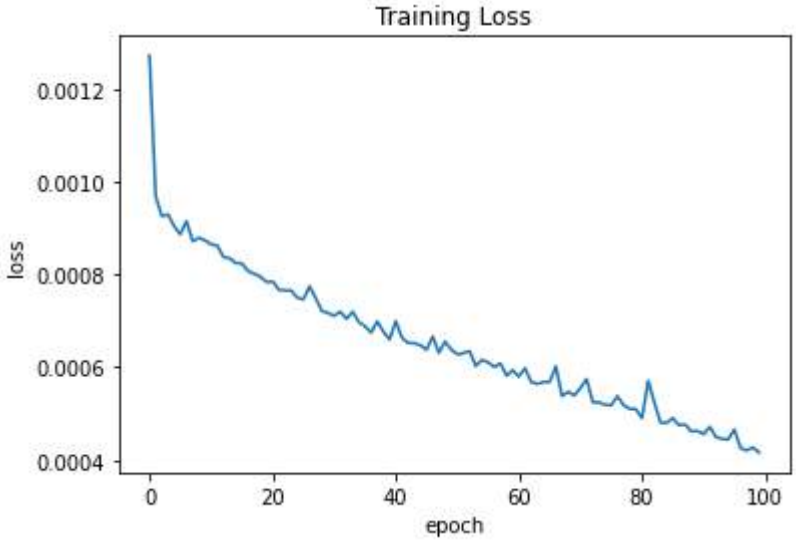
loss شبکه‌ی LSTM در هر حالت قبلی ۰.۰۰۵۳ بوده است که این موضوع نشان‌دهنده‌ی این است که imputation با وجود عملکرد خوب، باز هم به خوبی حالت اولیه نیست.

loss شبکه‌ی GRU در هر حالت قبلی ۰.۰۰۵۳ بوده است که این موضوع نشان‌دهنده‌ی این است که imputation با وجود عملکرد خوب، باز هم به خوبی حالت اولیه نیست.

در ادامه نمودارهای loss برای حالت بدون missing value را نمایش میدهیم.



شکل ۶۸- نمودار مقدار loss برای شبکه‌ی LSTM با دیتاست اصلی



شکل ۶۹- نمودار مقدار loss برای شبکه‌ی GRU با دیتاست اصلی