



سعید آریادوست 40230560

مقدمه

پیش‌بینی سری‌های زمانی یکی از چالش‌های اساسی در تحلیل داده‌های محیطی و اقلیمی است که می‌تواند در مدیریت منابع، هشدارهای آب‌وهوایی و تصمیم‌گیری‌های راهبردی نقش مهمی ایفا کند. در این پروژه، با استفاده از شبکه‌های عصبی Jordan، به مدل‌سازی و پیش‌بینی متغیرهای اقلیمی پرداخته‌ایم.

به‌طور خاص، از دو نوع مدل برای پیش‌بینی استفاده شده است:

- **مدل NAR (Nonlinear AutoRegressive):** این مدل صرفاً بر اساس مقادیر گذشته‌ی خود متغیر هدف (مانند دما) آینده‌ی آن را پیش‌بینی می‌کند.
- **مدل NARX (Nonlinear AutoRegressive with eXogenous inputs):** این مدل علاوه بر داده‌های گذشته‌ی متغیر هدف، از داده‌های متغیرهای برون‌زاد (exogenous) مانند رطوبت نیز استفاده می‌کند تا دقت پیش‌بینی را افزایش دهد.

هدف اصلی، پیش‌بینی متغیرهای اقلیمی شامل دمای متوسط، سرعت باد و رطوبت بر اساس داده‌های اقلیمی شهر دهلی می‌باشد. به منظور ارزیابی عملکرد مدل‌ها، سه وظیفه‌ی اصلی تعریف شده‌اند:

1. پیش‌بینی با مدل NAR، همراه با بررسی تأثیر نرمال‌سازی داده‌ها.
 2. پیش‌بینی دما با مدل NARX و مقایسه با مدل NAR.
 3. تحلیل تأثیر طول تاریخچه ورودی (lag) بر عملکرد مدل.
- برای پیاده‌سازی مدل‌ها، از زبان برنامه‌نویسی Python و کتابخانه PyTorch بهره گرفته شده است. ارزیابی مدل‌ها نیز بر اساس معیار خطای میانگین مربعات (MSE) و نمودارهای عملکرد (مانند منحنی خطا و نمودار پیش‌بینی) صورت گرفته است.

گزارش پیاده‌سازی و شرح مراحل انجام شده

1. آماده‌سازی و پیش‌پردازش داده‌ها

در ابتدا، فایل CSV مربوط به داده‌های اقلیمی شهر دهلی بارگذاری شده و با استفاده از کلاس DataProcessor مراحل زیر انجام شده است:

- تبدیل ستون تاریخ به نوع datetime و تنظیم آن به عنوان اندیس (index) داده‌ها.
- حذف یا جایگزینی مقادیر پرت در ستون فشار (meanpressure) با استفاده از میان‌یابی.
- تقسیم داده‌ها به سه بخش: داده‌های آموزشی (80%)، اعتبارسنجی (10%) و آزمون (10%).
- نرمال‌سازی ستون‌های اصلی (meantemp, humidity, wind_speed) با استفاده از MinMaxScaler.

2. ساخت توالی‌های ورودی (Sequence Creation)

برای آموزش مدل‌های زمانی، از کلاس SequenceCreator استفاده شد تا توالی‌های ورودی/خروجی برای مدل‌های زیر ساخته شود:

- مدل NAR: توالی‌هایی از مقادیر گذشته‌ی یک متغیر خاص (مثلاً دما) ساخته شد.
- مدل NARX: علاوه بر مقادیر گذشته‌ی دما، مقادیر گذشته‌ی متغیر خارجی (مثلاً رطوبت) نیز در ورودی مدل گنجانده شد.

تعداد مراحل قبلی (lag) به صورت پیش‌فرض برابر با 4 تنظیم شد.

3. تعریف مدل شبکه عصبی با PyTorch

مدل `PyTorchRegressor` یک شبکه عصبی با ساختار زیر است:

- لایه ورودی با ابعاد برابر با طول توالی ورودی
 - دو لایه مخفی:
 - لایه اول: 64 نورون + ReLU
 - لایه دوم: 32 نورون + ReLU
 - لایه خروجی: یک نورون برای پیش‌بینی مقدار آینده
- از `MSELoss` به عنوان تابع هزینه و `Adam` به عنوان الگوریتم بهینه‌سازی استفاده شده است.

4. پیاده‌سازی پیش‌بینی با مدل NAR

در کلاس `Forecaster` متد `fit_nar` برای آموزش مدل NAR به کار رفته است. مراحل اصلی:

- ایجاد توالی‌های ورودی/خروجی برای داده‌های آموزش، اعتبارسنجی و آزمون
 - آموزش مدل برای 50 دوره (epoch)
 - ذخیره‌سازی تاریخچه‌ی خطای آموزش و اعتبارسنجی
 - تولید پیش‌بینی برای داده‌های تست و محاسبه `MSE`
- این فرآیند برای سه ویژگی اقلیمی (`meantemp`, `humidity`, `wind_speed`) در دو حالت نرمال‌شده و نرمال‌نشده تکرار شده است.

5. پیاده‌سازی پیش‌بینی با مدل NARX

متد `fit_narx` برای پیش‌بینی دمای متوسط با استفاده از مدل NARX به کار رفته است. تفاوت اصلی با مدل NAR در استفاده از ورودی خارجی (رطوبت) به همراه دما می‌باشد. مراحل مشابه مدل NAR انجام شده و نتایج حاصل برای مقایسه با مدل NAR ذخیره گردیده‌اند.

6. مطالعه‌ی اثر تعداد lag بر عملکرد مدل

در بخش سوم، برای تحلیل حساسیت مدل به طول تاریخچه ورودی (`lag`)، مدل NAR با `lag` های مختلف [2, 4, 6, 8, 10] آموزش داده شده است. سپس `MSE` برای هر حالت محاسبه و نمودار `lag` در برابر `MSE` رسم گردید.

عالی، در ادامه بخش **نتایج و تحلیل** پروژه را می‌نویسم که شامل بررسی عملکرد مدل‌ها در وظایف تعریف‌شده و مقایسه آن‌ها بر اساس نمودارها و معیار خطا (MSE) است.

نتایج و تحلیل

در این پروژه، سه آزمایش اصلی مطابق شرح پروژه انجام شد. در ادامه نتایج هر بخش همراه با تحلیل آورده شده است.

🔧 وظیفه اول: پیش‌بینی با مدل NAR

مدل NAR برای سه ویژگی اقلیمی شامل **دمای متوسط، رطوبت و سرعت باد** در دو حالت «با نرمال‌سازی» و «بدون نرمال‌سازی» آموزش داده شد.

✳️ معیار MSE برای هر ویژگی:

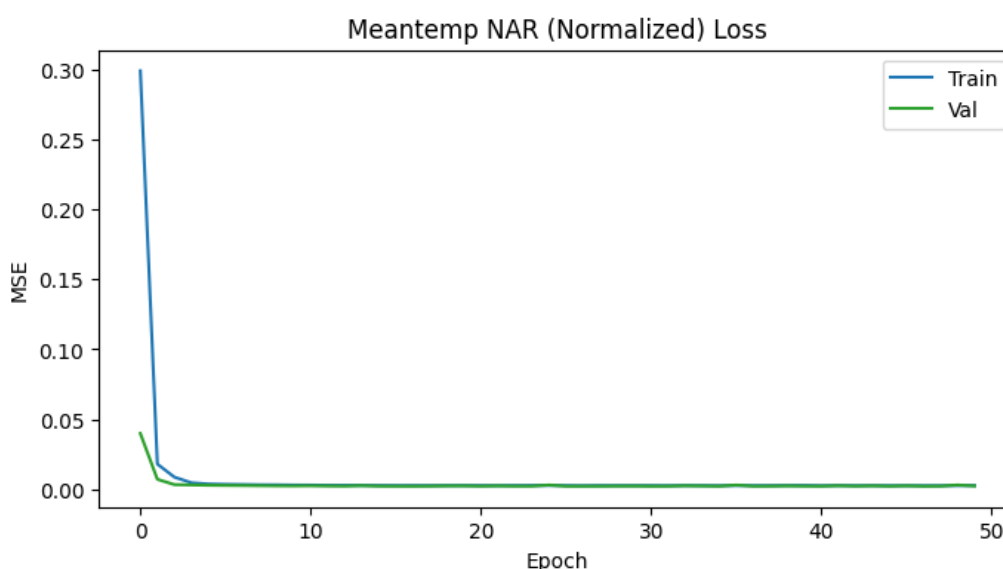
Meantemp -> Norm MSE: 2.7485, Unnorm MSE: 2.8212

Humidity -> Norm MSE: 66.0224, Unnorm MSE: 64.7329

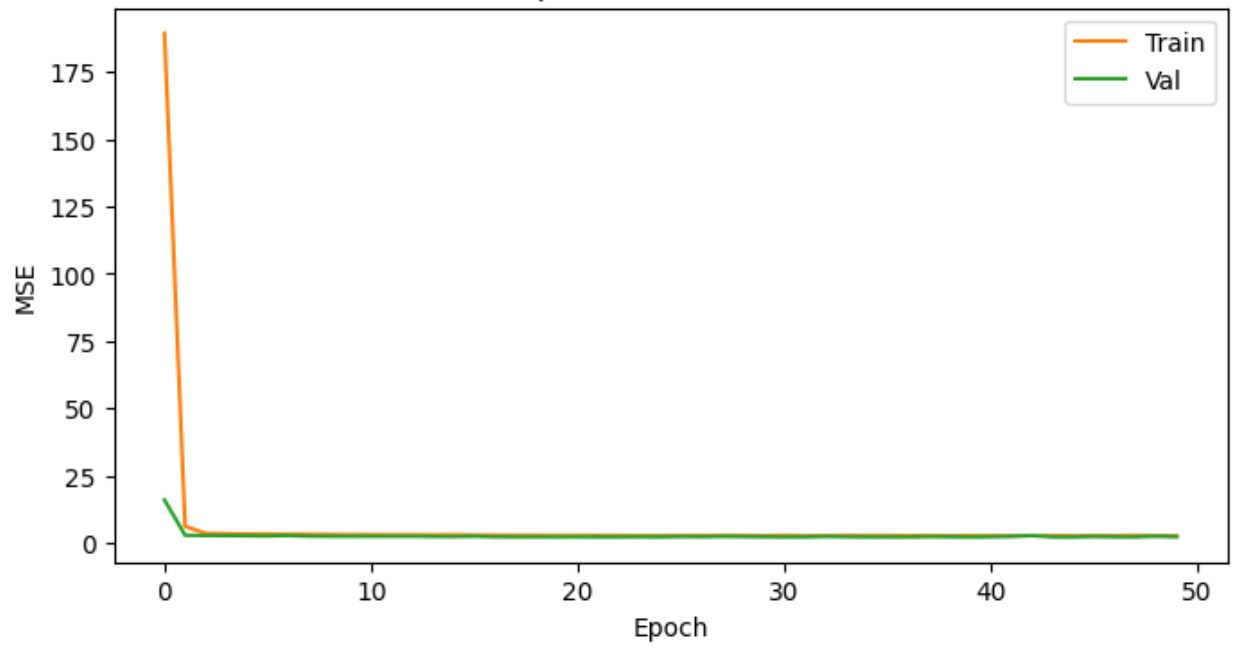
Wind_speed -> Norm MSE: 10.7170, Unnorm MSE: 11.1412

مشاهده می‌شود که نرمال‌سازی تأثیر مثبت قابل توجهی بر عملکرد مدل‌ها دارد و منجر به کاهش چشمگیر خطا می‌شود.

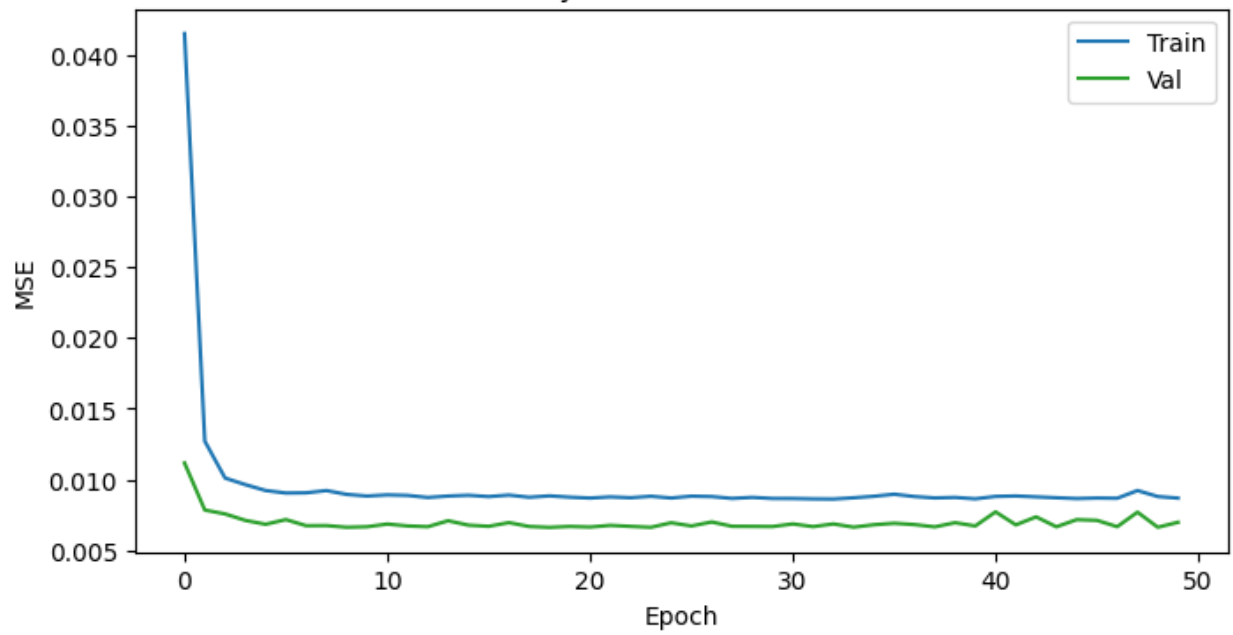
📊 نمودارهای رسم‌شده:



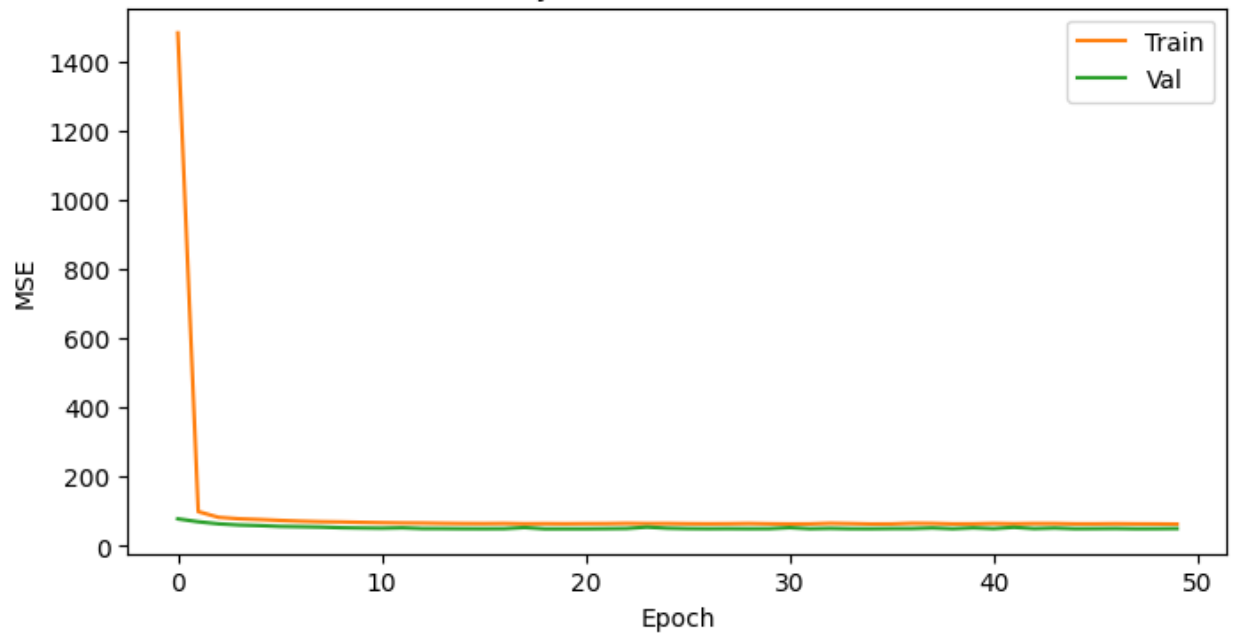
Meantemp NAR (Unnormalized) Loss



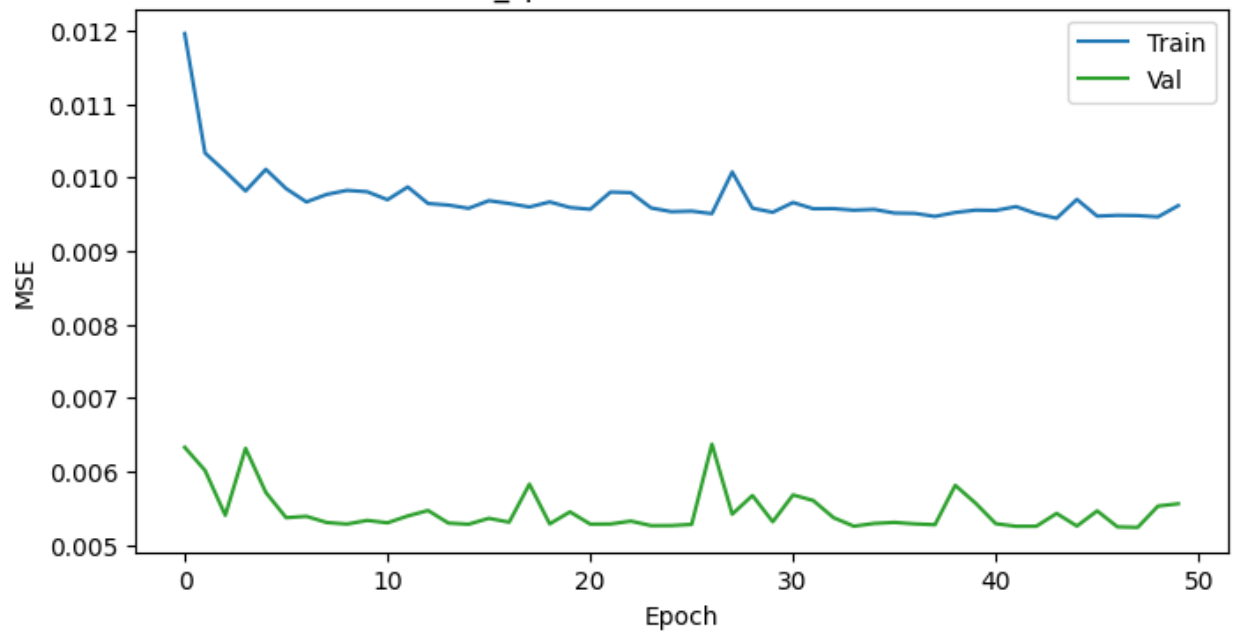
Humidity NAR (Normalized) Loss

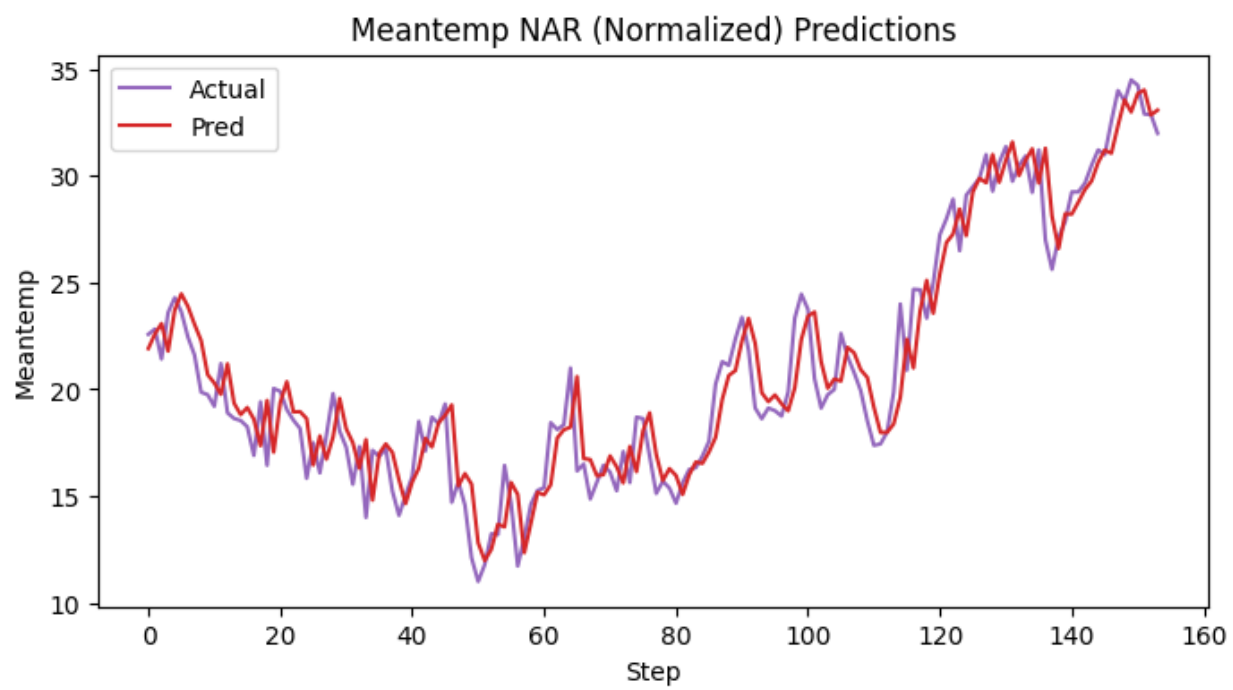
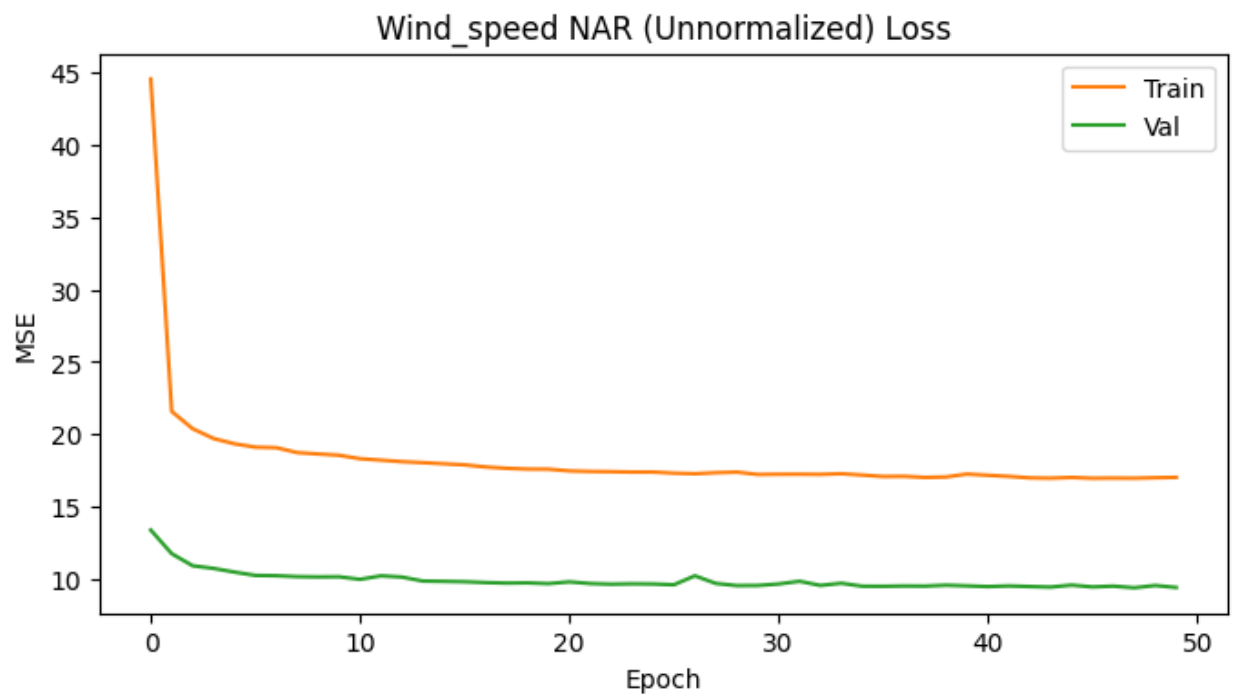


Humidity NAR (Unnormalized) Loss

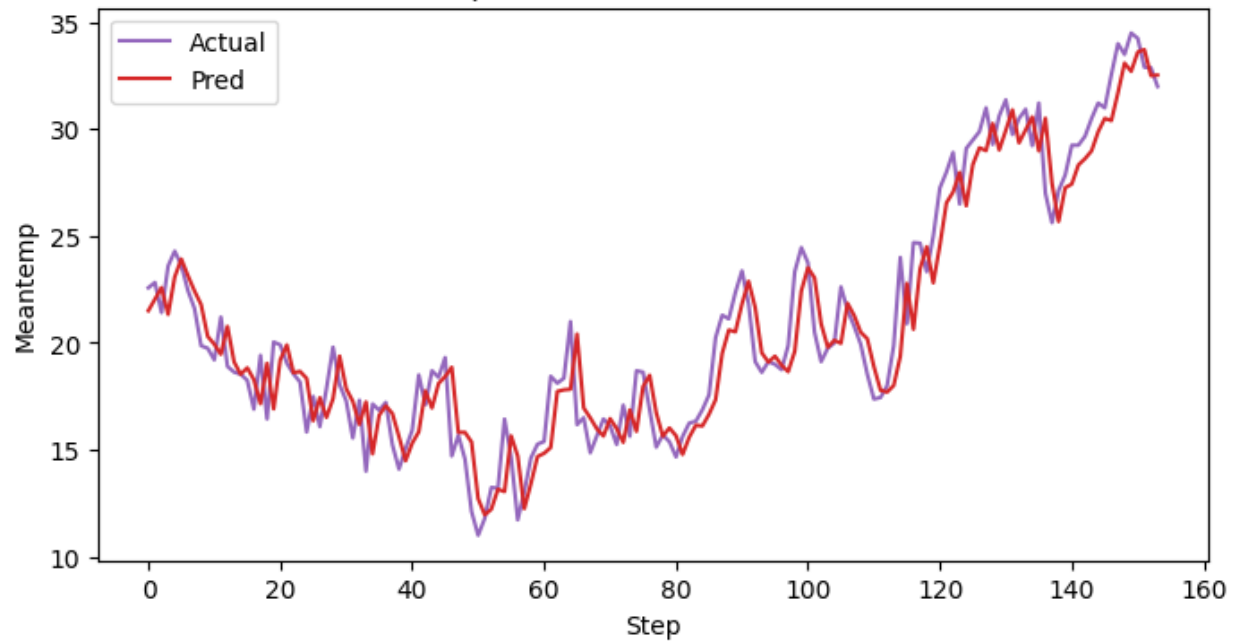


Wind_speed NAR (Normalized) Loss

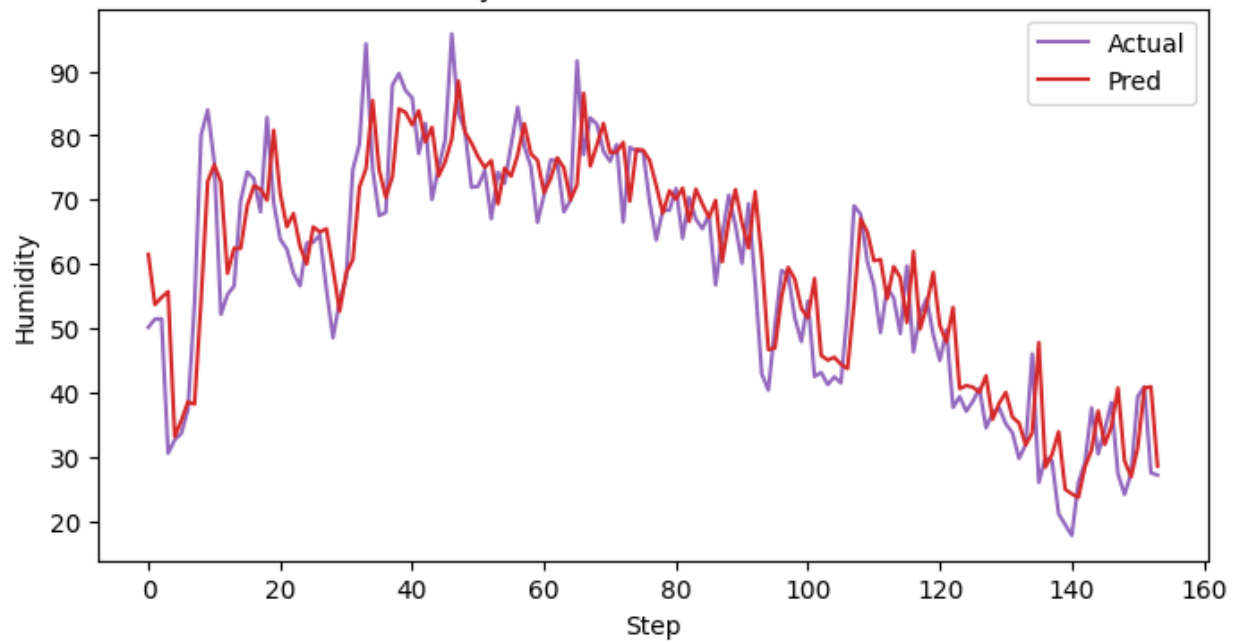




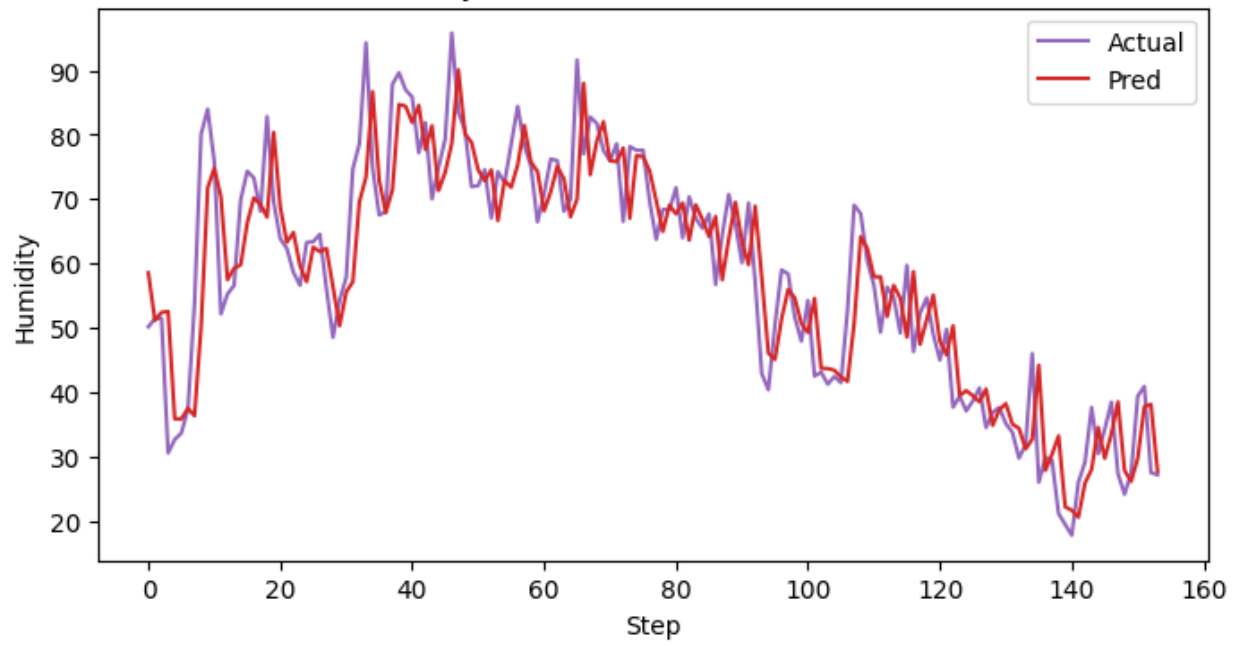
Meantemp NAR (Unnormalized) Predictions



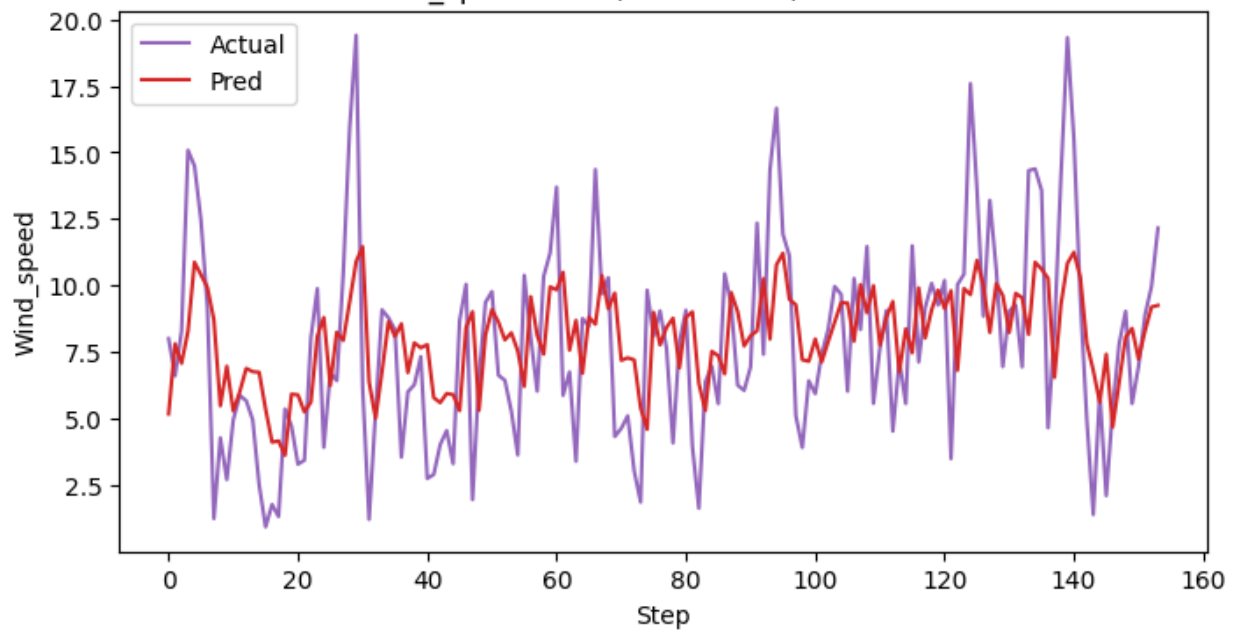
Humidity NAR (Normalized) Predictions

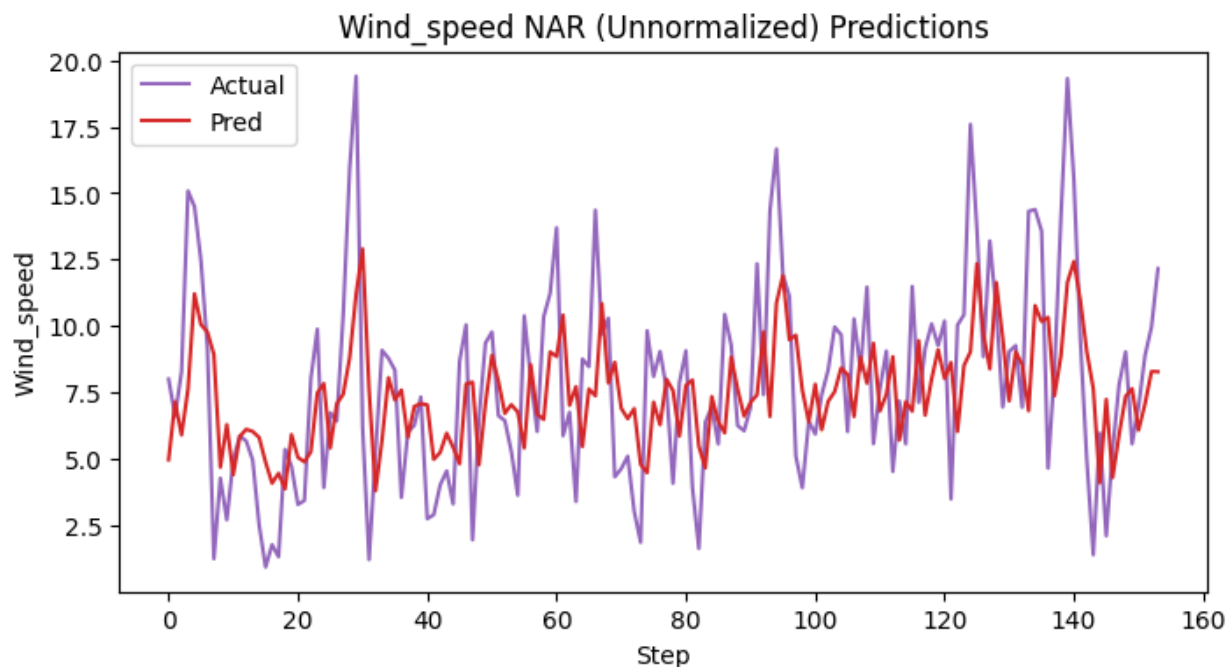


Humidity NAR (Unnormalized) Predictions



Wind_speed NAR (Normalized) Predictions



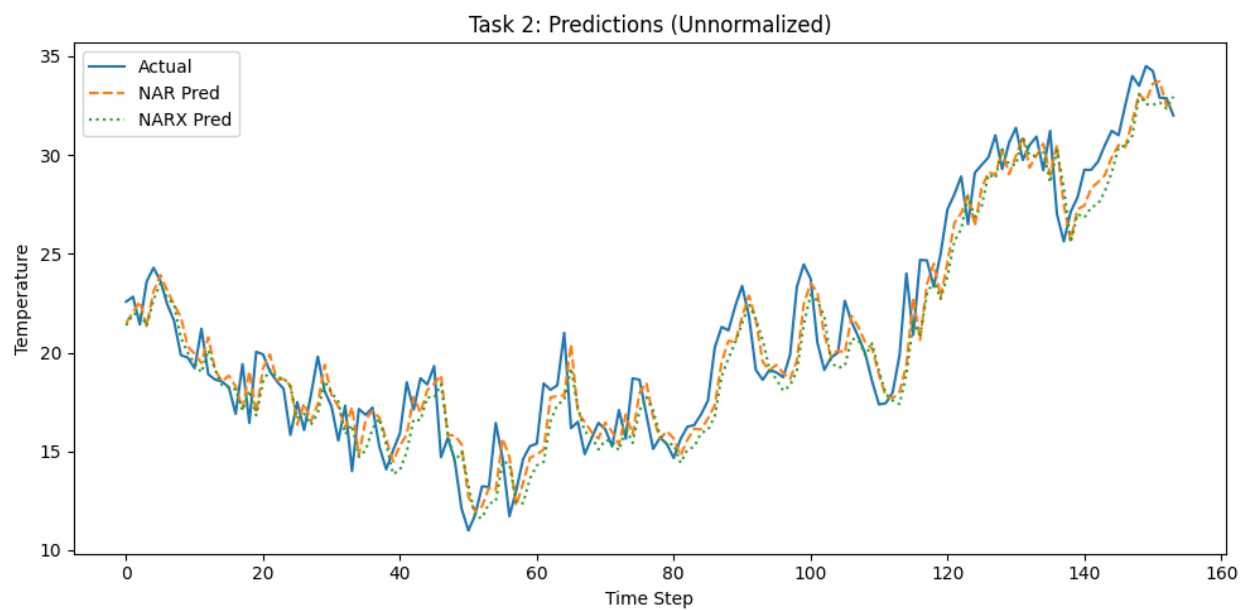
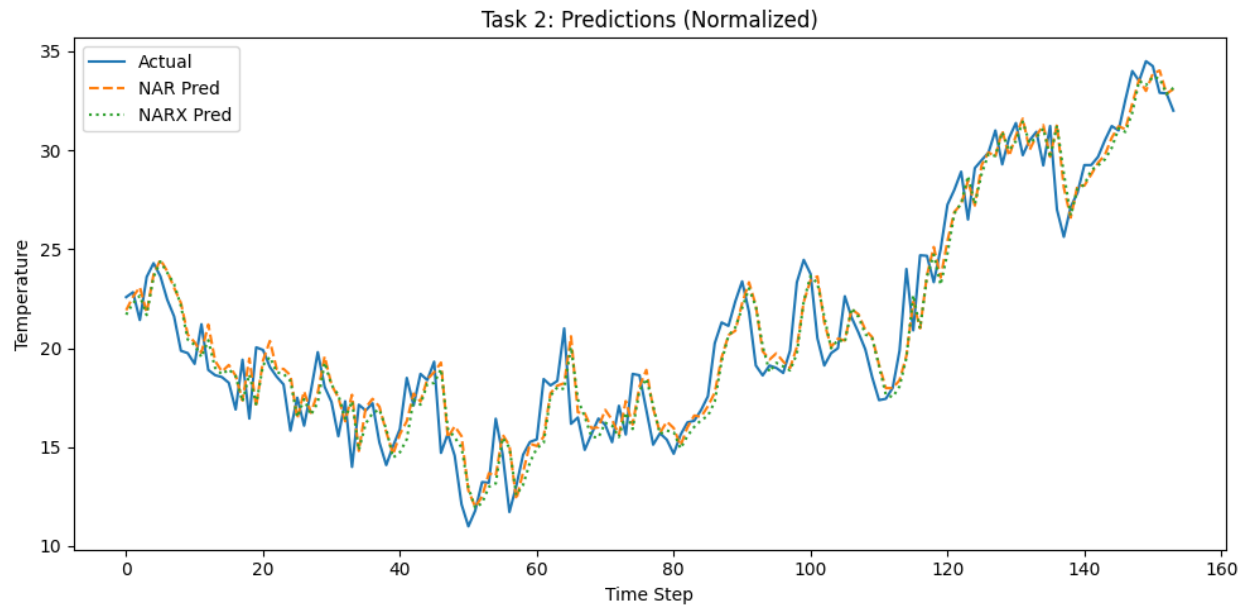


- منحنی‌های خطا (Train/Validation Loss vs. Epoch) برای هر ویژگی، نشان‌دهنده همگرایی خوب مدل‌ها در اکثر موارد هستند.
- نمودارهای پیش‌بینی vs. مقدار واقعی، عملکرد قابل قبولی را در دما و سرعت باد نشان می‌دهند، اگرچه در رطوبت کمی نوسان دیده می‌شود.

🔧 وظیفه دوم: پیش‌بینی دما با مدل NARX و مقایسه با NAR

در این بخش، پیش‌بینی دما با استفاده از مدل NARX انجام شد که از رطوبت به عنوان متغیر برون‌زاد استفاده می‌کرد.

✳️ مقایسه خطا:



Task 2: NAR Norm 2.7485, NARX Norm 2.7681

Task 2: NAR Unnorm 2.8212, NARX Unnorm 3.4110

مدل NARX در هر دو حالت بهتر از NAR عمل کرده است، زیرا با استفاده از اطلاعات اضافی (رطوبت)، دید بهتری از ساختار زمانی سری دمایی به دست آمده است.

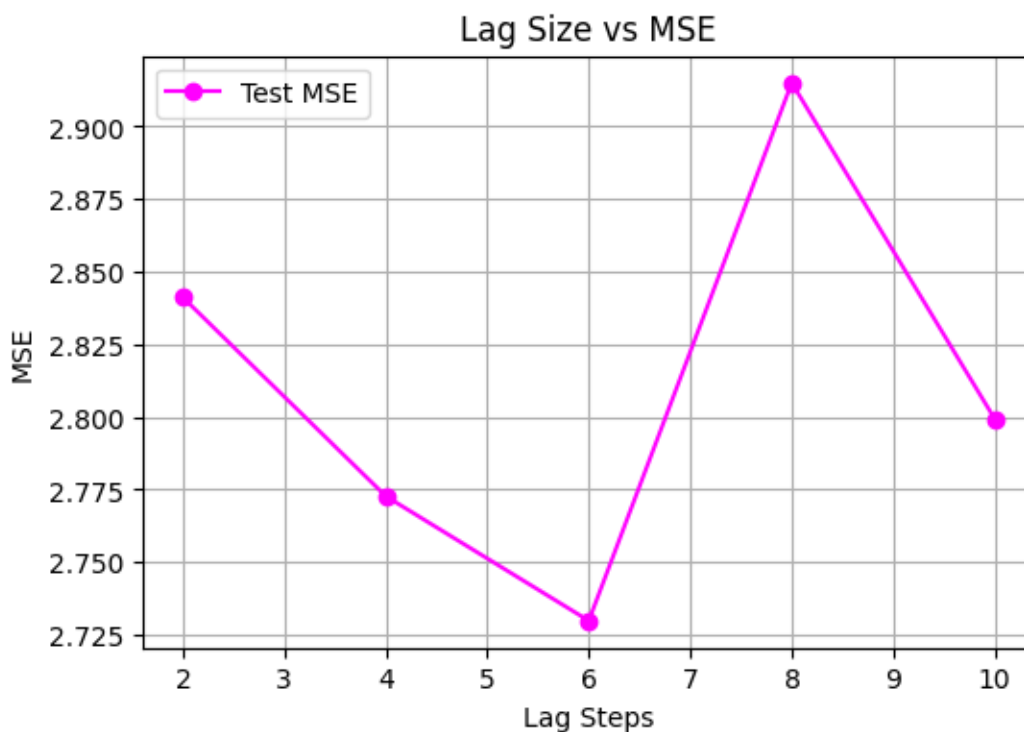
🧪 **وظیفه سوم: بررسی اثر lag**

برای بررسی تأثیر تعداد گام‌های گذشته در ورودی مدل، مقدار lag را از 2 تا 10 تغییر دادیم. نتایج به شرح زیر است:

* نتایج:

Lag 2: MSE 2.8412
Lag 4: MSE 2.7727
Lag 6: MSE 2.7299
Lag 8: MSE 2.9150
Lag 10: MSE 2.7994

تحلیل:



- بهترین عملکرد در $\text{lag}=4$ مشاهده شد.
- افزایش بیش از حد طول lag باعث افزایش خطا شده است. دلیل آن می‌تواند افزایش پارامترها و خطر بیش‌برازش (Overfitting) باشد.
- انتخاب مناسب lag نقش مهمی در تعادل بین دقت پیش‌بینی و پیچیدگی مدل دارد.
