Predicting Sales of Walmart Products

Rudraksh Mishra

Ambika Chundru

Deep Learning (Spring, 2022)

Dr. Youakim Badr

PennState

2 min to convince the audience

Context and Introduction

- In the retail industry, forecasting sales accurately is important as this will help improve the revenue by avoiding can avoid wastages and shortages of products.
- Walmart shared its historical sales data on Kaggle as an M5 competition to enhance its forecasting models.

PennState

Problem and Challenges

- **Aim:**
  - The main objective is to estimate or predict the unit sales of Walmart retail goods at stores in various locations for the next 28-days, and 180 days.
  - 30, 490 forecasts must be made for each day.
- **Challenges:**
  - A lot of intermittent values.
  - Some products are introduced later.

PennState

Related solutions

Learning Pytorch LSTM Deep Learning with M5 Data
Updated 2Y ago
37 comments · M5 Forecasting - Accuracy

【日本語】Starter Data Exploration と LSTM
Notebook copied with edits from Rob Mulla · Updated 2Y ago
20 comments · M5 Forecasting - Accuracy

Learning Pytorch Seq2Seq with M5 Data-Set
Updated 2Y ago
6 comments · M5 Forecasting - Accuracy +1

M5-Forecasting : EDA 📈 +LSTM Pytorch Modeling🔥
Updated 2Y ago
22 comments · M5 Forecasting - Accuracy

Transformer-XL Intro & Baseline
Updated 2Y ago
14 comments · M5 Forecasting - Accuracy +2

EDA and an Encoder-Decoder LSTM with 9 features.
Notebook copied with edits from RDizzl3 · Updated 2Y ago
Score: 1.21332 · 18 comments · M5 Forecasting - Accuracy

Baseline LSTM with Keras < 0.7
Updated 2Y ago
Score: 0.69361 · 22 comments · M5 Forecasting - Accuracy

M5 - RNN/Wavenet/N-BEATS Approach
Updated 2Y ago
13 comments · M5 Forecasting - Accuracy +3

Thoughts on creating a good NN model
Updated 2Y ago
3 comments · M5 Forecasting - Accuracy

M5 beginner EDA + CNN
Updated 2Y ago
11 comments · M5 Forecasting - Accuracy

M5: EDA (Plotly) + LSTM Neural Network Vs. XGBoost
Updated 2Y ago
3 comments · M5 Forecasting - Accuracy

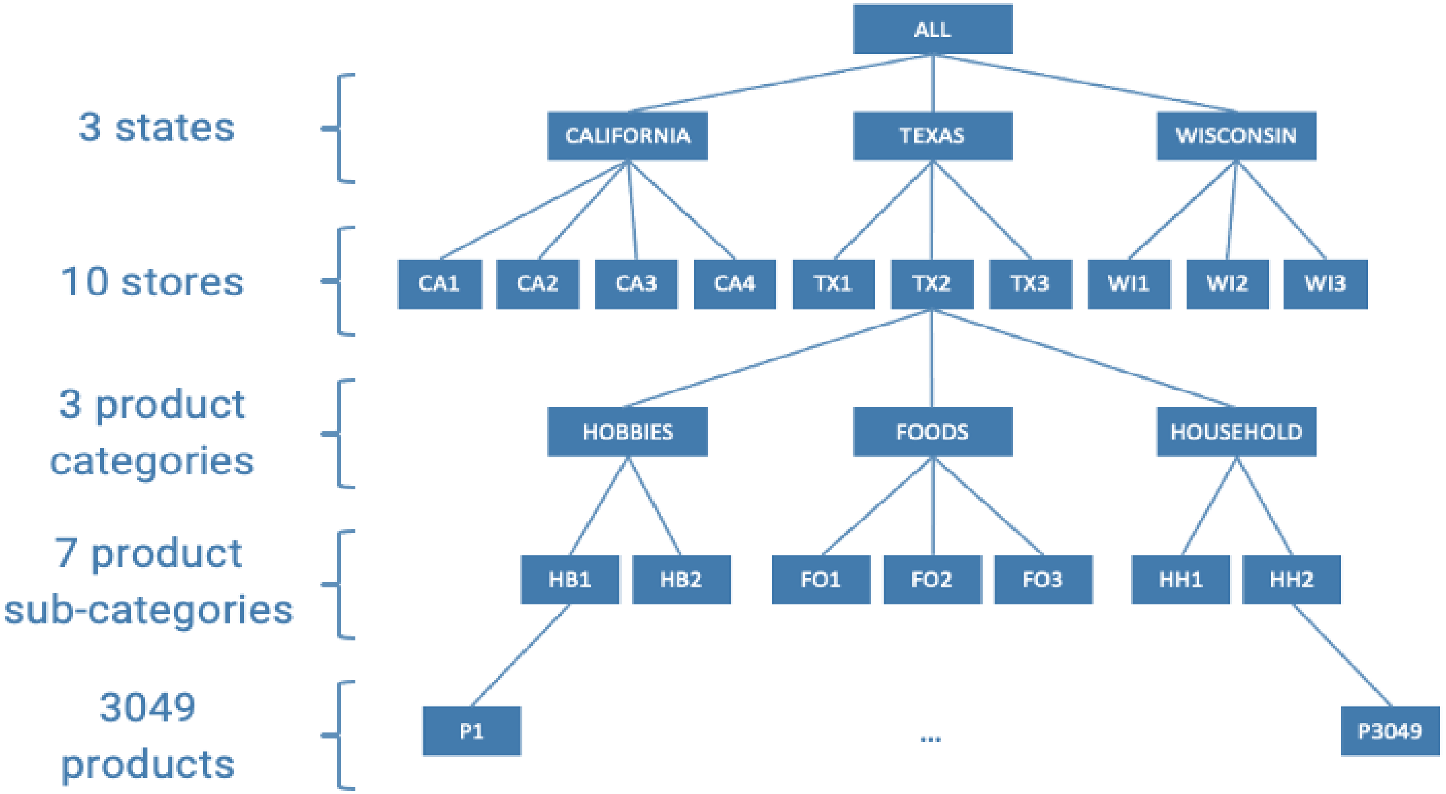
M5-Forecasting : Encoder Decoder with Attention
Updated 2Y ago
11 comments · M5 Forecasting - Accuracy +3

M5 - Web-Traffic-Implementation Pytorch❤
Updated 2Y ago
16 comments · M5 Forecasting - Accuracy

Deep Learning RNN for m5-forecasting
Updated 3mo ago
Score: 1.12716 · 10 comments · M5 Forecasting - Accuracy

About Data

- **3,0490 products**
- **3 product categories:** Hobbies, Foods, and Household
- **7 Departments:** 2, 3, 2
- **3 states:** CA, TX, WI
- **10 stores:** 4, 3, 3



Overview of Solution / Contributions

- Our Solution involves 2 models CNN-LSTM model and DA-RNN model.
- CNN-LSTM
- CNN: To learn the local trend features and remove noise.
- LSTM: To capture long temporal dependencies
- DA-RNN: Dual-stage Attention-based Recurrent Neural Network

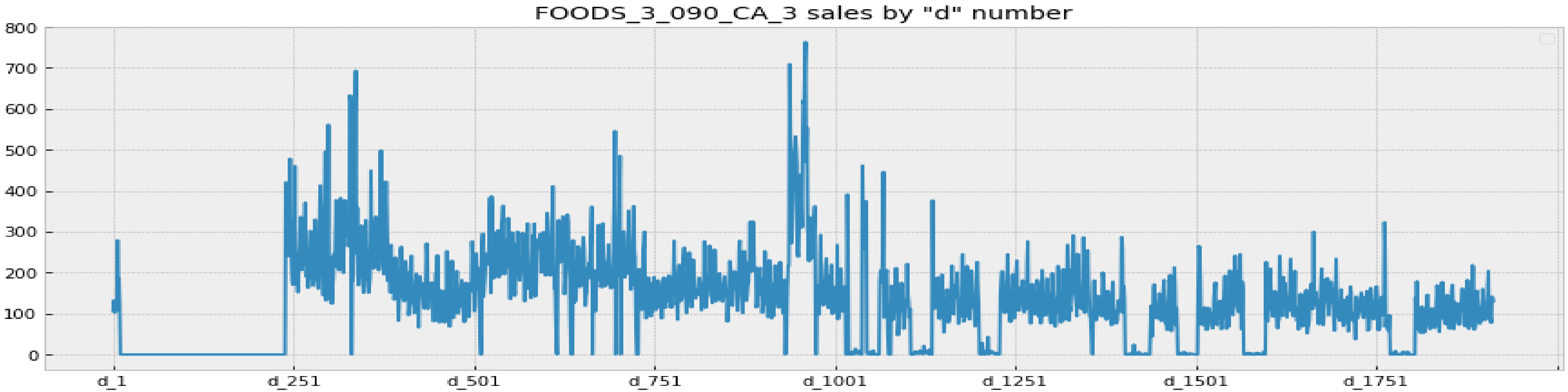
Data Collecting

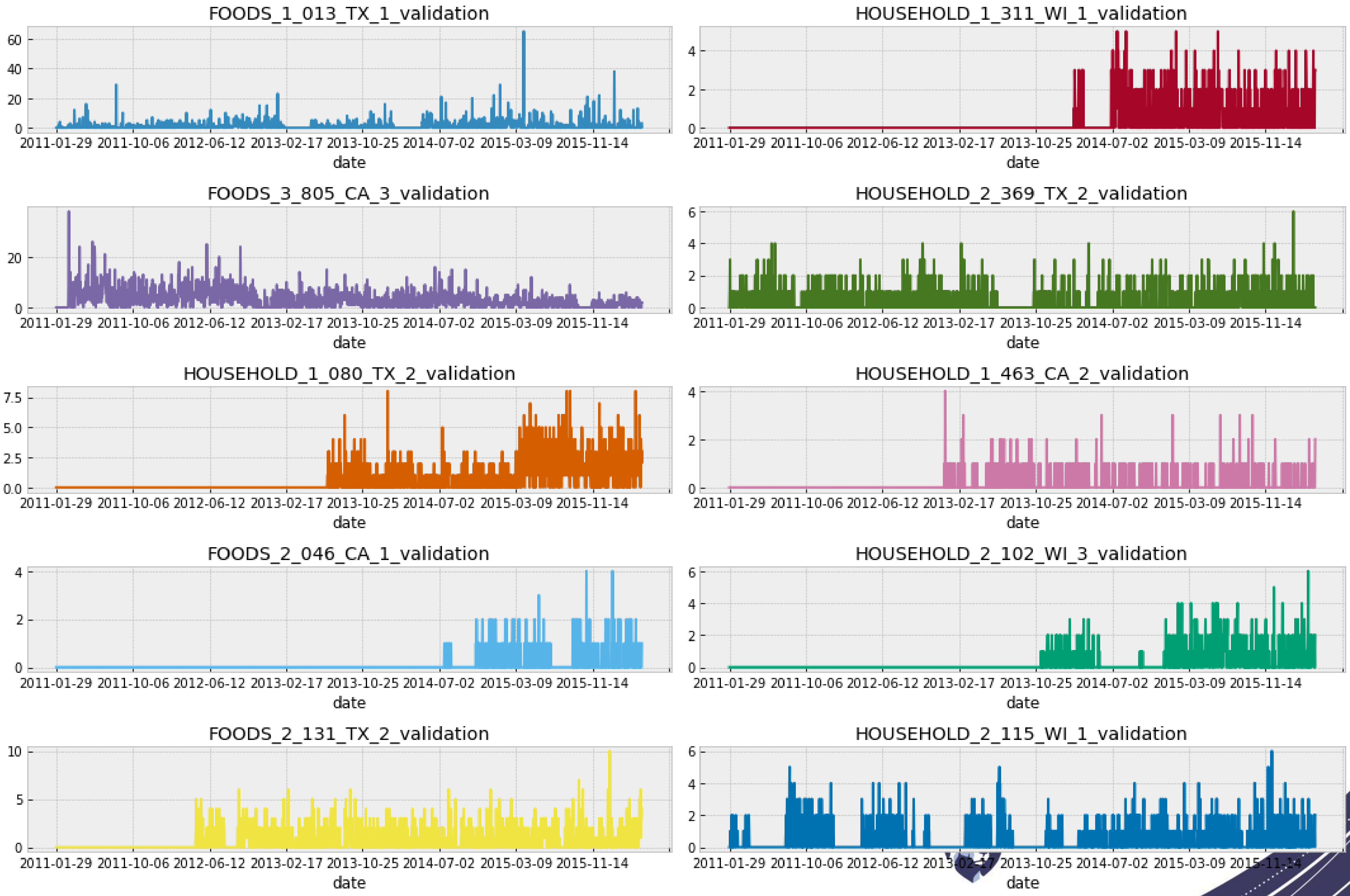
- The 5 years time-series sales data set of various product over various stores was published on Kaggle for the M5 completion.
- The data was published by Makridakis Open Forecasting Center (MOFC) at University of Nicosia who conduct cutting-edge forecasting research and provides business forecast training.

	id	item_id	dept_id	cat_id	store_id	state_id	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_10	d_11	d_12
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	0	0	0	0	0	0	0	0
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	0	0	0	0	0	0	0	0
2	HOBBIES_1_003_CA_1_evaluation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	0	0	0	0	0	0	0	0
3	HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	0	0	0	0	0	0	0	0
4	HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	0	0	0	0	0	0	0	0

EDA

The Below graph depicts the sales of a product over 1941 days.
0 sales indicate that the item is out of stock.

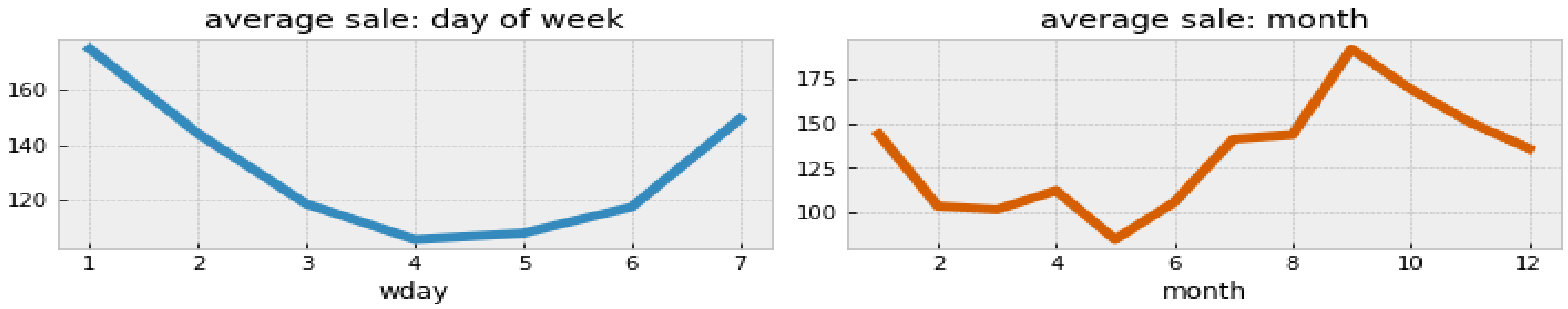



FOODS_1_013_TX_1_validation
0 20 40 60
2011-01-29 2011-10-06 2012-06-12 2013-02-17 2013-10-25 2014-07-02 2015-03-09 2015-11-14
date
HOUSEHOLD_1_311_WI_1_validation
0 2 4
2011-01-29 2011-10-06 2012-06-12 2013-02-17 2013-10-25 2014-07-02 2015-03-09 2015-11-14
date
FOODS_3_805_CA_3_validation
0 20
2011-01-29 2011-10-06 2012-06-12 2013-02-17 2013-10-25 2014-07-02 2015-03-09 2015-11-14
date
HOUSEHOLD_2_369_TX_2_validation
0 2 4 6
2011-01-29 2011-10-06 2012-06-12 2013-02-17 2013-10-25 2014-07-02 2015-03-09 2015-11-14
date
HOUSEHOLD_1_080_TX_2_validation
0.0 2.5 5.0 7.5
2011-01-29 2011-10-06 2012-06-12 2013-02-17 2013-10-25 2014-07-02 2015-03-09 2015-11-14
date
HOUSEHOLD_1_463_CA_2_validation
0 2 4
2011-01-29 2011-10-06 2012-06-12 2013-02-17 2013-10-25 2014-07-02 2015-03-09 2015-11-14
date
FOODS_2_046_CA_1_validation
0 2 4
2011-01-29 2011-10-06 2012-06-12 2013-02-17 2013-10-25 2014-07-02 2015-03-09 2015-11-14
date
HOUSEHOLD_2_102_WI_3_validation
0 2 4 6
2011-01-29 2011-10-06 2012-06-12 2013-02-17 2013-10-25 2014-07-02 2015-03-09 2015-11-14
date
FOODS_2_131_TX_2_validation
0 5 10
2011-01-29 2011-10-06 2012-06-12 2013-02-17 2013-10-25 2014-07-02 2015-03-09 2015-11-14
date
HOUSEHOLD_2_115_WI_1_validation
0 2 4 6
2011-01-29 2011-10-06 2012-06-12 2013-02-17 2013-10-25 2014-07-02 2015-03-09 2015-11-14
date

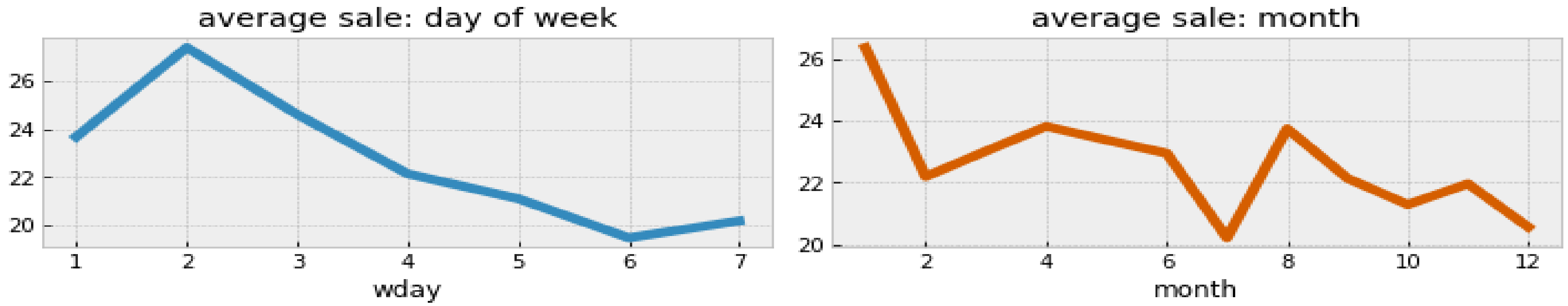
The below graphs give us a understanding of sales over different time components :

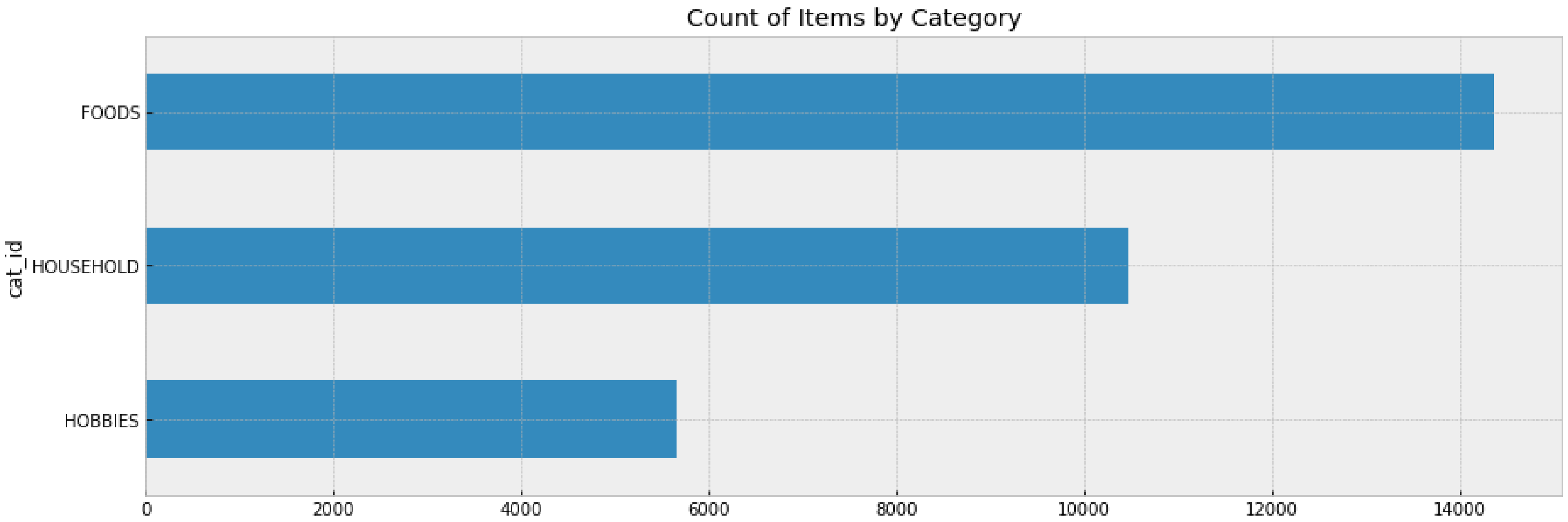
1) Week
2) Month

Food Item

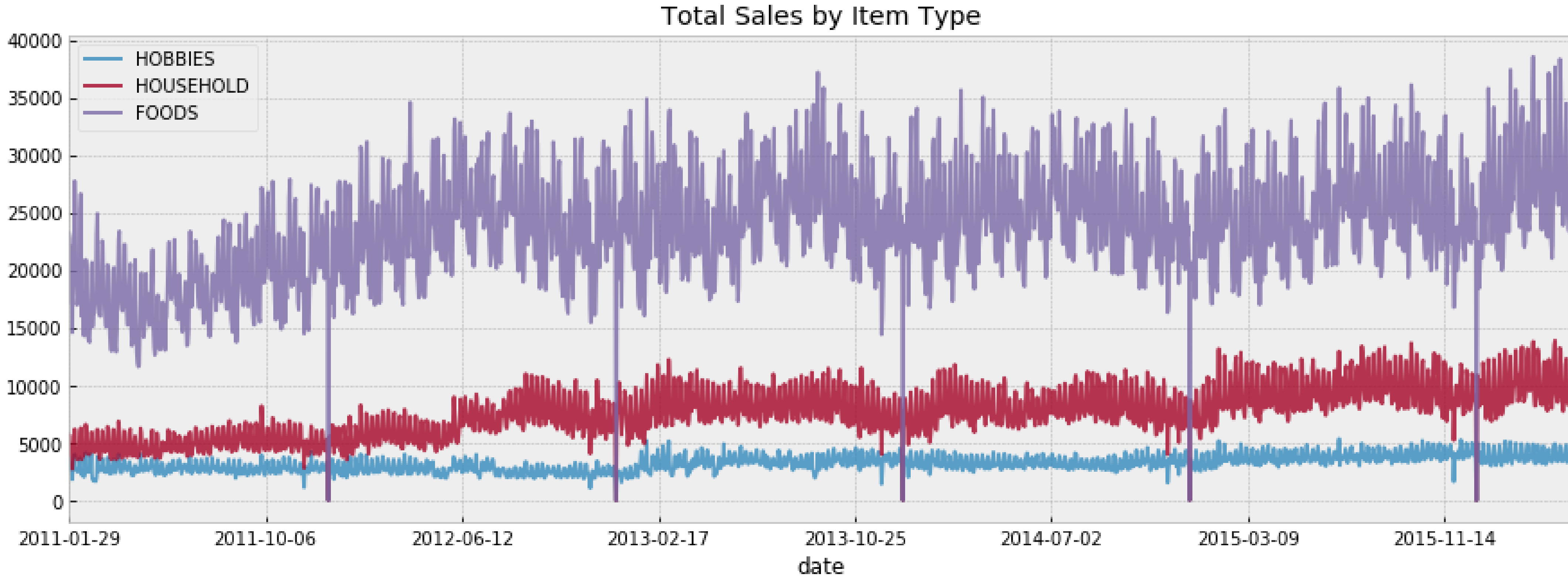


Household Item



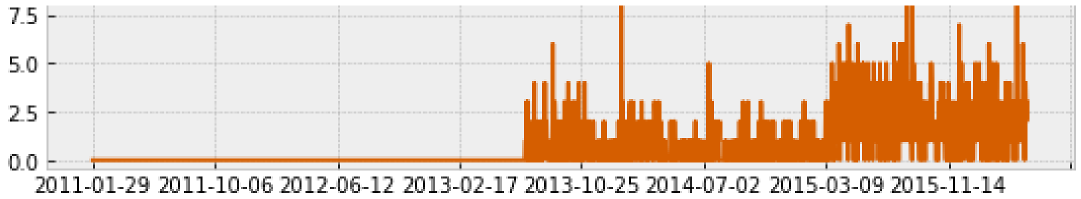


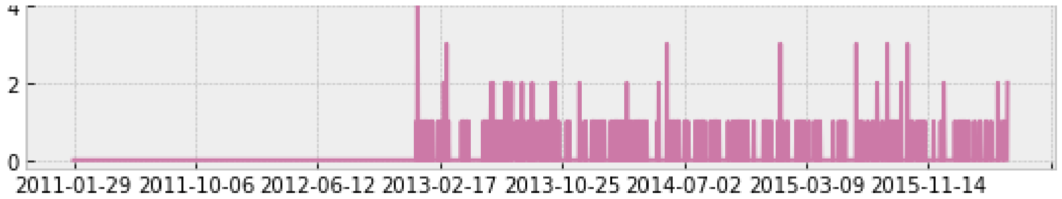
From the graph we see the number of unique products in each category



Data Preprocessing

- Backfill Sales for out of stock products
- Downcasting
- Remove the first 350 days due to zero inflated data





- Add features : One day before event, SNAP
- Creating training, validation, evaluation dataset
- Standardize features through scaling

PennState

CNN-LSTM Methodology

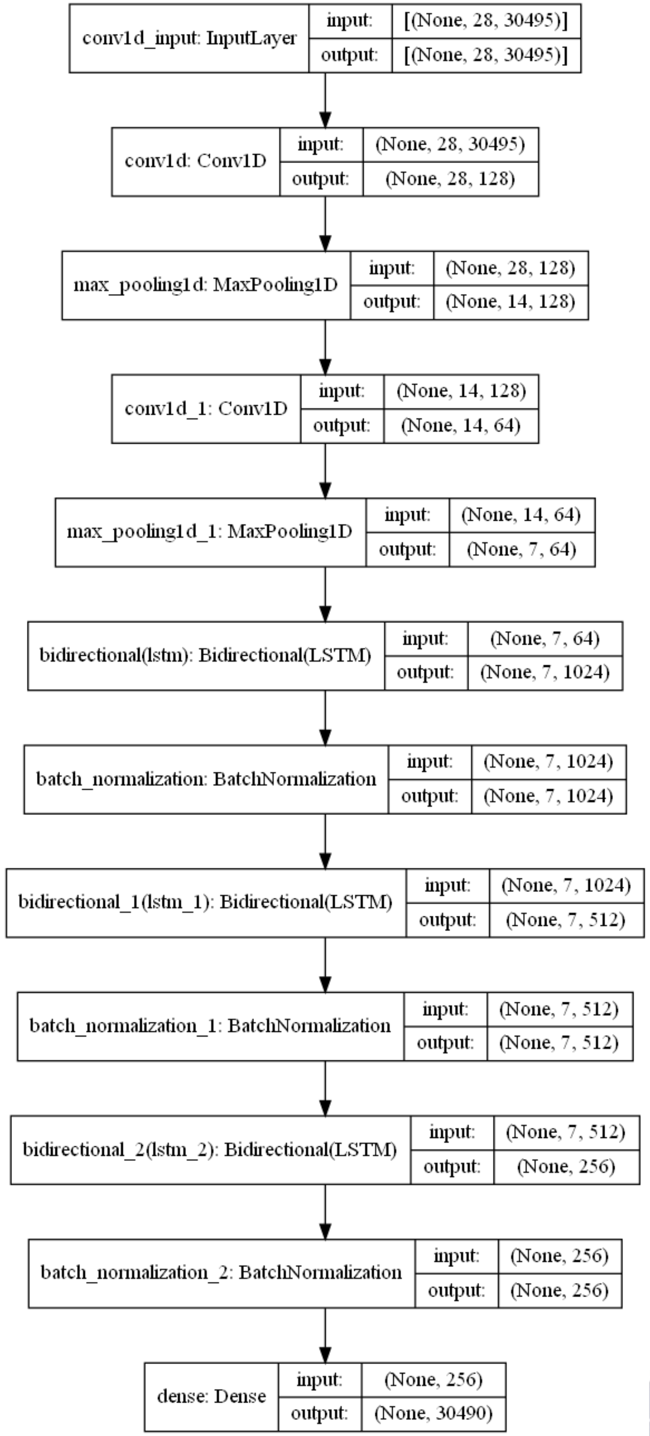
- CNN models is used to filter out the noise of the input data and extract more valuable features.
- LSTM models efficiently capture sequence pattern information for long-range prediction.
- We are adding Conv1D as our convolutional layer since we are dealing with a one-dimensional sequence.
- A filter is applied to the sequence to highlight certain features which are important.
- The pooling layer applies max function to the sections of the sequence. Pooling drastically reduces the size going into the next layer.

CNN-LSTM Methodology

- In short, the gated cell architecture keeps memory of important information uncovered earlier in the sequence of time steps.
- Bidirectionality will allow the LSTM to learn the input sequences both forward and backwards. In the backwards run, the bidirectional LSTM network preserves information from the future.
- Batch Norm is a normalization technique done on mini batches between the layers of a Neural Network. It helps to speed up training process and stabilizing the learning process

CNN-LSTM Summary

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_2 (Conv1D)            (None, 28, 128)           27323648
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 14, 128)           0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 14, 64)            57408
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 7, 64)             0
_________________________________________________________________
bidirectional_3 (Bidirection (None, 7, 1024)           2363392
_________________________________________________________________
batch_normalization_3 (Batch (None, 7, 1024)           4096
_________________________________________________________________
bidirectional_4 (Bidirection (None, 7, 512)            2623488
_________________________________________________________________
batch_normalization_4 (Batch (None, 7, 512)            2048
_________________________________________________________________
bidirectional_5 (Bidirection (None, 256)               656384
_________________________________________________________________
batch_normalization_5 (Batch (None, 256)               1024
_________________________________________________________________
dense_1 (Dense)              (None, 30490)             7835930
=================================================================
Total params: 40,867,418
Trainable params: 40,863,834
Non-trainable params: 3,584
_________________________________________________________________
```



Parameter calculation

- Conv1D Params = kernel size*input_depth*filters+bias

  =7*30495*128+128

  = 27323648

- LSTM params = 4*((sizeofinput + 1)*sizeofoutput+ sizeofoutput^2)

  =4*(65*512+512*512)

  = 1181696

- Since it is a bidirectional LSTM, total params= 2*1181686

  =2363392

- Batch normalization maintains the input shape

PennState

DA-RNN Methodology

DARNN

- One problem with encoder-decoder LSTM networks is that their performance will deteriorate rapidly as the length of input sequence is large.
- The attention-based encoder-decoder network uses an attention mechanism to select parts of hidden states i.e It takes all window size sales into account then assigning relative importance to each one of them.
- There are three parameters in the DA-RNN, i.e., the number of time steps in the window T, the size of hidden states for the encoder m, and the size of hidden states for the decoder p. We took time steps as 28, and did grid search over encoder and decoder layers to find 64 encoder/decoder size gave the best performance over val data.

PennState

Model Hyperparameter Comparisons

	CNN_LSTM	DA-RNN
Optimizer	Adam	Adam
Loss Function	MSE	MSE
Metric	MAE, MSE	MAE, MSE
Batch Size	64	16
Time per epoch	65 sec	17 sec
Total epochs	56	13
Total Time	60 min	3.6 min

PennState

Model Evaluation

- MAE:

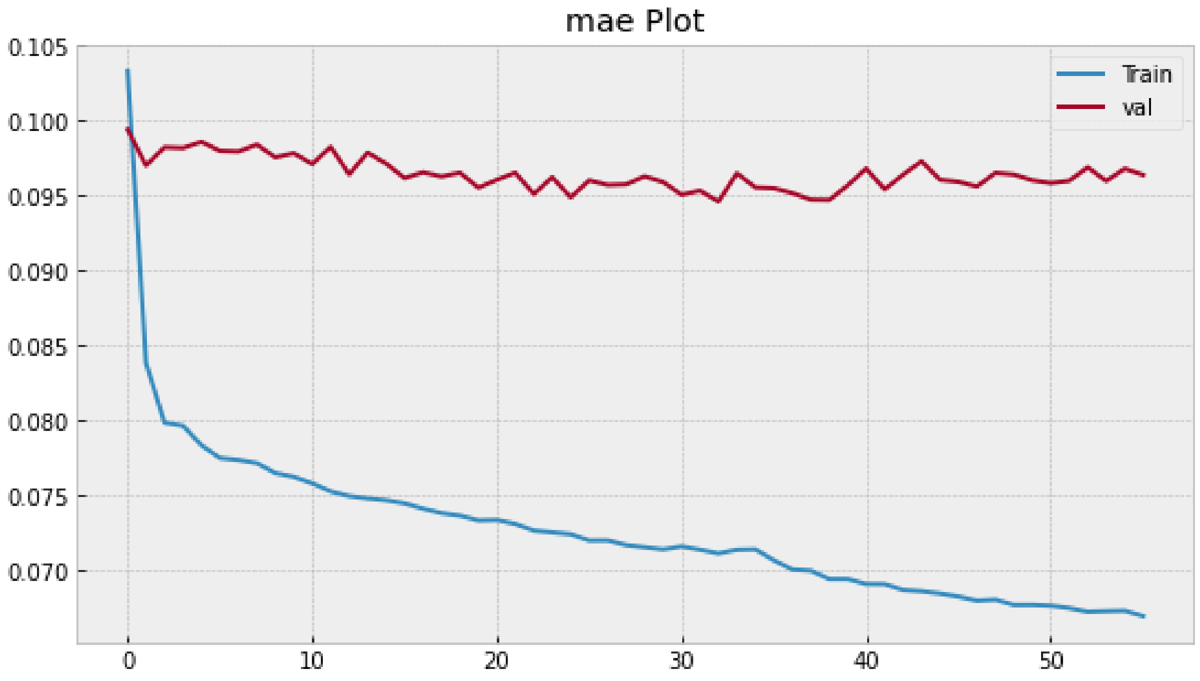
$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

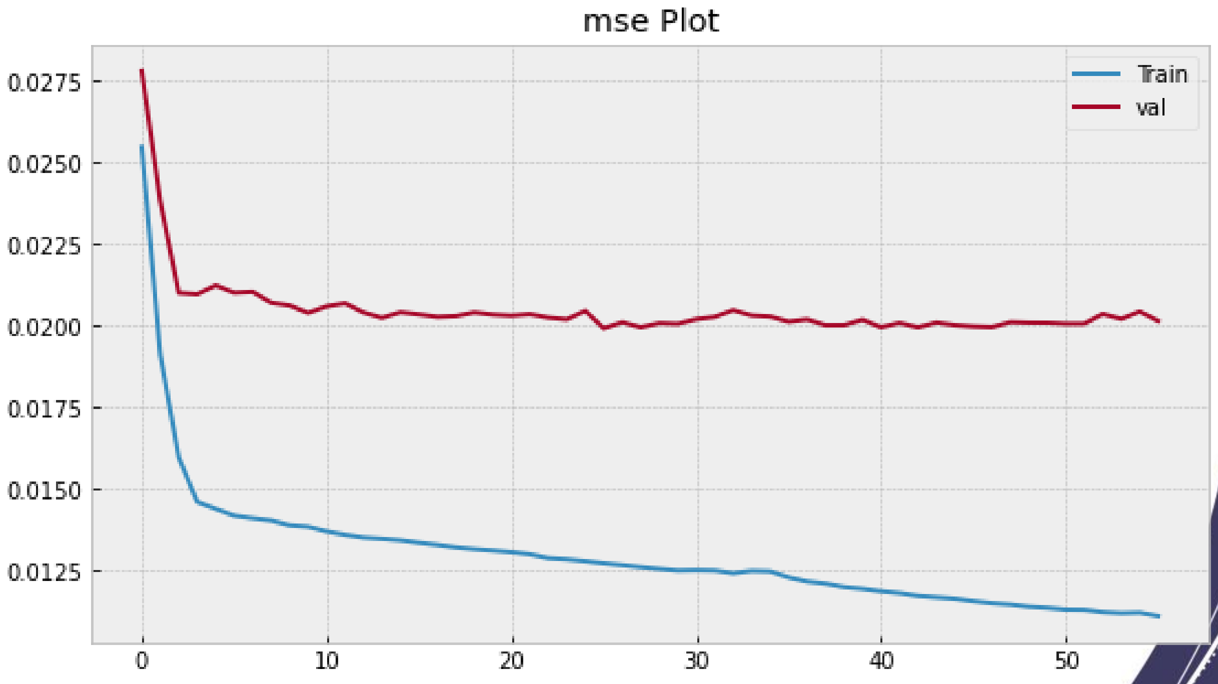
- MSE:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

	CNN-LSTM	DA-RNN
MAE	1.44	1.43
MSE	16.83	16.08

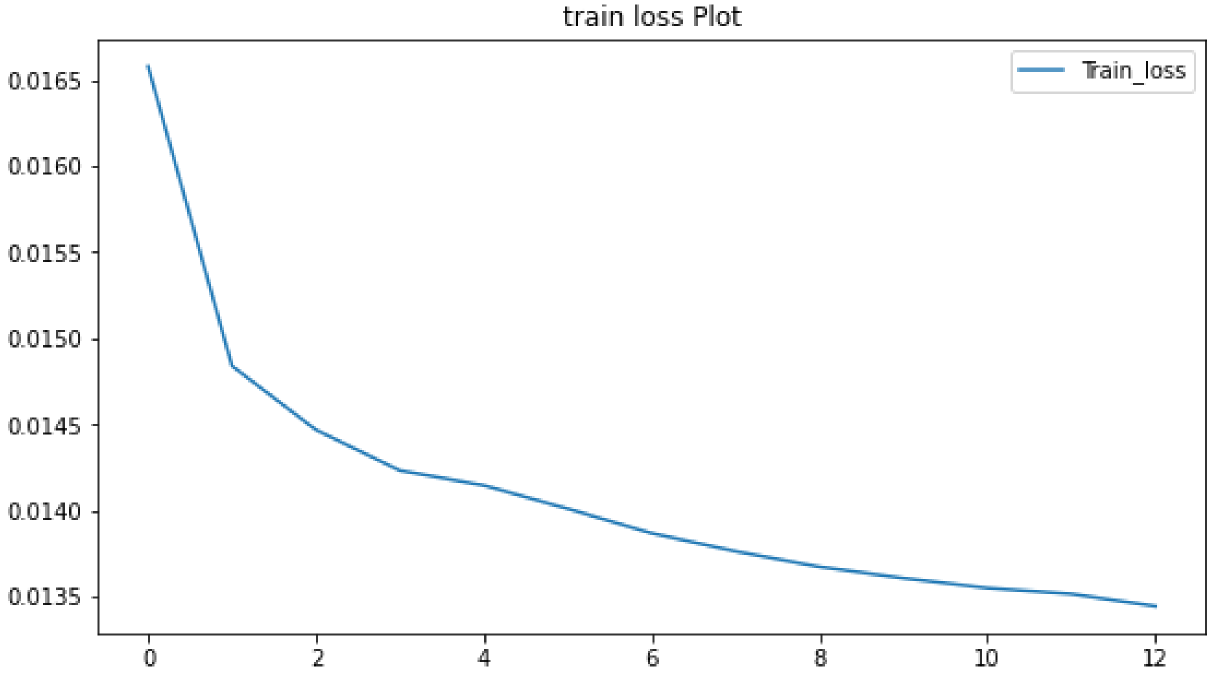
Training Validation Loss Plots CNN-LSTM Model

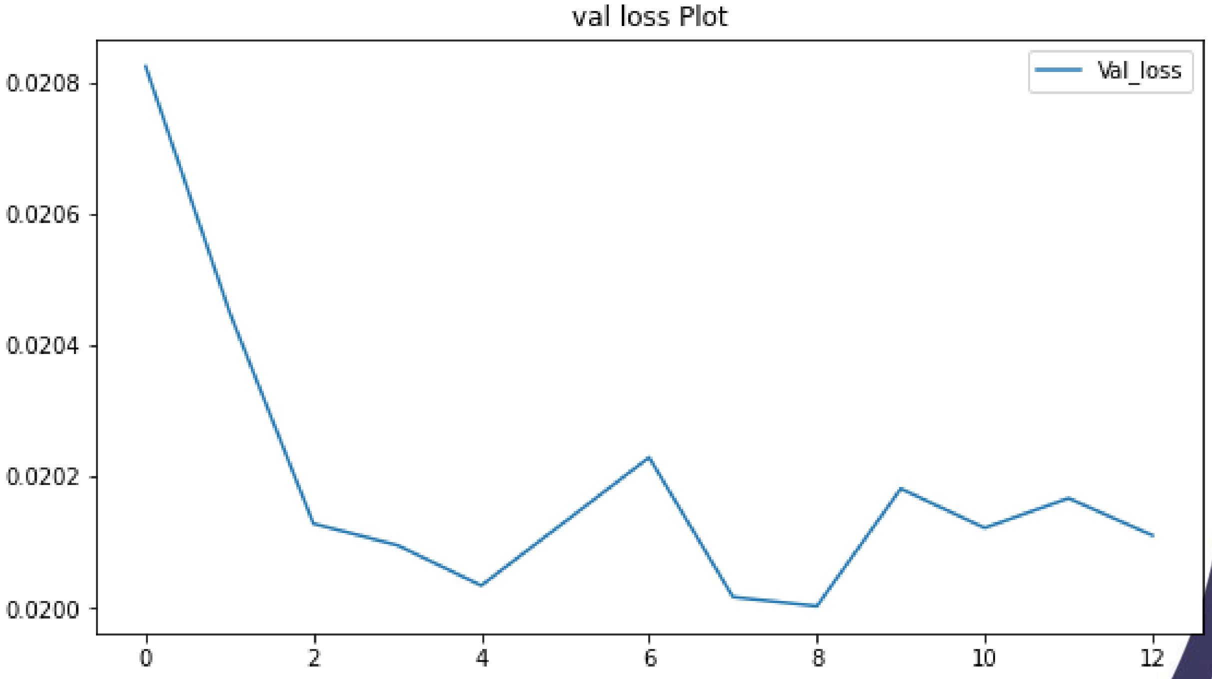




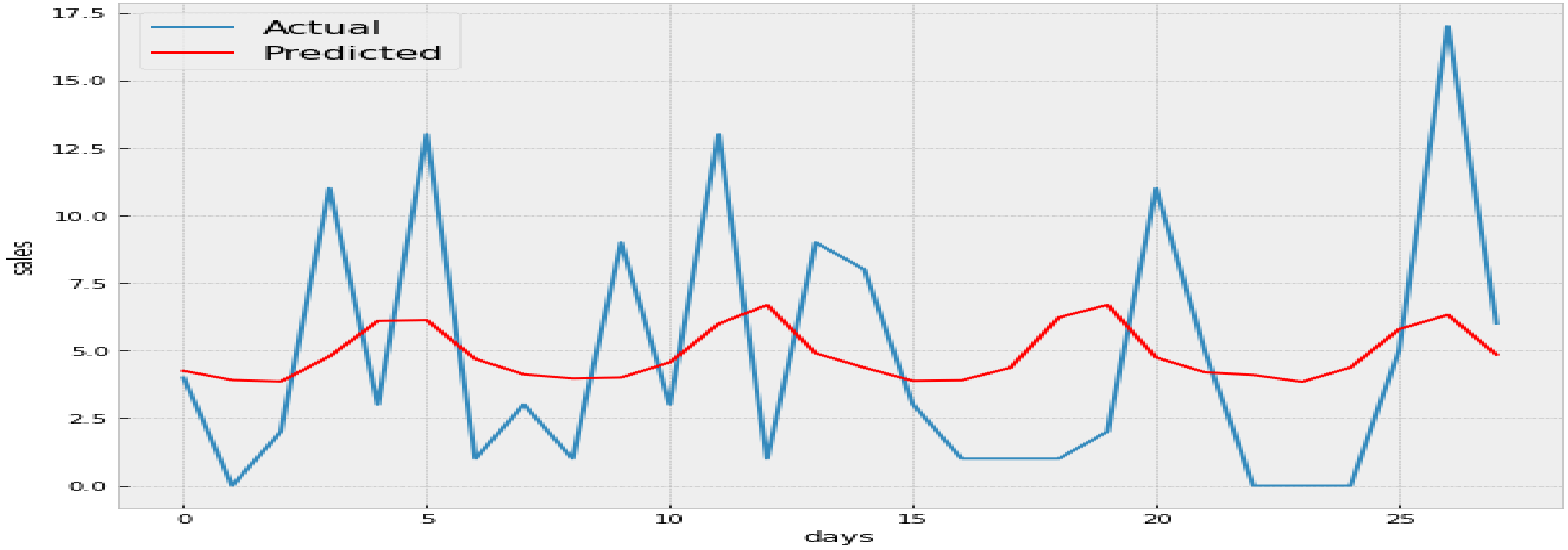
PennState

Training Validation Loss Plots DA-RNN Model

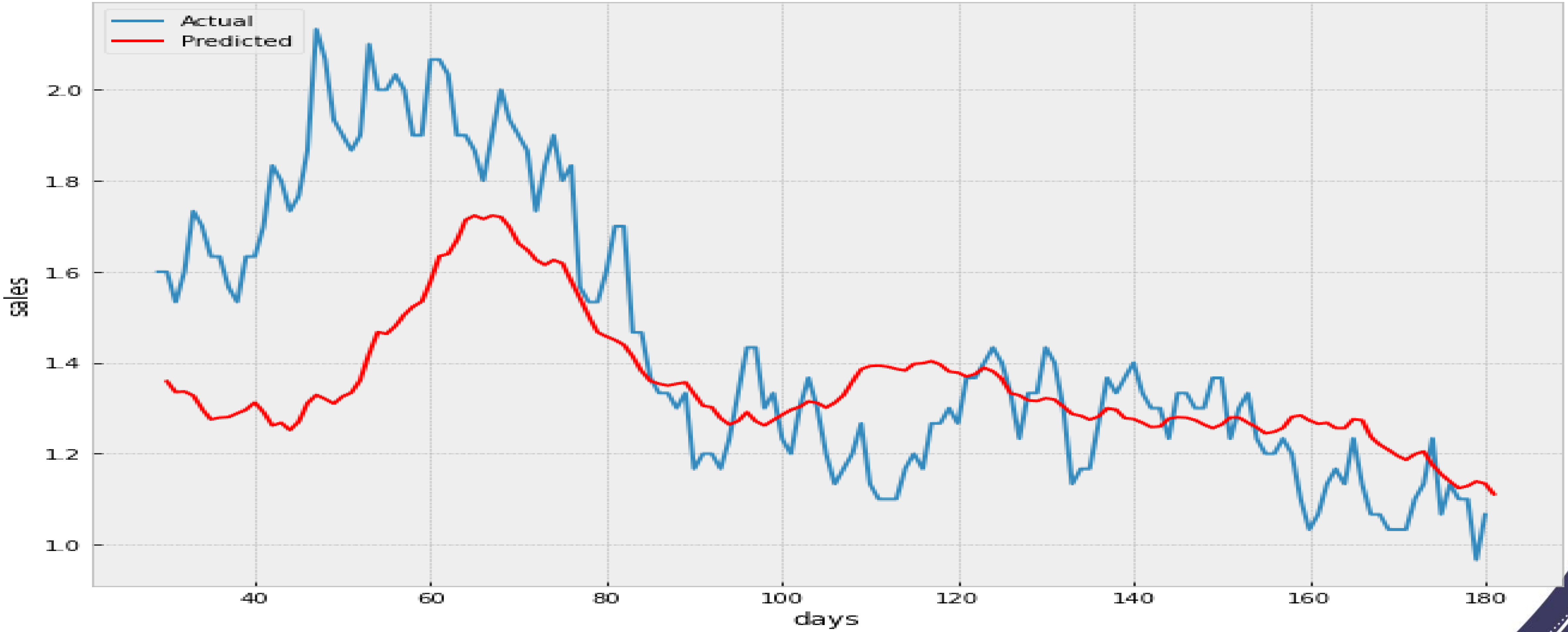




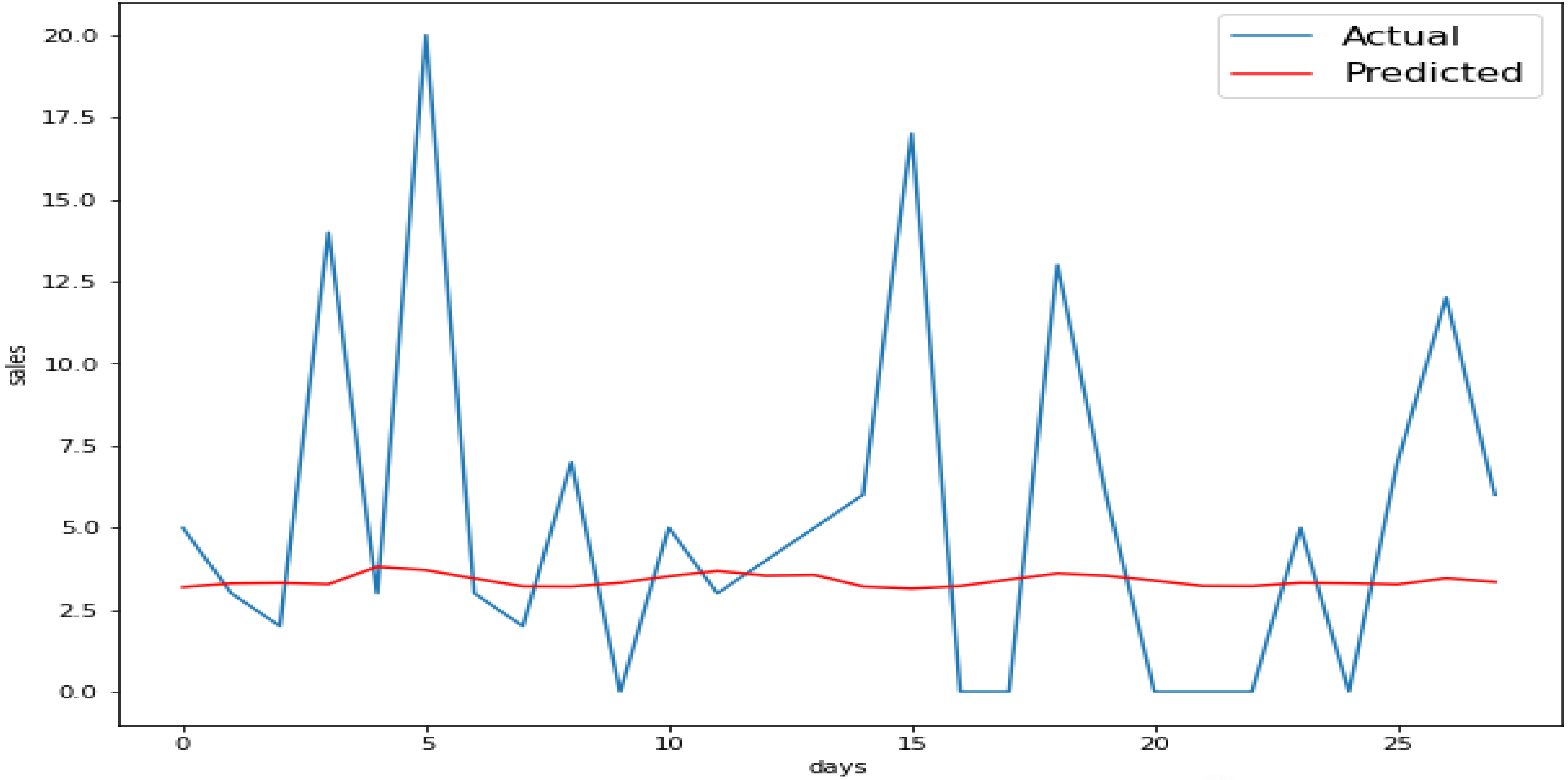
CNN-LSTM Outcomes (28days)



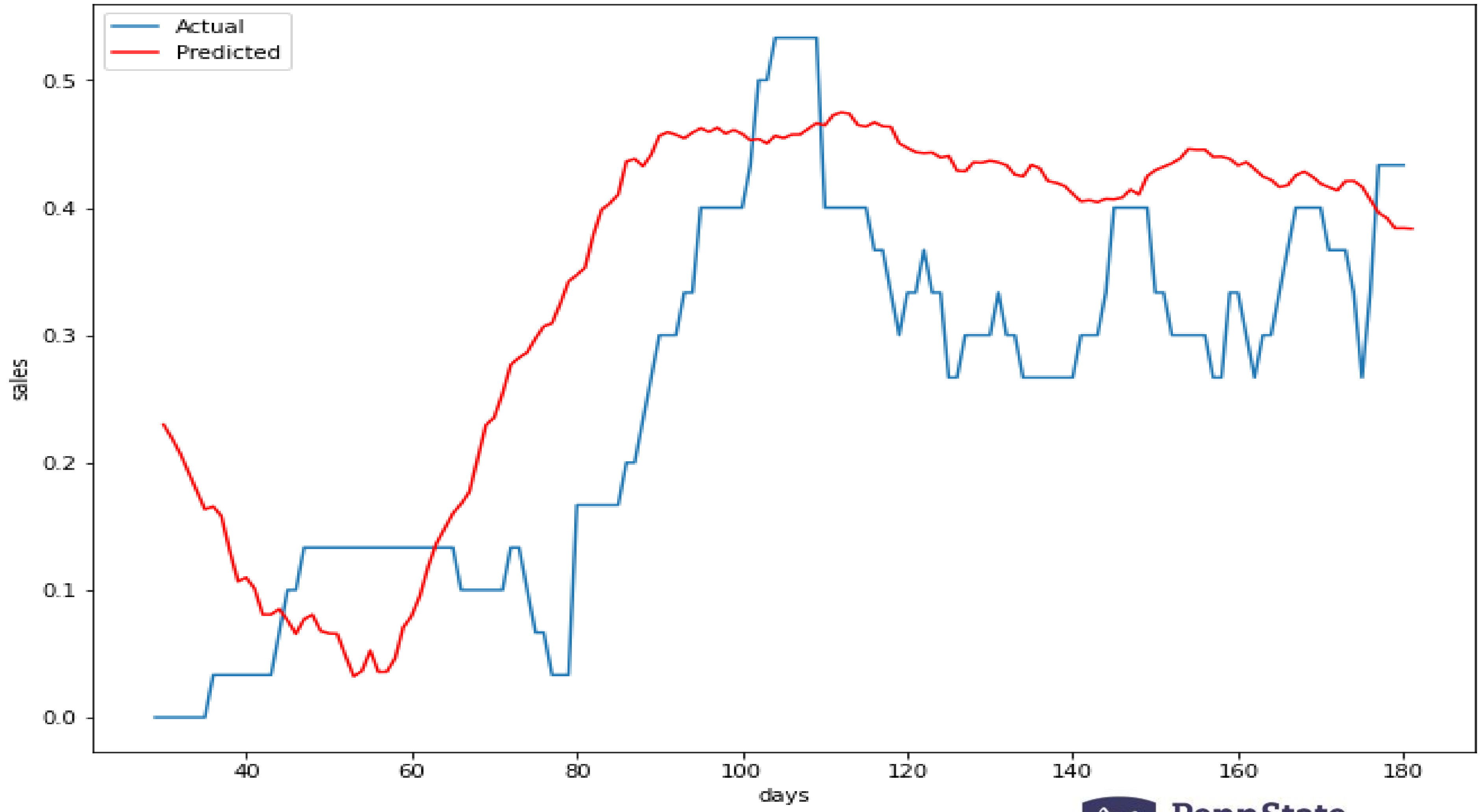
CNN-LSTM Outcomes(180 days)



DA-RNN Outcomes (28days)



DA-RNN Outcomes(180 days)



Lessons learned and Perspectives

Future work and improvements:

- Add more features for improving predictions. Through feature engineering we could embed season, quarter, month start/end, day of the week and week of the month.
- Moving standard deviation could also be explored.

PennState

Bibliographical References

- Livieris, Ioannis E., Emmanuel Pintelas, and Panagiotis Pintelas. "A CNN–LSTM model for gold price time-series forecasting." *Neural computing and applications* 32.23 (2020): 17351-17360.
- Lu, Wenjie, et al. "A CNN-LSTM-based model to forecast stock prices." *Complexity* 2020 (2020).
- Qin, Yao, et al. "A dual-stage attention-based recurrent neural network for time series prediction." *arXiv preprint arXiv:1704.02971* (2017).
- Ensafi, Yasaman, et al. "Time-series forecasting of seasonal items sales using machine learning–A comparative analysis." *International Journal of Information Management Data Insights* 2.1 (2022): 100058.

PennState

PennState