NVIDIA Next? Predicting Peaks and Spotting Sneaks in Stock Prices

Avinash Chaluvadi – CS5390

Motivation: Nvidia's stock has experienced remarkable growth over the past few years, driven by the increasing demand for its semiconductor chips, particularly those designed for AI applications. The objective of this project is to harness the capabilities of deep learning to predict Nvidia's stock prices, a task that is both complex and dynamic due to the volatile nature of financial markets.

Figure 1 illustrates the movement of Nvidia's stock price for today.

Dataset: For the experimental study, I downloaded a live dataset from the Yahoo Finance website. The input features include several key fields: Open, High, Low, Close, Adj Close, and Volume.

	Open	High	Low	Close	Adj Close	Volume	day_name	day_of_week
2012-05-10	3.1575	3.1625	3.0550	3.1050	2.848224	59709200	Thursday	3
2012-05-11	3.3625	3.4200	3.2900	3.3025	3.029391	143514000	Friday	4

Comprehensive Analysis of Nvidia Stock Performance (2012-2024)

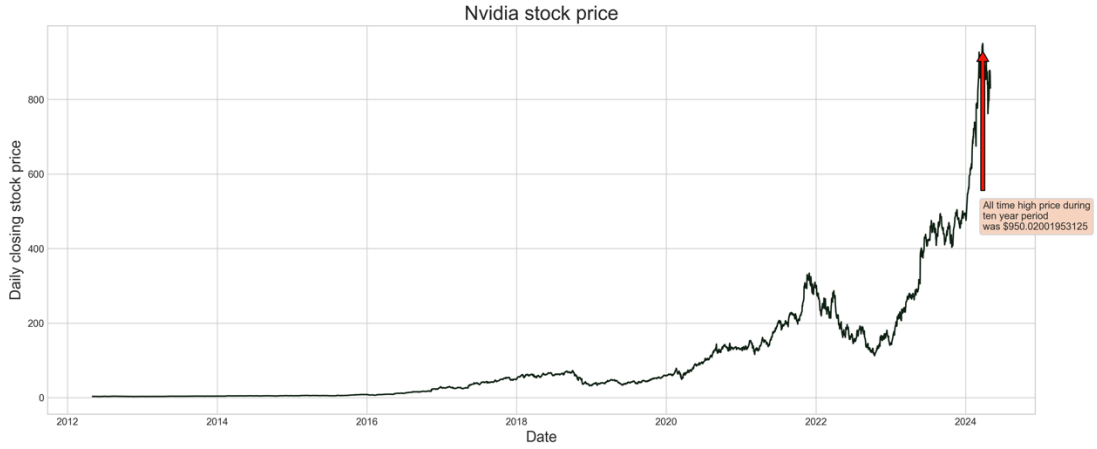


Figure 2 illustrates the daily closing stock prices of Nvidia over a decade, culminating a record high of $950 in 2024.

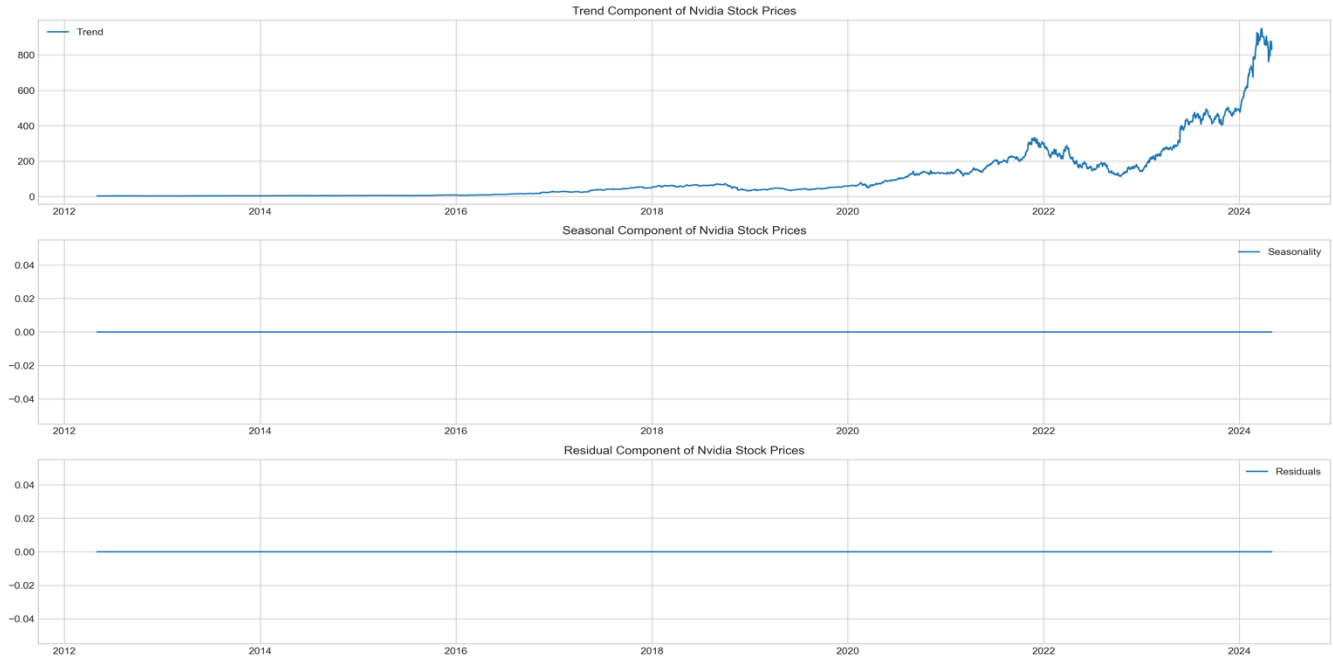


Figure 3: highlights a substantial upward movement post-2020, whereas the seasonal and residual subplots show relatively stable values across the observed period.

Data Processing: Data pre-processing is crucial prior to data modeling, particularly to manage the non-uniform distribution of data. For this project, the Min-Max Scaler was used.

Overview of Data Post-Normalization

Date	Close	Day of Week
2012-05-02	0.000388	0.500000
2012-05-03	0.000330	0.750000
2012-05-04	0.000232	1.000000
2012-05-07	0.000288	0.000000
2012-05-08	0.000285	0.250000

Modelling

An LSTM-based neural network architecture was employed to build the model, aimed at Nvidia stock price prediction. Long Short-Term Memory (LSTM) architectures were fitted to the data.

Overfitting

LSTM, Dense Layers(LSTM,Dense Units)	Loss	MAE	Windowed Data
1 LSTM(128)	6.3571e-05	0.0043	3 months
1 LSTM(128)	6.9634e-05	0.0045	6 months
2 LSTM(128-64), 1 Dense(8)	9.9356e-05	0.0059	3 months
2 LSTM(128-64), 1 Dense(8)	9.7770e-05	0.0057	6 months
3 LSTM(128-64-64), 2 Dense(8-1)	6.8383e-05	0.0044	3 months
3 LSTM(128-64-64), 2 Dense(8-1)	5.8841e-05	0.0043	6 months
4 LSTM(8-16-16-32), 3 Dense(32-16-16)	1.1268e-04	0.0067	3 months
4 LSTM(8-16-16-32), 3 Dense(32-16-16)	1.4419e-04	0.0068	6 months
5 LSTM(8-16-16-32-64), 2 Dense(8-8)	1.4442e-04	0.0077	3 months
5 LSTM(8-16-16-32-64), 2 Dense(8-8)	1.2227e-04	0.0071	6 months

Table 1: Model Performance for Different LSTM and Dense Layer Architectures

As shown in the table above, the basic architectures featuring 1, 2, or 3 LSTM layers, along with 1 or 2 dense layers, perform relatively similarly. I chose model configuration comprising of 3 LSTM layers and 2 dense layers.

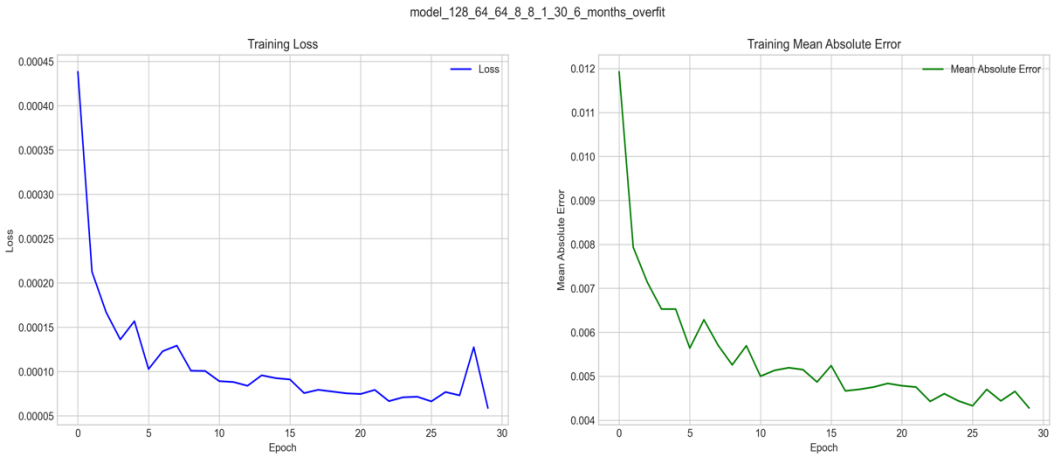


Figure 4: Curve showing change in over-fitting MSE and MAE vs epoch.

Model Evaluation

In this step, I evaluated all the models established during the overfitting phase, exploring various combinations of LSTM and dense layers, and adjusting the number of neurons and layers to determine the optimal model configuration.

Model Configuration	Window Period	Validation MAE	Validation MAPE
1 LSTM(128)	3 months	6.152192	0.037394
2 LSTM(128-64), 1 Dense(8)	3 months	5.962914	0.035794
3 LSTM(128-64-64), 2 Dense(8-1)	3 months	6.445919	0.038820
4 LSTM(8-16-16-32), 3 Dense(32-16-16)	3 months	9.583087	0.059262
5 LSTM(8-16-16-32-64), 2 Dense(8-8)	3 months	11.107743	0.070483

Table 2: Model Performance of Architectures: Validation with 3-Month Windowed Data.

Model Configuration	Window Period	Validation MAE	Validation MAPE
1 LSTM(128)	6 months	5.885041	0.039111
2 LSTM(128-64), 1 Dense(8)	6 months	5.801102	0.037494
3 LSTM(128-64-64), 2 Dense(8-1)	6 months	5.871108	0.038194
4 LSTM(8-16-16-32), 3 Dense(32-16-16)	6 months	10.011879	0.067544
5 LSTM(8-16-16-32-64), 2 Dense(8-8)	6 months	8.775115	0.059131

Table 3: Model Performance of Architectures: Validation with 6-Month Windowed Data.

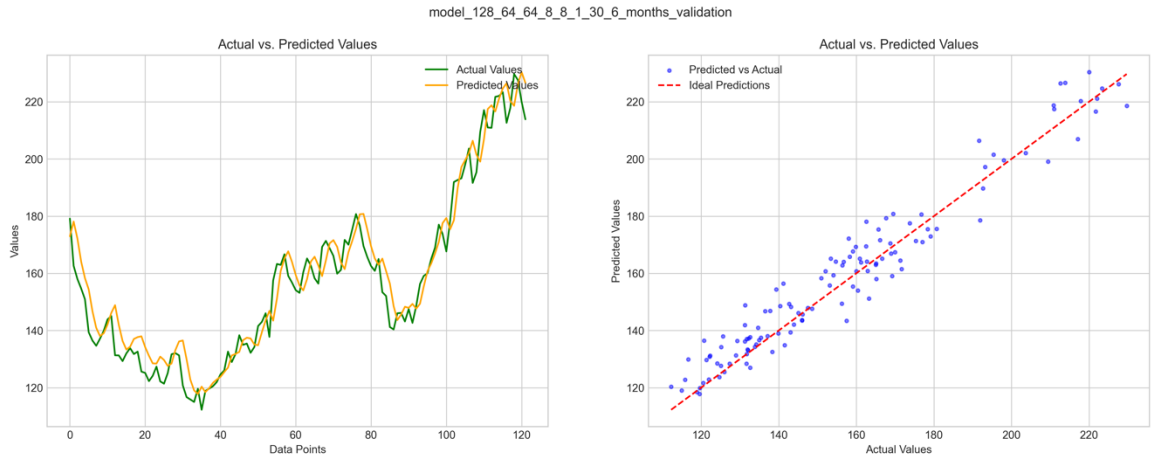


Figure 5: Comparison of Actual vs. Predicted Values and Scatter Plot of Predicted vs. Actual Values

Data Augmentation: In this step, I explored three different techniques to improve validation accuracy.

- Adding Jitter:
- Scaling Technique
- Magnitude Warping

Augmentation Technique	Period	Validation MAE	Validation MAPE	Test MAE	Test MAPE
Jitter 0.01	6 months	7.872104	0.050886	38.037239	0.052338
Jitter 0.03	6 months	12.814412	0.082905	92.741540	0.127254
Jitter 0.005	6 months	6.802560	0.044311	19.602779	0.029615
Scaling 0.1	6 months	6.249270	0.040435	99.480830	0.148395
Scaling 0.3	6 months	7.824560	0.049986	258.880185	0.359946
Scaling 0.01	6 months	6.142313	0.039434	18.831326	0.027169
Warping Sigma 0.2, Knot 1	6 months	10.648151	0.070893	203.769306	0.281648
Warping Sigma 0.2, Knot 4	6 months	10.371709	0.069998	204.302959	0.280349

Table 4: Model Configuration Results after Data Augmentation Technique

Observation: From the results, it's clear that the LSTM model trained with a scaling constant of 0.01 and a 6-month window performed better.

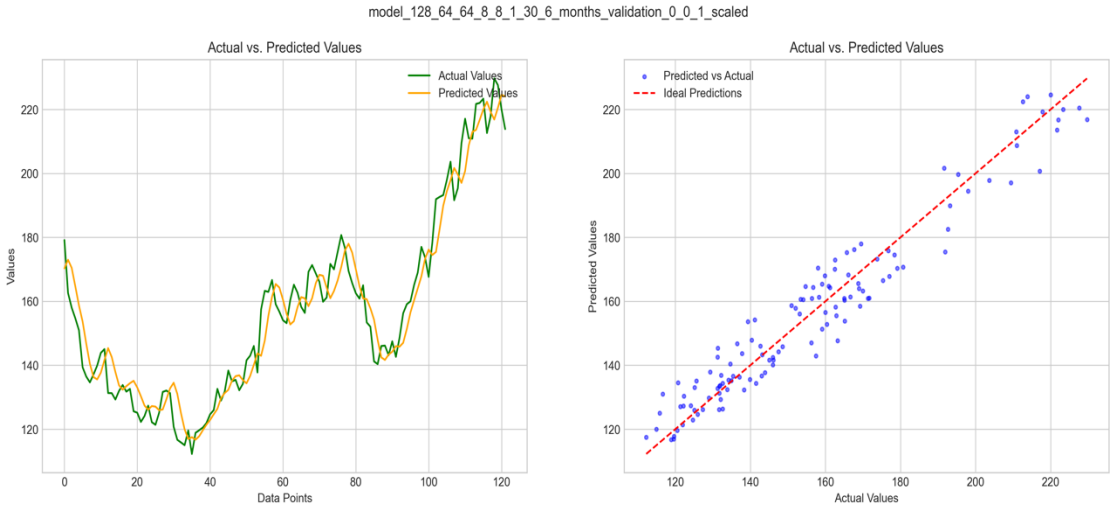


Figure 6: Comparison of Actual vs. Predicted Values and Scatter Plot of Predicted vs. Actual Values

Effects of Regularization: In this step, I explored two different methods to improve validation accuracy: Batch Normalization and Dropout.

Layer Configuration	Period	Validation MAE	Validation MAPE	Test MAE	Test MAPE
1 Batch Normalization layer	6 months	6.799067	0.044139	247.337539	0.331576
2 Batch Normalization layers	6 months	8.567218	0.055651	168.914832	0.233823
1 Dropout layer	6 months	6.603170	0.042127	21.736594	0.030256
2 Dropout layers	6 months	7.053871	0.045771	19.266117	0.027392

Table 5: Results with Batch Normalization and Dropout Layer.

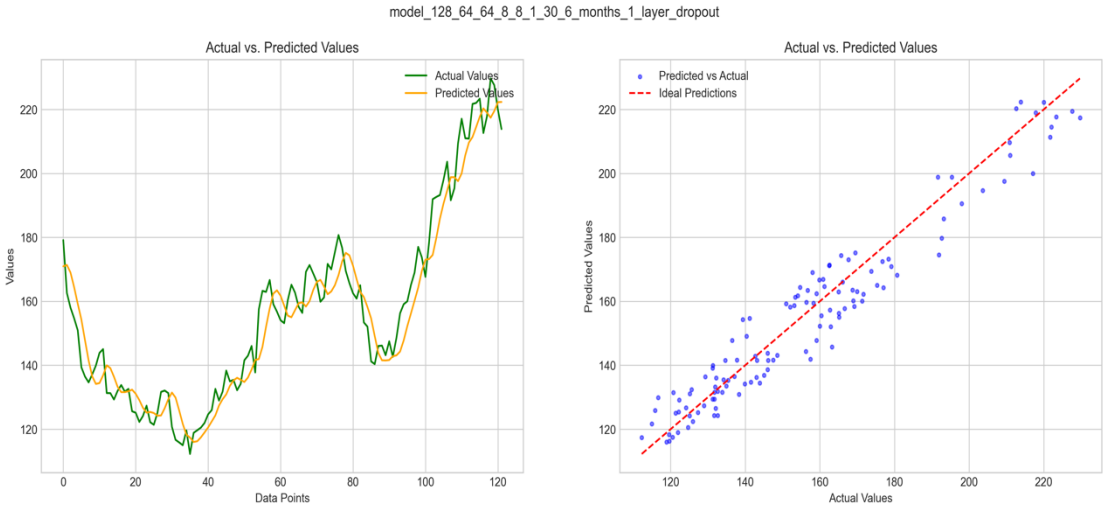


Figure7: Comparison of Actual vs. Predicted Values and Scatter Plot of Predicted vs. Actual Values

Exploring Powerful Architectures and PreTrained Models

In this step, I explored two different approaches: first, a dense bi-directional LSTM, and second, using a pre-trained model called Prophet.

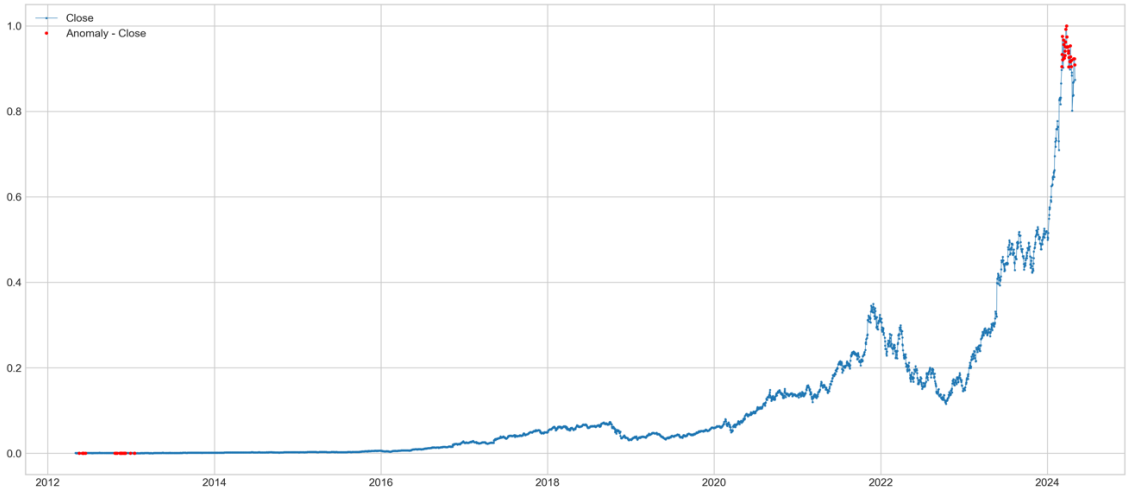
Model Configuration	Period	Validation MAE	Validation MAPE	Test MAE	Test MAPE
6 Bi-directional LSTM, 2 Dense	6 months	9.702311	0.064795	33.59239	0.046592
Prophet,	6 months			284.29	0.51141

Table6: Model Configuration with dense network and Pre-trained model(Prophet)

Observation: From the table above, it's clear that neither the pre-trained model nor the dense LSTM architecture resulted in a reduction in loss or Mean Absolute Error (MAE).

Anomaly Detection

To detect anomalies in the input, I used a quantile approach; specifically, I set the lowest quantile to 0.01 and the highest quantile to 0.99.



To detect anomalies in the output, I calculated the loss for all test data points. If the loss exceeds the minimum loss obtained during the overfitting phase, then that output is considered an anomaly.

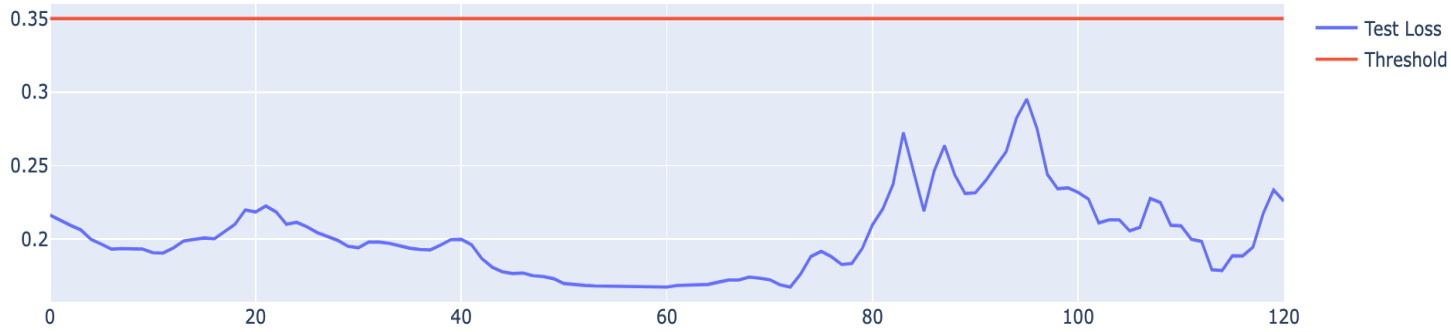


Figure 8: Output Anomaly Detection Plot

Predictions from the Optimal Model

Model Configuration	period	Test MAE	Test MAPE
model_128_64_64_8_8_1_30_6_months_validation	6 months	19.209361	0.028827

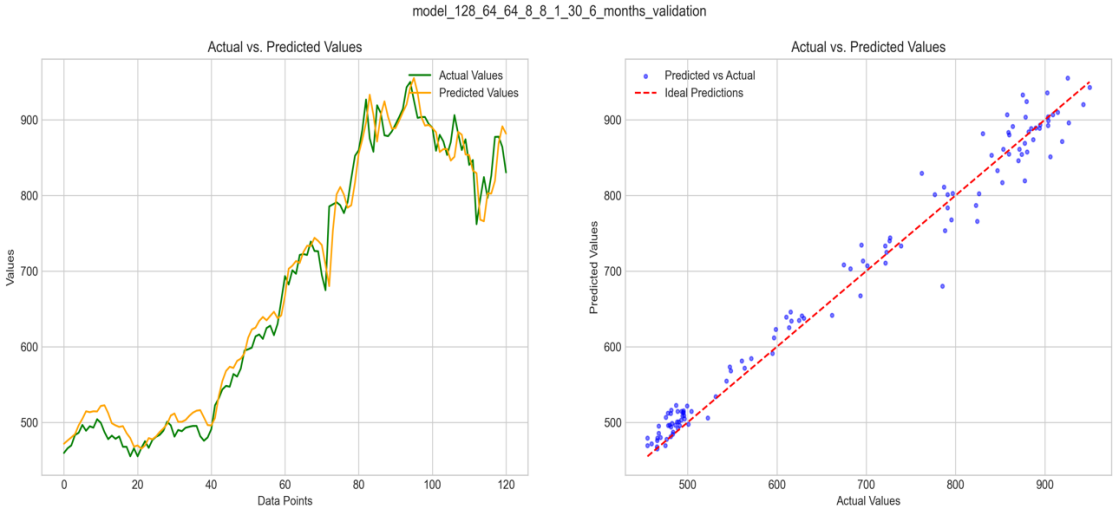


Figure9: Comparison of Actual vs. Predicted Values and Scatter Plot of Predicted vs. Actual Values

Conclusion

The findings reveal that the proposed methods effectively capture the interrelations within the data. After evaluating various model architectures with different configurations of LSTM and dense layers, the highest validation accuracy was achieved with a model containing three LSTM layers (128-64-64) and three dense layers (8-8-1). Despite experimenting with data augmentation techniques and incorporating Batch Normalization and Dropout layers, as well as more complex architectures and pre-trained models like Facebook Prophet, no better validation results were achieved.