Topic: Forecasting Stock Volatility

1. Abstract

This report presents a comprehensive approach for forecasting stock volatility using 7 input features, with the target variable being volatility. The aim of this report is to address the questions posed in the Assessment and compare the results of three NASDAQ-listed stocks. The model has shown robust generalization performance for [INTC], an older listed stock with more training data. However, the convergence quality of predictions for the other two stocks, TSLA and MSFT, is comparatively lower.

2. Introduction

Forecasting stock volatility is a challenging task, and several studies have been conducted in this area. The current understanding and background information about the topic suggest that predicting stock volatility is critical for investors to make informed decisions about their investment portfolios.

The purpose of this project is to forecast stock volatility using 7 input features. To achieve this objective, several techniques and models have been employed. The report provides a detailed explanation of the model architecture and the steps involved in implementing the LSTM model consisting of multiple layers with regularization, activation, and other means to reduce underfitting/overfitting. The report also explains the determination of optimal look back and horizon parameters through repeated iterations of the primary model for different [Look-back, Horizon] pairs.

The techniques implemented also include hyperparameter tuning using GridSearchCV and time-series split cross-validation. Walk-forward optimization is achieved by walk-forward rolling time series CV (also called Anchored - Walk). The LSTM autoencoders incorporate dimensionality reduction with a different number of compressed features and a different lookback window from the originally used lookback. Additionally, XGBoost can be used to combine the predictions from multiple LSTM models and improve overall performance.

XGBoost is a gradient boosting algorithm that can handle complex non-linear relationships and capture interactions between variables.

The potential outcomes of this project include accurate and reliable forecasts of stock volatility, which can be used by investors to make informed decisions. The report is organized into different sections, including the data preparation, model development, hyperparameter tuning, model evaluation, and conclusion, to provide a detailed description of the methodology used and the results obtained.

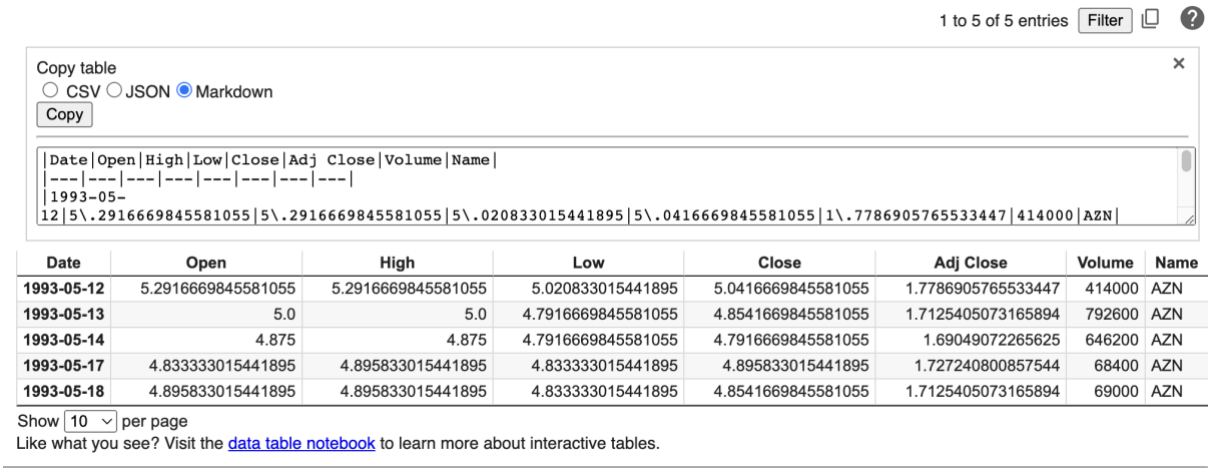
Constraints: Interpretability has not been implemented in the model.

3. Data

Data is downloaded from Yahoo Finance using yfinance library. The dataset contains top 50+ NASDAQ stocks and the user can input different stock symbols. With INTEL Corporation, being the oldest listed stock – the dataset has historical prices of all stocks since its inception in the markets.

3.1 Raw Structure:

The dataset has info on Open, High, Low, Close, Adj. Close, Volume, Stock Name as seen in the below figure.



Date	Open	High	Low	Close	Adj Close	Volume	Name
1993-05-12	5.2916669845581055	5.2916669845581055	5.020833015441895	5.0416669845581055	1.7786905765533447	414000	AZN
1993-05-13	5.0	5.0	4.7916669845581055	4.8541669845581055	1.7125405073165894	792600	AZN
1993-05-14	4.875	4.875	4.7916669845581055	4.7916669845581055	1.69049072265625	646200	AZN
1993-05-17	4.833333015441895	4.895833015441895	4.833333015441895	4.895833015441895	1.727240800857544	68400	AZN
1993-05-18	4.895833015441895	4.895833015441895	4.833333015441895	4.8541669845581055	1.7125405073165894	69000	AZN

Fig 3.1: Raw Data

Source: Colab Notebook of Author

3.2 Feature Engineering and Selection:

The 30-day arithmetic return (r_t) of the stock represents the average daily return of the stock over the past 30 trading days. The volatility of the stock is determined by calculating the standard deviation of the 30-day arithmetic returns over a rolling 30-day period. The data frame is filtered to have the following columns as displayed in below code snippet and in Fig 3.2.1.

df = chosen_stock[['Open', 'High', 'Low', 'Close', 'Adj Close', '30 day arithmetic return', 'volatility']]

The features can be defined as a vector

$$\mathbf{X}_t = (O_t, H_t, L_t, Close_t, AdjClose_t, r_t, VOL_t)^T$$

This project is to estimate the function $f(\cdot)$, that takes a sequence of historical $\mathbf{X}_t$ as input and generates vector $\mathbf{VOL}_t = \left(\mathbf{VOL}_{t,1}, \dots, \mathbf{VOL}_{t,H}\right)^T$ as output:

$$\mathbf{VOL}_t = f(\mathbf{X}_t, \mathbf{X}_{t-1}, \mathbf{X}_{t-2}, \dots, \mathbf{X}_{t-W})$$

Where $\boldsymbol{W}$ is the look back window, $j = 1, \dots, H$.

Fig 3.2.1: Definition of Feature X_t and Volatility VOL_t

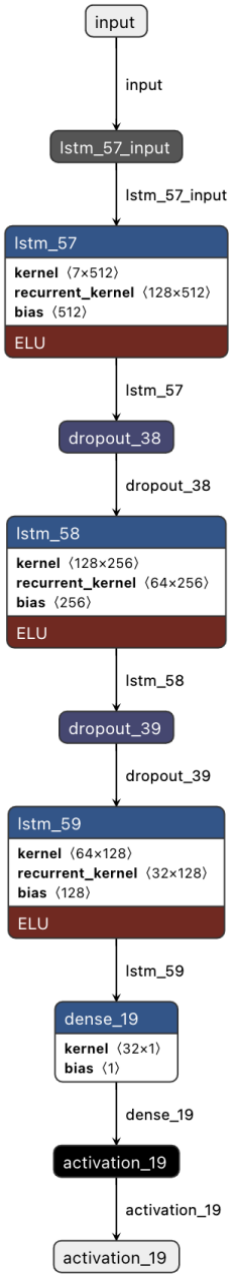
Source: Assignment Brief

Normalization: Standard Scaling (z-score) helps the model specifically time series data to converge faster and improves the training process. It is almost mandatory that we need to implement in any model.

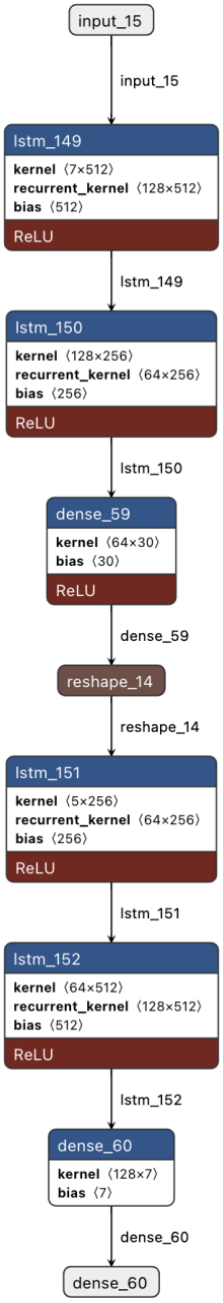
Train – Test split and Validation data: The trained data and test data are split by both ways – slicing and train_test_split function. Validation split argument ensures a percentage of train data is allocated to improve the model performance.

4. Model Architecture

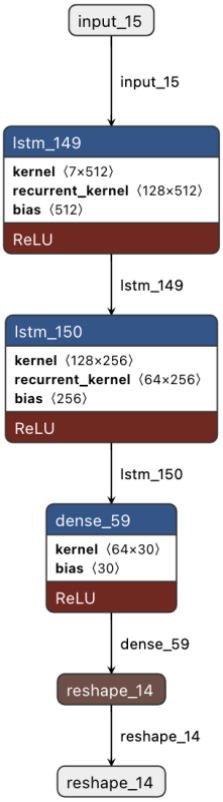
The model architecture appears like a “Hybrid Comparative Modelling” approach that implements all techniques and compares the performance of the resultant models with one another.



LSTM model



LSTM Autoencoder



Encoder

Fig 4.1: Model Architecture

Tool Used: https://netron.app/

LSTM Model: The first one is the primary model implemented at the beginning and this will serve as the basic block throughout the project. This is the representation of one of the calls made to the model from Walk – Forward Optimization section [discussed below].

Layers and Hyperparameters: The Exponential Linear Unit (ELU) was specifically chosen here since it gave better results than ReLu in implementing non-linearity in the system. ELU can better interpret negative values especially in case of stock prices where the prices and indicators can change swiftly. It also solves Bias shift issue by controlling the weights and distribution of inputs to the LSTM layers. In the case of stock volatility prediction, it is important to have a model that can capture the temporal dependencies in the time series data, and that can generalize well to new data. The early stopping criteria, dropout layers, used in this model can prevent overfitting and help to generalize the model better. Since the output needs to be a continuous value here, a linear activation output layer has been used instead of Sigmoid/SoftMax. Adam is the primary optimizer chosen in this model. Adam uses adaptive learning rates that change as the training progresses and by keeping a moving average of the past gradients. This allows the optimizer to quickly converge to a good solution and handle different types of data [1].

Hyperparameter tuning:

The parameters are tuned in a two-step process here. The optimal look-back, horizon pairs are first determined so that the comparative analysis use the same pair of values throughout the comparative model. In most cases, optimal pairs are found at look_back = 7 or 14. In most cases where horizon > 3, the prediction is so poor and has been overshot at peaks.

```
lb_range = [7, 14, 30, 60, 90]
horizon_range = range(1, 5)
```

Secondly with limited experimentation and research (Table 1) [2], it was found that it is better to tune the following parameters to improve the model performance and reduce underfitting/overfitting. All the parameters discussed in earlier section (Generalization/Overfitting) regulate fairly by themselves since multiple criteria is involved such as early stopping that monitors the values, stops the training, and restores weights from the best epoch.

```
n_neurons = [32, 64, 128]
dropout_rate = [0.05, 0.1, 0.2]
learning rate = [0.0001, 0.001, 0.01]
```

The GridSearchCV function is used to perform a grid search with 5-fold time series cross-validation to find the best hyperparameters for the LSTM model. A randomized search might not be helpful here

since it can miss some pair combinations and can significantly affect the predictions. An optimal level of neurons and regularization [dropout] prevents underfitting/overfitting.

Table 1: Learning rate - Impact on model performance

Too small	Training process may take too long to converge.
Too large	Training process may fail to converge or oscillate around the optimal point.
Optimal	Training process converges quickly to the optimal point.
Decay	Can help avoid overfitting by gradually reducing the learning rate during trainir

Input – Output Pairs Generator: This is one of the critical elements in the model where the function then creates input-output pairs by sliding a window of size look_back over the data and taking the next horizon time steps as the output. This process is repeated until the end of the data is reached. This is essential for sequential time series data to capture long term and temporal dependencies in data.

Walk – Forward Optimization: This technique uses rolling time series cross validation (anchored walk forward) where the data is split into 'k' walks (folds). Using rolling time series cross-validation allows us to evaluate the performance of the model on multiple testing sets and provides a more realistic estimate of the model's performance. Moreover, by shifting the training and testing periods forward, we can assess how well the model generalizes to future data, which is the goal of time series forecasting. While it is expected to get a better result at k^{th} walk, that's not always a guarantee. The same can be seen in MSFT and TSLA stock prediction where the 1^{st} walk didn't converge and comparatively converged better at 4^{th} walk. But the 5^{th} walk test_loss increased abruptly. The predictions at the 5^{th} walk were better in the INTC stock [Refer Appendix and Results Analysis]. Nevertheless, the technique is useful since it incorporates arising patterns as the model moves through the training data across walks [3].

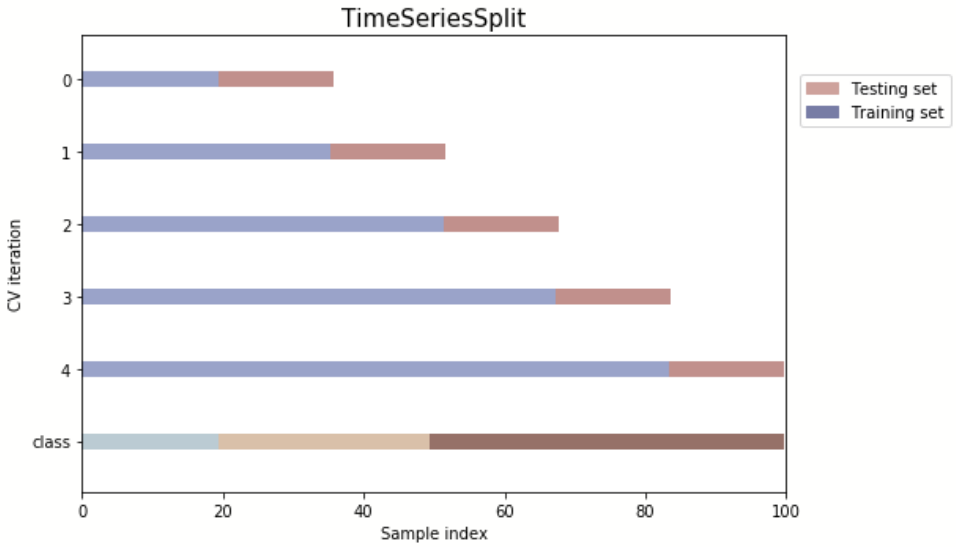
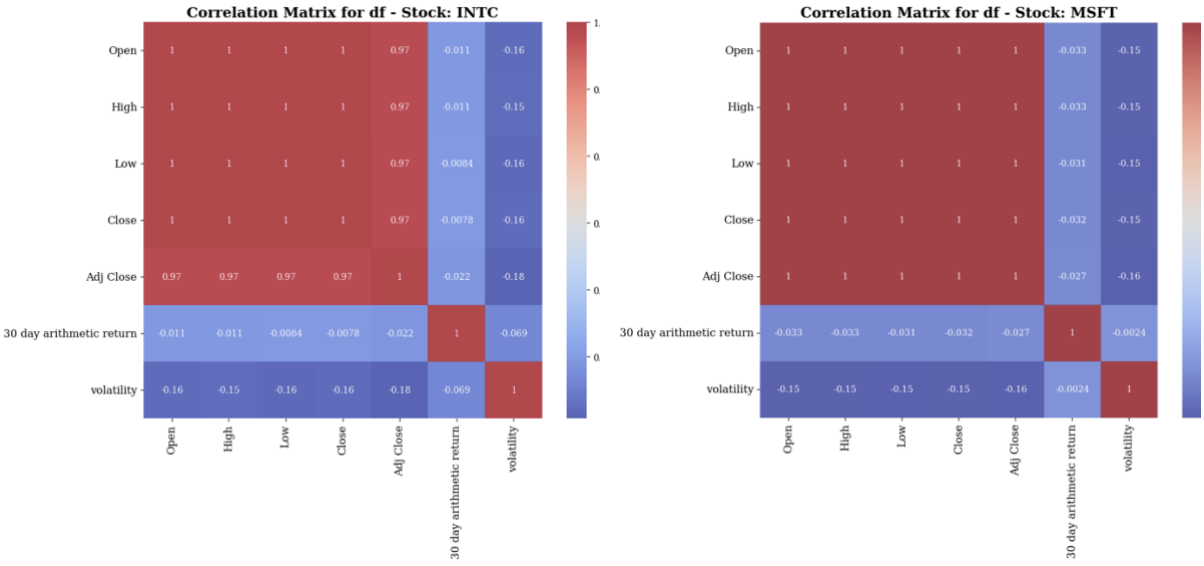


Fig 4.2: Rolling Time Series Cross Validation

Source: https://goldinlocks.github.io/

Autoencoder and Dimensionality Reduction:



From the matrices, we can clearly conclude that [Open, High, Low, Close, Adj. Close] are highly correlated with each other. Reducing dimensions using LSTM autoencoder can help remove noise and reduce computational costs during training. Therefore, the author has assigned reduced dimension range of [4, 5, 6] since that would be more reasonable and efficient to make predictions. Autoencoder is trained with a different lookback window (LB – 1) and the compressed X_train - X_test pairs are extracted from the Encoder. ReLu activation functions are commonly preferred, and it applies non-

linearity to the series. The compressed input train and test data are fed into the LSTM model and target variable Volatility is determined.

XGBoost – Ensemble Model:

For better predictions, an ensemble method XGBoost with 5 iterations of LSTM is implemented with weighted averages of final predictions. The hyperparameters in XGBoost are tuned to prevent overfitting and to ensure the model generalizes well to new data. For example, the max_depth and min_child_weight hyperparameters control the complexity of model. If these hyperparameters are too high, the model can become too complex and start fitting the noise in the data. Similarly, the gamma, reg_alpha, and reg_lambda hyperparameters control the regularization of the model, which can prevent overfitting.

5. Result Analysis:

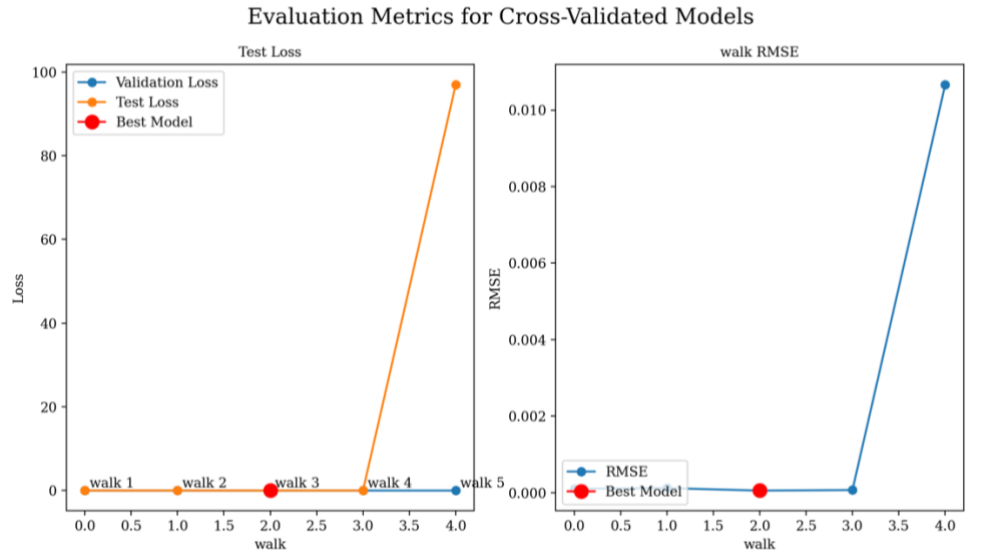


Fig 5.1: Walk – Forward Optimization: Walks 1 to 5 for 'MSFT' Stock

As discussed earlier, this clearly shows how the test loss has increased for walk 5 but remained a little over 0 until walk 4. While walk forward optimization captures temporal patterns efficiently, it might not decrease the test loss consistently in upcoming walks. In 'INTC' stock, it can be observed that the predictions converge consistently across all walks [Plots attached in Appendix].

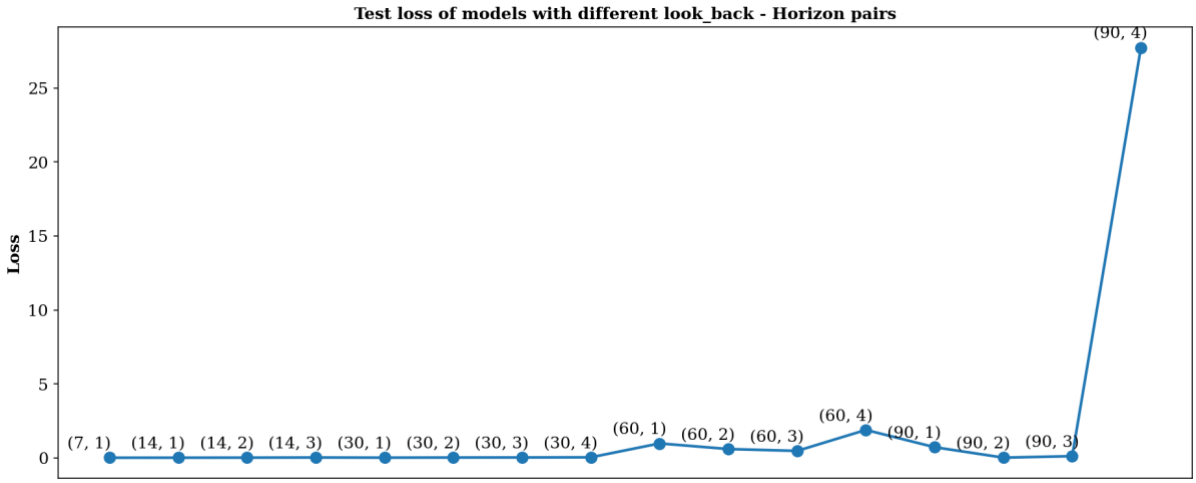


Fig 5.2: Loss of models with different (L, H) for INTC Stock

From Fig. 5.2, we can clearly see that the look_back and the horizon should not be too large for a good model. The Appendix plots contains all indicators and values inside the plots.

APPENDIX :

Figure 1 : Normalized and Denormalized Values of MSFT stock data

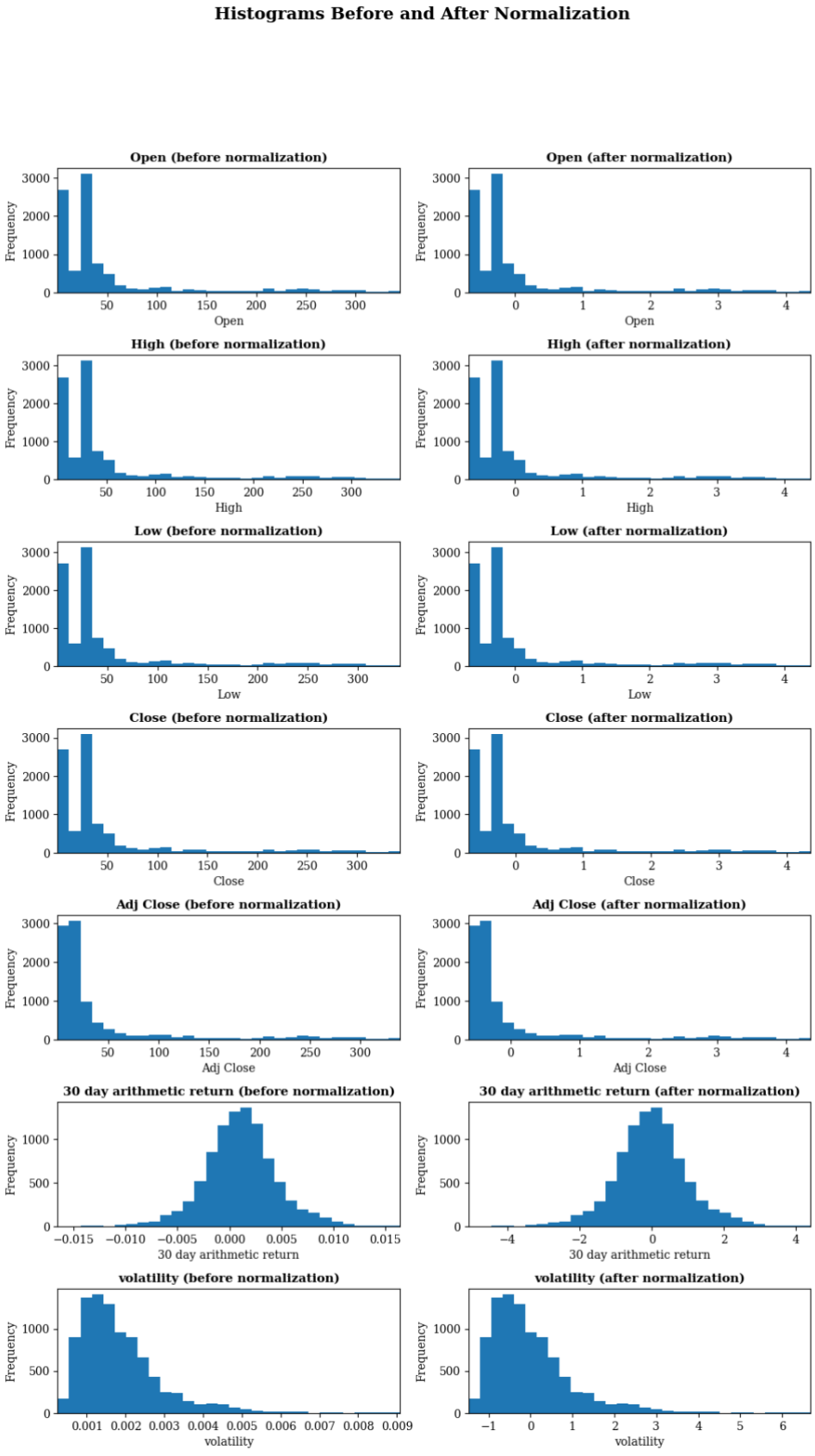
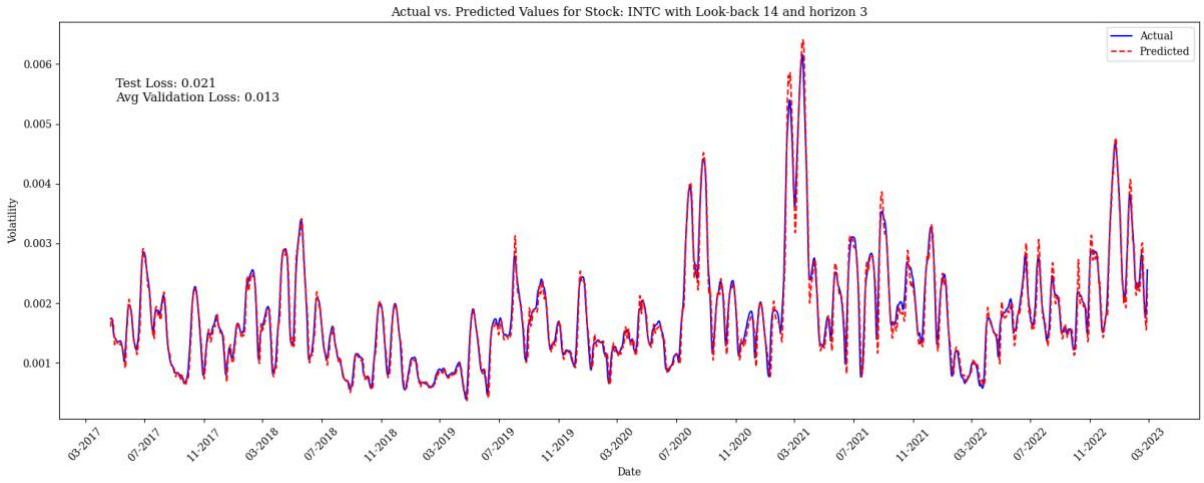
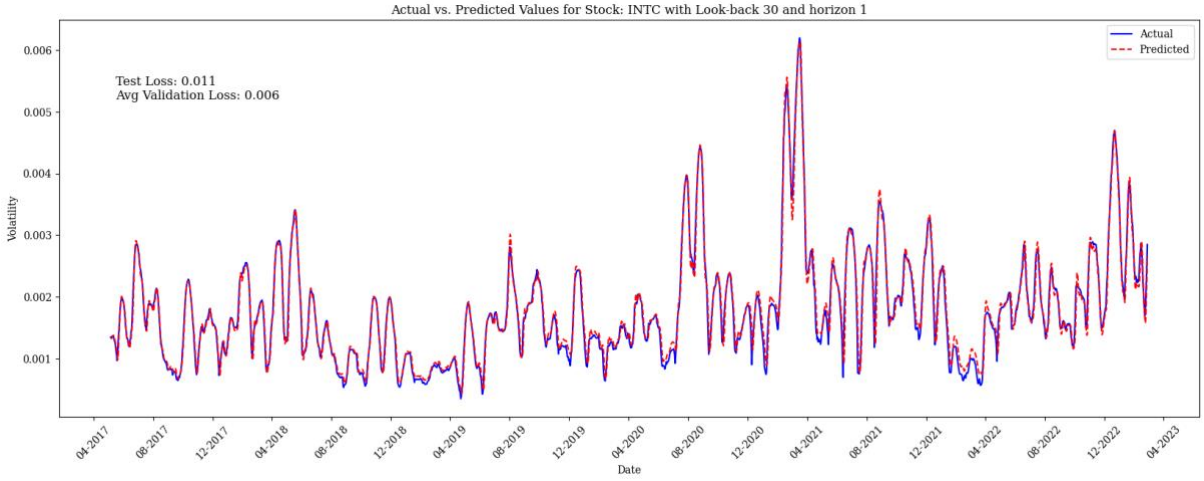
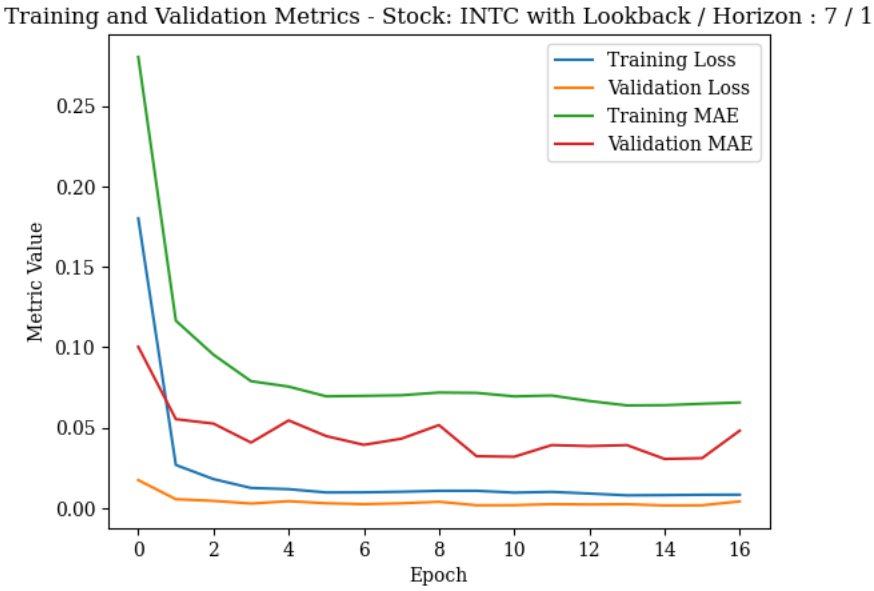
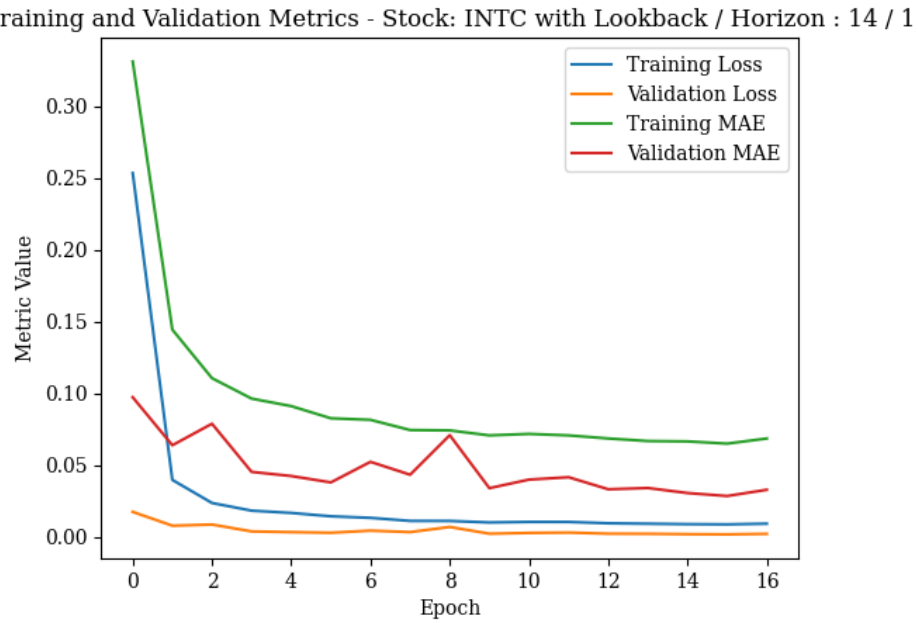


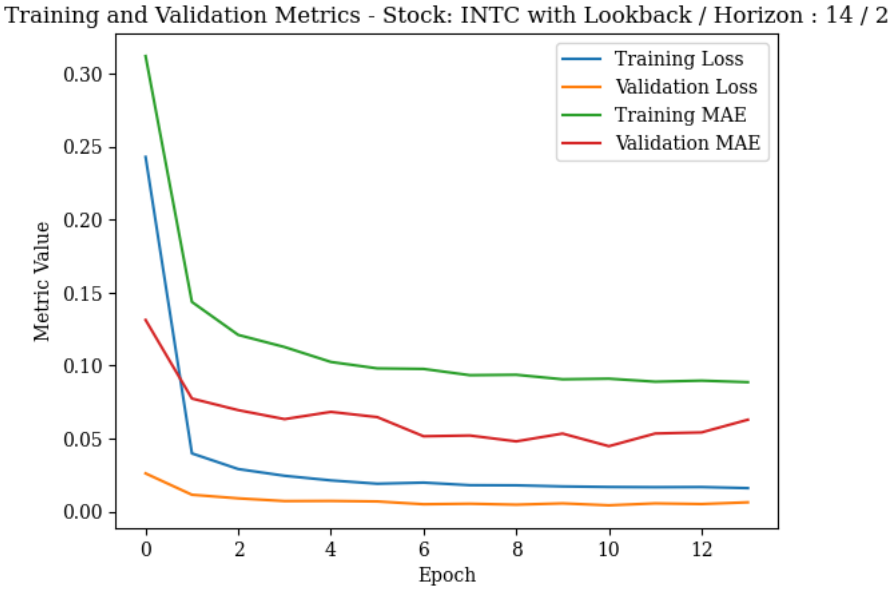
Figure 2 : Predictions for Look_back – Horizon Pair (INTC)

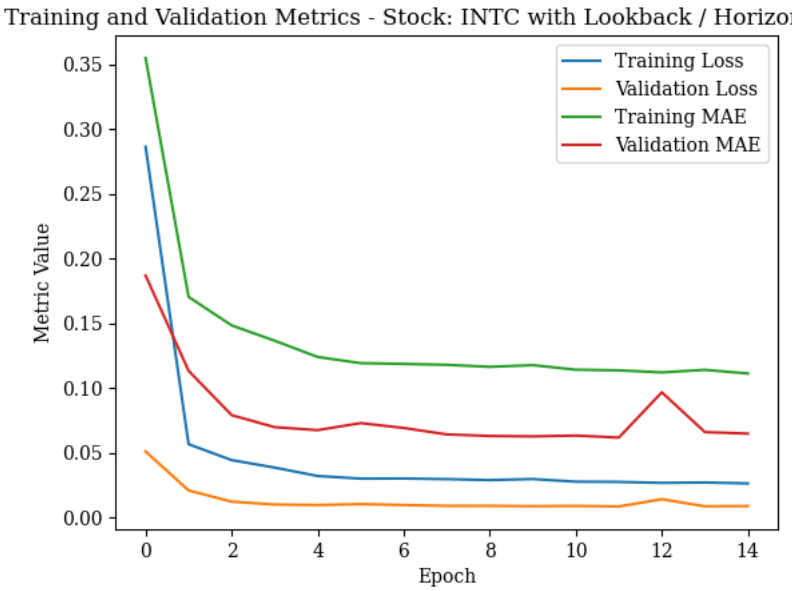

Actual vs. Predicted Values for Stock: INTC with Look-back 14 and horizon 3
Actual
Predicted
Test Loss: 0.021
Avg Validation Loss: 0.013
Volatility
Date


Actual vs. Predicted Values for Stock: INTC with Look-back 30 and horizon 1
Actual
Predicted
Test Loss: 0.011
Avg Validation Loss: 0.006
Volatility
Date


Training and Validation Metrics - Stock: INTC with Lookback / Horizon : 7 / 1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch


Training and Validation Metrics - Stock: INTC with Lookback / Horizon : 14 / 1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
2
4
6
8
10
12
14
16
Epoch


Training and Validation Metrics - Stock: INTC with Lookback / Horizon : 14 / 2
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
2
4
6
8
10
12
Epoch



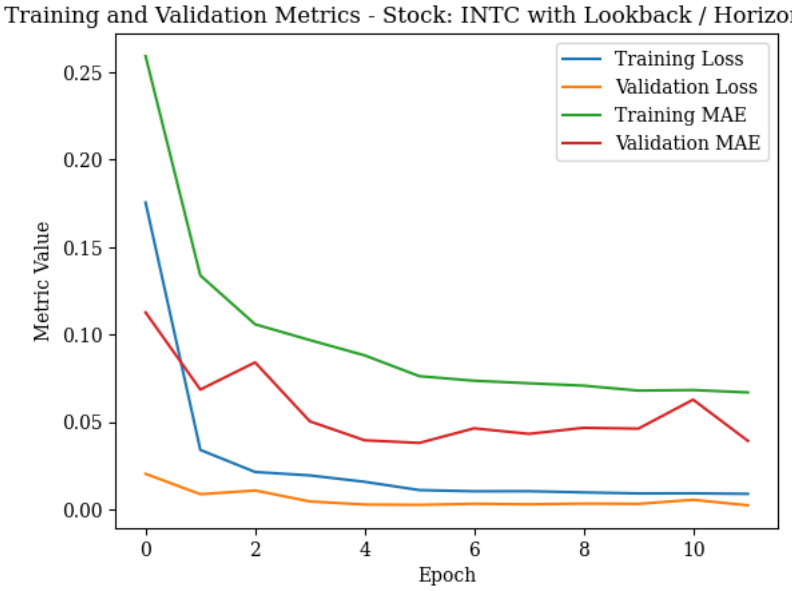
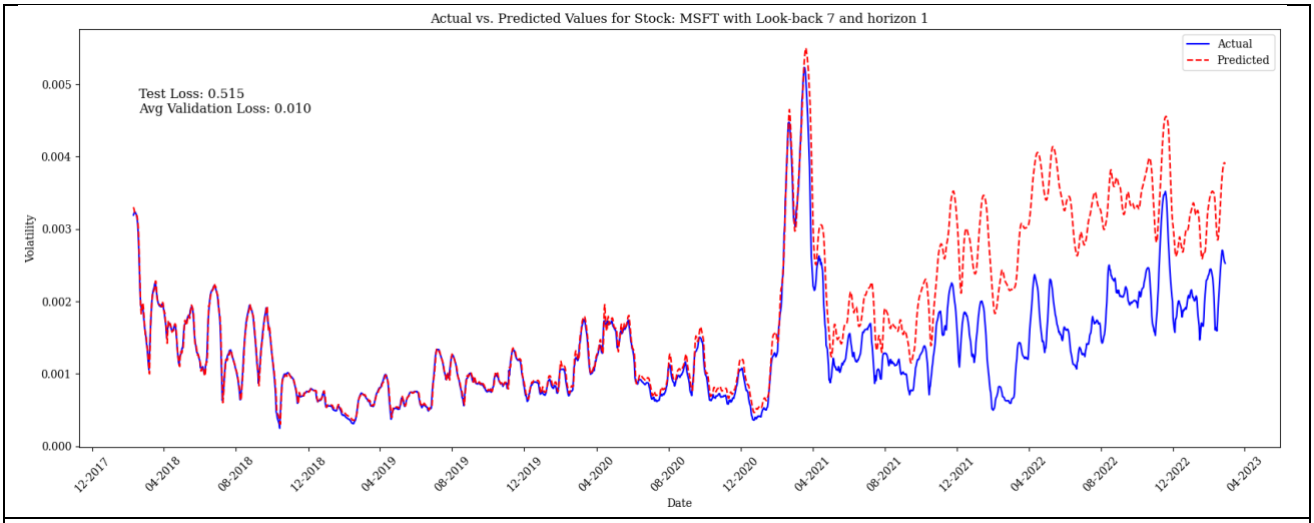
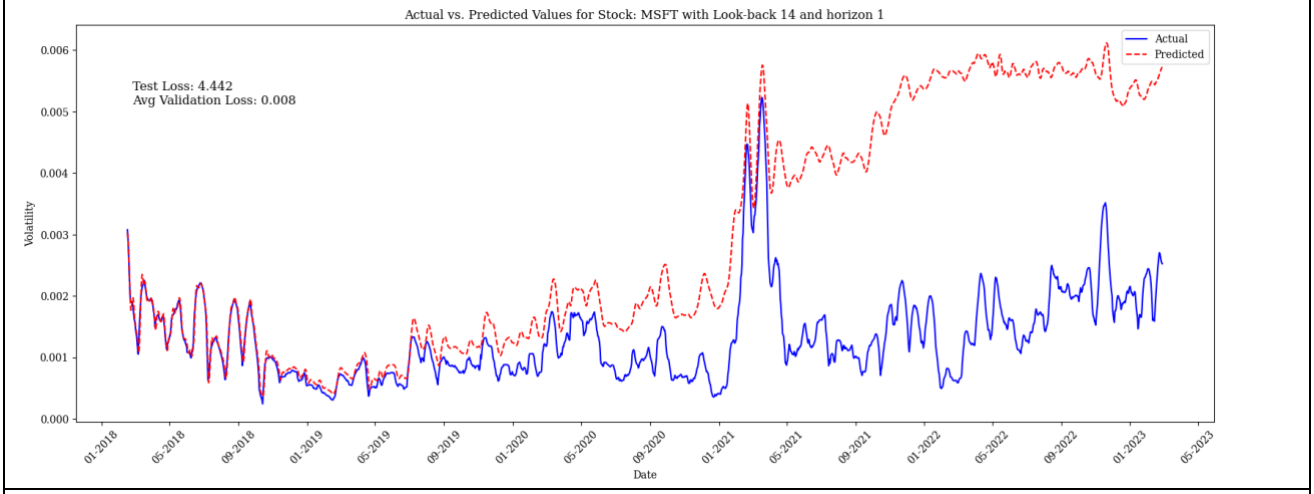
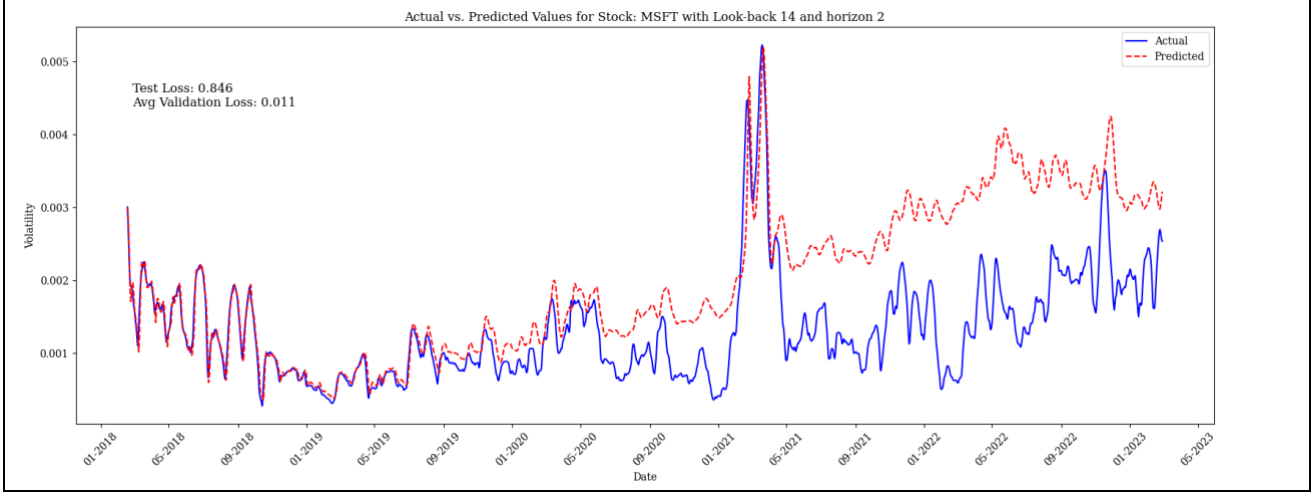
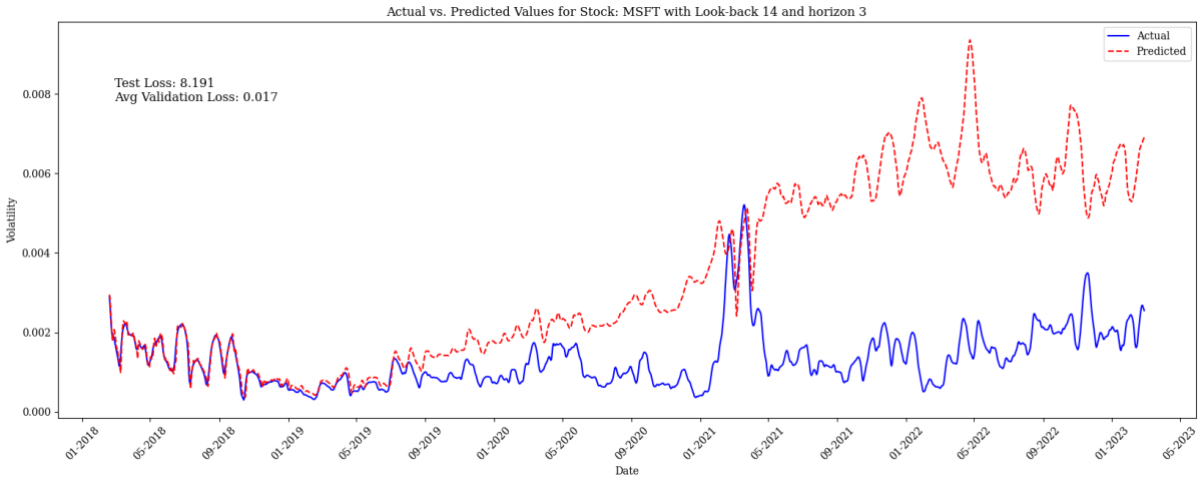


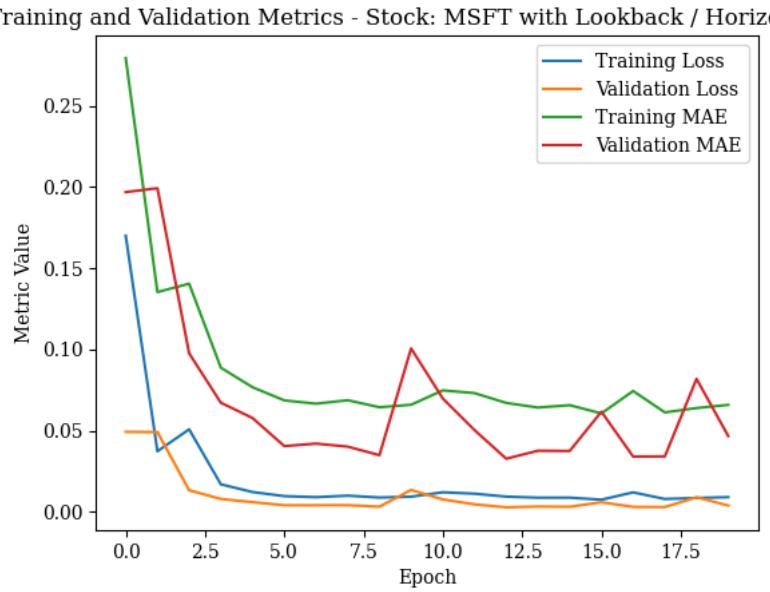
Figure 3 : Predictions for Look_back – Horizon Pair (MSFT)

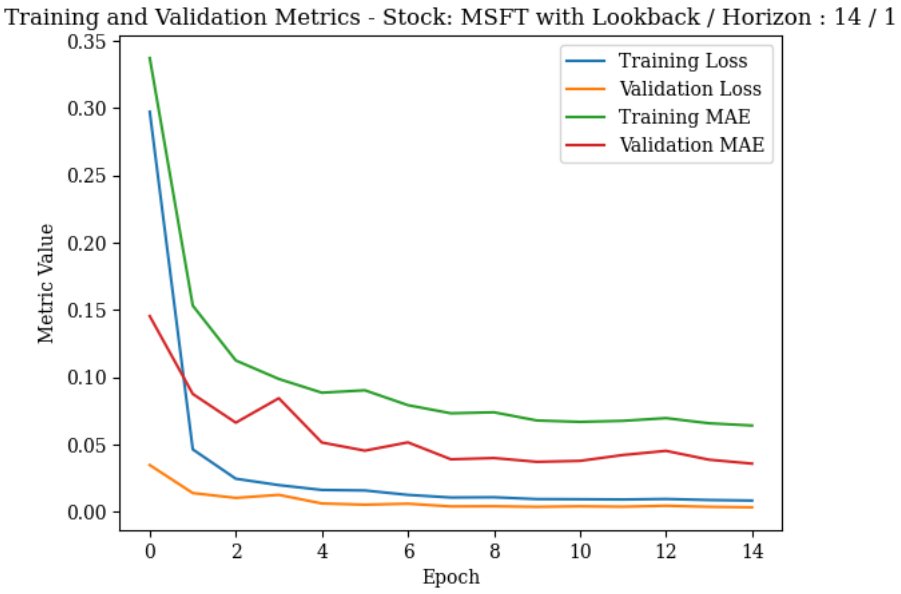

Actual vs. Predicted Values for Stock: MSFT with Look-back 7 and horizon 1
Test Loss: 0.515
Avg Validation Loss: 0.010
Actual
Predicted
Volatility
Date

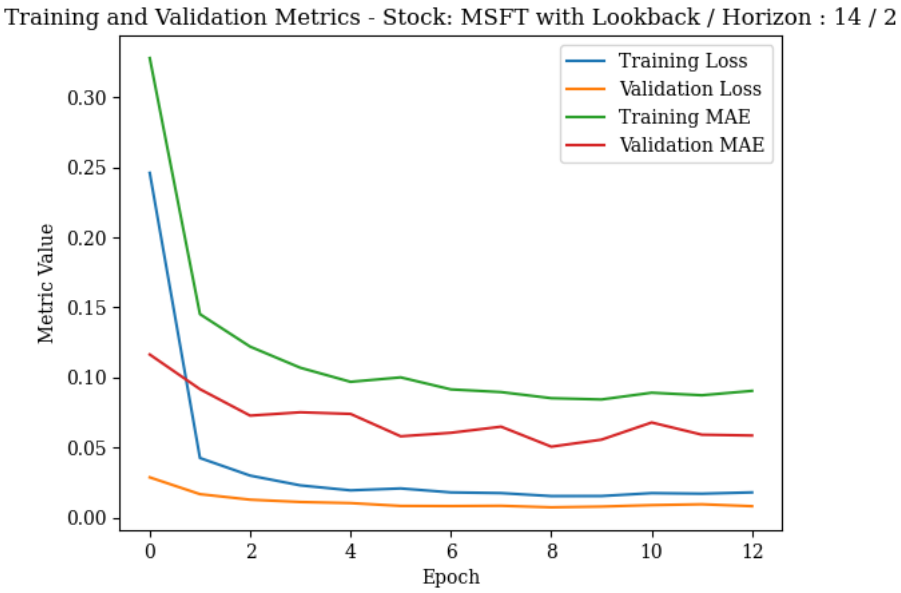

Actual vs. Predicted Values for Stock: MSFT with Look-back 14 and horizon 1
Test Loss: 4.442
Avg Validation Loss: 0.008
Actual
Predicted
Volatility
Date


Actual vs. Predicted Values for Stock: MSFT with Look-back 14 and horizon 2
Test Loss: 0.846
Avg Validation Loss: 0.011
Actual
Predicted
Volatility
Date


Actual vs. Predicted Values for Stock: MSFT with Look-back 14 and horizon 3
Actual
Predicted
Test Loss: 8.191
Avg Validation Loss: 0.017
Volatility
0.008
0.006
0.004
0.002
0.000
01-2018
05-2018
09-2018
01-2019
05-2019
09-2019
01-2020
05-2020
09-2020
01-2021
05-2021
09-2021
01-2022
05-2022
09-2022
01-2023
05-2023
Date


Training and Validation Metrics - Stock: MSFT with Lookback / Horizon : 7 / 1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
0.25
0.20
0.15
0.10
0.05
0.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch


Training and Validation Metrics - Stock: MSFT with Lookback / Horizon : 14 / 1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
10
12
14


Training and Validation Metrics - Stock: MSFT with Lookback / Horizon : 14 / 2
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
10
12

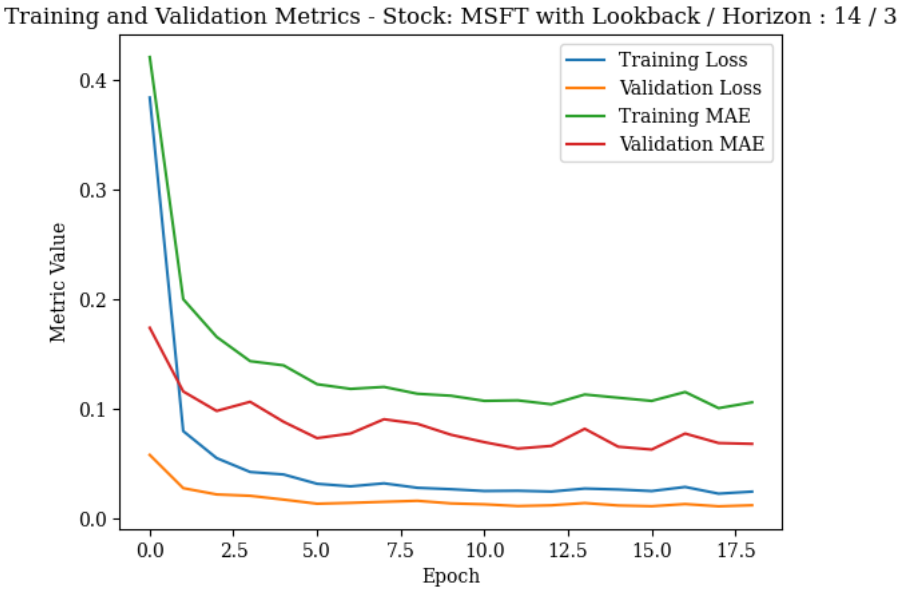
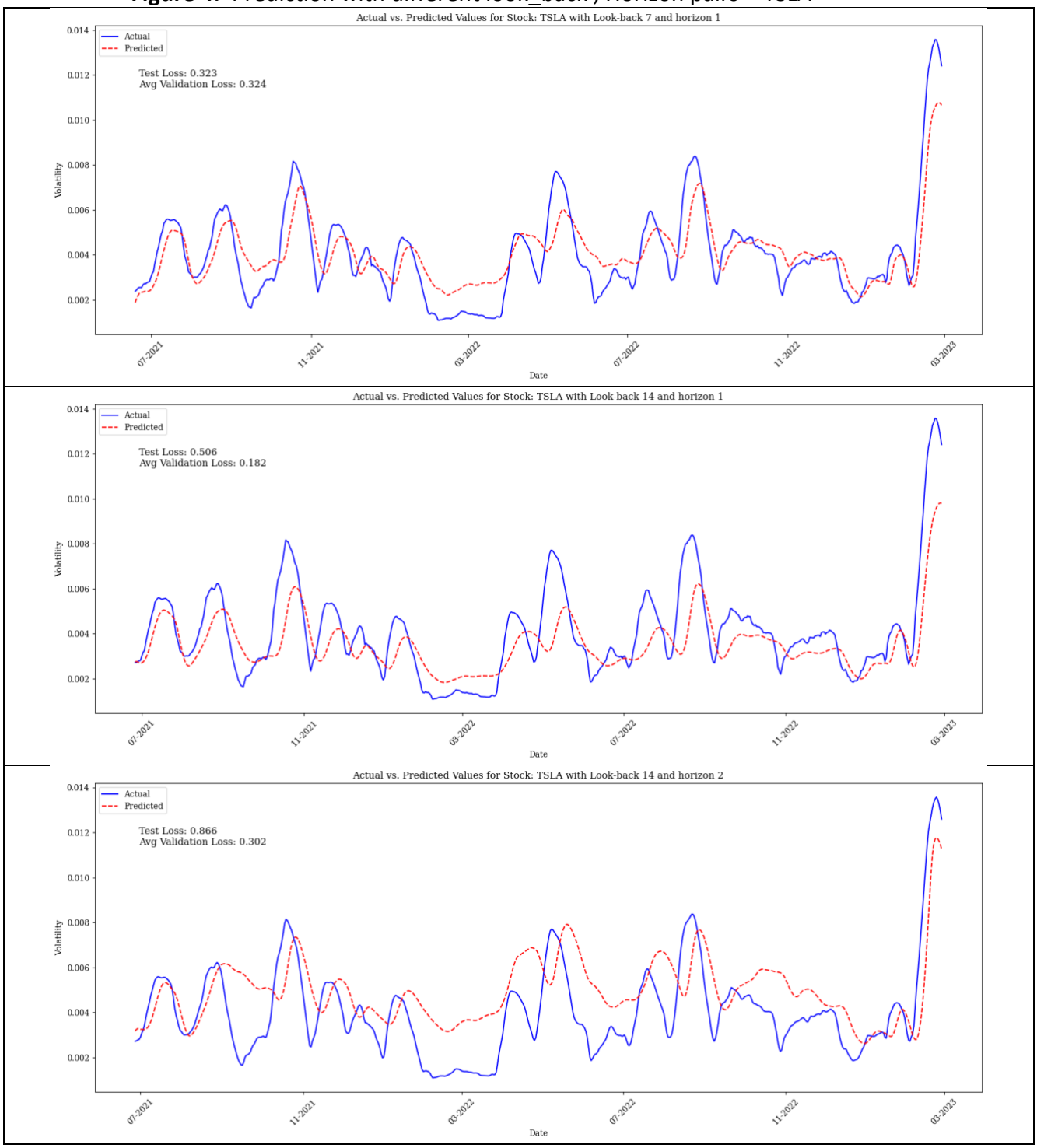
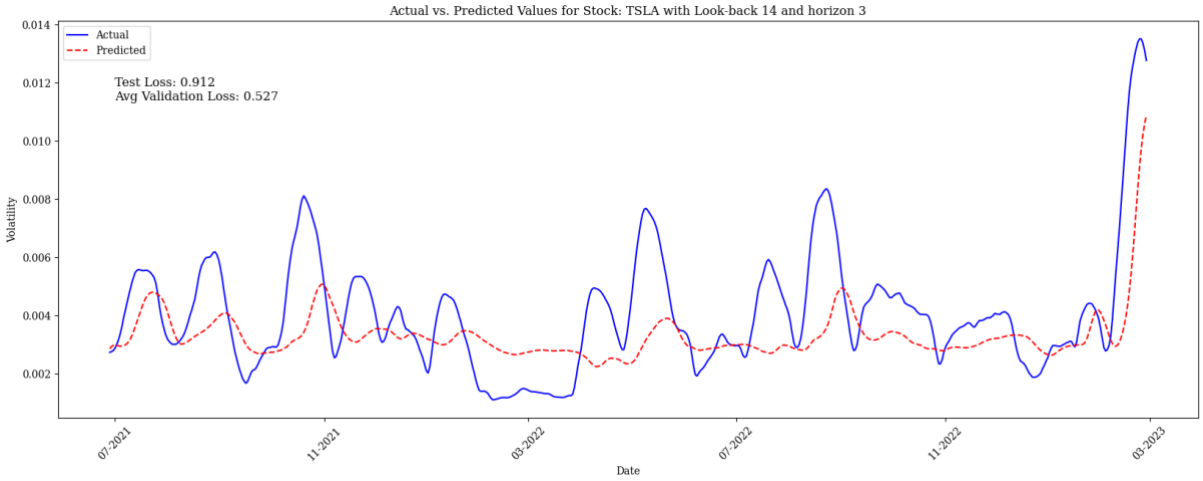
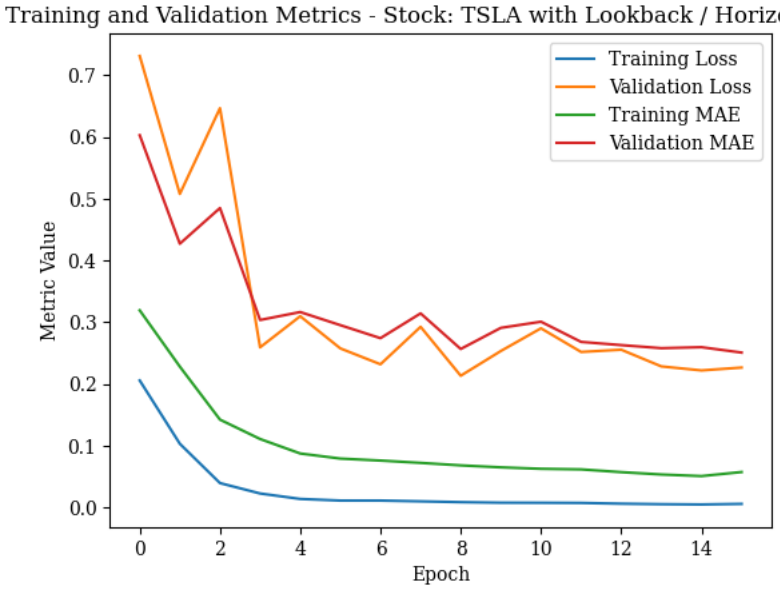
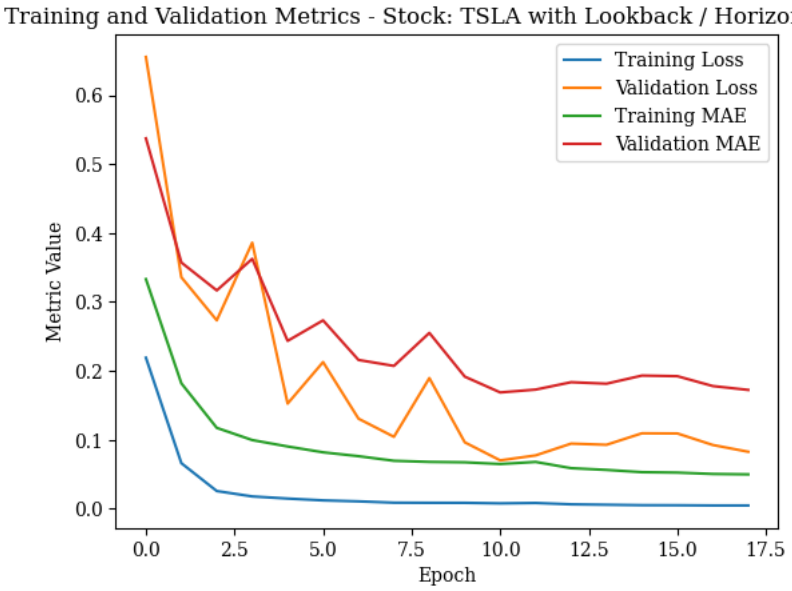

Training and Validation Metrics - Stock: MSFT with Lookback / Horizon : 14 / 3
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch
0.4
0.3
0.2
0.1
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

Figure 4: Prediction with different look_back , Horizon pairs – TSLA




Actual vs. Predicted Values for Stock: TSLA with Look-back 14 and horizon 3
Actual
Predicted
Test Loss: 0.912
Avg Validation Loss: 0.527
Volatility
Date
07-2021
11-2021
03-2022
07-2022
11-2022
03-2023


Training and Validation Metrics - Stock: TSLA with Lookback / Horizon : 7 / 1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch


Training and Validation Metrics - Stock: TSLA with Lookback / Horizon : 14 / 1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch

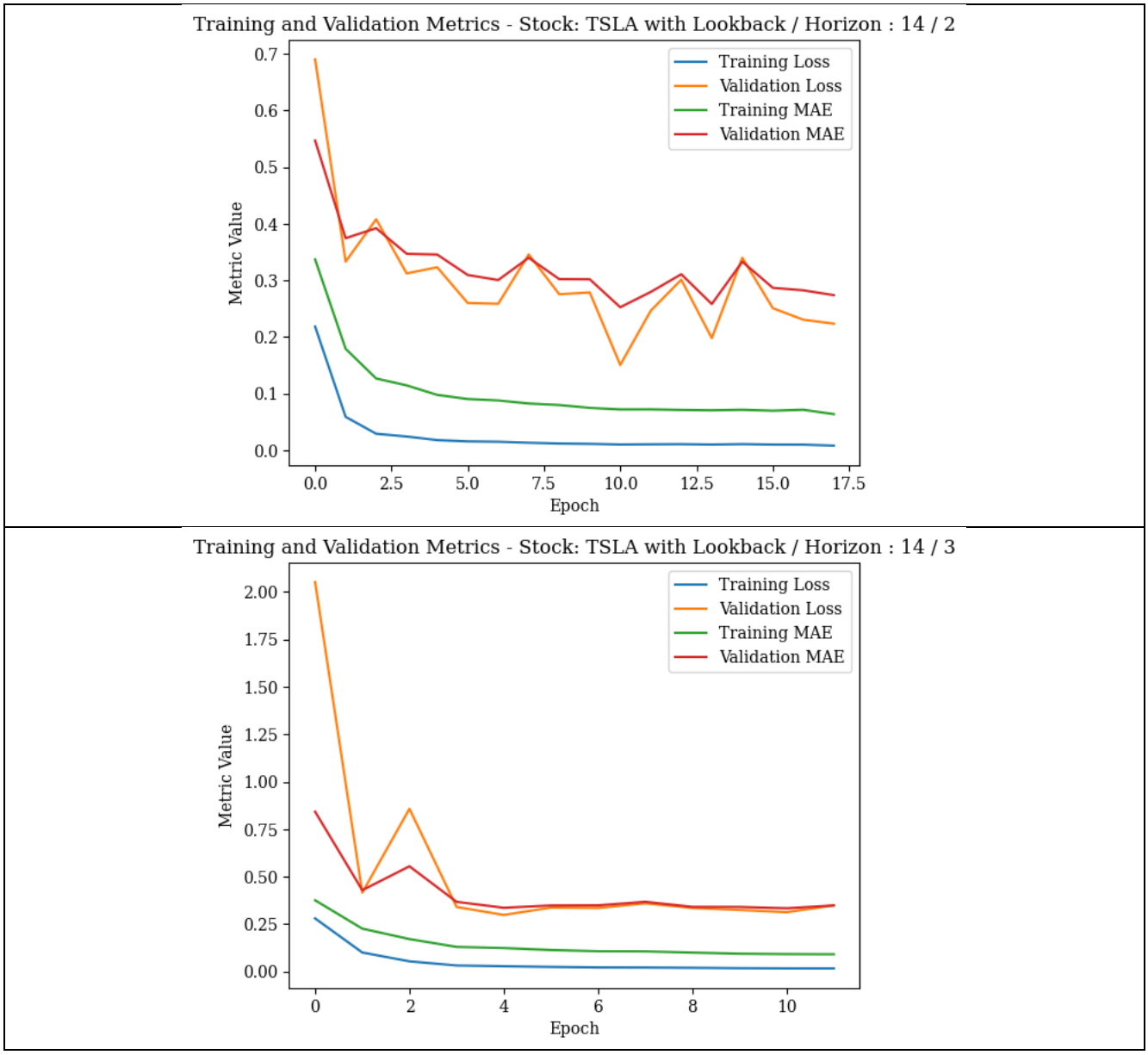
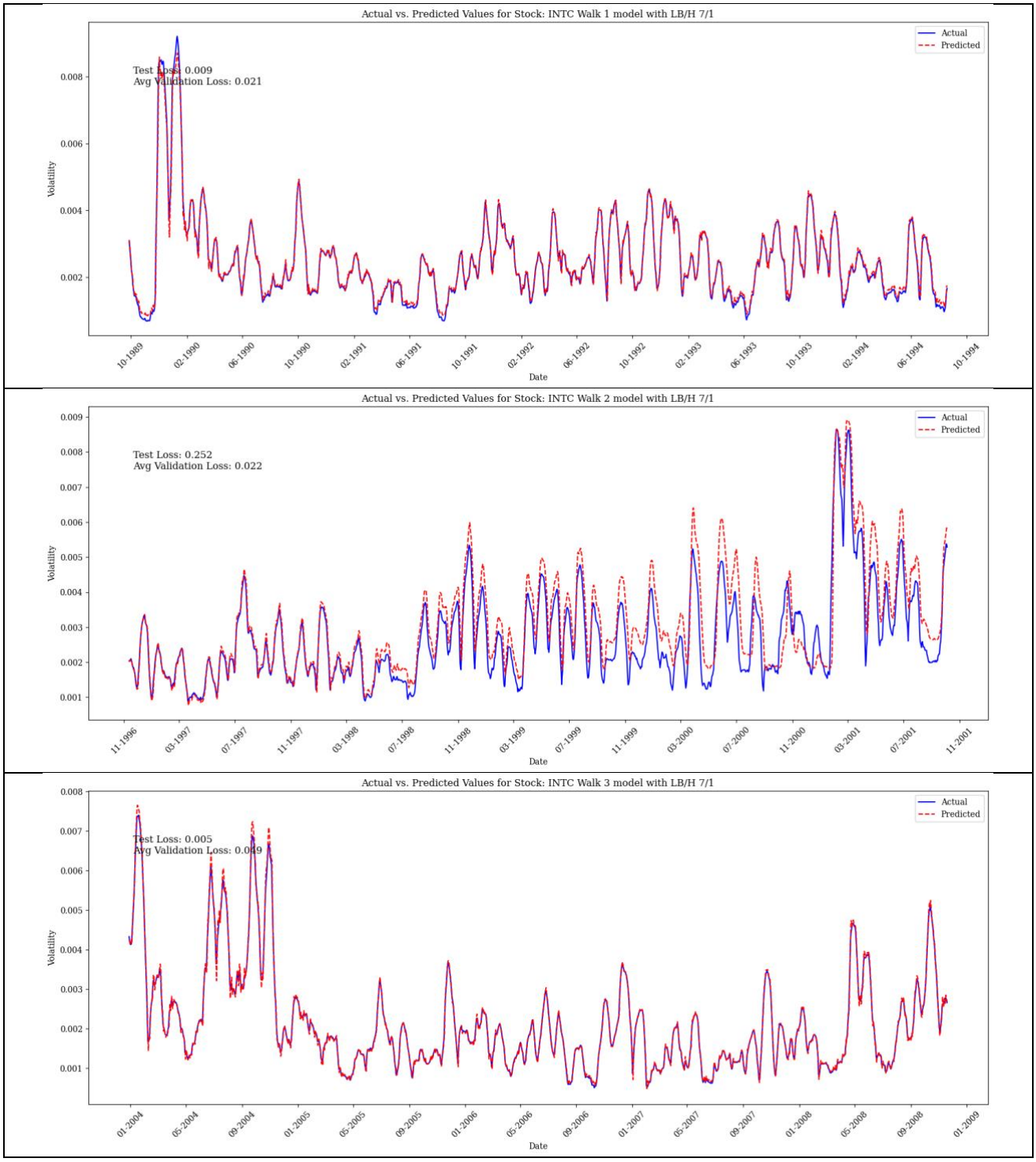
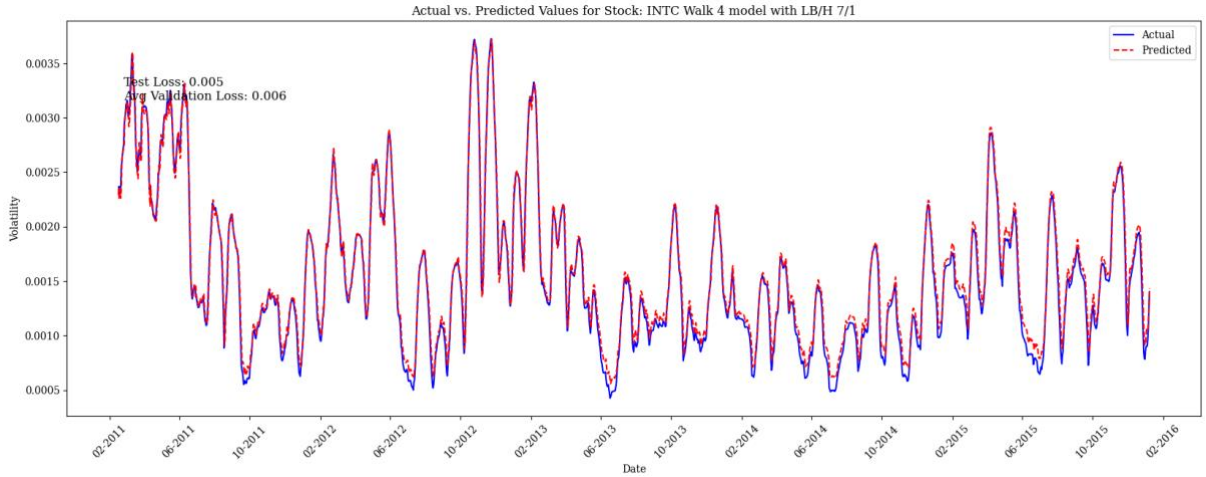
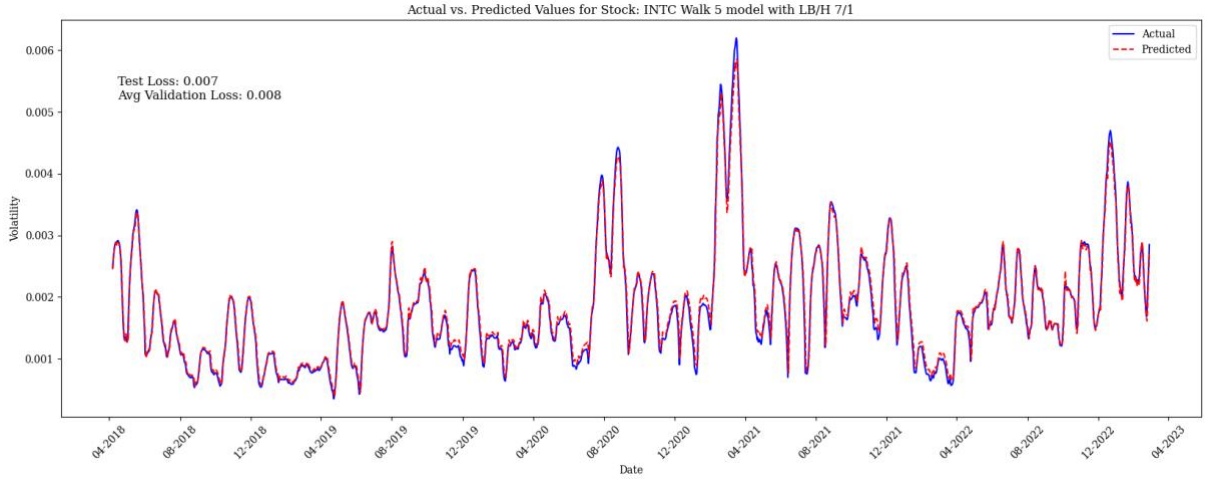
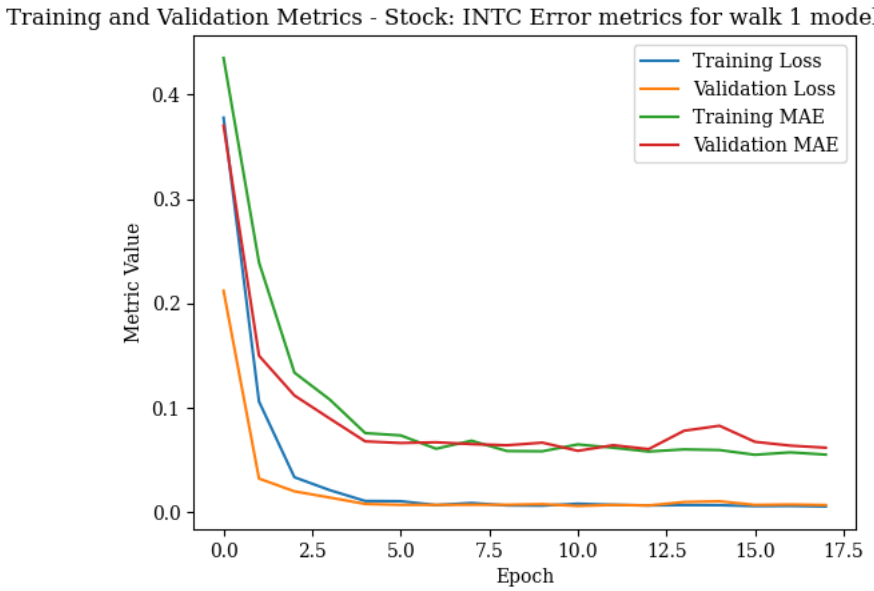


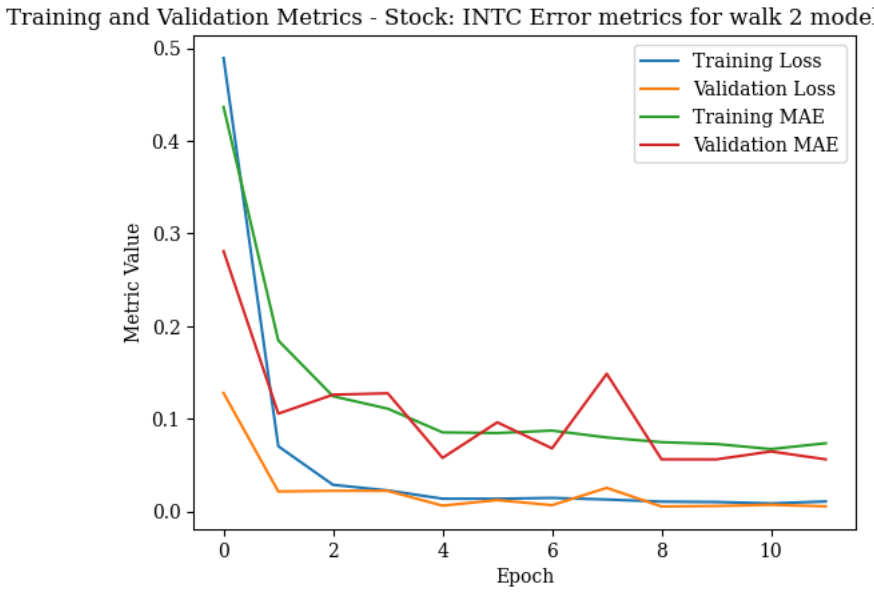
Figure 5: Prediction after Walk Forward Optimization – INTC

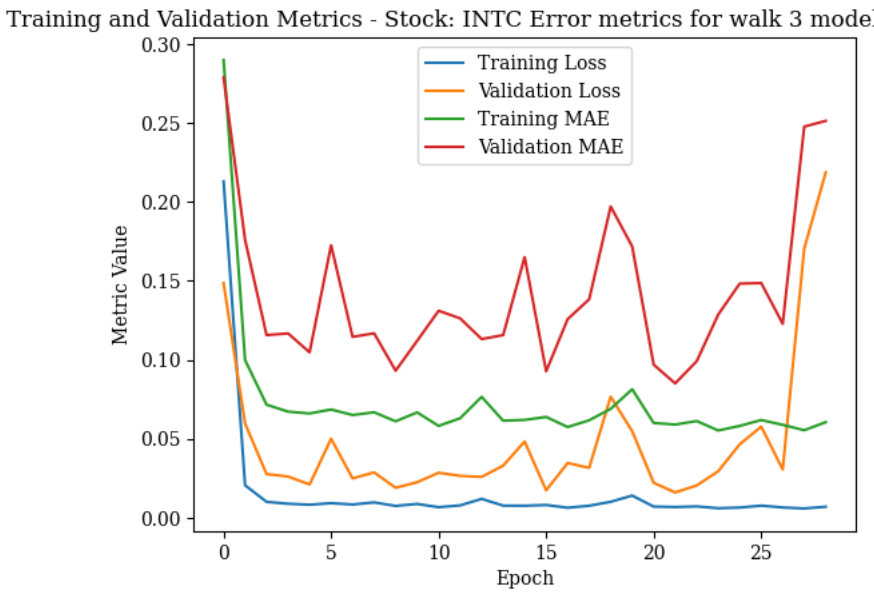

Actual vs. Predicted Values for Stock: INTC Walk 1 model with LB/H 7/1
Test Loss: 0.009
Avg Validation Loss: 0.021
Actual
Predicted
Volatility
Date
Actual vs. Predicted Values for Stock: INTC Walk 2 model with LB/H 7/1
Test Loss: 0.252
Avg Validation Loss: 0.022
Actual
Predicted
Volatility
Date
Actual vs. Predicted Values for Stock: INTC Walk 3 model with LB/H 7/1
Test Loss: 0.005
Avg Validation Loss: 0.049
Actual
Predicted
Volatility
Date

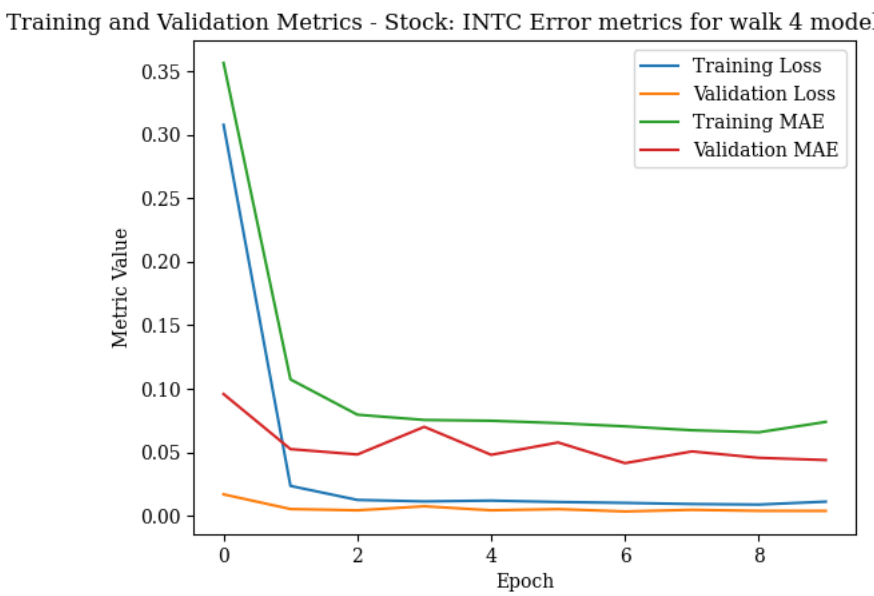

Actual vs. Predicted Values for Stock: INTC Walk 4 model with LB/H 7/1
Actual
Predicted
Test Loss: 0.005
Avg Validation Loss: 0.006
Volatility
Date


Actual vs. Predicted Values for Stock: INTC Walk 5 model with LB/H 7/1
Actual
Predicted
Test Loss: 0.007
Avg Validation Loss: 0.008
Volatility
Date


Training and Validation Metrics - Stock: INTC Error metrics for walk 1 model with LB/H 7/1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch


Training and Validation Metrics - Stock: INTC Error metrics for walk 2 model with LB/H 7/1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch
0.0
0.1
0.2
0.3
0.4
0.5
0
2
4
6
8
10


Training and Validation Metrics - Stock: INTC Error metrics for walk 3 model with LB/H 7/1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
5
10
15
20
25


Training and Validation Metrics - Stock: INTC Error metrics for walk 4 model with LB/H 7/1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
2
4
6
8

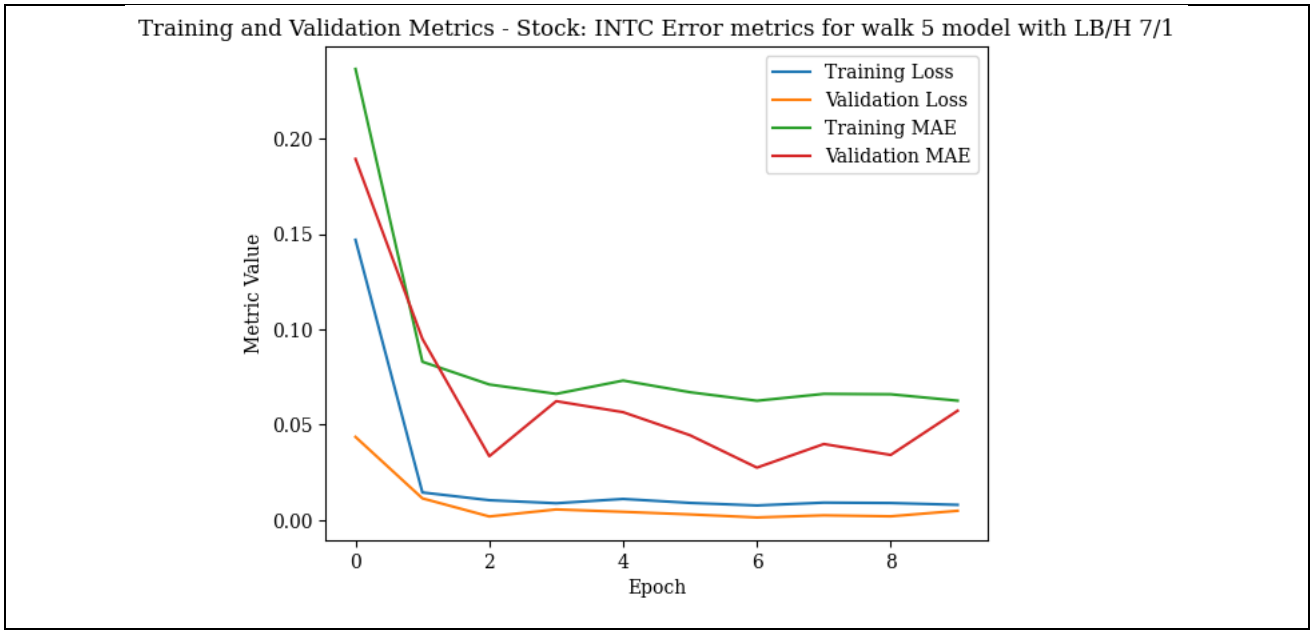
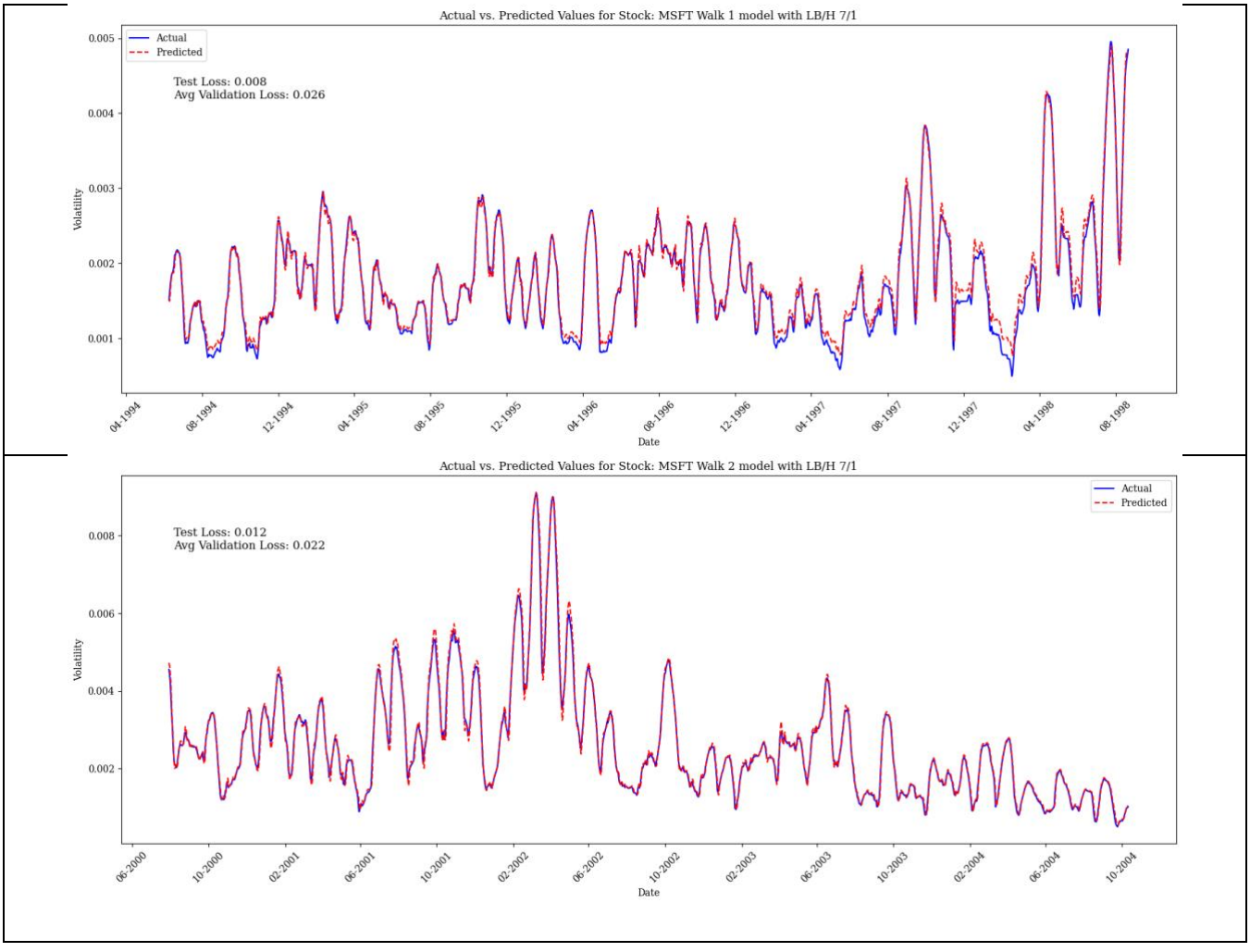
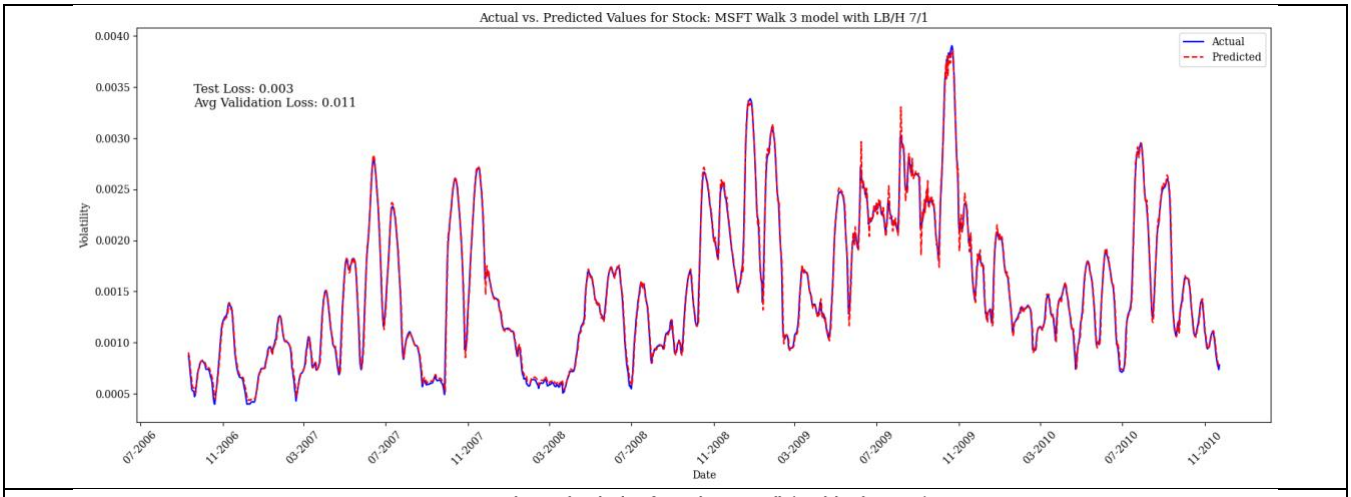
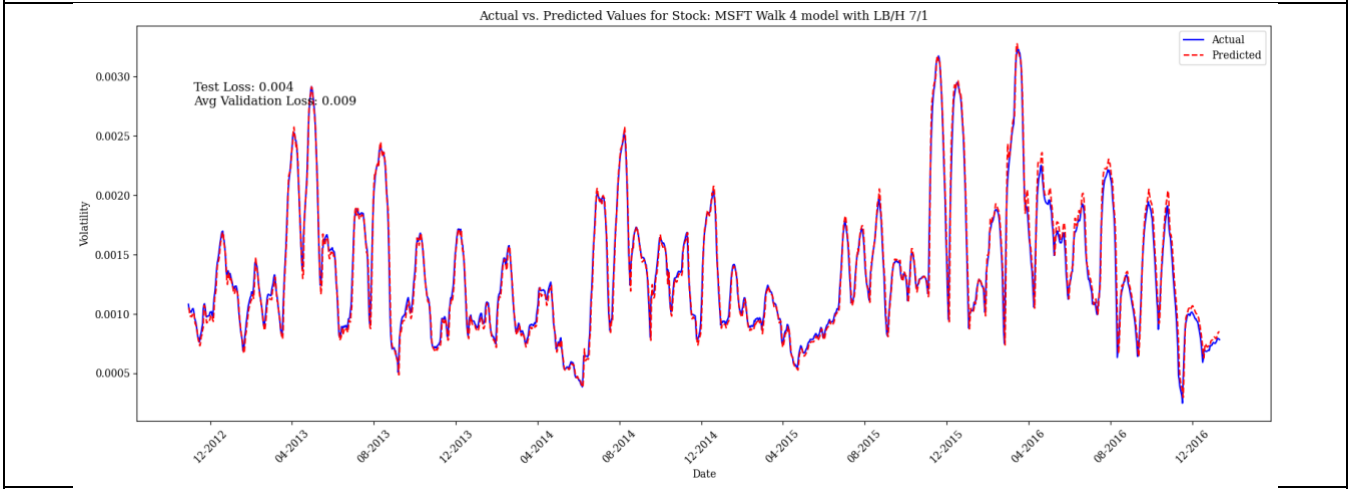
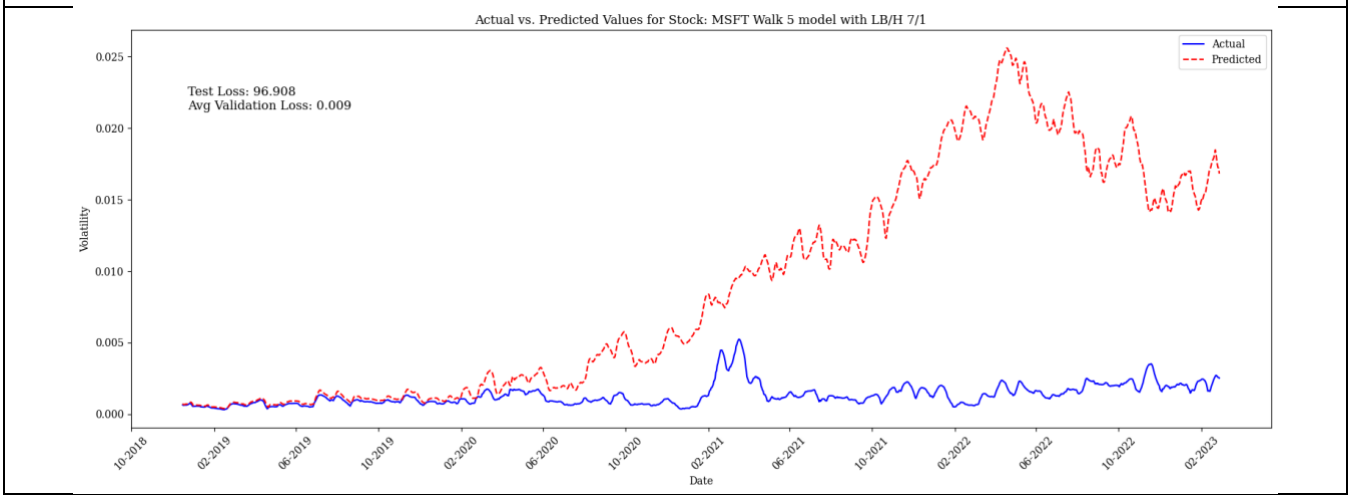


Figure 6: Prediction after Walk Forward Optimization – MSFT

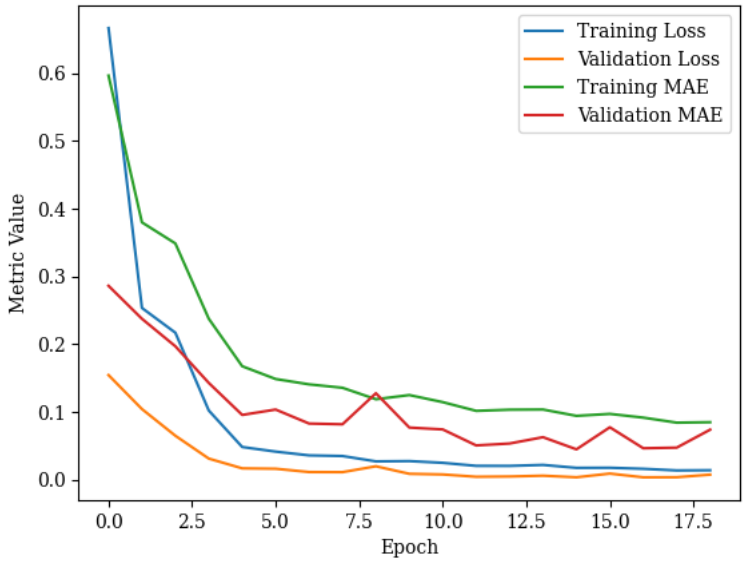



Actual vs. Predicted Values for Stock: MSFT Walk 3 model with LB/H 7/1
Test Loss: 0.003
Avg Validation Loss: 0.011
Actual
Predicted
Volatility
Date

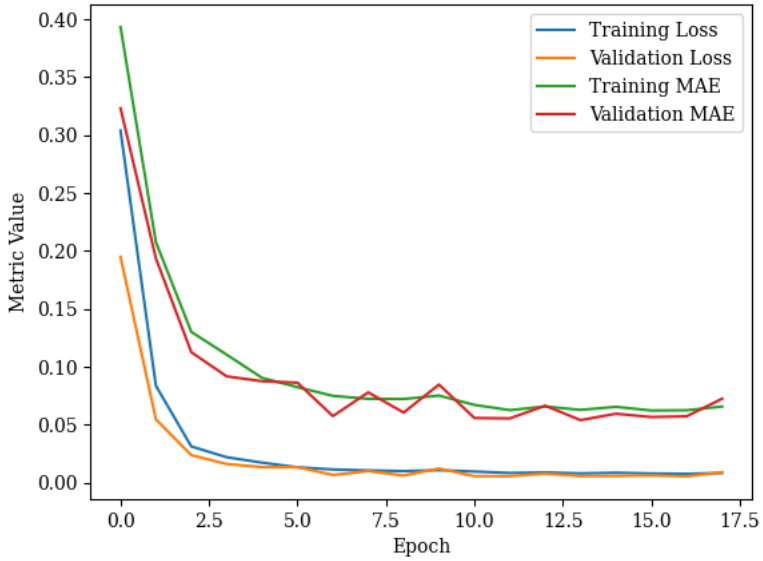

Actual vs. Predicted Values for Stock: MSFT Walk 4 model with LB/H 7/1
Test Loss: 0.004
Avg Validation Loss: 0.009
Actual
Predicted
Volatility
Date


Actual vs. Predicted Values for Stock: MSFT Walk 5 model with LB/H 7/1
Test Loss: 96.908
Avg Validation Loss: 0.009
Actual
Predicted
Volatility
Date

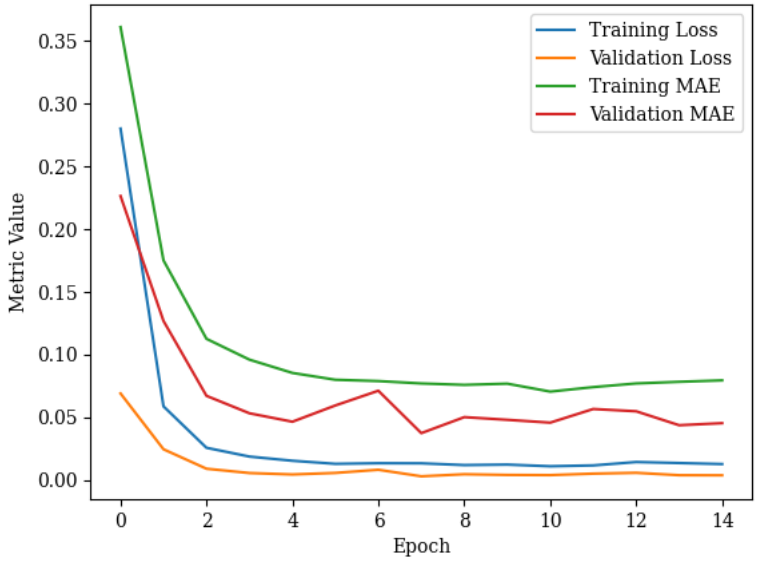
Training and Validation Metrics - Stock: MSFT Error metrics for walk 1 model with LB/H 7/1



Training and Validation Metrics - Stock: MSFT Error metrics for walk 2 model with LB/H 7/1



Training and Validation Metrics - Stock: MSFT Error metrics for walk 3 model with LB/H 7/1



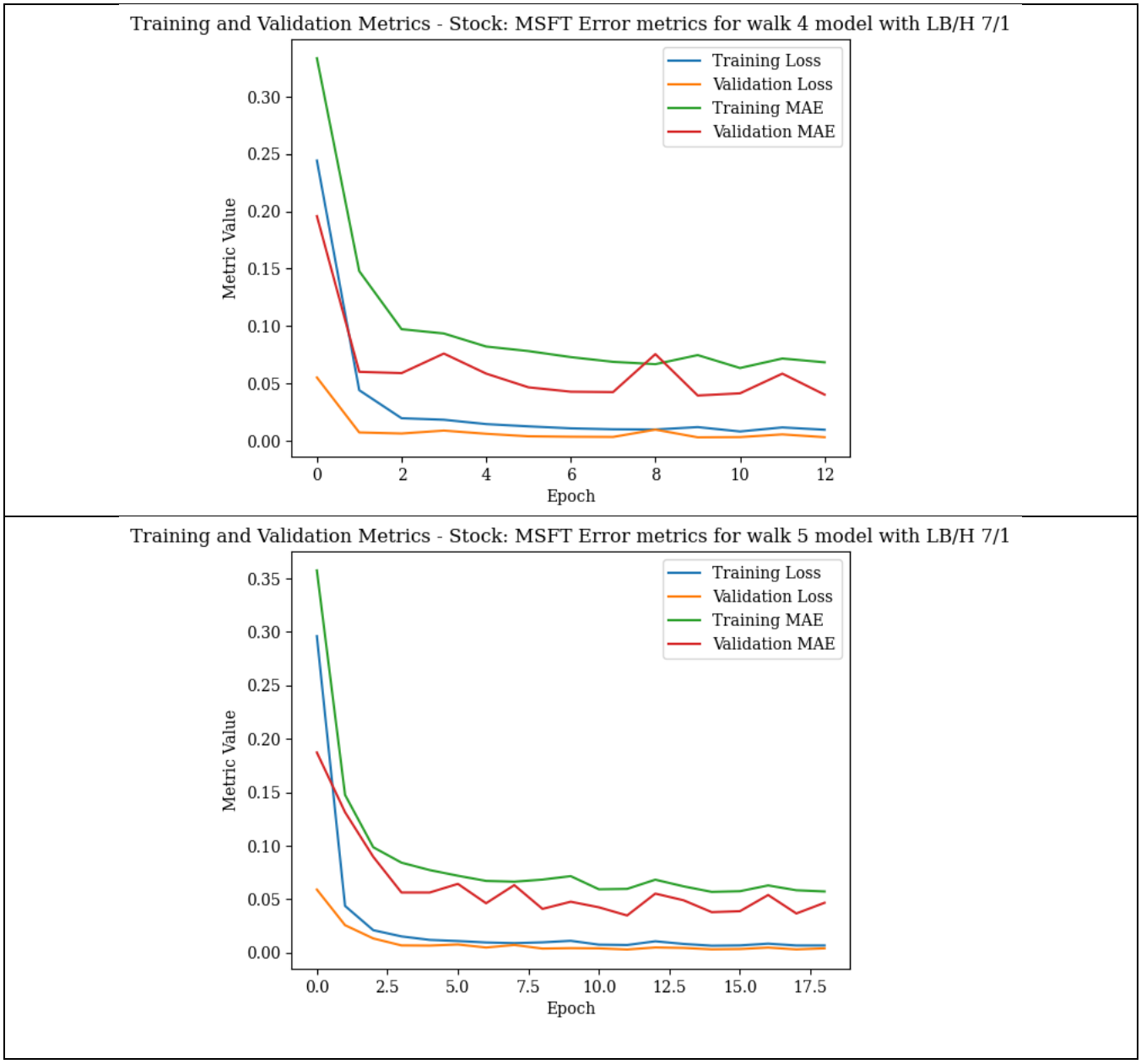
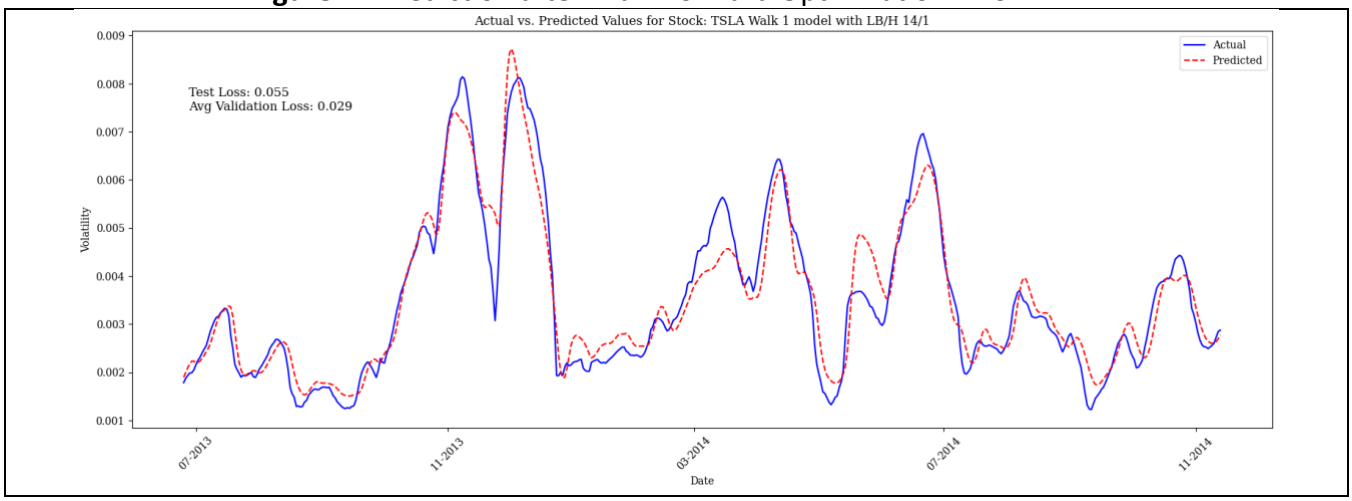
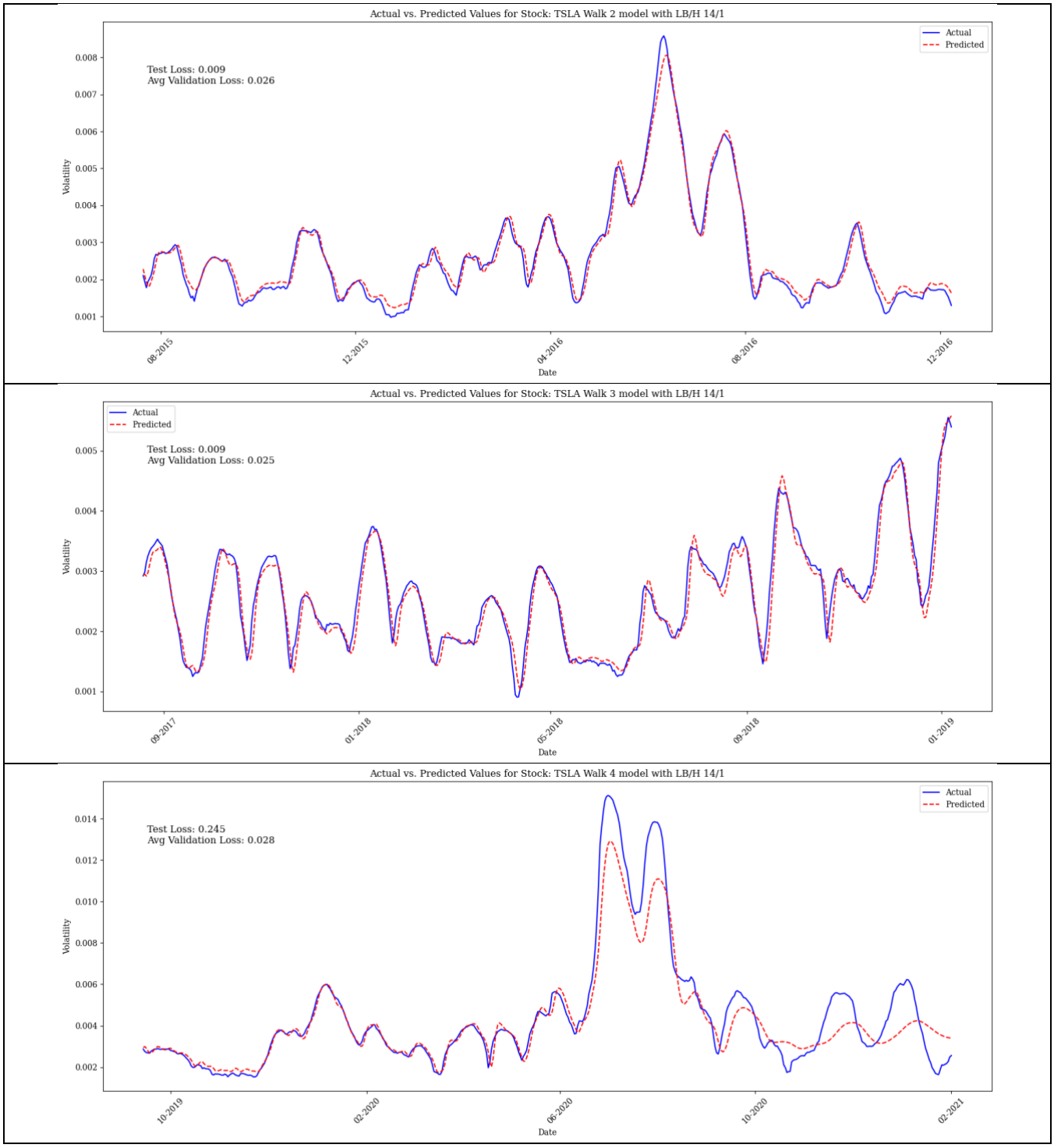
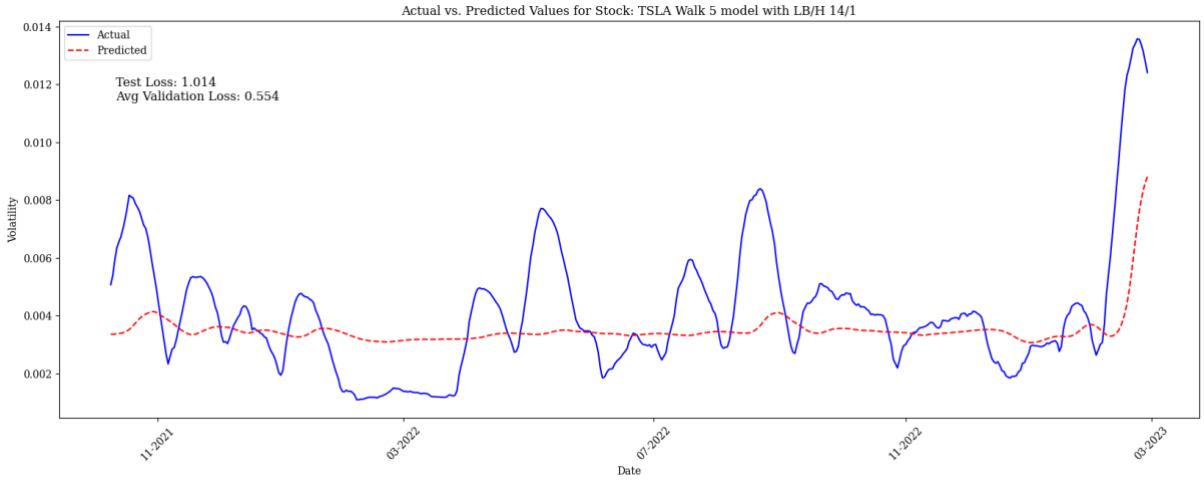
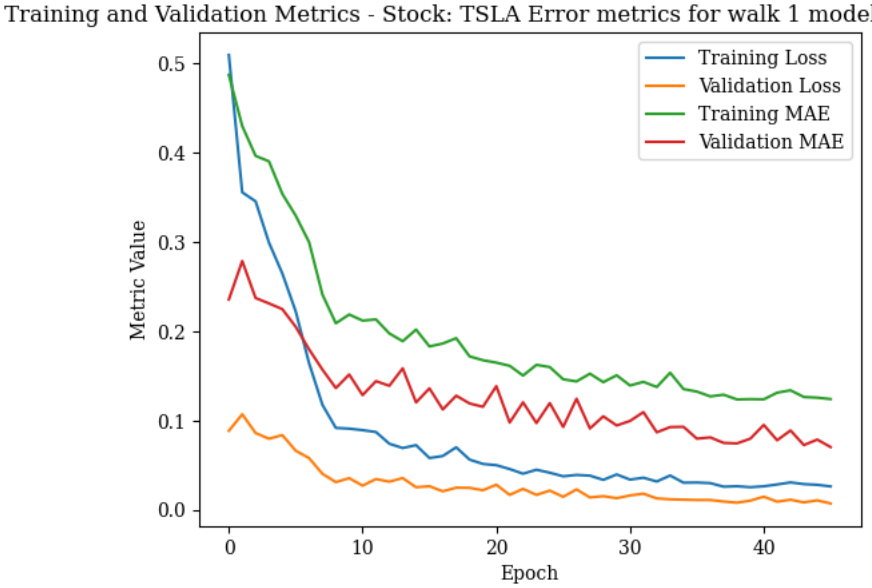


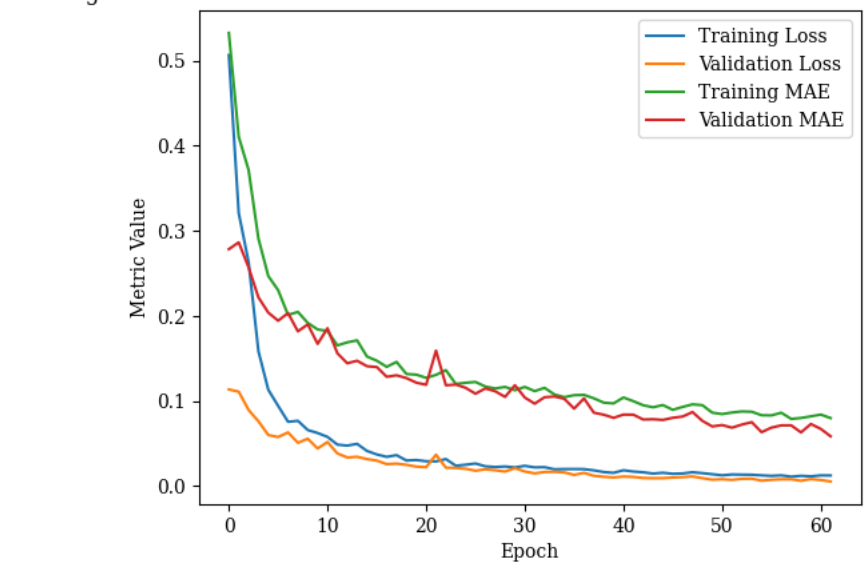
Figure 7: Prediction after Walk Forward Optimization – TSLA

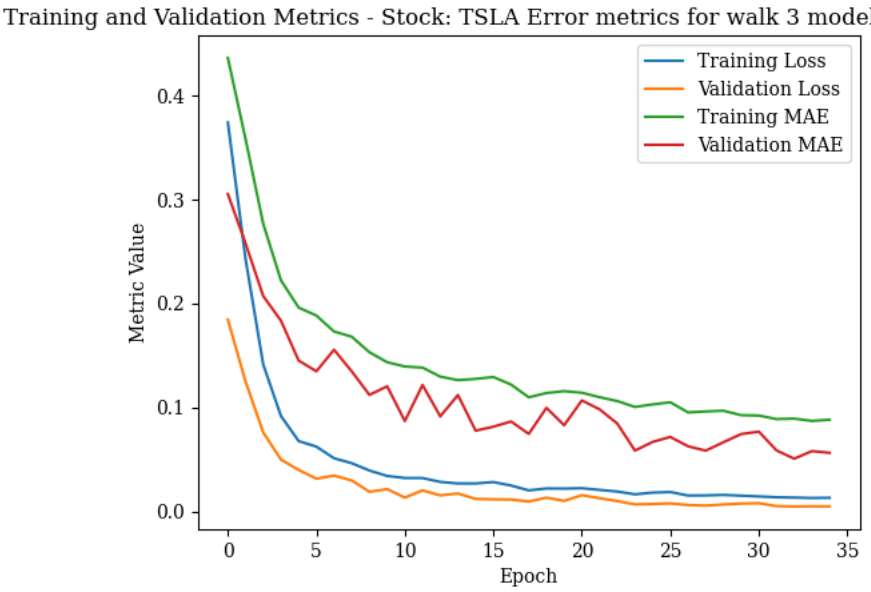


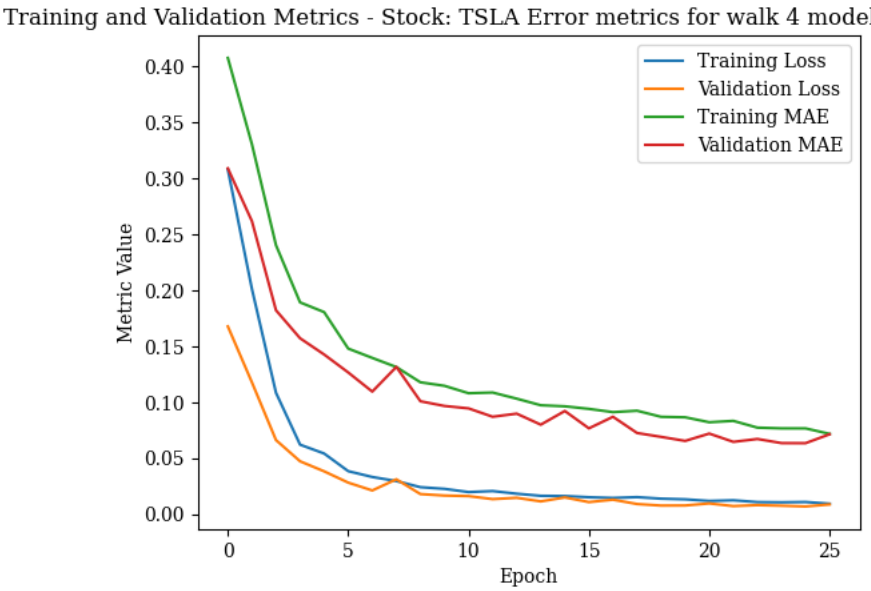

Actual vs. Predicted Values for Stock: TSLA Walk 2 model with LB/H 14/1
Actual
Predicted
Test Loss: 0.009
Avg Validation Loss: 0.026
Volatility
Date
08-2015
12-2015
04-2016
08-2016
12-2016
Actual vs. Predicted Values for Stock: TSLA Walk 3 model with LB/H 14/1
Actual
Predicted
Test Loss: 0.009
Avg Validation Loss: 0.025
Volatility
Date
09-2017
01-2018
05-2018
09-2018
01-2019
Actual vs. Predicted Values for Stock: TSLA Walk 4 model with LB/H 14/1
Actual
Predicted
Test Loss: 0.245
Avg Validation Loss: 0.028
Volatility
Date
10-2019
02-2020
06-2020
10-2020
02-2021


Actual vs. Predicted Values for Stock: TSLA Walk 5 model with LB/H 14/1
Actual
Predicted
Test Loss: 1.014
Avg Validation Loss: 0.554
Volatility
Date
0.014
0.012
0.010
0.008
0.006
0.004
0.002
11-2021
03-2022
07-2022
11-2022
03-2023


Training and Validation Metrics - Stock: TSLA Error metrics for walk 1 model with LB/H 14/1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40


Training and Validation Metrics - Stock: TSLA Error metrics for walk 2 model with LB/H 14/1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40
50
60


Training and Validation Metrics - Stock: TSLA Error metrics for walk 3 model with LB/H 14/1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch


Training and Validation Metrics - Stock: TSLA Error metrics for walk 4 model with LB/H 14/1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch

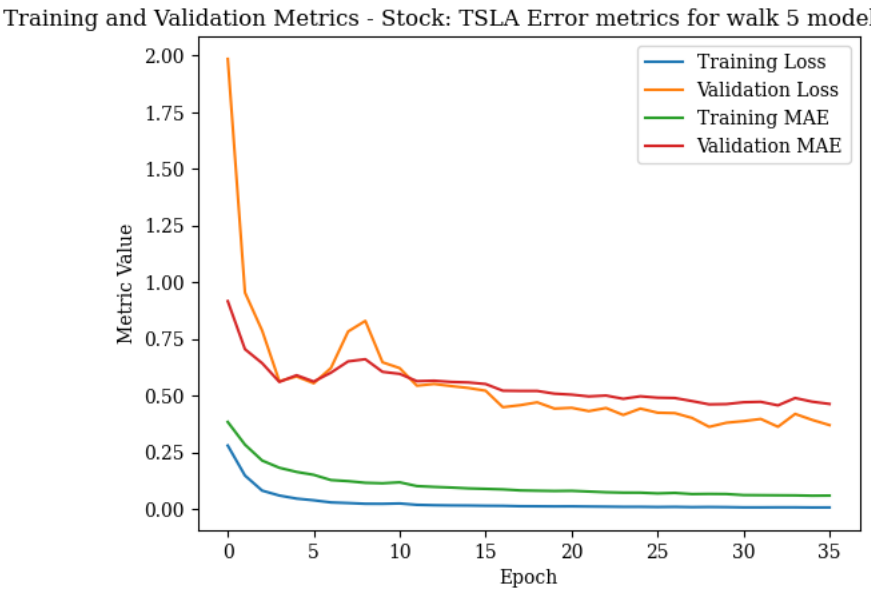

Training and Validation Metrics - Stock: TSLA Error metrics for walk 5 model with LB/H 14/1
Training Loss
Validation Loss
Training MAE
Validation MAE
Metric Value
Epoch

Figure 8: Prediction after Dimensionality Reduction Autoencoder – INTC

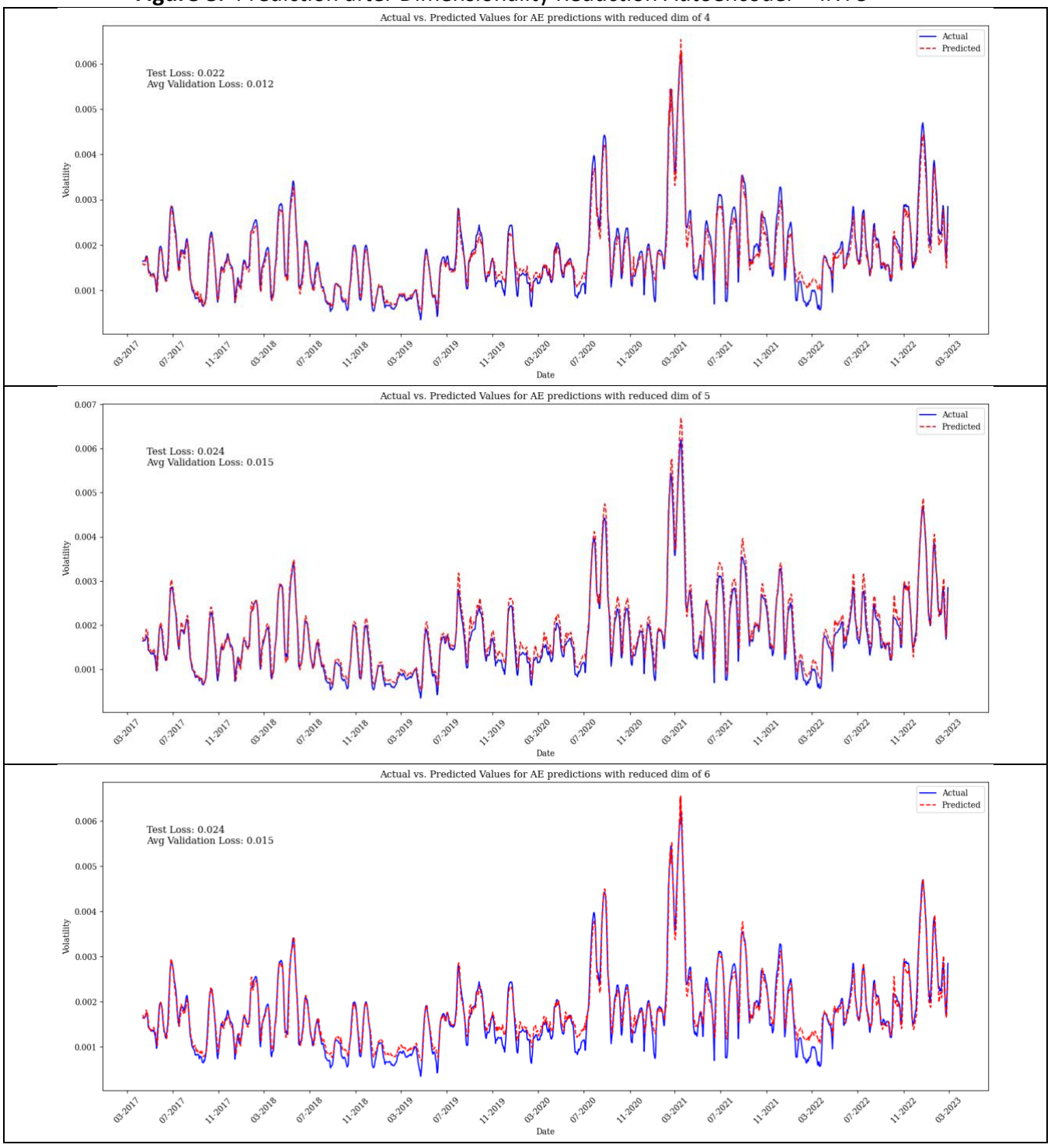


Figure 9: Prediction after Dimensionality Reduction Autoencoder – MSFT

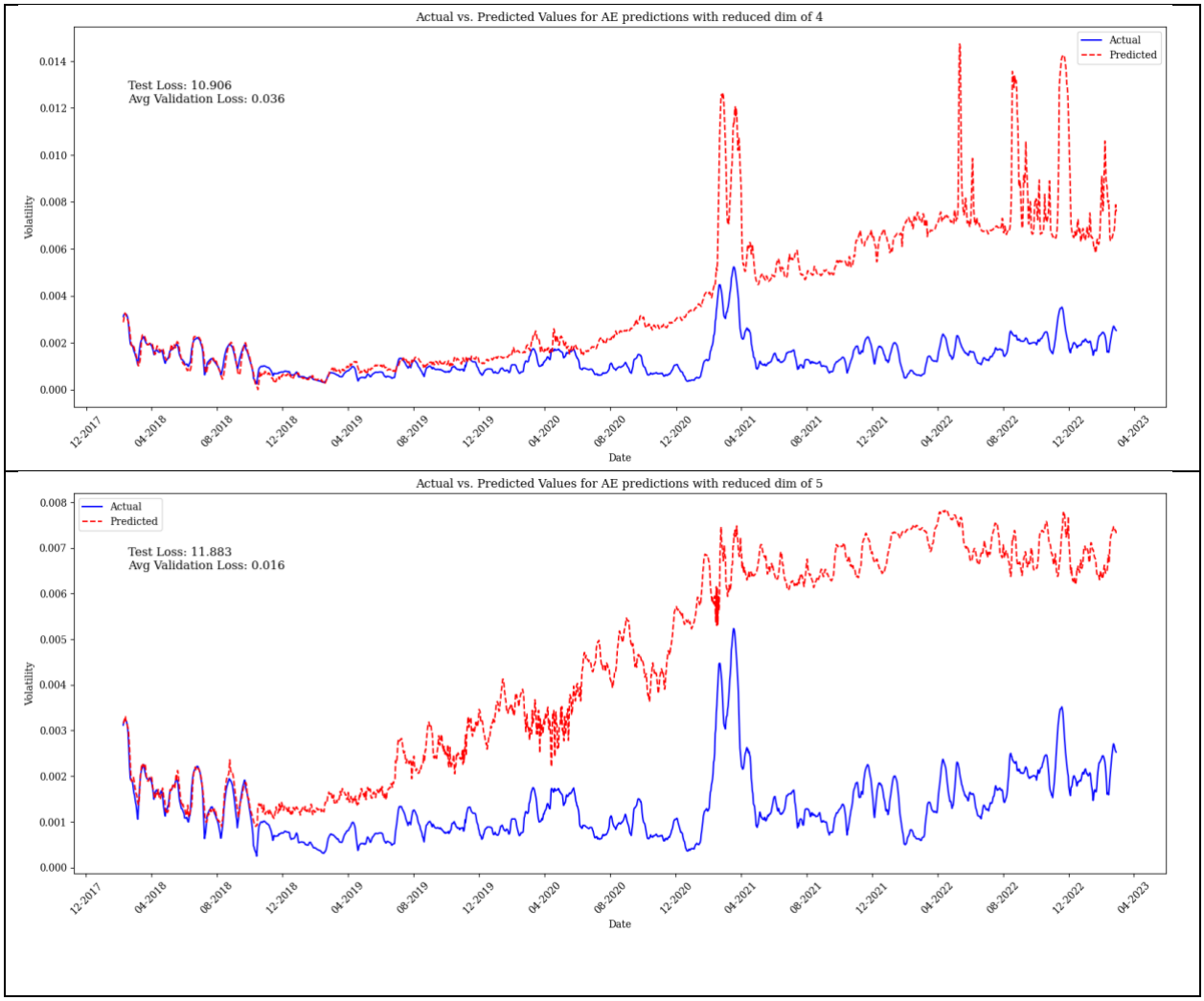
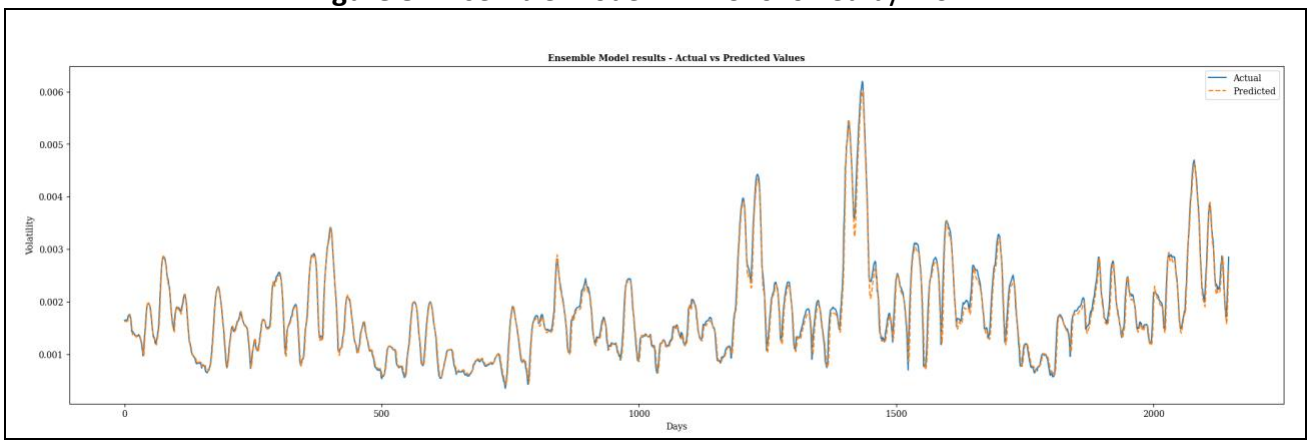
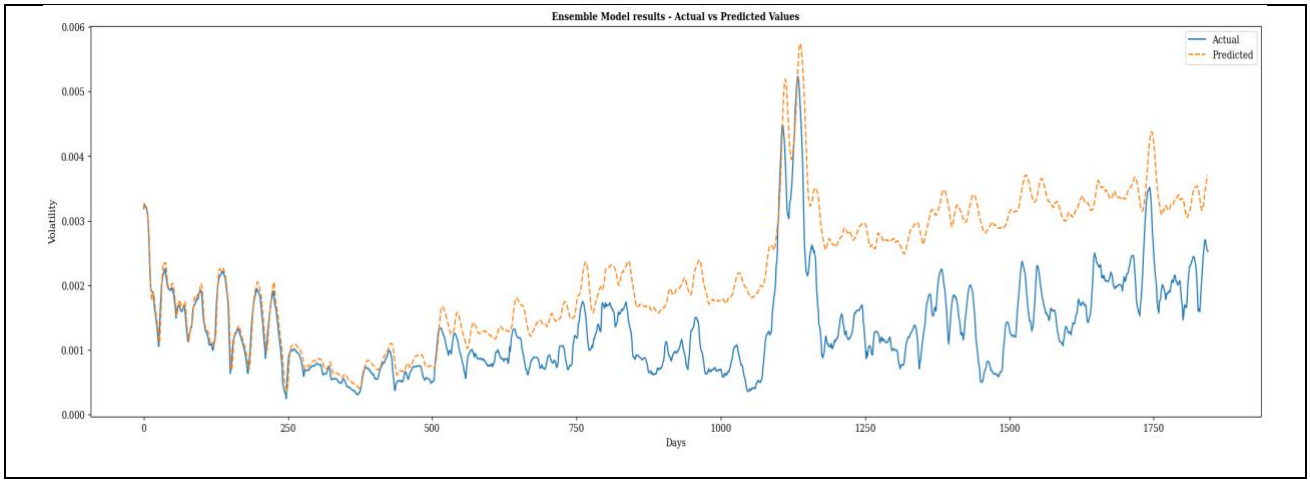


Figure 8: Ensemble Model – INTC followed by MSFT




Ensemble Model results - Actual vs Predicted Values
Actual
Predicted
Volatility
Days
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
250
500
750
1000
1250
1500
1750

References:

[1] Chang, Z., Zhang, Y. and Chen, W. (2019). Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform. *Energy*, 187, p.115804. doi:https://doi.org/10.1016/j.energy.2019.07.134.

[2] Konar, J., Khandelwal, P. and Tripathi, R. (2020). *Comparison of Various Learning Rate Scheduling Techniques on Convolutional Neural Network*. [online] IEEE Xplore. doi:https://doi.org/10.1109/SCEECS48394.2020.94.

[3] alphascientist.com. (n.d.). *Stock Prediction with ML: Walk-forward Modeling — The Alpha Scientist*. [online] Available at: https://alphascientist.com/walk_forward_model_building.html [Accessed 23 Mar. 2023].