

# Predicting Stock Prices with Deep Learning



University of Colorado **Boulder**

**DTSA 5511 Introduction to Deep Learning**

30 Oct 2025

MS-DS, [peculiar.d@colorado.edu](mailto:peculiar.d@colorado.edu)

# 1. INTRODUCTION

Challenging for retail investors to know when to buy or sell in the stock market





## 2. PROBLEM TO SOLVE

To forecast 14 days stock market price  
with traditional technical indicators and  
Deep Learning Model

### 3. PROJECT DATA

- 5 years daily closing prices from Yahoo Finance
- Stocks chosen: SPY S&P 500 Market Index

	Close	High	Low	Open	Volume
count	1510	1510	1510	1510	1510
unique	1497	1508	1510	1509	1510
top	285.3655700683594	283.1889036613753	667.7999877929688	666.8200073242188	65604500
freq	2	2	1	2	1

```
<class 'pandas.core.frame.DataFrame'>
Index: 1511 entries, Ticker to 2025-10-23
Data columns (total 5 columns):
#   Column  Non-Null Count  Dtype
---  -
0   Close   1510 non-null     object
1   High    1510 non-null     object
2   Low     1510 non-null     object
3   Open    1510 non-null     object
4   Volume  1510 non-null     object
dtypes: object(5)
memory usage: 70.8+ KB
```

## 4. DATA CLEANING

- Correct the header row
- Convert to price column to numerical
- Fix the data type
- Set the date column as index & sort
- Verified no missing data

```
<class 'pandas.core.frame.DataFrame'>  
Index: 1509 entries, 2019-10-23 to 2025-10-23  
Data columns (total 5 columns):  
#   Column  Non-Null Count  Dtype  
---  ---  
0   Close   1509 non-null   float64  
1   High    1509 non-null   float64  
2   Low     1509 non-null   float64  
3   Open    1509 non-null   float64  
4   Volume  1509 non-null   int64  
dtypes: float64(4), int64(1)  
memory usage: 70.7+ KB  
None
```

## 5. TARGET & FEATURE SELECTION

- **Target:** Close price
- **Features:**

- Close

- EMA 20

$$\text{EMA}_t = \text{Close}_t \times \alpha + \text{EMA}_{t-1} \times (1 - \alpha)$$

- RSI 14

$$RS = \frac{\text{Avg Gain}}{\text{Avg Loss}} \quad \text{RSI}_{14} = 100 - \left( \frac{100}{1 + RS} \right)$$

- Daily Returns

$$R_t = \frac{\text{Close}_t - \text{Close}_{t-1}}{\text{Close}_{t-1}}$$

- Volatility

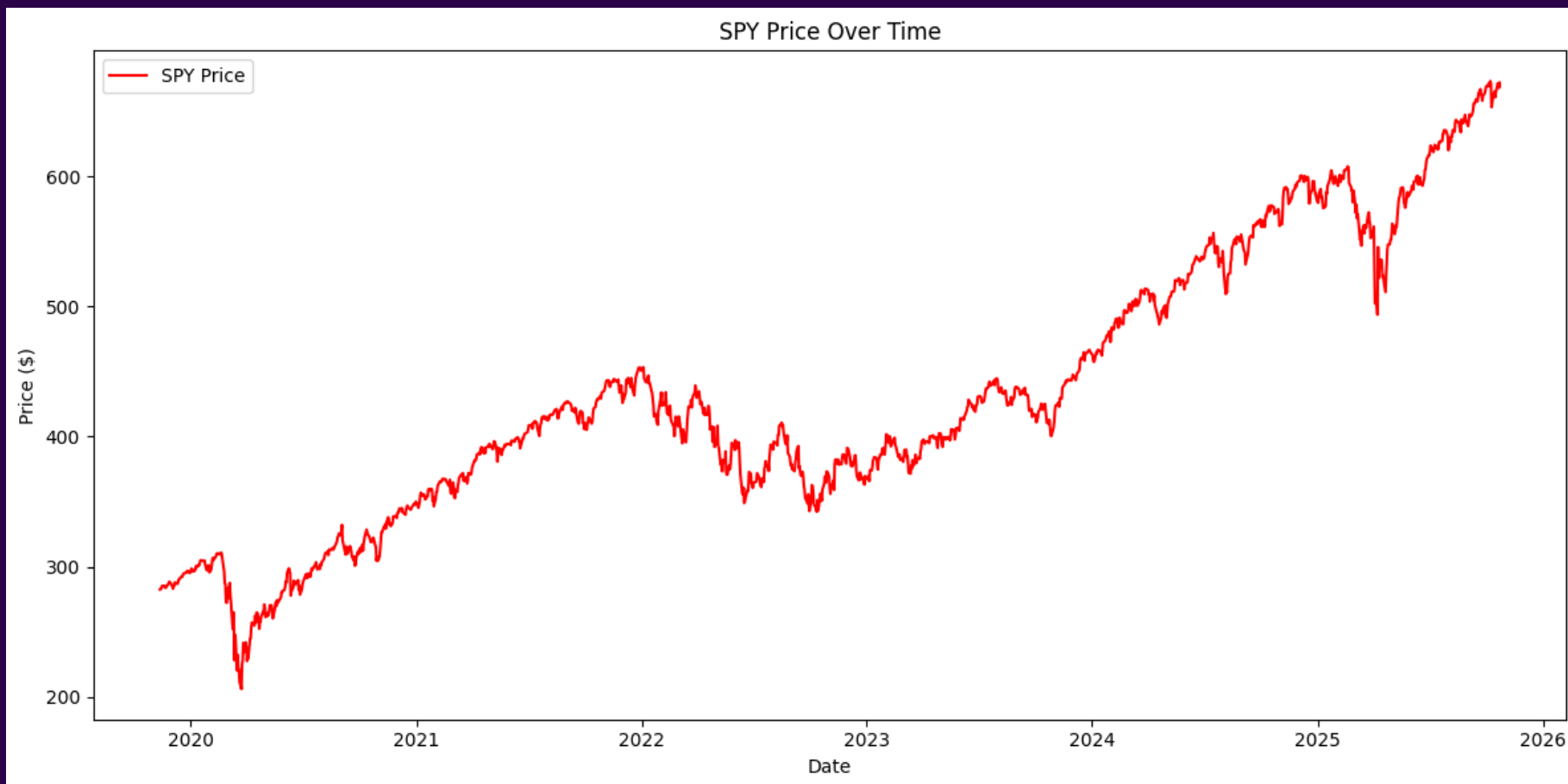
$$\text{Volatility}_t = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (r_{t-i} - \bar{r})^2}$$

- Volume

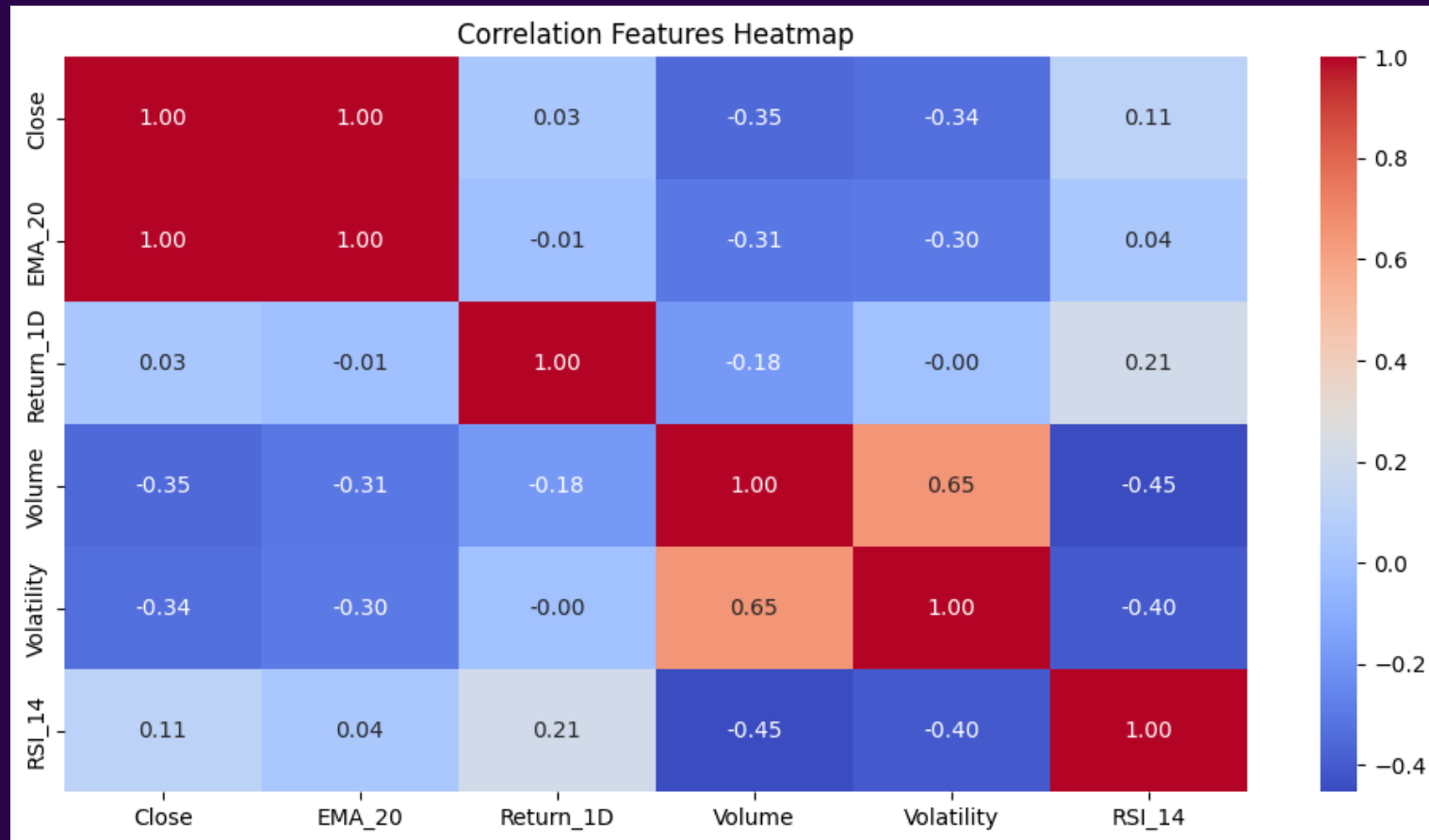
## 5. TARGET & FEATURE SELECTION



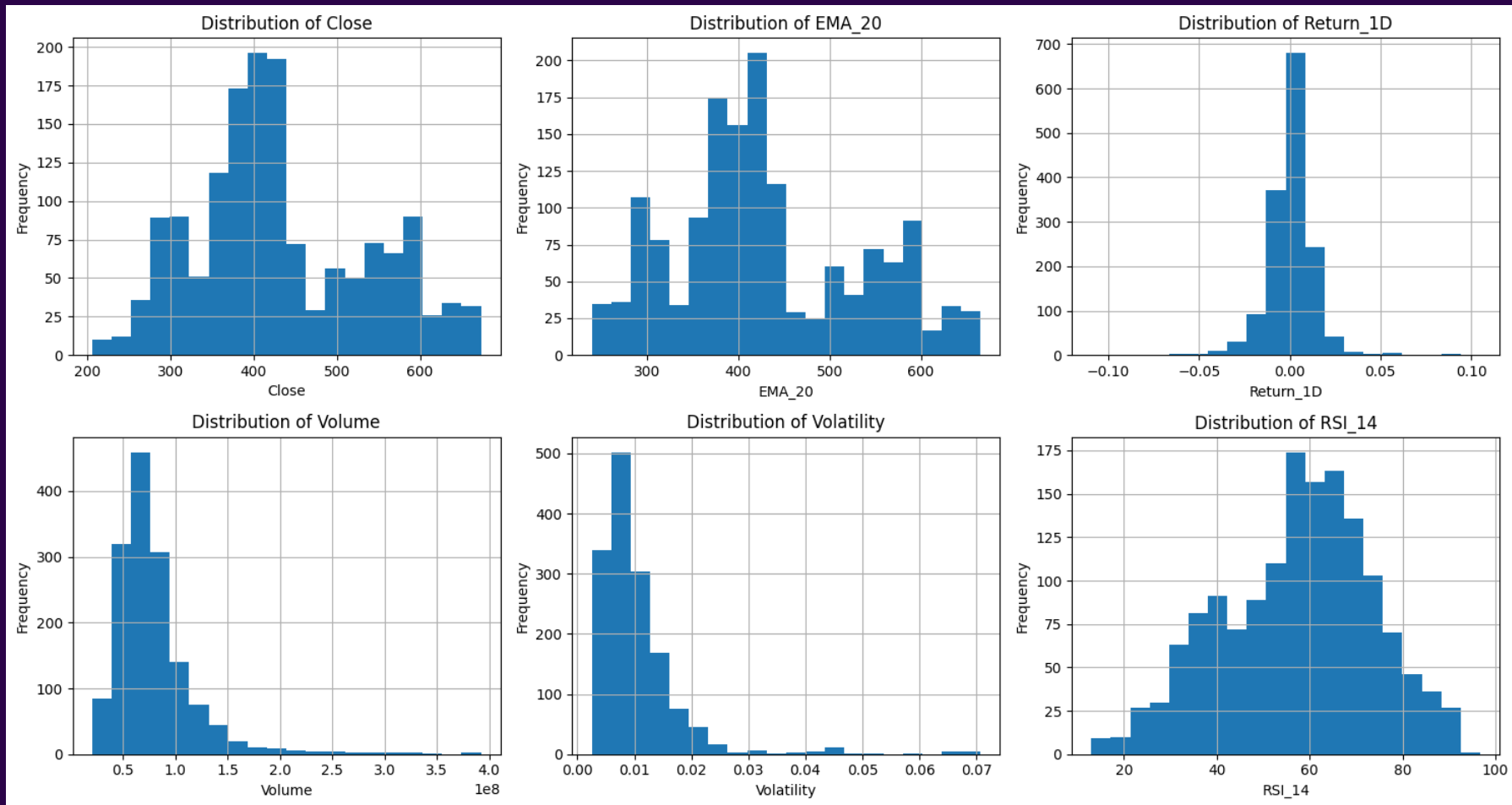
## 6. EXPLORATORY DATA ANALYSIS



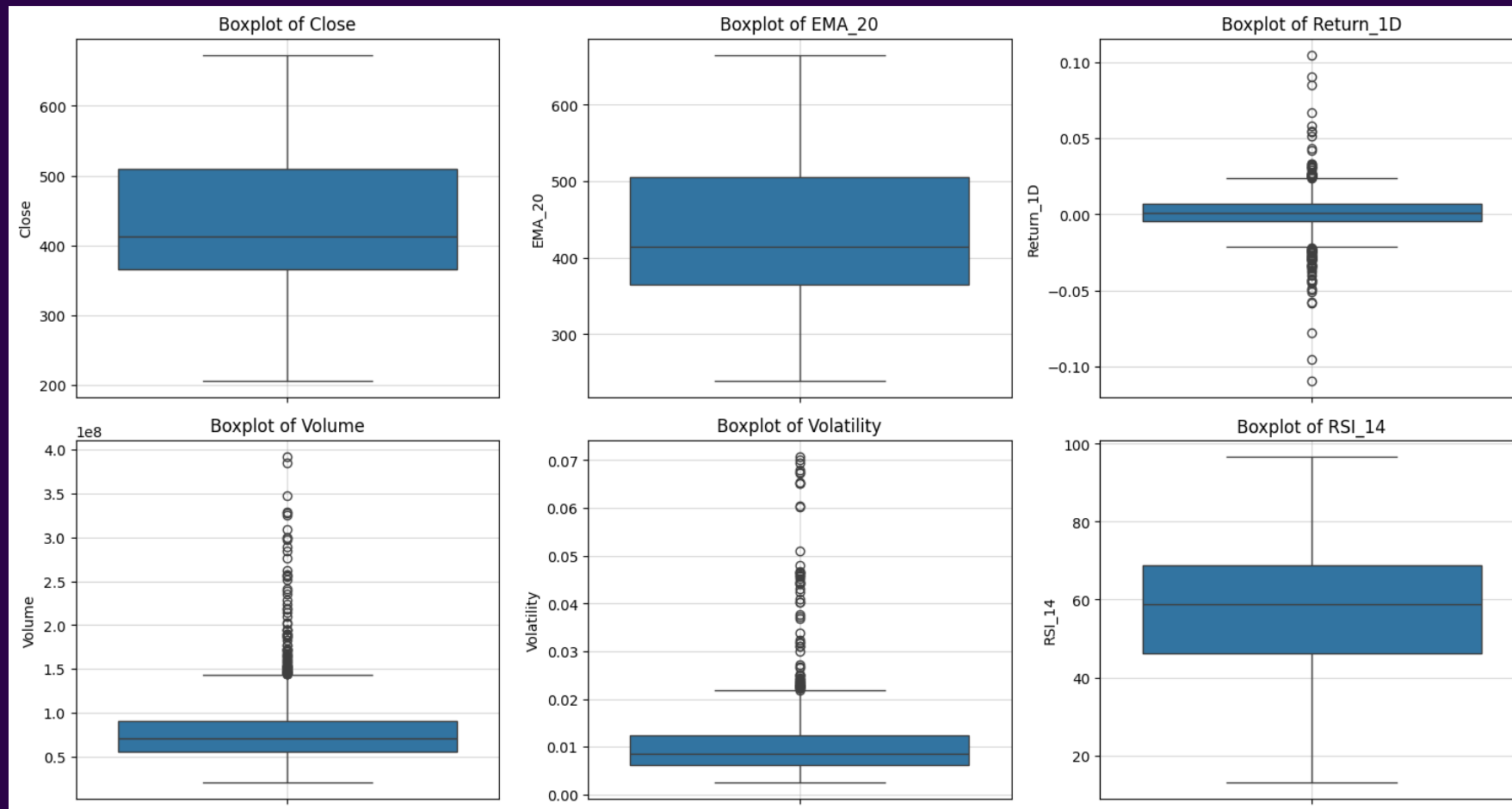
## 6. EXPLORATORY DATA ANALYSIS



## 6. EXPLORATORY DATA ANALYSIS



## 6. EXPLORATORY DATA ANALYSIS



## 7. DATA PREPROCESSING & POST CLEANING

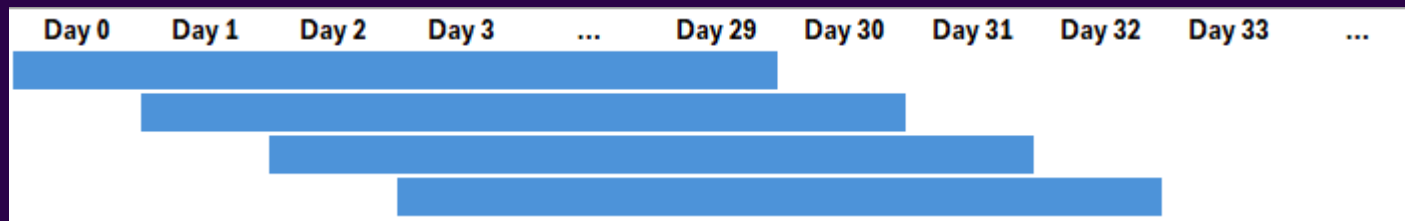
- Log Transformation for Volume

$$\text{Volume}_{\log} = \log(\text{Volume} + 1)$$

- MinMaxScaler to maintain proper scaling

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

- Sliding Window
  - model uses blocks of “sliding window” to predict the subsequent day’s return

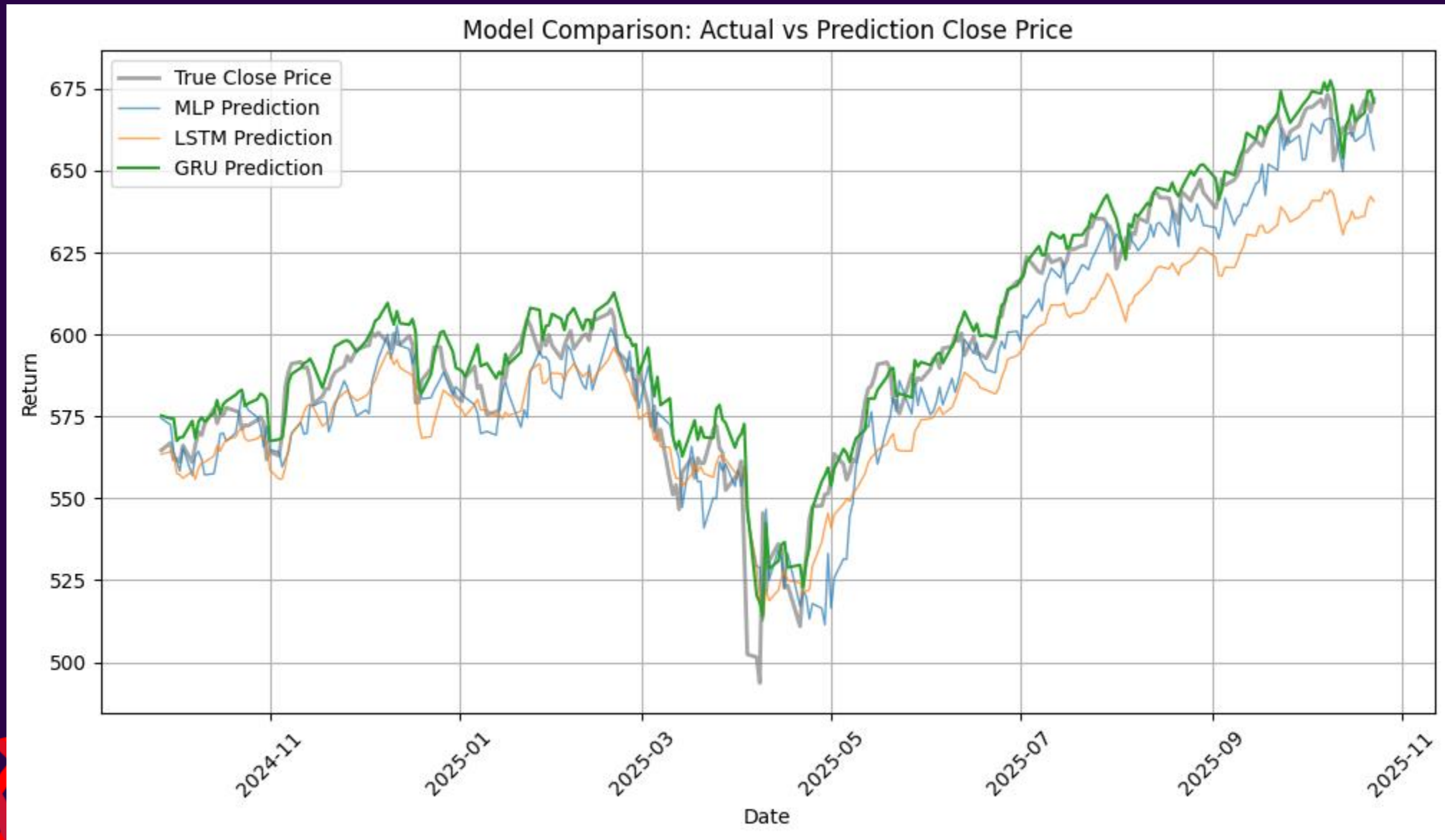


## 8. MODEL ARCHITECTURE

- Multilayer Perceptron (MLP)
- Long Short-Term Memory (LSTM)
- Gated Recurrent Unit (GRU)

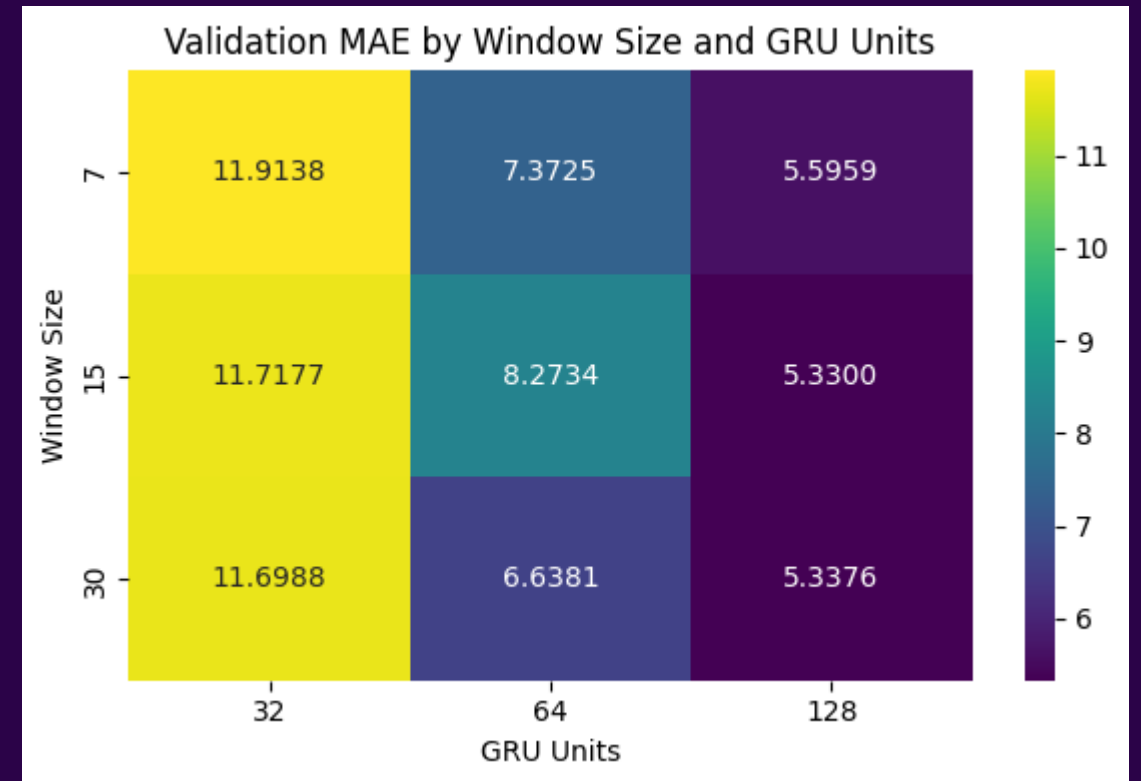
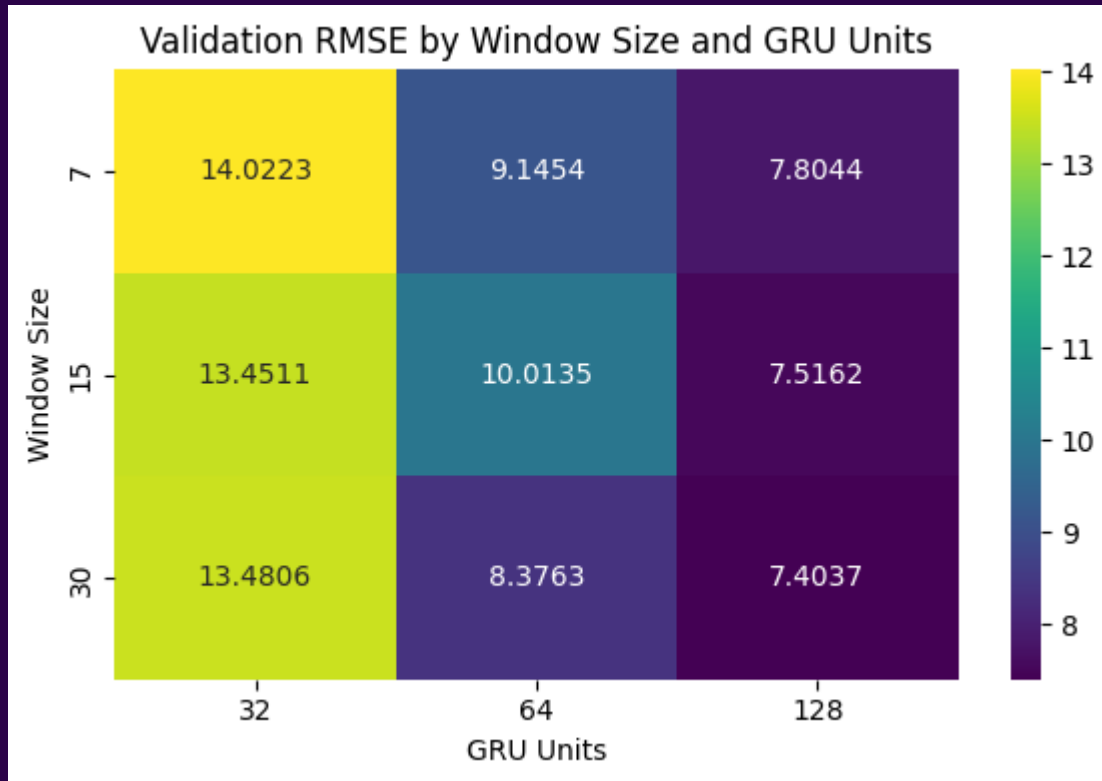
Model	RMSE	MAE
MLP	12.8487	10.0962
LSTM	17.0502	14.6350
GRU	<b>8.7818</b>	<b>6.4162</b>

## 8. MODEL ARCHITECTURE



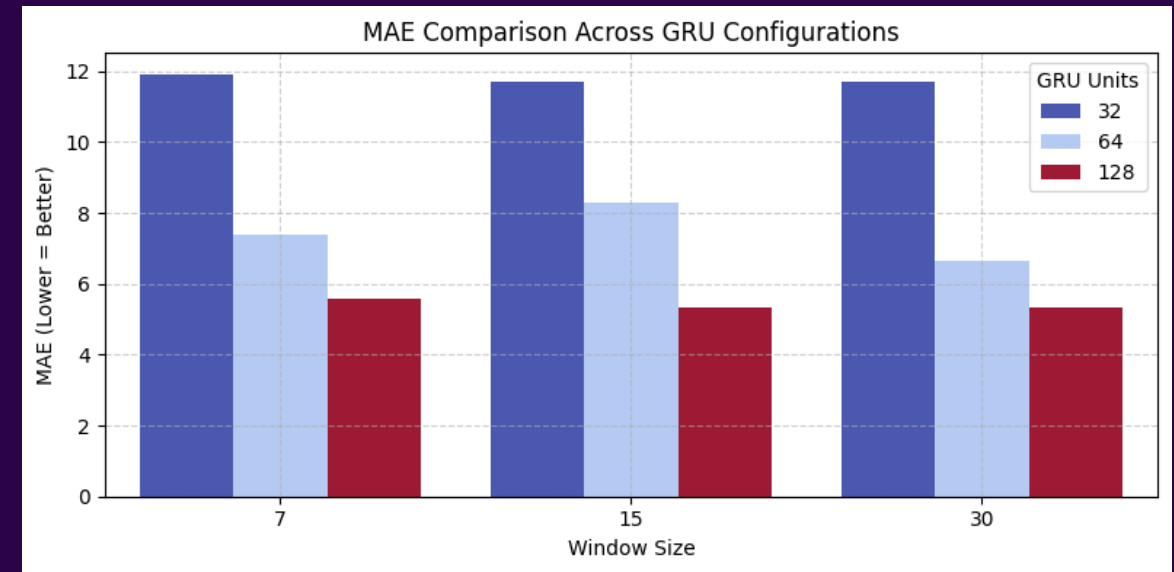
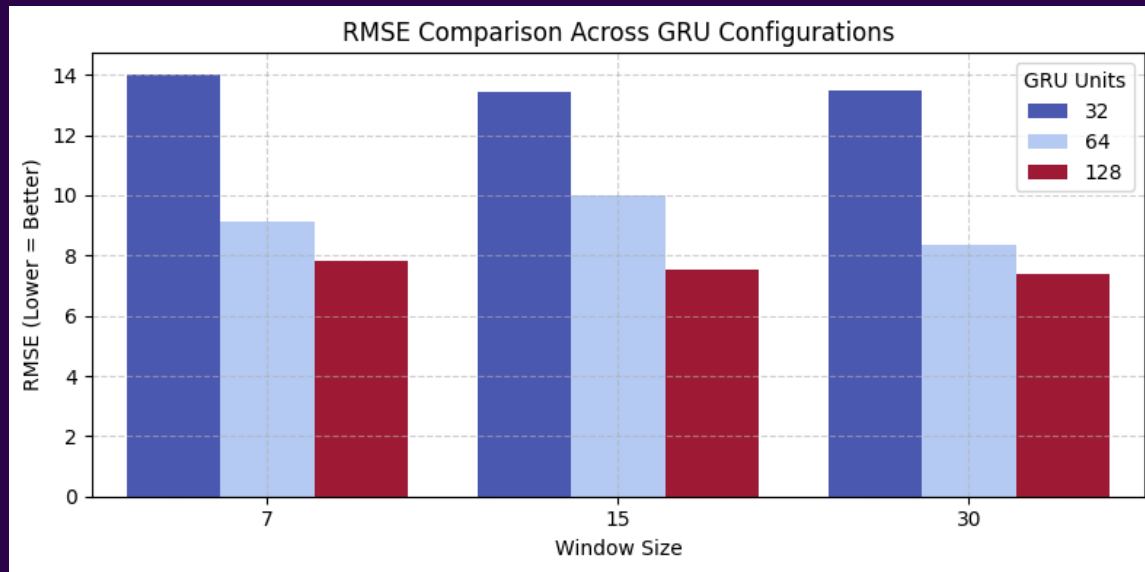
# 8. MODEL ARCHITECTURE

## Hyperparameter Tuning



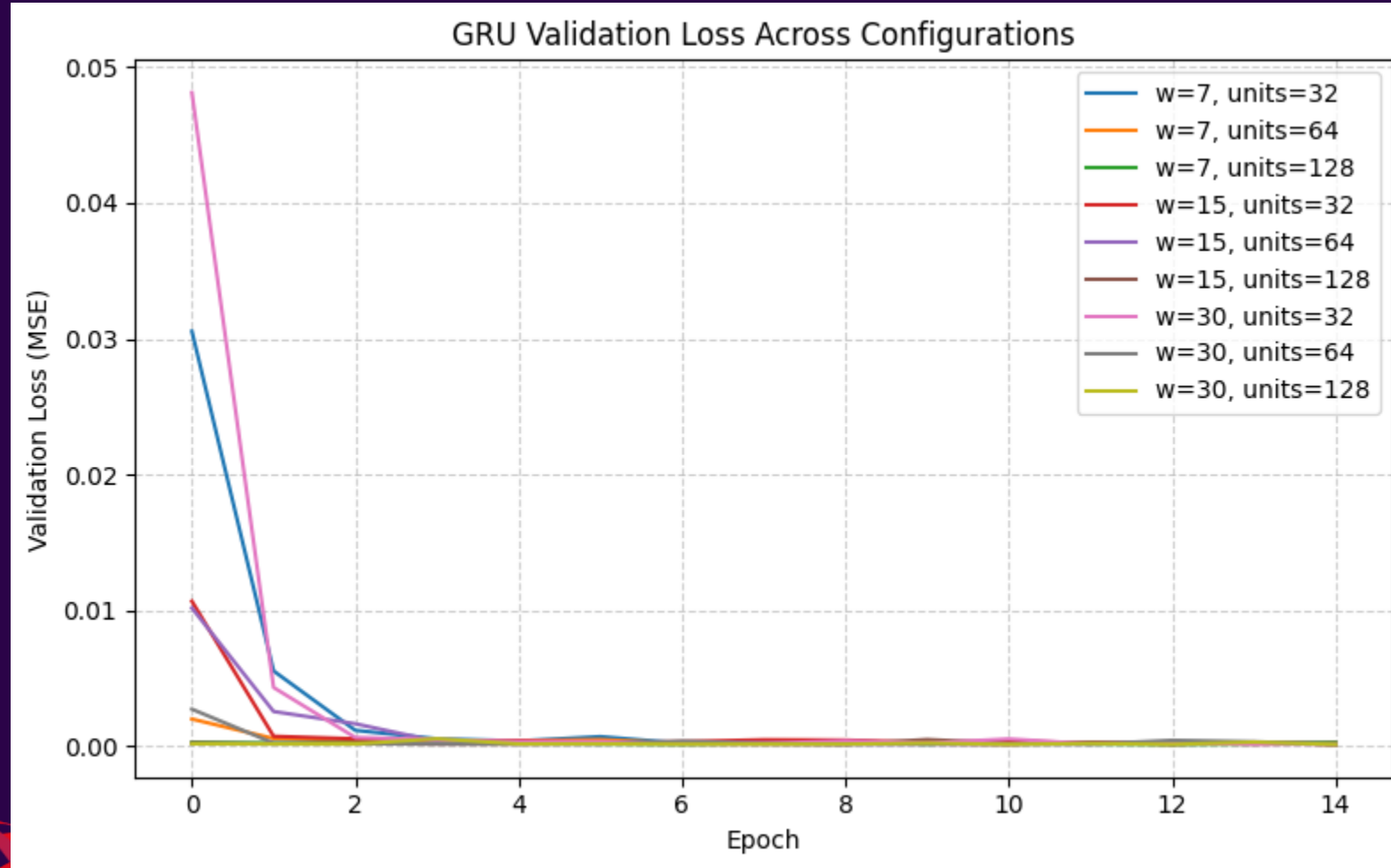
# 8. MODEL ARCHITECTURE

## Hyperparameter Tuning



# 8. MODEL ARCHITECTURE

## Validation Loss



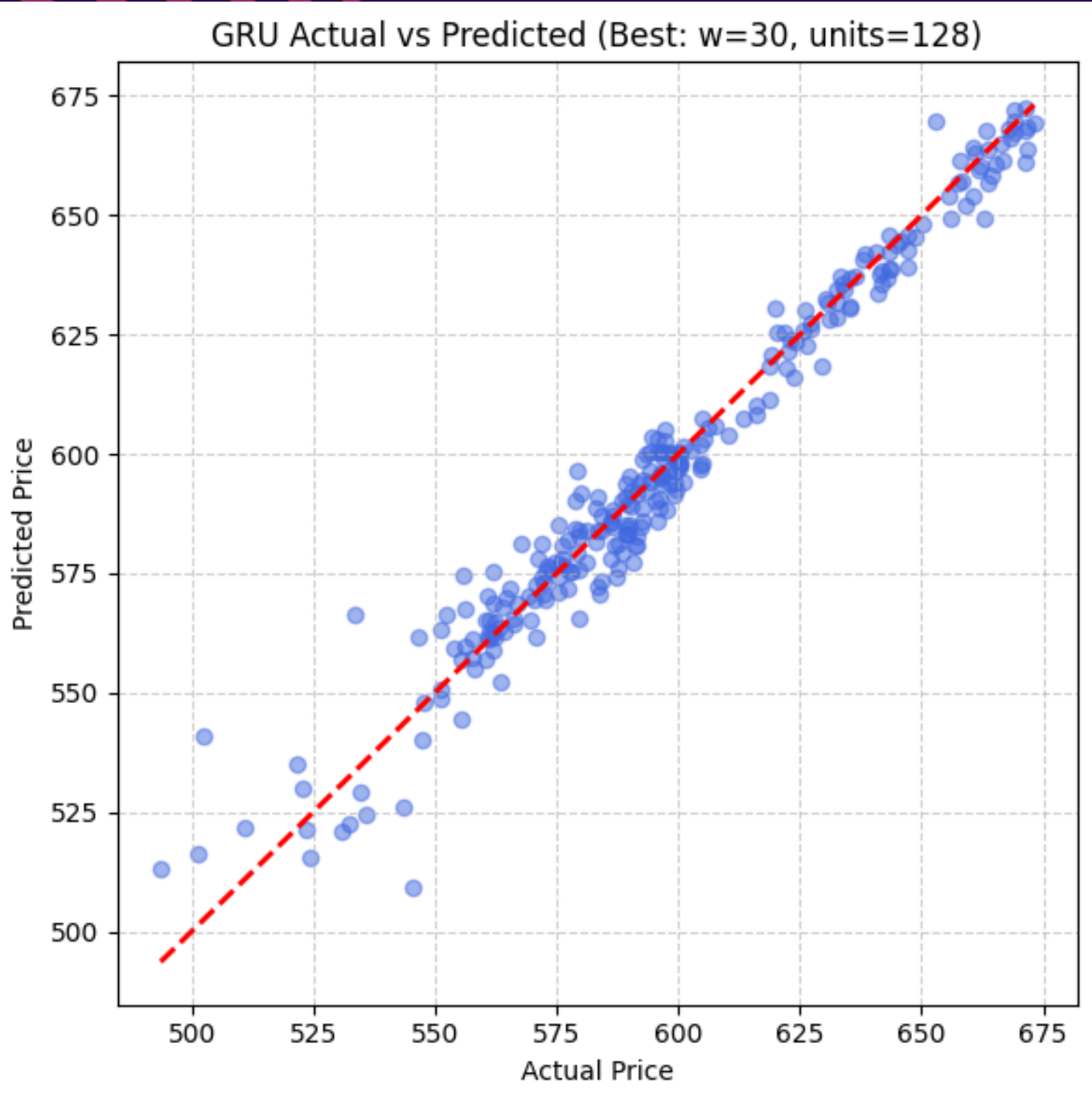
## 9. RESULT AND ANALYSIS

### MODEL COMPARISON

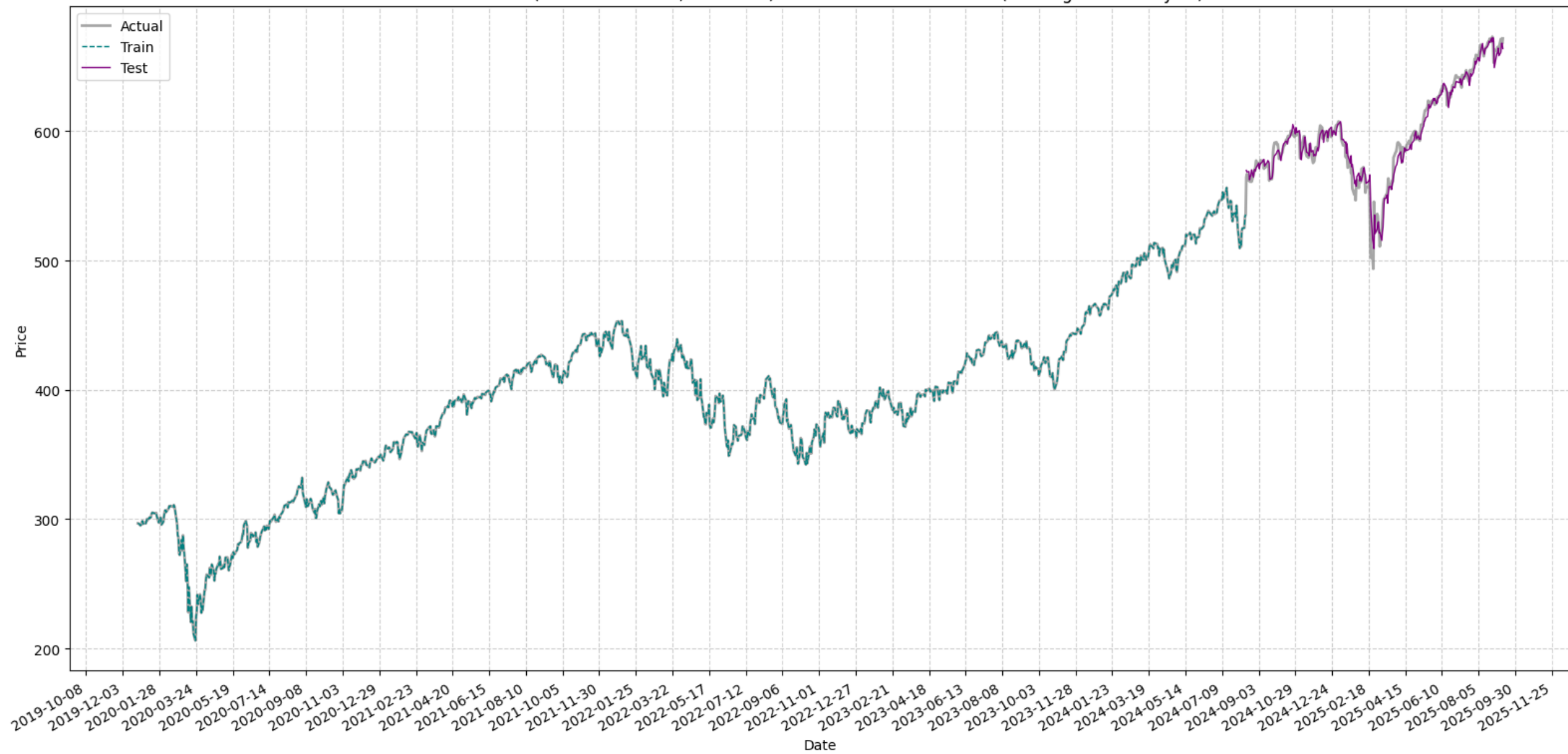
Model	RMSE	MAE
MLP	12.8487	10.0962
LSTM	17.0502	14.6350
GRU	<b>8.7818</b>	<b>6.4162</b>

### TOP 3 HYPERPARAMETER

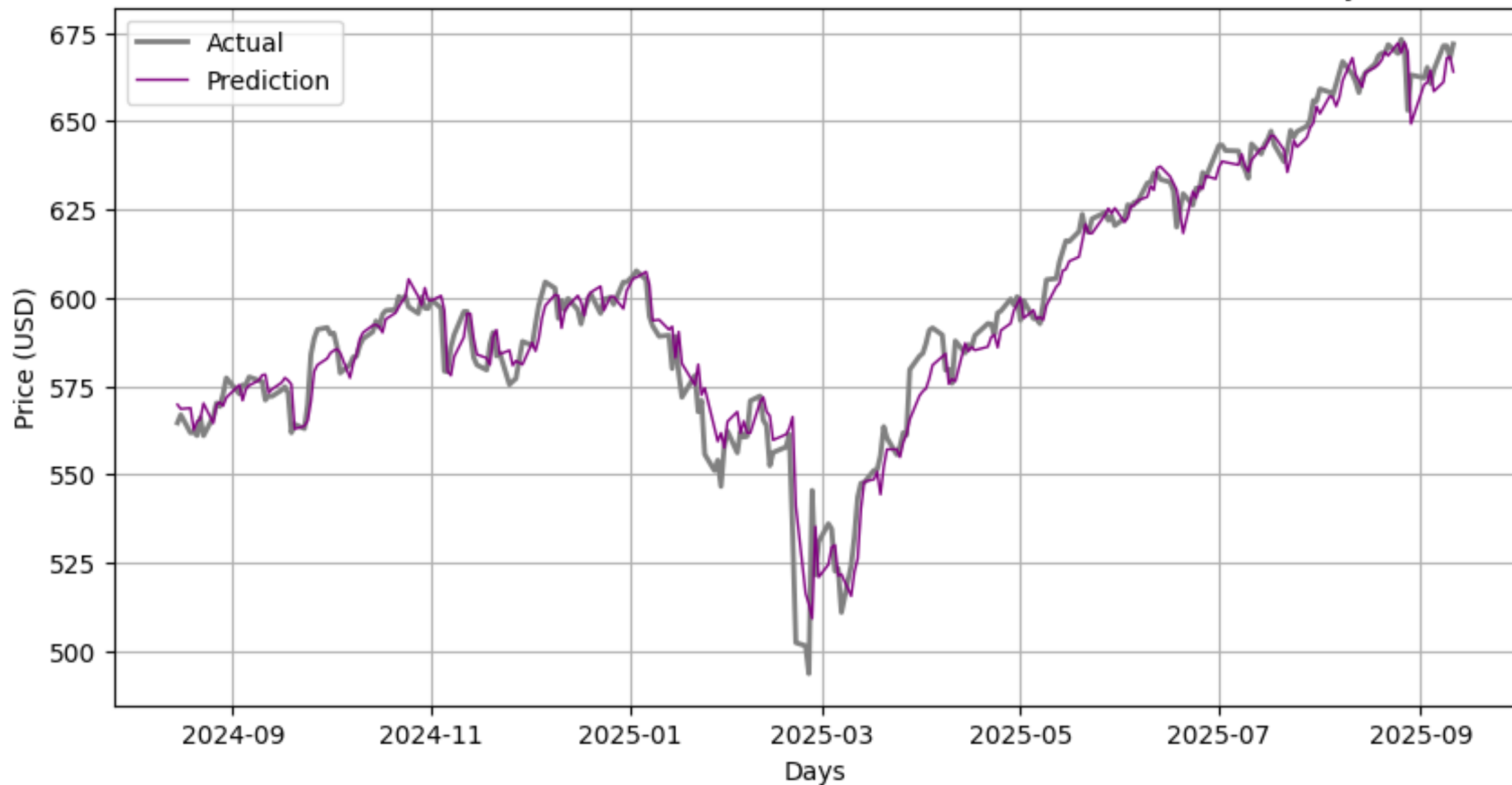
Window	Units	RMSE	MAE
30	128	<b>7.4037</b>	<b>5.3376</b>
15	128	7.5162	5.3300
7	128	7.8044	5.5959



GRU Model (Window Size=30, Unit=128): Actual vs Predicted Price (Training and Test Cycle)



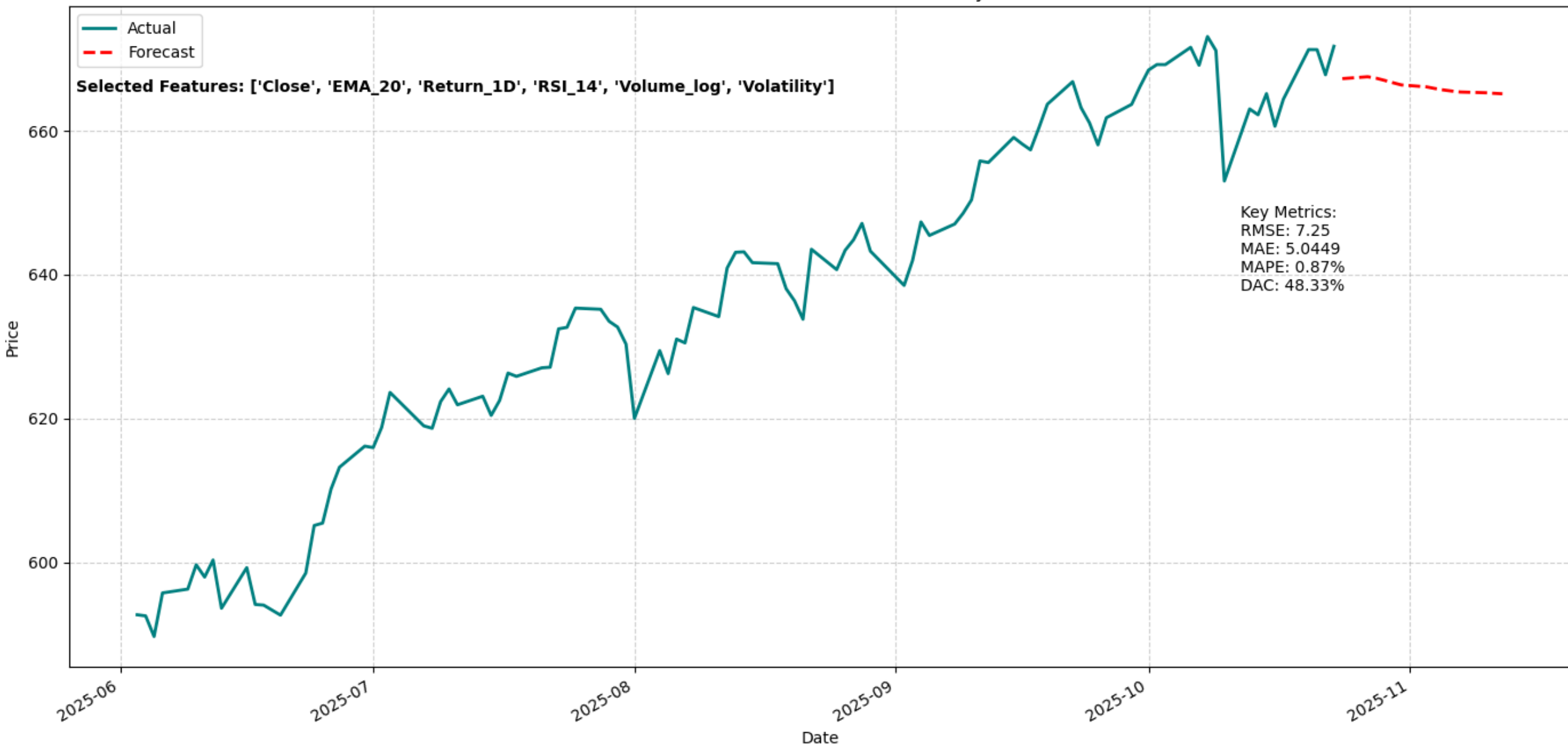
GRU Model (Window Size=30, Unit=128): Actual vs. Predicted Price (Test Cycle)



# PERFORMANCE SCORE

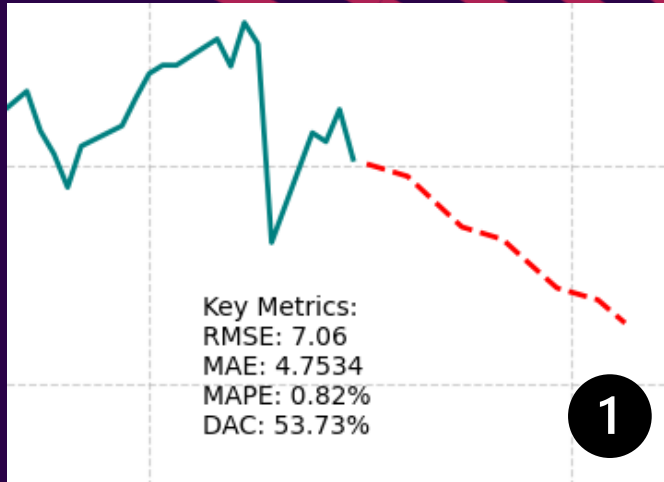
ROOT MEAN SQUARED ERROR (RMSE)	7.25
MEAN ABSOLUTE ERROR (MAE)	5.04
MEAN ABSOLUTE PERCENTAGE ERROR (MAPE)	0.87%
DIRECTIONAL ACCURACY (DAC)	48.33%

GRU Forecast Next 14 Business Days

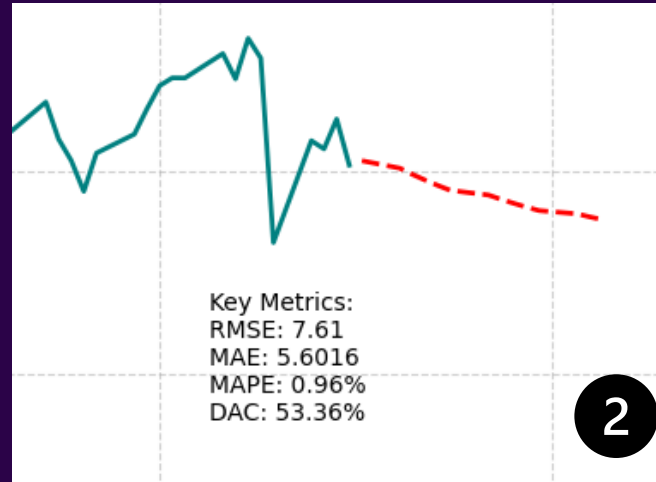


# Feature Comparison in Forecasting and Accuracy Scores

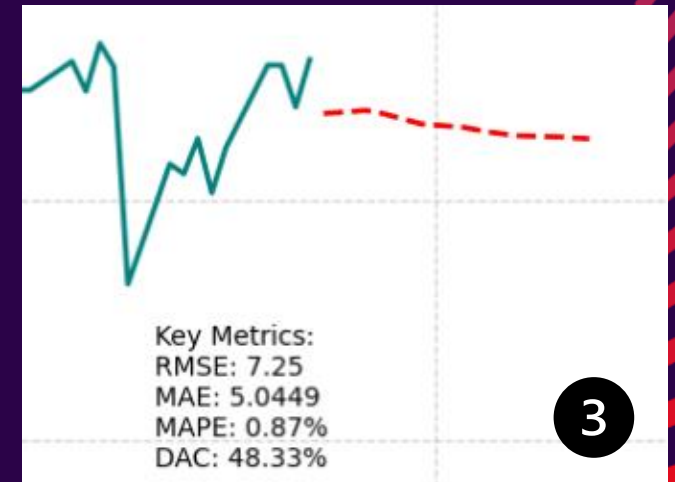
['Close']



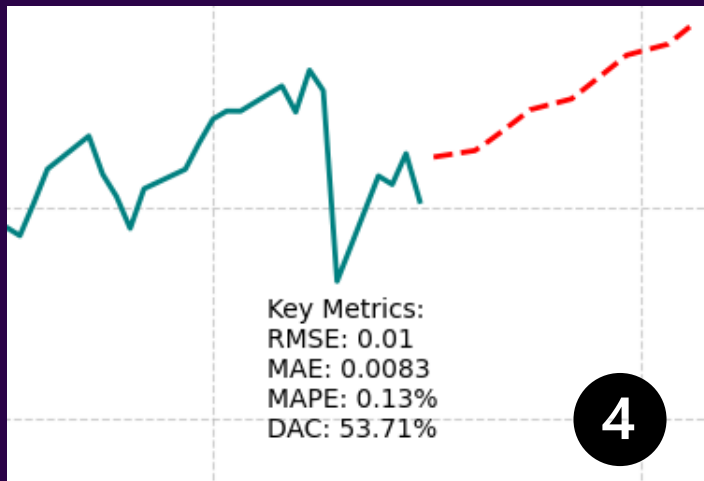
['Close', 'EMA\_20']



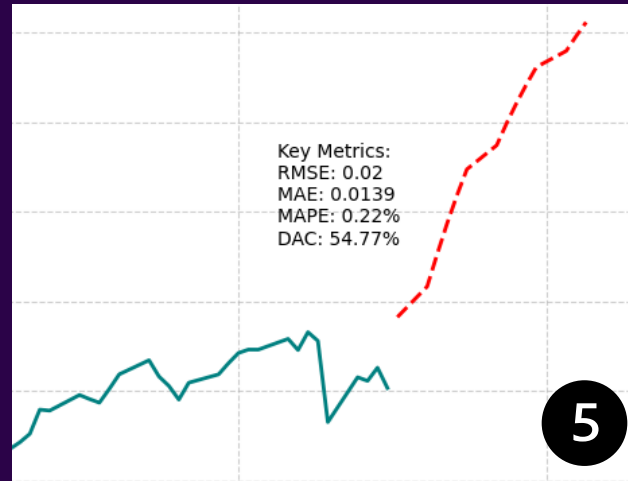
['Close', 'EMA\_20', 'Return\_1D',  
'RSI\_14', 'Volume\_log', 'Volatility']



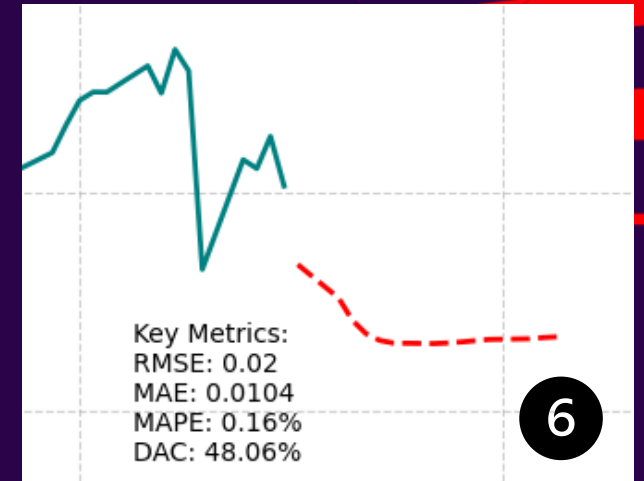
['Close\_log']



['Close\_log', 'EMA\_20\_log']

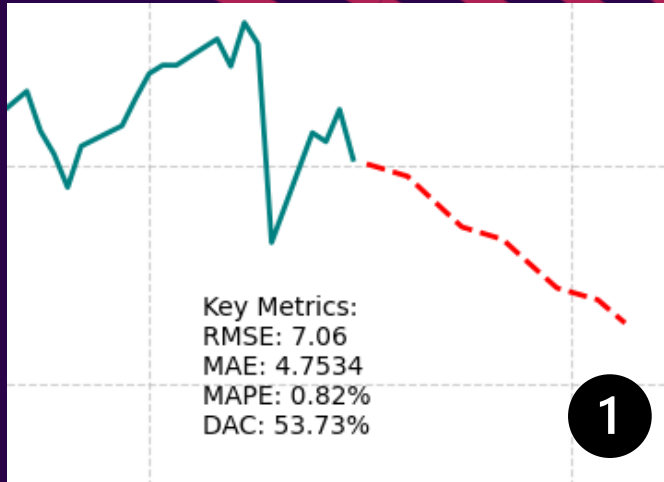


['Close\_log', 'EMA\_20\_log', 'Return\_1D',  
'RSI\_14', 'Volume\_log', 'Volatility']

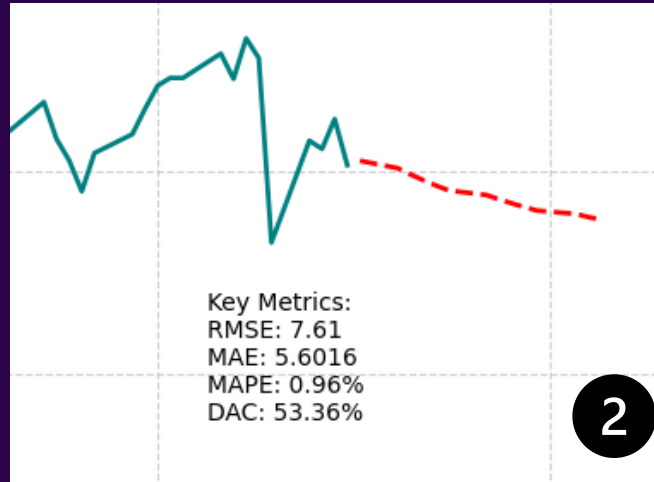


# Feature Comparison in Forecasting and Accuracy Scores

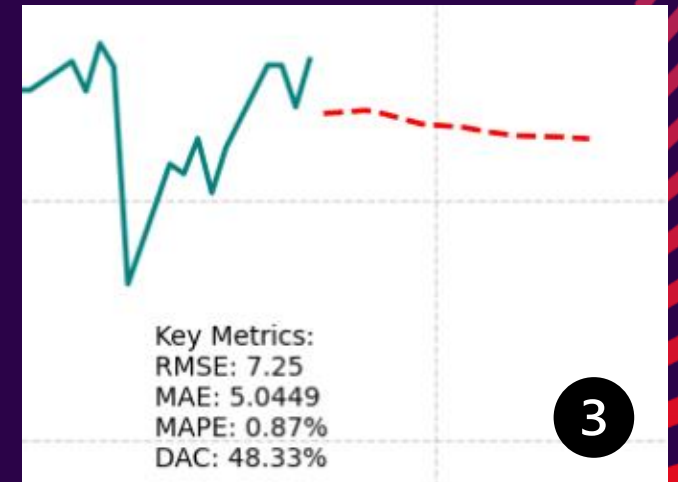
['Close']



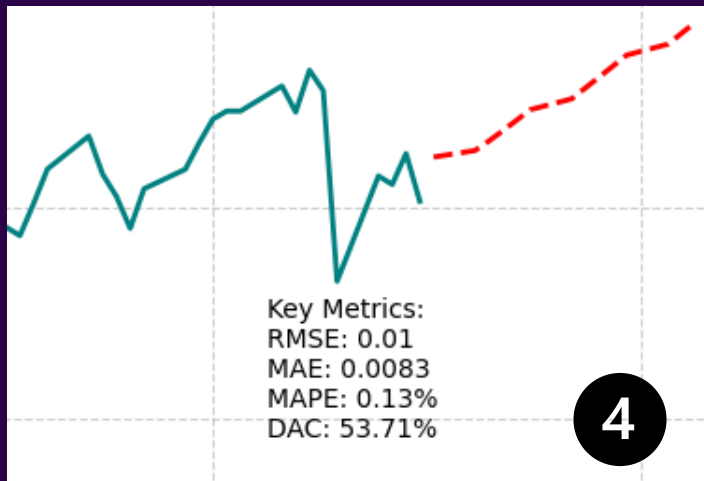
['Close', 'EMA\_20']



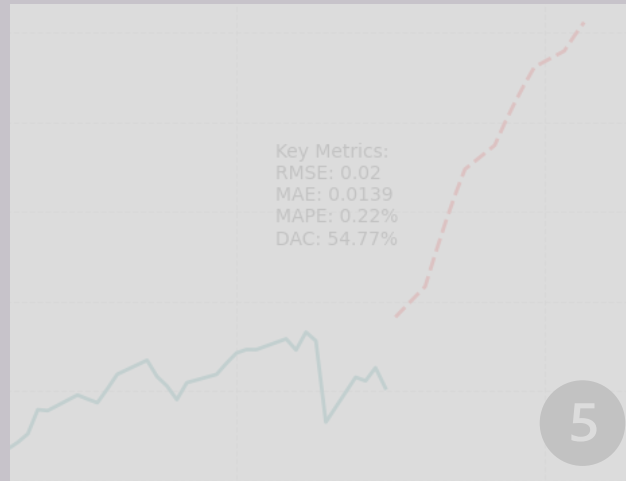
['Close', 'EMA\_20', 'Return\_1D',  
'RSI\_14', 'Volume\_log', 'Volatility']



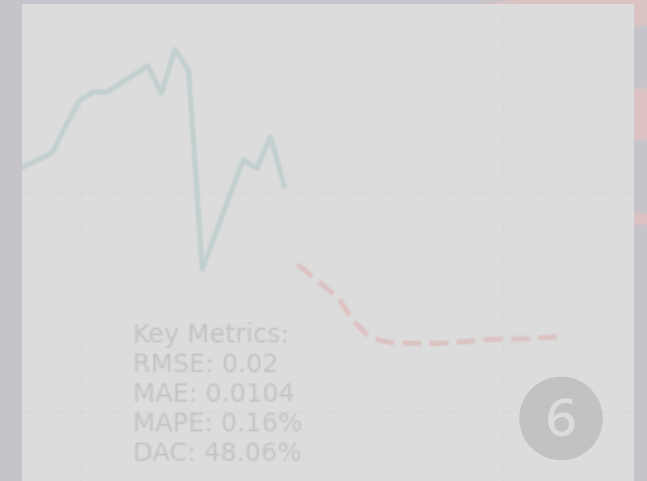
['Close\_log']



['Close\_log', 'EMA\_20\_log']



['Close\_log', 'EMA\_20\_log', 'Return\_1D',  
'RSI\_14', 'Volume\_log', 'Volatility']



## 10. DISCUSSION & CONCLUSION

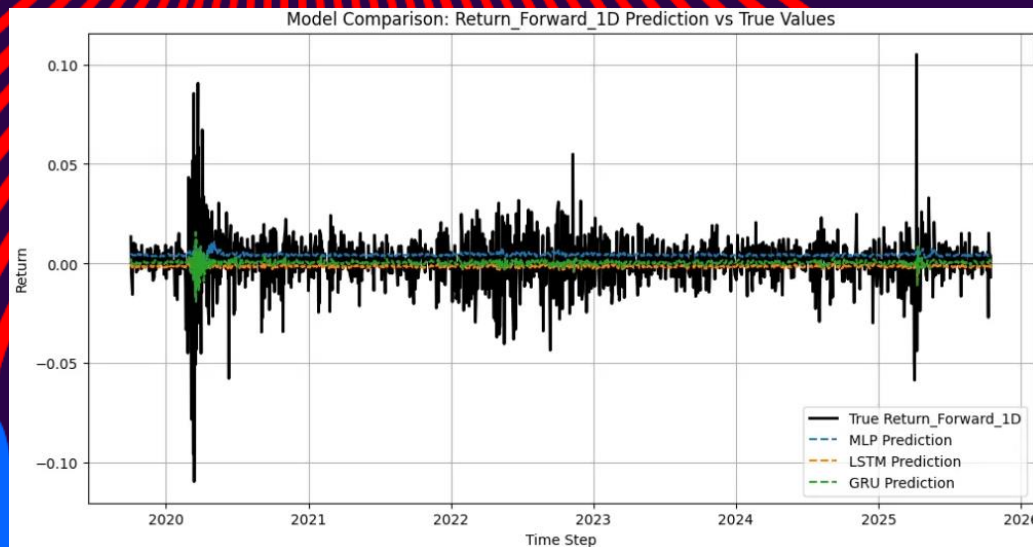
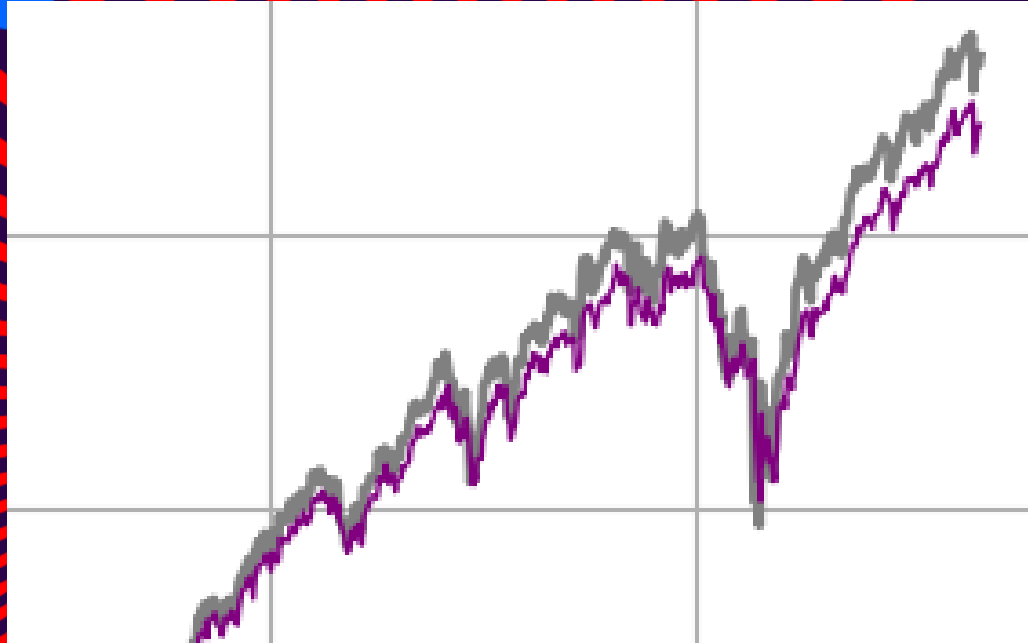
### Key Lessons:

- Directional Accuracy (DAC) (Yin & Zhao, 2023)
- Traditional technical indicators
- Strict data separation

## 10. DISCUSSION & CONCLUSION

### Limitation:

- GRU model underperformed
- Too much historical data
- Single-day returns



## 10. DISCUSSION & CONCLUSION

### Areas for Improvement:

- Hybrid and context-aware architectures (Zhu, 2025)
- Deep Reinforcement Learning (Liu, 2022)

# THANK YOU

# REFERENCES

- Brownlee, J. (2016, August 4). Time series prediction with LSTM recurrent neural networks in Python with Keras. Machine Learning Mastery. <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>
- Brownlee, J. (2020, July 15). Classification versus regression in machine learning. Machine Learning Mastery. <https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/>
- GRU layer. (n.d.). Keras. Retrieved October 28, 2025, from [https://keras.io/api/layers/recurrent\\_layers/gru/](https://keras.io/api/layers/recurrent_layers/gru/)
- Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and Practice (3rd ed.). OTexts. <https://otexts.com/fpp3/>
- Image classification using an MLP. (n.d.). Keras. Retrieved October 28, 2025, from [https://keras.io/examples/vision/mlp\\_image\\_classification/](https://keras.io/examples/vision/mlp_image_classification/)
- Liu, X. Y., Yang, H., Chen, Q., Yang, R., Zhang, R., Yang, L., Xiao, B., & Wang, C. D. (2022). FinRL: A deep reinforcement learning library for automated trading in quantitative finance. Applied Soft Computing, 118, Article 108486.
- LSTM layer. (n.d.). Keras. Retrieved October 28, 2025, from [https://keras.io/api/layers/recurrent\\_layers/lstm/](https://keras.io/api/layers/recurrent_layers/lstm/)
- Ohliati, J., & Yuniarty. (2024). Deep Learning for Stock Market Prediction: A Review. In 2024 International Conference on Information Management and Technology (ICIMTech). IEEE. <https://ieeexplore.ieee.org/document/10780931>
- Yin, T., & Zhao, C. (2023). Enhancing directional accuracy in stock closing price value prediction using a direction-integrated MSE loss function. Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023), 2, 77-88.
- Yulistiani, R., & Kurniadi, F. I. (2024). Stock Price Prediction With the Informer Model. In 2024 International Conference on Information Management and Technology (ICIMTech). IEEE. <https://ieeexplore.ieee.org/document/10780913>
- Zhu, P., Li, Y., Hu, Y., Xiang, S., Liu, Q., Cheng, D., & Liang, Y. (2025). MCI-GRU: Stock prediction model based on multi-head cross-attention and improved GRU. arXiv preprint arXiv:2410.20679.
- Wikipedia contributors. (n.d.). \*Long short-term memory\*. Wikipedia. [https://en.wikipedia.org/wiki/Long\\_short-term\\_memory](https://en.wikipedia.org/wiki/Long_short-term_memory)

# GITHUB REPOSITORY LINK

<https://github.com/peculiardatabits/DTSA-5511-Deep-Learning-Final-Project.git>