# Analysis of Machine and Deep Learning Models for Stock Prediction

Saptarshi Mukherjee[†] Ishan Chandra Gupta[†] Soumya Savarn[†]

## Abstract

This report explains our project on predicting stock prices using Machine Learning and Deep Learning models. We explore standard as well as novel approaches, and find out which is the best approach to use for each sector and index.

## 1. Introduction

The objective of this project is to explore and compare the performance of various machine learning (ML) and deep learning (DL) models for time-series forecasting, with a specific focus on stock market data. The study involves training models, including some incorporating novel architectures, to predict the closing prices of the Dow Jones Industrial Average. These models are further applied to other financial indices, such as the S&P 500, as well as individual stocks from the technology and healthcare sectors. The analysis includes varying the historical input window (e.g., 10, 30, and 60 days) to evaluate how the length of historical data impacts predictive accuracy. Additionally, the project investigates whether the optimal model depends on the industry the stock belongs to, examining how sector-specific patterns influence model performance. The ultimate goal is to identify the most effective modeling approach for time-series forecasting in financial markets by benchmarking their performance across different datasets, industries, and contexts.

## 2. Methods

### 2.1. LSTM (Long Short-Term Memory)

LSTM is a type of recurrent neural network (RNN) that addresses the vanishing gradient problem by using memory cells to capture long-term dependencies. It has three gates (input, forget, and output) to control the flow of information. LSTMs are widely used in time-series tasks like stock price prediction due to their ability to retain relevant temporal patterns.

### 2.2. GRU (Gated Recurrent Unit)

GRU is a simplified version of LSTM that uses fewer gates (update and reset gates) while still effectively capturing temporal dependencies. It is computationally more efficient and faster to train compared to LSTM. GRUs are ideal for time-series tasks where speed is critical without sacrificing much performance.

### 2.3. Bidirectional LSTM

Bidirectional LSTM processes sequences in both forward and backward directions, allowing the model to learn context from both past and future data points. This makes it especially powerful for tasks requiring full sequence information, such as stock trend analysis or language modeling. However, it comes with increased computational cost.

### 2.4. LSTM-GRU Hybrid

The LSTM-GRU hybrid combines the strengths of both LSTM and GRU layers, leveraging the long-term memory capabilities of LSTMs and the efficiency of GRUs. It is particularly useful for tasks where capturing diverse temporal dependencies is critical. This hybrid model can balance complexity and training speed effectively.

### 2.5. Bidirectional LSTM-GRU Hybrid

This model integrates bidirectional LSTM and GRU layers to capture temporal dependencies from both directions while benefiting from GRU's efficiency. It excels in tasks requiring contextual understanding of time-series data, such as market trend forecasting, but is more resource-intensive.

### 2.6. LSTM-CNN Hybrid

The LSTM-CNN hybrid combines convolutional layers for local feature extraction with LSTM layers for capturing sequential dependencies. CNNs detect short-term patterns, while LSTMs analyze temporal dependencies. This model is effective for time-series tasks where both spatial and temporal patterns are important.

### 2.7. GRU-CNN Hybrid

Similar to the LSTM-CNN hybrid, the GRU-CNN hybrid uses CNNs for feature extraction and GRUs for sequential modeling. It is faster to train compared to the LSTM-CNN hybrid and is useful for tasks requiring a balance of efficiency and accuracy. It's ideal for large-scale time-series datasets with temporal and spatial patterns.

### 2.8. Bidirectional LSTM-CNN hybrid

The CNN-BiLSTM hybrid integrates convolutional layers for spatial feature extraction with Bidirectional LSTMs for enhanced sequential modeling. By processing input sequences in both forward and backward directions, the BiLSTM captures bidirectional temporal dependencies, complementing the CNN's ability to identify spatial patterns. This model excels in tasks requiring deep temporal insights alongside spatial analysis, making it ideal for complex time-series datasets with intricate dependencies.

### 2.9. DTR (Decision Tree Regression)

Decision tree regression is a machine learning technique that uses a tree-like model to predict numerical values. It's a type of supervised learning algorithm that uses a flowchart-like structure to classify or regress data. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

### 2.10. Random Forest Regressor

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that works by creating a multitude of decision trees during training. For regression tasks, the output is the average of the predictions of the trees. Random forests correct for decision trees' habit of overfitting to their training set. While decision trees consider all the possible feature splits, random forests only select a subset of those features.

### 2.11. SVR (Support Vector Regression)

SVR is a supervised learning algorithm derived from Support Vector Machines (SVM), designed for regression tasks. It aims to find a function that predicts target values within a specified margin of tolerance, while minimizing the error. SVR uses kernel functions to model complex, non-linear relationships in data, making it versatile for various applications. It is particularly effective in handling high-dimensional datasets and outliers due to its robust optimization framework.

### 2.12. SVR on residuals

In this model we tried to fit SVR on the differenced version of the series which was stationary and regression on the trend part and clubbed the result at the time of reconstruction of the series. The outcome was variance stablised but the prediction error was high. However, in indicating the collective movement it performed well.

*Table 1.* Best MAE on indices by different models with corresponding k values in brackets.

| MODEL | DOW | IXIC | GSPC | W5000 | XLK |
|---|---|---|---|---|---|
| REG. LSTM | 310.74 (10) | 98.86 (10) | 34.92 (10) | 371.11 (10) | 1.03 (10) |
| SMALL LSTM | 383.19 (10) | 121.21 (60) | 43.12 (10) | 452.58 (10) | 1.16 (60) |
| BILSTM | 190.76 (10) | 66.95 (10) | 22.13 (10) | 233.95 (10) | 0.67 (10) |
| LSTM-GRU | 270.98 (60) | 82.80 (10) | 30.22 (10) | 319.77 (10) | 0.81 (10) |
| BILSTM-GRU | 344.30 (30) | 107.22 (30) | 39.19 (30) | 414.69 (30) | 1.01 (30) |
| CNN-LSTM | 205.95 (10) | 67.15 (60) | 23.15 (10) | 246.22 (10) | 0.65 (60) |
| GRU | 260.99 (90) | 100.73 (60) | 28.65 (120) | 274.96 (60) | 1.14 (90) |
| GRU-CNN | 312.68 (60) | 103.98 (10) | 34.46 (10) | 362.07 (10) | 1.03 (60) |
| BILSTM-CNN | 210.51 (10) | 67.97 (10) | 23.74 (10) | 252.54 (10) | 0.67 (10) |
| SVR | 275.59 (30) | 110.72(30) | 33.02 (30) | 459.89 (30) | 1.88 (30) |
| DTR | 2054.48 (1) | 1481.06 (1) | 306.81 (1) | 3075.92 (1) | 19.09 (1) |
| RANDOM FT | 2065.09 (1) | 1477.86 (1) | 302.15 (1) | 2962.75 (1) | 19.57 (1) |

*Table 2.* Best MAE on AMGN, ABBV, and ABT(health care stocks) by different models with corresponding k values.

| MODEL | AMGN (K) | ABBV (K) | ABT (K) |
|---|---|---|---|
| REGULAR LSTM | 7.62 (10) | 4.31 (10) | 1.87 (10) |
| SMALL LSTM | 7.02 (10) | 4.14 (10) | 1.75 (10) |
| BILSTM | 4.61 (10) | 2.70 (10) | 1.24 (10) |
| LSTM-GRU | 5.71 (10) | 3.33 (10) | 1.48 (10) |
| BILSTM-GRU | 6.29 (30) | 3.94 (30) | 1.61 (10) |
| CNN-LSTM | 5.54 (60) | 3.14 (30) | 1.42 (30) |
| GRU-CNN | 7.08 (10) | 3.95 (10) | 1.72 (10) |
| BILSTM-CNN | 5.90 (10) | 3.35 (10) | 1.50 (10) |
| DTR | 35.10 (1) | 13.65 (1) | 2.60 (1) |
| RANDOM FT | 34.79 (1) | 13.15 (1) | 1.77 (1) |

## Results

In this section, we present several tables comparing the performance of different models across various stocks and indices. The results highlight the effectiveness of each model, alongside the corresponding historical window size ($k$) that yielded the best Mean Absolute Error (MAE) for each dataset. These comparisons provide valuable insights into model suitability for different stock categories and sectors.

## Discussion and Future Work

### Discussion

#### Summary of Indices Performance

- **Best Overall Model**: **Bidirectional LSTM (BiLSTM)** consistently delivers the lowest MAE values across most indices, particularly with shorter historical windows ($k = 10$). Its ability to model both forward and backward dependencies makes it the most robust choice.

- **Notable Contender**: **CNN-LSTM Hybrid** performs competitively, especially for indices like **IXIC** and **XLK**, where longer historical windows ($k = 60$) im-

*Table 3.* Best MAE on INTC, AMZN, and META(tech sector stocks) by different models with corresponding k values.

| MODEL | INTC (K) | AMZN (K) | META (K) |
|---|---|---|---|
| REGULAR LSTM | 1.66 (10) | 5.04 (10) | 16.28 (10) |
| SMALL LSTM | 1.49 (10) | 4.54 (10) | 16.00 (10) |
| BILSTM | 1.03 (10) | 3.27 (10) | 10.41 (10) |
| LSTM-GRU | 1.26 (10) | 3.85 (10) | 12.84 (10) |
| BILSTM-GRU | 1.27 (30) | 4.26 (10) | 15.78 (10) |
| CNN-LSTM | 1.26 (30) | 3.84 (30) | 12.07 (30) |
| GRU-CNN | 1.65 (10) | 4.58 (10) | 14.69 (10) |
| BILSTM-CNN | 1.40 (10) | 3.89 (10) | 12.21 (10) |
| DTR | 1.44 (1) | 6.76 (1) | 55.47 (1) |
| RANDOM FT | 1.23 (1) | 5.89 (1) | 52.05 (1) |

*Table 4.* Best MAE on TSLA, GM, and F(automotive stocks) by different models with corresponding k values.

| MODEL | TSLA (K) | GM (K) | F (K) |
|---|---|---|---|
| REGULAR LSTM | 14.26 (10) | 1.60 (10) | 0.47 (10) |
| SMALL LSTM | 13.12 (10) | 1.53 (10) | 0.42 (30) |
| BILSTM | 8.70 (10) | 1.04 (10) | 0.29 (10) |
| LSTM-GRU | 9.90 (10) | 1.23 (10) | 0.33 (10) |
| BILSTM-GRU | 10.64 (10) | 1.47 (30) | 0.36 (10) |
| CNN-LSTM | 10.70 (30) | 1.24 (10) | 0.36 (30) |
| GRU-CNN | 13.06 (10) | 1.48 (10) | 0.44 (10) |
| BILSTM-CNN | 10.67 (10) | 1.23 (10) | 0.37 (10) |

prove performance.

- **Window Size Insights**:
  - **Shorter Windows** ($k = 10$): Optimal for indices like **DOW**, **GSPC**, and **W5000**, capturing recent trends effectively.
  - **Longer Windows** ($k = 60$): Beneficial for indices like **XLK**, where temporal patterns require extended lookbacks.
- **Model-Index Dynamics**: While BiLSTM is the most reliable across indices, CNN-based models (e.g., **CNN-LSTM Hybrid**) excel in cases with complex spatial-temporal dependencies, offering a strong alternative.

**Healthcare Sector (AMGN, ABBV, ABT)**

In the healthcare sector, **Bidirectional LSTM (BiLSTM)** demonstrated superior performance:

- For AMGN, it achieved the lowest MAE of 4.61 at $k = 10$, outperforming models such as Regular LSTM (7.62) and GRU-CNN Hybrid (7.08).
- Similar trends were observed for ABBV and ABT, where BiLSTM achieved MAE values of 2.70 and 1.24, respectively, at $k = 10$.

*Table 5.* Best MAE on GE, BA, and RTX by different models with corresponding k values.

| MODEL | GE (K) | BA (K) | RTX (K) |
|---|---|---|---|
| REGULAR LSTM | 4.54 (10) | 6.25 (60) | 1.98 (10) |
| SMALL LSTM | 4.62 (10) | 5.47 (30) | 2.08 (10) |
| BILSTM | 3.01 (10) | 4.17 (10) | 1.26 (10) |
| LSTM-GRU | 3.75 (10) | 4.50 (60) | 1.60 (10) |
| BILSTM-GRU | 4.75 (10) | 4.82 (30) | 2.03 (30) |
| CNN-LSTM | 3.46 (30) | 4.94 (30) | 1.46 (30) |
| GRU-CNN | 4.20 (10) | 6.77 (10) | 1.82 (10) |
| BILSTM-CNN | 3.48 (10) | 5.66 (10) | 1.46 (10) |

- Interestingly, the CNN-LSTM Hybrid performed well for ABT, with an MAE of 1.42 ($k = 30$), indicating that longer historical windows might benefit certain stocks.

**Technology Sector (INTC, AMZN, META)**

The performance of BiLSTM remained consistent in the tech sector:

- For INTC and AMZN, BiLSTM achieved the lowest MAE values of 1.03 and 3.27, both at $k = 10$.
- For META, BiLSTM achieved the best performance (10.41), closely followed by the CNN-LSTM Hybrid (12.07 at $k = 30$).

These results suggest that while BiLSTM is effective for most tech stocks, CNN-LSTM Hybrids may benefit stocks with complex temporal patterns, such as META, where slightly longer historical windows provide value.

**Automotive Sector (TSLA, GM, F)**

The results for automotive stocks reveal a mix of performance patterns:

- BiLSTM consistently achieved low MAE values across TSLA, GM, and F, with 8.70, 1.04, and 0.29, respectively, all at $k = 10$.
- The CNN-LSTM Hybrid showed competitive performance for TSLA and F, achieving MAE values of 10.70 ($k = 30$) and 0.36 ($k = 30$).

This suggests that while BiLSTM excels with short historical windows in this sector, CNN-based models may capture additional patterns for certain stocks using longer windows.

**Industrial Sector (GE, BA, RTX)**

In the industrial sector, **Bidirectional LSTM (BiLSTM)** and **CNN-LSTM Hybrid** models showed the best performance:

- For GE, BiLSTM achieved an MAE of 3.01 at $k = 10$, while CNN-LSTM Hybrid also performed well with an MAE of 3.46 at $k = 30$.
- For RTX, CNN-LSTM Hybrid slightly outperformed BiLSTM, with an MAE of 1.46 ($k = 30$) compared to BiLSTM's 1.46 ($k = 10$).
- BA showed a preference for BiLSTM at $k = 10$, achieving an MAE of 4.17, while the CNN-LSTM Hybrid and LSTM-GRU Hybrid performed well at longer windows ($k = 30$).

### General Observations

- **Bidirectional LSTM Dominance**: Across almost all indices and sectors, Bidirectional LSTM models outperformed other architectures, especially for shorter historical windows ($k = 10$). This highlights its ability to effectively capture bidirectional temporal dependencies.
- **CNN-LSTM Hybrid Strength**: CNN-LSTM Hybrids showed competitive performance, especially for stocks where spatial (convolutional) patterns may play a significant role. These models also benefit from longer historical windows ($k = 30$).
- **Sector-Specific Trends**:
  - For indices and healthcare stocks, shorter historical windows ($k = 10$) often sufficed for optimal predictions.
  - For tech and industrial stocks, there was a mix of preferences for $k$-values, indicating variability in temporal dependencies across different stocks.
- **Model-Specific Preferences**: While BiLSTM and CNN-LSTM Hybrid models generally performed the best, LSTM-GRU Hybrids and GRU-CNN Hybrids provided competitive alternatives, particularly for stocks with more volatile or complex patterns.
- **Feature Engineering**:
  - Before feature engineering, the models (Decision Tree, Random Forest) performed poorly, with high errors in predicting returns. After introducing well-engineered features (e.g., moving averages, lagged returns, RSI, MACD), the models achieved substantially lower Mean Squared Error (MSE), indicating much better accuracy.
  - Feature engineering added relevant financial indicators and patterns that the models could leverage to understand the stock's behavior.

### Conclusion

The results emphasize the importance of selecting both the appropriate model and historical window size for different datasets. **Bidirectional LSTM models** stand out as the most consistent performers, but **CNN-LSTM Hybrids** offer competitive alternatives in specific cases. Furthermore, the sector-specific analysis highlights that the choice of the best model may depend on the characteristics of the stocks or indices being analyzed.

### Future Work

This study highlights promising approaches for stock price prediction using hybrid deep learning models, but several avenues for future research remain:

1. **Incorporating Attention Mechanisms:** Attention mechanisms could be integrated with LSTM or BiLSTM layers to allow the model to focus on the most relevant time steps in the input sequence. This could enhance the interpretability and accuracy of predictions, particularly for datasets with non-uniform temporal importance.
2. **Advanced Hyperparameter Tuning:** While this study explored some hyperparameter combinations, a more exhaustive search using techniques like Bayesian optimization or genetic algorithms could be employed to fine-tune the model architecture and parameters. This would help identify optimal configurations for different stock categories and indices.
3. **Exploration of Long-Short Term Memory Networks with Transformers:** Combining LSTMs with transformers, which excel at capturing long-range dependencies, could provide a robust framework for financial time-series forecasting. This hybrid approach may improve performance for datasets with complex temporal patterns.
4. **Incorporating Large Language Models (LLMs):** Recent advancements in LLMs like GPT can be leveraged to incorporate textual data, such as financial news or sentiment analysis, into the predictive framework. This multimodal approach could complement the numerical analysis by providing additional context and improving predictive power.
5. **Sector-Specific Fine-Tuning:** While this study identified sector-based trends, future work could focus on fine-tuning models for individual sectors, incorporating domain-specific features such as industry metrics or macroeconomic indicators.
6. **Real-Time Prediction Systems:** Building real-time prediction pipelines using the developed models and

streaming data could enable applications in high-frequency trading or dynamic portfolio optimization. Integrating tools like Kafka and Spark could make such systems scalable and efficient.

7. **Explainability and Interpretability:** Future research could explore methods to make the predictions more interpretable, such as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations). This is particularly critical for applications in finance, where decision-making requires transparency.

8. **Application to Global Markets:** Extending the analysis to global stock indices and incorporating multi-currency datasets could provide insights into the generalizability of the models across diverse economic conditions and market behaviors.

These directions can further enhance the utility, accuracy, and versatility of deep learning models in financial forecasting and decision-making.

## References

[1] Varun Tyagi. (2023). *Stock Price Prediction with Machine Learning*. Available at: https://medium.com/@varun.tyagi83/stock-price-prediction-with-machine-learning-02c9cce35aab

[2] Neptune AI. (2023). *Predicting Stock Prices Using Machine Learning*. Available at: https://neptune.ai/blog/predicting-stock-prices-using-machine-learning

[3] GitHub. (n.d.). *Stock Price Prediction - Topics on GitHub*. Available at: https://github.com/topics/stock-price-prediction

[4] Lilian Weng. (2023). *Stock Price Prediction using RNN*. GitHub repository. Available at: https://github.com/lilianweng/stock-rnn

[5] Last Ancient One. (2023). *Deep Learning and Machine Learning for Stock Prediction*. GitHub repository. Available at: https://github.com/LastAncientOne/Deep_Learning_Machine_Learning_Stock

# A. Appendix

Looking at the predictions made by different models

```
Plotting DOW at k = 10
Model chosen: bidirectional
Testing for k = 10 models
117/117 1s 6ms/step
117/117 1s 5ms/step
117/117 1s 5ms/step
117/117 1s 5ms/step
Mean absolute error for k = 10: 190.7613803573032
```

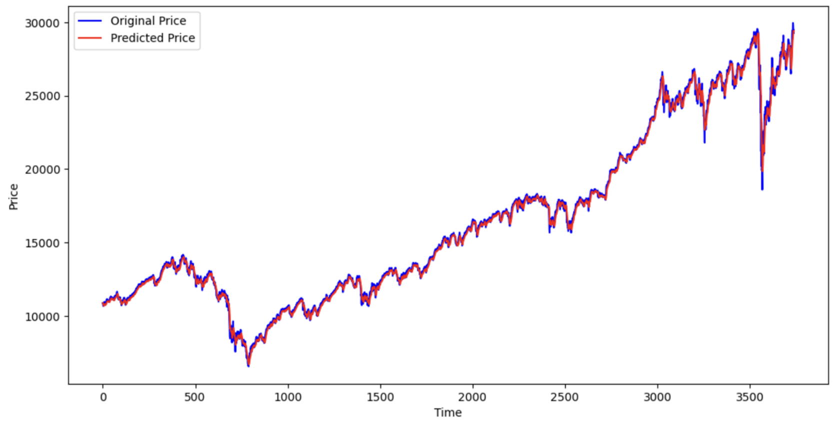


*Figure 1.* Prediction on DOW JONES

```
Plotting GSPC at k = 10
Model chosen: bidirectional
Testing for k = 10 models
117/117 1s 6ms/step
117/117 1s 5ms/step
117/117 1s 6ms/step
117/117 1s 6ms/step
Mean absolute error for k = 10: 22.126896337873273
```

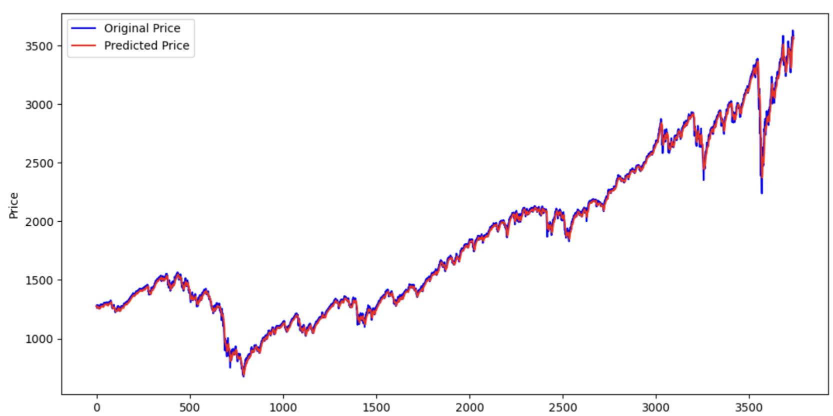


*Figure 2.* Prediction on GSPC

```
Plotting IXIC at k = 10
Model chosen: bidirectional
Testing for k = 10 models
117/117 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step
117/117 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step
117/117 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step
117/117 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step
Mean absolute error for k = 10: 66.94654605502559
```

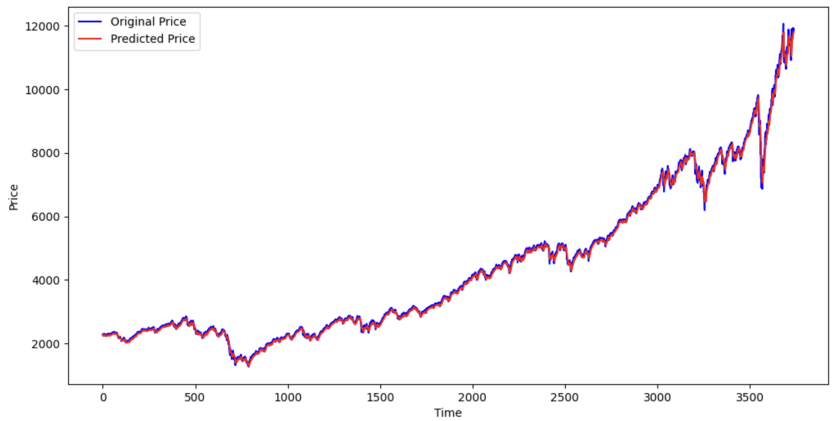


*Figure 3.* Prediction on IXIC

```
Plotting W5000 at k = 10
Model chosen: bidirectional
Testing for k = 10 models
117/117 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step
117/117 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step
117/117 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step
117/117 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step
Mean absolute error for k = 10: 233.95244290636833
```

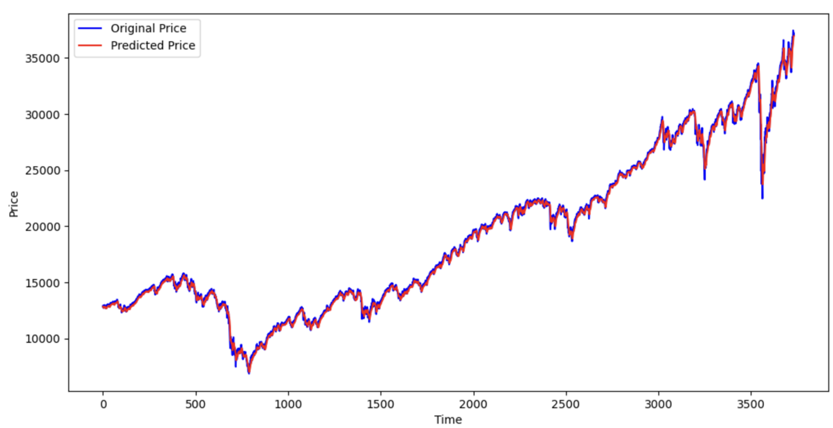


*Figure 4.* Prediction on W5000

```
Plotting XLK at k = 60
Model chosen: CNN LSTM hybrid
Testing for k = 60 models
116/116 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step
116/116 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step
116/116 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step
116/116 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step
Mean absolute error for k = 60: 0.6548110617417249
```

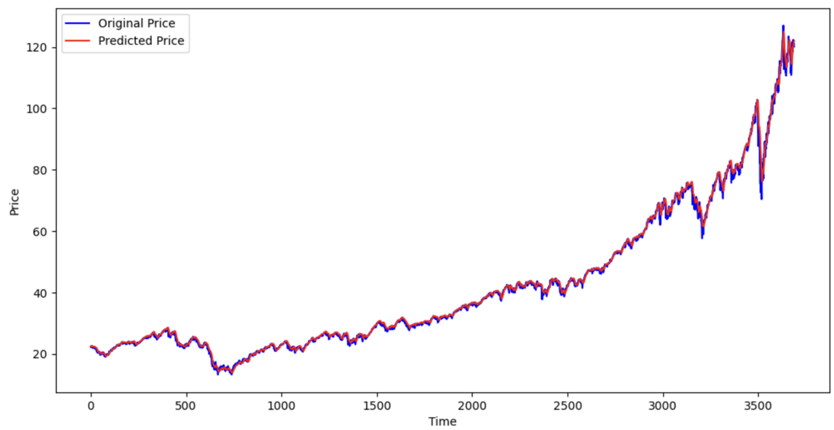


*Figure 5.* Prediction on XLK

```
Plotting AMGN at k = 10
Model chosen: bidirectional
Testing for k = 10 models
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 44ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 43ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 44ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 44ms/step
Mean absolute error for k = 10: 4.617892964490418
```

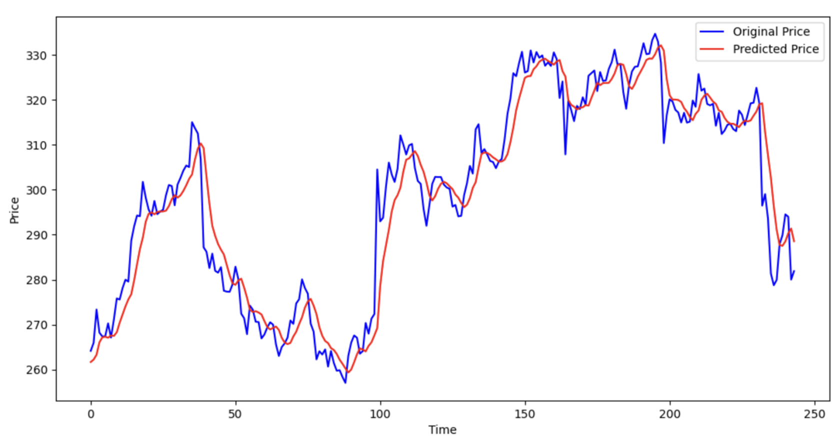


*Figure 6.* Prediction on AMGN

```
Plotting ABBV at k = 10
Model chosen: bidirectional
Testing for k = 10 models
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 44ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 43ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 42ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 41ms/step
Mean absolute error for k = 10: 2.715643819093537
```

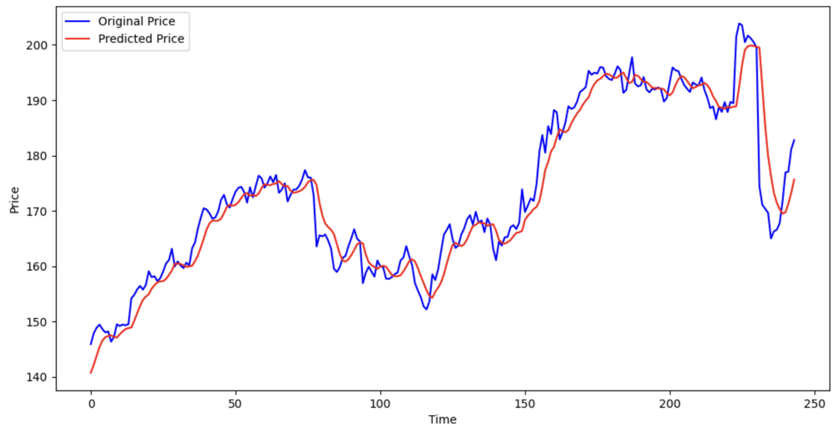


*Figure 7.* Prediction on ABBV

```
Plotting ABT at k = 10
Model chosen: bidirectional
Testing for k = 10 models
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 65ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 49ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 53ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 49ms/step
Mean absolute error for k = 10: 1.24131120776882
```

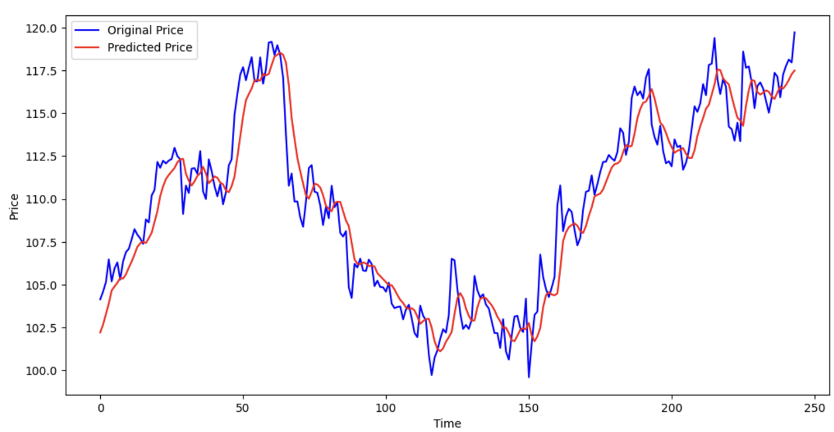


*Figure 8.* Prediction on ABT

```
Plotting INTC at k = 10
Model chosen: bidirectional
Testing for k = 10 models
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 44ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 43ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 42ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 43ms/step
Mean absolute error for k = 10: 1.0294160468659792
```

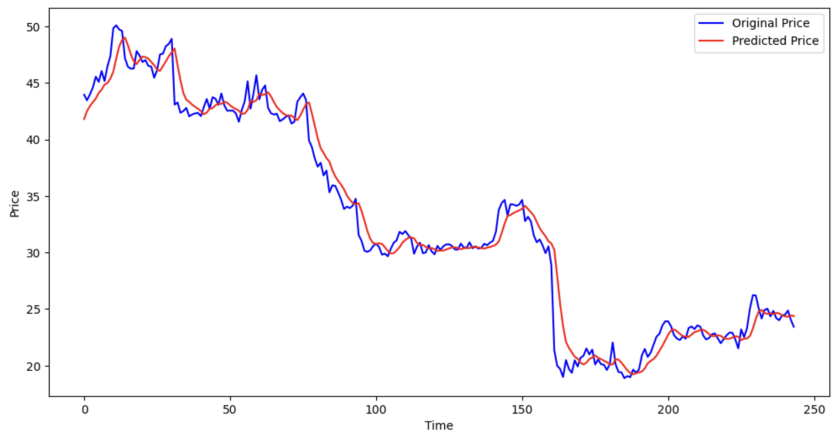


*Figure 9.* Prediction on INTC

```
Plotting AMZN at k = 10
Model chosen: bidirectional
Testing for k = 10 models
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 41ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 64ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 52ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 50ms/step
Mean absolute error for k = 10: 3.267092872511195
```

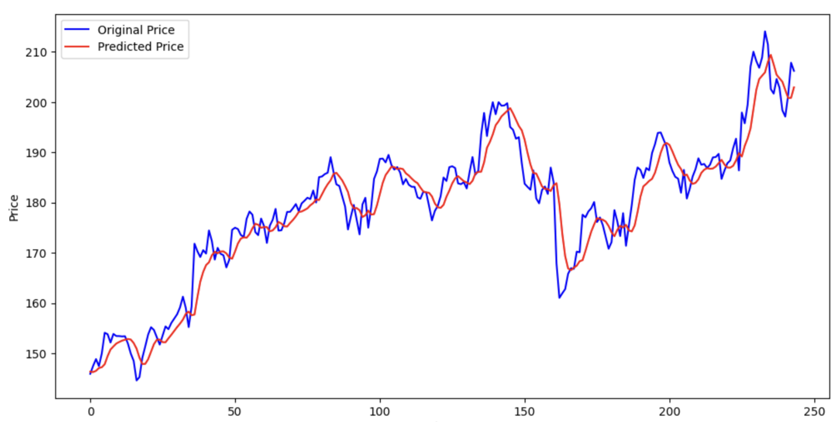


*Figure 10.* Prediction on AMZN

```
Plotting META at k = 10
Model chosen: bidirectional
Testing for k = 10 models
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 53ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 49ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 47ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 47ms/step
Mean absolute error for k = 10: 10.379398715766548
```

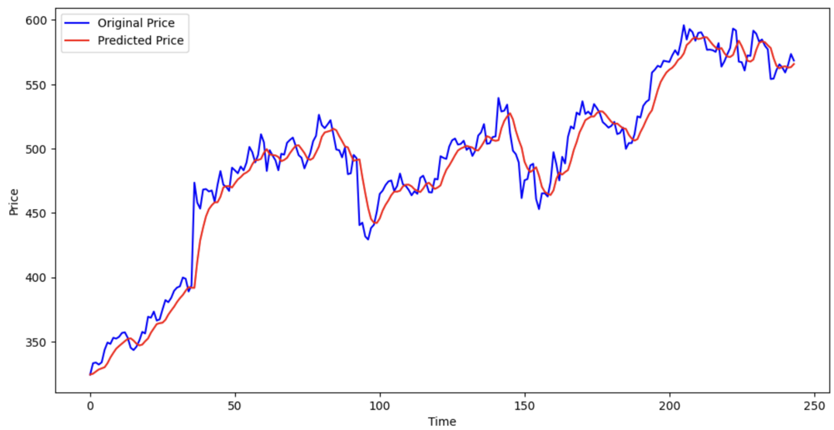


*Figure 11.* Prediction on META

```
Plotting TSLA at k = 10
Model chosen: bidirectional
Testing for k = 10 models
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 41ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 40ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 41ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 64ms/step
Mean absolute error for k = 10: 8.700456375884325
```

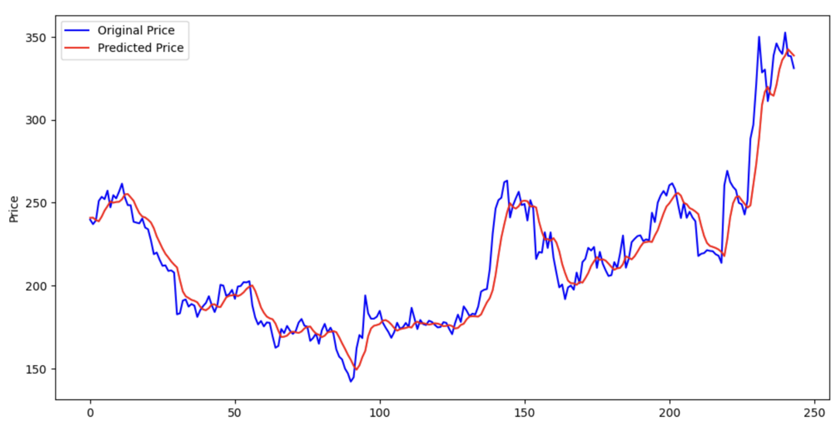


*Figure 12.* Prediction on TSLA

```
Plotting GM at k = 10
Model chosen: bidirectional
Testing for k = 10 models
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 53ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 55ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 49ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 46ms/step
Mean absolute error for k = 10: 1.0425604590979565
```

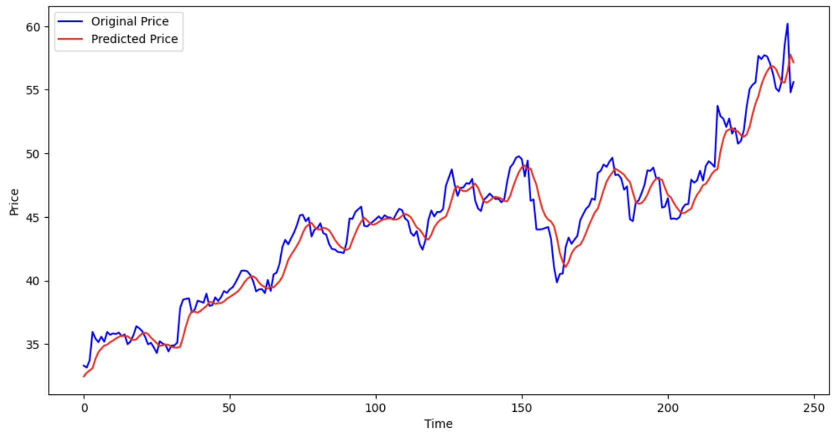


*Figure 13.* Prediction on GM

```
Plotting F at k = 10
Model chosen: bidirectional
Testing for k = 10 models
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 46ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 44ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 45ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 43ms/step
Mean absolute error for k = 10: 0.2879890900006586
```

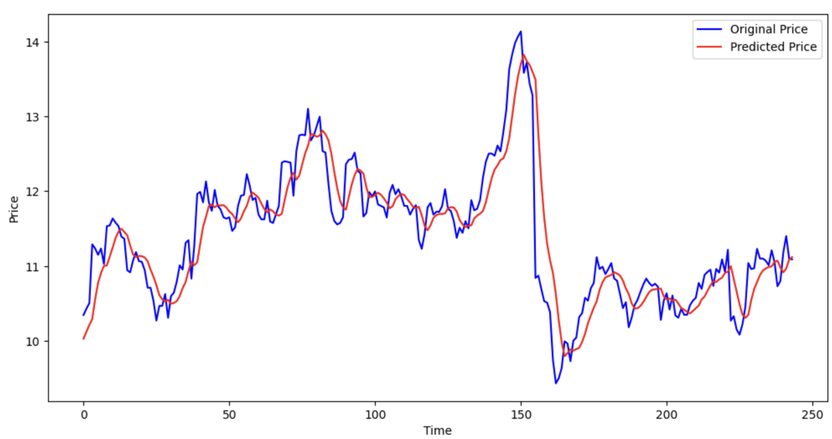


*Figure 14.* Prediction on F(Ford)

```
Plotting GE at k = 10
Model chosen: bidirectional
Testing for k = 10 models
8/8                      1s 41ms/step
8/8                      1s 41ms/step
8/8                      1s 41ms/step
8/8                      1s 41ms/step
Mean absolute error for k = 10: 3.0037716221987383
```

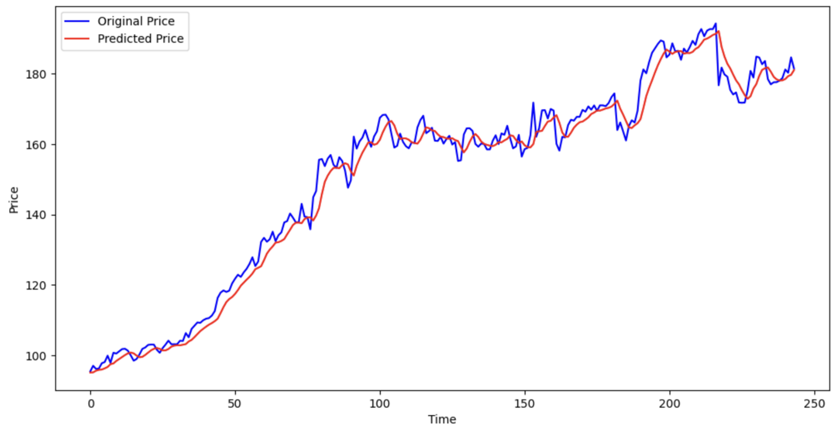


*Figure 15.* Prediction on GE

```
Plotting BA at k = 10
Model chosen: bidirectional
Testing for k = 10 models
8/8                      1s 64ms/step
8/8                      1s 61ms/step
8/8                      1s 54ms/step
8/8                      1s 52ms/step
Mean absolute error for k = 10: 4.163216355269931
```

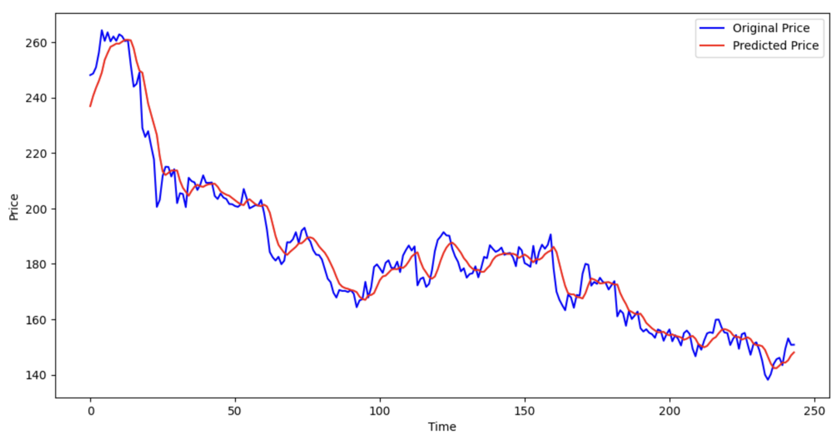


*Figure 16.* Prediction on BA

```
Plotting RTX at k = 10
Model chosen: bidirectional
Testing for k = 10 models
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 48ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 47ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 47ms/step
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 45ms/step
Mean absolute error for k = 10: 1.2584145442288057
```

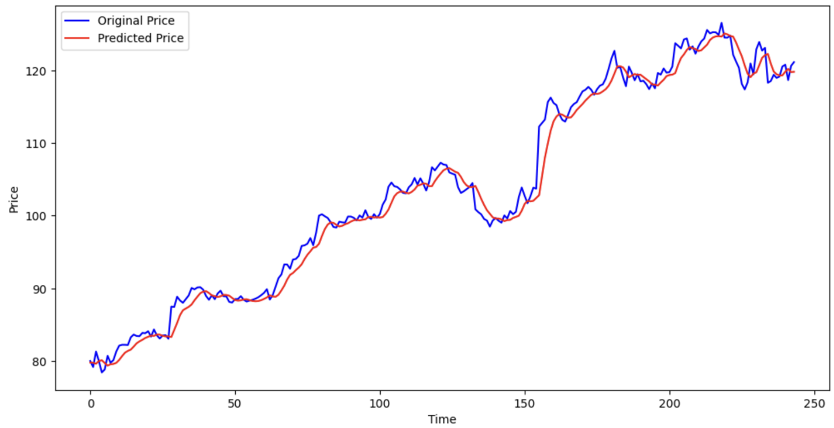


*Figure 17.* Prediction on RTX

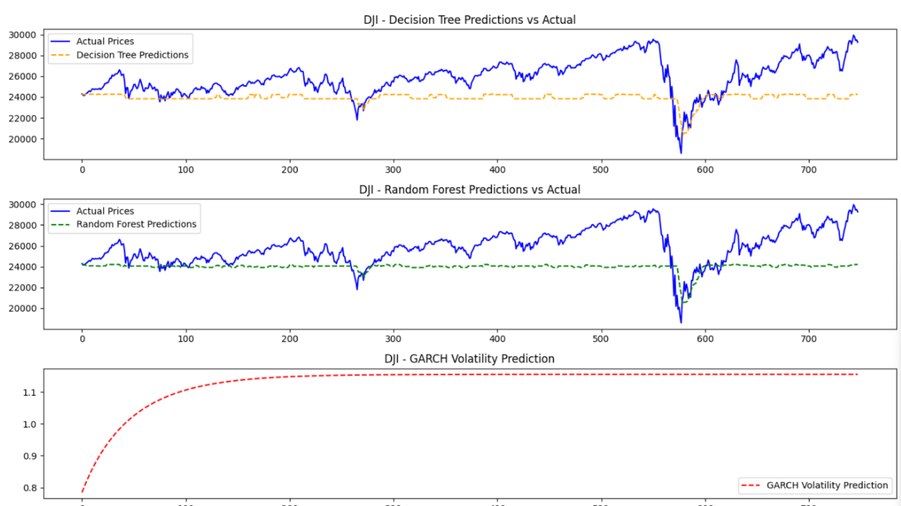


*Figure 18.* Prediction on DOW JONES

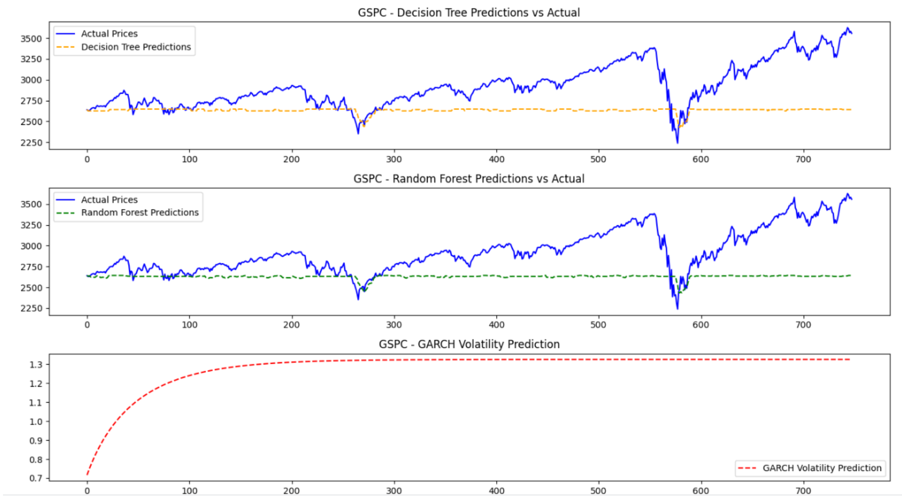


*Figure 19.* Prediction on GSPC

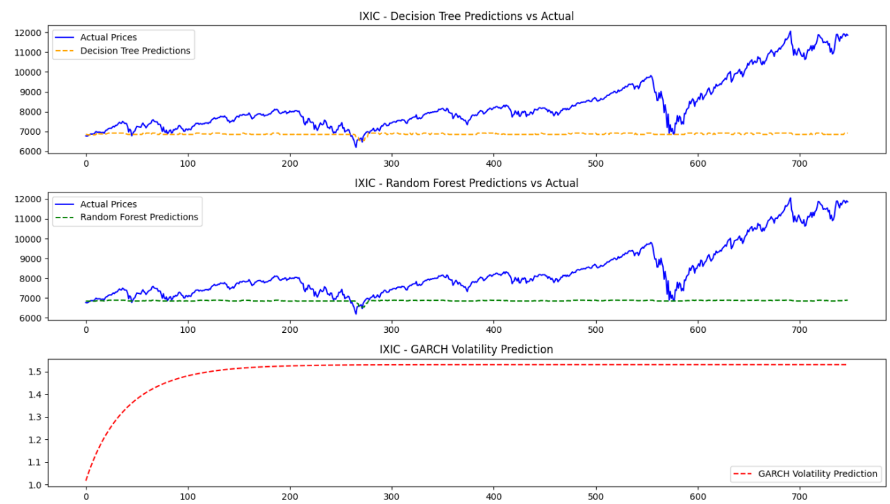


*Figure 20.* Prediction on IXIC

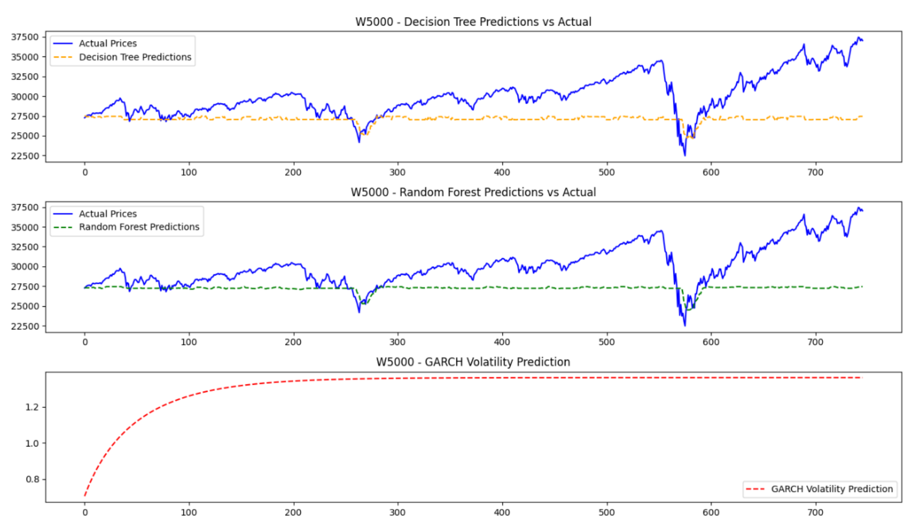


*Figure 21.* Prediction on W5000

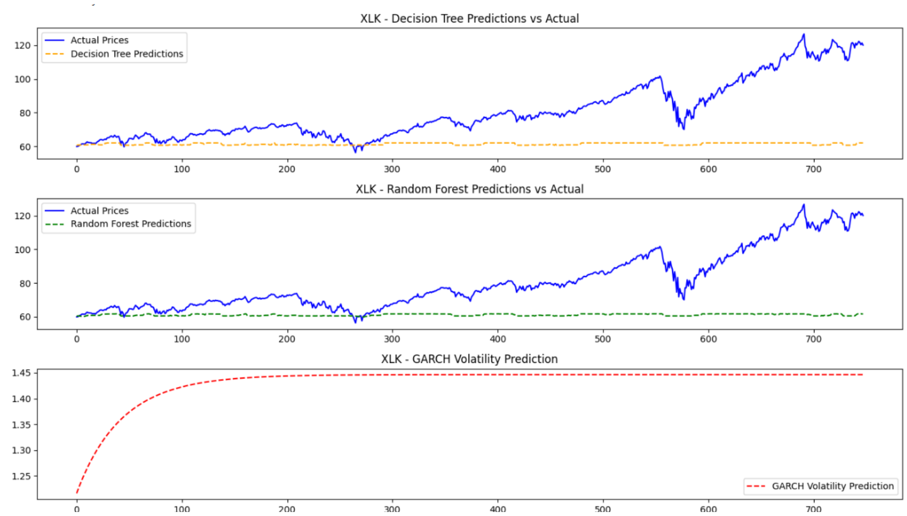


*Figure 22.* Prediction on XLK

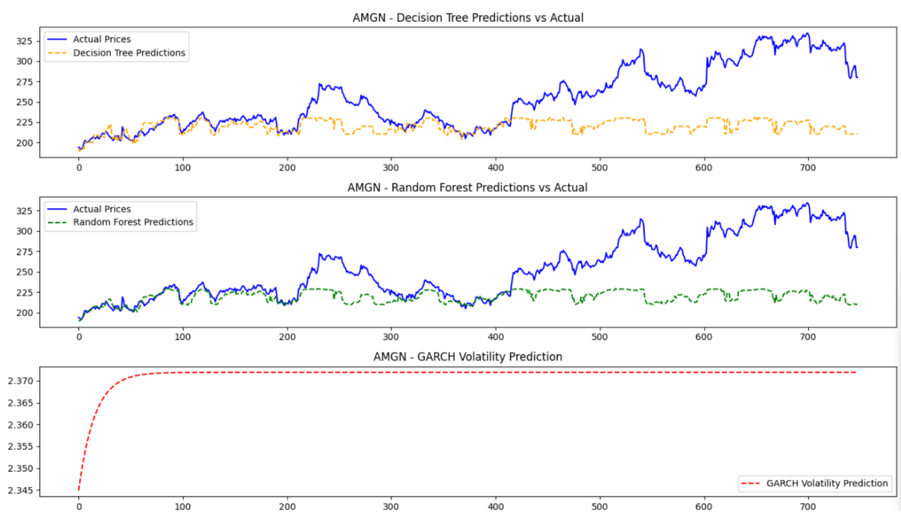


*Figure 23.* Prediction on AMGN


*Figure 24.* Prediction on ABBV

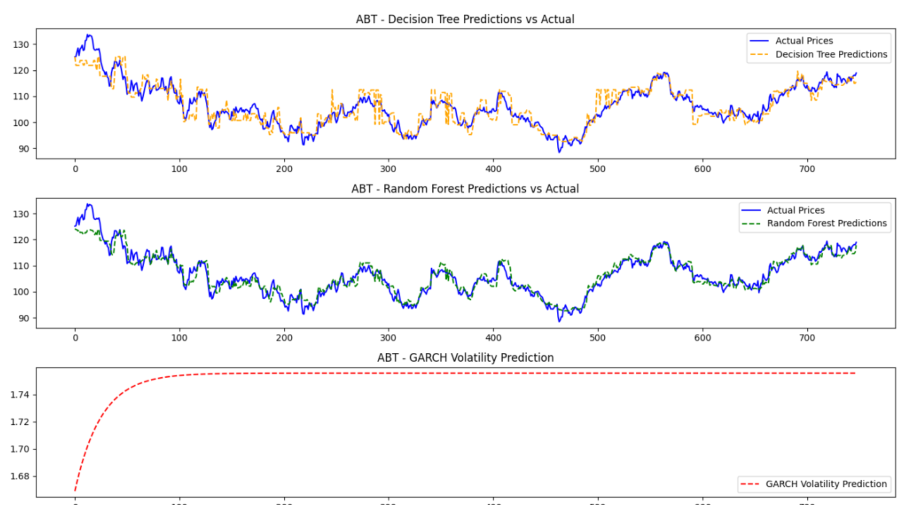


*Figure 25.* Prediction on ABT

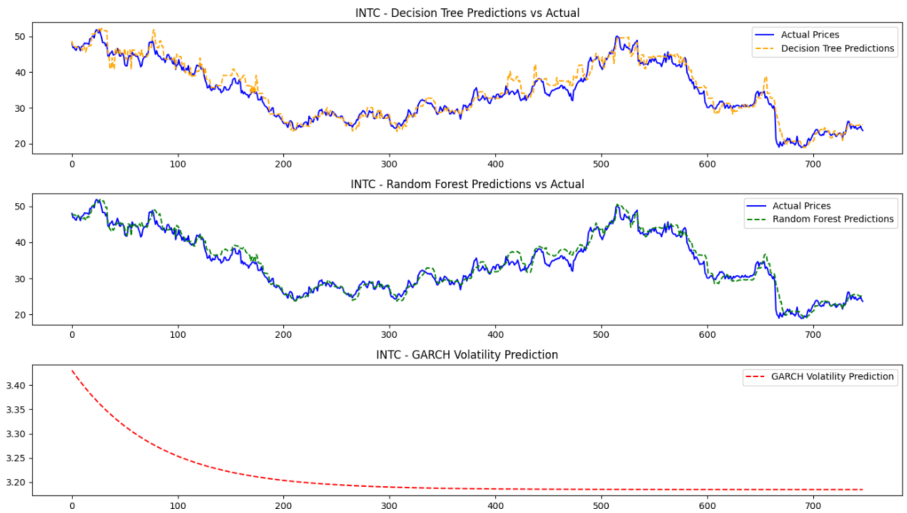


*Figure 26.* Prediction on INTC

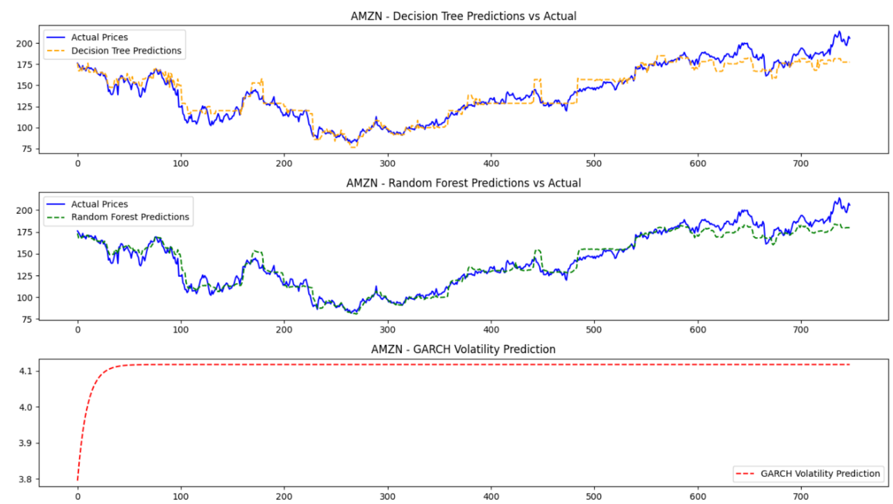


*Figure 27.* Prediction on AMZN

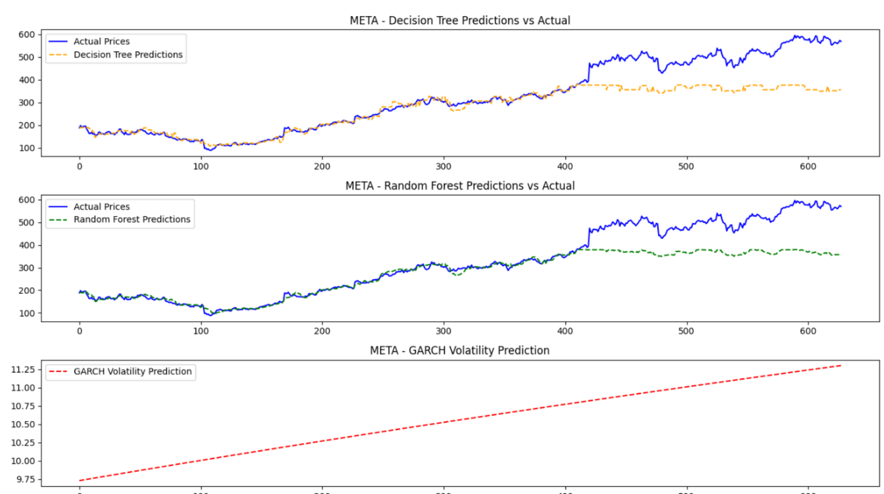


*Figure 28.* Prediction on META