Forecasting Stock Market Returns

Group Members: Elaa Hamdi,Aadip Thapaliya,
Rupesh Raut

Content

Motivation

Dataset

Baseline Models

Models

Model comparison

Conclusion

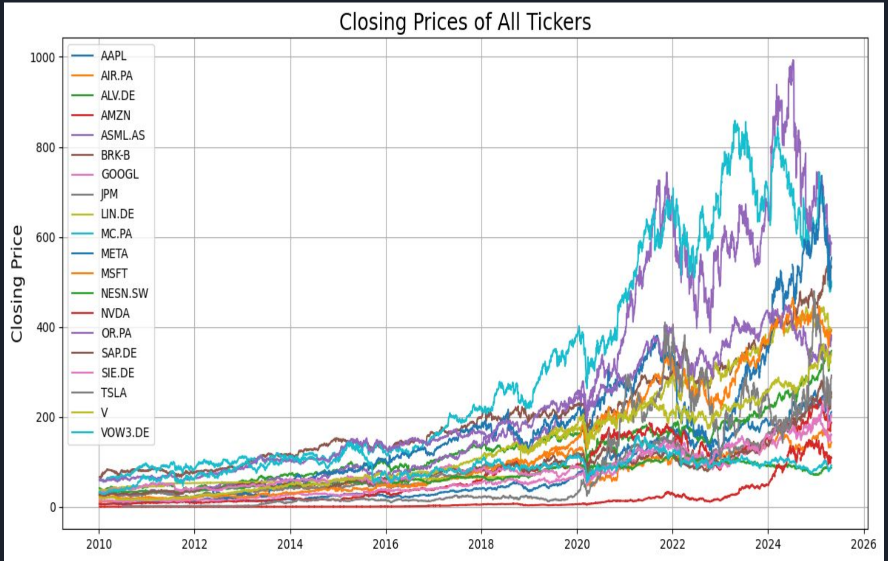
Motivation

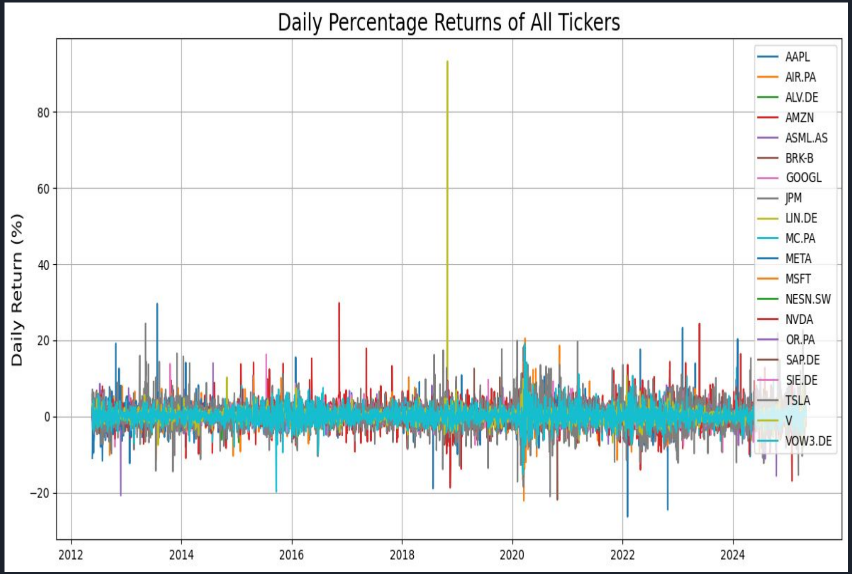
Our motivation is to explore the power of modern AI techniques in capturing nuanced financial patterns. By analyzing data from **20 global companies**, we aim to:

- Improve predictive accuracy in high-volatility environments
- Compare classic and contemporary models across sectors
- Uncover short-term signals that support better decision-making

Data set

- **20 globally recognized companies** across technology, finance, automotive, industrial, and consumer sectors.
- Data was sourced from **Yahoo Finance**, covering the period from **January 1, 2010 to March 31, 2025**.





Baseline Models

We have used ARIMA and GARCH as our baseline model

- **ARIMA**: Used for short-term price forecasting by modeling autoregressive, integrated, and moving-average components.

- **GARCH**: Used for volatility forecasting by modeling time-varying conditional variance (volatility clustering).

These two models serve as our foundational benchmarks for return and risk prediction.

ARIMA Model: Price Forecasting Workflow

Data Preparation

- Applied a log transformation to stabilize variance.
- Differenced the series (order d=1) to achieve stationarity.

Parameter Selection

- Performed a grid search over AR order p and MA order q (with d =1 fixed).
- Selected the (p, d, q) combination that minimized the Bayesian Information Criterion (BIC).

Model Fitting

- Fitted the ARIMA model on the training set with the chosen parameters.
- Validated performance on the hold-out test set.

Evaluation

- Implemented a rolling forecast on the test set, refitting the ARIMA model at each step.
- Generated separate one-step-ahead forecasts for each test point
- Computed MAE and RMSE to assess accuracy.

GARCH Model: Volatility Forecasting Workflow

Data Preparation

- Calculated daily log returns

Parameter Selection

- Performed a grid search over GARCH orders p and q using the training data.
- For each stock, selected the (p, q) pair that minimized the Bayesian Information Criterion (BIC).

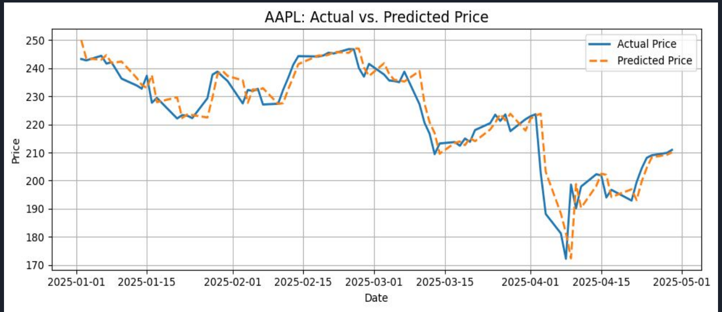
Model Fitting

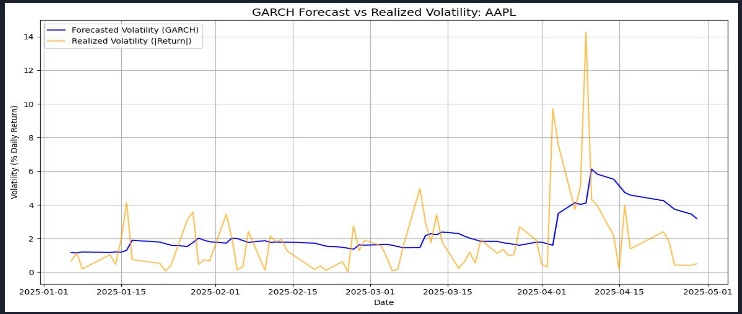
- Fitted the GARCH model on the training set with the chosen parameters.
- Validated performance on the hold-out test set.

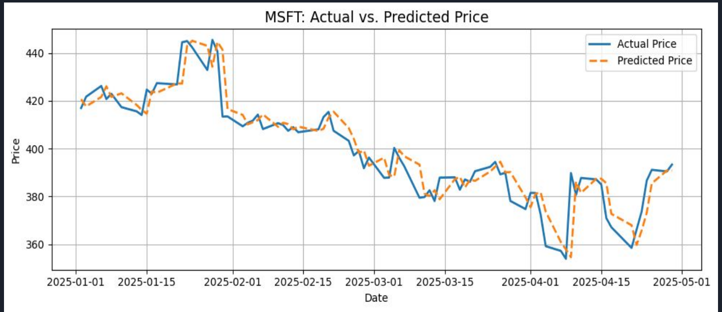
Forecasting & Evaluation

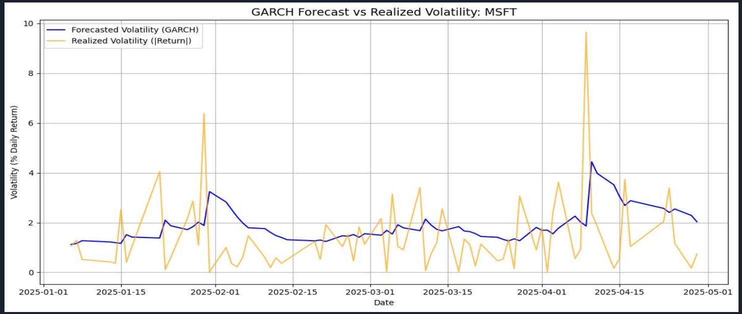
- Implemented a rolling forecast on the test set, refitting the GARCH model at each step.
- Forecasted one-step-ahead Volatility for each test point.
- Computed MAE and RMSE to assess accuracy.

Forecasting Performance: ARIMA & GARCH Actual vs. Predicted

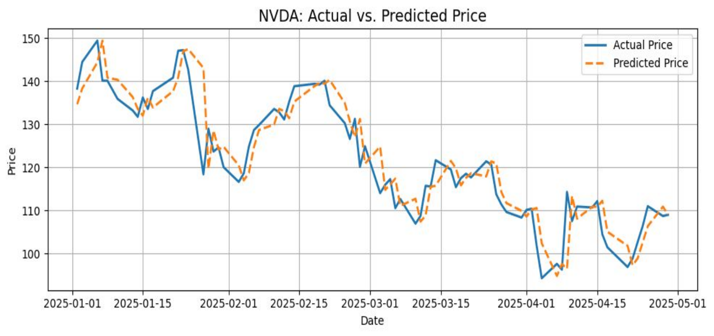


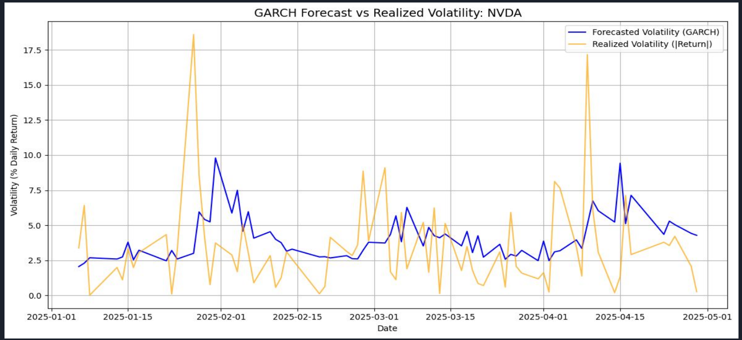


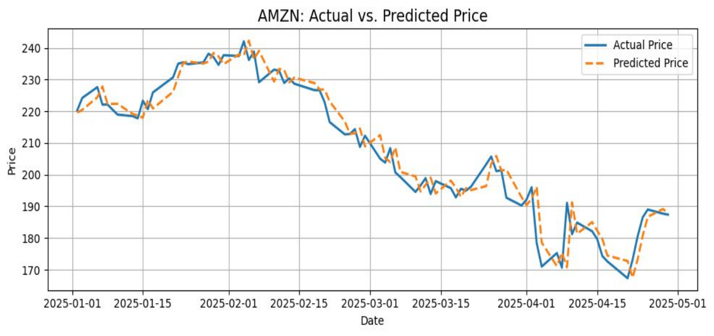


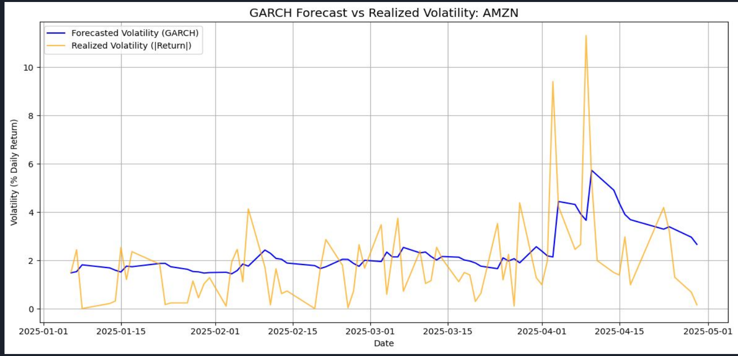


Forecasting Performance: ARIMA & GARCH Actual vs. Predicted

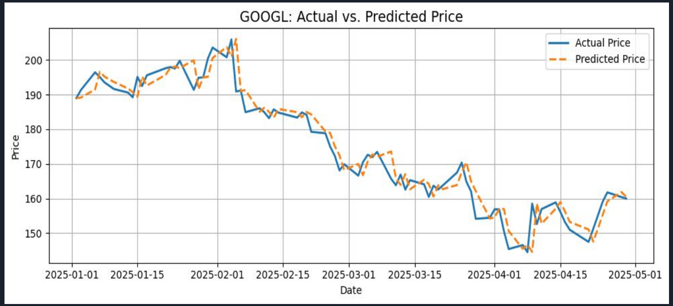


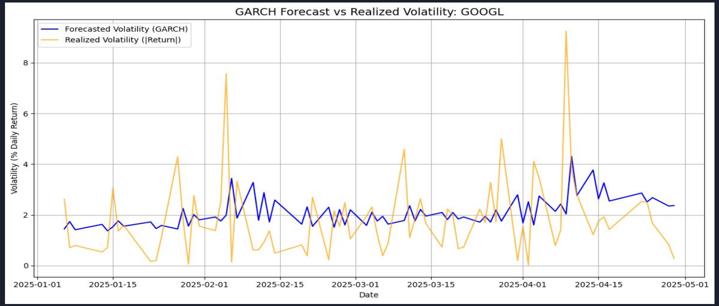


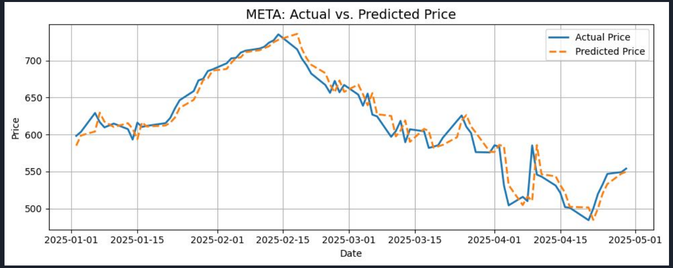


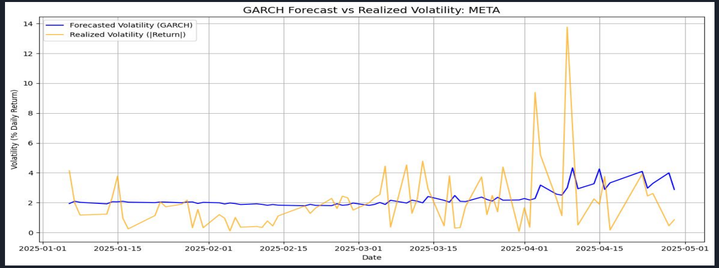


Forecasting Performance: ARIMA & GARCH Actual vs. Predicted

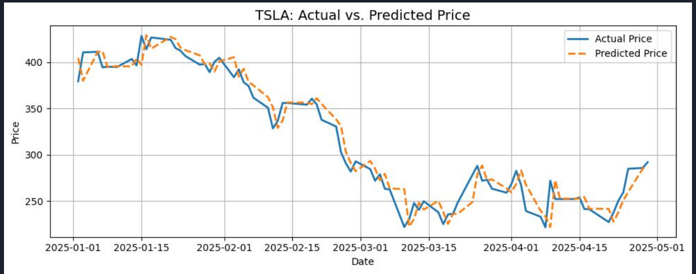


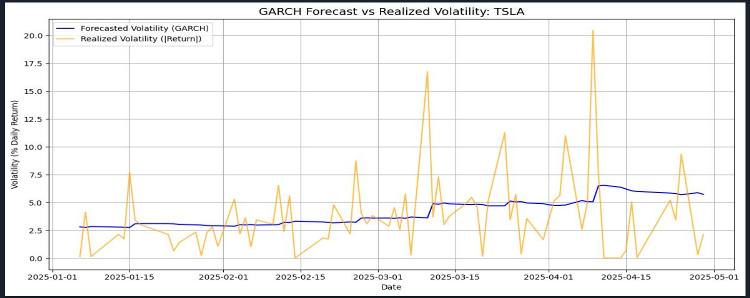


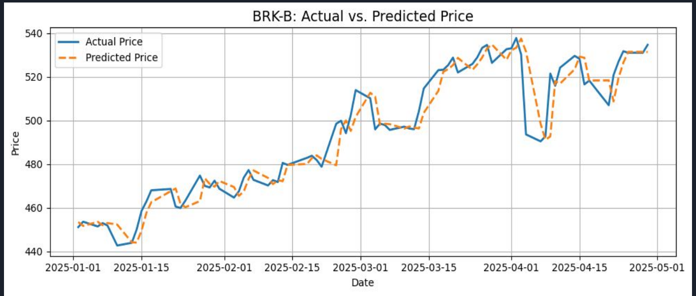


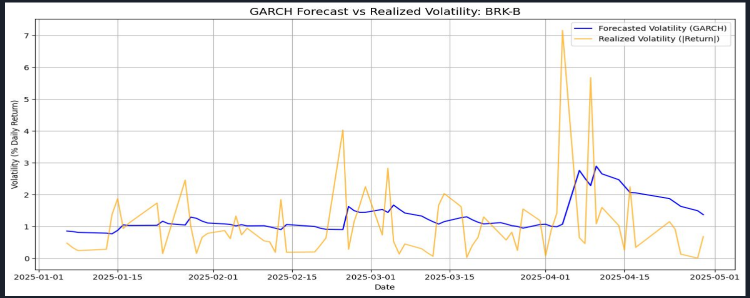


Forecasting Performance: ARIMA & GARCH Actual vs. Predicted

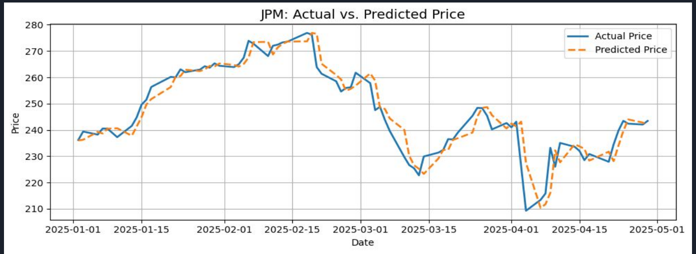


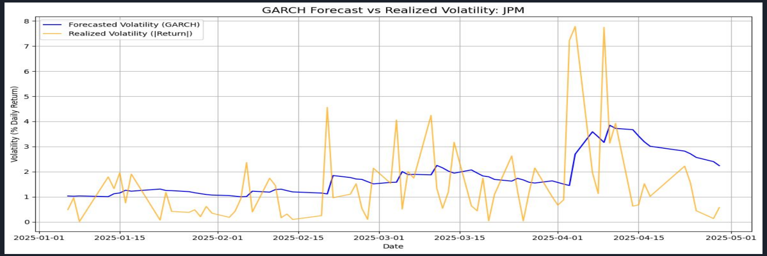


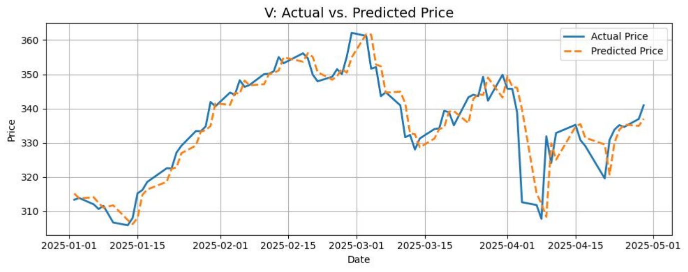


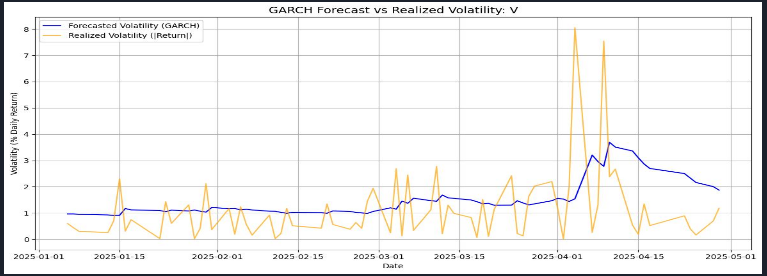


Forecasting Performance: ARIMA & GARCH Actual vs. Predicted

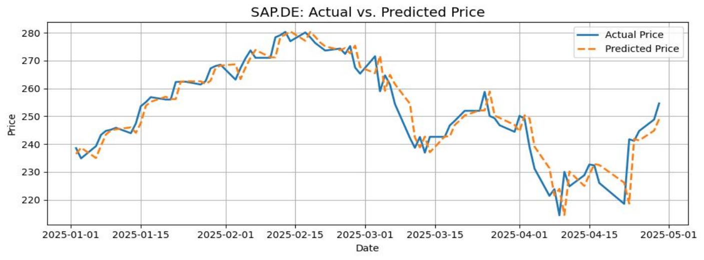


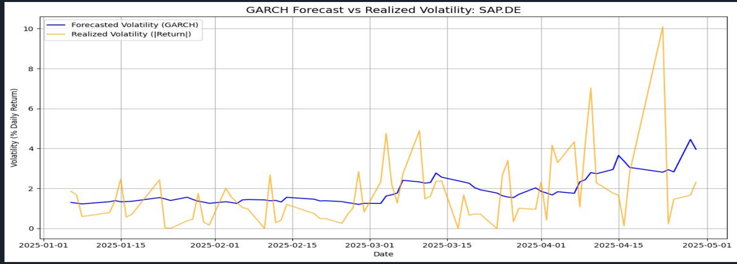


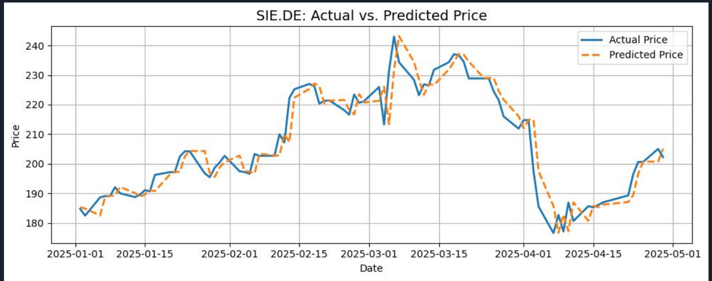


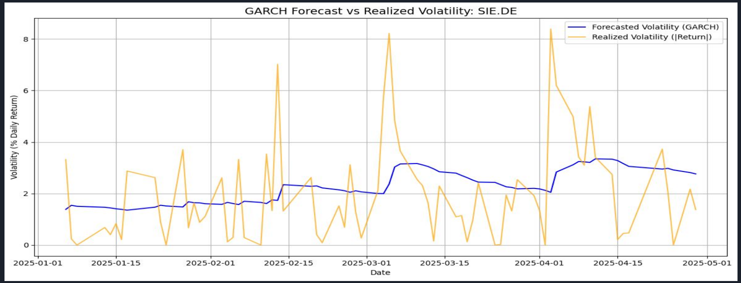


Forecasting Performance: ARIMA & GARCH Actual vs. Predicted

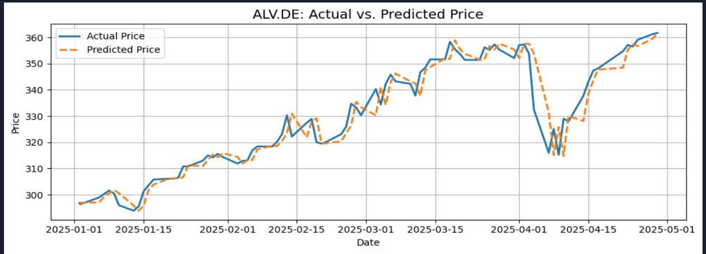


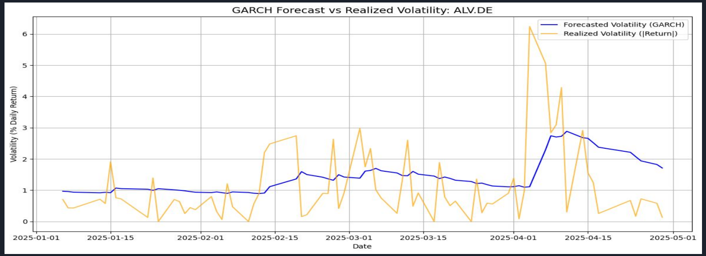


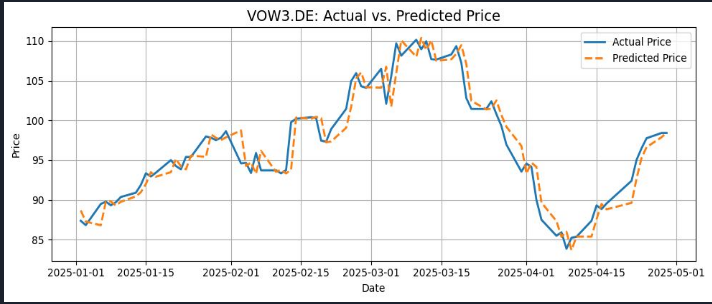


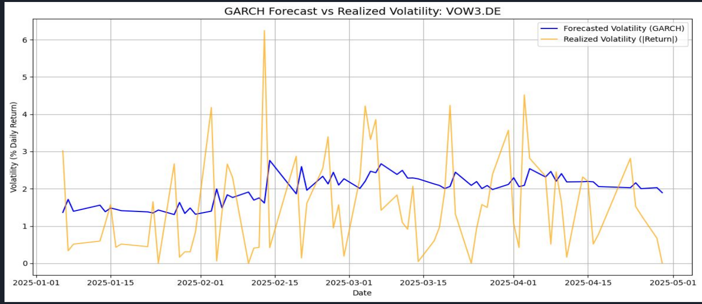


Forecasting Performance: ARIMA & GARCH Actual vs. Predicted

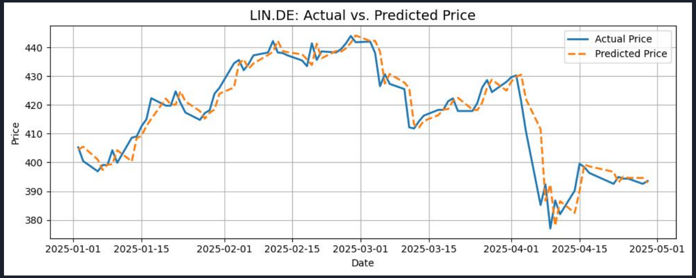


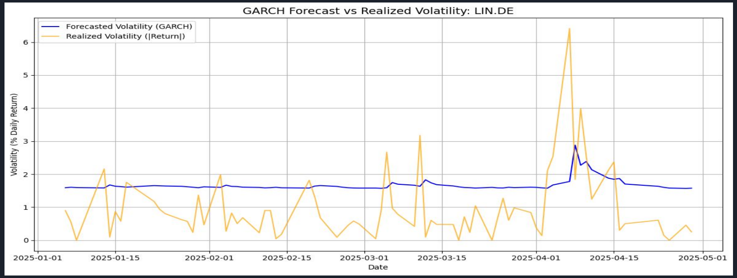


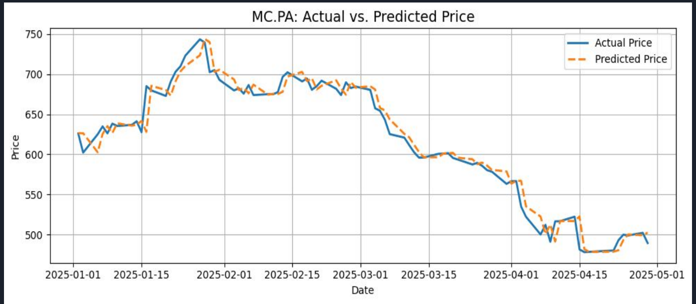


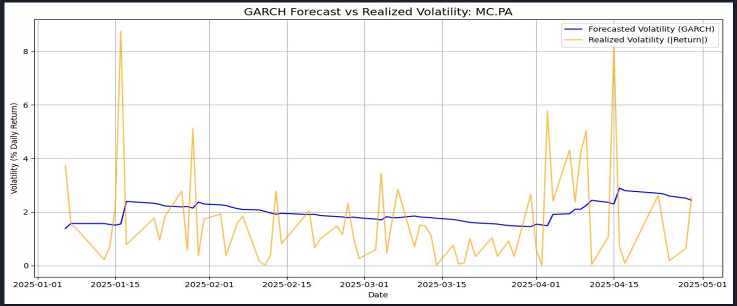


Forecasting Performance: ARIMA & GARCH Actual vs. Predicted

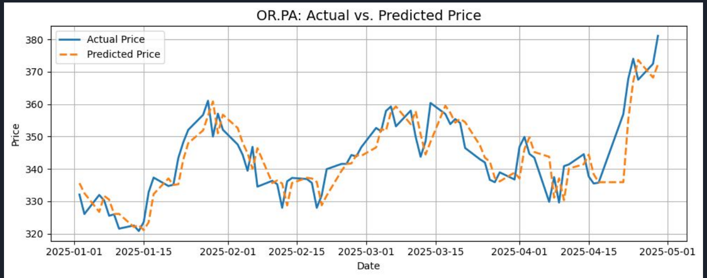


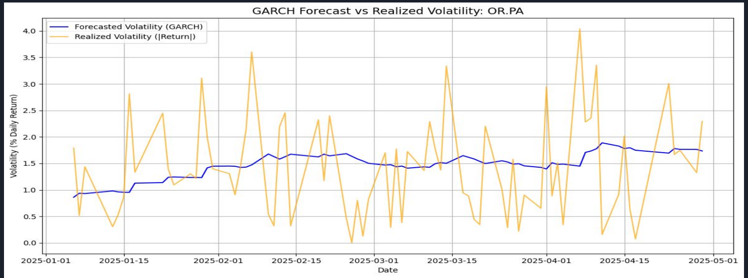


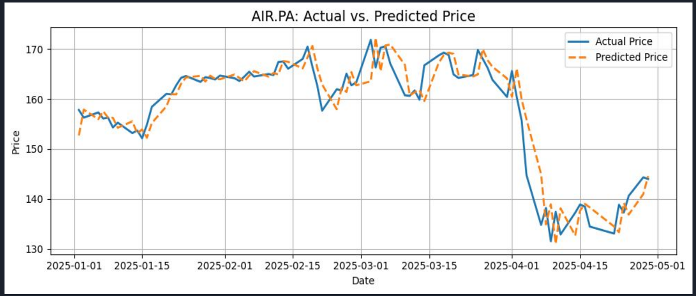


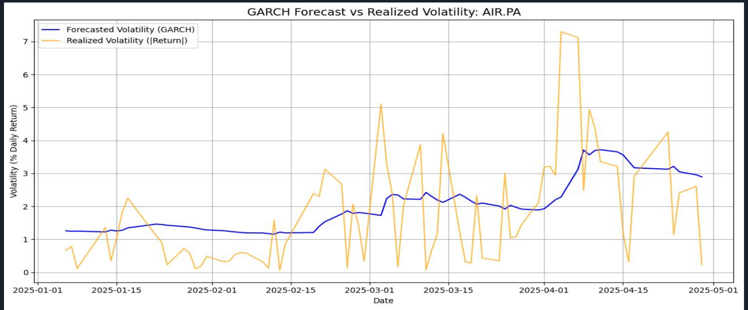


Forecasting Performance: ARIMA & GARCH Actual vs. Predicted

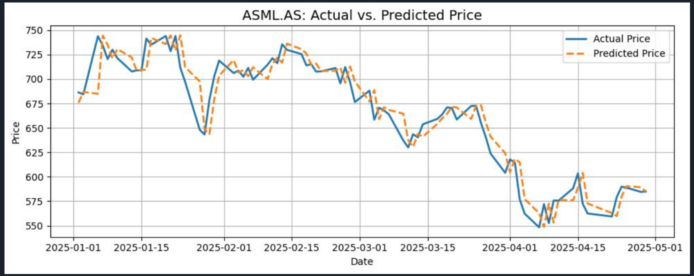


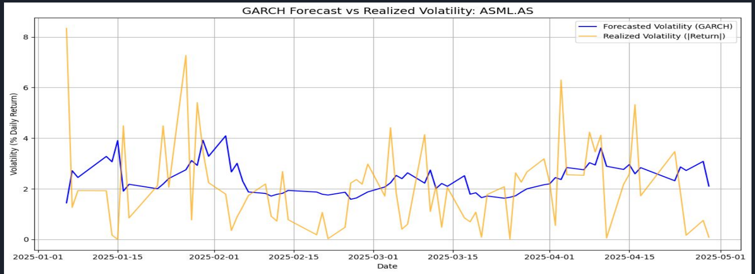


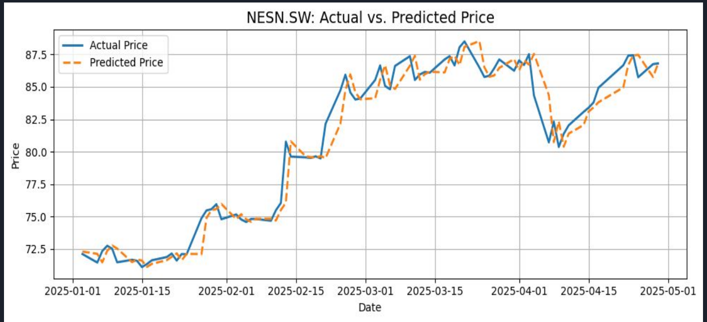


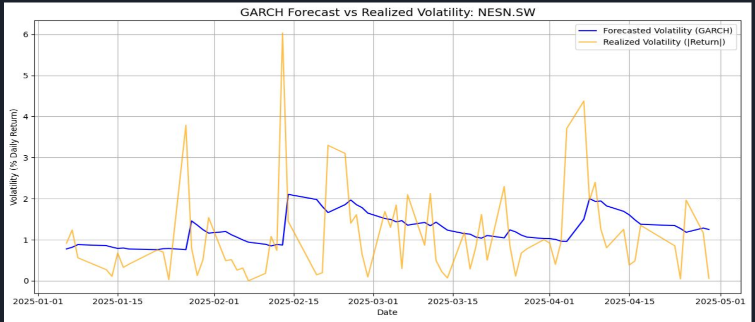


Forecasting Performance: ARIMA & GARCH Actual vs. Predicted









Stock Forecast Accuracy: MAE & RMSE for ARIMA & GARCH

ARIMA

Stock	MAE	RMSE
AAPL	4.091392	5.996919
MSFT	5.508590	7.950594
NVDA	4.018604	5.543244
AMZN	3.804370	5.093458
GOOGL	3.076819	4.055473
META	12.541686	17.143786
TSLA	11.530044	14.922781
BRK-B	5.296579	7.986166
JPM	3.527638	5.121703
V	3.797573	5.794724
SAP.DE	3.971100	5.516563
SIE.DE	3.878375	5.506522
ALV.DE	3.551259	5.233282
VOW3.DE	1.455966	1.924458
LIN.DE	3.831369	5.551702
MC.PA	9.869994	14.028798
OR.PA	4.862815	6.108709
AIR.PA	2.631216	3.548107
ASML.AS	13.1484	17.075604
NESN.SW	0.8481	1.222098

GARCH

Stock	MAE	RMSE
AAPL	1.522514	2.228308
MSFT	1.297048	1.714305
NVDA	2.665115	3.676411
AMZN	1.351453	1.880892
GOOGL	1.270257	1.742215
META	1.386151	2.079603
TSLA	2.400147	3.605987
BRK-B	0.910566	1.269367
JPM	1.201607	1.626328
V	1.057909	1.459868
SAP.DE	1.234072	1.642614
SIE.DE	1.504709	1.943803
ALV.DE	0.889942	1.170593
VOW3.DE	1.168535	1.414243
LIN.DE	1.060477	1.213820
MC.PA	1.344698	1.796585
OR.PA	0.793083	0.984570
AIR.PA	1.153930	1.460713
ASML.AS	1.414906	1.832806
NESN.SW	0.801679	1.136269

Tree-Based Models

XGBoost Model

Data Preprocessing for XGBoost

Feature engineering:

returns, volatility, dollar volume, lags

Target defined:

5-day future return

Data cleaning:

handled missing values & short series

XGBoost Model Architecture:

Input: return-based features (lags, volatility, volume)

Target: 5-day forward return

TimeSeriesSplit with 3 folds (preserves temporal order)

RobustScaler for outlier-resistant normalization

Hyperparameters: 500 trees, LR = 0.05, max depth = 6

XGBoost results

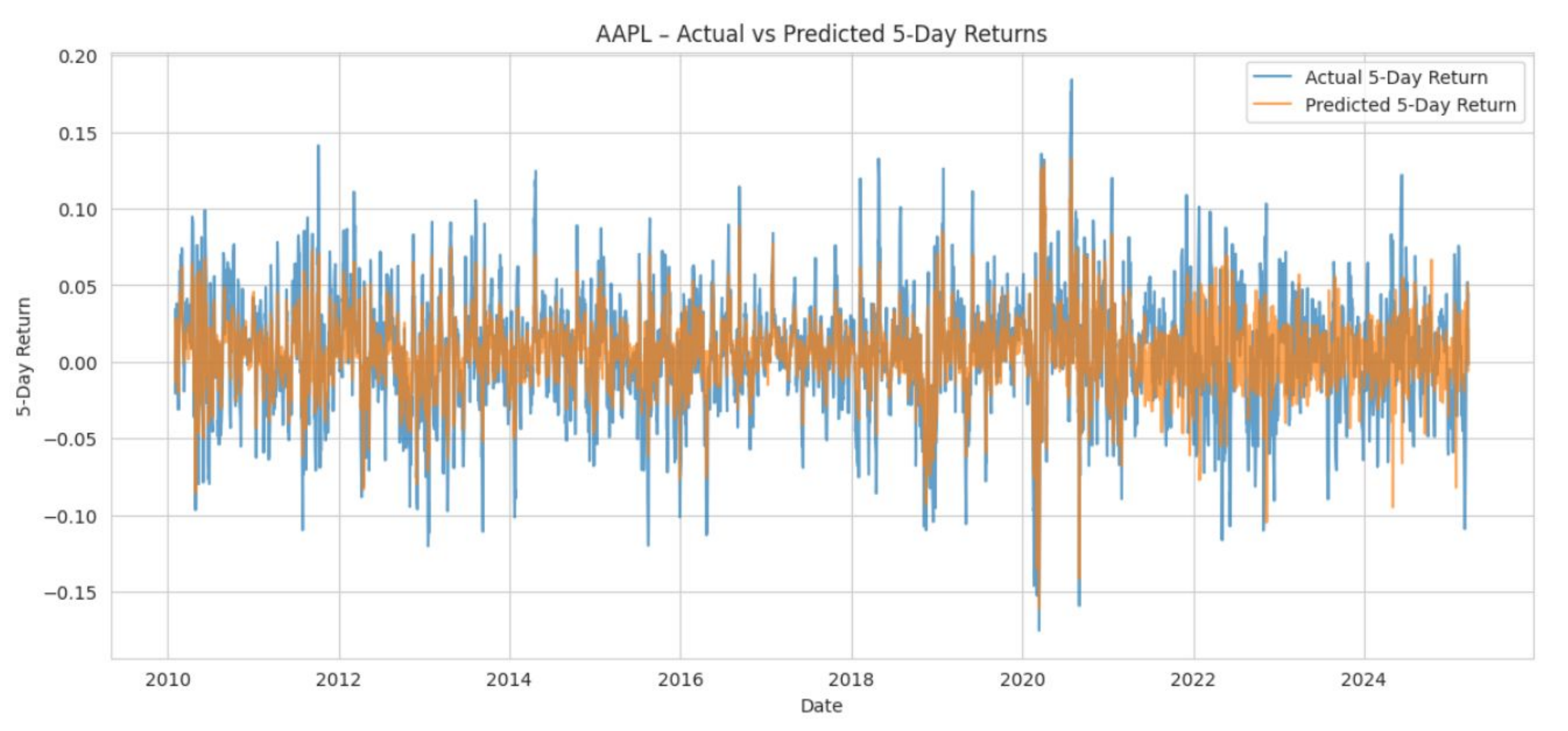
	Ticker	Avg_RMSE	Best_Fold_RMSE	Avg_MAE
19	NESN.SW	0.024639	0.024092	0.018497
7	BRK-B	0.027572	0.021902	0.020670
16	OR.PA	0.032493	0.030755	0.024702
9	V	0.034646	0.033648	0.025686
10	SAP.DE	0.036850	0.032248	0.027676
12	ALV.DE	0.036875	0.031931	0.025408
14	LIN.DE	0.037608	0.002633	0.018640
1	MSFT	0.038694	0.035373	0.029246
8	JPM	0.039785	0.033372	0.029225
11	SIE.DE	0.040361	0.035653	0.030095
0	AAPL	0.042180	0.039316	0.032619
15	MC.PA	0.042427	0.040568	0.032069
4	GOOGL	0.042991	0.038956	0.032431
17	AIR.PA	0.051736	0.044730	0.036435
3	AMZN	0.052627	0.050913	0.039513
18	ASML.AS	0.054921	0.044556	0.041566
13	VOW3.DE	0.055882	0.049084	0.040406
5	META	0.057962	0.050612	0.042245
2	NVDA	0.077435	0.072234	0.058788
6	TSLA	0.103376	0.097739	0.078288

Average RMSE: 0.046516

Average MAE: 0.03421

Better results on stable stocks, higher error on volatile ones (e.g., TSLA, NVDA)

XGBoost results



LightGBM

Preprocessing :

Applied same preprocessing pipeline as XGBoost

Model architecture:

Key Parameters

- n_estimators = 500 (number of boosting rounds)
- learning_rate = 0.05
- max_depth = 6

LightGBM results

	Ticker	Avg_RMSE	Best_Fold_RMSE	Avg_MAE
19	NESN.SW	0.023910	0.022769	0.018100
7	BRK-B	0.026692	0.020760	0.019876
16	OR.PA	0.031532	0.029568	0.024119
9	V	0.034029	0.032628	0.025357
10	SAP.DE	0.035512	0.029853	0.026554
12	ALV.DE	0.035825	0.031211	0.024638
1	MSFT	0.036255	0.034360	0.027318
14	LIN.DE	0.036446	0.004675	0.017527
8	JPM	0.037506	0.031056	0.027910
11	SIE.DE	0.039661	0.034118	0.029411
0	AAPL	0.040534	0.036753	0.031233
15	MC.PA	0.040678	0.039362	0.030852
4	GOOGL	0.041111	0.035796	0.031242
20	AVG_ALL	0.044420	0.037753	0.032669
17	AIR.PA	0.049975	0.041942	0.035089
3	AMZN	0.051169	0.049574	0.038378
18	ASML.AS	0.052920	0.043395	0.040360
13	VOW3.DE	0.053937	0.045771	0.038764
5	META	0.054271	0.042531	0.039552
2	NVDA	0.071271	0.063606	0.054410
6	TSLA	0.095159	0.085337	0.072679

XGBoost

Average RMSE: 0.046516

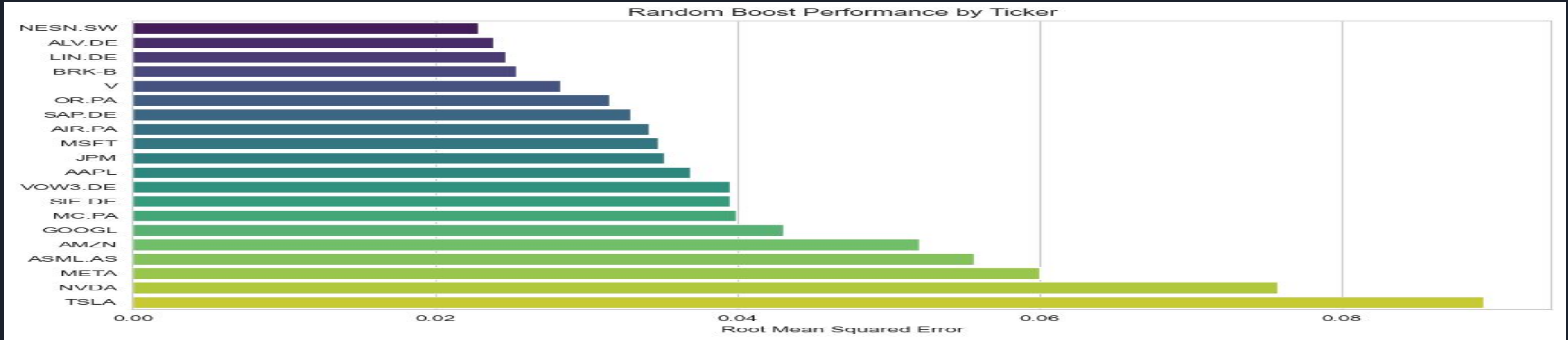
Average MAE: 0.03421

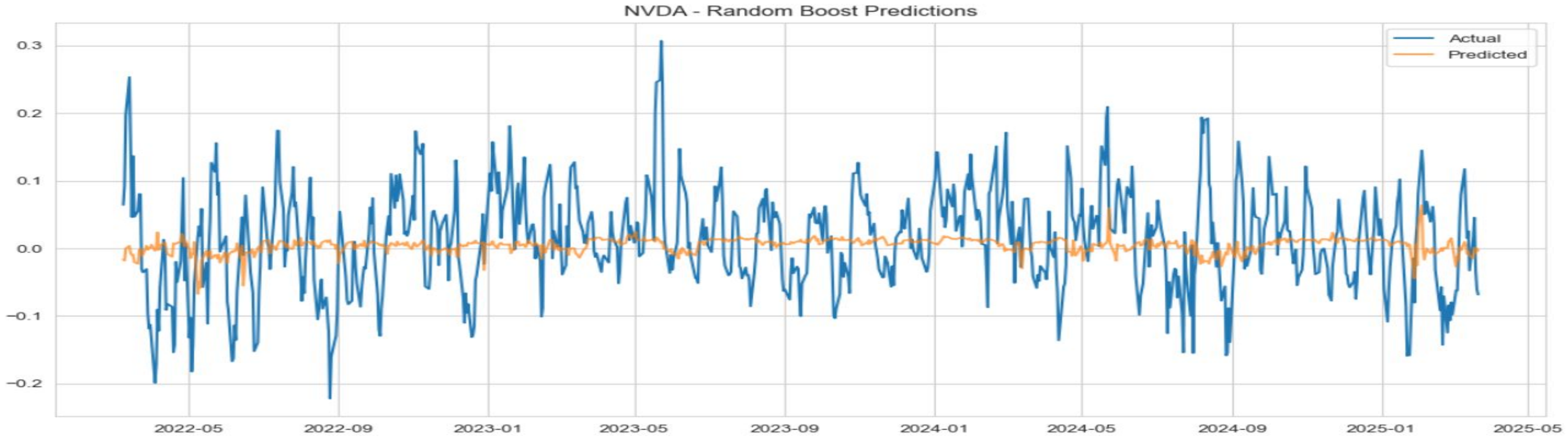
LightGBM

Average RMSE: 0.04592

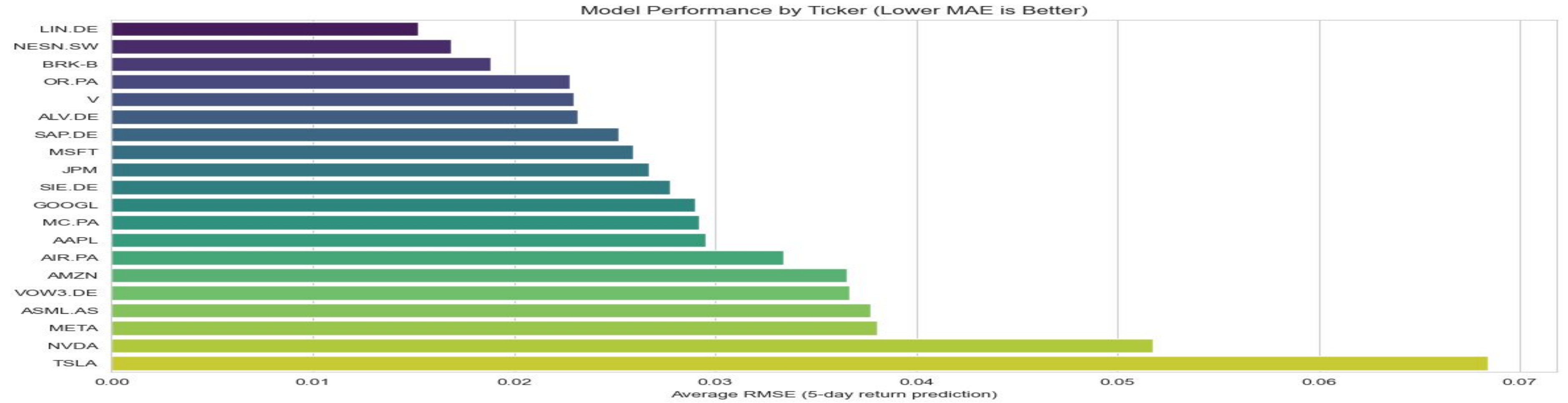
Average MAE: 0.03406

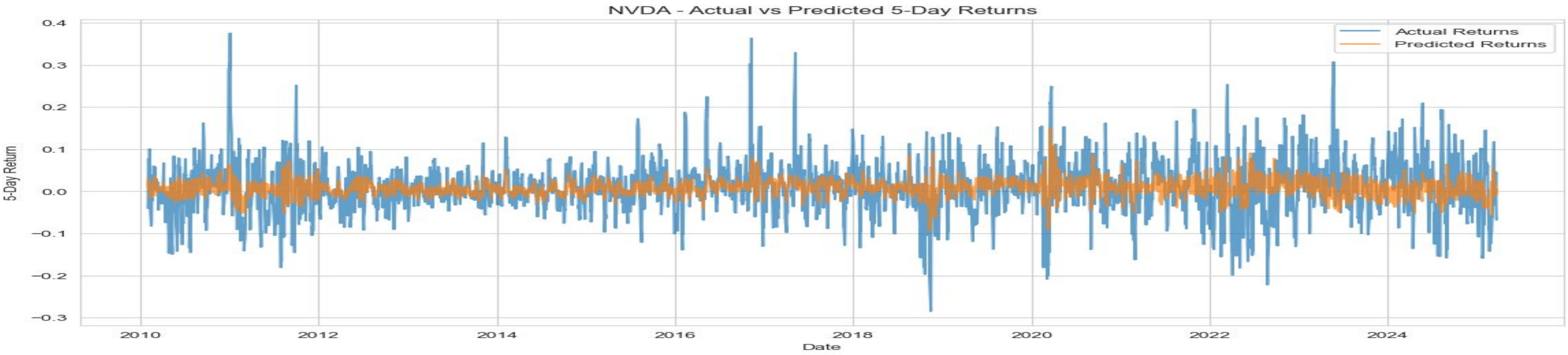
Random Boost





Cat Boost


Model Performance by Ticker (Lower MAE is Better)
LIN.DE
NESN.SW
BRK-B
OR.PA
V
ALV.DE
SAP.DE
MSFT
JPM
SIE.DE
GOOGL
MC.PA
AAPL
AIR.PA
AMZN
VOW3.DE
ASML.AS
META
NVDA
TSLA
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Average RMSE (5-day return prediction)


NVDA - Actual vs Predicted 5-Day Returns
Actual Returns
Predicted Returns
5-Day Return
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
2010
2012
2014
2016
2018
2020
2022
2024
Date

RNN Model

Data Preprocessing for RNN:

Calculated daily returns, 21-day volatility, and 5-day dollar volume

Created lagged return features (up to lag 5)

Defined the 5-day return as the prediction target

Constructed input sequences of length 10 for RNN training

RNN architecture:

SimpleRNN(32, activation='tanh') followed by Dense(1)

Optimizer: Adam | Loss: MSE

EarlyStopping on validation loss (patience=5)

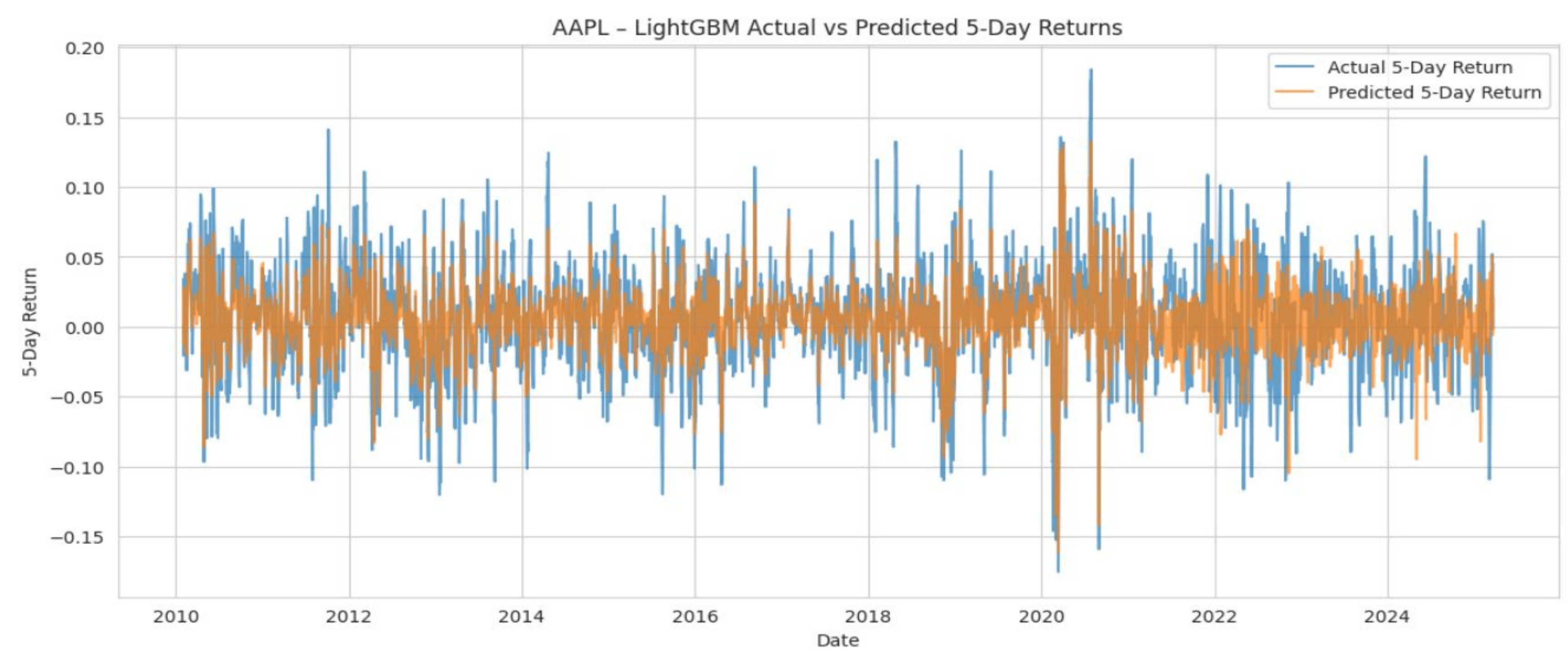
Trained for up to 50 epochs, batch size = 32

Separate model trained for each stock

RNN results

Average RMSE: 0.046516

Average MAE: 0.03421



N-BEATS Model

Data Preprocessing for N-BEATS:

Designed for univariate target forecasting using past features

Used a 30-day input window to form fixed-length sequences

Built custom PyTorch dataset and dataloader

Applied 3-fold time-series cross-validation

N-BEATS architecture:

Deep feedforward network for univariate time series forecasting

Uses stacked blocks (Trend & Seasonality) to model different signal components

Each block outputs a backcast (residual) and a forecast

Fully connected layers, no recurrence or convolution

Allows interpretable decomposition of predictions

N-BEATS results

Average RMSE: 0.04546

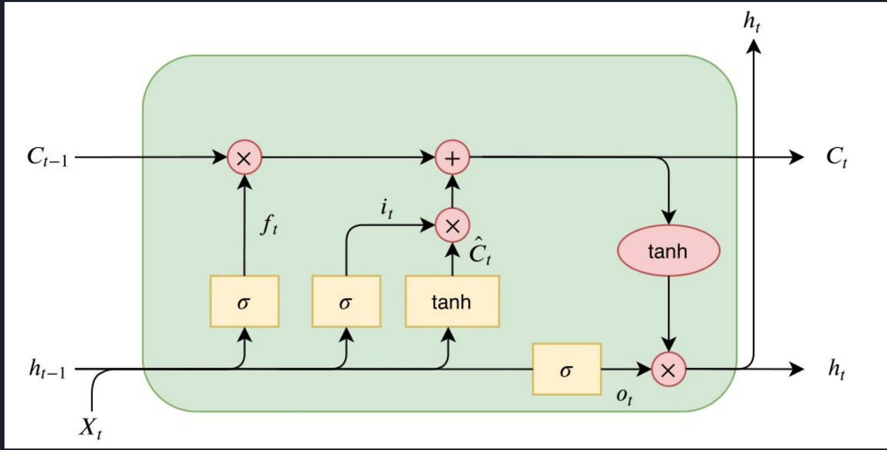
Average MAE: 0.03551

	Ticker	RMSE	MAE
19	NESN.SW	0.026577	0.020945
7	BRK-B	0.029092	0.022028
16	OR.PA	0.034283	0.026451
14	LIN.DE	0.035346	0.016190
12	ALV.DE	0.036568	0.026439
10	SAP.DE	0.036617	0.027290
1	MSFT	0.038346	0.029020
9	V	0.038438	0.029503
8	JPM	0.041774	0.031348
4	GOOGL	0.042158	0.032360
11	SIE.DE	0.043087	0.032918
20	AVG_ALL	0.047476	0.035554
15	MC.PA	0.048986	0.037277
3	AMZN	0.050789	0.038624
0	AAPL	0.051694	0.040888
18	ASML.AS	0.053004	0.040805
13	VOW3.DE	0.056423	0.041422
5	META	0.057567	0.042208
17	AIR.PA	0.058288	0.043734
2	NVDA	0.072096	0.054622
6	TSLA	0.098389	0.076999

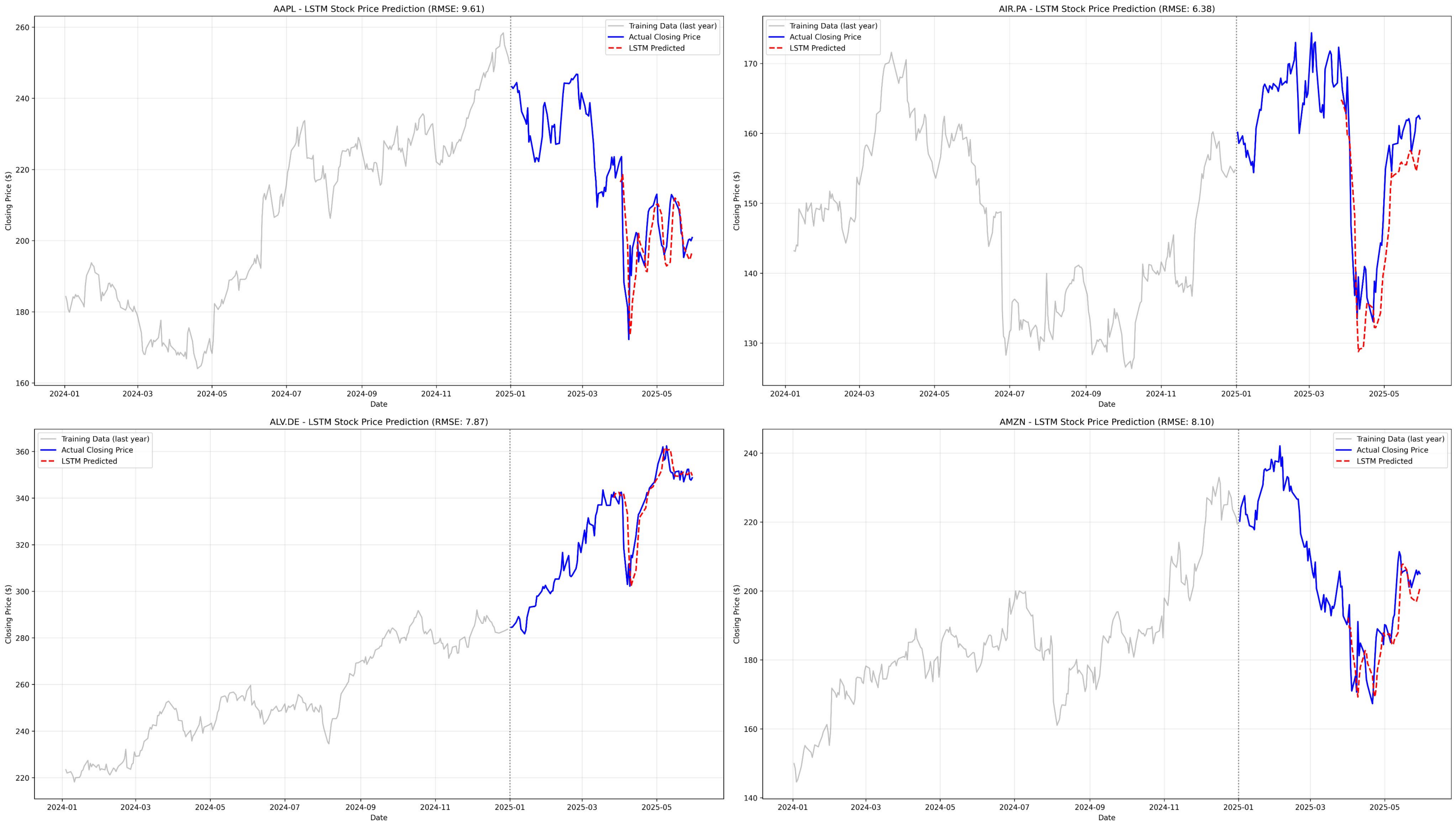
LSTM (Long Short-Term Memory)

Cell State (C): Memory of the network. **Hidden State (h):** Output at each time step. **Forget Gate:** Decides what to forget. **Input Gate:** Decides what to add.,**Output Gate:** Decides what to output

Architecture of lstm for this project

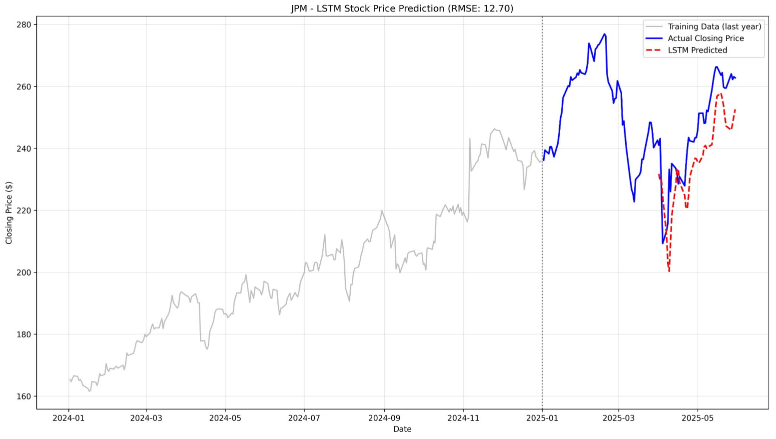


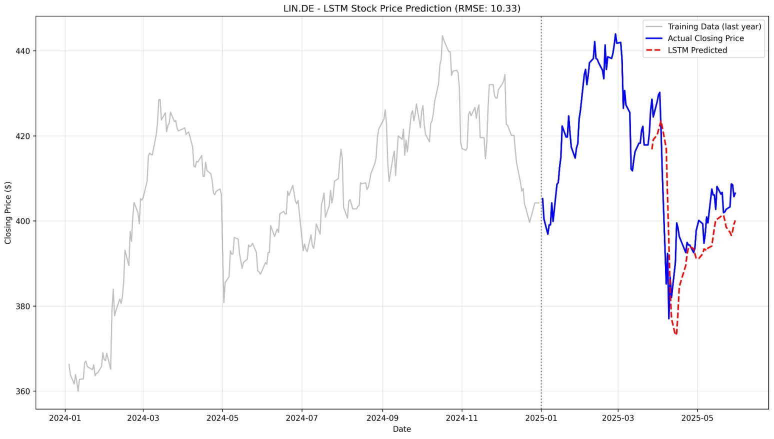
LSTM (Long Short-Term Memory)

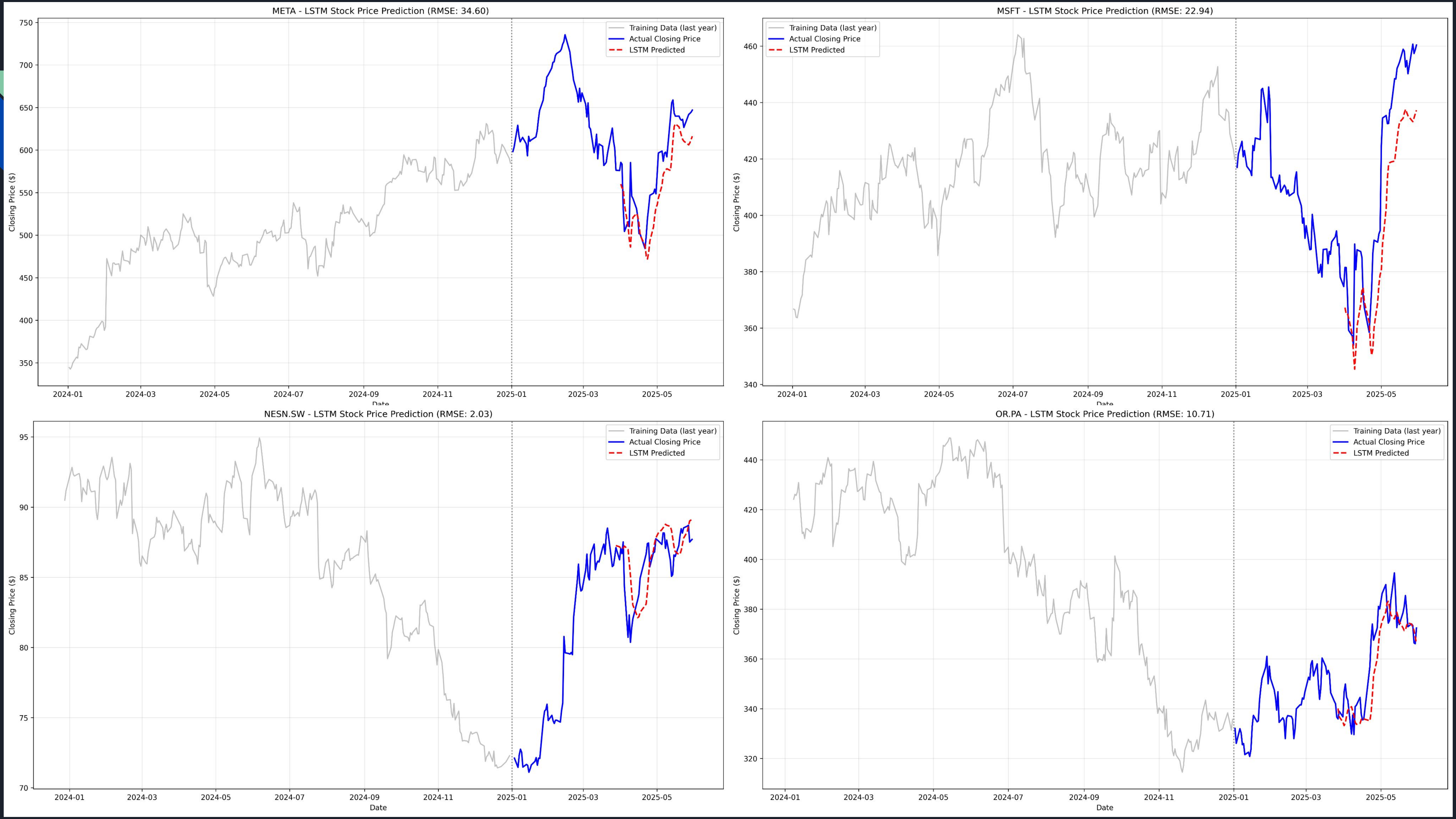



ASML.AS - LSTM Stock Price Prediction (RMSE: 29.83)
Training Data (last year)
Actual Closing Price
LSTM Predicted
Closing Price ($)
Date

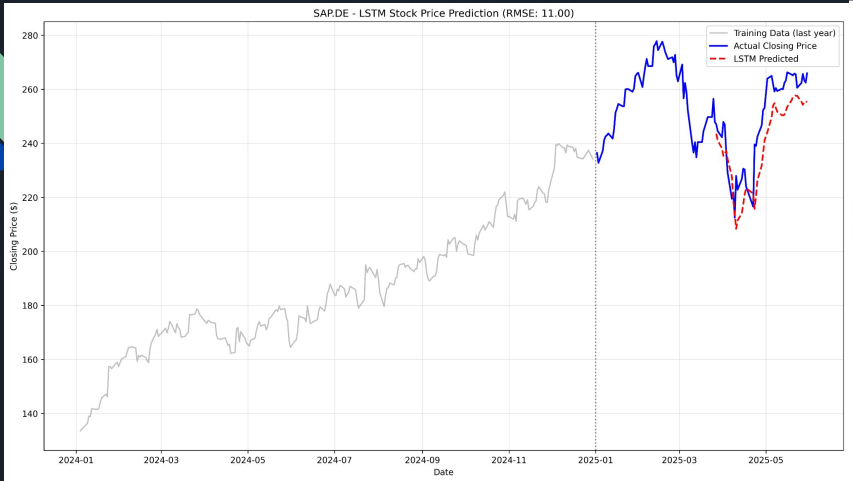

GOOGL - LSTM Stock Price Prediction (RMSE: 5.54)
Training Data (last year)
Actual Closing Price
LSTM Predicted
Closing Price ($)
Date

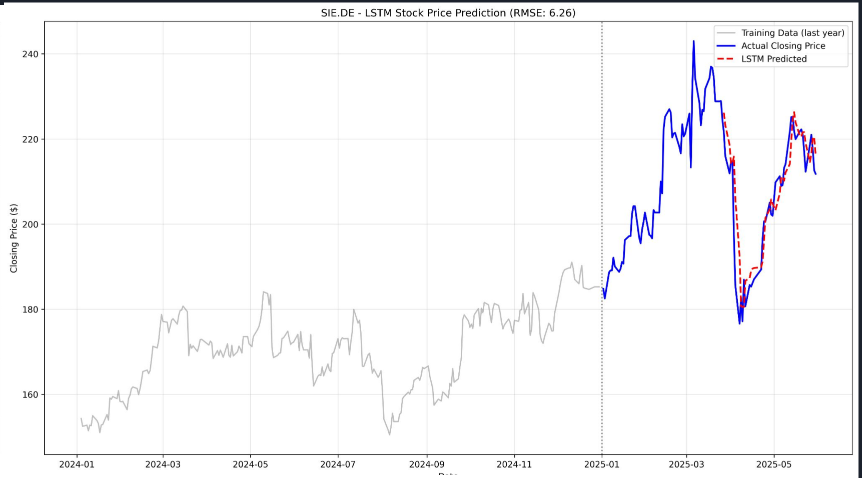

JPM - LSTM Stock Price Prediction (RMSE: 12.70)
Training Data (last year)
Actual Closing Price
LSTM Predicted
Closing Price ($)
Date

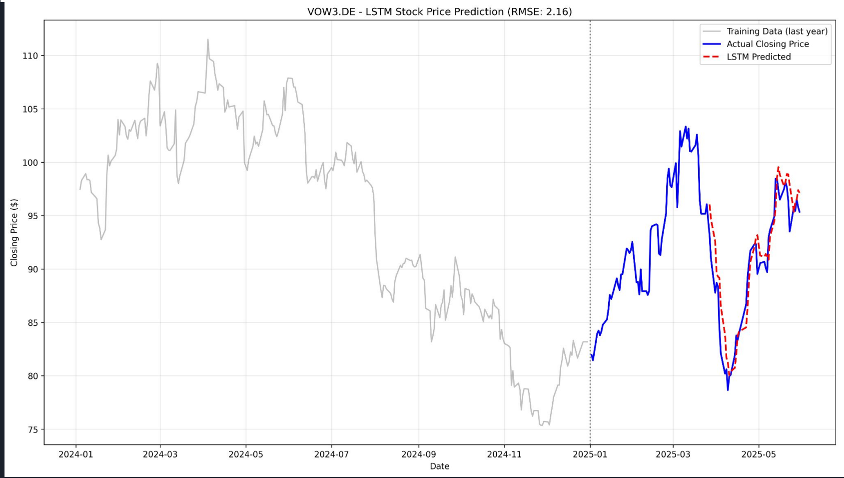

LIN.DE - LSTM Stock Price Prediction (RMSE: 10.33)
Training Data (last year)
Actual Closing Price
LSTM Predicted
Closing Price ($)
Date

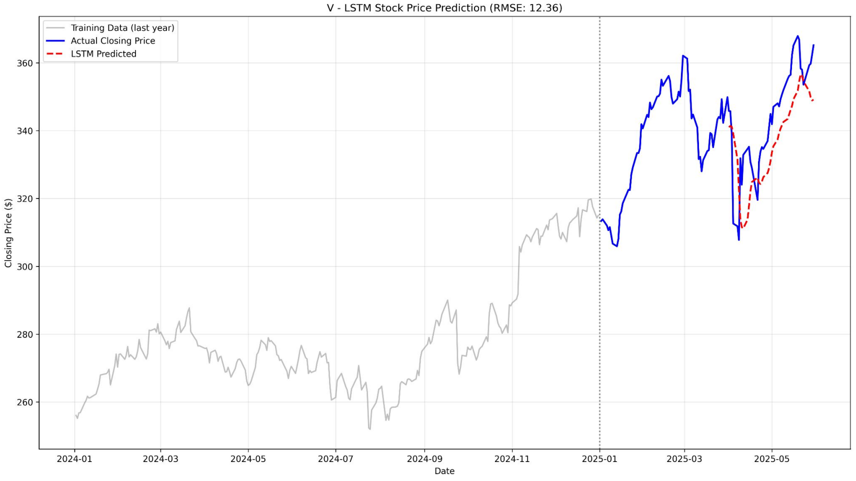


META - LSTM Stock Price Prediction (RMSE: 34.60)
MSFT - LSTM Stock Price Prediction (RMSE: 22.94)
NESN.SW - LSTM Stock Price Prediction (RMSE: 2.03)
OR.PA - LSTM Stock Price Prediction (RMSE: 10.71)
Training Data (last year)
Actual Closing Price
LSTM Predicted
Closing Price ($)
Date


SAP.DE - LSTM Stock Price Prediction (RMSE: 11.00)
Training Data (last year)
Actual Closing Price
LSTM Predicted
Closing Price ($)
Date


SIE.DE - LSTM Stock Price Prediction (RMSE: 6.26)
Training Data (last year)
Actual Closing Price
LSTM Predicted
Closing Price ($)

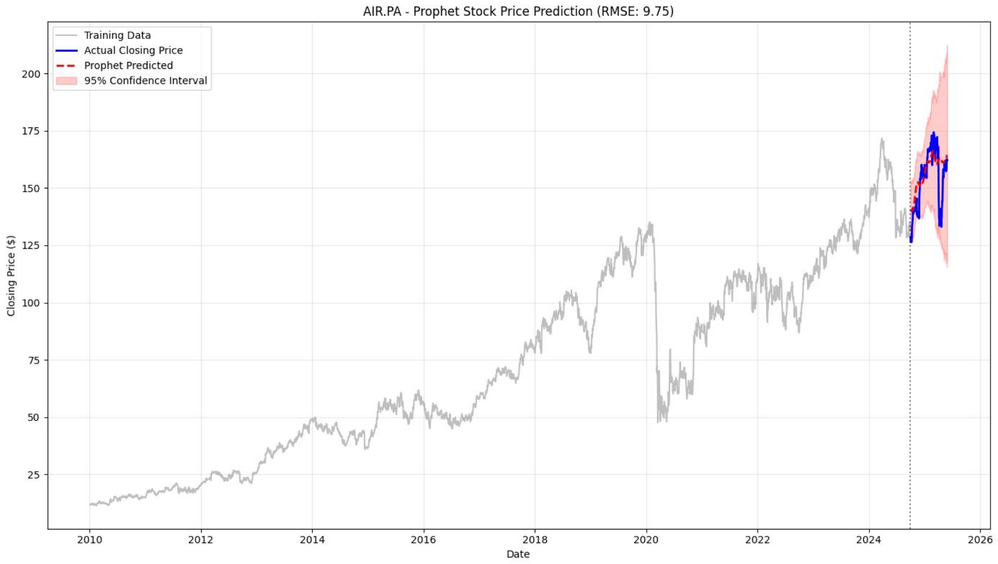

VOW3.DE - LSTM Stock Price Prediction (RMSE: 2.16)
Training Data (last year)
Actual Closing Price
LSTM Predicted
Closing Price ($)
Date

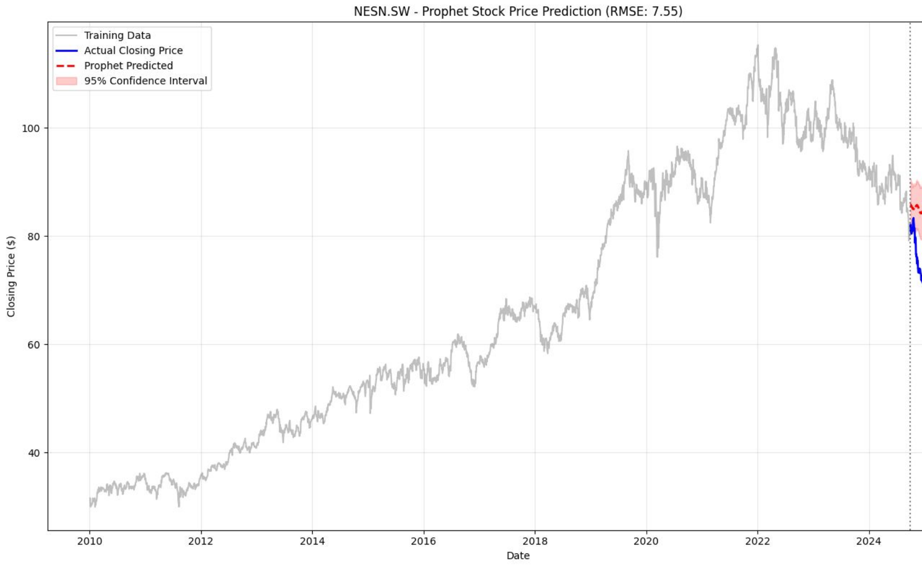

V - LSTM Stock Price Prediction (RMSE: 12.36)
Training Data (last year)
Actual Closing Price
LSTM Predicted
Closing Price ($)
Date

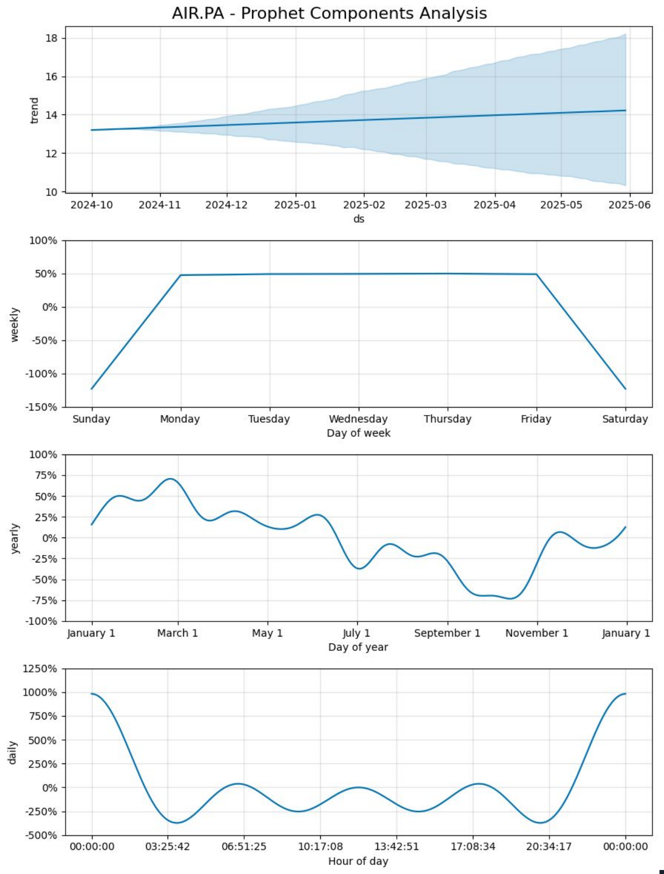
Prophet model- by **Facebook (Meta)**

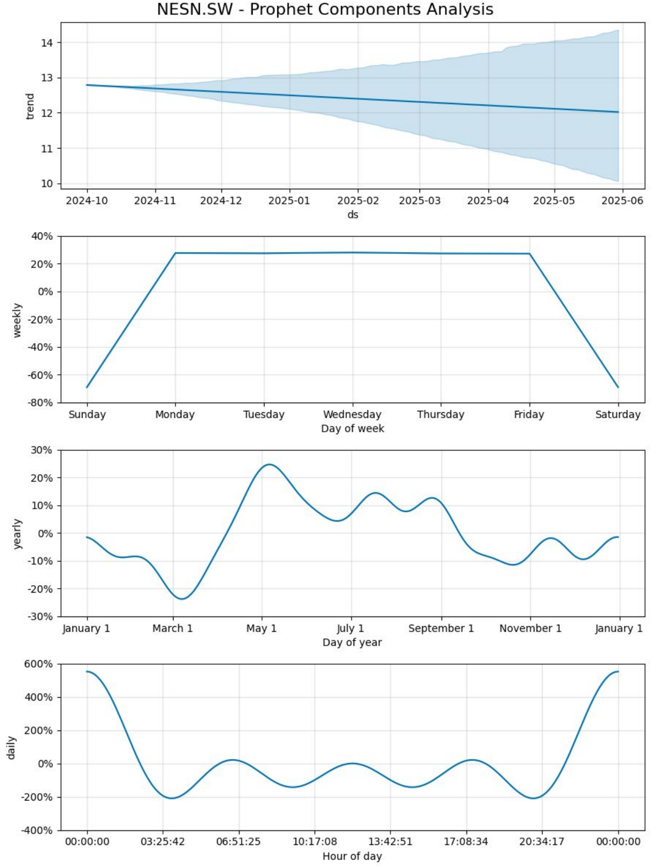
ease of use, scalability, and ability to handle:

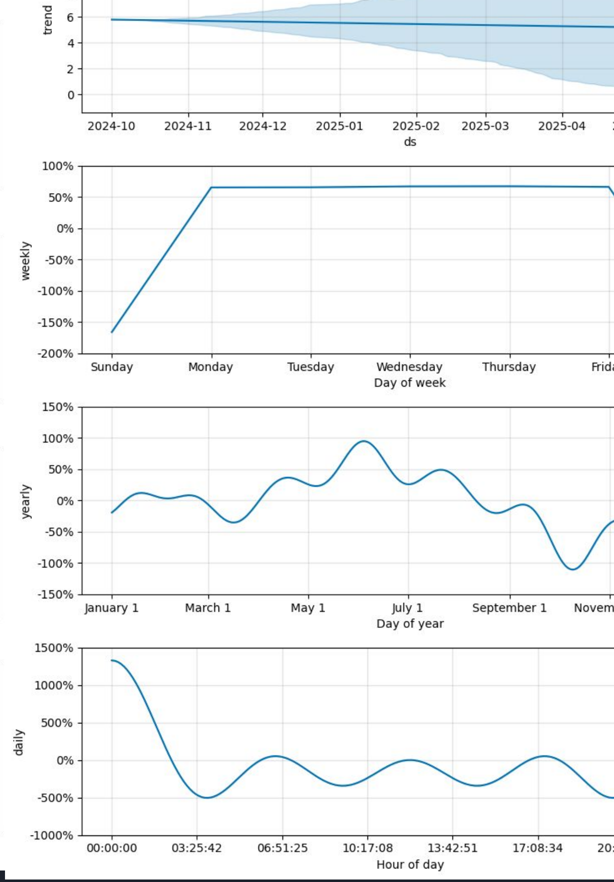
- **Seasonality**
- **Holidays and special events**
- **Missing data**
- **Outliers**
- **Trend changes (changepoints)**






AIR.PA - Prophet Components Analysis
trend
ds
2024-10
2024-11
2024-12
2025-01
2025-02
2025-03
2025-04
2025-05
2025-06
weekly
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of week
yearly
January 1
March 1
May 1
July 1
September 1
November 1
January 1
Day of year
daily
00:00:00
03:25:42
06:51:25
10:17:08
13:42:51
17:08:34
20:34:17
00:00:00
Hour of day


NESN.SW - Prophet Components Analysis
trend
ds
2024-10
2024-11
2024-12
2025-01
2025-02
2025-03
2025-04
2025-05
2025-06
weekly
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of week
yearly
January 1
March 1
May 1
July 1
September 1
November 1
January 1
Day of year
daily
00:00:00
03:25:42
06:51:25
10:17:08
13:42:51
17:08:34
20:34:17
00:00:00
Hour of day


trend
ds
2024-10
2024-11
2024-12
2025-01
2025-02
2025-03
2025-04
weekly
Sunday
Monday
Tuesday
Wednesday
Thursday
Day of week
yearly
January 1
March 1
May 1
July 1
September 1
Day of year
daily
00:00:00
03:25:42
06:51:25
10:17:08
13:42:51
17:08:34
Hour of day

Thank you!