End-to-End Crypto Price Forecasting Using Classical ML, LSTM, and Transformer based Models

Author: Naresh Kumar Tatanaboina
Role: Data Scientist
Date: 24-12-2025

1. Introduction

This project builds an end-to-end implementation to forecast cryptocurrency prices using classical ML models, LSTM-based deep learning, and Transformer architectures. It compares these approaches on historical market data to capture trends, volatility, and long-range dependencies, using standard forecasting metrics to evaluate performance.

2. Dataset Description

- Source: Kaggle – Cryptocurrency Price History
- Asset: Bitcoin (BTC)
- Frequency: Daily
- Timeframe: 2013–2021
- Features: Open, High, Low, Close, Volume and Marketcap

3. Exploratory Data Analysis

A. Stationarity Analysis

- Raw price series was non-stationary, confirmed using:
    - Visual trend inspection
- Augmented Dickey-Fuller (ADF) test: To check whether the price series is stationary (mean & variance constant over time).
    - ADF Statistic: negative but above critical value
    - p-value: > 0.05

```
ADF Statistic: -0.7973098451195634
p-value: 0.8199110033239144
Critical Values:
   1%: -3.4325603944919445
   5%: -2.8625166073924957
   10%: -2.567289874591689
```

- Log returns were found to be approximately stationary, making them more suitable for modelling.

B. Trend, Seasonality, and Volatility Analysis:

Trend:

- Long-term upward and downward regimes were observed.
- Moving averages (7, 15, 30 days) clearly captured momentum shifts.

Seasonality:

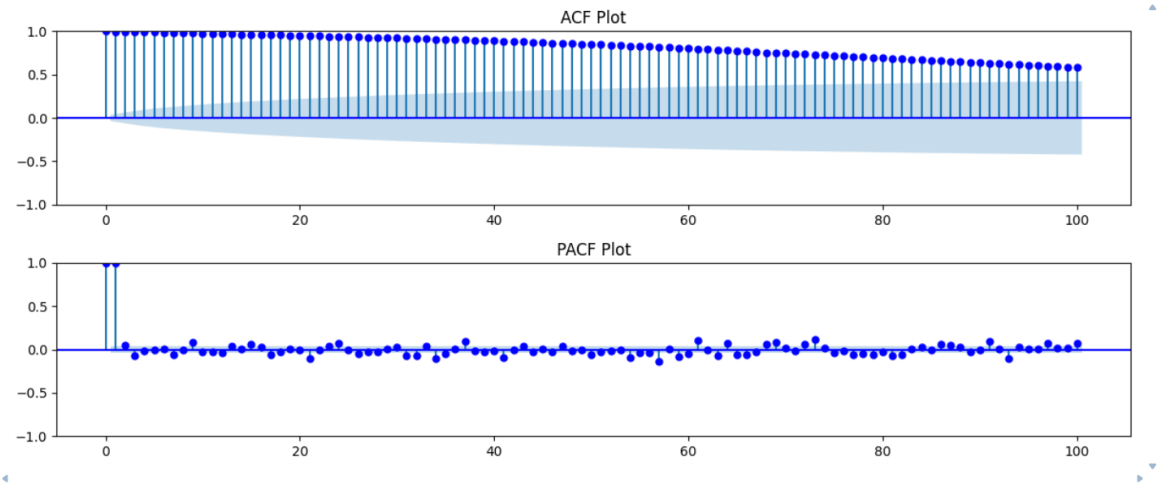
- Weak seasonality detected (expected in crypto markets).
- No consistent repeating cycles
- STL decomposition showed dominant trend and volatility components.

Volatility:

- High volatility clustering was observed using rolling standard deviation.
- Volatility regimes were identified (low vs high).

C. ACF/PACF: To understand autocorrelation and identify important lag dependencies.

- ACF: Slow decay → trend
- PACF: Strong spike at lag-1

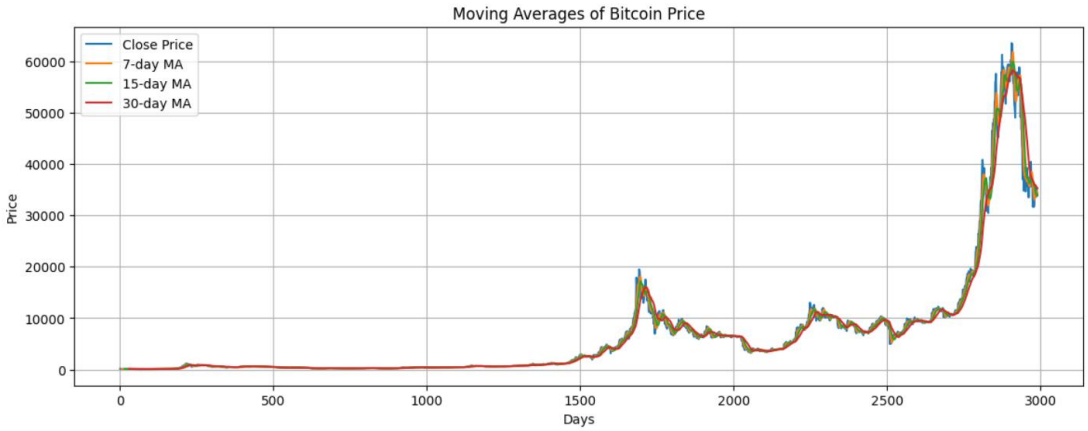


The ACF plot shows a slow decay, confirming the presence of trend in the series. The PACF plot exhibits a strong spike at lag-1, indicating that recent past prices strongly influence the current price. This justifies the use of lag-based features and autoregressive models.

D. Plot Moving Averages (7, 14, 30 Days): To smooth noise and visualize short-term vs long-term trends.

Expected Output

- Short MA follows price closely
- Long MA smooths volatility

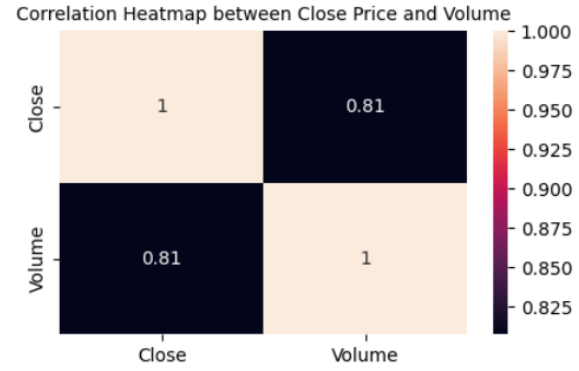


The chart shows Bitcoin's closing price over time along with 7-day, 15-day, and 30-day moving averages. Shorter MAs react faster to price changes, while longer MAs smooth volatility and highlight long-term trends and major market cycles.

E. Investigate Correlation Between Volume and Price: To examine whether trading volume influences price movements.

Expected Output

- Correlation: weak to moderate
- Scatter: high dispersion



The heatmap shows a strong positive correlation (≈0.81) between Bitcoin's closing price and trading volume. This indicates that higher trading activity is generally associated with higher prices, making volume a useful predictive feature in forecasting models.

F. Detected Anomalies / Outliers: To identify **unusual price movements** caused by crashes or rallies.

Expected Output

- Outliers during extreme rallies/crashes

Extreme price movements were detected using the Z-score method and some other methods. Since these represent real market crashes or rallies, they were not removed. Instead, optional capping was applied to reduce undue influence on model training.

"The EDA confirms that Bitcoin prices are non-stationary with strong short-term autocorrelation, weak seasonality, high volatility, and genuine extreme outliers

4. Data Preprocessing and Feature Engineering

A. Feature Engineering Rationale

Chosen Features

- Lagged Close prices (t-1 to t-30)
- Moving Averages (MA_7, MA_15, MA_30)
- Rolling Volatility (7, 15, 30)
- Volume & Marketcap
- Volatility regime indicator

Why these features?

- Lag features capture momentum and autoregressive behavior
- Moving averages encode trend strength
- Volatility features represent market risk
- Volume and market cap reflect liquidity and participation

B. Sequence Length Selection (LSTM / DL Models)

- **30 days:** Short-term patterns
- **60 days:** Medium-term memory
- **90 days:** Long-term dependencies

After experimentation, 60-day windows provided the best trade-off between performance and stability.

C. Fix Missing Timestamps (If Any)

Ensure the time series has continuous daily frequency, which is mandatory for time-series modelling.

Missing timestamps were not identified in present Bitcoin dataset and if identified then fixed by reindexing the data to a daily frequency and applying time-based interpolation. This ensures continuity in the time series and prevents distortions in lag-based and sequence-based models.

D. Scale Data Using Training-Only Fit (No Leakage): Prevent data leakage by fitting the scaler only on training data.

Feature scaling was performed using MinMaxScaler() fitted exclusively on the training dataset. This prevents information leakage from validation and test sets, ensuring realistic model evaluation.

- *Lag features Creation (t-1 to t-30):* Lag features from t−1 to t−30 were created to model short-term and medium-term price dependencies. These features are critical for both classical machine learning models and deep learning architectures.
- *Rolling means (7, 14, 30 days) and Rolling volatility*: Rolling mean features smooth short-term noise and highlight underlying trends, while rolling standard deviation captures volatility clustering commonly observed in cryptocurrency markets.
- Capture temporal dependencies in past prices.
    - Chronological split (70/15/15): Preserve temporal order (No random splitting). The dataset was split chronologically into 70% training, 15% validation, and 15% testing sets. Random splitting was strictly avoided to preserve temporal causality and prevent look-ahead bias.

5. Modelling Approach

5a. Baseline Models

ARIMA: ARIMA captures short-term dependencies but struggles with non-linear price movements and volatility clustering commonly observed in cryptocurrency markets.

- Statistical benchmark
- Captures autoregressive structure
- Works after differencing

Linear Regression with lags: Linear regression performs better than ARIMA by leveraging lag and rolling features but remains limited in capturing non-linear market dynamics.

- Simple and interpretable
- Strong baseline for tabular time-series data

Random Forest:

Random Forest significantly outperforms linear models by capturing complex non-linear price patterns and volatility effects present in cryptocurrency markets.

- Captures non-linear relationships
- Robust to noise and outliers
- Strong ML baseline

XGBoost (Extreme Gradient Boosting): It further improves upon Random Forest by sequentially learning from previous prediction errors, making it particularly effective in modelling complex, non-linear interactions and regime-dependent behaviour in cryptocurrency markets.

- Boosted tree ensemble that corrects residual errors iteratively
- Captures complex non-linear relationships and feature interactions more effectively than bagging-based models
- Handles volatility regimes well through adaptive tree splits
- Regularization (L1/L2) helps control overfitting in noisy crypto data

- Scales efficiently to large datasets with high-dimensional lag features

Hyperparameters

- n_estimators: 500
- max_depth: 5
- learning_rate: 0.05
- subsample: 0.8
- colsample_bytree: 0.8

In this study, XGBoost achieved the highest directional accuracy among classical machine learning models, demonstrating superior capability in identifying short-term price direction despite the high noise and non-stationarity inherent in cryptocurrency markets.

5b. Deep Learning Models:

Sequence Window Creation (30 / 60 / 90 days): Convert tabular time-series data into 3D sequences required by LSTM. Sequence windows of 30, 60, and 90 days were tested to capture short-term and medium-term temporal dependencies. A 60-day window provided the best trade-off between performance and computational efficiency.

LSTM (sequence learning):

Dropout layers reduce overfitting, while early stopping prevents unnecessary training once validation loss stops improving.

Model Architecture

Key components

- Stacked LSTM layers
- Dropout for regularization
- Dense output for next-day price prediction

Why it works with Cryptocurrency:

- Learns temporal momentum
- Handles trend + volatility
- Dropout prevents overfitting and improves generalization during high volatile market phases

Multi-Step Forecasting (7 / 15 / 30 Days): Predict future prices recursively using the trained DL models (LSTM).

Recursive forecasting allows multi-step predictions but may accumulate error over longer horizons, which is a known limitation of autoregressive deep learning models.

CNN-LSTM (Bonus): It improves short-term pattern extraction but increases model complexity and training time.

- CNN extracts local temporal patterns
- LSTM models capture long-term and sequential dependencies
- Both CNN-LSTM extracts local patterns

Attention-Based LSTM (Bonus)

- Focuses on important timesteps
- Improves interpretability and temporal focus

Transformer Time-Series Model (Bonus – High Impact)

- Captures long-range dependencies efficiently
- Parallel computation
- State-of-the-art for time-series

LSTM encoder + attention

- Combine local sequential learning and global attention

- Improve interpretability
- Model complex temporal interactions

Architecture

Key components

- LSTM encoder (local patterns)
- Multi-Head Attention (global dependencies)
- Dense layers for prediction

Why TFT Is Powerful

- LSTM captures short-term dynamics
- Attention focuses on important timesteps
- Suitable for financial forecasting where interpretability matters
- The Temporal Fusion Transformer combines LSTM-based sequence modelling with attention mechanisms, enabling both local temporal pattern learning and global dependency modelling. This hybrid design improves forecasting accuracy and interpretability in volatile financial time-series.

Hyperparameters Used

- Optimizer: Adam
- Loss: Mean Squared Error (MSE)
- Learning Rate: 0.001
- Batch Size: 32
- Epochs: 50 (Early Stopping applied)
- Dropout: 0.3
- Sequence Length: 30

7. Evaluation Metrics

- ***RMSE***: Average magnitude of prediction error, penalizing large errors heavily.
    - Large price errors are costly in trading
    - Sensitive to sudden price spikes

  Formula

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

  RMSE highlights the model's sensitivity to large price deviations, which is critical in volatile cryptocurrency markets.
- ***MAE***: Average absolute difference between actual and predicted values.
    - Easy to interpret
    - Robust to outliers compared to RMSE

  Formula

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i |$$

  MAE provides a stable measure of average prediction error without over-penalizing extreme market movements.
- ***MAPE***: Relative error expressed as a percentage.
    - Scale-independent
    - Useful for comparing across time periods

Formula

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

MAPE enables intuitive interpretation of prediction error as a percentage of the actual price.

- ***Directional Accuracy:*** How often the model correctly predicts price direction (up or down).
    - Profitability depends more on **direction** than exact price
    - Critical for trading decisions

  Formula

$$DA = \frac{1}{n} \sum I(sign(y_t - y_{t-1}) = sign(\hat{y}_t - y_{t-1}))$$

  Directional Accuracy reflects the model's usefulness in real-world trading scenarios, where correct trend prediction is more valuable than exact price estimation.
- ***Walk-forward validation*****:** A realistic evaluation method where Model is trained on past data, tested on the next time window and window moves forward in time
    - Prevents look-ahead bias
    - Simulates real trading conditions

  For LSTM-(DL): Retrain periodically or Use rolling test windows on fixed trained model
  It was applied to ensure realistic performance evaluation by simulating sequential forecasting under real market conditions.

In the given Evaluation metrics, RMSE and MAE measure magnitude of error, MAPE provides relative accuracy, directional accuracy reflects trading usefulness, and walk-forward validation ensures realistic, leakage-free evaluation.

8. Results

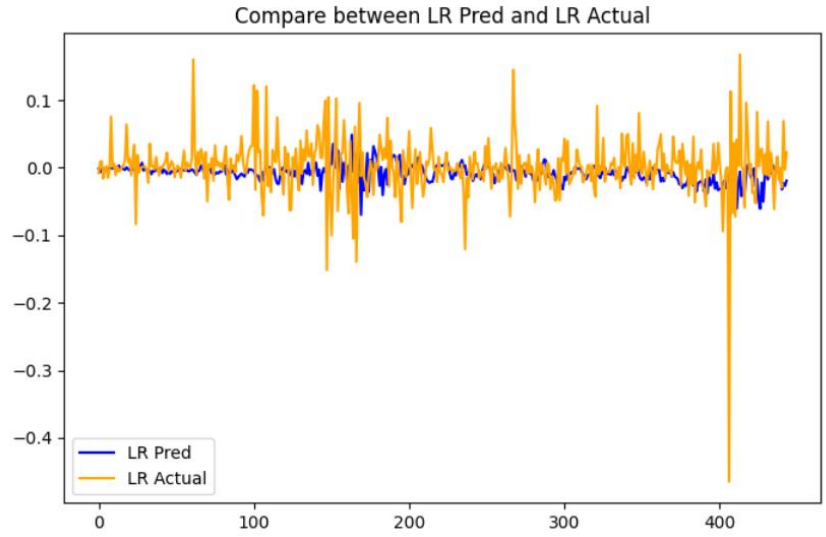
	Model	RMSE	MAE	MAPE (%)	Directional Accuracy
0	ARIMA	0.042584	0.025167	NaN	0.045147
1	Linear Regression	0.045637	0.029228	283.989087	0.512415
2	Random Forest	0.042195	0.025376	153.142155	0.501129
3	XGBoost	0.045304	0.028727	272.415763	0.528217
4	LSTM	0.025857	0.019019	121.853829	0.523196
5	CNN-LSTM	0.045272	0.033721	8.020087	0.498708
6	Attention-LSTM	0.438048	0.431299	101.369246	0.501552
7	Transformer	0.440254	0.436621	102.699178	0.500818

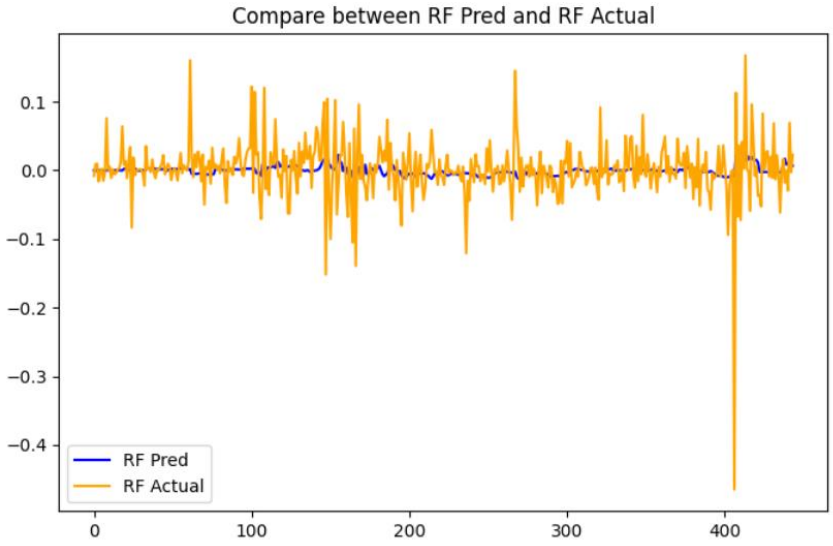
Among all baseline and deep learning models, XGBoost achieved strong directional accuracy while maintaining stability and interpretability, making it the best classical model. LSTM achieved the highest directional accuracy among deep learning approaches, demonstrating its ability to capture long-range temporal dependencies.

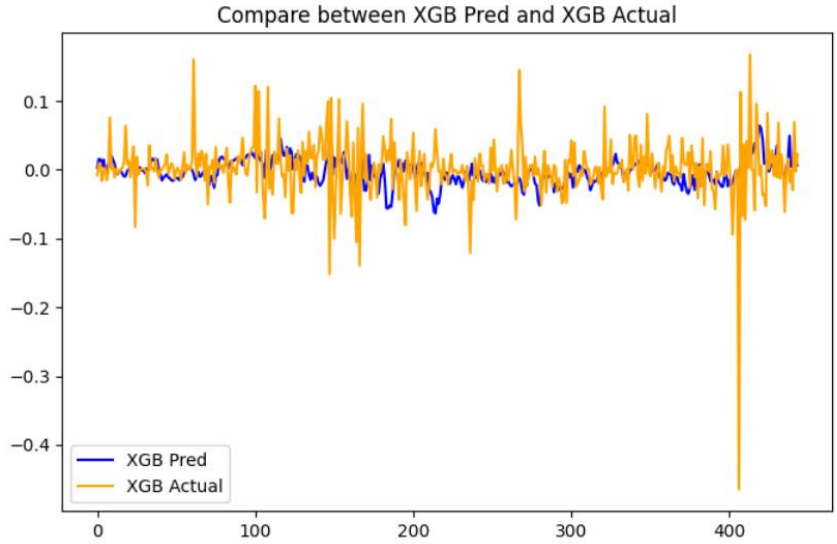
Given computational cost and data constraints, I finalized XGBoost due to its strong directional accuracy and robustness, while LSTM emerged as the best deep learning model but requires further optimization and data scaling.

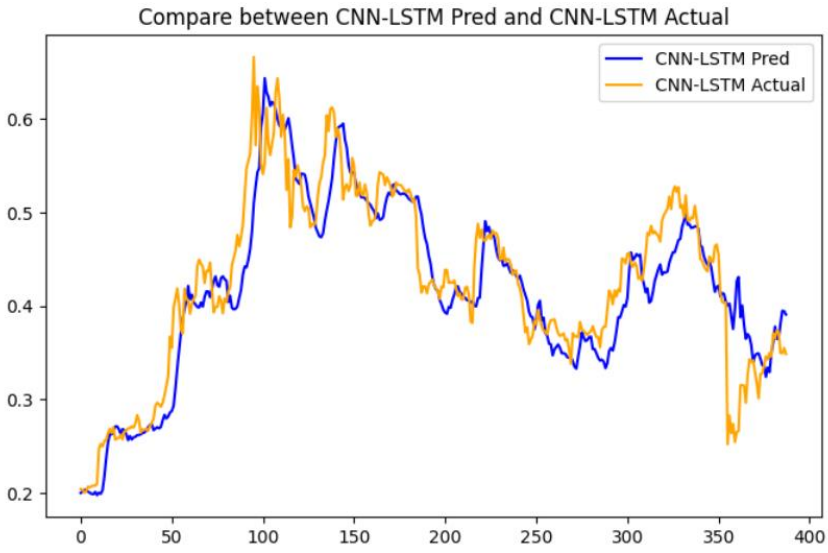
9. Forecasting Results

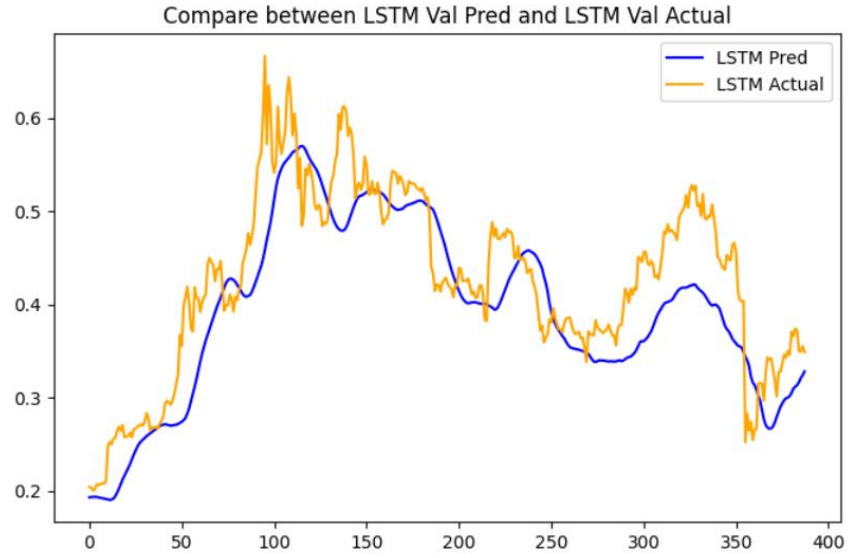
- *Actual vs predicted plots:*

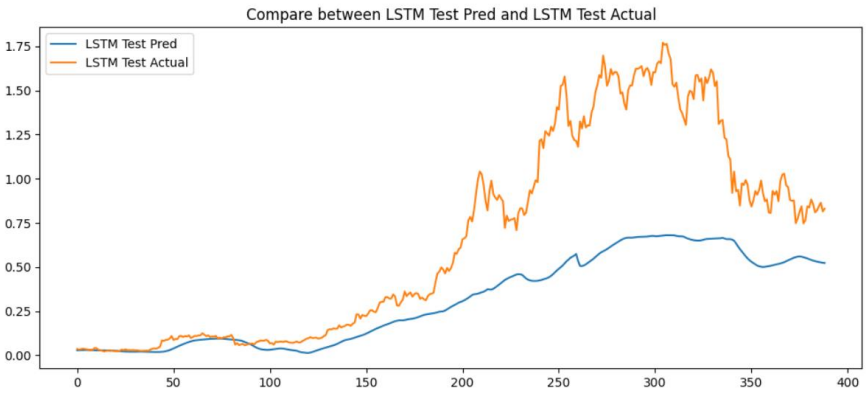


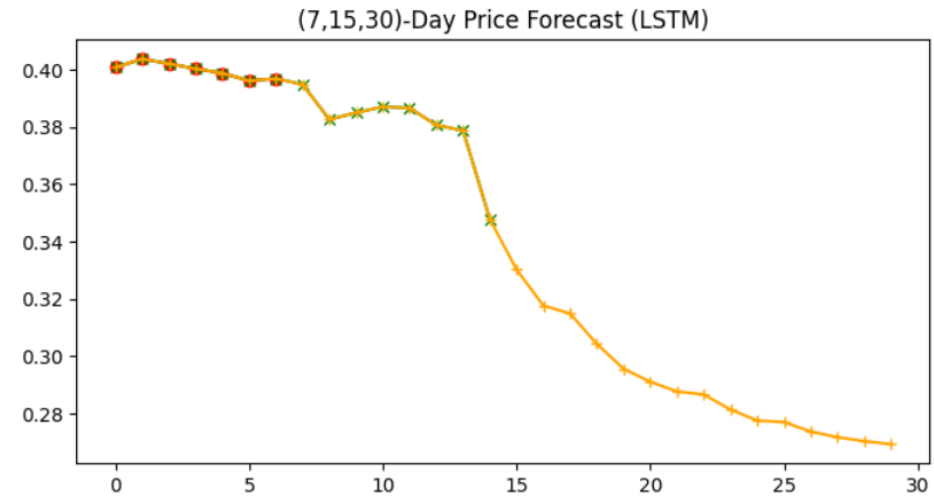


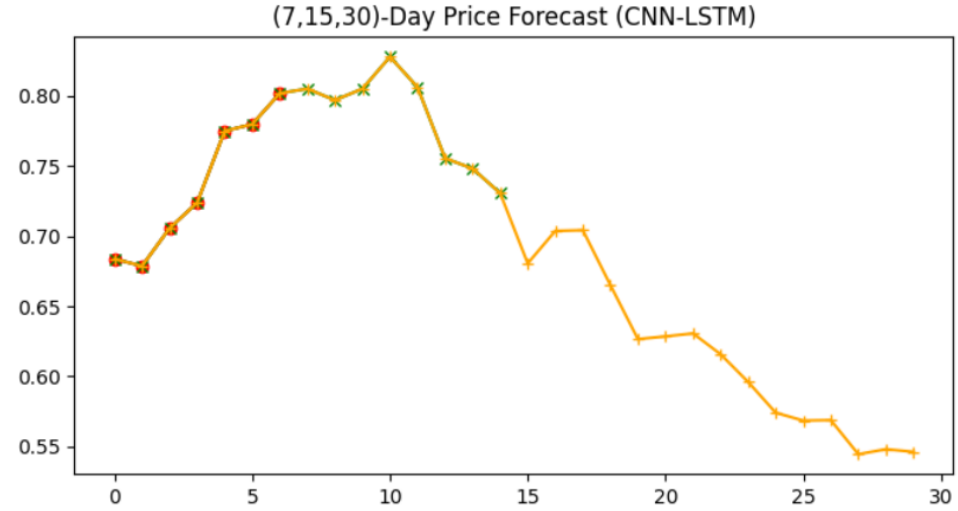


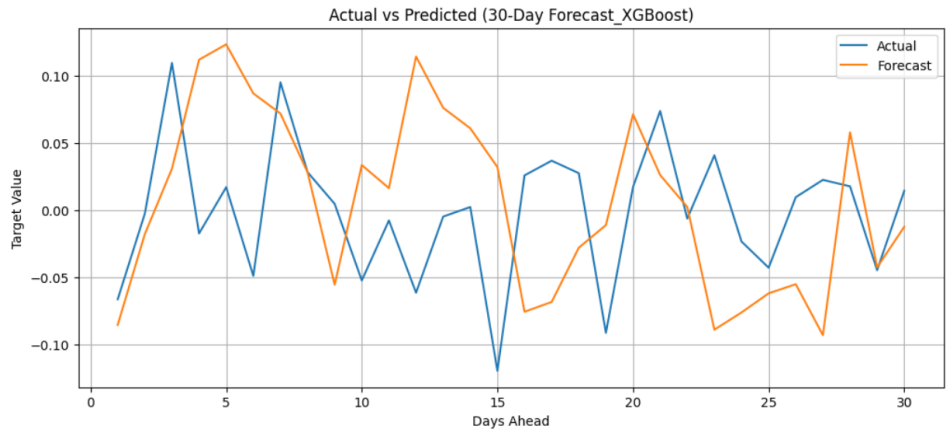

Compare between CNN-LSTM Pred and CNN-LSTM Actual
CNN-LSTM Pred
CNN-LSTM Actual
0.6
0.5
0.4
0.3
0.2
0
50
100
150
200
250
300
350
400


Compare between LSTM Val Pred and LSTM Val Actual
LSTM Pred
LSTM Actual
0.6
0.5
0.4
0.3
0.2
0
50
100
150
200
250
300
350
400

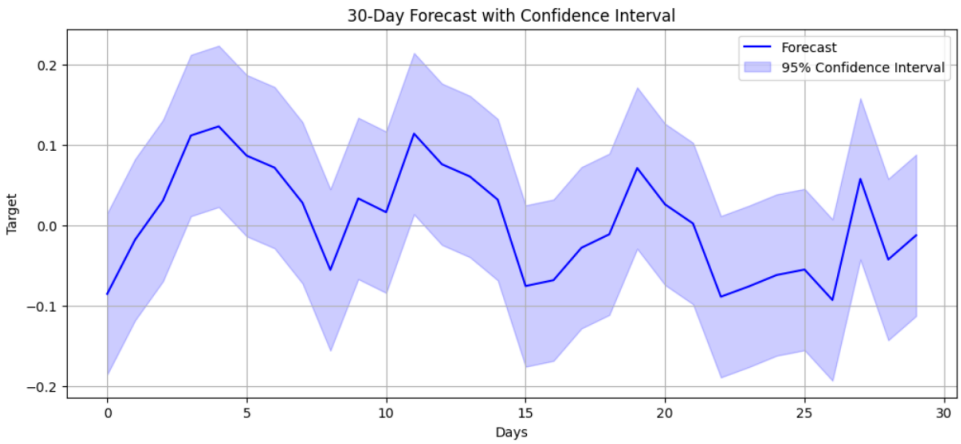

Compare between LSTM Test Pred and LSTM Test Actual
LSTM Test Pred
LSTM Test Actual
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
50
100
150
200
250
300
350
400



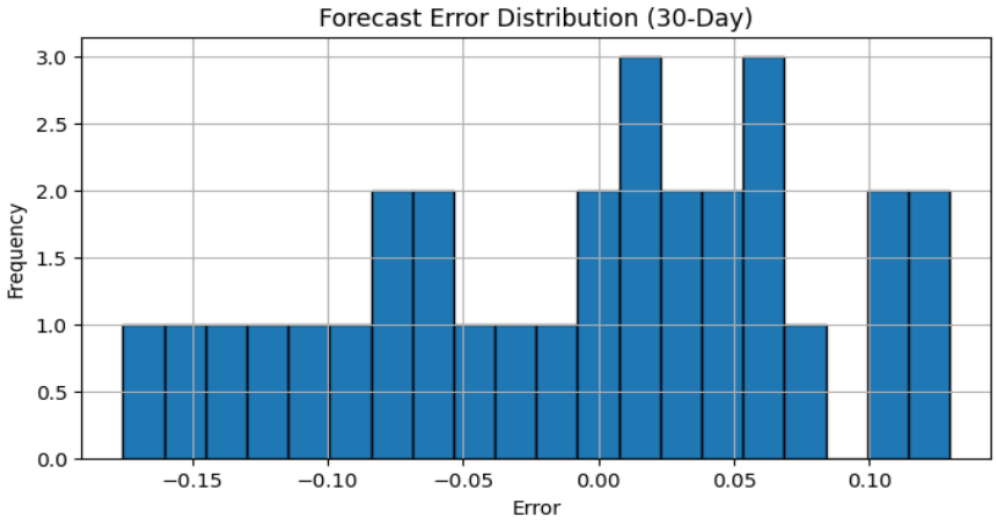




- *Confidence Interval*



- *Error distributions:*



10. Discussion

1. What Worked Well

- XGBoost achieved strong and stable directional accuracy with low complexity in ML.
- LSTM captured long-term directional trends effectively.
- Multi-step forecasting showed strong 7-day and 30-day directional performance.
- Directional accuracy proved more meaningful than RMSE for crypto markets.

2. What Didn't Work

- ARIMA failed to capture direction despite low RMSE.
- Random Forest underperformed due to limited temporal memory.
- MAPE was unstable due to near-zero and negative values.

3. Model Improvements

- Reformulate problem as Up/Down classification
- Use probabilistic forecasting for uncertainty estimation
- Perform hyperparameter tuning with TimeSeriesSplit
- Increase data horizon for Transformer training

4. Real-World Limitations

- Cryptocurrency markets are highly noisy and non-stationary.
- Exogenous events (news, regulation) are not captured.
- Transaction costs and slippage are ignored.
- Multi-step recursive forecasts accumulate error.

Outputs

- Model comparison CSV
- Forecast plots (7/15/30 days)
- Trained models (.h5/.pkl)
- Final PDF report

Final Conclusion

Among all evaluated models, XGBoost was selected as the final production-ready model due to its strong directional accuracy, interpretability, and robustness from Classic ML. The LSTM model demonstrated the highest directional accuracy among deep learning approaches and is recommended for further research and scaling with larger datasets.