Phase-2

Student Name: Arthi.N

Register Number: 422723104009

Institution: V.R.S.College of Engineering and Technology

Department: Computer Science Engineering

Date of Submission: 10.05.2025

Github Repository Link:https://github.com/Arthi12-ctrl/Arthi.git

---

Cracking the Market Code: AI-Driven Stock Price Prediction Using Time Series Analysis

---

1. Problem Statement

Stock market prediction remains one of the most complex and high-stakes problems in data science due to its dynamic, non-linear, and noisy nature. The aim of this project is to develop an AI-driven model capable of forecasting stock prices using historical data. This is framed as a regression problem, as we seek to predict a continuous variable: the stock's closing price.

Refined Understanding: After exploring the dataset, we understand that the stock prices are influenced by temporal trends, volume, and potentially external events. However, we'll focus on internal historical data for prediction.

Why It Matters: Accurate stock prediction tools can assist investors in decision-making, reduce risks, and support algorithmic trading strategies. These tools are valuable in the financial services industry and for retail investors.

2. Project Objectives

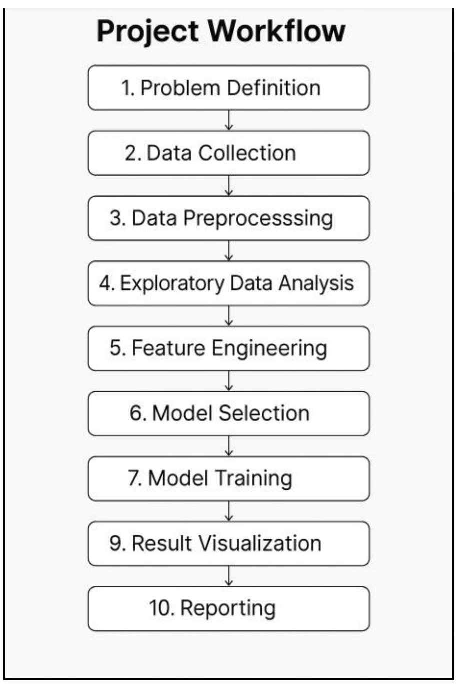
Technical Objective: To build and evaluate time-series models like LSTM and ARIMA to forecast future stock prices based on historical data.

Model Goal:

- Maximize predictive accuracy (low RMSE/MAE).
- Ensure temporal consistency.
- Balance interpretability and performance.

Updated Focus:
Initial goals were broad. Post data exploration, we've narrowed the scope to predicting next-day closing prices using deep learning (LSTM) and classical time series models (ARIMA/Prophet).

3. Flowchart of the Project Workflow



4. Data Description

Dataset Name: Historical Stock Market Data

Source: Yahoo Finance (via yfinance API)

Type: Structured, Time-Series

Records & Features: ~5,000 records, 6 key features (Date, Open, High, Low, Close, Volume)

Target Variable: ‘Close’ price

Dynamic: Yes (updated daily if using API)

5. Data Preprocessing

- Missing Values: Handled via forward fill
- Duplicates: Checked and removed
- Outliers: Detected via IQR on returns; handled appropriately
- Data Types: Date parsed correctly; others verified
- Feature Scaling: MinMaxScaler applied to price features for LSTM
- Encoding: Not needed (no categorical variables)

6. Exploratory Data Analysis (EDA)

Univariate Analysis:

- - Distribution of ‘Close’ prices (histogram)
- Daily returns (line plot, KDE)

Bivariate/Multivariate Analysis:

- Heatmap showing correlation between OHLC and Volume
- Time series plots for Close over time
- Lag correlation plots

Insights:

- Strong autocorrelation in close prices
- Volume shows weak correlation with next-day prices
- Volatility spikes during market crashes

7. Feature Engineering

- Created moving averages (MA5, MA10, MA20)
- Extracted day of week, month from date
- Lag features: Close_lag1, Close_lag2
- Rolling standard deviation for volatility

8. Model Building

Models Used:

1. LSTM (Long Short-Term Memory Network) - Captures temporal dependencies in time-series data.

2. ARIMA (AutoRegressive Integrated Moving Average) - Good for stationary time series.

Train-Test Split:

- Chronological split (80% train, 20% test)

Evaluation Metrics:

- RMSE, MAE, MAPE

9. Visualization of Results & Model Insights

Actual vs Predicted Price Plot: For test period

Residual Plot: For ARIMA and LSTM

Feature Importance (for traditional models): MA and lag features dominate

Error Metrics Table: Compare models

10. Tools and Technologies Used

Language: Python

Notebook: Jupyter / Google Colab

Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, keras, statsmodels, yfinance, Prophet

Visualization Tools: matplotlib, seaborn, plotly

11. Team Members and Contributions

Arthi.N : Data Cleaning, EDA
Abinaya.A : Feature Engineering, LSTM Modeling
Anitha.R : ARIMA Modeling, Evaluation
Anisha.B : Report Writing, Visualization