

Phase-1 Submission

Student Name: Ramkishor.S

Register Number: 410723104064

Institution: Dhanalakshmi College of Engineering

Department: Computer Science Engineering

Date of Submission: 30.04.2025

1.Problem Statement

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

2.Objectives of the Project

The main goal of this project is to develop an AI-powered model capable of predicting future stock prices based on historical stock market data. By leveraging time series analysis and machine learning/deep learning techniques, the project aims to assist investors and analysts in making more informed trading decisions and identifying patterns or trends that may not be visible through traditional analysis.

3.Scope of the Project

- Data Collection & Pre-processing – Gathering historical stock market data, cleaning, and normalizing it for analysis.
- Exploratory Data Analysis (EDA) – Visualizing trends, seasonality, and volatility in stock prices.

- Feature Engineering – Extracting relevant features (e.g., moving averages, RSI, MACD) to improve model performance.
- Model Development – Implementing and comparing:
 - Traditional time series models (ARIMA, SARIMA, GARCH)
 - Machine learning models (Random Forest, XGBoost)
 - Deep learning models (LSTM, GRU, Transformers)
- Model Evaluation – Using metrics like RMSE, MAE, and Sharpe Ratio to assess performance.
- Deployment (Optional) – Developing a web-based dashboard or API for real-time stock predictions.

4.Data Sources

- Yahoo Finance API – Free historical stock price data (OHLC - Open, High, Low, Close).
- Alpha Vantage – Provides real-time and historical stock data along with technical indicators.
- Quandl – Offers financial and economic datasets (e.g., NASDAQ, NYSE data).
- Kaggle Datasets – Pre-processed stock market datasets for benchmarking.

- Tiingo – Alternative API for stock prices, fundamentals, and crypto data.
- Dataset link: <https://tradingeconomics.com/united-states/stock-market>

5.High-Level Methodology

- **Data Collection** –We will collect historical stock market data (OHLC - Open, High, Low, Close prices) from sources such as Yahoo Finance API, Alpha Vantage, Quandl, and Kaggle. These sources provide both raw and pre-processed financial datasets. For advanced features like technical indicators (RSI, MACD), APIs like Alpha Vantage and Tiingo will be used. Optionally, we may also gather news and sentiment data using NewsAPI or Twitter API for sentiment analysis.
- **Data Cleaning** –The raw data may contain missing values, incorrect entries, or inconsistent formats. We'll handle missing data using forward-fill/backward-fill or interpolation techniques. Duplicates will be removed, and all timestamps will be converted to a consistent datetime format. Price and volume values will be normalized for modeling. This will be done using Python libraries such as Pandas, NumPy, and Scikit-learn.
- **Exploratory Data Analysis (EDA)** –EDA will help uncover trends, volatility patterns, seasonality, and anomalies in stock prices. We'll use visualizations like line plots for price trends, box plots for volatility, and heatmaps for correlation among technical indicators. Libraries such as Matplotlib, Seaborn, and Plotly will support visual exploration. This step helps in selecting meaningful features and shaping the modeling strategy.
- **Feature Engineering** – We will create features such as moving averages (SMA, EMA), Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Bollinger Bands, and lagged returns. These features capture market momentum, trends, and volatility. Additionally, rolling windows and time lags will be used to enhance temporal pattern recognition for ML/DL models. Sentiment scores (if used) will be added as auxiliary features.

- **Model Building** –We will implement and compare multiple models to capture different patterns in the data:
 - Traditional Models: ARIMA, SARIMA, GARCH to model trend and volatility.
 - Machine Learning Models: Random Forest and XGBoost to capture non-linear relationships and feature importance.
 - Deep Learning Models: LSTM and GRU to learn temporal dependencies. Transformer-based architectures may also be explored for long-term trend capturing.
 - All models will be trained on historical data split into training and testing sets. Hyperparameter tuning will be performed using cross-validation techniques.
- **Model Evaluation** –Model performance will be assessed using both regression and risk-related metrics:
 - Root Mean Squared Error (RMSE) – Measures the average magnitude of the prediction errors.
 - Mean Absolute Error (MAE) – Indicates how far predictions deviate from actual values.
 - R-squared Score – Indicates the proportion of variance explained.
 - Sharpe Ratio – (Optional) To assess return vs. risk in portfolio optimization.

- **Visualization & Interpretation** –Post-modeling, we will visualize predicted vs. actual stock prices using time series plots. Feature importance will be highlighted using SHAP or permutation importance. Cumulative returns and profit simulation graphs may be included. For deep learning models, attention or saliency maps might be explored to understand which inputs influenced predictions.
- **Deployment** –We plan to build a lightweight web dashboard using Streamlit or Flask to show real-time predictions, trend visualizations, and model summaries. The frontend will allow users to input a stock symbol and view forecasts with associated charts. The backend will run the trained models and retrieve updated data via APIs like Yahoo Finance or Alpha Vantage.

6.Tools and Technologies

- **Programming Language** – Python
- **Notebook/IDE** – Google Colab, Jupyter Notebook
- **Libraries** – pandas, numpy, matplotlib, seaborn, scikit-learn, statsmodels, keras, TensorFlow, xgboost, yfinance, transformers
- **APIs** – Yahoo Finance, Alpha Vantage, Quandl, Tiingo
- **Optional Tools** – Streamlit or Flask for web-based dashboards

7.Team Members and Roles

Vishwabharathi.S – Data Collection & Preprocessing

Ramkishor.S – Feature Engineering & Model Building

Safeera Nowsheen.M – EDA & Visualization

Snekha valli.K – Documentation & Deployment