Phase-2 Project Submission

Student Name: Gobinath P

Register Number: 620123106027

Institution: AVS Engineering College Salem

Department: Electronics and Communication Engineering

Date of Submission: 10.05.2025

Github Repository Link: [https://github.com/Gobinath14317]

---

1. Problem Statement

Stock markets are highly dynamic and influenced by many factors such as company news, global trends, and investor behavior. Because of this, predicting the future price of a stock is not only challenging but also very important. This project takes a data science approach to solve this problem. By using historical stock price data and applying artificial intelligence methods like time series analysis, we attempt to "crack the market code." The aim is to predict the future closing price of a stock based on past prices and trends. This kind of prediction task falls under regression, as it involves estimating a continuous numerical value. Building an accurate prediction system can support financial decision-making, automated trading, and personal investment planning.

2. Objectives

The goal of this project is to use AI and machine learning models to forecast stock prices in an understandable and practical way. We intend to use historical stock data and extract meaningful patterns to predict the next possible price. Objectives include:

- Applying time series models like ARIMA and deep learning models like LSTM.

- Explaining concepts using simple Python, C, and C++ code snippets for better understanding.
- Evaluating model performance using RMSE and R^2 metrics.
- Making predictions clear and interpretable using charts.
- Enhancing predictions with engineered features such as moving averages and lag data

3.Flowchart of the Project Workflow

Step 1: Data Collection
Step 2: Data Preprocessing
Step 3: Exploratory Data Analysis
Step 4: Feature Engineering
Step 5: Model Building
Step 6: Model Evaluation
Step 7: Prediction and Interpretation

4. Data Description

We collected stock data using the `yfinance` *Python package from Yahoo Finance. The dataset includes date, open, high, low, close, adjusted close prices, and volume. It is structured and time-series in format, meaning it is ordered by date and recorded daily. Since prices change continuously, this is a dynamic dataset. The target variable we aim to predict is the "Close" price.*

5. Data Preprocessing

The raw data was cleaned by:

- Filling missing values using forward fill.
- Removing duplicate rows.
- Converting dates to datetime format.
- Normalizing features using MinMaxScaler.
- Extracting date components like day and month.
- Treating outliers where necessary using boxplots.

These steps ensure that the model can train smoothly on clean and consistent data.

6.Exploratory Data Analysis (EDA)

EDA helped us understand the data structure. We used:

- Line plots to see trends in stock prices.
- Histograms for feature distributions.
- Correlation matrix to find relationships.
- Scatter plots and pair plots to study bivariate patterns.

We observed that stock prices followed trends and sometimes had seasonal spikes, especially during company announcements or quarterly reports.

7. Feature Engineering

We created:

- Moving averages (e.g., MA10, MA50) to show trend direction.
- Lag features to capture the value from previous days.
- Date-related features like day, month, and weekday to track periodic behavior.

These features help the models learn more effectively. In C++, this would involve simple loops to compute averages and shifts

8. Model Building

We built and compared:

- **Linear Regression** for simplicity.
- **ARIMA** for classical time series modeling.
- **LSTM** for deep learning with memory.

Python was used to implement all models. We evaluated them with RMSE and R^2 scores. LSTM showed the best results, as it learns from past sequences better than the others.

9. Visualization of Results & Model Insights

Visual tools used:

- Actual vs. Predicted price graphs
- Residual plots
- Feature importance charts

These helped us interpret model behavior and assess where improvements were needed

10. Tools and Technologies Used

- **Languages:** Python, C, C++
- **IDEs:** Jupyter Notebook, Code::Blocks
- **Libraries:** pandas, numpy, matplotlib, seaborn, scikit-learn, keras
- **Data Source:** Yahoo Finance (via `yfinance` API)
- **Visuals:** Line charts, scatter plots, heatmaps

11. Team Members and Contributions

- **[GOBINATH. P]:**. Pata collection, preprocessing, Python modeling, report writing
- **[ARUN S]:** Logic building in C/C++, feature creation
- **[KEERTHIVERAJ P]:** Visualizations, performance comparison, final editing