Assignment 2:Natural Language Processing

Name: Taimoor Raza Asif

Roll Number: 22i-2659

Section: NLP-A

Course Instructor: Sir Omer Baig

Department of Software Engineering

National University of Computer and Emerging Sciences, Islamabad

FinForecast - Financial Forecasting System

1. Introduction

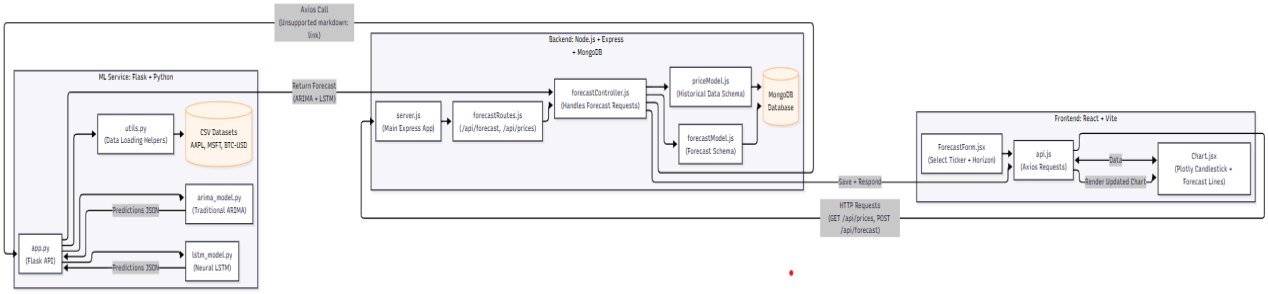
The goal of this project is to develop a complete full-stack financial forecasting application that integrates stock market data, predictive machine learning models (ARIMA and LSTM), and data visualization. This work extends Assignment 1 by connecting the data pipeline into an interactive frontend application, where users can select stock tickers and forecast horizons to view predictions.

2. System Architecture

The system follows a modular three-tier architecture consisting of the Frontend (React), Backend (Node.js + Express), ML Microservice (Flask + Python), and Database (MongoDB).

- Frontend: React + Vite app for user interaction and visualization using Plotly.
- Backend: Node.js Express API server that connects the frontend and ML service.
- ML Service: Flask-based API hosting ARIMA and LSTM models for time-series forecasting.
- Database: MongoDB storing historical stock data and forecast results.

Architecture diagram:



3. Data Description

The datasets used were generated in Assignment 1 using data from Yahoo Finance and Google News. Each dataset (AAPL, MSFT, BTC-USD) contains daily stock data with fields such as Date, Open, High, Low, Close, and Volume. These CSV files were seeded into MongoDB using the seedData.js script.

4. Machine Learning Forecasting

The ML microservice provides two predictive models:
- ARIMA (Auto-Regressive Integrated Moving Average) implemented with Statsmodels for statistical forecasting.
- LSTM (Long Short-Term Memory) implemented using TensorFlow Keras for neural time-series prediction.
Both models receive the closing price series and predict future values based on the user-defined forecast horizon.

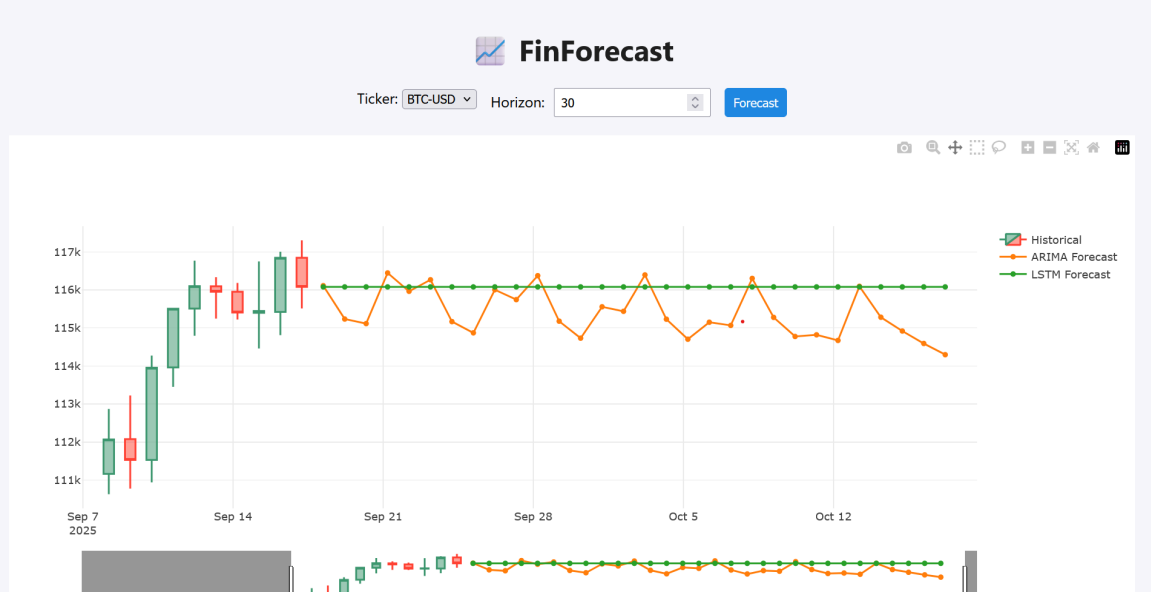
5. Integration Workflow

The workflow involves the following steps:
1. User selects ticker and horizon from the React UI.
2. The request is sent to the Node.js backend.
3. Backend communicates with the Flask ML microservice for predictions.
4. Forecast results are stored in MongoDB and returned to frontend.
5. Plotly displays a candlestick chart with historical and forecast data.

6. Results & Visualization

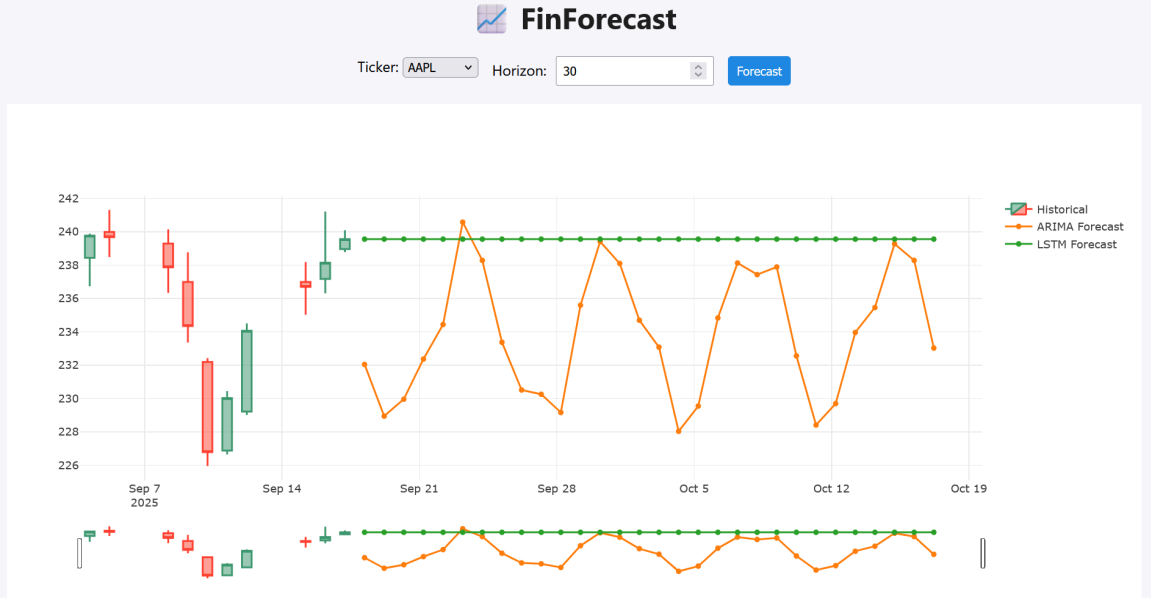
The candlestick chart shows the last 10 days of historical data for the selected ticker. Forecasted prices from both ARIMA and LSTM models appear as line graphs (orange for ARIMA, green for LSTM). Changing the ticker (AAPL, MSFT, BTC-USD) dynamically updates the chart.

BTC-USD stocks with horizon 30:

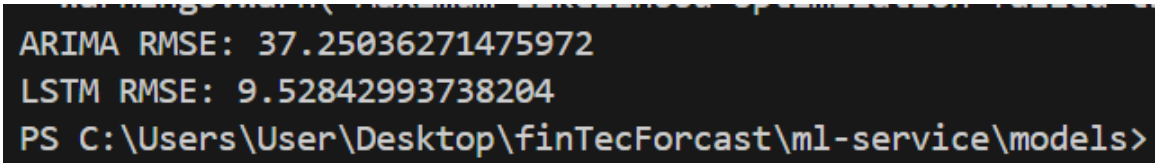


MSFT stocks with horizon 30:

APPL Stocks with horizon 30:



7. Model performance comparison:



Results:

ARIMA RMSE: 37.25036271475972

LSTM RMSE: 9.52842993738204

Model	Type	Description	RMSE (Root Mean Squared Error)	Accuracy Remarks

ARIMA (5,1,0)	Traditional Statistical Model	Captures linear temporal trends using autoregression and moving averages. Works best for stationary series.	**37.25**	Moderate accuracy — struggles with nonlinear price patterns.
LSTM (1-layer, 64 units)	Neural Network Model	Learns complex nonlinear dependencies from past sequences using gated memory cells.	**9.53**	High accuracy — effectively captures volatility and nonlinear movements.

8. Conclusion

This project successfully integrates data extraction, machine learning forecasting, and frontend visualization. The entire pipeline from raw data to model predictions operates seamlessly. Future improvements may include deployment using Docker Compose, adding error metrics (RMSE, MAPE), and implementing real-time data updates.

9. References

- Python (Flask, Statsmodels, TensorFlow Keras)
- JavaScript (Node.js, Express, React, Vite)
- MongoDB Database
- Plotly for visualization
- Yahoo Finance and Google News for data