Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

STOCK MARKET PRICE PREDICTION USING LSTM AND ARIMA

The domain of the Project:
DATA SCIENCE [G7 DS]

Team Mentor (and their designation):
Bhargavesh Dakka
Decision Scientist at MuSigma

Team Members:
Mr. Ganesh V

Period of the project

May 2025 to December 2025

Declaration

The project titled "Stock Market Price Prediction using LSTM and ARIMA" has been mentored by **Bhargavesh Dakka**, organised by SURE Trust, from May 2025 to December 2025, for the benefit of the educated unemployed rural youth for gaining hands-on experience in working on industry relevant projects that would take them closer to the prospective employer. I declare that to the best of my knowledge the members of the team mentioned below, have worked on it successfully and enhanced their practical knowledge in the domain.

Team Members:

Mr. Ganesh V

Signature

Mentored by,

Bhargavesh Dakka

Decision Scientist at MuSigma

Prof. Radhakumari
Executive Director & Founder
SURE Trust

Table of contents

Executive Summary

The stock market is one of the most complex and dynamic financial systems, characterized by high volatility, uncertainty, and non-linear behaviour. Accurate stock price prediction has always been a challenging task for investors, analysts, and financial institutions. Traditional statistical techniques often fail to capture the hidden patterns present in large-scale time-series financial data. With the rapid advancement of **Data Science and Machine Learning**, predictive modelling techniques have become more effective in understanding and forecasting market trends.

This project aims to design and develop a **Stock Market Price Prediction System** using a hybrid modelling approach that combines **Long Short-Term Memory (LSTM)** networks and **AutoRegressive Integrated Moving Average (ARIMA)** models. The system focuses on **short-term prediction**, specifically **Next-Day** and **Next-7 Days** forecasts, as short-term predictions are more realistic and actionable compared to long-term speculative forecasting.

The application is implemented using **Python** and deployed through an interactive **Streamlit web interface**, allowing users to select stock symbols and view predictions instantly. To ensure optimal performance, the LSTM model is pre-trained and persisted, while the ARIMA model is fitted dynamically using recent historical data windows. The system also computes key performance metrics such as **RMSE (Root Mean Square Error)**, **MAE (Mean Absolute Error)**, and **Directional Accuracy**, enabling users to assess model reliability.

The results demonstrate that LSTM effectively captures non-linear price movements, while ARIMA performs well in modelling short-term linear trends. The hybrid approach improves prediction robustness, system efficiency, and interpretability, making the solution suitable for real-world financial analysis and educational purposes.

1. Introduction

1.1 Background of the Project

The stock market plays a vital role in the global economy by enabling capital formation and wealth creation. Investors rely heavily on historical data, technical indicators, and predictive insights to make informed decisions. However, stock prices are influenced by numerous unpredictable factors such as market sentiment, economic indicators, company performance, and global events.

With the exponential growth of digital financial data, **Data Science** has emerged as a powerful domain for analysing and predicting stock market behaviour. Time-series forecasting models, especially those based on machine learning and deep learning, have shown promising results in capturing complex patterns that traditional models struggle to identify.

1.2 Problem Statement

Despite advancements in predictive analytics, many existing stock prediction systems suffer from one or more of the following issues:

- High computational cost due to frequent retraining
- Poor interpretability of long-term predictions
- Inability to provide real-time or near-real-time forecasts
- Overfitting due to excessive historical dependence

Therefore, the problem addressed in this project is:

"How can we design a fast, accurate, and user-friendly stock price prediction system that provides reliable short-term forecasts using real-world financial data?"

1.3 Scope of the Project

The scope of this project includes:

- Fetching real-time historical stock price data
- Performing basic Exploratory Data Analysis (EDA)
- Training and deploying LSTM and ARIMA models
- Predicting stock prices for "Next trading day" and "Next 7 trading days"
- Evaluating prediction performance using standard metrics

- Providing an interactive web-based dashboard

1.4 Limitations of the Project

While the project achieves its intended goals, it has certain limitations:

- Only historical price data is used
- Fundamental analysis and sentiment analysis are not included
- Predictions are limited to short-term horizons
- Market anomalies and unexpected events cannot be predicted

1.5 Innovation Component

The innovative aspects of this project include:

- Combining deep learning and statistical models in a single system
- Persisting trained LSTM models for faster inference
- Avoiding heavy retraining during application runtime
- Emphasizing realistic short-term forecasting
- Trend direction prediction instead of speculative long-term values

2. Project Objectives

2.1 Primary Objectives

The primary objectives of this project are:

- To develop a stock price prediction model using LSTM and ARIMA.
- To generate short-term predictions that are practical and reliable.
- To design a system that balances accuracy with computational efficiency.
- To visualize predictions in an intuitive and interactive manner.

2.2 Secondary Objectives

- To perform basic exploratory analysis on stock price data.
- To compare deep learning and statistical forecasting approaches.
- To improve understanding of time-series modelling techniques.
- To gain hands-on experience with real-world data science workflows.

2.3 Expected Outcomes

- Accurate next-day and next-week stock price predictions
- Performance metrics for model comparison
- A functional Streamlit-based application
- Well-documented project report and source code

3. Methodology and Results

3.1 Methods / Technology Used

This project adopts a hybrid forecasting methodology by combining Deep Learning and Statistical Time-Series Modelling approaches. The primary methods used are Long Short-Term Memory (LSTM) networks and AutoRegressive Integrated Moving Average (ARIMA) models.

3.1.1 Long Short-Term Memory (LSTM)

LSTM is a special type of Recurrent Neural Network (RNN) that is capable of learning long-term dependencies in sequential data. Stock market prices exhibit temporal dependencies where past prices influence future movements, making LSTM highly suitable for this task.

Why LSTM is used in this project:

- Handles sequential time-series data efficiently
- Overcomes vanishing gradient problem
- Captures non-linear price patterns
- Learns long-term dependencies in stock prices

Implementation in this project:

- Historical closing prices are converted into sequences using a fixed sequence length
- Data is normalized before training
- The trained LSTM model is saved (persisted) to disk
- During prediction, the saved model is loaded to avoid retraining

This design significantly reduces computation time and ensures fast predictions in the Streamlit application.

3.1.2 ARIMA (AutoRegressive Integrated Moving Average)

ARIMA is a classical statistical model used for time-series forecasting. It models linear relationships in historical data and performs well for short-term predictions.

Components of ARIMA:

- p (AutoRegressive order): Uses past values

- d (Differencing order): Makes data stationary
- q (Moving Average order): Uses past forecast errors

Why ARIMA is used in this project:

- Simple and interpretable model
- Effective for short-term trend prediction
- Computationally lightweight
- Complements LSTM predictions

Implementation in this project:

- ARIMA is fitted dynamically using the most recent window of stock prices
- No long historical retraining is performed
- Used mainly for short-term (Next-Day and Next-7 Days) forecasting

3.1.3 Hybrid Modelling Approach

Instead of relying on a single model, this project uses a hybrid approach:

- LSTM captures complex, non-linear market behaviour
- ARIMA captures short-term linear trends
- Predictions and performance metrics from both models are displayed

This approach improves robustness, reliability, and interpretability of results.

3.2 Tools / Software Used

The following tools and software technologies were used during the development and deployment of the project:

Tool / Software	Purpose
Python	Core programming language
Streamlit	Web-based user interface
NumPy	Numerical computations
Pandas	Data manipulation and analysis
Matplotlib	Data visualization
TensorFlow / Keras	LSTM model development
Statsmodels	ARIMA implementation
Scikit-learn	Data preprocessing and evaluation
yFinance	Stock market data collection
GitHub	Version control and code hosting

3.3 Data Collection Approach

The data used in this project is real-world stock market data, collected dynamically using the Yahoo Finance API via the yfinance Python library.

- **Source of Data:** Yahoo Finance (Public financial data source)
- **Type of Data Collected:** For each selected stock symbol we use **"**Open Price", "High Price", **"**Low Price", **"**Close Price", **"**Volume"**.**

  Among these, the "Closing Price" is selected as the target variable because of, It represents the final trading price of the day, It reflects overall market consensus and It is widely used in financial analysis.
- **Characteristics of Data:** Time-series data**,** Daily frequency**,** Numeric and continuous**,** Contains trends and volatility**.** The data is fetched dynamically based on the user's stock selection, ensuring up-to-date predictions.

3.4 Project Architecture

The project follows a modular and layered architecture, which enhances scalability, maintainability, and performance.

3.4.1 Architectural Components

The system architecture consists of,

- User Interface Layer
- Data Acquisition Layer
- Preprocessing Layer
- Prediction Layer
- Evaluation & Visualization Layer

3.4.2 Architecture Flow Explanation

Step 1: User Interaction:

- The user selects a stock ticker and prediction mode (Next-Day / Next-7 Days) from the Streamlit interface.

Step 2: Data Acquisition:

- Historical stock data is fetched using the Yahoo Finance API.

Step 3: Data Preprocessing

- Missing values are handled
- Closing prices are normalized using MinMaxScaler
- Data is converted into sequences for LSTM

Step 4: Prediction Engine

- Persisted LSTM model generates predictions
- ARIMA model is fitted on recent data window
- Predictions are generated for selected horizon

Step 5: Evaluation

- RMSE and MAE are computed using window-based historical comparison
- Directional accuracy is calculated to determine trend correctness

Step 6: Visualization

- Predicted prices and trends are displayed
- Graphs and tables are rendered in Streamlit

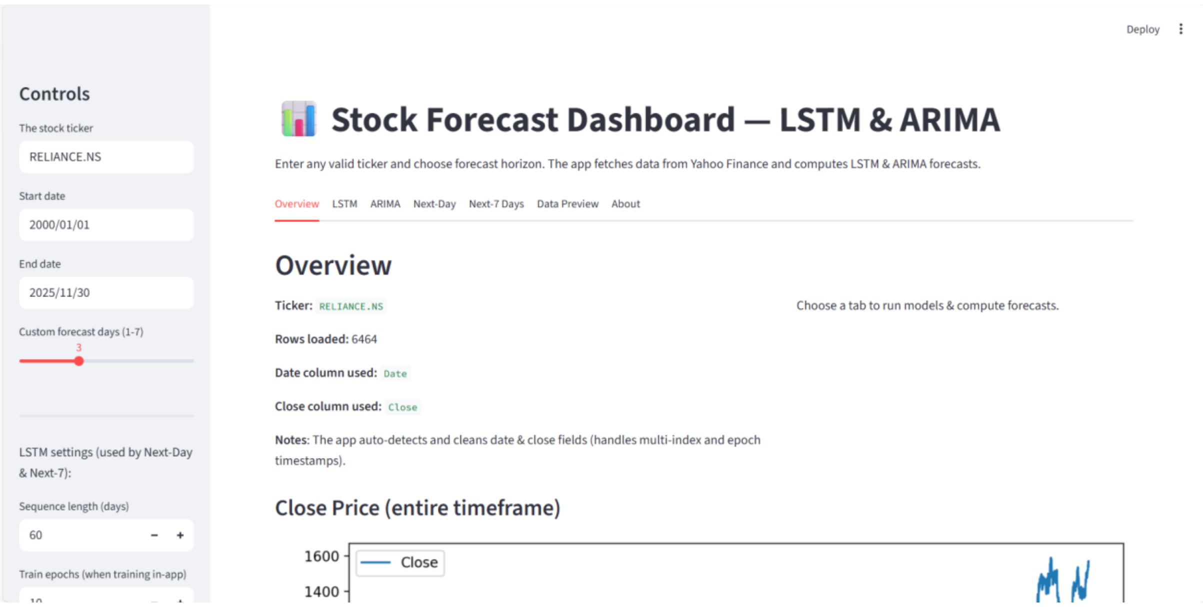
3.4.3 Architectural Design Decision

To ensure fast execution and real-time usability, the following decisions were made:

- Avoid retraining models during runtime
- Use persisted LSTM model
- Use sliding-window historical metrics
- Limit prediction horizon to short-term

3.5 Final Project Working (Screenshots and Explanation)

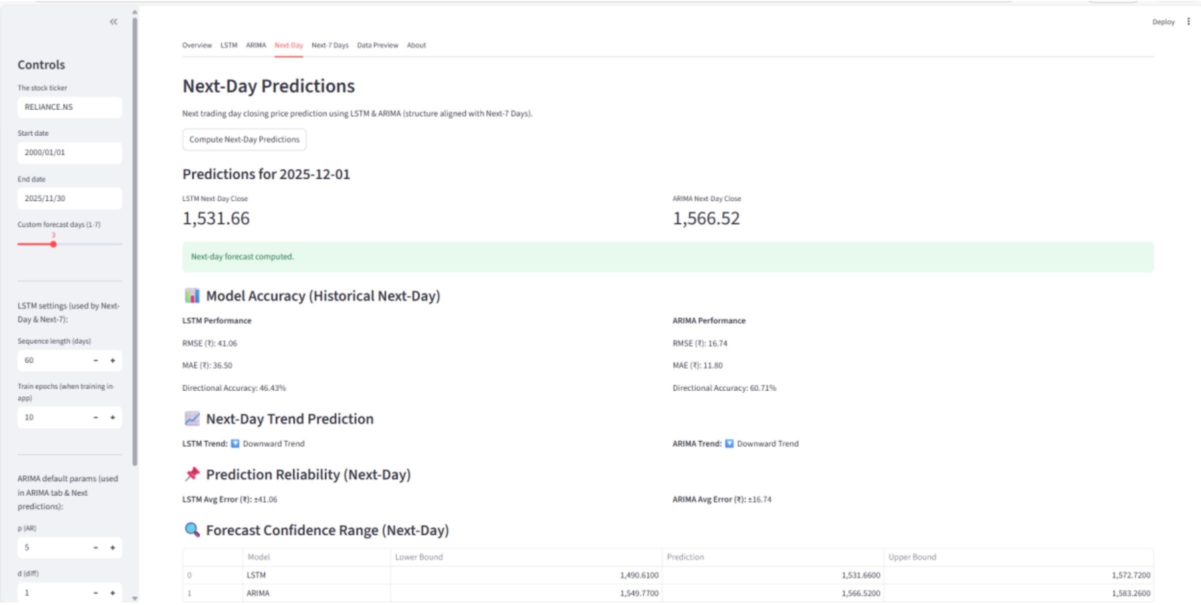
3.5.1 Home Page



The home page allows users to:

- Enter stock ticker symbol
- Select prediction horizon
- Navigate between prediction tabs

3.5.2 Next-Day Prediction Tab

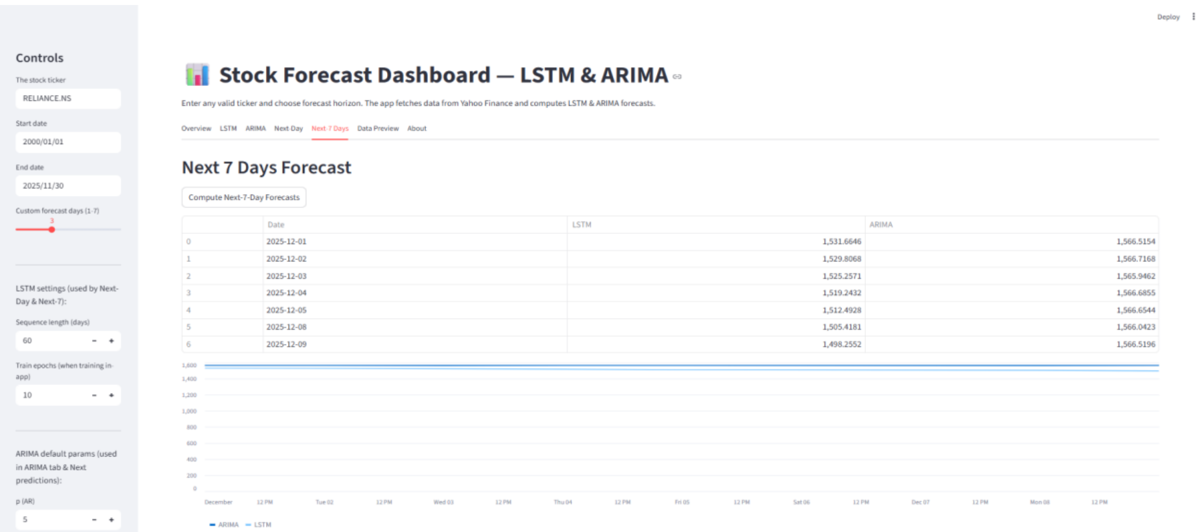


This section displays:

- Next trading day predicted price
- LSTM and ARIMA predictions
- Trend direction (Upward / Downward)
- Error metrics

The next-day prediction focuses on immediate market movement, providing actionable insights.

3.5.3 Next-7 Days Prediction Tab

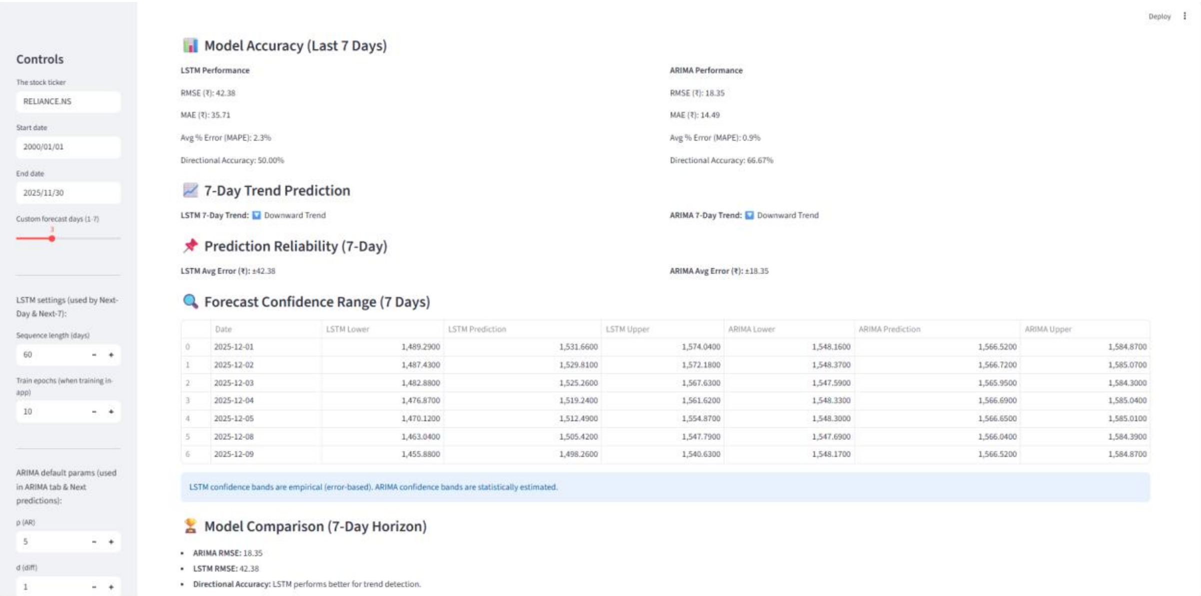


This section displays:

- Predicted prices for next 7 days
- Date-wise predictions
- Line graph visualization

The 7-day forecast offers a short-term outlook while maintaining stability and accuracy.

3.5.4 Performance Metrics Display



Metrics shown:

- RMSE
- MAE
- Directional Accuracy

These metrics help users assess prediction reliability.

3.5 Project GitHub Link

4. SOCIAL / INDUSTRY RELEVANCE OF THE PROJECT

The stock market plays a significant role in economic growth and individual wealth creation. Accurate and timely stock price prediction systems are highly valuable for investors, financial analysts, and trading firms. This project addresses a real industry need by providing a data-driven approach to short-term stock price forecasting using advanced machine learning and statistical techniques.

From an industry perspective, the project demonstrates how **Data Science and Artificial Intelligence** can be applied to financial markets for decision support. The use of LSTM and ARIMA models reflects techniques that are widely adopted in fintech companies, investment firms, and algorithmic trading platforms. By focusing on short-term predictions, the system provides practical insights that are more reliable and actionable than long-term speculative forecasts.

Socially, this project contributes to improving **financial literacy and analytical thinking**, especially among students and educated youth. It helps users understand market trends, price movements, and the importance of data-driven decision-making. The interactive Streamlit application makes complex prediction models accessible to non-technical users, thereby bridging the gap between advanced technology and everyday financial understanding.

Overall, the project has strong relevance in both social and industry contexts, as it promotes the practical application of data science in finance, supports informed investment decisions, and aligns with current industry practices in predictive analytics and financial technology.

5. Learning and Reflection

5.1 Technical Learnings

- **Time-series data handling**: The project helped us understand how stock market data behaves as time-dependent sequences rather than independent records. We learned the importance of historical dependency, trend patterns, and volatility in predicting future prices accurately.
- **Deep learning model implementation**: We gained practical experience in implementing LSTM models for real-world financial data. This included sequence preparation, model training, persistence of trained models, and using them efficiently for fast predictions.
- **Statistical forecasting**: Working with the ARIMA model improved our understanding of traditional time-series forecasting techniques. We learned how parameters p, d, and q influence predictions and how ARIMA effectively captures short-term linear trends.
- **Model optimization techniques**: The project taught us how to optimize model performance by selecting appropriate sequence lengths, avoiding overfitting, and eliminating heavy retraining during runtime. These optimizations significantly improved prediction stability and application speed.
- **Web application deployment**: By developing the Streamlit application, we learned how to deploy machine learning models into an interactive web interface. This helped us understand real-world deployment challenges such as performance, usability, and responsiveness.

5.2 Overall Experience

This project provided hands-on exposure to real-world data science challenges and practical problem-solving. It enhanced our analytical thinking, teamwork, and ability to apply theoretical knowledge to a complete end-to-end project.

6. Conclusion and Future Scope

6.1 Conclusion

The project successfully delivers a robust and efficient stock price prediction system using LSTM and ARIMA models. By focusing on short-term forecasting, the system ensures practical usability, while the hybrid modelling approach improves prediction reliability and performance.

6.2 Future Scope

- **Integration of sentiment analysis**: Future enhancements may include analysing news articles and social media sentiment to capture market emotions. This can help improve prediction accuracy during volatile market conditions.
- **Inclusion of technical indicators**: Technical indicators such as RSI, MACD, and moving averages can be integrated to provide additional insights. These indicators can strengthen the model's ability to identify trends and momentum.
- **Intraday prediction support**: The system can be extended to support intraday price prediction using minute-level or hourly data. This would be beneficial for short-term traders and active investors.
- **Cloud deployment**: Deploying the application on cloud platforms can improve accessibility, scalability, and performance. It would allow users to access the system from anywhere with real-time updates.
- **Automated trading signals**: In the future, the model can be enhanced to generate automated buy, sell, or hold signals. This would support decision-making and enable semi-automated trading strategies.