📘 Project Report : Stock Price Prediction using LSTM

Submitted by: VISHWAJEET KUMAR
Course: Bachelor of Technology, B Tech
Department: Computer Science & Engineering
Institution: Loknayak Jai Prakash Institute of Technology, Chapra
Submission Date: June 2025

Stock Price Prediction using LSTM

Abstract

This report presents a machine learning approach to forecasting stock prices using a Long Short-Term Memory (LSTM) neural network. It covers the end-to-end process from data acquisition to model training and deployment, and introduces a web application interface built with Streamlit for real-time interaction.

1. Introduction

Stock market prediction has long been a subject of interest for investors, economists, and data scientists. Traditional statistical methods often struggle with the non-linearity and volatility of stock prices. This project explores the use of LSTM networks, a type of recurrent neural network (RNN), well-suited for sequence prediction problems. The objective is to create a system that predicts stock prices for the next 7, 15, and 30 days and visualize the results through an interactive application.

2. Objectives

- To collect and preprocess historical stock price data.
- To design and train an LSTM model for time-series forecasting.
- To evaluate model performance by comparing predicted and actual prices.
- To predict future prices for 7, 15, and 30 days.
- To develop an interactive web application using Streamlit.

3. Literature Review

Numerous studies have explored the application of machine learning in financial forecasting. LSTM models are particularly effective for time-series prediction due to their ability to retain information over long sequences. Prior works have demonstrated LSTM's advantage over traditional models like ARIMA and linear regression in capturing stock price patterns.

4. Methodology

4.1 Data Collection

Data is sourced using the yfinance Python library, which fetches historical stock prices from Yahoo Finance. The focus is on the 'Close' price.

4.2 Data Preprocessing

- The 'Close' column is normalized using MinMaxScaler from Scikit-learn.
- Sequences of 60 time steps are created to predict the next price point.

4.3 Model Design

- LSTM Layer 1: 50 units, return sequences=True
- LSTM Layer 2: 50 units
- Dense Layer: 1 unit (output)
- Optimizer: Adam
- Loss Function: Mean Squared Error (MSE)

4.4 Model Training

The model is trained over 10 epochs using the training sequences, with validation on holdout data.

4.5 Model Evaluation

- Model predictions are compared with actual values on historical data.
- Graphs show the closeness of predicted values to the real trends.

4.6 Forecasting Future Prices

- Future predictions are made by using the most recent 60-day window.
- Predictions are made for 7, 15, and 30-day horizons.

5. Streamlit Web Application

The final model is deployed using Streamlit, allowing users to:

- Input any valid stock ticker (e.g., AAPL, TATAMOTORS.NS).
- View the current closing price.
- See past prediction accuracy.
- Explore future forecasts in an interactive dashboard.

6. Results and Analysis

- The LSTM model showed strong alignment with historical data.
- The predicted values for future days followed realistic price trends.
- The web app enables users to explore different stocks and date ranges.

7. Conclusion

This project demonstrates the feasibility and potential of LSTM networks in financial forecasting. The integration with Streamlit makes the model accessible and interactive for end users. While the current model focuses on single-feature prediction (Close price), future work can incorporate multi-feature inputs for enhanced performance.

8. Future Work

- Use of additional indicators like volume, RSI, MACD.
- Experiment with bidirectional LSTM or attention mechanisms.
- Add evaluation metrics like RMSE, MAE.
- Deploy the app on a cloud platform for broader accessibility.

9. References

- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation.
- Brownlee, J. (2017). Deep Learning for Time Series Forecasting.
- TensorFlow Documentation: https://www.tensorflow.org/
- yfinance Library: https://pypi.org/project/yfinance/
- Streamlit Documentation: https://docs.streamlit.io/