DSCI 560 Lab 4 Report

Group Code Submission Link:
https://github.com/shubhamdarekar/DSCI560---Shubham/tree/main/Lab_4

Code README:
https://github.com/shubhamdarekar/DSCI560---Shubham/blob/main/Lab_4/README.md

Meeting Notes Link:
https://docs.google.com/document/d/1TX3OkdmXz503WLHo-s17MIcVcAoM0mrUMVyjMKocDOs/edit?tab=t.0

Team Details: Group 1
Saavani Vaidya 9385579920
Yuxuan Liu 4780355176
Shubham Darekar 1641138809

1. Introduction

This report presents the implementation of an LSTM-based stock price prediction model for algorithmic trading. The goal of this lab was to develop an algorithm that predicts stock price movements and provides buy/sell signals to maximize trading profits. Our approach involves collecting, preprocessing, and analyzing stock data, incorporating technical indicators, and developing an LSTM-based model for forecasting stock prices and combining it with a SMA strategy to decide when to buy, sell, or hold shares in order to maximize profit..

2. Data Collection and Preprocessing

2.1 Data Collection

To build an effective stock price prediction model, we collected historical stock price data for Apple Inc (AAPL), Nvidia Corp (NVDA), Coca-Cola Consolidated Inc (COKE) using the yfinance Python library. The dataset spans 20 years with a daily interval.

2.2 Feature Engineering: Adding Technical Indicators

To improve predictive accuracy, we incorporated several widely used technical indicators:

- Momentum Indicator (MOM): Measures the rate of price change using a 5-day difference.
- Relative Strength Index (RSI): Identifies overbought/oversold conditions using a 14-day window.
- Moving Averages (MA5 & MA20): Smooths price trends over 5-day and 20-day periods.
- Volume Indicators (Volume_MA5 & Volume_MA20): Tracks volume trends using 5-day and 20-day moving averages.

2.3 Data Preparation

After adding technical indicators, we performed the following steps:

- Feature Selection: Selecting relevant features for model input.
- Data Normalization: Using MinMaxScaler to scale all features to [0,1].
- Sequence Generation: Creating 50-day sequences as input data for the LSTM model.

3. Algorithm Development

3.1 Model Selection

For stock price prediction, we chose a Long Short-Term Memory (LSTM) network due to its strength in capturing time-series patterns.

3.2 LSTM Model Architecture

The model consists of:

- Three LSTM layers (100-100-50 units) to extract time-dependent patterns.
- Dropout layers to prevent overfitting.
- Dense layers to predict the stock price.

3.3 Model Compilation and Training

- Optimizer: Adam with an exponential learning rate decay.
- Loss Function: Mean Squared Error.
- Early Stopping: Stops training if validation loss stops improving.

3.4 Model Evaluation

After training, we evaluated the model using:

- Mean Absolute Error (MAE)

- Root Mean Squared Error (RMSE)

3.5 Results: Predicted vs. Actual Prices

To visualize model performance, we plotted the predicted vs. actual stock prices.

4. Mock Trading Environment

To evaluate the effectiveness of our algorithmic trading model, we implemented a mock trading environment that simulates buy and sell decisions based on LSTM-predicted prices and Simple Moving Average (SMA) crossovers.

4.1 Trading Strategy

The trading strategy follows these principles:

We used a hybrid approach between LSTM and moving average (SMA). If the LTSM predicts a price increase and the SMA comparison shows an upward trend, then we buy. If the LTSM predicts a price decrease and the SMA comparison shows a downward trend, then we sell. Otherwise we hold.

1. LSTM Price Prediction for Buy/Sell Decisions:
    - If the predicted next-day price is at least 0.5% higher than the current price, the algorithm buys shares using up to 80% of available cash.
    - If the predicted price is at least 0.5% lower than the current price and the projected drop exceeds 0.1%, the algorithm sells all holdings.
2. Portfolio Management for Multiple Stocks:
    - Each stock in the portfolio is tracked separately.
    - Cash and stock holdings are updated dynamically with each trade.
    - The total portfolio value is calculated daily.
3. Risk Management Considerations:
    - Minimum profit threshold (0.1%) prevents excessive trading.
    - Portfolio diversification allows trading multiple stocks instead of a single asset.

4.2 Portfolio Simulation

- The initial investment is set to $10,000.
- The portfolio is adjusted dynamically based on buy and sell transactions.
- Portfolio value over time is tracked, considering both cash and stock holdings.

4.3 Trade Execution & Performance Evaluation

The simulated trading environment logs each trade, maintains transaction records, and evaluates strategy performance through financial metrics.

Key Performance Metrics:

1. Total Portfolio Value: Measures final wealth accumulation after trading.
2. Total Return (%): Percentage increase from the initial investment.
3. Annualized Return: Average projected return on a yearly basis.
4. Sharpe Ratio: Measures risk-adjusted returns using daily returns.

4.4 Results & Observations

Trading Performance Overview:

```
--- Final Trade Summary ---
Initial Investment: $10,000.00
Final Portfolio Value: $25,091.70
Total Return: 150.92%

Annualized Return: 27.93%
Sharpe Ratio: 0.93
Total Trades Executed: 477
```