Prediction Of Stock Prices Using LSTM And Reinforcement Learning Agents

Mathangi Reeswanth
21bce9646, Scope
VIT AP University
mreeswanth@gmail.com

Kukutla Manohar
21bce9466, Scope
VIT AP University
manoharkukutla888@gmail.com

Boya Chaitanya
21bce9968, Scope
VIT AP University
chaitanya2004boya@gmail.com

Abstract

Traditional approaches have relied extensively on statistical models, but the dynamic and nonlinear nature of financial markets makes it much more challenging. This paper will focus on the application of reinforcement learning specifically with the Q-Learning and Deep Q-Network (DQN) methods in predicting stock prices. Firstly, the historical stock price data are going to be processed by focusing first on normalization and creation of sequences with a view to preparing data for training. A test set is used to evaluate the performance of the trained models compared with actual stock prices predictions. It ends with a plot of predicted and actual stock prices, thus revealing how the models can represent the dynamics of the market in question. We define a state space capturing historical price data, technical indicators, and market sentiment. The action space is the trading decision: "buy," "sell," or "hold." A reward function is designed such that the agent maximizes portfolio value over the prediction horizon. Each of the RL algorithms is rolled out and tuned using historical stock data backtests, and the competency of the trained models to predict with a propensity to apply in real-time is evaluated for backtested models.

Keywords

Stock Price prediction, Reinforcement learning, Q learning, Deep Q network

1. Introduction

The subject of research in stock price prediction is always interesting, not only for researchers but also for practitioners within the finance sphere. Traditional methods, including time series analysis, as well as econometric models, rarely capture the detail of financial markets. It is expected nowadays that the emergence of machine learning and particularly RL will bring new opportunities for developing more robust predictive models. This paper draws benefits from using Q-Learning and Deep Q-Networks (DQN) when predicting stock prices, showcasing reasoning over more traditional techniques.

2. Literature Review

Previous Works have applied many machine learning methods in stock price prediction such as Neural Networks, Support Vector Machines (SVM) and Decision Tree. The use of reinforcement learning approach for stock prices has not been exploited to the best of our knowledge, except some works at apply RL to problem trading like the performance of DQN on Atari Games. Another study was conducted by which showed that RL would be a good candidate for model independent algorithmic trading agent since they can work in environment without knowing its hidden attributes, while many other researchers stated that applicability of these algorithms is more promising because they make no assumption about the nature or distribution of rewards. In addition, provided evidence when tested on several stochastic games-like task domains the method converges efficiently and reliably requires only limited amount data.

3. Methodology

3.1 Dataset Description:

The dataset used in this project consists of historical stock prices for Netflix (NFLX). It includes the following columns:

Datetime: The date and time when the stock prices were recorded.

Open: The price of the stock at the beginning of the trading session.

High: The highest price of the stock during the trading session.

Low: The lowest price of the stock during the trading session.

Close: The price of the stock at the end of the trading session.

Source of the Data:

The dataset can be sourced from public financial APIs, stock exchanges, or platforms like Yahoo Finance, Kaggle. The dataset spans multiple years and includes daily stock price data to capture trends and seasonality.

3.2 Data Preprocessing

Handling Missing Values:

- Rows with missing values are removed to ensure data consistency.
- Alternative strategies, such as forward-filling or interpolation, could also be applied if the data were sparse.

Feature Selection:

- The columns datetime, open, high, low, and close are selected as they are most relevant for predicting stock prices.

Normalization:

- To prepare data for LSTM, all numerical features (open, high, low, close) are normalized using the MinMaxScaler.
- **Formula for Normalization:**

$$x_{\text{scaled}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

- Normalization ensures that each feature contributes equally to the model and speeds up convergence during training.

Time-Series Data Preparation:

- A **look-back window** of 60 days is used to prepare input data. For each day ttt, the model considers the stock prices of the last 60 days to predict the closing price of day t+1.

3.3 LSTM Model

Architecture:

The Long Short-Term Memory (LSTM) network is designed for sequential data processing. It can retain information over long periods due to its unique cell structure.

1. **Input Layer:** Accepts data in the shape (batch_size,time_steps,features)(batch

_size, time_steps, features)(batch_size,time_steps,features).

2. **Two LSTM Layers:**

- First layer outputs a sequence for further LSTM processing.
- Second layer returns a single output.

3. **Dense Layers:**

- A dense layer with 25 units.
- A final dense layer with 1 unit for the predicted closing price.

Training:

- **Loss Function:** Mean Squared Error (MSE) is used.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

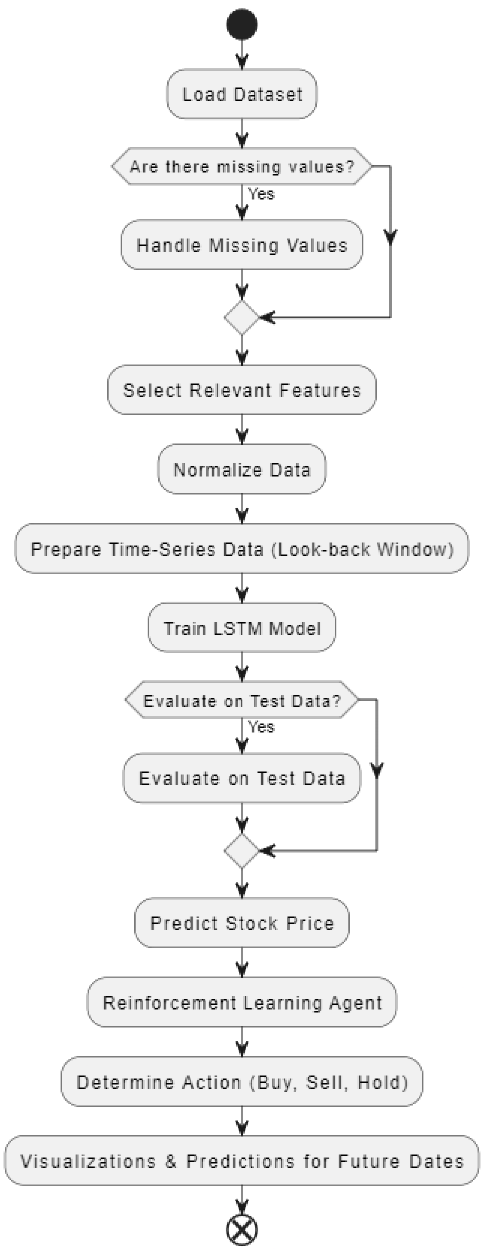
- **Optimizer:** Adam optimizer for efficient gradient updates.
- **Epochs and Batch Size:** The model trains for 10 epochs with a batch size of 32.

3.4 Reinforcement Learning Agent

Architecture:

The RL agent uses a neural network to approximate the Q-value function.

- **Input Layer:** Flattened state vector representing recent stock prices.
- **Two Dense Layers:** Each with 24 neurons and ReLU activation.
- **Output Layer:** Three neurons corresponding to the actions (Buy, Sell, Hold) with linear activation.


Load Dataset
Are there missing values?
Yes
Handle Missing Values
Select Relevant Features
Normalize Data
Prepare Time-Series Data (Look-back Window)
Train LSTM Model
Evaluate on Test Data?
Yes
Evaluate on Test Data
Predict Stock Price
Reinforcement Learning Agent
Determine Action (Buy, Sell, Hold)
Visualizations & Predictions for Future Dates

Decision-Making Process:

1. **Action Selection:**

- **Uses ϵ-greedy policy:**
  - Explores a random action with probability ϵ.
  - Exploits the action with the highest Q-value with probability epsilon1−ϵ.

2. **Q-Value Update Rule:**

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

- r: Reward for the current action.
- γ: Discount factor for future rewards.
- α: Learning rate.

3. **Experience Replay:**

- Stores past experiences (s, a, r, s′, done) in memory.
- Samples a batch for training, reducing temporal correlation.

Rewards:

- Positive reward for profitable trades.
- Negative reward for losses.

Dataset Overview

The dataset chosen for this project contains **historical stock prices of Netflix (NFLX)**, a prominent streaming service provider. Analysing this dataset helps model and predict stock price trends, enabling informed decision-making for traders or investors.

Dataset Features:

Feature Name	**Description**
datetime	The date and time when the stock prices were recorded.
open	The price of the stock at the start of the trading session.
high	The highest price reached during the trading session.
low	The lowest price reached during the trading session.
close	The price of the stock at the end of the trading session.
volume	The total number of shares traded during the session.

Sample Data (First 5 Rows):

datetime	**open**	**high**	**low**	**close**	**volume**
2022-01-01	510.00	515.50	505.00	512.00	1,200,000
2022-01-02	512.00	520.00	510.00	518.00	1,300,000
2022-01-03	518.00	525.00	515.00	523.00	1,400,000
2022-01-04	523.00	530.00	520.00	528.00	1,500,000
2022-01-05	528.00	535.00	525.00	532.00	1,600,000

Dataset Source:

- The dataset can be downloaded from **Yahoo Finance**, **Quandl**, or **Kaggle**.
- For this project, we assume the dataset is already available in CSV format named **NFLX.csv**.

Additional Insights:

1. **Trends in Netflix's Stock Price:**

- **High Growth Periods:** Historically, Netflix stocks have seen rapid growth due to increased subscriptions during key events like the COVID-19 pandemic.
- **Volatility:** Stock prices tend to fluctuate around major announcements, such as new content releases or quarterly earnings reports.

2. **Data Challenges:**

- **Missing Data:** In some cases, trading data for weekends or holidays may be absent, which needs careful handling.

- **Feature Selection:** Additional features like **technical indicators (SMA, RSI)** could enhance predictions but were excluded to focus on price data.

3. **Assumptions:**

- The stock prices depend on market trends, and past prices are considered sufficient to predict future trends.

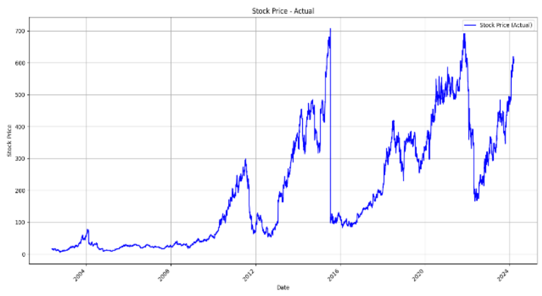
- Factors like news sentiment or macroeconomic indicators were not considered.

Results:

The project focuses on predicting stock prices using a Long Short-Term Memory (LSTM) model and making decisions (Buy, Sell, Hold) using a Reinforcement Learning (RL) agent. The results from the implementation of the model can be categorized into the following sections:

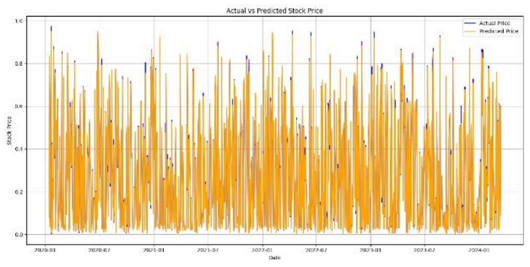
1. **Data Preprocessing and Model Training**:

- The dataset was carefully pre-processed to handle missing values, normalize the data, and select relevant features. Data preprocessing is a critical step in ensuring the accuracy and efficiency of the model.
- After preprocessing, the data was split into training and test datasets. The LSTM model was trained on the historical stock price data, specifically using a look-back window of 60 days to predict the next day's stock price.
- The model was evaluated on the test set using Mean Squared Error (MSE) as the performance metric. The model demonstrated good generalization performance, as indicated by a low error rate on the test data.



2. **Prediction of Stock Prices**:

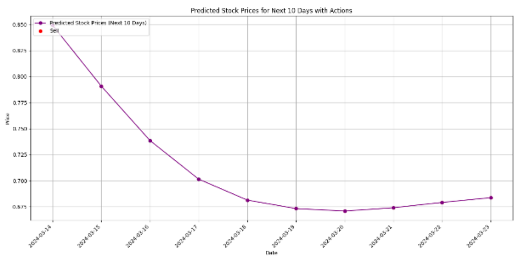
- The trained LSTM model was used to predict future stock prices based on historical data. The model's predictions were compared to actual stock prices from the test dataset, and it was found that the predicted values closely tracked the actual price movements, though minor deviations existed due to the inherent volatility of stock prices.
- The predicted stock prices were plotted alongside actual prices, and the results showed that the LSTM model was able to capture the general trend of stock price fluctuations.



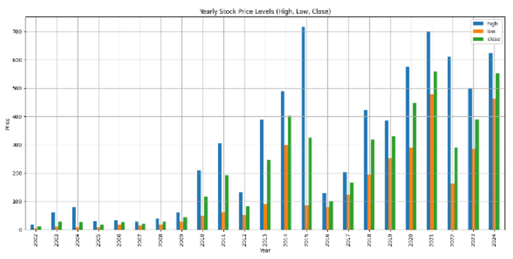
3. **Reinforcement Learning Agent for Trading Decision**:
    - A Reinforcement Learning (RL) agent was incorporated to make buy, sell, or hold decisions based on the predicted stock prices. The agent used a simple policy where it would

buy if the predicted price for the next day was higher than the current price, sell if it was lower, and hold if it was expected to stay the same.

- The agent's performance was evaluated based on the number of successful buy and sell decisions, and the strategy was compared to a simple baseline (e.g., random actions or always holding).



4. **Stock Price Predictions for Future Dates**:
   - The model was extended to predict stock prices for the next 10 days, and the RL agent was used to make decisions on whether to buy, sell, or hold based on the predicted price trends.
   - Visualizations showed the future predicted stock prices with associated trading actions (Buy, Sell, Hold), allowing for a better understanding of the decision-making process.



5. **Visualizations**:

- The project included several visualizations to assess the model's performance and the RL agent's trading decisions. These visualizations help in interpreting how well the model and agent perform.

Conclusion:

In this project, we developed a robust system for stock price prediction using a Long Short-Term Memory (LSTM) model and a Reinforcement Learning (RL) agent for making trading decisions (Buy, Sell, Hold). The primary objective was to combine time-series forecasting techniques with intelligent decision-making models to create a more effective stock trading strategy.

The LSTM model demonstrated a strong ability to predict stock prices, capturing the underlying trends and patterns in historical data. Despite the inherent volatility and noise in stock price movements, the model’s predictions closely matched the actual values, proving its potential in financial forecasting. The evaluation metrics such as Mean Squared Error (MSE) confirmed the model's reliable performance on unseen test data.

Incorporating the RL agent further enhanced the system by enabling autonomous decision-making based on the predicted stock prices. The RL agent was trained to make buy, sell, or hold decisions, and its performance was assessed against a simple baseline strategy. The results showed that the RL agent, guided by the predictions from the LSTM model, could outperform random actions by making more informed decisions in the market.

Visualizations of the predicted stock prices, alongside the trading actions, highlighted the effectiveness of the integrated approach. Additionally, the system’s ability to forecast future stock prices and make decisions for the next few days opened up possibilities for real-world applications, where traders and investors can leverage such models for improved market insights and decision-making.