ECEGR – 5760

PROJECT REPORT

ON

STOCK PRICE PREDICTION

Under the guidance of

Professor

Dr. Amin Riazi

By

Vivek Reddy Karra

INTRODUCTION

The goal of this project is to build a Recurrent Neural Network (RNN) model to predict stock market prices based on historical data, enabling the forecasting of future market trends. RNN models can be able to capture patterns in sequential data over time.

For this study, we focus on the stock data of Starbucks Corporation (SBUX), sourced from NASDAQ [1]. By training the model on historical stock prices, we aim to predict future stock price movements, which can be valuable for investment analysis and decision-making.

THEORETICAL BACKGROUND

Recurrent Neural Networks (RNN) is a type of neural network that is designed to process sequences of data. An RNN consists of three main layers:

- **Input Layer:** Receives the sequence data.
- **Hidden Layer(s):** Processes data sequentially, maintaining a 'memory' of previous steps using loops within the network.
- **Output Layer:** Produces the result based on the processed input sequence.

Types of Recurrent Neural Networks:

Based on the number of inputs and outputs in the network there are four types:

1. One to One
   This type of network takes only single input and produces only single output.
   Ex: Image classification

2. One to Many
   This type of network takes only single input and produces multiple outputs.
   Ex: Music or text generation (predicting a sequence of next words from a single input prompt)

3. Many to One
   This type of network takes multiple inputs and produces only single output.
   Ex: Sentiment Analysis (predicting the overall sentiment from a sequence of words)

4. Many to Many
   This type of network takes multiple inputs and produces multiple outputs.
   Ex: Language translation (translating a sentence from one language to another).

METHODOLOGY

Imported all the necessary Python libraries for data handling, preprocessing, model training and evaluation. Collected the last six months of stock data from 11th September 2025 to 10th March 2025 for Starbucks Corporation(SBUX) from NASDAQ. The dataset includes the following features: **Date, Close/Last, Volume, Open, High, Low.** This dataset is then loaded as dataframe and checked if there are any missing values.

	Date	Close/Last	Volume	Open	High	Low
0	03/10/2025	$101.125	15823320	$105.73	$106.00	$99.80
1	03/07/2025	$106.48	11637380	$104.73	$106.855	$103.4381
2	03/06/2025	$105.47	16321750	$110.01	$110.4266	$104.9054
3	03/05/2025	$111.69	11926820	$111.57	$112.5194	$110.21
4	03/04/2025	$112.06	12482360	$114.99	$115.27	$112.00

Fig 1: Original Dataset

Preprocessed the dataset by converting the 'Close/Last' column from string format to numeric by removing the '$' symbol and commas and converting the 'Date' column to datetime. Then, converted 'Date' column into index and considering only 'Close/Last' as the only column which has closing stock price of the day.

Date	Close/Last
2025-03-10	101.125
2025-03-07	106.480
2025-03-06	105.470
2025-03-05	111.690
2025-03-04	112.060

Fig 2: Dataset after Preprocessing

A line plot was then generated to visualize the trend of stock prices over time.

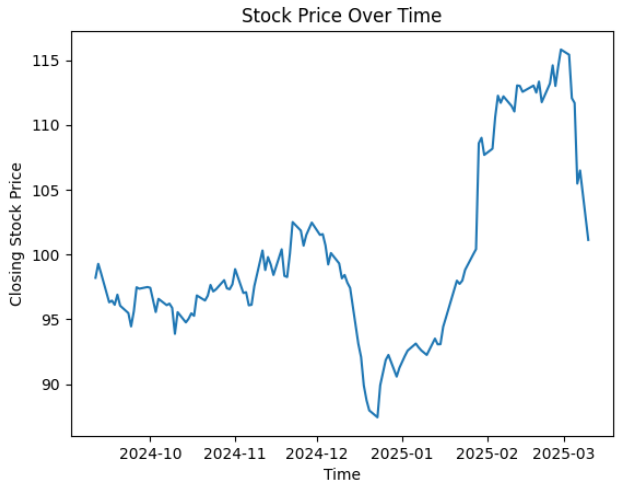

Fig 3: Closing Stock price over Time

To ensure efficient training of the neural network, the data was normalized using MinMaxScaler from the Scikit-learn library. The values were scaled to a range between -1 and 1, which helps to stabilize and speed up the learning process during model training.

Created sequential time series data function that is suitable for training an RNN-based model. Defined a time step of 30 to capture stock prices of previous 30 days and defined forecast length as 3 to predict the stock prices for 3 upcoming days from March 11, 2025 to March 13, 2025.

Splitted the dataset into training and testing sets with the ratio 80:20 for evaluation. Reshaped the data in the form of [samples, timestep, features] to match the input format required for RNN models.

Three different RNN models were build. Each model consists of stacked SimpleRNN layers with ReLU activation functions, and Dropout layers to prevent overfitting. The final layer is a Dense layer that outputs three predicted prices. All models are compiled using Mean Squared Error (MSE) as the loss function and Adam as the optimizer. Each model was trained using training dataset. Training was carried out for 100 epochs with a batch size of 32. The models were trained silently (verbose=0) to focus on performance metrics.

After training, the models were evaluated on the test data using two error metrics: **Mean Squared Error (MSE)** and **Root Mean Squared Error (RMSE)**. Predicted values were inverse-transformed using the scaler to return to actual stock price levels.

The trained models were then used to predict the next three stock prices based on the most recent sequence of data. To enhance the interpretability of predictions, future dates corresponding to the forecasted stock prices were generated using Python's datetime and timedelta libraries. The final output displayed the predicted prices along with their respective future dates.

EVALUATION METRICS

Mean Squared Error (MSE) measures the average squared difference between the actual and predicted values, while Root Mean Squared Error (RMSE) takes the square root of MSE to provide a more interpretable measure.

	MSE	RMSE
Model 1	**2.30**	**1.51**
Model 2	**2.61**	**1.61**
Model 3	**2.20**	**1.48**

Table 1: Evaluated metrics for each model

Comparing with all three models the Model 3 has performed well.

RESULTS AND DISCUSSION

Three different model architectures were tested, each with increasing complexity in terms of the number of layers and units.

MODEL 1

Date	Predicted Price
2025-03-11	**$98.41**
2025-03-12	**$98.40**
2025-03-13	**$98.03**

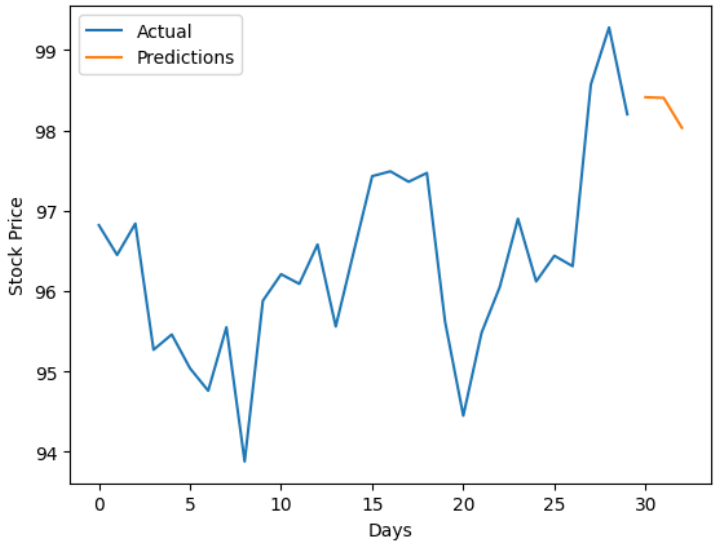

Fig 4: Model 1 Predicted Prices with Actual Prices

MODEL 2

Date	Predicted Price
2025-03-11	**$98.48**
2025-03-12	**$98.66**
2025-03-13	**$98.69**

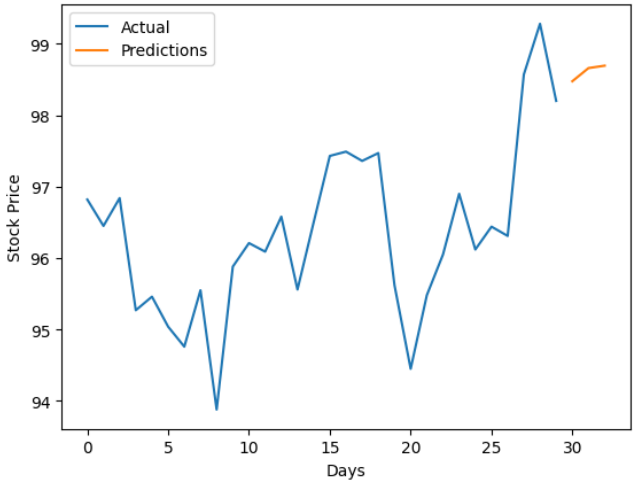

Fig 5: Model 2 Predicted Prices with Actual Prices

MODEL 3

Date	Predicted Price
2025-03-11	**$97.29**
2025-03-12	**$98.03**
2025-03-13	**$97.71**

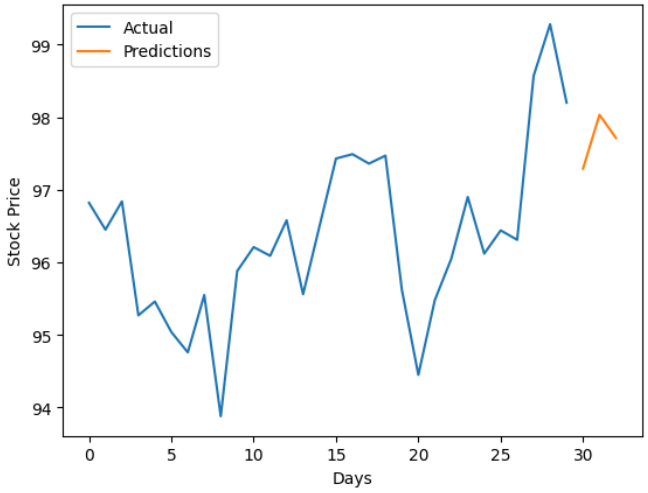

Fig 6: Model 3 Predicted Prices with Actual Prices

CONCLUSION

This project shows that the Recurrent Neural Networks (RNN) models can be able to forecast time-series data such as stock price. By using past sequences of data, the models were able to predict future stock prices with reasonable accuracy.

Key takeaways from the project:

- RNN are effective in learning temporal dependencies in stock market data.
- Model performance improves with deeper architecture (as seen with Model 3).

REFERENCES:

[1] **Dataset** **https://www.nasdaq.com/market-activity/stocks/sbux/historical?page=1&rows_per_page=10&timeline=m6**

[2] **Introduction to Recurrent Neural Networks**
https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/