NYSE Stock Market Prediction using Recurrent Neural Network

Yash Mahendrakumar Wani
Computer Engineering
The University of Texas at Dallas
yxw230008@utdallas.edu

Prathamesh Sanjay Gadad
Computer Engineering
The University of Texas at Dallas
psg220003@utdallas.edu

Prathamesh Kulkarni
Computer Science
The University of Texas at Dallas
pdk220001@utdallas.edu

***Abstract*—This code demonstrates stock price prediction using a simple recurrent neural network (RNN) implemented from scratch. It involves loading a dataset of Apple (AAPL) stock prices, preprocessing the data, and splitting it into training and testing sets. The model is trained using the training data and then used to predict the testing data. We tried the RNN model with multiple epochs and learning rates. The performance is measured using mean absolute error (MAE), and a plot compares predicted stock prices with actual prices. This approach showcases the potential of RNNs for time series forecasting in finance.**

***Index Terms*—Stock Market Prediction, AAPL stock, Forecasting, Features, Price Movements.**

I. Introduction

The New York Stock Exchange, also known as NYSE, plays a crucial role in the global financial market. One of the most closely watched stocks on this exchange is Apple Inc., (AAPL). Given Apple's significant impact on the broader market and the economy, accurately predicting the prices of AAPL stock is vital for investors and traders.

Multiple factors influence the performance of Apple's stock, including its innovation strategies, economic indicators, and market trends. To understand this complex web of dynamics, experts are increasingly turning to machine learning and statistical models. These models analyze vast amounts of data and make predictions about the movements of AAPL stock.

In this paper, we evaluate the effectiveness of different predictive models in forecasting AAPL stock prices in the short and long term. By looking at historical data, trading volumes, and market indicators, we assess various methods such as linear regression, decision trees, and neural networks. We also investigate how feature selection and data preprocessing impact the accuracy of these predictions [1].

Our findings not only provide insights into the factors that shape the trends of AAPL's stock but also offer a comparative analysis of different models. [1] This analysis can guide investment decisions in the stock market, helping investors make informed choices.

Overall, the use of machine learning and statistical models has become increasingly important in accurately predicting AAPL stock prices. These tools allow us to navigate the complexities of the market and make informed investment decisions.

II. Problem Definition

The problem is the prediction of the stock price of Apple Inc. (AAPL) based on past data on the New York Stock Exchange (NYSE). This project performs stock price prediction based on the current data provided for the New York Stock Exchange (NYSE). Data consists of past open, close, high, and low prices of stock and the model undertakes the task of predicting the closing price of the stock. So, the concern is how to predict the future prices with the most accurate way possible based on previous trends and patterns with lesser prediction error. Metrics like Mean Absolute Error (MAE) are used to evaluate the performance of the model by comparing the predicted and actual stock prices. By pre-processing the data, modeling by the use of sequence data, and optimization of the prediction, the approach is towards more accurate predictions.

III. Techniques/Algorithm Used

Following are the key elements required for stock prediction and used in the code:

A. Data Gathering

The proposed system requires data from a stock exchange, specifically focusing pn daily-format datasets for individual stocks. The analysis is centered around data gathered from the AAPL (Apple Inc.) stock on the New York Stock Exchange (NYSE).

The NYSE is one of the most prominent and established stock exchanges global, offering a vast selection of stocks for trading and investment. It provides extensive data, including daily price variations, volume traded, and other financial information essential for making informed investment decisions. For each trading dat, the data includes:

- **Date:** The data when the stock was traded.
- **Symbol:** The stock symbol (in this case, APPL for Apple Inc.)
- **Open:** The price at which the stock opened on a given day.
- **Close:** The price at which the stock closed on a given day.
- **Low:** The lowest price the stock reached during the trading day.
- **High:** The highest price the stock reached during the trading day.

- **Volume:** The total number of shares traded during the day.

Such datasets provide valuable insights into the stock's daily performance and allow investors to track changes in its price and volume over time. By focusing on a well-established stock like AAPL, traders can gain a better understanding of market trends and make more informed decisions. Additionally, monitoring a stable and consistently traded stock can provide insights into broader market movements and patterns.

This dataset is an essential resource for any stock market analysis, providing a comprehensive view of AAPl's trading performance over the specified period. Analyzing the data can reveal trends in stock behavior, volatility, and trading volume offering a foundation for future investment strategies and decisions.

B. Data Preprocessing

In the provided code, the data preprocessing steps are essential for preparing the dataset for training and testing the neural network model. Below is Fig. 1 showing data spread for AAPL stock. The main points of data preprocessing include:

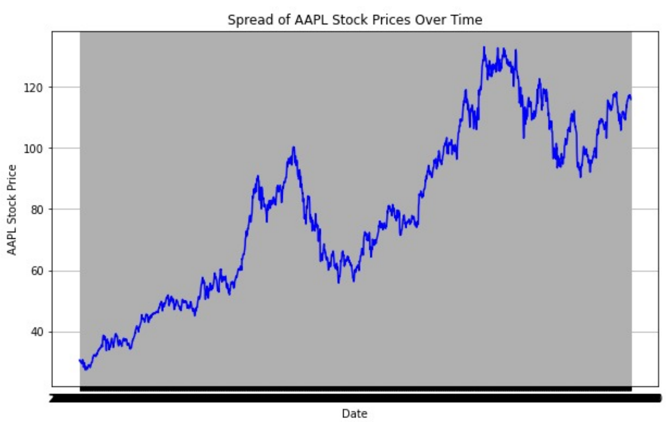

Fig. 1. Data spread

- **Data Loading:** The code loads data from a CSV file, selecting a specific stock symbol (AAPL) and dropping unnecessary columns (symbol and volume).
- **Data Splitting:** The data is split into a training set and a testing set based on a specific index, with the training set begin the first 1000 samples and the testing set being the rest.
- **Data Scaling:** MinMaxScaler is used to scale the training and testing data, transforming the features to a range between 0 and 1. This is important for the neural network model to work effectively.
- **Sequence Length Definition:** A sequence length is defined (20 in this case) to create sequences of data that the RNN can use for training.
- **Data Preparation for Training and Testing:** Based on the sequence length, training and testing data are prepared in the form of sequences (trainX and testX) and corresponding targets (trainY and testY) for each sequence.

These steps ensure that the data is appropriately structured and normalized for use in the neural network model, facilitating effective training and testing.

C. Stock Market Prediction

Stock price prediction using recurrent neural networks (RNN) offers several advantages for investors and analysts looking to forecast future movements in the stock market. RNNs are particularly well-suited for this task due to their ability to handle sequential data, making them adept at capturing temporal dependencies in stock prices over time. [1] This ability is crucial because stock prices are influenced by a wide range of factors, including market trends, economic indicators, and company-specific events, all of which may evolve over different timeframes. [2]

One of the key strengths of RNNs in stock price prediction is their capability to learn from historical data. By analyzing patterns in past prices, volume, and other relevant indicators, RNNs can identify trends and relationships that may influence future stock prices. This learning process allows the model to make predictions based on complex interactions within the data, potentially leading to more accurate forecasts.

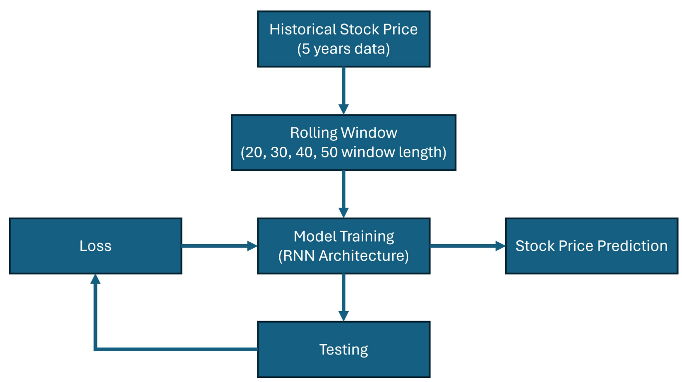

Fig. 2. Stock Market Prediction Architecture

RNNs also excel in handling varying sequence lengths, making them adaptable to different data formats and frequencies, such as daily, weekly, or monthly stock prices. This flexibility is beneficial for modeling different types of financial data and responding to changes in market conditions. Furthermore, the architecture of RNNs, including variants such as LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit), can effectively address issues like vanishing gradients and long-range dependencies, enhancing their predictive capabilities.

However, stock price prediction using RNNs is not without challenges. The stock market is inherently volatile and influenced by numerous unpredictable factors, making it difficult to achieve consistently accurate predictions. Additionally, RNNs may require substantial data preprocessing and fine-tuning of hyperparameters to perform optimally.

Despite these challenges, stock price prediction using RNNs remains a promising area of research and application, with the potential to provide valuable insights for traders and investors. By leveraging the capabilities of RNNs, market participants can make more informed decisions, identify emerging trends, and potentially gain a competitive edge in the dynamic world of finance.

D. Recurrent Neural Network

A Recurrent Neural Network (RNN) is a neural network architecture designed to handle sequential data, making it particularly useful for applications such as time series forecasting, natural language processing, speech recognition, and other tasks involving data with a temporal aspect. [4] RNNs stand out from other neural network types due to their ability to maintain a hidden state, or memory, which stores information about previous inputs in a sequence.

In an RNN, data is processed in a step-by-step fashion. At each time point, the network receives an input and combines it with the hidden state from the prior time point, generating an output. [5] This process enables the RNN to identify and represent patterns and dependencies in the data over time. The hidden state is updated at each step, allowing the network to retain past information and use it to guide future predictions.

The structure of the RNN includes input, hidden, and output layers. The input layer takes in the current input, while the hidden layer processes this input alongside the previous hidden state. The output layer generates the prediction or classification outcome based on the current hidden state. The model's weights and biases are adjusted during training to reduce the loss function, enabling the network to learn from the data effectively. [6]

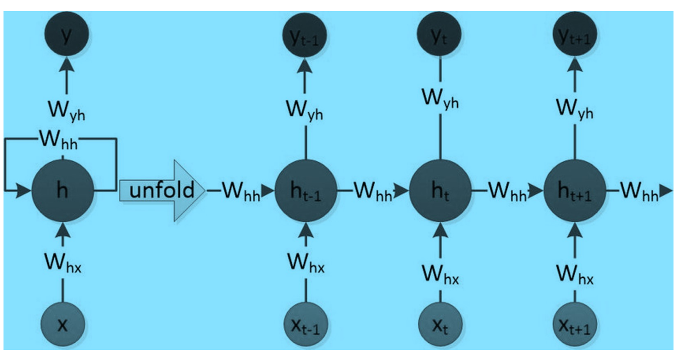

Fig. 3. Recurrent Neural Network Architecture

RNNs are known for their ability to handle variable sequence lengths and maintain context across different time steps. However, they can encounter challenges such as vanishing and exploding gradients, which can hinder training for long sequences. To address these issues, variants of RNNs such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been developed, offering improved performance and stability.

In summary, RNNs are powerful neural network architectures for processing sequential data. Their ability to learn from temporal patterns and dependencies makes them essential tools for various applications, including time series forecasting, language modeling, and speech recognition.

E. Model

- The model is a Recurrent Neural Network (RNN) with specified hyperparameters:
  - **Input Size**: The number of features (5 in this case: date, open, high, low, close).
  - **Hidden Size**: The number of hidden units in each RNN layer consists of 50 hidden units. These hidden units facilities the models ability to retain and utilize temporal information across different time stamps.
  - **Output Size**: The output size of the network (1, because we're predicting the closing price).

F. Loss Function and Optimizer

The Mean Absolute Error (MAE) loss function (`nn.MAELoss`) is used to measure prediction error.

G. Training

- The model is trained for multiple epochs (10, 20, 30, 40 ,50), with a learning rate of 0.005 and 0.01.
- In each epoch, the model predicts the outputs from the input sequences (trainX) and calculates the loss against the actual targets (trainY).
- The optimizer adjusts the model's parameters based on the gradients computed from the loss.

H. Testing

- The model's performance is evaluated on the test set, following a similar process to the training data preparation.
- Predictions are made for the test data sequences (testX), and the Mean Absolute Error (MAE) is calculated between the predicted data and actual target values (testY).

I. Visualization

- The predicted stock prices and actual stock prices are plotted on a graph to visually compare the model's performance.

By combining these techniques and algorithms, the code creates and trains an RNN model to predict future stock prices, evaluates its performance on test data, and visualizes the predictions against the actual prices.

Results and Discussion

The evaluation of different learning rates and epochs in the machine learning model reveals significant insights into the model's performance on a time series forecasting task. The goal was to identify the learning rate and number of epochs that yielded the lowest Mean Absolute Error (MAE) and, consequently, the best model performance.

Across the different epochs tested, the learning rate of 0.005 consistently resulted in lower MAE compared to the learning rate of 0.01. This trend suggests that a lower learning rate of 0.005 facilitates better model convergence and generalization, possibly by allowing the model to adjust its parameters more

Epoch	Learning Rate	MAE
10	0.005	0.17161
	0.01	0.20834
20	0.005	0.05741
	0.01	0.07783
30	0.005	0.0652
	0.01	0.06841
40	0.005	0.06947
	0.01	0.06264
50	0.005	0.06641
	0.01	0.05975

cautiously, reducing the risk of overshooting the optimal solution.

In terms of individual epochs, Epoch 20 with a learning rate of 0.005 demonstrated the best performance, achieving an MAE of 0.05741 (shown in Fig. 4). This was the lowest MAE observed in the experiments, indicating that this configuration of parameters allowed the model to predict unseen data with high accuracy. The graph for this setup visually supports this finding, showcasing a close alignment between the predicted and actual values.

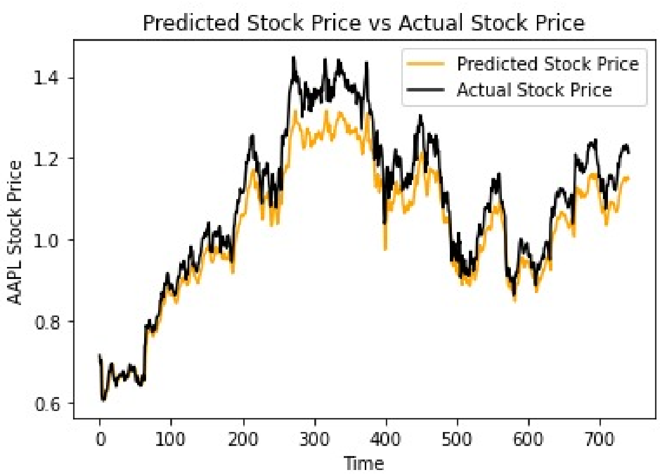

Fig. 4. Best Result achieved for Epoch: 20, Learning Rate: 0.005

On the other hand, the worst performance was observed in Epoch 10 with a learning rate of 0.01. The MAE was significantly higher at 0.20834 (shown in Fig. 5), indicating that the model struggled to converge effectively with a higher learning rate. This was also reflected in the graph, which showed noticeable deviations between the predicted and actual values, suggesting overfitting or instability in the learning process.

Epoch 50 with a learning rate of 0.005 presented a moderately good performance, with an MAE of 0.06641 (shown in Fig. 6). While not as low as Epoch 20, this configuration still demonstrated the advantage of a lower learning rate, maintaining stable performance over a longer training period.

In conclusion, the results suggest that the optimal learning rate for this model and dataset is 0.005, as it consistently led to lower MAE across different epochs. This finding aligns with the expectation that a lower learning rate often results in more

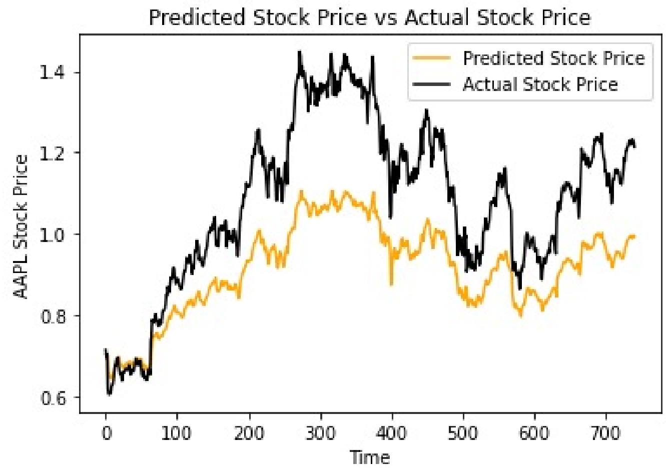

Fig. 5. Worst Result achieved for Epoch: 10, Learning Rate: 0.01

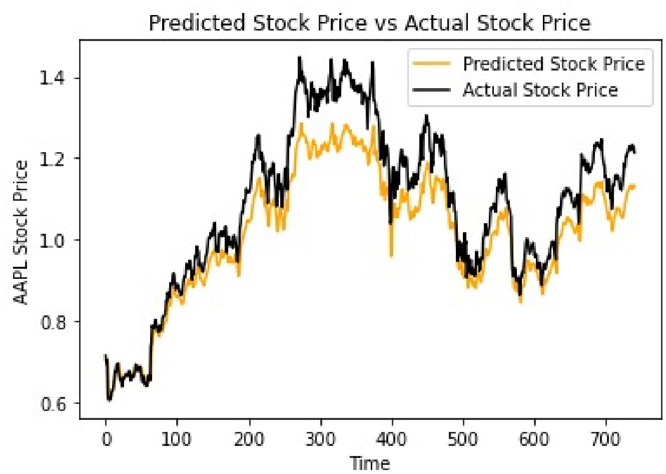

Fig. 6. Moderate Result achieved for Epoch: 50, Learning Rate: 0.005

stable and effective learning. However, further research and experimentation with different model architectures, datasets, and optimization algorithms are recommended to refine the understanding of the optimal hyperparameters for this and similar tasks.

Conclusion

To Conclude, as this paper delves into the optimized and smart approach to predict the AAPL stock prices on the NYSE, using the array of machine learning and statistical models. Through detailed evaluation of various techniques such as linear regression, decision trees, and recurrent neural networks (RNNs), we have highlighted the valuable insights of influencing the AAPL's performance in both short and long terms. By using historical data, trading volumes, along with other indicators, our analysis deepens the pivotal role of feature selection and data preprocessing in enhancing prediction accuracy. our study highlights the increasing importance of machine learning in helping investors and traders navigate the intricate landscape of the stock market. By providing a reliable framework for predicting AAPL stock prices, these advanced techniques offer valuable guidance for decision-making. As we refine our methods and explore new avenues, our research not only enhances our comprehension of stock market behavior but also lays the groundwork for even more powerful predictive models in the future.

REFERENCES

[1] https://link.springer.com/book/10.1007/978-981-99-5166-6.

[2] https://ijisrt.com/assets/upload/files/IJISRT23MAY575.pdf.

[3] P. A. Gunturu, R. Joseph, E. S. Revant and S. Khapre, ”Survey of Stock Market Price Prediction Trends using Machine Learning Techniques,” 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), Bangalore, India, 2023, pp. 1-5, doi: 10.1109/ICAIA57370.2023.10169745. keywords: Surveys;Measurement;Linear regression;Predictive models;Market research;User experience;Reliability;Stock Market;Machine Learning;Stock Price Predictions;Neural Networks.

[4] https://www.proquest.com/docview/2900927264.

[5] https://www.gresham.ac.uk/watch-now/ai-business.

[6] Kianiharchegani, Elham, ”Data-Driven Exploration of Coarse-Grained Equations: Harnessing Machine Learning” (2023). Electronic Thesis and Dissertation Repository. 9530. https://ir.lib.uwo.ca/etd/9530