Stock Market Prediction using LSTM

Divyam Prajapati (DBP230000)   Aaryan Singh Chauhan (AXC230019)   Bhanu Maneesh Reddy Mannem (BXM220055)

Snehal Kumar Ketala (SXK220463)

Abstract— **This project uses Recurrent Neural Networks (RNNs) in predicting stock market trends. RNNs have gained significant attention due to their ability to process sequential data by retaining memory, making them well-suited for analyzing time-series financial data. The study employs various architectures and training methodologies of RNNs, exploring their performance in forecasting stock prices, volatility, and trends. The experiments conducted aim to elucidate the strengths and limitations of RNNs in capturing the complexities of stock market dynamics and provide insights into enhancing their predictive capabilities. The findings contribute to the ongoing discourse on leveraging deep learning techniques for stock market forecasting and offer valuable implications for both investors and researchers in the financial domain.**

1. Introduction

Several studies have looked into how machine learning may be used in quantitative finance to predict asset values, comprehensively manage portfolios, and streamline different investing procedures. In this sense, the term "machine learning" describes a class of algorithms that use computer-based analysis to find patterns only in data, without the need for explicit programming instructions. It's quite difficult to predict anything, particularly when the future is unknown. Accurate prediction-making becomes more difficult in the stock market due to its intrinsic volatility and unpredictability. Stock price prediction can be divided into three categories: medium-term, long-term, and short-term forecasting. Predicting stock values for the next few seconds, minutes, days, weeks, or months is known as short-term forecasting.

While long-term forecasting goes beyond two years, medium-term forecasting covers one or two years. Short-term stock value prediction is significantly easier than long-term projections because of how different world events impact stock market movements[1]. Predictions can be made using three different forms of analysis: fundamental analysis, technical analysis and time series analysis. A technique for studying investments called fundamental analysis examines a firm's sales, earnings, profitability, and other economic aspects in order to determine the share value of the company. This method works especially well for long-term forecasting. Investors are constantly searching for methods to profit from market swings and seize opportunities. Decisions are nevertheless difficult even with the multitude of variables that affect stock prices, including supply and demand, market trends, the state of the world economy, company performance, historical data, public opinion, and sensitive financial information. Many people engage in the stock market because they want to profit from it; nevertheless, news about a company, whether good or bad, can influence purchasers and cause stock values to fluctuate. Making decisions is still a difficult undertaking, even after carefully considering many factors. There are constant obstacles in the way of forecasting future prices and enhancing stock market success.

The current study focuses on predicting the stock prices for the following day to validate the model daily during the year and then compare the forecasts with the actual daily values. The financial market includes exchanges between buyers and sellers of different financial assets, such as gold, silver, jewels, and other essential commodities.

Machine learning and deep learning algorithms come into play given the current bond situation, helping buyers and sellers estimate the possible consequences of financial market transactions. An additional layer of complexity is introduced by the stock market's sensitivity to political and economic circumstances. It is accepted that it might be difficult to get and trust both financial and political-economic data. Time series analysis is a tool that is used in many forecasting methodologies to help identify patterns, trends, and cycles in data. In the context of this study, a time series is a sequential collection of observations for a particular variable, such as stock prices. This project is primarily concerned with time series analysis and technical analysis. Even with its extensive use, stock market forecasting is still a rather mysterious and empirical field, with successful methods frequently kept under wraps. The goal of this project is to close the gap in the field of stock market prediction between theoretical understanding and real-world implementation.

Improving comprehension of the workings of the market is a primary goal since knowledgeable investors are better able to handle possible financial difficulties. The project aims to perform a quantitative assessment of new strategies and a comprehensive scientific analysis of current tactics. Economists, decision-makers, academics, and market participants have all been more interested in market forecasting in the past few decades. The proposed project intends to research and create stock price prediction-focused supervised learning systems. Even with its extensive use, stock market forecasting is still a rather mysterious and empirical field, with successful methods frequently kept under wraps. The goal of this project is to close the gap in the field of stock market prediction between theoretical understanding and real-world implementation.

These models offer a method for combining weaker information sources to create a special tool with practical applications. The recent combination of statistical methods and learning models has improved several machine learning algorithms, such as gradient-boosted regression trees, support vector machines, critical neural networks, and random forecasting.

These algorithms excel at displaying intricate connections and patterns that are hard to discern with linear algorithms due to their nonlinearity. In comparison to linear regression techniques, they also show higher efficacy and resilience to multicollinearity. Right now, a lot of research is being done on the application of ML methods in the finance sector. Tree-based models are used in some studies to predict portfolio returns, while deep learning is used in others to project future values of financial assets[2]. A distinct decision-making model created for day-trading stock market investments is employed by researchers in different field to forecast stock returns. These authors' model for portfolio selection incorporates both the mean-variance (MV) approach and the support vector machine (SVM) method.

An additional academic study explores the use of deep learning models for intelligent indexing. Furthermore, a thorough investigation has looked into several machine learning trends and uses in the field of quantitative finance. Most models, such as those for sentiment analysis, language processing, fraud detection, and decision-making, don't rely on archives of past data or long-term memory. A class of machine learning algorithms based on RNN has demonstrated remarkable efficacy in financial market price prediction and forecasting within this framework. The results showed that LSTM produced significantly better forecasts than autoregressive integrated moving average(ARIMA) when it came to the accuracy of time series data.

Two main questions are addressed in the study: 1) How can one forecast stock prices for the following day based solely on recent data? 2) What techniques can be used to verify the developed model's outcomes? Based on historical data, a RNN with LSTM as an automatic learning technique is used in this study to analyze and forecast future stock values. Our project's main goal is to find the most accurately trained machine learning algorithm for projecting future values in our portfolio by using it to predict the adjusted closing prices of an asset portfolio using an LSTM RNN-based model[3].

2. Data Preprocessing:

A normalization procedure was put in place to solve this issue and ensure fair and unbiased assessment. Normalizing the datasets helps the neural network weigh each feature precisely and eliminates bias and improper prioritization by scaling the values within a specified range. Completing the normalization step is necessary to improve the overall robustness and fairness of the machine learning model. Standardizing the scales of different features helps the neural network become more adept at finding patterns and relationships within the data, which contributes to the creation of a more accurate and reliable predictive model.

Following the normalization procedure, the datasets were further divided into training and testing sets. This part is essential to evaluating and validating the machine learning model's performance. The training set gives the model the ability to identify and adapt to the underlying patterns in the data. The testing set serves as an independent dataset that the model hasn't used for training. It is an essential reference point for assessing the model's ability to generalize and make accurate predictions on new, untested data[4]. These steps aid in mitigating biases, enhancing the model's adaptability, and fortifying its capacity to generate predictions over a range of datasets and real-world scenarios.

we used the S&P 500 Dataset from Kaggle. The dataset encompasses key financial indicators, such as opening and closing prices, high and low prices, trading volumes, and adjusted closing prices. These features are essential for conducting technical analysis, predicting stock trends, and assessing the overall market performance.

3. Methodology and Data

One kind of RNN that can gather data from earlier phases and use it to predict future events is called Long Short-Term Memory (LSTM). The three primary layers of an Artificial Neural Network (ANN) are usually the input layer, the hidden layers, and the output layer.

In a neural network with one hidden layer, the number of dimensions in the data determines the number of nodes in the input layer. There are associations between the nodes in the input layer and the hidden layer called "synapses." Each link, which depicts the relationship between any two nodes (from the input to the hidden layer), has a weight coefficient that serves as the signal processing determinant.

These weights are constantly being adjusted as part of the learning process. After learning is finished, the Artificial Neural Network (ANN) achieves the best weights for every synapse, which improves its capacity to make judgments based on the information it has learned[5]. The network can efficiently identify patterns and correlations in the data, which also guarantees the network's a good generalization to new, unknown inputs. A key component of the neural network's capacity to learn and enhance its prediction skills over time is this dynamic adaptation process.

The input layer's total weights are applied by the nodes in the hidden layer using either a sigmoid or tangent hyperbolic (tanh) function. A mathematical transformation known as the activation function is necessary for value production. The SoftMax function helps to achieve the goal of minimizing the error rate between the training and testing datasets.

The output layer of our neural network is made up of the outcomes of this transition. These numbers might not, however, indicate the best result. In these cases, the network's predictions are iteratively improved by starting a backpropagation process. By creating links between the hidden layer and the output layer, the backpropagation process sends signals that direct weight adjustments to produce the ideal error throughout the predetermined number of epochs. This process is repeated to improve forecasts and reduce total prediction error.

The model then goes through training when the task is done. RNN are a kind of neural network that are intended to forecast future values by using sequences of observations from the past. This kind of neural network makes use of knowledge from earlier phases to learn about the input and predict future patterns. To predict and predict future values, it is critical to hold onto data from previous phases. Here, the hidden layer serves as a history data store that is derived from sequential data; it is comparable to a cache of historical observations. "Recurrent" refers to the method of using components from previous sequences to forecast data in the future. By using its knowledge of past conditions, this recurrent process helps the neural network identify patterns and trends, allowing for a more perceptive and context-aware forecasting method.

LSTM networks were developed in response to the limitations of traditional RNNs in long-term memory storage. Adding a "memory line" to LSTMs has been quite helpful, especially when dealing with long-term temporal datasets[7]. Information from previous stages is stored in an LSTM using specialized gates that are precisely positioned along the memory line. This novel architectural design solves the problems that come with maintaining long-term memory in conventional RNNs by allowing the LSTM to efficiently learn and remember important patterns throughout lengthy sequences.

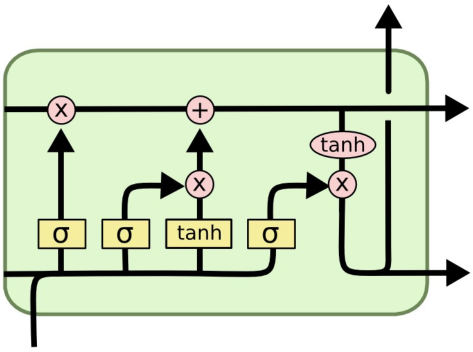

Figure 1: LSTM Cell Structure

The hidden layer's architecture is where RNN and LSTM architectures contrast most. The hidden layer of an LSTM from Figure 1 of the LSTM cell structure is a gated cell made up of four linked layers that work together to generate the state and output of the cell. These outputs are then sent to the following concealed layer. LSTMs have three logistic sigmoid gates and one tanh layer, as opposed to RNNs, which have one tanh neural net layer. To control the information flow within the cell, gates are added to Long Short-Term Memory (LSTMs)[8]. These gates are critical in identifying which data points must be considered for the next cell and which can be ignored.

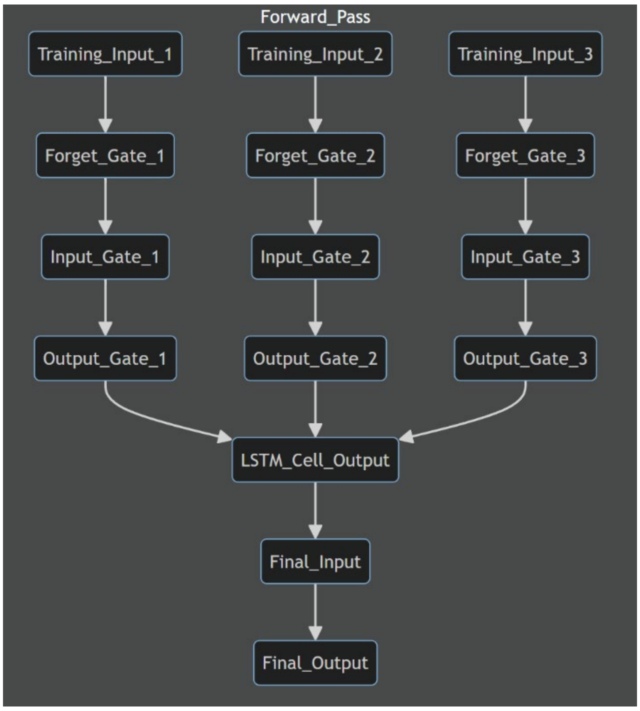

Figure 2.1: Diagrammatic Representation of Forward Pass of the Algorithm

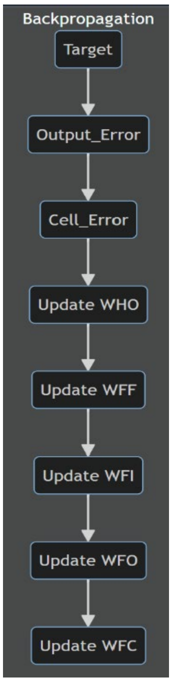

Figure 3.2: Diagrammatic Representation of Backpropagation of the Algorithm

A collection of cells devoted to maintaining historical data streams make up each LSTM node. Every cell has a top line that serves as a transport line, making it easier for data to go from earlier stages to the current ones. These cells' independence plays a critical role in enabling the model to add or filter values from one cell to another or to selectively filter data. The gate's sigmoidal neural network layer then decides whether to permit or forbid data flow, leading the cell to its optimal state.

Every layer of the sigmoid operates using a binary value: 0 means "allow practically nothing to transmit through," while 1 means "allow anything slides across."[9] because of their complicated design, LSTMs are especially good at managing and learning from complex temporal patterns in data since they can handle and comprehend sequential information with ease.

From Figure 2 the goal of the LSTM architecture is to use specific gate control to regulate each cell's state: The forget gate produces a number between 0 and 1 as its output. One indicates to "keep this in full," while zero indicates to "throw this away completely. "Which newly acquired information will be saved in the cell is decided by the Memory Gate. The variables that need to be changed are first chosen by a sigmoid layer called the "input door layer". A vector of possible new values that could be absorbed into the state of the cell is then produced by a tanh layer[10].

The output of every cell is mostly determined by the output gate. The cell state and the most current, filtered data additions are combined to produce this output result.

4. Results and Analysis

We used S&P 500 dataset for our problem from which we decided to predict the stock values of American Airlines Group Inc. and NVIDIA Corp. After training our model, the results showed that the number of epochs had a significant impact on the testing.

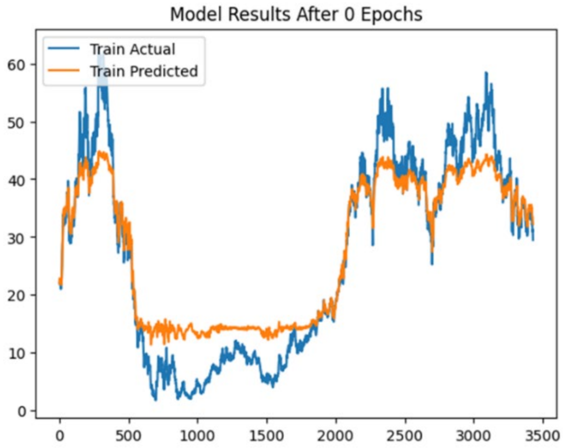

Figure 4.1: AAL - Result after 0 Epochs

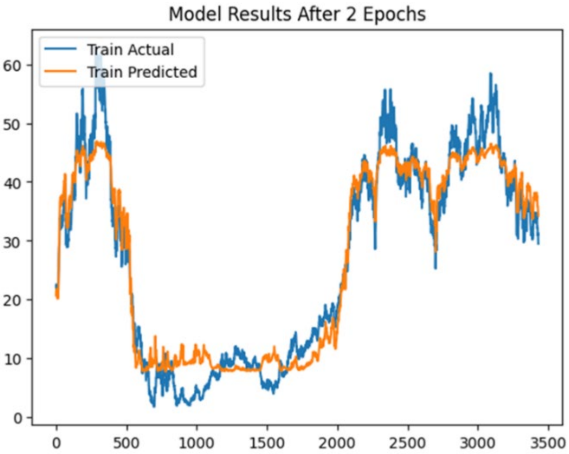

Figure 4.2: AAL - Result after 2 Epochs

As we can see in Figure 4.1 and 4.2, the results improved as we increased the number of epochs, after this point the model started to overfit the data, so we had to terminate it after two epochs.

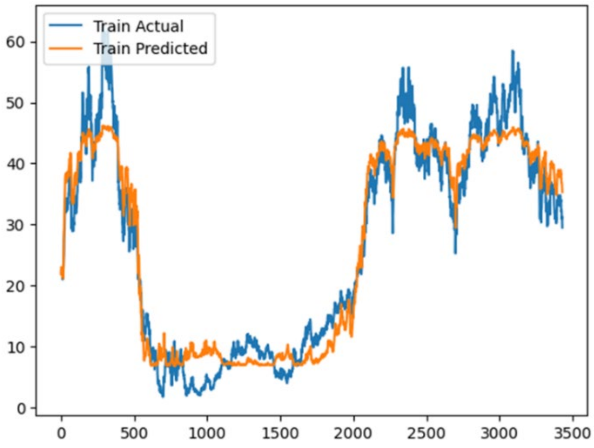

Figure 4.3: AAL - Performance of the model on the training dataset

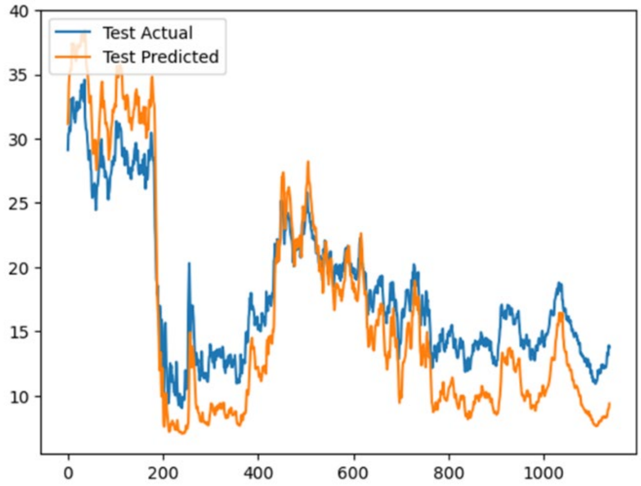

Figure 4.4: AAL - Performance of the model on the test dataset

Figure 4.4 shows how the model performed on the test dataset.

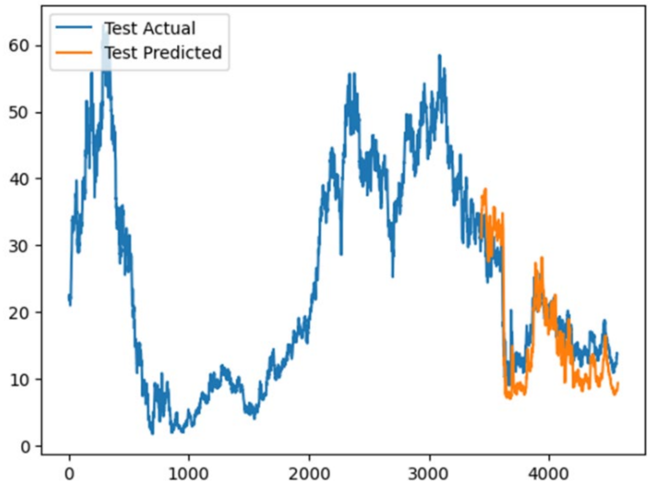

Figure 4.5: AAL - Performance of the model on the actual dataset

Figure 4.5 depicts the model performed well on the testing and training datasets. We can predict values with high accuracy even if we increase the number of data points for our model.

Performance Metrics for AAL	
Train	R2: 0.9433951051350675 RMSE: 4.0947637975419
Test	RMSE: 3.5761323131066947 R2: 0.6241687806169406

Table 4.1: AAL – Performance Metrics

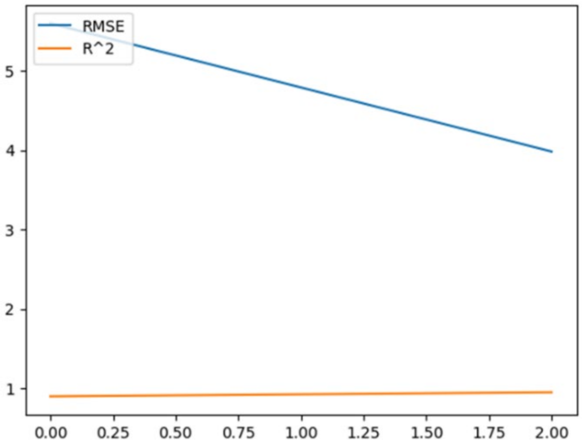

Figure 4.6: AAL – RMSE vs R2 score

Table 4.1 and Figure 4.6 depict that for the AAL dataset the model is improving in accuracy as it has lower RMSE, and it can capture more variance as it undergoes more training. Here is how the model performs with the NVDA dataset.

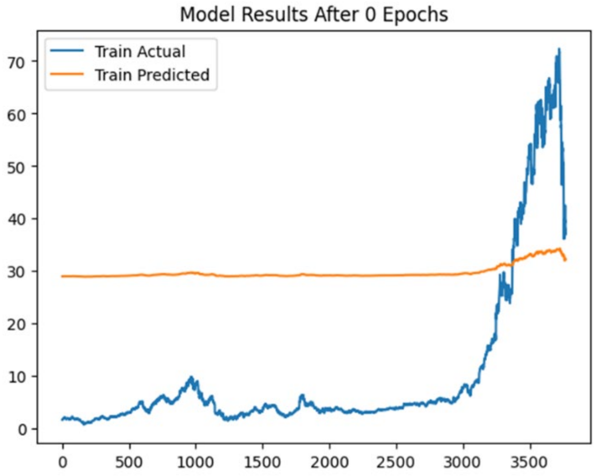

Figure 4.7: NVDA - Result after 0 Epochs

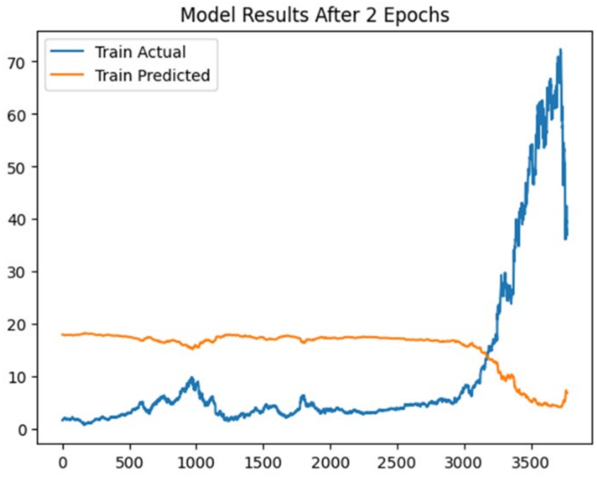

Figure 4.8: NVDA - Result after 2 Epochs

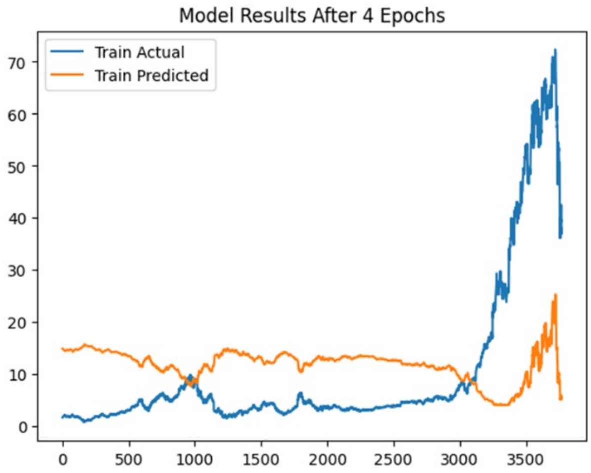

Figure 4.9: NVDA - Result after 4 Epochs

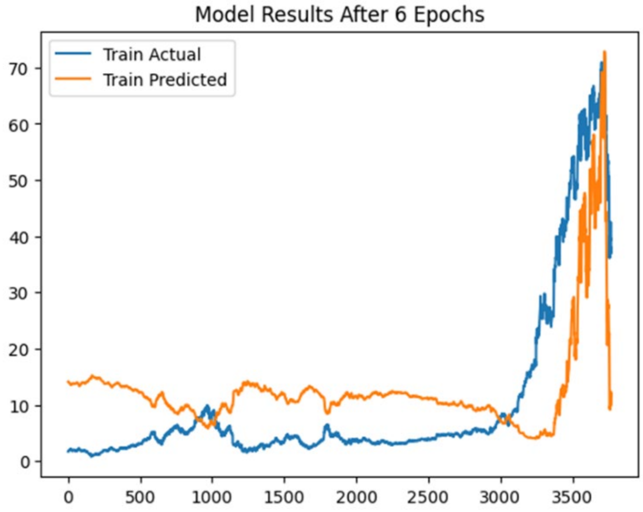

Figure 4.10: NVDA - Result after 6 Epochs

From Figure 4.7-4.10 we can see that initially the model was performing badly on the dataset but as we increased the number of epochs the model started to show signs of improvement.

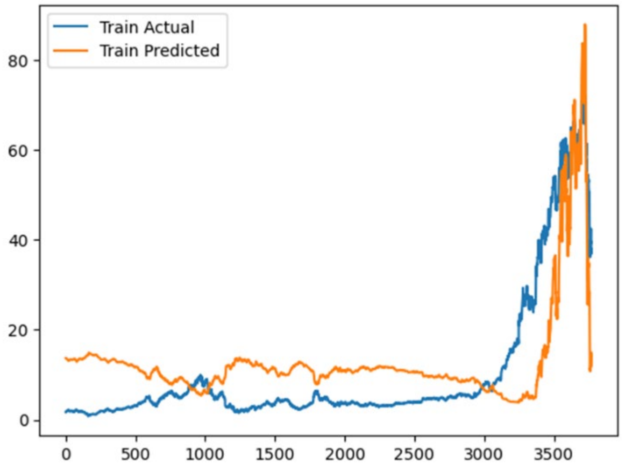

Figure 4.11: NVDA - Performance of the model on the training dataset

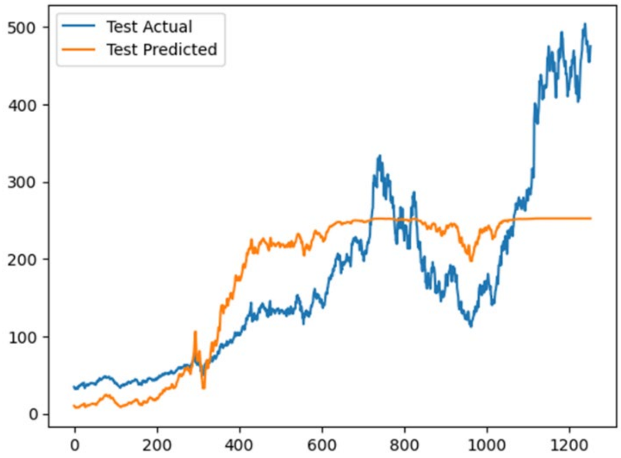

Figure 4.12: NVDA - Performance of the model on the test dataset

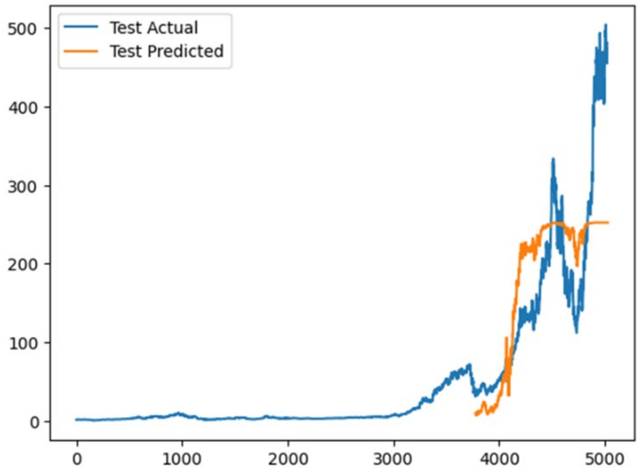

Figure 4.13: NVDA - Performance of the model on the actual dataset

Figure 4.12 and 4.13 show that the model performed well on the testing dataset, but it was not able to capture the intricacies of the test dataset. It performed relatively worse as compared to the prediction on the AAL dataset.

Performance Metrics for NVDA	
Train	RMSE: 10.022254672411615 R2: 0.5943731962511134
Test	RMSE: 82.16760452168778 R2: 0.5388880900119639

Table 4.2: NVDA - Performance metrics

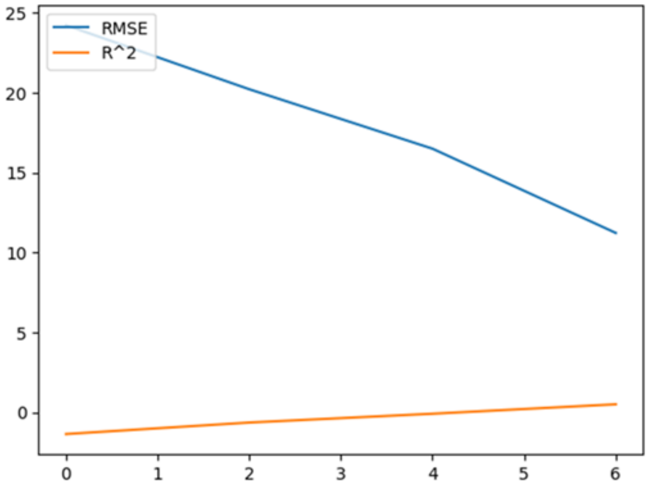

Figure 4.14: NVDA – RMSE vs R2 score

Table 4.2 and Figure 4.14 show that we got better results for the AAL dataset as compared to NVDA. The algorithm was performed moderately for NVDA which could be due to sudden spikes on the real-world dataset. The R2 score increases with several epochs indicating that if we provide more data points to the model it will improve.

5. Conclusion and Future work

This project depicts an RNN based on LSTM through which time series forecasting for AAL and NVDA assets was performed. The model showed promising results. It showed that it can capture the values for both the datasets in the testing phase. For future work the complexity of the model can be increased by adding more functions in the cell and more datapoints can be generated through real-time data so that the model can fare better for other stocks as well.

References

[1] Reddy, V. Kranthi Sai, and Kranthi Sai. "Stock market predictionusing machine learning." International Research Journal of

Engineering and Technology (IRJET) 5.10 (2018): 1033-1035.

[2] Reddy, V. Kranthi Sai, and Kranthi Sai. "Stock market predictionusing machine learning." International Research Journal of

Engineering and Technology (IRJET) 5.10 (2018): 1033-1035.

[3] Moritz, B., & Zimmermann, T. (2016). Tree-based conditional portfolio sorts: The relation between past and future stock returns. Available at SSRN 2740751.

[4] S. Goswami and S. Yadav, "Stock Market Prediction Using Deep Learning LSTM Model," 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Pune, India, 2021, pp. 1-5, doi: 10.1109/SMARTGENCON51891.2021.9645837.

[5] C. V. Gonzalez Zelaya, "Towards Explaining the Effects of Data Preprocessing on Machine Learning," 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 2019, pp. 2086-2090, doi: 10.1109/ICDE.2019.00245.

[6] A. Kumar and D. R. Rizvi, "Knowledge Weightage Calculation: an AI & ML based smart modelling for text- content summarization and quantification," 2023 1st International Conference on Intelligent Computing and Research Trends (ICRT), Roorkee, India, 2023, pp. 1-7, doi: 10.1109/ICRT57042.2023.10146669.

[7] A. Pulver and S. Lyu, "LSTM with working memory," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 2017, pp. 845-851, doi: 10.1109/IJCNN.2017.7965940.

[8] R. A. Viswambaran, G. Chen, B. Xue and M. Nekooei, "Evolutionary Design of Long Short Term Memory (LSTM) Ensemble," 2020 IEEE

Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, 2020, pp. 2692-2698, doi: 10.1109/SSCI47803.2020.9308393.

[9] J. P. Tan, A. L. A. Ramos, M. V. Abante, R. L. Tadeo and R. R. Lansigan, "A Performance Review of Recurrent Neural Networks Long Short-Term Memory (LSTM)," 2022 3rd International Conference for Emerging Technology (INCET), Belgaum, India, 2022, pp. 1-5, doi: 10.1109/INCET54531.2022.9824567.

[10] P. Hu, Z. Li, D. Tian and J. Zhang, "RUL Prediction Based on Improved LSTM Network Structure," 2022 34th Chinese Control and Decision Conference (CCDC), Hefei, China, 2022, pp. 1901-1906, doi: 10.1109/CCDC55256.2022.10033753.

[11] N. S. Malinović, B. B. Predić and M. Roganović, "Multilayer Long Short-Term Memory (LSTM) Neural Networks in Time Series Analysis," 2020 55th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), Niš, Serbia, 2020, pp. 11-14, doi: 10.1109/ICEST49890.2020.9232710.

Engineering and Technology (IRJET) 5.10 (2018): 1033-1035.

Education (TURCOMAT) 12.11 (2021): 2847-2854.

[12] Reddy, V. Kranthi Sai, and Kranthi Sai. "Stock market predictionusing machine learning." International Research Journal of

Engineering and Technology (IRJET) 5.10 (2018): 1033-1035.