Introduction

The opening price of a financial asset is a pivotal moment that dictates the trajectory of the day's trading activities. It serves as the initial reference point for investors and traders, shaping their perceptions and strategies for the trading session ahead. However, determining this opening price is far from straightforward, as it is influenced by a myriad of dynamic factors spanning market sentiment, economic indicators, breaking news events, and the intricate interplay of historical price movements.

Market sentiment, often driven by human emotions and perceptions, plays a significant role in setting the tone for trading sessions. Positive sentiment can fuel optimism and drive prices higher, while negative sentiment can instigate fear and lead to sell-offs. Economic indicators, ranging from employment data to GDP growth figures, provide valuable insights into the overall health of the economy and can sway investor sentiment accordingly.

Moreover, breaking news events, whether geopolitical developments, corporate earnings reports, or central bank announcements, have the power to cause sudden shifts in market sentiment and prices. The ability to anticipate and react to these events can make a crucial difference in trading outcomes.

Traditional methods of analysis, such as technical and fundamental analysis, have long been employed to forecast price movements. However, these approaches often fall short in capturing the complex and non-linear patterns inherent in financial data. Technical analysis relies on historical price charts and patterns to predict future price movements, while fundamental analysis assesses the intrinsic value of an asset based on factors such as earnings, dividends, and macroeconomic conditions. While valuable, these methods may struggle to adapt to rapidly changing market dynamics and unexpected events.

Enter deep learning, a subset of artificial intelligence that excels in learning intricate patterns and relationships from vast amounts of data. Deep learning models, powered by neural networks, can analyze massive datasets with speed and precision, uncovering hidden patterns that may elude traditional analysis techniques. By ingesting historical price data alongside a diverse array of contextual factors, deep learning models can potentially offer more accurate and nuanced predictions of opening prices.

The appeal of deep learning lies in its adaptability and ability to evolve with changing market conditions. Unlike rule-based algorithms or static models, deep learning models can continuously learn and refine their predictions over time, adapting to new information and market dynamics. This dynamic nature makes them well-suited for the inherently unpredictable nature of financial markets.

In conclusion, the accurate prediction of opening prices in financial markets is a challenging yet essential task for traders and investors. While traditional approaches have their merits, deep learning offers a promising avenue for enhancing the accuracy and reliability of price predictions. By harnessing the power of neural networks and advanced algorithms, deep learning models can provide valuable insights for making informed trading decisions in today's fast-paced and ever-changing financial landscape.

Objectives

1. Develop a robust deep learning model capable of accurately predicting opening prices of financial assets by leveraging historical price data and relevant contextual factors.
2. Explore and implement various deep learning architectures, such as recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and convolutional neural networks (CNNs), to identify the most effective model for price prediction in financial markets.
3. Evaluate the performance of the developed models using rigorous validation techniques and industry-standard metrics, including mean squared error (MSE), mean absolute error (MAE), and accuracy scores, to ensure the reliability and accuracy of predictions.
4. Investigate the impact of different hyperparameters, data preprocessing techniques, and feature engineering methods on the model's performance to optimize its predictive capabilities and generalization ability across diverse market conditions.

Literature Review

We apply LSTM recurrent neural networks (RNN) to forecast the correlation coefficient between the stock prices of two individual stocks. RNNs are particularly adept at capturing temporal dependencies inherent in time-series data, making them well-suited for analyzing stock price movements over time.

1. The utilization of LSTM cells within the RNN architecture enhances its ability to capture and model long-term dependencies in the stock price data. This enables the model to effectively learn and predict complex patterns and trends that unfold over extended periods.

2. In order to account for both linear and nonlinear relationships within the data, we integrate the Autoregressive Integrated Moving Average (ARIMA) model into our approach. The ARIMA model is proficient at identifying and filtering out linear trends present in the data, thus allowing the LSTM model to focus on capturing residual nonlinear patterns.

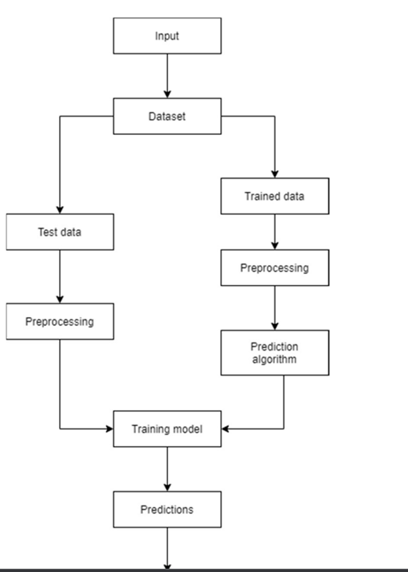
By combining the strengths of both LSTM networks and ARIMA models, we aim to develop a comprehensive forecasting framework capable of accurately predicting stock price correlation coefficients, thereby providing valuable insights for portfolio diversification and risk management strategies.

Furthermore, we compare the performance of Random Forests and LSTM networks (specifically CuDNNLSTM) as training methodologies for forecasting out-of-sample directional movements of constituent stocks of the S&P 500 index. This comparative analysis allows us to assess the effectiveness of these two methodologies in capturing and predicting intraday stock price movements over a historical period spanning from January 1993 to December 2018.

In our analysis, we introduce a multi-feature setting that incorporates not only the returns relative to the closing prices of stocks but also considers returns relative to the opening prices and intraday returns. By including these additional features, we aim to provide the model with richer information to better capture the dynamics of intraday price movements.

As part of our trading strategy, we adopt a systematic approach whereby, on each trading day, we identify and invest in the 10 stocks with the highest predicted probabilities of outperforming the market in terms of intraday returns. Conversely, we short sell the 10 stocks with the lowest predicted probabilities. By equalizing the monetary weight of each investment, we aim to optimize portfolio performance and potentially outperform the market benchmark.

Workflow diagram



Explanation

***Step1:*** ***Raw*** ***Stock*** ***Price*** ***Dataset:*** Day-wise past stock prices of selected companies are collected from the BSE (Bombay Stock Exchange) official website.

***Step*2:** ***Pre-processing***: This step incorporates the following:

a) Data discretization: Part of data reduction but with particular importance, especially for numerical data b) Data transformation: Normalization.

c) Data cleaning: Fill in missing values.

d) Data integration: Integration of data files. After the dataset is transformed into a clean dataset, the dataset is divided into training and testing sets so as to evaluate. Creating a data structure with 60 timesteps and 1 output.

***Step3:*** ***Feature*** ***Selection:*** In this step, data attributes are chosen that are going to be fed to the neural network. In this study Date & Close Price are chosen as selected features.

***Step*** ***4:*** ***Train*** ***the*** ***NN*** ***model***: The NN model is trained by feeding the training dataset. The model s initiated using random weights and biases. Proposed LSTM model consists of a sequential input layer

followed by 3 LSTM layers and then a dense layer with activation. The output layer again consists ofa dense layer with a linear activation function.

***Step5:*** ***Output*** ***Generation:*** The RNN generated output is compared with the target values and errordifference is calculated. The Backpropagation algorithm is used to minimize the error difference byadjusting the biases and weights of the neural network.

***Step*** ***6:*** ***Test*** ***Dataset*** ***Update:*** Step 2 is repeated for the test data set.

***Step*** ***7:*** ***Error*** ***and*** ***companies’*** ***net*** ***growth*** ***calculation:*** By calculating deviation we check thepercentage of error of our prediction with respect to actual price.

***Step*** ***8****:* ***Visualization***: Using Keras[21] and their function APIs the prediction is visualized.

***Step*** ***9***: Investigate different time interval: We repeated this process to predict the price at different time intervals. For our case, we took 2-month dataset as training to predict 3-month, 6-month, 1 year & 3 years of close price of the share. In this different time span, we calculate the percentage of error in the future prediction. This would be different for different sectors. So, this will help to find a framefor the particular sector to predict

future companies’ net growth.

Coding

1. In FrontEnd:
2. HTML:

Hypertext Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It defines the content and structure of web content. It is often assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript

1. CSS:

Cascading Style Sheets is a style sheet language used for specifying the presentation and styling of a document written in a markup language such as HTML or XML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

1. JavaScrip:

JavaScript, often abbreviated as JS, is a programming language and core technology of the Web, alongside HTML and CSS. 99% of websites use JavaScript on the client side for webpage behavior. Web browsers have a dedicated JavaScript engine that executes the client code.

1. In Backend:
2. Python :

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming

1. Machine learning:

Machine learning (ML) is a branch of artificial intelligence (AI) that involves developing algorithms and statistical models that allow computers to learn from data and perform tasks without explicit instructions. Machine learning algorithms can detect patterns in data and make predictions based on those patterns. ML algorithms gradually improve their accuracy.

1. Deep learning:

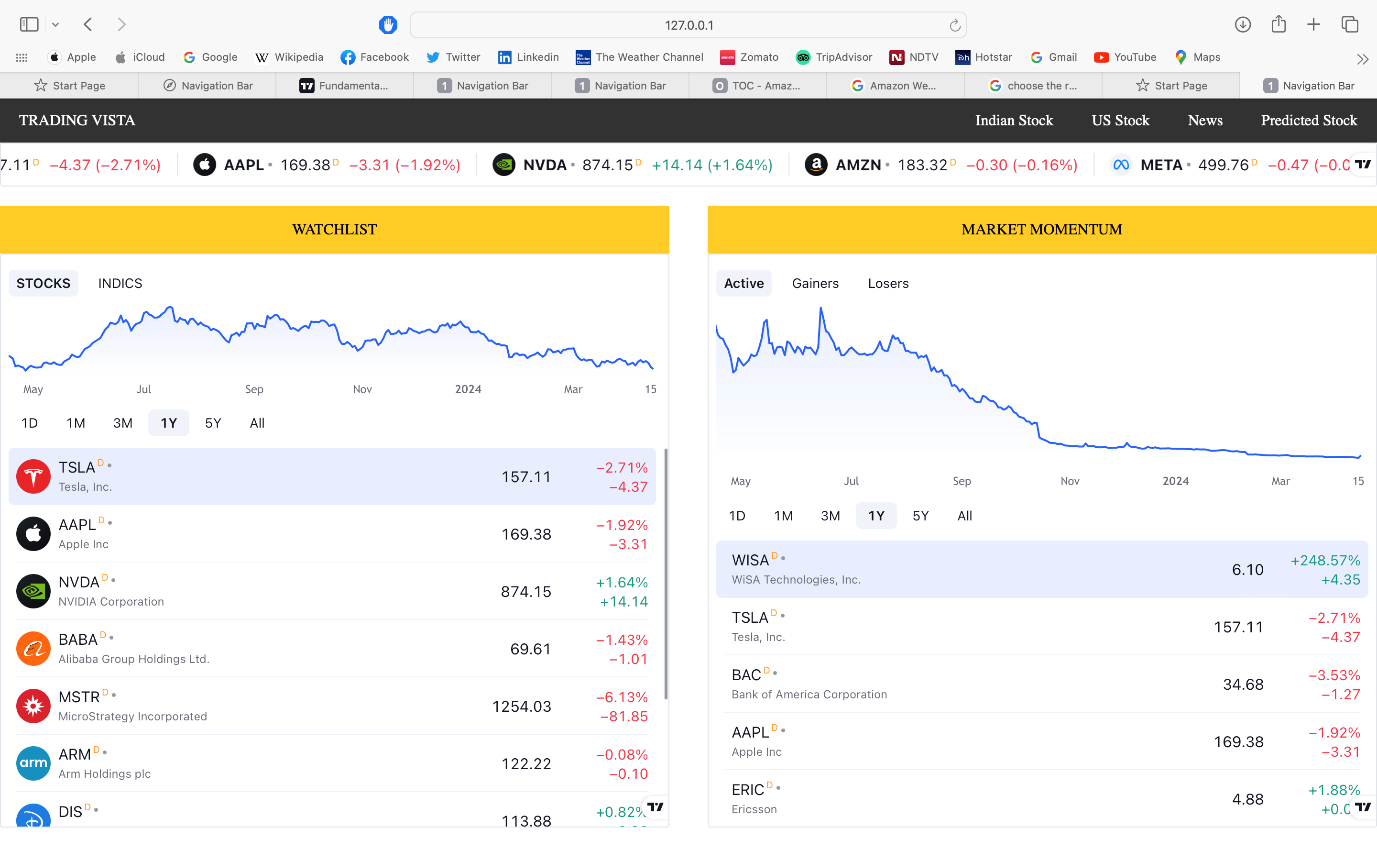
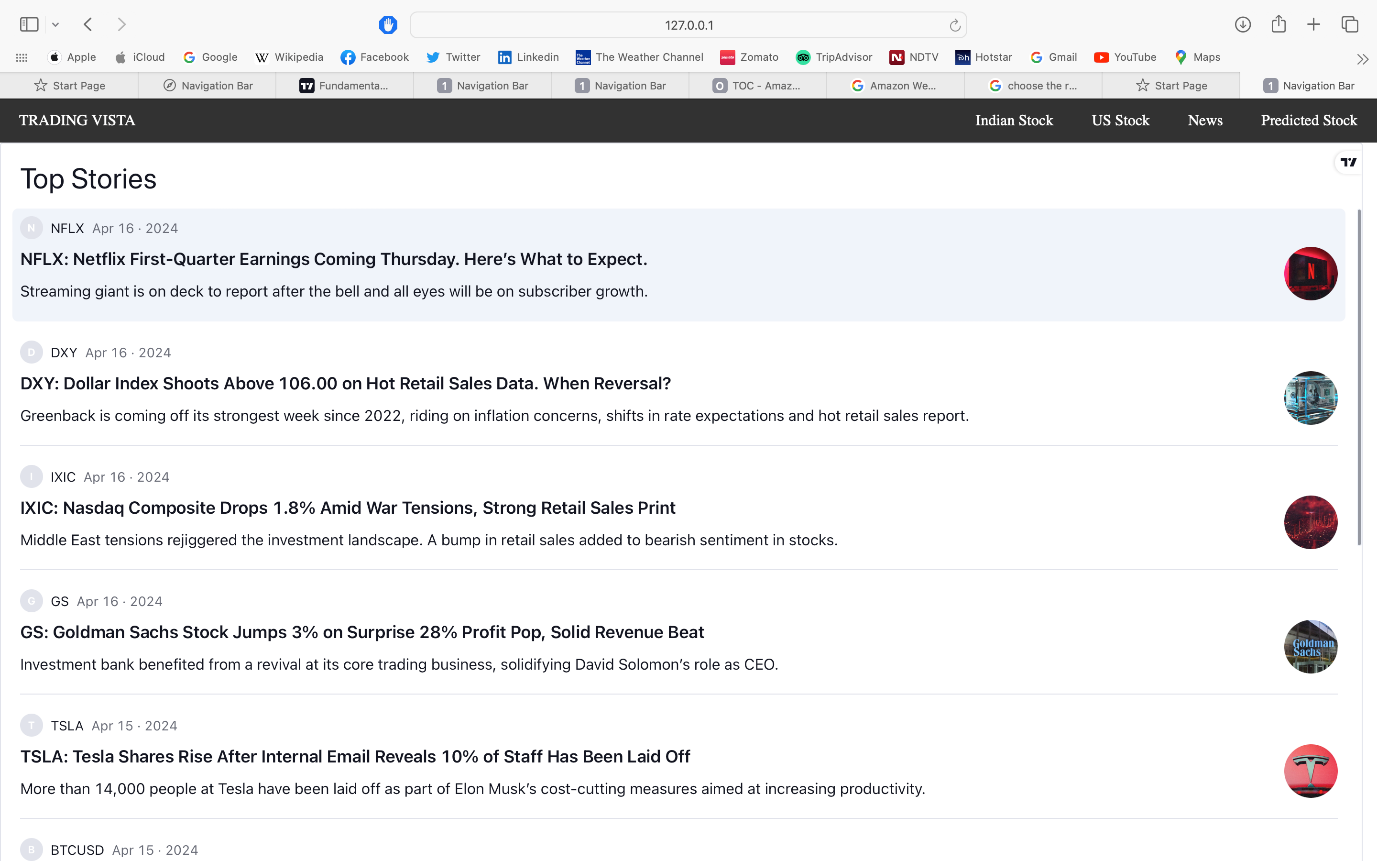
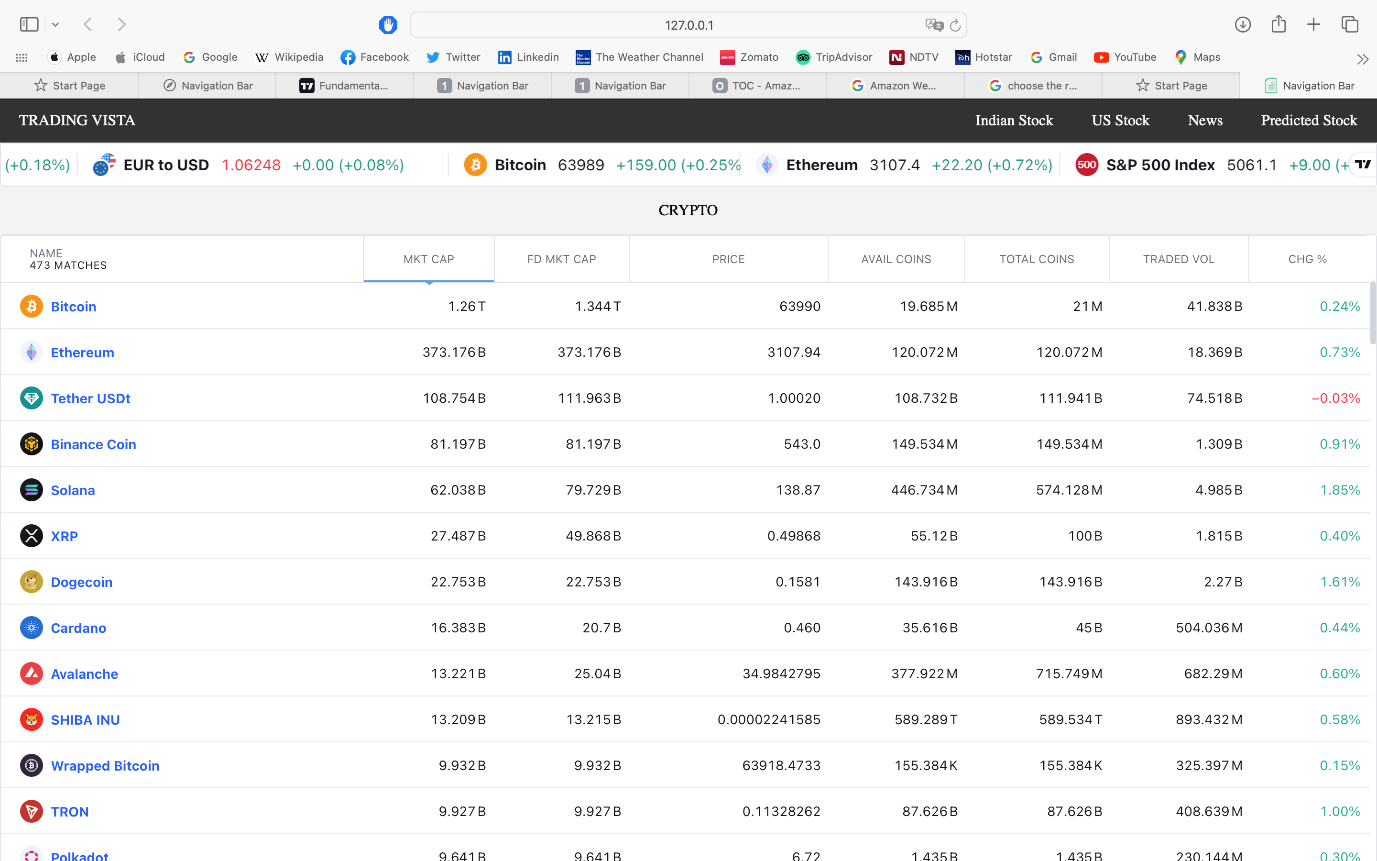
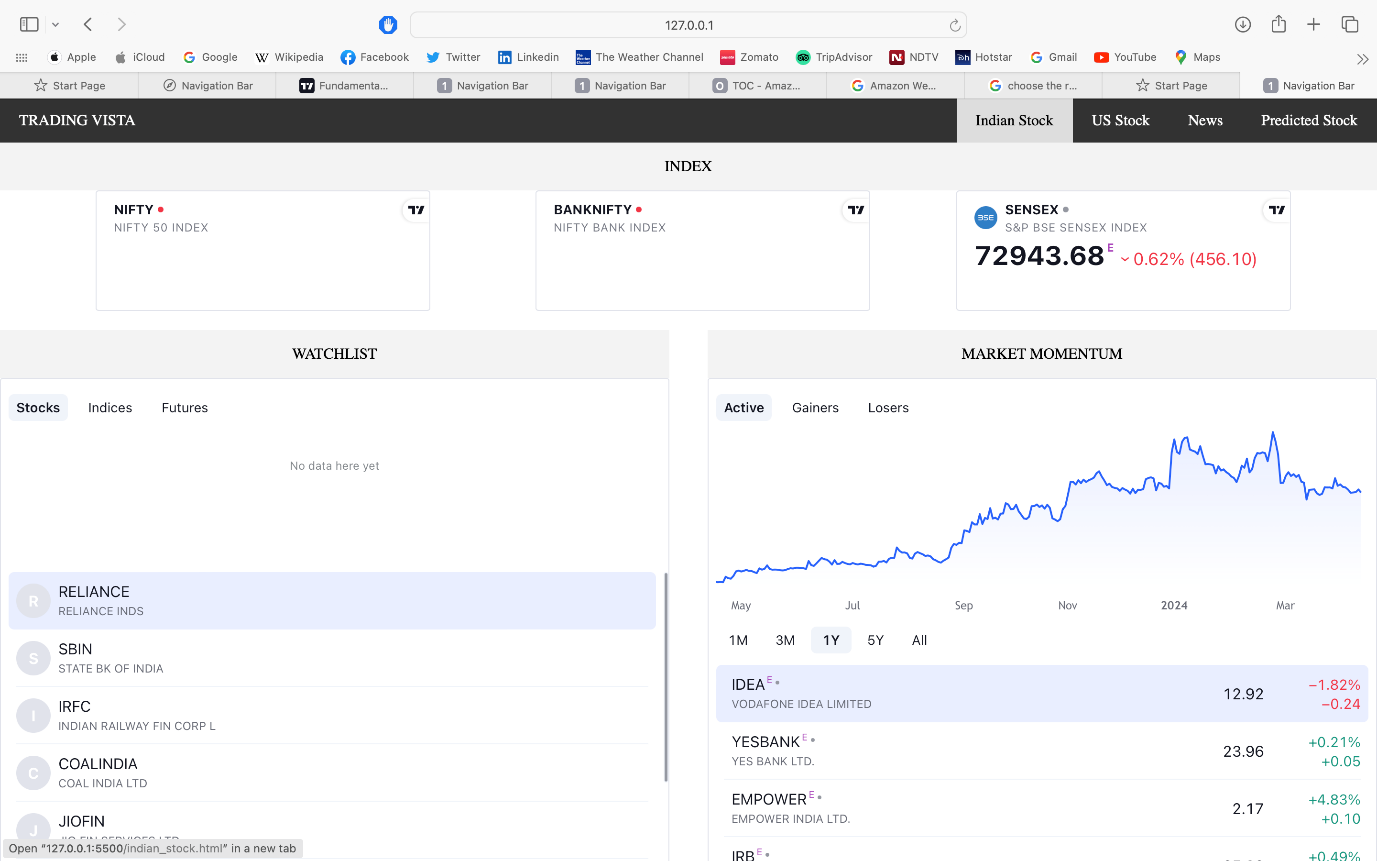
Deep learning is a method of artificial intelligence (AI) that teaches computers to process data in a way that mimics the human brain. Deep learning models can recognize complex patterns in pictures, text, sounds, and other data to produce accurate insights and predictions

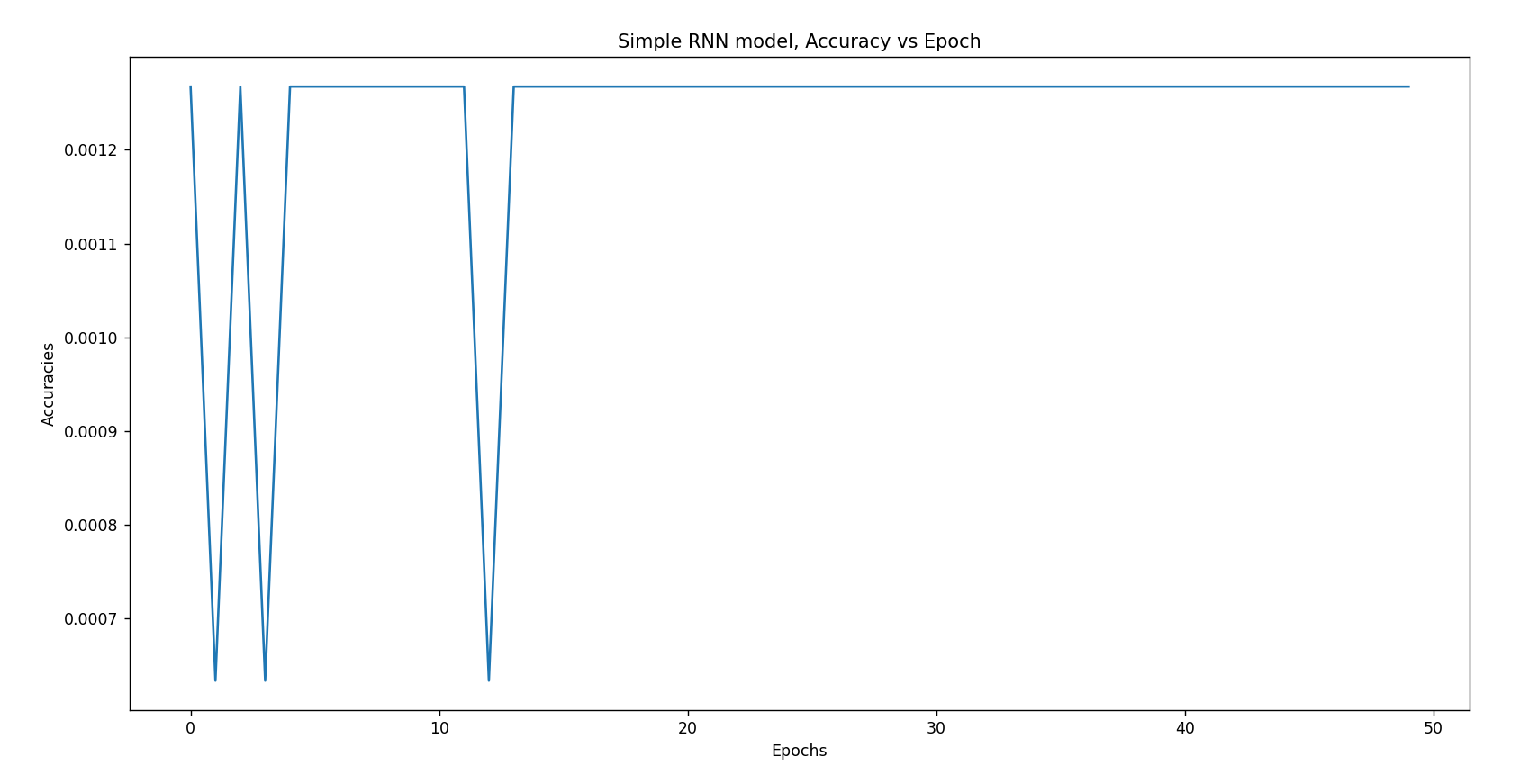
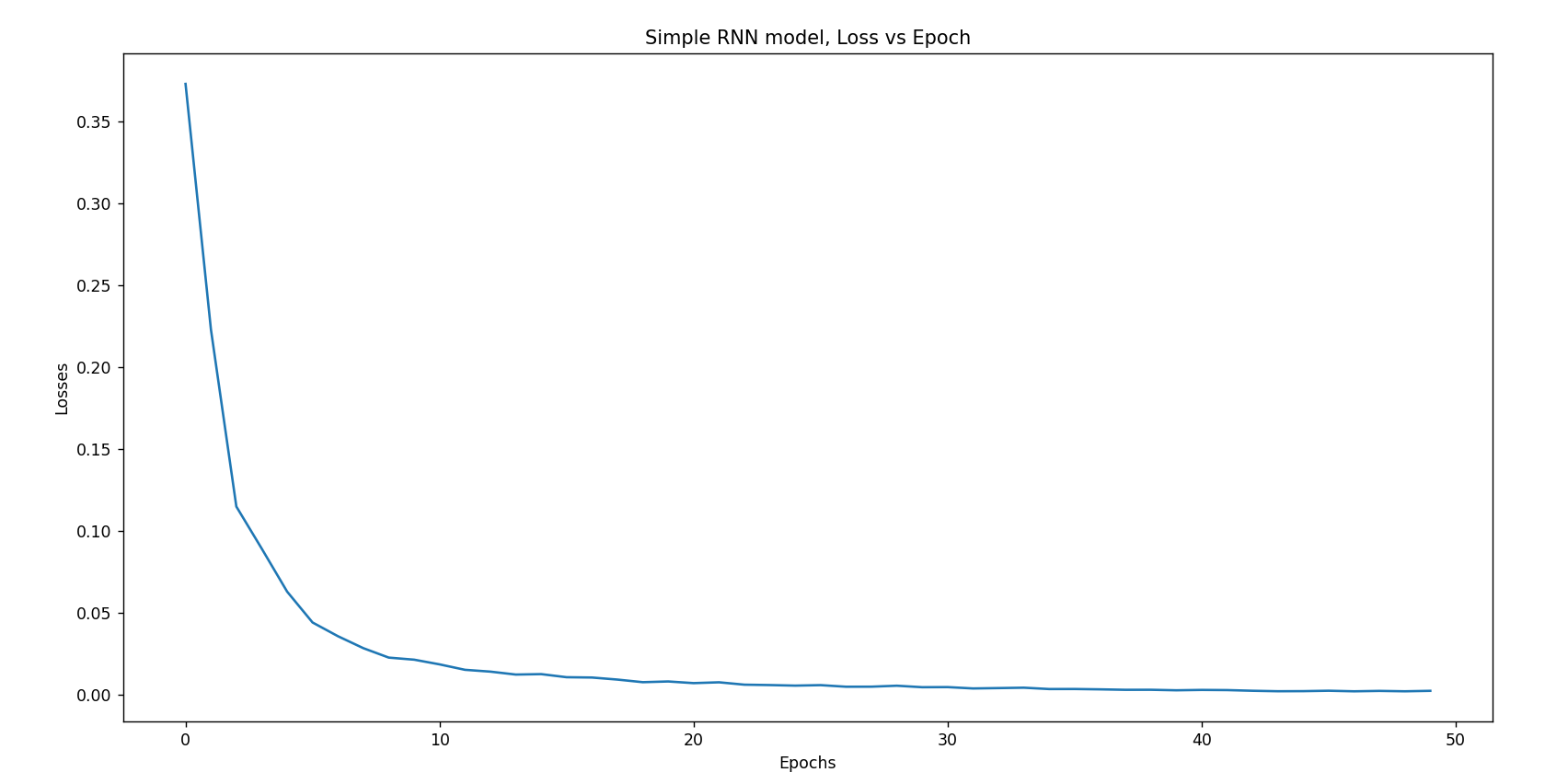
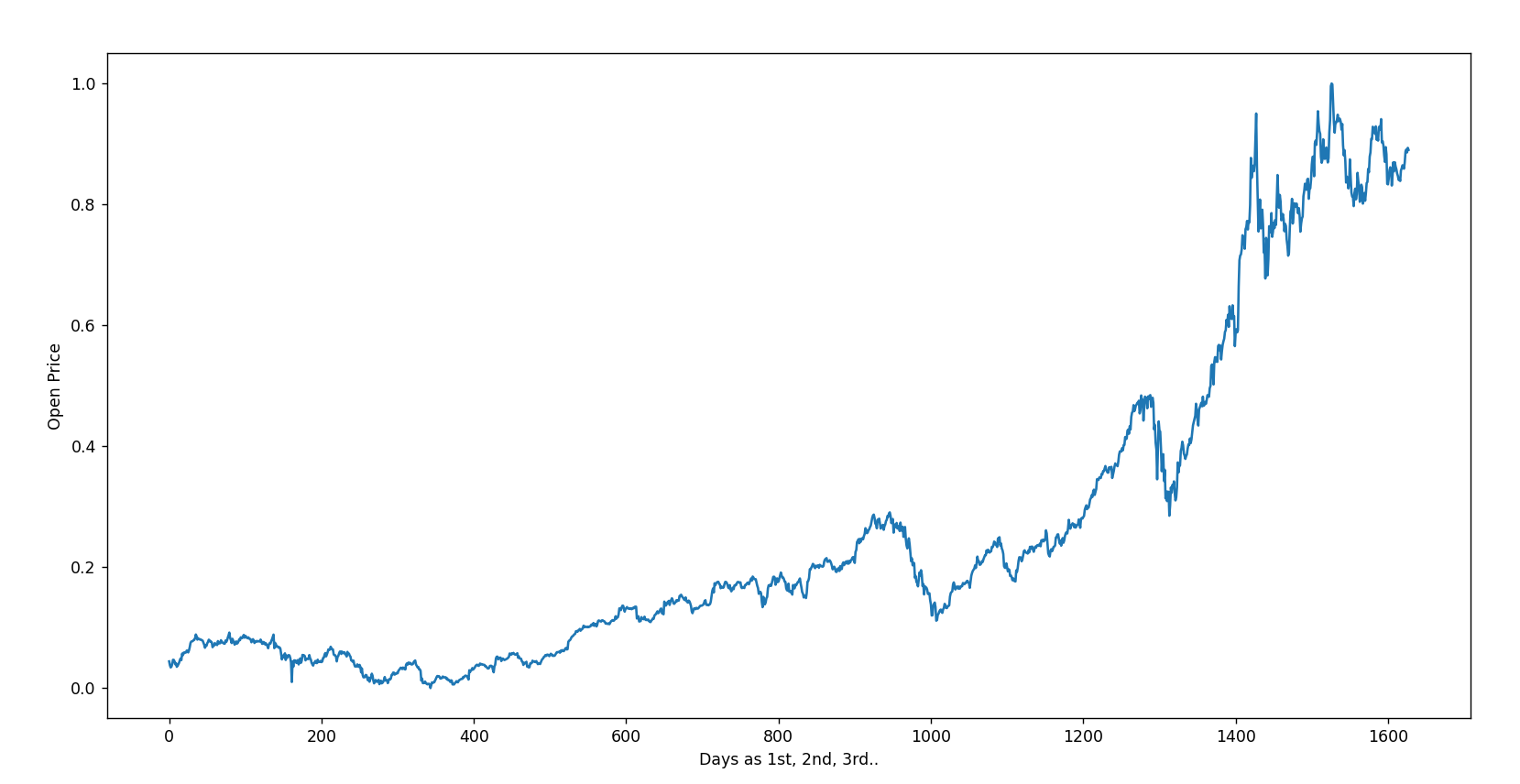
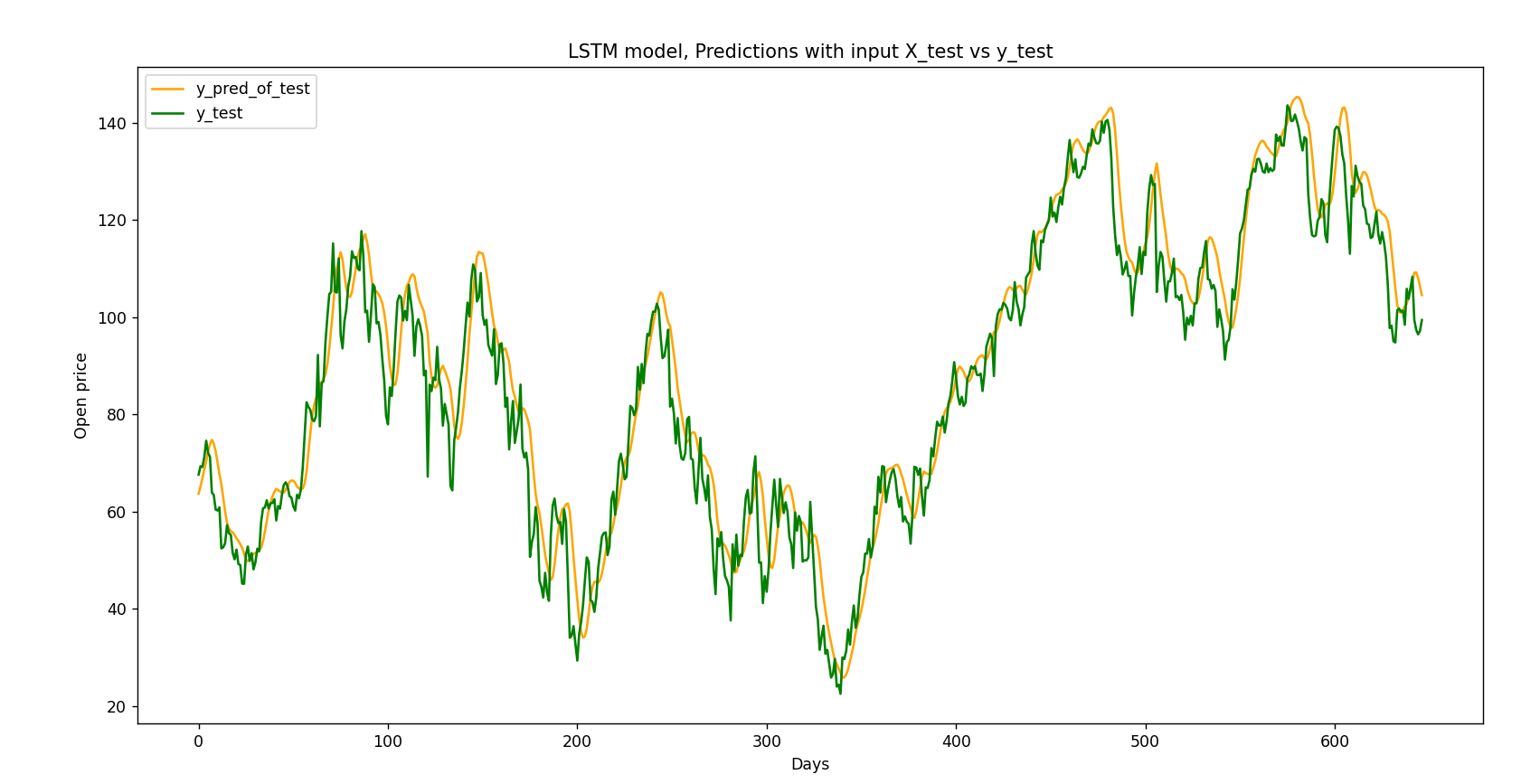
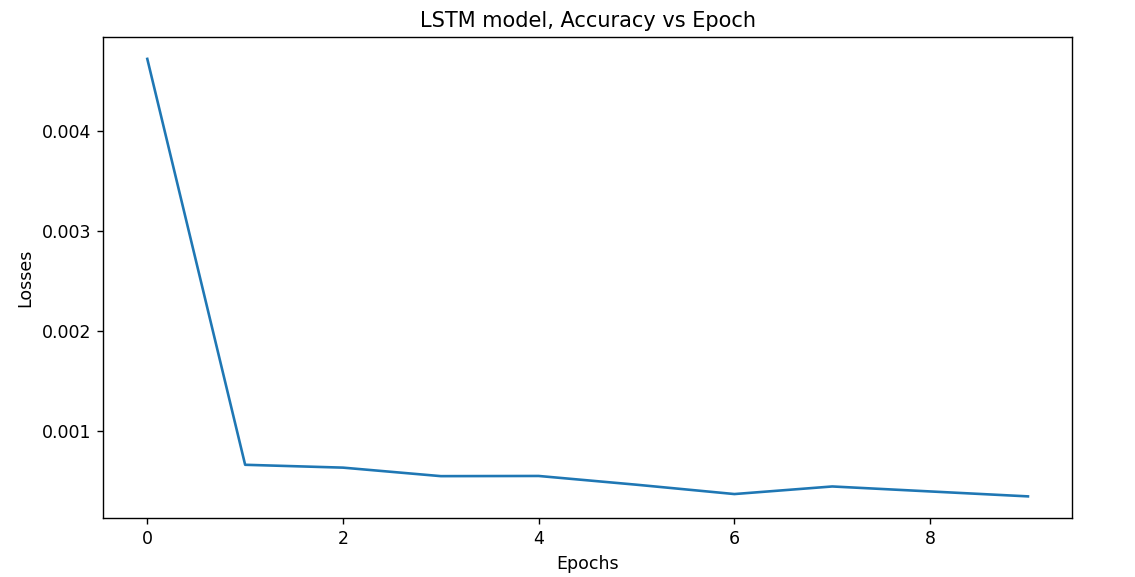
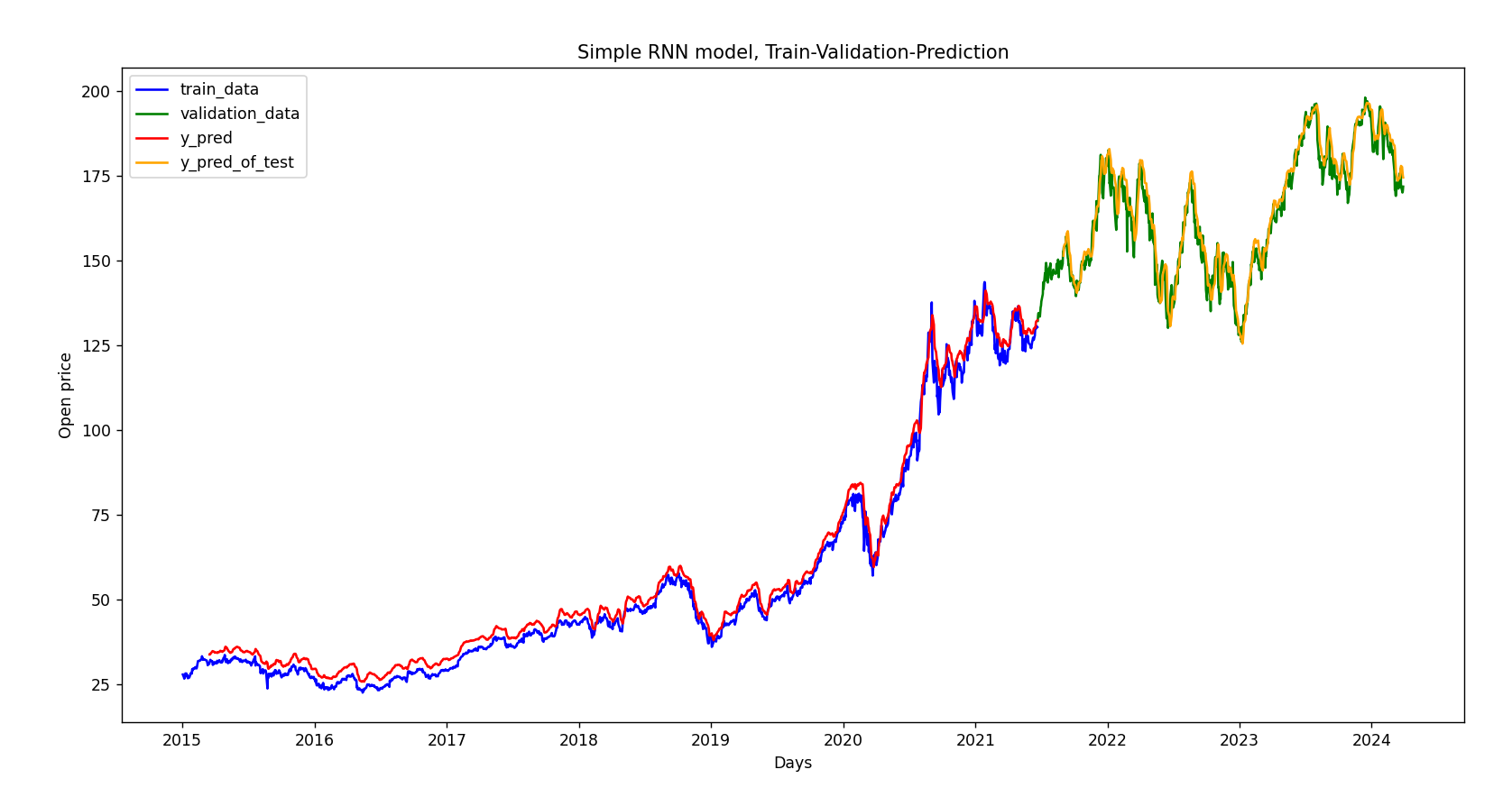
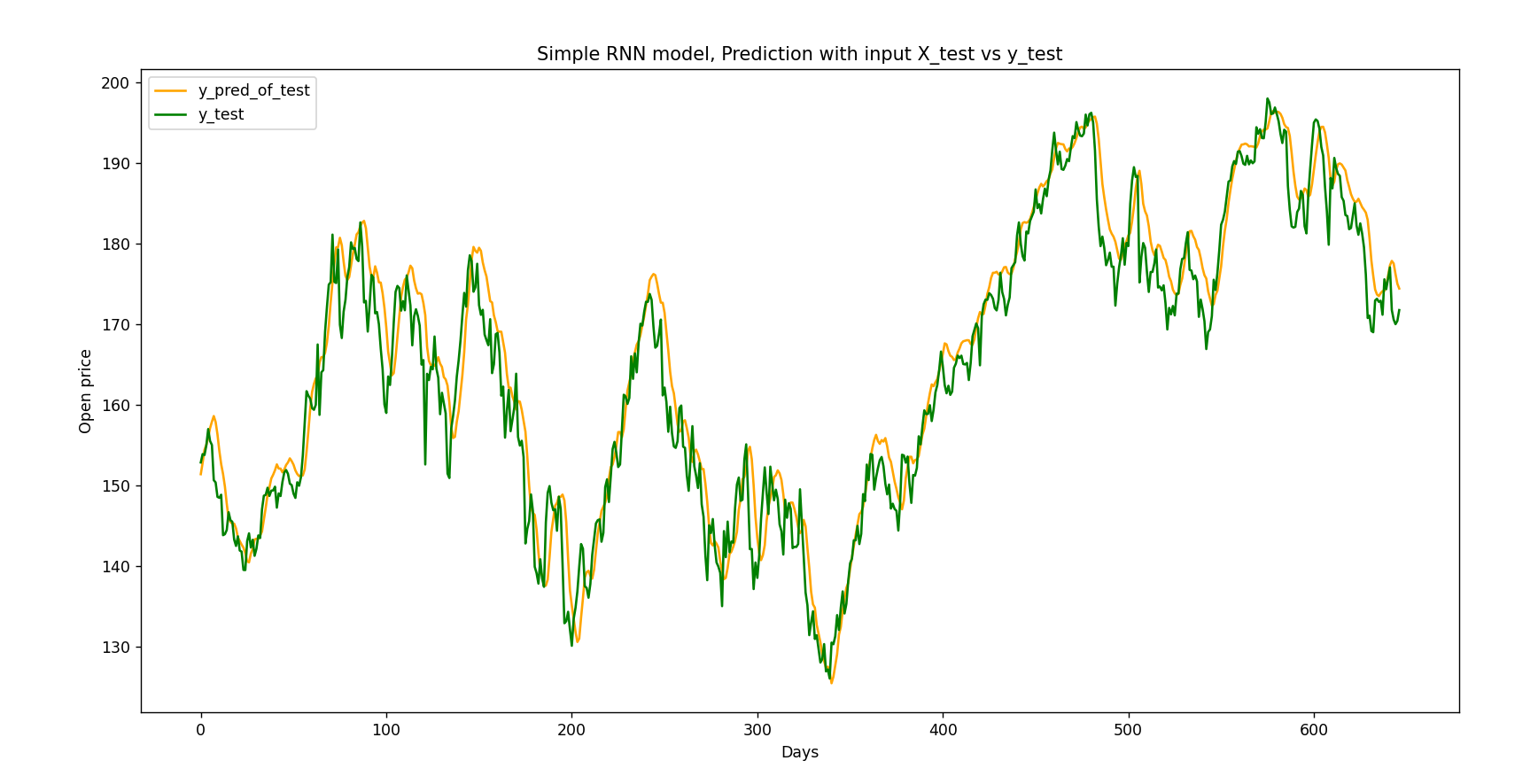
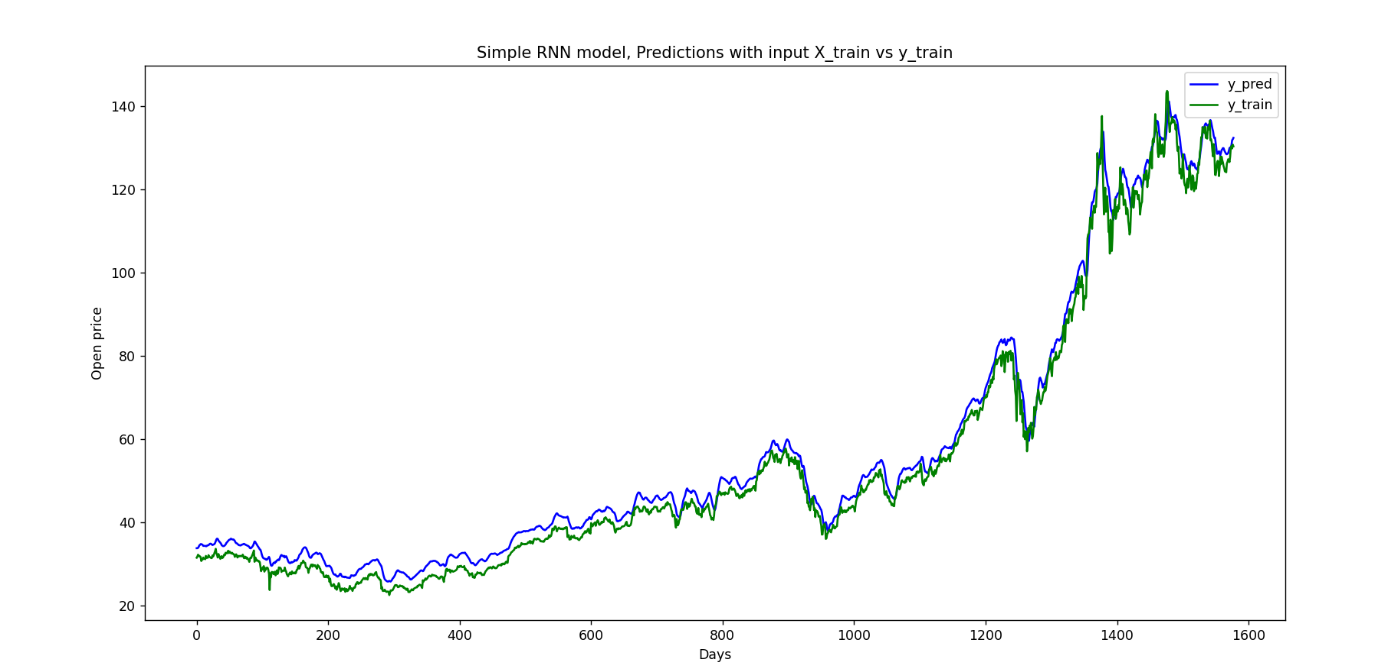
3)Framework:

1. Django:

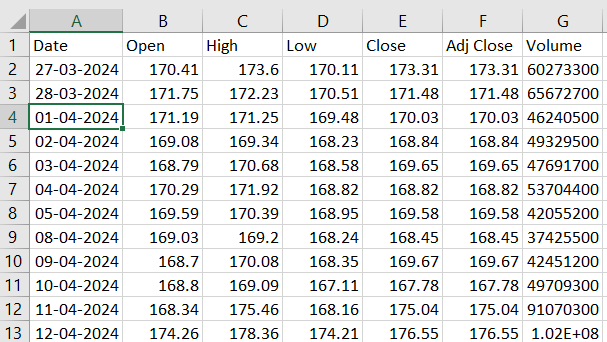
Django is a free and open-source, Python-based web framework that runs on a web server. It follows the model–template–views architectural pattern. It is maintained by the Django Software Foundation, an independent organization established in the US as a 501 non-profit.

Screenshots





|  |  |  |  |
| --- | --- | --- | --- |
| Apple | Prediction Date | Predicted Price | Real price |
| DateSet1 | 1-4 | 171.00285, 172.69574 | 171.19 |
| DateSet2 | 2-4 | 169.16489,  171.54726 | 169.08 |
| DateSet3 | 3-4 | 171.65458,  170.34845 | 168.78 |
| DateSet4 | 4-4 | 169.46512,  170.65476 | 170.29 |
| DateSet5 | 5-4 | 170.65161,  170.15676 | 169.59 |
| DateSet6 | 8-4 | 169.67155,  169.14753 | 169.03 |
| DateSet7 | 9-4 | 169.15413,  168.23416 | 168 |
| DateSet8 | 10-4 | 169.76135,  169.35972 | 168.80 |
| DateSet9 | 11-4 | 170.37885,  169.00175 | 168.34 |
| DateSet10 | 12-4 | 169.2161,  168.75496 | 174.26 |



Conclusion:

We were facing issues in technical analysis strategy vis-à-vis taking call based on the paradigm of supply and demand zones.

On that front we were incapable of fetching the precise opening point, we were getting the range instead, and the disparity between one end of the range and the other end were often so colossal that it undermined and seldom threatened our profit margin

Borrowing from our experience we arrived to the consensus that this disparity occurs due to one paramount factor - assumption of the closing price.

We came up with the notion of assuming the open price.

How did this project solve our conundrum and will continue to do so for the masses as well?

There’s a method in trading, namely BTST [Buy today, Sell tomorrow]

for that we require precise opening point which we weren’t getting earlier as aforementioned.

Hence, we fabricated a DEEP LEARNING MODEL of our own to grapple with this problem.

Our models accuracy for a single day and 10 day prediction is 99.75% and 99.002% respectively.

Furthermore, the derivation of accuracy :-

We calculated it by ascertaining MAPE by finding Absolute mean

MAPE is an acronym for MEAN ABSOLUTE PERCENTAGE ERROR

And the Error of our Deep Learning Model turned out to be 0.25% and 0.992 for a single day and 10 day respectively.

References:

1. <https://www.sciencedirect.com/science/article/pii/S1877050920307924>
2. <https://link.springer.com/article/10.1007/s00521-019-04504-2>
3. <https://ieeexplore.ieee.org/document/8342901>
4. <https://www.hindawi.com/journals/complexity/2021/5360828/>
5. <https://ieeexplore.ieee.org/abstract/document/8950831>
6. <https://www.researchgate.net/profile/Mallikarjuna-Pm/publication/348390803_Stock_Price_Prediction_Using_LSTM/links/5ffc6a23a6fdccdcb84a20f8/Stock-Price-Prediction-Using-LSTM.pdf>
7. <https://ieeexplore.ieee.org/abstract/document/9257950>