Stock Trend prediction using Deep Learning

Rhema K. Marneni
Rutgers University
Piscataway, NJ, USA
rkm110@scarletmail.rutgers.edu

Bhargavi Chinthapatla
Rutgers University
Piscataway, NJ, USA
bc837@scarletmail.rutgers.edu

***Abstract*— In the dynamic landscape of the stock market, where investors navigate the flow of company shares, informed decision-making is pivotal for successful investing. Beyond company performance, stock price movements are influenced by a myriad of factors such as economic conditions, inflation, political events, natural disasters, and even global pandemics, rendering the market highly unpredictable. In addition, social media has emerged as a significant influencer in investor decision-making. In response to these challenges, our project employs advanced deep learning algorithms for time series forecasting. By integrating the impact of social media into our models, we aim to enhance the accuracy of stock price movement predictions, providing valuable insights into future patterns and understanding the impact of using social media as a crucial factor in consideration.**

***Keywords*— AR-LSTM, CNN-LSTM, Deep Learning, Finance, LSTM, Mean Square Error, Netflix, Polarity, Sentiment Analysis, Social Media, Stock, Stock Market, Stock Prediction, Time Series Forecasting, Twitter**

I. Project Description

This project focuses on harnessing the power of advanced deep learning models, specifically CNN+LSTM and AR-LSTM, to forecast stock price movements. The chosen dataset, sourced through Yahoo Finance using the Python library yfinance, encompasses Netflix stock data spanning from 01 January 2020 to 10 December 2023. Recognizing the multifaceted nature of market dynamics, the project expands its predictive capabilities by integrating the influence of social media. Mokhtari et al, in their attempt to measure the impact of tweets on the stock market observed that in case of the company Apple "As the stock has been growing rapidly, people would expect it to continue likewise, resulting in many positive comments."[1] Tweets, being a significant driver of market sentiments, are scraped and processed to perform sentiment analysis using natural language processing techniques. The resulting sentiment polarity is seamlessly incorporated into the stock dataset as an additional feature, providing a nuanced understanding of the external factors impacting stock prices.
The project follows a systematic approach:

Data Acquisition:

- Utilizing the 'yfinance' Python library for stock data retrieval.
- Scraping Twitter data to capture social media sentiments.

Modeling:

- Implementing two distinct approaches, CNN+LSTM and AR-LSTM, for stock price prediction.
- Training models on historical Netflix stock data enriched with social media sentiment features. We aggregated the polarity value for each time period and calculated the mean as *p_mean*. We added this as an extra feature to our model while training.

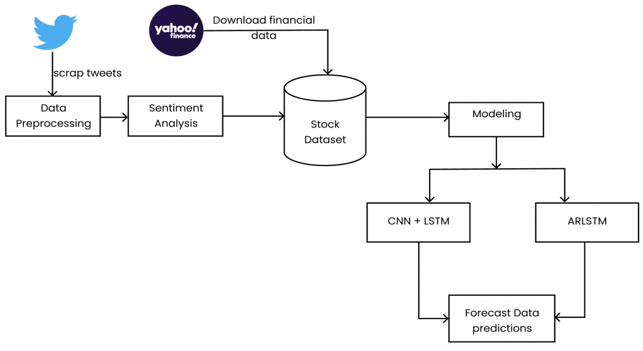

Figure 1: Conceptual model of the project

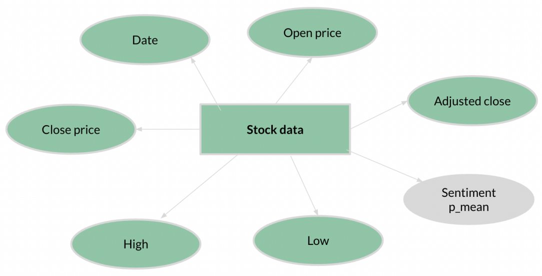

Figure 2: ER Diagram

Sentiment Analysis Integration:

Integrating sentiment polarity generated from Twitter data into the stock dataset.

Comparative Analysis:

- Evaluating the performance of the two models, CNN+LSTM and AR-LSTM, in predicting stock price movements.
- Employing mean square error as the loss function to assess model accuracy.

Real-time Evaluation:

- Assessing the robustness of the developed models against new, real-time data to validate their predictive capabilities.

This project extends beyond traditional stock prediction by incorporating the influential aspect of social media sentiments. The integration of sentiment analysis enables a more holistic understanding of market dynamics, recognizing the impact of external influences on stock prices. The comparative analysis provides insights into the efficacy of different deep learning approaches, offering valuable implications for future predictive modeling in financial markets.

Several essential libraries were used as listed below:
twitter-xlm-roberta-base-sentiment model
Numpy
Pandas
Tensorflow
Sklearn
Keras
Matplotlib

Target Users: Any general public. Investors, analysts, advisors etc. in the Finance Industry.

Project Timeline and Division of Labor:

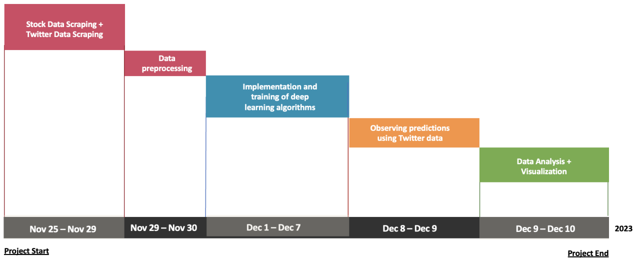
Stock Data Scraping + Twitter Data Scraping
Data preprocessing
Implementation and training of deep learning algorithms
Observing predictions using Twitter data
Data Analysis + Visualization
Nov 25 – Nov 29
Nov 29 – Nov 30
Dec 1 – Dec 7
Dec 8 – Dec 9
Dec 9 – Dec 10
2023
Project Start
Project End

Figure 3: Project Timeline

Rhema Marneni: Data Collection and Preprocessing, CNN+LSTM model for Stock Data without Twitter Data, AR-LSTM model for Stock Data with Twitter Data, Comparison of Trends and Data Analysis

Bhargavi Chinthapatla: Data Collection and Preprocessing, CNN+LSTM model for Stock Data with Twitter Data, AR-LSTM model for Stock Data without Twitter Data, Comparison of Trends and Data Analysis

II. Data Collection

2.1 Stock Price Data:

The primary data source for collecting historical stock prices is Yahoo Finance, accessed through the yfinance library in Python. Yahoo Finance is a reputable financial platform that provides comprehensive and reliable historical stock data. The yfinance library facilitates seamless data retrieval, allowing users to access a wealth of information pertaining to various financial instruments.

Attributes Description:

Open Price: The opening price of the stock at the beginning of a trading period.
High Price: The highest traded price of the stock during the specified period.
Low Price: The lowest traded price of the stock during the specified period.
Close Price: The closing price of the stock at the end of a trading period.
Adjusted Close Price: The closing price adjusted for dividends and stock splits, providing a more accurate representation of the stock's value.

Yahoo Finance is known for its data accuracy and reliability, contributing to the robustness of the collected stock price data. The yfinance library gives access to Yahoo Finance APIs to retrieve financial data like stocks, cryptocurrency etc., and further ensures that the retrieved information is structured and formatted for seamless integration with other components of the Stock Trend Prediction project.

2.2 Twitter Tweet Collection:

Twitter, a leading social media platform, serves as the primary data source for collecting real-time tweets related to specific stock tickers. The Twitter API is employed to access and retrieve a stream of tweets containing relevant information about the chosen stocks.
Access to the Twitter API is facilitated through authentication using API keys and tokens. These credentials are obtained by creating a Twitter Developer account. Once authenticated, users gain programmatic access to the wealth of public tweets on the platform. To focus on tweets relevant to specific stocks,

keywords, hashtags, or mentions associated with the stock tickers are defined. For instance, if the stock ticker of interest is "NFLX" for Netflix., keywords such as "AAPL," "Apple," or related hashtags may be utilized to filter tweets.

III. Data Preprocessing

3.1 Stock Data Preprocessing

Missing values, common in financial datasets due to market closures, were addressed by forward and backward filling to ensure a continuous and complete dataset. The date column was converted to a datetime format for consistency and ease of analysis. Relevant features, such as Open, High, Low, Close, and Adjusted Close prices, were selected for analysis. This provides a comprehensive view of the stock's historical performance. Furthermore, we normalized the stock prices data for training purposes. Normalization scales the input data to a consistent range. We used MinMaxScaler, which scales the data to a range between 0 and 1. This ensures that all features have a similar scale, preventing certain features from dominating others simply due to their original scale.

Hyperparameters and another information:

- Ratio of Training data to Testing data = 80:20
- Sequences of 60 days of stock prices as input.
- Number of records = 994
- Size per record = 10B
- Number of epochs = 50
- Dropout = 10%
- Learning rate = 0.01

3.2 Twitter Data Preprocessing

Twitter data preprocessing is a crucial step in refining raw text for sentiment analysis. The process begins with tokenization, breaking down tweets into individual words. Lowercasing ensures uniformity, preventing variations in letter cases from affecting the analysis. URLs and symbols are removed to focus on the textual content. Emojis are replaced with their corresponding textual representation, enhancing the interpretability of emotional context. Stopwords, common but less meaningful words, are excluded to reduce noise, and lemmatization consolidates word variations. These steps collectively contribute to a more streamlined and meaningful dataset, laying the groundwork for accurate sentiment analysis and interpretation of Twitter data in the context of our project.

IV. Algorithms

4.1 CNN-LSTM

CNN has the characteristic of paying attention to the most obvious features in the line of sight, so it is widely used in feature engineering. LSTM has the characteristic of expanding according to the sequence of time, and it is widely used in time series. According to the characteristics of CNN and LSTM, a stock forecasting model based on CNN-LSTM is established. The model structure diagram is shown in Figure 4, and the main structure is CNN and LSTM, including input layer, one-dimensional convolution layer, pooling layer, LSTM hidden layer, and full connection layer.

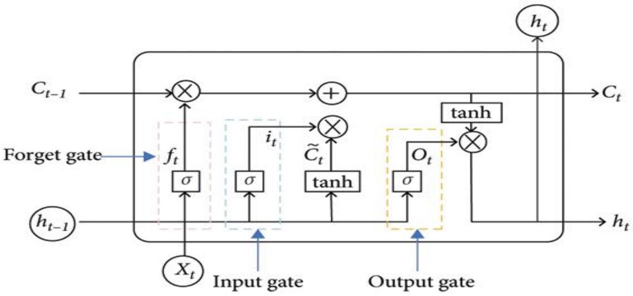

Figure 4: CNN-LSTM structure diagram

CNN-LSTM Training and Prediction Process:

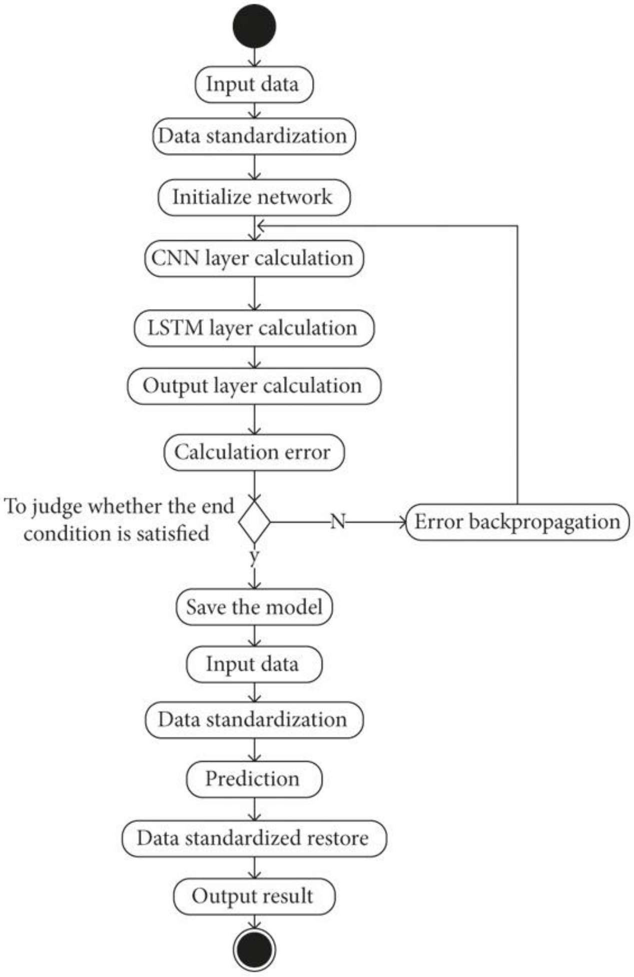

Figure 5: Activity diagram of CNN-LSTM training and prediction process

The main steps are as follows:

(1) Input data: Input the data required for CNN-LSTM training.

(2) Data Normalisation

(3) Initialize Network: Initialize the weights and biases of each layer of the CNN-LSTM.

(4) CNN Layer Calculation: The input data are successively passed through the convolution layer and pooling layer in the CNN layer, the feature extraction of the input data is carried out, and the output value is obtained.

(5) LSTM Layer Calculation: The output data of the CNN layer are calculated through the LSTM layer, and the output value is obtained.

(6) Output Layer Calculation: The output value of the LSTM layer is input into the full connection layer to get the output value.

(7) Calculation Error: The output value calculated by the output layer is compared with the real value of this group of data, and the corresponding error is obtained.

(8) To judge whether the end condition is satisfied: The conditions for the end are to complete a predetermined number of cycles, the weight is lower than a certain threshold, and the error rate of the forecasting is lower than a certain threshold. If one of the conditions for the end is met, the training will be completed, update the entire CNN-LSTM network, and go to step 10; otherwise, go to step 9.

(9) Error Backpropagation: Propagate the calculated error in the opposite direction, update the weight and bias of each layer, and go to step 4 to continue to train the network.

(10) Save the model: Save the trained model for forecasting.

(11) Input Data: Input the input data required for the forecasting.

(12) Data Standardization: The input data are standardized according to formula (8).

(13) Forecasting: Input the standardized data into the trained model of CNN-LSTM, and then get the corresponding output value.

(14) Data Normalised Restore: The output value obtained through the model of CNN-LSTM is the standard Normalized value, and the Normalized value is restored to the original value.

(15) Output Result: Output the restored results to complete the forecasting process.

4.2 AR-LSTM (AutoRegressive Long Short-Term Memory):

Autoregressive Component (AR):

The autoregressive component of the model leverages historical observations to predict the future value of the time series. It considers the linear relationships between past and present values, allowing the model to capture the inherent sequential dependencies in the data.

Long Short-Term Memory (LSTM) Component:

The LSTM component is incorporated to address the limitations of pure autoregressive models when dealing with complex, nonlinear patterns and long-range dependencies. LSTMs are well-suited for capturing and learning patterns across various time scales, making them effective in modeling intricate relationships in time series data.

Model Integration:

The AR-LSTM model integrates both components in a synergistic manner. The autoregressive output serves as an input to the LSTM, allowing the model to learn from the residual errors and capture higher-order patterns that might not be adequately captured by the autoregressive component alone.

Training and Optimization:

The AR-LSTM model undergoes a training process where it learns to minimize the difference between its predictions and the actual observed values. This involves optimizing model parameters, such as autoregressive coefficients and LSTM weights, through backpropagation and gradient descent.

V. Results

A. Stock Price Prediction

Based on the trained model, we predicted the stock price movement of Netflix until January 01, 2024 as shown in figure 6 and figure 7. CNN+LSTM seems to have caught the stock price trend, but CNN as a part of its training process, smoothes out some features, so there is still room for improvement. AR-LSTM not only predicted the price movement trend, but also gave more accuracy in terms of the closing price values. We had 2,266,914 trainable parameters.

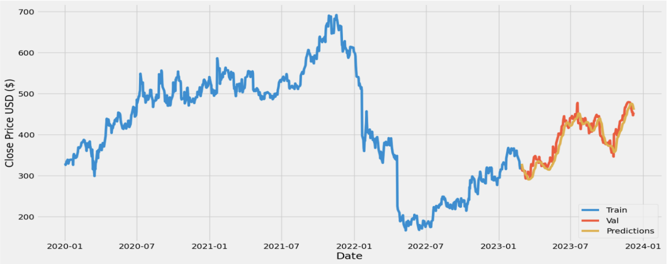

Figure 6: CNN+LSTM: Prediction of Closing Stock price till January 01, 2024

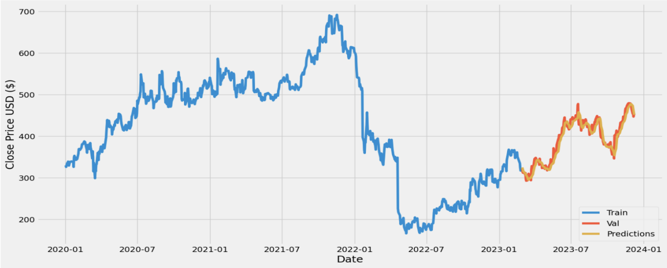

Figure 7: AR-LSTM: Prediction of Closing Stock price till January 01, 2024

B. Testing Accuracy

We chose the Mean Squared Error (MSE) as our loss function. Choosing the Mean Squared Error (MSE) as the loss function in stock prediction models is a common practice in regression problems, including time series forecasting. More particularly, because of its squared nature the curve converges smoothly.

$$\mathrm{MSE} = \overset{\text{Mean}}{\boxed{\frac{1}{n}\sum_{i=1}^{n}}}\Big(\overset{\text{Error}}{\boxed{Y_i - \hat{Y}_i}}\Big)^{\overset{\text{Squared}}{\boxed{2}}}$$

A lower MSE indicates that, on average, the predictions are closer to the actual values. We can observe from the prediction plots that the error reduced for the cases that considered twitter sentiment.

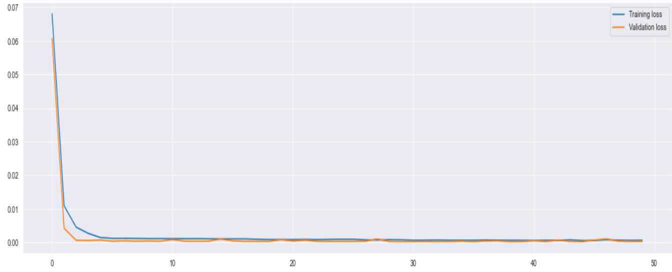

Figure 8: CNN+LSTM: Training and validation loss with Twitter Data

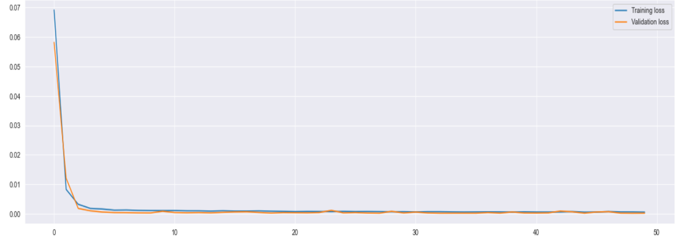

Figure 8: AR-LSTM: Training and validation loss with Twitter Data

Figure 8 and Figure 9 show the loss curve during training and validation. It is apparent that AR-LSTM converges faster than CNN+LSTM.

C. Stock Price Prediction using Twitter Data

Figure 10 and Figure 11 show the stock prediction on the testing data for CNN+LSTM without and with Twitter Data respectively.

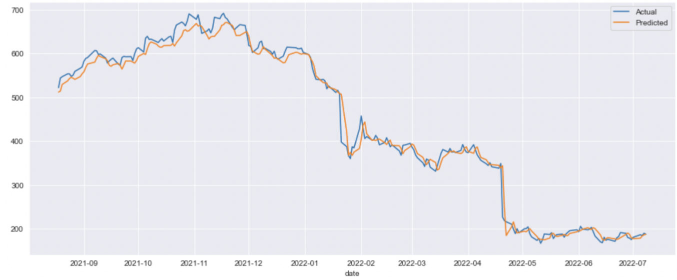

Figure 10: CNN+LSTM: Testing Prediction without Twitter Data

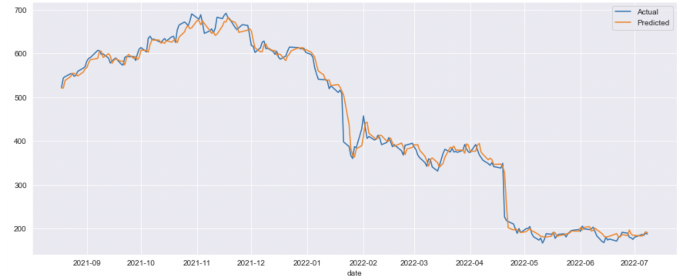

Figure 11: CNN+LSTM: Testing Prediction with Twitter Data

Although the price movement trend is captured well enough, adding an extra polarity feature resulted in more accuracy.

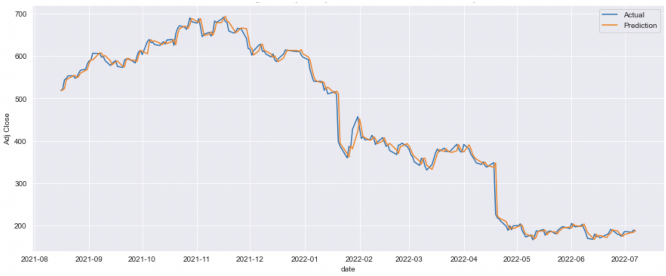

Figure 12: AR-LSTM: Testing Prediction without Twitter Data

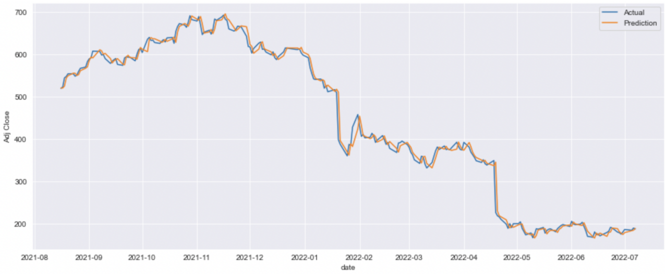

Figure 13: AR-LSTM: Testing Prediction with Twitter Data

Figure 12 and Figure 13 show the stock prediction on the testing data for AR+LSTM without and with Twitter Data respectively. Again, the price movement trend is captured more accurately when Twitter data is considered during training.

CNN+LSTM		**AR-LSTM**	
MSE Without Twitter	MSE With Twitter	MSE Without Twitter	MSE With Twitter
344.0639337957878	280.7222471708451	236.1959848642219	230.88280552424638

Table 1: Mean-Squared Error values for both the models with and without Twitter Data in consideration

VI. Conclusion

Our project successfully navigated through various stages, employing cutting-edge deep learning models—CNN+LSTM and AR-LSTM—to forecast Netflix stock movements until January 01, 2024. The models were trained and tested on a dataset comprising stock data from January 2020 to December 2023, augmented with the integration of sentiment features derived from Twitter data. The CNN+LSTM model showcased proficiency in capturing the overall stock price trend, albeit with some feature smoothing inherent to its training process. However, room for improvement was identified. On the other hand, the AR-LSTM model not only predicted the price movement trend but also demonstrated enhanced accuracy in terms of closing price values, showcasing its superior performance.

The inclusion of Twitter sentiment data emerged as a pivotal factor in refining time series analysis. By considering the sentiments expressed on social media, the models exhibited a noticeable reduction in Mean Squared Error (MSE) values, affirming the positive impact of incorporating external factors such as investor sentiments from Twitter. The MSE values further validate the success of our approach.

VI. Future Work

Future work for this project involves a multifaceted approach to enhance the predictive capabilities of the employed deep learning models and extend their applicability. Optimization efforts will delve into advanced hyperparameter tuning to refine CNN+LSTM and AR-LSTM models, aiming for heightened predictive accuracy. The exploration of sentiment analysis will broaden to encompass diverse social media platforms beyond Twitter, enriching the dataset and providing a more comprehensive understanding of market sentiments. The models' adaptability across various stocks and sectors will be assessed to enhance their generalization capabilities. Real-time sentiment integration mechanisms will be implemented to ensure the models' responsiveness to swiftly changing market conditions. Experimentation with additional influential features, such as macroeconomic indicators and news sentiment, will enrich the models' predictive power. Efforts will also be directed towards enhancing model interpretability through frameworks that highlight key contributing factors. Ethical considerations associated with social media data usage will be addressed, and long-term predictions will be explored to assess model adaptability to extended market trends. The dynamic interpretation of models and assessing their potential impact on market behavior will also be crucial focal points for future research.

VII. References

[1] Mokhtari, Melvin & Seraj, Ali & Saeedi, Niloufar & Karshenas, Adel. (2023). The Impact of Twitter Sentiments on Stock Market Trends. 10.48550/arXiv.2302.07244.

[2]https://www.kaggle.com/datasets/omermetinn/tweets-about-the-top-companies-from-2015 -to-2020