ADVANCE ARTIFICIAL NEURAL NETWORKS

BIOE 6306
Summer 2023
Final Project

Enhancing Stock Market Forecasting using Deep Learning: A Multi-Model Approach with Sentiment Analysis Integration

Team:
Lokesh Reddy Yenimireddy- 2160361

Abstract –

The rapid development of technology that uses large amounts of big data to provide useful insights has become critical to solving many timely problems. These deep learning methods have a wide range of applications such as text recognition, image recognition, medical disease diagnosis, self-driving cars, etc. It also has an application in the financial sector, where one of its application problems is stock market forecasting. In our project, we created a novel way to predict the opening price of Apple stock using not only its previous opening price but also the sentiment score of previous day. The models we tried to build are Hybrid CNN-Stacked LSTM, CNN-Stacked GRU and Ensemble of RNN, LSTM and GRU. We also imported Hugging Faces' Financial Bert model to calculate sentiment scores for news articles. Finally, we created 3 models with different datasets, one containing only Apple's opening stock price and the other a combination of open stock price and sentiment score. The metric used in the comparison is RMSE and we also visualize these results for better understanding. Finally, we found that both the open and sentiment score datasets produced the best results with the Ensemble model. We also used a 4-day window size to forecast Apple's stock price for the next few days, providing an analogue of price considered from Monday to Thursday to forecast for Friday.

Keywords – CNN-Stacked LSTM, CNN-Stacked GRU, Ensemble models, Apple stock

Hypothesis –

Forecasting remains one of the most tricky though challenging, aspects of financial investment. It is complicated because it is affected by many factors, including government legislation, company sales and earnings, new articles, P/E ratio, etc. Stock market data is non-linear, variable, non-stationary and complex with no discernible pattern. Various studies and linear models such as ARIMA, ARIMAX and others are used to solve these problems. These algorithms do not collect or understand complex stock data. That is why we have created several deep learning models that have produced good results in areas such as speech recognition, audio processing, natural language processing and video processing. These deep learning approaches have shown excellent results in stock market forecasting. So the major purpose of this research is to forecast the price of Apple shares utilising two approaches.One strategy considers only Apple stock price, while the other considers both opening price and sentiment score when comparing these two approaches.

Deep learning approaches such as RNN, LSTM and GRU performed excellently in serial data processing and produced the best possible prediction compared to other machine learning techniques. In particular, RNNs are very good with data that has a short sequence, but there is a slight problem with large sequence data due to the possibility of vanishing gradient problem. To solve these problems, we can use LSTMs whose memory state keeps important information and discards unimportant information. LSTM also contains gates like forget date, input gate and output gate. In this study, we propose three models with different combinations, such as CNN-stacked LSTM, CNN-stacked GRU, and RNN, GRU, LSTM ensemble model, and take the average of its output and send them back to the neural network. In our experiment, we also identified a noval technique for determining a stock market's starting price, in which the opening price is merged with the previous day's sentiment scores collected from Financial Apple news followed by Hugging-Face FinancialBert. This hugging face model provides three labels indicating whether the text is positive, negative, or neutral, as well as a Sentiment score ranging from 0 to 1. I got the financial data from Apple stock data from ytfinances and the dates range from 2020-07-14 to 2023-07-14.

Background –

CNN:

CNNs have been highly effective in dealing with the images and other structured data structured because of their special design, as they can take advantage of spatial correlations between pixels within an image [2]. CNN is made up of different layers which are convolutional layers, pooling layers, and fully connected layers while convolutional layers are designed to extract meaningful features from the data.
They use convolution techniques, in which tiny filters or kernels are applied to the input image to capture local patterns and generate feature maps that emphasize key areas. Followed by the pooling layers which are used to reduce the dimensions of the future maps. In our hybrid models the first layer of our model is CNN layer which consist 64 filters with relu activation function.

RNN:

Traditional neural networks rely on the independent nature of all inputs. When dealing with sequential data or inputs and outputs of different sizes, the situation gets more difficult [5]. For instance, with time series data or gene sequences, the order of the data points can have a substantial impact on how the data points are understood and processed. Traditional neural networks are incapable of handling this complexity. In this sense, RNNs are significant. It is a novel improvement on a traditional feedforward neural network. Sequential data that can be effectively managed includes time series, DNA sequences, and meteorological data. Because of their "memory loops," RNNs consider both the present input and the past input when determining the output. With these memory loops, which are recurrent hidden states, it can accommodate a variable length sequence of inputs. [6]. Nonetheless, RNNs are limited in their ability to look back in sequences for a very short period of time due to the vanishing gradient problem [7]. As a result, the gradient of the loss function may become either vanishing or exploding, which makes it difficult to train the network and may result in poor performance.

LSTM:

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is capable of storing information for long-term. Using this capacity, one can address the vanishing gradient problem that exists in traditional RNNs, causing them to lose sight of long-term data dependencies.
They successfully analyze sequence data because the information travels sequentially from one step to the next. To accomplish this, they employ the concept of 'cell state' or'memory cell'. This cell state transports information down the sequence chain as if it were a conveyor belt, with only minimal linear interactions. Unlike traditional RNNs, LSTMs feature four layers of neural networks instead of one, which aids in the preservation of long-term information.

A sigmoid activation function determines how much of each component is to be allowed through each of these layers. Each layer interacts uniquely, known as 'gates.'An input gate updates the current input state with new information, while an output gate determines the next hidden state. The forget gate determines what information should be thrown out or kept in the cell state. Because of the structure and functionality of these gates, LSTMs can retain critical information over longer sequences while learning which information to preserve or reject for future stages.

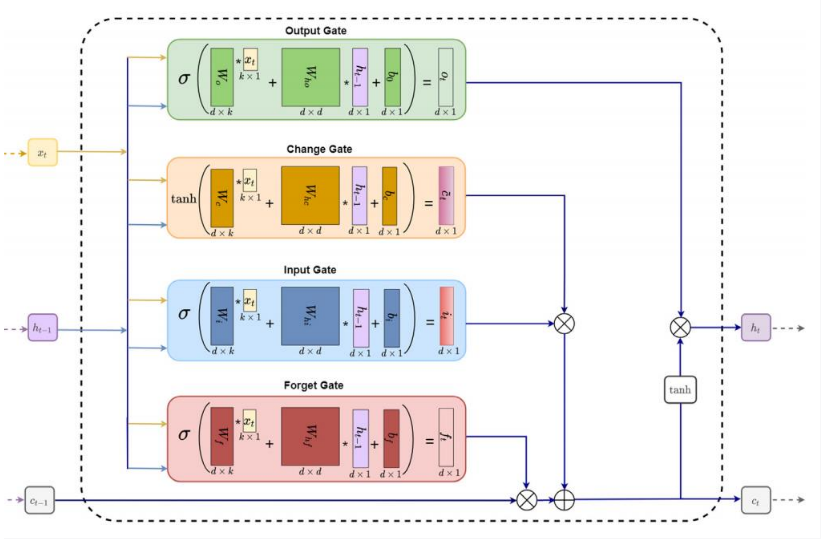


Fig-1 : Long – Short Term Memory Architecture

GRU:

Gated recurrent units (GRUs) have been designed to better capture dependencies in sequence data than conventional recurrent neural networks. GRU consists of only two gates, update and reset while LSTM has input gate, forget gate, update and output gates.

The reset gate determines how the new input is merged with the previously stored memory, and the update gate indicates how much of the previously stored memory is kept. In a GRU, the input gate and forget gate of an LSTM are combined to form the update gate. This simplification results in shorter training periods by lowering the number of tensor operations necessary during training.

In LSTMs hidden state and cell state are separated, whereas GRUs combine them. Because of this design choice, GRUs perform well while learning long-term dependencies, as they avoid the issues associated with RNNs that face vanishing or exploding gradients [8]. When training data is less, GRU will usually outperform an LSTM. When there is enough training data, LSTMs can be utilized to tackle tasks with long-term dependencies because they are more efficient at memorizing longer sequences [9].

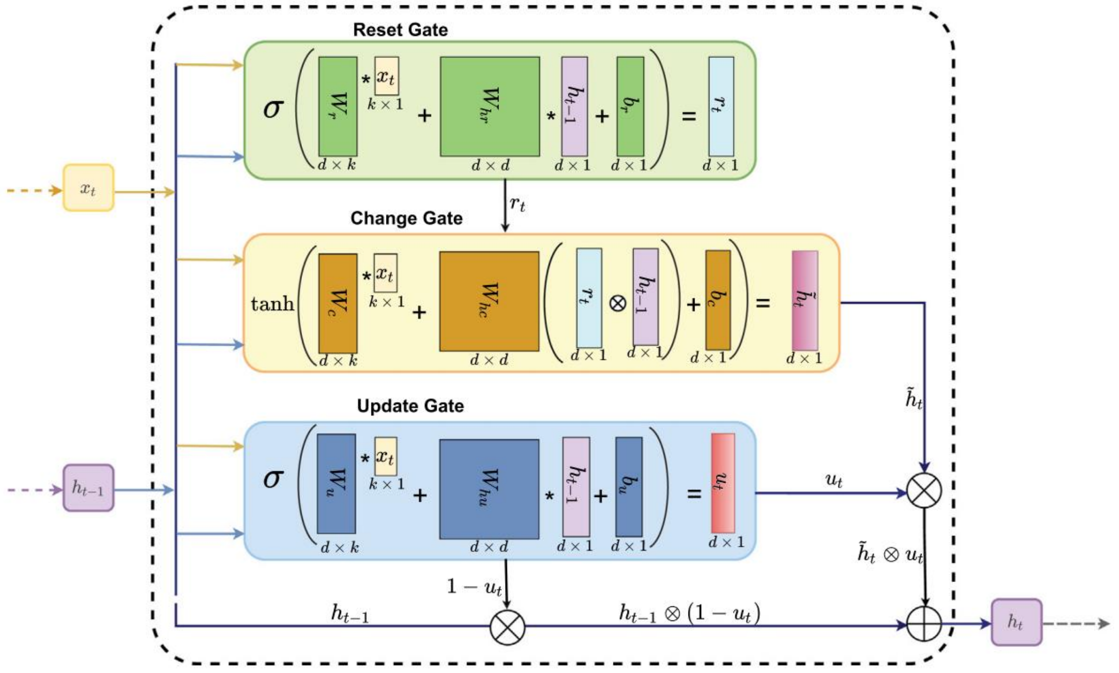


Fig-2: Gated Recurrent Unit Architecture

Ensemble model:

An ensemble model in machine learning is a means to merge various separate models into one to enhance overall performance. Ensemble modelling is essentially predicated on the notion that a collection of weak learners might become a strong one, hence improving model accuracy.Out of most commonly used ensemble methods which are bagging, boosting and stacking, where it creates different subsets for training different models. Finally, the prediction is made by taking the average of all the predictions. The different kind of models we used here are RNN.LSTM and GRU. This ensemble model efficiently leverages representations generated by each model.

FinancialBERT

FinancialBERT is a BERT (Bidirectional Encoder Representations from Transformers) model that has been specifically trained on a vast collection of financial texts. This model's primary purpose is to progress Natural Language Processing (NLP) research and applications in the financial sector.FinancialBERT has been trained on a vast collection of financial texts. This domain-specific training can help the model understand financial terminology, making it more effective at financial data-related tasks.By providing a pre-trained FinancialBERT model, these customers can use BERT for financial tasks without requiring broad resources. A dataset of Financial PhraseBank phrases labelled with their sentiment has been used to fine-tune the FinancialBERT model for sentiment analysis. The output will be a labelled as positive, negative or neutral while the score be in the range of 0 to 1.

Experimental Description:

Data Collection and visualization:

In our project Forecasting the stock market requires two sets of data. One is Apple stock price news for sentiment analysis, and the other is Apple stock price over the past three years. Apple stock price is available at ytfinances. This data includes open, close, low, high and volume columns. The opening price represents the initial price of the stock, while the closing price represents the highest price of the stock, while the closing price represents the highest price and the lowest price of the stock of the day. this day. Volume represents the number of slots sold and bought. A website called New.io which is an an open source API-based service, can be used to retrieve new articles. We can make 100 API key requests per day and for each request we get 10 outputs in json format which include Published At, Article Titles, Content, Author, Website etc. In each day's Sentiment analysis, we considered both headlines and content. We multiplied the acquired score by 1 if it was positive, 0 if it was neutral, and -1 if it was negative, and the line graph [Fig-4] of sentiment score over time is shown below. According to our data, there are 869 Positive news Articles and 165 negative Articles.

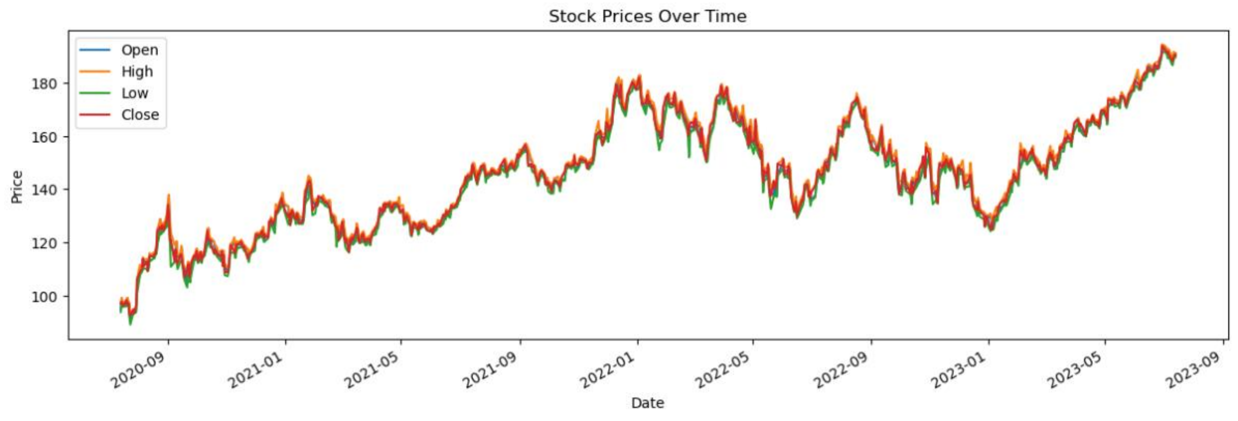


Fig-3: Apple Stock Price over Time

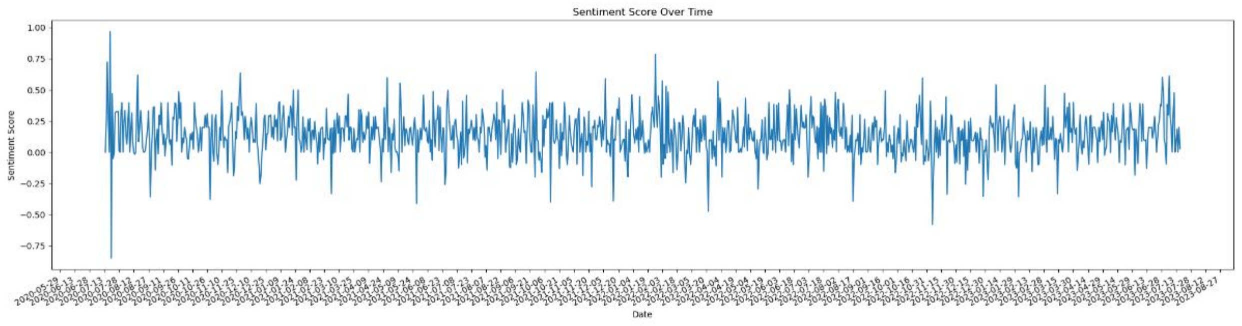


Fig-4: Sentiment Score Over Time

Experimental Setup:

a) Preprocessing:

For time series data, we first need to set the window size. The window size refers to the previous time steps that were taken into account to predict the current time step. I chose a window size of 4 for my problem, which is analogous to stating evaluate data from Monday to Thursday to anticipate Apple's share price on Fridayas shown in [Fig-5]. This experiment is broken down into three sections. To begin, we employ a benchmark model such as RNN,LSTM GRU on open stock price data to obtain a reference. In our second, we will use CNN-Stacked LSTM, CNN-Stacked GRU, and Ensemble models to create hybrid deep learning models. Finally,In third section we used the same hybrid model to predict sentiment score and open price. Finally, these models are compared in terms of train and test RMSE.In addition, I also forecast the price for the next four days and compare it to our original data. In the third segment, we utilised an inner join to connect two datasets based on dates. We did this since stock market data is only available while the market is open, whereas sentiment score is available 24 hours a day, seven days a week.As a result, the sentiment score dataset has more rows than the stock market dataset.As a consequence, we connected these tables using an inner join based on the criterion of matching dates in two datasets.We modify our input data from 2D to 3D for these sections.Our data is in the form [Samples,Fearutes], which we convert to [Samples,time-step/window-size,Feature].The number of features in sections 1 and 2 was one, whereas the number of features in section 3 was two.

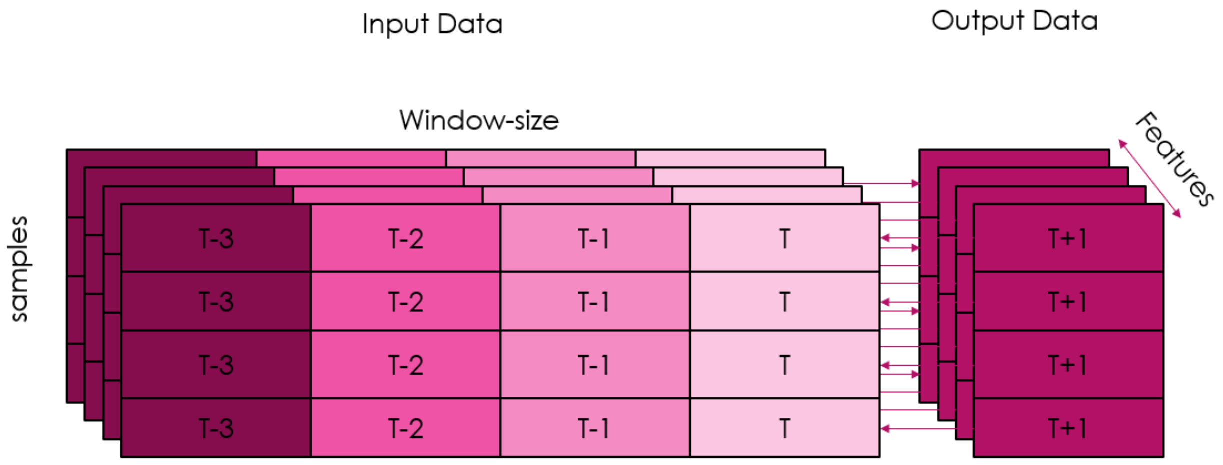


Fig-5: Data Architecture

` We standardised the data before fitting the model with the minmax scaling tool, resulting in a range of (0,1). I also divided the input data into train and test sets, with 80% representing the train set and 20% representing the test set. These proposed models were trained using the Mean squared Error loss function and the Adam optimizer. The number of epochs was set to 200, and the activation functions in the hidden layers were assigned to Relu and Tanh, respectively.

b)Benchmark Model:

The benchmark models we used were:

- RNN: Two RNN layers with 130 units and a dense layer with First layer having return sequence True .
- LSTM: An LSTM layer with 130 units and a dense layer with Adam optimizer and MSE loss.
- GRU: A GRU layer with 130 units and a dense layer with Adam optimizer and MSE loss.

c)Proposed Diagram:

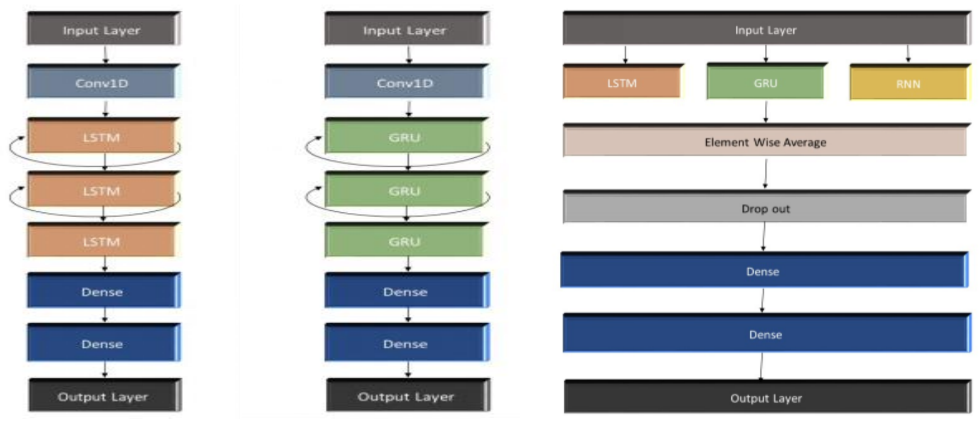


Fig-5: (a) Architecture of CNN -Stacked LSTM (b)Architecture of CNN-Stacked Gru (c)Architecture of Ensemble model

d)Hybrid Model Architecture

Table-1

CNN-Stacked LSTM	One-dimensional convolutional layer with 64 filters and Relu activation function
	LSTM layer with 256 units and Relu activation function and return sequence true
	LSTM layer with 256 units and Relu activation function and return sequence true
	LSTM layer with 128 units and Tanh activation function .
	Dense layer with 256 units and Relu activation Function.
	Dense layer for predict output.
CNN-Stacked GRU	One-dimensional convolutional layer with 64 filters and relu activation function
	GRU layer with 256 units and Relu activation function and return sequence true
	GRU layer with 128 units and Relu activation function and return sequence true
	GRU layer with 256 units and Relu activation function .
	Dense layer with 128 units and Relu activation Function.
	Dense layer for predicting output.
Ensemble Models	RNN layer with 125 units and the tanh activation function
	LSTM layer with 125 units and the relu activation function
	GRU layer with 125 units and the tanh activation function
	Average of all the hidden states from RNN, LSTM, and GRU
	Dropout layer with a rate of 0.2
	Dense layer with 32 units and the ReLU activation function
	Dense layer with a prediction of units

Table[1] clearly demonstrates the architecture and model parameters of all of the deep learning and machine learning models we used. We utilized CNN-Stacked LSTM since CNN are very effective at collecting essential patterns within a given data and LSTM,RNN,GRU are very good with sequence data, therefore we integrated both of these models. We also used the Ensemble model of RNN, GRU, and LSTM, which combines weak learners to build strong learners. The final output of an ensemble model is determined by taking the average of the outputs of each model, so that each model is given equal priority. We used two datasets, one with a combination of open price from Apple Stock price and a sentiment score, and the other with solely Apple Stock price. (window_size X feature_size) is the input size formula for the time series.

We trained in the train set for 200 iterations once the model was fitted. The value for both train and test was then predicted. We then used an inverse minimax scaler to get them in the same range as the output. Following that, we forecasted the Apple opening price for the next five days and plotted a graph of the opening price, predicted opening price, and forecasting price. These results and graphs are shown below and we got astonishing result for our novel way of combining both sentiment score and open price.

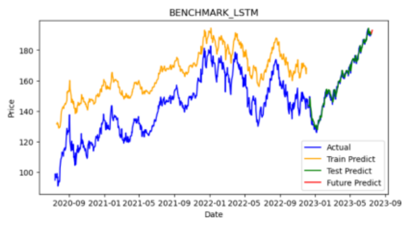
Result and Discussion:

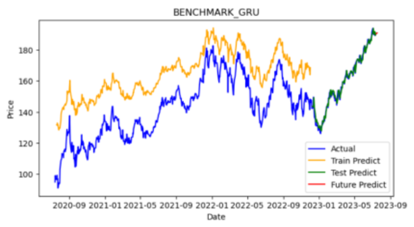
Table-2

Models	Forecast For 4-Days				RMSE(Train)	RMSE(Test)
RNN	190.84	191.44	192.06	192.58	23.27	2.23
LSTM	190.92	191.58	191.58	192.87	23.01	2.25
GRU	190.34	190.49	190.67	190.83	22.78	2.20
CNN-Stacked LSTM	192.25	194.01	195.75	197.35	23.25	2.37
CNN-Stacked GRU	189.11	188.13	187.22	186.41	22.19	2.35
Ensemble models	190.40	190.59	190.82	191.02	22.73	2.16
Senti-CNN-LSTM	189.01	188.28	187.28	186.23	2.20	1.74
Senti-CNN-GRU	**191.64**	**192.75**	**193.33**	**193.71**	2.74	2.32
Senti-Ensemble	190.37	190.42	190.34	190.25	**2.06**	**1.57**
Original Price	191. 89	193.35	193.10	195.08	0	0

Table [2] clearly shows the results of RMSE of train and test and prediction for the following 4 days for each model. Along with them, we have shown the original opening price of Apple for the next four days. We can see that Senti Ensemble has a very low RMSE for test data and test data. The training error of "Senti" models (models that incorporate sentiment analysis) is remarkably low when compared to benchmark and standard models, according to the discussion. This shows that including sentiment analysis has improved the model's capacity to capture and predict stock price patterns. Finally, when we compare the 4 day prediction to the original price, we can observe that Senti-CNN-GRU was able to provide the best outcome.

Graph and Interpretation:





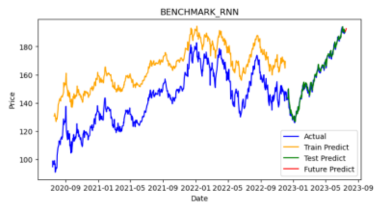


Fig:6- (a)Benchmark_LSTM (b)Benchmark_GRU (c)Benchmark_RNN

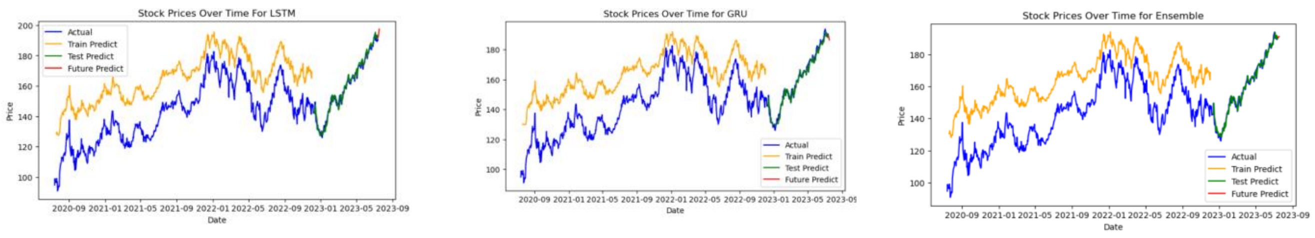


Fig:7- (a)CNN-Stacked LSTM (b) CNN-Stacked GRU (c) Ensemble models

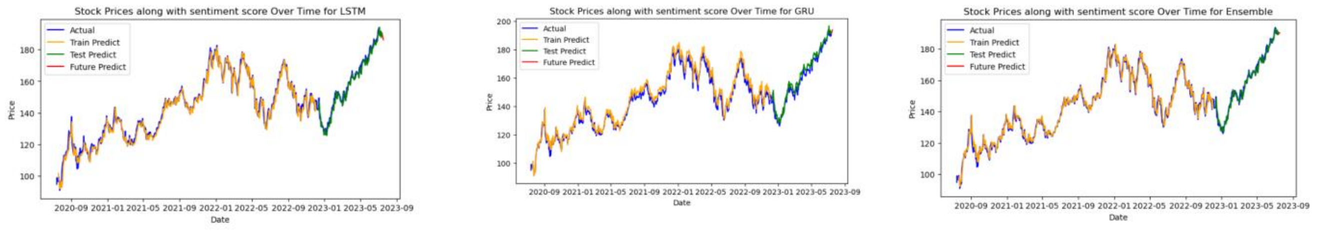


Fig:8- (a) Sentiment_LSTM (b) Sentiment_GRU (c)Sentiment_Ensemble

. Figures 6–8 show a graph of stock price over time, as well as anticipated stock price for both train and test, in yellow and green, respectively. Each model's forecast is represented by a red line. We can plainly see in Figures 6 and 7 that there is a difference between the original price and predict price for our train data set, but the test prediction fits the data flawlessly. Except forCNN-Stacked LSTM, all models indicate that the share price will decline in the next days. The results of our unique method of using both the opening price and the sentiment score are shown in Fig[8].We can see from Fig[8][a][b][c] that the out train prediction is perfectly aligned with the out share price. Senti-Ensemble model scored best when predicting test results.Senti-ensemble produced the best result wr.to test RMSE and the best result wr.to forecast. The forecast trend is negative for Senti-LSTM and Senti-Ensemble, but upward for Senti-GRU, which is consistent with the original data. This means that our new method outperformed other models in terms of overall performance.

Conclusion –

In our project, we created three deep learning models that are ideal for our time series data. We also performed NLP sentiment analysis on a new article on the Apple stock price and discovered a novel method of projecting open price by combining previous open price and sentiment score. We also created three benchmark models to compare our results, and we discovered that senti-models provided very good results when compared to all other models, with senti- Ensemble being the best model when compared to all other models. We also forecasted the price of Apple stock for the next four days and discovered that Senti-GRU is more aligned with the opening price of Apple. Overall, the trial of a novel method of determining the stock price by combining prior share and sentiment-score was successful.

Future Research-

Further research for this experiment could include developing two different models for section-3, one for predicting open price using two features (sentiment score and open price) and the other for predicting sentiment score using only one feature (sentiment score).We may also utilise temporal fusion transformers to forecast multivariant time series. The Temporal Fusion Transformers (TFT) architecture is used to increase the predictive capacity of models. TFTs are cutting-edge models that were created to deal with multivariate time series data. TFTs may outperform standard models such as CNN-stacked LSTM, CNN-stacked GRU, or ensembles in terms of forecasting accuracy since they incorporate temporal dependencies and interactions between several features. Investigating more advanced feature engineering techniques may also be advantageous. This could entail extracting more significant elements from financial data, market news mood, and external economic indicators that may affect stock performance.

References –

1. Walczak, S.; Cerpa, N. Artificial Neural Networks. In Encyclopedia of Physical Science and Technology, 3rd ed.; Academic Press: New York, NY, USA, 2003; pp. 631–645.
2. 2. Lecun, Y.; Bengio, Y. Convolutional Networks for Images, Speech, and Time-Series; MIT Press: Cambridge, MA, USA, 1997
3. Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings of the 27th International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010; pp. 807–814.
4. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J. Mach. Learn. Res. 2014, 15, 1929–1958.
5. Bao, W.; Yue, J.; Rao, Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE 2017, 12, e0180944
6. Rumelhart, D.E.; McClelland, J.L. Learning Internal Representations by Error Propagation. In Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations; MIT Press: Cambridge, MA, USA, 1987; pp. 318–362.
7. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. Neural Comput. 1997, 9, 1735–1780.
8. Shen, G.; Tan, Q.; Zhang, H.; Zeng, P.; Xu, J. Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions. Procedia Comput. Sci. 2018, 131, 895–903.
9. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In Proceedings of the NIPS 2014 Workshop on Deep Learning, Montreal, QC, Canada, 13 December 2014.
10. Wang, J., Zhang, S., Xiao, Y., & Song, R.. (2022, January 1). A Review on Graph Neural Network Methods in Financial Applications.
11. Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market.