Stock Closing Price Prediction Using Deep Learning Models

Akshada Joshi (joshiak) | Anitha Ganapathy (aganapa)| Archana Krishnamurthy (akrishn)
Deep Learning Systems, Final Project Fall 2021

ABSTRACT: Using time-series data analysis for stock-price forecasting (SPF) is complex and challenging because many factors can influence stock prices (e.g., inflation, seasonality, economic policy, societal behaviors). Such factors can be analyzed over time for SPF. Machine learning and deep learning have been shown to obtain better forecasts of stock prices than traditional approaches. This study, therefore, proposes a method to enhance the performance of an SPF system based on deep learning approaches.

1. INTRODUCTION

–Stock prices are affected by many factors, such as inflation, seasonality, economic policy, company performance, economic shocks, and political shocks. Such factors can decrease the accuracy of any forecasting system. Nevertheless, accurate SPF can bring benefits to companies, shareholders, and investors; it can also be used as a key measurement for assessing economic performance. Designing an accurate SPF system requires considering fundamental issues such as feature selection, model fitting, and prediction. Traditionally, Long Short-Term Memory (LSTM), autoregressive integrated moving average (ARIMA) are used for time series forecasting. In this project, we have also used Generative Adversarial Network (GAN) , GRU and hybrid ARIMA plus LSTM as additional deep learning models to improve the prediction. We have selected Apple Inc as the company to analyze the data from. The objective of the project is to compare the results from these deep learning models when trained on different datasets and predict the closing price for train and test datasets.

2. DATASETS AND MODELS

A. *Financial data Analysis* - This is the historical financial Apple data from Yahoo Finance consisting of 6 features, with the target feature being Closing price. The time period is from 24 Dec 2016 to 18 Nov 2021.

B. *Technical Indicators* - There are 8 technical indicators extracted from the closing price by performing moving average, calculating Bollinger bands and log momentum.

C. *Fast Fourier Transformation* - There are 6 features generated after performing fast Fourier transformation on the closing price. This has been done to extract the trend of the price, and minimize the fluctuations in the closing price. 3,6 and 9 components have been used.

D. *Sentimental Analysis* - This has been done to understand the overall investors' sentiment to buy/sell stocks. Comments from Reddit have been scrapped from 1 July 2017 to 1 Dec 2021. Daily bullish and bearish scores have been calculated by dividing the bullish/bearish comments per day by the total comments for that day.

E. *Google Trend* - Google trend data has been extracted from APIFY; an online API built for this purpose. This data consists of the normalized number of times the keyword AAPL has been searched worldwide. The data is collected from 24 Dec 2016 to 18 Nov 2021.

F. *Covid Data* - The data repository at Johns Hopkins has been used as the source to extract COVID data. The increase in the number of cases and increase in the number of deaths are the two features generated from this dataset, as they did have a strong impact on the stock market. The COVID data has been collected from 22 Jan 2020 to 3 Dec 2021.

G. *Final Dataset* : After concatenating all the datasets given above based on the date and dropping all NaNs, we came up with a final DataFrame, consisting of 1080 rows from 3 July 2017 to 1 Nov 2021 and 29 features.

H. *Data Preprocessing*: The final DataFrame is normalized using MinMaxScaler() to bring the data in [0,1] range. The normalized data has been split into train and test (validation) datasets such that 70% of the data lies in train and the rest 30% lies in test. The train and test 'X' dataset consists of the data on 3 consecutive days, and the train and test 'y' dataset consists of the closing price on the fourth day - which the models will predict. The closing price prediction for the test data is from 8 July 2020 to 1 Nov 2021.

I. *Deep learning models*: In order to successfully predict the closing price on the 4th day, the following deep learning models have been used and compared : LSTM, GRU, GAN and hybrid LSTM + ARIMA.

3. STOCK PRICE PREDICTION MODEL ARCHITECTURE

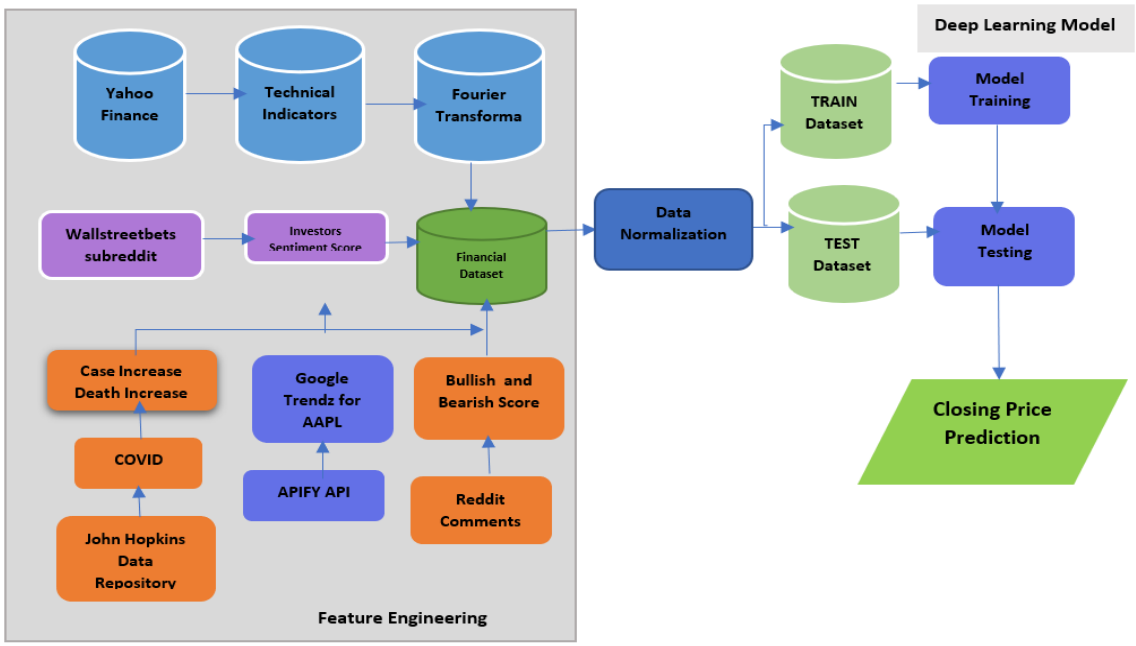

Fig : 1 Stock Price Prediction Architecture for different data sources and deep learning models.

4. DEEP LEARNING MODELS

A. LSTM – The Long Short-Term Memory (LSTM) based on "memory line" has proved to be very useful in forecasting cases with time series data. In an LSTM model, the memorization of earlier stages is performed through gates. We have used a simple model architecture consisting of an LSTM layer and 2 dense layers. The output of the last dense layer will be the fourth day closing price prediction.

Hyperparameter tuning and early stopping : Keras tuner was used for hyperparameter tuning for the learning rate and early stopping callback was used with a patience of 10. The best learning rate found is 0.0001 and the best number of epochs is 147. A batch size of 20 was used for the train and validation data. The first plot given below shows the validation and train loss over the epochs and the second plot shows the trend of the real and predicted closing price for the test dataset.

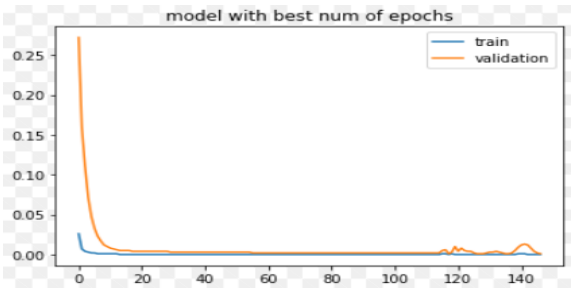

Fig 2 : Train / Validation Loss

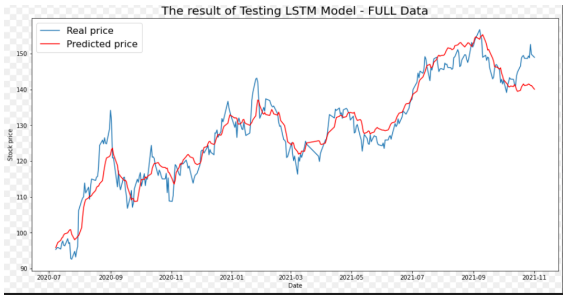

Fig 3 : LSTM Prediction / Closing price(Test data)

B. GRU – An alternative solution to RNN is called the gated recurrent units network (GRU), which is a modification to the LSTM model. It is a powerful model for predicting time series data because it implements most of the gates used in LSTM with less parameters to estimate. So, it is relatively easier to optimize. For this project, 2 GRU units with 2 dense units have been used. Manual hyperparameter tuning has been done. The best learning rate found is 0.0001 and the best number of epochs is 50, with a batch size of 20 for validation and train set. The first plot given below shows the validation and train loss for 50 epochs and the second plot shows the real and predicted closing price trend for the test data.

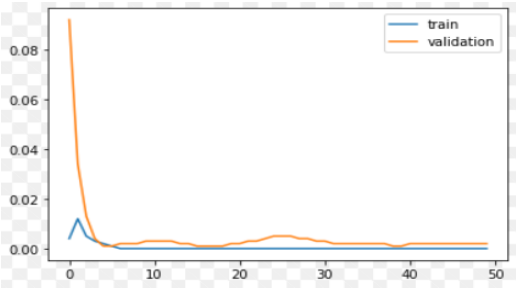

Fig 4 : Train / Validation Loss (GRU model)

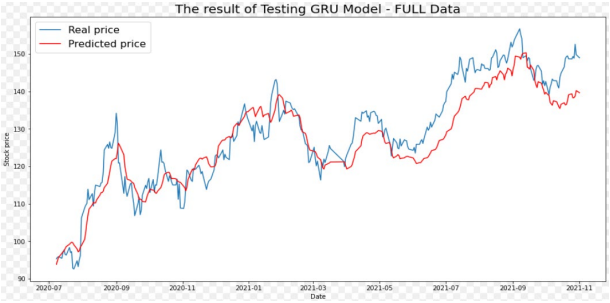

Fig 5 : GRU Prediction / Closing price(Test data)

C. GAN – GANs are a clever way of training a generative model by framing the problem as an unsupervised learning problem with two sub-models: the generator model that we train to generate new examples, and the discriminator model that tries to classify examples as either real (from the domain) or fake (generated). The two models are trained together in a zero-sum game, adversarial, until the discriminator model is fooled about half the time, meaning the generator model is generating plausible examples. The loss functions for the generator and the discriminator are given below:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[log\ log\ D\left(x^{(i)}\right) + log\ log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right] \quad \rightarrow \text{Discriminator loss}$$

$$\nabla \theta_g \frac{1}{m} \sum_{i=1}^{m} lo\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \quad \rightarrow \text{Generator Loss}$$

where z is the input data for generator, x is the target for the real dataset, G(z) is the generated data from the generator

The flow chart given below describes the working of generator and discriminator for our dataset.

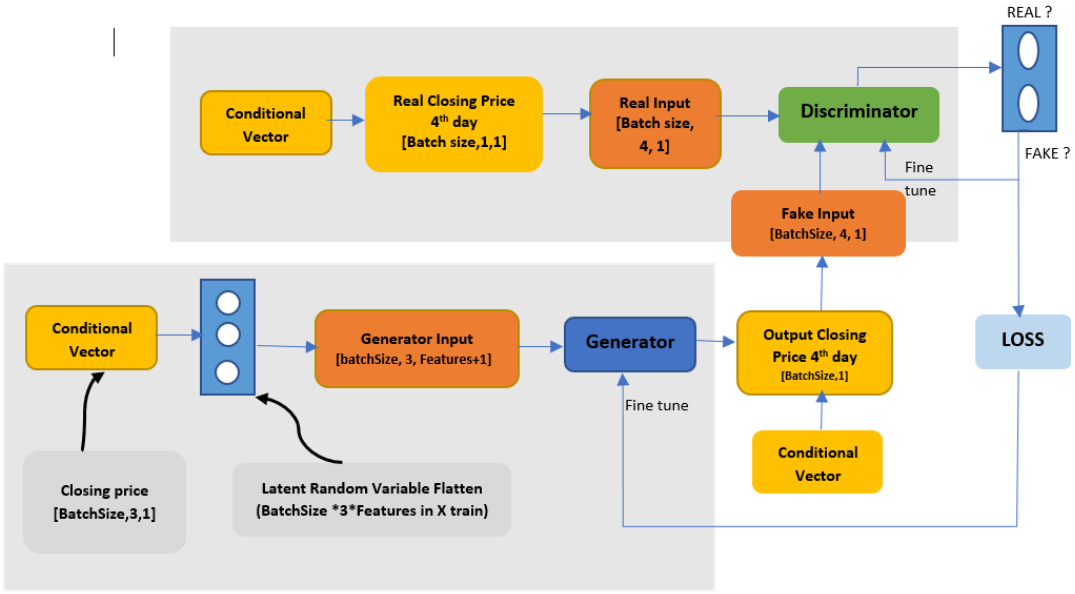

Fig 6 : GAN Stock Price Prediction model Description (Training)

- **GAN GENERATOR -** 3 GRU layers and 3 dense layers have been used for the generator. The input to the generator is a latent vector randomly sampled from a normal distribution. The conditional vector consists of the closing price of the past three days from (4,5,6 July 2017) to (4,5,6 July 2020). This vector is appended to the latent vector which forms the generator input. This forms a 3d array of size (batch size X 3 X Number of features + 1). The generator output is the closing price on the fourth day for the previous three days in the input. The shape of the output is (batch size X 1 X 1). As the generator is trying to fool the discriminator to classify the fake closing price as real, a cross entropy loss function between 1s and generator output has been used.

- **GAN DISCRIMINATOR -** 3 CNN layers and 3 dense layers have been used. The last dense layer uses a sigmoid activation function. The input to the discriminator can be divided into two categories. The first category consists of the fake closing price which is the output from the generator. The second category consists of the real closing price.

  The discriminator will try to make a decision and classify the real examples into positive class and fake examples into negative class. The discriminator loss is defined as the summation of the cross entropy loss from fake examples and cross entropy loss from real examples.

  The model is trained for 250 epochs with a learning rate of 0.0001 for both the discriminator and generator. The entire train dataset has been used as one batch. The first plot shows the variation of the generator loss, and the fake and real loss for the discriminator. The second plot shows the predicted and real closing price for the test dataset from the GAN model.

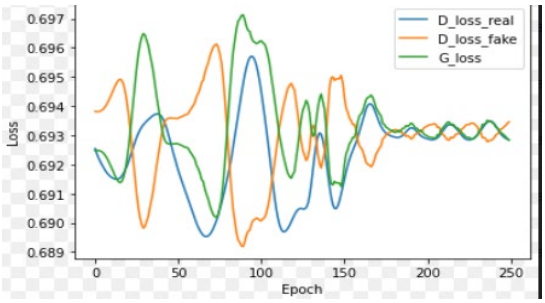

Fig 7 : Train / Validation Loss (GAN model)

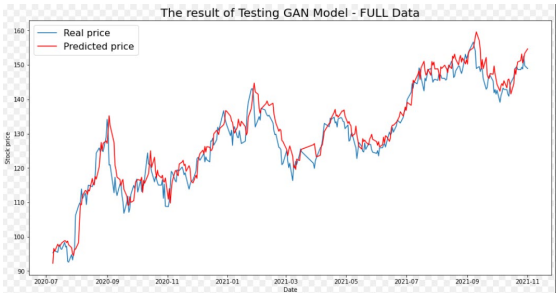

Fig 8 : GAN Prediction / Closing price(Test data)

D. HYBRID LSTM ARIMA – The hybrid ARIMA-LSTM model is an application of "Stock Price Prediction Based on ARIMA-RNN Combined Model[13]. This section of the project follows a similar pattern with additional experimentation on the moving averages fed into the ARIMA model & Neural Network architecture. This model proposes to overcome the challenge with the volatility of the stock data, and the problem in regards to overfitting from a neural network. Multiple experiments with various moving averages such as EMA (Exponential Moving Average), TEMA(Triple Exponential Moving Average), TRIMA (Triangular Moving Average) , KAMA (Kaufman Adaptive Moving Average), MIDPOINT(MidPoint over period) to name a few were introduced to improve upon this model and additional architectures such as bidirectional networks, RNN-LSTM were implemented. In addition, exogenous variables were introduced in the ARIMA model to generate a multivariate ARIMA forecast along with a multistep multivariate RNN model with a lookback of 3 days and 1 day forecast. The combinations of features introduced included data from google trends, covid death rate details & sentiments scores.

The moving averages are introduced for a smoothened distribution over time and were determined by the Kurtosis for a sample of 100 days. .The data was split between a high volatility and low volatility time series based on the moving average determined.

Below is the equation that sums up the final results from the hybrid ARIMA LSTM model.

$$\widehat{y_t} = \widehat{I_t} + \widehat{g_t} \quad \rightarrow \text{sum of predicted price and forecasted residual}$$

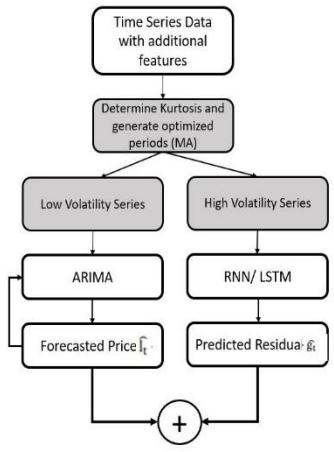

Fig 9 : Hybrid LSTM ARIMA Model

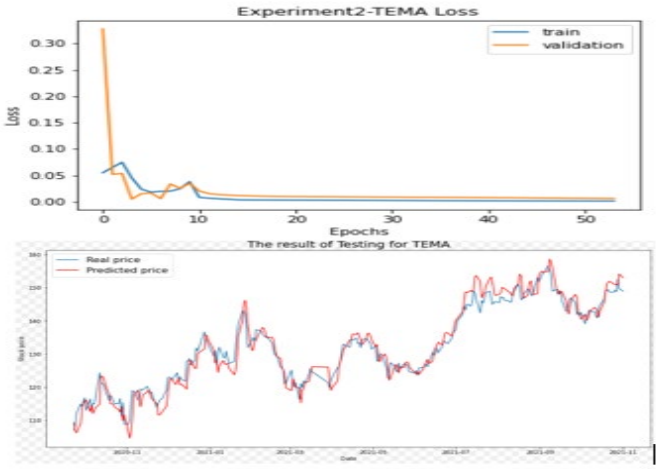

Fig 10 : Hybrid LSTM ARIMA Loss plot and Prediction / Closing price(Test data)

5. RESULTS

The following table shows the results from the key experiments which were performed on the datasets. The datasets have been split into 4 categories : full data, financial data + google trends, financial data + sentiment scores, financial data + COVID data.

Model	Dataset	Loss Metric	Full dataset	Dataset_with_Google_Trends	Dataset_with_Bull_Bear_score	Dataset_with_covid data
GRU	Test	Test MSE	29.2854	35.6801	67.3809	16.8599
		Test RMSE	5.4116	5.9733	8.2086	4.1061
		Test MAE	4.4592	5.0158	6.9443	3.1941
LSTM	Test	Test MSE	14.7534	31.4255	81.3366	67.2085
		Test RMSE	3.8410	5.6058	9.0187	8.1981
		Test MAE	3.0404	4.6626	7.8817	7.1253
GAN	Test	Test MSE	14.0359	10.3160	10.3871	11.0642
		Test RMSE	3.7465	3.2119	3.2229	3.3263
		Test MAE	2.8773	2.4255	2.4352	2.4860
HYBRID ARIMA LSTM - TEMA	Test	Test MSE	9.5644	32.0201	26.6083	72.3302
		Test RMSE	3.0926	5.6586	5.1583	8.5047
		Test MAE	2.4489	4.8400	4.3368	7.4137
HYBRID ARIMA LSTM - MIDPOINT	Test	Test MSE	19.8340	16.3987	18.2442	17.3813
		Test RMSE	4.4535	4.0495	4.2713	4.1691
		Test MAE	3.5744	3.2996	3.3888	3.3993

Fig 11 : Experiments performed with different DL models, datasets and metrics.

6. CONCLUSION -

In this paper, we proposed four deep learning models to predict the stock market and MAE, MSE and RMSE are the loss metrics used to evaluate the models. Along with raw financial data features, we have introduced new features based on sentiment analysis, google trends, COVID data, and Fourier transformation. We trained the models using different datasets with varying features. Based on the experimental results, some conclusions can be drawn. As the dataset changes, there is a considerable change in the error produced by each model (The highest being 60% increase in RMSE), which shows that each feature category plays a key role in the stock market prediction. For the full dataset, Hybrid ARIMA with TEMA moving average has the lowest RMSE of 3.09. For the rest of the datasets, GAN is the model with the least RMSE. The key conclusion which can be drawn is that nontraditional models namely hybrid LSTM and GAN are an improvement over traditional models like GRU and LSTM which are generally used for stock market prediction.

Future research should be focused on the development of sentiment analysis scores from other sources like business insider, Bloomberg BusinessWeek, TechCrunch etc. An improvement in the GAN model i.e. WGAN-GP model can be implemented to help stabilize and improve the basic GAN training, and generate even better results.

7. REFERENCES

[1] G. Zhang, Time series forecasting using a hybrid ARIMA and neural network model, Neuro computing, 2003, 50 (0):159-175

[2] C. Narendra Babu, B. Eswara Reddy, A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data[J]. Applied Soft Computing, 2014, 27-38.

[3] D;, Xu D;Zhang Q;Ding Y;Zhang. "Application of a Hybrid Arima-LSTM Model Based on the Spei for Drought Forecasting." *Environmental Science and Pollution Research International*, U.S. National Library of Medicine, https://pubmed.ncbi.nlm.nih.gov/34403057/.

[4] Gao, Zihao. "Stock Price Prediction with Arima and Deep Learning Models." 2021 IEEE 6th International Conference on Big Data Analytics (ICBDA), 2021, https://doi.org/10.1109/icbda51983.2021.9403037.

[5] Jin, Zhigang, et al. "Stock Closing Price Prediction Based on Sentiment Analysis and LSTM." Neural Computing and Applications, vol. 32, no. 13, 2019, pp. 9713–9729., https://doi.org/10.1007/s00521-019-04504-2.

[6] "Introduction to the Keras Tuner : Tensorflow Core." TensorFlow, https://www.tensorflow.org/tutorials/keras/keras_tuner.

[7] Brownlee, Jason. "How to Develop Multivariate Multi-Step Time Series Forecasting Models for Air Pollution." Machine Learning Mastery, 27 Aug. 2020, https://machinelearningmastery.com/how-to-develop-machine-learning-models-for-multivariate-multi-step-air-pollution-time-series-forecasting/.

[8] Owid. "Covid-19-Data/Public/Data at Master · Owid/Covid-19-DATA." GitHub, https://github.com/owid/covid-19-data/tree/master/public/data/.

[9] Hasan, Khan Saad Bin. "Stock Prediction Using Twitter." Medium, Towards Data Science, 3 Jan. 2019, https://towardsdatascience.com/stock-prediction-using-twitter-e432b35e14bd.

[10] Kala, Shagun. "Stock Market Prediction Using News Sentiments." Medium, Medium, 16 Aug. 2020, https://medium.com/@kala.shagun/stock-market-prediction-using-news-sentiments-f9101e5ee1f4.

[11] Maruya-Li, Keaton. "Stock Price Prediction Using Sentiment Analysis and Historical Stock Data." Medium, The Startup, 3 Nov. 2020, https://medium.com/swlh/stock-price-prediction-using-sentiment-analysis-and-historical-stock-data-587488db8576.

[12] Alazba, Amal, et al. "Saudi Stock Market Sentiment Analysis Using Twitter Data." Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 2020, https://doi.org/10.5220/0010026100360047

[13] YU, Shui-Ling, and Zhe Li. "Stock Price Prediction Based on Arima-RNN Combined Model." DEStech Transactions on Social Science, Education and Human Science, no. icss, 2018, https://doi.org/10.12783/dtssehs/icss2017/19384.