PROJECT REPORT

PREDICTING STOCK MARKET MOVEMENTS USING SENTIMENT ANALYSIS OF TWITTER DATA

ISE 244 – AI TOOLS AND PRACTICES FOR SYSTEMS ENGINEERING
Dr. Shilpa Gupta

Project Report by:

Name: Sai Teja Kandukuri
Student ID: 016709732

SJSU SAN JOSÉ STATE UNIVERSITY

TABLE OF CONTENTS

1. PROBLEM DEFINITION

Stock markets play a crucial role in the economy. They provide a platform for companies to raise money by selling their stocks and individuals can invest in these stocks to claim ownership of the assets and grow financially. Stock markets are accessible in almost every country today. Many studies have been conducted to understand the trends in stock market and predict stock prices accurately. Stock market prediction is important for various reasons. It can help investors and financial institutions make informed decisions and maximize their profits. However, predicting stock prices accurately is a difficult task because stocks time series behaves very randomly. With the growth of social media platforms like Twitter, there is a vast amount of data available. On average, Twitter users generate 140 million tweets every day about various topics including stocks and companies. This makes Twitter a rich source of real-time information. The rate of investment and business opportunities in the stock market can increase significantly with an effective model to predict stock market movements. The main goal of the project is to develop an efficient predictive model by combining the power of sentiment analysis and machine learning to forecast stock prices accurately.

2. PROJECT OBJECTIVE

The main objective of the project is to develop a predictive model that leverages sentiment analysis of Twitter data to predict stock market prices. By extracting and analyzing the sentiments expressed in tweets related to specific stocks, the project aims to build algorithms that can identify patterns and correlations between sentiment and stock prices to improve prediction accuracy. Companies usually hire investment professionals to analyze stocks data and provide insights and guidance on financial decisions. With the help of machine learning, large amounts of historical stocks data can be studied and analyzed to predict stock movements with greater precision within limited financial commitment and time. In today's digital era, social media has become easily accessible to anyone with internet access, providing a platform for anyone to express their opinions. When it comes to news about a particular company, these collective opinions expressed on social media can change the company’s reputation. Consequently, the positive or negative sentiments expressed on social media platforms regarding a company can potentially affect their stock prices. The motivation behind the idea to use Twitter data is that it can provide access to vast amount of real-time information from public. This project involves stocks data collection using Yahoo Finance API, Twitter data collection using SNScrape that uses Twitter API and web scraping, exploratory data analysis, sentiment analysis to extract sentiments in tweets, different classic machine learning and neural network-based algorithms to predict stock prices and performance evaluation of the models. Ultimately, the main objective is to build an accurate and efficient model that can predict stock prices.

3. LITERATURE SURVEY

While a lot of studies in the past investigated stock market predictions and sentiment analysis separately, this work aims to investigate the correlation between the two. Brian et. al proposed a framework to analyze the stock price changes based on the public sentiment of investors using the Pearson correlation coefficient. However, the novel framework from this paper to extract sentiments from social media data and predict stock market movements has outperformed the previous works. The core contribution of the work is developing a Sentiment analyzer and Correlation analyzer to investigate the strength of correlation. The paper analyzed the sentiments using two different textual representations: Word2Vec and N-gram and algorithms such as logistic regression, random forest, SMO, and libSVM. The work also focuses on Twitter data that represents real-time and large-scale social media data for the prediction of stock prices. Urvik et al. improved the work by predicting future stock prices using live-fetched data for the past 60 days. The selected paper was able to show the correlation and predict the prices with good accuracy but predicting future prices is more helpful in financial decision-making, stock trading, etc. Niveditha et al. discussed the limitations of using Word2Vec representations as it is not suitable to optimize for specific tasks. Furthermore, the study can be extended by using news data with Twitter data to improve prediction accuracy as it can provide an exhaustive public opinion collection.

4. DATA

4.1. DATA COLLECTION

This project uses stocks data over time to understand the market movements and tweets to analyze or extract sentiments from the tweets. I collected stocks data and Twitter data in the time frame of 1st March, 2022 to 1st May, 2023. I collected stocks data for Tesla, Inc. and BitCoin – USD stocks using Yahoo Finance API. This provides details about the stock such as open price, close price, adjusted close price, highest price, and lowest price for a given day. I collected Twitter data using SNScrape tool that uses Twitter API and webscrape to extract tweet information about the given stock i.e Tesla or BitCoin. Only English tweets with at least 50 likes are collected to ensure that the tweets are relevant and not repeated many times. In the given time frame of 14 months, I fetched around forty tweets per day with the details such as date and time, tweet ID, tweet text, and username for a given tweet. Ultimately, I collected over 17,000 tweets from 426 days for each stock. After preprocessing, tweets data and stock price data for each stock are combined into a single dataset for further analysis.

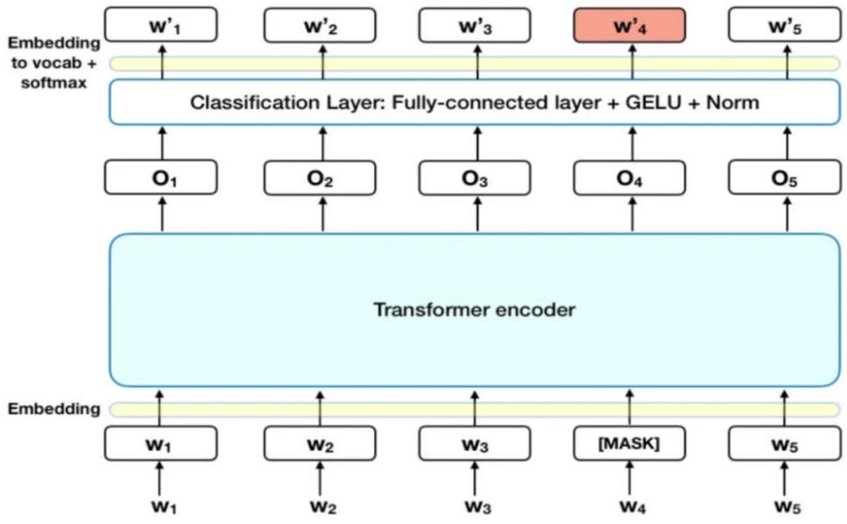
4.2. DATA PREPROCESSING

After Twitter data collection, text data from tweets is processed thoroughly so it is easier for models to extract meaningful information about sentiments. Initially, all the hyperlinks, irrelevant non English words, and stop words are removed from the tweets. Then, at the character level, special characters including hashtags (#) and asperand (@) are removed. To tokenize tweets text into a list of words, NLTK's word_tokenize function is used and then translate function is used to perform character level translation or deletion of irrelevant characters including punctuations. WordNetLemmatizer is used to lemmatize each word/token. Lemmatization is a text normalization technique used to switch any kind of a word to its base root mode. Lemmatization is responsible for grouping different inflected forms of words into the root form, having the same meaning. For stocks data, the date variable is changed to date format from datetime format as analyzing and prediction is done on a day basis. The stocks data is arranged with respect to dates and finally, the processed tweets text data is appended with stocks data on the basis of dates. After data preprocessing, there were around 12,000 entries for Tesla stock and over 16,000 entries for BitCoin-USD stock.

5. SENTIMENT ANALYSIS

Sentiment analysis is a powerful technique that involves the extraction and interpretation of emotions, attitudes, and opinions from textual data. It helps in understanding the sentiment expressed in various forms, such as social media content, customer reviews, or news articles. With the help of natural language processing and machine learning algorithms, sentiment analysis is used to uncover valuable insights that allows organizations to evaluate public sentiment and customer satisfaction to make data-driven decisions.

For this project, I performed sentiment analysis using pre-trained twitter-XLM-roBERTa-base model which is a transformer encoder-decoder architecture. It is a multilingual XLM-roBERTa-base model that is trained on approximately 198 million tweets and finetuned for sentiment analysis. The sentiment fine-tuning was done on 8 languages including English, Spanish and Hindi. RoBERTa is a state-of-the-art transformer-based language model introduced by Facebook AI Research. It builds upon BERT (Figure 1) and incorporates additional pretraining techniques and larger-scale training data, resulting in improved performance across various natural language processing tasks, including sentiment classification.



w'₁
w'₂
w'₃
w'₄
w'₅
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O₁
O₂
O₃
O₄
O₅
Transformer encoder
Embedding
w₁
w₂
w₃
[MASK]
w₅
w₁
w₂
w₃
w₄
w₅

Figure 1: BERT architecture

I used auto tokenizer, auto config and auto model for sequence classification that automatically selects appropriate options for the model. For a given text, it gives sentiment score between 0 to 1 for each of the three sentiments: positive, negative and neutral using softmax. Finally, the highest scored sentiment is considered as sentiment label for the text. Polarity of 1, 0, and -1 are assigned to positive, neutral and negative sentiments respectively. For a given day, the mean polarity is calculated based on sum of polarities and total tweets extracted on that day. Finally, polarity scores are appended with merged stocks and tweets data to perform stock price prediction.

6. DATA STATISTICS AND VISUALIZATION

The data statistics are dispayed in the table below to analyze minimum, mean, standard deviation and maximum values for different features (open, adj close, p_mean and #tweets) in the final data.

	Tesla (TSLA)				**BitCoin USD (BTC-USD)**			
	open	*Adj close*	*P_mean*	*#tweets*	*open*	*Adj close*	*P_mean*	*#tweets*
count	293	293	293	293	426	426	426	426
min	103	108	-0.5	34	15782	15787	-0.46	34
mean	232.3	231.8	-0.1	39.18	25386	25354	-0.14	40
std	61.43	61.12	0.144	1.64	8118	8075	0.11	1.17
max	378.8	381.8	0.3	41	47456	47465	0.125	41

Table 1: Overview of the data

- Open: open price for a stock on a given day
- Adj close: adjusted closing price for a stock on a given day
- P_mean: mean polarity for the tweets collected on a given day
- #tweets: number of means collected (in a given day).

Visualizations are performed for the final data with stocks information, tweets data along with polarity/sentiment scores obtained from sentiment analysis.

The first set of visualizations in this section depict time series chart for stocks, pie chart for sentiment labels and distribution of mean polarity of tweets for Tesla stocks and tweets data.

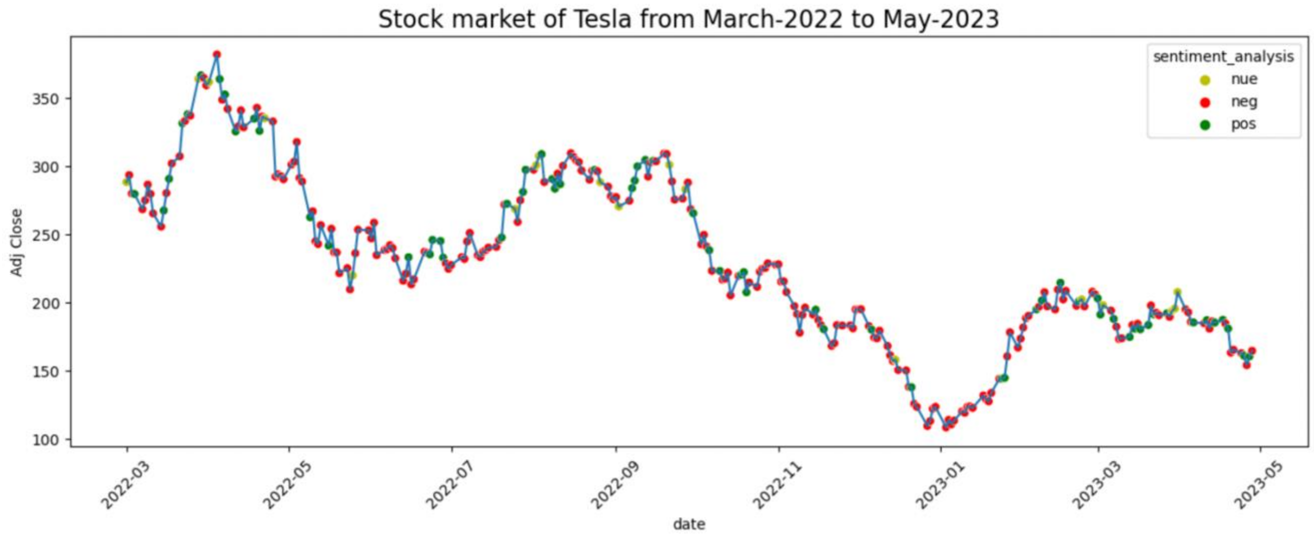


Figure 2: Time series chart for Tesla Stock market movements.

From the above figure, it is observed that the stock prices for Tesla has gone down over time with the lowest price during January 2023 and highest price around April 2022. In the recent months i.e after January 2023, there seems to be little growth in stock prices.

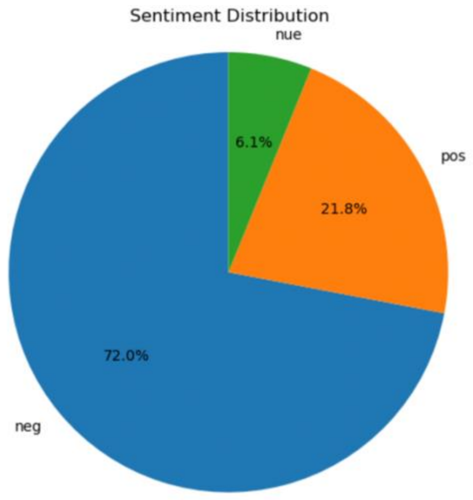


Figure 3: Pie chart of Sentiment distribution for Tesla tweets data

From the pie chart, it is observed that majority of tweets are of negative sentiment. The number of negative tweets are around 3.5 times the number of positive tweets and there are only a few neutral tweets.

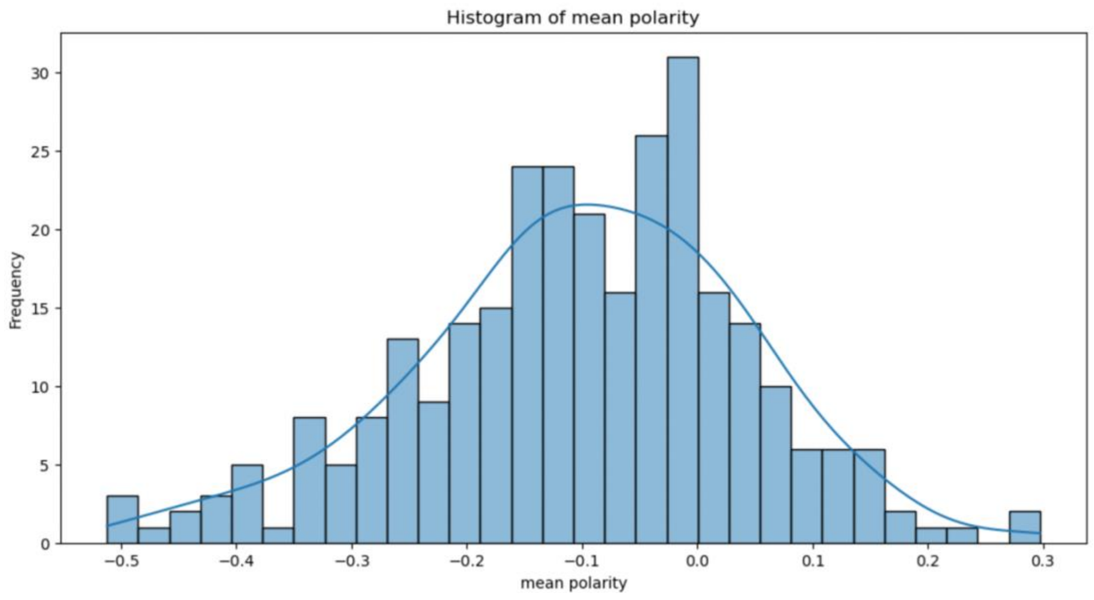


Figure 4: Distribution of mean polarity

From the histogram, it is observed that majority of tweets have almost equal distribution of positive and negative mean polarity. Most of the tweets have polarity near to -0.1.

The second set of visualizations in this section depict time series chart for stocks, pie chart for sentiment labels and distribution of mean polarity of tweets for BitCoin stocks and tweets data.

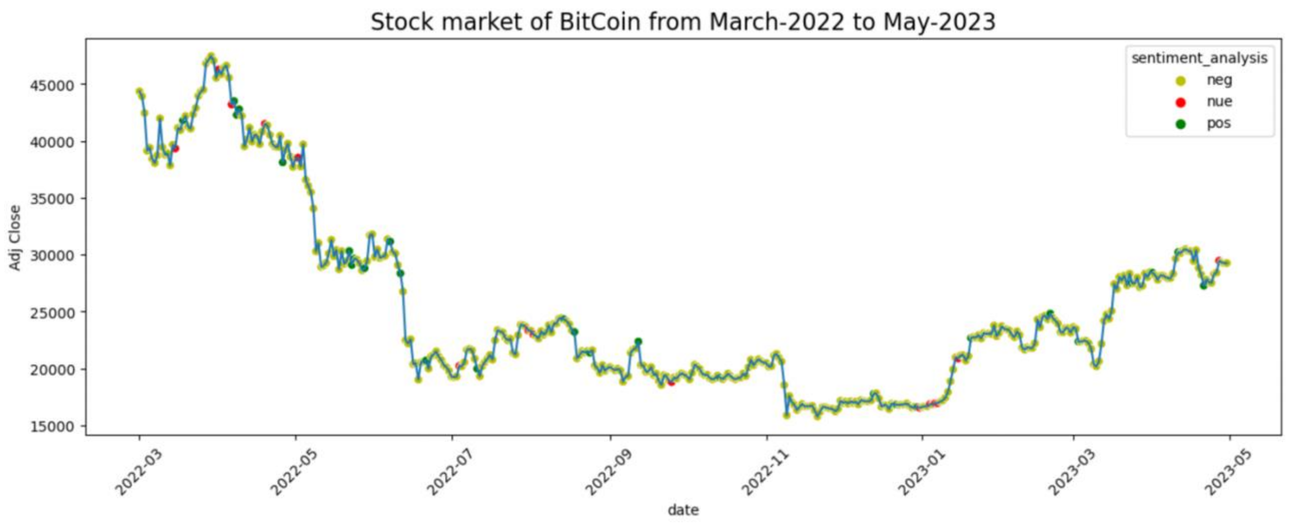


Figure 5: Time series chart for BitCoin-USD Stock market movements.

From the above figure, it is observed that the stock prices for BitCoin has gone down over time and then started to rise up with the lowest price during November 2022 and highest price around April 2022. In the recent months i.e after January 2023, there seems to be little growth in stock prices.

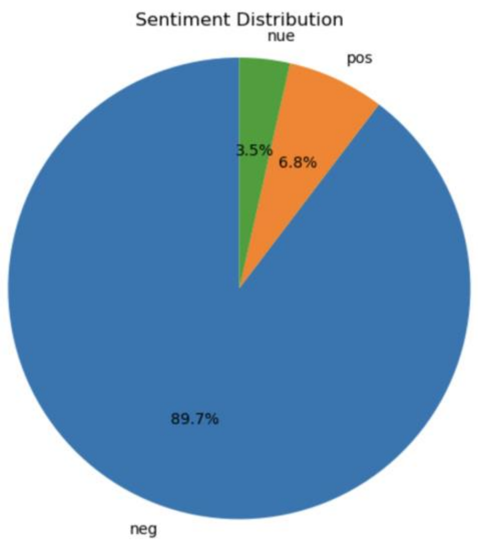


Figure 6: Pie chart of Sentiment distribution for BitCoin tweets data

From the pie chart, it is observed that majority of tweets are of negative sentiment. The number of negative tweets are over 14 times the number of positive tweets and there are only a few neutral tweets.

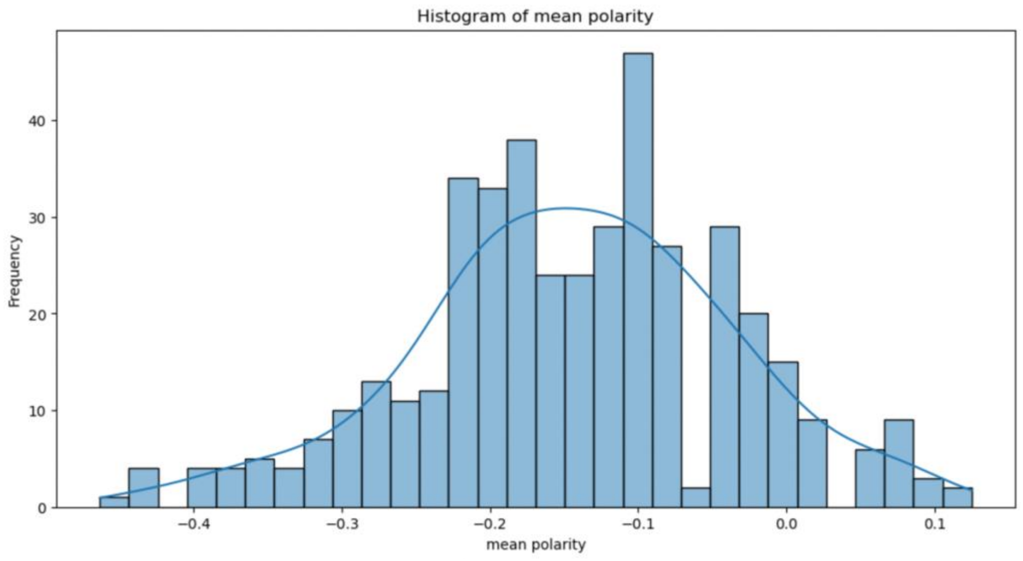


Figure 7: Distribution of mean polarity

From the histogram, it is observed that majority of tweets have almost equal distribution of positive and negative mean polarity.

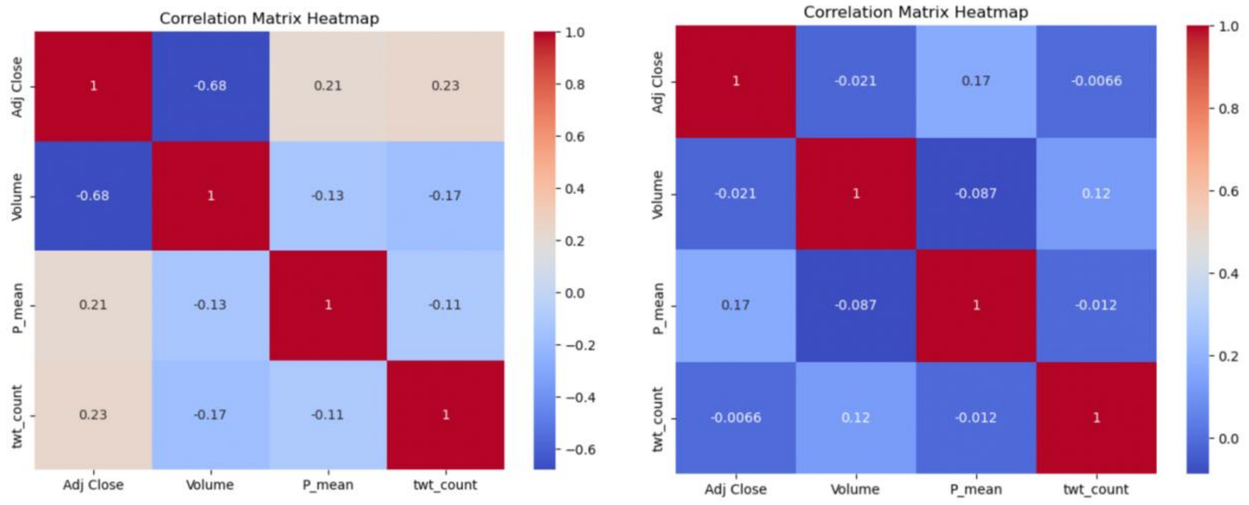


Figure 8: Correlation heatmaps for Tesla and BitCoin stocks

Finally, from the heatmaps, we can see that there is 0.21 correlation and 0.17 between mean polarity and adjusted close prices for Tesla and BitCoin stocks, respectively.

7. METHODOLOGY

I have implemented three algorithms – random forest regressor, TabNet regressor and a CNN-LSTM based neural network to predict stock prices. Details about each model are explained in detailed in the following subsections.

7.1. RANDOM FOREST REGRESSOR

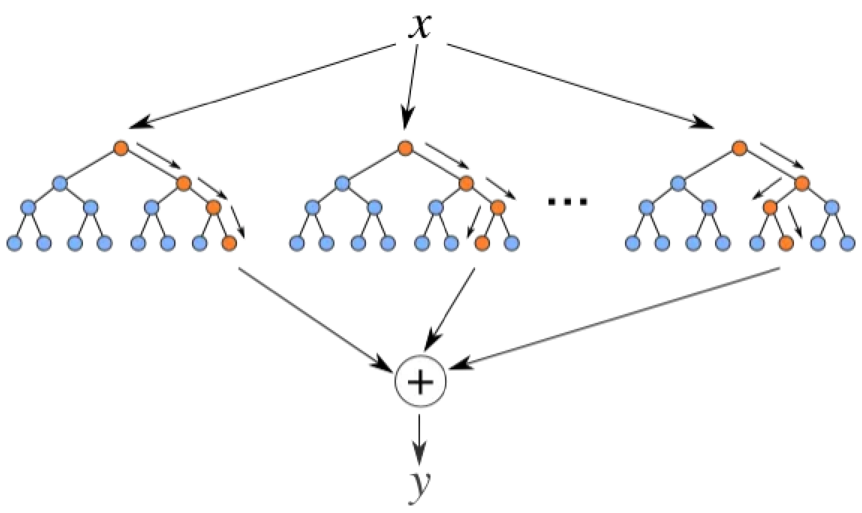


Figure 9: Random Forest Model

Random forest regression is an extension of Random forest algorithm that is tailored for regression problems. Random forest utilizes decision trees to predict. In regression tasks, the goal is to predict a continuous numeric output variable. This is suitable for the problem since I need to predict stock prices which are continuous. Random Forest Regression achieves this by combining multiple decision trees and generating a prediction based on the average (or sometimes weighted average) of the predictions from individual trees. A collection of decision trees is created, with each tree constructed using a different random subspace of features. Each decision tree is grown by recursively splitting the data based on different feature thresholds, aiming to minimize the variance of the target variable. During the prediction phase, the Random Forest Regression model takes a test input and passes it through each individual decision tree. Each tree produces a prediction based on the input features it was trained on. The final prediction is calculated by aggregating the predictions from all trees, typically by taking the average. I used default parameters to train this model after splitting the data into train, test and validation sets with 70%, 20% and 10% split, respectively.

7.2. TABNET REGRESSOR

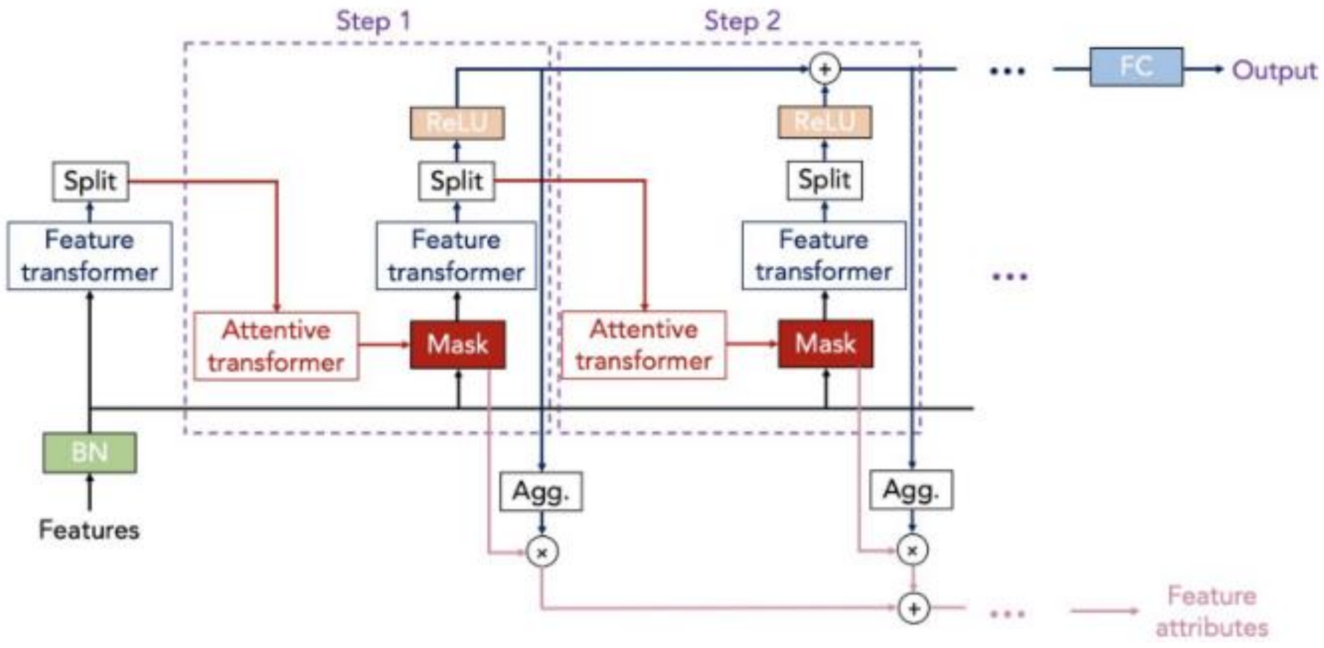


Figure 10: TabNet architecture

TabNet utilizes a modified version of Transformer which is a deep neural network. It consists of an encoder-decoder architecture. The encoder consists of a series of feature transformation steps, known as decision steps, where each step makes selections and transformations on the input features. The decoder learns to reconstruct the original input using the output from the encoder. Attentive transformer uses a sparse attention mechanism during feature selection, allowing the model to focus on the most relevant features and identify important features in data. It then assigns importance scores to the input features, indicating their relevance for making predictions. The decision steps use these importance scores to determine which features to retain and which to discard at each step. This process allows the model to focus on the most informative features. It employs adaptive learning by dynamically updating the importance scores and feature masks during the training process. In TabNet regressor, the final layer of the model is modified to produce a continuous numeric output, suitable for regression tasks.

Experimental settings: Widths of decision prediction layer and attention embedding are set to 6, n_steps is set to 5 with 1.5 gamma (coefficient for feature reusage in the masks), 2 independent and 2 shared Gated Linear Units are used. Adam based optimization with 0.01 learning rate is used. 'Entmax' masking function is used.

7.3. CNN – LSTM – BASED NEURAL NETWORK APPROACH

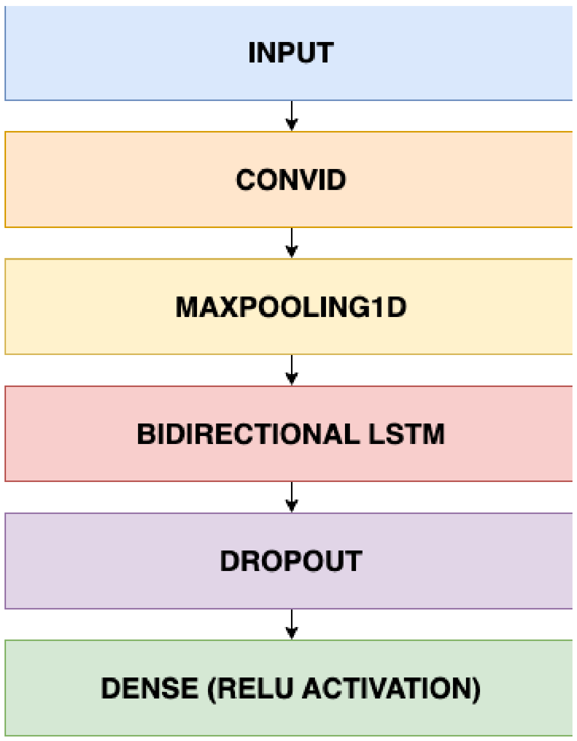


Figure 11: CNN-LSTM based neural network architecture

This approach is based on a sequential neural network. The model is designed for a regression task and combines convolutional neural network (CNN) and long short-term memory (LSTM) layers.

CNNs utilize convolutional layers to perform local feature extraction. This is particularly beneficial for time series data as it allows the network to identify relevant patterns and features at different temporal scales. CNNs can automatically learn filters that capture local structures, such as short-term patterns or trends, by applying convolutional operations over small regions of the input. By applying pooling operations, such as max pooling, CNNs can down sample the data, reducing its dimensionality. Dimensionality reduction helps improve computational efficiency and reduces the risk of overfitting. This makes CNN a good component for dealing with this project's data.

LSTM networks are designed to handle sequences of data with long-term dependencies, which is beneficial for analyzing time series data. The memory cells in LSTM allow them to retain information over longer time intervals, making them capable of capturing and learning patterns that span across multiple time steps. Time series data can have dependencies at various time scales, ranging from short-term patterns to long-term trends. LSTM networks with multiple layers can capture dependencies at different temporal scales, allowing them to extract meaningful features and make predictions at different levels of granularity. Therefore, LSTM is a good choice for this project's data.

The model starts with a 1D convolutional layer (Conv1D) with 128 filters, a kernel size of 2, and valid padding. This layer is designed to capture local patterns and features in the input data. The output is passed through a max pooling layer (MaxPooling1D) with a pool size of 2 and strides of 2, which reduces the spatial dimensions of the data. It is followed by another convolutional layer is added with 64 filters, a kernel size of 2, and valid padding. The output is then passed to second max pooling layer is included with a pool size of 1 and strides of 2, further reducing the spatial dimensions.

Two bidirectional LSTM layers are added (Bidirectional(LSTM)) with 256 units each. The bidirectional aspect allows the LSTM to capture temporal dependencies in the data in both forward and backward directions. A dropout layer with a dropout rate of 0.2 is applied after each LSTM layer to prevent overfitting by randomly dropping units during training. There are two dense layers. The final dense layer has the same number of units as the output shape of the training target. It uses the ReLU activation function suitable for producing the regression output.

The model is programmed using the Keras framework with a TensorFlow backend. The model is compiled with the Adam optimizer and the mean squared error (MSE) loss function, which is suitable for regression tasks.

8. EXPERIMENTAL RESULTS

All the models are evaluated using two metrics: Mean absolute error (MAE) and mean absolute percentage error (MAPE).

Mean Absolute Error (MAE) is a common evaluation metric used in regression tasks. It measures the average absolute difference between the predicted values and the true values, providing a measure of the average magnitude of the errors made by a regression model. Smaller MAE values indicate better model performance, with 0 being the ideal value where the predictions perfectly match the true values.

Mean Absolute Percentage Error (MAPE) is an evaluation metric commonly used to assess the accuracy of a regression model. It measures the average percentage difference between the predicted and actual values, providing a relative measure of the prediction errors.

When dealing with large values, MAE may have a larger magnitude, which can make it difficult to interpret and compare across different datasets. On the other hand, MAPE represents the error as a percentage of the actual values, which normalizes the error and makes it easier to compare across different scales. So I used both the metrics to evaluate models to compare between the models and between two datasets as well.

Stock	**Model**	**MAE - open price**	**MAPE - open price**	**MAE - Adj Close price**	**MAPE - Adj Close price**
Telsa (TSLA)	TabNet Regressor	10.7	0.06	8.5	0.05
	Random Forest Regressor	8.4	0.04	10.26	0.05
	CNN - LSTM	**5.06**	**0.027**	**6.29**	**0.033**
BitCoin USD (BTC-USD)	TabNet Regressor	666.01	0.025	898.63	0.034
	Random Forest Regressor	595.5	0.02	1550	0.06
	CNN - LSTM	**419.41**	**0.016**	**625.28**	**0.025**

Table 2: Results on MAE and MAPE for Tesla and BitCoin Stocks

From the table, it is evident that CNN-LSTM based neural network has outperformed all other models with the lowest MAE and MAPE scores. Random forest regressor and TabNet regressor were able to perform decently but are not consistent between open and adjusted close prices or datasets. By using convolutional and pooling layers, CNNs are able to reduce the dimensionality of the input data, focusing on the most relevant features while disregarding irrelevant details. They are also capable of identifying patterns regardless of their temporal position in the input data. This property proved to be valuable when analyzing stock price patterns that may repeat or occur at different time intervals. On the other hand, LSTM is able to handle input sequences of variable lengths, which proved to be beneficial when dealing with stock price data with different time intervals between data points. Ultimately, CNN was able to extract relevant features from stock price data and feeding these features into the LSTM was able to provide a comprehensive framework for stock prediction, incorporating both local pattern recognition and capturing long-term dependencies to get accurate stock price predictions.

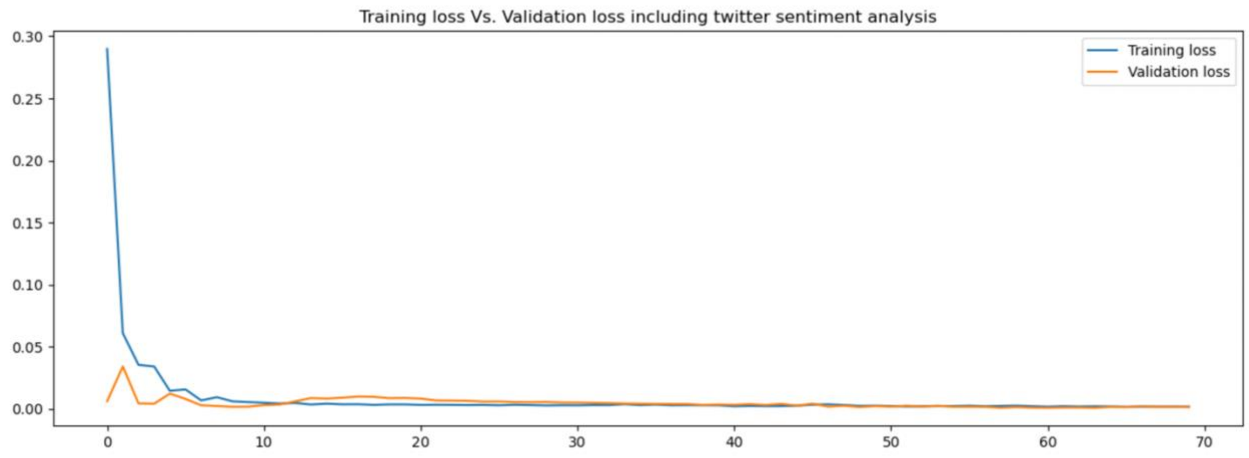


Figure 12: Training loss and validation loss for CNN-LSTM model on Tesla stocks

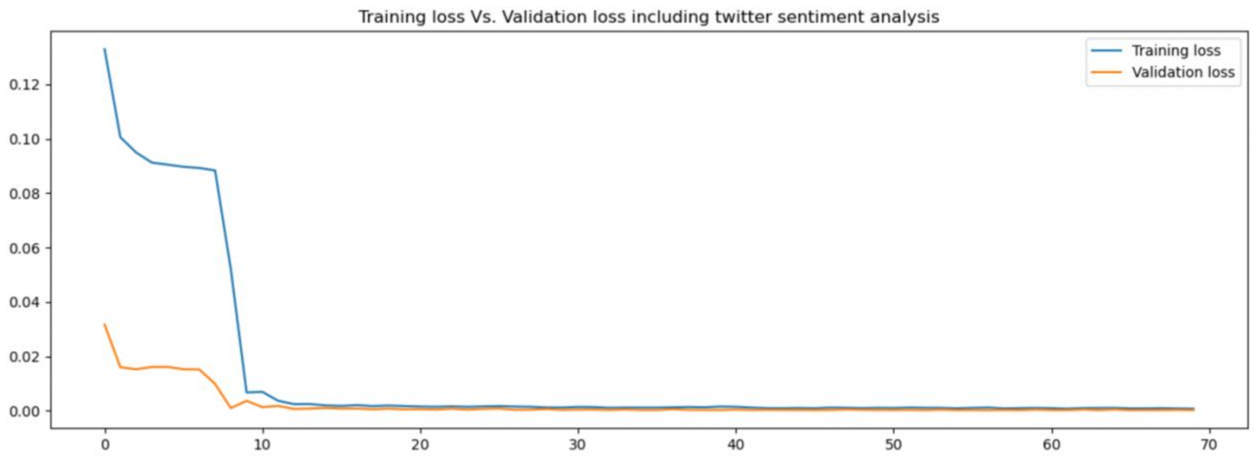


Figure 13: Training loss and validation loss for CNN-LSTM model on BitCoin stocks

From the training loss and validation loss plots, we can observe that neural network model converged well for both the datasets. For Tesla stocks, the model converged around the 10th epoch and figure 12 indicates that the model did not overfit for Tesla data as training and validation losses are close to each other. Also from table 2, we can observe that the test predictions are pretty accurate with low errors. For Bitcoin stocks data from figure 13, the model converged little after the 10th epoch and similar to the Tesla data, there is no overfit and the test predictions are accurate for this data as well. All in all, the performance of CNN-LSTM based model is good.

8.1 MODEL PREDICTION OUTPUTS

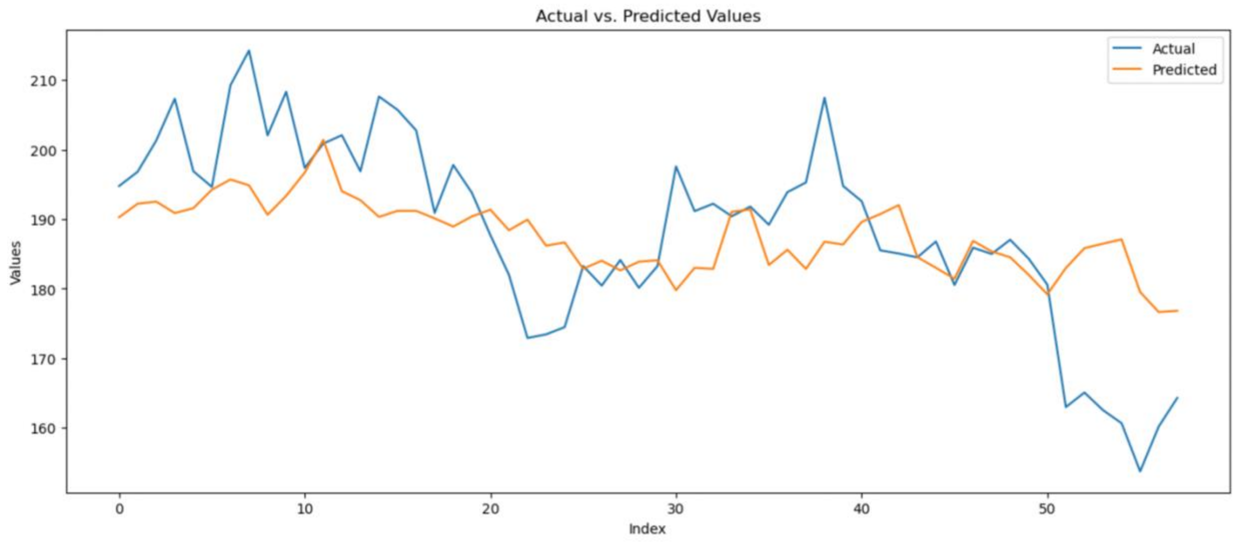


Figure 14: Actual vs Predicted values using TabNet Regressor on Tesla Stocks data.

From figure 14, it is seen that TabNet model did not perform well for Tesla stocks data. Irrespective of hyper parameter tuning, the model was not able to perform accurately. This depicts that the model was not able to learn patterns from the data well to predict as expected.

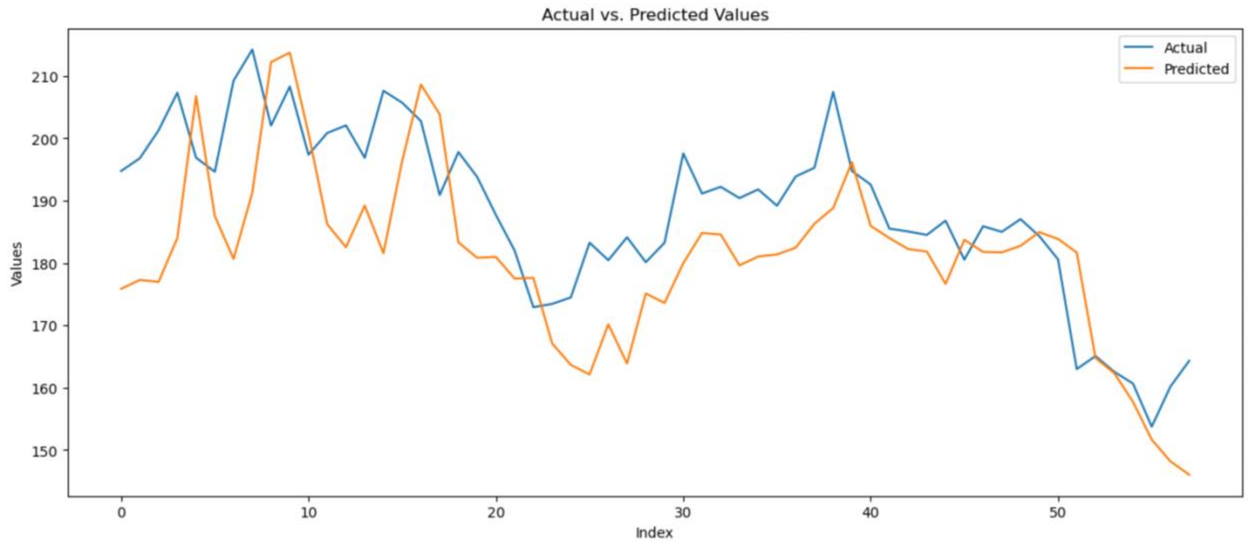


Figure 15: Actual vs Predicted values using Random Forest Regressor on Tesla Stocks data.

From figure 15, it is seen that random forest regressor model performed decently for Tesla stocks data. It was able to learn patterns from the data well and can predict the prices accurately.

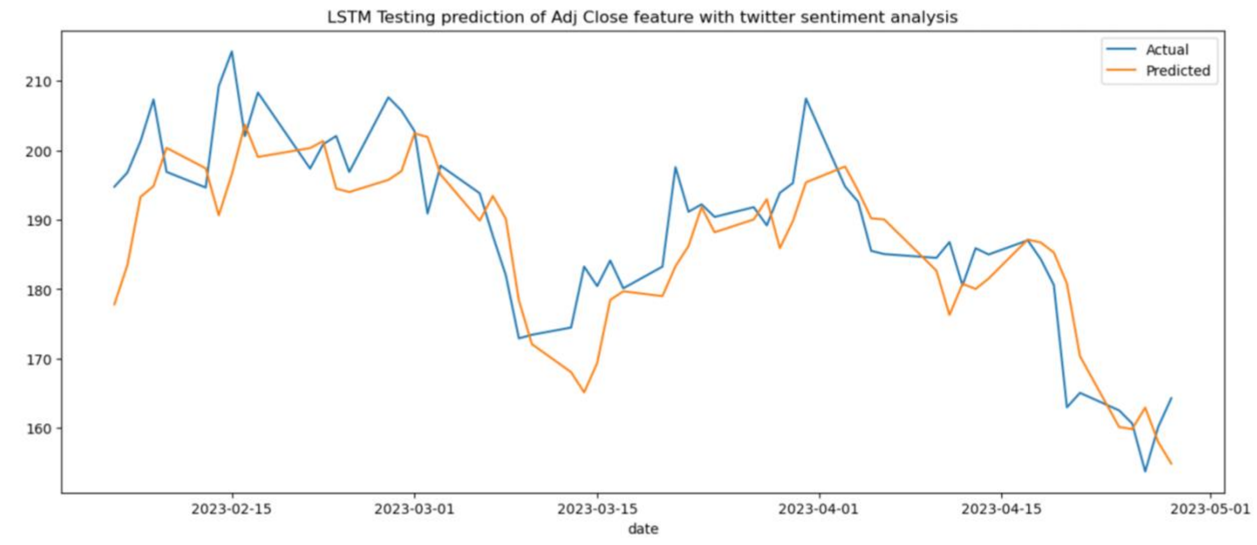


Figure 16: Actual vs Predicted values using CNN-LSTM based neural network model on Tesla Stocks data.

From figure 16, it is seen that CNN-LSTM based neural network model performed very well for Tesla stocks data. It was able to learn patterns the best and predict the stock prices with the lowest loss out of the three models

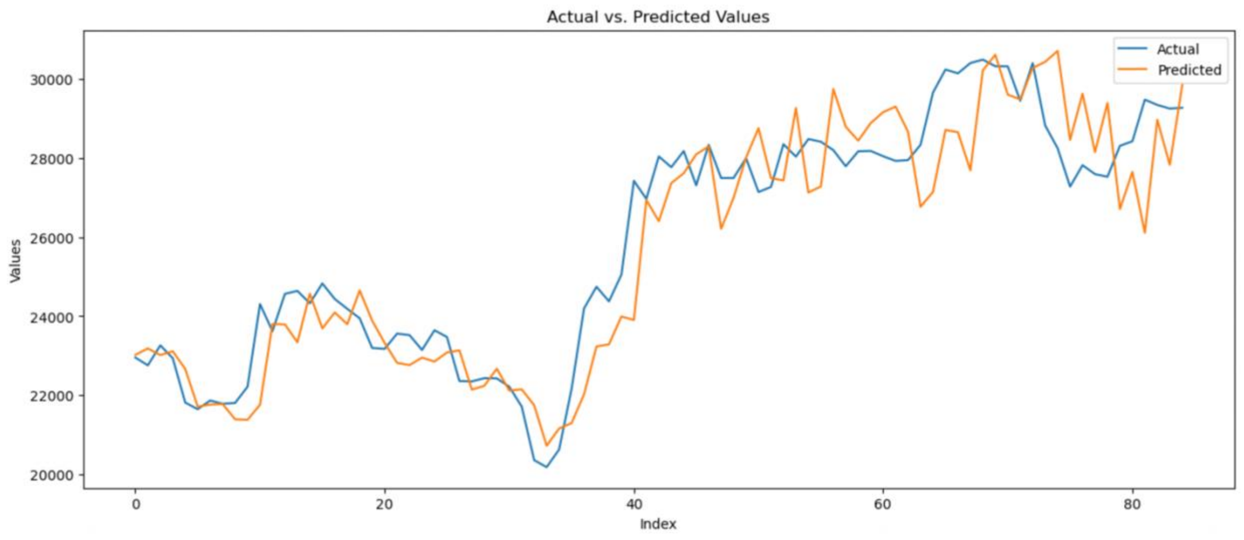


Figure 17: Actual vs Predicted values using TabNet Regressor on BitCoin Stocks data.

From figure 17, it is seen that TabNet model did not perform well for Bitcoin stocks data too. Irrespective of hyper parameter tuning, the model was not able to perform accurately. Overall, this shows that the model was not able to learn patterns from either of the datasets well to predict as expected.

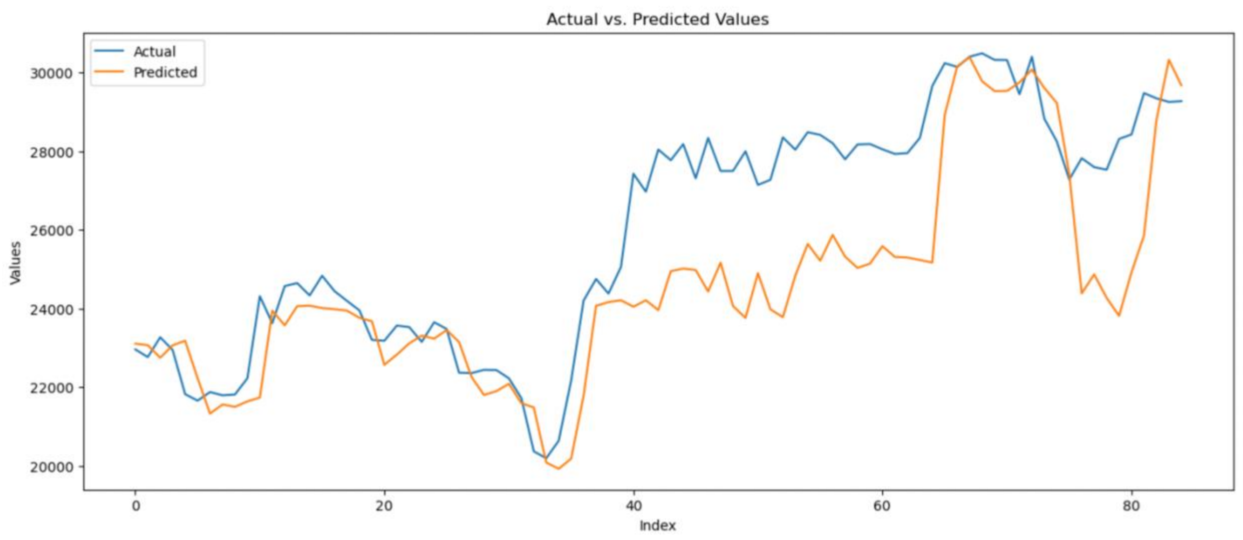


Figure 18: Actual vs Predicted values using Random Forest Regressor on BitCoin Stocks data.

From figure 18, it is seen that random forest regressor model performed decently for the most part on BitCoin stock data. It was able to learn patterns from the data well but the predictions were far apart when there is a big rise in the stock price. Following that, the fall in the stock price was also overly analyzed. Therefore, at rises and falls in the stock prices, it performed poorly.

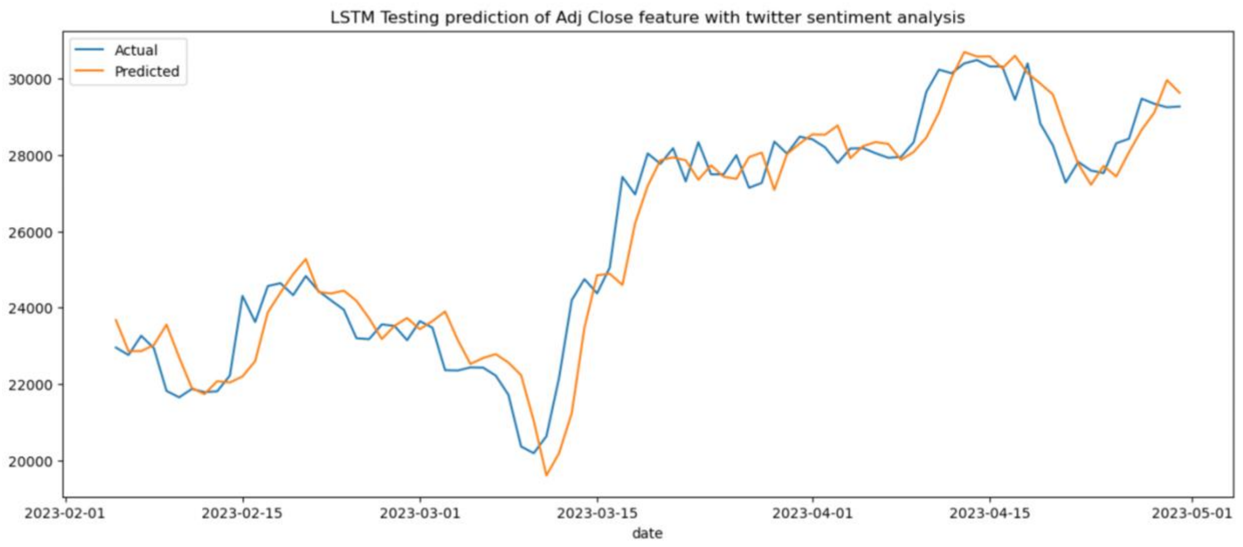


Figure 19: Actual vs Predicted values using CNN-LSTM based neural network model on BitCoin Stocks data.

From figure 19, it is seen that CNN-LSTM based neural network model performed very well for BitCoins stocks data. Similar to Tesla data, it was able to learn patterns the best and predict the stock prices with the lowest loss out of the three models to predict accurately.

9. DISCUSSION

In this project, I collected stocks data for two stocks – Tesla, Inc and BitCoin-USD and also collected Twitter data for these stocks over 14 months' time range. I performed sentiment analysis using Twitter RoBERTa model and then with the help of stocks data and extracted sentiment, I experimented with three different and unique models to predict stocks and finally, evaluated the performance of each model with respect to the dataset. The results depict that the RoBERTa model did a good job of extracting sentiment from twitter text data and classifying sentiments accurately. In addition to this, incorporating sentiment information with stock data resulted in good prediction accuracies over all.

I assessed the performances of TabNet regressor model, Random Forest regressor model and CNN-LSTM based neural network model on two evaluation metrics. I observed that TabNet regressor model was not able to perform well on either of the datasets and the prediction outputs indicated that the model was not able to predict any price rises or falls. On the other hand, random forest regressor performed better and was able to predict prices accurately unless there is a huge jump or fall in stock prices. From the MAE and MAPE scores, it is evident that CNN-LSTM based neural network model has outperformed other models and accurately predicted stock prices with minimal loss. The prediction outputs for this model show that the model has learned patterns in the data well. As discussed earlier, CNNs are inherently translation invariant, meaning that they can identify patterns regardless of their location in the input data. In the context of stock market prediction, this property allows CNNs to recognize patterns and trends in stock prices or indicators regardless of their position in the time series. It helps the model to generalize and make predictions even when the patterns occur at different time intervals. LSTM can effectively capture long-term dependencies in sequential data, which is essential for analyzing stock market time series data and it can also overcome vanishing gradient problem that we might encounter in other RNN based models. This allows LSTM to capture complex patterns and relationships in stock price movements over extended periods. Furthermore, LSTM's ability to capture long-term dependencies and non-linear mappings allows it to adapt to different market conditions, including periods of stability, volatility, and sudden changes in the stock prices. It can learn from historical data and adjust its predictions based on the prevailing market dynamics. Ultimately, the combination of CNN and LSTM has worked out well to predict stock prices accurately.

In conclusion, This project highlights the potential of leveraging social media data for financial analysis. I explored the whether and how sentiment analysis can help in accurate stock price prediction. I used stocks data and sentiment scores derived from tweets as a predictor for future stock prices. Out of the three models that were trained to predict stock prices based on sentiment scores and historical stock data, CNN-LSTM-based Neural Network model showed the most promising results, followed by the Random Forest Regressor. By incorporating sentiment scores derived from tweets along with historical stock data, we can improve the accuracy and reliability of stock price prediction models. This has potential implications for investors and financial analysts, providing them with additional insights and tools for making informed investment decisions. The combination of sentiment analysis and machine learning/deep learning techniques, particularly the CNN-LSTM-based Neural Network, holds promise for accurate stock price prediction.

10. EVALUATION AND REFLECTION

In this project, the main goal was to predict future stock prices using sentiment analysis scores obtained from Twitter data. After data collection and sentiment analysis using RoBERTa, random Forest Regressor, TabNet Regressor, and a CNN-LSTM based neural network were trained to achieve it. Upon evaluation, it was observed that the TabNet Regressor did not perform well, while the Random Forest Regressor gave decent results, and the CNN-LSTM model outperformed the other two models with accurate predictions.

The underperformance of the TabNet Regressor could be because of several factors. It is possible that the implementation or parameter tuning for the TabNet model was not optimal for this data used in this project. Additionally, the lack of adequate data might have limited the model's ability to capture the underlying patterns and relationships in the data, resulting in less accurate predictions. Random Forest Regressor gave a reasonable performance. Random Forest models are known for their ability to handle complex relationships in data and make accurate predictions. However, they might struggle to capture complex relationships among features, especially when it comes to sudden price movements. The relevance of sentiment analysis scores derived from Twitter data would have helped these models perform well. Finally, CNN-LSTM based model is well-suited for sequence-based data as it can capture spatial and temporal dependencies effectively. By combining convolutional and recurrent layers, the model could leverage both the local patterns in sentiment scores and the long-term dependencies in the stock data.

For the future work, data sources can be expanded beyond Twitter data and stock data. Integrating news articles, financial reports, or other relevant textual data sources to capture more factors that can affect stock price movements can help in understanding stock market behavior well. This can also provide a more detailed view of market sentiment and potentially improve the prediction accuracy. In addition to this, gated recurrent units (GRUs), or transformer-based models like the Transformer-XL or the WaveNet architecture can be explored to better capture temporal dependencies and long-term patterns in stocks data.

REFERENCES

[1] U. Shah, B. Karani, J. Shah, and M. Dhande, "Stock Market Prediction Using Sentimental Analysis and Machine Learning," 2021 2nd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2021, pp. 1-4, doi: 10.1109/GCAT52182.2021.9587898.

[2] B. P. Dickinson and W. Hu, “Sentiment Analysis of Investor Opinions on Twitter,” Social Networking, vol. 04, no. 03, pp. 62–71, Jul. 2015, doi: 10.4236/sn.2015.43008.

[3] F. Benrouba and R. Boudour, “Emotional sentiment analysis of social media content for mental health safety,” Social Network Analysis and Mining, vol. 13, no. 1, Jan. 2023, doi: 10.1007/s13278-022-01000-9.

[4] N. N. Reddy, E. Naresh, and V. Kumar B.P., "Predicting Stock Price Using Sentimental Analysis Through Twitter Data," 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2020, pp. 1-5, doi: 10.1109/CONECCT50063.2020.9198494.

[5] R. Gupta and M. Chen, Sentiment Analysis for Stock Price Prediction. 2020. doi: 10.1109/mipr49039.2020.00051.

[6] N. S. Reddy, E. Naresh, and V. S. R. P, Predicting Stock Price Using Sentimental Analysis Through Twitter Data. 2020. doi: 10.1109/conecct50063.2020.9198494.

[7] A. Porshnev, I. Redkin, and A. N. Shevchenko, Machine Learning in Prediction of Stock Market Indicators Based on Historical Data and Data from Twitter Sentiment Analysis. 2013. doi: 10.1109/icdmw.2013.111.

[8] J. Liu and J. Li, A Novel Twitter Sentiment Analysis Model with Baseline Correlation for Financial Market Prediction with Improved Efficiency. Cornell University, 2020. doi: 10.1109/snams.2019.8931720.

[9] L. Bing, K. C. C. Chan, and C. X. Ou, Public Sentiment Analysis in Twitter Data for Prediction of a Company’s Stock Price Movements. 2014. doi: 10.1109/icebe.2014.47.

[10] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, "Sentiment analysis of Twitter data for predicting stock market movements," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, India, 2016, pp. 1345-1350, doi: 10.1109/SCOPES.2016.7955659.

[11] E. Cano-Marin, M. Mora-Cantallops, and S. Sánchez-Alonso, “Twitter as a predictive system: A systematic literature review,” Journal of Business Research, vol. 157, p. 113561, Mar. 2023, doi: 10.1016/j.jbusres.2022.113561.