Sentiment Analysis on Cryptocurrency Patterns

Linda Dominguez
dlinda@mit.edu

Rui-Xi Wang
MIT
rxwangtw@mit.edu

1 INTRODUCTION

Cryptocurrency is a digital currency that serves as an alternative to physical payment. It's implemented through encryption algorithms. For example, some of the most common cryptos include Bitcoin, Ethereum, and Litecoin[3]. Interestingly, it tends fluctuate over time. Cryptocurrency itself tends to be more volatile than stocks, which makes it ideal for analyzing its potential association surrounding sentiments around it. This analysis sought to understand the patterns in sentiment surrounding the crypto and the performance. We looked at X(formerly known as twitter) cryptocurrency tweets to analyze sentiments around surrounding Bitcoin, the most-famous and the oldest crypto in the market. Our study aims to establish a relationship between sentiment and price patterns through statistical modeling and machine learning models such as LSTM(Long Short Term Memory) and GRU(Gated Recurrent Unit).

2 RELATED WORK

The fluctuation in the prices of various asset classes has remained a focal point for investors. Numerous models have been developed to forecast stock prices. Conversely, cryptocurrency represents a burgeoning financial instrument that has emerged in recent decades. Since the inception of Bitcoin in 2009 [2], the oldest and most widely traded cryptocurrency, the trading of this decentralized asset has gained considerable traction in the market. Previous research on cryptocurrency indicates that Recurrent Neural Network architectures, such as LSTM and GRU, can serve as valuable tools for predicting Bitcoin prices [4]. However, the influence of Sentiment Analysis on the predictive accuracy of these models has been underexplored. This study aims to investigate the impact of sentiment analysis, particularly on tweets concerning Bitcoin, on the efficacy of Bitcoin price prediction models.

3 METHODOLOGY

3.1 Alpaca API

The Alpaca API offers a comprehensive interface for accessing and interacting with market data and executing trades programmatically. This API enables people to quickly gather different types of financial data through simple queries and a human-friendly interface. In our work, we utilize the API to gather the bitcoin price for our analysis purpose.

3.2 Kaggle Tweets

The tweets were gathered from a kaggle[1] dataset that compiled Bitcoin tweets. All tweets in the dataset have Bitcoin and btc. The tweet dataset includes username, user location, followers, date, tex, hashtags, and other information.

3.3 LSTM

The Long Short-Term Memory (LSTM) network, a variant of Recurrent Neural Networks (RNNs), has earned significant attention in recent years for its efficacy in modeling sequential data and addressing the vanishing gradient problem encountered in traditional RNN architectures. Developed by Hochreiter and Schmidhuber in 1997, LSTMs introduce specialized memory cells capable of retaining information over extended time steps by utilizing a memory-controlled forget gate that control the weight of pass information, allowing for the capture of long-range dependencies within sequential data. This distinctive architecture enables LSTMs to effectively process and learn from temporal sequences, making them particularly well-suited for a diverse range of applications, including natural language processing, speech recognition, time series prediction, and more. In this paper, we developed time series protection networks with the assistance of the Keras API supported by tensorflow, enabling us to fastly construct recurrent neural networks suitable for time series prediction purposes.

3.4 GRU

Gated Recurrent Unit (GRU) neural networks represent a significant advancement in the field of sequence modeling

and prediction within the realm of artificial intelligence. Proposed as a more streamlined and computationally efficient alternative to traditional recurrent neural networks (RNNs), GRUs offer an elegant solution to the vanishing gradient problem while retaining the ability to capture long-range dependencies in sequential data. Similar to the LSTM architectures, GRUs incorporate gating mechanisms that regulate the flow of information within the network, allowing them to selectively update and forget information over time. The GRU model, however, possesses fewer parameters and hence enables faster training with relatively lower fine-grained accuracy. The versatility and effectiveness of GRU networks have made them suitable tools for us to develop time series prediction tools in parallel with the LSTM module. In this paper, We developed time series protection networks with Keras API.

3.5 Time Series forecasting

Time series forecasting is the process of predicting future values based on past observations of a time-dependent variable. It involves analyzing historical data to identify patterns, trends, and seasonality, and using this information to make predictions about future values. Time series forecasting is used in lots of fields, including finance, economics, meteorology, and machine learning.

The goal of time series forecasting is to accurately predict future values of a time-dependent variable, such as stock prices, sales figures, or temperature readings. Accurate forecasting can help businesses make informed decisions, plan resources effectively, and anticipate changes in market conditions.

3.6 NLTK sentiment analysis

Natural Language Toolkit, also known as NLTK is a library used for text processing and analysis. NLTK includes stop words, tokenization, frequency distribution and other capabilities. Amongst these capabilitites, NLTK has sentiment analysis abilities. We used NLTK's SentimentIntensityAnalyzer to gather sentiments from text. SentimentIntensityAnalyzer is part of NLTK's VADER(Valence Aware Dictionary and sEntiment Reasoner). The SentimentIntensityAnalyzer returns a dictionary of sentiments including negative, neutral, positive, and compound scores. The other scores(positive, negative, and neutral) add up to 1. The compound score is a sum of all of the sentiments that's then standardized into a range of -1 to 1.

4 DATA PROCESSING AND EXPERIMENT

4.1 Bitcoin Price Extraction

We used Alpaca API to create a dataset of Bitcoin daily prices from February 5, 2021 through January 9, 2023. The dataset contains the crypto name, the date, the open and close prices, the high and low prices, the average(low+high/2), volume, trade count, vwap(Volume-Weighted Average Price), and the timestamp.

4.2 Tweet Sentiment Analysis

In our data pre-processing pipeline, we begin by selecting the tweet with strings containing 'BTC' or 'Bitcoin' to get the Bitcoin-related tweets that we are interested in. Subsequently, we extract the days, months, and years from the timestamps of Twitter data, which is essential for deriving the labels necessary to compute the daily tweet count. Notably, the dataset exhibits a highly skewed distribution, with over 60 percent of days missing entries. To mitigate this challenge, we adopt the assumption that sentiment scores for absent dates can be estimated via interpolation or average substitution of existing data. Our approach involves initially computing sentiment scores for each tweet within the dataset, followed by aggregating these scores on a daily basis, computing the mean sentiment scores for each date. Subsequently, we employ interpolation and average substitution methods to infer sentiment scores for the remaining missing dates, thereby enriching our dataset for comprehensive sentiment analysis.

4.3 Time Series Forecasting

We conducted time series forecasting for Bitcoin using exponential smoothing. We began by splitting the data into testing and training data. Training data was all data before April 20, 2021 and testing data was everything forward. The training data was around 17.61% of the data and the testing data was around 82.39% of the data. The Mean Absolute Percentage Error(MAPE) was used as the model evaluation metric.

Firstly, a naive forecast model was created. Naive forecasting assumes future values will be the same as the last observed value.

Next, we crated an exponential smoothing model to the average Bitcoin prices using the ExponentialSmoothing function from the statsmodels library. Exponential smoothing is a technique for smoothing time series data by giving more weight to recent observations. We used exponential smoothing with an additive trend and seasonality to fit the model with specific smoothing parameters and seasonal periods of 7 days. We calculated the MAPE of the final exponential smoothing model and plotted the original prices and fitted values from the exponential smoothing model.

Finally, using the fitted exponential smoothing model, future predictions for Bitcoin we generated prices for the test period.

4.4 GRU

Two GRU models were created. Data was normalized using The data was prepped by converting the date column into DateTime format as well as feature selection and StandardScaler to ensure all features had the same scale. The dataset was divided into input sequences and corresponding output values. The model uses a sliding window approach where the past 20 days are used to predict the future 3 days. The GRU models use a Sequential model architecture in Keras with two layers(127 and 64 units). They have a dropout layer to prevent over fitting at a rate of .2 as well as Dense layer. Finally, the model is compiled with adam and uses MSE(mean squared error) as the loss function. The model is trained and RMSE(Root Mean Squared Error) is calculated to find the trained model performance. Next, both models were used on an 180 day forecast. They predict future stock prices for a given number of days(180) using the trained model. We gather the inverse predicted values to get the actual predicted values and finish by plotting the forecaster prices next to the actual prices to visualize model performance.

4.5 LSTM model

After interpolating existing data points, we now possess 704 dates suitable for time series analysis of sentiment scores and crypto prices. To assess the impact of integrating sentiment scores into the existing LSTM prediction network, we deploy two LSTM models. The first model utilizes cryptocurrency prices as the sequential input fed into the LSTM, with two hidden LSTM layers with 50-token wide and two dense layers to map the sequence to the final prediction value. The second model augments this input with sentiment scores as additional components within the time series vector. We use two LSTM layers with 100-token and 50-token wide and two dense layers to map the augmented sequence vectors to the predicted value. Our training regimen involves utilizing 80 percent of the dates for training and the remaining 20 percent for testing, employing a lookback window of 60 days. Evaluation of our models encompasses three pricing metrics (open, close, and average prices) and four categories of sentiment scores (neutral, negative, positive, and compound scores).

5 RESULT

5.1 Bitcoin Price Extraction

5.2 Tweet Sentiment Analysis

After selecting relevant tweets from the dataset, we obtained a total of 3,310,426 tweets pertaining to Bitcoin within a span of 704 days. However, upon examination, we observed a highly skewed distribution of tweet posting dates. The data points were predominantly concentrated within 221 days,

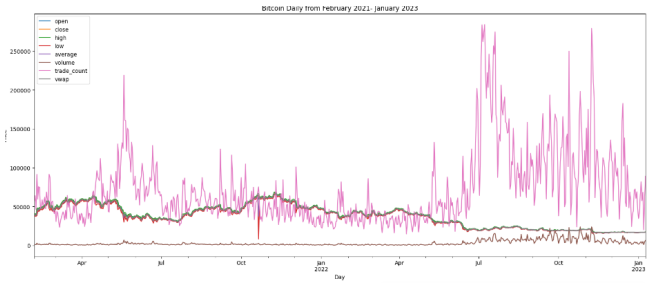

Figure 1: Bitcoin Daily from February 2021 - January 2023

with no tweets available on the remaining days. To address this bias, we employed two separate approaches: one is interpolation, and one is average substitution. Interpolation was used to fill in the missing dates between available data points, thereby creating a more evenly distributed dataset. Average Substitution, on the other hand, was used based on the assumption that the data is envely distributed. By assuming that the sentiment score on each day should be similar to that of neighboring dates, we utilized the interpolated data for further analysis. The resulting plot demonstrates a smoother trend in the interpolated data compared with average substitution, suggesting that the former approach can better mitigate the bias inherent in the dataset. This preprocessing step allows for a more robust examination of sentiment trends related to Bitcoin over the specified time period.

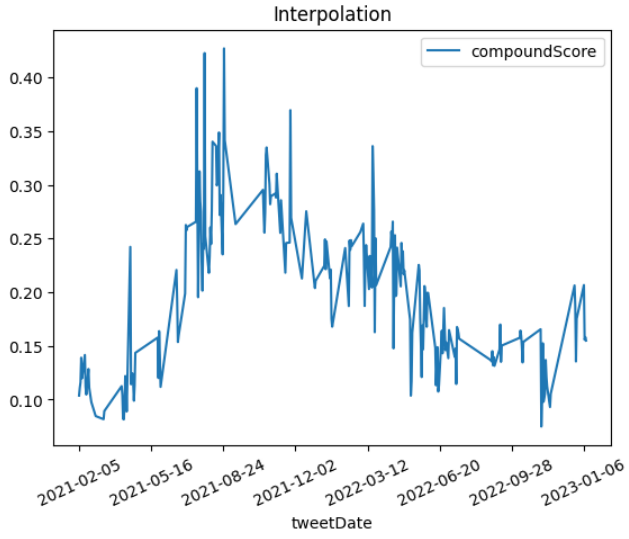

Figure 2: Interpolation Score

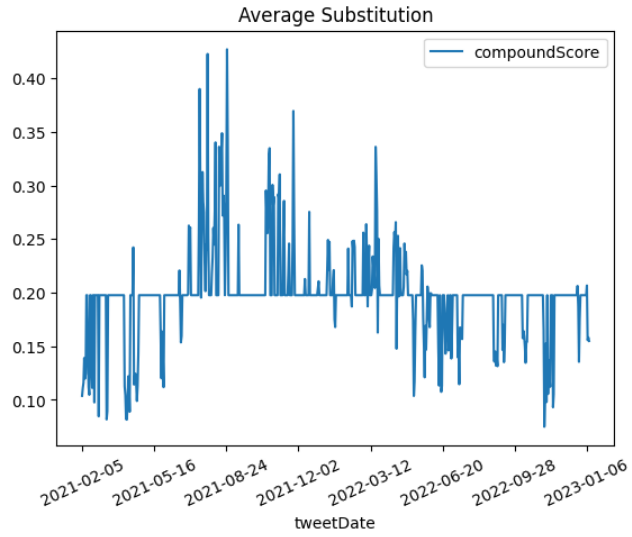

Figure 3: Average Subtition Score

5.3 Forecasting Model

The Naive Forecast model had a test MAPE of 47.86, which is great. However, it performs quite poorly on the testing data.

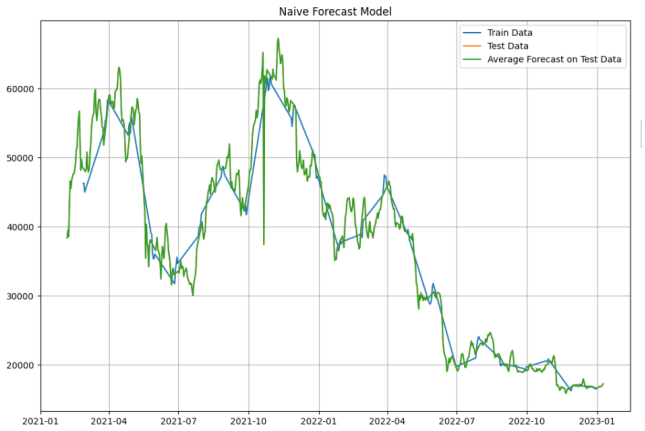

Figure 4: Naive Forecast Model Testing Data

The final exponential smoothing model using only the bitcoin price performed quite well. It had a final MAPE of 3.162, indicating great performance.

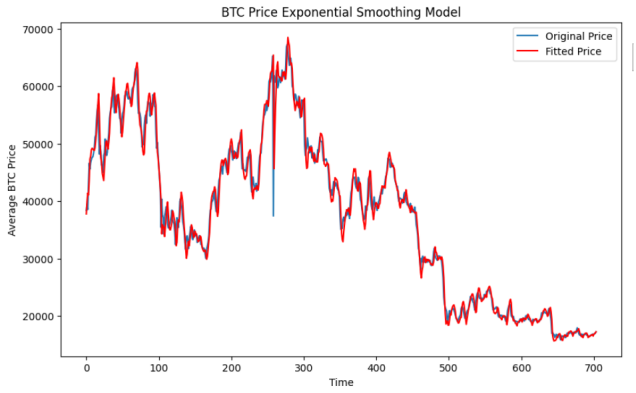

Figure 5: BTC Price Exponential Smoothing Model

The other model used only the sentiment scores. This model ends up with a final MAPE of 16.571%. The sentiment score performed considerably worse than the price score.

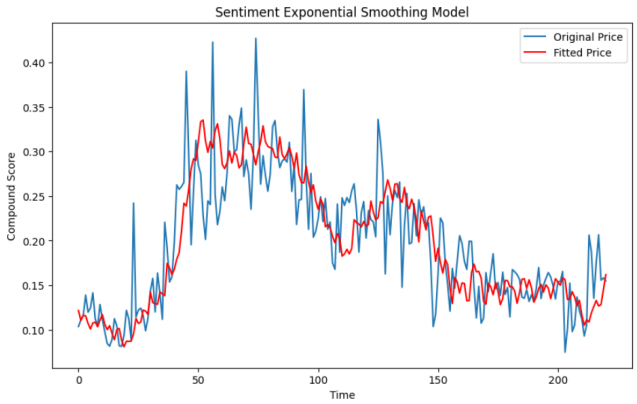

Figure 6: Sentiment Exponential Smoothing Model

5.4 GRU

The Gated Recurrent Unit model performed quite poorly.

The first model used a subset of features and the second GRU model used all the features. The first model's features were average price and the compound sentiment score. For this model, the RMSE was 4510.62, which is extremely high. In essence this means that on average, the predicted values are $4,510 off from the actual values.

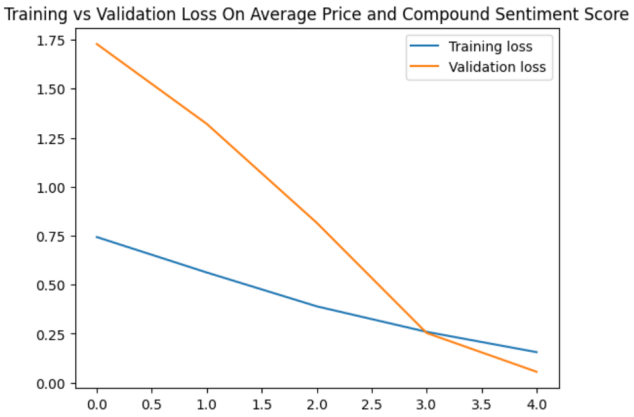

Figure 7: Training vs Validation Loss On Average Price and Compound Sentiment Score

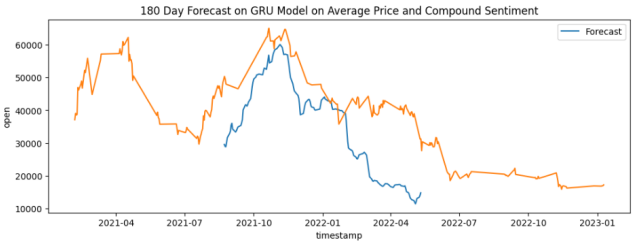

Figure 8: 180 Day Forecast on GRU Model on Average Price and Compound Sentiment

The second GRU model used all available features. These features included: close price, high price, low price, average price, volume, trade count, vwap, compound score, count, negative score, positive score and neutral score. The model performed better. It had an RMSE of 4457.89, which indicates using all testing data leads to better results than only using the compound sentiment score and bitcoin prices.

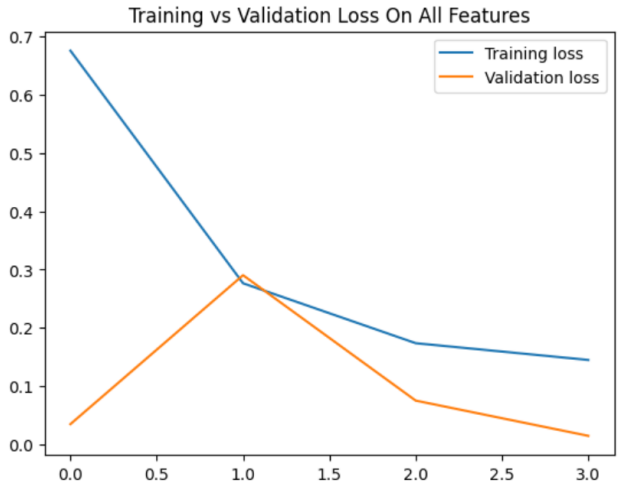

Figure 9: Training vs Validation Loss on All Features in GRU Model

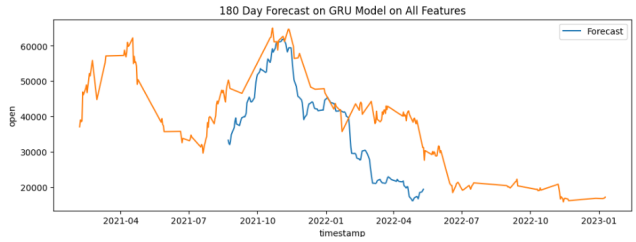

Figure 10: GRU Performance All Features

5.5 LSTM model

In both the sentiment and naive models, the LSTM model can successfully capture the general trend of crypto price fluctuation on the testing set, with an RMSE between the predicted value and actual prices ranging from 500 to 1700, varying between different models and types of sentiment scores. We observe that both the naive model and the sentiment model perform better in open price prediction, with RMSEs consistently lower than 800 for every type of sentiment score. This may be attributed to the nature of the open price. Given the current market situation and news, investors have more information and time to decide whether they should trade Bitcoin. This pattern enables the market to follow a more predictable and stable trajectory. However, both models perform the worst in average price prediction, with RMSEs averaging around 1500. This may be due to their reliance on relevant information flowing into the market, which cannot be deduced solely from the previous average price.

Regarding the performance of the two models, the sentiment model outperforms the naive model in 8 out of the

12 experiments. All sentiment scores, except for the Neutral scores, enable the sentiment model to outperform the naive model in at least 2 of the 3 types of price prediction. Among all 4 types of sentiment scores, the compound score emerges as the best general augmentation metric, as it enables better prediction by the sentiment model across all three types of market prices. This may be because compound scores provide a more comprehensive understanding of the sentiment in a sentence, making them applicable in a wider range of cases. Conversely, negative scores and positive scores perform better in close price prediction and average price prediction, respectively. This improvement can also be attributed to the nature of these prices. If negative comments flood Twitter, investors may panic and sell off their cryptocurrencies, leading to a sudden plummet in prices. Therefore, acquiring sentiment scores can help gauge the atmosphere of the current market and facilitate crypto price prediction. Similarly, positive tweets can potentially boost market confidence and consequently raise crypto prices. However, close price prediction using positive scores does not show improvement. A possible explanation is that once crypto prices skyrocket, investors may sell off their holdings, resulting in a sudden drop in price.

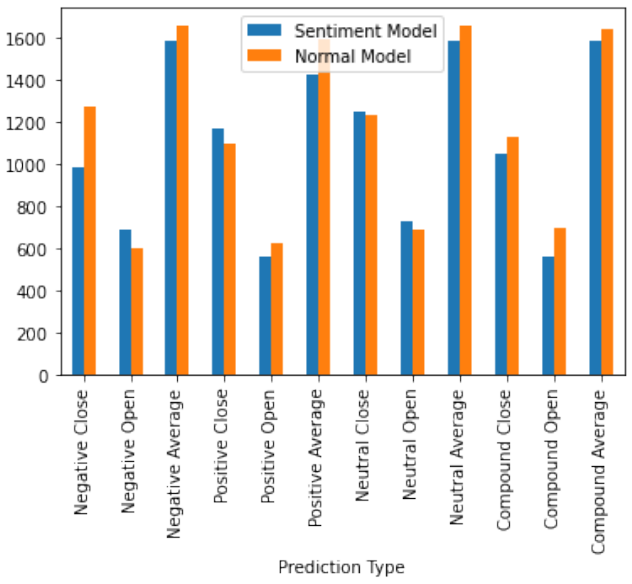

Figure 11: RMSE for different prediction tasks of two LSTM models

A prediction curve of the sentiment model and the naive model is illustrated in the figure below. It's evident that the sentiment model outperforms the naive model in capturing the general trend of market price fluctuation. However, upon closer inspection, the prediction curve of the sentiment model reveals a notable characteristic: it exhibits a shifting trend that fluctuates at a slower pace compared to the actual price values. One plausible explanation for this phenomenon is that investors react with a delay to market movements. Typically, investors observe market fluctuations and subsequently respond or post tweets reflecting their reactions to the market conditions. Consequently, these tweets tend to reflect a delayed response relative to the real-time market changes. Integrating these sentiment scores into the prediction model introduces an inherent delay into the model's predictions, as it reflects the lag in investors' reactions. This delay could potentially account for the slower fluctuation observed in the sentiment model's prediction curve compared to the actual market prices. This delay effect highlights an important consideration when incorporating sentiment analysis into price prediction models. While sentiment analysis provides valuable insights into market sentiment, it's essential to account for the time lag between market events and the corresponding sentiment expressed by investors. Strategies such as adjusting for this delay or implementing real-time sentiment analysis techniques could help mitigate the impact of this delay and improve the accuracy of price predictions.

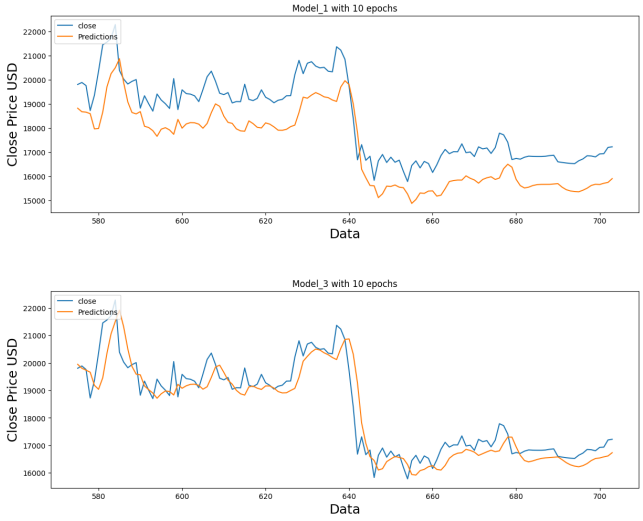

Figure 12: Close Price prediction with negative sentiment scores, model 1 is the naive model and model 3 is the sentiment model

6 CONCLUSION

We confirm that time series prediction techniques, such as LSTM recurrent neural networks and exponential smoothing methods, can capture the general trends of market price fluctuation. The assistance of extra input factors, such as different types of sentiment scores, can facilitate the prediction of forecasting systems. However, this extra information

should be used carefully, as one type of score may not be general enough to represent every kind of sentiment demonstrated in the tweets. Compound sentiment score can be a useful metric as it gives us the most general metric of all types of sentiment. The results suggest that a more fine-grained and general sentiment metric, such as the output of language models like GPT or BERT, can be a potential tool for the construction of a more accurate time-series and bitcoin price prediction system.

REFERENCES

[1] https://www.kaggle.com/datasets/kaushiksuresh147/bitcoin-tweets. Bitcoin Tweets. (https://www.kaggle.com/datasets/kaushiksuresh147/bitcoin-tweets).

[2] Satoshi Nakamoto. 2009. Bitcoin: A Peer-to-Peer Electronic Cash System. (May 2009). http://www.bitcoin.org/bitcoin.pdf

[3] State University of New York Oswego. 2024. The Basics about Cryptocurrency. (2024).

[4] Phumudzo Lloyd Seabe, Claude Rodrigue Bambe Moutsinga, and Edson Pindza. 2023. Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach. *Fractal and Fractional* 7, 2 (2023). https://doi.org/10.3390/fractalfract7020203