Twitter Sentiment Analysis and Bitcoin Price Prediction

Joel Pimenta
Department of Computer Science and Engineering
The University of Texas at Arlington
Arlington, TX USA
joel.pimenta@mavs.uta.edu

Nathaniel Fenoglio
Department of Computer Science and Engineering
The University of Texas at Arlington
Arlington, TX USA
nathaniel.fenoglio@mavs.uta.edu

***Abstract*—Twitter sentiment has been found to more successfully indicate the direction of Bitcoin price in comparison to news releases for non-cryptocurrency currencies. In this paper we investigate previously employed strategies for using Twitter sentiment to predict the price direction of Bitcoin in order to build a model in hopes of further contributing to the field. We have implemented Natural Language Processing (NLP) to examine how Bitcoin values can be estimated from public opinion. Our approach is developed in Python using the Polyglot NLP library, an LSTM (Long Short Term Memory Model) for model training and tested on manually gathered datasets.**

***Index Terms*—Bitcoin, Sentiment analysis, Prediction methods, Cryptocurrencies, Twitter, LSTM**

I. Introduction

Bitcoin is an open-source, public, decentralized, and digital peer-to-peer currency in which transactions are carried out collectively by the network. The transactions are verified through cryptography and recorded in a distributed ledger known as the blockchain. It was invented in 2008 by an unknown person by the name of ”Satoshi Nakamoto” and the term ”Bitcoin” was defined in their white paper published later in October 2008. The Bitcoin organization’s objective for the platform is to promote instantaneous transactions over the internet, rather than through traditional means involving intermediaries such as banks or governments. [4]

Sentiment analysis, also known as opinion mining, is an application of NLP used to classify text by obtaining the sentiment of the data through analysis. For example, it can be used to determine whether the tone of a message is positive, negative, or neutral. Sentiment Analysis has two stages, Preprocessing and Keyword Analysis. During the Preprocessing stage, sentiment analysis uses tokenization to break keywords into tokens, it uses Lemmatization to convert words to their root form, and it uses Stop-word removal to filter out words that do not add value to the sentence. In the Keyword Analysis stage, NLP is used to analyze extracted keywords at a deeper level. Finally, it gives the keywords a sentiment score, which is a measurement scale relative to the emotions being analyzed. [5]

The vast amount of data available through the social media platform Twitter makes the opportunity for sentiment analysis readily available. Various studies involving predicting the prices of digital currencies, such as Bitcoin, showed that price trends did not follow news releases like other global currencies, and instead Twitter sentiment analysis was an approach that produced better results for predicting future price moves. [1]

II. Previous Studies and Implementations

In this section we present our research on some of the previous implementations involving Twitter sentiment analysis and price prediction for the cryptocurrency Bitcoin.

In the last half decade cryptocurrencies, including Bitcoin, have enjoyed a major presence in the public eye due to their volatile nature. Past research work has often utilized social media platforms, namely Twitter, as prediction for forecasting future values. Masqood et al.[6] compare and contrast the performance of several Machine Learning algorithms in predicting future values of cryptocurrency. They ascertained this information by using a Sentiment Analyzer and gathered relevant tweets by utilizing features that had a hashtag with ”Bitcoin”. Also included in their calculation is the open, high, low, and close price of Bitcoin on the day each Tweet was collected. Their conclusive results expressed that Long Short-Term Memory (LSTM) was the highest-performing model across various performance metrics, such as Mean Absolute Error (MAE), Mean Absolute Scaled Error (MASE), Mean Absolute Percentage Error (MAPE), and Mean Squared Error (MSE).

In a study conducted by Sattarov et al.[7], they uncover if there is any correlation between the price fluctuation of Bitcoin and the sentiment of Twitter regarding it. As with Masqood, Sattarov and others used a sentiment analyzer on Bitcoin-related tweets scraped from Twitter. In regards to their methodology, they chose a time span of 60 days between March 12, 2018 and May 12, 2018. This was when Bitcoin was most explosive in its growth and decline. The specific sentiment analysis method they used was Valence Aware Dictionary and Sentiment Reasoner (VADER). The reason they chose this method, is because VADER is designed with social media in mind, and therefore will have a higher rate of accuracy from a social media website as Twitter. To evaluate this model, they used Random Forest Regression (RFR), as RFR is best for situations where the input has few relationships among its features. They observed the presence of a correlation between the price of Bitcoin and Twitter sentiment. Their final result was a 62.48% accuracy when making predictions based on bitcoin related tweets and the historical bitcoin price. They suggested in future research for studies to consider external factors, such as Twitter users, number of tweets, and emoticons.

The authors of “Bitcoin price change and trend prediction through twitter sentiment and data volume” [2] note that the current best Twitter/Bitcoin sentiment prediction models predict price direction but not the magnitude of the price increase or decrease and employ tweet sentiment along with tweet volume, recurrent neural network and convolutional neural network models with an additional model to forecast the magnitude of price change attempting to predict what interval range the price movement will fall in.

The optimal time interval by which to pair tweet sentiment and price change is explored with different time lags to find the interval that produces the best accuracy of sentiment to actual price change. [2] The datasets incorporated price data by shifting and merging price by 1-day, 3-day, and 7-day time interval lags producing 3 different datasets. Tweets are grouped by day and an average of the polarity scores are taken along with the total volume of tweets for the day and the closing price for the day.

The authors used the number of tweets per day in the dataset used to build the model based on the intuition that tweet volume would have a correlation with price movement. [2] We feel that adding tweet volume as a feature to train the algorithm would be relevant information, especially when attempting to predict not only the price direction but also the magnitude of the price change. A positive sentiment for a day with a relatively larger volume of tweets would seem to indicate a larger size movement in price for instance.

LSTM (Long Short-Term Memory) recurrent neural network, CNN (Convolutional Neural Network), and BiLSTM (Bidirectional Long Short-Term Memory) classifiers were all used separately in training models to predict Bitcoin price changes. The features used were the direction of price change for the day, the closing price, the positive polarity for the day, the negative polarity for the day, and the tweet volume for the day. [2] The results of their study will now be briefly detailed.

The 1-day and 7-day time lags produced the highest mean accuracy suggesting that immediate response to sentiment can be accurate but can also have a high variance with the smaller time interval variances being able to be smoothed out over longer time intervals when using the 7-day time lags. The BiLSTM (Bidirectional Long Short-Term Memory) architecture produces the highest accuracies with the lowest variance. The CNN (Convolutional Neural Network) architecture was the best at predicting the magnitude of price change with a 1-day time interval. [2] The authors state that future work should further explore the time lag aspect of matching up prices with tweet sentiment which we plan to investigate in our study.

The authors of “Combining Likes-Retweet Analysis and Naive Bayes Classifier within Twitter for Sentiment Analysis” [3] take note that Twitter has additional features that have not generally been used in sentiment analysis such as the number of likes and number of retweets of a tweet. The study combines textual classification with the non-textual number of likes and number of retweets information to improve the accuracy of sentiment prediction. The Fisher Score statistic tool is used to estimate the score or weight to determine the number of likes, number of retweets feature importance and this score is then combined with the textual sentiment analysis with a proportion of 60 percent textual and 40 percent non-textual. The number of likes and number of retweets does also seem like relevant information to include in model training that we plan to explore in our study. Similar to authors of [2] including the daily tweet volume in their model training, the overall sentiment of a tweet with a larger number of likes and retweets would seem to carry more weight than the sentiment of a tweet with fewer likes and retweets.

III. Sentiment Analysis Dealing With Raw Data and Feature Extraction

A. Preprocessing

The step of classifying the sentiment of tweets begins first with the preprocessing of the dataset. Preprocessing involves a series of techniques such as cleaning, tokenization, and stop word removal that are applied to raw textual data to transform it into a format that is better suited for analysis. The authors of [1] used data preprocessing steps to remove words and characters that do not benefit analysis such as stop words like “the”, “and”, and “but”, punctuation, and the conversion of words to their base forms are detailed and shown to help reduce the dimensionality of the data to be analyzed.

In the preprocessing stage the authors of [2] removed non-English tweets, duplicate tweets, and URLs which do not contribute to accurate sentiment analysis. A lemmatization process was performed to map words to their base form so that words with the same morphological base are not considered as different words when performing analysis. Other preprocessing involved the removal of stop words, replacing user mentions with only the word “USER”, removal of punctuation, removal of hashtags if the word that follows is not in an English wordlist, and the discarding of tweets of less than 4 words.

B. Feature Extraction

Features such as terms frequency, parts-of-speech labels, opinion words and phrases, and negations are valuable to extract from the data for performing sentiment analysis. Some feature extraction techniques are Bag of Words which represents the counts of every word in the document, Distributed Representation which is able to also take into account the context of words in contrast to Bag of Words which focuses only on the words' frequencies in a document. [1] The importance of feature extraction in the sentiment analysis process is that it can help reduce the dimensionality of the dataset to make processing faster for the machine learning algorithm. Sentiment analysis can many times involve processing large amounts of data and reducing the dimensionality of the dataset can improve the performance as well as the speed of the algorithm.

IV. General Overview of Sentiment Analysis Approaches

Approaches to sentiment analysis can be divided into three general categories, machine learning approaches, lexicon-based approaches, and hybrid approaches.

A. Machine Learning Approaches

Machine learning uses algorithms and linguistic features for classification and is further divided as supervised learning, when class labels are available with a dataset, and unsupervised learning, when class labels are not available. The machine learning strategies are able to learn patterns from specific types of datasets and can achieve better accuracy than other datasets for this reason. [1]

Some supervised learning approaches are Support Vector Machine (SVM), which attempts to find the hyperplane that splits the training data with the widest margin in between different classes, Artificial Neural Networks that use an input, hidden, and output layer of organized neurons with connections with weights that can be varied as the global error function is minimized between the three layers, Naïve Bayes algorithms that use a probabilistic approach and Bayes theorem which depends on Bag of Words feature extraction techniques and assumes independence of words, not taking into account context, but can achieve high classification accuracy, the Decision Tree approach that makes decision splits based on whether or not a word is found in a document in order to make classification determinations, and Ensemble Learning which combines multiple algorithms that in the end cast their vote with the majority vote determining the classification. [1]

Some unsupervised learning approaches use various clustering techniques to partition the data points by how similar points are to each other in order to determine classifications with unlabeled datasets. Another unsupervised learning approach is the graph-based approach where vertices represent sentences and edges are used to represent the similarity of different sentences. [1]

B. Lexicon-Based Approaches

Lexicon-Based approaches require a precompiled list of words mapped to their positive or negative sentiment rankings. The polarity value score can range from –1 to 1 with negative scores signifying negative sentiment and positive scores signifying positive sentiment. Individual words are scored and then the average of all of the scores is used to classify a document. The lexicon-based approach is categorized as an unsupervised approach and does not need any training data. The approach is limited by scores being assigned to words that may have different contextual meanings in different domains. Domain specific lexicons can be used to more accurately assign sentiment scores to words in relation to a particular domain. The approach also is relatively limited in comparison to some other machine learning approaches when the dataset is large enough for a machine learning algorithm to learn a more nuanced model that the lexicon-based approach is not able to adapt to. There are three different Lexicon-based approaches, the manual approach, the dictionary-based approach, and the corpus-based approach. [1]

- The manual approach of creating a lexicon involves humans manually assigning sentiment labels to words.
- The dictionary-based approach involves manually collecting a list of seed words, which are assigned labels, and then synonyms and antonyms are searched using known dictionaries to iteratively expand the list.
- The corpus-based approach also starts with a collection of seed words, then syntactic and co-occurrence patterns are used to build positive and negative sets of sentiment words with techniques such as clustering.

C. Hybrid Approaches

A hybrid approach combines machine learning with the lexicon-based approach to combine the benefits of stability from the lexicon-based approach and the higher accuracy benefits from machine learning techniques incorporating lexicon sentiment scores on the data which is then fed into sentiment analysis classifiers. [1] An example of a hybrid lexicon-based approach that combines dictionary-based and corpus-based methods for NLP classification that we are investigating is the Polyglot library which supports sentiment analysis for more than 130 different languages which we then use to feed into an LSTM recurrent neural network.

V. Introduction to Our Implementation Strategy

A. Preprocessing

We plan to include non-English tweets mined from Twitter in the dataset in contrast to the authors of [2] and will be able to score the sentiment for more than 130 different languages using the Polyglot library, providing the ability to have a more inclusive range of tweets from whichever language a tweet was composed in.

Stop words such as "the", "and", "but", and punctuation will be removed from tweets. Other elements of tweets that will be removed are URLs, non-ASCII characters, mentions of other Twitter users that follow the "@" symbol, and hashtags. Tokenization and lemmatization steps will be applied converting words to their base forms to assist in reducing the dimensionality of the data being analyzed. Tweets that contain only numbers will not be useful in analyzing sentiment and will also be discarded from the dataset.

B. Choice of sentiment analysis approach/machine learning algorithm

The tweets that will be used to perform sentiment analysis on were originally scraped from Twitter with the following six features saved: datetime, id, content, username, number of likes, number of retweets. After the average daily sentiment is classified using the Polyglot library, factoring in the number of likes and the number of retweets into a single tweet's score, the features that are preserved are: date, mean weighted sentiment by day, and the total number of tweets for a day.

These attributes are then merged with the Bitcoin close price per day. Columns for the close price difference in comparison to the previous day and a categorical attribute for whether the price direction was up or down are then calculated and integrated with the dataset.

Through visualizing the average sentiment and number of tweets over time, it appears that the number of tweets tend to increase significantly when the average sentiment decreases as can be seen in figure 1.

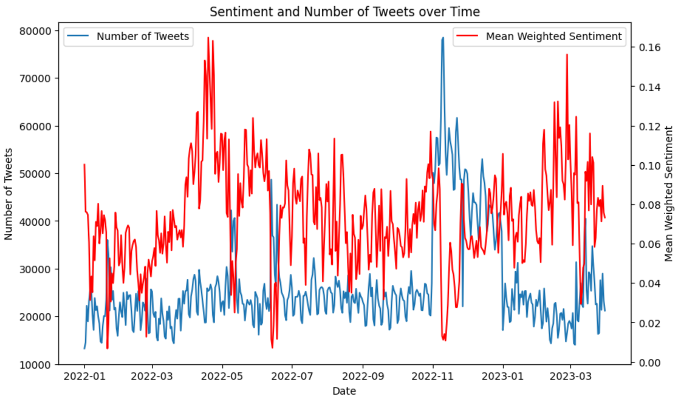

Fig. 1. Sentiment and Number of Tweets Over Time

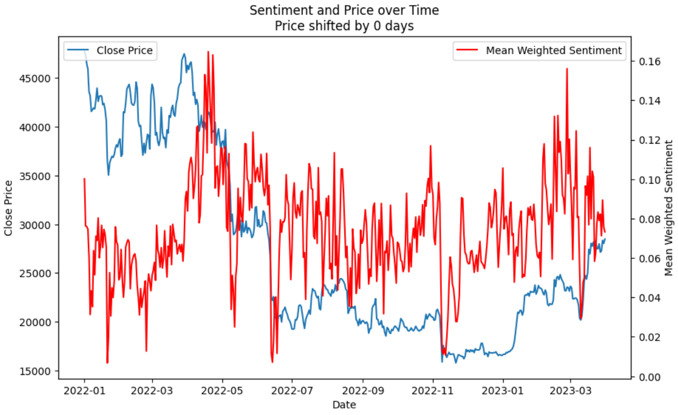

Fig. 2. Sentiment and Price Over Time Price Shifted by 0 Days

The volume of daily tweets will be incorporated like [2] and number of likes, number of retweets like [3] into the sentiment polarity score calculations for the per day averages because we think that the non-textual data is an important feature to include in sentiment analysis as a tweet with a large amount of likes and retweets should be given a higher weight than a tweet with a smaller amount of likes and retweets.

We will experiment with shifting future day close prices to prior days sentiment scores at various time lag intervals. The closing price and average sentiment over time without any shift can be seen in figure 2.

We will experiment with different time lag intervals for the LSTM window size to further investigate the optimal time lag furthering the research of [2] exploring in the 1-15 day range. LSTM window size is the number of time steps used as inputs to the neural network at each time step. If a larger window size is used, the LSTM will have available to it more information about long-term patterns, but this can make training more difficult due to the increase in the number of parameters needed to process. If a smaller window size is used, long-term patterns are more difficult for the LSTM to discern, but will make the neural network simpler and easier to train. The optimal window size will depend on the characteristics of the time series data.

The Polyglot NLP python package has been used to score the sentiment of tweets including the intensity (not just 1, 0, or -1 but the real number value in between –1 and 1 signifying the positive or negative intensity).

We plan to use the LSTM (Long Short-Term Memory) recurrent neural network algorithm that produced the highest accuracy in [6] to train our model with the average daily sentiment labels to predict the price direction of Bitcoin.

LSTMs present a way to successfully deal with the exploding or vanishing gradient problem in recurrent neural networks.,These situations can occur as the gradients are backpropagated through multiple layers. When the gradients become very small, this can cause the network to converge very slowly, or not at all, because the weight updates are too small to make a significant difference. On the other hand, when the gradients become very large, the weight updates can be too large, causing the network to oscillate or even diverge.

Two separate paths are used in order to make predictions. One path is used for long-term memory and the other is used for short-term memory. The long-term memory path does not have weights and biases that are able to directly affect it allowing for the long-term memory information to flow through the series of units without causing the gradient to explode or vanish. The short-term memory path is connected to weights and biases that are able to modify the memories. The short-term memory path is run through the sigmoid activation function and the output is then multiplied with the long-term memory path to make the current step's contribution to the percentage of long-term memory that is remembered.

Another step involves the short-term memory being run through the tanh activation function and together with the output of the sigmoid function which determines the percentage of this new memory that will be added to the long-term memory, updates the long-term memory. The short-term memory is then updated in the final step of a unit by factoring in the long-term memory using the tanh activation function and sigmoid function to determine the percentage of the contribution to the updated short-term memory and the final output of the LSTM unit.

We used the TwitterHashtagScraper feature from the sntwitter python library to scrape approximately 13 million tweets under hashtag bitcoin from the time period of January 2022 through March 2023 and will use the daily closing price data for the time period to further investigate the optimal time lag of sentiment affecting price direction using the 1-day, 3-day, and 7-day intervals similar to [2] and more time lag intervals in hopes of finding other meaningful sentiment to price direction correlations.

VI. Implementation and Results

A. Implementation

The LSTM layer takes in a 3D array of data size, timesteps, and number of features. Data size is the number of rows in the dataset, timesteps is the window size of the previous rows taken into account for predicting the current price, and number of features is the number of columns in the dataset.

We implemented an LSTM sequential model of a linear stack of layers. The input layer shape is the LSTM window size and the number of attribute columns used. The next layer is an LSTM recurrent neural network layer with 64 memory units. A Dense layer is then added with a Rectified Linear Unit (ReLU) activation function with 8 fully connected neurons in order to map the higher-dimensional output of the previous layer to a lower-dimensional representation. The final Dense output layer is then added with a linear activation function and a single neuron to output the actual predicted price for the day.

The LSTM model was compiled using mean squared error (MSE) as the loss function, Adam as the optimizer with a learning rate of 0.0001, and root mean squared error (RMSE) as the evaluation metric. The training data, using the validation data for evaluation, was then fit for 100 epochs and the best performing model was saved. The model with the lowest root mean squared error was then used to predict the Bitcoin price for the following 91 days.

The dataset was split into 60% training, 20% validation, and 20% testing. We experimented with training models on different configurations of including or not including the 3 different attributes and varied the LSTM window size from 1 to 15 days.

The performance metrics mean squared error, root mean squared error, mean absolute error, and R2 score were collected for each run of training a model. The models were trained five times with the same parameter configurations and the performance metrics were then averaged to compare and find the best model that produced the most accurate price predictions.

B. Results

For many window sizes, training with all 3 attributes, just daily close price and mean weighted sentiment, or just daily close price were actually very similar. The optimal window size range was found to be in the 3-7 day range with a gradual loss in accuracy as the window size is increased. Including the number of tweets as an attribute to train with did not benefit the model although it did produce more accurate models than training with just the daily close price alone. With larger window sizes (12-15 days), the models trained on all 3 attributes tended to perform worse even than the models trained on the daily close price attribute alone. The 12-15 day window is quite large and we have not seen previous experiments that incorporated such a large window size in training an LSTM.

Training with just daily close price and sentiment produced the best performance metrics out of the 3 attribute configurations that we experimented with. The window size of 3 days produced the best performance metrics out of the 1-15 day range experimented with.

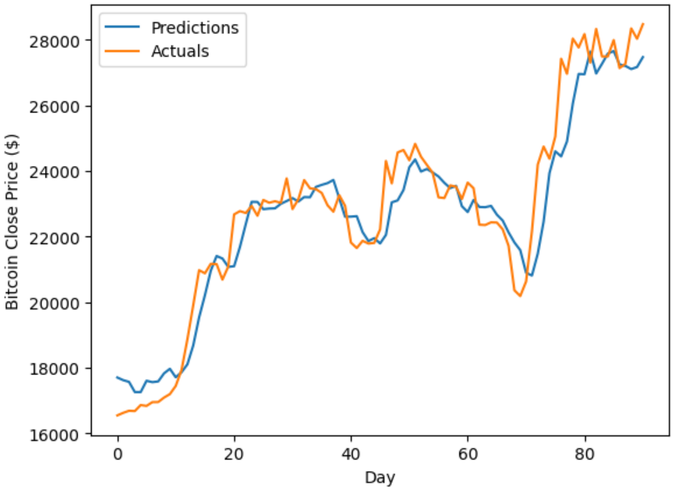

Fig. 3. Test Set 91 Day Price Prediction Comparison to Actual Price (Close Price and Sentiment Attributes Only)

Since the 3-day window size produced the best performance metrics for all 3 attribute configurations, we decided to take 55 samples of training the models with the different attribute configurations. Training the LSTM with a 3-day window size and just daily close price and sentiment produced an R2 score of 0.82 in comparison to 0.72 for the model trained on all 3 attributes (daily close price, sentiment, number of tweets) and 0.63 for the model trained on only the daily close price.

R2 Scores (3-day window size) average 55 trials each
(daily close price, sentiment, # of tweets) - 0.72
(daily close price and sentiment) – 0.82
(daily close price) – 0.63

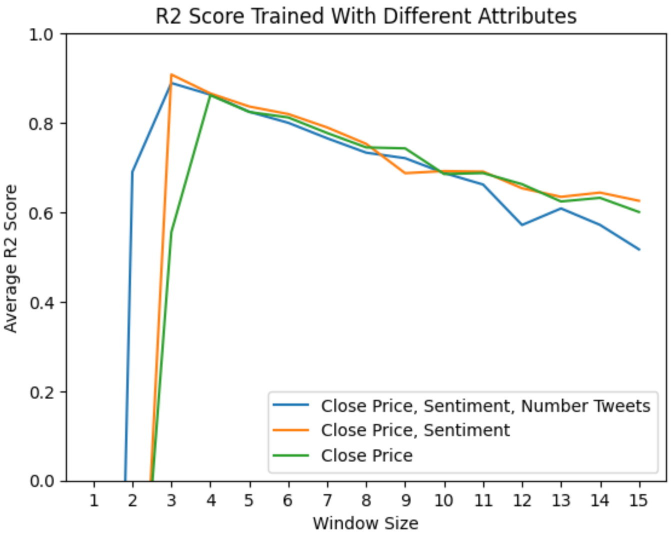

Fig. 4. R2 Score Trained With Different Attributes

Comparison for Mean Squared Error, Root Mean Squared Error, and Mean Absolute Error are as follows:

Mean Squared Error (3-day window size)
(daily close price, sentiment, # of tweets): 2,779,464.57
(daily close price, sentiment): 1,781,608.41
(daily close price): 3,571,994.47

Root Mean Squared Error (3-day window size)
(daily close price, sentiment, # of tweets): 1,667.17
(daily close price, sentiment): 1,334.77
(daily close price): 1,889.97

Mean Absolute Error (3-day window size)
(daily close price, sentiment, # of tweets): 1,130.61
(daily close price, sentiment): 996.44
(daily close price): 1,500.01

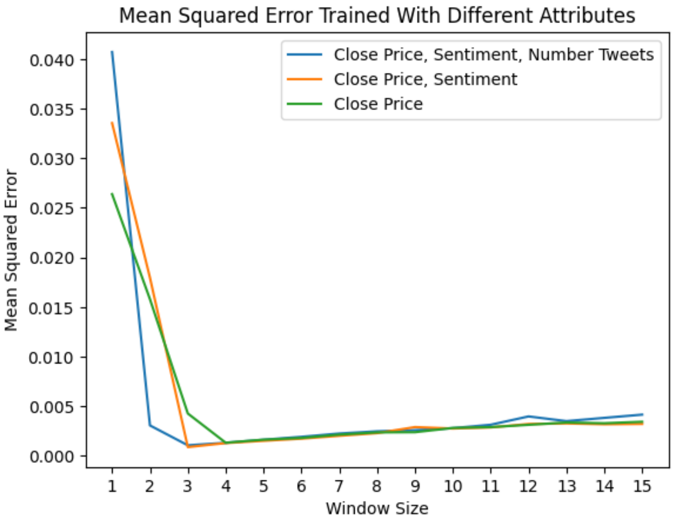

Fig. 5. Mean Squared Error Trained With Different Attributes

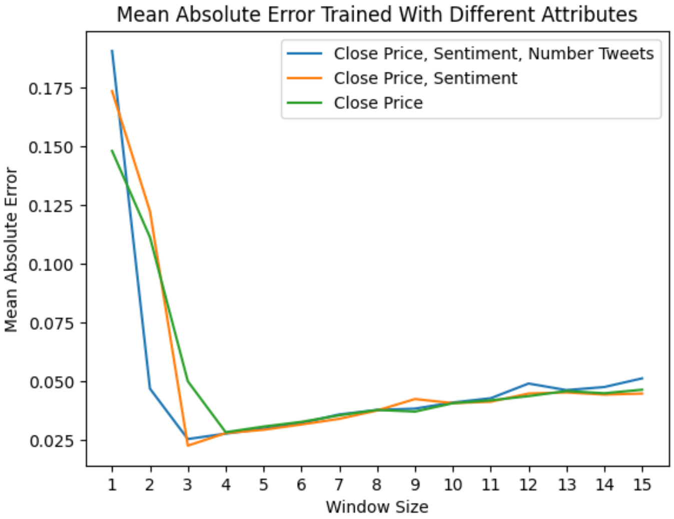

Fig. 6. Mean Absolute Error Trained With Different Attributes

VII. Conclusion

We conclude that price paired with sentiment and a 3-day window size is optimal for training an LSTM that can accurately predict the price of Bitcoin. Although inclusion of the volume of tweets per day in training our LSTM model did many times produce better accuracies than training with just the close price alone, better accuracy was achieved by training with only daily sentiment and close price. The hybrid approach of incorporating lexicon sentiment scores that are fed into LSTMs to train on, did prove successful and the optimal model trained produced a high enough accuracy that could be successfully used as a guide for making decisions on trading Bitcoin.

It would be interesting to see further side by side comparisons of LSTM models trained with only English tweets and models trained with multiple languages as were included in the model that we trained. Perhaps future experimentation will find different techniques for incorporating the daily volume of tweets to achieve higher accuracy in training models.

References

[1] Birjali, M., Kasri, M., Beni-Hssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and Trends. Knowledge-Based Systems, 226, 107134.

[2] Critien, J. V., Gatt, A., Ellul, J. (2022). Bitcoin price change and trend prediction through Twitter sentiment and Data Volume. Financial Innovation, 8(1).

[3] Perdana, R. S., Pinandito, A. (2018). Combining Likes-Retweet Analysis and Naive Bayes Classifier within Twitter for Sentiment Analysis. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(1-8), 41–46.

[4] Bitcoin Project. ”How does Bitcoin work?,” Bitcoin.org, https://bitcoin.org/en/how-it-works

[5] Amazon Web Services. ”What Is Sentiment Analysis?,” AWS, https://aws.amazon.com/what-is/sentiment-analysis/

[6] U. Maqsood, F. Y. Khuhawar, S. Talpur, F. H. Jaskani, (2022) ”Twitter Mining based Forecasting of Cryptocurrency using Sentiment Analysis of Tweets”

[7] O. Sattarov, H. S. Jeon, R. Oh, J. D. Lee, (2020) ”Forecasting Bitcoin Price Fluctuation by Twitter Sentiment Analysis”