

Feature-Rich Long-term Bitcoin Trading Assistant

Jatin Nainani^{1*}, Nirman Taterh^{1*}, Md. Ausaf Rashid¹, Ankit Khivasara^{1a}

¹K. J. Somaiya College of Engineering

^aCorresponding Author

*These authors have contributed equally to this work

Abstract—For a long time predicting, studying and analyzing financial indices has been of major interest for the financial community. Recently, there has been a growing interest in the Deep-Learning community to make use of reinforcement learning which has surpassed many of the previous benchmarks in a lot of fields. Our method provides a feature rich environment for the reinforcement learning agent to work on. The aim is to provide long term profits to the user so, we took into consideration the most reliable technical indicators. We have also developed a custom indicator which would provide better insights of the Bitcoin market to the user. The Bitcoin market follows the emotions and sentiments of the traders, so another element of our trading environment is the overall daily Sentiment Score of the market on Twitter. The agent is tested for a period of 685 days which also included the volatile period of Covid-19. It has been capable of providing reliable recommendations which give an average profit of about 69%. Finally, the agent is also capable of suggesting the optimal actions to the user through a website. Users on the website can also access the visualizations of the indicators to help fortify their decisions.

Keywords: Bitcoin, Cryptocurrency, Reinforcement Learning, Sentiment Analysis, Technical Analysis, Trend trading

I. INTRODUCTION

Cryptocurrency is a virtual currency which is secured by cryptography which uses cryptographic functions to facilitate financial transactions and form a system to store & transfer value. It leverages blockchain technology in order to be decentralized. Its most important feature is that it's not controlled by a central authority like the government or bank to uphold or maintain it. Bitcoin, which was released in 2009 is the first decentralized cryptocurrency.

While there are numerous solutions and trading bots that try to generate profits by trading and through short term patterns, there is a need for an investing assistant which tries to maximise profits by considering long term trends and real-time user sentiments.

It helps give the investor a perspective of the overall market situation and accordingly enables them to plan their investment and allocate an appropriate percentage of their financial portfolio to cryptocurrencies and Bitcoin.

It is important to note that more than 98% percentage of short-term traders do not make a profit net of fees. This is because it is complicated to time the market and make the right trading decision every time. This is because there are too many variables involved in predicting the short-term price of any asset, more so in the case of Bitcoin since it is one of the most volatile financial assets in the market.

The critical difference between a trading bot and an investing assistant is: In the case of the trading bot, the final trade decision is made by the bot itself, while in the investing assistant, the final investment decision is on the investor.

There are numerous challenges involved in building a robust and accurate reinforcement learning model that's a good enough representation of a cryptocurrency market's environment. Cryptocurrency price actions are extremely volatile and depend on a large amount of real-world and statistical factors. We will start by outlining the overall flow and architecture of the model. We then move forward to discuss the implementation of various features like Twitter sentiments, technical indicators and the custom index. Finally, all these features will be utilized in the reinforcement learning model. The model will be evaluated using robust evaluation metrics like entropy loss, policy loss and value loss. The final product is hosted on a website for the user to get the recommendations and gain insights from the visualizations.

II. LITERATURE SURVEY

Behavioural economists like Daniel Kahneman and Amos Tversky have established that decisions, including financial consequences, are impacted by the emotions and not just value alone [1]. Dolan et al.'s work in "Emotion, Cognition, and Behavior" further supports that emotions highly impact decision-making [2]. Insights from these researchers open up prospects to find advantages through various tools like sentiment analysis as it shows that demand for goods, and hence their price, may be impacted by more than its economic fundamentals. Panger et al. found that the Twitter sentiment also correlated with people's overall emotional state [3]. Few other researchers have studied the effectiveness of sentiment analysis of tweets. Pak et al. showed that separating tweets into either positive or negative or neutral categories will result in effective sentiment analysis [4]. Having established that emotions influence decisions, that social media can impact decisions, and that sentiment analysis of social media can accurately reflect the larger population's opinions towards something, Dennis et al. collected valence scores on tweets about companies within the S&P 500 and found that the scores correlated with stock prices [5]. De Jong et al. analyzed the minute-by-minute stock price as well as the tweet data for the 30 stocks within the DOW Jones Industrial Average and found that 87% of stock returns were influenced by the tweets [6]. Similar work has been done on cryptocurrencies after their introduction to see if such methods effectively predict cryp-

tocurrency price changes. In the paper by Stenqvist et al., the authors describe how they collected tweets related to Bitcoin and Bitcoin prices from May 11 to June 11 in 2017. First, tweets were cleaned of non-alphanumeric symbols (using “#” and “@” as examples of symbols removed). Then the tweets which were not relevant or were determined to be very influential were removed from the analysis. The authors then used VADER (the Valence Aware Dictionary and Sentiment Reasoner) to finally analyze the sentiment of each tweet and hence classify it as negative, neutral, or positive. Only tweets that could be considered positive or negative were kept in the final analysis [7]. The sentiment analysis done in this project builds off everything above but is unique, and we solve the problem of giving the tweets a weight.

Isaac et al. [8] used 10 second and 10-minute BTC price data from OKcoin and Coinbase API to accurately predict the bitcoin prices for short term periods. They focused on 25 features relating to the Bitcoin price and payment network over the course of five years. With this, training was done for Binomial GLM, SVM and Random Forests and these were compared with each other through metrics like sensitivity, specificity, accuracy and precision. The random forest model was able to beat all the models in all metrics except precision, where Binomial GLM bested it.

Zhengyao et al. [9] used a trading period of 30 minutes. Their primary focus was on portfolio management rather than price prediction, and the trading experiments were done in the exchange Poloniex. The Reinforcement Learning framework used a deterministic policy gradient algorithm. Agent was the software portfolio manager performing trading actions in a financial market environment. The reward function aimed to maximize the average logarithmic cumulative return R . For policy functions, three different methods were tested. CNN, RNN and LSTM. Performance metrics were: Accumulative portfolio value, Sharpe ratio. However with a fairly high commission rate of around 0.25% in the backtests, the framework was able to achieve at least 4-fold returns in 50 days.

Otabek et al. [10] created a recommendation system for cryptocurrency trading with Deep Reinforcement Learning. Data was taken from cryptodatadownload and focused on three currencies Bitcoin (BTC), Litecoin (LTC), and Ethereum (ETH). The environment was responsible for accounting for stock assets, money management, model monitoring, buying and holding or selling stocks and finally calculating the reward for actions taken. While the agent ran every day, constantly observing the environment to then choose an action with the policies learnt during the training. The environment monitoring, determining actions with policies, computing the gradient, recording and calculating rewards with discounted rewards, and updating the systems’ networks with gradients are all jobs that the agent is responsible for. The experiment on Bitcoin via DRL application shows that the investor got 14.4% net profits within one month. Similarly, tests when carried out on Ethereum and Litecoin also finished with 74% and 41% profit, respectively.

Fengrui et al. [11] regarded the transaction process as ac-

tions. The returns were regarded as awards and prices are considered to be states to align with the idea of reinforcement learning. A Deep Reinforcement Learning Algorithm - Proximal Policy Optimization (PPO) was used for high-frequency bitcoin trading. They first compared the results between advanced machine learning algorithms like support vector machine (SVM), multi-layer perceptron (MLP), long short-term memory (LSTM), Transformers and temporal convolutional network (TCN), by applying them to real-time bitcoin price and the experimental results demonstrated that LSTM outperforms. It was then decided to use LSTM as the policy for the PPO algorithm. The approach was able to trade the bitcoins in simulated environment paired with synchronous data and obtained a 31.67% more return than that of the best benchmark, improving the benchmark by 12.75%.

Gabriel et al. [12] created an agent that learns to trade the XB-TUSD (Bitcoin versus US Dollars) perpetual swap derivatives contract on BitMEX on an intraday basis. The cryptocurrency agent realized a total return of 350%, net transaction costs over five years, 71% of which is down to funding profit. The annualized information ratio that it achieves is 1.46. The echo state network provides a scalable and robust feature space representation.

Joonbum et al. [13] developed an action-specialized expert model designed specifically for each reinforcement learning action: buy, hold, and sell. Models were constructed by examining and defining different reward values that correlate with every action under some specific conditions, and the investment behaviour is reflected with each expert model. To verify the performance of this technique, the profits of the proposed system are compared to the profits of the common ensemble and single trading systems. In addition, sensitivity was checked with three varying reward functions: profits and Sharpe ratio and the Sortino ratio. All the experiments had been conducted with Hang Seng Index, S&P500 and Eurostoxx50 data. The model was 39.1% and 21.6% more effective than the single and common ensemble models, respectively. We can conclude a few points from the above literature survey. There is a distinct lack of a consistent environment, leading to some really restrictive while others are too free and ill-defined. Most of the agents are restricted to a single type of market. The variety of preprocessing techniques used led to the question of whether the improvement in the metric was the result of the model or the data fed. Most models are treated as a complete black box with a lot of hyperparameter tuning. Perhaps some form of explainable AI might find some use here to convince the investors and help them understand on what basis our model recommends actions.

III. APPROACH

A. Block Diagram

The block diagram is shown in Figure 1. It describes the flow of the project.

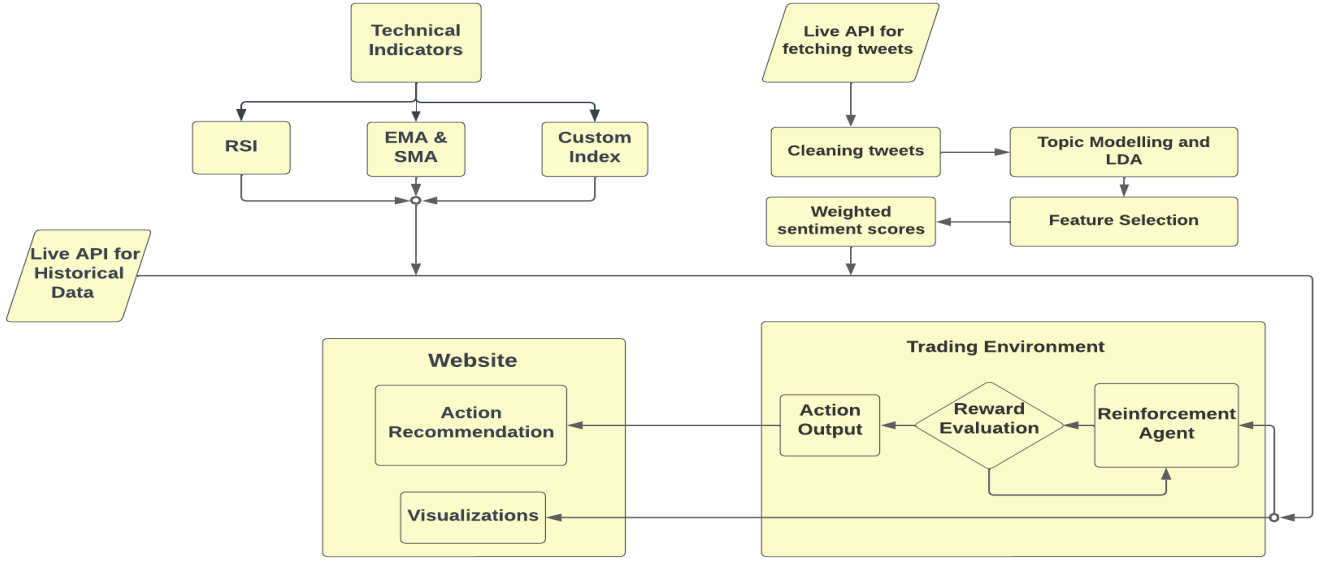


Fig. 1: Block Diagram

B. Data Description

1) *Bitcoin Historical Data:* We obtained the data on past prices from Cryptocompare as it regularly reviews exchanges, carefully monitoring the market to deliver the most accurate pricing indices. Bitcoin prices are known to be volatile, and the cryptocurrency market is still evolving. The main difference between BTC prices and usual stock prices is that the BTC prices change on a much larger scale than the local currencies. The overall data collection period is from January 2016 to April 2021 on a daily fidelity which gives us a total of 1918 values. This dataset consists of six different attributes, namely, Date, High, Low, Open, Close, and Volume. High and Low signifies the highest and lowest price at which Bitcoin was traded for that day. Close signifies the last recorded traded price of Bitcoin for that day. Volume here refers to the Bitcoin trading volume, which indicates how many Bitcoins are being bought and sold on specific exchanges on the day. Rows like High and Low are irrelevant for our analysis and can be hence dropped later.

2) *Bitcoin related Tweets:* Twitter is a minefield of information due to the volume and variety of users. This has resulted in Crypto Twitter being significantly influential on cryptocurrency trading decisions. Thus, we collected tweets to perform sentiment analysis on. We used the Twitter API to query tweets by applying to the developer role. However, the rate limit is limited to 10 calls per second, so fetching tweets for a period of 2016/1 to 2021/04 was time-consuming. ‘bitcoin,’ ‘BTC,’ ‘#Bitcoin,’ and ‘#BTC’ were used as the keyword filters for fetching tweets. Also, all tweets were fetched in the English language. This gave us a total of 4,265,266 tweets to work with. The data contains many columns, including the date of the tweet, username, tweet

location, tweet ID, number of replies, number of retweets, number of favorites, the text of the tweet, list of mentions, hashtags, and permalinks. Some of the columns are irrelevant to our analysis and will be dropped later.

C. Technical Analysis

1) Trend Indicators:

a) *Simple Moving Average:* SMA is just the mean price over the specified period. The mean average is called “moving” because it is plotted on the chart bar-wise, which forms a line that moves along the chart with respect to the average value. If the simple moving average points up, this means that Bitcoin’s price is increasing. Down pointing indicates a decrease in the Bitcoin’s price. Also, a longer time frame for moving average gives a much smoother simple moving average. A moving average with a shorter-term is more volatile, but the reading is much closer to the source data.

The shorter the time span(n) used to create the average, the more sensitive it will be to price changes. The longer the time span(n), the less sensitive the average will be to the price changes because more lag is introduced between the SMA and the Bitcoin prices. When it comes to our project, we aim at maximizing profits in the long run. So, we took a longer time span of 21 weeks to calculate our SMA to make sure that it is not sensitive to all the little changes in price but also does not overlook the major price changes. A time span of 21 weeks gives us a very balanced value of SMA, which is well suited for our project.

$$SMA = \frac{(A_1 + A_2 + \dots + A_n)}{n} \quad (1)$$

A_n is the price of Bitcoin at period n
 n is the number of total periods

b) *Exponential Moving Average*: Exponential Moving Average is comparable to SMA, gauging trend direction over a time period. On the other hand, SMA calculates average of price data whereas the EMA applies more weight to current data. Hence the EMA follows prices more closely than corresponding SMA for the same.

EMA is used to determine trend direction and to trade in that direction. When the EMA rises buying when prices dip near the EMA is a good move. When the EMA falls, selling when prices rally towards EMA is a good choice.

The formula for EMA at period n is:

$$EMA_n = \left(A_n * \left(\frac{S}{1+n} \right) \right) + EMA_{n-1} * \left(1 - \left(\frac{S}{1+n} \right) \right) \quad (2)$$

where, A_n is the price of Bitcoin at period n
 S is the smoothing factor
 n is the number of total periods

Although there are many options to choose from when considering the smoothing factor, we set up a value of 2. A value of 2 gives more credibility to recent data points available. The further we increase the smoothing factor value, the greater influence more recent data has on the moving average. So on testing for different values, we found the value of 2 to give just enough credibility to recent data. The EMA has a shorter delay than the SMA within the same period. Hence we decided to go for a period of 20 weeks for EMA calculations.

c) *Relative Strength Index*: The RSI is a momentum oscillator that measures the change of price movements. The RSI oscillates between zero and 100. Traditionally the RSI is considered to be overbought when its value is above 70 and oversold when the value is below 30. RSI is also helpful to identify the general trend.

Value of RSI is calculated with a two steps that starts with the below formula:

$$RS = \frac{Avg.Gain}{Avg.Loss} \quad (3)$$

$$RSI = 100 - \frac{100}{1 + RS} \quad (4)$$

The average percentage of the gain or loss during the look-back period is the average gain or loss used in the computation. The formula uses an overall positive value for the average loss. The periods of losses in price are counted as 0 in the computations of average gain, and the periods of increase in price are counted as 0 during the calculation of average loss. Usually 14 periods are used to calculate the initial RSI value. Once 14 periods of data is available, the second part of the RSI formula is to be done which smooths the results.

2) *Custom Indicator - BMSB*: BMSB index is a Bollinger band that helps us understand the current valuation of bitcoin against USD, vis., undervalued, overvalued, or fair valued. It ranges from -100 to 100, with the negative side corresponding

to extreme undervaluation and the positive side indicating extreme overvaluation with respect to recent price movements. Bull Market Support band is a type of Bollinger band that consists of 20w SMA and 21w EMA.

Usually, when the price of Bitcoin falls below BMSB, there is a bearish momentum in the market, and when the price of Bitcoin holds support or stays above the BMSB, it indicates the onset of a bull market. That being said, being over-extension (+70 to +100 range) does not necessarily imply a bull run, and under-valuation (-70 to -100 range) does not necessarily imply a bearish movement. In fact, it can indicate the end of a bull run and bear market, respectively. Therefore, in our project, when the BMSB index tends to -100, we show it as a buying opportunity, and when it tends to +100, we show it as a time to take profits.

a) *Algorithm*: Let us introduce a few of the crucial variables that will enable us to formulate the algorithm for the BMSB Index: We define Combined Average (μ) as the average of the current 21 weeks EMA and 20 weeks SMA. We define K_p as the Price Coefficient. The price coefficient decides the extent of the price range from which the Bitcoin price is considered to be normal. The normal price range varies from $\mu(1 - K_p)$ to $\mu(1 + K_p)$.

Algorithm 1: BMSB Indicator

Inputs: Price of Bitcoin (p), Exponential Moving Average (EMA), Simple Moving Average (SMA), Price Coefficient (K_P) and Scaling Coefficient (K_S)

Outputs: BMSB Index (I)

START

Combined Average (μ) = Average(SMA, EMA)

Upper Bound (B_U) = $\mu(1 - K_P)$

Lower Bound (B_L) = $\mu(1 + K_P)$

IF $p > B_L$ && $p < B_U$ **then**

$$I = \left(\frac{p - \mu}{\mu} \right) \frac{K_S}{K_P} \times 100$$

ELSE IF $p < B_L$ **then**

$$I = \left[\frac{p(1 - K_S)}{\mu(1 - K_P)} - 1 \right] \times 100$$

ELSE $p > B_U$

$$I = \left[1 - \frac{\mu(1 + K_P)(1 - K_S)}{p} \right] \times 100$$

end IF

STOP

K_s is defined as the scaling coefficient. The scaling coefficient decides the extent on the BMSB Index scale for the normal price range. It varies from $100(0-K_s)$ to $100(0+K_s)$ and corresponds to the $\mu(1 - K_p)$ to $\mu(1 + K_p)$ price range. We calculate the BMSB Index as shown in Algorithm 1.

D. Sentiment Analysis

1) *Preprocessing:*

a) *Cleaning*: After scraping the tweets, we had to drop some columns that we deemed irrelevant for the sentiment analysis. We kept the columns that we deemed useful, which include the date, username, number of replies, retweets, favorites, and the text of the tweet. Before we are able to start doing any form of sentiment analysis, the tweets collected have to be cleaned.

Sample tweet before cleaning:

“Bitcoin Climbs Above 11,316.7 Level, Up 6%
<https://yhoo.it/2YV6wKZ> #bitcoin #crypto #blockchain #btc
 #news #cryptocurrency pic.twitter.com/mPv3rd4kLn #eth #ltc
 #xrp”

These tweets contain a large amount of noise. Using regex expressions, all these noises were removed. Preprocessing is a very important aspect of sentiment analysis; if we were to pass these raw tweets to our analyzer, chances are it will perform very poorly and take up much more time

Sample Tweet after cleaning:

“Bitcoin Climbs Above Level Up”

Next, we set all the characters to lower cases and also removed stopwords in our tweets. Stop words are commonly used words such as “a,” “the,” and “as,” which provide no valuable information in Sentiment analysis. We made use of the Natural Language Toolkit(NLTK) to get rid of them. This was enough for VADER as it was specially tuned for social media content.

b) Topic Modeling and LDA for Ads and Spam handling:

In our first attempt on sentiment analysis, our sentiment analysis did not yield good results as it gave advertisement tweets a high positive sentiment score

E.g. “Free bitcoin faucet with extra bonuses Get free bitcoins now” : +0.8807 (Using Vader)

This is an issue as our prediction model was unable to differentiate between a useful tweet and ads. Hence, we identified and tagged these ads by using Latent Dirichlet Allocation (LDA) to cluster and identify possible tweet topics. This is useful as it allows us to gain insight into the type of Bitcoin topics people discuss on Twitter and, within these topics, to identify an ad topic cluster. The reason for choosing LDA is that it is most effective in generalizing to new documents easily.

To train our LDA model, the following parameters were varied:

- Corpus: Bag of Words generated from the pre-processed tweets
- id2word: Dictionary mapping of word ID to words in tweets

- Number of topics: The number of topic clusters to be generated.

The model parameters were varied by using different numbers of tweets and the number of topic clusters. Each tweet is presented as a distribution over topics, and every topic is represented as a distribution over words. To evaluate our model parameters, we used the following:

- Visual analysis of clusters with pyLDAvis:

Through the visual analysis of 10, 6,3,20 topic clusters, we found that 20 topic clusters gave the best result. In the case of 3 topic clusters, none of the clusters overlap in the Intertopic Distance Map indicating the topics were too broad and vague to use them to identify an ad cluster. The histogram on the right represents the most probable topic word for that cluster of tweets. Hence, we increased the cluster size until we found that our clusters gave us unique, most probable topic clusters. As seen here in Fig 2, we can see that the various possible Bitcoin topics are: Bitcoin, market, price, BTC, mining, analyst, new, day, transaction, news, time, analysis, blockchain, dont, eth, payment, exchange, million, social media, resistance. Based on the cluster topics, we suspected that the unusual 'don't' cluster(Fig 3) (cluster 14 on pyLDavis, topic 13) is likely to be the ad cluster.

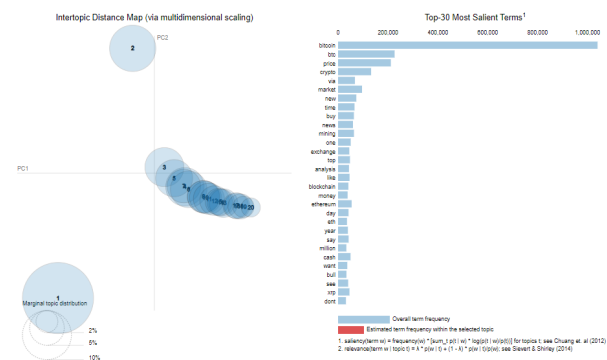


Fig. 2: *pyLDavis* interactive visualization of Topics generated from LDA. Model parameters: # of tweets used =All, # of clusters = 20.

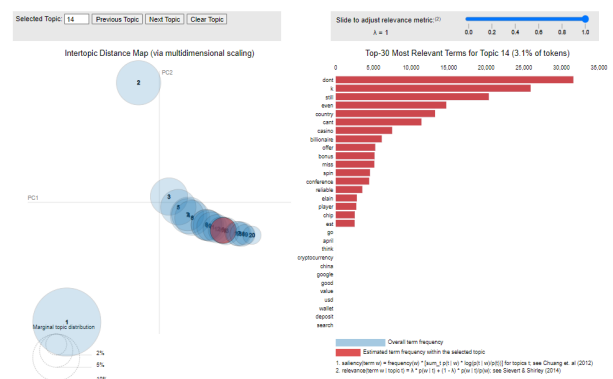


Fig. 3: *pyLDAvis* for suspected ad topic cluster

- Running new unseen documents through our model

Some examples of Ad tweets include, 'is airdropping den to participants sign up here'; 'claim daily bitcoins at for free'; 'bitcoin black v bitcoin get free coins value'; 'get free bitcoin'; 'claim free bitcoin now link moon bitcoin is a bitcoin'

To confirm our hypothesis, we used tweets that we know are ads (above) and ran it through the LDA model. We wanted to find the model parameters that will result in all 5 tweets' LDA results returning topic 13 is the most probable topic. Hence, we are able to establish that topic 13 is our ad cluster and our ideal model parameters are to use all the tweets with 20 topic clusters. To tag the ad tweets, we used all tweets and ran it through our LDA model with 20 clusters, and filtered out all Topic 13 tokenized tweets and extracted the words to form an "ad word list" which is used in the preprocessing stage to tag the ad tweets.

2) *Sentiment Analysis*: We then compare between VADER and TextBlob, which are lexicon-based approaches for sentiment analysis. We selected a few tweets to compare if VADER or TextBlob performs better. We noticed that the sentiment score for advertisements and spam tweets was mostly quite positive. Hence, we filtered out such tweets, which has been described in detail in the Topic Handling section. From the tests we conducted on the selected few tweets, VADER works better with things like slang, emojis, etc., which is quite prevalent on Twitter, and it is also more sensitive than TextBlob. So, we choose to go with VADER.

After the sentiment of each Tweet is computed, we take the daily average sentiment, but taking the average sentiment results in the loss of information. For example, a prominent user's positive tweet (score of +1.0) may have greater influence over public opinion as compared to a common users' negative tweet (score of 0). To address this issue, we decided to use weighted Sentiment analysis scores.

Feature Selection and Weighted Sentiment Score: Our collection of features consists of Tweets text, Time of Tweet, Number of Retweets, Number of Replies, Number of Favorites, Advertisement score, Tweet Volume, and Sentiment Analysis Score (VADER). To prevent the loss of information by taking the average sentiment, we have generated weights to create weighted sentiment scores. Each tweet will have a weighted score. Table 1 shows the Weight Rules we have established.

The final weights equation is as follows:

$$\begin{aligned} \text{Weight} &= [\text{weight}(\text{prominent user}) + \text{weight}(\text{keyword}) + \\ &\text{weight}(\text{ \# of retweets}) + \text{weight}(\text{ \# of replies}) + \text{weight}(\text{ \# of} \\ &\text{favorites}) + \text{weight}(\text{time})] * \text{ad_score} \\ \text{Weighted Sentiment per tweet} &= \text{sentiment score of tweet} * \\ &\text{Weight} \\ \text{Weighted Sentiment per day} &= \text{sum}(\text{Weighted Sentiment} \\ &\text{per tweet}) / \text{\# of tweets in one day} \end{aligned}$$

To address the issue with VADER assigning ads high polarity scores, we multiply the ad_score to the other weights. As ad_score for ads are assigned as 0 and so Weighted Sentiment

will be 0. As the weighted sentiment of ads are now neutral, we have effectively filtered out the ads as it will not affect our model prediction. Hence these newly generated Weighted Sentiment Scores are then passed onto the RL model as a feature.

E. Reinforcement Learning

1) Environment:

a) *Positions*: Positions of the market describe the amount of Bitcoin a particular user holds at that moment. In our environment, we have considered long and short positions. These suggest the two potential directions of the price required to maximize profit. Going Short is a trading technique in which a trader borrows an asset in order to sell it, with the expectation that the price will continue to decline. In the event that the price does decline, the short seller will then buy the asset at this lower price in order to return it to the lender of the asset, making the difference in profit. Going Long is when an investor gains exposure to cryptocurrency with the expectation that prices will rise at a later date, meaning that the asset can be sold for a profit.

With regards to our project, the Long position wants to buy shares when prices are low and profit by sticking with them while their value is going up, and the Short position wants to sell shares with high value and use this value to buy shares at a lower value, keeping the difference as profit.

The environment assigns a value of 0 to the agent if the position is discovered to be Short and 1 if the position is discovered to be Long. When starting a new environment, the position of the user is considered short as default.

b) *Actions*: The reinforcement agent aims at maximizing profit by recommending the best course of action to the user. There are numerous actions that a user can perform in the trading market like Buy, Sell, Hold, Enter, Exit, etc. However, these actions just make things complicated with no real positive impact. In fact, they increase the learning time. Therefore there is no need to have numerous such actions, and only Sell and Buy actions are adequate to train an agent just as well. That is to say that our agent will always be recommending the user to either Buy or Sell to make sure it does not miss a single penny. The environment assigns a value of 0 to the agent if the recommended action is Sell and 1 if recommended action is Buy.

However, performing a trade on every interval does not produce reliable results for a real-life situation. To bridge the gap between simulation and reality and maximize long-term profits, the algorithm accepts the recommendation from the model and the user's current position in the market to provide the final verdict. The algorithm first decides between trading or holding. To do that, it refers to the position of the environment. When you make a short trade, you are selling a borrowed asset hoping that its price will go down and you can buy it back later for a profit. So our algorithm will recommend the user to Buy after the short trade is satisfied and Sell after the long trade is satisfied. If the above conditions are not satisfied, then no trade occurs (Holding) because our conditional logic

	Rule	Weight Scores
1	Prominent Users: We generated a list of prominent Cryptocurrency Twitter users. This list of Twitter users was found through recommendations online and friends who actively follow Bitcoin news	If the Tweet is made by user on the list, +1 to the weight score
2	Bitcoin Keyword List: We generated a list of keywords that are used in Bitcoin. If any of the words are used, it is likely that the tweet contains useful information.	Add weight +1 if at least one or more keywords are used.
3	Number of Retweets, Replies and Favourites: The number of retweets, replies and favourites are indicators of impact and reach. The larger the number, the greater the weight.	Added the round(log(no of retweets + replies + favourites)) to the weight score. As some tweets may have thousands of retweets while some have none, we took log to normalise the data.
4	Time: As our price data is based on the closing price, the tweets made nearer to closing price are more reflective of the current public opinion during the closing price. Hence, we used a naïve method of splitting the time period into half. Tweets made in the later half carry more weight.	For daily, if the tweet is made from 13:00 to 23:59 and for hourly, if the tweet is made in the last 30 mins, +1 is added to the weights.
5	Advertisement Score: We needed a way to filter out the ad tweets. In the data pre-processing stage, based on the LDA ad cluster's results, we are able to tag each tweet with an ad score.	The Advertisement Score weight is the ad score obtained during data pre-processing.

TABLE I: *Weight Rules we have established*

suggests that profit can be maximized just by holding in such a case.

Algorithm 2: Final Recommendation

Inputs: Recommended Action, Current Position

Outputs: Final Recommendation

IF Recommended Action = Buy && Current Position = Short **then**

 Final Recommendation = BUY

ELSE IF Recommended Action = Sell && Current Position = Short **then**

 Final Recommendation = SELL

ELSE,

 Final Recommendation = HOLD

end IF

STOP

In the end, our algorithm can now recommend three independent actions depending on the situation of the market, namely, Buy, Hold or Sell.

2) Learning Algorithm:

a) Model Inputs: The reinforcement learning model predicts an action by taking the current state of the environment as input. We provide all the hand crafted features developed till now for the model to optimize its learning in Table 2.

b) Model Specifications: The Asynchronous Advantage Actor-Critic method (A3C) has been very influential since the paper [14] was published. The algorithm combines a few key ideas:

- Update rule that works on fixed-lengths of experience and uses these to compute estimators and advantage function.
- Architectures that share layers between the policy and value function.
- Asynchronous updates.

AI researchers questioned whether the asynchrony of A3C led to improved performance or if it was just an implementation detail that allowed for faster training with a CPU-based implementation. Synchronous A2C performs better than asynchronous implementations, and we have not seen any evidence that the noise introduced by asynchrony provides any performance benefit. The host of this algorithm was the ‘stable baselines 3’ library on Python. The library offers multiple learning policies depending on the type of input data.

We have used the MLP (Multi-Layer Perceptron) policy which acts as a base policy class for actor-critic networks allowing both policy and value prediction. It provides the learning algorithm with all the base parameters like the learning rate scheduler, activation functions, normalization, feature extraction class, and much more. This model is learned with a sliding window approach.

Feature	Definition
Closing Prices	The closing price is the value of the last transacted price of Bitcoin for the day.
Volume	The Bitcoin trading volume indicates how many Bitcoins are being bought and sold on specific exchanges.
Simple Moving Average	The simple moving average (SMA) is a way to calculate the average price of a cryptocurrency for a predetermined set of data and help to track the trend direction
Exponential Moving Average	EMAs are similar to SMAs. EMA involves a more complex calculation to place more weight on the most recent price changes. EMA are more sensitive than SMA
RSI	The RSI index measures momentum and oscillates on a scale between 0 and 100 and gives a gist of the market value of Bitcoin.
Bull Market Support Band	Our custom index which indicates whether Bitcoin is currently undervalued or overvalued in the current market. It ranges between -100 to 100.
Weighted Sentiment Scores	The weighted sentiment score indicates how negative or positive the overall sentiment for trading of Bitcoin has been for that day

TABLE II: Features used, along with definitions

IV. EXPERIMENTS

A. Evaluation Criteria

a) *Entropy Loss*: RL agents are known to be much more unstable to train in comparison to other types of machine learning algorithms. One of the ways that a reinforcement learning algorithm can under-perform is by becoming stuck during training on a strategy that is neither a good solution nor the absolute worst solution. This phenomenon is referred to as reaching a “local minimum”.

In reinforcement learning, a similar problem can occur if the agent discovers a strategy with better rewards than it was receiving when it first started but far from the optimally correct strategy. This might often manifest as the agent deciding to take a single move, over and over. Entropy loss, we believe first discussed in a 1991 paper, is an additional loss parameter that can be added to a model to help with local minima in the same way that momentum might and to provide encouragement for the agent to take a variety of moves and explore the set of strategies more.

To solve this, we will encourage the agent to vary its moves by adding a new loss parameter based on the entropy of its predicted moves. The equation for entropy here is described in Equation 5.

$$H(x) = \sum_{i=1}^n P(x_i) \log_e P(x_i) \quad (5)$$

This encourages the network to only make strong predictions if it is highly confident in them. Entropy loss is a clever and simple mechanism to encourage the agent to explore

by providing a loss parameter that teaches the network to avoid very confident predictions. As the distribution of the predictions becomes more spread out, the network will sample those moves more often and learn that they can lead to greater rewards.

B. Training of proposed model

The training of the RL agent was done on the Google Colab GPU, with *NVIDIA-SMI 460.67 Driver Version: 460.32.03 CUDA Version: 11.2*. The model was tested for multiple scenarios, and the below gave the most consistent results. The time steps were set to be 50000, which is equivalent to the number of iterations of training. The 1918 data instances were present, and 1233 were used for training the model with a window size of 30. The remaining 685 instances were used for testing.

C. Results and discussion

After training for 50000 time steps, the entropy loss of 0.0209 was obtained. The learning rate was optimized at 0.007.

Finally, the testing was performed on the latest 685 days in the data. Specifically, the testing period started from 17th May 2019 and ended on 1st April 2021. This period is highly significant as it captures the peak points of Covid-19 and its effects on the market. Given these extreme scenarios, the model was able to generate a total reward of 39,412 and a profit value of 1.69486, which is equivalent to a 69.486% increase over 685 days.

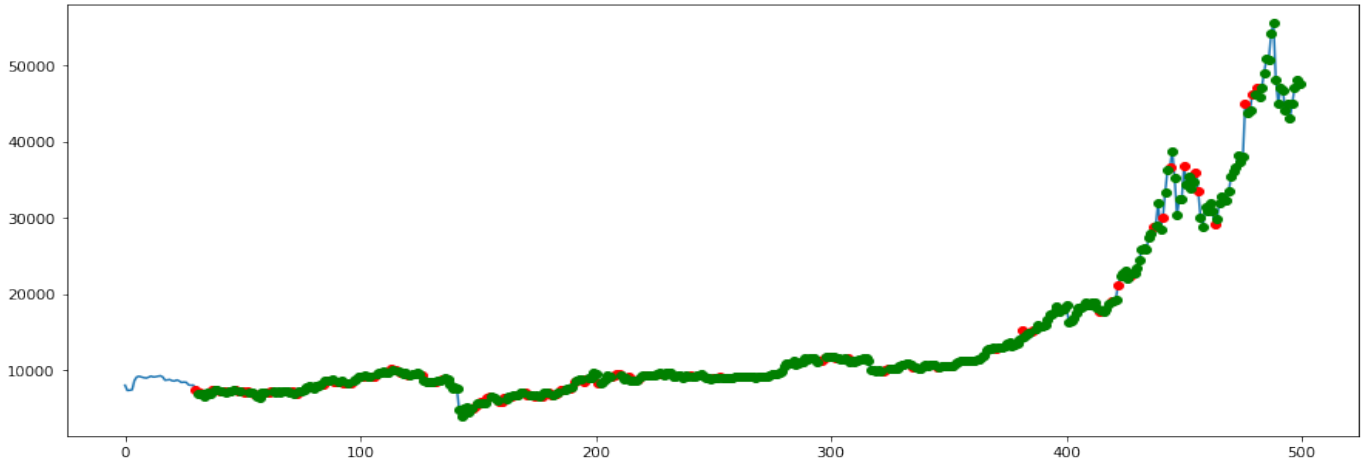


Fig. 4: Graph of recommendations

The green dots in the Fig. 4 indicate a long position, whereas the red dots indicate a short position. As discussed before, the agent itself recommends only Buy or Sell with the aim to capture every penny. Our algorithm then reviews the current situation/position of the user in the market and either recommend the agent action or asks them to hold their assets.

V. CONCLUSION

Investing in the crypto market is a tedious task and requires a lot of effort to make good trading strategies. Moreover, analyzing real-time data is a more difficult task. Traders make several trading decisions and keep on learning from them. During this process, they improve their decision-making skills. This is the major idea behind implementing a Reinforcement Learning agent, which trains against the environment to adapt and make better decisions. Thus, we can say that trading can be approached as a Reinforcement Learning Problem.

In the trading scene, experts use sophisticated tools to analyze the trend of prices better. Our plan was to allow the learning agent to learn in such a feature-rich learning environment. Specifically, we have used some of the most popular and tested long-term trend indicators like SMA, RSI, and EMA. We have also crafted our technical indicator, which is used by the agent to find opportunities for trade action. Another huge factor that impacts the flow of prices is the overall sentiment of the market. By including this factor in the environment, the agent was better able to understand the market situation.

The value of the added features was demonstrated as without these, the average profit of the model was not always positive. In addition to the features, we were able to get consistently positive profits over a long period of testing. With these hand-crafted features, our model was able to analyze the situation of the user and market and recommend smart decisions to the user.

We were able to provide an interface to the user for their better understanding of the current market situation through visualizations of the important indicators and sentiment scores. The interface was able to host the agent to provide its final recommendation to the user. The results show great potential for the approach, but the bitcoin markets are quite large, complex and volatile, so the modelling of this environment still presents a lot of challenges.

VI. FURTHER WORK

Many experts in the field of cryptocurrency and stock trading utilize trend analysis by identifying the popular patterns in the price action. Each of these patterns helps in the analysis of the price changes to occur in the future. The ascending and descending triangle pattern, as shown in the figure, leads to a continuity in the trend of prices. Another popular pattern experts use is the head and shoulders pattern shown in figure 2. This pattern is a strong sign of a reversal of the trend of the prices. Because of their impact, the recognition of these patterns becomes of most significance. A reliable recognition system will be sure to aid the reinforcement agent in better understanding the trading space and making smarter decisions. However, current pattern matching algorithms fail to work for different pattern spans. This problem is highly significant as even though we have an idea of what pattern we are looking for, most patterns occur at significantly different intervals. Nevertheless, there is still a promise of research in this department to take the project to the next level. In recent times, communities on Reddit have had a significant impact on the prices of cryptocurrencies. So another step forward for the project would be to include sentiments from specific subReddits to our sentiment scores. This will involve assigning weights to communities and comparing them with the scores from Twitter.

ACKNOWLEDGMENTS

We would like to express my very great appreciation to Ankit Khivasara Sir for their valuable and constructive suggestions during the planning and development of this project. Their willingness to give their time so generously has been very much appreciated.

We would also like to extend our gratitude to the entire team of OpenAI as well as that of Stable Baselines3 for implementing and maintaining their framework, without which this project would not be possible.

REFERENCES

- [1] Daniel Kahneman and Amos Tversky. “Prospect theory: An analysis of decision under risk”. In: *Handbook of the fundamentals of financial decision making: Part I*. World Scientific, 2013, pp. 99–127.
- [2] Raymond J Dolan. “Emotion, cognition, and behavior”. In: *science* 298.5596 (2002), pp. 1191–1194.
- [3] Galen Thomas Panger. *Emotion in social media*. University of California, Berkeley, 2017.
- [4] Alexander Pak and Patrick Paroubek. “Twitter as a corpus for sentiment analysis and opinion mining”. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10)*. 2010.
- [5] Hongkee Sul, Alan R Dennis, and Lingyao Ivy Yuan. “Trading on twitter: The financial information content of emotion in social media”. In: *2014 47th Hawaii International Conference on System Sciences*. IEEE. 2014, pp. 806–815.
- [6] Pieter de Jong, Sherif Elfayoumy, and Oliver Schnusenberg. “From returns to tweets and back: an investigation of the stocks in the Dow Jones Industrial Average”. In: *Journal of Behavioral Finance* 18.1 (2017), pp. 54–64.
- [7] Evita Stenqvist and Jacob Lönnö. *Predicting Bitcoin price fluctuation with Twitter sentiment analysis*. 2017.
- [8] Isaac Madan, Shaurya Saluja, and Aojia Zhao. “Automated bitcoin trading via machine learning algorithms”. In: *URL: <http://cs229.stanford.edu/proj2014/Isaac%20Madan>* 20 (2015).
- [9] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. “A deep reinforcement learning framework for the financial portfolio management problem”. In: *arXiv preprint arXiv:1706.10059* (2017).
- [10] Otabek Sattarov et al. “Recommending cryptocurrency trading points with deep reinforcement learning approach”. In: *Applied Sciences* 10.4 (2020), p. 1506.
- [11] Fengrui Liu et al. “Bitcoin transaction strategy construction based on deep reinforcement learning”. In: *Applied Soft Computing* 113 (2021), p. 107952.
- [12] Gabriel Borrageiro, Nick Firoozye, and Paolo Barucca. “The Recurrent Reinforcement Learning Crypto Agent”. In: *arXiv preprint arXiv:2201.04699* (2022).
- [13] JoonBum Leem and Ha Young Kim. “Action-specialized expert ensemble trading system with extended discrete action space using deep reinforcement learning”. In: *PloS one* 15.7 (2020), e0236178.
- [14] Volodymyr Mnih et al. “Asynchronous methods for deep reinforcement learning”. In: *International conference on machine learning*. PMLR. 2016, pp. 1928–1937.