IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 August 2022, accepted 5 September 2022, date of publication 8 September 2022, date of current version 16 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3205129

RESEARCH ARTICLE

Twitter Attribute Classification With Q-Learning on Bitcoin Price Prediction

SATTAROV OTABEK AND JAEYOUNG CHOI, (Member, IEEE)
School of Computing, Gachon University, Seongnam 13120, Republic of Korea

Corresponding author: Jaeyoung Choi (jychoi19@gachon.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) under Grant 2022R1C1C1004590 and in part by the Gachon University Research Fund of 2021 under Grant GCU-202109970001.

ABSTRACT Bitcoin price prediction based on people's opinions on Twitter usually requires millions of tweets, using different text mining techniques, and developing a machine learning model to perform the prediction. These attempts lead to the employment of a significant amount of computer power, central processing unit (CPU) utilization, random-access memory (RAM) usage, and time. To address this issue, in this paper, we consider a classification of tweet attributes that effects on price changes and computer resource usage levels while obtaining an accurate price prediction. To classify tweet attributes having a high effect on price movement, we collect all Bitcoin-related tweets posted in a certain period and divide them into four categories based on the following tweet attributes: (*i*) the number of followers of the tweet poster, (*ii*) the number of comments on the tweet, (*iii*) the number of likes, and (*iv*) the number of retweets. We separately train and test by using the Q-learning model with the above four categorized sets of tweets and find the best accurate prediction among them. We compare our approach with a classic approach where all Bitcoin-related tweets are used as input data for the model, by analyzing the CPU workloads, RAM usage, memory, time, and prediction accuracy. The results show that tweets posted by users with the most followers have the most influence on a future price, and their utilization leads to spending 80% less time, 88.8% less CPU consumption, and 12.5% more accurate predictions compared with the classic approach.

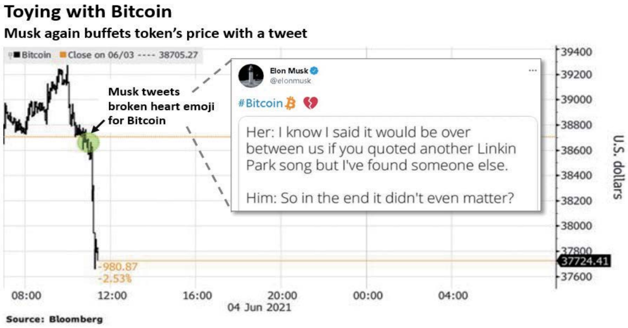
INDEX TERMS Bitcoin price prediction, Q-learning, reinforcement learning, tweet attributes, twitter sentiment analysis.

I. INTRODUCTION

Earlier stock market forecasting research relied on past stock values [1], [2], [3]. Most studies have discovered that analyzing previous prices is not sufficient to anticipate stock market changes because stock market prices are highly volatile. According to the efficient market hypothesis [4], financial market movements are influenced by news, current events, and product releases, all of which have a substantial impact on a company's stock value. As large stock market, Bitcoin has no central controlling authority and is regulated solely by the public. As a result, Bitcoin is viewed as a volatile cryptocurrency and its value is influencing by public ideas. According to the analysis of Kristoufek [5], several significant reductions have occured in the Bitcoin exchange rate and in its price during dramatic events in China. Another study conducted by the American Institute for Economic Research [6] shows that Bitcoin prices fluctuated substantially between 2016 and 2017 as a result of global news and emotions.

Owing to the rise of social media, information regarding popular sentiments has become more accessible. Social media is becoming an ideal medium for sharing public mood on any issue, and it has a significant effect on general public opinion. Twitter, a social networking service (SNS), has recently received significant academic attention. Twitter is a real-time micro-blogging service that allows users to follow and comment on others' thoughts and views [8]. Approximately 140 million tweets are sent to more than a million people daily. Each tweet is 140-characters long and expresses the public view of a particular issue. Information

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.



Her: I know I said it would be over between us if you quoted another Linkin Park song but I've found someone else.

Him: So in the end it didn't even matter?

FIGURE 1. Example of Elon Musk's (Tesla CEO) tweets affecting the Bitcoin price [7].

derived from tweets is valuable for forecasting [9]. Over a million Bitcoin-related tweets are available to researchers for processing and application in the field of predicting future Bitcoin prices. Processing a large amount of Bitcoin-related tweets normally consumes a high level of computer resources (CPU, RAM, memory) and time [31], [32], [33], [34]. Most of the previous works is focused on how to reduce the resource, so maximizing the prediction result at the same time is not considered. However, tweets written by an expert, public figure, or celebrity will become viral, with many replies, likes, and retweets. Tweets with few replies, likes, or retweets are unlikely to become viral because they are likely to circulate mainly among close friends. Consequently, viral tweets are expected to have a greater influence on price changes than other tweets. If we can separate tweets with the highest impact on future price changes from less important tweets, it gives the possibility to employ less computer resources usage while still obtaining accurate forecasts.

Hence, different from the previous approaches, in this study, we analyze how Bitcoin-related information on Twitter affects the actual Bitcoin price by considering four main attributes: (*i*) **the number of followers of the poster,** (*ii*) **the number of comments on a tweet,** (*iii*) **the number of likes,** and (*iv*) **the number of retweets.** For this, we gather all Bitcoin-related tweets within a particular period and divide them into four groups based on their attributes. Since we use the sentiment information of tweets as a resource for the prediction, yet there was no particular guidance to inform the model in what condition of sentiments the price will increase or decrease. Therefore, we need an optimal policy to achieve valuable prediction accuracy. The model can improve its initial non-optimal policy by receiving good/bad rewards based on the prediction results. Considering the above, we choose to use a well-known Q-learning method to obtain the most valuable attribute to predict Bitcoin's future price. To the best of our knowledge, our work is the first study on the predictable range of tweet attributes involving the term ''Bitcoin'' on the future returns and volatility of Bitcoin.

After classifying which attribute is helpful to separate highly effective tweets to make a prediction, we compare our approach to the classic approaches in which all Bitcoin-related tweets are utilized without being attribute-filtered, by looking at the CPU workload, RAM utilization, memory usage, required time to complete the same task, and prediction accuracy.

We summarize our main contributions in more detail as follows:

(*a*) First, we study the predictive power of four main tweet attributes: number of tweet poster's followers, number of comments, number of likes, and number of retweets. We create four datasets consisting of tweets sorted according to the above attributes. Next, we extract the sentiment of each tweet. By making four separate predictions based on the datasets and evaluating the prediction results, we detect the most useful attribute for the Bitcoin price prediction.

(*b*) Second, we develop the predictive model based on the Q-learning algorithm. For this, we first consider a Markov Decision Process (MDP) as follows: the current actual price of Bitcoin serves as a state, the prediction of Bitcoin price as an action, and the difference between the actual price and predicted price as a reward. In general, the state transition probability is often not provided which leads for us to adopt the model-free version of Reinforcement Learning (RL). Using this, we design several reward functions to improve the prediction accuracy of the Q-leaning.

(*c*) Finally, we improve the accuracy of prediction and minimize computer resources (CPU, RAM, and memory) utilization and researcher time. The Q-learning based model receives two different datasets as input data where the first dataset consists of all Bitcoin-related tweets without being attribute-filtered (classic approach) and the second dataset is the most useful dataset which we determined earlier among four datasets (proposed approach). With two different datasets, the model gives two different prediction outputs. By comparing the predictions' parameters, we get a conclusion about which approach is better one.

The remainder of this paper is organized as follows. Section II provides an overview of related research. Detailed information about the data collection, data preprocessing, and sentiment analysis, are provided in Section III. Section IV describes the model learning algorithm and its employment in our research. The experimental results are detailed in Section V. Section VI summarizes the limitations of the study and points the direction for further research. Finally, Section VII concludes the paper.

II. RELATED WORK

In this section, we classify the related researches into the following three categories: (1) Bitcoin price prediction with public opinion, (2) Striving for accurate prediction, and (3) Resource usage minimization. Table 1 shows general information about related studies along with the key algorithms/methods they used.

TABLE 1. Taxonomy of related works.

Goal	Methods
Bitcoin price prediction with public opinion	Pearson correlation [13], Bitcoin/Etherium [14], NewsSentiment [15], Word2vec and N-gram [16], Cumulative sentiment [17], Random Forest Regression [18], **Q-Learning [This paper].**
Striving for accurate prediction	Linear Regression, RNN, and LSTM [19], [20], [22], [23], ARIMAX [21], Linear discriminant analysis [24], ARIMA [25], Logistic regression, Naive Bayes, SVM [26], Multiple linear regression [27], Tweet corpus in COVID-19 era [28], Random Forest, Decision tree, AdaBoost [29], XGBoost-Composite model [30], **Q-Learning [This paper].**
Resource usage minimization	GPU over CPU [31], [33], GPU based system for SVM [32], Apache Spark [34], **Q-Learning [This paper].**

A. BITCOIN PRICE PREDICTION WITH PUBLIC OPINION

Sentiment analysis is an important field for researchers, as people's thoughts and emotions have become popular and an acceptable technique for examining and analyzing public opinion. Twitter, Facebook, and Instagram are examples of social media platform used to collect sentiment data for research. The major goal of adopting these approaches is to identify and extract emotions in spoken or written language using natural language processing techniques. Among other social media platforms, Twitter has recently attracted interest from a wide range of academic disciplines, as it is considered useful for analyzing economic and social datasets. Employment of machine learning algorithms on the data extracted from Twitter has opened widely opportunities including identification of hatred speeches [10], analyzing personalities based on profile pictures [11], prediction on offensiveness in tweets [12], etc.

Over the past decade, there have been some studies within the field of finding the links between price movements and sentiments extracted from Twitter. Kaminski *et al.* [13] found that the platform appears to have an impact on users and information dissemination. Ranasinghe *et al.* [14] demonstrated that Twitter may be related to a shift in the public image of Bitcoin. According to this research, there is a strong link between the probability of Twitter users' influence and the probability of being influenced, but the majority of users maintain a balance in terms of their attitudes in both circumstances. Nagar *et al.* [15] claimed that the sentiment of news obtained from the news corpus and stock price movements were highly correlated. Pagolu *et al.* [16] focused on forecasting stock price movements using Twitter sentiment, and revealed a strong connection between sentiments on Twitter and stock market movements. Sul *et al.* [17] developed a sentiment classifier and compared it with stock returns in 2.5 million tweets related to S&P 500 companies. The findings revealed that rapid sentiment was more likely to be reflected in a stock price on the same trading day, whereas slower-spreading sentiment was more likely to be reflected on upcoming trading days. In our previous research [18], we scrapped more than 9.2 thousand tweets that were posted in a two-month period, and found that when sentiment analysis was applied to tweets regarding Bitcoin and financial data, the sentiment on Twitter had a predictive impact on the Bitcoin findings.

B. STRIVING FOR ACCURATE PREDICTION

It is known that tweet sentiments have positive relationships with price fluctuations. Based on this fact, several techniques have been proposed to accurately predict the future price by the employment of different machine learning algorithms. Mittal *et al.* [19] gathered approximately 7.5 million tweets and obtained results on tweet sentiment after applying long short-term memory (LSTM), recurrent neural network (RNN), and Polynomial regression, whereas tweet volume and Google trends predicted accuracy of 77.01 percent and 66.66 percent for the Bitcoin direction, respectively. Pant *et al.* [20] conducted an another RNN model which categorized Bitcoin tweets as good/positive or negative. They used the percentage of them coupled with historical price of Bitcoin. The results showed total 77.62 percent of prediction accuracy.

While many studies that investigated the token economics based on the Bitcoin network, several researches was focused to analyze the network sentiment on the overall price of Bitcoin. Serafini *et al.* [21] compared two models used for Bitcoin time-series predictions: the Auto-Regressive Integrated Moving Average with eXogenous input (ARIMAX) and RNN. The flow of studies that adopted LSTM to make a price prediction has been continued by Ye *et al.* [22]. As an ensemble method along with LSTM, they used gate recurrent unit (GRU). The results showed that their model performance achieved 88.74% value based on real data from September 2017 to January 2021.

Thanekar *et al.* [23] demonstrated that artificial intellegence (AI) models using sentiment analysis of tweets containing the keywords "bitcoin" or "btc" predicted the volatility in Bitcoin values with higher accuracy than models that compared the values without sentiment analysis using machine learning through an autoregressive integrated moving average model and LSTM network. Gurrib *et al.* [24] achieved 0.828 accuracy in forecasting the next-day price direction by using linear discriminant analysis (LDA) with sentiment analysis of Bitcoin-related tweets. Another study [25] compared AutoRegressive Integrated

Moving Average (ARIMA) and LSTM model to make a real-time prediction of Bitcoin price using public sentiments in tweets and achieved more accurate results by using LSTM. Colianni *et al.* [26] studied how tweet sentiments may be utilized to influence investment decisions, focusing on Bitcoin. The authors employed supervised machine learning algorithms to achieve an hour-by-hour and day-by-day accuracy of above 90%. Similar with above researchers, Jain *et al.* [27] focused on current tweets by classifying positive, negative, and neutral sentiments and accumulating their numbers every two hour to predict the price of Bitcoin and Litecoin two hours in advance. Using multiple linear regression (MLR) model, they utilized more than 1.8 million Bitcoin-related and Litecoin-related tweets to investigate whether social factors were capable of predicting the future price of cryptocurrencies. The study notes that MLR model predicts the price of the Bitcoin and Litecoin with the score of 44% and 59% respectively.

As Bitcoin has no central authority to control and its fluctuations are relevant to ongoing news and events, some researchers have studied how COVID-19 outbreak data (number of new cases, recovery, and deaths) can impact the future price of Bitcoin. Pano *et al.* [28] provided a corpus of tweet text for Bitcoin-related tweets during the summer of the COVID-19 period. This dataset is publicly available and considers three months to perform unimpeded research. In order to make an accurate price prediction, Luo *et al.* [29] tried to feed four different machine learning models with three different data: Bitcoin exchange data, COVID-19 data, and Twitter data from January 2020 to July 2020. One of the findings of this study is COVID-19 data does not help to improve the prediction.

C. RESOURCE USAGE MINIMIZATION

Many researchers have studied how to minimize PC-resource employment while keeping the same working accuracy. One of such study, by Steinkraus *et al.* [31] reported over three times faster training and testing processes when the model was implemented on a graphic processing unit (GPU) rather than a CPU. A greater comparison difference was reported by Catanzaro *et al.* [32] where the classification time and speed were eight times faster when implementing support vector machine (SVM) on a GPU than when implementing an alternative SVM algorithm that ran on a CPU. In contrast to the above two studies, McNally *et al.* [33] ran LSTM model on a CPU and GPU to ascertain the accuracy of the direction of the Bitcoin price in USD. They reported the GPU outperforming by a result of 67.7%. As the dataset for the model to learn increases, Sumarsih *et al.* [34] compared GPU performance with the Apache Spark cluster, which is an in-memory data processing engine that uses RAM instead of an I/O disk. Their data processing simulation using linear regression (LR) to learn Bitcoin trading showed faster results when run on the Apache Spark cluster.

The common point of all the aforementioned researches is that they considered all types of tweets related to cryptocurrency, without considering the importance of the tweet attributes on price movements. To the best of our knowledge, our work is the first attempt to classify the tweet attributes involving the term ‘‘Bitcoin’’ and ‘‘BTC’’, that have effects on the future volatility of Bitcoin price.

TABLE 2. Statistical information of dataset.

Definition	Value
Time frame for prediction (start)	01.04.2014
Time frame for prediction (end)	14.11.2018
Number of total tweets	5,496,138
Tweet with keywords used once	3,462,567
Tweet with keywords used twice	1,154,192
Tweet with keywords used more than three time	879,379

III. DATA PREPARATION

In this section, we describe the data-preparation steps for Bitcoin price prediction. We consider the following four steps in data preparation: (*i*) **data collection**, (*ii*) **preprocessing**, (*iii*) **attribute division**, and (*iv*) **sentiment analysis**. In the data collection step, we collect data containing tweets relating to Bitcoin. Thereafter, we remove noise such as repeated tweets, URLs, user mentions, and extra repeated characters from the dataset in the preprocessing step. In the attribute division step, we build four datasets containing tweets sorted according to their attributes. We perform sentiment analysis on the gathered tweets in the final sentiment analysis step. The detailed explanation of each step is provided below.

A. DATA COLLECTION

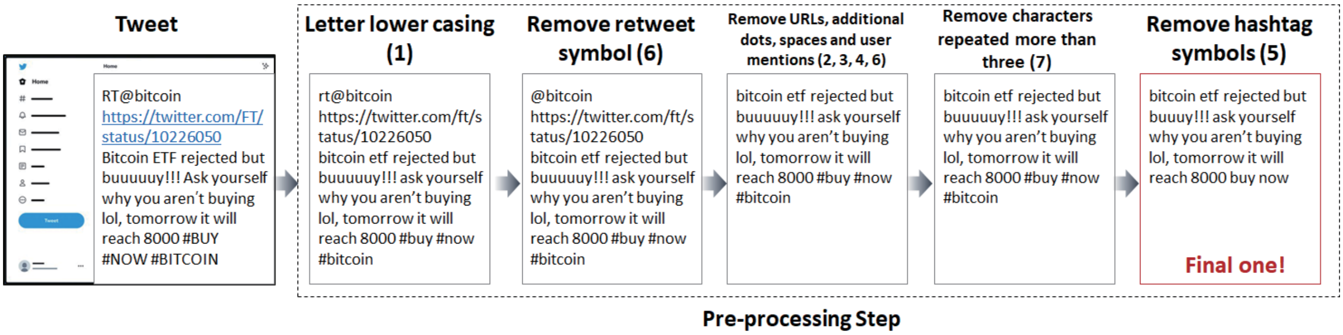
1) BITCOIN PRICE DATA

We use a total of 1690 days’ data that is in the time period from April 1, 2014 to November 14, 2018, in the Bitcoin price market (see [35]) as real data to predict the Bitcoin price because it was observed that the Bitcoin price fluctuated substantially during this period. This motivates us to verify the effectiveness of the proposed method during this period.

2) BITCOIN TWEET DATA

We use Tweepy and Twitter’s streaming API [36] for the Bitcoin-related tweet data. Tweepy is a Python-based open-source framework, makes it easier to gather tweets using Twitter API [37]. Tweepy allows data filtering based on hashtags or terms, which is an effective means of collecting relevant data. The filter keywords are selected using the most definitive Bitcoin context phrases; for example, ‘‘cryptocurrency’’ may contain attitudes towards other cryptocurrencies, and therefore, the scope must be narrowed even further to include only Bitcoin synonyms, such as ‘‘Bitcoin’’ and ‘‘BTC.’’ Using this method, we gather 5,496,138 Bitcoin-related tweets generated within the real data period of the Bitcoin price. Table 2 lists the statistical values for the dataset.

Tweets obtained directly from Twitter typically create noisy datasets. This is due to the social nature of social media use. Certain noises in tweets, such as URLs, emoticons, and user references, must be eliminated appropriately. For this


Tweet
Letter lower casing (1)
Remove retweet symbol (6)
Remove URLs, additional dots, spaces and user mentions (2, 3, 4, 6)
Remove characters repeated more than three (7)
Remove hashtag symbols (5)
Final one!
Pre-processing Step
Home
Tweet

RT@bitcoin https://twitter.com/FT/status/10226050 Bitcoin ETF rejected but buuuuuy!!! Ask yourself why you aren't buying lol, tomorrow it will reach 8000 #BUY #NOW #BITCOIN

rt@bitcoin https://twitter.com/ft/status/10226050 bitcoin etf rejected but buuuuuy!!! ask yourself why you aren't buying lol, tomorrow it will reach 8000 #buy #now #bitcoin

@bitcoin https://twitter.com/ft/status/10226050 bitcoin etf rejected but buuuuuy!!! ask yourself why you aren't buying lol, tomorrow it will reach 8000 #buy #now #bitcoin

bitcoin etf rejected but buuuuuy!!! ask yourself why you aren't buying lol, tomorrow it will reach 8000 #buy #now #bitcoin

bitcoin etf rejected but buuuuuy!!! ask yourself why you aren't buying lol, tomorrow it will reach 8000 #buy #now #bitcoin

bitcoin etf rejected but buuuy!!! ask yourself why you aren't buying lol, tomorrow it will reach 8000 buy now

FIGURE 2. Example of data preprocessing tasks in our approach. (At each task, the number in parentheses represents the number in Table 3.)

TABLE 3. Pre-processing tasks.

Number	Tasks
1	Change all letters in tweet to lower case
2	Check and switch 2 or more dots (.) with space
3	Switch 2 or more spaces with one single space
4	Remove user-mentioning symbol (@)
5	Change hashtags into typical words
6	Remove retweet symbol (RT) and URLs
7	Reduce characters repeated more than 3 times

purpose raw Twitter data must be formatted to build a dataset that can be easily processed by multiple classifiers. To this end, we consider several preprocessing steps to normalize the dataset, minimize its size, etc. Table 3 presents an example of our preprocessing tasks, in which the above order is not important. We use the data refined according to the corresponding processing.

B. ATTRIBUTE DIVISION

To determine the effects of tweet attributes, we divide the preprocessed data into the following four types: (1) number of followers of the poster, (2) number of comments on the tweet, (3) number of likes, and (4) number of retweets.

1) SORTING ACCORDING TO ATTRIBUTES

We consider that the tweet data covered tweets posted within 1,688 days, and we already obtain a single dataset with over 5 million tweets during this period.

To create datasets of interest, tweets posted on a particular day were separately sorted into four datasets according to the above attributes in decreasing order. That is, we sort the dataset by attribute (1), save it separately and sort it again by attribute (2), save it separately, and repeat this process with attributes (3) and (4). However, this is the same dataset.

2) AVOIDING SIMILAR DATA

To prevent similar data from appearing in each dataset, only the first half of each dataset is used in the experiment. In simple terms, all tweets posted in one day were sorted in decreasing order of their number of comments, and only the first half of the tweets were used as the first dataset. Subsequently, the tweets are disordered by the number of followers (1) and only the first half is used as the second dataset. Similarly, they are sorted according to the number of comments (2) and the number of retweets to create attributes (3) and (4).

C. SENTIMENT ANALYSIS

As a final step, we apply sentiment analysis to determine the subjective emotions or views expressed in the tweets on Bitcoin. We perform sentiment analysis by categorizing textual views into categories such as "positive," "negative," or "neutral." We use the Valence Aware Dictionary and Sentiment Reasoner (VADER) [38] to classify the content of each tweet. VADER is a sentiment analysis Python library that uses lexicons and rules to analyze sentiments posted on social media. VADER includes three valence scores for each sentiment, given text content: positive, negative, and neutral. The valence ratings of each word in the lexicon are added together, modified according to the rules, and then normalized into $[-1, 1]$, where -1 is extremely negative, $+1$ is extremely positive, and 0 is neutral. These statistics are good because they provide a single unidimensional estimate of the emotion for each tweet. Based on this, we use the compound score to describe the sentiment of each tweet. Subsequently, we perform proper Q-learning for the price prediction with sentimentally analyzed tweet data, as described in the following section.

IV. LEARNING ALGORITHM

In this section, we introduce our approach to predicting Bitcoin prices based on Twitter data. For this, we adopt simple reinforcement learning, in which the environment was the Bitcoin market. First, we briefly explain RL and the proposed approach with RL in the following subsection.

A. RL AND Q-LEARNING

Standard RL is formulated based on a Markov decision Process (MDP). An MDP is a tuple $<\mathscr{S}, \mathscr{A}, r, P, \gamma>$, where $\mathscr{S}$ and $\mathscr{A}$ are sets of states and actions, respectively, and $\gamma \in [0, 1]$ denotes the discount factor. A transition probability function $P : \mathscr{S} \times \mathscr{A} \rightarrow \mathscr{S}$ maps the states and actions to a probability distribution over the next states, and

$r : \mathscr{S} \times \mathscr{A} \rightarrow \mathbb{E}$ denotes the reward. The goal of RL is to learn a policy $\pi : \mathscr{S} \rightarrow \mathscr{A}$ that solves the MDP by maximizing expected discounted returns $R_t = \mathbb{R}\sum_{k=0}^{\infty}\gamma^k r_{t+k}|\pi$. The policy induces a value function $V^{\pi}(s) = \mathbb{R}_{\pi}[R_t|s_t = s]$ and an action value function $Q^{\pi}(s, a) = \mathbb{R}_{\pi}[R_t|s_t = s, a_t = a]$.

In general, the state transition probability is often not provided in the RL. In this case, the agent must learn the optimal policy using trial and error through exploration. In RL, determining a policy that maximizes the expected reward through this process is known as model-free learning. Q-learning is one of the most famous model-free algorithms. RL strategies (such as Q-learning) have recently been used in various sectors to improve prediction models in various areas of social network research [39]. Q-learning [40] is a simple RL algorithm that provides the current state and finds the best action to be taken in that state. This is an off-policy algorithm because it learns from random actions. It constructs a Q-table $Q(s, a)$, where the value of the table is the reward when the agent selects action $a \in \mathscr{A}$ at state $s \in \mathscr{S}$. The algorithm operates in three basic steps: (1) the agent starts in a state, takes an action, and receives a reward; (2) for the next action, the agent has two choices: either reference the Q-table and select an action with the highest value, or take a random action; and (3) the agent updates the Q-values (i.e., $Q(s, a)$) in the table. The main objective is to learn the Q-function. To describe this precisely, let s_t and a_t be the state and action at current time t. Before the iteration, Q is initialized to an arbitrary value. Subsequently, at each time t, the agent selects an action a_t at s_t and observes a reward r_t, following which it enters a new state, s_{t+1}. Subsequently, the values of Q are updated. At the core of the algorithm is the Bellman equation as a simple value iteration update using the weighted average of the old value and new information:

$$\begin{aligned} Q^{new}(s_t, a_t)cr &= Q(s_t, a_t) \\ &\quad + \theta * [r_t + \gamma Q^{*}(s_{t+1}, a') - Q(s_t, a_t)], \end{aligned} \tag{1}$$

where θ $(0 < \theta \leq 1)$ is the learning rate and γ is a discount factor with $0 \leq \gamma \leq 1$. The value of Q^* is the estimate of the optimal future value, which is expressed by

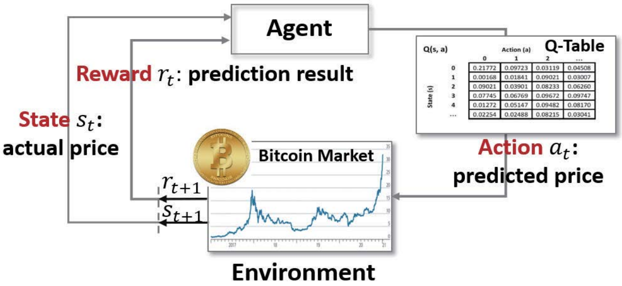
$$Q^{*} = \max_{a'} Q(s_{t+1}, a'). \tag{2}$$

This process continues until s_{t+1} reaches its final or terminal state. Due to the lack of model information (the transition probability of the Bitcoin price), we adopt Q-learning as an RL approach for our Bitcoin price prediction problem.

B. BITCOIN PRICE LEARNING

In this prediction problem, an agent interacts with the environment, which is the Bitcoin market, and learns how to predict future prices based on Q-learning. For this purpose, we define a tuple $< \mathscr{S}, \mathscr{A}, r >$, as follows:

- ***State Space*** $\mathscr{S}$. As a state $s_t \in \mathscr{S}$ of the agent at time t, the actual Bitcoin price at that time is considered. The Bitcoin price is usually expressed with two decimal places (e.g., 21,254.50 USD.). Hence, we note that this is a discrete value and the considered state space is also discrete. Let AP_t be the actual Bitcoin price at time t. Then, the state is $s_t := AP_t$. We assume that $s_t < M$ for a sufficiently large number $M < \infty$.



	0	1	2	...
0	0.21772	0.09723	0.03119	0.04508
1	0.00168	0.01841	0.09021	0.03007
2	0.09021	0.03901	0.08233	0.06260
3	0.07745	0.06769	0.09672	0.09747
4	0.01272	0.05147	0.09482	0.08170
...	0.02254	0.02488	0.08215	0.03041

FIGURE 3. Q-learning of our Bitcoin price prediction.

- ***Action Space*** $\mathscr{A}$. The action $a_t \in \mathscr{A}$ of the agent at time t is defined as a prediction of the current Bitcoin price. However, to reduce the number of action states, the percentage of the current price increasing, decreasing, or not changing is selected. That is, the action space is the rate of the price change as a percentage, which is expressed by $\mathscr{A} := \{-1000, -999, \ldots, 0, \ldots, 999, 1000\}$.[1] For example, if the agent selects 50, it means that the agent predicts the next price by increasing the current price by 50% of the current price; that is, $a_t = 1.5 \times AP_{t-1}$. We also denote this action by the predicted price PP_t at time t.
- ***Reward Function*** r. For the prediction of the actual Bitcoin price, we consider the following three reward functions: (1) simple difference reward (SDR), (2) relative difference reward (RDR), and (3) comparative difference reward (CDR). Detailed description for each function is as follows:

(*i*) ***SDR.*** This reward function is simply based on the difference between the actual price (AP_t) and the predicted price (PP_t). Considering that the model needs to receive a higher reward for a smaller difference, it receives only negative rewards with the highest possible reward $r_t = 0$ in case that AP_t and PP_t are the same. Formally, the SDR is defined by:

$$\text{SDR: } r_t = -|AP_t - PP_t|. \tag{3}$$

(*ii*) ***RDR.*** It is based on the relative difference between AP_t and PP_t, which is formally defined by:

$$\text{RDR: } r_t = \frac{-|AP_t - PP_t|}{AP_t} * 100\%, \tag{4}$$

where $AP_t > 0$. Therefore, $r_t \in [-\infty, 0]$ where $r_t = 0$ means perfect fit of PP_t to the AP_t.

(*iii*) ***CDR.*** In the prediction of actual price of Bitcoin, it will be an important information on how much has increased or decreased compared to the previous

[1] This is because it rarely increases or decreases by more than 1000% compared with the previous day in Bitcoin price prediction.

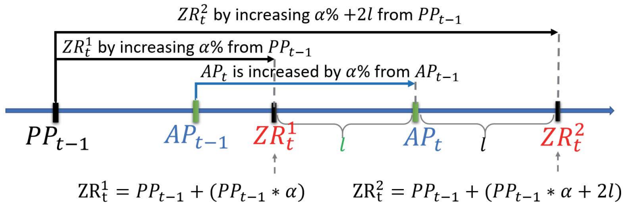

ZR_t^2 by increasing α% $+2l$ from PP_{t-1}
ZR_t^1 by increasing α% from PP_{t-1}
AP_t is increased by α% from AP_{t-1}
PP_{t-1}
AP_{t-1}
ZR_t^1
l
AP_t
l
ZR_t^2
$\mathrm{ZR}_t^1 = PP_{t-1} + (PP_{t-1} * \alpha)$
$\mathrm{ZR}_t^2 = PP_{t-1} + (PP_{t-1} * \alpha + 2l)$

FIGURE 4. Computation of two zero-valued rewards ZR_t^1 and ZR_t^2 used in the paper. (AP_t - actual price at t time, PP_t - predicted price at time t).

step. In the third reward function, we consider the additional information about this rate of change. To formally describe this, we first introduce a concept of zero-reward value as follow.

Definition 1: Let $\alpha = (AP_t - AP_{t-1})/AP_{t-1}$ where $AP_{t-1} > 0$ *i.e.*the rate of change of actual price. Let $l = AP_t - PP_{t-1}(1+\alpha) > 0$. We call a point by *Zero-value reward (ZR)* where the difference from AP_t is l.

Actually, we have two such zero-reward values as shown in Figure 4 since one point is less than l from AP_t and the other is larger than l from it. We denote the former by ZR_t^1 and the latter by ZR_t^2, respectively. Then, ZR_t^1 is computed by (See Figure 4):

$$ZR_t^1 = PP_{t-1} + (PP_{t-1} * \alpha), \tag{5}$$

and the ZR_t^2 is computed by

$$ZR_t^2 = PP_{t-1} + (PP_{t-1} * \alpha + 2l). \tag{6}$$

From two zero-reward points, we compute the reward value based on whether PP_t is higher or lower than the AP_t. The formula of computing the reward value is different according to the value of PP_t. The explanation of the possible PP_t cases and computing formulas is as follows.

(a) The case where the PP_t is smaller than the actual price ($PP_t < AP_t$). As AP_t value stands in the middle of positive rewards interval; the agent receives a negative reward if $PP_t < ZR_t^1$ or a positive reward if the PP_t is between ZR_t^1 and AP_t ($ZR_t^1 < PP_t \leqslant AP_t$). The value of the reward is calculated as follows:

$$r_t = \frac{PP_t - ZR_t^1}{AP_t - ZR_t^1} * 100\% \tag{7}$$

(b) The case where the prediction price is higher than the AP_t ($PP_t > AP_t$). As given in interval determination paragraph, when the PP_t is higher than the AP_t, the model computes ZR_t^2 value to decide whether the reward is positive or negative. If the PP_t is in between AP_t and ZR_t^2 ($AP_t \leqslant PP_t < ZR_t^2$) value range, then the reward will be positive. If the PP_t is higher than ZR_t^2, the reward will be negative. This is computed by:

$$r_t = \frac{PP_t - ZR_t^2}{AP_t - ZR_t^2} * 100\% \tag{8}$$

In all the cases, when the predicted price comes closer to the actual price, the reward value becomes higher. The second reward function, namely the RDR, also varies according to the current Bitcoin price. That is, even if the difference between the predicted and actual prices is the same, the reward value is higher if the current price is large.

Based on the defined reward functions and preprocessed tweet data, the agent learns the actual Bitcoin price and attempts to make a prediction by repeating the following working steps:

- Agent starts in a state (S_1 - Actual price of Bitcoin), takes an action (A_1 - a number between -1000 to 1000, as it will be applied as a percentage of change to actual price), and receives a reward (r - computed based on one of SDR, RDR, and CDR reward functions).
- The agent chooses an action by referring to the highest value in Q-table.
- Update Q-values.

As the Q-values are updated and the agent chooses the maximum value in the table to take the action, the agent performance also starts to improve. The model with the above parameters is tested using four different datasets to experimentally verify the predictability range of tweet attributes. A brief explanation of the experiment is presented in section V.

V. EXPERIMENT AND RESULTS

In this section, we present three different experimental results in order to determine the best reward function, tweet attribute that has the most influence on price, and computer resource working overloads during the performance of both classic and proposed approaches. For this, we use Python to create the experimental environment and the Pandas library for data preprocessing. Sentiment analysis is performed using the VADER analyzer tool, and TensorFlow and Keras are used for training and testing, respectively. For monitoring and analyzing of computer resources (CPU, RAM, and memory) usage we use one of Windows 10 standard tools called Performance Monitor [41]. It is useful with its options where anyone can customize what data to collect, when the collection begins, how long the analysis process needs to run, etc.

Training with Q-Learning. In the model training, we use a dataset of tweets posted between April 1, 2014, and June 30, 2017. The training process yielded promising results when the first part of the divided dataset was used to feed the model. We use $\gamma = 0.95$ as the discount factor because this value provided the best performance during the experiment.

A. PERFORMANCE MEASURES

To evaluate the performance of our model with reward functions, we use six metrics among a wide range of evaluation metrics, as they are the most suitable for the prediction task and provide a valuable evaluation. We briefly describe them as follow.

(i) ***Variance Accounted For (VAF).*** VAD [42] is used to verify the correctness of a model by comparing the real output with the predicted output. The values of VAF which is closed to 100% indicate highly accurate prediction. With the definition of the actual price - AP and the predicted price as - PP, the formula of VAF is given by:

$$VAF = \left(1 - \frac{var(AP - PP)}{var(AP)}\right) * 100\%, \tag{9}$$

where $var(x)$ is the variance of x, which is computed by $var(x) = (\sum_{t=1}^{n}(x_t - \bar{x})^2)/n - 1$. Here, x_t is a value of x at time t and $\bar{x}$ is the average value of x_t from $1 \leq t \leq n$. In our experiment, we set $n = 1690$ for all performance metrics.

(ii) ***Coefficient of Determination (R^2).*** R^2 is used to evaluate the forecast outputs and provides a measure of how well-observed outcomes are replicated by the model [43]. Formally, it is computed by:

$$R^2 = 1 - \frac{RSS}{TSS}. \tag{10}$$

where RSS is the residual sum of squares which is given by $RSS = \sum_{t=1}^{n}(AP_t - PP_t)^2$ and TSS is the total sum of squares that is $TSS = \sum_{t=1}^{n}(AP_t - \overline{AP})^2$. Here, AP_t and PP_t denote the actual price and predicted price of Bitcoin at time t, and $\overline{AP}$ is the average value of AP_t for time $1 \leq t \leq n$, respectively. Hence, the range of R^2 is [0, 1], where 1 indicates a perfect match of the prediction data with actual data.

(iii) ***Mean Absolute Percentage Error (MAPE).*** Like the aforementioned metrics, MAPE is also used to measure the prediction accuracy but unlike them, it is commonly used as a loss function in model evaluation because of its highly intuitive interpretation in terms of relative error [46]. The formal computation is given by:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{AP_t - PP_t}{AP_t}\right| * 100\%. \tag{11}$$

(iv) ***Nash–Sutcliffe model efficiency coefficient (NSE).*** The fourth evaluation metric that we consider to use is NSE, which is used to assess the predictive skill of models [47]. Following formula used to calculate the NSE value of the model prediction.

$$NSE = 1 - \frac{\sum_{t=1}^{n}(AP_t - PP_t)^2}{\sum_{t=1}^{n}(AP_t - \bar{A}P)^2}. \tag{12}$$

Hence, the NSE becomes one in the case of a perfect prediction.

TABLE 4. Summary of evaluation metrics in the paper. (AP_t - actual price at time t, PP_t - predicted price at time t, $\overline{AP}$ - average of actual price.).

Evaluation metrics	Equation
VAF	$\left(1 - \frac{var(AP-PP)}{var(AP)}\right) * 100\%$
R^2	$R^2 = 1 - (RSS/TSS)$ $RSS = \sum_{t=1}^{n}(AP_t - PP_t)^2$ $TSS = \sum_{t=1}^{n}(AP_t - \overline{AP})^2$
MAPE	$\frac{1}{n}\sum_{t=1}^{n}\left\lvert\frac{AP_t - PP_t}{AP_t}\right\rvert * 100\%$
NSE	$1 - \frac{\sum_{t=1}^{n}(AP_t - PP_t)^2}{\sum_{t=1}^{n}(AP_t - \overline{AP})^2}$
RMSE	$\sqrt{\frac{\sum_{t=1}^{n}(AP_t - PP_t)^2}{n}}$
WMAPE	$\frac{\sum_{t=1}^{n}\lvert AP_t - PP_t\rvert}{\sum_{t=1}^{n}AP_t} * 100\%$

(v) ***Root-mean-square error (RMSE)*** This evaluation metric is frequently used to measure the difference between values predicted by the model and observed values [48]. Formally, it is computed by:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(AP_t - PP_t)^2}{n}}. \tag{13}$$

Hence, we see that RMSE value is always non-negative, and a lower RMSE indicates a more accurate prediction than a higher RMSE.

(vi) ***Weighted Mean Absolute Percentage Error (WMAPE)*** WMAPE is a variant of MAPE in which errors are weighted by values of actuals [49]. The advantage of this metric over MAPE is that it overcomes the "infinite error" issue [50]. The formal metric is defined by:

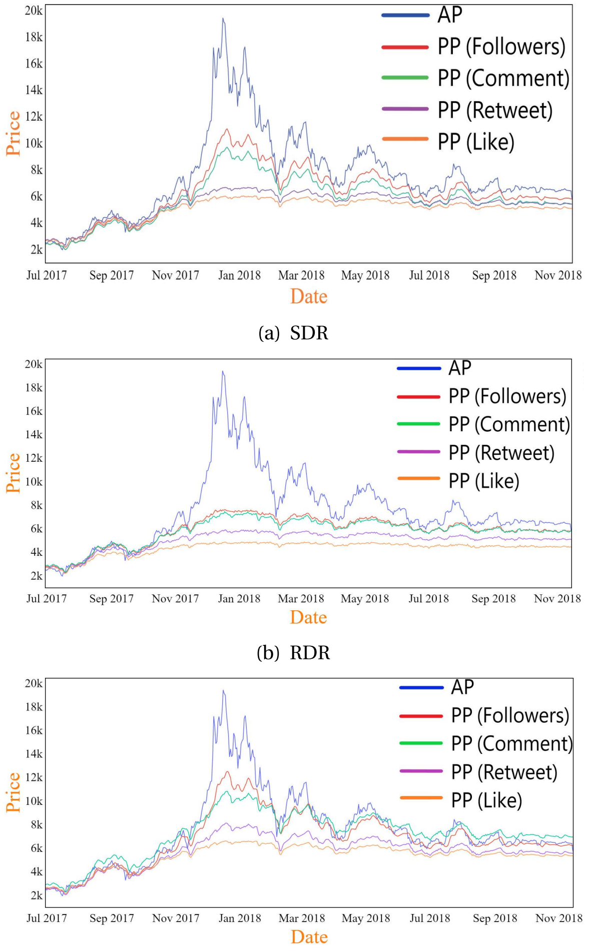
$$WMAPE = \frac{\sum_{t=1}^{n}|AP_t - PP_t|}{\sum_{t=1}^{n}AP_t} * 100\%. \tag{14}$$

B. RESULTS FOR EACH REWARD FUNCTION WITH FOUR ATTRIBUTES

As a first experiment result, we will show the prediction performance for the three reward functions to determine the most useful tweet attribute in predicting the price. For this, we use a dataset of tweets posted between July 1, 2017, and November 14, 2018. We obtain the prediction results based on four attributes: most commented, most liked, most retweeted, and the number of poster followers.

1) TWEET ATTRIBUTE CLASSIFICATION

First, in Figure 5, we see that tweets posted by those with the most followers and tweets with the most comments exhibit



(a) SDR

(b) RDR

(c) CDR

FIGURE 5. Prediction performance of Bitcoin price for four attributes with three reward functions ((a) SDR, (b) RDR, and (c) CDR) for testing datasets. (AP: actual price, PP(x): predicted price with attribute x.).

the best prediction results for all the SDR, RDR, and CDR reward functions. However, the prediction with CDR is better than that with SDR and RDR because the CDR provides a reward by comparing the current action a_t with the previous action a_{t-1}. Each action comparison with the previous action provides the opportunity to compare all actions relative to each other which boosts the learning process. The result shows that there is a high chance that people's tweets with the most followers catch the public's attention by being viral and have some influence on future events. Moreover, it can be seen from the results of the experiment, that there is a ranking among the attributes based on their predictive powers. Among three prediction outputs with the three different reward functions, the dataset sorted by the number of user followers shows the most accurate prediction. Next, the dataset created from tweets with the most comments shows a more accurate forecast than the remaining two datasets. As the prediction results in Figure 5, the most retweeted attribute comes in third place, whereas the most liked attribute is in the last place.

TABLE 5. Performance metrics for prediction with SDR, RDR, and CDR.

With SDR

	VAF	R^2	MAPE	NSE	RMSE	WMAPE
Follower	78.85	0.63	13.919	0.62	1944.3	16.9
Comment	71.84	0.43	19.257	0.44	2421.3	22.9
Retweet	42.79	0.32	22.833	0.32	3243.7	29.1
Like	36.31	0.22	26.785	0.23	3574.2	33.3

With RDR

	VAF	R^2	MAPE	NSE	RMSE	WMAPE
Follower	51.87	0.25	17.690	0.26	2789.5	23.1
Comment	47.41	0.18	18.948	0.18	2925.8	24.5
Retweet	33.13	0.10	27.408	0.11	3651.9	34.1
Like	22.96	0.06	35.439	0.06	4250.5	42.6

With CDR

	VAF	R^2	MAPE	NSE	RMSE	WMAPE
Follower	84.81	0.80	8.450	0.81	1441.2	10.7
Comment	71.87	0.70	11.510	0.69	1743.1	13.3
Retweet	57.68	0.34	19.078	0.33	2764.7	24.2
Like	42.64	0.26	23.378	0.27	3281.7	29.6

2) PERFORMANCE EVALUATIONS

To see the sufficient prediction performance, we obtain six different evaluation metrics during the assessment of performance for each reward function and each attribute are listed in Table 5. First, we see that, in the case of CDR, the VAF values show the most accurate prediction compared with SDR and RDR. Further, the attribute of number of poster's followers has the highest prediction performance as we expected.

In contrast to VAF, the R^2 takes values in the range [0, 1] where 1 indicates an ideal prediction. Keeping this definition in mind and by comparing the R^2 values of each reward function, we can determine that the model achieves a more precise prediction with CDR by having a maximum 0.8 value rather than SDR and RDR by having 0.63 and 0.25, respectively. The maximum R^2 values are achieved with the dataset that consists of posters' tweets with the highest number of followers.

By scoping the three prediction outputs with metric MAPE, we obtain a result that indicates the level of error in the predictions. Therefore, a lower MAPE value indicates higher accuracy. The MAPE value also shows no contradiction in the priority of the CDR over the SDR and RDR functions. For example, while SDR is being implemented by the model, the first attribute has a value of 13.919, which is the lowest among the second, third, and fourth attributes, with 19.257, 22.833, and 26.785 values, respectively. During the RDR implementation, the model has the lowest prediction quality. The MAPE value of the first attribute increased to 17.690 in this scenario, but still dominates the remaining attributes.

For the NSE metric, we observe similar results as the R^2 metric. Because the performance values are quite similar, we refrained from analyzing the reward functions' preferability and ranking of attributes.

In using the RMSE, taking into account the fact that RMSE measurement is based on errors, a low value of RMSE indicates a more accurate prediction than a high value RMSE.

While SDR is implementing by the model, the follower attribute has the lowest value among all attributes. The model has the poorest prediction quality during RDR implementation. In this case, the RMSE value of the follower attribute increased to 2789.5, but it still dominates the remaining attributes. As we expected, the RMSE also shows the best prediction when the model used CDR as a reward function.

The WMAPE is the last evaluation metric used in this study. Because it is a variant of MAPE, a smaller WMAPE value indicates an accurate prediction. With respect to CDR, WMAPE values indicate the most accurate forecast when compared to SDR and RDR. For example, the first attribute has a CDR value of 10.7, although this attribute has SDR and RDR values of 16.9 and 23.1, respectively.

3) PERFORMANCE COMPARISONS

In order to detect how good the model's performance is, we compare the accuracy of our prediction along with other similar studies that used different approaches to achieve an accurate prediction. However, there are several problems that resist making a fair comparison: types of data and its time period are different across studies; the model design and its implementation are not explained in detail in some studies; diversity of the metrics that are used to evaluate the model's performance; and difficulties on gathering all source codes and run in the same PC environment. Therefore, in Table 6, we briefly compare the results of previous relevant work with our proposed method. Most references are used Twitter as the main data source to obtain Bitcoin price predictions and yet only a few of them have considered analyzing the PC resource usage level. In the Table, we use the terms as follows: (*i*) Non-filtered: BTC historical price data is used as the main dataset without being filtered by any conditions and is used entirely in its form. (*ii*) Non-attribute filtered: Bitcoin-related tweets are used as the main dataset without being filtered by any Twitter attribute and are used entirely in its form. (*iii*) Attribute-filtered: Bitcoin-related tweets are used as the main dataset and the dataset has been used after filtering by the "number of followers" attribute. We see that the result in Ye *et al.* [22] shows the highest accuracy level with an 88.74% value but the resource usage did not considered. We observe that our proposed Q-learning model that considers only Bitcoin-related tweets that are posted by posters who have the most number of followers, considers the PC resource usage level while obtaining 84.81% accuracy which overcomes most of the previous studies results.

C. RESULTS WITH COMPUTER RESOURCE USAGE

In this subsection, we will describe the comparison results between our proposed approach and the classic approach. These two are explained as follows.

(*a*) ***Proposed approach:*** In the proposed approach, we obtain the Bitcoin-related tweets only from those who have the most followers, *i.e.*we use the data with attribute-filtering.

TABLE 6. Performance comparison with other results.

Studies	Datasets	Filter	Model	Accuracy	PC resource analysis
Mittal *et. al.* [19] - 2019	BTC price, Google trends	non-filtered	LSTM, RNN	66.66%	No
Pant *et. al.* [20] - 2018	BTC price, Tweets	non-attribute filtered	RNN	77.62%	No
McNally *et. al.* [33] - 2018	BTC price	non-filtered	RNN, LSTM	52.78%	Yes
Ye *et. al.* [22] - 2022	BTC price, Tweets	non-attribute filtered	LSTM, GRU	88.74%	No
Gurrib *et. al.* [24] - 2021	BTC price, Tweets	non-attribute filtered	LDA	82.8%	No
Sumarsih *et. al.* [34] - 2018	BTC price	non-filtered	LR	73.15%	Yes
Jain *et. al.* [27] - 2018	BTC price, Tweets	non-attribute filtered	MLR	44%	No
This paper	**BTC price, Tweets**	**attribute filtered**	**Q-learning**	**84.81%**	**Yes**

(*b*) ***Classic approach:*** In this approach, we obtain all Bitcoin-related tweets, *i.e.*we use all of the data without attribute-filtering.

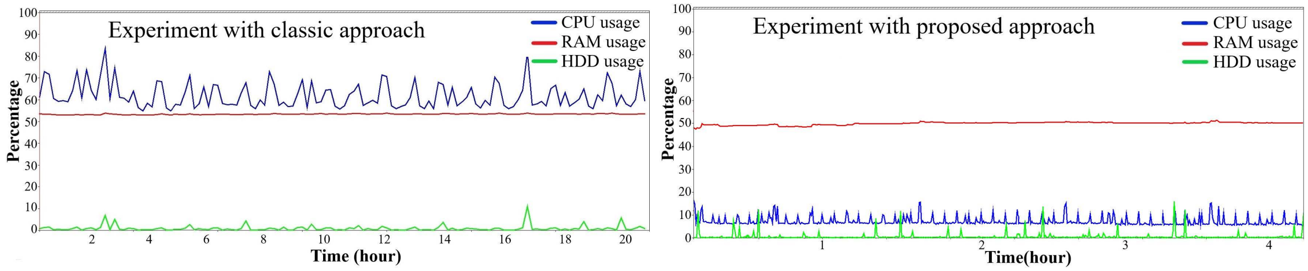
For this, we perform two different experiments as follows.

(*i*) ***Fixed running time:*** In the first experiment, we see how the resource usage and accuracy for each approach are different when the running time of same PC is equal to 1 hour.

(*ii*) ***Fixed target accuracy:*** In the second experiment, we check how much the performance difference are when the target accuracy of prediction is fixed for both approaches. During the experiment, we observe the status of the CPU workloads, RAM, and memory usage.

The resource usage information is given using three types of metrics: minimum, average, and maximum values during the experiment. At the end of the experiment, we calculate the accuracy of both approaches and a comparison of the observed results is presented in Table 7.

(*i*) ***First experiment result (Fixed running time).*** In the first experiment, we check that there is almost no noticeable difference in CPU usage between the approaches as in Table 7. However, some comparable results are observed in RAM usage, where the classic approach's minimum usage is 53.9%, average usage is 54.1%, 54.7% maximum, and the proposed approach's minimum usage level is 42%, 42.4% on average, and 45.1% maximum, respectively. In the memory usage, we check that our approach is more efficient. Further, the most noticeable difference between the approaches



(a) Resources usage graph with classic approach

(b) Resources usage graph with proposed approach

FIGURE 6. Computer resources usage results for the two approaches (a) Classic approach and (b) Proposed approach. In the experiment, we set the target accuracy by VAF = 84.81%. The blue line graph represents CPU workloads, red line indicates RAM usage, and green line describes memory usage.

TABLE 7. Experiment results of classic and proposed approaches.

Classic approach performance (runtime 1 hour)

	CPU usage %	RAM usage %	Memory usage %	Utilized tweets	Accuracy %
Minimum	5.11	53.909	0.85	193,406	23.814
Average	6.18	54.109	0.922		
Maximum	13.76	54.716	30.683		

Proposed approach performance (runtime 1 hour)

	CPU usage %	RAM usage %	Memory usage %	Utilized tweets	Accuracy %
Minimum	4.92	42.007	0.144	167,322	36.273
Average	6.892	42.482	0.852		
Maximum	14.453	45.131	13.17		

TABLE 8. Results from classic and proposed approaches after achieving the same target level of accuracy.

Classic approach performance (target accuracy is 84.81%)

	CPU usage %	RAM usage %	Memory usage %	Utilized tweets	Runtime
Minimum	46.125	51.809	0.31	5,185,742	**21h 36min 39sec**
Average	61.832	53.387	9.48		
Maximum	85.671	56.129	18.53		

Proposed approach performance (target accuracy is 84.81%)

	CPU usage %	RAM usage %	Memory usage %	Utilized tweets	Runtime
Minimum	4.334	48.377	0.15	785,469	**4h 17min 14sec**
Average	7.758	51.624	5.712		
Maximum	16.631	53.865	17.728		

is observed in the number of tweets utilized. During the 1-hour experimentation, the classic approach utilized 193,406 tweets, whereas the proposed approach utilized 167,322 tweets. Although the number of tweets used in the proposed approach is approximately 26,000 smaller than in the classical approach, it achieves a 36.2% accuracy, whereas the accuracy of the classic approach is 23.8%, which is 13% less accurate than the proposed approach.

(*ii*) ***Second experiment result (Fixed target accuracy).*** Finally, we perform same experiment to achieve the target accuracy of Bitcoin price prediction. To do this, we set a target accuracy level of - VAF = 84.81%, because we observed that the model with the proposed approach achieved this level of accuracy during the first experiment. We run both approaches until they reaches the target accuracy level and compare resource usage accordingly. We obtain our results in Figure 6 and Table 8, respectively.

As a result, we first see that there is a significant difference in CPU usage in this experiment. In the classic approach, the CPU workload is between 46.1% and 85.6%, with an average of 61.8%. The proposed approach shows a minimum of 4.3%, average of 7.7%, and maximum of 16.6%, which is almost 9 times less than the classic approach used. In the RAM usage, we check that there is no significant differences whereas we see that the average usage of memory in the proposed approach is better than that of classic one. Finally, we check that the classic approach runs for 21 hour 36 minute 39 second to achieve the target accuracy, which is almost five times more than the time required to achieve the same level by spending 4 hour 17 minute 14 second with the proposed model.

From the experiment results, we conclude that the proposed approach has much advantages over the classic approach. Considering the poster's tweets with the highest number of followers can lead to accurate prediction and prevent the computer from wasting its resources.

VI. DISCUSSION

In this work, we have checked that it is better for prediction performance and resource efficiency to extract and use data suitable for price prediction than to use all data in Bitcoin price prediction through tweeter data. In particular, for this purpose, even if only the attribute data of the most follower among the data on Twitter was used, the results were much better than the classic approach using all data. Furthermore, the model contributes to the literature on tweet sentiment studies and price prediction using reinforcement learning and provides reliable advice for further in-depth analysis.

However, there exist some limitations to the considered approach in this paper. First, we only used Twitter posted data to analyze people's feelings, which may be biased since not all crypto-traders express their opinions on Twitter. We realize that Bitcoin values are affected by a variety of variables that cannot be captured only through Twitter sentiments. Tweets and other social media (e.g., Reddit and Facebook) may be used to extract feelings in the real world, such as through news and other sources including photos and videos from YouTube or TV channels. Second, we analyzed the price prediction of Bitcoin by considering only four attributes of Twitter. Additional comparison results can be obtained by considering other attributes such as tweet language and tweet poster's location. In the case of tweet language, most of the data is expressed in one language (e.g., English), so it will not significantly affect the price prediction. However, it may be interesting to see how data according to the tweet poster's location affects the Bitcoin price and prediction performance. Third, the algorithm for the predictive model can be modified by extending it to deep reinforcement learning algorithms. This has the advantage of being able to express the Q-function used in Q-learning more accurately with the deep learning method, so it is expected to help improve prediction performance. Finally, considering other sources for sentiment data and other types of cryptocurrencies could also increase the accuracy of predictions. All of these things could be our further research.

VII. CONCLUSION

In this paper, we considered Bitcoin price prediction based on Q-learning using tweet data. We analyzed the manner in which Bitcoin-related information on Twitter affects the actual Bitcoin price by considering four main attributes: number of followers of the poster, number of comments on tweets, number of likes, and number of retweets. We predicted the actual Bitcoin price using a Q-learning method, and obtained the most valuable attributes with three reward functions. We verified that tweets with the most user-related attributes had the greatest effect on the future Bitcoin price. Next, we compare our approach with a classic approach where all Bitcoin-related tweets without being attribute-filtering, are uses as input data for the model, by analyzing the CPU workloads, RAM usage, memory, time, and prediction accuracy. We conclude that the proposed approach has much advantages over the classic approach.

REFERENCES

[1] O. Sattarov, A. Muminov, C. W. Lee, H. K. Kang, R. Oh, J. Ahn, H. J. Oh, and H. S. Jeon, "Recommending cryptocurrency trading points with deep reinforcement learning approach," *Appl. Sci.*, vol. 10, no. 4, p. 1506, Feb. 2020.

[2] M. Mallikarjuna and R. P. Rao, "Evaluation of forecasting methods from selected stock market returns," *Financial Innov.*, vol. 5, no. 1, pp. 1–6, Dec. 2019.

[3] G. S. Atsalakis and K. P. Valavanis, "Surveying stock market forecasting techniques—Part II: Soft computing methods," *Exp. Syst. Appl.*, vol. 36, no. 3, pp. 5932–5941, Apr. 2009.

[4] E. F. Fama, "The behavior of stock-market prices," *J. Bus.* vol. 38, no. 1, pp. 34–105, 1965. [Online]. Available: https://www.jstor.org/stable/2350752

[5] L. Kristoufek, "What are the main drivers of the bitcoin price? Evidence from wavelet coherence analysis," *PLoS ONE*, vol. 10, no. 4, Apr. 2015, Art. no. e0123923.

[6] *American Institute for Economic Research*. Accessed: May 24, 2022. [Online]. Available: https://www.aier.org/article/bitcoins-largest-price-changes-coincide-with-major-news-events-about-the-cryptocurrency/

[7] *Twitter*. Accessed: Jun. 20, 2022. [Online]. Available: https://itzone.com.vn/en/article/elon-musk-tweets-alluding-to-break-up-with-bitcoin/

[8] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Trans. Web*, vol. 1, no. 1, pp. 1–10, 2007.

[9] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proc. 7th Int. Conf. Lang. Resour. Eval. (LREC)*, vol. 10, May 2010, pp. 1320–1326.

[10] T. Zia, M. Akram, M. Nawaz, and M. Lali, "Identification of hatred speeches on Twitter," in *Proc. 52nd IRES Conf.*, Kuala Lumpur, Malasia, Nov. 2016, pp. 27–32.

[11] S. K. Bhatti, A. Muneer, M. I. Lali, M. Gull, and S. M. U. Din, "Personality analysis of the USA public using Twitter profile pictures," in *Proc. Int. Conf. Inf. Commun. Technol. (ICICT)*, Dec. 2017, pp. 165–172, doi: 10.1109/ICICT.2017.8320184.

[12] A. Imran, W. Aslam, and M. Ullah, "Quantitative prediction of offensiveness using text mining of Twitter data," *Sindh Univ. Res. J.-SURJ Sci.*, vol. 49, no. 1, pp. 87–92, 2017.

[13] J. Kaminski, "Nowcasting the bitcoin market with Twitter signals," 2014, *arXiv:1406.7577*.

[14] H. Ranasinghe and M. Halgamuge, "Twitter sentiment data analysis of user behavior on cryptocurrencies: Bitcoin and Ethereum," in *Analyzing Global Social Media Consumption*. Hershey, PA, USA: IGI Global, 2021, pp. 277–291.

[15] A. Nagar and M. Hahsler, "Using text and data mining techniques to extract stock market sentiment from live news streams," in *Proc. Int. Conf. Comput. Technol. Science.* vol. 47, 2012, pp. 91–95.

[16] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, "Sentiment analysis of Twitter data for predicting stock market movements," in *Proc. Int. Conf. Signal Process., Commun., Power Embedded Syst. (SCOPES)*, Oct. 2016, pp. 1345–1350.

[17] H. K. Sul, A. R. Dennis, and L. I. Yuan, "Trading on Twitter: Using social media sentiment to predict stock returns," *Decis. Sci.*, vol. 48, no. 3, pp. 454–488, Jun. 2017.

[18] O. Sattarov, H. S. Jeon, R. Oh, and J. D. Lee, "Forecasting bitcoin price fluctuation by Twitter sentiment analysis," in *Proc. Int. Conf. Inf. Sci. Commun. Technol. (ICISCT)*, Nov. 2020, pp. 1–4.

[19] A. Mittal, V. Dhiman, A. Singh, and C. Prakash, "Short-term bitcoin price fluctuation prediction using social media and Web search data," in *Proc. 12th Int. Conf. Contemp. Comput. (IC)*, Aug. 2019, pp. 1–6.

[20] D. R. Pant, P. Neupane, A. Poudel, A. K. Pokhrel, and B. K. Lama, "Recurrent neural network based bitcoin price prediction by Twitter sentiment analysis," in *Proc. IEEE 3rd Int. Conf. Comput., Commun. Secur. (ICCCS)*, Oct. 2018, pp. 128–132, doi: 10.1109/CCCS.2018.8586824.

[21] G. Serafini, P. Yi, Q. Zhang, M. Brambilla, J. Wang, Y. Hu, and B. Li, "Sentiment-driven price prediction of the bitcoin based on statistical and deep learning approaches," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8, doi: 10.1109/IJCNN48605.2020.9206704.

[22] Z. Ye, Y. Wu, H. Chen, Y. Pan, and Q. Jiang, "A stacking ensemble deep learning model for bitcoin price prediction using Twitter comments on bitcoin," *Mathematics*, vol. 10, no. 8, p. 1307, Apr. 2022.

[23] I. Gurrib, F. Kamalov, and L. Smail, "Bitcoin price forecasting: Linear discriminant analysis with sentiment evaluation," in *Proc. ArabWIC, 7th Annu. Int. Conf. Arab Women Comput. Conjunct 2nd Forum Women Res., Sharjah (UAE)*, Aug. 2021, pp. 148–152.

[24] A. Thanekar, S. Shelar, A. Thakare, and V. Yadav, "Bitcoin movement prediction using sentimental analysis of Twitter feeds," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 2, pp. 148–152, Feb. 2019.

[25] S. M. Raju and A. M. Tarif, "Real-time prediction of BITCOIN price using machine learning techniques and public sentiment analysis," 2020, *arXiv:2006.14473*.

[26] S. Colianni, S. Rosales, and M. Signorotti, "Algorithmic trading of cryptocurrency based on Twitter sentiment analysis," Stanford Univ., Stanford, CA, USA, Tech. Rep. CS229 proj2015/029, 2015, pp. 1–5.

[27] A. Jain, S. Tripathi, H. D. Dwivedi, and P. Saxena, "Forecasting price of cryptocurrencies using tweets sentiment analysis," in *Proc. 11th Int. Conf. Contemp. Comput. (IC)*, Aug. 2018, pp. 1–7.

[28] T. Pano and R. Kashef, "A corpus of BTC tweets in the era of COVID-19," in *Proc. IEEE Int. IoT, Electron. Mechatronics Conf. (IEMTRONICS)*, Sep. 2020, pp. 1–4, doi: 10.1109/IEMTRONICS51293.2020.9216427.

[29] J. Luo, "Bitcoin price prediction in the time of COVID-19," in *Proc. Manage. Sci. Informatization Econ. Innov. Develop. Conf. (MSIEID)*, Dec. 2020, pp. 243–247, doi: 10.1109/MSIEID52046.2020.00050.

[30] A. Ibrahim, "Forecasting the early market movement in bitcoin using Twitter's sentiment analysis: An ensemble-based prediction model," in *Proc. IEEE Int. IoT, Electron. Mechatronics Conf. (IEMTRONICS)*, Apr. 2021, pp. 1–5, doi: 10.1109/IEMTRONICS52119.2021.9422647.

[31] D. Steinkraus, I. Buck, and P. Y. Simard, "Using GPUs for machine learning algorithms," in *Proc. 8th Int. Conf. Document Anal. Recognit. (ICDAR)*, 2005, pp. 1115–1119.

[32] B. Catanzaro, N. Sundaram, and K. Keutzer, "Fast support vector machine training and classification on graphics processors," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 104–111.

[33] S. McNally, J. Roche, and S. Caton, "Predicting the price of bitcoin using machine learning," in *Proc. 26th Euromicro Int. Conf. Parallel, Distrib. Network-Based Process. (PDP)*, Mar. 2018, pp. 339–343, doi: 10.1109/PDP2018.2018.00060.

[34] S. C. Purbarani and W. Jatmiko, "Performance comparison of bitcoin prediction in big data environment," in *Proc. Int. Workshop Big Data Inf. Secur. (IWBIS)*, May 2018, pp. 99–106, doi: 10.1109/IWBIS.2018.8471691.

[35] *Bitcoin Data*. Accessed: Jul. 11, 2022. [Online]. Available: https://99bitcoins.com/bitcoin/historical-price/

[36] Twitter Inc. *API Overview Twitter Developers*. Accessed: Jul. 11, 2022. [Online]. Available: https://dev.twitter.com/overview/api

[37] *Tweepy*. Accessed: Jul. 11, 2022. [Online]. Available: http://www.tweepy.org/

[38] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. 8th Int. Conf. Weblogs Social Media (CWSM)*, Ann Arbor, MI, USA, Jun. 2014, 2014.

[39] S. Yang, M. Paddrik, R. Hayes, A. Todd, A. Kirilenko, P. Beling, and W. Scherer, "Behavior based learning in identifying high frequency trading strategies," in *Proc. IEEE Conf. Comput. Intell. Financial Eng. Econ. (CIFEr)*, Mar. 2012, pp. 1–8.

[40] J. C. H. C. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3, Mar. 1992, pp. 279–292.

[41] *Performance Monitor Tool*. Accessed: Jul. 23, 2022. [Online]. Available: https://www.windowscentral.com/how-use-performance-monitor-windows-10

[42] *Variance Accounted For*. Accessed: Jul. 23, 2022. [Online]. Available: https://www.dcsc.tudelft.nl/jwvanwingerden/lti/doc/html/vaf.html

[43] *Coefficient of Determination*. Accessed: Jul. 23, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Coefficient_of_determination

[44] *Residual Sum of Squares*. Accessed: Jul. 23, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Residual_sum_of_squares

[45] *Total Sum of Squares*. Accessed: Jul. 23, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Total_sum_of_squares

[46] *Mean Absolute Percentage Error*. Accessed: Jul. 23, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Mean_absolute_percentage_error

[47] *Nash-Sutcliffe Coefficient*. Accessed: Jul. 23, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Nash-Sutcliffe_model_efficiency_coefficient

[48] *Root Mean Square Error*. Accessed: Jul. 23, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Root-mean-square_deviation

[49] *Weighted Mean Absolute Percent Error*. Accessed: Jul. 23, 2022. [Online]. Available: https://en.wikipedia.org/wiki/WMAPE

[50] *Statistical Forecast Errors*. Accessed: Jul. 23, 2022. [Online]. Available: https://blog.olivehorse.com/statistical-forecast-errors

SATTAROV OTABEK received the B.S. degree in information teaching methods from Gulistan State University, Syrdarya, Uzbekistan, in 2018, and the M.S. degree in computer engineering from Konkuk University, Chungju-si, South Korea, in 2021. He is currently pursuing the Ph.D. degree with the AI-Software Department, Gachon University, Seongnam, South Korea. His research interests include machine learning algorithms, deep reinforcement learning, and natural language processing.

JAEYOUNG CHOI (Member, IEEE) received the B.S. and M.S. degrees from the Department of Mathematics, Korea University, South Korea, in 2008 and 2013, respectively, and the Ph.D. degree from the Department of Electrical Engineering, KAIST, in 2018. From 2018 to 2020, he was an Assistant Professor at the Department of Automotive Engineering, Honam University, South Korea. Since 2020, he has been an Assistant Professor at the School of Computing, Gachon University, South Korea. His research interests include the intersection of applied mathematics and statistical inference, including social networks, wireless vehicular networks, and graphical models.