# Cryptocurrencies-Prediction-Forecast

Team 14

Cheng-Hao Ho Ch3561

Shuoting Kao sk4920

Wei-Ren Lai wl2777

## 1. Introduction

The motivation is to make a lot of money and provide a powerful tool for investors to increase their profit. Our platform predicts the crypto's trend based on cryptocurrency historical transaction data, wallet transaction data, and sentiment analysis from the various data sources, including Twitter, Crypto News, Reddit, and Google Trend. Besides, our goal is to trace top performers' activities and detect the key signal through their activities to decide whether we should follow their moves. We also leverage all the heterogeneous data to predict the important points, such as buying or selling dates/times. Our solutions include deep learning, association rule learning and time series analysis.

### 1.1. Problems

To achieve high accuracy prediction, we use various types of data sources as features for the machine learning model to learn and train. The problems we encountered include data source exploration, data flow, algorithm implementation are described in this Section 1.

### 1.2. Sentiment analysis on Twitters

Our initial approach was to produce the sentiment metric of the associated twitters through Google Natural Language Processing API. Nevertheless, we are unable to call the API in the worker nodes because of configuration issues. The issue is critical since we need to find a way to do sentiment analysis. Certainly, we can analyze posts through the NLP API in a master node, but this is not distributed processing and is an inefficient approach.

Besides, considering the round-trip time of API response is a waste, we decide to use the NLTK package instead. The obvious advantage is not only the local computation for each RDD, but the NTLK is free. On the other hand, Google NLP API costs according to the number of calls. After using the NTLK, we can produce the sentiment metrics for machine learning. In fact, the result demonstrates the correlation between bitcoin and the sentiment metric.

### 1.3. Reddit data collection

There are several ways to scrape the Reddit posts, such as Reddit official API, pushshift.io, and web-scraping. Among these options, pushshift.io suspends its service currently, and web-scraping is time-consuming in development. Accordingly, we select Reddit's official API to scrape posts.

However, there are two problems. The first one is data scarcity. The official API has the cap on the number of posts that we fetch once, which is 100. We only select bitcoin relevant posts among 100, so we only have 30-50 posts per fetching. Furthermore, the fetching results would be the same within a certain period, and this is adverse to the data abundance. Specifically, we are unable to fetch posts from 101 to 200.

The possible solution is using specified before/after fullnames, Reddit post ID, to fetch posts in different sections. Another straightforward solution is fetching Reddit posts in random order persistently to have much more plentiful data.

### 1.4. Google trend and Wikipedia reviews fetching

We are now directly scraping data from pytrend api and Wikipedia-API, but we are still confirming how often we need to grab data is the most appropriate choice. The process is important, as it may significantly influence our model accuracy. For example, crawling data every minute may cause our model overfitting. Currently, we do it once an hour.

### 1.5. Price prediction

To predict the crypto currency's trend, we collect numerical data such as transaction ticks, transaction pairs and asset transfer adta. Integrating other data like text and google trends mentioned above, we are able to do an

ablation study on training a deep-learning-based model. Furthermore, instead of predicting the price directly, we reframed the problem into a classification problem - to predict important trading points. To get all the labels we need, we designed an algorithm to collect the labels automatically.

## 2. Related Work

This section compares the prior art and the relevant part to our project. Also, the upside and downside of these approaches are discussed in this section.

2.1 Sentiment Analysis

Twitter has massive amounts of posts online. One of the most challenging is where you drain data from. For example, Dibakar Raj Pant and Prasanga Neupane [4] collect tweets related to bitcoin through specific Twitter accounts, such as BitcoinNews(@BTCTN), CryptoCurrency(@cryptocurrency), BitcoinMagazine (@BitcoinMagazine). On the contrary, our project uses real-time streaming for any sources on Twitter. Comparing these two methods, the upside of scraping the specified accounts is less noisy because their posts are relevant to Bitcoin. Still, the downside is biased since the source is monotonic.

| | Dibakar Raj Pant and Prasanga Neupane [4] | This work |
|---|---|---|
| Tweets source | Predefined bitcoin accounts | Any |
| Pro | All tweets from these accounts are relevant and meaningful | Unbiased and universal opinion on social media |
| Con | Lack of diverse sources would bias and decrement the accuracy | Noisy data. It deducts the accuracy of the model. Need a classifier to pre-filter the noise |

Table 2.1.1 Strategies comparison with prior art along with advantages and disadvantages.

In terms of data preprocessing, Dibakar Raj Pant and Prasanga Neupane classified and labeled tweets into three categories, positive, negative, and neutral. After that, they remove the duplicates and irrelevant tweets. Besides, they applied Regex and Weighted Search, which avoid hyperlink HTTP and emoji in tweets. These are proper ways to remove noise. On the other hand, our approach handles noise-filtering by limiting the minimal length of tweets and the presence of the keywords. The former is cleaner yet costly since labeling is expensive.

Feature extraction is a critical part of sentiment analysis. Dibakar Raj Pant and Prasanga Neupane adapted a 300 dimensions word embedding layer with Word2Vector and Bag-of-word. They combined five different classifiers with doing major voting to boost the accuracy. In contrast, feature extraction and sentiment analysis is handled by the NLTK package. Behind the scenes, the NLTK sentiment analyzer also tokenizes the sentences into words, removes stop words, and builds the frequency of words. In a sense, the voting of diverse classifiers can achieve higher accuracy supported by mathematical proof. Yet, we do not know what models are built inside the NLTK. For a project scope, the package is sufficient to complete the task.

2.2 Machine learning towards price prediction

In this project, we would like to focus on three methods - CNN-based models, RNN-based models and Association rule learning.

**RNN and CNN-based models**

To predict the price of financial products, there are several surveys on the performance of CNN-based and RNN-based models. [8] compared the performance of three methods - LSTM, Seq2Seq and WaveNet [7] by measuring the correlation between the predicted price and the actual price. According to the experiment result, the WaveNet out-performed the other two models. Due to the noise and uncertainty of stock price, [9] used sparse autoencoders with 1-D residual convolutional networks to de-noise the data, and then feed the data into a LSTM model to predict the stock price. [10] proposed Corr2Vec model that is a WaveNet architecture to extract the price feature embeddings. [11] further did a thorough review on 88 papers from 2015 to 2021 on predicting stock / foreign exchange price movements. The review covered CNN, LSTM, DNN, RNN, RL and other deep learning methods, and analyzed the following metrics: MAE, MSE, accuracy, sharpe ratio and return rate. Overally, CNN-based models performed better than RNN-based models.
Since the noise and uncertainty of the price, we propose another prediction target - to predict the important occasions, such as buying and selling points. After surveying, we found some authors also had similar goals.

[12] labeled buying and selling dates by finding the max and min price within each 11 days window, and using a CNN model to predict the dates. [13] proposed a deep learning framework based on a hybrid convolutional recurrent neural network to predict the important trading points. Their method outperformed the market by 278.46% for annualized return.

Through the survey, we decided to adopt one of the CNN models - WaveNet as our training model, because: 1. CNN performance is better overall. 2. Unlike RNN models, CNN models do not need to execute recurrent steps, which makes them faster. 3. WaveNet performs well on price prediction.

**WaveNet**

WaveNet is an audio generative model based on the PixelCNN [14] architecture, originally introduced to solve the audio waveform generation problem. The main components of WaveNet are causal convolutions. The purpose of the causal convolutions is to guarantee the ordering of the data features. That is, the model emission at timestep *t* cannot depend on any of the future timesteps [7]. WaveNet also adopts dilated convolution to skip input values with a certain step that allows it to grow exponentially with depth. It achieved dazzling results on music audio modeling and speech recognition.

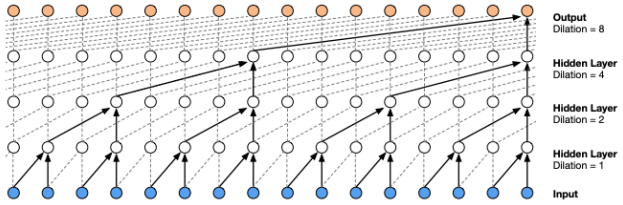


Figure 2.2.1. The architecture of WaveNet [7] with a stack of dilated causal convolutional layers..

**Association rule learning**

We are also interested in tracing the huge asset holders' activities. Intuitively, we thought we can use the association rule learning to filter out the important patterns. [15] gives us a sense of how to apply the association rule learning on financial data. After surveying the algorithms and feasibility, we decided our candidate algorithms are the following: FP-Growth [5] and PrefixSpan [6].

**Time series analysis**

To analyze time series data, we need a forecasting model to help us. The selection of the prediction model is of remarkable significance as it reveals the fundamental structure of the time series. [16] proposed a framework for stock market volatility based on parameters selection and the ARIMA model.

[17] also explained the relationship between google trend/Wikipedia views and Bitcoin prices. For the google trend analysis, we can see that practically the whole reaction comes from the positive feedback. However, for the Wiki reviews, we need to separate data between positive and negative feedback. Importantly, the result shows that the relationship is bidirectional, i.e. not only do the search queries influence the prices but also the prices influence the search queries.

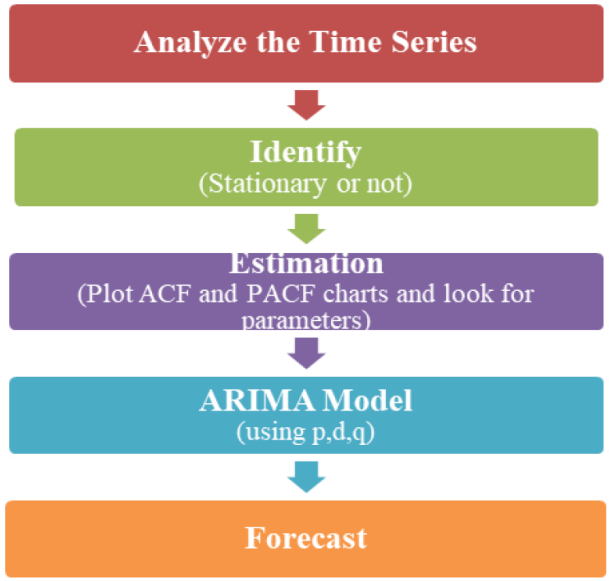


Figure 2.2.2. The framework for stock prediction in [16].

## 3. Current Progress

3.1 Sentiment Analysis on Twitter

We deploy a Twitter client and spark streaming on Dataproc shown in Fig 3.1.1 The current result is presented in Fig 3.1.2 and Fig 3.1.3. Within a 7-hours period, we can see some correlation between the metric and the price. Overall, the positive tone is slightly higher than the negative such that the overall (compound) is a little positive. During the period, the bitcoin price went high. Note that this is a short duration of observation. We will focus on the long evaluation period and do mathematical analysis through a correlation formula.

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

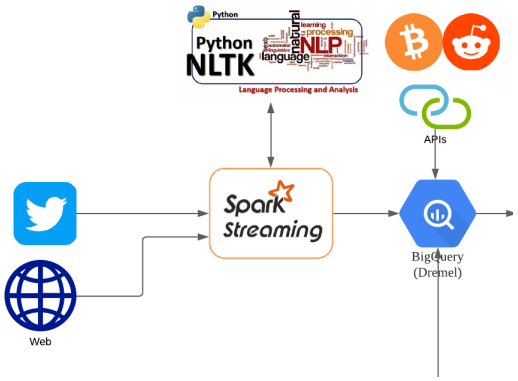


Fig 3.1.1 Twitter Streaming architecture

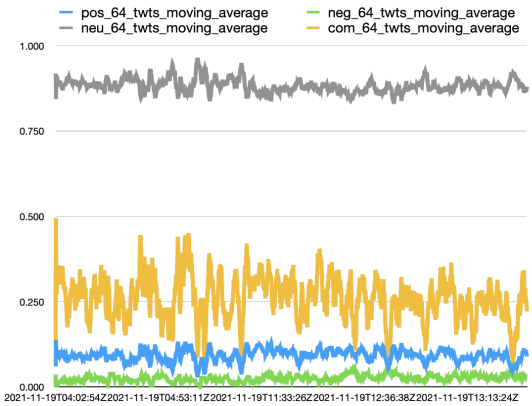


Fig 3.1.2 Sentiment analysis metric on tweets


Fig 3.1.3 Bitcoin price within the specific period

## 3.2 Sentiment analysis on Reddit

We have been populating the Reddit analyzed metric to Bigquery. Table 3.1.2 demonstrates the data content. Currently, the amount of data is insufficient to do meaningful correlative analysis. The implementation of the Reddit scraper is running on the virtual machine and is scheduled hourly by airflow. Figure 3.2.2 shows the DAG and expected running history through airflow.

| Row | id | time | pos | neg | neu | compound | title |
|---|---|---|---|---|---|---|---|
| 1 | r6ck2v | 2021-12-01 11:20:45 UTC | 0.199 | 0.176 | 0.625 | 0.101 | I wanna buy my 3rd 1000$ of Bitcoin but I'm scared o |
| 2 | r66vjx | 2021-12-01 05:00:01 UTC | 0.104 | 0.021 | 0.875 | 0.9371 | 🧬Unvaxxed Sperm🧬 \| $nuBTC \| Unvaccinated Sperr |
| 3 | r6fwmz | 2021-12-01 14:25:13 UTC | 0.125 | 0.0 | 0.875 | 0.8225 | Get 15 free euro in bitcoin with the Luno app |
| 4 | r67pii | 2021-12-01 05:49:40 UTC | 0.0 | 0.0 | 1.0 | 0.0 | Who are the worst guests at a dinner party? |
| 5 | r67ku0 | 2021-12-01 05:41:42 UTC | 0.0 | 0.0 | 1.0 | 0.0 | MonoX Finance bị hack, 31 triệu đô la bay màu |
| 6 | r5v99h | 2021-11-30 19:29:46 UTC | 0.0 | 0.0 | 1.0 | 0.0 | Bitcoin Rejected at $59K as Shiba Inu Explodes 27% |
| 7 | r5am3r | 2021-11-30 01:00:16 UTC | 0.0 | 0.0 | 1.0 | 0.0 | SATOSHI NAKAMOTO INU!! |

Table 3.2.1 Reddit data content in Bigquery

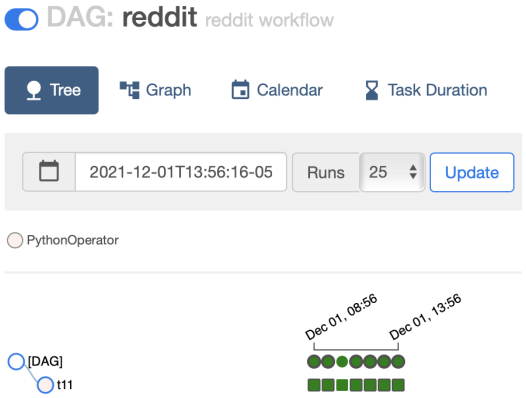


Figure 3.2.2 Reddit scraper running history on airflow.

## 3.3 Transaction tick and asset transfer real-time data collection

We didn't integrate the crawler into Airflow so far, so we use crontab to schedule the data crawling. In each 30 minutes, the crawler collects pairs (i.e. pairs of coin for exchanging) and ticks data into a NoSQL database, and the API also allows us to fetch the historical data. So far, we already have 1.6K pairs and 44 Million tick data. Also, we schedule the crawler to fetch huge asset transfer data every 10 seconds. We already have 790K records of huge asset transfers.
To visualize the real-time huge asset transfer activities, we built a simple web application to illustrate our result: http://34.75.110.94:8222/. It uses Flask as backend, binded with the SSE (server-side event). While the database detects the new data coming, it will trigger the SSE to push the data into the frontend.

## 3.4 Data labeling for machine learning model

To train a supervised learning model, we need input data and its corresponding labels. We designed an algorithm to collect the label automatically.
The core idea is to detect the peak and valley of the price oscillation. The algorithm is as below:

**Algorithm 1** Labelling Method

```
1: function CollectPeaks()
2:     peaks := array []
3:     price_max := 0
4:     price_min := inf
5:     dateStart := data[0].date
6:     profit := 0
7:     threshold := 0.2
8:     for (date, price in data)
9:         rise := (price - price_min) / price_mmin
10:        if (price > price_max)
11:            price_min = price
12:            profit = (price_max - price_min) / price_min
13:            if (profit >= threshold)
14:                peaks.add(dateStart)
15:        else if (price > price_max || rise >= threshold)
16:            price_min = price
17:            price_max = price
18:            dateStart = date

1: function CollectValleys()
2:     valleys := array []
3:     price_max := 0
4:     price_min := inf
5:     dateStart := data[0].date
6:     profit := 0
7:     threshold := 0.2
8:     for (date, price in data)
9:         drop := (price_max - price) / price_max
10:        if (price > price_max)
11:            price_max = price
12:            profit = (price_max - price_min) / price_min
13:            if (profit >= threshold)
14:                valleys.add(dateStart)
15:        else if (price < price_min || drop >= threshold)
16:            price_min = price
17:            price_max = price
18:            dateStart = date

1: function Labelling()
2:     peakDates := CollectPeaks()
3:     valleyDates := CollectValleys()
4:     flatDates := dates - peaks - valleys
5:     label "BUY" for peakDates
6:     label "SELL" for valleyDates
7:     label "HOLD" for flatDates
```

3.5 Google trend analysis

We have also parsed the information from google trend to see the relationship between the google trend and Bitcoin. In Fig 3.5.1, we can easily see that there exists a positive correlation between the transaction data and google trend data.

Refer to the method described in [16], we first need to make sure our data is stationary. Here, we applied ADF tests for the data. The p-values are all well below the 0.05 alpha level, therefore we can reject the null hypothesis. So the time series are stationary.

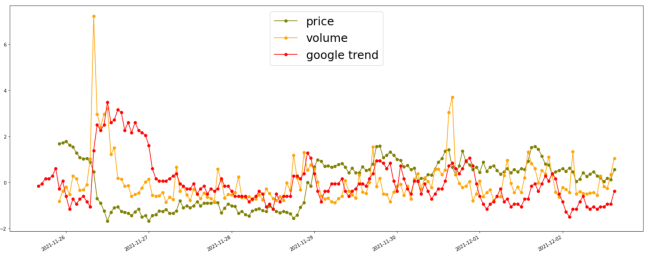


Fig 3.5.1 Bitcoin price and google trend within the specific period

```
Test Statistics: -3.0627571688590267
p-value: 0.029434929380172812
critical_values: {'1%': -3.435294916169133, '5%': -2.863723787379918, '10%': -2.5679326566037735}
Series is stationary
```

```
Test Statistics: -13.28837680822154
p-value: 7.448361558558406e-25
critical_values: {'1%': -3.4352212447633352, '5%': -2.863691278876476, '10%': -2.5679153445917}
Series is stationary
```

Fig 3.5.2 ADF tests for stationarity for Bitcoin data and google trend data

We then applied our stationary data to the Granger Causality test. The p-value for the test between google trend and Bitcoin (Fig 3.5.3) is smaller than significance level (0.05), we can therefore reject the null hypothesis and conclude that google trend results cause Bitcoin price. Likewise, we can assume that Bitcoin price causes google trend results (as shown in Fig 3.5.4). The relationship between the two data is bidirectional, which is the same as illustrated in [17].

```
Granger Causality
number of lags (no zero) 1
ssr based F test:         F=5.4218  , p=0.0200  , df_denom=1345, df_num=1
ssr based chi2 test:   chi2=5.4339  , p=0.0197  , df=1
likelihood ratio test: chi2=5.4230  , p=0.0199  , df=1
parameter F test:         F=5.4218  , p=0.0200  , df_denom=1345, df_num=1
```

Fig 3.5.3 Granger Causality between google trend and Bitcoin

```
Granger Causality
number of lags (no zero) 2
ssr based F test:         F=3.8644  , p=0.0212  , df_denom=1342, df_num=2
ssr based chi2 test:   chi2=7.7576  , p=0.0207  , df=2
likelihood ratio test: chi2=7.7354  , p=0.0209  , df=2
parameter F test:         F=3.8644  , p=0.0212  , df_denom=1342, df_num=2
```

Fig 3.5.4 Granger Causality between Bitcoin and google trend

## 4. The planned experiments

4.1 Sentiment analysis on tweets and Reddit

Long-term evaluation of the correlation between the price/volume and the metric is the next item we plan to do. In fact, we have launched 1-week Twitter streaming. Once we have it, we can run the mathematical correlation computation over the metric and the bitcoin price/volume to evaluate the result. This task can evaluate whether our sentiment analysis approach is appropriate or not. On the other hand, we would like to do the same evaluation on the Reddit sentiment metric. Currently, the airflow scheduler triggers a Reddit scraper. Once we have adequate data, we plan to do correlation analysis as well.

4.2 Machine Learning Model Training and Predicting

The price of the crypto currency changes dramatically, it might not be realistic to predict the price directly. Instead, we plan to train the models to predict the buying and selling point. Our model takes text data - twitter, reddit, and numerical data - google trend, transaction data and transfer data within a time interval as inputs, and the corresponding action (i.e. buy, sell or do nothing), as the outputs.
After a thorough survey on RNN and CNN models. We decided to adopt WaveNet [7] (Figure 3.3.1) as our training model.
We will also adopt a pre-trained word embedding (word2vec or globe) to encode out text data. Put the word embedding and the numerical data together, and with a given time period, we can construct our input data.

**Ablation study**

We have various sources of data. To validate the effectiveness of each data, we will conduct an ablation study on different combinations of input, to form the different feature embeddings. First of all, we will only use the transaction numeric data. Second, we will add other numeric data such as google trend and aggregated asset transfer data. Finally, we will combine text embedding into our model. By this incremental process, we will know which features are the most important and useful.

**Labeling**

To collect the correct action labels, we will write an algorithm to detect the peak and valley of the price, then mark the peak as selling point, the valley as the buying point and the others as doing nothing.

**Prediction Alert**

We will set up a daily-based schedule to train our model, and predict the result every hour. We also take the model output uncertainty into account. Once the model produces an output with high confidence, and the selling or buying point emerges, we will trigger the notifier to inform the users.

**Positive and negative feedbacks from google trend and Wikipedia reviews**

We will category our input from google trend and Wiki to see the volatility of data [17]. Currently, we have set up Cloud Functions and Cloud Schedulers on google cloud to parse information from both data sources once an hour. We also collect volume of transactions in each hour from Binance api as we know that there exists correlation between them from 3.5.

4.3 Association-Rule Learning

We are also interested in how big whales - large amounts of asset transfers into and from the trading platform - affect the crypto currency market. At the beginning, we attempted to excavate the relation between wallet addresses. However, we found that theoretically, the wallet owners can generate unlimited wallet addresses without any constraint, so it might be hard to use individual wallet addresses as an identity to execute the association-rule mining. Our new idea is to group the transfer amount into different buckets by the market share percentage of the transferred asset. And we also take the “direction” into account. That is, we consider the asset flow into the trading platform, such as Binance, CoinBase, as one direction, and the asset flow out from the trading platform as the other direction. And finally, we care about the influence introduced by those “big whales”, so we will encode the effect into three categories - up, neutral, and down. Put them together, we can do the association-rule mining. Since we collect the transfer data every 10 seconds, it might be noisy. We will find a proper window to aggregate the transfer data.
To calculate the result efficiently, we will use a distributed version of the algorithm on top of the Spark.

4.4 Pub/Sub subscription handler

In terms of publication and subscription, our plan is straightforward. We will build the subscription and publication features through Google pub/sub. The publisher is the alert producer on Dataproc, and the subscribers are users who want to receive the whale alerts.

After that, we will test the features by creating several events. Some are expected to be triggered, and some are not. Then, we check if we can receive the notification from the publisher upon the subscription of the threshold or the criteria gets hit.

## 5. References


[1] https://medium.com/general_knowledge/watch-the-whales-101-guide-to-wallet-tracking-8ff5799f3dc4

[2] https://github.com/manthanthakker/BitcoinPrediction

[3] https://col-jung.medium.com/how-to-make-realistic-cryptocurrency-price-predictions-436f3f6f54e3

[4] 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," *IEEE 3rd International Conference on Computing, Communication and Security (ICCCS). IEEE*, 2018, pp. 128–132.

[5] Jiawei Han, Jian Pei, Yiwen Yin. "Mining frequent patterns without candidate generation" *ACM SIGMOD Record, Volume 29, Issue 2*, June 2000, pp 1–12

[6] Jian Pei, Jiawei Han, B. Mortazavi-Asl, Jianyong Wang, H. Pinto, Qiming Chen, U. Dayal, Mei-Chun Hsu. "Mining sequential patterns by pattern-growth: the PrefixSpan approach" *IEEE Transactions on Knowledge and Data Engineering, Volume 16, Issue 11*, Nov. 2004

[7] Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." *arXiv preprint arXiv:1609.03499* (2016).

[8] Chun-Hung Cho , Guan-Yi Lee , Yueh-Lin Tsai , Kun-Chan Lan "Toward Stock Price Prediction using Deep Learning" *UCC '19 Companion: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing CompanionDecember* 2019 Pages 133–135

[9] Liu, Jialin, et al. "Stock prices prediction using deep learning models." *arXiv preprint arXiv:1909.12227* (2019).

[10] Marco Cerliani "Corr2Vec: a WaveNet architecture for Feature Engineering in Financial Market." https://towardsdatascience.com/corr2vec-a-wavenet-architecture-for-feature-engineering-in-financial-market-94b4f8279ba6

[11] Hu, Zexin, Yiqi Zhao, and Matloob Khushi. "A survey of forex and stock price prediction using deep learning." *Applied System Innovation* 4.1 (2021): 9.

[12] Asutosh Nayak. "Stock Buy/Sell Prediction Using Convolutional Neural Network" https://towardsdatascience.com/stock-market-action-prediction-with-convnet-8689238feae3

[13] Xinpeng Yu and Dagang Li. "Important Trading Point Prediction Using a Hybrid Convolutional Recurrent Neural Network." *Appl. Sci.* 2021, 11, 3984. https://doi.org/10.3390/app11093984

[14] van den Oord, Aaron, Kalchbrenner, Nal, and Kavukcuoglu, Koray. "Pixel recurrent neural networks." *arXiv preprint arXiv:1601.06759*, 2016a.

[15] Arvind Kalia, N. W. (2019). Association Rule Mining for Stock Data. *International Journal of Advanced Science and Technology, 28(19), 796 - 802*. Retrieved from http://sersc.org/journals/index.php/IJAST/article/view/2665

[16] S. M. Idrees, M. A. Alam and P. Agarwal, "A Prediction Approach for Stock Market Volatility Based on Time Series Data," in *IEEE Access*, vol. 7, pp. 17287-17298, 2019.

[17] Kristoufek, L. BitCoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era. Sci Rep 3, 3415 (2013). https://doi.org/10.1038/srep03415