Social Media based Cryptocurrency Sentiment Analysis and Visualization

Fei Yang
Ningbo Zhu
Mingzhi Zhu

UNIVERSITY OF ALBERTA

Multimedia Master Program
Department of Computing Science

Project Introduction

- The price of cryptocurrencies skyrocketed, many people invest their money into the cryptocurrency market.
- Like bitcoin, cryptocurrencies prices are impacted by many factors and forecast methodology may not be really accurate at a certain point.
- In this project, our team will try to use machine-learning based method to analyze the information from social media, like Twitter and predict the price increasing rate of Bitcoin.

Misunderstanding for Price Prediction

- Past price is independent of the future price
- Past price may lead a increase for future price but there is no direct relationship between past price and future price
- Future price only depends on how much people want to buy it. In another word, it depends on how the society trust it.
- Using past price to predict future price has logic problem

Solution

- Convert the target from price to the price increasing rate

- Using other society trust information for model training
  - Social media information
  - Hash rate

Proposed Method- Data Collecting:Twint

- Twint: A Python scraping library for collecting the Twitter users' data.
- Raw user data includes all tweets searched by keywords followed by their date.

{"id": 1211789890310066176, "conversation_id": "1211789890310066176", "created_at": "2019-12-30 16:23:18 ", "date": "2019-12-30", "time": "16:23:18", "timezone": "-0700", "user_id": 339061487, "username": "apompliano", "name": "Pomp 🌪", "place": "", "text": "Retweet to remind everyone that Bitcoin survived another year 🙏", "language": "en", "mentions": [], "urls": [], "photos": [], "replies_count": 94, "retweets_count": 910, "likes_count": 1910, "hashtags": [], "cashtags": [], "link": "https://twitter.com/APompliano/status/1211789890310066176", "retweet": false, "quote_url": "", "video": 0, "thumbnail": "", "near": "", "geo": "", "source": "", "user_rt_id": "", "user_rt": "", "retweet_id": "", "reply_to": [], "retweet_date": "", "translate": "", "trans_src": "", "trans_dest": ""}

- We also collect all year bitcoin prices and their hash rate from the Coindesk webpage and the Quandl webpage.

Date	Closing Price (USD)	Hash Rate
2019/1/1	3715.564441	43291797
2019/1/2	3882.304754	41615985
2019/1/3	3795.592741	42174589
2019/1/4	3812.381108	43291797

Proposed Method- Data preprocessing

- NLTK: A language analysis model to convert tweets into formalized keywords

  Retweet to remind everyone that Bitcoin survived another year 🙏

  ['retweet', 'remind', 'everyone', 'bitcoin', 'survive', 'another', 'year', '🙏', '🟫']

- Count most frequent keywords

  [('bitcoin', 20017), ('#bitcoin', 5552), ('’', 4839), ('btc', 2600), ('buy', 1994),

- Pick the keywords may influence the price

  key_word_list = ['btc','buy','go','people','year','time','like','...','money','price','crypto']

- Make one-hot vector for training

  'key_word_table': [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

Proposed Method- Data preprocessing

- vaderSentiment: A language analysis model to analyze the sentiment intensity.

Retweet to remind everyone that Bitcoin survived another year 🙏

```
'Sentiment_Analysis': 0.5106
```

- +1 is the strong positive sentiment, -1 is the strong negative sentiment, 0 is neutral sentiment.
- Use the sentiment analysis as a new feature for model training.

Proposed Method- Data Structure

- Features We Have: BTC Price, Blockchain Hashrate, Tweet Sentiment Analysis, Tweet Keywords.

- Different combination of features as input to find out which feature combination is useful.

  (e.g. Hashrate+Tweet Sentiment Analysis+Tweet Keywords)

Proposed Method- Data Structure

Time Range	Raw Data (Hashrate+Tweet Sentiment Analysis+Tweet Keywords)	Label (BTC Price Increasing Rate)
7 days (eg. 2019/06/23 ~ 2019/06/29)	[[51933576.06], [66666505.44], [65193212.5], [53775192.23], [60773333.69], [55539973.18], [64293187.57], [{[14, 10, 10, 7, 4, 10, 9, 3, 9, 10, 4], "2019-06-23", 0.19443106796116505}, {[15, 7, 7, 9, 11, 11, 3, 3, 2, 10, 6], "2019-06-24", 0.2174368421052632}, {[27, 7, 17, 11, 7, 16, 9, 14, 3, 10, 6], "2019-06-25", 0.19670406976744184}, {[22, 16, 20, 14, 16, 16, 10, 17, 6, 29, 8], "2019-06-26", 0.07637582938388628}, {[9, 11, 12, 2, 5, 4, 8, 10, 3, 12, 5], "2019-06-27", 0.10210416666666666}, {[7, 9, 8, 9, 6, 6, 5, 7, 3, 9, 4], "2019-06-28", 0.23671168831168823}, {[10, 5, 2, 4, 5, 8, 2, 5, 9, 2, 4], "2019-06-29", 0.150774626865671 6}]]	-0.0908
7 days (eg. 2019/06/24 ~ 2019/06/30)	[[66666505.44], [65193212.5], [53775192.23], [60773333.69], [55539973.18], [64293187.57], [68631991.64], [{[15, 7, 7, 9, 11, 11, 3, 3, 2, 10, 6], "2019-06-24", 0.2174368421052632}, {[27, 7, 17, 11, 7, 16, 9, 14, 3, 10, 6], "2019-06-25", 0.19670406976744184}, {[22, 16, 20, 14, 16, 16, 10, 17, 6, 29, 8], "2019-06-26", 0.07637582938388628}, {[9, 11, 12, 2, 5, 4, 8, 10, 3, 12, 5], "2019-06-27", 0.10210416666666666}, {[7, 9, 8, 9, 6, 6, 5, 7, 3, 9, 4], "2019-06-28", 0.23671168831168823}, {[10, 5, 2, 4, 5, 8, 2, 5, 9, 2, 4], "2019-06-29", 0.150774626865671 6}, {[8, 4, 6, 6, 2, 2, 4, 6, 5, 4, 1], "2019-06-30", 0.23243148148148146}]]	-0.0437
...	...	...

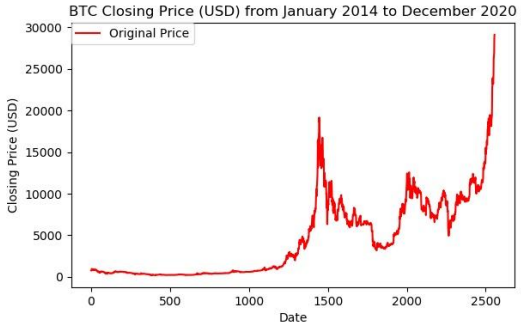
7-day hashrate

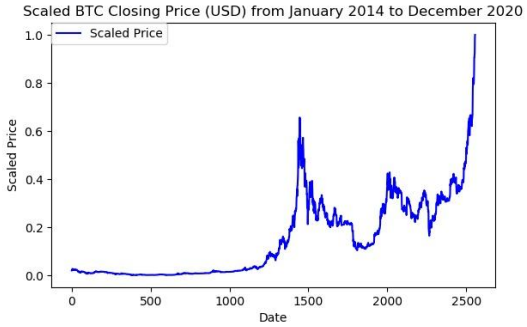
7-day Tweet Sentiment analysis and tweet keywords

8th day's price increasing rate

Proposed Method- Data Normalization

- For some numerical variables, because their value ranges vary greatly, it is often necessary to organize their value ranges into a reasonable range.
- MinMaxScaler
  - $$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$
  - Min-Max only change the the value range. It does not change the distribution





Proposed Method- Data Structure

Time Range	Raw Data (Hashrate+Tweet Sentiment Analysis+Tweet Keywords)	Input Data (Normalized)	Label (BTC Price Increasing Rate)
7 days (eg. 2019/06/23 ~ 2019/06/29)	[[51933576.06], [66666505.44], [65193212.5], [53775192.23], [60773333.69], [55539973.18], [64293187.57], [{[14, 10, 10, 7, 4, 10, 9, 3, 9, 10, 4], "2019-06-23", 0.19443106796116505}, {[15, 7, 7, 9, 11, 11, 3, 3, 2, 10, 6], "2019-06-24", 0.2174368421052632}, {[27, 7, 17, 11, 7, 16, 9, 14, 3, 10, 6], "2019-06-25", 0.19670406976744184}, {[22, 16, 20, 14, 16, 16, 10, 17, 6, 29, 8], "2019-06-26", 0.07637582938388628}, {[9, 11, 12, 2, 5, 4, 8, 10, 3, 12, 5], "2019-06-27", 0.10210416666666666}, {[7, 9, 8, 9, 6, 6, 5, 7, 3, 9, 4], "2019-06-28", 0.23671168831168823}, {[10, 5, 2, 4, 5, 8, 2, 5, 9, 2, 4], "2019-06-29", 0.1507746268656716}]]	[[-1], [0.7645901], [0.58813109], [-0.77942624], [0.05875406], [-0.56805517], [0.48033344], [1], [90, 84, 127, 174, 81, 73, 56, 48], 0.19443106796116505, 0.2174368421052632, 0.19670406976744184, 0.07637582938388628, 0.10210416666666666, 0.23671168831168823, 0.1507746268656716]	-0.0908
7 days (eg. 2019/06/24 ~ 2019/06/30)	[[66666505.44], [65193212.5], [53775192.23], [60773333.69], [55539973.18], [64293187.57], [68631991.64], [{[15, 7, 7, 9, 11, 11, 3, 3, 2, 10, 6], "2019-06-24", 0.2174368421052632}, {[27, 7, 17, 11, 7, 16, 9, 14, 3, 10, 6], "2019-06-25", 0.19670406976744184}, {[22, 16, 20, 14, 16, 16, 10, 17, 6, 29, 8], "2019-06-26", 0.07637582938388628}, {[9, 11, 12, 2, 5, 4, 8, 10, 3, 12, 5], "2019-06-27", 0.10210416666666666}, {[7, 9, 8, 9, 6, 6, 5, 7, 3, 9, 4], "2019-06-28", 0.23671168831168823}, {[10, 5, 2, 4, 5, 8, 2, 5, 9, 2, 4], "2019-06-29", 0.1507746268656716}, {[8, 4, 6, 6, 2, 2, 4, 6, 5, 4, 1], "2019-06-30", 0.23243148148148146}]]	[[0.7354092], [0.53707672], [-1], [-0.05792072], [-0.76242784], [0.41591672], [1], [84, 127, 174, 81, 73, 56, 48], 0.2174368421052632, 0.19670406976744184, 0.07637582938388628, 0.10210416666666666, 0.23671168831168823, 0.1507746268656716, 0.23243148148148146]	-0.0437
...	...	...	...

7-day hashrate

7-day Tweet Sentiment analysis and tweet keywords

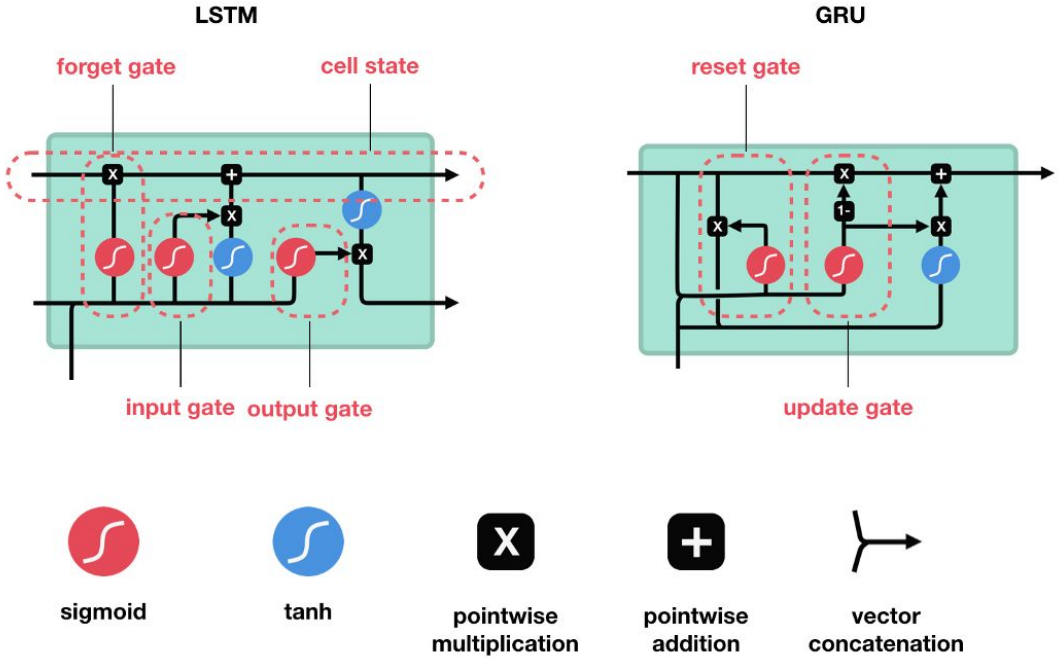
8th day's price increasing rate

Proposed Method- Training Model

- Long short-term memory (LSTM)
- Gated Recurrent Units (GRU)
- Bidirectional Long short-term memory (Bi-LSTM)

Proposed Method- Training Model

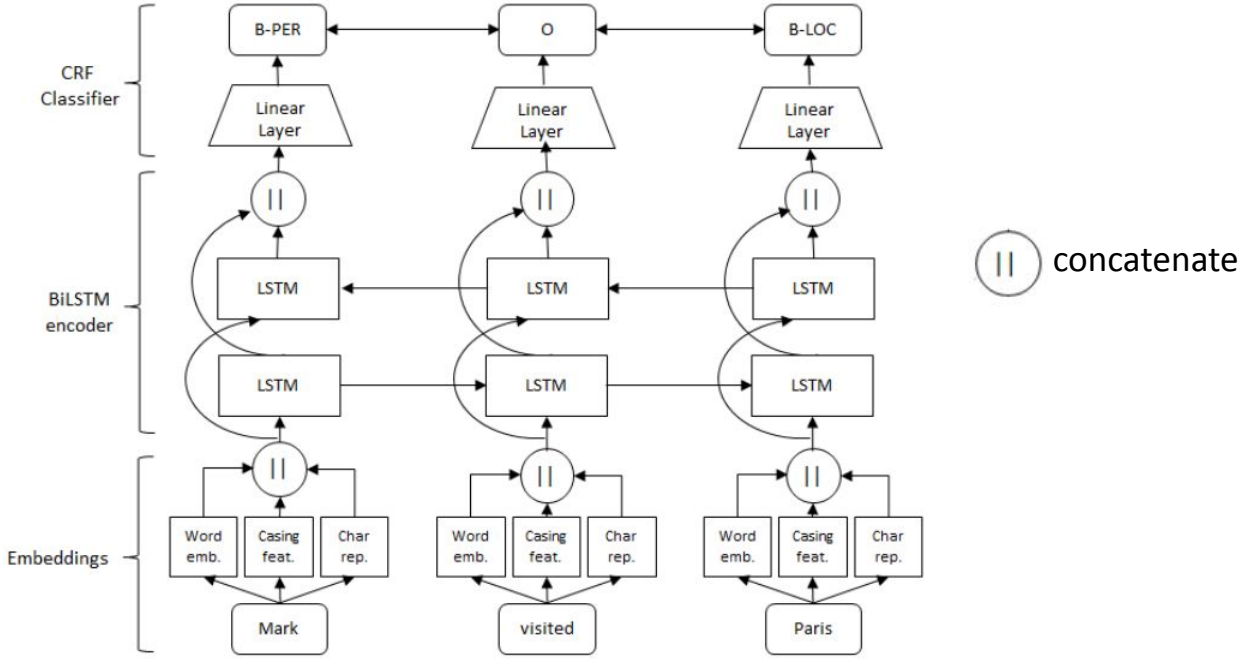
- Long short-term memory (LSTM) and Gated Recurrent Units (GRU)



https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Proposed Method- Training Model

- Bidirectional Long short-term memory (Bi-LSTM)



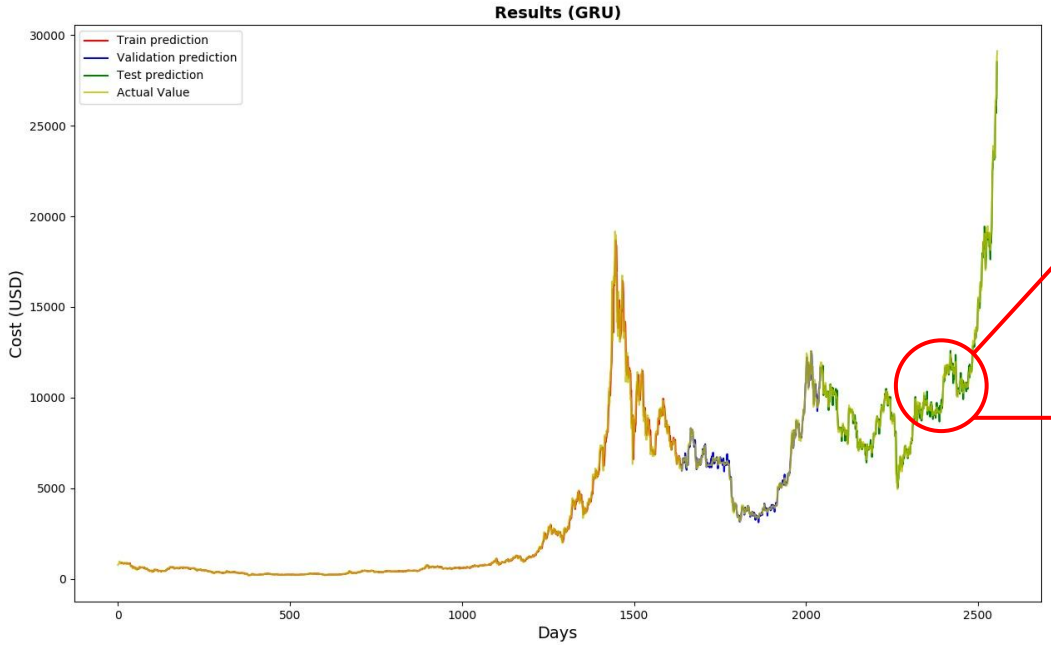
Reimers, N., & Gurevych, I. (2017). Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. arXiv preprint arXiv:1707.06799.

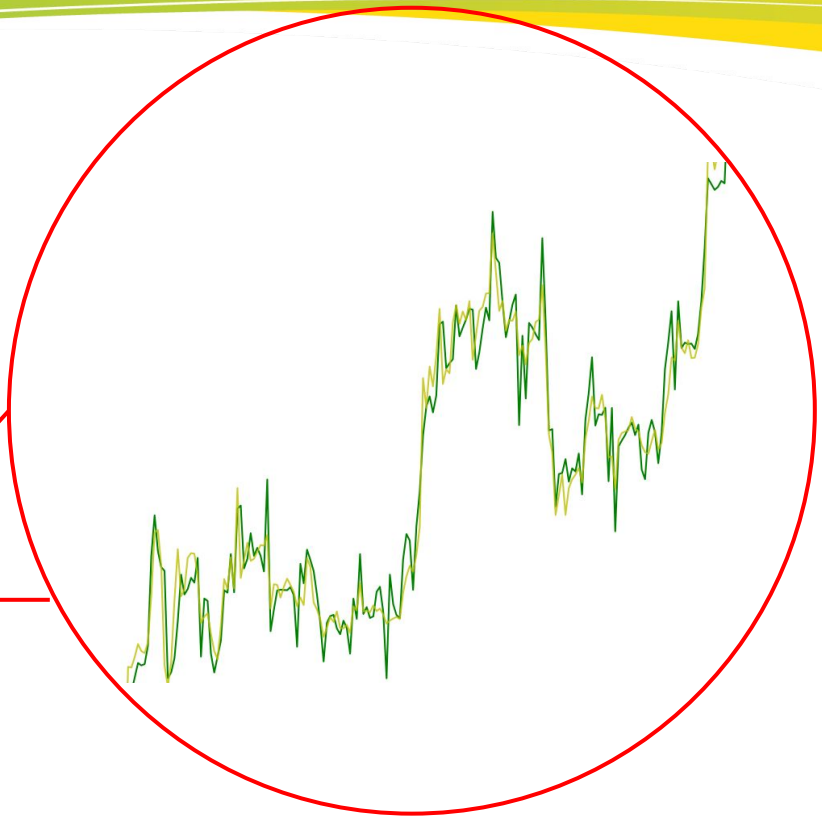
Price Predicted Result

- **Sign Correct**: The amount of result which correctly predict the fluctuation of price is up or down.
  - up: increasing rate > 0
  - down: increasing rate < 0
- **Error < 0.05**: The error between the predicted increasing rate and the groundtruth is less than 5%.
- Predicted 8th day's Price = 7th day's price * (Predicted Price Increasing Rate + 1)

Features		LSTM			bi-LSTM			GRU		
		Sign Correct	Error < 0.05	Sign Correct & Error < 0.05	Sign Correct	Error < 0.05	Sign Correct & Error < 0.05	Sign Correct	Error < 0.05	Sign Correct & Error < 0.05
Price	Train	0.77	0.93	0.71	0.77	0.93	0.71	0.80	0.94	0.74
	Validation	0.76	0.92	0.70	0.77	0.93	0.70	0.76	0.94	0.71
	Test	0.75	0.95	0.71	0.76	0.94	0.70	**0.77**	0.95	**0.72**
Hashrate	Train	0.57	0.85	0.48	0.54	0.85	0.46	0.53	0.85	0.46
	Validation	0.52	0.85	0.47	0.51	0.86	0.45	0.54	0.86	0.47
	Test	**0.56**	0.88	**0.48**	0.50	0.86	0.44	**0.56**	0.87	**0.48**
Hashrate+Tweet Sentiment Analysis+ Tweet Keywords	Train	0.52	0.85	0.45	0.53	0.85	0.45	0.53	0.85	0.45
	Validation	0.50	0.85	0.45	0.51	0.85	0.45	0.52	0.85	0.48
	Test	0.53	0.90	0.48	0.52	0.90	0.47	**0.54**	0.89	**0.50**

Price Predicted Result





Data visualization for using past 7-days price and GRU to predict price increasing rate

Price Predicted Result (After shuffle dataset)

- **Sign Correct**: The amount of result which correctly predict the fluctuation of price is up or down.
  - up: increasing rate > 0
  - down: increasing rate < 0
- **Error < 0.05**: The error between the predicted increasing rate and the groundtruth is less than 5%.
- Predicted 8th day's Price = 7th day's price * (Predicted Price Increasing Rate + 1)

Features	Train/ Validation/ Test	LSTM			bi-LSTM			GRU		
		Sign Correct	Error < 0.05	Sign Correct & Error < 0.05	Sign Correct	Error < 0.05	Sign Correct & Error < 0.05	Sign Correct	Error < 0.05	Sign Correct & Error < 0.05
Price	Train	0.7830	0.9675	0.7560	0.8197	0.9828	0.8056	0.7793	0.9510	0.7382
	Validation	0.7083	0.8627	0.6127	0.7353	0.8603	0.6446	0.7819	0.9118	0.7132
	Test	0.7725	0.9235	0.7098	0.7706	0.9235	0.7000	0.7863	0.9255	0.7235
Hashrate	Train	0.5383	0.8676	0.4746	0.5303	0.8719	0.4641	0.5267	0.8657	0.4629
	Validation	0.4975	0.8284	0.4093	0.4779	0.8284	0.3995	0.4951	0.8284	0.4093
	Test	0.5412	0.8549	0.4588	0.5510	0.8549	0.4725	0.5412	0.8549	0.4588
Price + Hashrate	Train	0.7474	0.9761	0.7284	0.7836	0.9859	0.7713	0.7284	0.9626	0.7032
	Validation	0.7034	0.8725	0.6029	0.6838	0.8431	0.5760	0.6961	0.8750	0.6078
	Test	0.7196	0.9039	0.6431	0.7235	0.8941	0.6431	0.7392	0.9098	0.6706
Price+Hashrate+Tweet Sentiment Analysis+ Tweet Keywords	Train	0.5310	0.8688	0.4697	0.5395	0.8700	0.4776	0.5671	0.8749	0.4954
	Validation	0.5049	0.8309	0.4314	0.4632	0.8309	0.3922	0.5294	0.8260	0.4485
	Test	0.5353	0.8549	0.4529	0.5235	0.8608	0.4529	0.5373	0.8510	0.4706

Evaluation

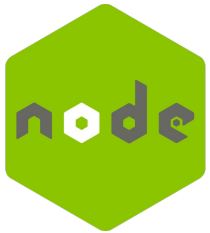
- Root-Mean-Square Error (RMSE)

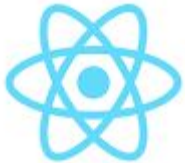
$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{n} (y_i - y_{i_p})^2}$$

Features	RMSE	LSTM		bi-LSTM		GRU	
		Groundtruth	Scaled	Groundtruth	Scaled	Groundtruth	Scaled
Price	Train RMSE	306.92	0.31	315.91	0.31	298.14	0.31
	Validation RMSE	355.56	0.34	360.56	0.34	337.44	0.34
	Test RMSE	479.25	0.33	472.48	0.33	469.90	0.33
Hashrate	Train RMSE	264.77	0.32	264.52	0.32	265.69	0.32
	Validation RMSE	279.75	0.35	282.78	0.35	279.28	0.35
	Test RMSE	354.78	0.34	354.50	0.34	354.66	0.34
Price + Hashrate	Train RMSE	302.99	0.31	299.11	0.31	303.33	0.31
	Validation RMSE	340.97	0.34	354.94	0.34	337.89	0.34
	Test RMSE	472.56	0.34	509.86	0.33	482.64	0.34
Price+ Hashrate+Tweet Sentiment Analysis+	Train RMSE	264.79	0.32	264.60	0.32	267.53	0.32
	Validation RMSE	282.4	0.35	281.15	0.35	290.72	0.35
	Test RMSE	357.17	0.34	357.72	0.34	359.60	0.34

Data Visualization

- MongoDB
- NodeJS Server
- React





MongoDB

Cloud-base DB

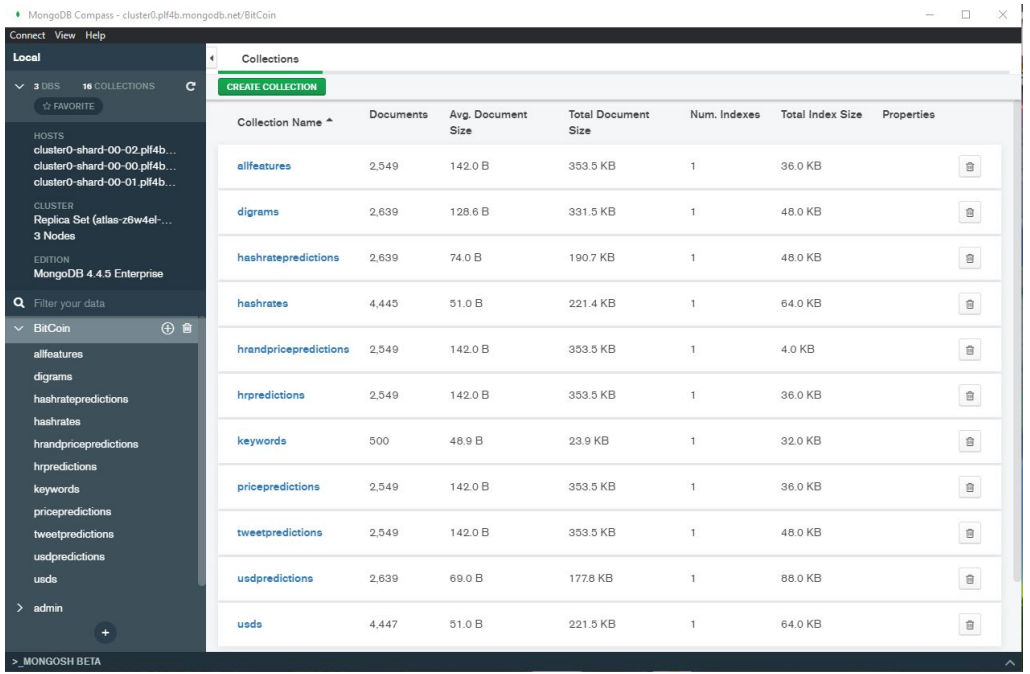
- Distributed DB
- Access anywhere
- High Security
- support dynamic data type
- Free

Collections

- USD
- Tweet prediction
- Price prediction
- Hash-rate prediction
- Mix combination prediction
- Keywords
- Digram

Total Size:

- approximate 4MB
- Continually growing



NodeJS Server

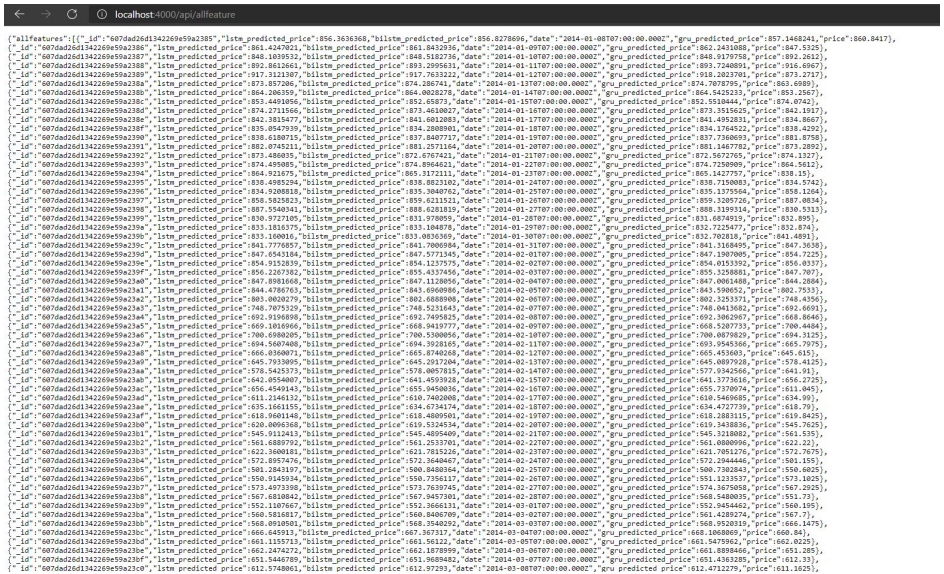
Server side application

Following REST rules

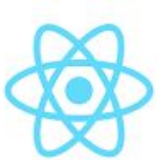
NodeJS server good practice

- Non-blocking Input/Output and asynchronous request handling
- using microservices ecosystem solution
- support multi-environment development
- Following good generic coding practices such as DRY and Separation of Concerns principles

Support strong type data model



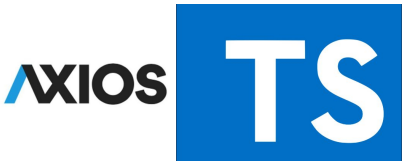
React

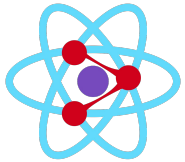


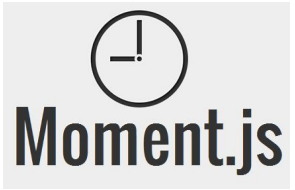
Modern web development tool

Libraries used:

- Typescript
- Sass
- Axios
- Ant-Design(AntD)
- Recharts
- React-wordcloud
- React-diagrams
- React-Router-Dom
- Moment.js







[Web Demo]

Future work

Models improvement

- More features input
  - std of the price
  - information from other social media, eg. Facebook
- Try our method on other cryptocurrency, eg. ETC
- Change model structure, hyperparameter Fine-tuning
- Change train data time range
- Balance the dataset

Web Improvement

- User-friendly
  - Interactive
  - Animation
- Informational
- Server host

Conclusion

- From the project, we can conclude that due to the past price is independent of the future price, using the future price as the target, the result always unsatisfactory.
- Convert the target from price to the price increasing rate has unexpected accuracy improvement.
- Social media information and hash rate are not useful for cryptocurrency price increasing rate prediction.
- Compare to the traditional stock price prediction, our cryptocurrency prediction method has better accuracy and timeliness.
- Our web app is convenient for users to analyze the relationship between social media and the cryptocurrency market and have a reasonable prediction of the future price.

Q&A