Bitcoin Price Prediction Through Opinion Mining

Germán Cheuque Cerda
gacheuque@uc.cl
Pontificia Universidad Católica de Chile
Santiago, Chile

Juan Reutter de La Maza
jreutter@ing.puc.cl
Pontificia Universidad Católica de Chile
Santiago, Chile

ABSTRACT

The Bitcoin protocol and its underlying cryptocurrency have started to shape the way we view digital currency, and opened up a large list of new and interesting challenges. Amongst them, we focus on the question of how is the price of digital currencies affected, which is a natural question especially when considering the price roller-coaster we witnessed for bitcoin in 2017-2018. We work under the hypothesis that price is affected by the web footprint of influential people, we refer to them as crypto-influencers.

In this paper we provide neural models for predicting bitcoin price. We compare what happens when the model is fed only with recent price history versus what happens when fed, in addition, with a measure of the positivity or negativity of the sayings of these influencers, measured through a sentiment analysis of their twitter posts. We show preliminary evidence that twitter data should indeed help to predict the price of bitcoin, even though the measures we use in this paper have a lot of room for refinement. In particular, we also discuss the challenges of measuring the correct sensation of these posts, and discuss the work that should help improving our discoveries even further.

KEYWORDS

Bitcoin, Recurrent Neural Networks, Twitter, Sentiment Analysis, Price Prediction.

ACM Reference Format:

Germán Cheuque Cerda and Juan Reutter de La Maza. 2019. Bitcoin Price Prediction Through Opinion Mining. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3308560.3316454

1 INTRODUCTION

The release of the Bitcoin Protocol [22] unleashed bitcoin as the world's first decentralized cryptocurrency. While it took some time to gain traction, in the last few years a huge number of alternate cryptocurrencies have appeared, and they have become popular to the point that almost every financial agent has at least considered investing in them. At the time of writing, the price of bitcoin has stabilized around 3500 USD, after an all time high of almost 20000 USD just two years ago. But while this price bubble is now understood to have burst –at least partially–, some other concerns about the condition of the bitcoin market are still not well understood.

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316454

One of the most recurrent question about the bitcoin market has to do with its volatility: what factors influence the variation in bitcoin price? There has been some research that links bitcoin price with Google searches [24], which is remarkable because other comparable assets such as gold are known to have very little correlation with this indicator. However, in conversation with people in the cryptocurrency market we were suggested another indicator: tweets and posts from a number of international influencers.

The goal of this research is to validate the hypothesis that the price of bitcoin is affected by the web footprint of popular actors in the international cryptomarket. Preliminary research has told us that the amount of digital material does not appear to be correlated with bitcoin price, so we cannot settle for a simple answer like in [24], But we can do more: what if positive messages affect the price in a positive way, and negative messages in a negative way? This idea gives us a clear map of what needs to be done: First, gather tweets by the most influential users in the cryptocurrency world. Next, analyze which of these messages speak positively of bitcoin and which ones do not, and then show that this data actually does affect bitcoin's price.

Influential users can be selected in a greedy way, starting from some well known popular users, and we can use sentiment analysis to classify the tweets of these users. But how can we show that this data affects the price? We proceed as in [1] and [2], and compare the prediction capabilities of a model that uses only the information of bitcoin's price in the past with a model that incorporates this external information into the time series.

In this paper we show promising preliminary results, showing that incorporating twitter data can lead to better forecasting models. However, we believe there is still much room for improvement, as there are many different ways to follow our general map, and we have only studied a small number of them. The study is presented as follows: in Section 2 we deepen our objectives, present the data we are working with and explain the type of prediction we will focus in the rest of the paper. Section 3 the architecture of our model, discusses the needed of an appropriate scaling and presents the results of the realistic problem of predicting price one day in future, with their new complications. Section 4 introduces the opinion information obtained from twitter and how their inclusion impacts on our results. Finally, Section 5 presents our conclusions and the proposed further work.

2 ON PREDICTING PRICE OF MONETARY GOODS

Predicting the price of monetary goods is a problem that has been thoroughly studied for several decades and from a series of different areas. As such, it is almost impossible to give a full overview of all the different versions of these problems, or the techniques used to solve them.

However, we do have one important aspect to discuss: the problem of predicting the immediate price of monetary goods that are subject to a high volume of transactions can be dealt with (up to a very reasonable error) standard machine learning techniques. For the case of Bitcoin, in Section 2.2 we provide a neural network that can effectively predict the price of bitcoin in the next 5 minutes, and we test it to show that the error is indeed quite small [1].

For this reason we focus instead on other problems which cannot be directly dealt with standard techniques. One problem is predicting the price of Bitcoin after 24 hours of reading the last input, and the other problem has to do with training a model that can properly simulate bitcoin prices for a long period of time. We define these in more detail in Section 2.3

2.1 Samples and range of prediction

We draw Bitcoin price data from bitcoincharts.com, and focus on the interval between July 1st and December 30th over 2018. The data contains information about the opening and closing prices of Bitcoin on intervals of 5 minutes, giving us a total of 58451 samples. In all our models we use the first 37339 data points as training data[2], and use the remaining to test our predictions. The Bitcoin price evolution over the referred dates is presented in Figure 1.

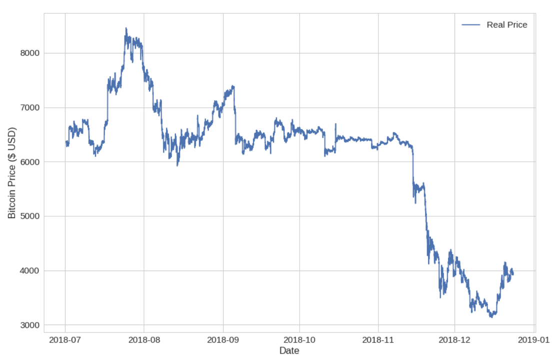

Figure 1: Bitcoin price time series.

As everyone in industry knows, the price of Bitcoin is still far from reaching a stable behavior. And while the high frequency of registries is definitely good news for learning models predicting its price, it also carries a curse: the large number of variations can be difficult to learn, and occasional points with high variablity would tend to be magnified by the necessary scaling we must perform on the data. This means that predictors may either choose to focus on these high variability changes, and therefore loosing constant small changes, or focus on small changes and treating big raises or drops as outliers. A way to solve this noisy effect without losing our high frequency chart is to introduce a new quantity, we refer to the *moving average* measure, defined by equation 1

$$MA = \frac{1}{N}\sum_{i=1}^{N} p_i \tag{1}$$

where p_i is the current price at date i and N indicates the number of samples that are considered on average. The idea behind this measure is to work with a constant window of size N and consecutively move their bounds to get a continue measure of how the price varies on average. Figure 2 show us how moving average captures the variability of the price with a smoother approximation. We remark that moving average is one of the most used metrics in stock exchange markets, to the point that its inclusion is considered folklore.

Figure 2: Comparison between price and moving average 1-day measure.

2.2 Predicting the next data point.

If all we want is to predict the next data point in the series, we can obtain reasonable prediction by using state of the art neural networks. Many studies have addressed the problem of forecasting using LSTM in the most diverse fields. Some studies have worked on e-commerce [5], other have made the same analyzing the weather [6], even studying the human blood pressure [7]. Other studies have made important advances in popular computer tasks, such as question Answering [8], [9] or speech recognition [10] and even in the development of autonomous vehicles [11] with the use of LSTM.

We chose to build a Recurrent Neural Network based in LSTM. Our network consist of five layers, each one of 30 neurons, except for the first one, with 288 neurons. Each one with their respective dropout regularization, the final two layers are fully connected to retrieve a single value for prediction. Further details about configuration and design decision can be found in Section 3. The network is fed with a moving window N of 288 data points, i.e., one full day of observations, on the other hand, the output is trained to predict the price of the next data point in the sequence. The predictor is

[1]But then, if one can predict the price, why is not everyone arbitraging the market? The problem is that, if one wants to arbitrage bitcoin prices by taking advantage of a model that predicts the price in the next 5 minutes, one would need a prediction that is essentially perfect.
[2]We always train with a random 20% of the sample reserved for cross validation.

trained to minimize both the root mean square (RMSE) and the mean absolute error (MAE).

As we briefly comment in last section, we also scale the values that feed our network, as it is known to accelerate the learning process of machine learning algorithms (as commented in [3]). Since we just need to scale the scalar value of a prediction, we use a simple MinMax function, applied over the total number of observations, as defined next:

$$X_{scaled} = \frac{X_{original} - X_{min}}{X_{max} - X_{min}} \tag{2}$$

where X_{min} and X_{max} are the minimum and maximum value of the range scaled, respectively. Clearly, this function maps the original values to the range [0,1]. With those considerations we train our model.

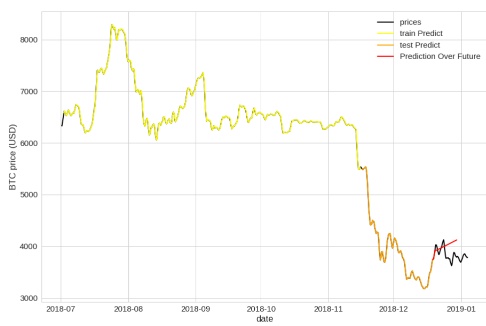

Figure 3: Network predictions after training. The black line represents the moving average measure introduced on last subsection, the yellow line the predicted value on the training part of the data, and the orange line the prediction on the test set. The part with the red line is zoomed-in and explained in Figure 4.

Figure 3 shows the quality of our regression. As we see, the model is able to predict the price of the next point with great accuracy, achieving a score of 9.91 on RMSE in training set and 10.87 on RMSE in testing set after recovering the scale.

2.3 Problems studied in this paper.

As we have seen, the idea of predicting the next point of a time series is a simple task for a neural network, and thus we prefer to focus on two more difficult problems.

The first problem we study is to predict the price of bitcoin in the next day (that is, to predict the price after 288 data points). This comes as a natural extension of the regression we have just adjusted, and it is both realistic and useful. In the following we refer to this problem as the *Next-day price*.

The second problem we study has more to do with our intention of using external data to predict prices. The motivation comes from Figure 3. If we see the orange line, every orange point takes as input the moving window of the latest 288 real data points. But what would happen if we start feeding the network with its own predicted prices? How well can we simulate the moving curves of bitcoin prices? We refer to this problem as the *Price Simulation* problem.

To understand the challenges behind Price Simulation, Figure 4 shows the poor performance of the model trained for Figure 3 when we start feeding the predicted data. Clearly, one expects that this price simulation cannot be carried away with price data only, as the high-raises and low-drops of bitcoin are probably related to external phenomena. This is where our analysis of twitter data comes in handy; we provide more details in Section 4.

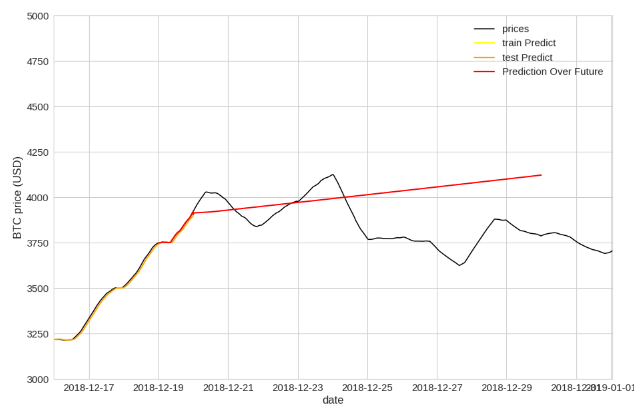

Figure 4: Simulation of Bitcoin Price. As the red curve shows, once we start feeding the network's own prediction as data we immediately loose generality and prediction power, and we get quickly stuck in an average value.

3 NEXT-DAY AND PRICE SIMULATION USING PRICE DATA

In this section we construct models that solve both the next-day price and the price simulation problem, and train them using price data only. More specifically, we want to answer the following two questions. First, can we get the same good results for next-day price as we got when predicting the next data point in the price history? The second question has to do with price simulation. As we have mentioned, in this setting we do not expect good results on the simulation, but the question remains: how far we can push this simulation using only price data?

3.1 Architecture

Let us briefly describe the architecture of our network. Choosing a recurrent network makes sense since we want to predict a single value from a time series. Moreover, LSTM networks are known for their ability to exploit temporal connections and to solve the gradient vanishing problem. Figure 5 depicts the main components of our model.

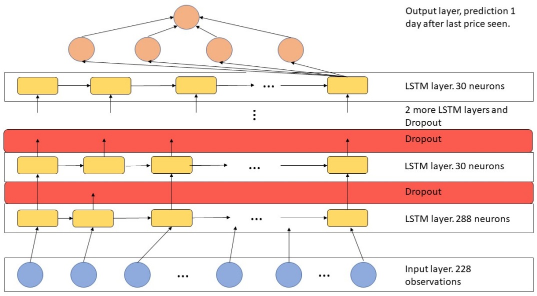

Figure 5: The architecture of our model is essentially a composition of multiple LSTM layers.

Our network consists of 5 LSTM recurrent layers, each consisting of 30 neurons, except from the first one, with 288 neurons. We drew these numbers from our preliminary testing, showing that such depth and width was good enough for our purposes. The input layer receives the 288 data points representing a complete day, encoding the sequential data. Then the output is passed across a sigmoid activation function and returned to feed the next layer. In these stages, the data is fed to the neural network and trained for prediction, initialized with random biases and weights. Every LSTM layer is followed by a Dropout regularization to avoid overfitting; the first layer has the biggest dropout probability and it decreases with each subsequent layer. After the last Dropout, two fully connected layers are added to obtain, first, new latent factors, and then to obtain the value that represents the prediction of the price on next day using a linear activation. When the output value is generated, is compared with the target value. The differences is then minimized using back propagation.

3.2 On scaling the data

With the introduced model, we proceed just as in Section 2, and start training our network for this new task of predicting the next day price. But the results, as Figure 6 shows were surprisingly worse: the model is practically useless.

The reason for this setback is that the prices of 1 day in the future have a much higher variation than what could be seen in consecutive data points. Together with our scaling function, this means that the network could not properly distinguish these changes. The issue is the following. If we scale all data points first, then most probably all data points of a 24 hours window would be mapped to very similar values. This would not be an issue for predicting values in the same range, but when forced to predict values for the next day, which are mapped to other ranges by the scaling, the network was no longer working how it should.

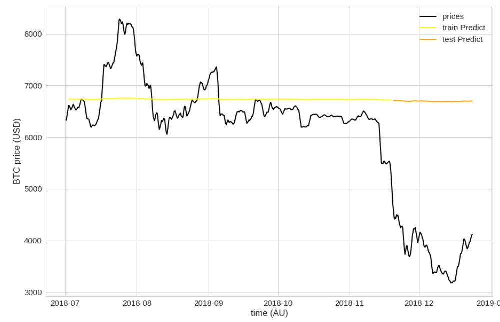

Figure 6: Next-day price prediction when all data is scaled as in Section 2. The curve is markedly displaced to an average value, not being able to recover the shape of the prediction curve and being flattened on all its extension

To solve this problem we use instead a scaling function that scales every 24 hours window separately, i.e., each observation of 288 points that feed our network is scaled with a common Min/Max scale function. Of course, there is a new problem with this solution: if the prices in a day vary in an (a, b) interval and the correct price in the next day is outside this interval, what should we expect the network to predict? To cope with this problem we alter the Min/Max bounds so that the lowest point (x_{min}) fed into the network does not get mapped to value 0 but to his equivalent value in scale, as a distance from the new lower bound in dataset $0.85 \times X_{min}$ and likewise for the higher point $1.20 \times X_{max}$. Figure 7 shows our results.

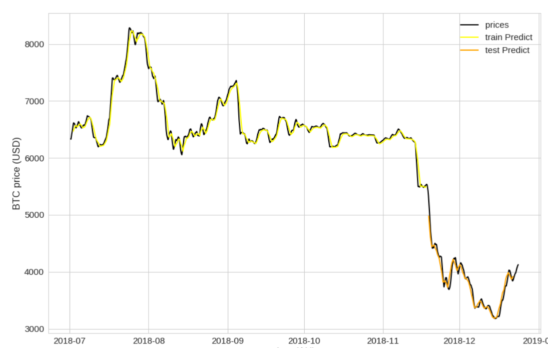

Figure 7: Next-day price prediction when all data is scaled in a per-day basis. Results are much stronger than what was obtained with a general scaling function.

There are of course many ways of improving these results: One can consider more complex networks, as in [4] where the authors develop generative adversarial networks to create a model of two part, one called Generator, who is an adversary that generate samples from time series and tries to compete with a Discriminator who determinate if the samples are real or generated by the adversary. Others, perhaps like [12] implement attention models to be aware of particular features on the sequence, and there are several other scaling functions to consider. However, since our goal is to later combine these models with twitter data, we choose to move onto the next problem.

3.3 Price Simulation using per-day scaling

Since we have seen that per-day scaling gives us a better model for the next-day problem, it also makes sense to update our price-simulation network so that the training is likewise done with this other scaling. Results are in Figure 8.

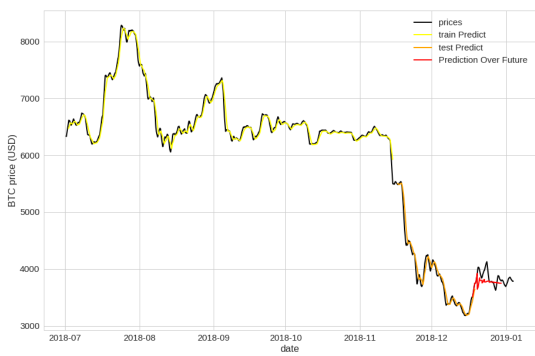

Figure 8: Price Simulation with per-day scaling. Test data is depicted on the red line

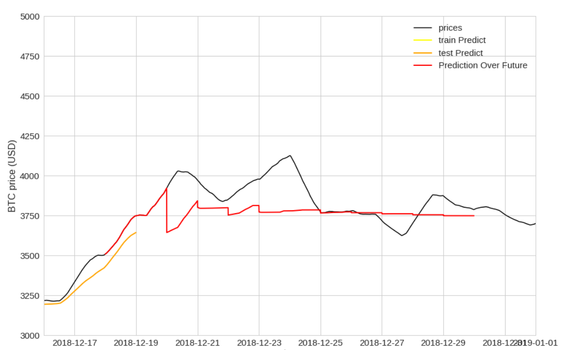

Figure 9: Zoom of the model's performance with test data. This model now manages to predict a decrease in price, but the prediction is still too coarse.

As we see, the new scaling gives our model the ability to identify slight trends over price data (shown in the red line in Figure 8). But we are still not able to detect the subsequent rise in price by the end of the test period: at this point the model has detected that prices tend to drop over time, and, unless we provide another form of external stimulus to the model, it will never change its direction. Our hypothesis is that twitter data may well be the external stimulus that we are looking for, and that "good" or "bad" tweets will give the network enough information to change the trend of what it is predicting. This is the focus of the next section.

4 INCORPORATING SENTIMENT ANALYSIS

The idea that the price of assets is affected by certain influential spokespersons is not new, and has already been studied when analyzing prices of other types of assets (see e.g. [1, 2], and our belief that twitter data is a good external source that should help in predicting Bitcoin data comes from gathering opinions of people working in the cryptomarket itself [23]. The goal of this section is to try out this hypothesis. To that extent, we first explain the way in which we collect data from these so-called influencers, then explain how to use their tweets to mine beliefs about the market, and then incorporate it into our predictor.

4.1 Twitter opinion data

There is a large body of work on how to compute the most central nodes in a social network, and to the best of our knowledge there is no agreement on a single method that fits all uses. Moreover, most of well known centrality measures (pagerank, betweeness centrality, degree centrality, etc.) require full access to the network to be computed, which is not feasible in the case of twitter. Instead, we use the following algorithm as an approximation for computing users with higher pagerank that are related to cryptocurrencies. This algorithm is inspired by [21]:

(1) We start with a seed of 10 users that are known to be influential.
(2) In each iteration, we create a list of users corresponding to all the people followed by the users in the seed.
(3) We order the list with the most followed users first, take the 10 most followed that are not already in the seed, and add them to the seed.

We selected a total of 135 influential users after 10 iterations of the algorithm. Then, we extract all tweets by these users between July and December 2018, but only store those tweets referring to one of the following terms: *bitcoin*, *btc* or *cryptocurrency* as well as their grammatical variations. This method gave us a total of 9146 tweets, which represent an average of 50.8 tweets per day across the six months considered. While this number is not small at all, is not comparable with the 288 measures per day that we use for studying price. For this reason, we propose to use a *moment*-like measure, i.e. we are going to sum all opinions over a movable window of 288 points, i. e.,

$$SM = \sum_{i}^{N} s_i \tag{3}$$

where s_i is the sentiment measure for an specific date i, while N is our movable window. We expect that tweets enclosed within this window will be a good information resource about the general appreciation of the cryptocurrency and how this is changing. Since we are interested on extracting a measure of the meaning of each tweet, a good tool to handle this is sentiment analysis.

Note that we could have obtained more tweets by simply enlarging our list of crypto influencers. However, there is a danger in doing this: with more users in our list, the tweets by the real influential people would by diluted by tweets from regular people, which we do not trust to affect the price in the same way. One direction we are currently following is to feed both the information of the tweets and the influence of the user into the neural network, but results are still too preliminary to report in this paper.

Sentiment Analysis. Sentiment analysis is another task that has received a tremendous amount of attention over the past decade (see e.g. [16–18]). Unfortunately, we cannot simply use known tools for sentiment analysis (or even sentiment of tweets) and expect that they would transfer well to the world of bitcoin. The main reason for this is that a big number of bitcoin-related tweets are very neutral in sentiment, but convey specific information that can be very good or bad for the price. Imagine, for instance, a tweet stating *Several cities in China forbid bitcoin mining*. This tweet is very neutral in sentiment, yet conveys information that should affect the price in a negative way.

For the scope of this paper, however, we will stick to traditional methods, and see whether they can still give us something about the price, and leave specific bitcoin sentiment analysis for future work. Specifically, we use *SentiWordNet*'s sentiword tool [20], an algorithm capable of exploring the semantic relations of words using a pre-trained dictionary with positive, negative and neutral scores over certain relations. Using this tool we get a sentiment moment distribution as presented Figure 10.

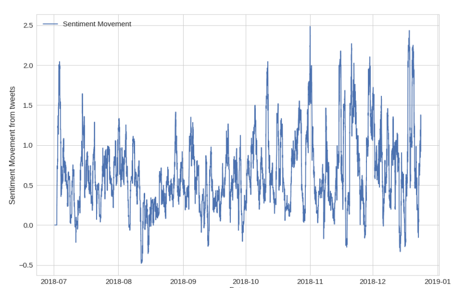

Figure 10: Evolution of sentiment movement, i.e. the sum of sentiment contained on a 1-day moving window.

It is remarkable how the sentiment score along the analyzed months is more positive than negative, with only just a few drops below zero. It is not evident if those scores are result of a generalized opinion about the topic or is just result of a high density of tweets over a particular date. However, we trust that a high density should be result of a determined event that produces a certain general opinion.

4.2 Results on prediction

With the introduction of a the sentiment measure we implement a final model, this one is similar to the described Figure 5 with the difference that the input layer is feed with observations of 288*2 dimensions, one additional dimension per point data for the moment sentiment of tweets in that point in time. With this model we train again our neural network. Our predictions are presented Figure 11 with a closer view over Figure 12.

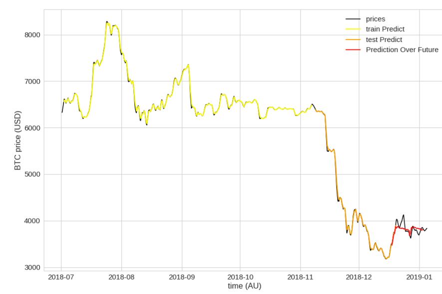

Figure 11: Prediction of the model when trained also with twitter data. The predictions for the next-day price are on the same level as the model trained without twitter data.

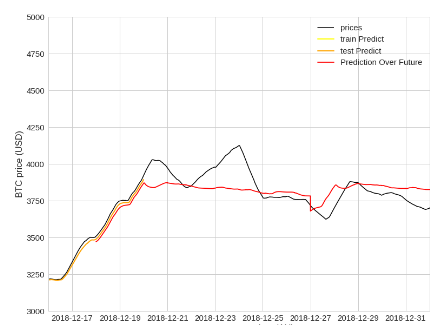

Figure 12: Zoom to final predictions. We note a fall over the seventh day that is recovered by the prediction.

Regarding the next-day problem, we noticed essentially the same predictive power, when compared via any of the two measures we were looking. However, the most interesting comparison comes from the price simulation problem, when looking at both Figure 12 and comparing it with Figure 9. We understand that the simulation of price predicted with the model of Figure 12 is much cleaner than the other one, and it is very interesting to note how twitter data predicts the movement of the curve around the 27th of December 2018. Looking at 9, we can only suspect that it was twitter data that helped shape the curve in this way.

For reference, we have included the sentiment evolution between our time window in Figure 13. We notice some trends by the end of December, but clearly there is not a huge visual correlation. It is surprising that this data still allows us to predict spike changes within the price simulation framework.

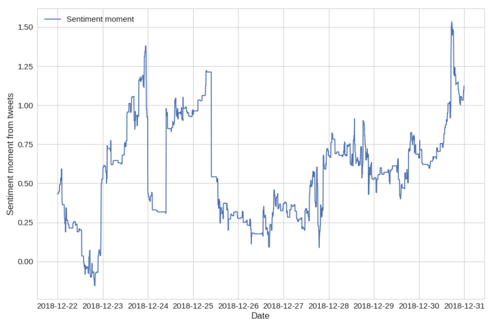

Figure 13: Evolution of sentiment movement, during prediction dates.

5 CONCLUSION AND FURTHER WORK

In studying the problem of predicting the price of bitcoin in the future, we have made advancements towards showing that the activity of influential personalities in the digital world does affect the price of this cryptocurrency.

The problem which is most interesting to us in this respect was the idea of simulating the price using a predictor model that is trained with real price data, but feeding with its own prediction and only uses twitter as its external stimulus. However, we have also shown a possible architecture that goes a long way into predicting the price when fed only with past price data: the error for predicting the next price point was almost negligible, and the predictions for the next-day price were accurate as well.

When fed with both price and twitter data, the resulting model had essentially the same predictive score than the model fitted only with price data. However, as we mentioned in the last section, the differences shown when simulating price leads us to believe that there is promising work to be done in refining our ideas.

Further work. As mentioned, evidence suggests that there is much room for improvement. In particular, we are currently pursuing the following main directions.

The first and most imperative refinement is to leverage state-of-the-art tools to produce a better classification of news into positive/negative. The main difficulty here is that the fact that a news text or an informative tweet ends up being positive or negative may not have anything in common with the actual sentiment of the tweet, as we expect most news to deliver information in a neutral or slightly positive tone. A new network classifying this precise task could be trained from scratch, but the problem here is that it is not trivial to amass the necessary data to produce a well-functioning classifier. Hence, the most promising approach appears to take advantage of the best models for standard sentiment analysis and then tune them to our specific task via transfer learning, as in e.g. [17].

The other important refinement is to study different neural architectures and different data input options for our predictors. As for data input options, we currently don't know a better way of integrating twitter data to the price stream, but we are not certain that these better ways not exists either, and perhaps there is a better way of aggregating this data. There are also several other ways of obtaining both influential users and/or their tweets: right now we are thinking on feeding the network with a greater density of tweets, but also with the information about the approximate centrality of the users posting these tweets so that the non-influential tweets do not end up diluting the valuable information. We are also interested on adding other types of data into the mix, such as news and reddit posts (again posted by influential users).

As for network architectures, one of our last ideas is to recover the results of Zhou et al. (2018) [4] about generative adversarial nets. The intuition here is that an adversarial net would use price history to train next data points, but an adversary would then use external twitter data to discriminate against the likelihood of the predicted price. This would perhaps not only improve our predictions, but would also give us a much more cleaner justification on the relation between price and tweets. Other ideas have to do with adding further penalization to the loss function so that models correctly predict inflection points.

6 ACKNOWLEDGEMENTS

This work have been partially funded by Millennium Institute for Foundational Research on Data (IMFD). We have also been partially supported by NIC Chile and by Pontificia Universidad CatÃşlica de Chile.

REFERENCES

[1] Xi Zhanga Yunjia Zhang, Senzhang Wang, Yuntao Yao, Binxing Fanga, Philip S. Yu. (2018). Improving Stock Market Prediction via Heterogeneous Information Fusion.
[2] Xi Zhang, Yixuan Li, Senzhang Wang, Binxing Fang, Philip S. Yu. (2018). Enhancing Stock Market Prediction with Extended Coupled Hidden Markov Model over Multi-Sourced Data.
[3] Ting Li, Bingyi Jing, Ningchen Ying , Xianshi Yu. Adaptive Scaling (2017).
[4] Xingyu Zhou, Zhisong Pan, Guyu Hu, Siqi Tang, and Cheng Zhao. (2018). Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets.
[5] Kasun Bandara, Peibei Shi, Christoph Bergmeir, Hansika Hewamalage, Quoc Tran, Brian Seaman. (2019). Sales Demand Forecast in E-commerce using a Long Short-Term Memory Neural Network Methodology.
[6] Zahra Karevan, Johan A. K. Suykens. (2018). Spatio-temporal Stacked LSTM for Temperature Prediction in Weather Forecasting.
[7] Uma M. Girkar, Ryo Uchimido, Li-wei H. Lehman, Peter Szolovits, Leo Celi, Wei-Hung Weng. (2018). Predicting Blood Pressure Response to Fluid Bolus Therapy Using Attention-Based Neural Networks for Clinical Interpretability.
[8] Dat Tien Nguyen, Shikhar Sharma, Hannes Schulz, Layla El Asri. (2018). From FiLM to Video: Multi-turn Question Answering with Multi-modal Context.
[9] Hyunwoo Jung, Moonsu Han, Minki Kang, Sungju Hwang. (2018). Learning What to Remember: Long-term Episodic Memory Networks for Learning from Streaming Data.
[10] Vidya Prasad K, Akarsh S, Vinayakumar R, Soman KP (2019). A Deep Learning Approach for Similar Languages, Varieties and Dialects.
[11] Bilal Wehbe, Octavio Arriaga, Mario Michael Krell, Frank Kirchner. (2018). Learning of Multi-Context Models for Autonomous Underwater Vehicles.
[12] Guoqiang Zhong, Xin Lin, Kang Chen. (2018). Long Short-Term Attention.
[13] Dzmitry Bahdanau, Dmitriy Serdyuk, Philémon Brakel, Nan Rosemary Ke, Jan Chorowski, Aaron Courville, Yoshua Bengio. (2016). Task Loss Estimation for Sequence Prediction.
[14] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, Guy Van den Broeck. (2018). A Semantic Loss Function for Deep Learning with Symbolic Knowledge.
[15] Sean Welleck, Zixin Yao, Yu Gai, Jialin Mao, Zheng Zhang, Kyunghyun Cho. (2017). Loss Functions for Multiset Prediction.

[16] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. (2018). Deep contextualized word representations.
[17] Jeremy Howard, Sebastian Ruder. (2018). Universal Language Model Fine-tuning for Text Classification.
[18] Shaunak Joshi, Deepali Deshpande. (2018). Twitter Sentiment Analysis System.
[19] Sara Rosenthal, Noura Farra, Preslav Nakov. (2017). SemEval-2017 Task 4: Sentiment Analysis in Twitter.
[20] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. (2006). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.
[21] Daniel Pérez. (2018). Crypto-Influencers Algorithm. Url = "https://github.com/dperezrada/crypto-influencers", lastaccessed on June 23, 2018.
[22] Nakamoto, Satoshi. (2018). Bitcoin: A peer-to-peer electronic cash system.
[23] Daniel Pérez Rada (2018-01-22). "Personal communication".
[24] Joost van der Burgt. (2018). Making Sense of Bitcoin Price Levels.