Analysing social media impact on forecasting the bitcoin price

Student name *** Supervisor

Mamouri sofia Dr Sergio Ioppolo

Student Number **190770297**

MSC Big Data Science

Abstract

Cryptocurrencies have gained popularity since 2017 due to consecutive exponential rises in their prices. Their value is based on the willingness of people to pay for them, this makes this field riskier than any other financial asset but also more profitable if smart strategies are taken. With the evolution of graphics processing unit technology (GPU), the computing power has increased significantly which helped advance neural networks algorithms. Such algorithms became successful in many fields. This success helped to forecast many problems. The cryptocurrency price is time-series data and we can identify its trends using deep learning methods. Recurrent neural network models (RNN) are well known to be suitable for time series problems. Here, we choose to use the evolved variant long short term neural network (LSTM) to predict Bitcoin price. Other external factors affect the cryptocurrency price like daily news, financial news and social media. In this paper, we consider social media as a solution to analyse social media impact on forecasting one of the largest cryptocurrencies "Bitcoin" by applying LSTM neural network on the closing price historical data and text sentiments of the related tweets to bitcoin.

Keywords: RNN, LSTM, Bitcoin, SA

I. Introduction

In today's globalised world, more than 6600 cryptocurrencies are created to address the drawbacks of the traditional currency and the lack of confidence in the financial system that is due mainly to the international financial crisis in 2008. The problem with traditional currency is centralisation. The money is issued, regulated and controlled by a central authority such as the government or a central bank, and it is unlimited by quantity as it can be printed by the government whenever needed. Thus, whoever has the control over the money supply has unlimited power as Sir John Dalberg-Acton said "Absolute power corrupts absolutely", for instance, the Wells Fargo fraud scandal the bank created millions of fake accounts without the consent of their clients. The issue with centralisation is that the central authority might print money to save certain institutions like banks from collapsing which increases the money supply and causes inflation. Also, people can lose control of their money, since the central authority has the power to freeze their bank accounts and deny them access to the funds. An authoritarian government can use this power unlawfully. On the other hand, cryptocurrency offers security and more freedom even though some people believe that it encourages money laundering and makes life easier for criminals.

In 2009, an alternative decentralised digital monetary system arose named cryptocurrency. What is a Cryptocurrency ? and why do we need to use It? Cryptocurrency is a decentralised digital currency that uses cryptography which is an encryption technique for protecting information and secure communications. We can use cryptocurrency the same way as any traditional currency. We can buy, sell and exchange goods directly without any third party to ease the instant payment. Cryptocurrency has no intrinsic value, thus cannot be touched by hand nor issued or regulated by any central authority. Bitcoin is the first cryptocurrency created and the most traded in the cryptocurrency market.

What is Bitcoin?

An anonymous group of people with the pseudonym Satoshi Nakamoto [1] introduced the whitepaper “Bitcoin: A Peer-to-Peer Electronic Cash System”. They described the need for an electronic payment system depending on cryptographic algorithms instead of trust and without middleman intervention to facilitate the instant payment. In 2009, the open-source software that enables the use of this currency was released. Bitcoin is created as a challenge prize where users use their computing power to record and verify transactions into a public distributed transactions ledger system, known as the blockchain. This task is known as mining and only successful miners are awarded newly fixed mined bitcoins but periodically decreasing in rate. The cryptocurrency market is one of the largest unregulated markets and its value exceeds $2 trillion. Bitcoin accounts for over half of this figure, as data from the CoinMarketCap site show https://coinmarketcap.com/. This rise is driven by the growth of institutional and individual demand.

To analyse the impact of social media on forecasting the Bitcoin price, we will apply the LSTM neural network to the Bitcoin historical data to predict the closing price at first, then we will introduce the sentiment of the tweets as a new feature in the Bitcoin dataset, so this time the model will be trained using previous input along with the new sentiment feature to predict the daily closing price. The sentiment will be classified as neutral, positive or negative. Finally, we will compare the obtained predictions from both LSTMs, i.e., before and after adding the sentiment of the tweets, by using Root Mean Square Error RMSE as an evaluation metric for their performance to see if there is any correlation between the social media and the price of bitcoin.

A.Literature review

In this section, we will describe research that is related to LSTM neural networks, forecasting and sentiment analysis.

LSTM neural networks

The LSTM neural network is a variant of RNN. RNN possesses loops in them that allow information to persist and pass from one step of the network to the next, i.e. the output from the previous step is passed to the next step as input, they have a memory that remembers the previous input. This chain-like nature makes it very suitable for time series problems and they were applied into many fields such as language modelling, speech recognition. However, RNN cannot process very long sequences.

LSTM was proposed by Hochreiter et.al [2] and later developed by other researchers. It was designed to improve the long-term memory issue of RNN and combat vanishing gradient problems. The LSTM neural network has a structure called memory cell that contains neurons and three main gates called input , forget and output gates. These gates decide which information is relevant to be kept and which one to discard. This mechanism retains only relevant information and thus provides a good solution for our predictive model.

Cryptocurrency price forecast

The continuous expansion of the cryptocurrency market in the last years attracted institutional and individual demand as well as the attention of media coverage. This success resulted in the creation of many other cryptocurrencies such as litecoin, ether.

Those cryptocurrencies only differ from Bitcoin by a few factors such as currency supply and block time. Any trading service whether it uses traditional currency or a digital one, won't be free of risk. Cryptocurrencies are traded freely without any regularisations or interventions from any central authority and are categorised as highly volatile and non-linear. They are affected by many factors, such as investors behaviour, politics, institutional demand and the global economy. This is why we see the extreme fluctuation of market cap value. Many financial firms and hedge funds like JP Morgan, Goldman SACHS are including cryptocurrencies in their portfolios. On the other hand, the research community is still conducting research in cryptocurrency trading based on deep learning algorithms.

Längkvist et.al [3] presented a review on the late "developments in the deep learning field along with unsupervised feature learning for time-series problems."

Kim et.al [4] studied the comments and replies of users in cryptocurrency forums to predict the variation of prices as well as the transactions of bitcoin, ripple and ethereum.

According to Li et.al [5], the analysis of bitcoin prices was driven by risky investment and as the market grows, the price adjusted to changes in the economy.

Jang et.al [6] analysed the time series of Bitcoin using Bayesian neural networks (BNNs) and revealed that not only BNN performed well in predicting Bitcoin price but could even explain the high volatility of the late Bitcoin price.

In Phillips et.al [7], to build profitable trading strategies on multiple cryptocurrencies, hidden Markov models were applied depending on online social media indicators behavior

Ji et.al [8] analysed multiple state-of-the-art deep learning techniques like convolutional neural network (CNN), deep neural network (DNN), LSTM model as well as the combinations of those models for Bitcoin price prediction. They revealed that the performances of the proposed models were comparable.

Lahmiri et.al [9] showed that the predictability of LSTM is superior to the generalised regression neural networks.

In Sebastião et.al [10], based on machine learning techniques, they propose an ''examination of the predictability and profitability of bitcoin, ethereum, and litecoin.''

B. Sentiment analysis

Nowadays, people express their opinions, comments and even exchange information using different online platforms such as social media, blogs and online forums. We can benefit from analysing the sentiment expressed by the public users which is related to cryptocurrency to understand the online public opinions. Sentiment analysis (SA) studies the subjectivity of the information in an expression. It can be an emotion, opinion or attitude. Expressions can be categorised as positive, neutral or negative. SA belongs to the natural language processing (NLP) field and many studies like Li et.al [11], Kumar et.al [12] analyse and compare several methods of context-based sentiment analysis to extract emotions from online text data. Nofsinger et.al [13] realise that investors behaviour is an important element in the financial market. Investors conduct high trading after good news is released which leads to a rise in stock prices and vice versa.

II. Methodology

The fundamental idea behind LSTM is that it maintains its state over time through the gating mechanism existing inside the LSTM cell[14]. Each cell takes the current input data, the short-term memory is known as the hidden state and the long-term memory referred to as the cell state.

What is the inner work of those gates and how are they trained to return accurate results?

The input gate determines the new values to store in the cell state using two layers as shown in fig.1. The 1*st* layer decides the value to be retained by passing the current input and the hidden state from the previous stage multiplied by the weights as input to the sigmoid function to convert the values to be between 0 or 1. zero indicates that the value is irrelevant and discards it whereas one means the value is important and thus kept. This 1st layer is trained through backpropagation to update the weights in order to learn to only ignore the insignificant value as in (1).

The *2nd* layer uses the same input as the *1*st layer but with different weights and then passes the input to the *tanh* activation function to regulate the network as in (2).

The multiplication of the obtained outputs as in (3) represents the information retained in the long-term memory.

The forget gate shown in fig.2 decides what information from the cell state should be retained or rejected. Similar to the 1*st* layer of the input gate, we feed the same input again to the sigmoid function but with different weights and we get a vector of 0s and 1s that we multiply with the hidden state to decide what value to retain or to ignore from the cell state as in (4). Then, a pointwise addition occurs between the output of the input gate and the forget gate to produce the new cell state as in (5).

The output gate illustrated in fig.3 produces the new hidden state by taking the current input, the previous hidden states as input and passing it again through a sigmoid function but with different weights as in (6).

Next, we feed in the newly computed cell state to a tanh activation function as in (7)

Then we multiply both outputs to produce the new hidden state as in (8).

The newly produced hidden state and cell state are passed to the next LSTM cell and the process is repeated to the end.

$$i_1 = \sigma(W_{i1} * (H_{t-1}, x_t) + bias_{i1}). \quad (1)$$

$$i_2 = tanh(W_{i2} * (H_{t-1}, x_t) + bias_{i2}). \quad (2)$$

$$i_{input} = i_1 * i_2 . \quad (3)$$

$$fv = \sigma(W_{forget} * (H_{t-1}, x_t) + bias_{forget}). \quad (4)$$

$$C \quad = C_{t-1} * fv + i_{input}. \quad (5)$$

$$O_1 = \sigma(W_{out1} * (H_t, x_t) + bias_{out\,1}). \quad (6)$$

$$O_2 = tanh(W_{out2} * C_t + bias_{out2}). \quad (7)$$

$$H_t, O_t = O_1 * O_2 . \quad (8)$$

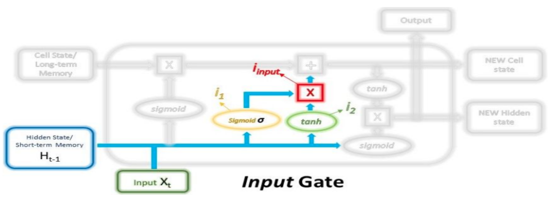


Fig.1 indicates the undergoing operations inside the input gate.

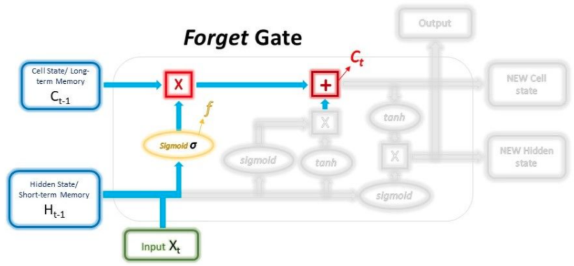


Fig.2 indicates the undergoing operations inside the forget gate.

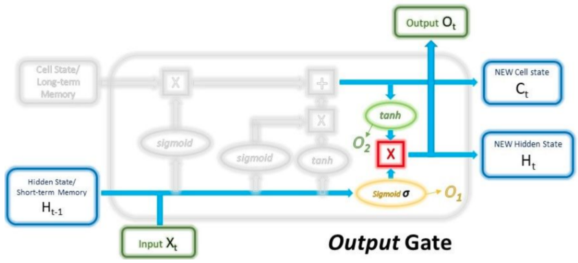


Fig.3 indicates the undergoing operations inside the output gate.

A. Data collection

We collected bitcoin historical data from Bitmex.

1. We created an account on Bitmex and used the API key on the Bitmex crypto exchange.
2. We generated two keys: API key and Secret key for authentication purposes.
3. We created a function in python to fetch the selected data.

There are twelve features in the basic transaction dataset. Open is the opening price and it is the first transaction price per share after the opening of the market on a trading day. Close is the closing price and it is the final price that day. High is the highest price a cryptocurrency trades in a day. Low is the lowest price that day. Trades refers to the number of transactions in a time unit for a transaction. Volume is the number of shares traded. Symbol XBTUSD is simply XBT which is the new abbreviation for bitcoin and USD is the dollar unit. VWAP is Volume Weighted Average Price and it indicates the daily average price of bitcoin weighted by the Volume. Trades is how many bitcoins were traded during the day. Turnover is the daily bitcoin gain. Lastsize is how many bitcoins were traded in the last transaction. Home Notional refers to how many coins this trade is worth. Foreign Notional refers to how many US dollars was this trade worth. Table1 lists some data samples of the Bitcoin price dataset and we are only interested in forecasting the closing price.

timestamp	symbol	open	...	homeNotional	foreignNotional
2015-09-26 00:00:00+00:00	XBTUSD	239.99	...	448.441887	1.057927e+05
2015-09-27 00:00:00+00:00	XBTUSD	235.20	...	604.837298	1.415267e+05
2015-09-28 00:00:00+00:00	XBTUSD	234.51	...	376.046459	8.782600e+04
2015-09-29 00:00:00+00:00	XBTUSD	233.29	...	759.499444	1.807558e+05
2015-09-30 00:00:00+00:00	XBTUSD	240.07	...	878.655118	2.093607e+05
...	...	...	...	...	...
2021-08-13 00:00:00+00:00	XBTUSD	45518.50	...	30032.259918	1.344272e+09
2021-08-14 00:00:00+00:00	XBTUSD	44387.50	...	31750.850896	1.469991e+09
2021-08-15 00:00:00+00:00	XBTUSD	47821.00	...	23622.575650	1.108554e+09
2021-08-16 00:00:00+00:00	XBTUSD	47097.50	...	28043.405975	1.298315e+09
2021-08-17 00:00:00+00:00	XBTUSD	47030.00	...	36246.755941	1.696843e+09

[2153 rows x 12 columns]

From the curve in fig.4 we can notice the existence of random variations and not consistent trends.

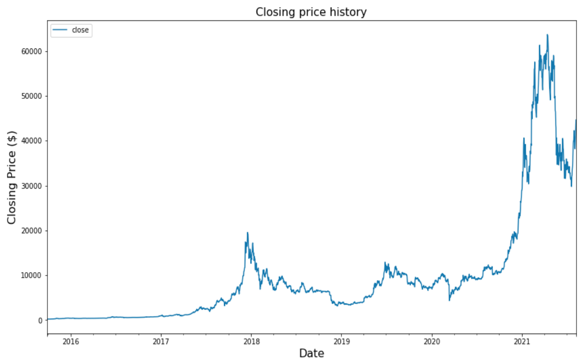


Fig.4 represents the daily bitcoin closing price.

B. Data preprocessing

The Bitcoin dataset collected is already cleaned, there are no missing values and the dataset is indexed using the timestamp column before being divided into training, validation and test datasets as shown in fig.5 and then scaled using Standardscaler from sklearn library to help the model converge faster. Also, we use a sequence length of 30 days.

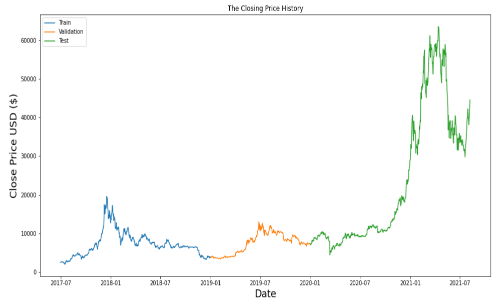


Fig. 5 represents the Bitcoin daily closing price history.

The aim of the LSTM neural network is to learn to map an input which is a sequence of past information to output, so we transform the long sequence of past information into multiple samples from which it learns, i.e. shorter sequences. Then, we divide the sequence into multiple input and output shapes where *n* steps, referred to as sequence length, are used as input and the *n+1* step is used as output, i.e. we shift by one. The obtained samples from this transformation are converted to tensors since we are using Pytorch as illustrated in fig.6.

```python
from torch.autograd import Variable
def transform_data(ar, seq_len):
    x,y = [],[]
    for i in range(seq_len,len(ar)):
        x_i = ar[i-seq_len:i]
        y_i = ar[i]
        x.append(x_i)
        y.append(y_i)
    x_ar = np.array(x)
    y_ar = np.array(y)
    x_var = Variable(torch.from_numpy(x_ar).float())
    y_var = Variable(torch.from_numpy(y_ar).float())
    return x_var,y_var
```

Fig.6 represents the code of the transformation function.

C. Building the LSTM model

The LSTM model was implemented in Python using Pytorch. The LSTM was instantiated using the LSTM layer provided with the following argument:

- Input dimension: represents the size of the input features.
- Hidden neurons: represents the size of the cell state and the hidden state at each timestep.

- Number of layers: represents the number of stacked layers in the network.
- Output dimension: represents the size of the output features.

The model has two layers, 128 hidden neurons, a dropout function of 20% and a fully connected linear layer for the final output as shown in fig.7. The dropout function reduces overfitting and improves the performance.

```
LSTM(
  (lstm): LSTM(1, 128, num_layers=2, batch_first=True, dropout=0.2)
  (linear): Linear(in_features=128, out_features=1, bias=True)
)
```

Fig.7 represents the LSTM structure.

The profit is usually determined by it at the end of the day, therefore, we are only interested in forecasting the closing price. The closing price will be considered as input and target variable since the LSTM model will make the predictions of the closing price for the upcoming days and thus the output and the input dimension is 1.

To start, we feed in the input which is a tensor of the shape (batch-size, sequence length and input dimension) along with the hidden states. Then, the model will process the data through its gates and produce an output at each timestep.

We train the LSTM model on the training set along with the hyperparameters tuning, then we test on the validation set and evaluate the final results on the testing set. There are no rules on how to determine some hyperparameters such as the number of the hidden neurons, learning rate or the number of stacked layers. Their tuning depends on many characteristics like the complexity of data, the generating data process, domain knowledge. Hence, we tuned them by trial and error and we selected the best configuration based on its performance on the validation set. The model is trained using 150 epochs and a learning rate of 0.01.

- Epoch: indicates the number of times the training set is passed to the model. In our model the batch size is the number of the whole training set thus, epoch is the number of iterations.
- Learning rate: controls the step size taken towards the convergence of the model.

Mean square error (MSE) is defined as a loss function to calculate the model error to see how far the predicted value differs from the ground truth. Adam optimiser is used as the optimiser function to optimise the error. Once the model is trained it is used to make predictions on the validation set and the model with best performance is then evaluated on unseen data,i.e testing set.

D. Tweets collection

To analyse the impact of social media on the Bitcoin price forecasting, we collected the tweets related to Bitcoin from Twitter API, i.e. only tweets mentioning the words Bitcoin and btc to apply sentiment analysis on those tweets. We applied polarity detection to extract the sentiment. To achieve this, we created a Twitter application and generated the access token key and access token secret key for authentication purposes and used Tweepy library to access Twitter API easily. Next, the tweets are cleaned by removing any irrelevant characters before calculating its sentiments, i.e polarity of the tweets using Textblob library. The resulting polarity score falls into three categories: positive, neutral and negative. Those scores are added as a new column on the Bitcoin dataset used before, so this time the model will be trained using two input features to forecast the bitcoin price. However, Twitter's API has limitations, it limits the extraction of the tweets to the last 3200 tweets in his timeline and does not return tweets older than 7 days unless you pay for it. To address this issue we collected the tweets related to Bitcoin from kaggle, however the data was insufficient for our model to learn from it. To deal with the lack of data, we generated synthetic data containing a random score of the tweet’s polarity: 0 indicates neutral, 1 shows positive, whereas -1 as negative as seen in fig.8

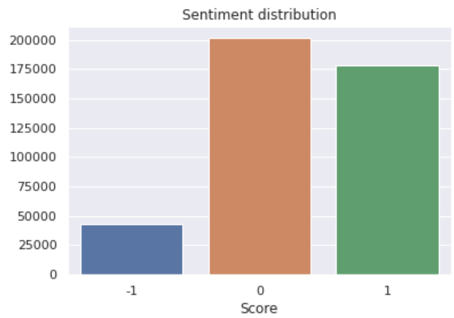


Fig.8 represents the score of sentiment distribution.

After training, testing and evaluating the LSTM model with the new sentiment feature, we compared its performance with the previous LSTM model to see if there is any correlation between the social media and the price of bitcoin.

E. Model performance metrics

The prediction results were evaluated by the root mean square error (RMSE). RMSE is one of the widely used measures to judge the quality of the predictions. It is used frequently on standardised data. The smaller the RMSE, the better is the performance and the closer the predicted value is to the ground-truth value.

RMSE can be expressed as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2}$$

N: is the number of observations.
y_i: is the ground-truth.
$\widehat{y}_i$: is the corresponding prediction.

To compute RMSE, we calculate the residual, i.e. the difference between the prediction and the groundtruth for each observation. Next, we compute the square of the residual for observations, then the mean of those squared residuals and we take the square root of that mean.

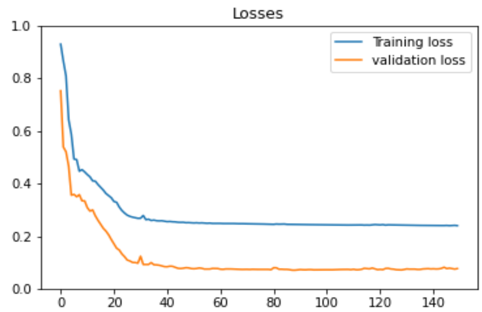
III. Experimental results and discussion

The dataset is shortened to only include observations occurring in 2017 where the price bubble occurred. At the beginning of the year, the price was around $1,000 and then had a short decline to $975 before reaching a peak of $20,000 on December 17, as shown in the fig.4 above.

Fig.9 represents the learning curves of the loss function and it shows that the trained model is neither underfitting nor overfitting since the training and validation loss both decrease to stability. However, the gap between the two curves is not minimal. Additionally, the validation loss is lower than the training loss.

Fig.9 plots the training and validation loss

This lowerness is due to applying the regularisation



dropout function only to the training set and to computing the validation loss once the current training epoch is completed. However, the generalisation gap could be a result of an unrepresentative validation dataset that fails to capture specific features that are efficient for perfect generalisation.

In fig.10, we can see that the LSTM predictions defined by the orange line are able to follow the pattern, i.e. it captures the unexpected jumps and downs along with the fluctuations. However, LSTM is unable to apprehend the excessive-high jumps to return good results.

Table.2 indicates the RMSE scores returned.

Training set	Validation set	Testing set
0.47 RMSE	0.31 RMSE	0.36 RMSE

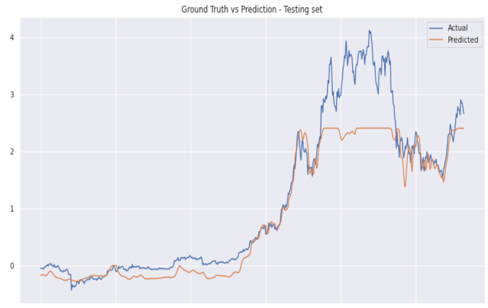


Fig.10 plots the ground truth values and the predicted values without including the tweets feature.

We believe this situation is due to the large difference between observations caused by real variations and external factors in the global economy which sometimes are unexplainable such as the increased demand on the cryptocurrency during the burst of the covid-19 pandemic. Also, the LSTM model might have some limitations that prevent him from predicting beyond a certain value or perhaps the model needs more features to learn and perform better.

The same applies to the fig.11. Adding the tweet's polarity as a new feature to the dataset did not impact the model. We can see that the model followed the pattern but clearly could not capture the excessive-high jumps. We can say that the model took the tweets into account but did not show any correlation between the tweets and the price prediction.

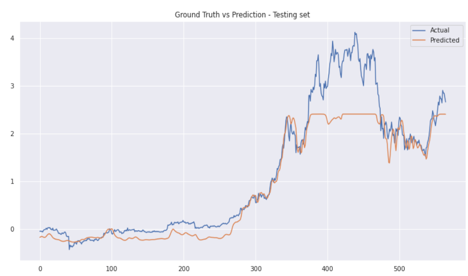


Fig.11 plots the actual values and the predicted values using a testing set and including the sentiment feature.

IV. Conclusion and Future Work

Even by adding the sentiment of the tweets referred to as polarity to our dataset we did not get any clear improvement, this can be considered as acceptable since in reality, a tweet can be positive or negative but has no impact or it can be neutral but retweeted frequently and this might increases the awareness and attract the attention of readers, followers and media to the cryptocurrency field. In our opinion, the sentiment tweet feature alone in the context of our LSTM is not enough to determine the impact on the prediction of the bitcoin closing price. In general, tweets tweeted by powerful or famous personalities influence the behavior of their followers as when Elon musk announced that Tesla would not anymore accept the payment with Bitcoin and this announcement caused Bitcoin a loss of 15% of its value in the first 24 hours. Accordingly, adding a certain weight to tweets related to users having a high number of followers might have an impact and it is interesting to try it. Also, we can include other features such as tweet volumes and see how well the model performs.

Moreover, the LSTM model could not capture the excessive jumps due probably to some architecture limitations such as their sensitivity to different random weight initialisations since the weights and bias are initialised by using small weights and it is well known that the performance of a neural network model is affected by learnable parameters initialisations. Therefore, it is worth using other existing initialisation methods. Furthermore, data that is highly non-linear and this might prevent LSTM from representing this complexity, trying another model like a Deep LSTM (DLSTM) [15] which is an extension of the original LSTM may be more beneficial taking in account its ability to learn the non-linearity and the complexity of time-series data through its structure that includes multiple LSTM layers and where each layer contains multiple LSTM cells. This model might be the best solution.

Bibliography

[1]Nakamoto, SATOSHI. “A Peer-to-Peer Electronic Cash System.” *A Peer-to-Peer Electronic Cash System*, https://bitcoin.org/bitcoin.pdf.

[3]Längkvist, M., Karlsson, L. & Loutfi, A. Review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognit. Lett.* 42, 11–24(2014).https://doi.org/10.1016/j.patrec.2014.01.008.

[4]Young Bin Kim, Jun Gi Kim, Wook Kim, Jae Ho Im, Tae Hyeong Kim, Shin Jin Kang, Chang Hun Kim.Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies. https://doi.org/10.1371/journal.pone.0161197.

[5]Li, Jin & Quan -2020- Li W, Jin B, Quan Y. Review of research on text sentiment analysis based on deep learning.

[6]Huisu Jang; Jaewook Lee An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information.https://ieeexplore.ieee.org/document/8125674.

[7]Phillips R C, Gorse D -2017- Predicting cryptocurrency price bubbles using social media data and epidemic modelling. IEEE symposium series on computational intelligence (SSCI). https://doi.org/10.1109/SSCI.2017.8280809.

[8]Ji, Suhwan, Jongmin Kim, and Hyunseung Im. 2019. "A Comparative

[2]Hochreiter sepp. *Long short term memory*, https://ieeexplore.ieee.org/abstract/document/6795963.

Study of Bitcoin Price Prediction Using Deep Learning" Mathematics 7, no. 10: 898. https://doi.org/10.3390/math7100898.

[9]Lahmiri S, Bekiros S -2019, Cryptocurrency forecasting with deep learning chaotic neural networks. https://doi.org/10.1016/j.chaos.2018.11.014.

[10]Sebastião, H., Godinho, P. Forecasting and trading cryptocurrencies with machine learning under changing market conditions. https://doi.org/10.1186/s40854-020-00217-x.

[11]Jie Li,Lirong Qiu.A Sentiment Analysis Method of Short Texts in Microblog.https://ieeexplore.ieee.org/document/8005904.

[12]Akshi Kumar Teeja Mary Sebastian., Sentiment Analysis on Twitter.https://www.researchgate.net/publication/304579281_Sentiment_Analysis_on_Twitter

[13]John R.Nofsinger, Johan Sulaeman, Abhishek Varma. Institutional investors and corporate social responsibility https://doi.org/10.1016/j.jcorpfin.2019.07.012.

[14]https://blog.floydhub.com/long-short-term-memory-from-zero-to-hero-with-pytorch/

15-Sagheer, A. & Kotb, M. Time series forecasting of petroleum production using deep lstm recurrent networks. Neurocomputing 323, 203–213 (2019).

MSc Project - Reflective Essay

Project Title:	Analysing social media impact on forecasting the bitcoin price
Student Name:	Mamouri Sofia
Student Number:	190770297
Supervisor Name:	Loppolo Sergio
Programme of Study:	MSC Big Data Science

My research topic is to analyse social media impact on forecasting one of the largest cryptocurrencies “Bitcoin” by applying deep learning techniques. We chose to apply the evolved variant Long short term (LSTM) neural network to predict the price of Bitcoin since the model deals with vanishing gradients and long-term dependencies. Cryptocurrency is affected by some external factors such as daily news, financial news and social media. In this study, we considered the sentiment of the tweets as a solution to analyse social media impact on forecasting the Bitcoin closing price historical data.

1. The difference between the theory and practical work

Forecasting has always been important in making decisions. Both individuals and organisations seek to maximise profit and reduce the risks. The future is surrounded by uncertainty which is exciting and challenging simultaneously. Time series forecasting is a popular field but very difficult to handle due to its temporal aspect since it involves a time component that adds more information. Cryptocurrency price is a time series data that is volatile due to its nature as a speculative asset. Also, it is noisy, non-linear, non-stationary and prone to global economic changes. Those complex features make forecasting the cryptocurrency prices data a challenge.

Many approaches and methods address this field and with the evolution of graphics processing unit technology that is designed for parallel processing and the growing availability of data, deep learning techniques show a lot of promise. Those techniques analyse the input data, learn features and draw conclusions. The recurrent neural network (RNN) is a promising model in handling sequential data models and is known to be suitable for time series problems since they have a short-term memory obtained from the feedback loop. The RNN was tested in many problems such as speech recognition, translation, language modelling.

Their success lies in their ability to address the vanishing gradient issue. The evolved variant Long short term neural network (LSTM) [1,2] has a structure labelled memory cell that contains neurons referred to as LSTM cell and three main gates: input, forget and output

gates. These gates decide which information to retain and which one to discard,i.e. the information flow is regulated through these gates.
Thus, only relevant information remains and this provides a good solution for our predictive model in this study. All exciting results based on RNNs were successfully realised by LSTM models and for this reason they became a state-of-the-art model within the deep learning field.

The LSTM model is being used by Google for speech recognition [3], and also for machine translation improvement on Google Translate [4]. Amazon is using it to improve Alexa's functionalities [5] and Facebook to improve the translation to ameliorate the consumer's experience [6]. It was demonstrated that The LSTM models are very successful in many application fields, however, they are less commonly applied to financial time series predictions. Few research papers like Fisher and Krauss [7], applied the model to financial market predictions and found that LSTM networks outperform memory-free classification methods such as DNN, a random forest (RAF), and a logistic regression classifier. Yan and Ouyang [8] empirical results show that LSTM outperforms multi-layer perceptrons, support vector machines and K-nearest neighbours. Chung and Shin [9] proposed a hybrid approach integrating LSTM and genetic algorithm (GA) on KOSPI index data and similarly showed that the method achieved satisfactory performance. Cryptocurrency prices are highly volatile and non-linear, maybe this is the reason why our proposed model did not adapt to the price change.

2-Strengths and weakness

In this section, we will be discussing the strengths and weaknesses of the LSTM model. LSTM is a powerful recurrent neural network that models sequential data, i.e. Each input is assumed to be dependent on previous ones and it is specifically designed to handle the vanishing gradient problems that arise when learning long-term dependencies. The Lstm has two important components: cell states and hidden states(long-term memory and short-term memory). The cell state has the ability to regulate the information flow through the gate mechanism and this later can remove or add information to the cell. That is why LSTM should be able to overcome long-term dependency and learn complex patterns. This means that learning and remembering long sequences of inputs from previous input is a default behaviour.

In the proposed Lstm model we can see from our experimental results that it relatively succeeded in following the overall pattern of the historical data of bitcoin closing price. It learns the up and down trends. Thus, we almost calibrated the model; however, it fails to capture the peak values and returns unsatisfactory results. Maybe this poor performance is due to frequent fluctuations in the closing price,i.e the trend is highly non-linear. It is important to remember that empirical results will differ with the heterogeneity of problems and that financial time-series are not easy to forecast because otherwise, the cryptocurrency and stock markets would be solved and people would be extremely rich. However, time-series data contain inherent noise which refers to random fluctuations due to unknown and unexplained factors and thus is impossible to capture no matter how prominent the proposed model is.

In addition to the inability of the model to capture the peak values, LSTM has other weaknesses. It is true that the popularity of LSTM lies in their ability to solve the problem of vanishing gradients. But in reality, they fail to remove this issue completely due to the input data being moved from one LSTM cell to another to be evaluated. This process becomes more complex when adding additional features such as input, forget and output non-linear gates to filter out the data. To train the LSTM with such complexity a lot of resources and time is required. For instance, a high memory-bandwidth is needed to remember the previous information for a longer period.

Moreover, tuning hyperparameters can be a weakness too, since there is no guidance on how to tune the hyperparameters such as the number of the hidden neurons, learning rate or the number of stacked layers. Their tuning depends on many characteristics like data, the complexity of data, data-generating process, domain knowledge, etc. Hence, In our model we tuned them by exploring different configurations through trial and error.

Furthermore, the LSTMs are vulnerable to overfitting; it is recommended to use a regularisation method to regularise the network. In our project we used the dropout function on randomly selected nodes with a probability of 20% to drop them out at each weight update, i.e. each epoch during training the model. We believe that applying the dropout function exclusively to the training caused the validation loss of the model to be lower than the training loss since the validation error is computed once the training epochs are completed. However, the gap between the losses could be a result of an unrepresentative validation dataset that fails to capture specific features that are important in generalising efficiently or perhaps due to an inherent random noise. However, even by adding an additional feature which the sentiment of the tweets referred to as polarity we did not get any clear improvement which is acceptable since in reality a tweet can be positive or negative but has no impact or it can be neutral but retweeted frequently which increases the awareness and attract the attention to the cryptocurrency field. In our opinion, the Sentiment tweet feature alone in the context of our LSTM is not enough to determine the impact on the prediction of the bitcoin closing price.

3-Future work

The LSTM model could not capture the excessive jumps due possibly to some architecture limitations such as their sensitivity to different random weight initialisations. In our study we used the Pytorch library to make the forecast. This library handles the tuning of the learnable parameters, i,.e the weights and bias are initialised by small weights and it is known that the performance of a neural model is affected by the learnable parameter initialisation. Therefore, it is worth using other initialisation methods. Moreover, data that is highly non-linear and this might prevent LSTM from representing this complexity, trying another model like a Deep LSTM neural network (DLSTM) [10] which is an extension of the original LSTM may be more beneficial taking in account its ability to learn the non-linearity and the complexity of time-series data through its structure that includes multiple LSTM layers and where each layer contains multiple cells. This model may be the best solution.

Furthermore, tweets tweeted by powerful or famous personalities influence the behavior of their followers as when Elon musk announced that Tesla would not anymore accept the payment with Bitcoin and this announcement caused Bitcoin a loss of 15% of its value during the first 24 hours. Accordingly, adding a certain weight to tweets related to users having a high number of followers might have an impact and it is worthy to try it and we can also include other features such as tweet volumes and see how well the model performs.

4-Ethics

Ethical concerns arise when working with Twitter data regarding the privacy of the author of the tweets collected along with the responsibility to preserve the privacy of persons who entrusts you with their personal tweets. In my project, all of the tweets collected are available publicly by the authors through Twitter, since those tweets were written and sent in public. Therefore, those tweets are accessible through the Twitter API.

Being conscious of the fact that sometimes twitter users are not informed of their tweet status or simply believe that only their friends and or their followers would be authorised to see those tweets on their profile. The collected tweets were not used in any public websites such as Github, since we believe that collecting those tweets through Twitter Api is quite an invasion of their privacy, especially that the authors of those tweets are unaware that other people have the ability to collect any public tweets and use them publicly, in their studies, etc. Additionally, we only extract the sentiment of the tweet without any additional information about the tweets.

Références.

1-Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* 9, 1735–1780 (1997).

3-Francoise Beaufays, Google voice search. URL https://ai.googleblog.com/2015/08/the-neural-networks-behind-google-voice.html

3-Gers, F. A., Schmidhuber, J. & Cummins, F. Learning to forget: Continual prediction with lstm. *Neural Comput.* 12, 2451–2471 (2000).

4-An Infusion of AI Makes Google Translate More Powerful Than Ever (2016).URL **https://www.wired.com/2016/09/google-claims-ai-breakthrough-machine-translation/**

5-Bringing the Magic of Amazon AI and Alexa to Apps on AWS. URL https://www.allthingsdistributed.com/2016/11/amazon-ai-and-alexa-for-all-aws-apps.html

6-Transitioning entirely to neural machine translation. https://engineering.fb.com/2017/08/03/ml-applications/transitioning-entirely-to-neural-machine-translation/

7-Fischer.,krauss.,Deep learning with long short-term memory networks for financial market predictions. URL https://EconPapers.repec.org/RePEc:eee:ejores:v:270:y:2018:i:2:p:654-669

8-Yan.,Ouyang., URL https://www.researchgate.net/publication/321634593_Financial_Time_Series_Prediction_Based_on_Deep_Learning

9-Chung., Shin.,Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction.URL https://www.researchgate.net/publication/328373411_Genetic_Algorithm-Optimized_Long_Short-Term_Memory_Network_for_Stock_Market_Prediction

10-Sagheer, A. & Kotb, M. Time series forecasting of petroleum production using deep lstm recurrent networks. Neurocomputing 323, 203–213 (2019).