Forecasting Cryptocurrency Prices Time Series using Recurrent Neural Networks

Catarina Pinheiro – m20200654

Henrique Eduardo Espadinha Renda – m20200610

Nguyen Huy Phuc – m20200566

Deep Learning

Master Degree Program in Data Science and Advanced Analytics, with a specialization in Data Science

NOVA IMS Information Management School

Abstract

In October 2011, Litecoin, an alternative cryptocurrency, was created. Like Bitcoin, Litecoin is based on an open-source global payment network that is not controlled by any central authority. Cryptocurrency price prediction has become a trending research topic globally. Deep learning algorithms such as Gated Recurrent Unit (GRU), Recurrent Neural Networks (RNN), and Long short-term memory (LSTM) have been used in several projects to predict and to analyze elements disturbing cryptocurrency prices. In this paper, we used LSTM, GRU and BiLSTM to make a cryptocurrency prediction, which focuses only on Litecoin. In order to get better results, we used these models using both single input and multiple inputs. In the end, comparing them, we got to the conclusion that the models, with multiple input, were showing worse results that the ones with single input. The data is time data series from April 28, 2013.

Keywords: Price prediction; Litecoin; Recurrent Neural Network; Long Short-Term Memory; Gated Recurrent Unit

1. INTRODUCTION

1.1. Cryptocurrency

A cryptocurrency is a digital or virtual currency that is secured by cryptography, which secure transaction records, control the creation of additional coins, and verify the transfer of coin ownership. Cryptocurrencies typically use decentralized control as opposed to centralized digital currency and central banking systems. It is a peer-to-peer system that can enable anyone anywhere to send and receive payments.

Instead of being physical money that is carried around and exchanged in the real world, cryptocurrency payments exist purely as digital entries to an online database that describe specific transactions. When you transfer cryptocurrency funds, the transactions are recorded in a public ledger. You store your cryptocurrency in a digital wallet.

Cryptocurrencies are usually built using blockchain technology. Blockchain describes the way transactions are recorded into "blocks" that contains a cryptographic hash of the previous block, a timestamp, and transaction data that are resistant to modification. This is because once recorded, the data in any given block cannot be altered retroactively without alteration of all subsequent blocks.

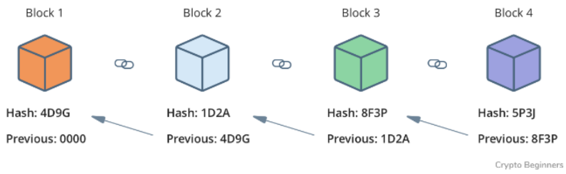

Figure *1*. Blockchain Architecture

1.2. Objective

In recent years, cryptocurrencies are becoming increasingly popular, getting a significant amount of media attention worldwide. The first blockchain-based cryptocurrency was Bitcoin created in 2009, which remains the most popular and most valuable. Today, there are thousands of alternate cryptocurrencies with various functions and specifications emerging: Alternative Coins or Altcoins.

In our project we will focus on this type of assets since we have seen an increased interest on these tokens and we believe that some of these projects can be useful to fix some flaws that Bitcoin present such as scalability (transactions per second), environmental cost and lack of regulation.

We will analyze the behavior of some of the best projects in the recent years and try to predict the value of these tokens in the future using Deep Learning models. To time series data we will use the variants of Recurrent Neural Networks (RNNs) such as Long short-term memory (LSTM), Bidirectional LSTM (biLSTM), and Gated Recurrent Units (GRU).

This prediction will not aim to be a financial advice for the reader, but rather an analysis of the growth of the various projects since every year several ambitious ideas emerge, but few manage to assert themselves and resolve their purpose.

1.3. Dataset

To train the model, we use a dataset taken from Kaggle containing historical price information from a rank 10 cryptocurrency: **Litecoin**.

This information was taken from Coin Market Cap and is available daily from April 28, 2013.

Features	**Description**
Date	Date of observation.
Open	Opening price on the given day.
High	Highest price on the given day.
Low	Lowest price on the given day.
Close	Closing price on the given day.
Volume	Volume of transactions on the given day.
Market Capitalization	The total market value of a cryptocurrency's circulating supply during the day.

Table 1. Variable's description.

2. BACKGROUND

2.1. Long Short-Term Memory (LSTM)

LSTM is an Recurrent Neural Network architecture designed to deal with long time-dependencies. It processes data passing on information as it propagates forward. The differences from RNN are the operations within the LSTM's cells.

2.2. Bidirectional LSTM

A Bidirectional LSTM, or biLSTM, is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. Using bidirectional will run your inputs in two ways, and what differs this approach from unidirectional is that in the LSTM that runs backward you preserve information from the future and using the two hidden states combined you are able in any point in time to preserve information from both past and future.

In problems where all timesteps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTMs on the input sequence. This can provide additional context to the network and result in faster and even fuller learning on the problem.

2.3. Gated Recurrent Unit

The GRU is the newer generation of Recurrent Neural networks and is similar to a LSTM. GRU's don't use the cell state and substitute it by a hidden state to transfer information. It also only has two gates, a reset gate and update gate.

2.4. Regularization L2

In order to further prevent overfitting, we apply L2 regularization in all the models' layers. This type of regularization is called Ridge Regression and adds "squared magnitude" of coefficient as penalty term to the loss function. Mostly, this technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables.

3. MODELING

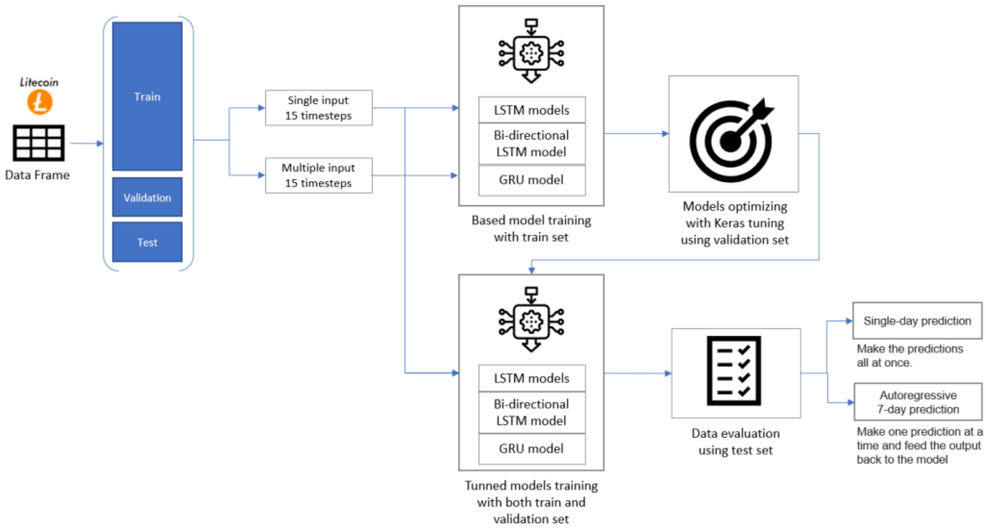

Figure 2 - Proposed methodology

3.1. Data Preprocessing

Firstly, the dataset will be split into 3 sets: training set, validation set and test set for models training and evaluation. Subsequently we scaled our data into range 0 to 1, using min-max normalization.

We select daily close price as our primary feature and train the model with the data of the past 15 days. In our proposed method, we will have 2 different experiment, one using the models with single input and in the other one, we will use multiple input models that are fed by all the input features obtaining in the original dataset.

Single input method	Multiple input method
Training set multiple input: (1988, 15, 1)	Training set multiple input: (1988, 15, 6)
Validation set multiple input: (414, 15, 1)	Validation set multiple input: (414, 15, 6)
Test set multiple input: (415, 15, 1)	Test set multiple input: (415, 15, 6)

Table 2. Dataset description

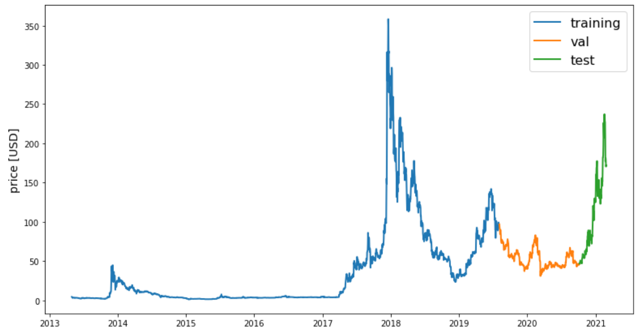

Figure 3 Train, Validation and Test set range

3.2. Model Selection

In the selection stage of this project, 4 RNNs based models are used. The first LSTM, the GRU model and BiLSTM model are referenced from [an article](). The second LSTM model is from the course lecture of RNNs. All four models are then trained with the training set and the hyperparameters were tuned using the validation set. The evaluation metrics used for models training is mean absolute error (MAE).

Figure 4 Training and Validation Loss of single input models

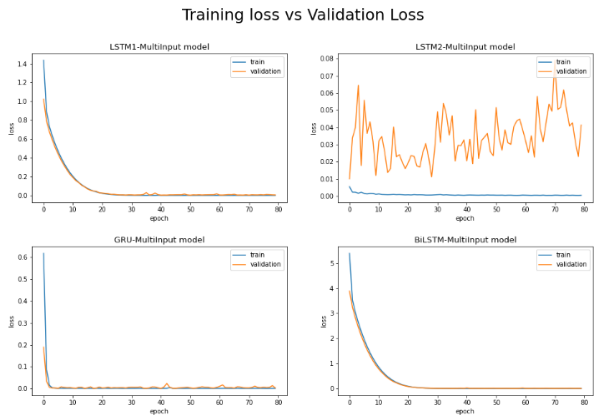

Figure 5 Training and Validation Loss of multiple input models

We can clearly observe that for the model without the regularization (LSTM2) the validation loss is significantly lower than the training loss. This can be due to the higher volatileness in the training data compare to the validation data. Thus, the model is a little bit underfit without the regularization.

3.2.1. Single Input

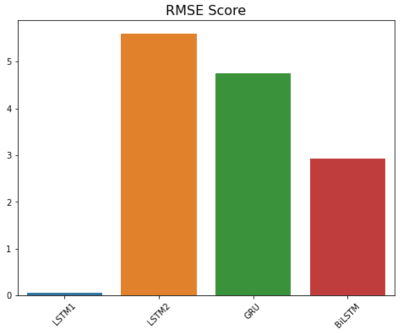

Figure 6 RMSE single input models

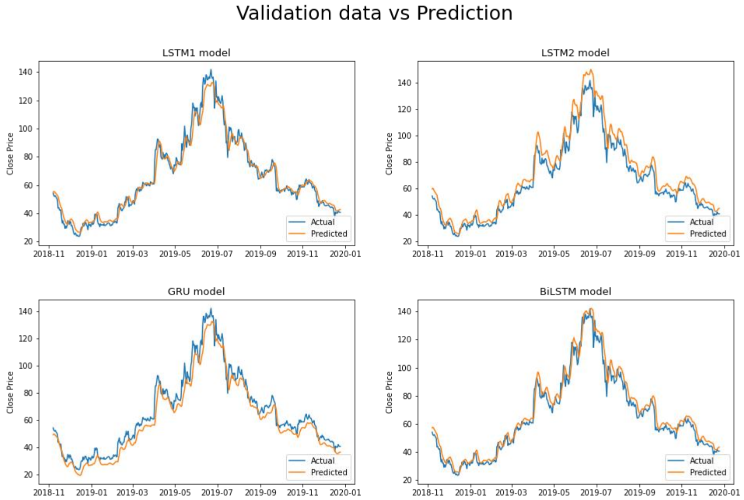

Figure 7 Prediction vs actual of single input models

For the prediction of the validation set, each model was tuned so we can get the best results, besides the one that we used in the practical class. The LSTM1 and BiLSTM were the best models for the single input approach, however since the validation data is considerably low volatile, this might not be the actual performance of the models in practical cases.

3.2.2. Multiple Input

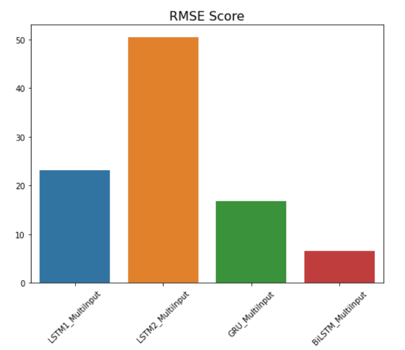

Figure 8 RMSE multiple input models

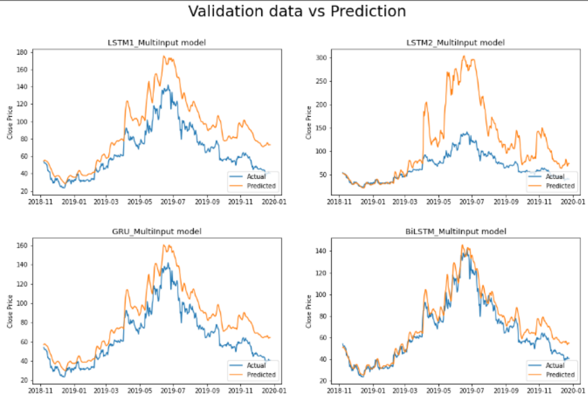

Figure 9 Prediction vs actual of single input models

Even with the best result for multiple input models, the RMSE is still higher than the worse model of the single input. This showed that the base-line LSTM models and its variant is not performing well with this type of data input. However, it is needed to be tried with the test set to conclude this finding.

In the next step, all the models' hyperparameter will be optimized to find the best combinations.

3.3. Model optimization

In order to improve the performance of our model, we used the Keras Tuner library. With this Keras extension we were able to iteratively evaluate the hyperparameter referring to the number of neurons in each layer using a Bayesian Optimization to reduce the value of the validation loss.

We run all our models in ten steps with different number of neurons avoiding having a big number of units to prevent overfitting, and in the final we get the combinations that you can see in the notebook.

3.4. Evaluation

To evaluate our models, we combine the validation data and the test data in the same set, so we get a best prediction in the end. The metrics to measure is Root Mean Square Error (RMSE) and indicates the absolute fit of the model to the data – how close the observed data points are to the model's predicted values.

3.4.1. Single Input

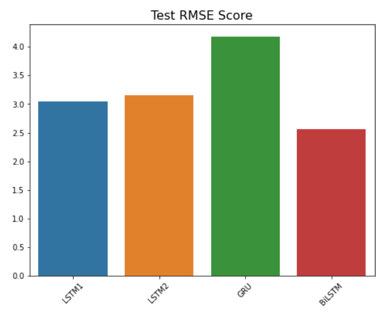

Figure 10 - RMSE test set single input

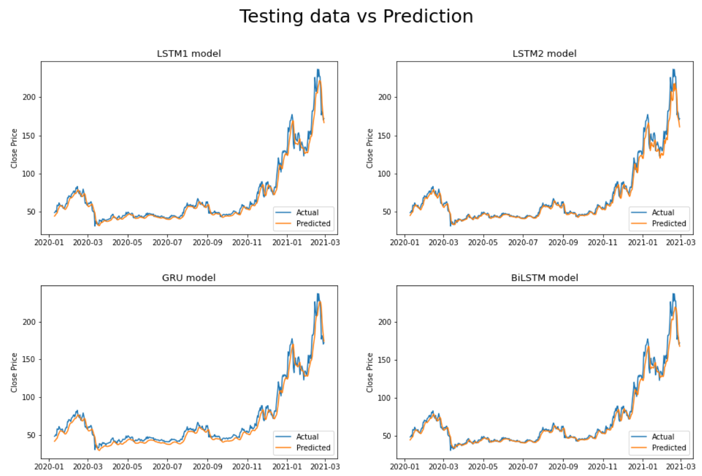

Figure 11 - Prediction vs Actual of test set single input

The result of the test set is good. The two best models are still the LSTM2 and the GRU model. We can conclude that these are the best model for predicting Litecoin price.

3.4.2. Multiple Input

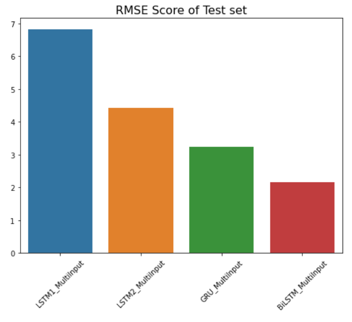

Figure 12 - RMSE test set multiple input

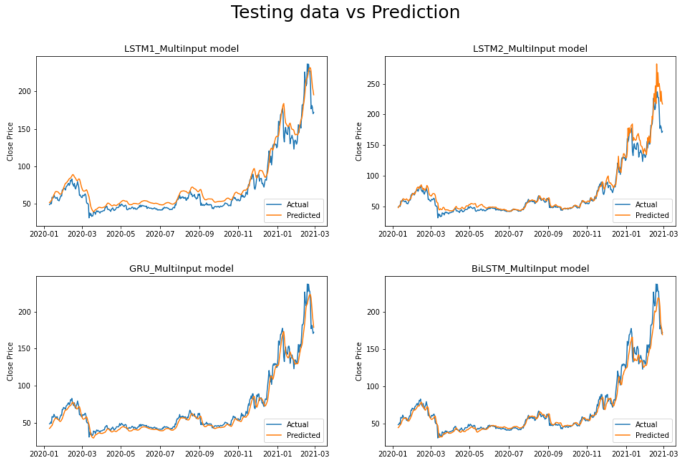

Figure 13 - Prediction vs Actual of test set multiple input

Although, the results of the tunned models show significant improvement in the models' performance on the test set comparing to the validation set. However, once again, the models with multiple input did not show a better result than the ones with single input

4. RESULTS

Having the best performance, single input models will be continued to test in actual recent data that captured from https://finance.yahoo.com/. The data in range from 01/03/2021 to 21/03/2021, with the first 15 days will be set up for the input window-frame. Then we will try to predict 1-day step and 7-day step with the best models obtained from the previous section

With 1 day-step prediction, the best model seems to be the BiLSTM model, which is closely captured the price of the coin

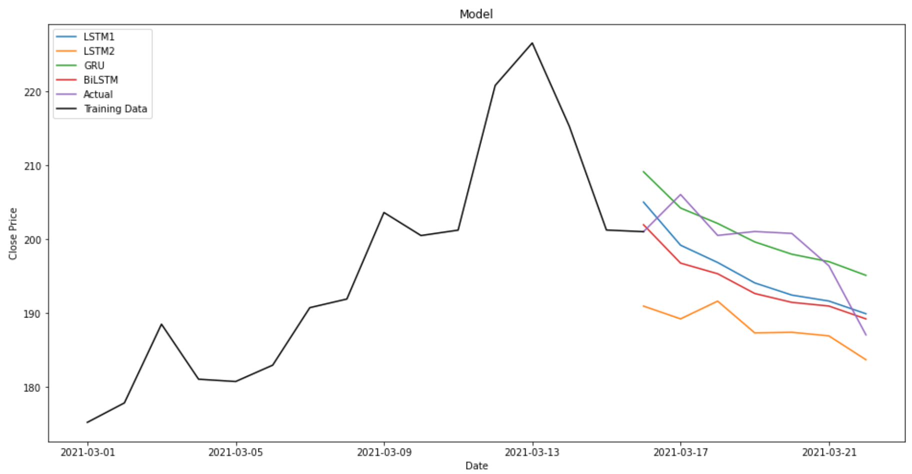

Figure 14 Single-shot 1 day prediction

With the 7 day-step, the predictions of the models are doing worse since the gaps are increasing by time. However, all 4 models were able to capture the trend of the price in the next 7 days with the LSTM1 and BiLSTM having the best results

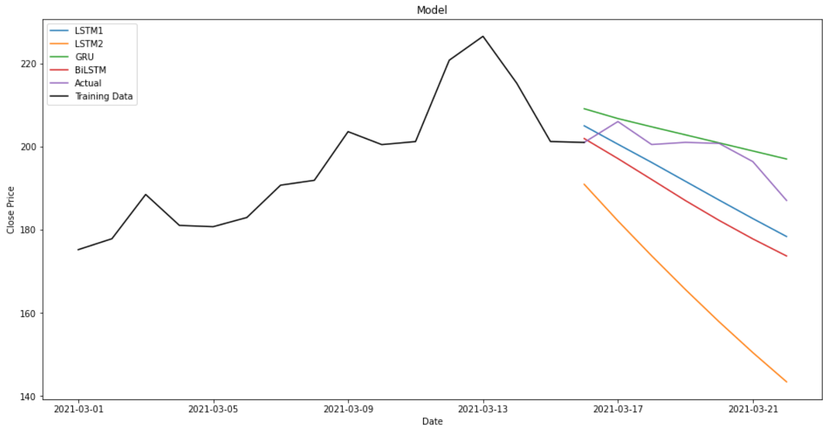

Figure 15 Autoregressive 7-day prediction

5. CONCLUSION

In this project, we presented a general introduction about some variants of RNN, such as LSTM, BiLSTM and GRU architectures and how to use it to train the data to build predictions on unseen data. We used two different approaches, single inputs and multiple inputs and concluded that the last one gave us worse results than the first one. In order to improve the prediction accuracy of the model we tuned the different hyper-parameters using the Keras Tuner library. Even though the best model for single input and multiple input is GRU and Bidirectional LSTM, respectively, our results revealed that the models used in this project are inefficient and unreliable cryptocurrency price predictors. This is probably due to the fact that cryptocurrency prices follow almost a random walk process and the deep learning techniques used in this project are not able to solve this problem. New sophisticated algorithmic methods and alternative approaches should be explored to find the hidden patters that perhaps exist in this problem and then make accurate and reliable forecasts.

6. REFERENCES

[1] Time series forecasting | TensorFlow Core
https://www.tensorflow.org/tutorials/structured_data/time_series

[2] Predictive Analytics: LSTM, GRU and Bidirectional LSTM in TensorFlow
https://towardsdatascience.com/predictive-analysis-rnn-lstm-and-gru-to-predict-water-consumption-e6bb3c2b4b02

[3] Patel, M.M., Tanwar, S., Gupta, R. and Kumar, N., 2020. A deep learning-based cryptocurrency price prediction scheme for financial institutions. *Journal of Information Security and Applications*, *55*, p.102583.