Predicting Price Fluctuations of Cryptocurrencies Using a Temporal Convolutional Network.

by **Jack Featherstone**

Project Supervisor**: Dr. Andrea Visentin**

Presented To

University College Cork

The School of Computer Science and Information Technology.

**Abstract**

Featherstone, Jack Bartholomew

**A Final Year Research Project On: Predicting Price Fluctuations of Cryptocurrencies Using a Temporal Convolutional Network**

Cryptocurrency fluctuation prediction is a conversation that can be shared among various disciplines, including Computer Science, Mathematics, Data Science and Economics.

Despite the various factors that contribute to cryptocurrencies volatile behaviour, many machine learning and deep learning networks have been used to predict cryptocurrency prices with great accuracy.

It is no secret then, why price forecasting for cryptocurrency has become a trending research topic globally.

In this project a Temporal Convolutional Network (TCN) algorithm was used to predict the prices of three types of cryptocurrencies, namely, Bitcoin (BTC), Ethereum (ETH) and Ripple (XRP).

The time-series data that was used was of hourly duration, from 17/08/2017 up until 01/01/2022.

The results show that the proposed algorithm accurately predicts the prices with high accuracy and can therefore be a dependable model that investors can rely on.

**Introduction**

Featherstone, Jack Bartholomew

**A Final Year Research Project On: Predicting Price Fluctuations of Cryptocurrencies Using a Temporal Convolutional Network**

* 1. **Cryptocurrency Growing market**

Since April 2011, when Bitcoin first surpassed one dollar, the term crypto has become progressively recognized as something that is here to stay, rather than some convoluted pyramid-scheme.

At the time of writing this 10/03/2022, the global crypto market capitalization is 1.83 trillion dollars with 18,000 crypto currencies available. – from Investopedia

Binance is one of the largest cryptocurrency exchanges in the world, **Figure 1** shows the increasing rate of users per year.

Table

Description automatically generated

**Figure 1.** Binance users per year

* 1. **Importance**

Before cryptocurrencies, Investors were always on the search for tools and techniques that would increase profit and reduce risk within the stock market. Buying a certain stock involves risk, the aim of the investor is to keep this risk as low as possible while maximizing profits. So naturally, any instrument that could minimise the risk would be valued highly.

Since the virtual explosion of cryptocurrencies as outlined above, the exact same can now be said about cryptocurrency price prediction. Although Stocks and Cryptocurrency are fundamentally different, the way in which they are treated are quite typically the same (a store of value in which you can hold onto or sell).

Cryptocurrency price prediction has become a trending research topic globally and it has created a big opportunity for research.

The importance of being able to predict these high valued currencies, cannot be understated.

Students from various majors are constantly working on ways to more accurately predict the upswing or downswing of a particular currency to know when to invest and when not to.

As technology is advancing, this is the opportunity to find the most informative indicators to make better predictions.

* 1. **General Outline**

In this project a time series forecasting machine learning model was used to accurately predict fluctuations of cryptocurrency prices.

A time series is a sequence of data points that are listed in order of time. Time-series forecasting is a common technique used in many real-world applications such as weather forecasting and financial market prediction. It uses the continuous data for a particular period, to predict the result for the next period (e.g., using the past 3 years of daily temperature in a particular area, to predict next week’s temperature in the same area).

The stock market and now cryptocurrency market is a typical area that presents time-series data and many researchers have proposed various models using this data, for various aims.

It is no secret that Stock markets are rather unpredictable and are affected by many factors causing the high volatility in the market. The exact same can be said about cryptocurrency if not more factors that contribute to the volatile behaviour, one rather comical example of this would be the ‘Musk Effect’ a term coined from when Elon Musk’s tweets had a direct and substantial effect on the price of certain crypto’s. With factors such as media hype lending itself as a cause to why cryptocurrency produces unstable behaviour, as well as many other factors, predicting something in this unstable market will be extremely difficult.

Despite cryptocurrencies volatile behaviour many time-series forecasting algorithms have shown their effectiveness in practice. The most common algorithms are now based on Long-Short Term Memory Networks which are based off Recurrent Neural Networks. One model that was been showing promising results in the Financial Time Series is the Temporal Convolutional Network (TCN).

In this project, A TCN was used to predict the upswing or downswing of a particular crypto based off its closing price (The price this crypto was during a specific time of the day).

**Background**

Featherstone, Jack Bartholomew

**A Final Year Research Project On: Predicting Price Fluctuations of Cryptocurrencies Using a Temporal Convolutional Network**

In the past, statistical time series models have been used to predict the price of stock prices.

The most popular statistical forecasting method for finance is the ARIMA method.

The ARIMA model stands for Auto Regressive Integrated Moving Average model. It includes both an Auto-Regressive component and a Moving Average component with differencing.

The main benefit of using the ARIMA model is that its transforms a non-stationary timeseries into a timeseries without seasonality or trend by predicting the differences of the timesteps from one timestep to the next, giving a constant mean over time as shown in **Figure 2**.

The ARIMA model does require statistical assumptions about the data. Hence why it would be used in situations where the timeseries is stationary or stationary except for a moving mean.

Some previous work has been done on a model for forecasting Bitcoins’ Exchange Rate Bakar N.B. et al. [1]. Although this method produced a reliable forecasting model, highly volatile environments such as cryptocurrency price prediction creates larger error.

More recent techniques in deep learning have come out such as Recurrent Neural Networks (RNN). Two examples of RNN’s are Long Short-Term Memory (LSTM) and Grated Recurrent Unit (GRU) algorithms. These networks behave in the same way, that is, the output of the previous timestep is fed back into the current timestamp, to help in memorize previously seen inputs, **Figure 3**.

Hamayel M.J. et al. [2]. The deep learning models considered for this study involves learning algorithms such as LSTM, bidirectional LSTM and GRU to predict the prices of three different cryptocurrency prices -Ethereum (ETH) , Bitcoin (BTC) and Litecoin (LTC).

The GRU algorithm outperformed the other algorithms. It was shown to be reliable and acceptable for cryptocurrency prediction, **Figure 4**. However, all algorithms were noted as having excellent predictive results.

The influence the news and social networking cites have on cryptocurrency prices has also been a topic of great research. Kraaijeveld O. et al. [3]. It was found that Twitter sentiment analysis has predictive power for the returns of BTC , Bitcoin Cash (BHS) and LTC. It was also noted that favourable sentiment is less likely than unfavourable sentiment with BTC.

Shumin Deng et al.[4] discovered that TCNs significantly outperformed baseline models like ARIMA and LSTM when forecasting stock trend with abrupt changes. It achieves much better performance than either traditional ML models, or deep neural networks (such as LSTM and CNN), suggesting that TCN has a more obvious edge in sequence modelling and classification problems. Hence the reason why a Temporal Convolutional Network (TCN) will be implemented in this paper.

**Graphical user interface, text, application, email

Description automatically generated**

Google, (2017) , [Stationary and non-stationary data]. Retrieved Wednesday 23 , 2022 from <https://www.quora.com/What-are-stationary-and-non-stationary-series>

**Figure 2.** Transformation from non-stationary data to stationary, having a constant mean over time.

Diagram

Description automatically generated

**Figure 3.** An intuition on how the internal memory works, X2 relying on the output of X1.

Table

Description automatically generatedTable

Description automatically generated

**Figure 4.** Here we have the predictive results from M.J. Hamayel et al. [2].

Featherstone, Jack Bartholomew

**A Final Year Research Project On: Predicting Price Fluctuations of Cryptocurrencies Using a Temporal Convolutional Network**

**Dataset**

Featherstone, Jack Bartholomew

**A Final Year Research Project On: Predicting Price Fluctuations of Cryptocurrencies Using a Temporal Convolutional Network**

The data used in this project was collected from Binance.com, an open-access website. Binance is one the largest cryptocurrency exchanges where users can trade cryptocurrencies.

The data consisted of three different csv files, the first sheet for BTC (**Figure 5)**, the second for ETH and the least sheet for XRP. The recorded prices in this dataset were collected on an hourly basis, for BTC and ETH the data was collected from 17 August 2017 to 11 February 2022 with 39221 records and for XRP the data was collected from 04 May 2018 to 11 February 2022 with 33017 records.

The data collected had five features, which are as follows.

- **Open**: Opening price of the cryptocurrency for the day.

- **Close**: Closing price of the cryptocurrency for the day.

- H**igh**: Highest price of the cryptocurrency for the day.

- **Low**: Lowest price of the cryptocurrency for the day.

- **Volume**: Volume of the cryptocurrency traded in the day.

Data Pre-Processing

In the pre-processing step the data was normalized using the minmax normalization to convert the values in the range of 0-1.

Only the closing price was used to predict the fluctuation of each cryptocurrency (univariate).

After normalization, the data is then arranged into multiple input-output pairs (rolling window format) in preparation to be fed into the model. **Figure 6.** Shows an example of data in the rolling window format.

The model then predicts the next hourly closing price. This predicted price can then be fed back into the model to forecast the subsequent timestep. This process will proceed until the number of predictions made equals the number of timesteps you want to predict in the future. **Figure 7.** Illustrates predicting 3 timesteps into the future.

The Models were trained with 70% of the input-output pairs. The validation data used 20% of the input-output pairs and the Testing dataset was left with 10% of the input-output pairs.

Table

Description automatically generated

**Figure 5.** Screenshot showing a sample of the data from the BTC dataset.

Text

Description automatically generated with low confidence

**Figure 6.** A simplified view of how rolling window splits the data for input to a model

Text

Description automatically generated with medium confidence

**Figure 7.** A simplified view of predicant 3 timestamps into the future using window size of 16.

**Baseline Model**

Featherstone, Jack Bartholomew

**A Final Year Research Project On: Predicting Price Fluctuations of Cryptocurrencies Using a Temporal Convolutional Network**

A baseline model was picked so the results found using the TCN, could be compared against the findings of this baseline model. The baseline that was decided on was Patel M.M. et al. [5].

The reasons why this model was picked as a baseline was that it was recently published - 04/06/2020, it had been moderately cited and its subject matter suited our project description.

The model proposed in the Baseline was a hybrid model, consisting of two Long Term Short-Term Memory Networks (LSTM) and a single Gated Recurrent Unit (GRU) network.

**Figure 8.** The model was compiled using Adam as an optimizer and ReLU as the activation function. The model was trained for 154 epochs.

LSTM’s and GRU’s serve relatively the same function (attempting to capture long term temporal patterns). A separate model using only the two LSTMs was also trained and tested on, to see if the separate GRU captured temporal patterns that the two LSTMs could not. **Figure 9.** The model was compiled using Adam as an optimizer and ReLU as the activation function also. The model was trained for 148 epochs.

Both the baseline models were trained and tweaked using the BTC dataset, then the same models were trained and tested on the XRP and ETH datasets.

Chart, waterfall chart

Description automatically generated

**Figure 8.** The LSTM and GRU based hybrid model.

Chart, waterfall chart

Description automatically generated

**Figure 9**. The LSTM model.

**TCN Model**

Featherstone, Jack Bartholomew

**A Final Year Research Project On: Predicting Price Fluctuations of Cryptocurrencies Using a Temporal Convolutional Network**

In the past, Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been the architecture of choice for sequence modelling. There has been results put forward, to prove that convolutional networks should be considered in the conversation as the primary model for predicting sequential data (S.Bai , 2018).

These results showed that convolutional networks outperformed the forementioned RNNs in many tasks, while simultaneously avoiding the vanishing gradient problem, a problem that plagues RNN’s. As the backpropagation gradient flow is not dependent over time, this brings additional benefits such as the ability to train in parallel for much faster training using GPU optimization.

Temporal Convolutional Networks (TCNs) are a class of time-series models that capture long-range patterns using the order of temporal convolutional filters.

A TCN is a variation of a CNN with 2 main features. Firstly, each hidden layer is the same length as the input layer. This is achieved using zero padding.

Secondly, the network only uses information from past time steps. This is implemented by using casual convolutions, both features can be seen in **Figure 10**.

The last thing to consider for TCN models is the receptive field. The receptive field is the outputs dependencies from the original input. The larger the receptive field the greater the chance of capturing long range dependencies. With casual convolutions it is not feasible to capture long range dependencies without your network growing too large from adding too many layers. It is not recommended to use larger kernel sizes either as this adds weights in order to describe the kernel, this is not parameter efficient as most of the time we have multiple filters per layer, this could quite quickly lead to overfitting.

The solution is to use dilation casual convolutions, we stretch the kernel position out by a dilation factor so as a result, the receptive field is also extended by a dilation factor. We stack these dilated casual convolutional layers, so we use all inputs as a receptive field as seen in **Figure 11**.

The formula’s below are to calculate the number of layers in the network and the padding for each dilated layer.

A picture containing text, clock

Description automatically generated

Figure to calculate the number of layers needed in the network, l = input\_size, b = dilation\_base, k = kernel\_size. Roundup the result.

Text

Description automatically generated with low confidence

Figure to calculate padding for each layer, b = dilation\_base, I = number of layers beneath , k = kernel\_size

Timeline

Description automatically generated

**Figure 10.** Here we can see 24 timesteps, padded at each convolutional layer to keep input same length as output. The output of timestep 1 is only ever dependent on timesteps that come on and before timestep 1.

Chart, bubble chart

Description automatically generated

**Figure 11.** Here we can see that less layers are needed using dilated convolutional layers.

**Experimental**

Featherstone, Jack Bartholomew

**A Final Year Research Project On: Predicting Price Fluctuations of Cryptocurrencies Using a Temporal Convolutional Network**

**6.1. Google Cloud**

Training sophisticated deep learning models can take a significant amount of computational power. Google Cloud enables users to run an instance in the cloud to handle all the heavy computation.

The models presented in this report were not extremely complex. Despite this, the models training process would have taken significantly more time if ran on a local machine than with Google Cloud.

The specifications of the machine used in this project:

**GPU**: NVIDIA Tesla T4.

**CPU**: 8vCPU, 30GB memory.

**Pre – Packaged with**: TensorFlow2.8/Keras. CUDA11.3.

**Boot Disk Image:**  Debian GNU/Linux 10 (buster) with 200GB (standard persistent disk).

TheNVIDIA Tesla T4 was picked because it was specifically designed for high performance computing and deep learning training. This GPU was the least powerful GPU on offer.

The option was there to upgrade the hardware including the GPU in the instance, although it was not necessary as these models were not in need of anymore computation power.

This instance enabled immediate learning, with all the pre-packaged frameworks, libraries, and drivers pre-installed.

**6.2. Configuration**

The TCN that produced the best results in training consisted of 3 layers, each layer consisting of 64 filters, kernel size = 3 and dilation rate = 3 and window size = 24. The smaller network was picked as it prevented overfitting. The activation function used was ReLU and the optimizer used was Adam.

Keras Tuner was used to optimize the hyper-parameter selection process. These hyper-parameters included the learning rate, kernel size, window size and the number of filters per layer.

**6.3. Evaluation Metrics**

The evaluation of the proposed scheme was done using mean absolute error (MAE), mean squared error (MSE), mean absolute percentage error (MAPE) and root mean squared error (RMSE).

**6.4. Results**

The price prediction has been made for different window lengths: *1 hour, 3 hours, 5 hours, 7 hours, 9 hours and 11 hours*. In order to judge the actual performance of the proposed approach, these predictions are computed from the test set, a set that was not utilized to train model.

Figures 12 – 17 shows the comparison of errors in prediction from the LSTM baseline, GRU and LSTM hybrid and the proposed approach based off the number of days of the prediction.

Figures 18 – 21 shows the comparison of errors in prediction from the LSTM baseline, GRU and LSTM hybrid and the proposed approach based off the evaluation metrics.

**6.5. Conclusions**

The GRU algorithm outperformed the other algorithms. It was shown to be reliable and acceptable for cryptocurrency prediction,

In our study, we not only compared ARIMA with simple LSTM, GRU, and RNN but also compared it with TCNs,

In our analysis we found that ARIMA, GRU and TCN are one of the best performing model for univariate single step price prediction.

We show that temporal convolutional networks are able to learn inherent patterns in sequential data automatically and so TCN could be a feasible model to learn normal time series behaviors, and can be used for anomaly detection. Our TCN-based approach works well on three open real-world datasets. Besides, with skip connection for multi-scale feature mixture it can get greater performance at a small cost.

many machine learning and deep learning algorithms have been used for prediction purposes. In recent times, neural networks have shown promising results in time-series data prediction. Many variants of neural networks are there to analyze cryptocurrency prices. Among all, LSTM has been proved to be the best in until now. This is due to their ability to remember and extract the temporal features of data. In this paper, we proposed a cryptocurrency price prediction scheme that utilizes a hybrid model of GRU and LSTM. Our proposed scheme has proved to be better than the LSTM network, which is evident from the errors of prediction.

**6.6. Future Work**

In the future, one can introduce a more complex model along with the incorporation of multivariate and sentiment data, which can improve the prediction results for cryptocurrencies.

**References**

Featherstone, Jack Bartholomew

**A Final Year Research Project On: Predicting Price Fluctuations of Cryptocurrencies Using a Temporal Convolutional Network**

[1] Bakar A.B. , Rosbi S. (2017) Autoregressive Integrated Moving Average (ARIMA) Model for Forecasting Cryptocurrency Exchange Rate in High Volatility Environment: A New Insight of Bitcoin Transaction.

[2] Hamayel M.J. , Owda A.Y. (2021) A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms.

[3] Kraaijeveld O. , Smedt J.D. (2020) The predictive power of public Twitter sentiment for forecasting cryptocurrency prices

[4] Shumin D. , Zhang N. , Zhang W. , Pan J.Z. , Chen H. , Chen J. (2019) Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network.

[5] Patel M.M. , Tanwar S. , Gupta R. , Kumar N. (2020) A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions.

Increasing the batch size, and also getting rid of the data generators increased the usage of the Graphics Processing Unit (GPU). By calculating the batches in parallel in the GPU memory, this reduced the communication overhead in the CPU. However, it was about finding the balance because I didn’t want the batch sizes too big so that the model could not converge to the best of its ability. The batch size which I ended up deciding on was 512.

Bouncing back and forth , and not going down a lot , we lower the learning rate

Too small traiing will progress very slowly ,