Logo, icon

Description automatically generated

Cryptocurrency Price Prediction using Deep Learning

Prateek Rana

SID: 1824482

**MOD002691: Final Project Report**

**BEng Computer Science**

**Submitted: 25th April 2022**

# Abstract

The cryptocurrency market continues to grow larger despite fluctuation in prices and there are currently over 18,000 cryptocurrencies in use with a combined market cap of £1.55 trillion. Investments in cryptocurrency have a high risk but offer great returns. Many people have started investing in cryptocurrencies to diversify their portfolio. There is a need for an accurate and efficient method of forecasting cryptocurrency prices so investors can make better decisions and make more profit. This research aims to develop a model for predicting the price of Bitcoin, Ethereum, Monero and Ripple using deep learning and evaluate its performance using certain metrics. This research should also be able to identify the optimal values of parameters like the number of epochs and dropout rates to obtain the best results. The proposed model uses a neural network with three LSTM (Long Short Term Memory), three dropout layers and one dense layer. This network is trained using historical time-series data. The actual and predicted prices of the selected cryptocurrency are plotted using python libraries. The proposed model is successful in predicting the prices and it works the best for predicting Ethereum prices with a normalized RMSE of 0.0564. The best performance is achieved with a dropout rate of 0.2 and 25 epochs. All the primary objectives of this research were achieved. Even though the model can predict most of the fluctuation in prices, it should not be used alone to make investment decisions as people may lose money. However, it can help to study cryptocurrency price trends.

# List of Figures

Figure 1: Proposed Method……………………………………………………………………...11

Figure 2: Sample Dataset………………………………………………………………………..12

Figure 3: Cryptocurrency Selection……………………………………………………………..13

Figure 4: Data Preparation……………………………………………………………………....14

Figure 5: Data Preparation………………………………………………………………………15

Figure 6: Creating the Neural Network…………………………………………………………16

Figure 7: Testing the Model…………………………………………………………………….17

Figure 8: Predicting Prices……………………………………………………………………...17

Figure 9: Calculating Evaluation Metrics………………………………………………………18

Figure 10: Plotting the Graph…………………………………………………..........................18

Figure 11: Bitcoin Price Prediction…………………………………………………………….19

Figure 12: Ethereum Price Prediction………………………………………………………….20

Figure 13: Ripple Price Prediction……………………………………………………………..20

Figure 14: Monero Price Prediction……………………………………………………………21

# List of Tables

Table 1: Evaluation Metrics for BTC, ETH, XMR and XRP Prediction…………………….22

Table 2: RMSE & Normalized RMSE for Different Dropout Rates…………………………23

Table 3: RMSE & Normalized RMSE for Different Number of Epochs…………………….23

# Table of Contents

[Abstract ii](#_Toc101768749)

[List of Figures iii](#_Toc101768750)

[List of Tables iv](#_Toc101768751)

[Table of Contents v](#_Toc101768752)

[1. Introduction 1](#_Toc101768753)

[1.1 Aims 2](#_Toc101768754)

[1.2 Background Study 3](#_Toc101768755)

[2. Deep Learning and Artificial Neural Networks 6](#_Toc101768756)

[3. Requirements of the Project 8](#_Toc101768757)

[4. Methodology 10](#_Toc101768758)

[4.1 Proposed Method 10](#_Toc101768759)

[4.2 Data Collection 11](#_Toc101768760)

[4.3 Data Normalization 12](#_Toc101768761)

[5. Program Design and Implementation 13](#_Toc101768762)

[5.1 Cryptocurrency Selection 13](#_Toc101768763)

[5.2 Collecting and preparing the data 14](#_Toc101768764)

[5.3 Creating the Neural Network 15](#_Toc101768765)

[5.4 Testing the model and Predicting Prices 16](#_Toc101768766)

[5.5 Calculating evaluating metrics and Plotting the Predictions 17](#_Toc101768767)

[6. Analysis and Results 19](#_Toc101768768)

[6.1 Price Prediction Plots 19](#_Toc101768769)

[6.2 Evaluation Metrics 21](#_Toc101768770)

[6.3 Parameters 22](#_Toc101768771)

[7. Discussion 24](#_Toc101768772)

[8. Conclusion 26](#_Toc101768773)

[9. References 27](#_Toc101768774)

[Appendix I: Interim Report 30](#_Toc101768775)

[Table of Contents 31](#_Toc101768776)

[1.Research Proposal 31](#_Toc101768777)

[2.Ethics Certificate 33](#_Toc101768778)

[3.Work Plan (Gantt Chart) 33](#_Toc101768779)

[4.Curriculum Vitae 34](#_Toc101768780)

[5.Exit Plan 35](#_Toc101768781)

[Appendix II: Poster 36](#_Toc101768782)

[Appendix III: Instructions to Run the Program 37](#_Toc101768783)

[Appendix IV: Source Code 38](#_Toc101768784)

# Introduction

The first cryptocurrency transaction happened in May 2010, when Laszlov Hanyecz used 10,000 bitcoins to buy two pizzas. Today, these bitcoins would have been worth more than 300 million pounds.

The idea for a currency that could be sent untraceably ad without the need of centralized entities first came up in the 1980s. Digicash, a type of cryptographic electronic payment was implemented in 1995. It required specific encrypted keys to send and receive money. In 1998, Bit Gold was designed by Nick Szabo. It is often called a precursor to Bitcoin as it required the participants to solve cryptographic puzzles using computer power, and those who solved the puzzle received Bit Gold.

On October 31, 2008, a person anonymously published the first paper on Bitcoin using the fictitious name Satoshi Nakamoto and paved the way for Bitcoin and ensuing cryptocurrencies. This paper discussed the functionality of blockchain technology, without which Bitcoins and other subsequent cryptocurrencies would not be possible. The first block of bitcoins was mined in January 2009, but for several months, bitcoins had no value. One bitcoin was less than 10 pence in April 2010, but it soon rose to 7 pounds by May 2011.

Litecoin appeared in 2011 along with several other cryptocurrencies. These cryptocurrencies which were not bitcoins were termed “alt-coins”. Venture capitalists funded another cryptocurrency Ripple in 2012. There were around 500 different cryptocurrencies by the end of 2014. Bitcoin prices rose exponentially from £333 in January 2016 to £15000 by December 2017. A new blockchain project named Ethereum became popular at around the same time. Ethereum blockchain helped generate over 200,000 distinct projects and counting. There are currently over 18,000 cryptocurrencies in use with a combined market cap of £1.55 trillion. Everyday £55 billion worth of cryptocurrencies are traded. Several establishments have begun to accept bitcoin as a means of payment. Bitcoins are even accepted as a legal tender in El Salvador. Unlike the stock market, the price of Bitcoin is unaffected by major events or political intervention.

The overall cryptocurrency market continues to rise despite fluctuations, with Bitcoin leading the way. Many sophisticated investors are investing in cryptocurrency and using it to diversify their portfolio. It is a high-risk investment with great returns. As the market becomes more stable, there is a need of an efficient way of predicting cryptocurrency prices. This would help people take better investment decisions. Deep learning could be implemented to accurately and efficiently forecast cryptocurrency prices.

Section 1 defines the aims of the project and discusses related works in predicting cryptocurrency prices. Some basic machine learning terms are explained in section 2. Section 3 lists the software requirements of this project. The research methodology and data collection strategy are discussed in section 4. Section 5 focusses on the program design and implementation of the proposed model. The results obtained in this study are displayed in section 6, followed by the discussion in section 7 and the conclusion in section 8.

## 1.1 Aims

This study aims to develop a novel approach to predict cryptocurrency prices using deep learning and artificial neural networks. The success of this study will be based on achieving the following objectives:

* The study should be able to predict the prices of Bitcoin, Ethereum, Monero and Ripple.
* It should be able to evaluate the accuracy of the predictions using various metrics.
* It should ideally be able to identify the optimal values of different parameters to offer the most accurate prediction.
* The secondary objective would be to produce an application that would display predicted prices for selected cryptocurrency and offer investment advice.

## 1.2 Background Study

Nakamoto (2008) emphasized on the need for an electronic payment system that was based on “cryptographic proof” and enabled direct transactions between two parties in his white paper. A cryptocurrency is a decentralized online currency that is based on the blockchain technology. Blockchain is a decentralized system that secures and maintains all cryptocurrency transactions. There are only a few price prediction models available as cryptocurrency is a relatively new technology.

Greaves and Au (2015) developed an approach to predict bitcoin prices using Logistic Regression and SVM. Their method could predict the prices with an accuracy of 55%. Shah and Zhang (2015) worked on a time series data approach using Bayesian Regression. This research stated that it could help investors double their earning in 50 days, but it is no longer relevant due to the volatile nature of cryptocurrency prices. Madan, Saluja and Zhao (2015) created three time series data sets with an interval of 30, 60, and 120 minutes using the same dataset. Then they generated three linear models from the datasets using Generalized Linear Models and Random Forest algorithm. To estimate the price of Bitcoin, these models were integrated linearly. The price for the next 10 minutes was predicted with an accuracy of 55% and the daily price was predicted with an accuracy of 98.7% using these models.

As the value of cryptocurrencies varies rapidly over time, showing fundamentally chaotic and unpredictable behaviour, cryptocurrency price prediction is a tough and complex endeavour. Therefore, deep learning techniques could be a viable solution to this problem. Sin and Wang (2017) predicted the next day bitcoin prices using Selective Neural Network ensemble based on Genetic Algorithm. Multi-Layered Perceptron (MLP) was used as the basic unit for the model. This ensemble was employed to predict price trends by solving a binary classification.  The ensemble method GASEN predicted the next day price with an accuracy of 58% to 63%. GASEN is an acronym for Genetic Algorithm based Selective Neural Network Ensemble.

Selvin et al. (2017) used LSTM and RNN approach to accurately predict stock prices in the Indian stocks market NSE (National Stock Exchange). This model can also be used to predict cryptocurrency prices as they vary just like stock prices, even though they are affected by different factors.

A multiple regression approach was discussed by Saad and Mohaisen (2018), which was based on highly correlated features such as hash rate, transaction rate and price. Pant et al. (2019) used twitter sentiment analysis to predict the price of bitcoins. Their approach collected tweets about bitcoins and then classified them into negatives and positives. This data was fed to a recurrent neural network along with historical data to predict bitcoin prices. The main objective of the study was to create an accurate sentiment analyser, but the overall price prediction accuracy was found to be 77.82%, which showed potential for neural networks in predicting cryptocurrency prices.

Ayaz, et al. (2020) discussed the use of traditional Auto Regressive Integrative Moving Average (ARIMA) for predicting the price of bitcoin. Kumar & Rath (2020) analyzed how different deep learning techniques can be used to predict Ethereum price. To predict the price, they used MLP and LSTM techniques on historical data. The performance of this model was evaluated using metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Their model was found to be too simple and inaccurate.

# Deep Learning and Artificial Neural Networks

Machine learning (ML) focusses on educating computers to learn from the data provided instead of programming them explicitly. A machine learning model improves and learns as it is used. ML algorithms are trained to identify patterns in massive datasets, and then use these patterns to make better decisions and predictions. The larger the training data set, the more accurate the decisions become. The initial data used to instruct a machine learning algorithm to accomplish the required task is known as training data. The data used to evaluate how well the model is performing is known as testing data

Deep Learning refers to a class of robust ML algorithms that specialise in tackling nonlinear and complicated issues by utilising large quantities of data to create efficient prediction models. It is a subset of ML.

A neural network is a program that tries to mimic how people and animals think and learn. It is made up of multiple neurons that are wired up in certain ways and produce output based on a weighted function of inputs. There are a variety of approaches, but for now, a neural network can be considered as a series of layers of neurons, each of which outputs a weighted sum of the neurons in the preceding layer.

The number of neurons in a typical artificial neural network can vary greatly according to the need of the application. These neurons are organised in layers that are linked to one another. An input layer receives input from the outside world, which the network recognises and responds to. There are output units that provide an output that corresponds to the applied inputs. The output is determined by how the network processes and learns about the input. Between the input and output layers, there exist some hidden layers. A quantity known as weight, which can be positive or negative, is used to express the connections between two units.

Artificial neural networks (ANNs) are one of the most extensively used forecasting tools in a variety of industries, including engineering, banking, and healthcare. This is due to their ability to execute complicated tasks with high accuracy, such as categorization, pattern recognition, and prediction. As the financial market is highly non-linear and temperamental, artificial neural networks are gaining popularity in the finance sector.

In this study, the focus would be on recurrent neural networks (RNNs). These networks are artificial neural networks that use sequential or time-series data.

# Requirements of the Project

This project has been developed using Python on PyCharm 2021 IDE. Python can be efficiently used to model and solve machine-learning problems. It is a powerful high-level language that is open-source and easy to read. Various open-source python libraries exist, which help the developer in solving advanced problems and creating complex programs. Some important libraries that were used to develop this project are discussed in the next subsections.

1. Numpy

Numpy is used to execute various mathematical calculations in python. It has powerful data structures that speed up calculations. It features several complicated mathematical functions that can be applied to arrays, lists, and matrices.

1. Pandas

Pandas is a robust Python library for data analysis and manipulation. It is based on the numpy library. It's designed to operate on tabular data. It includes classes that help to read, write, and process.csv files.

1. Matplotlib

Matplotlib is a plotting library for python. It is used to plot static graphs and interactive visualizations. This library, coupled with Numpy, is a viable open-source MATLAB replacement. It has been used in this project to plot graphs of predicted vs actual cryptocurrency prices.

1. Pandas-datareader

This python package creates pandas DataFrame objects using various online sources. It has been used in this project to get finance data using Yahoo API.

1. TensorFlow

Tensorflow is a Google-developed library for machine learning, mainly deep learning. It features a versatile architecture that operates on the CPU, GPU and TPU (Tensorflow Processing Unit). It may be used for a variety of tasks, but in this project, it is mostly used to implement and train neural networks.

1. Keras

Keras is a library for building robust neural networks. It is based on TensorFlow. It contains classes for the implementation for activation functions and layers. It's made to run utilizing both the CPU and the GPU. It supports CNN and recurrent neural networks.

1. Scikit-learn

Sci-kit learn is a machine-learning package written in Python used for data analysis. It consists of classes that assist in the solving various classification, regression, and clustering problems.

# Methodology

This project employs the LSTM (Long Short Term Memory) architecture, which is a RNN used for deep learning. Hochreiter & Schmidhuber (1997) created an innovative, efficient, gradient-based technique that results in several successful runs and learns significantly quicker called LSTM. Recurrent Neural Network is an artificial neural network that operates on the current input while considering the previous output (feedback) and temporarily keeping it in memory. This memory is called short-term memory. RNN memory can be extended using the LSTM model. LSTM uses a set of gates to inject long-term memory into RNNs. This avoids the gradient problem, which occurs when the updates to the various weights inside a neural network get smaller and the neural network stops learning. The relationship between the hidden layers distinguishes it from ordinary RNNs. Unlike RNNs, LSTM does not have the problem of being unable to manage memory at each input with use of memory cells and gate units.

Three LSTM layers, three dropout layers, and one Dense Layer are used in development of the proposed model. These have been discussed in section 5.3.

## 4.1 Proposed Method

Figure 1 shows the proposed prediction method. First, historical data is collected and pre-processed. This data is then divided into training and testing data. The prediction model is trained with the training data. Then, it is evaluated using the test data. Finally, the processed data is used to obtain the prediction result.

Diagram

Description automatically generated

**Figure 1: Proposed Method**

## 4.2 Data Collection

The data is gathered from Yahoo finance API as DataFrame objects using the pandas-datareader library for Python. It is collected from 1st January 2016 to the present date for four cryptocurrencies: Bitcoin, Ethereum, Ripple and Monero. This research uses time-series data, and the testing data is the data collected from 1st January 2021 to the present date. Figure 2 shows a sample of the data obtained using Yahoo finance API for predicting bitcoin prices

**Text, table

Description automatically generated****Figure 2: Sample Dataset**

## 4.3 Data Normalization

The dataset being used in this project is large and heterogenous. Hence, the data has been scaled to values between 0 and 1 using MinMaxScaler as this makes the neural network learning easier. The MinMaxScaler scales all input values such that they all fall inside a range of min to max using this equation:

# Program Design and Implementation

This section discusses the development of the python script that predicts the price of various cryptocurrencies using the proposed model. It can be divided into the subsequent sections.

## 5.1 Cryptocurrency Selection

The cryptocurrency whose price is to be predicted can be selected by changing the value of the variable “currency\_crypto”. It can be set to ‘BTC’ for Bitcoin, ‘ETH’ for Ethereum, ‘XRP’ for Ripple and ‘XMR’ for Monero. The prices predicted are against traditional currencies such as GBP, USD, etc. The against currency can be selected by changing the values of the variable “currency\_against”. Figure 3 shows the python code for this sub-section. The different layers and models imported have been discussed in section 5.3.

Text

Description automatically generated**Figure 3: Cryptocurrency Selection**

## 5.2 Collecting and preparing the data

The data has been collected using Yahoo finance API for a period starting from the start date to the end date. First, a MinMaxScaler is created. The data is then scaled to values between 0-1 using this scaler with the use of fit\_transform. The training data is prepared using the data available for the last 60 days. Figures 4 and 5 show the python code for this sub-section.

Text

Description automatically generated**Figure 4: Data Preparation**

Text

Description automatically generated**Figure 5: Data Preparation**

## 5.3 Creating the Neural Network

The neural network is built using the Sequential model of Keras. This model consists of linearly stacked layers. The various layers used are LSTM (Long Short Term Memory), Dropout & Dense. Three LSTM layers, three dropout layers, and one Dense Layer are used in development of the proposed model. The model is the compiled and trained. The python code for this sub-section is shown in Figure 6.

* LSTM: These layers memorize the important information and feed it back to the neural network.
* Dropout Layer: This layer prevents over-fitting of the neural network. Over-fitting is a scenario that happens when a neural network learns too much from the training data or the noise present in it.
* Dense Layer: This is the final layer. It is used at the end to produce a single value which is the predicted price.

Text

Description automatically generated**Figure 6: Creating the Neural Network**

## 5.4 Testing the model and Predicting Prices

The testing data is collected from 1st January 2021 to the present date. The total dataset is obtained by concatenating the training and test data. Then, the model inputs are created for predicting the cryptocurrency price. These inputs are reshaped and scaled down. Finally, the prices are predicted, and scaled inversely to get the actual predicted price. Figures 7 and 8 show the python code for this sub-section.

Text

Description automatically generated**Figure 7: Testing the Model**

Text

Description automatically generated**Figure 8: Predicting Prices**

## 5.5 Calculating evaluating metrics and Plotting the Predictions

The MAE, MSE and RMSE for the proposed model are calculated using functions from sklearn.metrics. These metrics and their formulae are discussed in section 6.2. After prediction, the actual and predicted prices are plotted using matplotlib. The actual prices are represented by black colour and the predicted prices by green colour. Figures 9 and 10 show the python code for this section.

Text

Description automatically generated**Figure 9: Calculating Evaluation Metrics**

**Text

Description automatically generated****Figure 10: Plotting the Graph**

# Analysis and Results

In this section, the results obtained using the proposed model that uses multiple LSTM layers are presented. It also discusses the various evaluation metrics that can be used to assess the proposed model's performance.

## **6.1 Price Prediction Plots**

The price prediction for Bitcoin (BTC) is displayed in figure 11. Figure 12 displays the price prediction for Ethereum (ETH). Similarly. Ripple (XRP) and Monero (XMR) price prediction is depicted in figures 13 and 14 respectively. In figures 11 to 14, the black curve shows the actual price, and the green curve shows the predicted price. All the prices are in pounds.

Chart, line chart, histogram

Description automatically generated

**Figure 11: Bitcoin Price Prediction**

Chart, histogram

Description automatically generated

**Figure 12: Ethereum Price Prediction**

Chart, histogram

Description automatically generated

**Figure 13: Ripple Price Prediction**

Chart, histogram

Description automatically generated

**Figure 14: Monero Price Prediction**

## 6.2 Evaluation Metrics

In every machine learning model, determining the model's accuracy is a critical step. The metrics Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) were used in this research to assess the proposed model's performance. The formulae and definitions for these metrics were provided by Chai and Draxler (2014).

1. Mean Absolute Error: It is the average of all the differences between actual and predicted values.
2. Mean Squared Error: It is the average of the square of all the differences between actual and predicted values.
3. Root Mean Squared Error: It is defined as the Mean Squared Error’s square root.

According to Brownlee (2016) the RMSE of a model will always be equal to or greater than its MAE and it determines how well a model can predict a continuous value. It has the same units as the data for which it is used. The RMSE is not greatly affected by small errors, but large errors can cause significant variation in the RMSE value.  A low value of RMSE means that the model has a high accuracy. Table 1 depicts the MAE, MSE and RMSE values for different cryptocurrency prediction. As RMSE value depends on the range of values in the dataset, it would be greater for cryptocurrency like Bitcoin that is very expensive compared to Ripple. Hence, the RMSE value should be normalized to compare different cryptocurrencies. This can be done by dividing the RMSE value by the difference of the maximum and minimum values in the dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cryptocurrency | MEA | MSE | RMSE | Normalized RMSE |
| Bitcoin (BTC) | 1685.84 | 4605400.68 | 2146.02 | 0.0757 |
| Ethereum (ETH) | 131.34 | 28731.66 | 169.96 | 0.0564 |
| Monero (XMR) | 12.46 | 290.16 | 17.03 | 0.0675 |
| Ripple (XRP) | 0.107 | 0.0212 | 0.145 | 0.1241 |

**Table 1: Evaluation Metrics for BTC, ETH, XMR and XRP Prediction**

## 6.3 Parameters

The accuracy of the proposed model was analysed using the following parameters:

* Dropout Rate: This number represents the proportion of neurons that are disabled to prevent over-fitting in the model. It has a value in the range of 0 to 1. If the dropout rate is 0, no neurons are disabled. The dropout values 0.1, 0.2 and 0.3 were tried in predicting the price of Ethereum (ETH). Table 2 shows the RMSE and normalized RMSE values for Ethereum price prediction.

|  |  |  |
| --- | --- | --- |
| Dropout Rate | RMSE | Normalised RMSE |
| 0.1 | 240.62 | 0.07985 |
| 0.2 | 169.96 | 0.0564 |
| 0.3 | 250.87 | 0.0832 |

**Table 2: RMSE & Normalized RMSE for Different Dropout Rates**

* Number of Epochs: One full iteration over the training data is one epoch. Therefore, as the number of epochs increases, training the model takes more time. The epoch values 25, 50 and 75 were tried in predicting the price of Ethereum (ETH). Table 3 shows the RMSE and normalized RMSE values for Ethereum price prediction.

|  |  |  |
| --- | --- | --- |
| Number of Epochs | RMSE | Normalized RMSE |
| 25 | 169.42 | 0.0564 |
| 50 | 233.64 | 0.0775 |
| 75 | 147.25 | 0.0489 |

**Table 3: RMSE & Normalized RMSE for Different Number of Epochs**

# Discussion

The proposed model has been successfully able to predict the price of Bitcoin, Ethereum, Monero and Ripple. In figures 11 to 14, the black curve shows the actual price, and the green curve shows the predicted price. The points at which these curves intersect represent the days on which the cryptocurrency prices were predicted with a 100% accuracy using the proposed model, but there are only few such points. The price of cryptocurrencies can be influenced by uncertain factors, so it is difficult to give a 100% accurate prediction every day. However, the proposed model can predict the trends in cryptocurrency prices accurately. When the actual prices are increasing over time, the predicted prices follow the same trend. Similarly, when the actual prices are decreasing over time, the predicted prices decrease too.

Although several theories and algorithms for predicting the price of cryptocurrencies have been developed, most of them have overfitting issues and inaccuracies caused by large datasets. The proposed model solves these shortcomings with the use of multiple LSTM and dropout layers.  Table 1 shows the MAE, MSE, RMSE and normalized RMSE values for four cryptocurrencies. As lower RMSE and normalized RMSE values indicate high accuracy, the proposed model makes the most accurate prediction for Ethereum (ETH) with a normalized RMSE of 0.0564 and the least accurate prediction for Ripple (XRP) with a normalized RMSE of 0.1241. It fairly predicts Bitcoin (BTC) and Monero (XMR) prices with a normalized RMSE of 0.0757 and 0.0675.

The results indicate that the proposed model could be unreliable at times because predicting cryptocurrency prices is a complex issue, that even modern deep learning techniques such as LSTM and CNNs fail to tackle effectively. As investments in cryptocurrency are high risk, the proposed model alone should not be used to make investment decisions. However, it can help to study cryptocurrency price trends.

In the future, this model can be used along with other deep learning models like ARIMA or GASEN to make better predictions. It could also be integrated with twitter sentiment analysis to create a tool that reduces the uncertainty in predicting cryptocurrency prices. The effect of changing the LSTM layers, number of neurons, batch size and optimizer on the prediction RMSE can be studied to develop a more accurate model. Furthermore, an application that predicts cryptocurrency prices using different models and offers investment advice could be developed.

From the results in table 2, it is evident that the proposed model performs the best and has the lowest RMSE when the dropout rate is 0.2. From the results in table 3, it is observed that the model has the highest normalized RMSE using 50 epochs and the lowest normalized RMSE using 75 epochs. The RMSE and normalised RMSE values at 25 epochs are not too high and close to the values at 75 epochs. As the number of epochs increases, training the model takes more time. Therefore, it is better to have 25 epochs and a normalized RMSE value of 0.0564 than having 75 epochs and a normalized value of 0.0489.

# Conclusion

The proposed model has been successful in predicting the prices of Bitcoin, Ethereum, Monero and Ripple. The most accurate prediction was for Ethereum (ETH) with a normalized RMSE of 0.0564. The model performs the best with a dropout of 0.2 and 25 epochs. Even though the model could predict most of the fluctuation in prices, it should not be used alone to make investment decisions as people may lose money. All the primary objectives of this research were achieved as the model could predict prices, evaluate performance and identify parameters for the best predictions. The secondary objective could not be achieved as this LSTM based model is not good enough to offer investment advice.

In the future, the proposed model could be integrated with other deep learning models or twitter sentiment analysis to obtain better results. Peer-to-peer transactions are undergoing a breakthrough transformation and with better knowledge of cryptocurrencies, better models and techniques to predict cryptocurrency prices will be discovered soon.

# References

Ayaz, Z., Fiaidhi, J., Sabah, A. and Anwer Ansari, M. (2020). Bitcoin Price Prediction using ARIMA Model.

Brownlee, J. (2019). *Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/> [Accessed 19 Feb. 2022].

Chai, T. and Draxler, R.R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? *Geoscientific Model Development Discussions*, 7(1), pp.1525–1534.

Greaves, A. and Au, B. (2015). *Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin*. [online] Available at: [http://snap.stanford.edu/class/cs224w-2015/projects\_2015/Using\_the\_Bitcoin\_Transaction\_Graph\_to\_sPredict\_the\_Price\_of\_Bitcoin.pdf](http://snap.stanford.edu/class/cs224w-2015/projects_2015/Using_the_Bitcoin_Transaction_Graph_to_Predict_the_Price_of_Bitcoin.pdf) [Accessed 18 Dec. 2021].

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), pp.1735–1780.

IBM Cloud Education (2020). *What are Neural Networks?* [online] www.ibm.com. Available at: <https://www.ibm.com/uk-en/cloud/learn/neural-networks> [Accessed 12 Jan. 2022].

Jones, E. (2022). *A Brief History of Cryptocurrency*. [online] CryptoVantage. Available at: <https://www.cryptovantage.com/guides/a-brief-history-of-cryptocurrency/> [Accessed 15 Feb. 2022].

Kumar, D. and Rath, S.K. (2020). Predicting the Trends of Price for Ethereum Using Deep Learning Techniques. *Advances in Intelligent Systems and Computing*, pp.103–114.

Madan, I., Saluja, S. and Zhao, A. (2015). *Automated Bitcoin Trading via Machine Learning Algorithm*. [online] Available at: <http://cs229.stanford.edu/proj2014/Isaac%20Madan,%20Shaurya%20Saluja,%20Aojia%20Zhao,Automated%20Bitcoin%20Trading%20via%20Machine%20Learning%20Algorithms.pdf> [Accessed 15 Jan. 2022].

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. *SSRN Electronic Journal*.

Pant, D.R., Neupane, P., Poudel, A., Pokhrel, A.K. and Lama, B.K. (2018). Recurrent Neural Network Based Bitcoin Price Prediction by Twitter Sentiment Analysis. *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*.

Reddy, L.S. and Sriramya, D.P. (2015). A Research On Bitcoin Price Prediction Using Machine Learning Algorithms. *International Journal of Scientific & Technology Research*, [online] 9(4), pp.1600–1604. Available at: <https://www.ijstr.org/paper-references.php?ref=IJSTR-0420-33830> [Accessed 25 Jan. 2022].

Saad, M. and Mohaisen, A. (2018). Towards characterizing blockchain-based cryptocurrencies for highly-accurate predictions. *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E.A., Menon, V.K. and Soman, K.P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.

Sin, E. and Wang, L. (2017). *Bitcoin Price Prediction Using Ensembles of Neural Networks*. [online] Available at: <https://personal.ntu.edu.sg/elpwang/PDF_web/08393351.pdf> [Accessed 22 Nov. 2021].

# Appendix I: Interim Report

Logo, icon

Description automatically generated

Predicting Cryptocurrency Prices using Deep Learning

Prateek Rana

SID: 1824482

**MOD002691: Interim Report**

**BEng Computer Science**

**Submitted: 25th April 2022**

# Table of Contents

1. Research Proposal………………………………………………………….....31-32
2. Ethics Certificate……………………………………………………………........33
3. Work Plan (Gantt Chart) ……………………………………….……………......33
4. Curriculum Vitae…………………………………………………….……….......34
5. Exit Plan……………………………………………………………………….....35

# 

# 1.Research Proposal

This study aims to develop a novel approach to predict cryptocurrency prices using deep learning and artificial neural networks

**PROJECT PROPOSAL FORM**

|  |  |
| --- | --- |
| ***Name:*** | Prateek Kumar Rana |
| ***SID:*** | 1824482 |
| ***Email:*** | [pkr114@student.aru.ac.uk](mailto:pkr114@student.aru.ac.uk) |
| ***Degree course:***  *Include designation eg BSc, BEng etc* | BEng (Hons) Computer Science |
| ***Start of project:*** month/year | 11/2021 |
| ***Expected project completion/ submission:*** month/year | 05/2022 |
| ***Draft Project Title:*** | Predicting Cryptocurrency Prices using Deep Learning |
| ***Possible supervisor:***  Suggest a supervisor (1st and 2nd choices, or leave blank) |  |
| ***Aim(s):***  Suggest one or two aims | * The study should be able to predict the prices of Bitcoin, Ethereum, Monero and Ripple. * It should be able to evaluate the accuracy of the predictions using various metrics. * It should ideally be able to identify the optimal values of different parameters to offer the most accurate prediction. |
| ***Previous work:*** Give two literature sources relevant to this work you have consulted (optional, but will help verify the topic is worthy of study) | * Pant, D.R., Neupane, P., Poudel, A., Pokhrel, A.K. and Lama, B.K. (2018). Recurrent Neural Network Based Bitcoin Price Prediction by Twitter Sentiment Analysis. *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*. * Sin, E. and Wang, L. (2017). *Bitcoin Price Prediction Using Ensembles of Neural Networks*. [online] Available at: <https://personal.ntu.edu.sg/elpwang/PDF_web/08393351.pdf> [Accessed 22 Nov. 2021]. |
| ***Methodology and outcomes:***  Describe how you hope to achieve your aims and how you will measure the success of the work (50 words max) | The proposed model will use a neural network with three LSTM layers, dropout layers and a dense layer to predict cryptocurrency prices. The success of the model will be based on the accuracy of the prediction and will be evaluated using the metrics RMSE and MAE. |

# 2.Ethics Certificate

![Graphical user interface, text, application, email

Description automatically generated]()

# 3.Work Plan (Gantt Chart)

Timeline

Description automatically generated

# 4.Curriculum Vitae



# 5.Exit Plan

After graduation, I will look for an internship or a job at a reputed IT organization. I will update my resume and apply to various firms. I would like a job that allows me to utilize my problem-solving skills to further develop my abilities in the field of computer science. I plan to work for a few years and save money to pursue a post-graduate degree from a good university. Ultimately, I would like to use my experience and knowledge to start-up a business.

# Appendix II: Poster

Text

Description automatically generated

# Appendix III: Instructions to Run the Program

1) Make sure you have python and the following python libraries/packages installed:

* numpy
* pandas
* matplotlib
* keras
* tensorflow
* pandas-datareader
* scikit-learn or sklearn

If you don't have some of the packages installed, open command prompt/terminal and type:

pip install package\_name

2) Make sure you have an internet connection

3) Select the cryptocurrency whose price you want to predict by chaning the value of the varible "curr\_crypto".

* By default it is set to curr\_crypto='ETH' for Ethereum price prediction. You can change it to:
* curr\_crypto='BTC' for Bitcoin
* curr\_crypto='XRP' for Ripple or
* curr\_crypto='XMR' for Monero.

4) You can also change the currency against which the cryptocurrency prices are predicted, by default is is set to 'GBP'

This can be changed by changing the value of the variable "curr\_against".

Example: curr\_against='USD' for US dollars.

5) Run the python script "CryptoPrediction.py" and please wait to get the results.

Thanks!

# Appendix IV: Source Code

import math

import numpy as my\_npy

import matplotlib.pyplot as my\_plot

import pandas as my\_panda

import pandas\_datareader as my\_reader

import datetime as my\_date

import sklearn.metrics

# To scale the data in between 0-1 to make the neural network work better

import sklearn.metrics

from sklearn.preprocessing import MinMaxScaler

# Importing dense,dropout, lstm layers

from keras.layers import Dropout, LSTM, Dense

from keras.models import Sequential

# select cryptocurrency (BTC/ETH/XRP/XMR)

curr\_crypto = 'ETH'

# select against currency (USD/GBP/etc.)

curr\_against = 'GBP'

# start date

begin\_dt = my\_date.datetime(2016, 1, 1)

# end date

ending\_dt = my\_date.datetime(2020, 12, 31)

# get finance data from yahoo api

my\_data\_set = my\_reader.DataReader(f'{curr\_crypto}-{curr\_against}', 'yahoo', begin\_dt, ending\_dt)

# print(my\_data\_set.head())

# Preparing data

# scaling data to values between (0,1)

sclr\_minmax = MinMaxScaler(feature\_range=(0, 1))

dataset\_scld = sclr\_minmax.fit\_transform(my\_data\_set['Close'].values.reshape(-1, 1))

# number of days the prediction is based on

prdctn\_period = 60

# preparing training data

training\_x, training\_y = [], []

for t in range(prdctn\_period, len(dataset\_scld)):

training\_x.append(dataset\_scld[t - prdctn\_period: t, 0])

training\_y.append(dataset\_scld[t, 0])

# converting to a numpy array

training\_x, training\_y = my\_npy.array(training\_x), my\_npy.array(training\_y)

# reshaping

training\_x = my\_npy.reshape(training\_x, (training\_x.shape[0], training\_x.shape[1], 1))

# Creating the neural network

model = Sequential()

# Adding layers

model.add(LSTM(units=50, return\_sequences=True, input\_shape=(training\_x.shape[1], 1)))

# to prevent over-fitting

model.add(Dropout(0.2))

model.add(LSTM(units=50, return\_sequences=True))

# to prevent over-fitting

model.add(Dropout(0.2))

model.add(LSTM(units=50))

# to prevent over-fitting

model.add(Dropout(0.2))

# to get a single value which would be the prediction

model.add(Dense(units=1))

# Compiling the model

model.compile(optimizer='adam', loss='mean\_squared\_error')

# Training the model

model.fit(training\_x, training\_y, epochs=50, batch\_size=32)

# Testing the model

# start date for test data

test\_begin = my\_date.datetime(2021, 1, 1)

# end dae for test data

test\_ending = my\_date.datetime.now()

# testing data set

testing\_data\_set = my\_reader.DataReader(f'{curr\_crypto}-{curr\_against}', 'yahoo', test\_begin, test\_ending)

# getting actual prices

actual\_crypto\_price = testing\_data\_set['Close'].values

# obtaining the total dataset from testing and training datasets

total\_data\_set = my\_panda.concat((my\_data\_set['Close'], testing\_data\_set['Close']), axis=0)

# model inputs

my\_inputs = total\_data\_set[len(total\_data\_set) - len(testing\_data\_set) - prdctn\_period:].values

# reshaping the model inputs

my\_inputs = my\_inputs.reshape(-1, 1)

# scaling down model inputs

my\_inputs = sclr\_minmax.fit\_transform(my\_inputs)

testing\_x = []

for x in range(prdctn\_period, len(my\_inputs)):

testing\_x.append(my\_inputs[x - prdctn\_period:x, 0])

testing\_x = my\_npy.array(testing\_x)

# reshaping to add a 3rd dimension

testing\_x = my\_npy.reshape(testing\_x, (testing\_x.shape[0], testing\_x.shape[1], 1))

# predicting the price

prdctd\_crypto\_price = model.predict(testing\_x)

# inverse scaling the predicted price to get actual values

prdctd\_crypto\_price = sclr\_minmax.inverse\_transform(prdctd\_crypto\_price)

# mae, mse, rmse calculation

mse = sklearn.metrics.mean\_squared\_error(actual\_crypto\_price, prdctd\_crypto\_price)

print("Mean Square Error: ")

print(mse)

rmse = math.sqrt(mse)

print("\nRoot Mean Square Error: ")

print(rmse)

mae = sklearn.metrics.mean\_absolute\_error(actual\_crypto\_price, prdctd\_crypto\_price)

print("\nMean Absolute Error: ")

print(mae)

# Normalised RMSE

maxP = max(actual\_crypto\_price)

minP = min(actual\_crypto\_price)

normalizedRMS = rmse/(maxP-minP)

print("\n Normalized RMSE: ")

print(normalizedRMS)

# plotting the predicted and actual prices

my\_plot.plot(actual\_crypto\_price, color='black', label='Actual Price')

my\_plot.plot(prdctd\_crypto\_price, color='green', label='Predicted Price')

my\_plot.title(f'{curr\_crypto} Price Prediction')

my\_plot.xlabel('Number of Days')

my\_plot.ylabel(f'Price in {curr\_against}')

my\_plot.xlim(0, 100)

my\_plot.legend(loc='upper left')

my\_plot.savefig("predictionPlot1.png")

my\_plot.show()