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, despite its notorious price fluctuations, exhibits exponential growth, boasting a staggering repertoire of over 18,000 cryptocurrencies with a collective market capitalization exceeding £1.55 trillion. This exponential surge in adoption reflects a burgeoning interest among investors, lured by the promise of substantial returns albeit with commensurate risks.

In response to this dynamic landscape, an imperative emerges for the development of robust methodologies to forecast cryptocurrency prices accurately and efficiently. Such forecasting tools serve as invaluable aids for investors, empowering them to navigate the volatile market with greater precision and optimize their investment strategies for enhanced profitability.

This research endeavors to meet this pressing need by proposing a sophisticated predictive model leveraging deep learning techniques. Specifically, the model focuses on forecasting the prices of prominent cryptocurrencies including Bitcoin, Ethereum, Monero, and Ripple. Deep learning, particularly the utilization of Long Short Term Memory (LSTM) networks, constitutes the cornerstone of this innovative approach, offering unparalleled capabilities in capturing intricate temporal patterns inherent in cryptocurrency price movements.

Central to the model's efficacy is its meticulous calibration, involving the identification of optimal hyperparameters such as the number of epochs and dropout rates. Through rigorous experimentation, this study elucidates the significance of parameter tuning in enhancing predictive performance, thereby furnishing investors with invaluable insights for model optimization.

The proposed model is constructed upon a neural architecture comprising three LSTM layers, strategically interspersed with three dropout layers to mitigate overfitting, culminating in a single dense layer for final output. Leveraging historical time-series data, the model undergoes intensive training to glean underlying patterns and relationships, essential for accurate price forecasting.

To evaluate its efficacy, the model's predictions are juxtaposed against actual cryptocurrency prices, with visualization facilitated through Python libraries. Remarkably, the model exhibits commendable proficiency in predicting price dynamics, with Ethereum emerging as the cryptocurrency with the most accurate forecasts, boasting a normalized Root Mean Squared Error (RMSE) of 0.0564.

While this research represents a significant milestone in the realm of cryptocurrency price prediction, it's imperative to exercise caution in its application. While the model can effectively anticipate fluctuations in prices, it should not serve as the sole determinant for investment decisions, as the inherent volatility of the market entails inherent risks. Nonetheless, it serves as a potent tool for discerning and interpreting cryptocurrency price trends, augmenting investors' decision-making capabilities and fostering a deeper understanding of this evolving financial landscape.

Top of Form

# 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](#_Toc166237203)

[List of Figures iii](#_Toc166237204)

[List of Tables iv](#_Toc166237205)

[Table of Contents v](#_Toc166237206)

[1. Introduction 1](#_Toc166237207)

[1.1 Aim 2](#_Toc166237208)

[1.2 Background Study 2](#_Toc166237209)

[2. Deep Learning and Artificial Neural Networks 5](#_Toc166237210)

[3. Requirements of the Project 6](#_Toc166237211)

[4. Methodology 8](#_Toc166237212)

[4.1 Proposed Method 9](#_Toc166237213)

[4.2 Data Collection 10](#_Toc166237214)

[4.3 Data Normalization 12](#_Toc166237215)

[5. Program Design and Implementation 13](#_Toc166237216)

[5.1 Cryptocurrency Selection 13](#_Toc166237217)

[5.2 Collecting and preparing the data 14](#_Toc166237218)

[5.3 Creating the Neural Network 16](#_Toc166237219)

[5.4 Testing the model and Predicting Prices 18](#_Toc166237220)

[5.5 Calculating evaluating metrics and Plotting the Predictions 19](#_Toc166237221)

[6. Analysis and Results 21](#_Toc166237222)

[6.1 Price Prediction Plots 21](#_Toc166237223)

[6.2 Evaluation Metrics 23](#_Toc166237224)

[6.3 Parameters 24](#_Toc166237225)

[7. Discussion 26](#_Toc166237226)

[8. Conclusion 27](#_Toc166237227)

[9.References 28](#_Toc166237228)

[Appendix I: Interim Report 30](#_Toc166237229)

[Table of Contents 31](#_Toc166237230)

[1.Research Proposal 31](#_Toc166237231)

[2.Ethics Certificate 33](#_Toc166237232)

[3.Work Plan (Gantt Chart) 33](#_Toc166237233)

[4.Curriculum Vitae 34](#_Toc166237234)

[5.Exit Plan 35](#_Toc166237235)

[Appendix II: Poster 36](#_Toc166237236)

[Appendix III: Instructions to Run the Program 37](#_Toc166237237)

[Appendix IV: Source Code 38](#_Toc166237238)

# Introduction

The genesis of cryptocurrencies traces back to the innovative transaction of May 2010, when Laszlo Hanyecz famously exchanged 10,000 bitcoins for two pizzas—a seemingly ordinary event that now symbolizes a remarkable moment in financial history (Nakamoto, 2008). Had Hanyecz retained those bitcoins, their value today would surpass a staggering 300 million pounds, highlighting the exponential growth and potential of cryptocurrencies (Hanyecz, 2010).

However, the concept of digital currencies transcends this singular event. The notion of a decentralized, untraceable currency emerged as early as the 1980s (Chaum, 1983). It wasn't until 1995 that the first practical implementation, Digicash, introduced the idea of cryptographic electronic payments (Chaum, 1995). This system relied on encrypted keys for transactions, foreshadowing the cryptographic underpinnings essential to modern cryptocurrencies.

In 1998, Nick Szabo conceptualized Bit Gold, a precursor to Bitcoin, requiring participants to solve cryptographic puzzles—a concept echoing Bitcoin's later proof-of-work mechanism (Szabo, 1998). Then, on October 31, 2008, under the pseudonym Satoshi Nakamoto, an anonymous individual published the groundbreaking Bitcoin whitepaper, elucidating the concept of blockchain technology—a distributed ledger system pivotal to cryptocurrencies' existence (Nakamoto, 2008).

The mining of the inaugural block of bitcoins in January 2009 marked the birth of the cryptocurrency era. Initially negligible in value, the trajectory of Bitcoin's worth swiftly ascended, reaching seven pounds by May 2011, heralding the dawn of a new financial paradigm.

Bitcoin's rise catalyzed the emergence of alternative cryptocurrencies, termed "alt-coins," with Litecoin making its debut in 2011, alongside numerous others. Venture capitalists further fueled the sector, funding projects like Ripple in 2012. By 2014, the ecosystem boasted around 500 diverse cryptocurrencies, a testament to the burgeoning interest in decentralized finance.

The ascent of Bitcoin's price, surging from £333 in January 2016 to an astonishing £15000 by December 2017, captivated global attention. Concurrently, Ethereum, a groundbreaking blockchain platform, gained prominence, facilitating the development of a multitude of decentralized applications.

Today's cryptocurrency landscape is vast and dynamic, encompassing over 18,000 digital assets with a combined market capitalization exceeding £1.55 trillion. Daily trading volume exceeds £55 billion, underscoring the growing significance of cryptocurrencies in global finance.

Despite volatility, the cryptocurrency market remains resilient, attracting sophisticated investors seeking diversification and high returns. As the sector matures, the need for robust price prediction models becomes imperative. Leveraging deep learning algorithms presents a promising avenue for accurately forecasting cryptocurrency prices, offering insights crucial for informed investment decisions.

Outlined in the subsequent sections, this project delineates its objectives, reviews existing literature on cryptocurrency price prediction, elucidates fundamental machine learning concepts, outlines software requirements, details research methodology and data collection strategies, and proposes a model design and implementation. Results, discussions, and conclusions follow, paving the way for advancements in cryptocurrency forecasting and investment strategies.

## 1.1 Aim

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 the necessity for an electronic payment system grounded in "cryptographic proof" that facilitated direct transactions between two parties, laying the foundation for cryptocurrencies as outlined in his seminal white paper. A cryptocurrency operates as a decentralized online currency, leveraging blockchain technology—a decentralized system that securely records and maintains all cryptocurrency transactions.

Given the relative novelty of cryptocurrency technology, only a limited number of price prediction models exist. Greaves and Au (2015) proposed an approach employing Logistic Regression and Support Vector Machines (SVM) to forecast bitcoin prices, achieving an accuracy of 55%. Conversely, Shah and Zhang (2015) employed Bayesian Regression on time series data, claiming potential earnings doubling in 50 days, though the volatile nature of cryptocurrency prices renders this approach less relevant over time.

Madan, Saluja, and Zhao (2015) contributed by generating time series datasets at varying intervals and applying Generalized Linear Models and the Random Forest algorithm to predict Bitcoin prices. Their ensemble of models achieved a 55% accuracy in predicting prices over 10 minutes and an impressive 98.7% accuracy in daily price predictions.

Recognizing the chaotic and unpredictable nature of cryptocurrency markets, Sin and Wang (2017) proposed a Selective Neural Network Ensemble based on Genetic Algorithm to predict bitcoin prices. The ensemble, comprising Multi-Layered Perceptron (MLP) models, demonstrated 58% to 63% accuracy in forecasting the next day's prices.

Drawing parallels between stock and cryptocurrency markets, Selvin et al. (2017) applied Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN) techniques to predict stock prices on the Indian National Stock Exchange (NSE), suggesting the applicability of such models to cryptocurrency price forecasting.

Saad and Mohaisen (2018) discussed a multiple regression approach leveraging highly correlated features like hash rate, transaction rate, and price to predict cryptocurrency prices, while Pant et al. (2019) explored sentiment analysis of Twitter data alongside historical data, employing recurrent neural networks to achieve a 77.82% accuracy in bitcoin price prediction, highlighting the potential of neural networks in this domain.

Finally, Ayaz et al. (2020) advocated for traditional Auto Regressive Integrative Moving Average (ARIMA) models for bitcoin price prediction, while Kumar and Rath (2020) experimented with various deep learning techniques, such as Multi-Layered Perceptron (MLP) and LSTM, to forecast Ethereum prices, albeit with mixed results in terms of accuracy and complexity.

# Deep Learning and Artificial Neural Networks

Deep Learning and Artificial Neural Networks (ANNs) represent a paradigm shift in machine learning (ML), offering powerful tools to tackle nonlinear and complex problems by leveraging vast amounts of data to construct efficient prediction models. Unlike traditional programming approaches, where computers are explicitly programmed, ML educates computers to learn from data, improving and refining their performance over time. ML algorithms are trained on massive datasets to identify patterns, enabling them to make better decisions and predictions. The training dataset, used to instruct the algorithm, is distinct from the testing dataset, which evaluates the model's performance.

At the forefront of deep learning, neural networks seek to emulate the workings of human and animal brains. They consist of interconnected neurons arranged in layers, each neuron producing an output based on weighted inputs. Typically, neural networks comprise input, output, and hidden layers, with the number of neurons varying based on the application's requirements. Inputs are received by the input layer from the external environment, while outputs are generated by the output layer. Hidden layers process and learn from the input data, facilitating complex computations and feature extraction. The connections between neurons are defined by weights, which adjust during the training process to optimize the network's performance.

Artificial neural networks (ANNs) are widely deployed across industries, including engineering, finance, banking, and healthcare, owing to their ability to perform tasks such as categorization, pattern recognition, and prediction with high accuracy. In the finance sector, where markets exhibit nonlinear and volatile behavior, ANNs are increasingly utilized for forecasting and decision-making tasks.

Within the realm of forecasting, recurrent neural networks (RNNs) hold particular significance. RNNs are a specialized type of artificial neural network designed to process sequential or time-series data, making them well-suited for analyzing financial data characterized by temporal dependencies and trends. RNNs employ feedback loops to incorporate information from previous time steps, enabling them to capture temporal dynamics and make predictions based on historical data patterns.

In this study, the focus will be on leveraging recurrent neural networks (RNNs) to analyze and forecast financial data, harnessing their capability to handle sequential data and capture complex temporal relationships inherent in financial markets. By employing RNNs, this study aims to enhance forecasting accuracy and provide valuable insights for decision-making in the finance sector.

# Requirements of the Project

This project has been meticulously crafted using Python within the PyCharm 2021 Integrated Development Environment (IDE), leveraging the language's versatility and the IDE's robust features for efficient development. Python, renowned for its simplicity and readability, serves as the cornerstone for modeling and solving machine-learning problems, offering a rich ecosystem of open-source libraries that streamline complex tasks.

1. **Numpy**: Numpy emerges as a foundational component, facilitating various mathematical computations within Python. Its sophisticated data structures accelerate calculations, providing a plethora of intricate mathematical functions applicable to arrays, lists, and matrices. Numpy's efficiency is instrumental in handling large-scale numerical computations with precision and speed.

2. **Pandas**:Pandas stands as a stalwart Python library for comprehensive data analysis and manipulation, building upon the foundation laid by Numpy. Tailored for tabular data operations, Pandas furnishes a suite of classes optimized for reading, writing, and processing .csv files, empowering developers to manipulate and analyze datasets seamlessly.

3. **Matplotlib:** Matplotlib assumes the role of a versatile plotting library, empowering developers to visualize data through static graphs and interactive visualizations. Paired with Numpy, Matplotlib emerges as a potent open-source alternative to MATLAB, offering extensive customization options and enabling the creation of insightful visualizations, including those depicting predicted versus actual cryptocurrency prices.

**4. Pandas-datareader:** Pandas-datareader emerges as a valuable Python package, facilitating the creation of Pandas DataFrame objects sourced from diverse online repositories. Leveraging APIs such as Yahoo Finance, Pandas-datareader enables seamless integration of financial data into the project, enriching analyses and insights.

**5. TensorFlow:** TensorFlow, developed by Google, stands as a preeminent library for machine learning, particularly deep learning. Renowned for its versatile architecture compatible with CPUs, GPUs, and TPUs (TensorFlow Processing Units), TensorFlow serves as a robust framework for implementing and training neural networks. Within this project, TensorFlow assumes a pivotal role in modeling and training intricate neural network architectures.

**6. Keras:** Keras, a high-level neural network library built atop TensorFlow, offers developers a user-friendly interface for constructing and deploying neural networks. Boasting compatibility with both CPU and GPU environments, Keras provides a rich assortment of classes for implementing activation functions, layers, and model architectures. Its support for convolutional neural networks (CNNs) and recurrent neural networks (RNNs) empowers developers to tackle diverse machine-learning tasks with ease and efficiency.

7. **Scikit-learn:** Scikit-learn emerges as a versatile machine-learning package written in Python, catering to a myriad of data analysis tasks. Featuring an extensive array of classes and algorithms, Scikit-learn facilitates the resolution of classification, regression, and clustering problems, providing developers with powerful tools for data exploration and modeling.

Collectively, these Python libraries form the backbone of the project, enabling developers to harness the full potential of machine learning and deep learning techniques in analyzing and forecasting cryptocurrency prices with precision and efficacy.

# Methodology

This project harnesses the Long Short-Term Memory (LSTM) architecture, a specialized type of recurrent neural network (RNN) tailored for deep learning tasks. Conceived by Hochreiter & Schmidhuber (1997), LSTM represents a groundbreaking advancement in neural network design, offering a solution to the vanishing gradient problem that plagued traditional RNNs. Unlike conventional RNNs, LSTM introduces a sophisticated mechanism for managing memory over extended time intervals, thereby enabling more effective learning and retention of temporal dependencies.

At its core, an RNN processes sequential data by iteratively updating its internal state based on the current input and previous output (feedback). However, traditional RNNs struggle to retain long-term dependencies due to the vanishing gradient problem, wherein updates to network weights diminish exponentially over time, hindering learning. LSTM addresses this challenge by incorporating a set of specialized gates—namely, input, forget, and output gates—that regulate the flow of information through the network, facilitating the preservation of long-term memory across multiple time steps.

The key innovation of LSTM lies in its ability to maintain two types of memory: short-term memory, which captures immediate dependencies within the sequence, and long-term memory, which stores essential information over extended intervals. By selectively updating and preserving memory through the gating mechanism, LSTM mitigates the vanishing gradient problem and enables more robust learning across diverse temporal contexts.

In the context of this project, the LSTM architecture is employed to model and forecast cryptocurrency prices, leveraging its ability to capture intricate temporal patterns and dependencies inherent in financial data. The proposed model comprises three LSTM layers, each equipped with dropout layers to prevent overfitting, and a final Dense Layer for output generation. These architectural components are meticulously configured and optimized to strike a balance between model complexity and predictive accuracy, as discussed in detail in section 5.3 of the project documentation.

By harnessing the power of LSTM networks, this project endeavors to unlock deeper insights into cryptocurrency price dynamics, enabling more accurate forecasting and informed decision-making within the volatile landscape of financial markets. Through rigorous experimentation and fine-tuning of model parameters, the aim is to develop a robust predictive framework capable of adapting to evolving market conditions and delivering actionable insights for stakeholders in the finance sector.

## 4.1 Proposed Method

Figure 1 illustrates the systematic approach of the proposed prediction method. The process begins with the collection and pre-processing of historical data, which serves as the foundational dataset for model training and evaluation. Subsequently, the pre-processed data is partitioned into distinct sets for training and testing purposes, enabling the assessment of the model's performance on unseen data The prediction model undergoes rigorous training using the designated training dataset, wherein it learns to discern patterns and relationships within the input data. Leveraging techniques such as backpropagation and gradient descent, the model iteratively adjusts its parameters to minimize prediction errors and optimize performance.

Following training, the model is subjected to evaluation using the test dataset, allowing for an objective assessment of its predictive capabilities on unseen data. Through this evaluation process, the model's generalization ability and robustness to new inputs are scrutinized, providing insights into its performance under real-world conditions.

Finally, the processed data is fed into the trained model to obtain prediction results, which offer valuable insights into future trends or outcomes based on historical patterns. These predictions serve as actionable intelligence for stakeholders, guiding decision-making processes and informing strategic initiatives within the domain of cryptocurrency trading and investment.

Diagram

Description automatically generated

**Figure 1: Proposed Method**

## 4.2 Data Collection

The data collection process utilizes the Yahoo Finance API in conjunction with the pandas-datareader library for Python, enabling the retrieval of cryptocurrency data as DataFrame objects. Specifically, data is gathered spanning from January 1st, 2016, up to the present date, encompassing four prominent cryptocurrencies: Bitcoin, Ethereum, Ripple, and Monero.

Given the temporal nature of the analysis, this research leverages time-series data to model and forecast cryptocurrency prices accurately. To facilitate rigorous evaluation and validation of the prediction model, a subset of the collected data spanning from January 1st, 2021, to the current date is earmarked as the testing dataset. This partitioning ensures that the model is evaluated on unseen data, thereby assessing its ability to generalize to new market conditions

Figure 2 provides a visual representation of a sample dataset obtained through the Yahoo Finance API, focusing specifically on the cryptocurrency Bitcoin. This sample data serves as a snapshot of the comprehensive dataset utilized for predictive modeling, showcasing key attributes such as date, opening price, closing price, high and low prices, volume, and adjusted close. Through detailed analysis and processing of such data, predictive models can be trained and evaluated to generate insightful forecasts and predictions regarding future cryptocurrency price movements.

Overall, the utilization of the Yahoo Finance API and pandas-data reader library facilitates the acquisition of reliable and up-to-date cryptocurrency data, enabling robust analysis and modelling of price trends. By leveraging time-series data and delineating distinct testing datasets, this research ensures the development of accurate and dependable prediction models capable of informing strategic decisions within the cryptocurrency market landscape.

**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 process initiates with the acquisition of cryptocurrency data from the Yahoo Finance API, spanning from a designated start date to an end date. To prepare this data for subsequent analysis and modeling, a crucial step involves standardizing its scale. To accomplish this, a MinMaxScaler is instantiated.

The MinMaxScaler is a preprocessing technique that transforms features by scaling them to a specified range, typically between 0 and 1. This normalization process ensures that all features are on a consistent scale, mitigating potential issues related to varying magnitudes across different features.

Subsequently, the collected data undergoes scaling using the MinMaxScaler, achieved through the fit\_transform method. This method computes the minimum and maximum values of the data and then scales each feature accordingly to fit within the designated range (0 to 1). By applying this scaling transformation, the data is homogenized, facilitating more effective analysis and modeling.

Following scaling, the training dataset is prepared by selecting a window of historical data. In this context, the training data comprises observations from the last 60 days, providing a temporal context for model training. This sliding window approach allows the model to learn from sequential patterns within the data, capturing temporal dependencies and trends that may influence future cryptocurrency prices.

Figures 4 and 5 in the documentation illustrate the Python code implementing these preprocessing steps, delineating the instantiation of the MinMaxScaler, the application of scaling using fit\_transform, and the preparation of training data using the sliding window approach. Through these meticulous preprocessing steps, the collected cryptocurrency data is refined and structured, laying the foundation for subsequent modeling and prediction tasks with enhanced accuracy and reliability.

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 architecture is constructed using the Sequential model from the Keras library, comprising linearly stacked layers. The model incorporates several crucial layers, including LSTM (Long Short-Term Memory), Dropout, and Dense layers, each serving distinct purposes in the model's design. Specifically, the proposed architecture integrates three LSTM layers, three dropout layers, and one Dense layer, meticulously configured to optimize predictive performance (Brown & Harris, 1994).

**LSTM (Long Short-Term Memory) Layers**: LSTM layers are pivotal components of the neural network, designed to capture and retain important temporal dependencies within the input data. Unlike traditional recurrent neural networks (RNNs), LSTM layers excel in memorizing crucial information over extended time intervals, mitigating issues such as vanishing gradients and facilitating more effective learning and prediction. These layers function by selectively retaining and propagating relevant information through time, thereby enhancing the model's ability to capture long-term dependencies and patterns (Hochreiter & Schmidhuber, 1997).

**Dropout Layers**: Dropout layers play a crucial role in preventing overfitting within the neural network. Overfitting occurs when the model learns to memorize noise or idiosyncrasies present in the training data, leading to poor generalization on unseen data. Dropout layers mitigate this phenomenon by randomly deactivating a fraction of neurons during training, effectively introducing a form of regularization that encourages the network to learn robust and generalizable representations of the data (Srivastava et al., 2014).

**Dense Layer**: The Dense layer serves as the final component of the neural network, responsible for producing the model's prediction output. This layer aggregates and integrates the information learned by the preceding layers, ultimately generating a single value that represents the predicted price of the cryptocurrency. By synthesizing the learned features and patterns, the Dense layer encapsulates the model's predictive capabilities, providing valuable insights into future price trends (LeCun et al., 2015).

The model is subsequently compiled and trained using the configured architecture, leveraging optimization algorithms and loss functions to iteratively refine the model parameters and enhance predictive accuracy (Rumelhart et al., 1986). Figure 6 showcases the Python code implementing this subsection, encapsulating the construction, compilation, and training of the neural network architecture.

In summary, the neural network architecture comprises LSTM, Dropout, and Dense layers, each contributing to the model's ability to capture temporal dependencies, prevent overfitting, and generate accurate predictions of cryptocurrency prices. Through meticulous design and training, the model emerges as a powerful tool for forecasting and decision-making within the dynamic landscape of cryptocurrency markets.

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

In this section, the Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) metrics are computed to evaluate the performance of the proposed model. These metrics provide insights into the accuracy and precision of the model's predictions, shedding light on its efficacy in forecasting cryptocurrency prices.

The MAE represents the average absolute difference between the actual and predicted values, providing a measure of the model's average prediction error. Similarly, the MSE quantifies the average squared difference between actual and predicted values, offering a more sensitive measure to outliers. Finally, the RMSE, derived from the MSE, represents the square root of the average squared difference, providing a measure of the average magnitude of the prediction errors.

These metrics, along with their corresponding formulae, are elaborated upon in detail in section 6.2 of the research document, offering a comprehensive understanding of their significance in evaluating model performance.

Following the computation of these metrics, the actual and predicted prices are visualized using the matplotlib library. The actual prices are depicted in black, while the predicted prices are represented in green. This graphical representation facilitates a qualitative assessment of the model's predictive accuracy, allowing for visual comparison between actual and predicted price trends.

Figures 9 and 10 present the Python code responsible for generating these visualizations, showcasing the seamless integration of matplotlib for data visualization purposes. Through the juxtaposition of actual and predicted price trajectories, stakeholders gain valuable insights into the model's performance and its ability to capture and forecast cryptocurrency price dynamics effectively.

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 demonstrates remarkable success in forecasting the prices of Bitcoin, Ethereum, Monero, and Ripple, as evidenced by the comparison between actual and predicted prices depicted in Figures 11 to 14. In these visualizations, the black curve represents the actual price, while the green curve represents the predicted price. While the model achieves 100% accuracy on certain days, such instances are limited due to the inherent uncertainty in cryptocurrency markets. Despite this limitation, the model adeptly captures trends in cryptocurrency prices, aligning with the upward or downward trajectory of actual prices over time.

Despite the existence of numerous theories and algorithms for cryptocurrency price prediction, many encounter challenges such as overfitting and inaccuracies stemming from extensive datasets. However, the proposed model addresses these shortcomings through the utilization of multiple LSTM and dropout layers. Table 1 provides a comprehensive overview of the model's performance, showcasing metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and normalized RMSE values for each cryptocurrency. Notably, the model achieves the highest accuracy for Ethereum (ETH) with a normalized RMSE of 0.0564, while Ripple (XRP) exhibits the lowest accuracy with a normalized RMSE of 0.1241. Bitcoin (BTC) and Monero (XMR) prices are predicted fairly accurately, with normalized RMSE values of 0.0757 and 0.0675, respectively.

While the proposed model serves as a valuable tool for studying cryptocurrency price trends, it should not be solely relied upon for investment decisions due to the inherent volatility and unpredictability of cryptocurrency markets. However, it can offer valuable insights when used in conjunction with other deep learning models such as ARIMA or GASEN, or integrated with sentiment analysis from social media platforms like Twitter to mitigate uncertainty in price predictions.

In future iterations, the model's performance can be further enhanced by exploring the effects of varying LSTM layers, neuron counts, batch sizes, and optimizers on prediction accuracy. Additionally, the development of an application that aggregates predictions from multiple models and provides investment advice could prove beneficial to cryptocurrency investors.

The findings from Table 2 suggest that the model achieves optimal performance with a dropout rate of 0.2, exhibiting the lowest RMSE values. Furthermore, the results from Table 3 indicate that while the model achieves the lowest normalized RMSE value with 75 epochs, a value of 25 epochs offers a competitive performance with a normalized RMSE of 0.0564. Given that training time increases with the number of epochs, a balance must be struck between computational efficiency and prediction accuracy. Therefore, employing 25 epochs presents a practical compromise, delivering satisfactory performance while minimizing computational overhead.

# Conclusion

The proposed model has demonstrated notable success in accurately predicting the prices of Bitcoin, Ethereum, Monero, and Ripple. Particularly noteworthy is the high accuracy achieved for Ethereum (ETH), with a normalized RMSE of 0.0564, indicating precise forecasting capabilities. By optimizing key parameters such as dropout rate and number of epochs, the model's performance was further enhanced, particularly with a dropout rate of 0.2 and 25 epochs.

However, it is essential to exercise caution when relying solely on the model for investment decisions. While the model effectively captures price fluctuations, the inherently volatile nature of cryptocurrency markets introduces inherent risks. Therefore, investors should supplement model predictions with additional research and analysis to mitigate potential losses.

Nevertheless, the primary objectives of this research have been successfully accomplished. The model effectively predicts prices, evaluates performance metrics, and identifies optimal parameters for enhanced prediction accuracy. Despite the model's limitations in providing investment advice, it serves as a valuable tool for understanding and analyzing cryptocurrency price trends.

Looking ahead, there are opportunities for further refinement and improvement of the proposed model. Integration with other deep learning models or sentiment analysis from platforms like Twitter could yield enhanced predictive capabilities. As peer-to-peer transactions and cryptocurrency technologies continue to evolve, advancements in modeling techniques and predictive methodologies are anticipated, paving the way for more accurate and reliable cryptocurrency price predictions.

In summary, while the proposed model offers valuable insights into cryptocurrency price dynamics, it should be utilized cautiously and supplemented with additional research. Continued research and innovation in the field hold promise for the development of more robust and sophisticated models capable of navigating the complexities of cryptocurrency markets with greater precision and reliability.

# 9.References

* Ayaz, Z., Fiaidhi, J., Sabah, A., & 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. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/
* Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? Geoscientific Model Development Discussions, 7(1), 1525–1534.
* Greaves, A., & Au, B. (2015). Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin. Retrieved from http://snap.stanford.edu/class/cs224w-2015/projects\_2015/Using\_the\_Bitcoin\_Transaction\_Graph\_to\_sPredict\_the\_Price\_of\_Bitcoin.pdf
* IBM Cloud Education. (2020). What are Neural Networks? Retrieved from https://www.ibm.com/uk-en/cloud/learn/neural-networks
* Jones, E. (2022). A Brief History of Cryptocurrency. CryptoVantage. Retrieved from https://www.cryptovantage.com/guides/a-brief-history-of-cryptocurrency/
* Kumar, D., & Rath, S. K. (2020). Predicting the Trends of Price for Ethereum Using Deep Learning Techniques. Advances in Intelligent Systems and Computing, 103–114.
* Madan, I., Saluja, S., & Zhao, A. (2015). Automated Bitcoin Trading via Machine Learning Algorithm. Retrieved from http://cs229.stanford.edu/proj2014/Isaac%20Madan,%20Shaurya%20Saluja,%20Aojia%20Zhao,Automated%20Bitcoin%20Trading%20via%20Machine%20Learning%20Algorithms.pdf
* Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. SSRN Electronic Journal.
* Pant, D. R., Neupane, P., Poudel, A., Pokhrel, A. K., & 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., & Sriramya, D. P. (2015). A Research On Bitcoin Price Prediction Using Machine Learning Algorithms. International Journal of Scientific & Technology Research, 9(4), 1600–1604. Retrieved from https://www.ijstr.org/paper-references.php?ref=IJSTR-0420-33830
* Saad, M., & 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., & 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., & Wang, L. (2017). Bitcoin Price Prediction Using Ensembles of Neural Networks. Retrieved from https://personal.ntu.edu.sg/elpwang/PDF\_web/08393351.pdf

# 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()