Evaluating and Comparing the Predictability of Cryptocurrency Prices from Historical Price Data using different Deep Learning methods

FINAL YEAR PROJECT

John Kennedy

6575452

# Abstract:

Cryptocurrencies have been becoming increasingly prevalent in the modern day and have become a very appealing medium for investors due to their high volatility and the massive returns possible as a result. However, they are also seen as very risky investments for the same reason.

This project aims to study the predictability of cryptocurrencies such as Bitcoin or Ether through Machine Learning (ML) methods. It does this by comparing the effectiveness of different models and neural networks in forecasting prices based on historical price data; Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRU) are used to perform analysis on historical information, predicting the price of the currencies and compares these to evaluate performance, in order to aid in financial forecasting research in relation to cryptocurrencies as well as those who are interested in short-term investment in the crypto exchange market.

Different datasets are also used and compared to see how this affects predictability, specifically between hourly and daily price datasets. Certain data pre-processing techniques and normalisation will be applied to improve results achieved, while data smoothing techniques used in financial forecasting shall also be used to see if they improve results.

Machine Learning is well-suited to this task as it is capable of analysing the time-series data as a sequence and finding high-dimensional relationships between inputs to make predictions.

The final models generated are further tested against historical data for accuracy and performance metrics and evaluated against each other.

# Declaration of Originality

“*I confirm that the submitted work is my own work. No element has been previously submitted for assessment, or where it has, it has been correctly referenced. I have clearly identified and fully acknowledged all material that is entitled to be attributed to others (whether published or unpublished) using the referencing system set out in the programme handbook.*  
  
*I agree that the University may submit my work to means of checking this, such as the plagiarism detection service Turnitin® UK and the Turnitin® Authorship Investigate service. I confirm that I understand that assessed work that has been shown to have been plagiarised will be penalised.*"

# Acknowledgements

I would like to thank the University of Surrey for supporting me throughout this project and giving me the opportunity to perform this study.

Additionally, several key people have been key in helping me throughout my journey and sparking my interest in this subject, so special thanks must go out to them, my friends and family.

Table of Contents

[Abstract: 1](#_Toc113536373)

[Declaration of Originality 2](#_Toc113536374)

[Acknowledgements 3](#_Toc113536375)

[Table of Figures 7](#_Toc113536376)

[1. Introduction 8](#_Toc113536377)

[1.1 Problem Background 8](#_Toc113536378)

[1.2 Project Description 10](#_Toc113536379)

[1.3 Project Aims 11](#_Toc113536380)

[1.4 Report Structure 12](#_Toc113536381)

[2. Literature Review 14](#_Toc113536382)

[2.1 Cryptocurrency 14](#_Toc113536383)

[2.1.1 Bitcoin and the Fundamentals of Cryptocurrency 14](#_Toc113536384)

[2.1.2 Altcoins 16](#_Toc113536385)

[2.1.3 Trading 17](#_Toc113536386)

[2.2 Financial Forecasting 19](#_Toc113536387)

[2.2.1 Stock Price Prediction 19](#_Toc113536388)

[2.2.2 Forecasting Techniques – Technical Analysis 20](#_Toc113536389)

[2.3 Time Series 23](#_Toc113536390)

[2.4 Machine Learning 24](#_Toc113536391)

[2.4.1 Fundamentals 24](#_Toc113536392)

[2.4.2 Functions & Hyperparameters 25](#_Toc113536393)

[2.4.3 Neural Networks 30](#_Toc113536394)

[2.4.4 Deep Learning 30](#_Toc113536395)

[2.4.5 Recurrent Neural Networks (RNNs) 31](#_Toc113536396)

[2.4.6 Long Short-Term Memory (LSTM) 31](#_Toc113536397)

[2.4.7 Convolutional Neural Networks (CNNs) 32](#_Toc113536398)

[2.4.8 Gated Recurrent Unit (GRU) 33](#_Toc113536399)

[2.4.10 Regularisation - Dropout 34](#_Toc113536400)

[2.5 Related Studies 35](#_Toc113536401)

[3. Choice of Experiments 37](#_Toc113536402)

[4. Data Overview 38](#_Toc113536403)

[5. Data Pre-processing 40](#_Toc113536404)

[5.1 Data Normalisation 40](#_Toc113536405)

[5.2 Sequence Extraction 40](#_Toc113536406)

[5.3 Financial Forecasting Algorithms 41](#_Toc113536407)

[5.4 Dataset Split 42](#_Toc113536408)

[6. Neural Networks 43](#_Toc113536409)

[Neural Networks to be implemented 43](#_Toc113536410)

[Architecture 44](#_Toc113536411)

[7. Methodology 47](#_Toc113536412)

[7.1 Data Pre-processing 47](#_Toc113536413)

[7.1.1 Normalisation 47](#_Toc113536414)

[7.1.2 Dataset Split 48](#_Toc113536415)

[7.1.3 Sequence Extraction 49](#_Toc113536416)

[7.2 Evaluation Methods 49](#_Toc113536417)

[7.3 Experimentation with Dataset Size 50](#_Toc113536418)

[7.4 Implementing Data Smoothing 50](#_Toc113536419)

[8. Analysis and Evaluation of Results (Main Results in Appendix) 51](#_Toc113536420)

[9. Conclusions and Future Work 55](#_Toc113536421)

[10. Evaluation 56](#_Toc113536422)

[11. Statement of Ethics 57](#_Toc113536423)

[11.1 Overview 57](#_Toc113536424)

[11.2 Informed Consent 57](#_Toc113536425)

[11.3 Copyright & Intellectual Property 57](#_Toc113536426)

[11.4 Confidentiality of Data 58](#_Toc113536427)

[11.5 Social Responsibility 58](#_Toc113536428)

[11.6 BCS Codes of Conduct and Ethics 58](#_Toc113536429)

[References 59](#_Toc113536430)

[Libraries 64](#_Toc113536431)

[Tensorflow: https://www.tensorflow.org/ 64](#_Toc113536432)

[Keras: https://keras.io/ 64](#_Toc113536433)

[Numpy: https://numpy.org/ 64](#_Toc113536434)

[Pandas: https://pandas.pydata.org/ 64](#_Toc113536435)

[Requests: 64](#_Toc113536436)

[JSON: 65](#_Toc113536437)

[Matplotlib PyPlot: https://matplotlib.org/ 65](#_Toc113536438)

[Scikit-Learn: https://scikit-learn.org/stable/ 65](#_Toc113536439)

[Appendix 66](#_Toc113536440)

[Sage Ethics Form 66](#_Toc113536441)

[Daily Training Results – 6 months 79](#_Toc113536442)

[Daily Training Results – 1 year 80](#_Toc113536443)

[Daily Training Results – 3 years 81](#_Toc113536444)

[Daily Training Results – 5 years 82](#_Toc113536445)

[Hourly Training Results – 500 hours 83](#_Toc113536446)

[Hourly Training Results – 1000 hours 84](#_Toc113536447)

[Hourly Training Results – 1500 hours 85](#_Toc113536448)

[Hourly Training Results – 2000 hours 86](#_Toc113536449)

[Data Smoothing Experimentation Results 87](#_Toc113536450)

# Table of Figures

[Figure 1 - Diagram of Bitcoin transaction verification; Satoshi Nakamoto, Bitcoin: A peer-to-peer electronic cash system. The public key for the transaction is used to verify the transaction by miners, but only the Owner’s Private Key can be used to sign and authenticate the transaction or decrypt the transaction to reveal non-public information. 15](#_Toc113536451)

[Figure 2 - Largest Cryptocurrencies Market Share 2015/2019; Springer Nature, Open Economies Review 16](#_Toc113536452)

[Figure 3 - Daily Revenue of the largest crypto exchanges, calculated with trading volume data from CoinMarketCap.com; Economic Times, Cryptocurrency exchanges are raking in billions of dollars 17](#_Toc113536453)

[Figure 4 - ADX Formulae; Investopedia, Average Directional Index (ADX) 20](#_Toc113536454)

[Figure 5 - RSI Formula 1; Investopedia, Relative Strength Index (RSI) Indicator Explained With Formula 21](#_Toc113536455)

[Figure 6 - RSI Formula 2; Investopedia, Relative Strength Index (RSI) Indication Explained With Formula 21](#_Toc113536456)

[Figure 7 - MA Formula; Investopedia, Moving Average (MA): Purpose, Uses, and Examples 22](#_Toc113536457)

[Figure 8 - EMA Formula; Investopedia, Moving Average (MA): Purpose, Uses, and Examples 22](#_Toc113536458)

[Figure 9 - Artificial Neuron against biological inspiration; R Pramoditha, The Concept of Artificial Neurons (Perceptrons) in Neural Networks, 2021 24](#_Toc113536459)

[Figure 10 - Underfitting & Overfitting Examples; IBM, Overfitting 28](#_Toc113536460)

[Figure 11 - Overfitting and Iterations; IBM, Overfitting 28](https://d.docs.live.net/967ef39f97508245/Desktop/FYP/Report/Project%20Report.docx#_Toc113536461)

[Figure 12 - Overshooting due to high learning rate; Google Developers, Reducing Loss: Learning Rate 29](#_Toc113536462)

[Figure 13 - Diagram of Neurons in a Neural Network; IBM, What are Neural Networks? 30](#_Toc113536463)

[Figure 14 - Recurrent (left) vs Feed-Forward NNs (right); IBM, Recurrent Neural Networks 31](#_Toc113536464)

[Figure 15 - Example of CNN Architecture; S. Saha 2018, A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way 32](#_Toc113536465)

[Figure 16 – Update/Reset Gate Formula; S. Kostadinov 2017, Understanding GRU Networks 33](#_Toc113536466)

[Figure 17 - Diagram of Dropout Implemented on a Neural Network; N. Srivastava 2014, Dropout: A Simple Way to Prevent Neural Networks from Overfitting 34](#_Toc113536467)

[Figure 18 - Historical Price Data for BTC – Last 5 Years, from the dataset 38](#_Toc113536468)

[Figure 19 - Historical BTC Price Data from the dataset - 1 Year 39](#_Toc113536469)

[Figure 20 - Range Scaling Function; Google Developers, Machine Learning - Normalization 40](#_Toc113536470)

[Figure 21 - Simple Moving Average Equation; CIN7, Moving Average 41](#_Toc113536471)

[Figure 22 - Exponential Moving Average Equation; Trading View, Exponential Moving Average 41](#_Toc113536472)

[Figure 23 - Base RNN Architecture 44](#_Toc113536473)

[Figure 24 - Deeper RNN Architecture 45](#_Toc113536474)

[Figure 25 - CNN-RNN Hybrid Architecture 46](#_Toc113536475)

[Figure 26 - Normalisation Functions from code 47](#_Toc113536476)

[Figure 27 (Repeated) - Historical Price Data for BTC, from the dataset, including the dataset split 48](#_Toc113536477)

[Figure 28 - Sequence Extraction Code 49](#_Toc113536478)

[Figure 29 - Data Quantity/Best Winrate Achieved Graphs 51](#_Toc113536479)

[Figure 30 - Example of Loss Graph for Overtrained Model 52](#_Toc113536480)

[Figure 31 - Daily Training Best Models 52](#_Toc113536481)

[Figure 32 - Hourly Training Winners 53](#_Toc113536482)

# 1. Introduction

## 1.1 Problem Background

After the financial crisis of 2008, the resulting loss of trust in the global financial establishment led to the appearance and popularization of Bitcoin, the first decentralized peer-to-peer payment system (Vasilis Kostakis, 2014). Independent of any government or individual, these currencies have attracted the attention of the general public, as well as investors, academics and politicians. The cryptocurrency market today has grown to become one of the world’s biggest unregulated markets.

Since the rise of Bitcoin as the first cryptocurrency, many other “altcoins” have entered the spotlight, such as Ethereum, Tether and Polkadot. Since the early days of obscurity and mistrust, crypto has become increasingly common as a medium for investment and the currencies themselves are being accepted as forms of payment by more and more companies. With 15,000 businesses worldwide now accepting cryptocurrency payments (Flynn, 2022), the shift towards this new medium is undeniable.

Bitcoin was originally intended to be an alternative form of electronic payment and was successful in this regard at first due to the advantages afforded by the decentralization made possible by the blockchain. The blockchain’s presence, recording all transactions in a public, transparent, quick and secure manner allowed Bitcoin to thrive. The highly divisible nature of cryptocurrency also allows for transactions to be very small and/or very exact as required, while the pseudonymous nature of these transactions allows people to remain anonymous while also keeping track of transactions by referring only to wallet addresses rather than their owners.

Despite this, the rapid rise in the value of Bitcoin attracted the attention of investors as well as the popular imagination, prompting more people to treat cryptocurrencies as speculative assets rather than an actual currency as the high volatility of the medium made it possible to make huge returns very quickly (Dirk G. Baur, 2018).

In the US alone, roughly 46 million people (22% of the adult population) own a share of Bitcoin (Howarth, 2022). This shows the massive relevance of crypto as a platform for investment, in large part due to its high volatility despite the fact that the supply of bitcoin has a 21 million unit cap (A. Hayes, 2022), since this limit has not yet been reached. This is also the reason that many people (up to 97% of day traders) lose money trading cryptocurrencies, and why many refuse to invest in it.

Ether (Ethereum/ETH), is the next biggest cryptocurrency, and does not have a cap on the number of possible units. Ethereum is distinct from BTC as Bitcoin was intended to provide a new form of decentralised digital currency for payments, while Ethereum uses the blockchain in another way, to allow the formation of contracts and applications that are tamper-proof and public. Ether was intended to be used to compliment Bitcoin, aiding in transactions and business dealings while building on Bitcoin as the first cryptocurrency, however as these have become used more and more as speculative investable assets, Ether has come to be a major competitor in the crypto exchange market.

Another major example of an altcoin is Litecoin (LTC), which was based on Bitcoin and uses the same protocol, however it was designed to be significantly faster and more efficient, by speeding up transactions. In addition, it seeks to address issues with mining additional coins, in that it takes a huge amount of processing power, time, and energy to improve overall processing speed on the blockchain for mining as well as transactions. This increased speed makes it highly appealing for both business transactions that are time-sensitive, and miners who wish to earn Litecoin, the increased number of whom contributes to increasing the speed of the validation of transactions on the blockchain. This also allows Litecoin to maintain a higher supply cap of 84 million coins.

Asset pricing based on empirical data is an essential and growing area within the world of finance, and over time the use of Machine Learning (ML) in this field is growing due to its ability to take into account multiple different factors that have complex, interdependent relationships with many dimensions in order to discern the mathematical link between these individual features and the target result.

ML techniques are used for financial forecasting of assets including equities and bonds, however the reliability of these methods can be questionable, particularly when the asset in question is highly volatile, such as in the case of cryptocurrency price prediction.

## 1.2 Project Description

This study attempts to compare and statistically analyse the performance of multiple different machine learning methods in the prediction of cryptocurrency prices, trained on historical data from Bitcoin as well as specific altcoins, the aforementioned Ether and LTC, in order to draw conclusions on the usefulness of these strategies on their own for trading and forecasting within the crypto exchange market. This report hopes to make clear the difference between the different neural networks that can be used, as well as presenting the most efficient methods for each of the studied coins through experimentation and comparison, particularly the training and validation loss values over iterations for each model, as well as the “win-rates” of each model in predicting either a rise or fall in price.

This sort of experimentation is necessary as there is variation in the application of machine learning, where different methods face distinct issues and hold unique advantages when approaching certain tasks, particularly in this case due to the “time-series” nature of the data, although this can be altered with data pre-processing. Examples of this include the “vanishing gradient problem” which is a problem faced by many gradient-based methods which can be avoided by changing the activation function or the type of neural network (such as to an LSTM), and overfitting of larger networks that benefit from Deep Learning or are simply overtrained, which can cause problems when the models are applied to new data that they were not trained on.

These differences in the efficacy of different methods are particularly prominent when comparing different categories of ML neural networks, such as between Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Long Short-Term Memory (LSTM) models, all of which are suited to different tasks and types of data input. All of these will be compared and covered within this study.

In addition, the effects of applying data smoothing to the training data will be examined through thorough experimentation to see and explain the effects that these techniques have on the accuracy of predictions.

The results of models trained on hourly and daily datasets will also be compared to show the difference that the frequency and volume of the data make in the ability of ML methods to form predictions, particularly due to the high volatility of cryptocurrencies.

## 1.3 Project Aims

In this section, the aims and objectives of the study are outlined to make clear the success of the study when these are later reflected upon in the evaluation of the project.

1. **Acquire historical price data for multiple cryptocurrencies**

In order to use ML to generate price predictions, large datasets of historical data will have to be used for training. The first step in this study is to acquire this data, both for hourly and daily prices, to train models to generate results for comparison.

1. **Research techniques used in normal financial price forecasting**

Existing techniques used for financial stock analysis, as well as ML methods that are applicable, particularly in relation to time-series analysis, will be researched as this will be crucial in producing working models that can perform good price forecasting.

1. **Apply data pre-processing techniques and normalisation**

Normalisation and pre-processing of data before use in training can greatly help the models form better predictions by removing unnecessary complexity from the data. These techniques also ensure that the data provided is used for training in the best way to improve the quality of predictions.

1. **Apply and adapt Machine Learning and financial price forecasting techniques to the prediction of cryptocurrency prices**

Techniques such as exponential moving average smoothing, used in financial stock analysis, would be highly applicable to this study and would allow for more accurate predictions. Different neural network types must be selected to provide good comparisons between categories of neural network, and for greatest likelihood of success.

1. **Generate and train models of different types for comparison**

Different models must be trained in order to allow for comparison to find the best possible model for each dataset/cryptocurrency combination.

1. **Find and compare the best-performing models across different cryptocurrencies and datasets**

Information gathered through the training of models and experimentation with different datasets and cryptocurrencies will be compared to find the best configuration for price prediction of each cryptocurrency.

1. **Present and explain the results**

Results must be presented to show the most effective model for each dataset, and the influence of whether the price dataset is hourly/daily must be evaluated, in addition to the effect of applying data smoothing.

## 1.4 Report Structure

**Section 1 – Introduction**

In this section I briefly go over the background of cryptocurrency and how machine learning can be used to predict cryptocurrency prices. A brief description of the project is given, and the project’s aims are outlined.

**Section 2 – Literature Review**

This section details relevant background information on cryptocurrencies and stock price forecasting, in addition to explaining the technologies and theories behind the machine learning methods used.

**Section 3 – Choice of Experiments**

This section briefly goes over the choice of experiments to be performed in this study, based on my review of the relevant literature.

**Section 4 – Data Overview**

Here the structure and content of the datasets used are covered and briefly explained.

**Section 5 – Data Pre-processing**

This section contains information on the pre-processing that I have decided to implement on the data given my previous analysis in order to assist in the training of the model and to ensure that the data is usable.

**Section 6 – Neural Networks**

In this section, I explore the different neural networks that may be used in my models, and I explain my choices regarding the types of neural networks that I compare and their architectures, as well as the advantages and disadvantages of each.

**Section 7 – Methodology**

Each model is trained with the different historical price datasets, and with data from this I can draw preliminary comparisons between the models. In this section, the implementation of these models, and my testing methodology are covered.

**Section 8 – Analysis & Evaluation of Results**

Here I present the results of my experimentation and compare the performances of the various models across databases and cryptocurrencies, allowing me to draw conclusions.

**Section 9 – Final Conclusions & Future Work**

Here I briefly outline the final conclusions I have drawn from the data on the most effective model, the most predictable currency, the application of data smoothing, and the overall viability of machine learning for price prediction based on historical price data, as well as additional work that could be done in the future to improve the quality of the findings and the performance of the models.

**Section 10 – Evaluation**

In this section, I evaluate the project as a whole against its aims and objectives, judging its effectiveness and success, and what has been achieved.

**Section 11 – Statement of Ethics**

This section explores the ethical considerations, implications, and consequences of the project to ensure that my work does not legally or morally violate anyone’s rights, including their privacy, and that the study cannot be used to do harm.

# 2. Literature Review

## 2.1 Cryptocurrency

Cryptocurrencies are a decentralised form of digital currency which arose due to widespread mistrust in the established banking and finance industries after the 2008 financial crash (Vasilis Kostakis, 2014). They are known as cryptocurrencies since they are protected cryptographically, allowing for secure and pseudonymous transactions (reference is never made to the owner of a crypto “wallet”, but rather simply the address of the wallet allowing people to remain mostly anonymous) (N Alsalami, 2019).

The decentralisation of these currencies is permitted by use of a public ledger, the “blockchain”. This allows the currencies to be completely independent of third parties such as banks, while being secure and transparent as all transactions can be publicly viewed (A. Hayes, 2022).

Despite the original intention of these currencies to be a digital version of normal currency, their prices are incredibly volatile and highly subject to the market forces of supply and demand. This has resulted in, and is also in turn caused by, the public view of cryptocurrencies as speculative assets for investing, particularly as they have no intrinsic value in and of themselves (Dirk G. Baur, 2018).

The following sections will outline and cover the principles and history behind Bitcoin, the first and most popular cryptocurrency, as well as some of the “Altcoins” that have followed it. The unpredictability of the market and important factors to be considered when attempting to predict changes in prices will also be explored. The Machine Learning techniques to be used and compared in the study are then detailed, as well as how then can be evaluated and compared. At the end, related studies and work are presented.

### 2.1.1 Bitcoin and the Fundamentals of Cryptocurrency

Cryptocurrencies can vary quite widely in their characteristics, however a large proportion of them share many similarities as most are based off Bitcoin, the first successful cryptocurrency.

Proposed in 2008, the creator of Bitcoin, known by the alias Satoshi Nakamoto, brought Bitcoin online in 2009 (Velde, 2013) with the publication of its first open-source client. Bitcoin revolutionised digital currency in certain key ways - by use of the blockchain, Nakamoto protected Bitcoin from double-spending attacks, to which previous digitally distributed currencies were vulnerable, allowing someone to exploit the system by spending their money multiple times (G. Karame, 2012). This was avoided through the achievement of distributed consensus to secure the blockchain, with mechanisms such as “Proof-of-Work”, to verify that the transactions verified and authenticated on the blockchain are authentic and have not been tampered with by use of hash functions for validation (A. Hayes, 2022).

This subsection goes over the fundamentals and principles behind the basic functionality of Bitcoin as outlined by Nakamoto in his 2008 “white paper”.

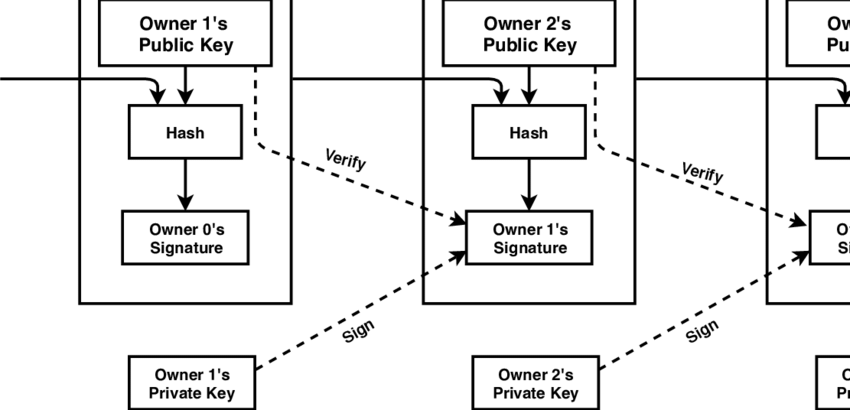


Figure 1 - Diagram of Bitcoin transaction verification; Satoshi Nakamoto, Bitcoin: A peer-to-peer electronic cash system. The public key for the transaction is used to verify the transaction by miners, but only the Owner’s Private Key can be used to sign and authenticate the transaction or decrypt the transaction to reveal non-public information.

**Encryption**

Nakamoto uses encryption to ensure that Bitcoin transactions are secure through the use of ECDSA (the Elliptic Curve Digital Signature Algorithm) (A. Narayanan, 2016), a form of asymmetric or “public key” encryption.

In asymmetric cryptography, there is a public and private key, one which can be accessible to the public and one which must remain private. One allows a user to encrypt, while the other allows a user to decrypt a message (K. Brush, 2021).

Bitcoin implements this by using the user’s private key to generate the transaction signature, which is then used along with the public key for verification of transactions. Once the owner has signed the transaction, it goes on to be processed by miners who can use the public key to verify the authenticity of each transaction (Nakamoto, 2009).

**Blockchain**

The “blockchain” is a distributed database which acts as a public ledger containing records of all Bitcoin transactions (A. Hayes, 2022).

Records of transactions are encrypted and compiled into “blocks”, which are verified by miners and added to the “chain” of blocks (A. Hayes, 2022). Each block contains an encrypted hash of the previous block, which means that previous blocks cannot be edited as even a small change would change the hash value significantly, which would be detected as the chain would no longer be valid. Changing one block would require changing all following blocks to ensure validity. This creates security in the system.

Transparency is also promoted by this blockchain system - for example even if a hacker attacks an exchange and gains access to the coins, it will be known if these coins are transferred or spent as the ledger of transaction records is public (A. Hayes, 2022).

This system allows for anonymity, as records are encrypted, and only the owner can decrypt them to reveal their identity through use of their private key (A. Hayes, 2022).

Transactions must be checked for validity before they can be added to other transactions to create a block so that they can be added to the blockchain. Only valid transactions can be added, so each transaction must be checked for authentication by the sender, and it must be verified that the sender has sufficient balance in their account for the payment, which enabled by the public transaction history held in the blockchain itself. This history contains all Bitcoin transactions since its inception.

### 2.1.2 Altcoins

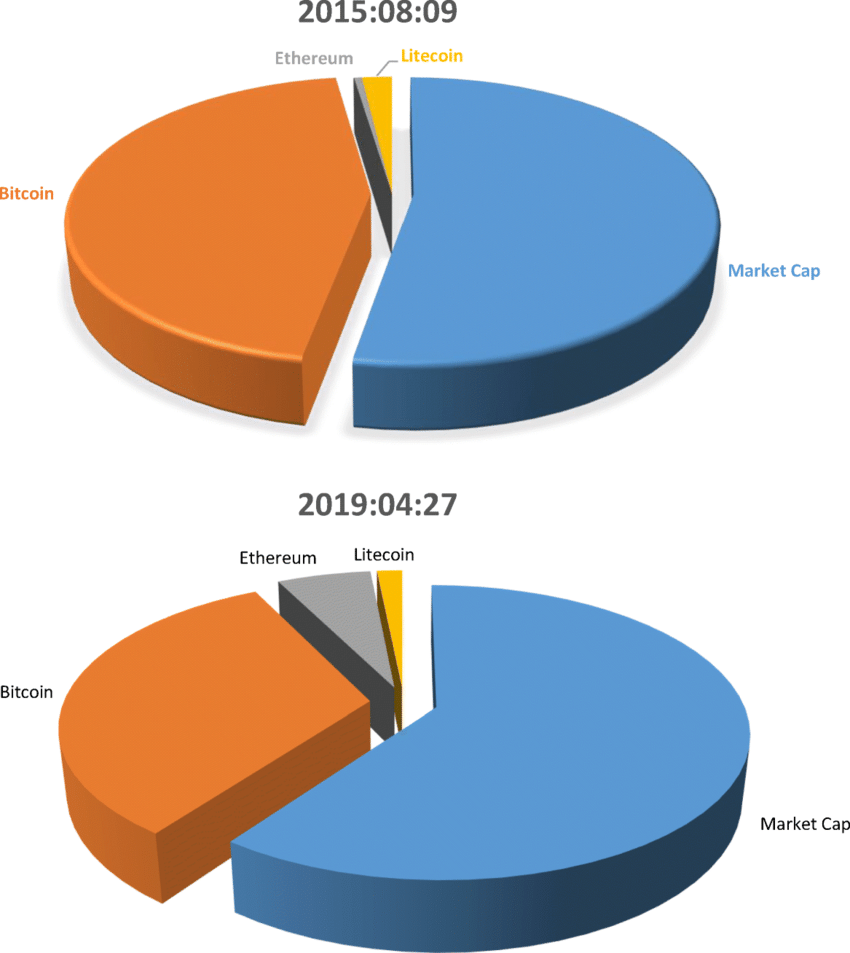
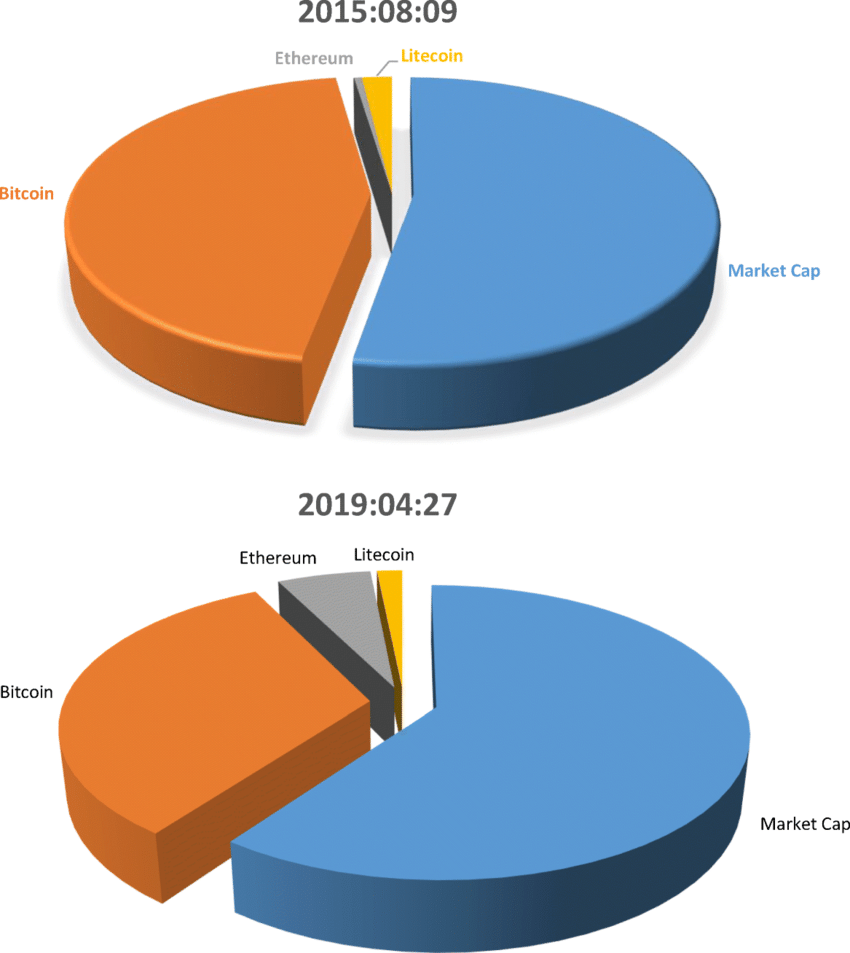


Figure 2 - Largest Cryptocurrencies Market Share 2015/2019; Springer Nature, Open Economies Review

Over 10,000 active “altcoins”, or non-Bitcoin cryptocurrencies, have been released since the advent of Bitcoin in 2009 (Howarth, 2022). They normally aim to improve on/add a feature of Bitcoin or address one of the cryptocurrency’s flaws.

The leading Altcoin in the market is Ethereum (Investopedia, 2022), the second biggest cryptocurrency by market capitalisation. Ethereum allows the formation of “smart contracts” (A. Hayes, 2022) which are accounts with a balance that execute a transaction upon fulfilment of certain conditions automatically without any intermediary. This is made possible as it allows distributed applications to be deployed across the blockchain (A. Hayes, 2022).

Another leading early altcoin is Litecoin, which has many similarities to Bitcoin, however it uses a different mining algorithm to promote the use of CPUs rather than GPUs, as this was an issue with Bitcoin mining that had practically rendered CPUs obsolete for mining purposes. In addition, its targeted block time is 2.5 seconds as opposed to Bitcoin’s 10, which means that Litecoin can be 4 times faster at confirming transactions (Genesis Mining, n.d.).

Ripple is an altcoin that improves on the scalability of Bitcoin, which is one of the cryptocurrency’s recognised flaws, by using server nodes that find consensus over short intervals rather than using the blockchain as most cryptocurrencies do (M. Travis, 2017).

These are all examples of how altcoins have emerged to meet specific tailored needs and improve upon the work that was begun by Nakamoto with Bitcoin.

It is very important to consider these altcoins in addition to Bitcoin, as Figure 2 above shows how the market share of BTC, although still the majority, has decreased greatly. In addition, these different cryptocurrencies are affected by external events to different degrees and can follow different trends which affects their volatility and predictability.

### 2.1.3 Trading

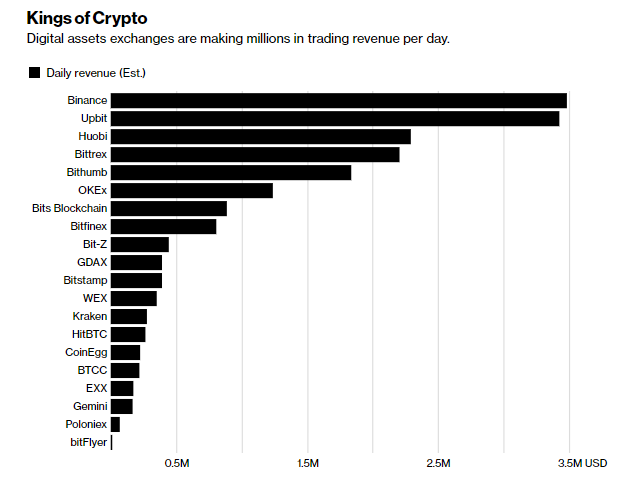


Figure 3 - Daily Revenue of the largest crypto exchanges, calculated with trading volume data from CoinMarketCap.com; Economic Times, Cryptocurrency exchanges are raking in billions of dollars

Cryptocurrency exchanges act as intermediaries in the trading of cryptocurrencies. Trading is the principal way that people obtain cryptocurrency, in addition to mining. The price of these coins is determined by market forces as they have no intrinsic value, solely being driven by interest and demand. This is shown by strong correlation with sentiment data such as Google Trends (Smuts, 2018). This is similar to the behaviour of stocks, however stock values are also driven by the performance of the company.

Cryptocurrency trading is attractive over stock trading for multiple reasons:

The nature of cryptocurrency as digital and transparent through the use of the blockchain means that it is available to be traded all the time, and exchanges do not close for weekends or at night.

The open-source and transparent aspect of these digital currencies means that all live data is freely available with free APIs online to use this data quickly and easily such as “CryptoCompare”, which is used in this study. This is opposed to live financial analytical data which can be highly expensive. An example of this is the Bloomberg Terminal which costs approximately US$24,000 yearly for real time price data, analytics and trading services (Investopedia, 2022).

Cryptocurrency exchanges charge lower fees for their services, often simply earning money from trading fees and withdrawal fees without any upfront charges (Bitcoin, n.d.).

This lower barrier to entry for crypto trading is attractive to investors as it makes it very quick, easy and cheap to begin trading. This also makes cryptocurrency highly suited for the testing of predictive models due to the wealth of price data readily available, and as they can be tested additionally by the increased volatility.

## 2.2 Financial Forecasting

### 2.2.1 Stock Price Prediction

There are multiple types of analysis used in stock trading to augment and determine trading strategies or draw useful information from the available data. These include fundamental analysis of the asset value, technical analysis of trading information and sentimental analysis.

**Fundamental Analysis**

“Fundamental analysis (FA) is a method of [measuring a security's intrinsic value](https://www.investopedia.com/terms/v/valuation.asp) by examining related economic and financial factors. Fundamental analysts study anything that can affect the security's value, from macroeconomic factors such as the state of the economy and industry conditions to microeconomic factors like the effectiveness of the company's management.

This method of stock analysis is considered to be in contrast to [technical analysis,](https://www.investopedia.com/terms/t/technicalanalysis.asp) which forecasts the direction of prices through an analysis of historical market data such as price and volume.” (T. Segal, 2021)

Fundamental Analysis allows an investor to decide whether a stock is overvalued or undervalued based on the valuation obtained from the analysis in relation to the current market price of the security (T. Segal, 2021).

**Sentimental Analysis**

Sentiment analysis, also known as opinion mining or subjectivity analysis (Mejova, 2009), studies the subjective opinion elements present in a unit. This can be a word, sentence or phrase (Mejova, 2009).

When sentiment analysis is performed, it allows you to determine – through analysis of the *implicit* and *explicit* sentiments expressed, and the specific features mentioned (Mejova, 2009) – the overall opinion and interest in a subject can be gauged, in addition to specific merits and concerns that may be highlighted.

This relates particularly to the analysis of stocks and cryptocurrency due to the strong correlation between their prices and positive sentiment/public interest.

**Technical Analysis**

Technical Analysis studies the behaviour of the market price of a stock, and its relation to previous prices and volumes traded to determine underlying trends for prediction and stock analysis.

“Technical analysis is a trading discipline employed to evaluate investments and identify trading opportunities by analyzing statistical trends gathered from trading activity, such as price movement and volume. Unlike fundamental analysis, which attempts to evaluate a security's value based on business results such as sales and earnings, [technical analysis](https://www.investopedia.com/terms/t/technical-analyst.asp) focuses on the study of price and volume.” (A. Hayes, 2022)

The nature of this type of analysis as being focused on the market value of the security being exchanged makes this directly applicable to this study and will be further explored in the next subsection.

The high-dimensional, non-linear nature of the relationships between these factors lends itself to application through Machine Learning methods, which has resulted in the rising popularity of this method for stock price forecasting, being viewed as essential for successful trading strategies in addition to other forms of analysis (Soni, 2022).

### 2.2.2 Forecasting Techniques – Technical Analysis

Technical analysis is used to infer statistical trends within the data for the purposes of stock analysis and price forecasting. This can be performed in a variety of ways in order to estimate the movement of prices and the strength of a current trend. These methods include Moving Averages, Directional Moving Index and Relative Strength Index techniques (Credit Suisse, n.d.).

In stock price forecasting, data smoothing is employed to reduce the noise from the data by smoothing the curves of the graph mathematically. This addresses the volatility of the data and improves a model’s ability to gauge the overall trend of the prices.

**Average Directional Moving Index (ADX)**

Average Directional Index is used in technical analysis to determine the strength of a trend in the data (C. Mitchell, 2021).

A picture containing text

Description automatically generated

Figure 4 - ADX Formulae; Investopedia, Average Directional Index (ADX)

This formula uses the previous calculated ADX, in conjunction with a calculated directional movement value to produce a positive and negative directional index which represents the strength of the positive or negative trend in the current behaviour of the data (C. Mitchell, 2021).

**Relative Strength Index**

Relative Strength Index, or RSI, is another momentum indicator used to determine the speed and magnitude of trends in recent price changes to determine if a stock is over or under valued (J. Fernando, 2022).

It utilises two calculations, the first of which is used to determine the first several RSI values (generally 14) using the average percentage gain/loss.

Diagram

Description automatically generated with medium confidence

Figure 5 - RSI Formula 1; Investopedia, Relative Strength Index (RSI) Indicator Explained With Formula

After the first number of values are calculated, the second formula is employed to calculate further RSI values. This aims to smooth the data so that it is more representative of the current trends, and will have a high value (near 100), when the stock is overbought and trending strongly up, while it will maintain a low value if the stock is oversold and trending strongly downwards (J. Fernando, 2022).

A picture containing diagram

Description automatically generated

Figure 6 - RSI Formula 2; Investopedia, Relative Strength Index (RSI) Indication Explained With Formula

**Moving Average**

The Simple Moving Average, or simply Moving Average, aims to smooth the price data by calculating a “moving average” at each point (J. Fernando, 2022), which will be used in place of the actual data to allow it to be used more effectively for training.

This lowers the impact of short-term fluctuations in price and addresses the volatility of stocks and cryptocurrency.

Graphical user interface, text, application

Description automatically generated

Figure 7 - MA Formula; Investopedia, Moving Average (MA): Purpose, Uses, and Examples

The moving average is calculated with the formula shown above and calculates a simple average that updates across each point in the dataset, computed using a given window size.

**Exponential Moving Average**

The Exponential Moving Average (EMA) is a variation of the Moving Average which smooths the data, however EMA weights more recent price changes higher, making it more reactive to price changes (J. Fernando, 2022).

Text

Description automatically generated

Figure 8 - EMA Formula; Investopedia, Moving Average (MA): Purpose, Uses, and Examples

## 2.3 Time Series

“A time series is a sequence of data points that occur in successive order over some period of time. This can be contrasted with [cross-sectional data](https://www.investopedia.com/terms/c/cross_sectional_analysis.asp), which captures a point in time.

In investing, a time series tracks the movement of the chosen data points, such as a security’s price, over a specified period of time with data points recorded at regular intervals. There is no minimum or maximum amount of time that must be included, allowing the data to be gathered in a way that provides the information being sought by the investor or analyst examining the activity.” (A. Hayes, 2022)

Time series data is important as it is possible to use time series and technical analysis to create forecasts, and it is particularly suited to training with certain neural networks.

Time series analysis (or trend analysis) examines the trend of the data on each specific day based on its previous performance (A. Hayes, 2022), which is the form of analysis that will be employed in the training of the models.

## 2.4 Machine Learning

Machine Learning seeks to develop systems that, through experience, can improve their own performance (T Mitchell, 1990). This is often done through the use of “neural networks”, which are made up of many individual “neurons”. These are based on their biological counterparts, taking multiple inputs, and calculating an output based on a mathematical “activation function”.

Several of these “neurons” were connected to create the Perceptron by Rosenblatt et al. in 1957, which laid the foundation for neural networks.

Neural networks approximate non-linear functions mathematically through the use of these neurons. Each neuron takes multiple inputs, which are given weights and biases which will affect the output. Through a backpropagation algorithm, the output given is used to readjust the weights to improve the model’s results. The neurons are arranged into layers, including an input layer, output layer, and at least one hidden layer.

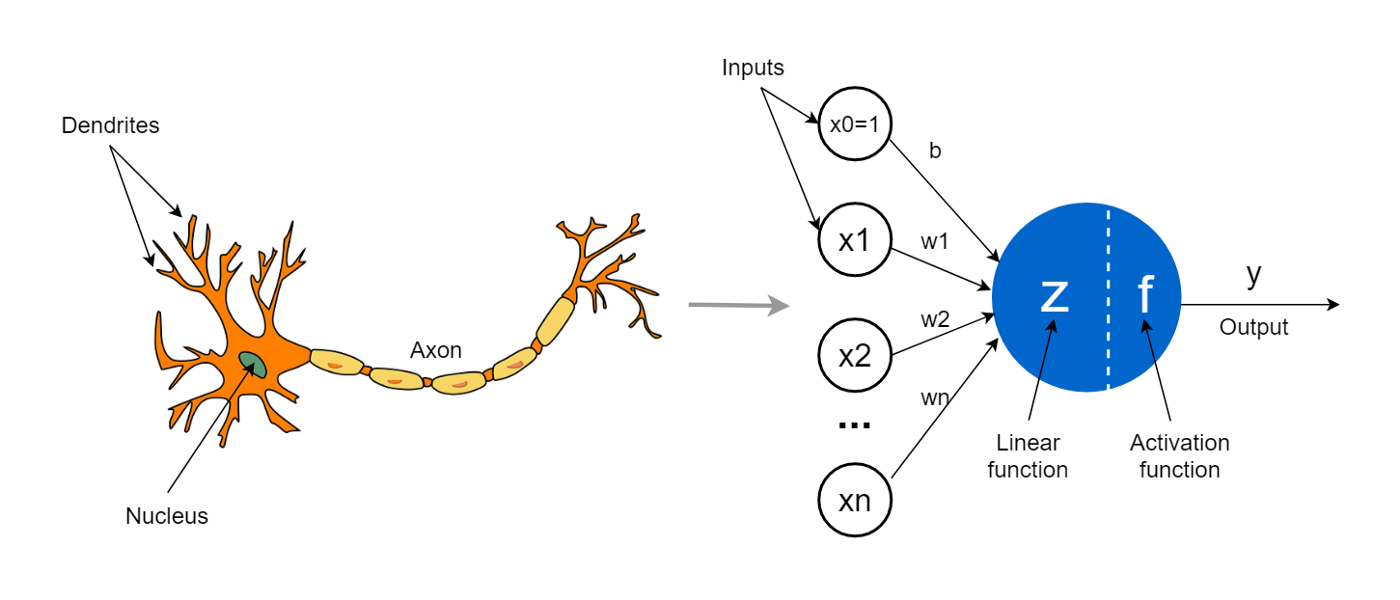


Figure 9 - Artificial Neuron against biological inspiration; R Pramoditha, The Concept of Artificial Neurons (Perceptrons) in Neural Networks, 2021

### 2.4.1 Fundamentals

Machine Learning tasks are classed as either regression or classification problems. Classification is when ML is applied to learn to separate data between categories, such as differentiating a dog from a cat, or determining if a tumour is benign or malevolent based on previous data, whereas a regression problem involves the calculation of an output value through the approximation of a function.

There are four basic categories of ML algorithms:

* Supervised Learning is the most basic form of Machine Learning, in which a model is provided example data with labels (in the case of classification) or correct output values or “ground truth” (in the case of regression). This data is used to train the model, and the model is then evaluated on its performance against unlabelled test data to determine its effectiveness (Zhang, 2010).
* Unsupervised Learning trains the model based on a set of inputs, usually by looking for similarities between the inputs provided, without labels or the corresponding ground truth values (Zhang, 2010).
* Semi-supervised Learning utilises both labelled and unlabelled data in the training of models (Zhang, 2010).
* Reinforcement Learning is learning based on the impact of the models’ actions on the environment in order to assess and improve itself based on its previous performance (Zhang, 2010).

### 2.4.2 Functions & Hyperparameters

Certain functions are applied within neural networks for training. Which functions are used will determine how well the model trains and the output of the model. There are 3 types of these functions: activation functions, optimizer functions and loss/error functions.

**Activation Functions**

Activation functions determine when the neurons will “fire” by defining the activation threshold which must be exceeded by the input to allow an output. These activation functions allow the neuron’s output to learn and recognise complex features in the data due to the additional complexity, as otherwise the output would simply be a linear function (S. Sharma, 2020).

Common examples of activation functions include (S. Sharma, 2020):

* **Sigmoid:**

*f*(x) =

* **Hyperbolic Tangent (Tanh):**

*f*(x) = 2(sigmoid(2x))-1

* **Rectified Linear Unit (ReLU):**

*f*(x) = max(0, x)

ReLU returns 0 for negative values, and returns the input if greater than 0

* **LeakyReLU:**

*f*(x) = 0.01x, x < 0

*f*(x) = x, x >= 0

LeakyReLU scales down negative values linearly, while values greater than 0 remain the same

**Loss/Error Functions**

Loss functions determine how the model will train by calculating the “loss” or “error” between the results achieved and the validation and test data. This allows the model to determine how well it is performing so that it can improve its predictions in further training runs, and how this is calculated will determine exactly how the model decides to adapt its weights and biases when determining its new outputs as it trains.

Common examples of loss functions include:

* **Mean Squared Error (MSE):**

*f*(x) = Σ(observed - predicted)2

* **Mean Absolute Error (MAE):**

*f*(x) =

* **Mean Absolute Percentage Error (MAPE):**

*f*(x) =

**Optimiser Functions**

Optimiser functions are the functions that decide how the weights and parameters of a neural network are changed or “optimised” in order to reduce error. The choice of optimiser can affect the rate of learning hugely and choosing the correct one can great improve the efficiency of a model (Doshi, 2019).

These optimiser functions are implemented as a form of mini-batch gradient descent which takes a set number of samples to train with before updating the weights and moving on to the next batch in the epoch. This is opposed to batch gradient descent which only updates the weights after training with all samples, and stochastic gradient descent which updates the weights after each sample (Doshi, 2019).

Common examples of optimisers include:

* **Adaptive Gradient Algorithm (AdaGrad):**

AdaGrad is an optimiser function which automatically adapts the learning rate, eliminating the need for manual tuning. It also adjusts the learning rate for each parameter, assigning higher learning rates to parameters related to infrequent features and lower rates to those related to more frequent features. This allows AdaGrad to converge more quickly and reliably than SGD (Databricks, n.d.).

* **Root Mean Squared Propagation (RMSProp):**

RMSProp is a gradient optimisation algorithm related to AdaGrad. However it uses a decaying/moving average when calculating learning rates rather than the total sum, which results in learning rates that are not reduced as quickly. This can lead to better results as low learning rates can slow down progress and training can stop before the minimum is actually reached (Brownlee, 2021).

* **Adam:**

Adam is an optimiser that is often considered to capture the benefits of both AdaGrad and RMSProp. This is because it uses an exponential moving average in its learning rate calculations, rather than the sum or a simple moving average. It is often recognised as the default optimiser to use as well as SGD (Brownlee, 2017).

* **NAdam:**

NAdam is an extension to the Adam optimiser which utilises Nesterov momentum in its calculations, which is an improved form of momentum that utilises a decaying average (Brownlee, 2021).

* **Stochastic Gradient Descent (SGD):**

Stochastic Gradient Descent is recognised as a standard optimiser, which updates weights and biases after each training sample. It is the most basic optimiser and is often used by default.

There are many more optimisers, loss functions and activation functions, including variations of the ones mentioned above, however this study will focus on the aforementioned examples as the foremost methods used in the field to compare performance.

**Hyperparameters**

Hyperparameters are the parameters used within the training of the model, including basic parameters such as the number of epochs, training batch size and learning rate. These determine how much and how well the model will train.

The number of epochs determines how many “runs” the model will perform through the data as it trains. If this value is too low, the model can undertrain and will not perform as well as it could, however if this value is too high then it can lead the model to underperform due to overtraining to the training data, rendering it unable to adapt to changes that are present in the test data.

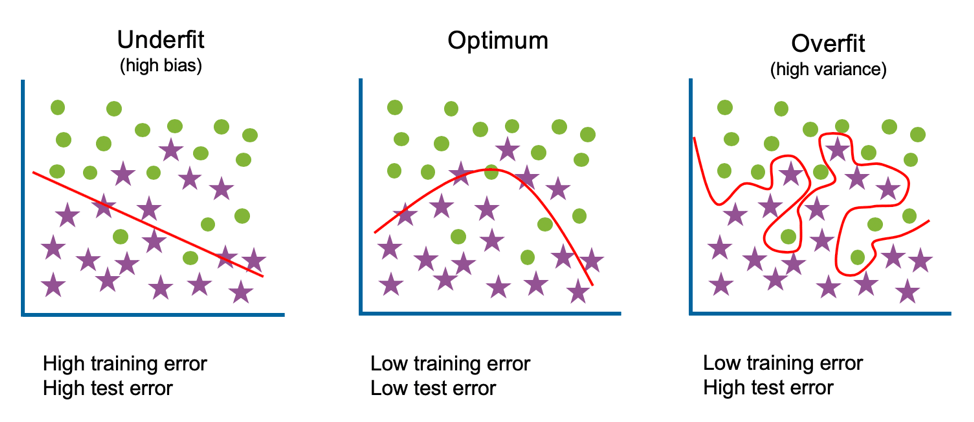


Figure 10 - Underfitting & Overfitting Examples; IBM, Overfitting

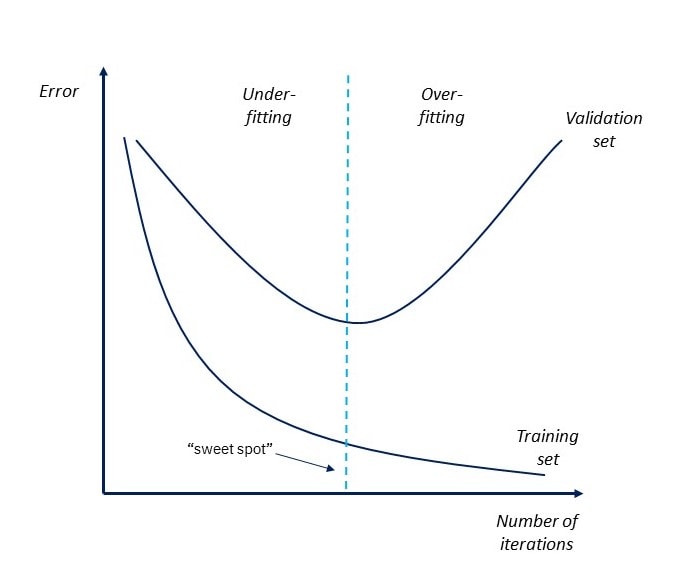
Overfitting is very important in determining the number of iterations a model should undergo.

Figure 11 - Overfitting and Iterations; IBM, Overfitting

High variance and low error rates in training are symptomatic of overtraining and will therefore result in higher error in testing (IBM, 2021).

This bias-variance trade-off must be taken into account in experimentation in order to attempt to find the “sweet spot” to maximise the efficiency of training (IBM, 2021).

This would enable the model to train from the data as much as possible without sacrificing performance later by overfitting (IBM, 2021).

However, for deep learning models and neural networks this does not always apply, and they can continue to train after interpolation. In these cases, remedies such as dropout and early stopping can be detrimental (IBM, 2021).

The batch size determines how many datapoints are used in each iteration of training before the weights and biases are updated. This is repeated in batches until all of the data has been used, after which a new epoch begins and training restarts (Sharma, 2017). Changing this number affects how the model learns and trains from the data, as the weights will be set based on comparisons from different groups of data. In addition, the use of smaller batches can allow training of neural networks with lots of data without utilising large amounts of RAM.

The learning rate is a multiplier which decides the size of the “steps” that the model takes when adjusting gradients. This value is important and can be very consequential to the performance of a model. If the value is too low, it will take much longer to train, and the gradient descent algorithm may get stuck in a local minimum. Otherwise, if the value is too large, it can overshoot minima, which reduces the effectiveness of the predictions (Google Developers, n.d.).

Diagram

Description automatically generated

Figure 12 - Overshooting due to high learning rate; Google Developers, Reducing Loss: Learning Rate

There are additional hyperparameters involved in the architecture of the layers themselves, which include the number of neurons included in the network and the values used in specific layers, such as dropout.

The number of neurons determines the depth of the network, which can affect its performance depending on the data.



Figure 13 - Diagram of Neurons in a Neural Network; IBM, What are Neural Networks?

### 2.4.3 Neural Networks

Neural Networks are algorithms which learn to perform tasks through training and optimisation processes. These networks are made up of individual nodes called “neurons”, based off of their biological equivalent. These nodes are arranged into “layers”, through which the data is passed during training to allow the model to assign new weights and biases to the parameters in order to improve its performance (B. Muller, 1995).

### 2.4.4 Deep Learning

Deep Learning is a form of machine learning that attempts to utilise higher numbers of larger layers in order to take advantage of this increased depth to find complicated relationships between large amounts of data (IBM Cloud Education, 2020).

This allows deep learning models with higher precision and less data pre-processing, depending on the task (IBM Cloud Education, 2020).

A neural network with 3 or more layers is generally considered an example of deep learning (IBM Cloud Education, 2020). While all of the models used in this study will fit this criterion, a version of each with more layers will be tested to examine the effect of increased depth on the network’s performance.

### 2.4.5 Recurrent Neural Networks (RNNs)

Diagram

Description automatically generated

Figure 14 - Recurrent (left) vs Feed-Forward NNs (right); IBM, Recurrent Neural Networks

Recurrent Neural Networks are a type of artificial neural network designed to use sequential or time series data. They are known as recurrent networks as the nodes do not simply output to the next layer but are interconnected. In addition, weight are shared between layers in a Recurrent NN (Education, 2020).

They leverage BPPT (Backpropagation through time), rather than normal backpropagation in the calculation of weights, which differs as it is specific to sequence data (Education, 2020). This specialisation in this type of data suggests that RNNs are strongly suited for the task presented by this study.

RNNs run into problems with “exploding” or “vanishing” gradients when the gradient of the slope of the error function is too high or too low. This issue can be reduced by lowering the complexity of the model by removing layers, changing activation/loss functions or using a variation of the RNN, such as the LSTM which is covered in the next section.

### 2.4.6 Long Short-Term Memory (LSTM)

LSTM is a variation of a Recurrent Neural Network which uses a slightly different structure in its individual units in order to specifically deal with the exploding and vanishing gradients problem. Due to this, LSTM RNN models are versatile and very well suited to the analysis of time-series data (Brownlee, 2017).

### 2.4.7 Convolutional Neural Networks (CNNs)

Diagram, engineering drawing

Description automatically generated

Figure 15 - Example of CNN Architecture; S. Saha 2018, A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way

A Convolutional Neural Network is a neural network most often used in a 2-dimensional format with images as an input. It aims to extract high-level features such as edges from the image, while keeping the computational cost low (Saha, 2018).

CNNs utilise a “kernel”, which confines the networks “attention” to a small section of the image, which moves through the image based on the chosen “stride length”. This allows the network to pick up high-level features in the data, as well as the spatial and temporal relations between each data point, while keeping the computational cost low as the resolution increases (Saha, 2018).

These networks also utilise “pooling” layers, which can return the average or max values from the portion of the image covered by the kernel. Both of these methods of pooling can be used to reduce the computational complexity of the problem, while also reducing additional noise in the images which can interfere with training (Saha, 2018).

**1-Dimensional CNN**

There are variations of the classic 2-Dimensional CNN, one of which is the 1D CNN. In this case, the kernel only slides in one dimension, rather than covering a 2-dimensional image, while still capturing the spatial properties of the data. This makes this type of network very well-suited for the processing of time-series data (Verma, 2019).

### 2.4.8 Gated Recurrent Unit (GRU)

Gated Recurrent Unit Networks are another form of RNN designed to combat the vanishing gradient problem. It was introduced by Cho, et al. in 2014 (K. Cho, 2014). Due to similarities in design and purpose, it is often considered a variation of the LSTM (Kostadinov, 2017).

It utilises two “gates” to address the vanishing gradient problem, the “update” and “reset gates.

**Update Gate:**

Diagram

Description automatically generated with medium confidence

Figure 16 – Update/Reset Gate Formula; S. Kostadinov 2017, Understanding GRU Networks

The update gate utilises weights through the formula shown above to decide how much past information should be passed along. This allows it to use previous data to varying degrees in its predictions, which allows it to adapt to different problems. The ability to set the weights to use all previous data allows it to eliminate the vanishing gradient problem if necessary (Kostadinov, 2017).

**Reset Gate:**

The Reset Gate, in turn, utilises the same formula to decide how much past information should be forgotten and not used in the processing of future data. The difference between these gates lies in the usage of the gate and the weights applied (Kostadinov, 2017).

The use of these gates allows the GRU networks to store and filter information to address vanishing gradients, allowing them to perform very well with complex data if properly trained (Kostadinov, 2017).

### 2.4.10 Regularisation - Dropout

Diagram, schematic

Description automatically generated

Figure 17 - Diagram of Dropout Implemented on a Neural Network; N. Srivastava 2014, Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Dropout is a technique used in machine learning which allows the model to randomly “drop” neurons based on a set probability. Dropout can prevent overfitting in a model and can also be used over runs to simulate training with multiple different network architectures, however it can also interfere with the model’s ability to learn from the data, particularly if the probability used is too high (Srivastava, 2014).

### 2.5 Related Studies

In this section I will be going over related work that has been done with a similar goal or methodology in mind to this project.

Due to the nature of the subject, papers and work relating to this topic only exist in the last few years, which makes the literature quite limited and further work is to be expected in the near future.

Much of the work done on this topic focuses on Bitcoin, as the first, largest and most popular cryptocurrency. This, however, is not able to truly capture how Machine Learning can be used in this task, as less-volatile cryptocurrencies such as Ethereum should also be covered (S. Charandabi, 2021).

This is shown by Jaquart et al. in their 2021 paper “Short-term bitcoin market prediction via machine learning” (Jaquart, 2021) analyses the predictability of Bitcoin using both technical and sentiment-based features. This paper focuses on the short-term predictability of the market, with forecast horizons ranging from 1 minute to 60 minutes. They use variations of LSTM and GRU models in order to achieve this, using data from 2019.

This experimentation yielded results with a true accuracy of approximately 50%, however each model is shown to outperform a random classifier. The study concludes that RNN models are more appropriate than GRU or Feed-Forward models and finds that longer prediction horizons are more accurate.

This study, while quite comprehensive in terms of the models implemented and through the inclusion of sentiment data (which is beyond the scope of this study), does not include altcoins in its experimentation.

This study hopes to address this by testing and training the models on multiple cryptocurrencies.

E. Pintelas et al. explores the performance of deep learning models in the prediction of cryptocurrency prices from BTC, ETH and XRP in their 2020 paper: “Investigating the Problem of Cryptocurrency Price Prediction: A Deep Learning Approach” (Pintelas, 2020). It compares the performance of these models against SVM, 3NN and DTR machine learning models using a 70% training set split of data from 2018 to 2019. These models also achieve accuracies of approximately 50%.

This study concludes that, of the deep learning approaches evaluated, CNN-LSTM performed the best across datasets, however it finds that these models are “inefficient and unreliable” (Pintelas, 2020) in the prediction of cryptocurrency prices due to the complexity of the problem and the volatility of the currencies involved.

Non-deep learning approaches, such as Genetic Algorithms, are also covered in relevant work, such as by Sin et al. in 2017 (Sin, 2017). In this study, the authors explore the performance of a Genetic Algorithm based Selective Neural Network Ensemble (GASEN) in Bitcoin price prediction, using the previous 2 years of historical data. This study reports a “consistent accuracy of around 58% to 63%” (Sin, 2017).

As can be seen from these examples, machine learning approaches generally result in a prediction accuracy of 50-60%, which shows that machine learning has some promise for this application, but it is limited by the complexity of the problem of predicting cryptocurrency prices.

Most of the relevant work already completed suffers from a lack of training data, particularly the earlier studies, due to the novelty of cryptocurrencies in general. This results in many of the models used being overfitted due to the small dataset sizes.

Now, in 2022, with more data available, I am able to address this problem within my own work by utilising more historical data. This also allows me to vary the dataset size used in order to evaluate the effect this has on the predictions. I also hope to address the problem of overfitting through the inclusion of dropout layers in the neural network architectures.

This project aims to present a comprehensive comparison of the performance of different neural networks for this task, the effect of changing the depth of the architectures of the models on predictive performance.

The potential for Data Smoothing to be helpful in forecasting will also be explored due to the similarities between the behaviours or cryptocurrency and the stock market, which is not represented in the current literature.

Although Sentiment Analysis has been shown to improve the accuracy of results and have a correlation with changes in the price of Bitcoin due to the lack of intrinsic value in cryptocurrencies, this study will focus solely on training using historical price data.

# 3. Choice of Experiments

Due to my research of the relevant literature and technologies as well as related work already done on the subject, I have chosen to focus my experimentation in 3 particular areas: training data, neural network types, and data smoothing. By including all 3 of these areas, I hope to provide a more comprehensive analysis of predictability than is currently available in the literature, while attempting to evaluate the inclusion of data smoothing techniques used in the financial sector.

**Training Data**

In order to experiment with the relation between the data used and the efficacy of the model trained with said data, I will be training and testing each of the models with different datasets.

These datasets will be from 2 cryptocurrencies, namely Bitcoin & Ethereum. The inclusion of ETH addresses the focus on Bitcoin mentioned in my assessment of related work. This allows the models to be assessed on their performance with different data, and on currencies with different behaviours and volatility.

In addition to the use of an altcoin, I will be comparing the performance of the models using hourly and daily price datasets for each cryptocurrency, in order to assess how this affects the ability of the model to predict future prices.

Experimentation will also be performed with the length of the data used (for example, with daily data performance will be compared when using 1 year of data, 3 years of data, 5 years of data, etc.)

**Neural Networks**

Through the course of this experiment, I will experiment with different Neural Network architectures to evaluate the differences in performance for the purposes of price prediction. This addresses the problem in the relevant work already done in this field that many of these studies focus solely on one or a few types of networks, as this project aims to show the differences in performance side-by-side.

**Data Smoothing**

I will test the implementation of Data Smoothing techniques used in Stock Price Forecasting, as explored in the literature review, and examine how the application of these techniques affect the performance of the models. This is important due to the similarity in behaviour of cryptocurrencies to stocks, as mentioned in the previous section, so predictive techniques as used in the financial sector are highly applicable to this study.

# 4. Data Overview

All daily and hourly datasets are obtained using CryptoCompare API: https://min-api.cryptocompare.com/

The datasets used contain the historical price data necessary to train and test the neural networks. These include the high & low prices for the period (daily or hourly), the opening and closing prices for the period, the cryptocurrency volume traded within that time, and the date/time. For this study we are using the previous 5 years of historical price data for each cryptocurrency.

Fields:

|  |  |
| --- | --- |
| Field | Description |
| High | The highest price reached during the day |
| Low | The lowest price reached during the day |
| Open | The price at the first transaction of the day |
| Close | The price at the end of the day |
| Volume from | Volume traded at the beginning of the day |
| Volume to | Volume traded at the end of the day |

The nature of the data as regular measurements over time within a specified time frame allows it to be analysed as a time series data sequence by neural networks.

All of these fields will be used in the training of the neural networks as they contain relevant information about the price and volume changes during the time period which will be used to determine future prices.

Chart, histogram

Description automatically generated

Figure 18 - Historical Price Data for BTC – Last 5 Years, from the dataset

From simply looking at the visualised graph data for the historical prices, it is clear that cryptocurrency prices are highly volatile, as was explored in the literature, and very prone to outside influences. This can be seen particularly in the case of Bitcoin, which rose dramatically during the COVID-19 pandemic, only to then fall sharply during 2021 due to negative comments from Elon Musk (R. Browne, 2021) and the Chinese Government’s crackdown on cryptocurrency (BBC, 2021). The currency recovered after this slump, but due to further external factors such as a potential looming recession due to COVID-19 and the war in Ukraine, the price has been steadily crashing since.

This volatility is visible not simply on the grand scale with historical data, but in the price movements day-to-day. This can be seen when the amount of data is reduced to just the last year of price data:

Chart, line chart

Description automatically generated

Figure 19 - Historical BTC Price Data from the dataset - 1 Year

The abnormal behaviour of the cryptocurrency prices in the recent period due to external factors could additionally serve to test the ability of the models to deal with anomalous circumstances.

# 5. Data Pre-processing

Data Pre-Processing is applied to data in order to configure the data to allow models to train with it. In addition, further processing is performed to adjust the data to improve the ability of models to learn from the data, through processes such as normalisation, data smoothing, or the removal of anomalous data.

In this case, I have elected not to remove or ignore any of the data points as they are representative of the overall volatility of the subject prices being predicted which is inherent in cryptocurrencies, and therefore it would be more beneficial to have these trained into the models.

## 5.1 Data Normalisation

Normalisation or standardisation is applied to the data in order to improve the model’s ability to train. This is done by transforming the features of the data to be in the same scale.

It is mainly performed to remove information irrelevant to the predictions and works to normalise the data between given range.

It scales the price data down by the range of all of the data with the following function:



Figure 20 - Range Scaling Function; Google Developers, Machine Learning - Normalization

This allows the neural network to ignore information from outside this range, such as how high the prices are themselves, and simply focus on the change between them to determine its predictions of future behaviour.

## 5.2 Sequence Extraction

To address the volatility of the data, I need to ensure that the neural networks are trained to predict the price of the cryptocurrency based on the data from the previous few days. To do this, I extract windows of data as a sequence to use in the training of the models.

Each sequence is associated with a day and contains the price data for a given number of days previous to that day. This allows the model to train to predict prices based on current price behaviour rather than attempting to decide an overall trend from the whole dataset of prices where one does not exist.

This is an implementation of “feature engineering” and “feature extraction” as part of the pre-processing of the data to improve the ability of the model to extract meaningful information and make accurate predictions.

For these experiments, I will be using windows of 5 days (or 120 hours for the hourly dataset)

## 5.3 Financial Forecasting Algorithms

In this study I explore the effect of applying data smoothing, as implemented in financial stock price forecasting, on the performance of the predictive models. I have chosen to implement Simple and Exponential Moving Average smoothing in my experimentation as this can help the model to train by calculating a moving average for the data to cut out the noise of the volatile day-to-day price changes to find the underlying trend.

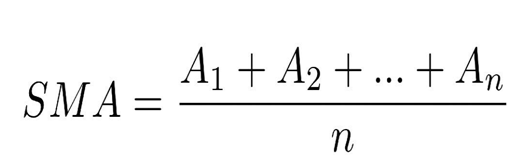


Figure 21 - Simple Moving Average Equation; CIN7, Moving Average

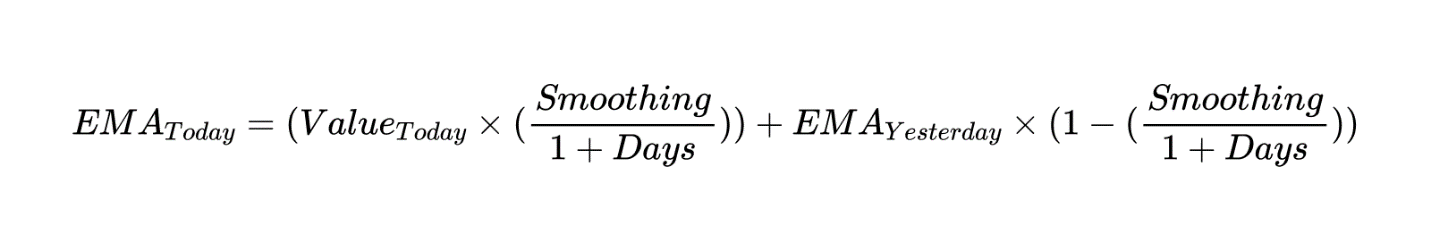


Figure 22 - Exponential Moving Average Equation; Trading View, Exponential Moving Average

The Exponential Moving Average is more likely to be effective, as it is more sensitive to price changes due to the higher weighting of recent data, making it more suitable due to the volatility of the subject data (Cory Mitchell, 2021), however I have also elected to include the Simple Moving Average in my experimentation for comparison.

As the focus of this study is on forming predictions based on historical price data, I have chosen not to implement sentiment analysis. Additionally, the lack of intrinsic value in cryptocurrency has meant that a fundamental analysis is not possible. Due to these considerations, I will only be testing the use of Moving Average Smoothing for technical analysis.

## 5.4 Dataset Split

When training Machine Learning models, data is commonly split into training, validation and testing data. The training data is used to train the models, while the validation data is used to provide metrics to evaluate the performance of the models as they train between epochs. This validation data acts as a reference to represent the performance of the model on data other than that which was used to train it before the final testing. Finally, the testing data is utilised at the end, unseen by the models, to properly assess their performance on unseen data.

Convention dictates that training data should make up the vast majority of the data, while a smaller segment of it is used for testing.

This is backed mathematically, as p ≈ 80% is empirically the ideal value for the test data split (A Gholamy, 2018). For this reason, I have split the data into 75% training data and 25% test data, slightly increasing the test data split in order to slightly more effectively determine the model’s ability to predict future prices.

One third of this training data is used as validation data for the model to compare its performance against in between training epochs.

Therefore, the final split is 50% training data, 25% validation data and 25% test data as determined by convention, research, and my personal goals for the project. This is consistent with the relevant work explored in the literature review.

# 6. Neural Networks

## Neural Networks to be implemented

From my research into relevant ML technologies, particularly neural networks, I have selected 6 different Neural Networks for comparison in this study: RNN, LSTM, GRU, CNN, CNN-RNN and CNN-LSTM.

These have been selected as they are specifically designed to be used with time-series data, as covered in my review of the relevant technologies, and are the most likely to be effective in this application. In addition, my examination of the relevant literature suggests that the use of RNN models, particularly LSTM (as well as CNN-LSTM) is most effective for this task.

Additionally, to explore the effects of utilising deep learning with larger networks, 2 models will be created for each NN type, one of which will have additional layers to explore the effect of added depth. Hybrid models with a CNN and RNN/LSTM layer are also included for evaluation.

In order to evaluate the performance of these models against each other, each of the models with lower depth shall have the same architecture, utilising the same layers, functions, and hyperparameters to ensure that they are trained and tested fairly. The same is true of the deeper model architectures, which will have additional layers compared to the shallower networks, however they will be identical in architecture to each other.

## Architecture

Diagram

Description automatically generated

Figure 23 - Base RNN Architecture

The basic architecture for the models implemented includes 4 layers, other than the input layer.

The first layer is the main layer being assessed for its performance, RNN/LSTM/GRU/CNN.

It also includes a ReLU activation layer, 40% dropout, and a dense layer. These were chosen from a combination of my research as presented in the literature review and experimentation.

All architectures used utilise NAdam optimisation, as it is believed to be more effective than the classic Adam and has proven suitable for this purpose through my own testing.

The loss function used is Mean Squared Error, which was selected as it proved more effective than Mean Absolute Percentage Error and Mean Absolute Error when used in training.

Each Neural Network contains 150 neurons in its layers, which is sufficient to capture the relationships in the data.

Diagram

Description automatically generated

Figure 24 - Deeper RNN Architecture

The slightly deeper models are identical to these previous models, however they contain an additional activation layer and dense layer to increase the depth of the network and allow us to see if this significantly improves the performance of the neural nets.

The inclusion of the additional Dense Layer should allow the networks to capture higher-dimensional relationships and features within the data if they exist, while the dropout layer should ensure that the models are not too overfitted to the data due to the increased complexity of the network.

Diagram

Description automatically generated

Figure 25 - CNN-RNN Hybrid Architecture

Finally, the hybrid models included are identical to the RNN and LSTM models already mentioned, however they contain a 1-Dimensional Convolutional Layer after the input layer, which is followed by an Average Pooling Layer to reduce the dimensionality of the data for further processing with Recurrent layers.

This further allows us to explore the effects of depth on the performance of deep learning in this application, as well as evaluating the performance of CNN-RNN (including CNN-LSTM) hybrid networks to assess how their prediction accuracies differ from the simpler neural networks.

# 7. Methodology

## 7.1 Data Pre-processing

### 7.1.1 Normalisation

Two normalisations are applied to the data in order to improve the model’s predictions:

The first normalises the data against the first item, using it as a reference and dividing the rest of the data by that value to reduce the scale of the data while maintaining the relationships between the values.

A second normalisation is applied to the data according to its range.

Text, letter

Description automatically generated

Figure 26 - Normalisation Functions from code

Each value in the dataset has the minimum price value subtracted from it. This reduces all of the data and removes additional information that is irrelevant to the relative price changes. These values are then divided by the total range of the data, calculated by subtracting the minimum price from the maximum price.

Often, “Feature Clipping” is employed when normalising data, which involves removing outliers from the data, however I do not employ this method of data normalisation as none of the price data is invalid, and this would simply emphasise the volatility of the prices, which is important to include in training.

### 7.1.2 Dataset Split

Chart, line chart, histogram

Description automatically generated

Figure 27 (Repeated) - Historical Price Data for BTC, from the dataset, including the dataset split

Figure 3 above shows the dataset split employed on the Bitcoin dataset. This ratio remains the same across all of the datasets to allow for fair comparisons.

The ratio of training data, validation data, and test data are 2:1:1. Therefore, 50% of the data is used for the initial training of the model; 25% is used for validation during the training; and 25% is used in the final testing of each model.

This split follows convention within the field, which is to use approximately 70-80% of the data for training overall and the remaining 20-30% for testing afterwards (A Gholamy, 2018). This allows the model sufficient information with which to train without overtraining, while leaving sufficient test data afterwards. These values can be tweaked and can have an impact on the performance of the model, however due to the volume of the data in the datasets this impact would be negligible. In addition, lowering the test data split could lead to overtraining, which I would prefer to avoid due to the general volatility of cryptocurrency.

This split should also prove valuable in the assessment of the models, as the test and validation portions of the data are affected by abnormal external influences including COVID-19, China declaring all crypto-currency transactions illegal (BBC, 2021) and the war in Ukraine, as explored in the Preliminary Data Analysis. This allows the models, trained on data from previously, to be tested on their performance when the prices are subjected to strong external influences. The presence of the validation data in this time period also means that it will also contribute to the training of the models and the metrics generated for evaluation.

### 7.1.3 Sequence Extraction

Text

Description automatically generated

Figure 28 - Sequence Extraction Code

Sequences of values for the previous “length\_of\_seq” days for each target day are extracted and passed to the models as they train to predict each day’s closing price. This allows the model to learn how the future price is influenced by recent changes in price rather than attempting to find a pattern in the overall behaviour of the historical data.

Changing the length\_of\_seq parameter allows us to experiment with the length of the analysed window and see if a shorter window based on the immediate trend of the data is more/less effective than a longer window containing more historical information which may mislead the model.

## 7.2 Evaluation Methods

In order to evaluate the performance of each model, the principal measurement of performance are the calculated “win-rates”. These win rates reflect the percentage of predictions performed on the test data which correctly forecast whether the price rises or falls in that period.

These win-rates are calculated for 4 separate forecasting horizons. For the daily data, win-rates are calculated for predictions 1 day, 3 days, 5 days, and 1 week in advance; models trained with hourly data are assessed on the performance of their predictions 1 hour, 6 hours, 12 hours, and 1 day ahead.

These varied win-rate calculations allow the evaluation of these models for both short-term and longer-term predictions. The shorter prediction horizons are more representative of the model’s ability to precisely ascertain the price changes and deal with the volatility of the data, while the longer forecast win-rates reflect the performance of the model in terms of capturing the trend of the data over a longer period.

Loss values are also calculated between the predictions of the models and the true testing values. The Mean Absolute Error, Mean Squared Error and Mean Absolute Percentage Error are all calculated to help compare the models. Graphs are also generated for the Training and Validation Loss of each model as they train, as this would help to determine if the winning models are overfitted or if they could be further trained, as a low training loss value with a validation loss that has plateaued would indicate overtraining.

## 7.3 Experimentation with Dataset Size

Each model is evaluated on its performance against both hourly and daily price data for both Bitcoin and Ethereum. In addition to this, they are trained and tested with different amounts of data in order to ascertain the effect that this has on the performance of the models and their ability to capture features of the data.

With the daily datasets, each model is trained with 6 months, 1 year, 3 years, and 5 years of historical price data. With hourly data, models are trained with 500, 1000, 1500, and 2000 hours of data.

Each of these dataset sizes should be sufficient for the training of Deep Neural Networks, however it is important to assess the effect of changing the dataset size like this for multiple reasons:

In the literature review it is briefly explored that much of the existing work in this area suffers from a lack of training data, particularly in less recent work, as cryptocurrencies themselves are very new.

The inclusion of larger quantities of data allows us to see if the model performs better when solely provided with the most recent data or if additional data allows the neural networks to better train and understand the behaviour of the prices. In addition, this will show if larger amounts of data are excessive and result in overtraining.

## 7.4 Implementing Data Smoothing

After experimentation with dataset sizes is complete, the optimal model for each cryptocurrency/dataset/data-quantity combination will have been established.

Using this information, the 3 best performing models for each dataset - using the optimal data quantity and win-rate - will be trained and tested twice more, once using Simple Moving Average smoothing, and once using Exponential Moving Average Smoothing.

By testing both of these methods, in addition to the initial testing without data smoothing, comparison of model performance should show whether the application of these financial forecasting techniques of technical analysis are useful in the training of Machine Learning models for cryptocurrency price prediction.

# 8. Analysis and Evaluation of Results (Main Results in Appendix)

Figure 29 - Data Quantity/Best Winrate Achieved Graphs

For daily data, the trend of the results suggests that a larger dataset size is more beneficial for longer prediction horizons, as the best models received the highest 1-Week win-rates when using 5 years of daily data (LSTM: 69.4%), while a lower score was achieved when using 6 months of data (LSTM: 62.5%). The same holds true for the best-performing 1-Day win-rate models for hourly data, with the best performer using 500 hours of data (GRU: 55%) performing worse than the best model that used 2000 hours of data (CNN-LSTM: 56.2%). However, this trend is much less prominent with the hourly data.

It can be seen by visualising the win-rate data that the highest win-rates achieved with daily price data utilised a 5-day predictive horizon and 1 year of historical data. Due to the volatility of the data, it is likely that forecasts both longer and shorter than this are more unpredictable, while additional training data beyond 1 year leads to overfitting and lower performance.

Chart, line chart

Description automatically generated

Figure 30 - Example of Loss Graph for Overtrained Model

This potential overfitting can be seen with some of the models, particularly in their loss graphs as the validation loss sometimes plateaus quite early on, while the model continues to train and reduce training loss. However, this plateau could also simply be due to the volatile and unpredictable nature of the data, and it is possible that further training could prove beneficial. This leaves room for further experimentation with hyperparameters, particularly the use of dropout and different epochs in the future.

By contrast, for shorter forecast horizons, the opposite is shown to be the case, with models performing better when provided with less data. Daily price data training with 6 months of historical data returned the most accurate model (CNN-RNN: 67.5%) compared to a significantly lower performance using 5 years of data (CNN-RNN: 49.9%), while the same is shown in the hourly-data-trained models with the best performing 1-Hour win-rate model (RNN: 53.3%) having been trained with the least data, 500 hours.

BTC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Used | Daily Winner | Score | Weekly Winner | Score |
| 6 months | CNN-RNN | 0.675 | LSTM | 0.625 |
| 1 year | CNN-RNN | 0.581 | GRU | 0.686 |
| 3 years | CNN-LSTM | 0.519 | LSTM | 0.694 |
| 5 years | CNN-RNN | 0.499 | LSTM | 0.694 |

ETH

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Used | Daily Winner | Score | Weekly Winner | Score |
| 6 months | GRU | 0.65 | GRU | 0.525 |
| 1 year | CNN-RNN | 0.57 | CNN-LSTM | 0.616 |
| 3 years | CNN-LSTM | 0.537 | CNN-RNN | 0.664 |
| 5 years | CNN-LSTM | 0.517 | CNN-LSTM | 0.661 |

Figure 31 - Daily Training Best Models

This trend could be due to the fact that a smaller training set would make the networks more sensitive to the volatility of the data, which would be beneficial to short-term predictions, while a larger dataset would allow the model to learn the trend and larger features of the data more accurately, lending itself towards longer-term predictions. This trend is present for both Bitcoin and Ethereum prediction results.

BTC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Used | Hourly Winner | Score | Daily Winner | Score |
| 500 hours | RNN | 0.533 | GRU | 0.55 |
| 1000 hours | LSTM | 0.506 | CNN-RNN | 0.527 |
| 1500 hours | RNN2 | 0.516 | CNN-RNN | 0.559 |
| 2000 hours | CNN-LSTM | 0.521 | CNN-LSTM | 0.562 |

ETH

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Used | Hourly Winner | Score | Daily Winner | Score |
| 500 hours | RNN | 0.517 | CNN-RNN | 0.458 |
| 1000 hours | RNN2 | 0.551 | CNN-LSTM | 0.6 |
| 1500 hours | RNN | 0.522 | CNN-RNN | 0.581 |
| 2000 hours | RNN2 | 0.521 | LSTM | 0.576 |

Figure 32 - Hourly Training Winners

The CNN-RNN and CNN-LSTM hybrid models are shown to be the most effective at middling-length predictions, scoring the highest out of all the models when using a 1-day win-rate with both hourly data and daily data (achieving 56.2% and 67.5% respectively). By contrast, Recurrent Neural Networks appear to be best-suited to predictions on both extremes, as the Simple RNN and LSTM models consistently return the best results for 1-week win-rates when using daily data, while outperforming other models with hourly data when utilising a 1-hour win-rate. These hybrid models are also strongly favoured by the Ethereum trained models using daily price data.

The results obtained do not suggest that increasing the depth of these models results in increased performance, as the deeper architectures generally obtained similar results to the models with fewer layers, but the simpler networks actually obtained the best win-rates.

Overall, the deep neural networks trained with daily data performed better, which could be due to the hourly data simply being too volatile and unpredictable.

Higher scores were achieved on BTC daily data than with ETH, which could be due to a number of factors. As mentioned in the Data Overview and Literature Review, Bitcoin’s strong relationship with public interest and sentiment data could have caused it to be more predictable in the testing period, as it experienced a steady decline in value due to a number of factors (COVID-19, the war in Ukraine, etc.)

These results show that the best model to use depends on the cryptocurrency being analysed and the amount of data used for training, however the overall highest-performing models were the GRU and LSTM, achieving a win-rates of almost 80% when trained with 1 year of historical Bitcoin price data using a 5-day prediction horizon. This is in line with my expectations and my review of the literature, which suggested that RNN models (particularly LSTM) are best suited for time-series analysis tasks such as this, while other research promotes the use of GRU-based networks.

For Ethereum, the most consistent winning models were the CNN hybrid networks, suggesting that further experimentation with hybrid models and deeper neural networks could be beneficial to performance.

Data Smoothing Experimentation (shown in the appendix) reveals that these technical analysis techniques are not beneficial to predictions when applied to the data before training the neural networks, and they in fact perform better when using just the raw data.

In general, the Exponential Moving Average returns better results than the Simple Moving Average for hourly data, while the SMA is more effective with daily data. Despite this, both data smoothing methods return results that are either the same or worse than the original result.

Overall, the final highest performing model was the Gated Recurrent Unit network (GRU), achieving a very high prediction win-rate of 77.9%. This was achieved using a 5-day win-rate, analysing the last 1 year of historical daily Bitcoin price data. In the figure below, the predictions of this model are shown mapped against the true historical price values and can be seen to closely follow the trend of the actual data.

Chart, line chart

Description automatically generated

Figure 33 - Final Best Model - GRU - Predictions vs Targets

# 9. Conclusions and Future Work

The results achieved by the models are almost always above 50%, and are often significantly higher when using daily data, reaching up to 69.4%. This shows that there is a lot of promise in the application of Machine Learning techniques to predict the price changes of cryptocurrencies. However, the low performance of many of the models shows that this is limited by the nature of the data as being highly volatile and driven by public interest.

This suggests that a project with wider scope, including Sentiment Analysis in tandem with price data would be more effective in predicting the future trends of cryptocurrency prices. This is also supported by my research in the literature review, which showed that there is a correlation between price changes in Bitcoin and interest reflected by data from Google Trends.

The fact that significantly higher results were achieved with longer prediction horizons for daily price data suggests that predictions even further into the future, such as 1 month, could prove even more promising, possibly as this allows the problem of short-term volatility to be largely circumvented in favour of detection of larger trends in the data. Further testing with different cryptocurrencies and prediction horizons would help to establish this connection further.

The high performance of the LSTM and GRU neural nets as the models that obtained the best results (≈80%), in addition to the high performance of the simple RNN and CNN-LSTM/CNN-RNN hybrid models, supports the hypothesis and relevant studies that suggest that RNN/LSTM models are best suited for this task due to the time-series nature of the data, however this is also shown to not always be the case, the exact outcome depending on the variables and data applied to the model.

The slight increase in the depth of the neural networks did not yield improved results, and the deeper networks largely obtained the same results as their shallower counterparts. While this could suggest that increased depth does not help in the identification of features within such data, this could require further experimentation with significantly deeper networks in order to ascertain if they can capture higher-dimensional relationships better.

Experimentation with SMA and EMA data smoothing shows that these methods are not useful for this application, and actually worsen results when applied to the raw data.

Given more time and wider scope, further investigation of the actual profitability of these methods could be conducted through the implementation of automated trading (simulated or real) using an exchange based on the forecasts produced by these models. This would allow for a more complete picture of the efficacy of these strategies, as while prediction accuracies just above 50% for some of the worse-performing models suggest that they can predict changes in the price somewhat, these would be unlikely to turn a profit in a cryptocurrency exchange and are therefore of limited use.

Further improvement of the performance of these models could be undertaken through experimentation with different network architectures and thorough examination of the effects of the use of different hyperparameters. A more comprehensive study on the overall application of Machine Learning in this type of task through the inclusion of Sentiment Analysis and non-Deep Learning methods, such as Random Decision Forests, Genetic Algorithms and Support Vector Machines.

# 10. Evaluation

1. **Acquire historical price data for multiple cryptocurrencies**
2. **Research techniques used in normal financial price forecasting**
3. **Apply data pre-processing techniques and normalisation**
4. **Apply and adapt Machine Learning and financial price forecasting techniques to the prediction of cryptocurrency prices**
5. **Generate and train models of different types for comparison**
6. **Find and compare the best-performing models across different cryptocurrencies and datasets**
7. **Present and explain the results**

Overall, I believe that this study fulfils its objectives of assessing the impact of changing the dataset, neural network, and data smoothing applied on the predictive performance of the models.

Data pre-processing and normalisation techniques are successfully applied to allow proper training of the models with the raw price data obtained with the CryptoCompare API.

Deep Learning and Data Smoothing methods are successfully adapted and applied to the task, returning comprehensive results to compare the efficacy of each model and the impact of data smoothing, which was shown to have a negative effect on results.

The fact that the best model for predictions was reliant on the specific combination of dataset and win-rate is valuable information and shows that there are multiple valid methods for price prediction, however the results support the idea that RNN/LSTM and GRU models are best suited to this sort of time-series analysis.

Further work can be undertaken to improve performance; however, this study succeeds in its aim of assessing whether Deep Learning models can be used in cryptocurrency price prediction, as well as presenting the differences in performance between models in different situations.

This study is also comprehensive in its inclusion of elements not covered in much of the existing literature, such as varying datasets and dataset sizes, comparing the use of hourly price data to daily price data, and the inclusion of Ethereum for comparison with performance using Bitcoin, as most of the literature focuses heavily on the first and most popular cryptocurrency.

These conclusions are drawn through extensive analysis of the results achieved, and have shown what future work is required in order to further research into this topic (i.e. the inclusion of non-Deep Learning ML methods, Sentiment Analysis, longer prediction horizons and additional hyperparameter experimentation)

# 11. Statement of Ethics

## 11.1 Overview

Ethics, as a set of principles, in the world of computer science, serve to regulate the use of technologies in order to ensure that they do not infringe upon the rights and privacies of society or individuals and that they follow a certain moral standard.

In this section we shall assess this study from several perspectives such as legal, ethical and social to ensure this standard, focusing on issues of copyright, privacy, intellectual property and the impact of the project on society at large, in addition to its contributions.

It is important to consider projects ethically as this viewpoint extends beyond just legality; while ethical codes can be ambiguous and subjective, they are important to guarantee that the spirit of the project is not malicious and recognise the potential repercussions of the end result, whether it be the development of a technology or the results of a study.

This study respects privacy, legality and confidentiality concerns in a number of ways.

## 11.2 Informed Consent

Consent is a key feature of ethical discussion and is very important so that any participants are aware of and have agreed to all details of their participation, the use of their personal information, and any information that may affect them as subjects within the study. However, this does not apply in this case as this study does not require any volunteers, nor does it use any non-public personal data. All data for the training of the neural networks is sourced through the use of an online cryptocurrency API, CryptoCompare, which has access to historical price data for many cryptocurrencies.

There is no active human participation in this study, and the only possible interaction is through the visualised outputs provided by the models and their performance metrics.

## 11.3 Copyright & Intellectual Property

All code used in the project was made by myself and libraries used in the implementation of the study are open source and referenced. In addition, all sources used are referenced in order to avoid plagiarism and copyright infringement.

None of the data used by or generated by the project is stored for future use.

## 11.4 Confidentiality of Data

The Data Protection Act (DPA) was introduced to ensure that the handling of personal and sensitive data is up to ethical standards by introducing into law certain requirements in the processing and storage of such data. This addresses Confidentiality of Data as one of the key ethical considerations in the world of computing.

It requires that all stored data is maintained so that it is accurate and up to date to the extent that is possible. The data stored and the manner of its storage cannot violate a person’s rights or privacy, so they must be aware of what data is stored and have access to the data themselves, in addition to having the ability to stop or prevent further processing of their data.

Data is required to be used fairly and lawfully and kept for no longer than necessary.

This study does not use, access. or store any personal or sensitive information that would come under the purview of the DPA. There is no user input nor any interaction with the system outside of receiving the outputs.

## 11.5 Social Responsibility

Any ethical assessment of a project must evaluate the contributions and impact of the project to society at large. This ensures that the work cannot be used in an unethical manner in order to do harm to the public, and instead will be designed, implemented, and used in a socially responsible manner which takes into account moral, legal and social concerns.

This study respects the principles of data confidentiality, copyright, and informed consent. All data used is publicly available, obtained using open-source APIs and referenced. This gives credit to others for their work and is according to intellectual property laws and conventions.

## 11.6 BCS Codes of Conduct and Ethics

Due to the nature of the project, few points covered in these codes apply to this study, however as a member of BCS I have taken into account the spirit and intention behind these codes to ensure that no harm or unethical wrongdoing will result from my work. I have worked to ensure that sources used in research were referenced and that libraries used in the implementation are open-source and referenced for credit so as not to infringe on their intellectual property.

# References

A Gholamy, V. K. O. K., 2018. *Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation.* [Online]   
Available at: https://scholarworks.utep.edu/cs\_techrep/1209/

A. Hayes, I., 2022. *Technical Analysis.* [Online]   
Available at: https://www.investopedia.com/terms/t/technicalanalysis.asp

A. Hayes, I., 2022. *What Happens to Bitcoin After All 21 Million Are Mined?.* [Online]   
Available at: https://www.investopedia.com/tech/what-happens-bitcoin-after-21-million-mined/

A. Hayes, I., 2022. *What is a Blockchain?.* [Online]   
Available at: https://www.investopedia.com/terms/b/blockchain.asp

A. Hayes, I., 2022. *What Is a Time Series?.* [Online]   
Available at: https://www.investopedia.com/terms/t/timeseries.asp

A. Narayanan, J. B. E. F. A. M. S. G., 2016. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction.* s.l.:s.n.

B. Muller, J. R. M. T. S., 1995. *Neural Networks: An Introduction.* [Online]   
Available at: https://books.google.co.uk/books?hl=en&lr=&id=EFUzMYjOXk8C&oi=fnd&pg=PA3&dq=neural+networks&ots=WkArMiNoXE&sig=TU536f8dbe9sR5VdkzVJDih3Wps&redir\_esc=y#v=onepage&q=neural%20networks&f=false

BBC, 2021. *China declares all crypto-currency transactions illegal.* [Online]   
Available at: https://www.bbc.co.uk/news/technology-58678907

Bitcoin, n.d. *How Bitcoin Exchange Works.* [Online]   
Available at: https://www.bitcoin.com/get-started/how-bitcoin-exchange-works/

Brownlee, J., 2017. *A Gentle Introduction to Long Short-Term Memory Networks by the Experts.* [Online]   
Available at: https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/

Brownlee, J., 2017. *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning.* [Online]   
Available at: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

Brownlee, J., 2021. *Gradient Descent Optimization With Nadam From Scratch.* [Online]   
Available at: https://machinelearningmastery.com/gradient-descent-optimization-with-nadam-from-scratch/

Brownlee, J., 2021. *Gradient Descent With RMSProp from Scratch.* [Online]   
Available at: https://machinelearningmastery.com/gradient-descent-with-rmsprop-from-scratch/#:~:text=Root%20Mean%20Squared%20Propagation%2C%20or,step%20size%20for%20each%20parameter.

C. Mitchell, I., 2021. *Average Directional Index (ADX).* [Online]   
Available at: https://www.investopedia.com/terms/a/adx.asp

CIN7, 2019. *Moving Average.* [Online]   
Available at: https://www.cin7.com/industry-terms/moving-average/

Cory Mitchell, I., 2021. *Simple vs. Exponential Moving Averages: What's the Difference?.* [Online]   
Available at: https://www.investopedia.com/articles/trading/10/simple-exponential-moving-averages-compare.asp

Credit Suisse, n.d. *Technical Analysis - Explained.* [Online]   
Available at: https://www.credit-suisse.com/pwp/pb/pb\_research/technical\_tutorial\_de.pdf

Databricks, n.d. *AdaGrad.* [Online]   
Available at: https://www.databricks.com/glossary/adagrad#:~:text=Adaptive%20Gradient%20Algorithm%20(Adagrad)%20is,incorporating%20knowledge%20of%20past%20observations.

Dirk G. Baur, K. H. A. D. L., 2018. *Bitcoin: Medium of exchange or speculative assets?.* [Online]   
Available at: https://www.sciencedirect.com/science/article/pii/S1042443117300720?casa\_token=JwYqUa5LdzIAAAAA:fJT3JzMaMQhEr\_B40W8ZuVtVNRJc6jvjEXjiCcfZ8zr-mCw3qDNScFVPwoXGNsLUYd4bjBee

Doshi, S., 2019. *Various Optimization Algorithms For Training Neural Network.* [Online]   
Available at: https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6

Education, I. C., 2020. *Recurrent Neural Networks.* [Online]   
Available at: https://www.ibm.com/cloud/learn/recurrent-neural-networks

Flynn, J., 2022. *HOW MANY BUSINESSES ACCEPT BITCOIN? [2022]: 21 IMPORTANT BITCOIN STATISTICS.* [Online]   
Available at: https://www.zippia.com/advice/how-many-businesses-accept-bitcoin/#:~:text=Approximately%2015%2C174%20businesses%20worldwide%20accept,businesses%20accept%20cryptocurrency%20as%20payment.

G. Karame, E. A. S. C., 2012. *Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin.* [Online]   
Available at: https://eprint.iacr.org/2012/248.pdf

Genesis Mining, n.d. *LITECOIN MINING: A HELPFUL GUIDE.* [Online]   
Available at: https://www.genesis-mining.com/litecoin-mining-guide#:~:text=Litecoin's%20block%20generation%20time%20is,algorithm%3B%20they%20are%20very%20different.

Google Developers, n.d. *Reducing Loss: Learning Rate.* [Online]   
Available at: https://developers.google.com/machine-learning/crash-course/reducing-loss/learning-rate

Howarth, J., 2022. *How Many Cryptocurrencies are There In 2022?.* [Online]   
Available at: https://explodingtopics.com/blog/number-of-cryptocurrencies

Howarth, J., 2022. *How Many People Own Bitcoin? 95 Blockchain Statistics (2022).* [Online]   
Available at: https://explodingtopics.com/blog/blockchain-stats

IBM Cloud Education, 2020. *Deep Learning.* [Online]   
Available at: https://www.ibm.com/cloud/learn/deep-learning

IBM, 2020. *What are Neural Networks?.* [Online]   
Available at: https://www.ibm.com/cloud/learn/neural-networks

IBM, 2021. *Overfitting.* [Online]   
Available at: https://www.ibm.com/cloud/learn/overfitting#:~:text=When%20the%20model%20memorizes%20the,that%20it%20was%20intended%20for.

Investopedia, 2022. *10 Important Cryptocurrencies Other Than Bitcoin.* [Online]   
Available at: https://www.investopedia.com/tech/most-important-cryptocurrencies-other-than-bitcoin/10 Important Cryptocurrencies Other Than Bitcoin

Investopedia, 2022. *Beginner’s Guide to the Bloomberg Terminal.* [Online]   
Available at: https://www.investopedia.com/articles/professionaleducation/11/bloomberg-terminal.asp#:~:text=For%20a%20standard%20license%2C%20a,month%2C%20or%20%2424%2C000%20per%20year.

J. Fernando, I., 2022. *Moving Average (MA): Purpose, Uses, and Examples.* [Online]   
Available at: https://www.investopedia.com/terms/m/movingaverage.asp

J. Fernando, I., 2022. *Relative Strength Index (RSI) Indicator Explained With Formula.* [Online]   
Available at: https://www.investopedia.com/terms/r/rsi.asp#:~:text=The%20relative%20strength%20index%20(RSI)%20is%20a%20momentum%20indicator%20used,scale%20of%20zero%20to%20100.

Jaquart, P., 2021. *Short-term bitcoin market prediction via machine learning.* [Online]   
Available at: https://www.sciencedirect.com/science/article/pii/S2405918821000027

K. Brush, T., 2021. *asymmetric cryptography (public key cryptography).* [Online]   
Available at: https://www.techtarget.com/searchsecurity/definition/asymmetric-cryptography#:~:text=Asymmetric%20cryptography%2C%20also%20known%20as,from%20unauthorized%20access%20or%20use.

K. Cho, e. a., 2014. *Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation.* [Online]   
Available at: https://arxiv.org/pdf/1406.1078v3.pdf

Kostadinov, S., 2017. *Understanding GRU Networks.* [Online]   
Available at: https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be

M. Travis, R., 2017. *Ripple: The Most (Demonstrably) Scalable Blockchain.* [Online]   
Available at: http://highscalability.com/blog/2017/10/2/ripple-the-most-demonstrably-scalable-blockchain.html

Mejova, Y., 2009. *Sentiment Analysis: An Overview.* [Online]   
Available at: https://d1wqtxts1xzle7.cloudfront.net/3243118/CompsYelenaMejova-with-cover-page-v2.pdf?Expires=1661258037&Signature=MHAqbTfc0r7jOaInjjHXK0Z-~oIc6CAyTCDBTf91aDOxnFO7HZIiOhHLsHPwvwRdu13YzRc5rXbhaJ~31Hrx7N4YRX9LXxaSQL1Z77m6Rt9EZ~zyDbb2AX~RmtB~~xCgFFN414z0bkx

N Alsalami, B. Z., 2019. *SoK: A systematic study of anonymity in cryptocurrencies.* [Online]   
Available at: https://ieeexplore.ieee.org/abstract/document/8937681?casa\_token=19TKXf11o4sAAAAA:tUhwvh1sYC2QXhcc4jKzdYh3m9V3HWYiotY8CyjH7OPzMGx85WIYp0XXbJV7p3PRe\_3cUSuxuQ

Nakamoto, S., 2009. *Bitcoin: A Peer-to-Peer Electronic Cash System.* [Online]   
Available at: https://www.ussc.gov/sites/default/files/pdf/training/annual-national-training-seminar/2018/Emerging\_Tech\_Bitcoin\_Crypto.pdf

Pintelas, E., 2020. *Investigating the Problem of Cryptocurrency Price Prediction: A Deep Learning Approach.* [Online]   
Available at: https://link.springer.com/chapter/10.1007/978-3-030-49186-4\_9#Sec3

Pramoditha, R., 2021. *The Concept of Artificial Neurons (Perceptrons) in Neural Networks.* [Online]   
Available at: https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc

R. Browne, C., 2021. *Bitcoin Falls after Elon Musk posts breakup meme.* [Online]   
Available at: https://www.cnbc.com/2021/06/04/bitcoin-falls-after-elon-musk-tweets-breakup-meme.html

S. Charandabi, K. K., 2021. *Prediction of Cryptocurrency Price Index Using Artificial Neural Networks: A Survey of the Literature.* [Online]   
Available at: https://ejbmr.org/index.php/ejbmr/article/view/1138/603

S. Sharma, S. S., 2020. *Activation Functions in Neural Networks.* [Online]   
Available at: https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf

Saha, S., 2018. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.* [Online]   
Available at: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Sharma, S., 2017. *Epoch vs Batch Size vs Iterations.* [Online]   
Available at: https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9

Sin, E., 2017. *Bitcoin price prediction using ensembles of neural networks.* [Online]   
Available at: https://ieeexplore.ieee.org/abstract/document/8393351

Smuts, N., 2018. *What Drives Cryptocurrency Prices?: An Investigation of Google Trends and Telegram Sentiment.* [Online]   
Available at: https://dl.acm.org/doi/abs/10.1145/3308897.3308955?casa\_token=8yV1Sa-7\_nIAAAAA:ynpo4ldaDxg0U62jcWP97schZIW\_tHcdYwLmfqj1AK3jQUYwRVZDiyS-BmNEicbSvPR7c7O2YWvI

Soni, P., 2022. *Machine Learning Approaches in Stock Price Prediction: A Systematic Review.* [Online]   
Available at: https://iopscience.iop.org/article/10.1088/1742-6596/2161/1/012065/pdf

Springer Nature, n.d. *Open Economies Review.* [Online]   
Available at: https://www.researchgate.net/journal/Open-Economies-Review-1573-708X

Srivastava, N., 2014. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting.* [Online]   
Available at: https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm\_content=buffer79b43&utm\_medium=social&utm\_source=twitter.com&utm\_campaign=buffer,

T Mitchell, B. B. G. D. T. D. P. R. A. W., 1990. *Machine Learning - Annual Review of Computer Science.* [Online]   
Available at: https://www.annualreviews.org/doi/abs/10.1146/annurev.cs.04.060190.002221?journalCode=arcompsci

T. Segal, I., 2021. *Fundamental Analysis.* [Online]   
Available at: https://www.investopedia.com/terms/f/fundamentalanalysis.asp

The Economic Times, 2018. *Cryptocurrency exchanges are raking in billions of dollars.* [Online]   
Available at: https://economictimes.indiatimes.com/markets/stocks/news/cryptocurrency-exchanges-are-raking-in-billions-of-dollars/articleshow/63182900.cms

Trading View, n.d. *Exponential Moving Average.* [Online]   
Available at: https://www.tradingview.com/support/solutions/43000592270-exponential-moving-average/

Vasilis Kostakis, C. G., 2014. *The (A)Political Economy of Bitcoin.* [Online]   
Available at: https://www.triple-c.at/index.php/triplec/article/view/606

Velde, F., 2013. *Bitcoin: A Primer.* [Online]   
Available at: http://blog.philippe-poisse.eu/public/monnaie\_locale/bitcoin/cfldecember2013\_317.pdf

Verma, S., 2019. *Understanding 1D and 3D Convolution Neural Network | Keras.* [Online]   
Available at: https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610

Zhang, Y., 2010. *New Advancements in Machine Learning.* [Online]   
Available at: https://books.google.co.uk/books?hl=en&lr=&id=XAqhDwAAQBAJ&oi=fnd&pg=PA19&dq=types+of+machine+learning&ots=r2Lj9UxiJs&sig=p3ny1ttP4GtrLX-\_o-MB3SORT4M#v=onepage&q=types%20of%20machine%20learning&f=false

# Libraries

## Tensorflow: <https://www.tensorflow.org/>

Machine Learning Platform for Python which allows the implementation of ML techniques.

Offers tensor data structure to input data into models for training.

## Keras: <https://keras.io/>

A Tensorflow API which implements a Python interface for working with Neural Networks.

Simplifies the implementation and distribution of neural networks, particularly deep learning models. Used to create and train the neural networks, in addition to providing the optimizers, activation functions and loss functions used in each model.

## Numpy: <https://numpy.org/>

Offers functions and data structures, mathematical functions, random number generators. Numpy arrays are used to store the short sequences extracted from the data.

## Pandas: <https://pandas.pydata.org/>

Library for data manipulation and analysis, which introduces certain data structures, including the Dataframe in which each dataset is stored, and the Series which is used in plotting the model predictions.

## Requests:

Built-in python library to streamline HTTP protocol requests. Used to access API for historical price data

## JSON:

Built-in python library to work with data stored in JSON format. Used to access the dataset once it is obtained from the API

## Matplotlib PyPlot: <https://matplotlib.org/>

Allows for the visualisation of data in Python. Used to generate graphs for evaluation.

## Scikit-Learn: <https://scikit-learn.org/stable/>

Provides metrics for evaluation of data. Used to evaluate model performance with Mean Absolute Error, Mean Squared Error and Mean Absolute Percentage Error.

# Appendix

## Sage Ethics Form

Graphical user interface, application

Description automatically generated

Text

Description automatically generated with medium confidence

Table

Description automatically generated

Graphical user interface, table

Description automatically generated Graphical user interface

Description automatically generated with medium confidence Table

Description automatically generated Graphical user interface, table

Description automatically generated Graphical user interface, application, table

Description automatically generated

Table

Description automatically generated with medium confidence Graphical user interface, application, table

Description automatically generated Table

Description automatically generated Table

Description automatically generated Table

Description automatically generated

# Daily Training Results – 6 months

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bitcoin | MAE | MSE | MAPE | 1 Day WR | 3 Day WR | 5 Day WR | 7 Day WR |
| RNN | 0.06 | 0.01 | 1.86 | 0.55 | 0.625 | 0.6 | 0.5 |
| LSTM | 0.03 | 0.01 | 1.33 | 0.525 | 0.65 | 0.675 | 0.625 |
| CNN | 0.06 | 0.01 | 7.45 | 0.425 | 0.35 | 0.3 | 0.35 |
| GRU | 0.03 | 0.01 | 1.05 | 0.55 | 0.625 | 0.675 | 0.625 |
| CNNRNN | 0.03 | 0.01 | 2.09 | 0.675 | 0.65 | 0.65 | 0.55 |
| CNNLSTM | 0.03 | 0.01 | 0.98 | 0.525 | 0.65 | 0.7 | 0.625 |
| RNN2 | 0.06 | 0.01 | 2.44 | 0.575 | 0.675 | 0.625 | 0.55 |
| LSTM2 | 0.03 | 0.01 | 1.33 | 0.525 | 0.65 | 0.675 | 0.625 |
| CNN2 | 0.06 | 0.01 | 7.45 | 0.425 | 0.35 | 0.3 | 0.35 |
| GRU2 | 0.03 | 0.01 | 1.05 | 0.55 | 0.625 | 0.675 | 0.625 |
| CNNRNN2 | 0.03 | 0.01 | 2.09 | 0.675 | 0.65 | 0.65 | 0.55 |
| CNNLSTM2 | 0.03 | 0.01 | 0.98 | 0.525 | 0.65 | 0.7 | 0.625 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ethereum | MAE | MSE | MAPE | 1 Day WR | 3 Day WR | 5 Day WR | 7 Day WR |
| RNN | 0.09 | 0.01 | 3.79 | 0.425 | 0.525 | 0.65 | 0.5 |
| LSTM | 0.05 | 0.01 | 3.14 | 0.575 | 0.65 | 0.65 | 0.5 |
| CNN | 0.08 | 0.01 | 15.17 | 0.475 | 0.325 | 0.4 | 0.475 |
| GRU | 0.05 | 0.01 | 1.07 | 0.65 | 0.65 | 0.675 | 0.525 |
| CNNRNN | 0.05 | 0.01 | 4.24 | 0.55 | 0.575 | 0.65 | 0.475 |
| CNNLSTM | 0.05 | 0.01 | 5.99 | 0.575 | 0.65 | 0.65 | 0.475 |
| RNN2 | 0.06 | 0.01 | 4.4 | 0.45 | 0.6 | 0.575 | 0.525 |
| LSTM2 | 0.05 | 0.01 | 3.14 | 0.575 | 0.65 | 0.65 | 0.5 |
| CNN2 | 0.08 | 0.01 | 15.17 | 0.475 | 0.325 | 0.4 | 0.475 |
| GRU2 | 0.05 | 0.01 | 1.07 | 0.65 | 0.65 | 0.675 | 0.525 |
| CNNRNN2 | 0.05 | 0.01 | 4.24 | 0.55 | 0.575 | 0.65 | 0.475 |
| CNNLSTM2 | 0.05 | 0.01 | 5.99 | 0.575 | 0.65 | 0.65 | 0.475 |

# Daily Training Results – 1 year

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bitcoin | MAE | MSE | MAPE | 1 Day WR | 3 Day WR | 5 Day WR | 7 Day WR |
| RNN | 0.05 | 0.01 | 2.92 | 0.523 | 0.721 | 0.709 | 0.651 |
| LSTM | 0.05 | 0.01 | 1.4 | 0.535 | 0.674 | 0.767 | 0.674 |
| CNN | 0.06 | 0.01 | 9.21 | 0.523 | 0.605 | 0.674 | 0.605 |
| GRU | 0.04 | 0.01 | 1.57 | 0.512 | 0.674 | 0.779 | 0.686 |
| CNNRNN | 0.05 | 0.01 | 1.68 | 0.581 | 0.628 | 0.721 | 0.674 |
| CNNLSTM | 0.04 | 0.01 | 3.84 | 0.5 | 0.663 | 0.756 | 0.674 |
| RNN2 | 0.05 | 0.01 | 6.39 | 0.43 | 0.698 | 0.733 | 0.651 |
| LSTM2 | 0.05 | 0.01 | 1.4 | 0.535 | 0.674 | 0.767 | 0.674 |
| CNN2 | 0.06 | 0.01 | 9.21 | 0.523 | 0.605 | 0.674 | 0.605 |
| GRU2 | 0.04 | 0.01 | 1.57 | 0.512 | 0.674 | 0.779 | 0.686 |
| CNNRNN2 | 0.05 | 0.01 | 1.68 | 0.581 | 0.628 | 0.721 | 0.674 |
| CNNLSTM2 | 0.04 | 0.01 | 3.84 | 0.5 | 0.663 | 0.756 | 0.674 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ethereum | MAE | MSE | MAPE | 1 Day WR | 3 Day WR | 5 Day WR | 7 Day WR |
| RNN | 0.07 | 0.01 | 2.25 | 0.5 | 0.651 | 0.698 | 0.605 |
| LSTM | 0.07 | 0.01 | 6.55 | 0.512 | 0.605 | 0.709 | 0.605 |
| CNN | 0.11 | 0.01 | 377.44 | 0.419 | 0.453 | 0.326 | 0.326 |
| GRU | 0.07 | 0.01 | 2.35 | 0.535 | 0.674 | 0.698 | 0.605 |
| CNNRNN | 0.07 | 0.01 | 2.32 | 0.57 | 0.628 | 0.674 | 0.593 |
| CNNLSTM | 0.07 | 0.01 | 2.21 | 0.5 | 0.616 | 0.709 | 0.616 |
| RNN2 | 0.07 | 0.01 | 7.84 | 0.57 | 0.616 | 0.744 | 0.605 |
| LSTM2 | 0.07 | 0.01 | 6.55 | 0.512 | 0.605 | 0.709 | 0.605 |
| CNN2 | 0.11 | 0.01 | 377.44 | 0.419 | 0.453 | 0.326 | 0.326 |
| GRU2 | 0.07 | 0.01 | 2.35 | 0.535 | 0.674 | 0.698 | 0.605 |
| CNNRNN2 | 0.07 | 0.01 | 2.32 | 0.57 | 0.628 | 0.674 | 0.593 |
| CNNLSTM2 | 0.07 | 0.01 | 2.21 | 0.5 | 0.616 | 0.709 | 0.616 |

# Daily Training Results – 3 years

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bitcoin | MAE | MSE | MAPE | 1 Day WR | 3 Day WR | 5 Day WR | 7 Day WR |
| RNN | 0.04 | 0.01 | 2.07 | 0.493 | 0.642 | 0.728 | 0.642 |
| LSTM | 0.04 | 0.01 | 7.9 | 0.504 | 0.679 | 0.728 | 0.694 |
| CNN | 0.06 | 0.01 | 15.39 | 0.496 | 0.455 | 0.571 | 0.522 |
| GRU | 0.04 | 0.01 | 2.76 | 0.504 | 0.668 | 0.731 | 0.69 |
| CNNRNN | 0.04 | 0.01 | 5.39 | 0.507 | 0.66 | 0.765 | 0.657 |
| CNNLSTM | 0.04 | 0.01 | 2.24 | 0.519 | 0.634 | 0.754 | 0.679 |
| RNN2 | 0.04 | 0.01 | 2.56 | 0.519 | 0.668 | 0.709 | 0.683 |
| LSTM2 | 0.04 | 0.01 | 7.9 | 0.504 | 0.679 | 0.728 | 0.694 |
| CNN2 | 0.06 | 0.01 | 15.39 | 0.496 | 0.455 | 0.571 | 0.522 |
| GRU2 | 0.04 | 0.01 | 2.76 | 0.504 | 0.668 | 0.731 | 0.69 |
| CNNRNN2 | 0.04 | 0.01 | 5.39 | 0.507 | 0.66 | 0.765 | 0.657 |
| CNNLSTM2 | 0.04 | 0.01 | 2.24 | 0.519 | 0.634 | 0.754 | 0.679 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ethereum | MAE | MSE | MAPE | 1 Day WR | 3 Day WR | 5 Day WR | 7 Day WR |
| RNN | 0.06 | 0.01 | 2.32 | 0.504 | 0.601 | 0.653 | 0.653 |
| LSTM | 0.06 | 0.01 | 1.84 | 0.534 | 0.649 | 0.679 | 0.649 |
| CNN | 0.09 | 0.01 | 87.53 | 0.511 | 0.526 | 0.481 | 0.448 |
| GRU | 0.05 | 0.01 | 1.57 | 0.515 | 0.657 | 0.694 | 0.634 |
| CNNRNN | 0.06 | 0.01 | 1.89 | 0.519 | 0.601 | 0.679 | 0.664 |
| CNNLSTM | 0.06 | 0.01 | 1.58 | 0.537 | 0.638 | 0.687 | 0.649 |
| RNN2 | 0.06 | 0.01 | 15.65 | 0.47 | 0.634 | 0.649 | 0.638 |
| LSTM2 | 0.06 | 0.01 | 1.84 | 0.534 | 0.649 | 0.679 | 0.649 |
| CNN2 | 0.09 | 0.01 | 87.53 | 0.511 | 0.526 | 0.481 | 0.448 |
| GRU2 | 0.05 | 0.01 | 1.57 | 0.515 | 0.657 | 0.694 | 0.634 |
| CNNRNN2 | 0.06 | 0.01 | 1.89 | 0.519 | 0.601 | 0.679 | 0.664 |
| CNNLSTM2 | 0.06 | 0.01 | 1.58 | 0.537 | 0.638 | 0.687 | 0.649 |

# Daily Training Results – 5 years

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bitcoin | MAE | MSE | MAPE | 1 Day WR | 3 Day WR | 5 Day WR | 7 Day WR |
| RNN | 0.04 | 0.01 | 1.83 | 0.488 | 0.63 | 0.716 | 0.672 |
| LSTM | 0.04 | 0.01 | 4.36 | 0.483 | 0.636 | 0.705 | 0.694 |
| CNN | 0.06 | 0.01 | 10.99 | 0.486 | 0.534 | 0.534 | 0.523 |
| GRU | 0.04 | 0.01 | 2.43 | 0.488 | 0.654 | 0.703 | 0.692 |
| CNNRNN | 0.04 | 0.01 | 2.38 | 0.499 | 0.612 | 0.716 | 0.674 |
| CNNLSTM | 0.04 | 0.01 | 2.77 | 0.494 | 0.601 | 0.721 | 0.667 |
| RNN2 | 0.04 | 0.01 | 3.34 | 0.463 | 0.632 | 0.714 | 0.678 |
| LSTM2 | 0.04 | 0.01 | 4.36 | 0.483 | 0.636 | 0.705 | 0.694 |
| CNN2 | 0.06 | 0.01 | 10.99 | 0.486 | 0.534 | 0.534 | 0.523 |
| GRU2 | 0.04 | 0.01 | 2.43 | 0.488 | 0.654 | 0.703 | 0.692 |
| CNNRNN2 | 0.04 | 0.01 | 2.38 | 0.499 | 0.612 | 0.716 | 0.674 |
| CNNLSTM2 | 0.04 | 0.01 | 2.77 | 0.494 | 0.601 | 0.721 | 0.667 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ethereum | MAE | MSE | MAPE | 1 Day WR | 3 Day WR | 5 Day WR | 7 Day WR |
| RNN | 0.05 | 0.01 | 1.95 | 0.488 | 0.625 | 0.67 | 0.641 |
| LSTM | 0.05 | 0.01 | 1.81 | 0.486 | 0.625 | 0.681 | 0.643 |
| CNN | 0.08 | 0.01 | 19.42 | 0.457 | 0.521 | 0.548 | 0.488 |
| GRU | 0.05 | 0.01 | 1.71 | 0.497 | 0.632 | 0.678 | 0.643 |
| CNNRNN | 0.06 | 0.01 | 2.54 | 0.475 | 0.612 | 0.665 | 0.652 |
| CNNLSTM | 0.06 | 0.01 | 1.64 | 0.517 | 0.594 | 0.661 | 0.661 |
| RNN2 | 0.05 | 0.01 | 1.29 | 0.488 | 0.627 | 0.627 | 0.645 |
| LSTM2 | 0.05 | 0.01 | 1.81 | 0.486 | 0.625 | 0.681 | 0.643 |
| CNN2 | 0.08 | 0.01 | 19.42 | 0.457 | 0.521 | 0.548 | 0.488 |
| GRU2 | 0.05 | 0.01 | 1.71 | 0.497 | 0.632 | 0.678 | 0.643 |
| CNNRNN2 | 0.06 | 0.01 | 2.54 | 0.475 | 0.612 | 0.665 | 0.652 |
| CNNLSTM2 | 0.06 | 0.01 | 1.64 | 0.517 | 0.594 | 0.661 | 0.661 |

# Hourly Training Results – 500 hours

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bitcoin | MAE | MSE | MAPE | 1 Hr WR | 6 Hr WR | 12 Hr WR | 1 Day WR |
| RNN | 0.03 | 0.01 | 1.0 | 0.533 | 0.658 | 0.592 | 0.508 |
| LSTM | 0.01 | 0.01 | 3.02 | 0.492 | 0.65 | 0.492 | 0.425 |
| CNN | 0.01 | 0.01 | 1.92 | 0.5 | 0.533 | 0.433 | 0.475 |
| GRU | 0.01 | 0.01 | 3.35 | 0.475 | 0.65 | 0.558 | 0.55 |
| CNNRNN | 0.01 | 0.01 | 2.32 | 0.5 | 0.625 | 0.517 | 0.492 |
| CNNLSTM | 0.01 | 0.01 | 8.36 | 0.467 | 0.725 | 0.508 | 0.433 |
| RNN2 | 0.01 | 0.01 | 1.48 | 0.5 | 0.608 | 0.492 | 0.433 |
| LSTM2 | 0.01 | 0.01 | 3.02 | 0.492 | 0.65 | 0.492 | 0.425 |
| CNN2 | 0.01 | 0.01 | 1.92 | 0.5 | 0.533 | 0.433 | 0.475 |
| GRU2 | 0.01 | 0.01 | 3.35 | 0.475 | 0.65 | 0.558 | 0.55 |
| CNNRNN2 | 0.01 | 0.01 | 2.32 | 0.5 | 0.625 | 0.517 | 0.492 |
| CNNLSTM2 | 0.01 | 0.01 | 8.36 | 0.467 | 0.725 | 0.508 | 0.433 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ethereum | MAE | MSE | MAPE | 1 Hr WR | 6 Hr WR | 12 Hr WR | 1 Day WR |
| RNN | 0.04 | 0.01 | 10.68 | 0.517 | 0.475 | 0.475 | 0.383 |
| LSTM | 0.01 | 0.01 | 5.5 | 0.483 | 0.567 | 0.517 | 0.417 |
| CNN | 0.02 | 0.01 | 1.82 | 0.458 | 0.492 | 0.5 | 0.367 |
| GRU | 0.02 | 0.01 | 3.08 | 0.45 | 0.575 | 0.55 | 0.408 |
| CNNRNN | 0.01 | 0.01 | 22.84 | 0.45 | 0.517 | 0.558 | 0.458 |
| CNNLSTM | 0.01 | 0.01 | 2.39 | 0.483 | 0.592 | 0.525 | 0.442 |
| RNN2 | 0.03 | 0.01 | 2.09 | 0.492 | 0.517 | 0.475 | 0.433 |
| LSTM2 | 0.01 | 0.01 | 5.5 | 0.483 | 0.567 | 0.517 | 0.417 |
| CNN2 | 0.02 | 0.01 | 1.82 | 0.458 | 0.492 | 0.5 | 0.367 |
| GRU2 | 0.02 | 0.01 | 3.08 | 0.45 | 0.575 | 0.55 | 0.408 |
| CNNRNN2 | 0.01 | 0.01 | 22.84 | 0.45 | 0.517 | 0.558 | 0.458 |
| CNNLSTM2 | 0.01 | 0.01 | 2.39 | 0.483 | 0.592 | 0.525 | 0.442 |

# Hourly Training Results – 1000 hours

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bitcoin | MAE | MSE | MAPE | 1 Hr WR | 6 Hr WR | 12 Hr WR | 1 Day WR |
| RNN | 0.03 | 0.01 | 1.61 | 0.49 | 0.531 | 0.469 | 0.445 |
| LSTM | 0.01 | 0.01 | 5.8 | 0.506 | 0.702 | 0.567 | 0.51 |
| CNN | 0.02 | 0.01 | 2.07 | 0.469 | 0.457 | 0.445 | 0.392 |
| GRU | 0.01 | 0.01 | 1.97 | 0.449 | 0.69 | 0.539 | 0.506 |
| CNNRNN | 0.01 | 0.01 | 3.21 | 0.433 | 0.633 | 0.535 | 0.527 |
| CNNLSTM | 0.01 | 0.01 | 1.63 | 0.473 | 0.714 | 0.576 | 0.522 |
| RNN2 | 0.02 | 0.01 | 2.42 | 0.457 | 0.539 | 0.49 | 0.469 |
| LSTM2 | 0.01 | 0.01 | 5.8 | 0.506 | 0.702 | 0.567 | 0.51 |
| CNN2 | 0.02 | 0.01 | 2.07 | 0.469 | 0.457 | 0.445 | 0.392 |
| GRU2 | 0.01 | 0.01 | 1.97 | 0.449 | 0.69 | 0.539 | 0.506 |
| CNNRNN2 | 0.01 | 0.01 | 3.21 | 0.433 | 0.633 | 0.535 | 0.527 |
| CNNLSTM2 | 0.01 | 0.01 | 1.63 | 0.473 | 0.714 | 0.576 | 0.522 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ethereum | MAE | MSE | MAPE | 1 Hr WR | 6 Hr WR | 12 Hr WR | 1 Day WR |
| RNN | 0.02 | 0.01 | 2.8 | 0.482 | 0.596 | 0.531 | 0.551 |
| LSTM | 0.01 | 0.01 | 1.74 | 0.49 | 0.624 | 0.584 | 0.547 |
| CNN | 0.02 | 0.01 | 4.82 | 0.49 | 0.535 | 0.424 | 0.433 |
| GRU | 0.01 | 0.01 | 2.6 | 0.531 | 0.637 | 0.596 | 0.551 |
| CNNRNN | 0.01 | 0.01 | 2.41 | 0.498 | 0.645 | 0.645 | 0.531 |
| CNNLSTM | 0.02 | 0.01 | 2.27 | 0.49 | 0.661 | 0.637 | 0.6 |
| RNN2 | 0.03 | 0.01 | 1.6 | 0.551 | 0.612 | 0.547 | 0.506 |
| LSTM2 | 0.01 | 0.01 | 1.74 | 0.49 | 0.624 | 0.584 | 0.547 |
| CNN2 | 0.02 | 0.01 | 4.82 | 0.49 | 0.535 | 0.424 | 0.433 |
| GRU2 | 0.01 | 0.01 | 2.6 | 0.531 | 0.637 | 0.596 | 0.551 |
| CNNRNN2 | 0.01 | 0.01 | 2.41 | 0.498 | 0.645 | 0.645 | 0.531 |
| CNNLSTM2 | 0.01 | 0.01 | 2.27 | 0.49 | 0.661 | 0.637 | 0.6 |

# Hourly Training Results – 1500 hours

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bitcoin | MAE | MSE | MAPE | 1 Hr WR | 6 Hr WR | 12 Hr WR | 1 Day WR |
| RNN | 0.02 | 0.01 | 1.72 | 0.511 | 0.635 | 0.557 | 0.522 |
| LSTM | 0.01 | 0.01 | 2.23 | 0.478 | 0.714 | 0.565 | 0.538 |
| CNN | 0.01 | 0.01 | 2.86 | 0.505 | 0.527 | 0.457 | 0.446 |
| GRU | 0.01 | 0.01 | 1.56 | 0.489 | 0.714 | 0.581 | 0.546 |
| CNNRNN | 0.01 | 0.01 | 2.3 | 0.497 | 0.716 | 0.576 | 0.559 |
| CNNLSTM | 0.01 | 0.01 | 2.47 | 0.473 | 0.722 | 0.573 | 0.516 |
| RNN2 | 0.02 | 0.01 | 2.28 | 0.516 | 0.651 | 0.557 | 0.557 |
| LSTM2 | 0.01 | 0.01 | 2.23 | 0.478 | 0.714 | 0.565 | 0.538 |
| CNN2 | 0.01 | 0.01 | 2.86 | 0.505 | 0.527 | 0.457 | 0.446 |
| GRU2 | 0.01 | 0.01 | 1.56 | 0.489 | 0.714 | 0.581 | 0.546 |
| CNNRNN2 | 0.01 | 0.01 | 2.3 | 0.497 | 0.716 | 0.576 | 0.559 |
| CNNLSTM2 | 0.01 | 0.01 | 2.47 | 0.473 | 0.722 | 0.573 | 0.516 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ethereum | MAE | MSE | MAPE | 1 Hr WR | 6 Hr WR | 12 Hr WR | 1 Day WR |
| RNN | 0.02 | 0.01 | 14.03 | 0.522 | 0.559 | 0.522 | 0.511 |
| LSTM | 0.01 | 0.01 | 1.49 | 0.489 | 0.649 | 0.638 | 0.562 |
| CNN | 0.02 | 0.01 | 6.08 | 0.47 | 0.4 | 0.408 | 0.457 |
| GRU | 0.01 | 0.01 | 1.76 | 0.489 | 0.643 | 0.619 | 0.546 |
| CNNRNN | 0.01 | 0.01 | 1.74 | 0.497 | 0.63 | 0.6 | 0.581 |
| CNNLSTM | 0.01 | 0.01 | 2.23 | 0.505 | 0.657 | 0.627 | 0.568 |
| RNN2 | 0.02 | 0.01 | 2.27 | 0.505 | 0.659 | 0.592 | 0.559 |
| LSTM2 | 0.01 | 0.01 | 1.49 | 0.489 | 0.649 | 0.638 | 0.562 |
| CNN2 | 0.02 | 0.01 | 6.08 | 0.47 | 0.4 | 0.408 | 0.457 |
| GRU2 | 0.01 | 0.01 | 1.76 | 0.489 | 0.643 | 0.619 | 0.546 |
| CNNRNN2 | 0.01 | 0.01 | 1.74 | 0.497 | 0.63 | 0.6 | 0.581 |
| CNNLSTM2 | 0.01 | 0.01 | 2.23 | 0.505 | 0.657 | 0.627 | 0.568 |

# Hourly Training Results – 2000 hours

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bitcoin | MAE | MSE | MAPE | 1 Hr WR | 6 Hr WR | 12 Hr WR | 1 Day WR |
| RNN | 0.01 | 0.01 | 2.1 | 0.497 | 0.618 | 0.549 | 0.543 |
| LSTM | 0.01 | 0.01 | 26.87 | 0.495 | 0.717 | 0.574 | 0.558 |
| CNN | 0.01 | 0.01 | 5.59 | 0.489 | 0.481 | 0.477 | 0.487 |
| GRU | 0.01 | 0.01 | 2.29 | 0.477 | 0.725 | 0.562 | 0.547 |
| CNNRNN | 0.01 | 0.01 | 2.62 | 0.507 | 0.729 | 0.572 | 0.56 |
| CNNLSTM | 0.01 | 0.01 | 1.57 | 0.521 | 0.741 | 0.574 | 0.562 |
| RNN2 | 0.02 | 0.01 | 8.41 | 0.511 | 0.673 | 0.598 | 0.541 |
| LSTM2 | 0.01 | 0.01 | 26.87 | 0.495 | 0.717 | 0.574 | 0.558 |
| CNN2 | 0.01 | 0.01 | 5.59 | 0.489 | 0.481 | 0.477 | 0.487 |
| GRU2 | 0.01 | 0.01 | 2.29 | 0.477 | 0.725 | 0.562 | 0.547 |
| CNNRNN2 | 0.01 | 0.01 | 2.62 | 0.507 | 0.729 | 0.572 | 0.56 |
| CNNLSTM2 | 0.01 | 0.01 | 1.57 | 0.521 | 0.741 | 0.574 | 0.562 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ethereum | MAE | MSE | MAPE | 1 Hr WR | 6 Hr WR | 12 Hr WR | 1 Day WR |
| RNN | 0.02 | 0.01 | 2.82 | 0.503 | 0.62 | 0.588 | 0.558 |
| LSTM | 0.01 | 0.01 | 1.34 | 0.499 | 0.638 | 0.616 | 0.576 |
| CNN | 0.02 | 0.01 | 7.81 | 0.485 | 0.521 | 0.475 | 0.473 |
| GRU | 0.01 | 0.01 | 1.66 | 0.491 | 0.653 | 0.608 | 0.57 |
| CNNRNN | 0.01 | 0.01 | 4.75 | 0.503 | 0.651 | 0.618 | 0.576 |
| CNNLSTM | 0.01 | 0.01 | 1.03 | 0.505 | 0.657 | 0.614 | 0.572 |
| RNN2 | 0.02 | 0.01 | 1.6 | 0.521 | 0.628 | 0.58 | 0.568 |
| LSTM2 | 0.01 | 0.01 | 1.34 | 0.499 | 0.638 | 0.616 | 0.576 |
| CNN2 | 0.02 | 0.01 | 7.81 | 0.485 | 0.521 | 0.475 | 0.473 |
| GRU2 | 0.01 | 0.01 | 1.66 | 0.491 | 0.653 | 0.608 | 0.57 |
| CNNRNN2 | 0.01 | 0.01 | 4.75 | 0.503 | 0.651 | 0.618 | 0.576 |
| CNNLSTM2 | 0.01 | 0.01 | 1.03 | 0.505 | 0.657 | 0.614 | 0.572 |

# Data Smoothing Experimentation Results

BTC Daily

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Neural Net | Data/Winrate | Score (no MA) | Score (SMA) | Score (EMA) |
| GRU | 1 year – 5 days | 0.779 | 0.779 | 0.721 |
| GRU2 | 1 year – 5 days | 0.779 | 0.779 | 0.721 |
| LSTM | 1 year – 5 days | 0.767 | 0.686 | 0.628 |

BTC Hourly

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Neural Net | Data/Winrate | Score (no MA) | Score (SMA) | Score (EMA) |
| CNN-LSTM | 2000 hour – 6 hours | 0.741 | 0.711 | 0.725 |
| CNN-LSTM2 | 2000 hour – 6 hours | 0.729 | 0.711 | 0.725 |
| CNN-RNN | 2000 hour – 6 hours | 0.729 | 0.681 | 0.711 |

ETH Daily

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Neural Net | Data/Winrate | Score (no MA) | Score (SMA) | Score (EMA) |
| RNN2 | 1 year – 5 days | 0.744 | 0.709 | 0.744 |
| LSTM | 1 year – 5 days | 0.709 | 0.663 | 0.581 |
| CNN-LSTM | 1 year – 5 days | 0.709 | 0.64 | 0.64 |

ETH Hourly

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Neural Net | Data/Winrate | Score (no MA) | Score (SMA) | Score (EMA) |
| CNN-LSTM | 1000 hours – 6 hours | 0.661 | 0.624 | 0.608 |
| CNN-LSTM2 | 1000 hours – 6 hours | 0.661 | 0.624 | 0.608 |
| CNN-RNN | 1000 hours – 6 hours | 0.645 | 0.58 | 0.637 |

# Experimentation Code

All code used for the purposes of experimentation within the scope of this study are publicly available and can be found on GitHub at:

<https://github.com/jk00845/FYP-Deep-Learning-Cryptocurrency-Prediction>