Project Report

MINOR PROJECT III

PRESENTED TO

Kushagra Agrawal

(E13465)

PRESENTED BY

Aniket Mittal

(20BCS6819)

SHORT TERM CRYPTO CURRENCY PRICE PREDICTOR USING ML

A Project Work

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

AIML

Submitted by:

Aniket Mittal

20BCS6819

Under the Supervision of:

Kushagra Agrawal (E13465)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING APEX INSTITUE OF TECHNOLOGY

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,

PUNJAB

MAY 2023

DECLARATION

I, **'Aniket Mittal**, student of **'Bachelor of Engineering in CSE (AIML)'**, **session:2020 - 24**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled '**Short-Term crypto currency price predictor using ML'** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

(Aniket Mittal)

Candidate UID: 20BCS6819

Date:

Place:

ACKNOWLEDGEMENT

I have dedicated myself in this project. Although, it would not have been achievable without the support and help of many personal and administration. I would like to acknowledge all of them.

I am extremely grateful to Chandigarh University for their guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

In the achievement of fulfilment of my project on Short-term crypto currency price predictor using ML, I would like to send my special gratitude to my mentor Mr. Kushagra Agrawal, of AIT Department. I would like to thank him for giving his precious time and valuable efforts, guidance and suggestions that helped me in various phases of the completion of this project. I will always be grateful to him.

Ultimately, as one of the students, I would like to acknowledge my teachers for their support and coordination. As well as their abilities in developing the project and have willingly helped me out.

TABLE OF CONTENTS

List of Tables

Table title **page**

List of Figures

List of Symbols

1. LSTM	Long Short-Term Memory
2. RNN	Recurrent Neural Network
3. BTC	Bitcoin
4. USD	United State Dollar
5. P2P	Peer to Peer
6. AODE	Averaged one-dependence estimator
7. RF	Random Forest
8. SVM	Support Vector Machine
9. DNN	Deep Neural Network
10. GRNN	Generalized Regression Neural
11. RSM	Random Sampling Method

Abstract

Cryptocurrency has grown significantly in recent years. Additional progress in the planetarium has recognized the importance of embracing quantitative benefits and rapid progress in this field. In today's financial markets, the decision to buy or sell cryptocurrency is an interesting challenge that traders face every day. During the year, it reached unprecedented highs, leading to the idea that explains the growth trend. The question of whether the movement of financial assets can be predicted has been of great interest to investors, economists and researchers in recent years. Therefore, the paper uses machine learning to build a model to predict the price of Cryptocurrency using technical indicators, which is the most important to learn market trends. This study explores how to adapt Long Short-Term Memory (LSTM) to build cryptocurrency price prediction models. The main factors used are the available price, closing price, high price, low price, volume and market capitalization among several cryptocurrencies, based on the size of important trading characteristics that affect the unpredictability of using the model to improve the efficiency of the process. However, the cryptocurrency market lacks a strong and unpredictable regulatory structure, making price prediction more difficult and complex. From the analysis, it was found that the machine learning model provides better performance in cryptocurrency price prediction.

In this project, we use LSTM version of recurrent neural network, price for Bitcoin. To better understand the price impact and make an overview of this great invention, we first take a brief look at the Bitcoin economy. Next, we define a database that includes data from the market index, sentiment, blockchain, and Coinmarketcap. In this analysis, we show the use of LSTM structure and the aforementioned time. Finally, we pull the results of Bitcoin price prediction 30 and 60 days in advance

The purpose of this research paper is to derive an algorithmic model with high prediction accuracy for the price of Bitcoin the next day through random forest regression and LSTM and explain the variables that affect the price of Bitcoin. There is more literature on Bitcoin price forecast research and research methods mainly based on time series ARMA model and deep learning LSTM algorithm.

Although the Diebold-Mariano test cannot prove that the prediction accuracy of random forest regression is better than LSTM, the RMSE and MAPE errors of random forest regression are better than LSTM. Changes in the variables that determine the price of Bitcoin in each period are also obtained by random forest regression. From 2015 to 2018, three US stock indexes, NASDAQ, DJI, S&P500, and oil prices and ETH prices were influenced by Bitcoin prices.

Important variables since 2018 have been the price of ETH and the Japanese stock index JP225. The relationship between accuracy and the number of explanatory variables introduced into the model shows that the model with only one lag explanatory variable has the best prediction accuracy for predicting the next day's price of Bitcoin.

Keywords

Bitcoin, Crypto Currency, Machine Learning, Blockchain, Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Prediction

1. INTRODUCTION

Bitcoin and other crypto currencies are a decentralized digital currency that uses cryptography for security and is not controlled by governments or financial institutions. It was created in 2008 with a paper entitled "Bitcoin: A Peer-to-Peer (P2P) Electronic Cash System" by an individual or a group of individuals using the pseudonym Satoshi Nakamoto (2008). Bitcoin transactions are recorded on the public blockchain, which allows anyone to see the history of a particular Bitcoin. The decentralized nature of Bitcoin allows it to be used independently of central banks and can be instantly transferred worldwide. It became popular as a medium of exchange and store of value. In the last 10 years, in November 2021, one coin has exceeded USD 68,000, and the total value once exceeded USD 1.2 trillion.

However, Bitcoin as a commodity has high volatility. During the seven-year period from April 2015 to April 2023, Bitcoin's average daily return was 3.85%, which is 2.68 times the return rate of gold in the same period, 3 times the S & P500. This is 36 times higher. Bitcoin's functionality as a commodity, as a store, and as a currency has been called into question due to its large price swings.

Taking advantage of Bitcoin's security and decentralization, it has become a challenge how to understand the trend of Bitcoin in order to reduce the risk of Bitcoin floating. Many researchers try to understand the trend of Bitcoin through the correlation between the price of Bitcoin and the price of other commodities.

In past studies, another form of research to understand the Bitcoin price trend is to predict the price of Bitcoin in the future using AI algorithms and powerful computing power of computers. Machine learning technology has become a hot research area in the 21st century with improvements in hardware performance. Initially, machine learning was used in various fields such as stocks, crude oil market, gold market and futures market.

Predicting AI by Bitcoin mainly falls into two categories. The first category is classification research to predict whether Bitcoin will rise or fall in the future. Standard errors of DA and F1. Another category is the regression test on Bitcoin price prediction, with relative error RMSE and MAPE. Because the price of Bitcoin fluctuates so much, just knowing whether the price of Bitcoin will rise or fall in the future will not allow investors to avoid risk. Instead, it is more useful to take the price of bitcoin as a reference value.

Predicting AI by Bitcoin mainly falls into two categories. The first category is classification research to predict whether Bitcoin will rise or fall in the future. Standard errors of DA and F1. Another category is the regression test on Bitcoin price prediction, with relative error RMSE and MAPE. Because the price of Bitcoin fluctuates so much, just knowing whether the price of Bitcoin will rise or fall in the future will not allow investors to avoid risk. Instead, it is more useful to take the price of bitcoin as a reference value.

Based on the need to avoid price risk as Bitcoin, this study chooses a machine learning random forest regression algorithm and a neural network algorithm LSTM model to predict Bitcoin's price. I mainly focus on the performance of random forest regression in predicting Bitcoin price while using LSTM prediction results as a comparison. Random forest regression is a variant of random forest regression. Unlike black-box neural network technology, random forest regression, like machine learning, can provide the value of each explanatory variable in predicting Bitcoin through the results of weak learners.

1.1 Problem Definition

Cryptocurrency is a type of digital currency similar to the dollar, euro and yen. The difference is that instead of being backed by a nation or federal bank, it uses an online ledger with the strong cryptography so to secure online transactions.

Through cryptocurrency exchanges, one can buy and sell cryptocurrency. It can also be "mining". The popularity of cryptocurrencies skyrocketed in 2017 as a result of the several months of exponential growth in their market capitalization.

As geopolitical and economic issues have escalated over the past two years, global currency values have plummeted, stock markets have seen a poor run and investors have lost their wealth. This has rekindled interest in digital currencies. Our system helps predict the price of cryptocurrencies using machine learning.

Problem Overview

Buying and selling cryptocurrencies like Bitcoin can yield tremendous gains if done correctly. It has proved lucky for many people in the past and is still earning them a lot of money today. But this does not come without its downsides. If not thought and calculated properly, one can lose all your money. There should be an incredible understanding of how and precisely crypto costs change (according to market news, various crypto events, technical analysis and so on), which suggests that realizing how individuals make their crypto predictions is very important. Considering these things (supply and demand, regulations, news, etc.), the prices of various crypto currencies are finally predicted.

Short term crypto currency price predictor using ML is a very useful project. Crypto currency market is growing like a boom day by day.

This project will help you predict the future price of various crypto currencies based on the historical data of that particular currency, so that it becomes easier and more efficient for a crypto trader to forecast the market and earn money by trading in various. Crypto currency.

Hardware Specification

1. RAM: 6GB or more
2. PROCESSOR: 64bit
3. Laptop with GPU with more than or equal to 4cores.

Software Specification

1. Python
2. Anaconda software
3. Jupyter Notebook

Tools Required:

1. NumPy
2. Pandas
3. Matplotlib
4. Scikit learn
5. Tensorflow

1.2 Literature Review

We're all predicting where bitcoin spending will be in one year, two years, five years, or 10 years. Waiting is hard, but we all love to do it. Buying and selling bitcoins can be done accurately every time. It has proven to be an asset for many people in the past and is still earning a lot of money. But it won't hurt either. If it is not properly thought and considered correctly, you can lose a lot of money. You must have a great understanding of how and exactly the price of bitcoin changes (organic market, trends, news, etc.), which means you must understand how we make bitcoin predictions. With this in mind (supply and demand, regulations, news, etc.), bitcoin technology and its development should be considered. In addition to this, we have to deal with the technical aspects of using various algorithms and technologies that can accurately predict the price of bitcoin. We have found some models that exist today, such as the biological nervous system. (BNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Automatic Regressive Integral Moving Average (ARIMA) etc. Time series is usually a series of numbers over time. This is because, as a time series data set, the total data set must be divided into two parts: input and output. In addition, LSTM is excellent compared to classical statistical linear models because it can easily solve multi-input prediction problems.

The price of bitcoin is mainly based on public recognition, as Lee and Wan (2017) have analyzed social media messages, Google searches, Wikipedia views, tweets or comments on Facebook or special forums, among others. crypto currency. For example, Kim et al. (2016) considers user comments and responses specific to bitcoin to predict daily price changes and transactions of bitcoin, ether, and ether. Phillips and Gorse (2017) used a hidden Markov model based on online social media indicators to formulate successful trading strategies in several cryptocurrencies. Corbett et al. (2018b) found that bitcoin, ripple and litecoin were uncorrelated with several economic and financial variables in the time and

frequency domains. Sovbetov (2018) showed that factors such as market beta, trading volume, volatility and attractiveness affect the weekly price of bitcoin, ether, dash, Litecoin and monero. Phillips and Gorse (2018) investigate whether online and social media factors influence the price of bitcoin, ether, icocoin, and monero based on the market regime; they found that short-term correlations are strongly amplified in regimes such as crowdfunding, while short-term correlations are triggered by specific market events such as hackers or security regulations.

Some researchers, such as Stavroyannis and Babalos (2019), study the hypothesis of irrational behaviour such as herding in the crypto market. Gurdgiev and O'Loughlin (2020) investigated the price dynamics of 10 cryptocurrencies and their proxies due to fear, uncertainty (US stock market uncertainty index), investor attitudes to cryptocurrencies (based on investor opinions on bitcoin forums), and investors in general finance. their perception of rudeness/incivility (measured by CBOE call ratio). They noted that investor sentiment is a good predictor of the price direction of cryptocurrencies and that cryptocurrencies can be used as a hedge in times of uncertainty.

Along these lines, Chen et al. (2020a) analysed the impact of fear sentiment on the price of bitcoin and showed that an increase in the fear of coronavirus leads to negative results and trading volume. The author notes that during a market crisis (for example, during the coronavirus pandemic), bitcoin behaves like any other financial asset — it is not a safe haven. In other areas of literature, some authors have directly studied the efficiency of the cryptocurrencies market, especially bitcoin. Using a different methodology, Urquhart (2016) and Bariviera (2017) argue that bitcoin is not efficient, while Nadarajah and Chu (2017) and Tiwari et al. (2018) argue the opposite. However, Urquhart (2016) and Bariviera (2017) note that after the initial transition phase, bitcoin tends to be efficient as the market begins to grow.

In the last three years, interest in predicting and profiting from cryptocurrencies using ML methods has increased. Table 1 summarizes several papers in chronological order since the work of Madan et al. (2015), to our knowledge, is one of the first works to address this issue. We do not intend to provide an exhaustive list of papers for this literature; instead, our goal is to summarize our research and highlight its main contributions. For a comprehensive study of Cryptocurrency trading and more information on ML trading.

EXISTING SYSTEM

Initial research on Bitcoin disputes whether it is actually another form of currency or a purely speculative asset, with most authors favouring the latter view due to its high volatility, very short-term returns, and bubble-like behaviour. (For example, Yermack 2015; Dwyer 2015; Cheung et al. 2015; Cheah and Fry 2015). This claim has been carried over to other well-executed cryptocurrencies such as Ethereum, Litecoin, and Bitcoin (e.g., Gkillas and Katsumata 2018; Catania et al. 2018; Corbet et al. 2018 Charfeddine and Mauchi 2019).

The idea that cryptocurrencies are purely speculative assets with no intrinsic value investigated the possible relationship between macroeconomic and financial variables and price determinants in investor behavior. These factors also proved to be very important for traditional markets. For example, Wen et al. (2019) shows that Chinese firms with greater exposure to retail investors have a lower risk of stock price collapse.

Christoufek (2013) showed that there is a high correlation between Google Trends and search queries on Wikipedia and the price of bitcoin. Kristoufek (2015) reinforced the previous results and found no significant correlation with basic variables such as the Financial Stress Index and the gold price of the Swiss

Franc. Bouoiyour and Selmi (2015) investigated the relationship between the price of bitcoin and several variables such as the gold market price, Google search, and bitcoin rate, and found that only Google search has a significant effect at the 1% level. Polasik et al. (2015) showed that bitcoin price formation is mainly related to the volume of news, news sentiment, and the number of traded bitcoins.

PROPOSED SYSTEM

In our approach, LSTM will use historical data to predict the closing price of bitcoin 30 days in advance. In the approach we use, we apply a Bayesian optimized recurrent neural network (RNN) and a long-term memory network (LSTM). The highest classification accuracy was achieved by LSTM with an accuracy of 52% and an RMSE of 8%. We now apply the popular Auto Backward Revolving Average (ARIMA) model for timing as a correlation and deep learning model. ARIMA forecasting is performed better by nonlinear deep learning methods. Finally, both deep learning models are defined on GPU and CPU. Training time on CPU is 67.7% higher than GPU performance. Among the key papers we selected, the authors collected more than 25 years of feature sets related to bitcoin prices and payment systems, recorded daily, and were able to predict the sign of daily bitcoin price changes. with an excellent accuracy of 98.7%.

In the second phase of our experiment, we focus only on bitcoin price data and use data for 10 minutes and 10 seconds. This is because we see a great opportunity to accurately evaluate price forecasts at different levels of granularity and granularity. This results in excellent results with 50-55% accuracy in predicting future bitcoin price changes in 10-minute time intervals.

LITERATURE REVIEW SUMMARY

Article	Dependent variable	Sample Period	Model	Input Set	Main Findings
Madan et al. (2015)	Bitcoin prices in USD from Coinbase	5 years since the inception of Bitcoin	Binomial logistic regressions (BLR) and random forest (RF)	Prices and 16 blockchain features	10-min data give a better sensitivity and specificity ratio than the 10-s data
Kim et al. (2016)	Bitcoin, ethereum and ripple prices	Bitcoin: Dec-2013 to Feb-2016 Ethereum: Aug-2015 to Feb-2016 Ripple: Sept-2015 to Jan-2016	Averaged one-dependence estimators (AODE)	Trading information, and comments and replies posted in online communities	Comments and replies are good predictors of Bitcoin prices
Żbikowski (2016)	Bitcoin prices in USD from Bitstamp	Jan-2015 to Feb-2015	Box (SVM) and volume weighted SVM	10 technical analysis indicators	VW-SVM is the best model in terms of average return and maximum drawdown
Jiang and Liang (2017)	Prices in USD of the 12 most traded cryptocurrencies at Poloniex	Jun-2015 to Aug-2016	Convolutional neural networks (CNN) with deep reinforcement learning	Returns	Mixed results between CNN portfolio and Online Newton Step and Passive Aggressive Mean Reversion portfolios

Jang and Lee (2018)	Bitcoin price index in USD	Sep-2011 to Aug-2017	Bayesian networks (BNN), linear regression and (SVM)	Trading information, exchange rates and macroeconomic variables	The BNN is the best prediction model
McNally et al. (2018)	Bitcoin prices in USD from CoinDesk	Aug-2013 to July-2016	Bayesian neural (RNN) and (LSTM)	OHLC prices, difficulty, and hash rate of blockchain	The best time lengths are 100 days for the LSTM and 20 days for the RNN
Nakano et al. (2018)	Bitcoin returns in USD from Poloniex	July- 2016 to Jan-2018	Artificial neural networks (ANN)	Returns and 4 technical analysis indicators	Higher performance of the ANN strategy, except in the last month of data. Results are highly sensitive to the model specification and input data
Vo and Yost-Bremm (2018)	Bitcoin prices in USD, CNY, JPY, EUR from 6 online exchanges	Jan-2012 to Oct-2017	Random forests (RF) and a deep learning model	5 technical analysis indicators	RF is the best model for a frequency of 15-min
Alessandretti et al. (2019)	Price indexes of 1681 cryptocurrencies in USD	Nov-2015 to Apr-2018	Ensemble of regression trees built by XGboost and long short-term memory network	Price, market capitalization, market share, rank, volume, and age	All strategies, produce a significant profit (expressed in bitcoin) even with transaction fees up to 0.2%

Atsalakis et al. (2019)	Bitcoin ethereum, litecoin and ripple returns	Sep-2011 to Oct-2017	PATSOS—a hybrid neuro-fuzzy model	Returns and prices	PATSOS outperforms other competing methods and produces a return significantly higher than the Buy-and-Hold (B&H) strategy
Catania et al. (2019)	Bitcoin, ethereum, litecoin and ripple returns in USD	Aug-2015 to Dec-2017	Linear univariate and multivariate regression models, and selections and combination	Returns and several exogenous financial variables	Statistically significant improvements in forecasting returns when using combinations of univariate models
de Souza et al. (2019)	Bitcoin prices in USD	May-2012 to May-2017	Artificial neural network (ANN) and support vector machine (SVM)	OHLC prices	SVM provides conservative returns on the risk adjusted basis, and ANN generates abnormal profits during short run bull trends
Han et al. (2019)	Bitcoin returns in USD	April-2013 to Mar-2018	NARX Neural Network	Returns	NARX is effective in predicting the tendency but not the jumps
Huang et al. (2019)	Bitcoin returns in USD	Jan-2012 to Dec-2017	Trees	124 technical indicators computed from the OHLC prices	Lower volatility, higher win-to-loss ratio and information ratio than those of every simple cut-off strategy or the B&H strategy

Ji et al. (2019b)	Bitcoin returns in USD from Bitstamp	Nov.-2011 to Dec.-2018	Deep Neural Network (DNN), (LSTM), (CNN), (ResNet), (CRNN) and their combination	Prices and 17 blockchain features	Performances of the prediction models were comparable, LSTM is the best prediction model, DNN models are the best classification models
Lahmiri and Bekiros (2019)	Bitcoin, digital cash and ripple prices in USD	Bitcoin: July-2010 to Oct-2018 Digital Cash: Feb-2010 to Oct-2018 Ripple: Jan-2015 to Oct-2018	LSTM and Generalized Regression Neural Networks (GRNN)	Prices	Predictability of LSTM is significantly higher than of GRNN
Mallqui and Fernandes (2019)	Bitcoin prices in USD	Apr-2013 to Apr-2017	Artificial neural networks (ANN), support vector machine (SVM) and ensembles	OHLC prices, Blockchain information and several exogenous financial variables	Ensemble of recurrent neural networks and a Tree classifier is the best classification model, while SVM is the best regression model
Shintate and Pichl (2019)	Bitcoin returns in CNY and USD from OkCoin	Jun-2013 to Mar-2017	Random sampling method (RSM), Long-short term memory	OHLC prices	The proposed RSM outperforms several alternatives.

Smuts (2019)	Bitcoin and ethereum prices in USD	Dec-2017 to Jun-2018	Long short-term memory recurrent neural network (LSTM)	Prices, volumes, Google trends, and Telegram chat groups dedicated to bitcoin and ethereum trading	Telegram data is a better predictor of bitcoin, while GoThe ensemble, by unweighted average of the four trading signals from the four models
Borges and Neves (2020)	Prices from Binance 100 cryptocurrencies pairs with the most traded volume in USD	For each pair since beginning of trading at Binance until oct-2018	Logistic regression, random forest, support vector machine, and gradient tree boosting and an ensemble of these models	Returns, resampled returns, and 11 technical indicators	~
Chen et al. (2020b)	Bitcoin price index and trading prices from Binance in USD	July-2017 to Jan-2018 for 5-min and Feb-2017, to Feb-2019 for daily	Logistic Regression (LR), (LDA), (RF), (XGB), (SVM), and Long Short-Term Memory (LSTM)	5-min: OHLC prices and trading volume. Daily: 4 Blockchain features, 8 marketing and trading variables media search volume, and gold spot price	For 5-min data machine learning models achieved better accuracy than LR and LDA, with LSTM achieving the best result (67% accuracy). For daily data, LR and LDA are better, with an average accuracy of 65%

Chu et al. (2020)	Bitcoin, ethereum, dash, litecoin, MaidSafeCoin, monero and ripple from CryptoCompare in USD	Feb-2017 to Aug-2017	Exponential Moving Averages (EMA) for time series and cross-sectional portfolios	Trading prices	Momentum trading does not beat the passive trading strategies
Sun et al. (2020)	42 crypto currencies	Jan-2018 to Jun-2018	LightGBM, SVM support vectors (SVM) and RF	Trading data and macroeconomic variables	LightGBM outperforms SVM and RF

Table 1. Different crypto prediction models

1.3 Machine Learning

Machine learning (ML) is a field devoted to understanding and building methods that allow machines to "learn," that is, methods that use data to improve computer performance on various problems. It is seen as a broad field of artificial intelligence. Machine learning algorithms build models based on sample data, called training data, to make predictions or decisions without precise planning. Machine learning algorithms are used in many applications, such as medicine, email filtering, word recognition, agriculture, and computer vision, where it is difficult or impossible to develop conventional algorithms to perform the required tasks. Part of machine learning is closely related to computational statistics, which uses computers to make predictions, but not all machine learning is a statistical study. The study of mathematical optimization provides methods, theories, and practical applications for machine learning systems. Data mining refers to research that focuses on the analysis of survey data through unsupervised learning.

Some machine learning processes use data and neural networks in a way that mimics the way the biological brain works. When applied to business problems, machine learning is also called predictive analytics. Learning algorithms are based on the likelihood that strategies, algorithms, and benchmarks that have worked well in the past will continue to work well in the future. This reference can sometimes be explicit, such as "every day for the last 10,000 days the sun will rise and rise tomorrow." Sometimes they can be more nuanced, like:X% of birds have geographically distinct species and colour patterns, so there's a chance Y% of blackbirds are undiscovered.

Machine learning applications can do this without explicit programming. This involves computers learning from data to perform certain tasks. For a simple task given to a computer, it is possible to program an algorithm that tells the machine

how to perform all the necessary steps to solve the problem at hand; no computer training required. For more advanced problems, it can be difficult for humans to create the necessary algorithms by hand. In practice, it may be more efficient to help the machine develop its own algorithm instead of having a human programmer determine the necessary steps. Machine learning techniques use a different approach to teach computers to perform tasks for which no algorithm is satisfactory. If there is a large number of possible answers, it is important to mark some of the correct answers. It can then be used as training data for computers to improve the algorithm(s) used to determine the correct answer. For example, the MNIST dataset of handwritten digits is often used to train systems for digital character recognition problems.

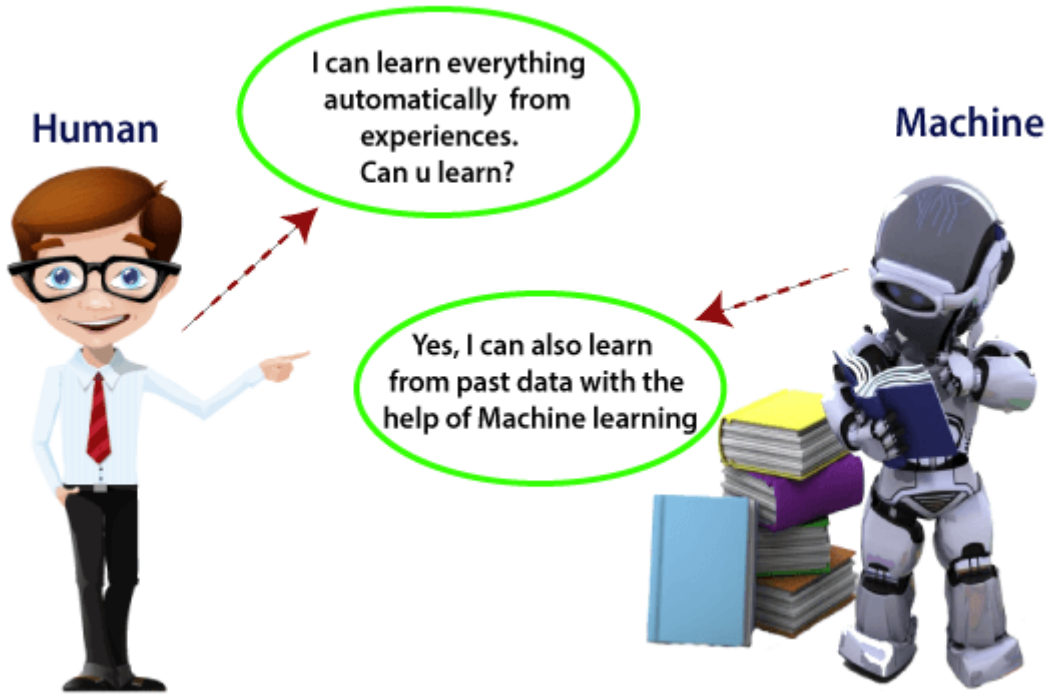


Figure 1. Machine Learning

Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

The need for machine learning is increasing day by day. The reason behind the need for machine learning is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannot access the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy. We can train machine learning algorithms by providing them the huge amount of data and let them explore the data, construct the models, and predict the required output automatically. The performance of the machine learning algorithm depends on the amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money. The importance of machine learning can be easily understood by its uses cases, Currently, machine learning is used in self-driving cars, cyber fraud detection, face recognition, and friend suggestion by Facebook, etc. Various top companies such as Netflix and Amazon have built machine learning models that are using a vast amount of data to analyse the user interest and recommend product accordingly.

Following are some key points which show the importance of Machine Learning:

- Rapid increment in the production of data
- Solving complex problems, which are difficult for a human
- Decision making in various sector including finance

- Finding hidden patterns and extracting useful information from data.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm:

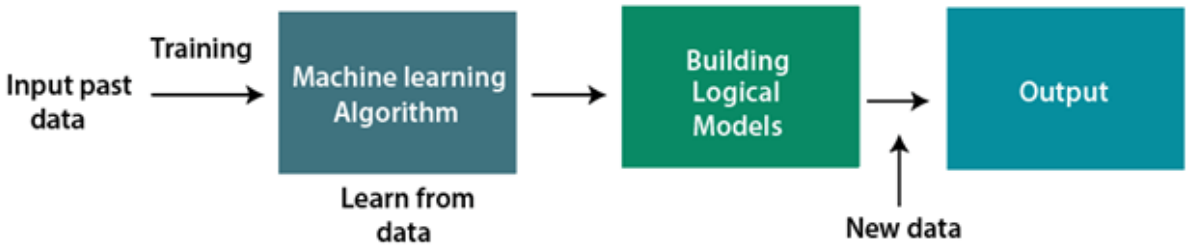


Figure 2. Working of Machine learning

Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

Supervised Learning

Supervised learning is a machine learning technique where we provide the machine learning system with specific data to train it and based on that, it predicts the output. The system creates a model by using the given data to understand the

database and learn each data, after training and processing, we test the model by providing sample data to see if it predicts the correct output. The purpose of supervised learning is to map input data to output data. Supervised learning is based on supervision and it is as if students learn things under the teacher's supervision. An example of supervised learning is spam filtering. Supervised learning algorithms can be divided into two categories:

- Classification
- Regression

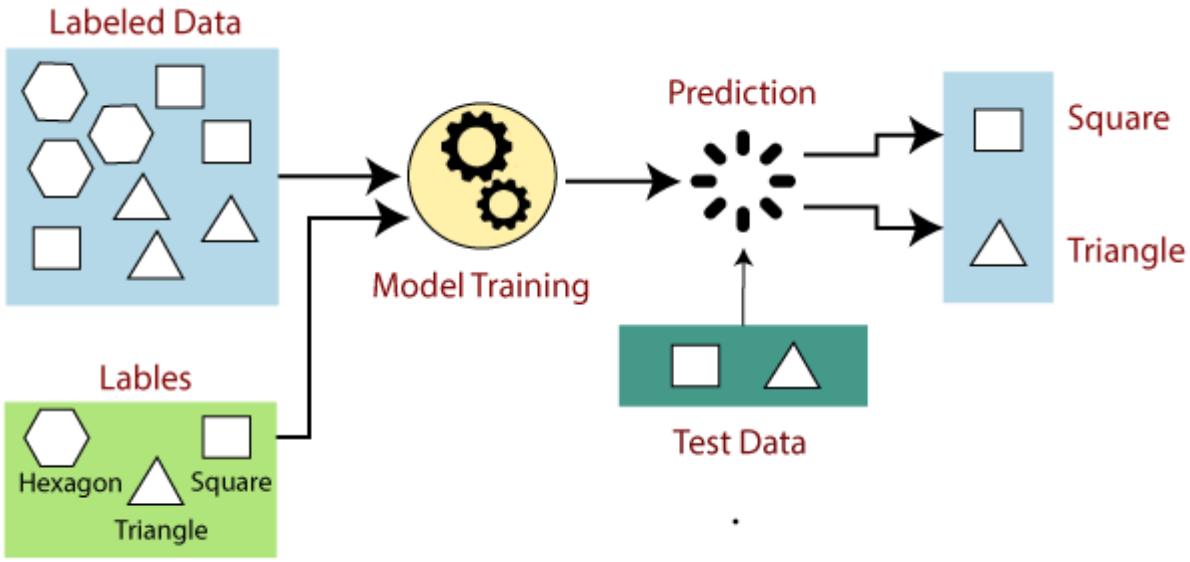


Figure 3. Supervised Machine Learning

Unsupervised Learning

Unsupervised learning is a learning technique where the machine learns without supervision. Training is given to the machine with a set of unlabelled, unclassified, or uncategorized data, and the algorithm must act on the data without any control. The purpose of unsupervised learning is to change input data into new features or groups of objects with similar patterns. In an uncontrolled

study, there are no predetermined outcomes. The tool tries to find useful insights from large amounts of data. Algorithms can further be divided into two categories:

- Clustering
- Association

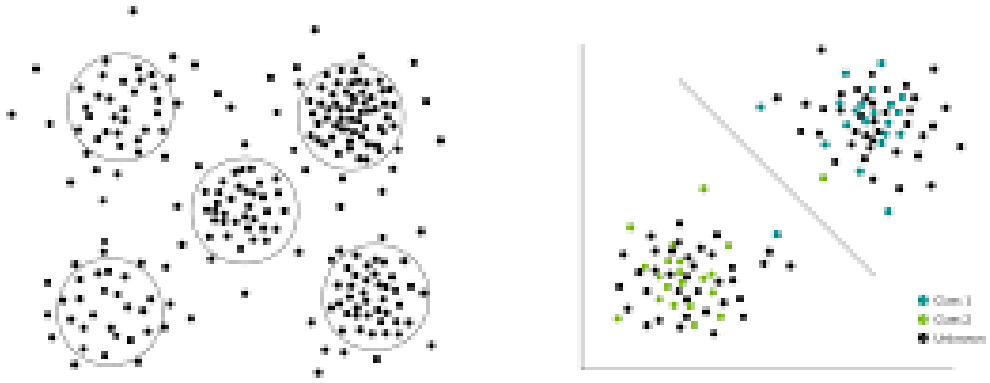


Figure 4. Unsupervised Machine Learning

Reinforcement Learning

Reinforcement learning is a learning technique based on feedback where the learning agent receives a reward for every correct behaviour and a punishment for every wrong behaviour. Agents automatically learn and improve their performance with this feedback. In reinforcement learning, the agent interacts with the environment and learns about it. The agent's goal is to earn the most reward points and thus improve his performance. A robotic dog that automatically learns its hand movements is an example of Reinforcement Learning.

1.4 Blockchain

Blockchain is a distributed database or record shared among the nodes of a computer network. They are known for their important role in the cryptocurrency system to maintain secure and decentralized transaction records, but they are not limited to using cryptocurrency themselves. Blockchain can be used in any industry to make data immutable - a term used to describe the fact that it cannot be changed. Since there is no way to change the block, the only trust required is where the user or application enters the data. This aspect will reduce the need for trusted third parties, which are usually auditors or other people who add costs and make mistakes. Since the launch of Bitcoin in 2009, the use of blockchain has exploded, spawning various cryptocurrencies, decentralized finance applications (DeFi), non-traceable tokens (NFTs), and smart contracts. You may be familiar with spreadsheets or databases. Blockchain is somewhat similar in that it is a database where data is entered and stored. But the main difference between traditional databases or spreadsheets and blockchain is the structure and availability of data.

Blocking is made up of programs called scripts that perform the tasks you would normally do in a database: Access and retrieve data, store and save it. Blocks are distributed, which means many copies are stored on many machines, and they all have to be correct. Blockchain collects transaction data and stores it in groups, like cells in a spreadsheet, that contain data. Once complete, the data is run through an encryption algorithm that creates a six-digit number called a Hash. This hash is then inserted into the header of the following block and encoded with the other data in the block. This creates a group of blocks that are chained together.

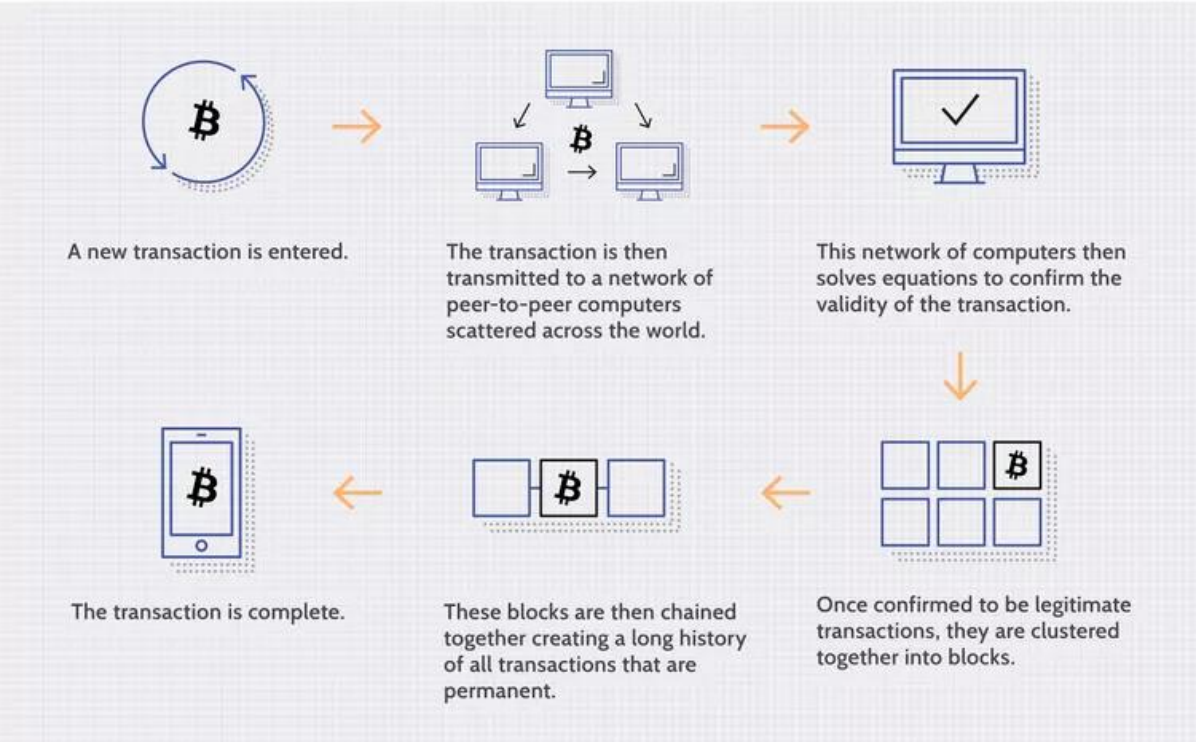


Figure 5. Blockchain Transaction Process

Blockchain Decentralization

This allows information in the blocking database to be distributed between different network nodes - between computers or devices running blocking software - in different locations. This not only creates redundancy but also maintains data integrity. For example, if someone tries to modify a record in one instance of the database, another node will prevent it. Therefore, even one point in the network cannot change the data in it. Because of this distribution and transaction evidence is encrypted, data and history (such as cryptocurrency transactions) cannot be reversed. Such a record can be a list of transactions (for example with cryptocurrency), but the blockchain can store many other data, such as legal contracts, government ID cards, or company records.

Bitcoin vs. Blockchain

Blockchain technology was first described in 1991 by Stuart Haber and W. Scott Stornett, two researchers who wanted to implement an immutable document seal system. But almost two decades later, with the launch of Bitcoin in January 2009, blockchain made its first real-world application. The Bitcoin protocol is built on the blockchain. In a research paper proposing a digital currency, Satoshi Nakamoto, the eponymous creator of Bitcoin, called it "a new, fully peer-to-peer electronic money system with no trusted third parties." The key thing to understand is that Bitcoin uses the blockchain to record payments or other transactions between parties.

As we now know, the Bitcoin blockchain stores transaction information. More than 23,000 other cryptocurrencies currently operate on the blockchain. But it turns out, blockchain is a secure way to store information about other types of transactions. Some of the companies experimenting with blockchain include Walmart, Pfizer, AIG, Siemens and Unilever, among others. For example, IBM created the Food Trust blockchain to track the journey of groceries to their location. Using blockchain allows brands to track a food product from its origin, to each stop, along the delivery route. Not only that, but this company can now see other things that may be in the relationship, which can save lives by allowing problems to be detected sooner. This is an example of blocking in action, but there are many other ways to implement blocking.

Currency

Blockchain is the basis of cryptocurrencies such as Bitcoin. The US dollar is managed by the Federal Reserve Bank. In this centralized system, user data and currency are technically at the discretion of the bank or government itself. If a user's bank is hacked, the customer's personal information is at risk. If the

customer's bank fails or the customer lives in a country with an unstable government, the value of their currency may be at risk. In 2008, several failed banks were bailed out, some with taxpayer money. This is an early and advanced issue of Bitcoin. Blockchain can provide a more stable currency and financial system to countries with unstable currencies or financial infrastructure. They have access to more programs and a greater number of individuals and institutions to conduct domestic and international business.

By distributing its operations across a network of computers, blockchain allows Bitcoin and other cryptocurrencies to operate without the need for a central authority. This not only reduces risk, but also reduces processing and transaction costs. Using a cryptocurrency wallet for a savings account or as a means of payment makes special sense for non-state holders. Some countries may be ravaged by war or governments have no real detection infrastructure. Citizens of such countries do not have access to savings or brokerage accounts and therefore cannot safely store their wealth.

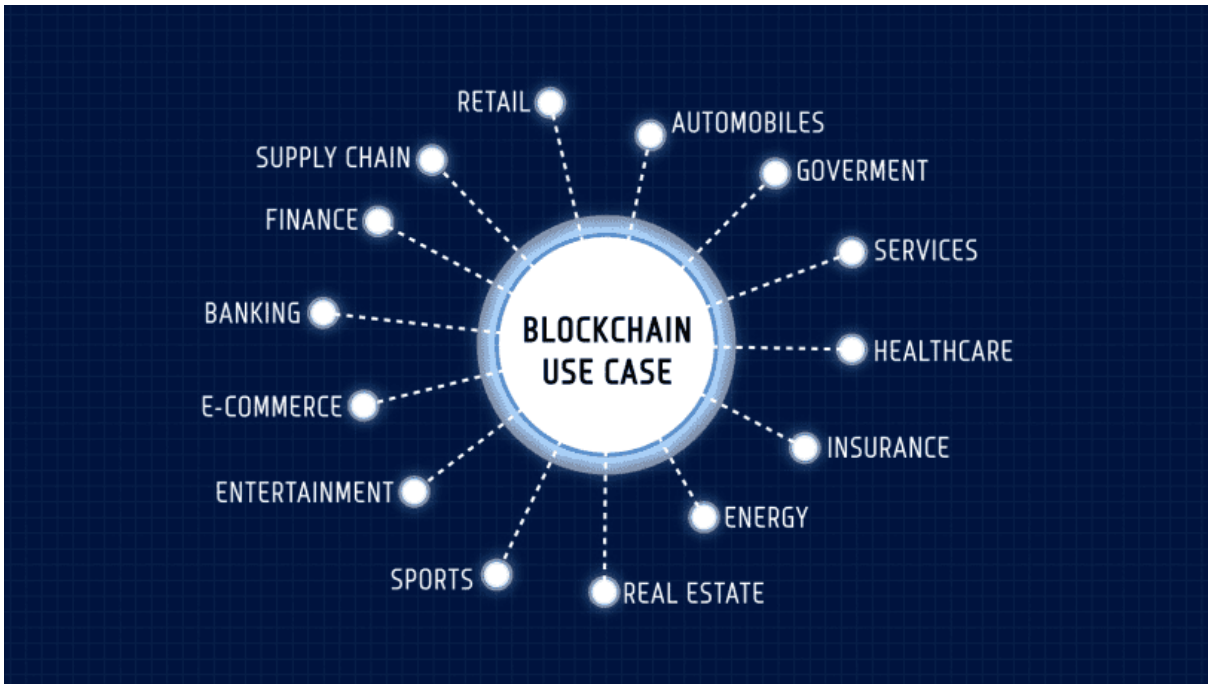


Figure 6. Blockchain Use Cases

Benefits of Blockchain

Accuracy of the Chain

Transactions on the blockchain are verified by thousands of computers and devices. This eliminates almost all human verification, leading to human error and inaccurate data recording. Even if a computer in the network should make a calculation error, that error will only be made for one copy of the block and will not be accepted by the rest of the network.

Cost Reductions

Typically, customers pay a bank to verify transactions or a notary to sign documents. Blockchain eliminates the need for third-party verification and associated costs. For example, borrowers pay lower fees when getting a credit card because banks and payment processing companies have to process the transaction. Bitcoin, on the other hand, is decentralized and has limited transaction fees.

Decentralization

Blockchain does not store data in a centralized location. Instead, the blockchain is copied and distributed across computer networks. Whenever a new block is added to the blockchain, each computer in the network updates its own blockchain to reflect the change. nstead of storing that information in a central database, it becomes more difficult to connect to the blockchain by spreading it across the network.

Efficient Transactions

Transactions made through central authorities may take several days to process. For example, if you deposit a check on Friday evening, you may not see the funds

in your account until Monday morning. Business hours for financial institutions are usually five days a week, but they are open 24 hours a day on weekends, seven days a week and 365 days a year. In some cases, transactions can be completed within minutes and considered secure after a few minutes. This is especially useful for cross-border transactions as time zones matter and all parties must agree on payment processing.

Private Transactions

Most blockchain systems operate as public databases, meaning that anyone with an internet connection can view the network's transaction history. While users can access transaction details, they cannot access identifying information about the user who made the transaction. It is common knowledge that blockchain networks like Bitcoin are completely anonymous; it is actually nicknamed because it has a visible address users link to when the information comes out.

Secure Transactions

After the transaction is recorded, its authenticity must be verified by the blocking system. Once the transaction is confirmed, it is added to the blockchain. Each block in the blockchain has a unique hash and a unique hash of the previous block. Therefore, the block cannot change if the network has confirmed them.

Transparency

Most blockchains are completely software. This means that anyone can see your code. This allows auditors to review cryptocurrencies such as Bitcoin for security. However, it also means that you have no real control over who manages or edits the Bitcoin code. Therefore, anyone can suggest changes or improvements to the system.

Banking the Unbanked

The most profound aspect of blockchain and cryptocurrency is that it can be used by anyone, regardless of nationality, race, location or cultural background. According to the World Bank, approximately 1.3 billion adults do not have a bank account or a means to store their money or assets. Moreover, almost all of these people live in developing countries that are in the early stages of the economy and are completely dependent on cash These people are often paid in physical cash. This physical cash can then be stored in their home or elsewhere, encouraging robberies or violence. While it is impossible to steal, crypto makes it more difficult for would-be thieves. The hives of the future is not only an account unit to store wealth, but also a solution to store medical records, property rights and various other legal documents.

Drawbacks of Blockchains

Technology costs

Although blockchain can save users money on transaction fees, the technology is not free. For example, the operating system consumes a lot of computing power to confirm transactions on the Bitcoin network. In the real world, millions of devices in the Bitcoin network consume more energy than Pakistan consumes every year. Several solutions to this problem are beginning to emerge. For example, bitcoin mining farms are designed to use solar energy, excess natural gas from pore space, or energy from wind farms.

Speed and data inefficiencies

Bitcoin is a perfect example of the potential inefficiency of the blockchain. Bitcoin's Pow system takes about 10 minutes to add a new block to the blockchain. At that rate, it is assumed that the blockchain can only handle three

transactions per second (TPS). While other cryptocurrencies such as Ethereum are doing better than Bitcoin, blockchain limits them. Legacy brand Visa can process up to 65,000 TPS per context. Solutions to this problem have been developed for years. Today, there are more than 30,000 TPS-boasting brokers. With the integration between the Ethereum main net and the beacon chain (September 15, 2022), many devices (phones, tablets, and laptops) will be able to run Ethereum. This is expected to increase network throughput, reduce congestion and increase traffic speed. Another problem is that each block can store only so much data. The debate about block size has been, and continues to be, one of the most important issues for the scalability of blockchains going forward.

Illegal activities

Privacy on the blockchain protects users from hackers and maintains privacy, as well as prevents illegal trade and activity on the blockchain. The most prominent example of blockchain being used for illegal activity is Wallet Road, an online dark site for illegal drugs and money laundering markets that operated between February 2011 and October 2013 when the FBI was shut down. The dark web allows users to buy and sell illegal goods and make illegal purchases in Bitcoin or other cryptocurrencies without tracking their Tor browser. This is in stark contrast to US regulations that require financial service providers to obtain information about their customers when they open an account. We must verify the identity of each customer and ensure that they are not on the list of known or suspected terrorist organizations. Illegal activity will account for only 0.24% of all cryptocurrency transactions in 2022. It allows anyone to access your financial account, but allows criminals to operate more easily. Many argue that the good use of crypto, such as the banking of the unbanked world, outweighs the bad use of cryptocurrency, especially since most illegal activities are still carried out with unrestrained cash.

Many in the crypto space are concerned about government regulation of cryptocurrencies. As decentralized networks grow, it becomes more difficult and, if not impossible, to stop something like Bitcoin, but governments could theoretically make it illegal to own cryptocurrencies or participate in their networks. This concern has diminished over time as major companies such as PayPal began allowing customers to use cryptocurrencies on their e-commerce platforms.

Cryptocurrency: Blockchain vs Cryptocurrency

The most popular (and most controversial) use of Blockchain is in cryptocurrencies. Cryptocurrencies are digital currencies (or tokens) such as Bitcoin, Ethereum or Litecoin that can be used to purchase goods and services. As a form of digital cash, crypto can be used to buy everything from lunch to the next house. Unlike cash, crypto uses blockchain as a public ledger and advanced cryptographic security system, so online transactions are always recorded and secured.

For example, the terms Bitcoin blockchain and cryptocurrency are used interchangeably, but they are still two separate entities. The first application of blockchain appeared in 2009 as Bitcoin, a crypto system that uses distributed ledger technology. Also marked Bitcoin as the first "pack". The blockchain aspect that is being used to become this new digital currency is what brought the two organizations together and is quickly gaining attention. Blockchain Bitcoin only describes the technology on which the currency is based, while cryptocurrency Bitcoin only describes the currency itself.

HOW DOES CRYPTOCURRENCY WORK?

Cryptocurrencies are digital currencies that use blockchain technology to record and protect every transaction. Cryptocurrencies (e.g. Bitcoin) can be used as a form of digital cash to pay for everyday items and large purchases such as cars and houses. Using one of several digital wallets or trading platforms, once an item is purchased, it can be digitally transferred and purchased with a blockchain record of the transaction and the new owner. The appeal of cryptocurrencies is that everything is publicly recorded and secured using cryptography, making every payment an irrefutable, timely and secure record.

To date, there are more than 20,000 cryptocurrencies in the world with a combined value of nearly $1 trillion, with Bitcoin accounting for most of the value. These tokens have become incredibly popular over the past few years, with the price of a single Bitcoin starting from several thousand dollars.

The main reasons for the recent popularity of Cryptocurrency are:

Blockchain security makes theft more difficult because each cryptocurrency has an irrefutable identifier attached to its owner. Crypto will reduce the need for separate currencies and central banks. With blockchain, cryptocurrencies can be sent anywhere in the world to anyone without the need for exchanges or central bank interference. Cryptocurrencies can make some people rich. Speculators run up the value of crypto, especially Bitcoin, helping some adopters become billionaires. It remains to be seen whether this is actually positive, as some critics believe that speculators do not consider the long-term benefits of crypto.

More and more large corporations have come to the idea of blockchain-based digital currency for payments. In February 2021, Tesla announced that it will invest $1.5 billion in Bitcoin, which will be accepted as payment for its cars. Of course, there are many legitimate arguments against blockchain-based digital

currencies. First, crypto is not a highly regulated market. Many governments are quickly getting into crypto, but there is little codified legislation on the matter. Also, crypto is incredibly volatile because of speculators. The lack of stability makes some people rich, while many still lose thousands of dollars. Whether or not digital currency has a future is still unclear. For now, blockchain's meteoric rise seems to be little more than pure hype. Although it is still a very new, highly researched business, blockchain shows promise beyond Bitcoin.

1.5 LSTM

LSTM Architecture

In our introduction to long-term memory, we learned how RNN solves the vanishing gradient problem, so now in this section we will see how LSTM solves this problem by exploring its architecture. At a high level, an LSTM works very much like an RNN cell. This is the deep way of LSTM network. LSTM network architecture consists of three parts as shown in the figure below and each part performs a specific function.

The first part chooses whether the information from the previous time point should be remembered or irrelevant and forgotten. In the second phase, the cell tries to learn new information from the input to the cell. Finally, in the third step, the cell transfers the updated data from the current timestamp to the next timestamp. One LSTM cycle is considered as one time step. These three parts of the LSTM section are known as gates. They control the flow of data in and out of the cell or cell etc. The first gate is called the Gate of Oblivion, the second gate is called the Gate of Entry, and the last gate is called the Gate of Exit. An LSTM unit consisting of these three gates and a memory cell or lstm cell can be thought

of as a neuron layer in a typical feedforward neural network, with each neuron having a hidden layer and current state.

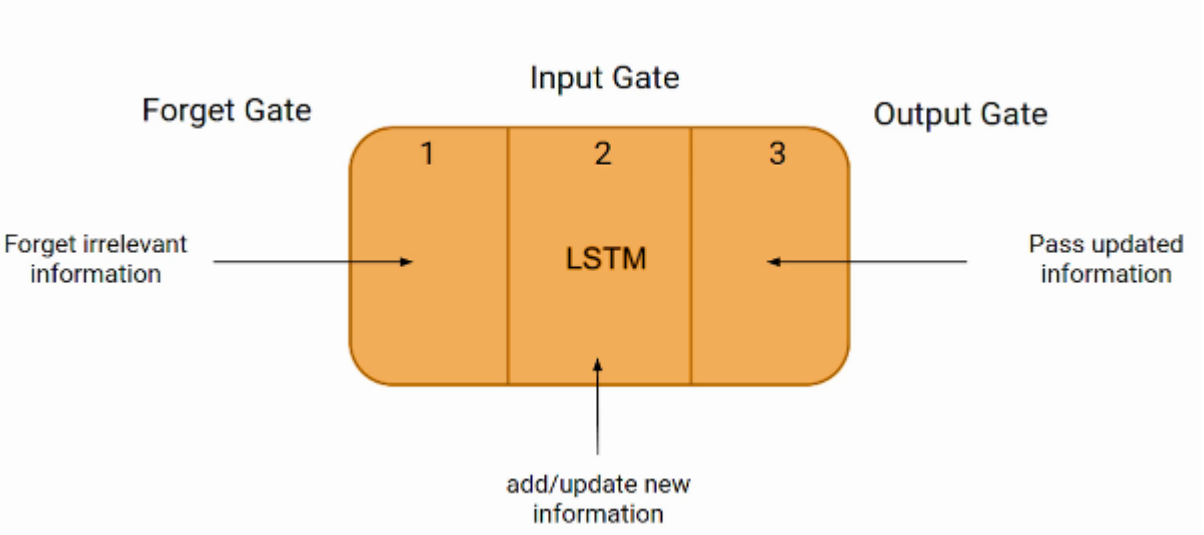


Figure 7. LSTM Architecture

Like simple RNN, in LSTM, H(t-1) represents the hidden state of the previous time number and Ht represents the hidden state of the current time number. In addition, LSTM has cell states denoted by C(t-1) and C(t) for the previous and current timestamps, respectively. Here, the latent state is called short-term memory and the cellular state is called long-term memory. See the picture below.

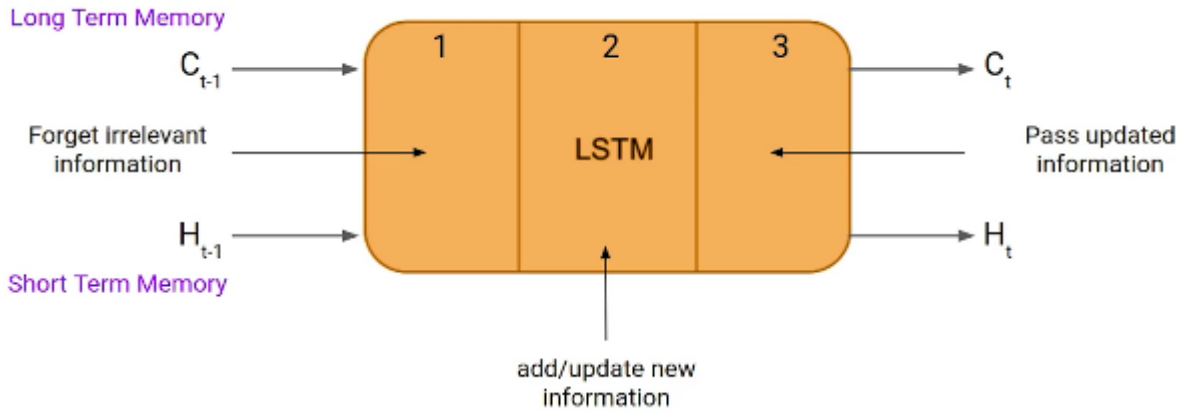


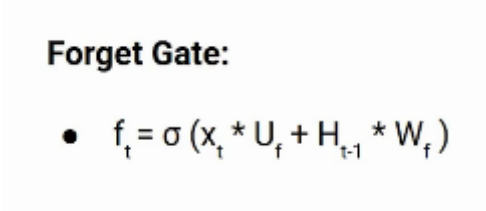
Figure 8. States in LSTM Architecture

It is interesting to note that the cell state carries the information along with all the timestamps.

Let's take an example to understand how LSTM works. These are two sentences separated by a period. The first sentence is "Bob is a good man" and the second sentence is "On the other hand, Dan is bad". In the first sentence, it is clear that we are talking about Bob, and after introducing a full stop (.), we start talking about Dan. When we move from the first sentence to the second sentence, our system must understand that we are no longer talking about Bob. Now our topic is Dan. Here, the network forget gateway allows you to forget it. Let us understand the role played by these gates in LSTM architecture.

Forget gate

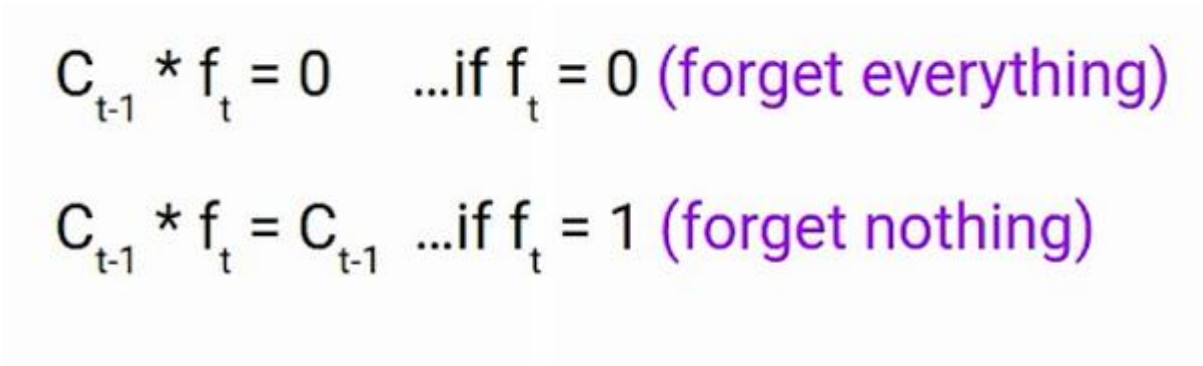
The first step in an LSTM neural network is whether to keep or forget the information from the previous step. This is the equation to forget the gate.



Let's try to understand the equation, here

- Xt: input to the current timestamp.
- Uf: weight associated with the input
- Ht-1: The hidden state of the previous timestamp
- Wf: It is the weight matrix associated with the hidden state

Later, a sigmoid function is applied to it which always has values between 0 and 1. This ft is later multiplied with the cell state of the previous timestamp, as shown below.



Input Gate

"Bob knows how to swim. He told me over the phone that he served four years in the Navy."

So, in these two sentences we are talking about Bob. However, both provide different information about Bob. In the first sentence, we learn that he can swim. As stated in the second sentence, he used a telephone and served in the Navy for four years. Now think based on the context given in the first sentence, what information is important in the second sentence? He started using the phone to talk or serve in the Navy. In this context, it does not matter whether he uses the telephone or some other means of communication to provide information. The important information contained in the fleet is what we want our model to remember for future calculations. This is the Gateway function.

Input gates are used to determine the importance of new data that is brought through the input. This is the input gate equation

Input Gate:

- $i_t = \sigma (x_t * U_i + H_{t-1} * W_i)$

- Xt: Input at the current timestamp t
- Ui: weight matrix of input
- Ht-1: A hidden state at the previous timestamp
- Wi: Weight matrix of input associated with hidden state

Again, we have applied the sigmoid function over it. As a result, the value of I at timestamp t will be between 0 and 1.

New Information

- $N_t = \tanh(x_t * U_c + H_{t-1} * W_c)$ (new information)

Now the new data that will be sent to the state of the cell is the hidden state at the previous timestamp t-1 and the input x at timestamp t. Here the activation function

is tanh. Because of the Tanh function, the value of the new data will be between -1 and 1. If the value of nt is negative, the data is removed from the state of the cell, and if the value is zero, the data is added to the cell, and it shows in the current timestamp. However, Nt is not directly involved in the state of the cell. Here is the updated equation:

$$C_t = f_t * C_{t-1} + i_t * N_t \text{ (updating cell state)}$$

Here, Ct-1 is the cell state at the current timestamp, and the others are the values we have calculated previously.

Output Gate

"Bob fought the enemy alone and died for his country. Courage _____ for their attachment."

During this exercise we have to complete the second sentence. Now, when we see the word hero, we know we are talking about a person. In this sentence, we can't just say Bob the hero, the enemy of the hero, or the country of the hero. So based on the current expectations, you have to give the right words to fill in the blanks. This word is our output and this is our output gate function. Here is the equation for the Exit Gate, which is the same as the previous two gates.

Output Gate:

- $o_t = \sigma(x_t * U_o + H_{t-1} * W_o)$

The value will lie between the 0 and 1 as we used sigmoid function. Now for the current hidden state calculation, Ot and tanh of the updated cell state will be used. As shown below.

$$H_t = o_t * \tanh(C_t)$$

It turns out that the hidden state is a function of long-term memory (Ct) and the current output. If you need to take the output of the current timestamp, just apply the SoftMax activation on hidden state Ht.

$$\text{Output} = \text{Softmax}(H_t)$$

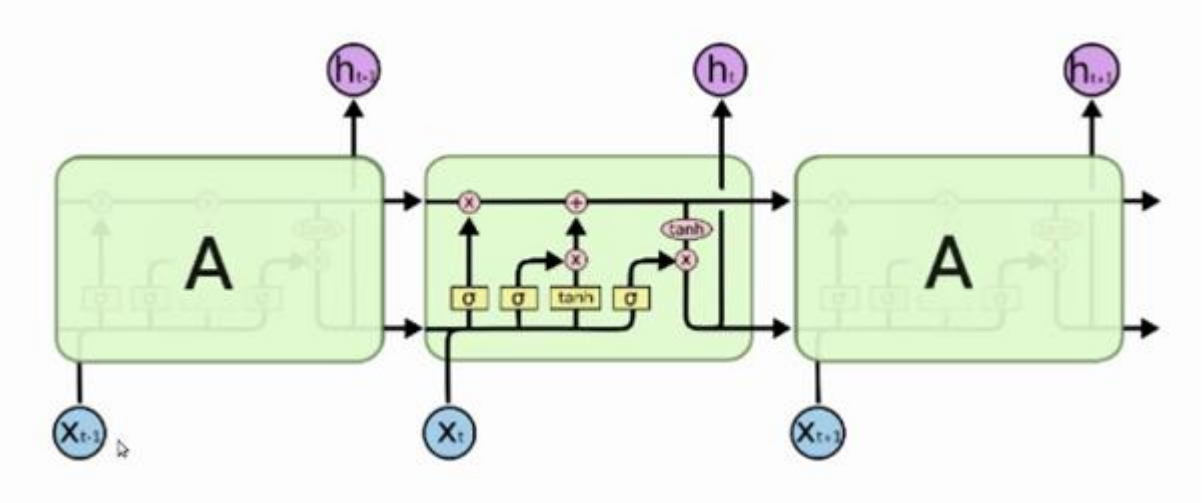


Figure 9. LSTM Output gate

The RNN algorithm is different from the traditional DNN algorithm. When data is exchanged with the model, it not only produces output values, but also changes the parameters of the model. The RNN algorithm has the task of retaining the previous input data in the model. This paper uses an LSTM model that simulates the short-term memory deficit of an RNN. Changing the data is an input to the RNN model and the memory model through three paths of the forgetting gate activation function, the input gate, and the output gate.

Based on the fact that the output value of the LSTM model can be transformed into another layer of the LSTM model and the application of the layer layer

mentioned in the literature, the structure of the LSTM model of this experiment is as follows. As for setting the output layer parameters, I experimented with [min = 10%, max = 50%] for each output layer. It can be seen that if the total dropout value is small, the training data performs well but the prediction error of the validation data is large. When the total dropout rate is large, the errors of training data and validation data become large. Then, the test revealed that the accuracy of prediction from descending order is worse than that from ascending order. The number of LSTM layers [min = 2, max = 6] and [32, 64, 128, 256, 512] parameters of each layer were tested. After balancing precision and excess risk. The activation function of each layer is set to ReLU, which has better performance than sigmoid and tanh. The specific cost and LSTM framework are shown in Table 1 and Figure 2 below. The last 10% of the training data is defined as the validation data.

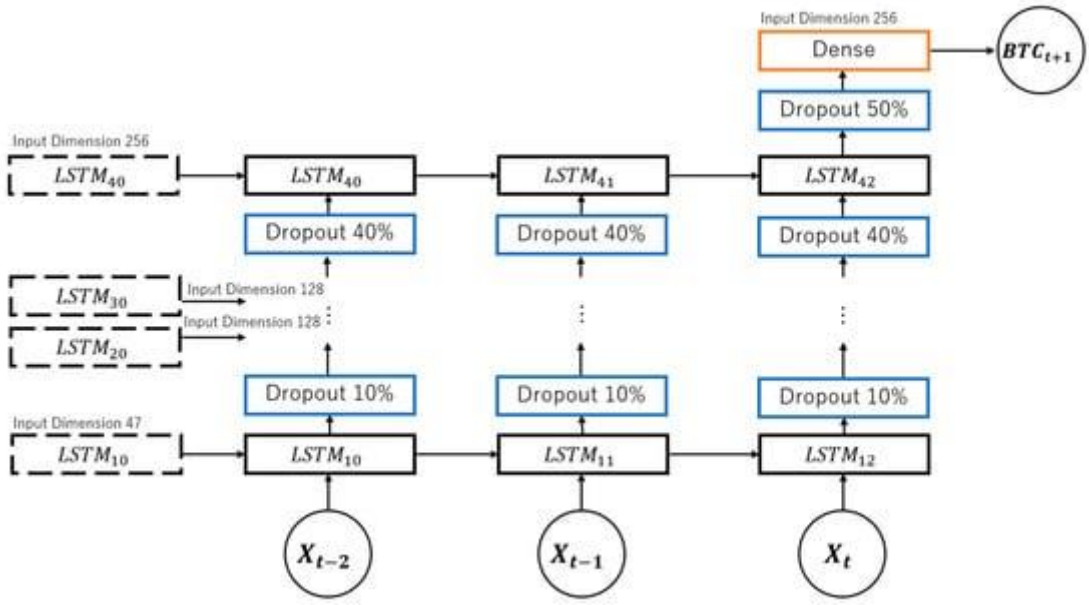


Figure 10. Parameters and framework of LSTM

In addition to the model framework setting, another important hyperparameter for deep learning is epochs. The period value indicates the number of passes to learn train data. The bigger the interval, the smaller the prediction error of the training data. However, when the period is too big, it causes a lot of problems. Therefore, by graphing the training and validation losses in periods 1 and 2 in Figure11 below, period 1 is set to 250 and period 1 is set to 75.

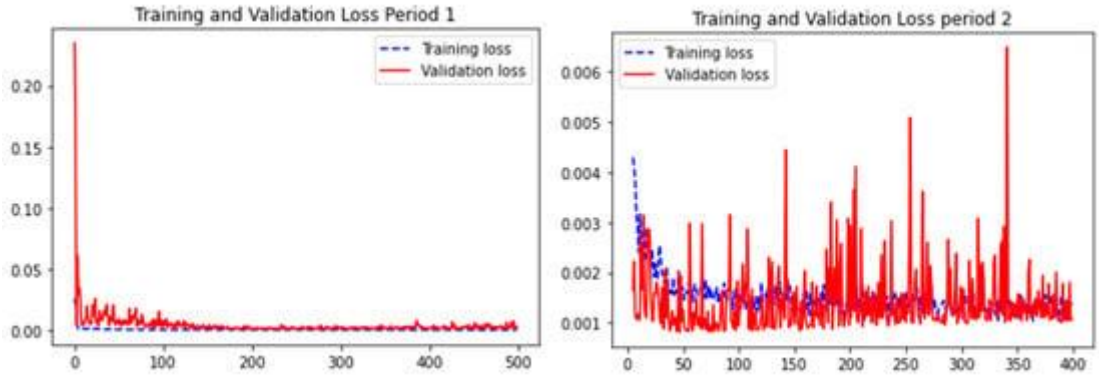


Figure 11. Training and validation loss of LSTM

2. METHODOLOGY

Machine learning is an important branch of artificial intelligence (AI). Depending on whether or not there is a target variable, it can be classified into supervised training, unsupervised training, and strengthening training. The purpose of this study is to predict the future price of Bitcoin, so the regression function and supervised learning are used. The logic of the integration of machine learning is that after the algorithm has been determined, a learner is generated and the learner-high accuracy is obtained by repeatedly training the learner through the training data and validation process. Finally, tests replace students who are trained to evaluate and use data.

The random forest regression and LSTM model training in this paper were performed using the Python open-source machine learning library. sklearn, the library used by Random Forest Regression, uses Keras for LSTM search. Pre-processing and data collection is done by panda.

Random forest is a type of ensemble of multiple regression trees. The advantages are clear, but the predicted results are limited to study samples. The principle of the regression tree is to divide the main group into subgroups using a certain variable index, and the classification is based on minimizing the average squared residuals of each group shown in the equation below.

$$\frac{1}{n_1}\sum_{i=1}^{n_1}\left(y_i-\overline{y_{(1:n_1)}}\right)+\frac{1}{n_2-n_1}\sum_{j=n_1+1}^{n_2}\left(y_j-\overline{y_{(n_1+1:n_2)}}\right)\rightarrow min$$

Regarding parameter settings, the maximum depth of a sub-regression tree is 10, and the number of sub-regression trees in the random forest is 500. I tried the maximum depth of the interval [min = 3, max = 20] and the number of sub-regression trees in the interval [min = 200, max = 1000] respectively. My future tests show that when the maximum depth is greater than 10 or the number of sub-regression trees is greater than 500, the training data and prediction errors do not change.

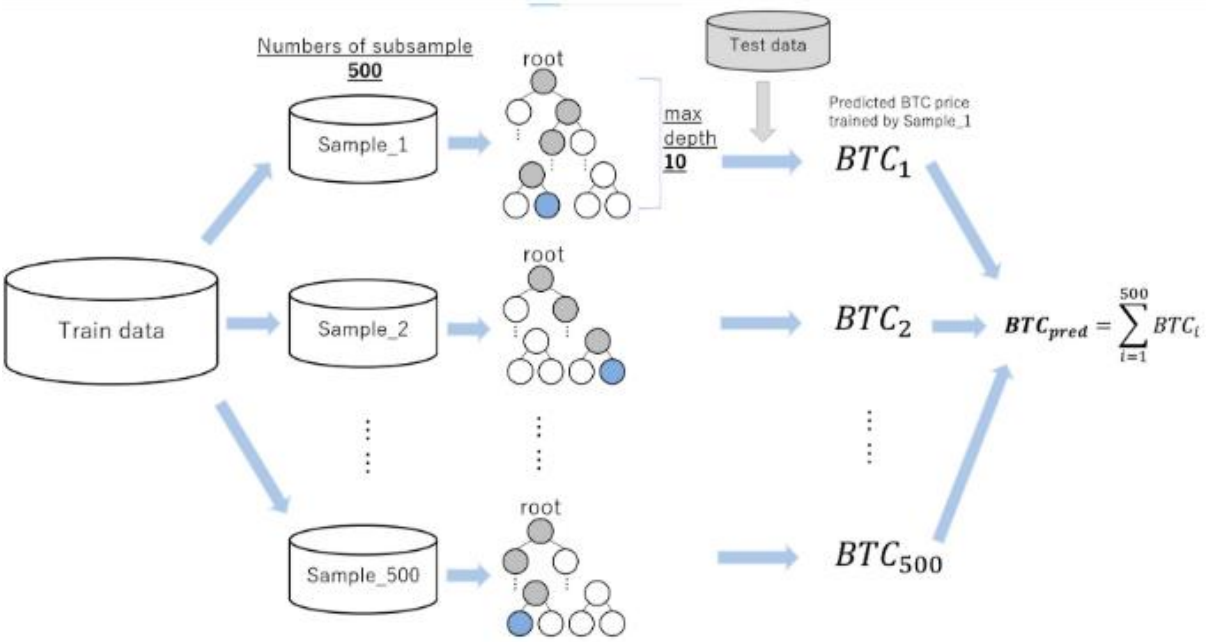


Figure 12. Parameter and framework for Random Forest Regression

As an important criterion to assess the accuracy of machine learning predictions, this study determines the prediction performance of the model using three errors, MAPE (percentage absolute error) and RMSE (root mean square error) and DA (decision accuracy).

$$MAPE = \frac{1}{m}\sum_{t=1}^{m}\left|\frac{y(t)-\hat{y}(t)}{y(t)}\right|$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{t=1}^{m}(y(t)-\hat{y}(t))^2}$$

$$DA = \frac{1}{m}\sum_{t=1}^{m}a(t)\times 100\%$$

In addition to comparing the forecast accuracy of different models to get the performance of each model in predicting the future price of Bitcoin, this study expects to compare the forecast error in explanatory variables to analyse the long-term characteristics of the Bitcoin market.

In addition to the MAPE, RMSE, and DA errors of each prediction result, this paper tests the hypothesis of a significant difference between two different algorithms using the Diebold-Mariano test and the Clark-West test. The principle of DM testing can be summarized in a simple way: given two sequences of forecast errors, then determine the loss function when MSE and MAE.

$$DM_t = \frac{\bar{d}_t}{se(d_t)}$$

Based on Diebold–Mariano's loose assumption, DMt as shown above, is asymptotically distributed in N (0, 1), and finally a one-sided hypothesis test is performed on the statistic DMt. The Clark–West test adds the (et−e′t)2 item in the loss function of the Diebold–Mariano test of MSE, which is also asymptotically distributed in N (0, 1), and finally performs a one-tailed hypothesis test on the statistic f_t.

3. PROBLEM FORMULATION

3.1 Download Crypto Currency Price Data

I am using mwclient library for getting data from Wikipedia API. mwclient is a lightweight Python client library to the MediaWiki API which provides access to most API functionality. It works with Python 2.7 as well as 3.5 and above, and supports MediaWiki 1.16 and above. I am fetching the data of bitcoin from wiki as shown in the figure below:

```
import mwclient
import time

site = mwclient.Site("en.wikipedia.org")
page = site.pages["Bitcoin"]
```

```
revs = list(page.revisions())
```

```
revs[0]
```

```
OrderedDict([('revid', 1149274508),
             ('parentid', 1149262239),
             ('user', 'Jtbobwaysf'),
             ('timestamp',
              time.struct_time(tm_year=2023, tm_mon=4, tm_mday=11, tm_hour=6, tm_min=23, tm_sec=3, tm_wday=1, tm_yday=101, tm_i
sdst=-1)),
             ('comment', '/* Creation */ correct')])
```

```
revs = sorted(revs, key=lambda rev: rev["timestamp"])
```

```
revs[0]
```

```
OrderedDict([('revid', 275832581),
             ('parentid', 0),
             ('user', 'Pratyeka'),
             ('timestamp',
              time.struct_time(tm_year=2009, tm_mon=3, tm_mday=8, tm_hour=16, tm_min=41, tm_sec=7, tm_wday=6, tm_yday=67, tm_is
dst=-1)),
             ('comment', 'creation (stub)')])
```

Figure 13. mwclient code figure

Performing sentiment analysis using transformers pipeline

Pipeline transformer is made by huggingface.io. This tries to answer the various challenges we face in the field of NLP; pre-built models, tokenizers, configurations, various APIs, ready-made pipelines for our ideas, etc. provide. The Transformers package provides the advantage of using pre-built language models and data processing tools. Most models are provided directly and are available in the PyTorch and TensorFlow libraries. The Transformers package requires running TensorFlow or PyTorch and can train models on multiple paths and easily process our text data.

Sentiment analysis is the automated process of labelling data based on its sentiment, such as positive, negative, and neutral. Sentiment analysis enables companies to analyze data at scale, identify insights and automate processes. A natural language processing technique that determines the polarity of a given text. Sentiment analysis comes in many flavours, but one of the most widely used techniques categorizes data as positive, negative, and neutral. It enables data processing at scale and in real time. For example, do you want to analyze thousands of tweets, product reviews or support tickets? Instead of manually sorting through this data, you can use sentiment analysis to automatically understand how people talk about certain topics, make informed decisions based on data, and automate business processes.

In the below figures it is clearly shown how I used sentiment analysis on the data downloaded from Wikipedia using mwclient in above steps:

```
from transformers import pipeline
sentiment_pipeline = pipeline("sentiment-analysis")

def find_sentiment(text):
    sent = sentiment_pipeline([text[:250]])[0]
    score = sent["score"]
    if sent["label"] == "NEGATIVE":
        score *= -1
    return score
```

```
edits = {}

for rev in revs:
    date = time.strftime("%y-%m-%d", rev["timestamp"])

    if date not in edits:
        edits[date] = dict(sentiments=list(), edit_count=0)


    edits[date]["edit_count"] += 1

#     comment = rev["comment"]
    comment = rev.get("comment", "")
    edits[date]["sentiments"].append(find_sentiment(comment))
```

```
from statistics import mean

for key in edits:
    if len(edits[key]["sentiments"]) > 0:
        edits[key]["sentiment"] = mean(edits[key]["sentiments"])
        edits[key]["neg_sentiment"] = len([s for s in edits[key]["sentiments"] if s < 0]) / len(edits[key]["sentiments"])
    else:
        edits[key]["sentiment"] = 0
        edits[key]["neg_sentiment"] = 0

    del edits[key]["sentiments"]
```

Figure 14. Sentiment analysis

As shown in above figure I created a variable edits an stored the sentiment, neg_sentiment data with respective date. Here I am using data frame from pandas library for arranging my data in the form of a table so that it is very systematic and easy for further use. The edits variable is shown in the figure below:

```
{'09-03-08': {'edit_count': 4,
  'sentiment': -0.550525039434433,
  'neg_sentiment': 0.75},
 '09-08-05': {'edit_count': 1,
  'sentiment': 0.7481209635734558,
  'neg_sentiment': 0.0},
 '09-08-06': {'edit_count': 2,
  'sentiment': 0.995745837688446,
  'neg_sentiment': 0.0},
 '09-08-14': {'edit_count': 1,
  'sentiment': 0.9300214052200317,
  'neg_sentiment': 0.0},
 '09-10-13': {'edit_count': 2,
  'sentiment': -0.22749841213226318,
  'neg_sentiment': 0.5},
 '09-11-18': {'edit_count': 1,
  'sentiment': 0.8839512467384338,
  'neg_sentiment': 0.0},
 '09-12-08': {'edit_count': 1,
  'sentiment': -0.9869275689125061,
  'neg_sentiment': 1.0},
 '09-12-17': {'edit_count': 1,
  'sentiment': -0.9975171089172363,
  'neg_sentiment': 1.0},
 '10-02-23': {'edit_count': 1,
  'sentiment': -0.9994946718215942,
  'neg_sentiment': 1.0},
 '10-03-18': {'edit_count': 1,
  'sentiment': 0.875878095626831,
  'neg_sentiment': 0.0},
 '10-04-13': {'edit_count': 4,
  'sentiment': 0.844355896115303,
  'neg_sentiment': 0.0},
 '10-04-15': {'edit_count': 8,
  'sentiment': 0.5781015157699585,
  'neg_sentiment': 0.125},
```


Figure 15. Edits variable data

Using Pandas library

Pandas is an open-source library designed specifically for easy and intuitive work with linked and tagged data. It provides a variety of data structures and operations for manipulating numerical and time series data. This library is built on top of the NumPy library. Pandas is fast and has high performance and productivity for users.

Pandas DataFrame is a two-dimensional, resizable, homogeneous tabular data structure with labelled axes (rows and columns). The data frame is a two-dimensional data structure, which is the data arranged in a table in rows and columns. A Pandas DataFrame consists of three main components, data, rows, and columns. A Pandas DataFrame will be created by loading a real-world database from an existing repository, which can be a SQL database, a CSV file, an Excel file. Pandas DataFrame consists of lists, dictionaries and lists of dictionaries etc. can be made.

```
import pandas as pd

edits_df = pd.DataFrame.from_dict(edits, orient="index")
```

I am using pandas DataFrame to convert the sentiment data as shown in Figure 15 which is one-dimensional and unsystematic into the highly arranged manner. Basically, in the form of a table in rows and columns as shown in the table 2 below. Also, I am taking the dataset from 8th of March 2009 to the present date from datetime library and then again using DataFrame for converting these dates into the table and further reindexing the edits DataFrame with these dates processed as shown in the figure below:

	edit_count	sentiment	neg_sentiment
09-03-08	4	-0.550525	0.75
09-08-05	1	0.748121	0.00
09-08-06	2	0.995746	0.00
09-08-14	1	0.930021	0.00
09-10-13	2	-0.227498	0.50
...	...	...	...
23-03-29	1	-0.999457	1.00
23-03-30	1	-0.997922	1.00
23-04-03	1	-0.996604	1.00
23-04-11	2	0.000964	0.50
23-04-22	1	-0.998953	1.00

2596 rows × 3 columns

Table 2. Edits DataFrame for sentiment data

```
from datetime import datetime

dates = pd.date_range(start="2009-03-08", end=datetime.today())
```

```
dates
```

```
DatetimeIndex(['2009-03-08', '2009-03-09', '2009-03-10', '2009-03-11',
               '2009-03-12', '2009-03-13', '2009-03-14', '2009-03-15',
               '2009-03-16', '2009-03-17',
               ...
               '2023-04-15', '2023-04-16', '2023-04-17', '2023-04-18',
               '2023-04-19', '2023-04-20', '2023-04-21', '2023-04-22',
               '2023-04-23', '2023-04-24'],
              dtype='datetime64[ns]', length=5161, freq='D')
```

	edit_count	sentiment	neg_sentiment
2009-03-08	NaN	NaN	NaN
2009-03-09	NaN	NaN	NaN
2009-03-10	NaN	NaN	NaN
2009-03-11	NaN	NaN	NaN
2009-03-12	NaN	NaN	NaN
...	...	...	...
2023-04-20	0.633333	-0.117631	0.162500
2023-04-21	0.700000	-0.117651	0.179167
2023-04-22	0.600000	-0.084758	0.145833
2023-04-23	0.600000	-0.084758	0.145833
2023-04-24	0.600000	-0.084758	0.145833

5161 rows × 3 columns

Table 3. Rolling Edits table

As in the above figure there are lots of null values which can affect our model. They can reduce the accuracy of our model and also affect the predictions made in further steps. So, dropping the null values form rolling edits table in order to clean our data. This step of data cleaning is also called as the Data Pre-processing.

Now we get our rolling edits data processed and it is ready for further usage in our model in order to go for predictions. So I am storing this data of around 5161 rows in the form of a csv file.

So that I do not need to do these steps again and again when I use the model in order to save the time and efforts made by the model.

	edit_count	sentiment	neg_sentiment
2009-04-06	1.266667	-0.193400	0.233987
2009-04-07	1.266667	-0.193400	0.233987
2009-04-08	1.266667	-0.193400	0.233987
2009-04-09	1.266667	-0.193400	0.233987
2009-04-10	1.266667	-0.193400	0.233987
...	...	...	...
2023-04-20	0.633333	-0.117631	0.162500
2023-04-21	0.700000	-0.117651	0.179167
2023-04-22	0.600000	-0.084758	0.145833
2023-04-23	0.600000	-0.084758	0.145833
2023-04-24	0.600000	-0.084758	0.145833

5132 rows × 3 columns

Table 4. Processed rolling edits table

```
rolling_edits.to_csv("wikipedia_edits.csv")
```

In the above figure I am storing my data in the csv file by the name "wikipedia_edits.csv". This data contains 5161 rows and 3 columns of "edit_count", "sentiment", "neg_sentiment" along with the respective dates.

Downloading the bitcoin price data

I am using yfinance library for downloading the price data of crypto currency. Yfinance is one of the popular Python modules used to collect data online, and with it we can collect Yahoo's financial data. With the help of the yfinance

module, we receive and collect the company's financial data (financial indicators, etc.), as well as marketing data history, using the company's functions. But before we learn more about this module and its implementation and application, we need to install the yfinance module in our system (because it is not a built-in module in Python).

Downloading the bitcoin price data from 17th April 2014 to present date using yfinance library. The data contains 7 columns: 'Date', 'Open', 'High', 'Low', 'Close', 'AdjClose' and 'Volume' with total 3132 rows as shown below:

Date	Open	High	Low	Close	Adj Close	Volume
2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	21056800
2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	34483200
2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	37919700
2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	36863600
2014-09-21	408.084991	412.425995	393.181000	398.821014	398.821014	26580100
...	...	...	...	...	...	...
2023-04-10	28336.027344	29771.464844	28189.271484	29652.980469	29652.980469	19282400094
2023-04-11	29653.679688	30509.083984	29609.300781	30235.058594	30235.058594	20121259843
2023-04-12	30231.582031	30462.480469	29725.574219	30139.052734	30139.052734	18651929926
2023-04-13	29892.740234	30539.845703	29878.623047	30399.066406	30399.066406	17487721001
2023-04-14	30407.134766	31005.607422	30366.443359	30811.263672	30811.263672	20907202560

3132 rows × 6 columns

Table 5. Bitcoin price data

After getting the raw data, we have to clean the data or undergo data pre-processing. We can analyse that the columns 'Close' and 'Adj Close' have the same data. So, it is better to delete the column "Adj Close" because I don't need redundant data in my model for future prediction.

Exploratory Data Analysis

EDA is an approach to analysing the data using visual techniques. It is used to discover trends, and patterns, or to check assumptions with the help of statistical summaries and graphical representations. While performing the EDA of the Bitcoin Price data we will analyze how prices of the cryptocurrency have moved over the period of time and how the end of the quarters affects the prices of the currency.

Now plotting the graph of closing price of bitcoin from our dataset as shown in the figure below:

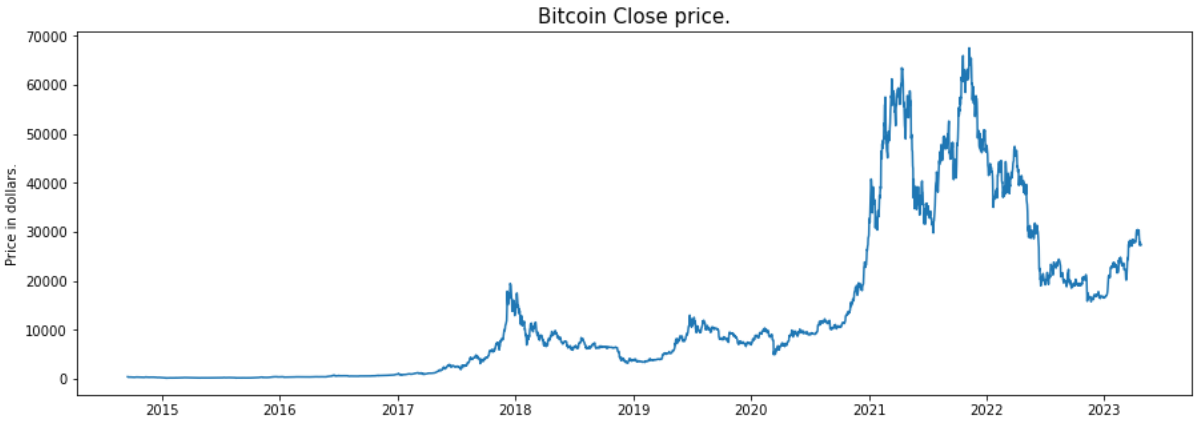


Fig. 16 Variation in the price of bitcoin

Now I am checking for the null values if any present in the DataFrame to proceed further: there are no null values present.

So, in order to check for the patterns in the dataset I have created the distribution plot for continuous features from the dataset i.e. ‘close’, ‘open’, ‘high’, ‘low’ as shown in the figure below:

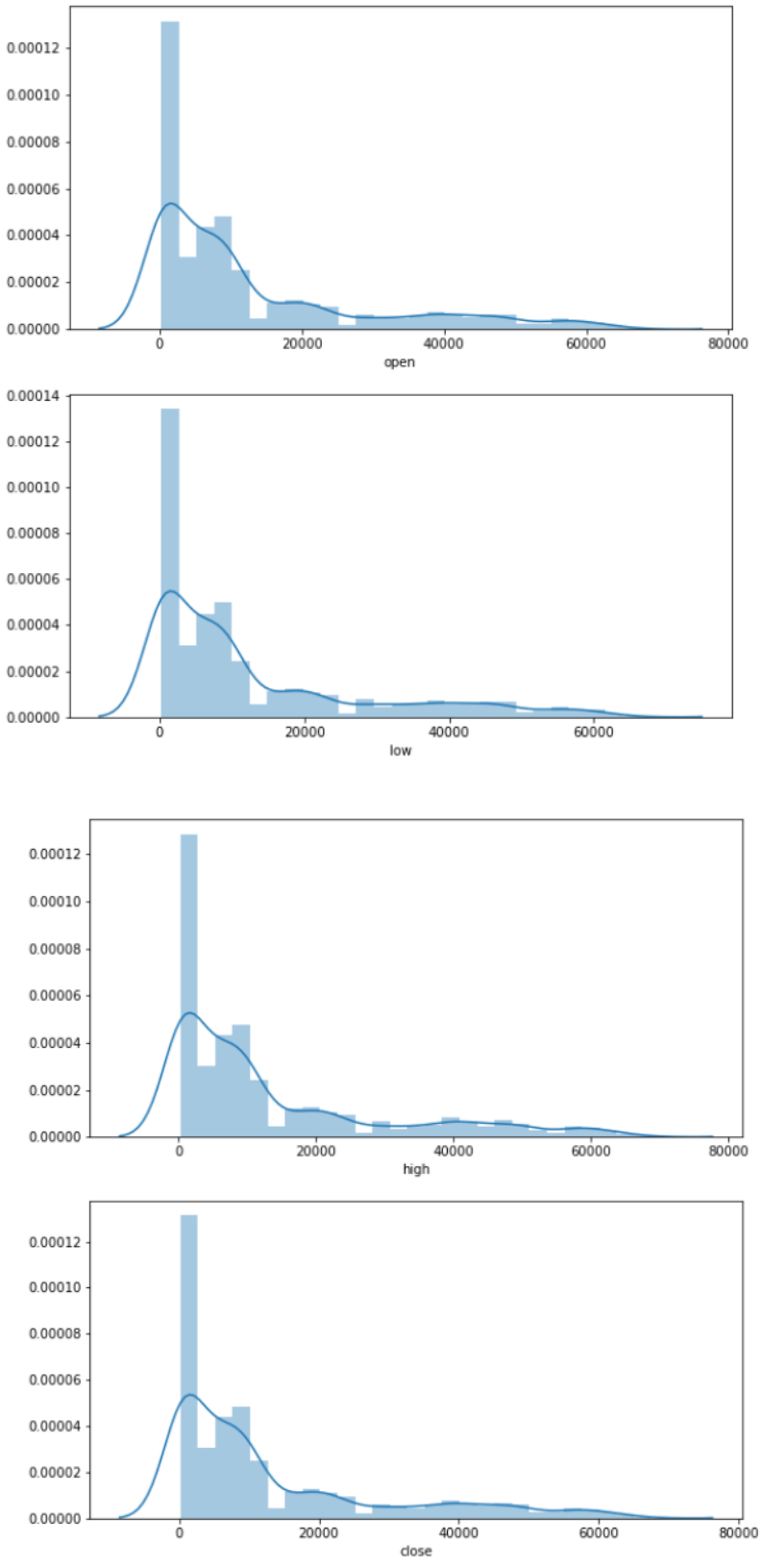


Fig. 17 Distribution plot for the OHLC data

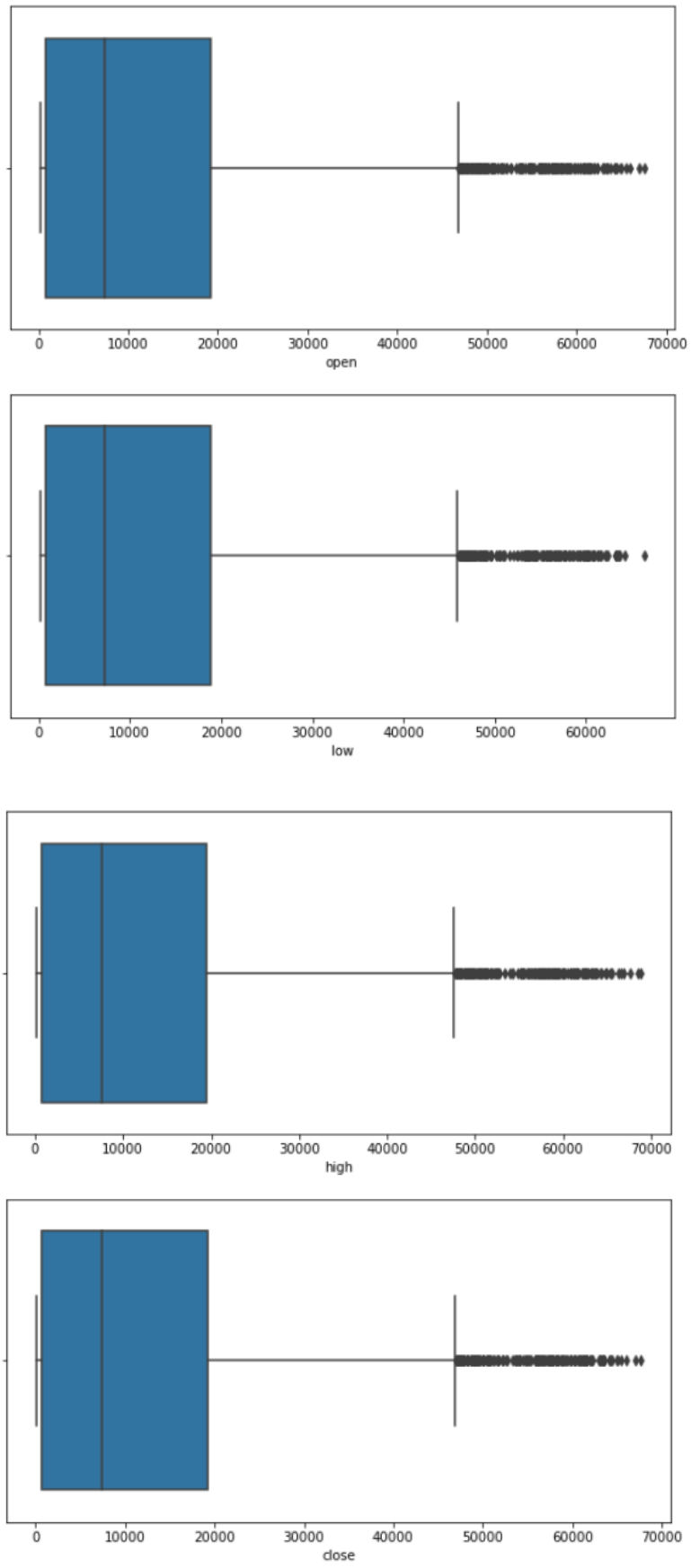


Fig. 18 Box plot for the OHLC data

3.2 Preparing Data for Machine Learning

Until now I have two datasets one is of sentiment data stored in rolling_edits variable as described in previous steps and the data was saved in a .csv file named as “Wikipedia_edits.csv” as shown in table 4 on page number 60.

The second data is of the crypto currency bitcoin which I downloaded recently from yahoo finance as shown in table 5 on page number 61.

Now finally I will merge both of these datasets into a single table to perform forward steps. The merged data is shown in table 6. After that I will create new features in my dataset which will further come into use during time of training data and then I will check whether the variables are balanced or not using pie chart and the correlation between these features using pie chart and heat maps.

	open	high	low	close	volume	edit_count	sentiment	neg_sentiment
2014-09-17	465.864014	468.174011	452.421997	457.334015	21056800	0.533333	-0.109741	0.154444
2014-09-18	456.859985	456.859985	413.104004	424.440002	34483200	0.566667	-0.142785	0.187778
2014-09-19	424.102997	427.834991	384.532013	394.795990	37919700	0.600000	-0.176097	0.221111
2014-09-20	394.673004	423.295990	389.882996	408.903992	36863600	0.600000	-0.176097	0.221111
2014-09-21	408.084991	412.425995	393.181000	398.821014	26580100	0.600000	-0.109894	0.187778
...	...	...	...	...	...	...	...	...
2023-04-21	28249.230469	28349.968750	27177.365234	27276.910156	20759504330	0.700000	-0.117651	0.179167
2023-04-22	27265.894531	27872.142578	27169.570312	27817.500000	13125734602	0.600000	-0.084758	0.145833
2023-04-23	27816.144531	27820.244141	27400.314453	27591.384766	12785446832	0.600000	-0.084758	0.145833
2023-04-24	27591.730469	27979.982422	27070.849609	27525.339844	17703288330	0.600000	-0.084758	0.145833
2023-04-26	28281.115234	28446.505859	28262.826172	28349.996094	18316503040	0.600000	-0.084758	0.145833

3143 rows × 8 columns

Table 6. Final data after merging

So now after merging the dataset I am creating two new important features: ‘tomorrow’ and ‘target’. The tomorrow feature or column will show the closing price for next day i.e., what is the predicted price for a new data in future for

trading. The target column will compare the todays closing price and tomorrow closing price to give us the signal whether the market will go up or down in future. This is basically a signal for future trading.

	open	high	low	close	volume	edit_count	sentiment	neg_sentiment	tomorrow	target
2014-09-17	465.864014	468.174011	452.421997	457.334015	21056800	0.533333	-0.109741	0.154444	424.440002	0
2014-09-18	456.859985	456.859985	413.104004	424.440002	34483200	0.566667	-0.142785	0.187778	394.795990	0
2014-09-19	424.102997	427.834991	384.532013	394.795990	37919700	0.600000	-0.176097	0.221111	408.903992	1
2014-09-20	394.673004	423.295990	389.882996	408.903992	36863600	0.600000	-0.176097	0.221111	398.821014	0
2014-09-21	408.084991	412.425995	393.181000	398.821014	26580100	0.600000	-0.109894	0.187778	402.152008	1
...	...	...	...	...	...	...	...	...	...	...
2023-04-21	28249.230469	28349.968750	27177.365234	27276.910156	20759504330	0.700000	-0.117651	0.179167	27817.500000	1
2023-04-22	27265.894531	27872.142578	27169.570312	27817.500000	13125734602	0.600000	-0.084758	0.145833	27591.384766	0
2023-04-23	27816.144531	27820.244141	27400.314453	27591.384766	12785446832	0.600000	-0.084758	0.145833	27525.339844	0
2023-04-24	27591.730469	27979.982422	27070.849609	27525.339844	17703288330	0.600000	-0.084758	0.145833	28349.996094	1
2023-04-26	28281.115234	28446.505859	28262.826172	28349.996094	18316503040	0.600000	-0.084758	0.145833	NaN	0

3143 rows × 10 columns

Table 7. Added tomorrow and target feature in data

I have added the target feature which is a signal whether to buy or not we will train our model to predict this only. But before proceeding let's check whether the target is balanced or not using a pie chart.

A pie chart is a circular statistical plot that can only show a series. The area of the chart is a percentage of the total data given. The area of the pie slice represents the percentage of pieces of data. A piece of the pie is called the whole. The area of the curve is determined by the length of the arc of the arc. The area of the curve shows the relative percentage of that section relative to the entire data. Pie charts are used in business presentations and provide a quick overview of sales, operations, research results, resources, and more.

The pie chart for the target variable is shown below in figure 19:

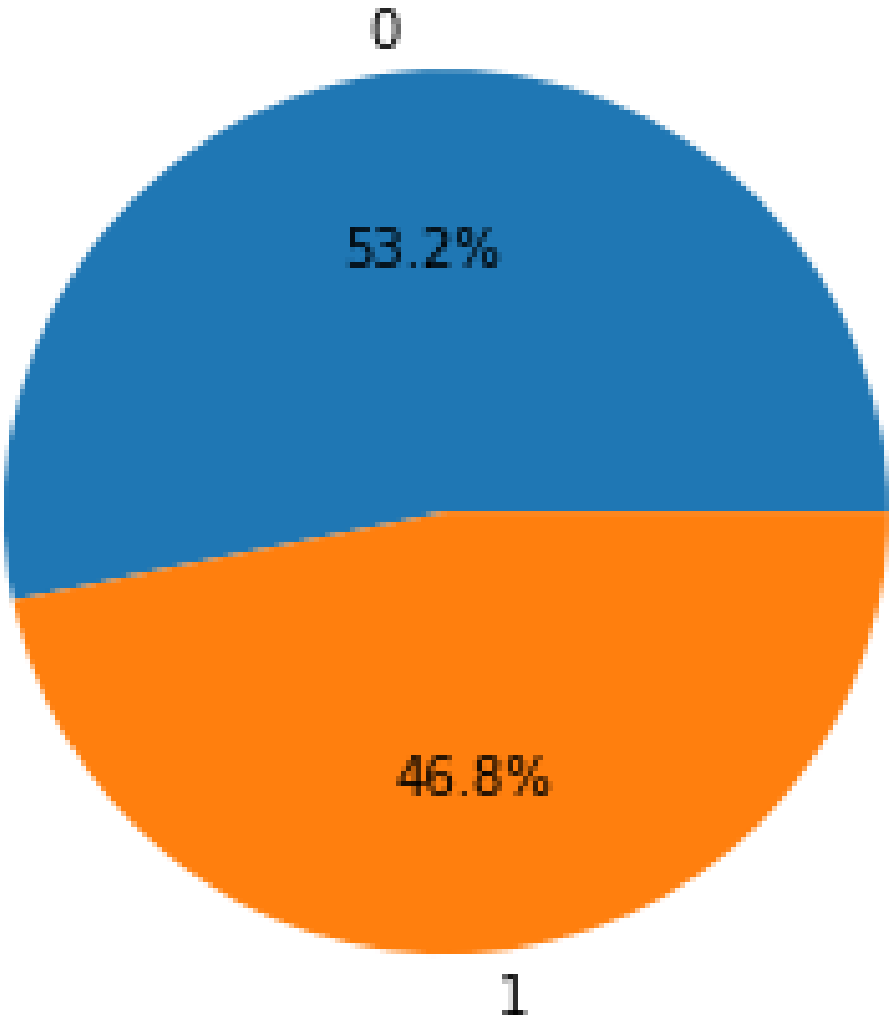


Fig 19. Pie char for target variable

In this pie chat, 0's represents the negative signal for buying the crypto means the market will fall and the 1's represent the positive signal which denotes the right time for purchasing the currency.

From this pie chart we can observe that the number of 0's and the number of 1's is almost equal or we can say that our data is balanced.

Whenever we add new features to our dataset, we have to ensure that there are no highly correlated features as they do not help in the learning process of the algorithm. For this we will create a heat map to verify or check the correlation between features in our data.

The heat map is created using seaborn library in python, Seaborn is a great visualization library for plotting statistical graphs in Python. It provides standard styles and a nice colour palette to make statistical charts more attractive. It is built on top of the Matplotlib library and is tightly coupled to the data structures of pandas.

Seaborn aims to make visualization a central part of learning and understanding data. It provides a database-oriented API so that you can switch between different visual representations of the same variable to better understand the database.

Plots are mainly used to visualize the relationship between variables. The variables can be numerical or categorical such as group, class or division. Beach divides the plans into these categories – Categorical plots, Distribution plots, Matrix plots and Multi plot grids.

A heat map is defined as a graphical representation of data using colour to visualize matrix values. In this case, lighter colours, mostly red, are used to represent more general values or high activity, and darker colours are chosen to represent less general or activity values. Heatmaps are also known as shadow matrices.

The heat map for our data is show below in figure 20:

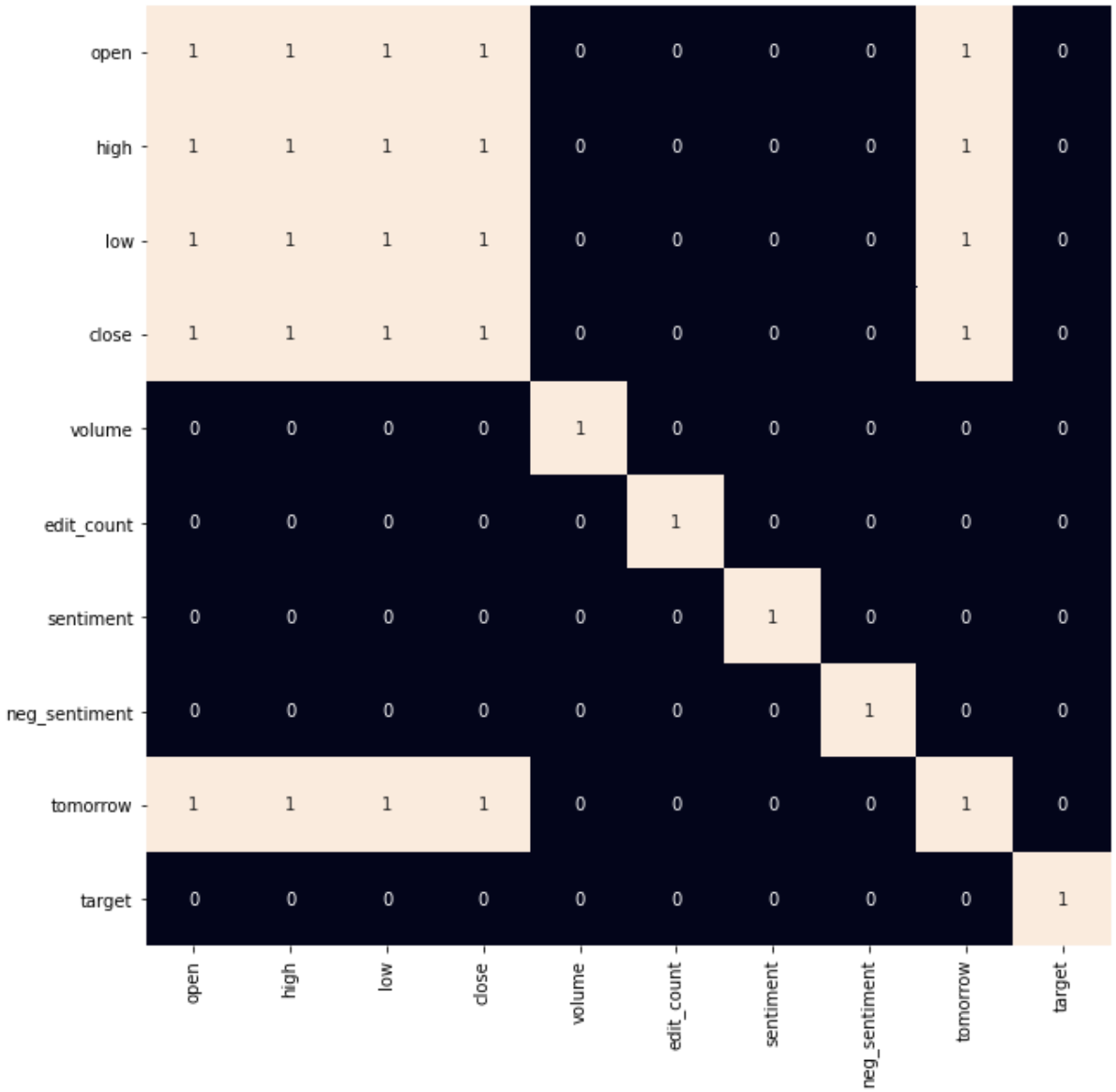


Fig 20. Heat map

From the above heatmap, we can say that there is a high correlation between OHLC (open, high, low, close) which is pretty obvious, and the added features are not highly correlated with each other or previously provided features which means that we are good to go and build our model.

I also created a scatter plot using seaborn between open and closing price of the currency as shown below:

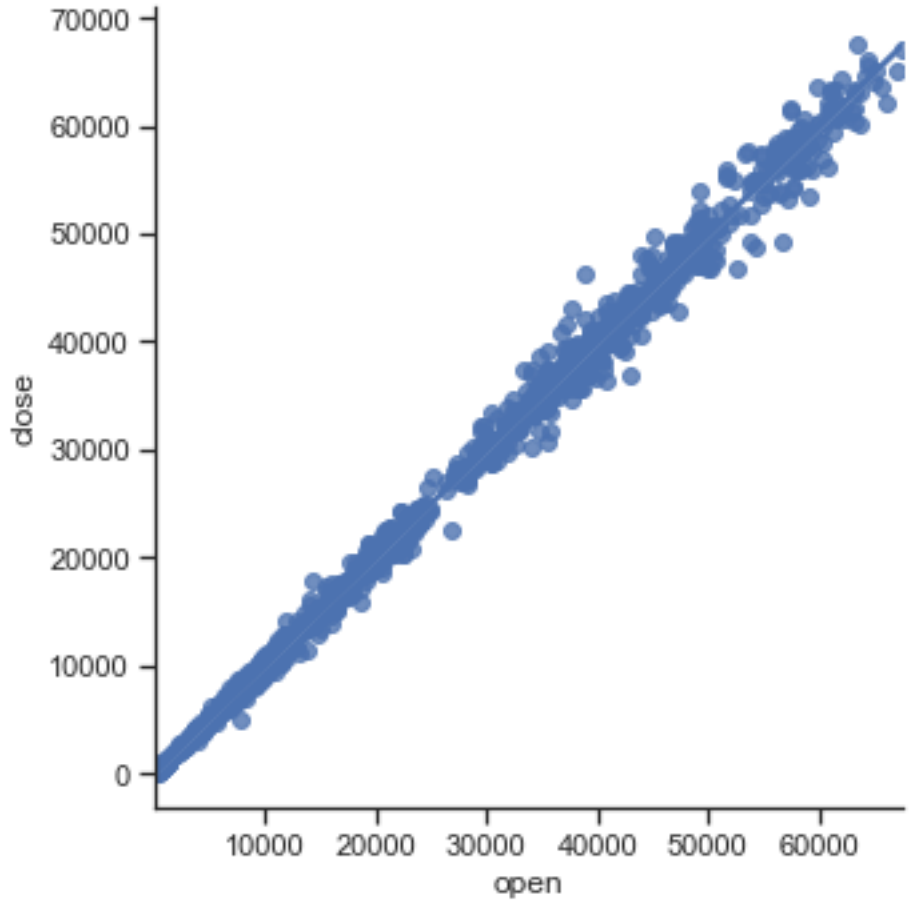


Fig 21. Scatter Plot between open and closing price

3.3 Training Baseline ML Model

I am using Random Forest classifier to create my model using sklearn. ensemble python library.

Scikit Learn library python:

Scikit-learning is an open-source data analysis library and the gold standard for Machine Learning (ML) in the Python ecosystem. Key concepts and features: Algorithmic decision-making techniques, including: Classification (identification and classification of data based on patterns) and Regression: (forecasting or predicting data values based on the average of available and

projected data) and Clustering: (automatic grouping of similar data into the database).

Algorithms that support predictive analysis, from simple linear regression to neural network pattern recognition. Working with NumPy, pandas and matplotlib libraries. ML is a technology that allows computers to learn from input data and build/train predictive models without explicit programming. ML is a subset of artificial intelligence (AI). Whether you're just looking for an introduction to ML, want to get up and running quickly, or looking for the latest ML research tools, you'll find that scikit-learning is both well-documented and easy to learn/use.

As a high-level library, it allows you to define a predictive data model in a few lines of code, and then fit that model to your data. It integrates well with other Python libraries such as matplotlib for plotting and plotting, numeric for array vectorization, and pandas for data frames.

Random Forest classifier python:

Random forest or random decision forest is a supervised Machine Learning algorithm used for classification, regression and other problems using decision trees. A random forest classifier generates a set of decision trees from a randomly selected part of the training set. This is basically a set of decision trees (DT) from a randomly selected part of the training set, and then collect noise from different decision trees to determine the final prediction.

Random Forest fits several decision tree classifiers in different database subsamples and uses the average to improve prediction accuracy and control overfitting. The sub-sample size is controlled by the max_samples parameter if bootstrap = True (default), otherwise the entire data set is used to build each tree.

```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=100, min_samples_split=50, random_state=1)

train = btc.iloc[:-200]
test = btc[-200:]

predictors = ["close", "volume", "open", "high", "low", "edit_count", "sentiment", "neg_sentiment"]
model.fit(train[predictors], train["target"])
```

After creating the random forest classifier model and fitting the training and testing data in our model I am importing the precision score from sklearn metrices to check the precision of my model.

Precision Score:

Classification models are used in classification problems to predict the target class of data samples. The classification model assumes that each instance belongs to one class or another group. It is important to evaluate the performance of classification models in order to reliably use these models in industry to solve real-world problems. In the machine learning classification model, performance measures are used to evaluate the performance of the machine learning classification model in a certain context. This performance measure includes precision, accuracy, recall, and F1-score. Model performance is very important for machine learning because it helps to understand the strengths and limitations of these models when making predictions in new situations.

The accuracy of the model shows that the positive predictions are actually correct. Accuracy is also known as positive predictive value. Accuracy is used in conjunction with the return of false positives and false negatives. Accuracy affects class distribution. If there are more samples in the minority class, the accuracy will be lower. Accuracy can be thought of as a measure of accuracy or quality. If we want to reduce false negatives, we will choose a model with high accuracy. Conversely, if we want to reduce false positives, we will choose a

model with a high recall. Accuracy is mainly used when we need to predict the positive class, and false positives have a higher rate than false negatives, such as medical diagnosis or spam filtering. For example, if a model is 99% accurate, but only 50% accurate, that means half the time it predicts that an email is spam, it's not spam.

The precision score is a useful measure of the success of prediction when the classes are very imbalanced. Mathematically, it represents the ratio of true positive to the sum of true positive and false positive.

Precision Score = TP / (FP + TP)

From the above formula, you could notice that the value of false-positive would impact the precision score. Thus, while building predictive models, you may choose to focus appropriately to build models with lower false positives if a high precision score is important for the business requirements.

```
from sklearn.metrics import precision_score

preds = model.predict(test[predictors])
preds = pd.Series(preds, index=test.index)
precision_score(test["target"], preds)
```

From my model I get the precision score of -> 0.5357142857142857 which is less, the value of precision score lies between 0 and 1. In the forward steps I will work on my model to increase the precision.

3.4 Evaluating Error with Back Testing

The concept of backtesting is actually quite simple. This is to save the last part of the data as Test data so that the model can be evaluated against the test data. Let's see how it works step by step.

1. Divide the data into Training data and Test data:

Before building the model, we divided the data into two parts, namely Training data and Test data.

2. Build a model based on training data:

Then, we can build a prediction model based on training data and Prophet.

3. Use the model to predict the duration of the test data:

After the model is built, we can use the model to predict the duration of the test data.

4. Compare between expected and actual:

We have an answer by saying that we have real data for this test period. So, we can compare the predicted value with this actual data and see how big or small the difference between the predicted and the actual is.

The code I used for backtesting is shown below:

```
def predict(train, test, predictors, model):
    model.fit(train[predictors], train["target"])
    preds = model.predict(test[predictors])
    preds = pd.Series(preds, index=test.index, name="predictions")
    combined = pd.concat([test["target"], preds], axis=1)
    return combined
```

```
def backtest(data, model, predictors, start=1095, step=150):
    all_predictions = []

    for i in range(start, data.shape[0], step):
        train = data.iloc[0:i].copy()
        test = data.iloc[i:(i+step)].copy()
        predictions = predict(train, test, predictors, model)
        all_predictions.append(predictions)

    return pd.concat(all_predictions)
```

Fig 22. Evaluating error with Back Testing

3.5 Using an XG Boost Model

XGBoost is an open-source Python library that provides a gradient boosting framework. It helps to produce highly efficient, flexible and portable models.

When it comes to prediction, XGBoost outperforms other algorithms or machine learning frameworks. This is due to increased accuracy and productivity. Combine multiple models into one model to correct errors in the existing models.

XGBoost is built on top of the gradient booster framework. Gradient boosting is a machine learning technique used for classification, regression, and clustering problems. It optimizes the model when predictions are made. In this method, different models are grouped together to perform the same task.

The basic model is known as weak learning. They work on the principle that a weak student makes bad guesses when alone, but makes the best guesses when in a group.

XGBoost creates strong learners based on weak learners. Adding routine models. Therefore, the error of the weak model is corrected by the next model of the chain to obtain the optimal solution. This is known as an ensemble.

```
from xgboost import XGBClassifier

model = XGBClassifier(random_state=1, learning_rate=.1, n_estimators=200)
predictions = backtest(btc, model, predictors)

predictions
```

	target	predictions
2017-09-16	0	0
2017-09-17	1	0
2017-09-18	0	1
2017-09-19	0	0
2017-09-20	0	0
...	...	...
2023-04-21	1	0
2023-04-22	0	0
2023-04-23	0	0
2023-04-24	1	0
2023-04-26	0	0

2048 rows × 2 columns

Table 8. Prediction using XGBoost model

3.6 Improving Precision with Trends

Machine Learning and Deep Learning models are around us in modern organizations. The number of AI use cases is growing rapidly with the rapid development of new algorithms, cheaper computing, and more data. Banking, healthcare, education, manufacturing, construction, etc., every industry has relevant machine learning and deep learning applications. One of the biggest challenges in all these ML and DL projects across industries is model optimization.

One of the keys to success is model accuracy and performance. Model performance is primarily a technical factor, and for some machine learning and deep learning use cases, there is no point in deploying it unless the model is accurate.

```
def compute_rolling(btc):
    horizons = [2, 7, 60, 365]
    new_predictors = ["close", "sentiment", "neg_sentiment"]

    for horizon in horizons:
        rolling_averages = btc.rolling(horizon, min_periods=1).mean()

        ratio_column = f"close_ratio_{horizon}"
        btc[ratio_column] = btc["close"] / rolling_averages["close"]

        edit_column = f"edit_{horizon}"
        btc[edit_column] = rolling_averages["edit_count"]

        rolling = btc.rolling(horizon, closed="left", min_periods=1).mean()
        trend_column = f"trend_{horizon}"
        btc[trend_column] = rolling["target"]

        new_predictors += [ratio_column, trend_column, edit_column]
    return btc, new_predictors
```

Fig 23. Improving precision with trends

There is no one-size-fits-all strategy to improve existing machine learning and deep learning models. I will review a set of guidelines and best practices that can be evaluated to consistently identify potential sources of improvement in model accuracy and performance.

	open	high	low	close	volume	edit_count	sentiment	neg_sentiment	tomorrow	target
2014-09-17	465.864014	468.174011	452.421997	457.334015	21056800	0.533333	-0.109741	0.154444	424.440002	0
2014-09-18	456.859985	456.859985	413.104004	424.440002	34483200	0.566667	-0.142785	0.187778	394.795990	0
2014-09-19	424.102997	427.834991	384.532013	394.795990	37919700	0.600000	-0.176097	0.221111	408.903992	1
2014-09-20	394.673004	423.295990	389.882996	408.903992	36863600	0.600000	-0.176097	0.221111	398.821014	0
2014-09-21	408.084991	412.425995	393.181000	398.821014	26580100	0.600000	-0.109894	0.187778	402.152008	1
...	...	...	...	...	...	...	...	...	...	...
2023-04-21	28249.230469	28349.968750	27177.365234	27276.910156	20759504330	0.700000	-0.117651	0.179167	27817.500000	1
2023-04-22	27265.894531	27872.142578	27169.570312	27817.500000	13125734602	0.600000	-0.084758	0.145833	27591.384766	0
2023-04-23	27816.144531	27820.244141	27400.314453	27591.384766	12785446832	0.600000	-0.084758	0.145833	27525.339844	0
2023-04-24	27591.730469	27979.982422	27070.849609	27525.339844	17703288330	0.600000	-0.084758	0.145833	28349.996094	1
2023-04-26	28281.115234	28446.505859	28262.826172	28349.996094	18316503040	0.600000	-0.084758	0.145833	NaN	0

trend_2	close_ratio_7	edit_7	trend_7	close_ratio_60	edit_60	trend_60	close_ratio_365	edit_365	trend_365
NaN	1.000000	0.533333	NaN	1.000000	0.533333	NaN	1.000000	0.533333	NaN
0.0	0.962696	0.550000	0.000000	0.962696	0.550000	0.000000	0.962696	0.550000	0.000000
0.0	0.927789	0.566667	0.000000	0.927789	0.566667	0.000000	0.927789	0.566667	0.000000
0.5	0.970419	0.575000	0.333333	0.970419	0.575000	0.333333	0.970419	0.575000	0.333333
0.5	0.956729	0.580000	0.250000	0.956729	0.580000	0.250000	0.956729	0.580000	0.250000
...	...	...	...	...	...	...	...	...	...
0.0	0.932216	0.666667	0.142857	1.038110	1.554444	0.433333	1.191986	1.514247	0.465753
0.5	0.962443	0.661905	0.285714	1.056418	1.545556	0.450000	1.217347	1.514247	0.468493
0.5	0.967648	0.652381	0.285714	1.045579	1.536667	0.450000	1.209176	1.514155	0.468493
0.0	0.974706	0.642857	0.285714	1.040725	1.527222	0.450000	1.208014	1.514064	0.468493
0.5	1.014416	0.628571	0.285714	1.068436	1.517778	0.466667	1.246020	1.513973	0.468493

Table 9. Various trends in the dataset

As shown in the above table I added a lot of new trends in my dataset which helps my model to predict accurate results and give me good precision scores. This makes my model more accurate and much useful and trustworthy for trading various crypto currencies.

3.7 Generating Future Predictions

Finally generating the future predictions, model will predict the price accurately and give us the signal whether the price will go up or down using the target variable I created in the dataset. The predictions are shown below in the table:

	target	predictions
2017-09-16	0	1
2017-09-17	1	0
2017-09-18	0	0
2017-09-19	0	1
2017-09-20	0	1
...	...	...
2023-04-21	1	1
2023-04-22	0	0
2023-04-23	0	1
2023-04-24	1	1
2023-04-26	0	1

2048 rows × 2 columns

Table 10. Final predictions

3. CONCLUSION AND DISCUSSION

Bitcoin and other crypto currencies are a decentralized digital currency that uses cryptography for security and is not controlled by governments or financial institutions. It was created in 2008 with a paper entitled "Bitcoin: A Peer-to-Peer (P2P) Electronic Cash System" by an individual or a group of individuals using the pseudonym Satoshi Nakamoto (2008). Bitcoin transactions are recorded on the public blockchain, which allows anyone to see the history of a particular Bitcoin. The decentralized nature of Bitcoin allows it to be used independently of central banks and can be instantly transferred worldwide. It became popular as a medium of exchange and store of value. In the last 10 years, in November 2021, one coin has exceeded USD 68,000, and the total value once exceeded USD 1.2 trillion.

Based on the need to avoid price risk as Bitcoin, this study chooses a machine learning random forest regression algorithm and a neural network algorithm LSTM model to predict Bitcoin's price. I mainly focus on the performance of random forest regression in predicting Bitcoin price while using LSTM prediction results as a comparison. Random forest regression is a variant of random forest regression. Unlike black-box neural network technology, random forest regression, like machine learning, can provide the value of each explanatory variable in predicting Bitcoin through the results of weak learners.

Although there are already several applications of ML for the cryptocurrency market, there are some aspects that researchers and market experts may find informative. In particular, the more recent period shows market confusion and bear market conditions since mid-2017; using not only trade variables but also network variables as important inputs to the data set; and provide a thorough statistical and economic analysis of reviewed trading strategies in the crypto market. It is important to note that the price during the approval period

experienced an explosive behaviour, followed by a sudden and significant drop; However, mean reversion is positive. In the test sample, the price is more stable, but the average return is negative. Therefore, analysing the performance of trading strategies in this strict framework can be seen as a rigorous test of their profitability.

The accuracy of predictions varies widely across models and cryptocurrencies, and there is no clear pattern that allows us to determine which model is superior or which cryptocurrency is the most predictive during the trial period or the trial period. However, in general, the prediction accuracy of individual models seems to be low compared to similar studies. This is not surprising, since the best-in-class model is based on the average increase in one-step-ahead returns rather than on reducing the forecast error. The main apparent pattern is that the prediction accuracy in the validation subsample is lower than in the test sample, which is likely due to significant differences in price trends in the previous period.

Taking into account the poor prediction performance of individual models in the research sample and the results have been reported in the literature that the collection of models gives the best results, the analysis of profitability in the cryptocurrency market is carried out according to the trading strategy. The rule states that a long position in the market will be created if at least four, five or six different patterns agree on a positive trade mark for the next day. trading strategy only involves the creation of long positions, assuming that short selling is difficult or impossible in the cryptocurrency market. This restriction is very necessary because the test period is characterized by a bear market, with average daily returns of less than 0.20%.

The winning rate of the strategy was never below 50%, the best results of ensembles 5 and 6 for Ethereum reached 60.71% and 63.33% respectively, but the average daily profit is not impressive. In general, this strategy can beat the market significantly. Furthermore, this trading strategy is subject to high tail risk,

with CVaRs ranging between 3.88% and 13.40%, and maximum drawdowns between 11.15% and 48.06%. In particular, the results show that the best trading strategy is ensemble 5 used for Ethereum and Litecoin with an annual Sharpe ratio of 80.17% and 91.35% and an annual return of 0.5%, 9.62% and 5.73, respectively- each. each other. This price looks low compared to the daily minimum and maximum returns of these cryptocurrencies during the subsample test. However, the positives of the ML method can support the belief that the trading strategy is still valid in the cryptocurrency market, that is, when the price is low and the probability of extreme negative events is high. A positive return after trading costs can indicate that this strategy can hold up even in favourable market conditions.

Best practices in ML applications include data partitioning, parameterization, attribute space, etc. It is clear that many decisions must be made about In this study, the main objective is not to extensively test alternative forecasting and trading strategies; therefore, there is no guarantee that we use the best methods available. Instead, our goal is simpler, as we try to find out whether ML can lead to a profitable strategy in the cryptocurrency market in general, or whether this profitability is present when market conditions change and more realistic market characteristics are taken into account. High frequency data, for example using the actual transaction price of a particular online exchange; a broader set of inputs that include more nuanced attributes such as technical analysis indicators; consider bitcoin futures, where short positions are easy to make and transaction costs are lower - all of which can lead to better results.

4. REFFERNCES

[1] Garcia, D. & Schweitzer, F. (2015) Social signals and algorithmic trading of Bitcoin. *Royal Society Open Science* 2(9), 150288. Retrieved from https://dx.doi.org/10.1098/rsos.150288 10.1098/rsos.150288

[2] Kathyayini, R. S., Jyothi, D. G. & Crypt". Currency Price Prediction using Machine Learning". *International Journal of Advanced Research in Computer and Communication Engineering*

[3] Alessandretti L, ElBahrawy A, Aiello LM, Baronchelli A (2019) Anticipating cryptocurrency prices using machine learning. Complexity 2018:8983590

[4] Balcilar M, Bouri E, Gupta R, Roubaud D (2017) Can volume predict Bitcoin returns and volatility? A quantiles-based approach. Econ Model 64:74–81

[5] Cheah ET, Fry J (2015) Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of Bitcoin. Econ Lett 130:32–36

[6] Dorfleitner G, Lung C (2018) Cryptocurrencies from the perspective of euro investors: a re-examination of diversification benefits and a new day-of-the-week effect. J Asset Manag 19(7):472–494

[7] Foley S, Karlsen JR, Putniņš TJ (2019) Sex, drugs, and bitcoin: how much illegal activity is financed through cryptocurrencies? Rev Financ Stud 32(5):1798–1853

[8] Hyun S, Lee J, Kim JM, Jun C (2019) What coins lead in the cryptocurrency market: using Copula and neural networks models. J Risk Financ Manag 12(3):132.

[9] Jiang Z, Liang J (2017) Cryptocurrency portfolio management with deep reinforcement learning. In: 2017 intelligent systems conference (intelliSys). IEEE, New York, pp 905–913

[10] Mallqui DC, Fernandes RA (2019) Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques. Appl Soft Comput 75:596–606

[11] Corbet S, Meegan A, Larkin C, Lucey B, Yarovaya L (2018b) Exploring the dynamic relationships between cryptocurrencies and other financial assets. Econ Lett 165:28–34

[12] de Souza MJS, Almudhaf FW, Henrique BM, Negredo ABS, Ramos DGF, Sobreiro VA, Kimura H (2019) Can artificial intelligence enhance the Bitcoin bonanza. J Finance Data Sci 5(2):83–98

[13] Fang F, Ventrea C, Basios M, Kong H, Kanthan L, Martinez-Rego D, Wub F, Li L (2020) Cryptocurrency trading: a comprehensive survey.

[14] Hyun S, Lee J, Kim JM, Jun C (2019) What coins lead in the cryptocurrency market: using Copula and neural networks models. J Risk Financ Manag 12(3):132.

[15] Erdas, Mehmet Levent, and Abdullah Emre Caglar. 2018. Analysis of the relationships between Bitcoin and exchange rate, commodities and global indexes by asymmetric causality test. *Eastern Journal of European Studies* 9: 27–45

[16] Guarino, Alfonso, Luca Grilli, Domenico Santoro, Francesco Messina, and Rocco Zaccagnino. 2022. To learn or not to learn? Evaluating autonomous, adaptive, automated traders in cryptocurrencies financial bubbles. *Neural Comput & Applic* 34: 20715–56

[17] Kim, Alisa, Y. Yang, Stefan Lessmann, Tiejun Ma, M.-C. Sung, and Johnnie E. V. Johnson. 2020a. Can deep learning predict risky retail investors? A case

study in financial risk behavior forecasting. *European Journal of Operational Research* 283: 217–34.

[18] McNally, Sean, Jason Roche, and Simon Caton. 2018. Predicting the Price of Bitcoin Using Machine Learning. Paper presented at 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Cambridge, UK, March 21–23; pp. 339–43

[19] Phaladisailoed, Thearasak, and Thanisa Numnonda. 2018. Machine learning models comparison for bitcoin price prediction. Paper presented at 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), Bali, Indonesia, July 24–26; pp. 506–11

[20] Shin, MyungJae, David Mohaisen, and Joongheon Kim. 2021. Bitcoin Price Forecasting via Ensemble-based LSTM Deep Learning Networks. Paper presented at 2021 International Conference on Information Networking (ICOIN), Jeju Island, Republic of Korea, January 13–16; pp. 603–8