CRYPTOCURRENCY PRICE PREDICTION USING DEEP LEARNING

Major project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

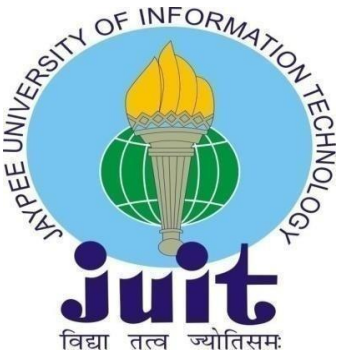
By

JYOTI SARPAL(181314)

TANISHA RANTA(181342)

UNDER THE SUPERVISION OF

Dr. MONIKA BHARTI



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh,
INDIA

DECLARATION

I hereby declare that; this project has been done by me under the supervision of **(Dr Monika Bharti, Assistant Professor (CSE/IT)),** Jaypee University of Information Technology.

I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:
(Dr Monika Bharti)
Assistant Professor
(CSE/IT)
Department of Computer Science & Engineering and Information Technology Jaypee University of Information Technology

(Jyoti Sarpal) **(Tanisha Ranta)**
(181314) (181342)

Computer Science & Engineering Department
Jaypee University of InformationTechnology

CERTIFICATE

This is to certify that the work which is being presented in the project report titled **"CRYPTOCURRENCY PRICE PREDICTION USING DEEP LEARNING** " in partial fulfillment of the requirements for the award of the degree of B Tech in Computer Science and Engineering submitted to the Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat ,Solan is an authentic record of work carried out by "Jyoti Sarpal (181314)**",** "Tanisha Ranta (181342)" during the period from January 2022 to May 2022 under the supervision of **Dr Monika Bharti**, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat ,Solan.

Jyoti Sarpal (181314)

Tanisha Ranta (181342)

The above statement made is correct to the best of my knowledge.

(**Dr Monika Bharti**)
Assistant Professor
(CS/IT)
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat,Solan

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for his divine blessing makes it possible for us to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor Dr Monika Bharti Assistant **Professor (CSE/IT**) Department of Computer Science and Engineering, Jaypee University of Information Technology, Wakhnaghat,Solan. Deep Knowledge & keen interest of my supervisor in the field of "**CRYPTOCURRENCY PRICE PREDICTION USING DEEP LEARNING**" to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr Monika Bharti, Department of Computer Science and Engineering, for her kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge ,with due respect, the constant support and patients of my parents.

TABLE OF CONTENTS

Chapters **Page No.**

1 INTRODUCTION

1.1 Introduction 1

1.2 Problem Statement 5

1.3 Objectives 6

1.4 Methodology 6

1.5 Organization 7

2 LITERATURE SURVEY

2.1 Literature Survey 9

2.2 Requirements 13

2.2.1 Tools and Technologies 13

2.2.2 Description of The Proposed Approach 19

3 SYSTEM DEVELOPMENT

3.1 Benchmark Dataset 20

3.2 Dataset Features and Definitions 21

3.3 The Proposed Approach 22

3.4 Building Neural Network 25

3.4.1 Recurrent Neural Network 26

4 PERFORMANCE ANALYSIS

4.1 Long Short Term Memory 27

4.2 Gated Recurrent Unit 28

4.3 AutoRegressive Integrated Moving Average 30

4.4 Screenshots of the various Stages

4.4.1 Discovering/Analyzing the data 30

4.4.2 Validation and Preprocessing 31

4.4.3 Exploration and Visual Analysis 32

4.4.4 Data Scaling 33

4.4.5 Train Test Split 34

4.4.6 Performance Metrics 35

5 CONCLUSIONS

5.1 Results and Discussions 37

5.2 Model Evaluation 39

5.2 Future Sco**pe** 44

REFERENCES 45

ABSTRACT

Cryptocurrencies are a sort of digital currency in which all transactions are carried out through the internet. It is a soft currency that does not exist in hard cash form. We emphasize the difference between a decentralized currency and a centralized currency in that all virtual currency users can acquire services without the intervention of a third party. Using these cryptocurrencies, however, has an influence on international relations and trade because of their severe price volatility. Furthermore, the rapid variations in cryptocurrency prices indicate that a reliable method for estimating this price is urgently required.

Price control by a number of organizations has had a significant impact on the level of one main or central control over them, affecting relationships with other businesses and international trade. Furthermore, the ever-changing oscillations suggest a more accurate means of projecting this price is desperately needed. Thus, using deep learning techniques such as the recurrent neural network (RNN) and the long short-term memory (LSTM), gated recurrent unit (GRU), which are effective learning models for training data, we must design a method for the accurate prediction of by considering various factors such as market cap, maximum supply and, volume, circulating supply.

The proposed method is written in Python and tested on benchmark datasets. The results show that the proposed method can be used to make reliable predictions. Thus, the neural network, which has been used by academics in numerous fields over the past ten years as one of the intelligent data mining tools.

Stock market data is critical in today's economy. Linear (AR, MA, ARIMA, ARMA) and non-linear models are the two types of forecasting methodologies (ARCH, GARCH, Neural Network). To anticipate a company's stock price based on past prices, we employed the Autoregressive Integrated Moving Average (ARIMA), Recurrent Neural Network, Long Short-Term Memory (LSTM), and gated recurrent unit deep learning architectures (GRU).

CHAPTER 1

1.1 INTRODUCTION

In financial systems, cryptocurrency is a virtual or digital currency. It is protected by encryption, which prevents counterfeiting and double-spending. It is also distinguishable from traditional currencies because it is not issued by a central authority or central banks, and it is a decentralized virtual currency that can be converted via cryptographic techniques. The other aspect is that it is powered by block chain technology, which is incredibly complicated and aims to store data in such a way that it is difficult or impossible to alter, hack, or cheat the system. Bitcoin has begun to carve out a niche for itself, which could either help cryptocurrencies gain widespread acceptance or spell their doom. Cryptocurrencies are still in the experimental stage and it is difficult to predict whether they will ever be widely used in global markets or not. The most prominent cryptocurrency, Bitcoin, was established in 2009 and for more than two years was the sole Block chain-based cryptocurrency.

Recurrent neural networks (RNNs), which are the state-of-the-art approach for sequential data, are used in both Apple's Siri and Google's voice search. It's an algorithm that uses its internal memory to recall its input, making it perfect for machine learning issues that require sequential data. It is a deep learning algorithm that produces good results. We'll look at how to anticipate Bitcoin values using data from the last six years in this article.

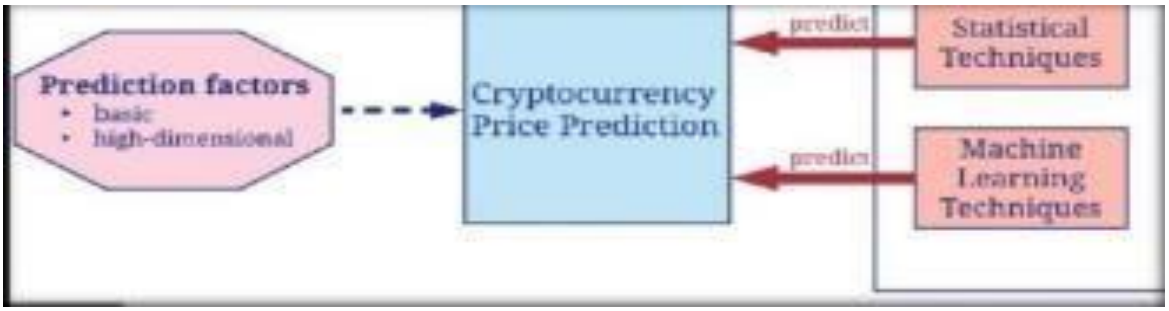


Fig 1: Machine Learning for Cryptocurrency Prediction

The rapid rise in market capitalization and price of bitcoin produced a swarm of other cryptocurrencies, the majority of which differ from bitcoin in only a few areas (block time, currency supply, and issuance scheme). With more than 5.7 thousand cryptocurrencies, 23 thousand online exchanges, and a market capitalization of more than 270 billion USD, the cryptocurrency business had evolved to become one of the world's largest unregulated marketplaces by July 2021.

Bitcoin and other cryptocurrencies soon gained a reputation as pure speculative ventures, despite its origins as a peer-to-peer electronic payment system. Their prices are unpredictable because they are mostly influenced by behavioral factors and are unrelated to the primary types of financial assets; nonetheless, their informational efficiency is high.
As a result, many hedge funds and asset managers began to include cryptocurrencies in their portfolios, while researchers concentrated on cryptocurrency trading, namely machine learning (ML) algorithms. Bitcoin's early success as a peer-to-peer virtual currency was due to its cryptography-based technology, which eliminates the need for a trusted third party and solves the problem of double-spending.

Bitcoin is based on block chain technology, which functions as a public (permission-less) digital ledger that records user transactions. Because there is no central authority, this ledger can be replicated by network participants (nodes), who collaborate to maintain it using dedicated software. The first Bitcoin logo has undergone two changes. The revisions were clearly evident in both cases. The graphic elements were adjusted by the designers to match the theme.

Bitcoin is a digital money that was first introduced in January 2009. It is the most valuable cryptocurrency in the world, and it is traded on more than 40 exchanges worldwide, accepting more than 30 different currencies. Bitcoin, as a currency, presents a novel potential for price forecasting due to its extreme volatility, which is significantly higher than that of traditional currencies.

The bitcoin system consists of a collection of decentralized nodes that run the bitcoin code and keep track of its block chain. A block chain can be thought of as a collection of blocks in metaphorical terms. There are a number of transactions in each block. Because all computers running the block chain have the same list of blocks and transactions and can observe these fresh blocks being filled with new bitcoin transactions in

Because all the computers running the block chain have the same list of blocks and transactions, and can transparently see these new blocks being filled with new bitcoin transactions, no one can cheat the system.

It is mainly a digital ledger of transactions that is distributed across the entire network of computer systems on the block chain. The block chain consists of two fundamental components; the first one is a transaction, and the second is a block. The transaction represents the action triggered by the participant, and the block is a data collection that records the transaction and additional details such as the correct sequence and creation timestamp. Block chains have a signaling system of multi-domain, block chain-based, cooperative DDoS defense systems in which each autonomous system (AS) joins the defensive line.

Effects of networks on competition in the nascent cryptocurrency market over a period of time regarding exchange rates among cryptocurrencies depends on two aspects: (1) competition among different currencies and (2) competition among exchanges. There are hundreds of cryptocurrencies, but Bitcoin is the most popular one as it is a stubborn competitor and did not emerge from the cryptocurrency competition track. As a result, it has become the dominant cryptocurrency. The authors describe the competition between cryptocurrency as “healthy competition” and suggest that new technology and security innovation.

The aim of this research is to examine whether the price of Bitcoin can be predicted similar to other stock market tickers. This will have a basis on whether we can further use it as a

medium of payment. Block chain keeps track of all Bitcoin transactions occurring anywhere in the world. It is a cryptographic implementation that provides the highest security.The popularity of cryptocurrencies soared in 2017 as their market value grew exponentially for several months in a row. In January 2018, prices reached a high of around $800 billion.

Although machine learning has been successful in predicting stock market prices using a variety of time series models, it has been limited in its use to predicting cryptocurrency prices. The reason for this is obvious: cryptocurrency values are influenced by a variety of factors such as technological advancements, internal competitiveness, market pressure to produce, economic troubles, security concerns, political factors, and so on. Because of their tremendous volatility, they have a huge profit potential if smart invention tactics are used. Cryptocurrencies are, unfortunately, less predictions like stock market predictions.

1.2 PROBLEM STATEMENT

Are machine learning models capable of forecasting the bitcoin market?
Unlike traditional paper currency which can be printed as per market needs, Cryptocurrency has a limited supply. Should you invest or not?

We investigate this research question by contrasting well-known machine learning models trained on crypto currency data. Our research makes two major contributions. To begin, we add to the research by rigorously comparing the predictive abilities of several prediction models (e.g., recurrent neural networks, gated recurrent unit (GRU), Long Short-Term Memory LSTM), ARIMA(Autoregressive Integrated Moving Average**)** ,feature sets (e.g., technical, block chain-based), and prediction horizons. As a result, our research provides a comprehensive baseline for the predictive accuracy of short-term crypto market prediction models. Recurrent neural networks appear to be well-suited for this prediction challenge, according to the overall picture emerging from the analysis.

We hope to use Machine Learning Algorithms which also are widely utilized by many organizations. This report will walk through a simple implementation of analyzing and forecasting the prices by using various Machine Learning Algorithms.

1.3 OBJECTIVES

The main goal of this study is to use technical trade indicators and machine learning to build and integrate price prediction for various cryptocurrencies.

- Is Bitcoin a publicly traded commodity with a price prediction possible?
- Can cryptocurrencies be used to regulate the price like any other traditional money, such as the US dollar?
- Is cryptocurrency, such as Bitcoins, capable of replacing traditional currencies as a primary mode of transaction?
- Using machine learning, create an automated application for predicting a price increase in cryptocurrencies for various time series.

1.4 METHODOLOGY

Due to price volatility and dynamism, cryptocurrency prices are difficult to forecast. Hundreds of cryptocurrencies are used by clients all around the world. We'll look at three of the more popular ones in this paper. As a result, the study intends to do the following by employing deep learning algorithms, which may uncover hidden patterns in data, integrate them, and generate considerably more accurate predictions:

A full examination of the many existing systems for predicting BTC cryptocurrency prices is presented. LSTM, ARIMA, and GRU AI algorithms are used to reliably anticipate cryptocurrency prices. For prediction, various AI algorithms are used which enable an auto machine learning method.Using evaluation matrices such as, evaluating the proposed hybrid models such as RMSE.

1.5 ORGANIZATION

We used APIs to fetch this Bitcoin Cryptocurrency dataset. The obtained dataset was then averaged into one dataset for consistency and in order to fill in the gaps created by missing data in the dataset. Building Neural Network Model Machine Learning is the most suitable technique which can be used here to predict cryptocurrency prices prediction.

The model to be built had to achieve several goals in order to produce a near to accurate prediction.

This included selecting the framework which could produce a good prediction accuracy, take in consideration of other parameters in its prediction algorithm and be trainable. Below is the organization of our information.

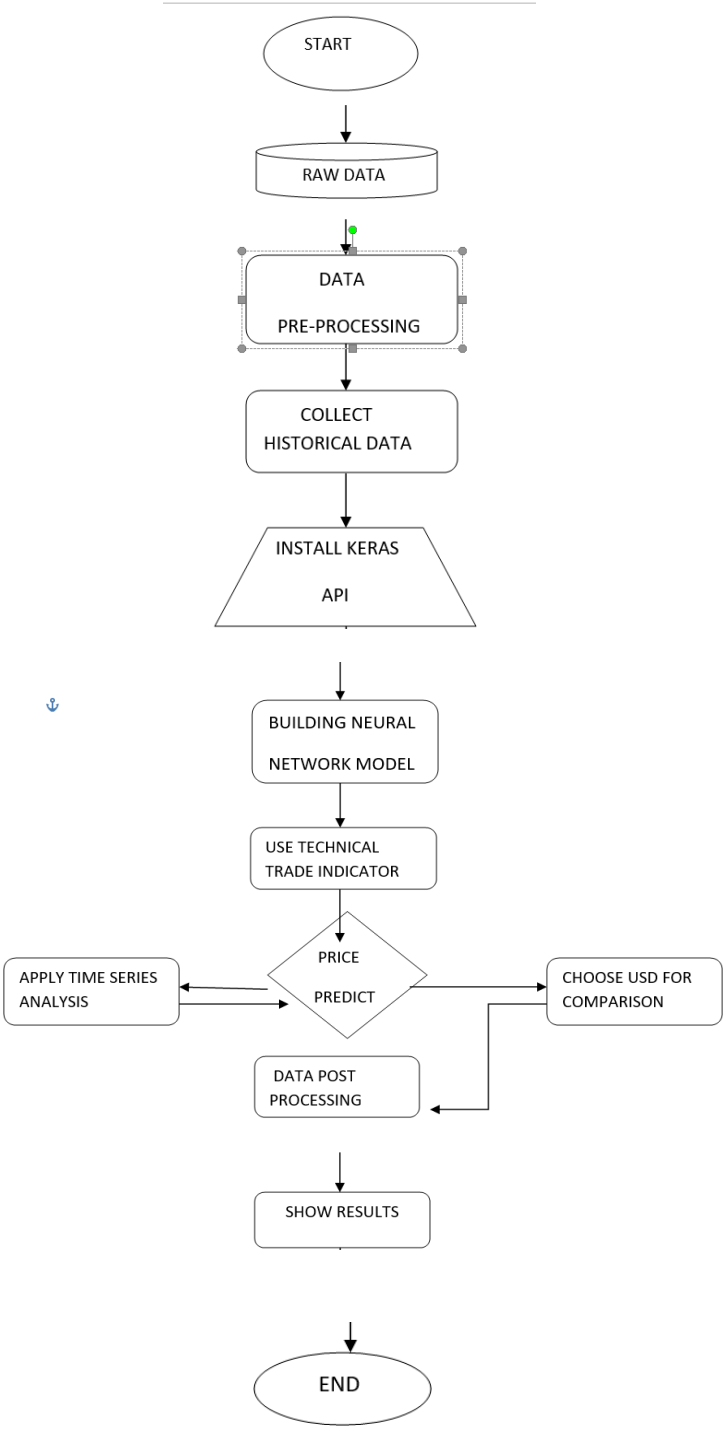


Fig 3 : Organization of Model

In Fig 3 we have shown how our respective model is organized by the various steps that are included in this.

CHAPTER 2

2.1 LITERATURE SURVEY

1) A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms

Mohammad J. Hamayel and Amani Yousef Owda
Department of Natural, Engineering and Technology Sciences, Arab American University, Ramallah P600, Palestine; m.abdjoudeh@student.aaup.edu

Hamayel and Owda Proposed a prediction model for predicting the prices of three types of cryptocurrency BTC ETH LTC

Performance measures were conducted to test the accuracy of different models . Then, they compared the actual and predicted prices. The results show that GRU outperformed the other algorithms with a MAPE of 0.2454%, 0.8267%, and 0.2116% for BTC, ETH, and LTC, respectively. The RMSE for the GRU model was found to be 174.129, 26.59, and 0.825 for BTC, ETH, and LTC, respectively. Based on these outcomes, the GRU model for the targeted cryptocurrencies can be considered efficient and reliable. This model is considered the best model. However, bi-LSTM represents less accuracy than GRU and LSTM with substantial differences between the actual and the predicted prices for both BTC and ETH. The experimental results show that:

- The AI algorithm is reliable and acceptable for cryptocurrency prediction.
- GRU can predict cryptocurrency prices better than LSTM and bi-LSTM but overall all algorithms represent excellent predictive results.

2) Deep Learning-Based Cryptocurrency Price Prediction Scheme with Interdependent Relations

SUDEEP TANWAR 1 , (Senior Member, IEEE), NISARG P. PATEL 1 , SMIT N. PATEL 2 , JIL R. PATEL 3 , GULSHAN SHARMA 4 , AND INNOCENT E. DAVIDSON 4 , (Senior Member, IEEE)
Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, Gujarat 382481, India 2Department of Information and Technology, Government Engineering College, Gandhinagar, Gujarat 382028, India 3Department of Information and Technology, Hasmukh Goswami College of Engineering, Vehlal, Ahmedabad, Gujarat 382330, India 4Department of Electrical Power Engineering, Durban University of Technology, Steve Biko Campus, Durban 4001, South Africa Corresponding authors: Gulshan Sharma (gulshanS1@dut.ac.za) and Sudeep Tanwar; sudeep.tanwar@nirmauni.a

Tanwar et al. proposed a prediction model on Forecasting cryptocurrency prices.
They proposed a hybrid model of GRU and LSTM with an interdependent relationship to the parent currency in this study. The suggested approach uses the direction of Bitcoin with four window widths to estimate the price of Litecoin and Zcash. The suggested model's MSE losses for 1-day and 3-days are 0.02038 and 0.02103, respectively, for Litecoin, and 0.00461 and 0.00483 for Zcash. The presented model accurately predicts bitcoin prices for 7 and 30 days.

For Zcash, however, it followed a stochastic pattern. Compared to the bigger window size for Zcash, the proposed model performs well for the smaller window size. For a greater window size of -days and 30-days for Zcash, the proposed model demonstrates the stochastic character.
They will use the proposed methodology to work on cryptocurrencies with multiple interdependencies in the future. We will also add emotive factors to the suggested algorithm, such as Twitter and Facebook posts and messages, to improve the accuracy of the forecast findings. Traditional commodities like gold and oil prices can be used to improve the

prediction outcome.

The crypto market, on the other hand, is less stable than traditional commodities markets. Many technological, sentimental, and legal elements can influence it, making it very volatile, uncertain, and unexpected. Many studies have been conducted on various cryptocurrencies in order to estimate correct prices, but the bulk of these methods are not applicable in real time. In this work, they present a solution based on the previous debate To predict the price of lit coin and Zcash with inter-dependency of the parent currency, a deep-learning-based hybrid model (including Gated Recurrent Units (GRU) and Long Short Term Memory (LSTM)) was used. The suggested model is well trained and tested using standard data sets and can be employed in real-time applications. In comparison to existing models, the suggested model estimates prices with a high degree of accuracy.

3) Machine Learning for Bitcoin Pricing — A Structured Literature Review

Patrick Jaquart1, David Dann1, and Carl Martin1

1 Karlsruhe Institute of Technology, Institute of Information Systems and Marketing (IISM), Karlsruhe, Germany

{patrick.jaquart, david.dann}@kit.edu, carl.martin@student.kit.edu

Jaquart et al. proposed a study in which they use machine learning to assess the existing corpus of literature on empirical bitcoin pricing and organized it into four main concepts.

They demonstrate that research on this subject is quite different, and that the findings of multiple studies can only be compared to a limited extent. They also develop standards for future field papers to ensure a high level of transparency and reproducibility.

2.2 REQUIREMENTS

In this we are going to list all the requirements that are needed in the model. The various Tools, Technologies and Libraries which are used.

2.2.1 TOOLS AND TECHNOLOGIES AND LIBRARIES

Tensor board:

TensorBoard is a visualization toolbox for TensorFlow that lets you watch metrics like loss and accuracy, display the model graph, observe histograms of weights, biases, and other tensors as they change over time, and much more. It is a component of the TensorFlow ecosystem and is free source.

The dashboards for Scalars, Graphs, Histograms, Distributions, and HParams are now available. Over time, more TensorBoard dashboards will be introduced.

Anyone with a link can see any data published to TensorBoard.dev. It should not be used to store sensitive information.

Tensor Board provides the necessary visualization tools which are needed for machine learning model building. Also helps us in:

a) Keep track and visualize metrics like loss and accuracy

b) Model the graph and its various layers.

c) Develop histograms, boxplots, line graphs, bar charts, and other tensors which change over (i.e., include new and additional features) from time to time.

Py Torch:

Py Torch is an open source machine learning framework based on Facebook's AI Research division's Torch library for applications such as computer vision and natural language processing (FAIR). It's free software released under the Modified BSD license. Although PyTorch has a C++ interface, the Python interface is more developed and is the main focus of development. Deep learning software developed on PyTorch includes Tesla Autopilot, Uber's Pyro,

Hugging Face's Transformers, PyTorch Lightning, and Catalyst, to name a few.

PyTorch offers two high-level features:

- Tensor computation (like NumPy) with powerful graphics processing unit acceleration (GPU)
- Deep neural networks with an autonomous differentiation method based on tape.

Modules

Module Autograd:

Automated differentiation is a PyTorch approach. The outputs of operations are recorded and then played backwards to compute the gradients using a recorder. This method is very beneficial for building neural networks since it allows for parameter differentiation during the forward pass, which saves time on a single epoch. The torch module could be improved.

The optim module implements different optimization strategies for neural network construction. There is no need to construct the most widely used methods from scratch because they are already supported.

The PyTorch auto grad module makes it simple to create computational graphs and compute gradients, but raw autograd is too low-level for defining sophisticated neural networks. The nn module can help with this.

Data Reader:

To obtain data with a DataReader, first construct an instance of the Command object, then call Command to build a DataReader. To obtain rows from a data source, use the ExecuteReader method. The DataReader offers an unbuffered stream of data that allows procedural logic to process data from a data source consecutively and effectively. Because the data is not stored in memory, the DataReader is an excellent choice for obtaining massive

volumes of data. To get a row from the query results, use the read method. Each column in the returned row may be accessed by passing the column's name or ordinal number to the DataReader.

Numpy:

For scientific computing, NumPy is the most important Python package. It's a Python library with a multidimensional array object, derived objects (like masked arrays and matrices), and a variety of routines for performing fast array operations like mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and more.

The NumPy library includes multidimensional array and matrix data structures (more on this in subsequent sections). It provides methods for working efficiently with ndarray, a homogeneous n-dimensional array object. NumPy allows you to perform a variety of array-based mathematical operations. It includes a large library of high-level mathematical functions that interact with these arrays and matrices, as well as advanced data structures that ensure quick array and matrix calculations.

NumPy (Numerical Python) is a widely used open source Python library in research and engineering. It's the de facto standard for working with numerical data in Python, and it's at the centre of the scientific Python and PyData ecosystems. NumPy users range from inexperienced other data science and scientific Python tools utilizing the NumPy API extensively.

Python lists are more compact and slower than NumPy arrays. An array consumes less memory and is simpler to manipulate. NumPy saves data in a much smaller amount of memory and allows you to choose data types. This makes it possible to optimize the code even further.

Pandas:

Pandas is a data manipulation and analysis software package for the Python programming language. It includes data structures and methods for manipulating numerical tables and time series, in particular. It's open-source software with a three-clause BSD license.

The word "panel data" is an econometrics term for data sets that comprise observations for the same persons over several time periods.

Features of the library:

- DataFrame object with inbuilt indexing for data manipulation.
- Data reading and writing tools for in-memory data structures and various file formats.
- Data alignment and integrated missing data handling
- Data sets may be reshaped and pivoted.
- Slicing of big data sets based on labels, clever indexing, and subsetting
- Insertion and deletion of data structure columns.
- Split-apply-combine procedures on data sets may be performed by grouping by engine.
- Merging and joining data sets.
- To work with high-dimensional data in a lower-dimensional data structure, use hierarchical axis indexing.
- Date range generation and frequency conversions, moving window statistics, moving window linear regressions, date shifting and lagging are all examples of time series capability. Data filtration is available.
- Critical code paths are written in Cython or C, and the library is heavily optimized for efficiency.

Dataframes:

Pandas is mostly used in Dataframes for data analysis and related data manipulation. Pandas can read data from CSV, JSON, Parquet, SQL database tables or queries, and Microsoft Excel files. Pandas is capable of merging, reshaping, choosing, data cleansing, and data wrangling, among other data manipulation activities. With the advent of pandas, many similar characteristics of dealing with Dataframes that were created in the R programming language were included into Python. The pandas library is based on the NumPy library, which is designed to operate with arrays rather than Dataframes.s.

Seaborn:

Seaborn is a Python tool that allows you to create statistical visualisations. It is built on top of matplotlib and works closely with pandas data structures.
Seaborn can help you explore and understand your data. Its graphing capabilities work with dataframes and arrays holding entire datasets, using internal semantic mapping and statistical aggregation to produce informative graphs. Because of its declarative, dataset-oriented API, you can focus on the meaning of your charts rather than the mechanics of generating them. We only need to import the Seaborn library for our short example. The acronym sns is commonly used when importing it.

Seaborn draws its plots with matplotlib behind the scenes. It's best to utilise a Jupyter/IPython interface in matplotlib mode for interactive work, or else you'll have to execute matplotlib.pyplot.show() to view the plot.

To obtain rapid access to an example dataset, much of the code in the documentation will utilise the load dataset() method. These datasets are nothing special: they're just pandas dataframes that we could have imported using pandas.read csv() or manually generated. Although most of the examples in the manual use pandas dataframes, seaborn is fairly versatile when it comes to data structures.

Using a single call to the seaborn function relplot, this plot depicts the relationship between five variables in the tips dataset (). Notice how we merely gave the variables' names and their functions in the graphic. It wasn't essential to give plot element properties in terms of colour values or marker codes, unlike when using matplotlib directly. Behind the scenes, seaborn managed the conversion of dataframe values into matplotlib-friendly arguments. This declarative method allows you to concentrate on the questions you want to answer rather than the specifics of matplotlib control.

In seaborn, there are several specific plot styles for showing categorical data. They may be found using catplot (). Different levels of granularity are available in these charts. At the most basic level, you could want to make a "swarm" plot, which is a scatter plot in which the points along the category axis are adjusted such that they don't overlap:

Pyplot :

Pyplot is a Python matplotlib API that essentially turns matplotlib into a viable open source alternative to MATLAB. Matplotlib is a data visualization package that generates plots, graphs, and charts.

Pyplot is stateful, which means it remembers the state of an object when you first plot it. This is required for usage inside the same loop or session state until plt.close() is called. When establishing many plots in a row, state is also significant.

The pyplot API is a hierarchy of Python code objects that comprises various functions, the most important of which being matplotlib.pyplot.

Scripting layer - this layer is used to construct a figure that comprises one or more plots with axes (i.e., x axis ,y axis, and possibly z axis)
Artist Layer - used to edit plot components such as labels, lines, and so on.
Backend Layer: This layer is used to format the plot for presentation in a specific target application, such as a Jupyter Notebook.

There are times when you have data in a format that allows you to use strings to access certain variables. For instance, numpy.recarray and pandas. DataFrame.

The data keyword parameter in Matplotlib lets you supply such an object. If these variables are available, you may create graphs using the strings that correspond to them.

Tensor flow:

An open-source platform for problem-solving. Includes a number of comprehensive, flexible tools and multi-purpose libraries, as well as community resources, to assist academics in developing Machine Learning models and quickly building and deploying Machine Learning-based applications.

TensorFlow is a free and open-source machine learning and artificial intelligence software library. It can be used for a wide range of tasks, however it is primarily focused on deep neural network training and inference.
TensorFlow was created by the Google Brain team for internal Google research and production. The Apache License 2.0 was used to release the initial version in 2015.

TensorFlow is compatible with a wide range of programming languages, including Python, Javascript, C++, and Java.
This adaptability lends itself to a wide range of applications in a variety of industries.

2.2.2 DESCRIPTION OF PROPOSED APPROACH

TABLE 1: DESCRIPTION OF PROPOSED APPROACH

Parameter Name	Description
Processor	Intel Core i7 processor, up to 3.8 GHz
Operating System	WINDOWS
RAM	RAM 16 GB
Graphics Processor:	NVIDIA GeForce 930M

The above table shows the description of the proposed approach that we have used.

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 BENCHMARK DATA SET

The suggested method was tested using one of the most well-known and oldest cryptocurrencies, Bitcoin (BTC) The BTC dataset included exchanges from January 2016 to December 2021, with OHLC (Open, High, Low, Close) updates every minute, the volume of BTC and the specified currency, and weighted Bitcoin prices. The dataset was publicly accessible over the Internet.

Table 2: Bitcoin Datasets

Parameter	**Value/Description**
Dataset Details	USD (Large in number named as L1)
Memory usage	919.8 MB
Range Index	1259 entries, 0 to 1258
Total Data Columns	14
Date	int64
Open	float64
Close	float64
High	float64
Low	float64
Volume	float64
adjClose	float64
adjHigh	float64
adjOpen	float64
adjVolume	float64
divCash	float64
splitFactor	float64

Given above is the Bitcoin dataset which includes the description of the parameters of the dataset.

symbol	date	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjOpen	adjVolume	divCash	splitFactor
AAPL	2016-12-05 00:00:00+00:00	109.11	110.03	108.25	110.00	34324540	25.672732	25.889200	25.470380	25.882142	137298160	0.0	1.0
AAPL	2016-12-06 00:00:00+00:00	109.95	110.36	109.19	109.50	26195462	25.870377	25.966847	25.691555	25.764496	104781848	0.0	1.0
AAPL	2016-12-07 00:00:00+00:00	111.03	111.19	109.16	109.26	29998719	26.124493	26.162139	25.684496	25.708025	119994876	0.0	1.0
AAPL	2016-12-08 00:00:00+00:00	112.12	112.43	110.60	110.86	27068316	26.380961	26.453902	26.023317	26.084493	108273264	0.0	1.0

Fig 4: **BTC Dataset**

3.2 FEATURES AND THEIR DEFINITIONS

High: The highest bitcoin price in one-minute period.

Last: The bitcoin price at that one-minute interval.

Timestamp: The bitcoin server's one-minute timestamp. Bitcoin volume in that one-minute period.

Low: The lowest bitcoin price in that one-minute period.Bitcoin's open price at that one-minute interval.

Adj Close: The adjusted close is the closing price after all applicable splits have been adjusted.

Adj High: The highest price of bitcoin after adjusting for all applicable splits.

Adj Low: The adjusted low price of bitcoin is the lowest price after adjusting for all applicable splits.

Adj Open: Adjusted low is bitcoin's initial price after adjusting for applicable splits.

Split: A 2-for-1 split gives you two shares for the price of one.

3.3 THE PROPOSED APPROACH

Different phases of the proposed approach are described as follows:

Data Analysis Phase: This phase examines data and its parameters for redundancy in data values that might impact prediction outcomes.

If a dataset contains any unnecessary parameters, the data values for those parameters are eliminated. This step also analyses data for prospective data merging to increase model predictability.

Data Filtration Step: In this phase, data is filtered to eliminate any empty or redundant values.

The Train-Test Split Phase divides data into training and testing subgroups.
For example, data is split into two sections, with 65 percent training data and 35 percent test data.

Data-Scaling Phase: Data are scaled according to model requirements before being delivered to the model. This step reshapes data in this way to make it more suited for the model.

Modeling Phase: The suggested technique is written in Python. In Python, the two most powerful libraries for machine learning models are keras and tensor flow. Keras with Tensor Flow is utilized as the backend library to improve the accuracy of the model. The Keras sequential model is made up of two layers: LSTM and dense layers. These layers thoroughly examine data in order to assess all types of patterns created in the dataset in order to improve the model's precision. The data is then fed into the model for training.

Phase of Model Learning and Evaluation: Data is trained using several LSTM units. There are four gates in this circuit: a memory cell, an input gate, an output gate, and a forget gate. These gates allow information to pass through.

Then the same data is trained using ARIMA forecasting model where used various features and measures to test it then we have the GRU model which is also used to train the model,

All these models are used for training and testing of data so that we get results and we can test them for the next 30 days to compare their results.

Prediction Phase: The stored model is used to make the prediction. The model is fed input data and produces projected values as output. The output is then compared against testing data to determine accuracy and losses. The stored model is used to make the prediction. The model is fed input data and produces projected values as output. The output is then compared against testing data to determine accuracy and losses.

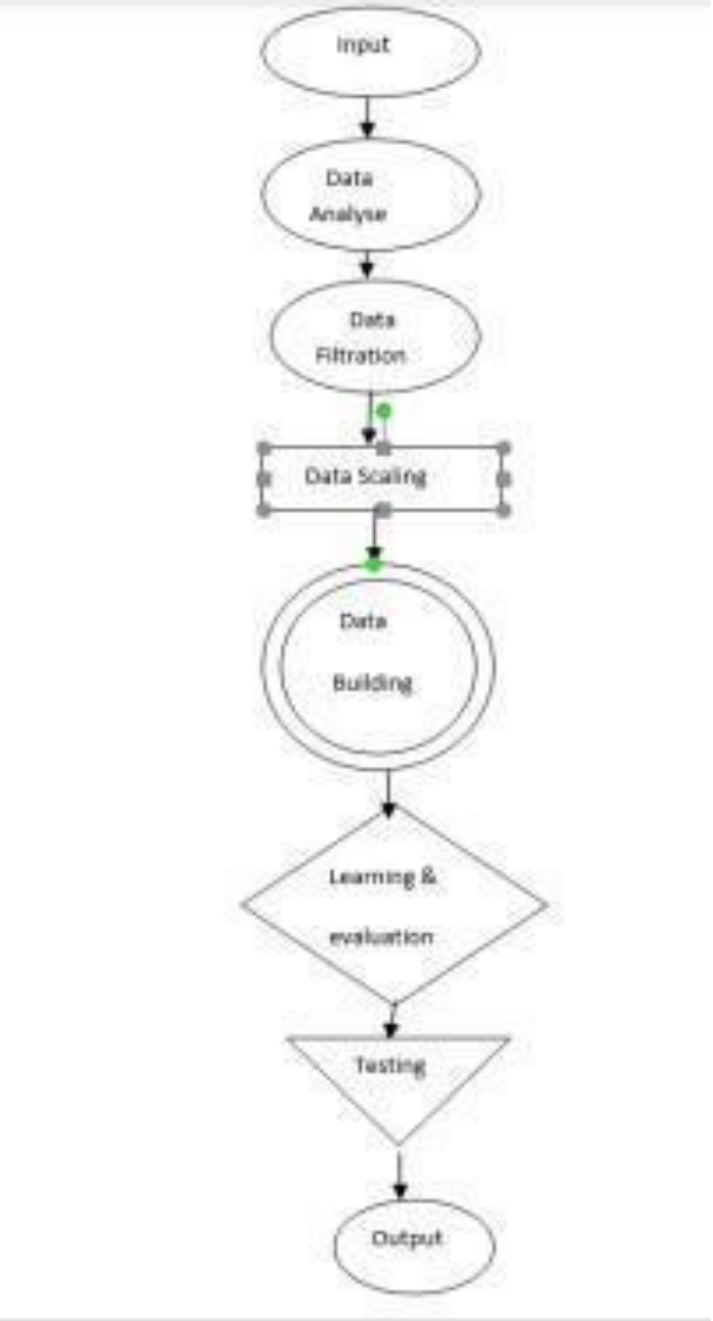


Fig 5: Proposed Approach

The above figure explains the various proposed steps used to make our model.

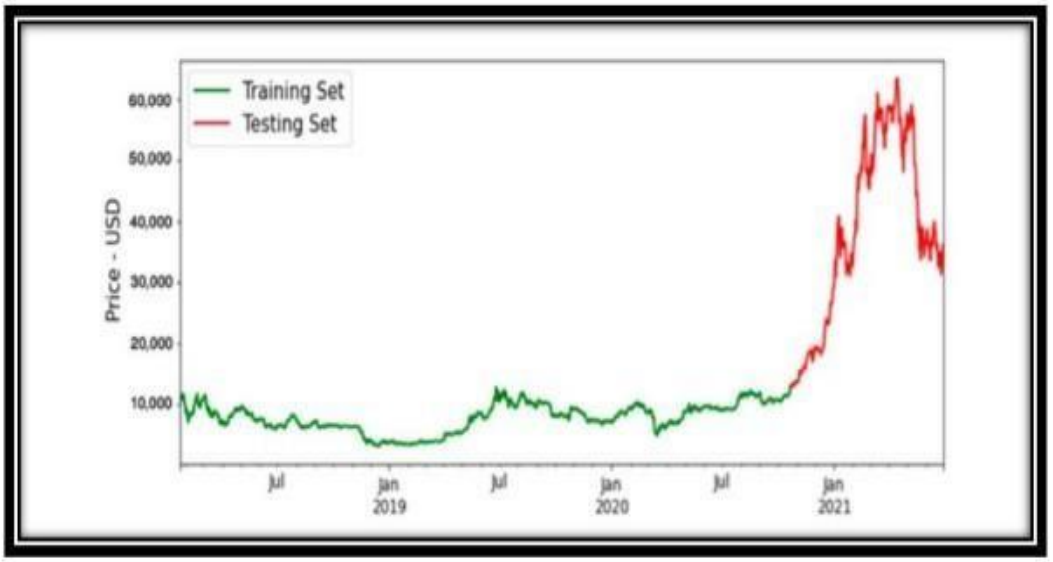


Fig 6: Training and Testing Datasets

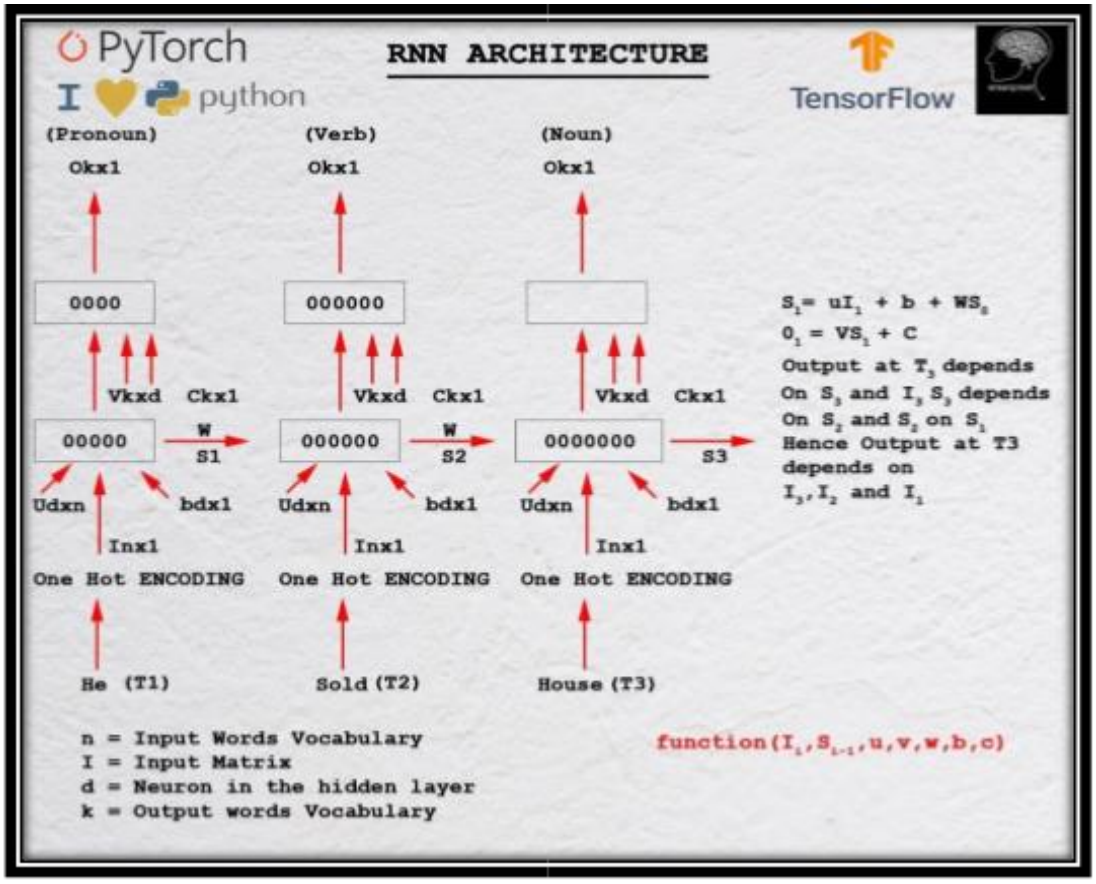
3.4 BUILDING NEURAL NETWORK

Model Machine Learning is the most appropriate approach for predicting Bitcoin cryptocurrency values in this case. In order to create a near-accurate forecast, the model had to accomplish numerous goals. This involved picking a framework that could yield excellent prediction accuracy, take other parameters into account in its prediction process, and be trainable.

Following that, it was necessary to determine which layers would be included and how many would be required, as well as the epoch rates. The training dataset was normalized and altered since it is better appropriate for various activation functions.

The square of the correlation coefficient is used in this method to determine the link between characteristic fields in the collection of data. This assists the dominant parameters in determining the values for the other fields. The bitcoin price is then established using linear and exponential forecasting. To anticipate cryptocurrency prices, the suggested method employs the LSTM model and the GRU model.

3.4.1 Recurrent neural networks (RNN):



Output is affected by both current and previous inputs. Let I1 be the initial input with a dimension of n*1, where n is the vocabulary length. S0 represents the hidden state of the first RNN cell with d neurons. Input hidden state for each cell should be one prior. Because no prior state is visible, initialise S0 with zeros or a random value for the first cell. U is another d*n-dimensional matrix, where d is the number of neurons in the first RNN cell and n is the size of the input vocabulary. W is another d*d-dimensional matrix. b is bias with a size of d*1. Another matrix V with the size k*d is used to find the output from the first cell.

Mathematically, outputs from the first RNN cell are as below:

S1= UI1+ WS0 + b

O1= VS1+c

In General,

Sn= UIn+ WSn-1 + b ; On= VSn+c

CHAPTER 4

PERFORMANCE ANALYSIS

4.1 LSTM (Long Term Memory Network):

The long short-term memory network, or LSTM, solves the problem of fading gradients that plagues recurrent neural networks. This is a type of recurrent neural network used in deep learning because it can learn very large designs. The LSTM is an RNN-like architecture with gates that control data flow between cells. The input and forget gate structures can alter information passing through the cell state, with the final output being a filtered version of the cell state based on the context of the inputs. The mathematical representation of the LSTM forward training procedure is as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (1)$$

$$i_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_i\right) \quad (2)$$

$$C_t = f_t * C_{t-1} + i_t * tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

$$O_t = \sigma(W_0 \cdot [h_{t-1}, x_t] + b_0) \quad (4)$$

$$h_t = o_t * tabh(C_t) \quad (5)$$

Below is the LSTM model being applied:

```
model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
```

```
[ ] model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 100, 50)           10400

 lstm_1 (LSTM)               (None, 100, 50)           20200

 lstm_2 (LSTM)               (None, 50)                20200

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

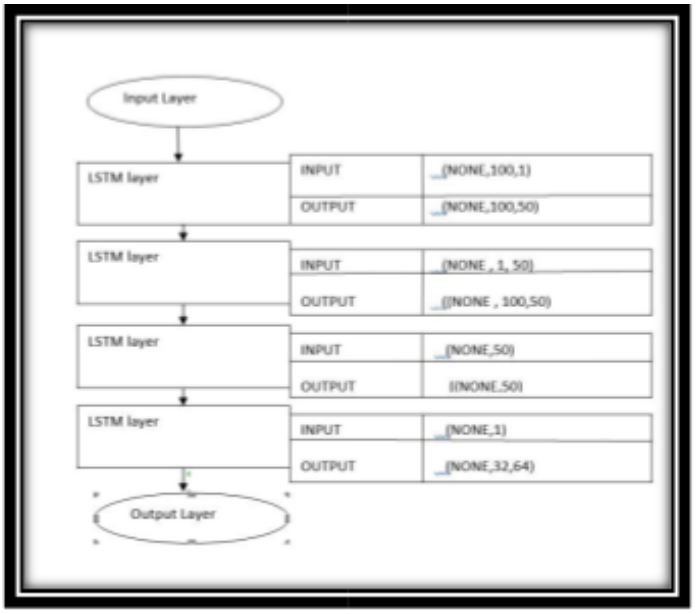


Fig 8: LSTM OUTPUT

4.2 GATED RECURRENT UNIT (GRU):

Gated recurrent neural networks (Gated RNNs) have demonstrated their effectiveness in a variety of applications requiring sequential or temporal data. The transition functions in hidden units of GRU are given as follows:

$$z_t = \sigma(W^z x_t + V^z h_{t-1} + b^z)$$

$$r_t = \sigma(W^r x_t + V^r h_{t-1} + b^r)$$

$$\overline{h}_t = tanh(W^c x_t + V^c(r_t \cdot h_{t-1}))$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \overline{h}_t$$

Below is the GRU model being applied:

```
[ ] regressorGRU = Sequential()
    # First GRU layer with Dropout regularisation
    regressorGRU.add(GRU(units=50, return_sequences=True, input_shape=(X_train.shape[1],1), activation='tanh'))
    regressorGRU.add(Dropout(0.2))
    # Second GRU layer
    regressorGRU.add(GRU(units=50, return_sequences=True, input_shape=(X_train.shape[1],1), activation='tanh'))
    regressorGRU.add(Dropout(0.2))
    # Third GRU layer
    regressorGRU.add(GRU(units=50, return_sequences=True, input_shape=(X_train.shape[1],1), activation='tanh'))
    regressorGRU.add(Dropout(0.2))
    # Fourth GRU layer
    regressorGRU.add(GRU(units=50, activation='tanh'))
    regressorGRU.add(Dropout(0.2))
    # The output layer
    regressorGRU.add(Dense(units=1))
```

```
[ ] regressorGRU.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru (GRU)                   (None, 100, 50)           7950

 dropout (Dropout)           (None, 100, 50)           0

 gru_1 (GRU)                 (None, 100, 50)           15300

 dropout_1 (Dropout)         (None, 100, 50)           0

 gru_2 (GRU)                 (None, 100, 50)           15300
```

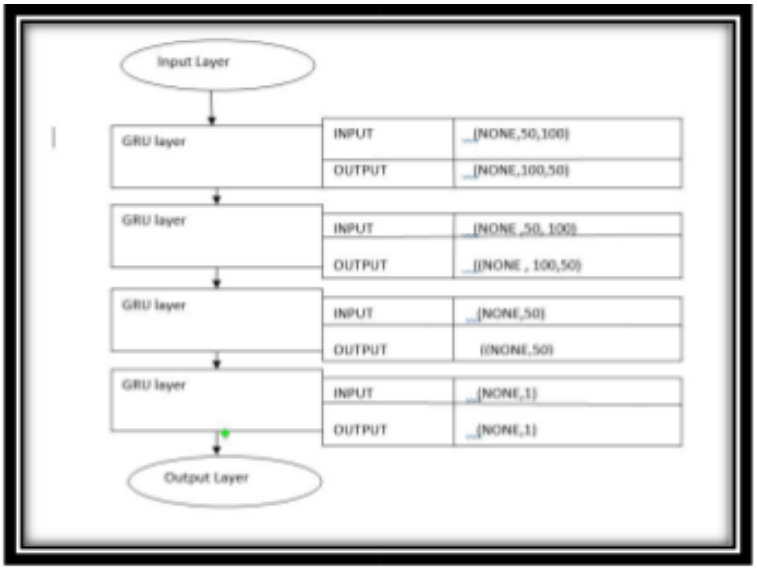


Fig 8: GRU OUTPUT

4.3 AUTO REGRESSIVE MOVING AVERAGE (ARIMA):

ARIMA (Auto Regressive Integrated Moving Average) is a family of models that 'explains' a given time series based on its own previous values, such as delays and prediction errors, such that the equation can be used to predict future values.

An ARIMA model is defined by three terms: p, d, and q.

p, d, q, where p is the AR term's order,

The order of the AR word is p.

The order of the MA word is q.

where d is the number of differencings required to stabilize the time series.

```
# ARIMA Model
model = ARIMA(df.close, order=(6, 1, 3))
model_fit = model.fit()
# summary of fit model
print(model_fit.summary())
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-static
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.7/dist-packages/statsmodels/base/model.py:606: ConvergenceWarning: Maximum Likelih
  ConvergenceWarning)
                               SARIMAX Results
==============================================================================
Dep. Variable:                  close   No. Observations:                 1753
Model:                 ARIMA(6, 1, 3)   Log Likelihood               -3005.594
Date:                Tue, 08 Mar 2022   AIC                           6031.188
Time:                        05:17:55   BIC                           6085.874
Sample:                             0   HQIC                          6051.402
                                 - 1753
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.5474      0.072     -7.605      0.000      -0.688      -0.406
ar.L2         -0.1769      0.073     -2.417      0.016      -0.320      -0.033
ar.L3         -0.6871      0.083     -8.278      0.000      -0.850      -0.524
ar.L4         -0.0830      0.027     -3.021      0.003      -0.137      -0.029
ar.L5          0.0818      0.023      3.487      0.000       0.036       0.128
ar.L6          0.0900      0.016      5.651      0.000       0.059       0.121
ma.L1          0.3607      0.074      4.865      0.000       0.215       0.506
ma.L2          0.0939      0.068      1.374      0.169      -0.040       0.228
ma.L3          0.7179      0.076      9.406      0.000       0.568       0.867
sigma2         1.8014      0.027     66.212      0.000       1.748       1.855
===================================================================================
Ljung-Box (L1) (Q):                   0.10   Jarque-Bera (JB):             19207.86
```

4.4 SCREENSHOTS OF THE VARIOUS STAGES OF THE PROJECT

4.4.1 Discovering/Analyzing the data:

Understand a given dataset.

Discovering/Analyzing the data

```
[ ] df.shape            #shows number of rows and columns in the dataset
```

```
(1258, 14)
```

```
[ ] df.columns
```

```
Index(['symbol', 'date', 'close', 'high', 'low', 'open', 'volume', 'adjClose',
       'adjHigh', 'adjLow', 'adjOpen', 'adjVolume', 'divCash', 'splitFactor'],
      dtype='object')
```

```
[ ] df.describe()
```

	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjOpen	adjVolume	divCash	splitFactor
count	1258.000000	1258.000000	1258.000000	1258.000000	1.258000e+03	1258.000000	1258.000000	1258.000000	1258.000000	1.258000e+03	1258.000000	1258.000000
mean	188.950545	190.749544	186.896469	188.723430	4.927098e+07	71.238460	71.953692	70.444818	71.179304	1.205522e+08	0.009380	1.002385
std	70.181322	70.961984	69.061847	69.900707	3.873169e+07	39.502144	39.950835	39.005628	39.483081	5.591478e+07	0.079172	0.084583
min	106.840000	110.030000	103.100000	104.540000	1.136204e+07	25.672732	25.889200	25.470380	25.708025	4.099995e+07	0.000000	1.000000
25%	143.187500	144.300000	142.202500	143.433750	2.371339e+07	41.041037	41.458644	40.611750	41.047695	8.360832e+07	0.000000	1.000000
50%	170.415000	171.860000	169.120000	170.415000	3.336960e+07	51.330761	52.059583	50.904935	51.303798	1.053918e+08	0.000000	1.000000
75%	209.557500	211.997500	207.605000	209.870000	6.390696e+07	112.518578	114.510343	111.335425	113.101473	1.406886e+08	0.000000	1.000000
max	506.090000	515.140000	500.330000	514.790000	3.326072e+08	165.300000	170.300000	164.530000	167.480000	4.479402e+08	0.820000	4.000000

4.4.2 Data validation and Preprocessing:

Cleaning up the data by converting raw data to clean dataset.

Cleaning up the data (Cleansing the data).

Converting raw data to clean dataset.

```
df.duplicated()   #Checking for duplicate values and eliminating them one by one
```

```
0       False
1       False
2       False
3       False
4       False
        ...
1253    False
1254    False
1255    False
1256    False
1257    False
Length: 1258, dtype: bool
```

```
[ ] df.isnull().sum()    #Checking for missing values
```

```
symbol        0
date          0
close         0
high          0
low           0
open          0
volume        0
adjClose      0
adjHigh       0
adjLow        0
adjOpen       0
adjVolume     0
divCash       0
splitFactor   0
dtype: int64
```

```
[ ] df1=df.reset_index()['close']
```

4.4.3 Exploration of data and it's visual analysis:

Visualization provides an effective way to identify summaries, structure, relationships, differences, and abnormalities in the data.

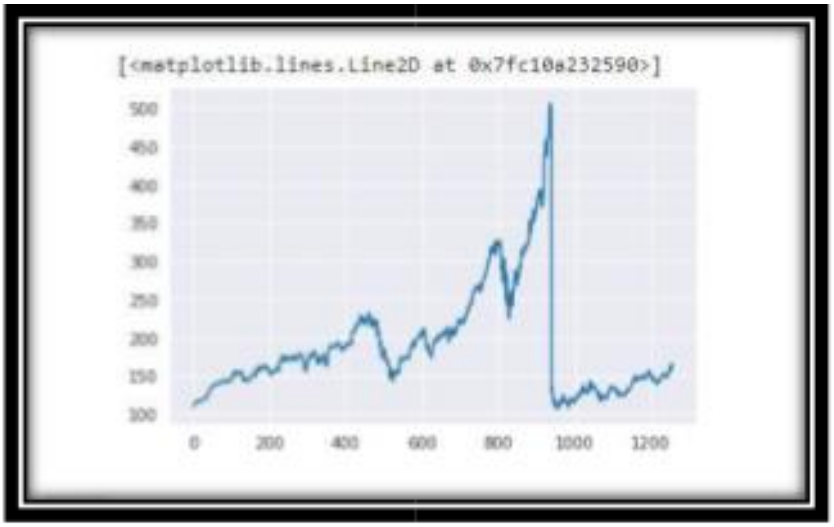


Fig 9: XY Function Plot

Fig 10: **Pair Plot**

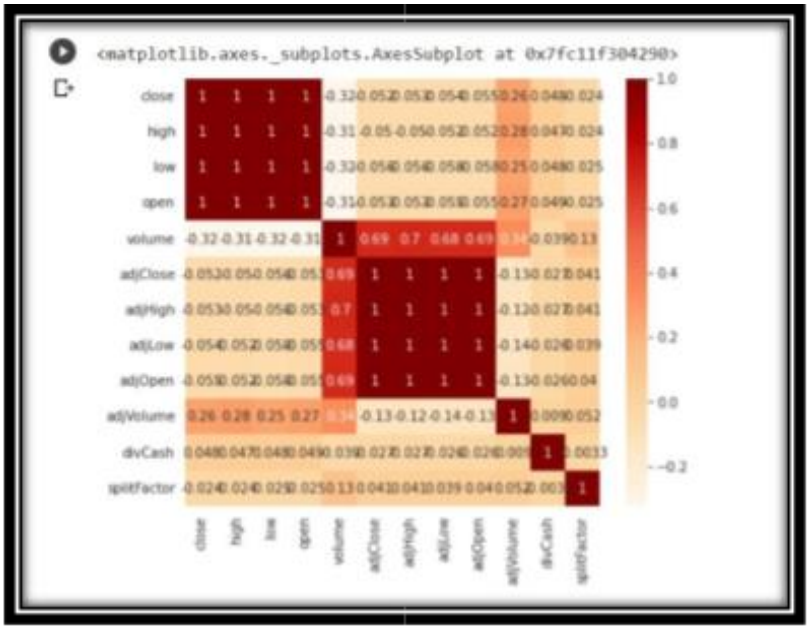


Fig 11: Heat Map

4.4.4 Data Scaling:

Before data are passed to the model, the data are scaled according to model requirements. In this way, this phase reshapes data to make them more suitable for the model.

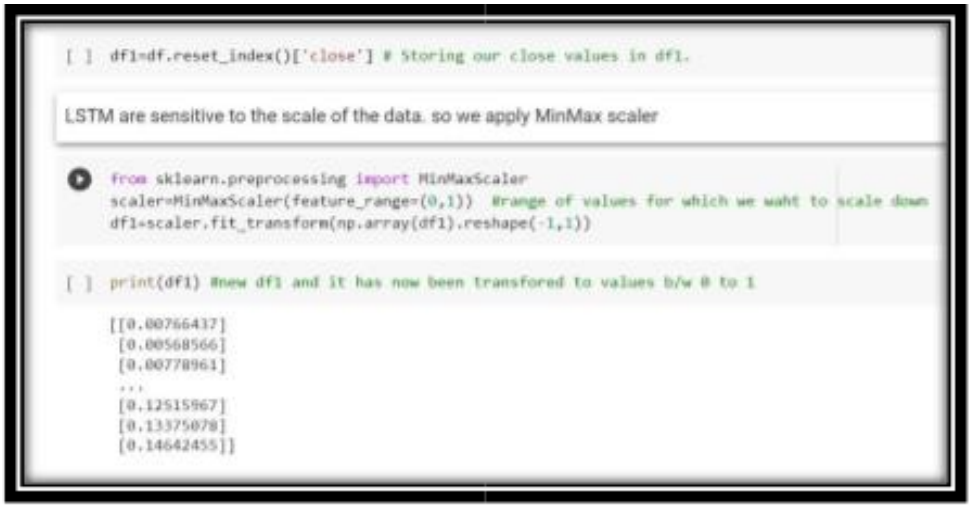


Fig 12: Scaling Data

4.4.4 Train-Test Split:

This phase splits data into training and testing data subsets. For example, data are divided into two parts per a ratio of 65% training data and 35% test data.

The final phase Model evaluated will be discussed ahead.

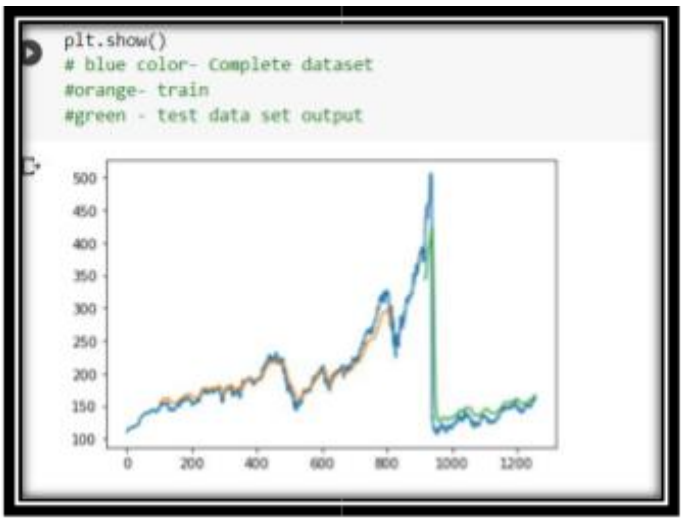


Fig 12: Train-Test Graph

The given blue line represents the whole dataset whereas orange is our training data used for model construction and green is the testing data.

Final model evaluation will be shown ahead.

4.5 Performance Metrics:

The performance of the proposed approach generally used metrics:

Number of epochs: This is defined as the total quantity of data that the machine must learn in a single iteration during the training stage.

Correlation coefficient: A measurement of the strength of the link between two variables. The Person correlation coefficient is the most used technique, and its formula for any two connections x and y is as follows:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2]\,[n\sum y^2 - (\sum y)^2]}}.$$

Root mean square error: This calculates the difference between two datasets. This is determined as the total of all observations divided by the number of observations (n) as the difference between the anticipated value (Pi) and the observed value (Oi):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(P_i - O_i)^2}{n}}.$$

CHAPTER 05

CONCLUSION

5.1 RESULTS AND DISCUSSION

For benchmark datasets, the model was run and implemented. From the entire dataset, the square of the correlation coefficient was utilized to discover a dominant feature, and then correlations between Close and High, Close and Low, Close and Open, and Close and Volume were obtained. Correlations between several types of market data are displayed.

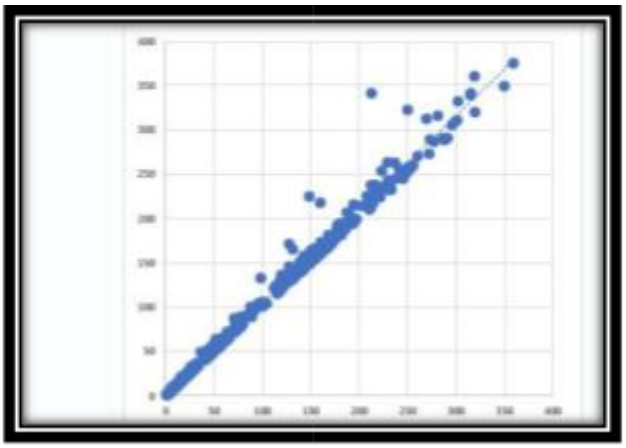
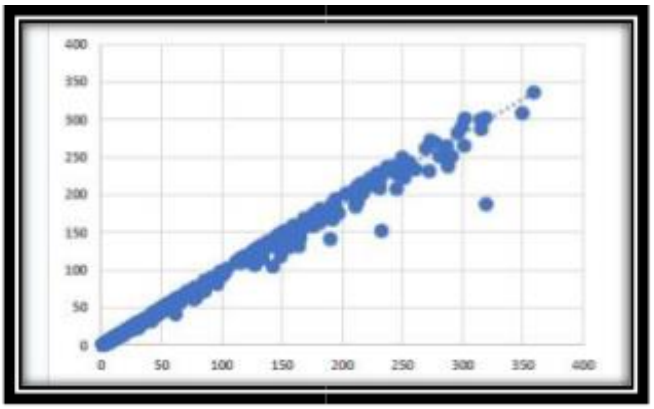


Fig 13: Correlations between open and market close

Any value of open can be found using the following equation:

Open = 1.0528 * Market Close – 0.1297.



Any value of Low can be found by putting the value of close in the following equation:

Low= .9342* Close+ .4016

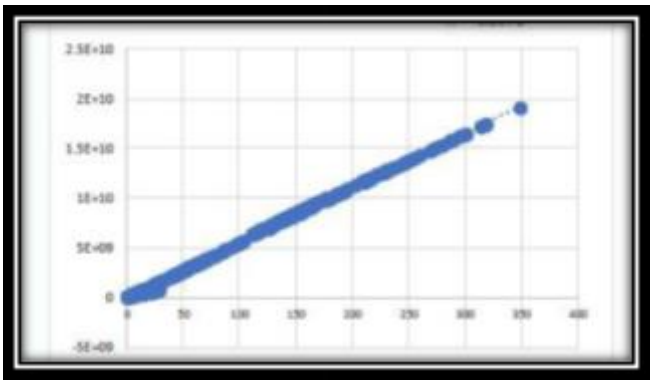


Fig 15: Correlations between close and open

Any value of open can be found by putting the value of close in the following equation:

Market Close= 0.9959* Market Open + 0.1777.

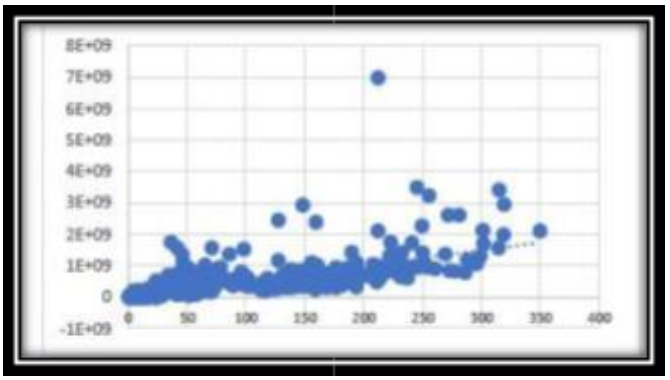


Fig 16: **Correlations between close and volume**

This means volume was not entirely dependent on close

5.2 MODEL EVALUATION

5.2.1 LSTM

Table 3: Values of different statistics for errors for LSTM

Statistic	Train Data Set Error	Test Data Set Error
Root mean square error (RMSE)	198.24714	174.26122

PREDICTION GRAPH OF LSTM:

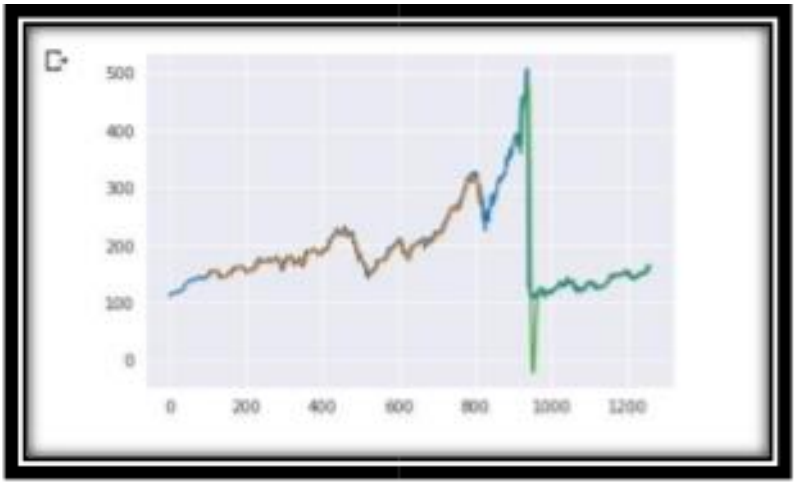


Fig 17: Actual vs Predicted values for LSTM

The blue line shows the values from the original dataset, while the orange line shows the training values and the green line shows the projected test values.

Further prediction for next 30 days is shown:

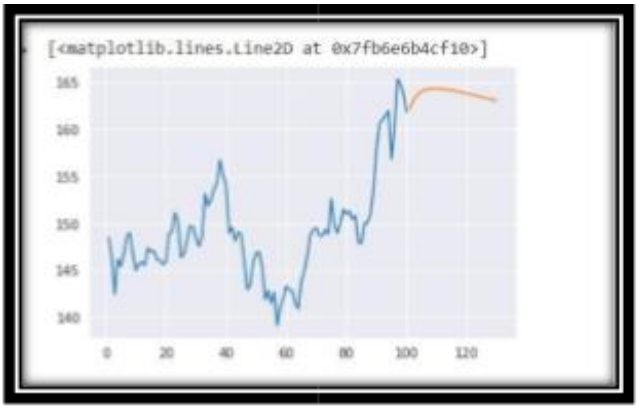


Fig 18: LSTM Prediction for 30 days

5.2.2 GRU MODEL:

Table 4: Values of different statistics for errors for GRU

Statistic	Train Data Set Error	Test Data Set Error
Root mean square error (RMSE)	198.24714	167.6786

PREDICTION GRAPH OF GRU:

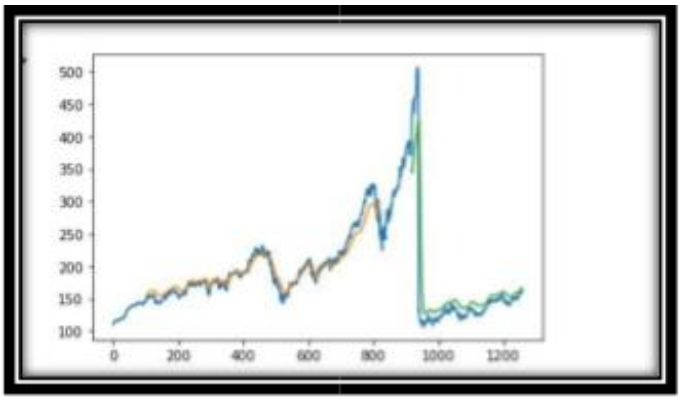


Fig 20: Actual vs Predicted value of GRU

The blue line shows the values from the original dataset, while the orange line shows the training values and the green line shows the projected test values.

The following is a forecast for the next 30 days:

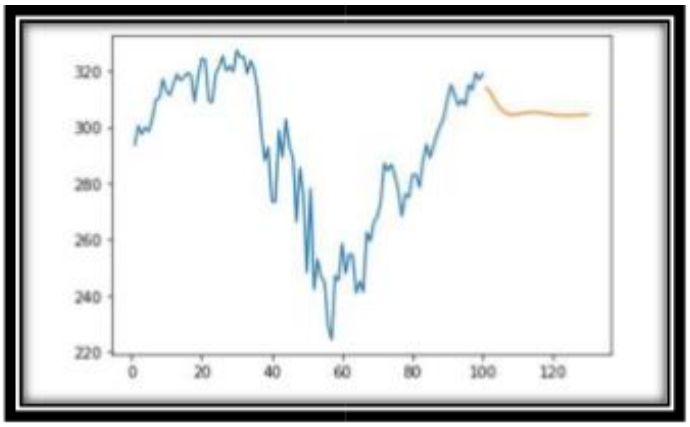


Fig 20: GRU Prediction for 30 days

5.2.3 ARIMA MODEL:

Table 4: Values of different statistics for errors for ARIMA

Statistic	Train Data Set Error	Test Data Set Error
Root mean square error (RMSE)	198.24714	167.6786

PREDICTION GRAPH OF ARIMA:

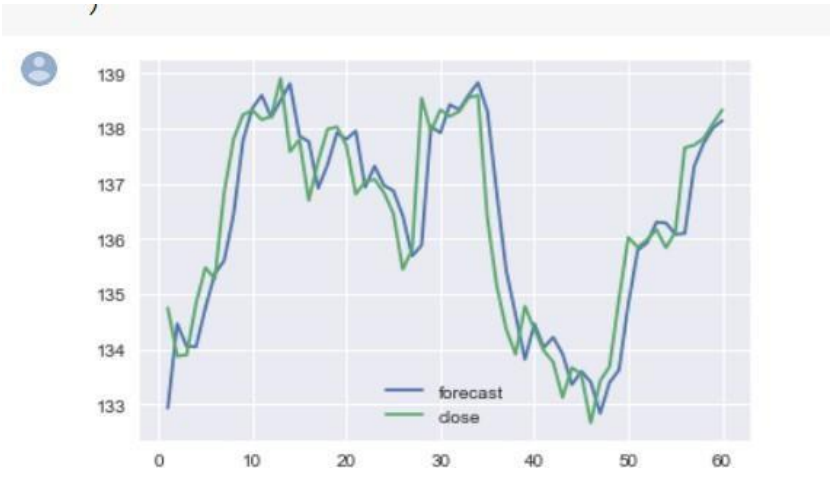


Fig 19: Actual vs Predicted values for ARIMA

RESIDUAL ERRORS OF ARIMA

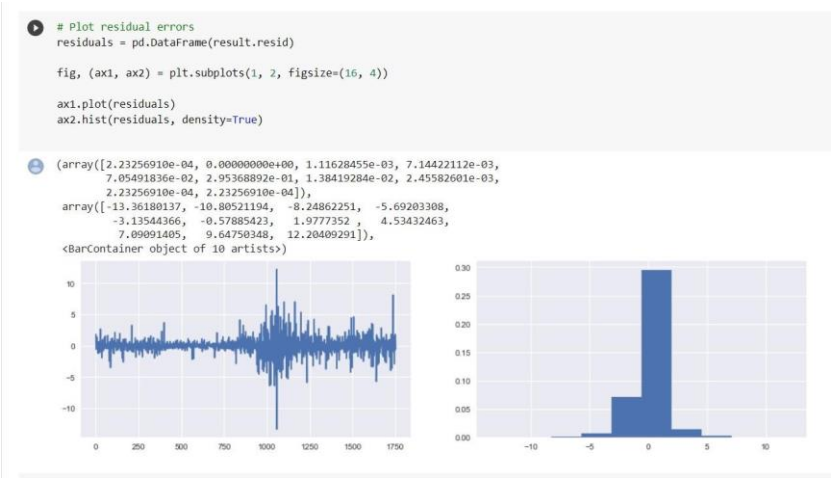


fig 22: Actual vs Predicted values for ARIMA

In conclusion, the findings suggest that proximity may play an important role in impacting the other characteristics. Furthermore, as seen by the findings for the proposed model trained on big datasets, the size of the dataset may impact future predictions.

To create the predicted model, the study solely considers the bitcoin closing price. It does not take into account other economic elements such as bitcoin news, government regulations, and market attitudes, which may be the project's future scope in order to estimate the price with greater precision. The forecast is confined to historical data. The ability to forecast on streaming data would increase the model's performance and predictability. The study solely includes the comparison between ARIMA vs LSTM vs GRU.

Based on these findings, the GRU model for the cryptocurrencies under consideration may be regarded as efficient and dependable. This model is regarded as the best. However, LSTM and ARIMA have worse accuracy than GRU and have significant disparities between the real and forecasted prices for our dataset.

The results of the experiments reveal that:

- The AI algorithm is trustworthy and suitable for bitcoin prediction.
- GRU predicts bitcoin prices better than LSTM and ARIMA, but all algorithms produce outstanding predictions overall.

5.3 FUTURE SCOPE

- We'd experiment with new machine learning methods in order to improve accuracy and reduce errors.
- We will create a website that contains all of the information about our project.
- To make this project accessible to everyone, we will also deploy it on cloud platforms.
- We'll look into other factors that could influence bitcoin market values.

The price volatility of cryptocurrency is affected and determined by factors such as a country's political system, public relations, and market policy. Other cryptocurrencies such as ripple, ethereum, lite coin, and others were not examined in our research. We will improve the model by applying it to these coins, making it more stable. Fuzzification can also be applied at the input.

REFERENCES

[1] D.L.K. Chuen, Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data, Academic Press, 2015.

[2] S. Nakamoto, "*Bitcoin: A peer-to-peer electronic cash system*," 2008.

[3] M. Amjad and D. Shah, "*Trading Bitcoin and Online Time Series Prediction*," in NIPS 2016 Time Series Workshop, 2017.

[4] D. Garcia and F. Schweitzer, "*Social signals and algorithmic trading of Bitcoin*," Royal Society Open Science, vol. 2, no. 9, 2015.

[5] R. Chen and M. Lazer, "*Sentiment Analysis of Twitter Feeds for the Prediction of Stock Market Movement*," Stanford Computer Science, no. 229, 2011, p. 15.

[6] A. Go, L. Huang and R. Bhayani, "*Twitter Sentiment Classification using Distant Supervision*," Stanford Computer Science, 2009.

[7] B. Pang, L. Lee and S. Vaithyanathan, "*Thumbs up: sentiment classification using machine learning techniques*," in ACL-02 conference on Empirical methods in natural language processing, Philadelphia, PA, USA, 2002.

[8] M. Dixon, D. Klabjan and J. H. Bang, "*Classification-based financial markets prediction using deep neural networks*," ArXiv, 2017.

[9] S. McNally, J. Roche and S. Caton, "*Predicting the price of Bitcoin using machine learning*," in 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), 2018.

[10] M. Daniela and A. BUTOI, "*Data mining on Romanian stock market using neural networks for price prediction*," Informatica Economica, vol. 17, no. 3, 2013.

[11] D. Shah and K. Zhang, "*Bayesian regression and Bitcoin*," in 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2015.