A

Major Project

On

CRYPTO CURRENCY PRICE ANALYSIS USING AI

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Yashwanth Kumaar Kakani	(217R1A05R8)
U. Vinod Kumar	(227R5A0523)
G. Manoj Reddy	(217R1A05M7)

Under the Guidance of

Dr. B. LAXMAIAH

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

April, 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled " **CRYPTO CURRENCY PRICE ANALYSIS USING AI**" being submitted **YASHWANTH KUMAAR KAKANI (217R1A05R8), U. VINOD KUMAR (227R5A0523)** and **G. MANOJ REDDY (217R1A05M7)** in partial fulfillment of the requirements for the award of the degree of B. Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2024-25.

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. B. Laxmaiah
(Associate Professor)
INTERNAL GUIDE

Dr. Nuthanakanti Bhaskar
HoD

Dr. A. Raji Reddy
DIRECTOR

Signature of External Examiner

Submitted for viva voice Examination held on ________________

ACKNOWLEDGEMENT

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project, we take this opportunity to express our profound gratitude and deep regard to our guide **Dr. B. Laxmaiah**, Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Dr. J. Narasimha Rao, Mr. K. Ranjith Reddy, Dr. K. Maheshwari, Mrs. K. Shilpa** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

My sincere appreciation also goes to **Dr. Nuthanakanti Bhaskar**, Head, for his encouragement and continuous support in ensuring the successful completion of my project.

I am deeply grateful to **Dr. A. Raji Reddy**, Director, for his cooperation throughout the course of this project. Additionally, I extend our profound gratitude to **Sri. Ch. Gopal Reddy**, Chairman, **Smt**. **C. Vasantha Latha**, Secretary and **Sri. C. Abhinav Reddy**, Vice-Chairman, for fostering an excellent infrastructure and a conducive learning environment that greatly contributed to our progress.

I also acknowledge and appreciate the guidance and assistance provided by the faculty and staff of **CMR Technical Campus**, whose contributions have been invaluable in bringing this project to fruition.

Lastly, I sincerely thank our families for their unwavering support and encouragement. We also extend our gratitude to the teaching and non-teaching staff of CMR Technical Campus for their guidance and assistance. Their contributions, along with the support of everyone who helped directly or indirectly, have been invaluable in the successful completion of this project.

K. YASHWANTH KUMAAR (217R1A05R8)

U. VINOD KUMAR (227R5A0523)

G. MANOJ REDDY (217R1A05M7)

ABSTRACT

Cryptocurrency is playing an increasingly important role in reshaping the financial system due to its growing popular appeal and merchant acceptance. While many people are making investments in Cryptocurrency, the dynamical features, uncertainty, the predictability of Cryptocurrency are still mostly unknown, which dramatically risk the investments. It is a matter to try to understand the factors that influence the value formation. In this project, we use advanced artificial intelligence frameworks of fully connected Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Recurrent Neural Network to analyse the price dynamics of Bitcoin, Etherum, and Ripple. We find that RNN tends to rely more on short-term history while LSTM tends to rely more on long-term dynamics, which indicate the efficiency of LSTM to utilize useful information hidden in historical memory is stronger than RNN. However, given enough historical information LSTM can achieve a higher accuracy, compared with RNN. This project provides a unique demonstration that Cryptocurrency market price is predictable. However, the explanation of the predictability could vary depending on the nature of the involved machine-learning model.

Keywords: cryptocurrency, artificial intelligence, price prediction, machine learning, market trends, RNN, LSTM, Bitcoins, Etherum, Ripple.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

Cryptocurrency is the peer-to-peer digital monetary and payment system that exist online via a controlled algorithm. When a miner cracks an algorithm to record a block of transactions to public ledger named blockchain and the cryptocurrency is created when the block is added to the blockchain. It allows people to store and transfer through encryption protocol and distributed network. Mining is a necessary and competitive component of the cryptocurrency system. The miner with more computational power has a better chance of finding a new coin than that of less. Bitcoin is the first and one of the leading digital currencies (its market capitalization had more than $ 7 billion in 2014, and then it increased significantly to $ 29 billion in 2017) which was first introduced by Satoshi Nakamoto in 2008. Among many features of bitcoin, the most impressive one is decentralization that it can remove the involvement of traditional financial sectors and monetary authorities effectively due to its blockchain network features. While an increasing number of people are making investments in Cryptocurrency, the majority of investors cannot get such profit for being inconsiderable to cryptocurrencies' dynamics and the critical factors that influence the trends of bitcoins. Therefore, raising people's awareness of vital factors can help us to be wise investors. Although market prediction is demanding for its complex nature the dynamics are predictable and understandable to some degree. For example, when there is a shortage of the bitcoin, its price will be increased by their sellers as investors who regard bitcoin as a profitable investment opportunity will have a strong desire to pay for bitcoin. Furthermore, the price of bitcoin may be easily influenced by some influential external factors such as political factors . Although existing efforts on Cryptocurrency analysis and prediction is limited, a few studies have been aiming to understand the Cryptocurrency time series and build statistical models to reproduce and predict price dynamics. Traditional time series methods are not very useful as cryptocurrencies are not precisely the same with stocks but can be deemed as a complementary good of existing currency system with sharp fluctuations features. Therefore, it is urgently needed to understand the dynamics of cryptocurrencies better and establish a suitable predictive modelling framework.

1.1 PROJECT PURPOSE

The purpose of this project is to develop an AI-powered system for cryptocurrency price analysis and prediction. By leveraging machine learning algorithms such as RNN, LSTM, and moving averages, the system aims to provide accurate price forecasting, trend analysis, and decision-making support for traders and investors. The project integrates blockchain technology for secure transactions and data management, ensuring transparency and reliability. It also features a user-friendly interface for users, agents, and admins to manage transactions efficiently

1.2 PROJECT FEATURES

This project offers a comprehensive cryptocurrency price analysis and trading system powered by AI and blockchain technology. It features advanced price prediction using RNN and LSTM models, enabling users to make informed investment decisions based on historical trends and real-time data. A secure authentication system ensures only verified users and agents can access the platform, with separate dashboards for users, agents, and administrators. The project supports seamless cryptocurrency transactions, allowing agents to buy from the admin and users to purchase from agents. Blockchain integration guarantees transparency and security by maintaining a distributed ledger for all transactions.

2. LITERATURE SURVEY

1) Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin

AUTHORS: Greaves, A., & Au, B.

Bitcoin is the world's leading cryptocurrency, allowing users to make transactions securely and anonymously over the Internet. In recent years, The Bitcoin the ecosystem has gained the attention of consumers, businesses, investors and speculators alike. While there has been significant research done to analyze the network topology of the Bitcoin network, limited research has been performed to analyze the network's influence on overall Bitcoin price. In this paper, we investigate the predictive power of blockchain network-based features on the future price of Bitcoin. As a result of blockchain-network based feature engineering and machine learning optimization, we obtain up-down Bitcoin price movement classification accuracy of roughly 55%.

2) Cryptocurrency value formation: an empirical analysis leading to a cost of production model for valuing bitcoin

AUTHORS: Hayes, A. S.

This paper aims to identify the likely source(s) of value that cryptocurrencies exhibit in the marketplace using cross sectional empirical data examining 66 of the most used such 'coins'. A regression model was estimated that points to three main drivers of cryptocurrency value: the difficulty in 'mining 'for coins; the rate of unit production; and the cryptographic algorithm employed. These amount to relative differences in the cost of production of one coin over another at the margin, holding all else equal. Bitcoin-denominated relative prices were used, avoiding much of the price volatility associated with the dollar exchange rate. The resulting regression model can be used to better understand the drivers of relative value observed in the emergent area of cryptocurrencies. Using the above analysis, a cost of production model is proposed for valuing bitcoin, where the primary input is electricity. This theoretical model produces useful results for both an individual producer, by setting breakeven points to start and stop production, and for the bitcoin exchange rate on a macro level. Bitcoin production seems to resemble a competitive commodity market; in theory miners will produce until their marginal costs equal their marginal product.

We consider the problem of planning the ISS cosmonaut training with different objectives. A pre-defined set of minimum qualification levels should be distributed between the crew members with minimum training time differences, training expenses or a maximum of the training level with a limitation of the budget.
First, a description of the cosmonaut training process is given.
Then four models are considered for the volume planning problem.
The objective of the first model is to minimize the differences between the total time of the preparation of all crew members, the objective of the second one is to minimize the training expenses with a limitation of the training level, and the objective of the third one is to maximize thetraining level with a limited budget. The fourth model considers the problem as an n-partition problem. Then two models are considered for the calendar planning problem.
For the volume planning problem, two algorithms are presented.
The first one is a heuristic with a complexity of (n) operations. The second one consists of a heuristic and exact parts, and it is based on the n-partition problem appro

3. Economic prediction using neural networks: the case of IBM daily stock returns

AUTHORS: H. White

A report is presented of some results of an ongoing project using neural-network modeling and learning techniques to search for and decode nonlinear regularities in asset price movements. The author focuses on the case of IBM common stock daily returns. Having to deal with the salient features of economic data highlights the role to be played by statistical inference and requires modifications to standard learning techniques which may prove useful in other contexts

4. Designing a neural network for forecasting financial and economic time series

AUTHORS: Kaastra and M. Boyd

Artificial neural networks are universal and highly flexible function approximators first used in the fields of cognitive science and engineering. In recent years, neural network applications in finance for such tasks as pattern recognition, classification, and time series forecasting have dramatically increased. However, the large number of parameters that must be selected to develop a neural network forecasting model have meant that the design process still involves much trial and error. The objective of this paper is to provide a practical introductory guide in the design of a neural network for forecasting economic time series data. An eight-step procedure to design a neural network forecasting model is explained including a discussion of trade offs in parameter selection, some common pitfalls, and points of disagreement among practitioners.

2.1 REVIEW OF RELATED WORK

The prediction of cryptocurrency prices has been a focal point in financial research, leading to the development of various models aimed at capturing the volatile nature of these digital assets. This review examines the evolution of methodologies employed, emphasizing the transition from traditional statistical approaches to advanced deep learning techniques, particularly focusing on Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks.

1. Traditional Statistical Approaches

Early attempts at forecasting cryptocurrency prices predominantly utilized statistical methods such as Autoregressive Integrated Moving Average (ARIMA) models. While these models offered insights into linear patterns within time series data, they often fell short in accurately capturing the nonlinear and volatile behaviors inherent in cryptocurrency markets.

2. Machine Learning-Based Approaches

With advancements in computational power, machine learning techniques were introduced to enhance predictive accuracy. Algorithms like Support Vector Machines (SVM), Random Forests, and Decision Trees were employed to model complex relationships within financial datasets. Despite their improved performance over traditional statistical methods, these approaches were limited by their reliance on handcrafted features and struggled with capturing long-term dependencies in sequential data.

3. Deep Learning-Based Approaches

The advent of deep learning brought significant improvements to time series forecasting. RNNs, designed to process sequential data, emerged as a natural fit for modeling temporal dependencies in cryptocurrency prices. However, standard RNNs faced challenges such as vanishing gradients, hindering their ability to learn long-term patterns.

To address these limitations, LSTM networks were introduced. LSTMs are a specialized form of RNNs capable of learning long-term dependencies through their unique gating mechanisms. Studies have demonstrated the effectiveness of LSTMs in predicting cryptocurrency prices, highlighting their ability to capture intricate temporal patterns and improve forecasting accuracy.

4. Comparison with the Proposed Approach

While existing methodologies have contributed significantly to cryptocurrency price prediction, challenges remain in achieving high accuracy and generalization across different market conditions. The proposed approach in this project leverages the capabilities of RNNs and LSTMs to model temporal dependencies effectively. By integrating these models, the system aims to enhance predictive accuracy, reduce errors, and provide a robust tool for stakeholders in the cryptocurrency market.

This review underscores the progression from traditional statistical methods to advanced deep learning models in cryptocurrency price prediction. The proposed methodology seeks to build upon previous research by offering a comprehensive and efficient solution tailored to the complexities of cryptocurrency markets.

2.2 EXISTING SYSTEM

The existing systems for cryptocurrency price analysis primarily rely on traditional algorithms that often struggle to cope with the rapid fluctuations and complexities of the market. Techniques such as linear regression, moving averages, and exponential smoothing have been commonly employed to analyze historical price data and generate forecasts. While these methods can provide basic insights, they typically fail to account for the non-linear relationships and high volatility inherent in cryptocurrency trading. Additionally, these older algorithms often lack the ability to process vast amounts of unstructured data, such as news articles and social media sentiment, which can significantly influence market movements. As a result, these systems may produce less accurate predictions and leave traders vulnerable to unforeseen market shifts, highlighting the need for more advanced analytical approaches that can adapt to the dynamic nature of the cryptocurrency landscape.

Limitations of the Existing System

- Existing systems cannot fully account for sudden events such as fuel price hikes or airline promotions.
- Broad recommendations (e.g., "buy now" or "wait") are given without clear explanations for users.
- Limited incorporation of real-time data, leading to less accurate predictions.
- The systems lack personalized predictions based on individual user preferences like preferred airlines or frequent flyer programs.
- Existing tools are slow to adapt to new data, which can result in missed opportunities to secure the best prices.

2.3 DEFINITION OF PROBLEM STATEMENT

The cryptocurrency market is highly volatile, with prices fluctuating rapidly due to factors such as trading volume, market demand, global regulations, and investor sentiment. Traditional financial models and manual analysis often fail to predict these price changes accurately, leading to increased investment risks. Investors and traders struggle to make informed decisions due to the unpredictable nature of cryptocurrency prices. Additionally, cryptocurrency transactions require secure and transparent management to prevent fraud, unauthorized access, and data manipulation. The lack of a reliable system for price prediction and secure transactions further complicates decision-making for investors. This project aims to address these challenges by developing an AI-powered cryptocurrency price prediction system using machine learning models like Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM). These models analyze historical price data and market trends to generate accurate price forecasts.

2.4 PROPOSED SYSTEM

The proposed system for cryptocurrency price analysis leverages advanced artificial intelligence and machine learning algorithms to enhance predictive accuracy and adaptability in the volatile cryptocurrency market. By integrating various data sources, including historical price data, technical indicators, and real-time sentiment analysis from social media platforms and news articles, this system can better capture the multifaceted influences on price movements. The use of deep learning techniques, such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, allows for the modeling of complex temporal patterns, improving the system's ability to forecast future price trends. Additionally, the proposed system employs a feedback loop mechanism that continuously updates its models based on new data, ensuring that the predictions remain relevant in the face of rapidly changing market dynamics. Overall, this approach aims to provide investors with more accurate insights and improved risk management strategies, thereby enhancing decision-making in cryptocurrency trading.

Advantages of the Proposed Systems

- **Enhanced Predictive Accuracy**: By utilizing advanced machine learning algorithms, the proposed system can capture complex patterns and non-linear relationships in the data.
- **Comprehensive Data Integration**: The system incorporates a wide range of data sources, including historical prices, technical indicators, and real-time sentiment analysis.
- **Adaptability to Market Changes**: The use of a feedback loop mechanism enables the system to continuously learn and adapt to new market conditions.
- **Improved Risk Management**: With more accurate forecasts and insights, investors can make better-informed decisions, leading to enhanced risk management strategies.

2.5 OBJECTIVES

- Accurate Price Forecasting: Leverage the capabilities of RNNs and LSTMs to model temporal dependencies in cryptocurrency market data, enabling precise predictions of future price movements.
- Risk Mitigation for Investors: Provide reliable forecasts to assist investors in making informed decisions, thereby reducing potential financial risks associated with the inherent volatility of cryptocurrency markets.
- Real-Time Predictive Analytics: Implement models capable of processing live data feeds to offer up-to-date price predictions, facilitating timely investment strategies.
- Enhanced Trading Strategies: Utilize predictive insights to optimize trading algorithms, aiming to maximize returns and improve the overall efficiency of cryptocurrency trading operations.
- Comprehensive Market Analysis: Incorporate various market indicators and sentiment analyses to enrich the predictive model, ensuring a holistic approach to understanding and forecasting price dynamics.

2.6 HARDWARE & SOFTWARE REQUIREMENTS

2.6.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Hard Disk : 256 GB.
- Mouse : Optical Mouse.
- RAM : 6 GB.

2.6.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system : Windows 10 or Above.
- Coding Language : Python.
- Designing : Html, CSS, JavaScript.
- Data Base : MySQL.

3.SYSTEM ARCHITECTURE & DESIGN

Project architecture refers to the structural framework and design of a project, encompassing its components, interactions, and overall organization. It provides a clear blueprint for development, ensuring efficiency, scalability, and alignment with project goals. Effective architecture guides the project's lifecycle, from planning to execution, enhancing collaboration and reducing complexity.

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for Crypto currency price analysis using AI.

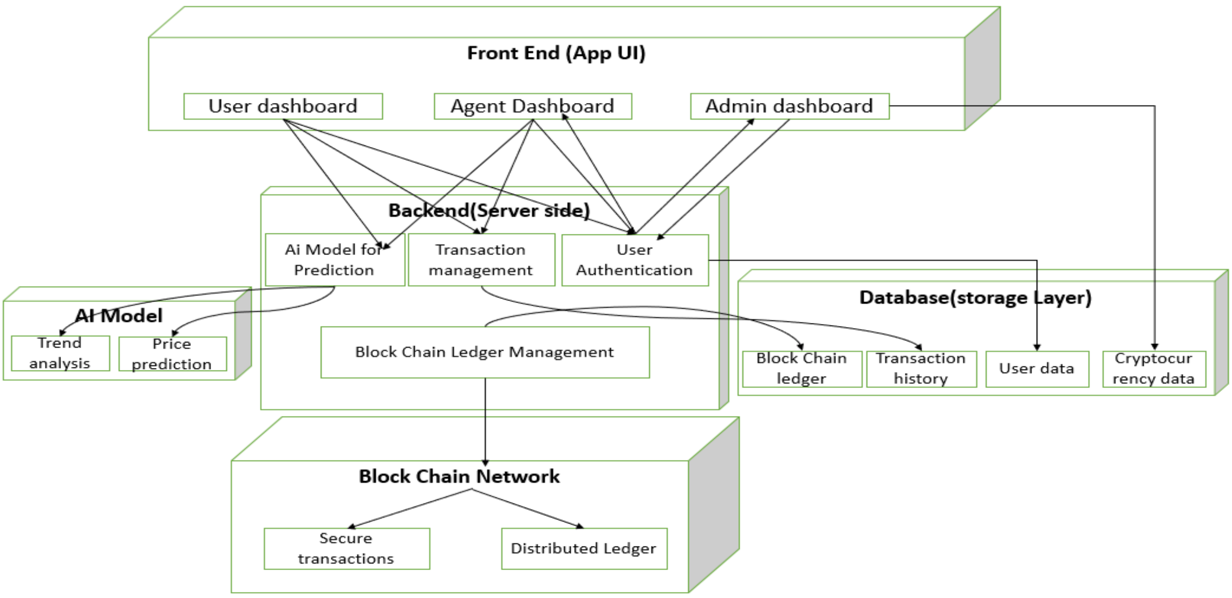

Figure 3.1: Project Architecture of Crypto currency price analysis using AI.

3.2 DESCRIPTION

1. **Objective**: To accurately predict airfare prices using machine learning algorithms, helping users identify the best times to book flights.
2. **Machine Learning Algorithms**: Utilizes algorithms such as Support Vector Machines (SVM), regression trees, random forest, linear regression, Extreme Learning Machine (ELM), and Multi-Layer Perceptron (MLP) for prediction.
3. **Data Analysis**: Analyzes historical flight data to identify patterns in pricing, influenced by factors like booking time, demand, and airline competition.
4. **Real-Time Data Integration**: Incorporates real-time data, such as current airline pricing trends, to provide dynamic and timely predictions.
5. **Personalization**: Offers personalized predictions tailored to user preferences (e.g., favorite airlines, travel dates, loyalty programs), making the forecasts more relevant.
6. **Transparency**: Provides clear explanations for price changes, offering insights into why prices are expected to rise or fall.
7. **Use Case**: Aids both travelers in saving on airfare and airlines in optimizing pricing strategies, providing a data-driven tool for the travel industry.
8. **Adaptability**: The system can quickly adapt to market changes, ensuring that users receive the most accurate and up-to-date forecasts.

3.3 DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

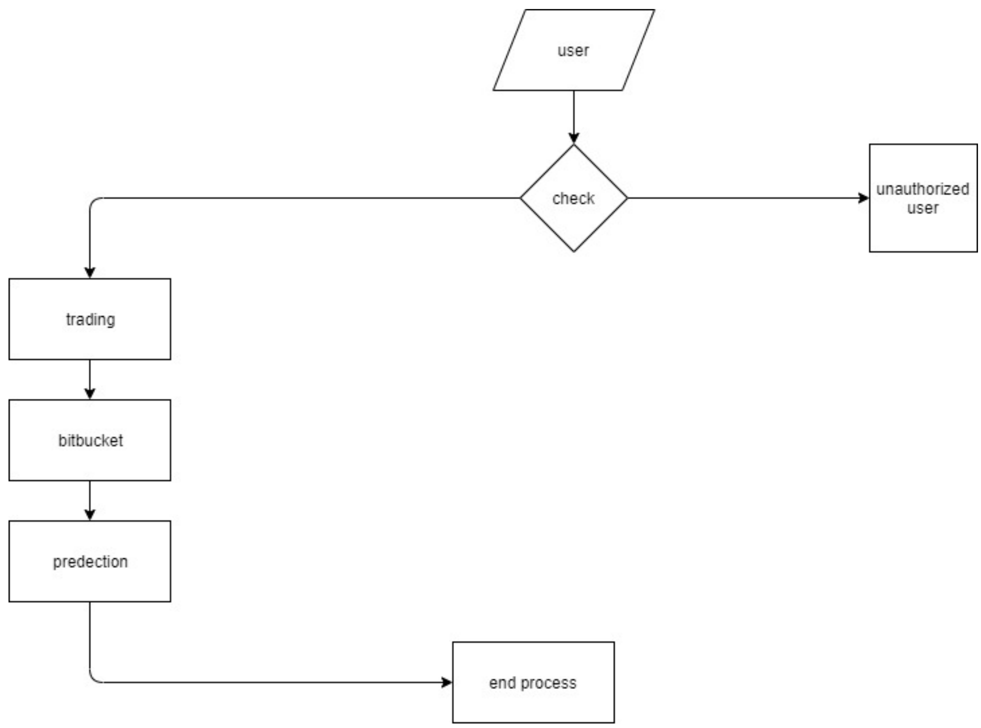

Figure 3.2 : Data flow diagram of A Crypto Currency Price Analysis Using AI for Admin.

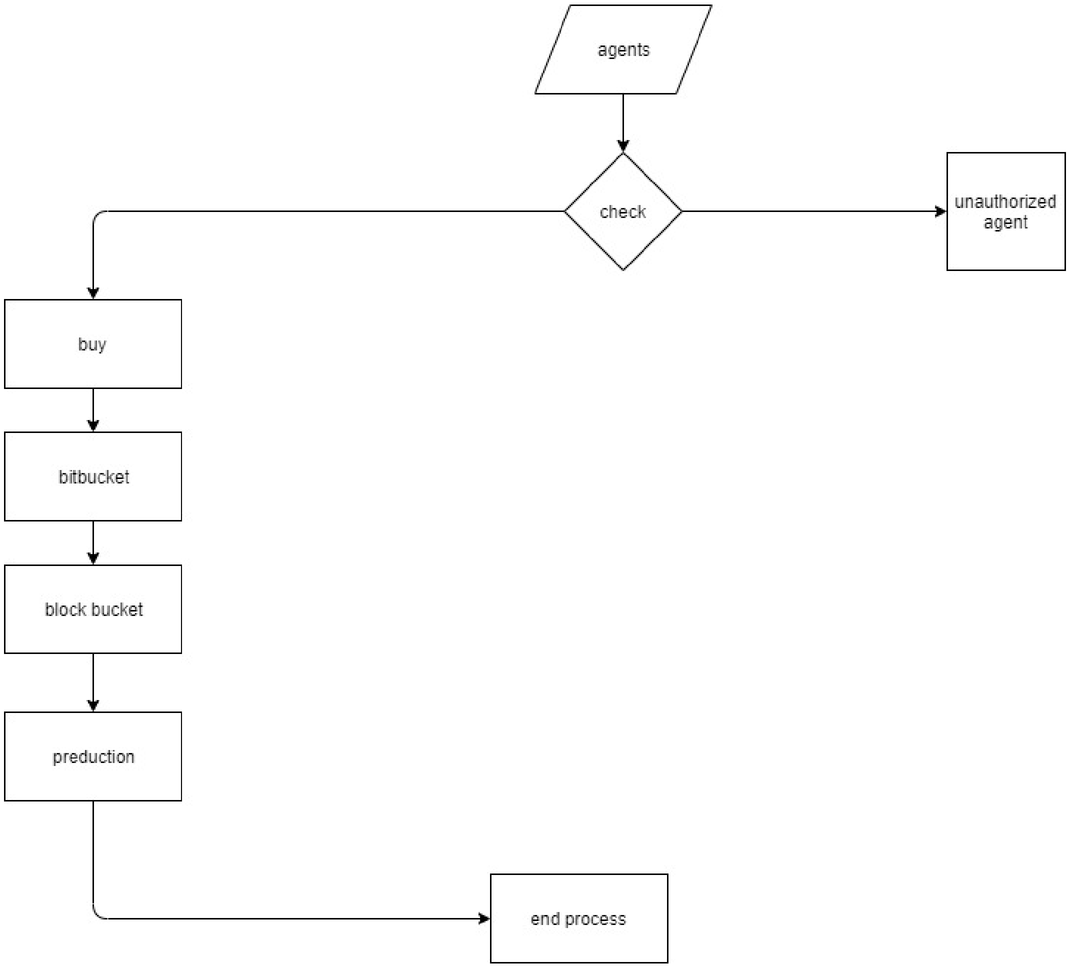

Figure 3.3 : Data flow diagram of A Crypto Currency Price Analysis Using AI for Agent.

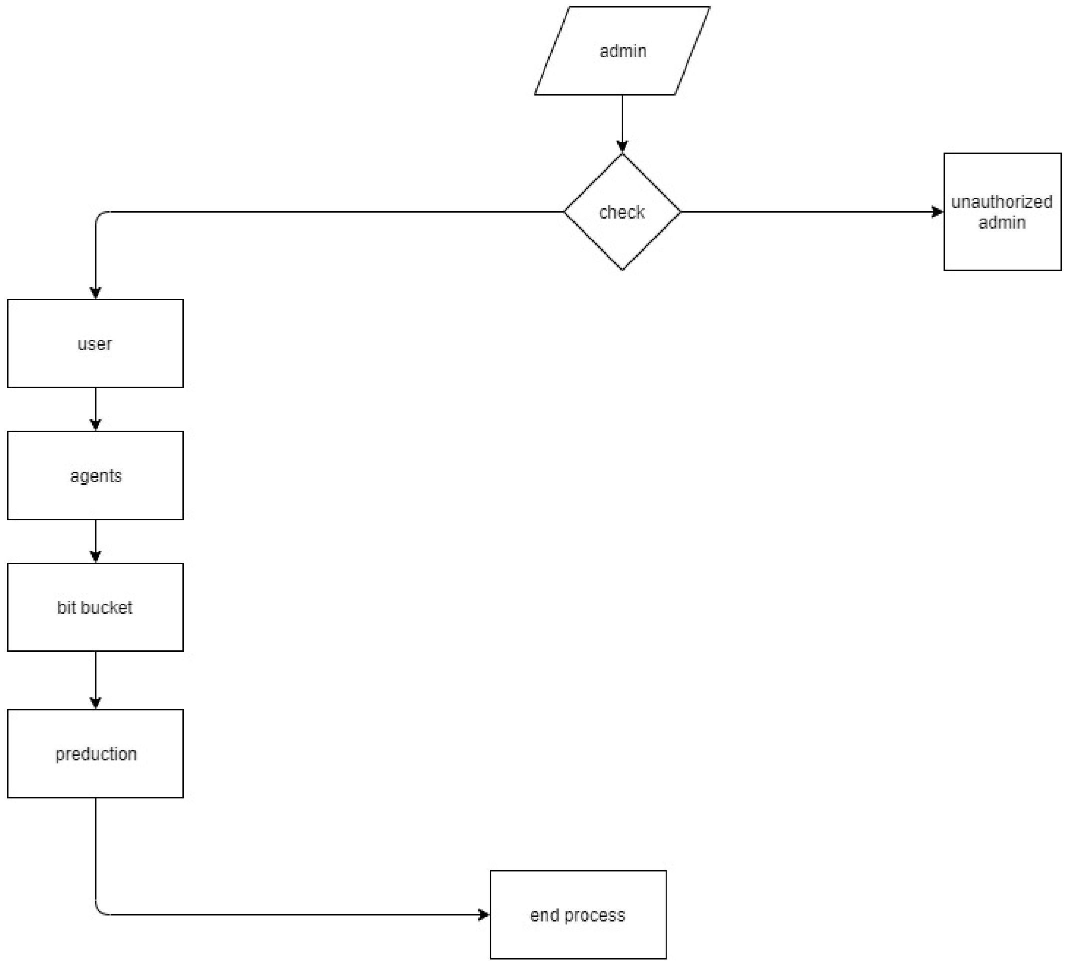

Figure 3.4 : Data flow diagram of A Crypto Currency Price Analysis Using AI for Admin.

4. IMPLEMENTATION

The implementation phase of a project involves executing the planned strategies and tasks. It requires meticulous coordination, resource allocation, and monitoring to ensure that objectives are met efficiently. Effective implementation is crucial for achieving project goals and delivering expected outcomes within the set timeline and budget constraints.

4.1 ALGORITHMS USED

Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed to process sequential data by leveraging their inherent temporal dynamics. Unlike traditional neural networks, RNNs possess internal memory that enables them to maintain information about previous inputs, making them particularly effective for tasks where context and order are essential. This characteristic allows RNNs to exhibit temporal behavior, which is crucial for applications such as time-series forecasting, natural language processing, and speech recognition.

Advantages of RNNs:

- Temporal Data Processing: RNNs are adept at handling sequential data, capturing temporal dependencies that are vital for understanding context over time.
- Parameter Sharing: They utilize the same parameters across all time steps, reducing the overall number of parameters and enhancing generalization.

Disadvantages of RNNs:

- Vanishing and Exploding Gradients: During training, RNNs can suffer from vanishing or exploding gradient problems, making it challenging to learn long-term dependencies.
- Limited Long-Term Memory: Standard RNNs may struggle to retain information over extended sequences, hindering their ability to model long-range dependencies effectively.

Long Short-Term Memory Networks (LSTMs)

Long Short-Term Memory (LSTM) networks are a specialized type of RNN designed to address the limitations of traditional RNNs, particularly the challenges associated with learning long-term dependencies. Introduced by Hochreiter and Schmidhuber in 1997, LSTMs incorporate a unique architecture comprising memory cells and gating mechanisms—namely, input, output, and forget gates. These gates regulate the flow of information, allowing the network to maintain and update its memory over extended periods. This structure enables LSTMs to effectively capture both short-term and long-term patterns in sequential data, making them highly suitable for tasks such as language modeling, machine translation, and time-series prediction.

Advantages of LSTMs:

- Long-Term Dependency Learning: LSTMs are specifically designed to learn and retain long-term dependencies, overcoming the vanishing gradient problem inherent in traditional RNNs.
- Effective Sequence Modeling: They excel in modeling sequences where context over long durations is crucial, enhancing performance in complex temporal tasks.

Disadvantages of LSTMs:

- Computational Complexity: The intricate architecture of LSTMs, with multiple gating mechanisms, leads to increased computational requirements and longer training times.
- Resource Intensive: Training LSTMs can be resource-intensive, necessitating substantial computational power, especially for large-scale applications.

Applications in Cryptocurrency Price Prediction

In the context of cryptocurrency price prediction, both RNNs and LSTMs can be employed to analyse historical price data to forecast future price movements. RNNs process sequences of past prices to identify temporal patterns, while LSTMs, with their ability to capture long-term dependencies, can model more complex patterns in the price data, potentially leading to more accurate predictions. However, due to the inherent volatility and complexity of cryptocurrency markets, LSTMs are often preferred over standard RNNs for this application.

By leveraging the strengths of RNNs and LSTMs, models can effectively analyse temporal data, capturing both short-term fluctuations and long-term trends, thereby enhancing the accuracy and reliability of predictions in various sequential data tasks.

To implement this project we have designed following modules:

Implementing a cryptocurrency price prediction system using Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks involves several key modules to ensure accurate and efficient forecasting. The primary components include:

1. Data Collection and Preprocessing Module: This module is responsible for gathering historical cryptocurrency price data and other relevant features, such as trading volumes and market indicators. The collected data is then pre-processed to handle missing values, normalize features, and structure the dataset appropriately for model training.

2. Feature Engineering Module: In this module, additional features are derived from the raw data to enhance the predictive power of the model. This may include technical indicators, moving averages, and sentiment analysis scores obtained from social media platforms.

3. Model Development Module: This core component involves designing and implementing the RNN and LSTM architectures. The models are configured to capture temporal dependencies in the data, with LSTMs specifically addressing long-term dependencies to improve prediction accuracy.

4. Training and Evaluation Module: Here, the models are trained using historical data, and their performance is evaluated using metrics such as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Cross-validation techniques may be employed to ensure robustness and prevent overfitting.

5. Prediction and Visualization Module: Once trained, the models generate future price predictions, which are then visualized through interactive dashboards or graphical representations. This module aids users in interpreting the forecasts and making informed decisions.

6. Deployment and Monitoring Module: This final module focuses on deploying the predictive models into a production environment, enabling real-time forecasting.

4.2 SAMPLE CODE

Urls.py

```
from django.contrib import admin
from django.urls import path
from .views import index,users,agents,admins,usersignup,agentsignup,logout
from users.views import
bituserregister,userlogincheck,StartUserTrading,UserBuyQuantity,UserBuyingCoins,Us
erTransactionsHistory,UserPredictionTest,UserPredictTestProcess
from agents.views import
bitagentregister,agentlogincheck,AgentBuyCrypto,agentbuycurrency,AgentTransactions
,AgentHadCoins,AgentLedgerStatus,AgentPredectionTest,AgentredictTestProcess
from admins.views import
adminlogincheck,viewusers,viewagents,activatewaitedusers,activatewaitedagents,curren
trate,updatecryptocurrency,AdminGetLedger

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',index,name='index'),
    path('index/', index, name='index'),
    path('users/',users,name='users'),
    path('agents/',agents,name='agents'),
    path('admins/',admins,name='admins'),
    path('usersignup/',usersignup,name='usersignup'),
    path('agentsignup/',agentsignup,name='agentsignup'),
    path('logout/',logout,name='logout'),

    path('bituserregister/',bituserregister,name='bituserregister'),
    path('userlogincheck/',userlogincheck,name='userlogincheck'),
    path('StartUserTrading/',StartUserTrading,name='StartUserTrading'),
    path('UserBuyQuantity/',UserBuyQuantity,name='UserBuyQuantity'),
    path('UserBuyingCoins/',UserBuyingCoins,name='UserBuyingCoins'),

path('UserTransactionsHistory/',UserTransactionsHistory,name='UserTransactionsHistor
y'),
    path('UserPredictionTest/',UserPredictionTest,name='UserPredictionTest'),



    path('bitagentregister/',bitagentregister,name='bitagentregister'),
    path('agentlogincheck/',agentlogincheck,name='agentlogincheck'),
    path('AgentBuyCrypto/',AgentBuyCrypto,name='AgentBuyCrypto'),

path('agentbuycurrency/<currencyname>',agentbuycurrency,name='agentbuycurrency'),
    path('AgentTransactions/',AgentTransactions,name='AgentTransactions'),
    path('AgentHadCoins/',AgentHadCoins,name='AgentHadCoins'),
    path('AgentLedgerStatus/',AgentLedgerStatus,name='AgentLedgerStatus'),
    path('AgentPredectionTest/',AgentPredectionTest,name='AgentPredectionTest'),
```

```
path('AgentredictTestProcess/<value>',AgentredictTestProcess,name='AgentredictTestP
rocess'),


    path('adminlogincheck/',adminlogincheck,name='adminlogincheck'),
    path('viewusers/',viewusers,name='viewusers'),
    path('viewagents/',viewagents,name='viewagents'),
    path('activatewaitedusers/',activatewaitedusers,name='activatewaitedusers'),
    path('activatewaitedagents/',activatewaitedagents,name='activatewaitedagents'),
    path('currentrate/',currentrate,name='currentrate'),

path('updatecryptocurrency/<curr>',updatecryptocurrency,name='updatecryptocurrency'
),
    path('AdminGetLedger/',AdminGetLedger,name='AdminGetLedger'),

path('UserPredictTestProcess/<value>',UserPredictTestProcess,name='UserPredictTestP
rocess'),



]
```

Admin side **models.py**

```
from django.db import models
import datetime
from django.utils import timezone

# Create your models here.

class cryptcurrencyratemodel(models.Model):
    currencytype=models.CharField(max_length=100, primary_key=True)
    doller=models.FloatField()
    rupee=models.FloatField()
    originalprice = models.FloatField()

    def __str__(self):
        return self.currencytype
    class Meta:
        db_table = 'currencyrate'

class CurrencyUpdateModel(models.Model):
    id = models.AutoField(primary_key=True)
    currencyname = models.CharField(max_length=100)
    conversionRate = models.FloatField()
    newCurrencyValue = models.FloatField()
    originalCurrencyValue = models.FloatField()
    chnageValue = models.FloatField()
    profitorloss = models.CharField(max_length=50)
    changedate = models.DateTimeField()
```

```
    def __str__(self):
        return self.currencyname
    class Meta:
        db_table = 'currencychnagetable'
        unique_together = ('currencyname', 'changedate',)
```

Admins Side **Views.py**

```
from django.shortcuts import render,HttpResponse
from django.contrib import messages
from users.models import BitUserRegisterModel,BlockChainLedger
from agents.models import BitAgentRegisterModel
from .models import cryptcurrencyratemodel,CurrencyUpdateModel
import string
import random
from datetime import date
from django.db.models import Sum

# Create your views here.

def adminlogincheck(request):
    if request.method=='POST':
        usrid = request.POST.get('adminid')
        pswd  = request.POST.get('pswd')
        print("User ID is = ", usrid)
        if usrid == 'admin' and pswd == 'admin':
            return render(request, 'admins/adminhome.html')
        else:
            messages.success(request, 'Please Check Your Login Details')
    return render(request, 'admins.html')

def viewusers(request):
    dict = BitUserRegisterModel.objects.all()
    return render(request, 'admins/userslist.html', {'objects': dict})

def viewagents(request):
    dict = BitAgentRegisterModel.objects.all()
    return render(request,'admins/agentslist.html',{'objects':dict})
def activatewaitedusers(request):
    if request.method=='GET':
        id = request.GET.get('uid')
        status = 'activated'
        print("PID = ", id, status)
        authkey = genSecretKey(8)
        BitUserRegisterModel.objects.filter(id=id).update(status=status,authkey=authkey)
        registerusers = BitUserRegisterModel.objects.all()
        return render(request, 'admins/userslist.html', {'objects': registerusers})
def activatewaitedagents(request):
    if request.method=='GET':
        id = request.GET.get('uid')
```

```
        status = 'activated'
        print("PID = ", id, status)
        authkey = genSecretKey(8)
        BitAgentRegisterModel.objects.filter(id=id).update(status=status,
authkey=authkey)
        registerusers = BitAgentRegisterModel.objects.all()
        return render(request, 'admins/agentslist.html', {'objects': registerusers})

def genSecretKey(stringLength=8):
    """Generate a random string of letters and digits """
    lettersAndDigits = string.ascii_letters + string.digits
    return ''.join(random.choice(lettersAndDigits) for i in range(stringLength))

def currentrate(request):
    dict = cryptcurrencyratemodel.objects.all()
    dict2 = CurrencyUpdateModel.objects.all()
    return render(request,'admins/cryptoratecurrent.html',{'objects':dict,'objects1':dict2})

def updatecryptocurrency(request,curr):
    rate = request.GET.get('rate')
    print('Rate = ',type(rate),' Currency ',type(curr))
    incrementRate = float(rate)
    if incrementRate>0:
        check = cryptcurrencyratemodel.objects.get(currencytype=curr)
        currentRate = check.doller
        currentRupee = check.rupee
        originalDollerrate = check.originalprice
        originalRupee = check.originalprice

        newRupee = (incrementRate * currentRupee) / 100
        newCurrencyVal = (incrementRate * currentRate) / 100
        print('Updated Currency ', newCurrencyVal)
        today = date.today()
        print("Today's date:", today)

        # changes = newCurrencyVal - originalDollerrate
        changes = newCurrencyVal + currentRate
        newRup = newRupee + currentRupee
        print("Chnages is ", changes)
        currencygain = ''

        if changes > currentRate:
            currencygain = 'Gain'
        else:
            currencygain = "loss"
        print('Currency is ', currencygain)
        CurrencyUpdateModel.objects.create(currencyname=curr,      conversionRate=rate,
newCurrencyValue=changes,
                                    originalCurrencyValue=originalDollerrate,
chnageValue=changes,
```

```
                        profitorloss=currencygain, changedate=today)
        cryptcurrencyratemodel.objects.filter(currencytype=curr).update(doller=changes,
rupee=newRup)

        dict = cryptcurrencyratemodel.objects.all()
        dict2 = CurrencyUpdateModel.objects.all()
        return render(request, 'admins/cryptoratecurrent.html', {'objects': dict, 'objects1':
dict2})
    elif incrementRate==0:
        print("Please Check Yhe Conversion rate")
    else:
        print("Currency Decrease Starts")
        check = cryptcurrencyratemodel.objects.get(currencytype=curr)
        currentRate = check.doller
        currentRupee = check.rupee
        originalDollerrate = check.originalprice
        originalRupee = check.originalprice

        newRupee = (abs(incrementRate) * currentRupee) / 100
        newCurrencyVal = (abs(incrementRate) * currentRate) / 100
        print('Updated Currency ', newCurrencyVal)
        today = date.today()
        print("Today's date:", today)

        # changes = newCurrencyVal - originalDollerrate
        changes =currentRate - newCurrencyVal
        newRup = currentRupee - newRupee
        print("Chnages is ", changes)
        currencygain = ''

        if changes > currentRate:
            currencygain = 'gain'
        else:
            currencygain = "loss"
        print('Currency is ', currencygain)
        CurrencyUpdateModel.objects.create(currencyname=curr,     conversionRate=rate,
newCurrencyValue=changes,
                        originalCurrencyValue=originalDollerrate,
chnageValue=changes,
                        profitorloss=currencygain, changedate=today)
        cryptcurrencyratemodel.objects.filter(currencytype=curr).update(doller=changes,
rupee=newRup)

        dict = cryptcurrencyratemodel.objects.all()
        dict2 = CurrencyUpdateModel.objects.all()
        return render(request, 'admins/cryptoratecurrent.html', {'objects': dict, 'objects1':
dict2})
```

```
def AdminGetLedger(request):
    check = BlockChainLedger.objects.aggregate(Sum('blockchainmoney'))
    x = check.get("blockchainmoney__sum")
    x = round(x, 2)
    print('Totoal Ledger Sum ', x)
    dict = BlockChainLedger.objects.all()
    return render(request, 'admins/adminsblock.html', {'objects': dict, 'ledger': x})
```

agent **models.py**

```
from django.db import models
from django.utils import timezone

    currencyName = models.CharField(max_length=100)
    customeremail = models.CharField(max_length=100)
    quantity = models.IntegerField()

    def __str__(self):
        return self.customeremail
    class Meta:
        db_table = "CustomerContainCoins"
        unique_together = ('currencyName', 'customeremail',)
class UserBuyingCryptoModel(models.Model):
    id = models.AutoField(primary_key=True)
    customername = models.CharField(max_length=100)
    email = models.CharField(max_length=100)
    currencyname = models.CharField(max_length=100)
    quantity = models.IntegerField()
    agentemail = models.CharField(max_length=100)
    singlecoingamount = models.FloatField()
    payableammount = models.FloatField()
    cardnumber = models.CharField(max_length=100)
    nameoncard = models.CharField(max_length=100)
    cardexpiry = models.CharField(max_length=100)
    cvv = models.IntegerField()
    cdate= models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.id
    class Meta:
        db_table = 'UserBuyingCryptoModel'

class BlockChainLedger(models.Model):
    id = models.AutoField(primary_key=True)
    customeremail = models.CharField(max_length=100)
    agentemail = models.CharField(max_length=100)
    currencyname = models.CharField(max_length=100)
    quantity = models.IntegerField()
    paidammout = models.FloatField()
    blockchainmoney = models.FloatField()
```

```
    def __str__(self):
        return self.id
    class Meta:
        db_table = "BlockChainLedger"
```

user side **views.py**

```
from django.shortcuts import render, HttpResponse,redirect
from .models import BitUserRegisterModel, CustomerHadCoins, UserBuyingCryptoModel,BlockChainLedger
from django.contrib import messages
from agents.models import AgentHadCrypto
from admins.models import cryptcurrencyratemodel
from django.conf import settings
import os
import pandas as pd
import datetime as dt
from datetime import datetime
import matplotlib.pyplot as plt
from .lstmann import predictionstart
from .algo.generatedata import GetData


# Create your views here.

def bituserregister(request):
    if request.method == 'POST':
        email = request.POST.get('email')
        pswd = request.POST.get('pswd')
        username = request.POST.get('username')
        mobile = request.POST.get('mobile')
        pan = request.POST.get('pan')
        state = request.POST.get('state')
        location = request.POST.get('location')
        print("Valid Form = ", email)
        try:
            rslts = BitUserRegisterModel.objects.create(email=email, pswd=pswd, username=username, mobile=mobile,
                                                        pan=pan, state=state, location=location)
            if rslts is None:
                print("Invalid Data ", rslts)
                messages.success(request, 'Email ID already exist, Registration Failed ')
            else:
                print("Valid Data ", rslts)
                messages.success(request, 'Registration Success')
        except:
            messages.success(request, 'Email ID already exist, Registration Failed ')
            return render(request, 'users/usersignup.html', {})
    else:
        print("Invalid Form Data")
        messages.success(request, 'Email ID already exist, Registration Failed ')
```

5. RESULTS & DISCUSSION

The following screenshots showcase the results of our project, highlighting key features and functionalities. These visual representations provide a clear overview of how the system performs under various conditions, demonstrating its effectiveness and user interface. The screenshots serve as a visual aid to support the project's technical and operational achievements.

5.1 HOME :

In below screen, Shows that home page/Main interface of the crypto currency price analysis using AI.

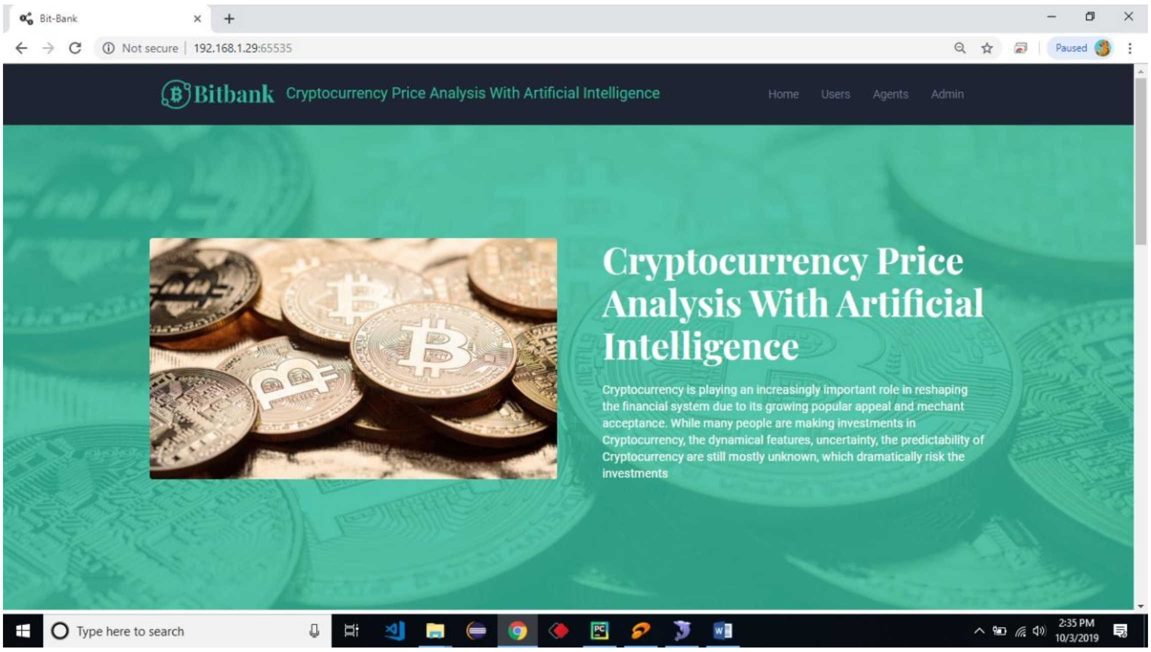

Figure 5.1 : GUI/ main Interface of a Crypto Currency Price Analysis Using AI

5.2 USER REGISTRATION PAGE :

In below screen, A new user can create trading account by clicking register option, after filling required details. Account will be created on wating for the activation by the admin.

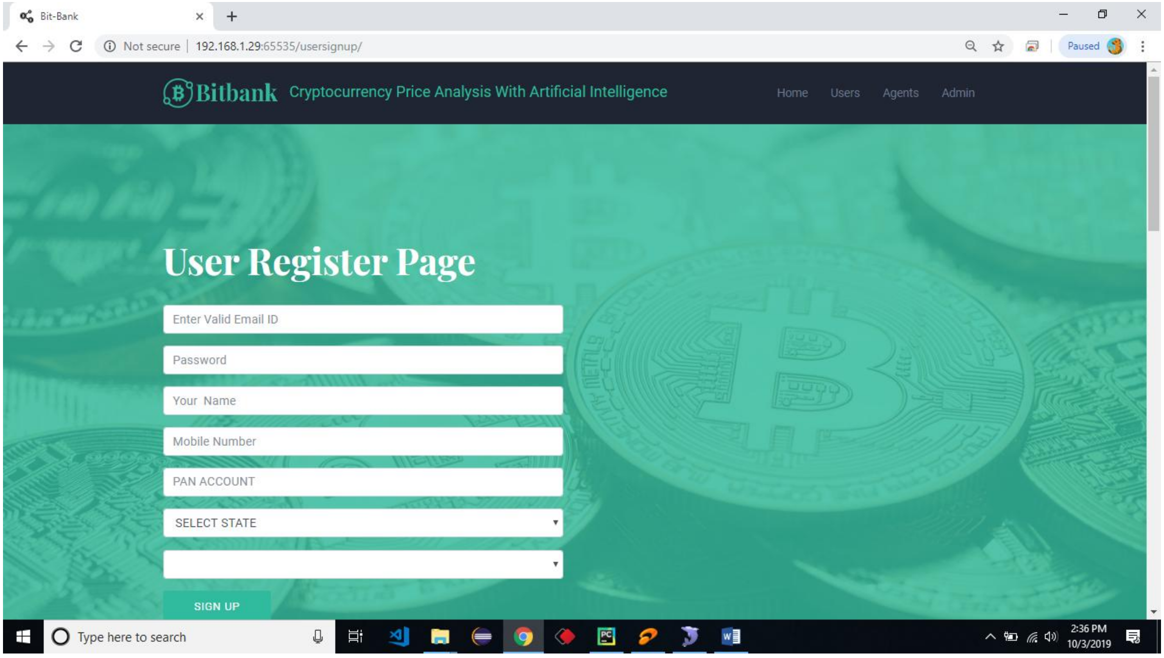

Figure 5.2 : User registration page of a Crypto Currency Price Analysis Using AI.

5.3 USER LOGIN PAGE :

In below screen, User can log into their account using logging credentials. Once the account is activated by the admin.

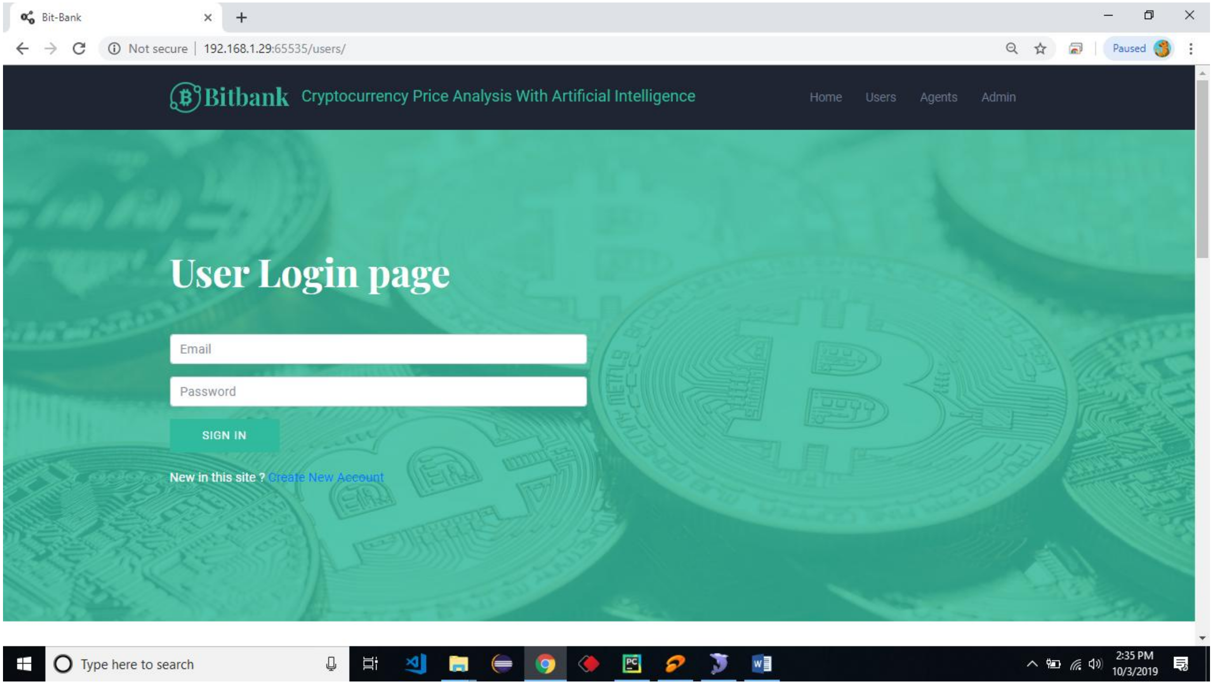

Figure 5.3 : User login page of a Crypto Currency Price Analysis Using AI.

5.4 AGENT REGISTRATION PAGE :

In below screen, A new Agent can create trading account by clicking register option, after filling required details. Account will be created on wating for the activation by the admin.

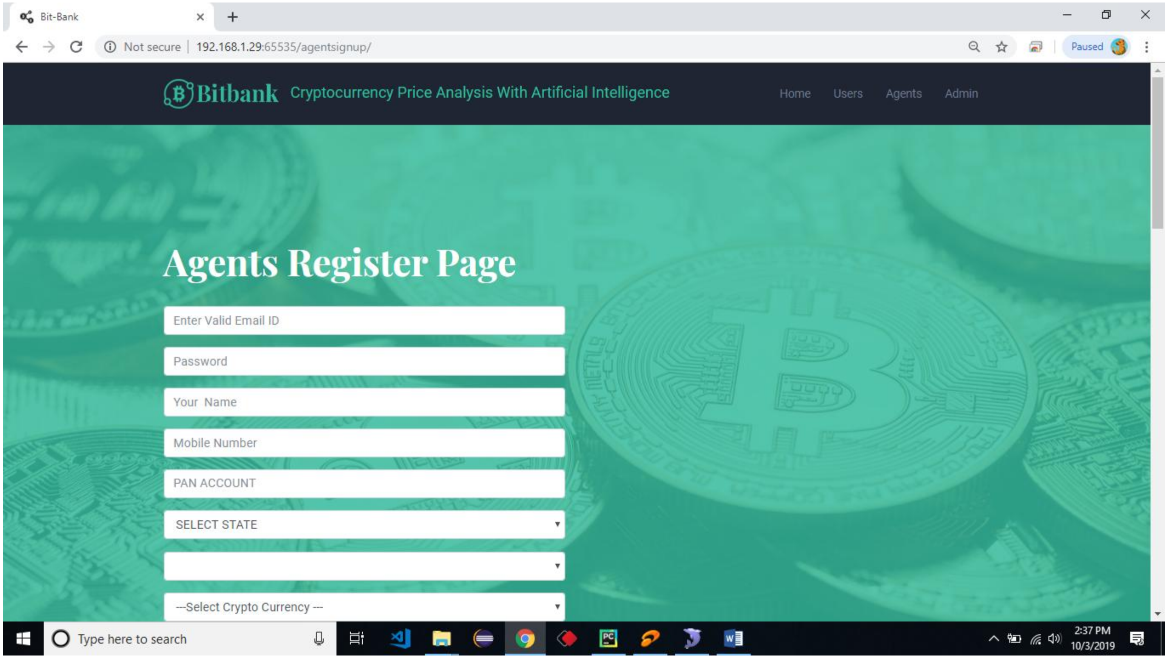

Figure 5.4: Agent Registration Page of a Crypto Currency Price Analysis Using AI.

5.5 AGENT LOGIN PAGE :

In below screen, Agent can log into their account using logging credentials. Once the account is activated by the admin.

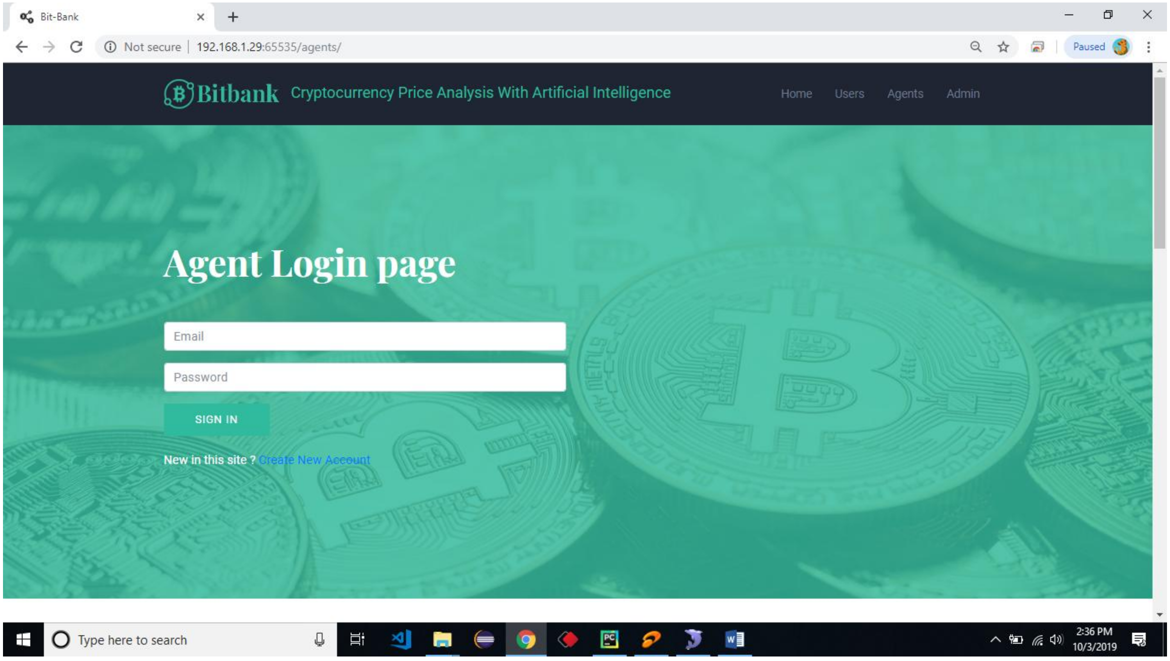

Figure 5.5: Agent Login Page of a Crypto Currency Price Analysis using AI.

5.6 ADMIN LOGIN PAGE :

In below screen, Admin can log into Administration account using logging credentials.

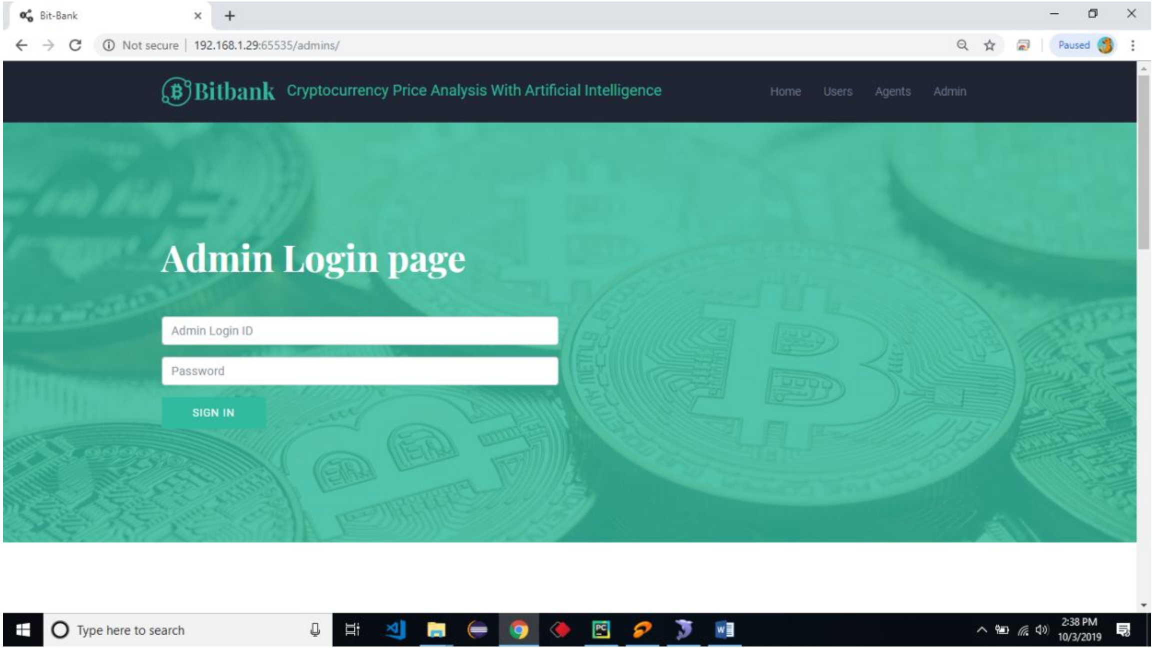

Figure 5.6 : Admin Login Page of a Crypto Currency Price Analysis using AI.

5.7 ACTIVATION OF USER APPLICATIONS :

In below screen, It shows the list of users registered into Crypt Currency Digital trading.

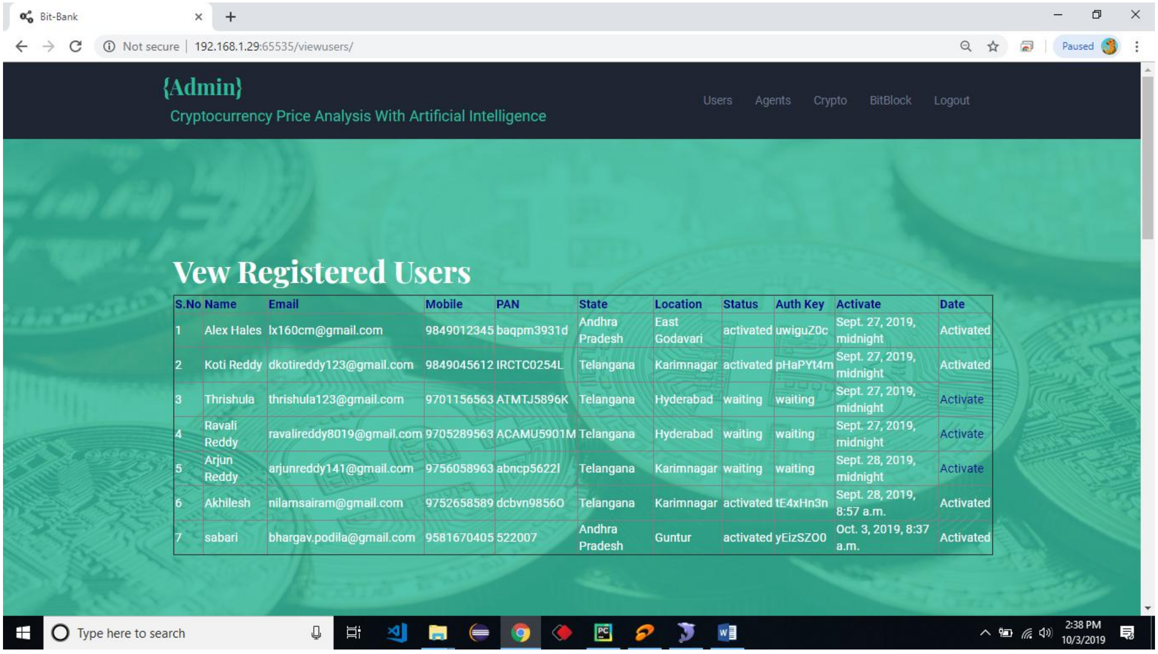

Figure 5.7: List of the Users Registered Into website of a Crypto Currency Price Analysis Using AI.

5.8 ACTIVATION OF AGENTS :

In below screen, It shows the list of Agents registered into Crypt Currency Digital trading.

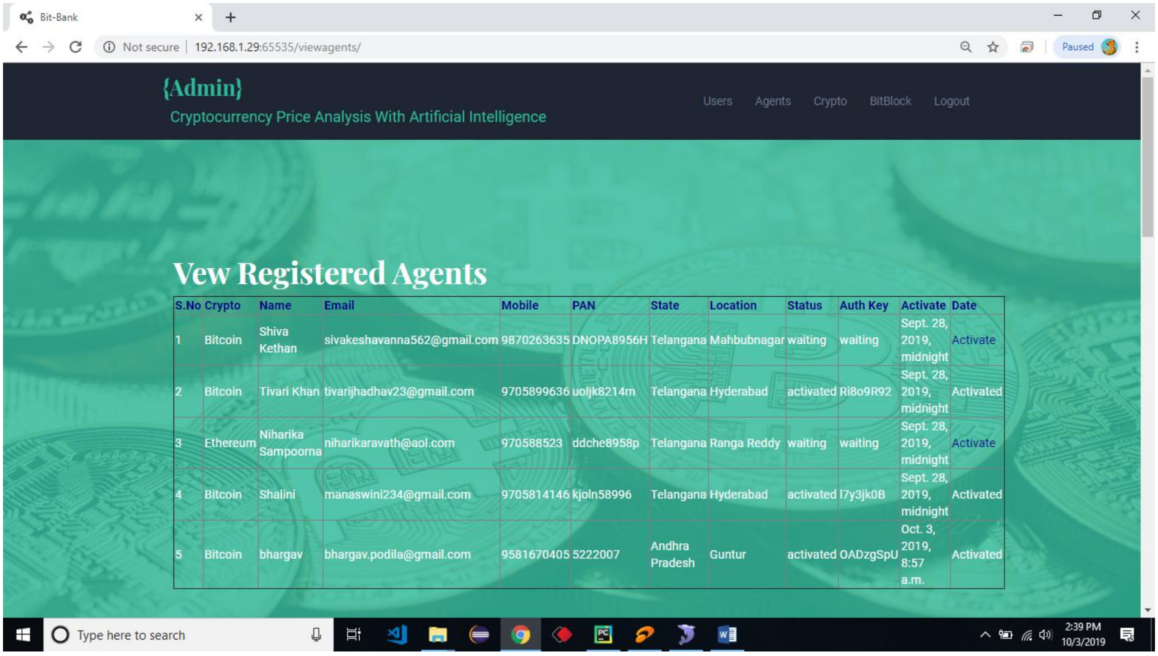

Figure 5.8 : List of the Users Registered Into website of a Crypto Currency Price Analysis Using AI.

5.9 PRICE UPDATION :

In below screen, Admin can monitor the current market prices and updates the prices of the crypto currencies like bitcoin, etherum, and ripple. The changed price will be visible to the users and agents.

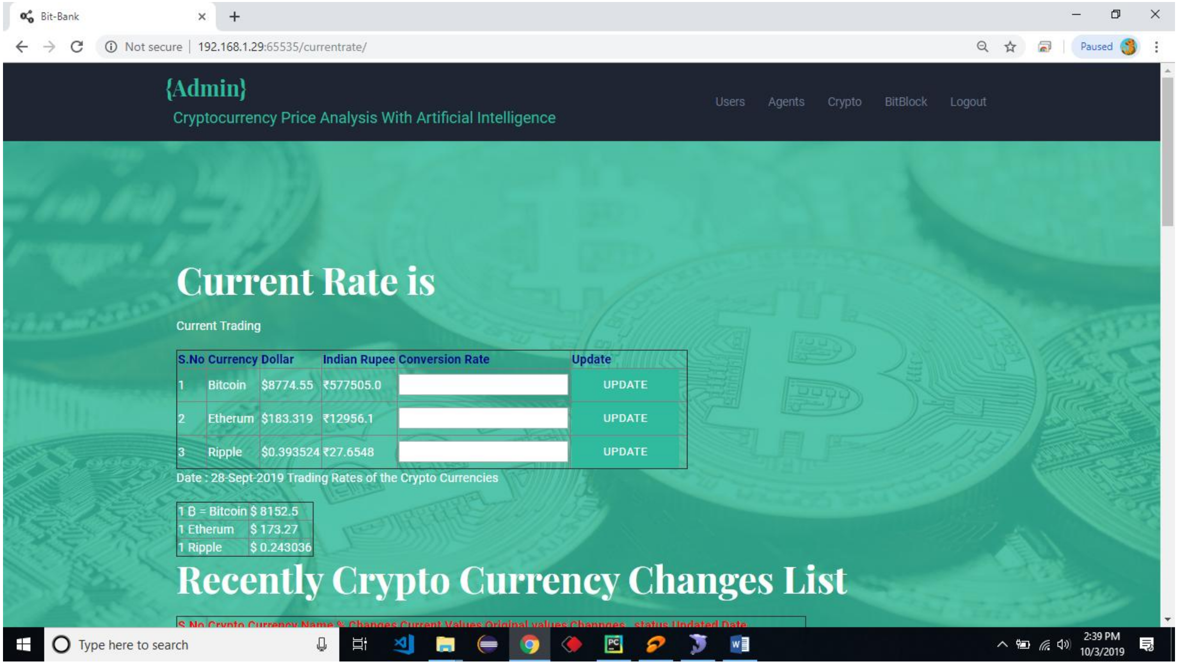

Figure 5.9 : Admin can Update the Crypto Currency Prices as the Market Fluctuates.

5.10 UPDATE HISTORY OF PRICES :

In below screen, The updated prices of the currencies history will be shown.

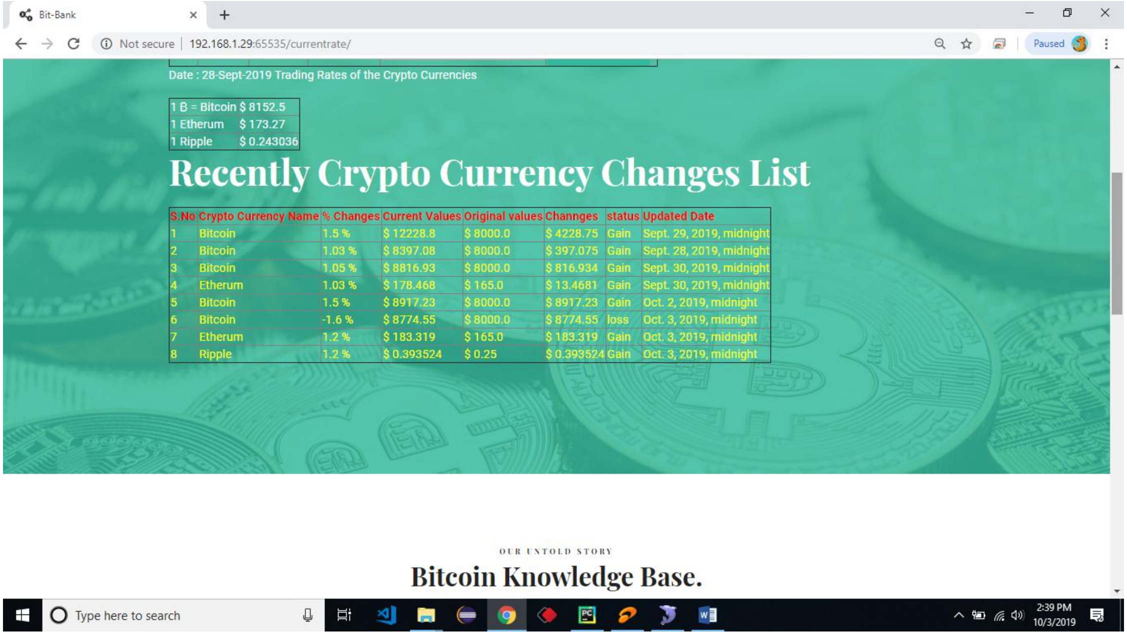

Figure 5.10: Updated prices history of a Crypto Currency Price Analysis using AI.

5.11 BLOCK CHAIN LEDGER BALANCE :

In below screen, Admin can monitor the transaction details of the agent. Can monitor how much money paid by the agent and how much ledger amount is still agent had in his/her account.

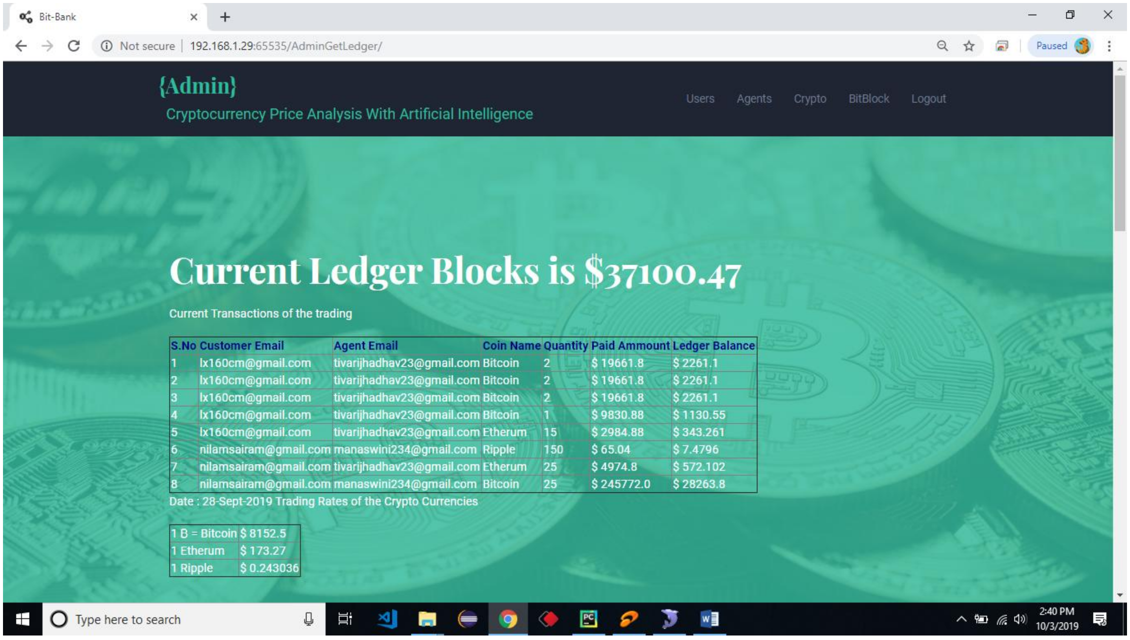

Figure 5.11 : Admin can track the Transaction details of a agent in Crypto Currency Price Analysis Using AI.

5.12 AGENT BUYING CRYPTO :

In below screen, Agent buying crypto currency based on price and the amount agent had in his/her account. Agent can select which currency and quantity of that particular currency. Once enter all the required details then transaction will be successful.

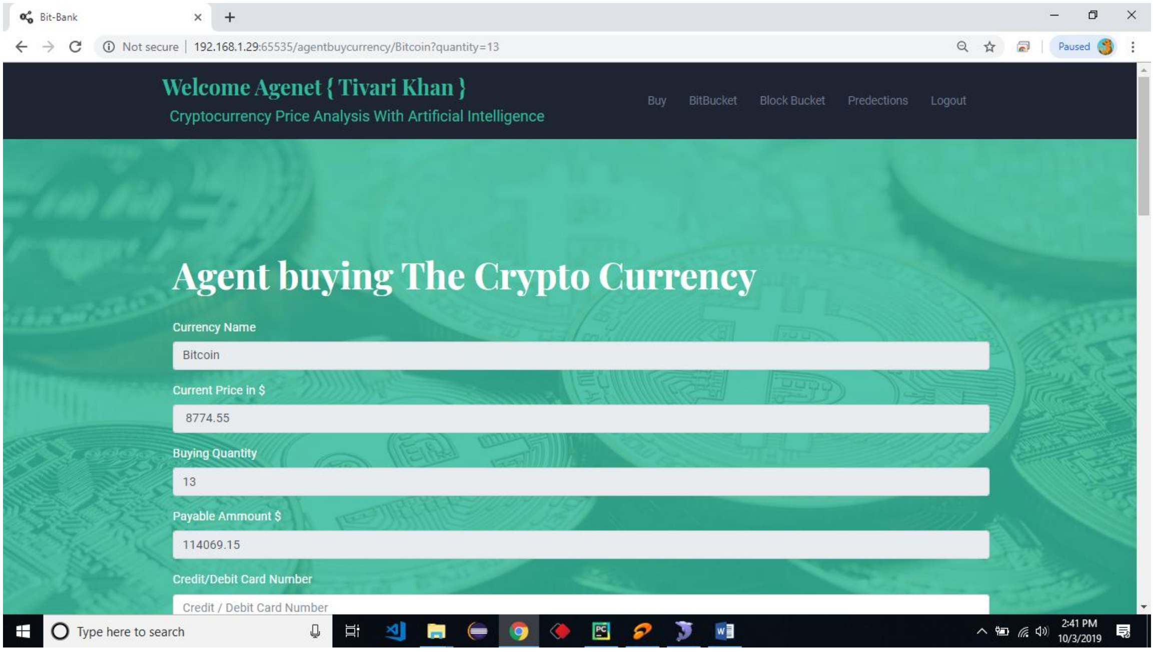

Figure 5.12 : Agent buying the Crypto Currency in Crypto Currency Price Analysis using AI.

5.13 TRANSACTION HISTORY :

In below screen, The complete transaction details will be shown i.e, either agent or users transactions they made.

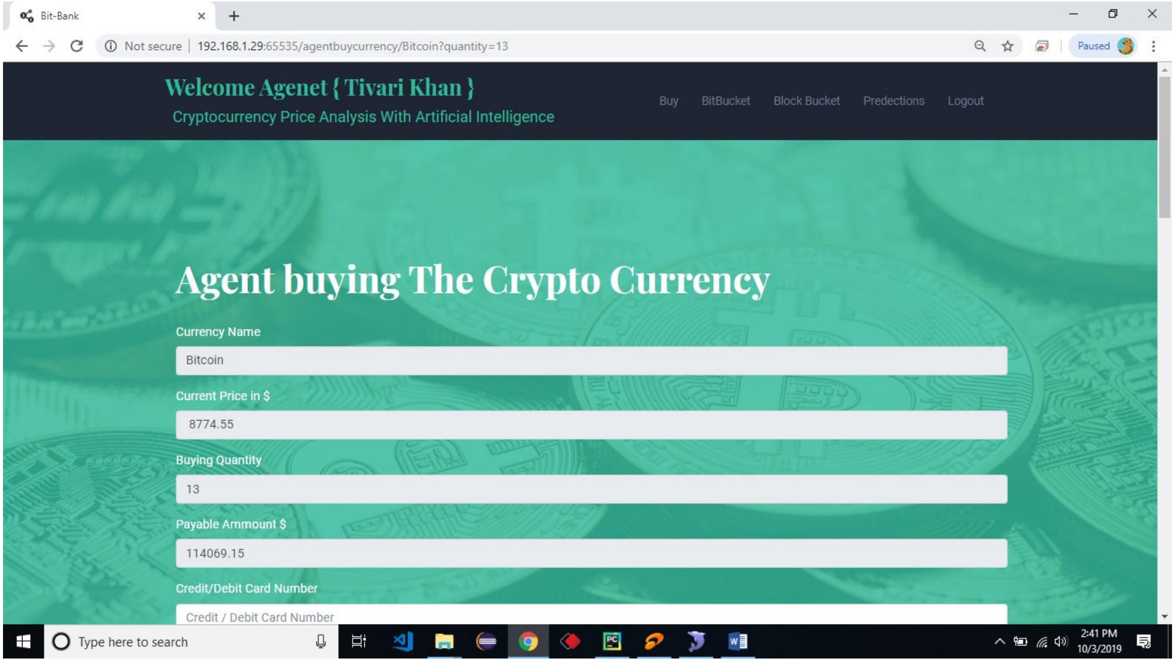

Figure 5.13 : Transactions history of an Agent/User of Crypto Currency Price Analysis using AI.

5.14 : DATASETS FOR PREDICTION :

In below screen, datasets will be available for predictions. These predictions will be helpful for clear understanding of market fluctuations and helps to make smarter decisions in investment.

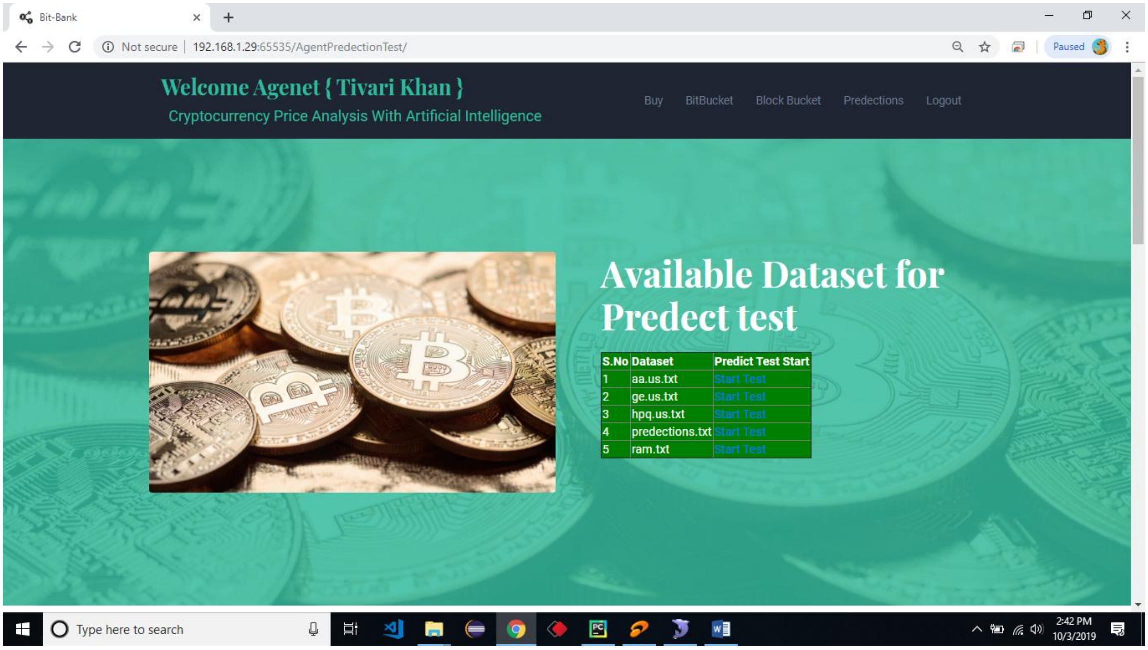

Figure 5.14 : Datasets used for Predictions in Crypto Currency Price Analysis using AI.

5.15 PREDICTION GRAPH :

In below screens, The datasets are used to understand the market fluctuations. And these accuracy graphs ensures that the algorithms predicts with almost similar accuracy of market fluctuations, which helps to predict better future price trends.

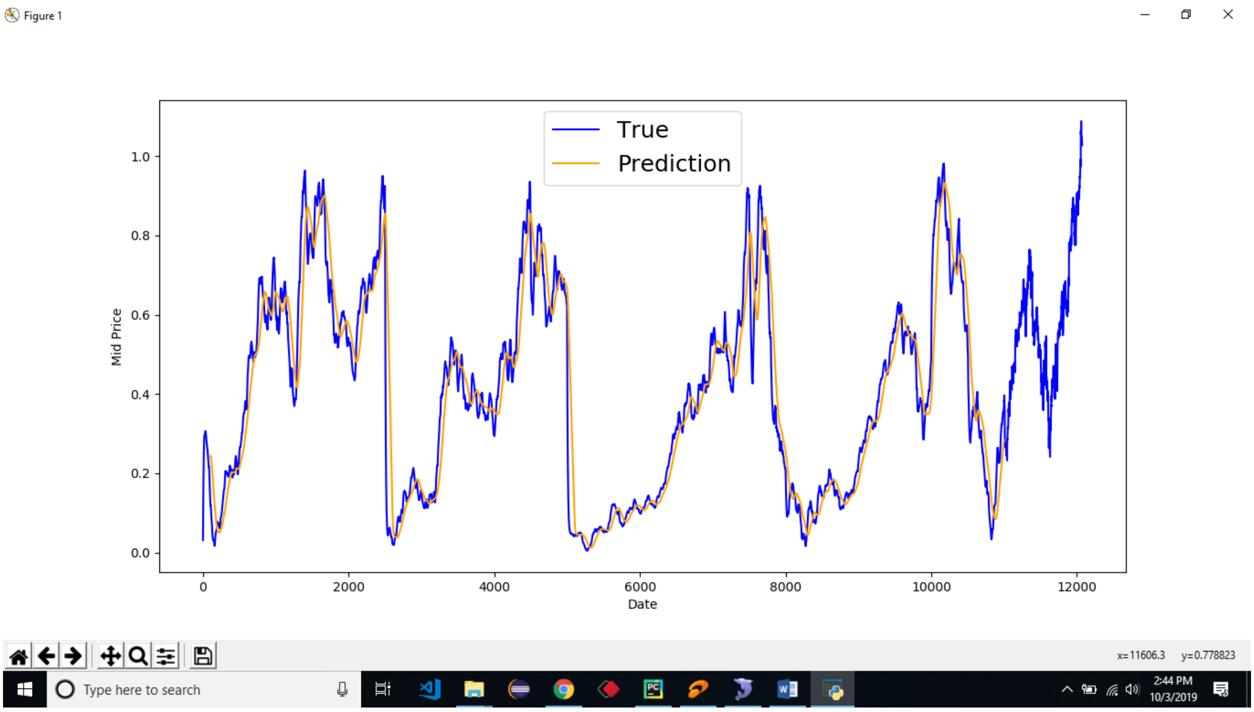

Figure 5.15 : Dataset Accuracy graph of a Crypto Currency Price Analysis using AI.

6. VALIDATION

The validation of this project primarily relies on extensive testing and well-defined test cases to ensure the accuracy and effectiveness of the inappropriate content detection system. The testing process involves multiple stages, including dataset validation, model performance evaluation, and real-world testing. By implementing a structured validation approach, we can ensure that the system consistently delivers high accuracy in detecting inappropriate content while minimizing false positives and false negatives

6.1 INTRODUCTION

The development of a cryptocurrency price prediction system leveraging Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks necessitates a comprehensive and meticulous validation process to ensure the model's robustness and reliability. This structured approach encompasses data partitioning, model evaluation, performance benchmarking, and real-world testing, collectively enhancing the system's predictive accuracy and generalization capabilities.

Data Partitioning

Initially, the historical cryptocurrency price data is systematically divided into training and testing sets, commonly employing an 80-20 split. The training set serves to instruct the deep learning model, while the testing set is reserved for assessing its ability to generalize to unseen data. To further bolster the model's robustness, K-fold cross-validation is implemented, wherein the dataset is partitioned into 'K' subsets. The model is iteratively trained on 'K-1' subsets and validated on the remaining subset, ensuring comprehensive evaluation across multiple data partitions and mitigating the risk of overfitting.

Model Evaluation

The system's predictive performance is quantified using key metrics, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics provide insights into the average magnitude of prediction errors, the square root of the average squared differences between predicted and actual values, and the average absolute percentage error, respectively. Additionally, a confusion matrix is utilized to delineate correct versus incorrect classifications, offering a granular view of the model's performance and guiding refinements to enhance accuracy.

Performance Benchmarking

To substantiate the efficacy of the proposed RNN and LSTM-based models, their performance is benchmarked against alternative models, such as Gated Recurrent Units (GRU) and Bidirectional LSTMs (Bi-LSTM). Comparative analyses have demonstrated that Bi-LSTM models often exhibit superior accuracy in predicting cryptocurrency prices, attributed to their capability to process information in both forward and backward directions, thereby capturing intricate temporal patterns more effectively.

Real-World Deployment Testing

Subsequent to validation and benchmarking, the model undergoes real-world deployment testing to simulate live trading environments. This phase involves processing real-time data feeds to evaluate the model's responsiveness and accuracy in dynamic market conditions. Continuous monitoring and iterative improvements are implemented based on performance outcomes, ensuring the system's adaptability and sustained efficacy in predicting cryptocurrency price movements.

By adhering to this structured validation process, the proposed system is poised to deliver reliable, scalable, and high-accuracy predictions, thereby serving as a valuable tool for investors and traders navigating the volatile cryptocurrency market.

TABLE 6.2 : TEST CASES

Test cases for different modules to test, validate and verify the AI modules.

S.no	Test Case	Excepted Result	Result	Remarks(IF Fails)
1	User REGISTERED	If user registration successfully.	Pass	If user is not registered.
2	Agent REGISTERED	If agent registration successfully.	Pass	If agent is not registered.
3	ADMIN	user rights will be accepted here.	Pass	If user are not registered.
4	ADMIN	agent rights will be accepted here.	Pass	If agent are not registered.
5	user LOGIN	If user name and password is correct then it will getting valid page.	Pass	If user name or password is not correct.
6	agent LOGIN	If agent name and password is correct then it will getting valid page.	Pass	If agent name or password is not correct.
7	Agent buying crypto currency from admin	If agent is correct then it will getting valid page.	Pass	If sale crypto currencies are not available .
8	User buying crypto currency from agent	If user is correct then it will getting valid page	Pass	If sale crypto currencies are not available

6.2.1 DESCRIPTION

The test cases for the cryptocurrency price analysis project cover essential functionalities to ensure smooth platform operations. User registration verifies that users can sign up successfully, storing their information in the database without errors. Similarly, agent registration ensures agents can join the platform seamlessly. Admin user rights management confirms that administrators have proper permissions to manage users, agents, and cryptocurrency data, while admin agent rights management validates the assignment and management of agent permissions. User login and agent login test cases check if users and agents can access their respective dashboards with valid credentials, identifying any issues with incorrect login details or system errors. The agent buying cryptocurrency from admin test ensures that agents can purchase available cryptocurrencies from the admin without any problems, and the user buying cryptocurrency from agent test verifies that users can buy from agents when the cryptocurrency stock is sufficient. These test cases collectively validate user roles, transactions, and access rights, ensuring a robust and efficient system.

7. CONCLUSION AND FUTURE SCOPE

In conclusion, the project has successfully achieved its objectives, showcasing significant progress and outcomes. The implementation and execution phases were meticulously planned and executed, leading to substantial improvements and insights. Looking ahead, the future aspects of the project hold immense potential. Future developments will focus on expanding the scope, integrating new technologies, and enhancing sustainability. These advancements will not only strengthen the existing framework but also open new avenues for growth and innovation, ensuring the project remains relevant and impactful in the long term. This strategic approach will drive continuous improvement and success.

7.1 PROJECT CONCLUSION

Cryptocurrency, such as Bitcoin, has established itself as the leading role of decentralisation. There are a large number of cryptocurrencies sprang up after Bitcoin such as Ethereum and Ripple. Because of the significant uncertainty in its prices, many people hold them as a means of speculation. Therefore, it is critically important to understand the internal features and predictability of those cryptocurrencies. In this project, we use two distinct artificial intelligence frameworks, namely, fully-connected Recurrent Neural Network (RNN) and Long-Short-Term-Memory (LSTM) to analyse and predict the price dynamics of Bitcoin, Etherum, and Ripple. We showed that the RNN and LSTM models are comparable and both reasonably well enough in price prediction, although the internal structures are different. Then we further analyse the influence of historical memory on model prediction. We find that RNN tends to rely more on short-term history while LSTM tends to rely more on long-term dynamics, which indicate the efficiency of LSTM to utilise useful information hidden in historical memory is stronger than RNN. However, given enough historical information This prject provides a unique demonstration that Cryptocurrency market price is predictable. However, the explanation of the predictability could vary depending on the nature of the involved machine-learning model.

7.2 FUTURE ASPECTS

Advancements in cryptocurrency price prediction using Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) models are poised to benefit from integrating alternative data sources, such as social media sentiment and macroeconomic indicators, to enhance predictive accuracy. For instance, incorporating sentiment analysis from platforms like Sina-Weibo has been shown to improve price trend predictions. Developing hybrid architectures that combine RNNs and LSTMs with other neural network models, like Convolutional Neural Networks can further improve feature extraction capabilities. Additionally, implementing real-time prediction systems can facilitate dynamic trading strategies, potentially increasing profitability. Exploring advanced sentiment analysis techniques also contribute to more accurate and robust cryptocurrency price forecasting. Furthermore, addressing the inherent volatility of cryptocurrency markets remains a significant challenge, necessitating continuous refinement of these models to adapt to rapidly changing market conditions.

These are the future aspects : Regulatory Insights, Financial Advisory Services, Investment and Trading, Risk Management in Digital Currency trading, Market Analysis, Real-Time Prediction and Automation, Integration with Blockchain Analytics, Collaboration and Open-Source Contribution.

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] Greaves, A., & Au, B. (2015). Using the bitcoin transaction graph to predict the price of bitcoin. No Data.

[2] Hayes, A. S. (2017). Cryptocurrency value formation: An empirical study leading to a cost of production model for valuing bitcoin. Telematics and Informatics, 34(7), 1308-1321.

[3] Shah, D., & Zhang, K. (2014, September). Bayesian regression and Bitcoin. In Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on (pp. 409-414). IEEE.

[4] Indra N I, Yassin I M, Zabidi A, Rizman Z I. Non-linear autoregressive with exogenous input (mrx) bitcoin price prediction model using so-optimized parameters and moving average technical indicators. J. Fundam. Appl. Sci., 2017, 9(3S), 791-808`

[5] Adebiyi AA, Ayo C K, Adebiyi MO, Otokiti SO. Stock price prediction using a neural network with hybridized market indicators. Journal of Emerging Trends in Computing and Information Sciences, 2012, 3(1):1-9

[6] Adebiyi AA, Ayo C K, Adebiyi MO, Otokiti SO. Stock price prediction using a neural network with hybridized market indicators. Journal of Emerging Trends in Computing and Information Sciences, 2012, 3(1):1-9

[7] Ariyo AA, Adewumi AO, Ayo CK. Stock price prediction using the ARIMA model. In UKSim-AMSS 16th IEEE International Conference on Computer Modelling and Simulation (UKSim), 2014, pp. 106-112

[8] Ron, D., & Shamir, A. (2013, April). Quantitative analysis of the full bitcoin transaction graph. In International Conference on Financial Cryptography and Data Security (pp. 6-24). Springer, Berlin, Heidelberg.

8.2 GITHUB LINK

https://github.com/Vinodkumar0113/crypto_major