CRYPTOCURRENCY PRICE ANALYSIS WITH ARTIFICIAL INTELLIGENCE

A project report submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, ANANTAPUR

In partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

IV B. Tech II Semester

M. HARIKA SRI SAI	20KH1A0564
B. DURGA SUJITHA	20KH1A0508
L. BHARGAVI	20KH1A0555
D. MEGHANA	20KH1A0528
B. KEERTHI BHAVANI	20KH1A0516

Under the Esteemed Guidance of

Mr. P.V. MADHUSUDHAN, M. Tech.,(Ph.D)
ASSISTANT PROFESSOR

Department of Computer Science and Engineering

Sri Venkateswara College of Engineering
(Affiliated to JNTUA, Anantapuramu and Approved by AICTE, New Delhi)
Balaji Nagar, Kadapa-516003.

(2020-2024)

CRYPTOCURRENCY PRICE ANALYSIS WITH ARTIFICIAL INTELLIGENCE

Project report submitted in partial fulfillment of the requirement for the award of the Degree of **B. Tech in Computer Science and Engineering**

By

M. HARIKA SRI SAI	20KH1A0564
B. DURGA SUJITHA	20KH1A0508
L. BHARGAVI	20KH1A0555
D. MEGHANA	20KH1A0528
B. KEERTHI BHAVANI	20KH1A0516

Under the Esteemed Guidance of

Mr. P.V. MADHUSUDHAN, **M. Tech.,(Ph.D)**

ASSISTANT PROFESSOR

Department of Computer Science and Engineering

SV COLLEGE OF ENGINEERING

Approved by AICTE, New Delhi, Affiliated to JNTUA, Anantapuram

Website: www.svck.edu.in

Sri Venkateswara College of Engineering

(Affiliated to JNTUA, Anantapuramu and Approved by AICTE, New Delhi)

Balaji Nagar, Kadapa-516003.

Department of Computer Science and Engineering

Certificate

This is to certify that the project work entitled

CRYPTOCURRENCY PRICE ANALYSIS WITH ARTIFICIAL INTELLIGENCE

is the bonafide work done by

M. HARIKA SRI SAI	20KH1A0564
B. DURGA SUJITHA	20KH1A0508
L. BHARGAVI	20KH1A0555
D. MEGHANA	20KH1A0528
B. KEERTHI BHAVANI	20KH1A0516

In the Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Kadapa is affiliated to JNTUA-Anantapur in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering during 2020-2024.

This work has been carried out under my guidance and supervision.

The results embodied in this Project report have not been submitted in any University or Organization for the award of any degree or diploma.

INTERNAL GUIDE
Mr. P. V. MADHUSUDHAN, M. Tech.,(Ph.D)
Assistant Professor

HEAD OF THE DEPARTMENT
Mr. G. SREENIVASULU, M. Tech.,(Ph.D)
Assistant Professor & HoD

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We are extremely thankful to our beloved chairman and establisher **Sri A. Gangi Reddy** for providing necessary infrastructure and resources for the accomplishment of our project at **Sri Venkateswara College of Engineering, Kadapa.**

We are highly indebted to **Dr. R. Veera Sudarsana Reddy, Principal** of **Sri Venkateswara College of Engineering, Kadapa**, for his support during the tenure of the project.

We are very much obliged to our beloved **Mr. G. Sreenivasulu, Head of the Department of Computer Science & Engineering**, Sri Venkateswara College of Engineering for providing the opportunity to undertake this project and encouragement in completion of this project.

We here by wish to express deep sense of gratitude to **Mr. P.V. Madhusudhan**, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering for the esteemed guidance, moral support and invaluable advice provided by him for the success of the project.

We are also thankful to all the staff members of Computer Science and Engineering department who have co-operated in making our project a success. We would like to thank all our parents and friends who extended their help, encouragement and moral support either directly or indirectly in our project work.

Thanks for Your Valuable Guidance and kind support.

DECLARATION

We hereby declare that project report entitled **CRYPTOCURRENCY PRICE ANALYSIS WITH ARTIFICIAL INTELLIGENCE** is a genuine project work carried out by us, in B. Tech (Computer Science and Engineering) degree course of **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR** and has not been submitted to any other courses or University for award of any degree by us.

Signature of the Student

1. M. Harika Sri Sai
2. B. Durga Sujitha
3. L. Bhargavi
4. D. Meghana
5. B. Keerthi Bhavani

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 study, we use advanced artificial intelligence frameworks of fully connected Artificial Neural Network (ANN) and Long Short-Term Memory (LSTM) Recurrent Neural Network to analyze the price dynamics of Bitcoin, Ethereum, and Ripple. We find that ANN tends to rely more on long-term history while LSTM tends to rely more on short-term dynamics, which indicate the efficiency of LSTM to utilize useful information hidden in historical memory is stronger than ANN. However, given enough historical information ANN can achieve a similar accuracy, compared with LSTM. This study 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.

LIST OF FIGURES

LIST OF PHOTOGRAPHS

TABLE OF CONTENTS

1. INTRODUCTION

1.1. ABOUT THE PROJECT

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 study, we use advanced artificial intelligence frameworks of fully connected Artificial Neural Network (ANN) and Long Short-Term Memory (LSTM) Recurrent Neural Network to analyse the price dynamics of Bitcoin, Ethereum, and Ripple. We find that ANN tends to rely more on long-term history while LSTM tends to rely more on short-term dynamics, which indicate the efficiency of LSTM to utilize useful information hidden in historical memory is stronger than ANN. However, given enough historical information ANN can achieve a similar accuracy, compared with LSTM. This study 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. Cryptocurrency is the peer-to-peer digital money 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. Significant efforts have been made to analyse and predict the trends of traditional financial markets especially the stock market however, predicting cryptocurrencies market prices is still at an early stage. Compared to these stock price prediction models, 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. In this study, we hypothesise that time series of cryptocurrencies exhibits a clear internal memory, which could be used to help the memory-based time series model to works more appropriately if the length of internal memory could be quantified. We aim to use two artificial intelligence modelling frameworks to understand and predict the most popular cryptocurrencies price dynamics, including Bitcoin, Ethereum, and Ripple.

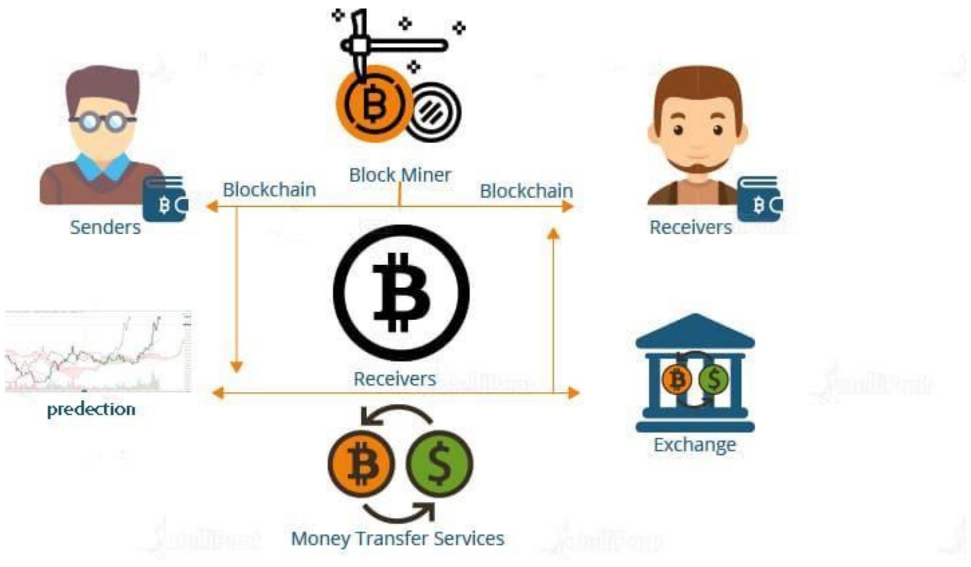


Fig.1.1: System Architecture

1.2 MODULES

1.2.1. User

Registration of User details next admin activated by the user. go to user page and login. User view some fields [start trading, bit bucket, prediction, logout] now open start trending page contains sale available cryptocurrencies and user buy the currency. User views the transaction history details and user view the prediction for available datasets.

1.2.2. Agent

First registration of the agent page. next activated by the admin. Login agent page. agent view some fields [buy, bitbucket, block bucket, prediction, logout]. here three types of digital currencies are there [bitcoins, ripple, Ethereum] agent can buy digital currency (any one or all three). Agent buying the crypto currency. agent transaction history also available.

1.2.3. Admin

The aim of admin is to approve the users and agents. Admin contains some fields [user, agents, crypto, bit block]. Admin view user and agent registered details and in crypto field user update the currency rate and view the recently crypto currency changes list. Admin view the current transaction details. 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.

1.2.4. Artificial intelligence

The application of advanced digital, smart technologies, robotic systems, new materials and design techniques, creation of large data processing systems, computer-aided learning and artificial intelligence (AI) are relevant for various branches of science and technology, including manned space programs. Some technology concepts and pilot systems based on the AI (3-D computer vision, automated systems for planning and evaluating the activities of cosmonauts, inquiry and communications system) were developed in the industry over several decades.

2. LITERATURE SURVEY

A literature survey or literature review is the study of references and old algorithms that we have read for designing the proposed methods. It also helps in reporting summarization of all the old references papers, and their drawbacks. The detailed literature survey for the project helps in comparing and contrasting various methods, algorithms in various ways that have implemented in the research.

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

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 analyse the network topology of the Bitcoin network, limited research has been performed to analyse 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] 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.

This paper aims to identify the likely determinants for cryptocurrency value formation, including for that of bitcoin. Due to Bitcoin's growing popular appeal and merchant acceptance, it has become increasingly important to try to understand the factors that influence its value formation. Presently, the value of all bitcoins in existence represents approximately $7 billion, and more than $60 million of notional value changes hands each day. Having grown rapidly over the past few years, there is now a developing but vibrant marketplace for bitcoin, and a recognition of digital currencies as an emerging asset class. Not only is there a listed and over-the-counter market for bitcoin and other digital currencies, but also an emergent derivatives market. As such, the ability to value bitcoin and related cryptocurrencies is becoming critical to its establishment as a legitimate financial asset.

Using cross-sectional empirical data examining 66 of the most widely used cryptocurrencies, a regression model was estimated that points to three main drivers of cryptocurrency value: the level of competition in the network of producers, the rate of unit production, and the difficulty of algorithm used to "mine" for the cryptocurrency. These amount to relative differences in the cost of production of one digital currency over another at the margin, pointing to differences in relative cost of production – electricity goes in, cryptocurrency comes out. Using that as a starting point, a no-arbitrage situation is established for Bitcoin-like cryptocurrencies followed by the formalization of a cost of production model to determine the fair value of a bitcoin.

[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.

Abstract—In this paper, we discuss the method of Bayesian regression and its efficacy for predicting price variation of Bitcoin, a recently popularized virtual, cryptographic currency. Bayesian regression refers to utilizing empirical data as proxy to perform Bayesian inference. We utilize Bayesian regression for the so-called "latent source model". The Bayesian regression for "latent source model" was introduced and discussed by Chen, Nikolov and Shah [1] and Bresler, Chen and Shah [2] for the purpose of binary classification. They established theoretical as well as empirical efficacy of the method for the setting of binary classification. In this paper, instead we utilize it for predicting real-valued quantity, the price of Bitcoin. Based on this price prediction method, we devise a simple strategy for trading Bitcoin.

[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`

Indra N I, Yassin I M, Zabidi A, and Rizman Z I (2017) proposed a non-linear autoregressive with exogenous input (MRX) model for Bitcoin price prediction. This model incorporates SO-optimized parameters along with moving average technical indicators. In their research, Indra N I et al. aimed to enhance Bitcoin price prediction by employing a sophisticated modelling approach.

By integrating SO-optimized parameters and moving average technical indicators into the MRX framework, they sought to capture the complex dynamics of Bitcoin price movements. This study contributes to the field of cryptocurrency forecasting by exploring innovative methodologies to improve prediction accuracy and inform investment decisions.

[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

Stock prediction with data mining techniques is one of the most important issues in finance being investigated by researchers across the globe. Data mining techniques can be used extensively in the financial markets to help investors make qualitative decision. One of the techniques is artificial neural network (ANN). However, in the application of ANN for predicting the financial market the use of technical analysis variables for stock prediction is predominant. In this paper, we present a hybridized approach which combines the use of the variables of technical and fundamental analysis of stock market indicators for prediction of future price of stock in order to improve on the existing approaches. The hybridized approach was tested with published stock data and the results obtained showed remarkable improvement over the use of only technical analysis variables. Also, the prediction from hybridized approach was found satisfactorily adequate as a guide for traders and investors in making qualitative decisions.

3.SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

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. Madan et al. collected bitcoins price with the time interval of 0.5, 1and 2 hours, and combined it with the blockchain network, the underlying technology of bitcoin. Their predictive model leveraging random forests and binomial logistic regression classifiers and the precision of the model is around 55% in predicting bitcoin’s price. 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. Currently highly reliable GPS systems are used to track vehicle speeds in US. Cost-effectiveness is a concern in such a case. In our method moving vehicle video from any video camera or mobile source is utilized. The algorithms are implemented in ‘C’ language using OpenCV and Visual Studio. Later this code can be ported to a simple processor where vehicle speed can be measured. Example: a simple smart phone with average processing capacity. Our aim was to implement real- time vehicle speed detector

3.1.1 Limitations in Existing System

- 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.
- By using random forests and binomial logistic regression We cannot predict the 100% results

3.2. PROPOSED SYSTEM

Among many features of bitcoin, the most impressive one is decentralisation that it can remove the involvement of traditional financial sectors and monetary authorities effectively due to its block chain network features. In proposed system we are using ANN algorithm and LSTM easily predict the time series of crypto currency prices. we use an ANN model to predict the price of Bitcoin one day into the future using five different lengths of memory. While LSTM is intentionally designed to model the internal memory flow and its impact on future prediction, therefore, both ANN and LSTM are suitable for the crypto currencies price time series prediction

3.2.1. Features of the Proposed System

- The bitcoin has introduced the controllable anonymity scheme, and this enhances users' safety and anonymity by using this technology, for instance, we can take advantage of this property of blockchain to make identification cards, and it not only can protect our privacy but verify our identity.
- By using ANN and LSTM We can predict the future price of cryptocurrency and time series also successful We can find the 100% result.

ALGORITHMS

Artificial Neural Networks (ANN)

Artificial Neural Networks (ANNs) are computational models inspired by the structure and functioning of biological neural networks in the human brain. They consist of interconnected nodes, called neurons or units, organized in layers. ANNs are widely used in various machine learning tasks, including classification, regression, pattern recognition, and optimization. Here are the key components and concepts related to artificial neural networks:

Neurons (Nodes): Neurons are the fundamental units of an ANN. Each neuron receives input signals, processes them, and produces an output signal. In a typical feedforward neural network, neurons are organized in layers: an input layer, one or more hidden layers, and an output layer.

Weights and Biases: Connections between neurons are associated with weights, which represent the strength of the connection. Additionally, each neuron has an associated bias, which allows it to adjust the output along with the weighted sum of inputs. During training, weights and biases are adjusted to minimize the difference between the predicted and actual outputs.

Activation Functions: Activation functions introduce non-linearities to the output of neurons, enabling ANNs to learn complex relationships in the data. Common activation functions include sigmoid, tanh, ReLU (Rectified Linear Unit), and softmax (for multi-class classification).

Feedforward Propagation: In feedforward propagation, the input data is passed through the network layer by layer, with each layer performing a weighted sum of inputs followed by an activation function to produce the output.

Backpropagation: Backpropagation is an algorithm used to train ANNs by adjusting the weights and biases based on the error between the predicted and actual outputs. It involves propagating the error backward through the network and updating the parameters using gradient descent or its variants.

Loss Function: The loss function quantifies the difference between the predicted and actual outputs of the network. During training, the goal is to minimize the loss function, typically using optimization algorithms like stochastic gradient descent.

Training Data: ANNs require labeled training data to learn the underlying patterns in the data. The training process involves iteratively presenting the input data to the network, adjusting the weights and biases, and evaluating the performance until convergence.

Regularization: Regularization techniques such as dropout and L2 regularization are employed to prevent overfitting and improve the generalization performance of ANNs.

Hyperparameters: ANNs have several hyperparameters that need to be tuned, including the number of hidden layers, the number of neurons per layer, learning rate, batch size, and activation functions.

ANNs can be implemented using various libraries and frameworks, such as TensorFlow, Keras, PyTorch, and scikit-learn, making them accessible to both researchers and practitioners in the field of machine learning.

LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to overcome the limitations of traditional RNNs in learning long-term dependencies in sequential data. LSTM networks are particularly useful in machine learning tasks involving time series data, natural language processing (NLP), speech recognition, and other sequential data domains. Here's a brief overview of how:

LSTM works:

Sequential Data Representation: LSTM processes sequential data one step at a time. Each step involves inputting a new data point into the network.

Memory Cells: LSTM networks have memory cells that can maintain information over time, allowing them to capture long-term dependencies. These memory cells are equipped with gates that regulate the flow of information.

Gates: There are three types of gates in an LSTM unit:

Forget Gate: Decides what information to discard from the cell state.

Input Gate: Decides which new information to incorporate into the cell state.

Output Gate: Decides what information to output based on the current input and the past state.

Cell State: The cell state runs along the entire sequence, and its information can be modified or retrieved by the gates. This mechanism enables LSTM networks to remember or forget information selectively.

Training: LSTM networks are trained using backpropagation through time (BPTT) or similar algorithms. The parameters (weights and biases) of the network are adjusted iteratively to minimize the difference between the predicted output and the actual output.

When working with LSTMs, it's crucial to tune hyperparameters, preprocess data appropriately, and monitor training performance to achieve the best results for a given task. Additionally, understanding the principles behind LSTM's.

architecture and functioning are essential for effectively applying them in ML projects.

The advantages of LSTM (Long-Short Term Memory) are as follows:

Long-term dependencies can be captured by LSTM networks. They have a memory cell that is capable of long-term information storage.

In traditional RNNs, there is a problem of vanishing and exploding gradients when models are trained over long sequences. By using a gating mechanism that selectively recalls or forgets information, LSTM networks deal with this problem.

LSTM enables the model to capture and remember the important context, even when there is a significant time gap between relevant events in the sequence. So, where understanding context is important, LSTMS are used. e.g. machine translation.

4. SOFTWARE AND HARDWARE REQUIREMENTS

4.1 SOFTWARE SPECIFICATIONS

Operating System	:	Windows 7+
Coding Language	:	Python
Server	:	WAMP Server
Libraries Used	:	NumPy, Pandas, Matplotlib
Database	:	MYSQL

4.2 HARDWARE SPECIFICATIONS

Processor	:	I3/Intel
Hard Disk	:	128 GB
RAM	:	4 GB

5. SYSTEM DESIGN

1.2. SYSTEM ARCHITECTURE

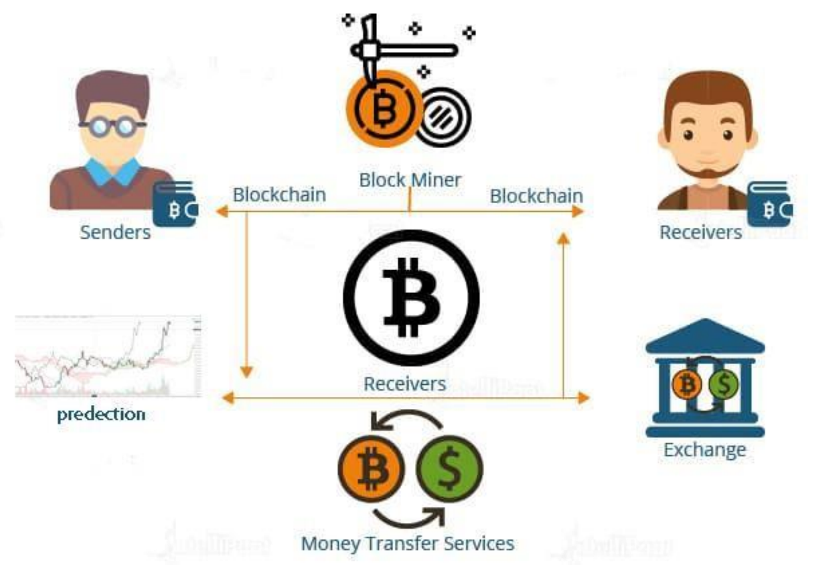


Fig 5.1: System Architecture

1.3. DATA FLOW DIAGRAMS

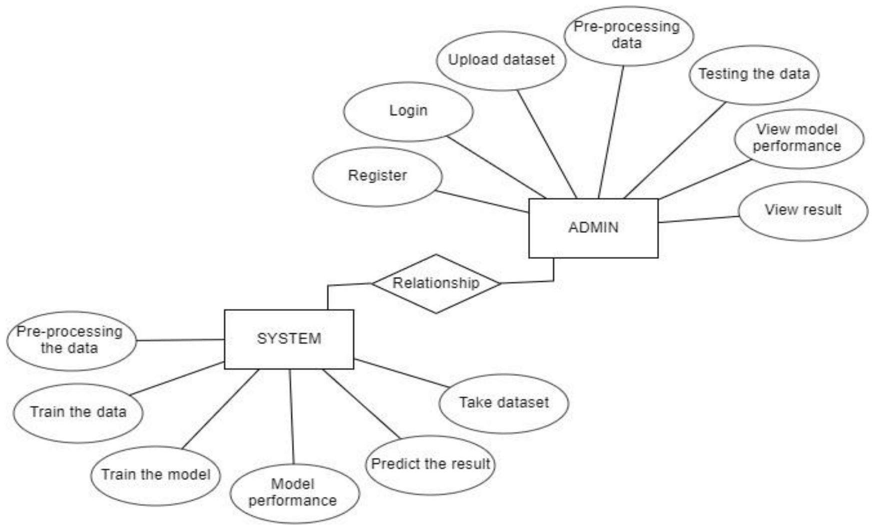


Fig.5.2: Data Flow Diagram

Dataflow Diagram works as follows:

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.

1.4. UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

5.3.1. Use Case Diagram

A use case diagram is a type of behavioral diagram created from a Use-case analysis. The purpose of use case is to present overview of the functionality provided by the system in terms of actors, their goals and any dependencies between those use cases. In the below diagram the use cases are depicted with actors and their relationships.

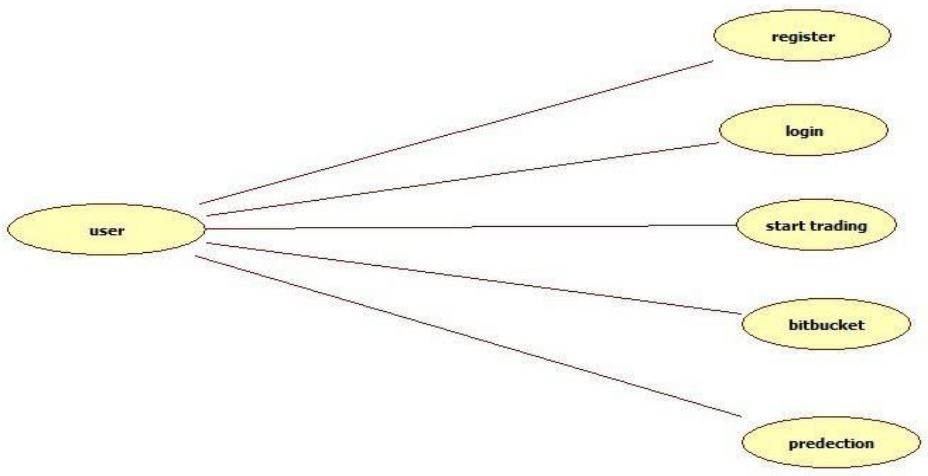


Fig.5.3: Use Case Diagram for Overall Project

5.3.2. Class Diagram

A class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. Private visibility hides information from anything outside the class partition. Public visibility allows all other classes to view the marked information. Protected visibility allows child classes to access information they inherited from a parent class.

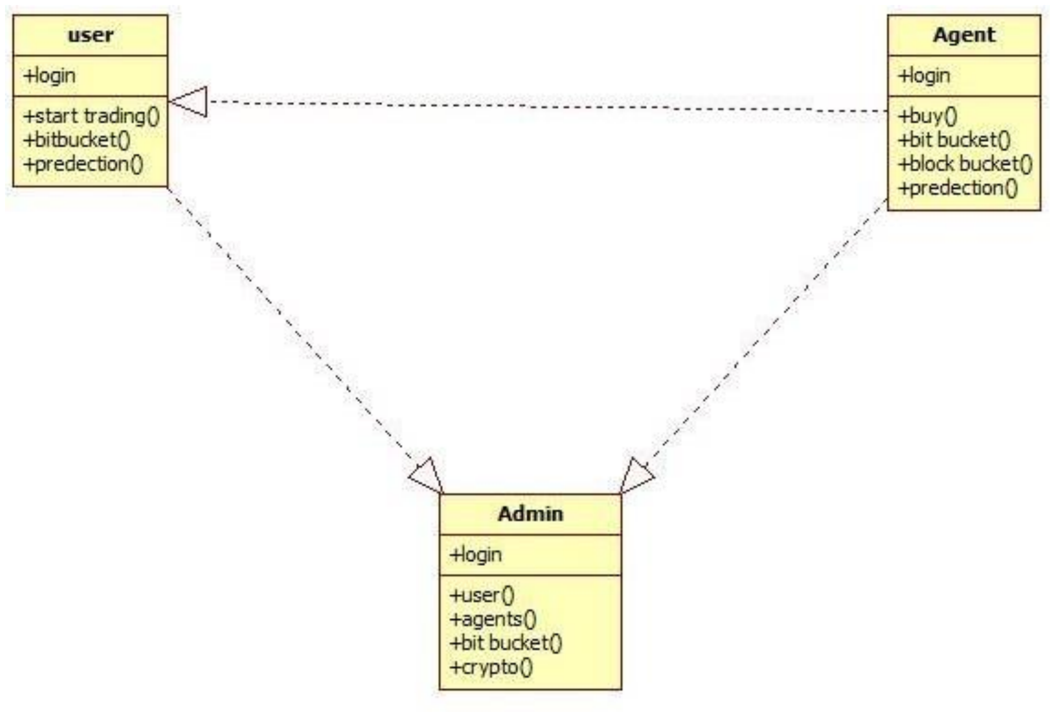


Fig.5.4: Class Diagram for Overall Project

5.3.3. Sequence Diagram

The sequence diagram describes the flow of messages being passed from object to object. Unlike the class diagram, the sequence diagram represents dynamic message passing between instances of classes rather than just a static structure of classes. In some ways, a sequence diagram is like a stack trace of object messages.

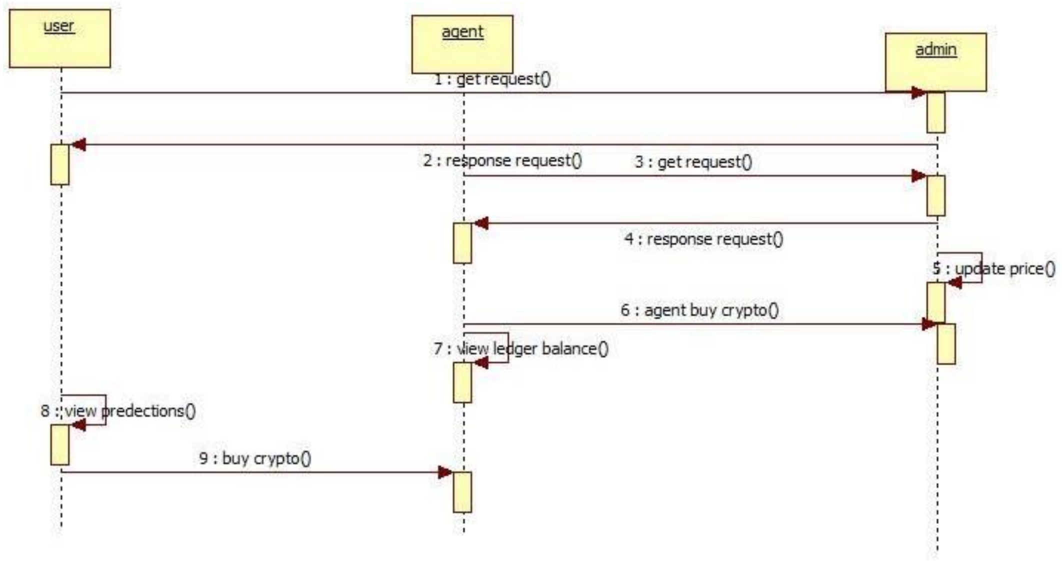


Fig.5.5: Sequence Diagram for Overall Project

5.3.4. Activity Diagram

Activity Diagram in some ways is like a flowchart with states. With the activity diagram you can follow flow of activities in your system in the order that they take place. An activity diagram illustrates the dynamic nature of a system by modelling the flow of control from activity to activity. Because an activity diagram is a special kind of state chart diagram, it uses some of the same modelling conventions.

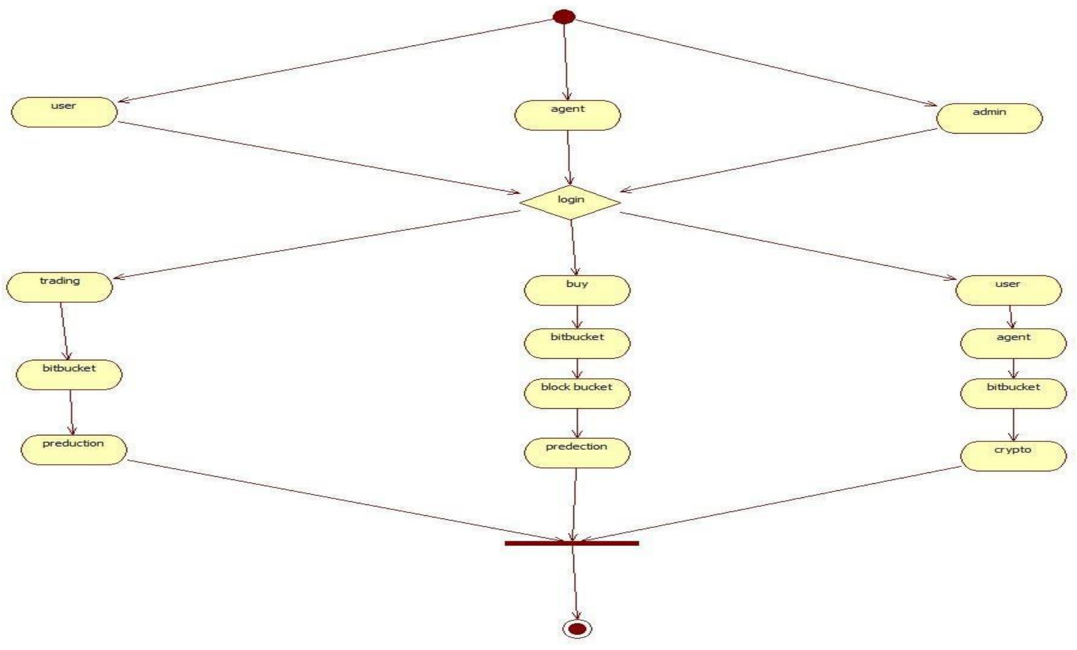


Fig.5.6: Activity Diagram for Client

6. MODULE IMPLEMENTATION

6.1. PYTHON

Below are some facts about Python. Python is currently the most widely used multi-purpose, high-level programming language.Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard libraries which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, Beautiful Soup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python:

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3.Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

4.Improved Productivity

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. These further aids the readability of the code.

8. Object-Oriented

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

9. Free and Open-Source

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages. Any doubts till now in the advantages of Python? Mention in the comment section.

6.2. HISTORY OF PYTHON

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum Wiskunde & Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So, I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

What is Machine Learning:

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data. Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects.

Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”. Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programming logic, in the problems that cSVM & NAVI BAYESot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale.

Challenges in Machines Learning:

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cSVM & NAVI BAYESot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

Applications of Machines Learning:

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cSVM & NAVI BAYESot be solved with traditional approach. Following are some real-world applications of ML –

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

6.3. LEARNING MACHINE LEARNING

Arthur Samuel coined the term "Machine Learning" in 1959 and defined it as a "Field of study that gives computers the capability to learn without being explicitly programmed".

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer is the Best Job of 2019 with a 344% growth and an average base salary of $146,085 per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So, this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on math’s as there are many common libraries available.

7.SOURCE CODE

VIEW.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)

        # 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)
        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.now()
        print("Today's date:", today)
        CurrencyUpdateModel.objects.
        create(currencyname=curr,
        conversionRate=rate,
        newCurrencyValue=changes,
  originalCurrencyValue=origin
```

```
        alDollerrate,
        changeValue=changes,profitor
        loss=currencygain,
        changedate=today)
        cryptcurrencyratemodel.object
        s.filter(currencytype=curr).upd
        ate(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.now()
        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.py

```
from django.shortcuts import render, HttpResponse,redirect
from .models import BitAgentRegisterModel,
AgentHadCrypto,AgentBuyCryptoModel
from django.contrib import messages
from admins.models import cryptcurrencyratemodel, CurrencyUpdateModel
from users.models import BlockChainLedger
from django.db.models import Sum
from django.conf import settings
import os
from users.lstmann import predictionstart
from users.algo.generatedata import GetData
# Create your views here.



def bitagentregister(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')
        crypttype = request.POST.get('cryptocurrencies')
        print("Valid Form = ", email)
        try:
        rslts = BitAgentRegisterModel.objects.create(email=email, pswd=pswd,
        username=username, mobile=mobile,
        pan=pan, state=state, location=location, cryptcurrency=crypttype)

            if rslts is None:
            print("Invalid Data ", rslts)
                messages.success(request, 'Email ID already exist, Registration Failed ')
            else:
                print("Valid Data ",
                results)messages.success(request, 'Registration
                Success')
        except:
            messages.success(request, 'Email ID already exist, Registration Failed ')
            return render(request, 'agents/agentsignup.html', {})
    else:
        print("Invalid Form Data")
        messages.success(request, 'Email ID already exist, Registration Failed ')
    return render(request, 'agents/agentsignup.html', {})
def agentlogincheck(request):
    if request.method == "POST":
        email = request.POST.get('email')
        pswd = request.POST.get('pswd')
        print("Email = ", email, ' Password = ', pswd)
        try:
            check = BitAgentRegisterModel.objects.get(email=email, pswd=pswd)
            status = check.status
            print('Status is = ', status)
            if status == "activated":
                request.session['id'] = check.id
                request.session['loggedagent'] = check.username
                request.session['email'] = check.email
                print("User id At", check.id, status)
                return render(request, 'agents/agentpage.html', {})
            else:
                messages.success(request, 'Your Account Not at activated')
                return render(request, 'users.html')
            # return render(request, 'user/userpage.html',{})
        except Exception as e:
            print('Exception is ', str(e))
            pass
        messages.success(request, 'Invalid Email id and password')
    return render(request, 'agents.html', {})
```

```
def AgentBuyCrypto(request):
    dict = cryptcurrencyratemodel.objects.all()
    dict2 = CurrencyUpdateModel.objects.all()
    return render(request, 'agents/buycurrencybyagent.html', {'objects': dict, 'objects1':
  dict2})



  def agentbuycurrency(request, currencyname):
    quntity = int(request.GET.get('quantity'))
    check = cryptcurrencyratemodel.objects.get(currencytype=currencyname)
    currentPrice = check.doller
    payableAmount = quntity * currentPrice
    print("1 Bitcoint value = ", currentPrice, " Currency is = ", currencyname, "
  Quanity = ", quntity,
          " Payable Ammount = ", payableAmount)
    dict = {
        "currentPrice": currentPrice,
        "currencyname": currencyname,
        "quntity": quntity,

"PayableAmmount": payableAmount
    }
    return render(request, 'agents/agentbuycrypto.html', dict)



  def AgentTransactions(request):
    if request.method == 'POST':
        currencyname = request.POST.get('currencyname')
        currentprice = float(request.POST.get('currentprice'))
        quantity = int(request.POST.get('quantity'))
        payableammount = float(request.POST.get('payableammount'))
        cardnumber = request.POST.get('cardnumber')
        nameoncard = request.POST.get('nameoncard')
        cardexpiry = request.POST.get('cardexpiry')
        cvv = int(request.POST.get('cvv'))
        agentName = request.session['loggedagent']
        email = request.session['email']

        agentQuantities = checkusercrypto(email, currencyname)
        print("Agents Quantity ", agentQuantities)
        if agentQuantities == 0:
            print("AM in IF block")
            AgentHadCrypto.objects.create(currencyName=currencyname,
useremail=email, quantity=quantity)
        else:
            totalQuanty = int(agentQuantities) + quantity
            print("AM in else block ",totalQuanty )
            AgentHadCrypto.objects.filter(currencyName=currencyname,
useremail=email).update(quantity=totalQuanty)
    AgentBuyCryptoModel.objects.create(agentName =
agentName,agentemail=email,currencyname=currencyname,currentprice=currentprice
,quantity = quantity,payableammount = payableammount,cardnumber =
cardnumber,nameoncard = nameoncard,cardexpiry = cardexpiry,cvv= cvv)
    dict1 = AgentHadCrypto.objects.filter(useremail=email)
    dict2 = AgentBuyCryptoModel.objects.filter(agentemail=email)
    return render(request, 'agents/agentbuyed.html', {"object1": dict1, 'object2': dict2})


def checkusercrypto(useremail, currencyname):
    qty = 0
    try:
        obj = AgentHadCrypto.objects.get(currencyName=currencyname,
useremail=useremail)
```

```
        qty = obj.quantity
    except Exception as e:
        qty = 0
        print('Error is ', str(e))
    return qty
def AgentHadCoins(request):
    email = request.session['email']
    dict1 = AgentHadCrypto.objects.filter(useremail=email)
    dict2 = AgentBuyCryptoModel.objects.filter(agentemail=email)
    return render(request,'agents/agentbuyed.html',{"object1":dict1,'object2':dict2})
def AgentLedgerStatus(request):
    email = request.session['email']
    check = BlockChainLedger.objects.aggregate(Sum('blockchainmoney'))
    x = check.get("blockchainmoney__sum")
    x = round(x, 2)
    print('Totoal Ledger Sum ',x)
    dict = BlockChainLedger.objects.filter(agentemail=email)
    return render(request,'agents/agentblock.html',{'objects':dict,'ledger':x})

def AgentPredectionTest(request):
    dict = {}
    dirName = settings.MEDIA_ROOT
    listOfFile = getListOfFiles(dirName)
    # print('List Files ',listOfFile)
    count = 0;
    for x in listOfFile:
        count += 1
        x1 = os.path.basename(x)
        dict.update({count: x1})
    print('List Of Files = ', dict)
    return render(request, 'agents/agentpredictTest.html', {'dict': dict})
```

```
def getListOfFiles(dirName):
    # create a list of file and sub directories
    # names in the given directory
    listOfFile = os.listdir(dirName)
    allFiles = list()
    # Iterate over all the entries
    for entry in listOfFile:
        # Create full path
        fullPath = os.path.join(dirName, entry)
        # If entry is a directory then get the list of files in this directory
        if os.path.isdir(fullPath):
            allFiles = allFiles + getListOfFiles(fullPath)
        else:
            allFiles.append(fullPath)
    return allFiles

defAgentredictTestProcess(request,value):
     print("FIle is ",value)

    fileName = settings.MEDIA_ROOT + "\\" + value
    print('Dataset Name  is ', fileName)
    obj = GetData()
    list = obj.generateTrading()
    # print("List Data is ",list)
    pPath = settings.MEDIA_ROOT + "\\" + "predections.txt"
    with open(pPath, 'a') as f:
        # f.write("Date,Open,High,Low,Close,Volume,OpenInt")
        # f.write('\n')
        for item in list:
            for x in item:
                f.write("%s," % x)
            f.write('\n')

    predictionstart(fileName)
    return redirect("AgentPredectionTest")
```

8. SYSTEM TESTING

8.1. SOFTWARE TESTING TECHNIQUES

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, designing and coding.

8.1.1. Unit Testing

In software testing, Unit testing mainly focuses on verification effort on the smallest unit of program or software design that is also called a module. In unit testing the procedural or functional design provides a detailed description as a guide, focal the control paths are tested to uncover errors occurred in the designed software within the boundaries of the module. The unit testing of software is normally white box or open testing oriented and the series of steps can be conducted in corresponding or parallel for multiple modules or functions.

8.1.2. Integration Testing

Integration testing is another Testing for systematic technique and product module integrating which constructs the program structure and makes the data flow between the modules, while conducting Integration Testing it requires to uncover errors associated with various interfaces. The main objective is to take unit tested methods and activities to build a program structure that have been dictated by design.

8.1.3. Validation Testing

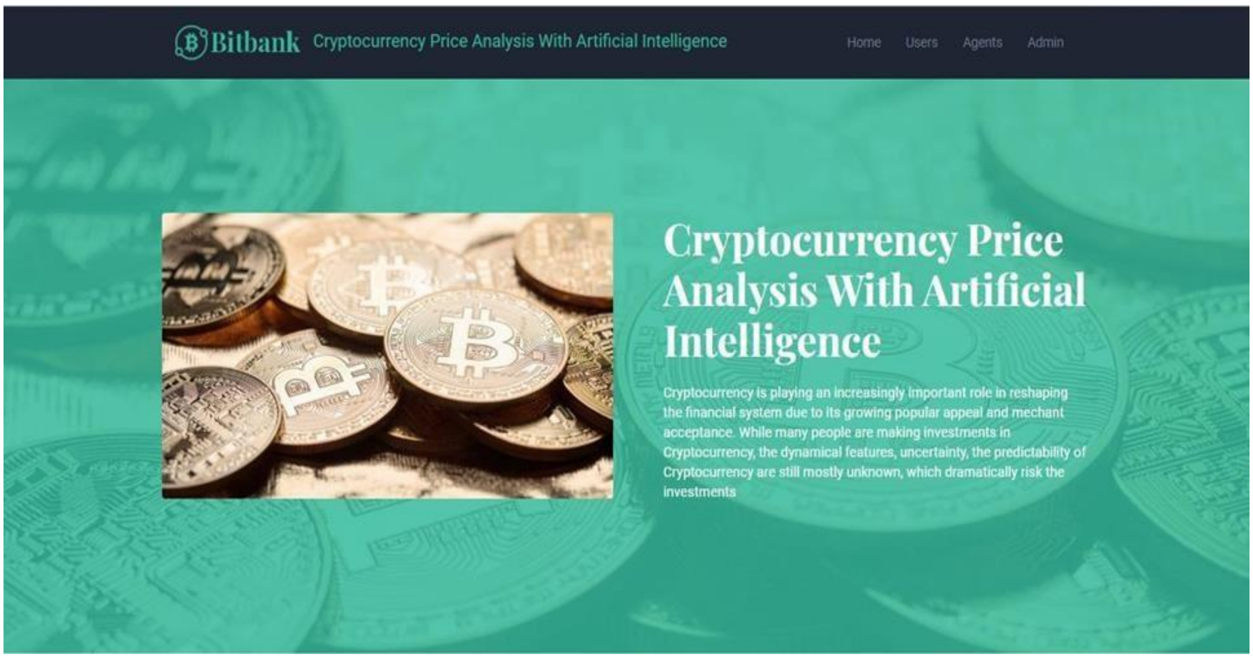
The Validation Testing is integration testing for software which is completely assembled as a package. The Validation testing is the next stage in Testing Activities, which can be defined as successful testing process for the software functions in the mSVM & NAVIBAYESer reasonably expected by the customer. The validation Testing is mainly performed at the end approach of the user needs in testing the information inputed to the product and information contained in those sections are to validated through various testing approaches.

8.2. TEST CASES

S. No.	TEST CASES	INPUT	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	User Registration	Enter all fields	User gets registered	Registration is successful	pass
2	User Registration	if user miss any field	User not registered	Registration is un successful	fail
3	Admin Login	Give the user's name and password	Admin home page should be opened	Admin home Page has been opened	Pass
4	Upload Currency Dataset	Test whether the Currency Dataset is uploaded or not into the system	If Currency Dataset is not uploaded	We cannot do further operations	Currency Dataset uploaded we will do further operations
5	Preprocess Dataset	Verify the Dataset is Per – processed or not	Without loading the dataset	We cannot Per-processing Dataset	We Can Pre-process Dataset successfully
6	Run algorithm	Verify the algorithm will work or not	Without training model	We cannot run algorithm	We can run algorithm

Table 8.1: Test Case Results

9. OUTPUT SCREENSHOTS



Cryptocurrency Price Analysis With Artificial Intelligence

Cryptocurrency is playing an increasingly important role in reshaping the financial system due to its growing popular appeal and mechant 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

Fig 9.1: Home Page

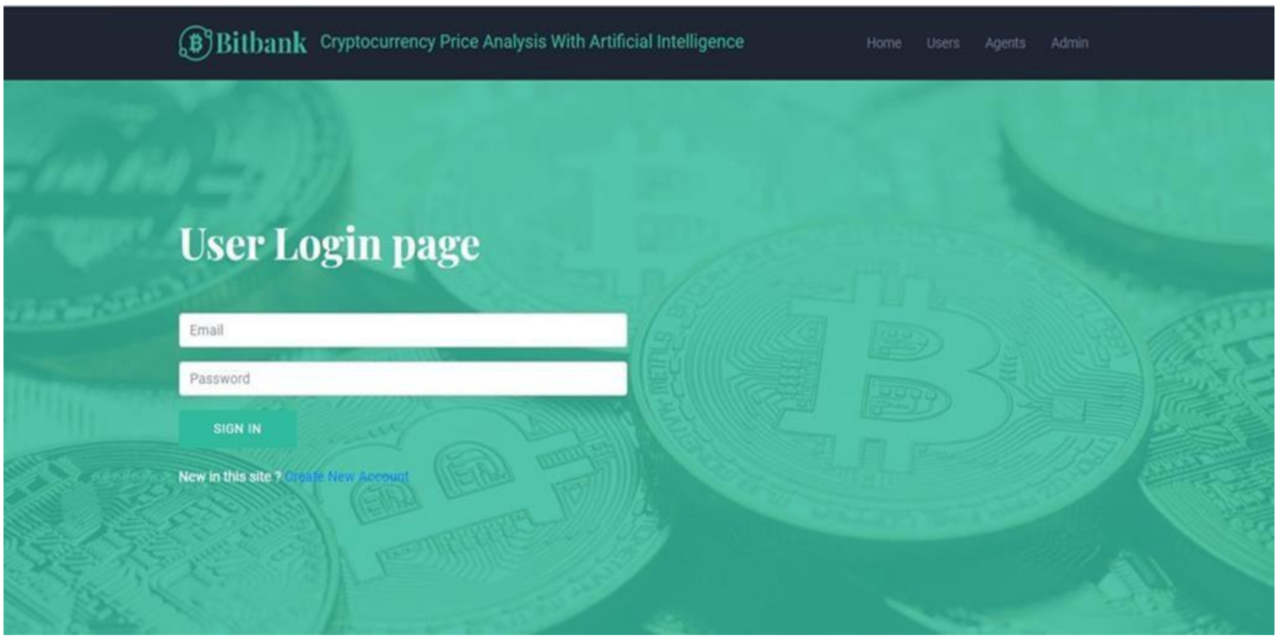

Bitbank
Cryptocurrency Price Analysis With Artificial Intelligence
Home
Users
Agents
Admin
User Login page
Email
Password
SIGN IN
New in this site ? Create New Account

Fig 9.2: User Register Page

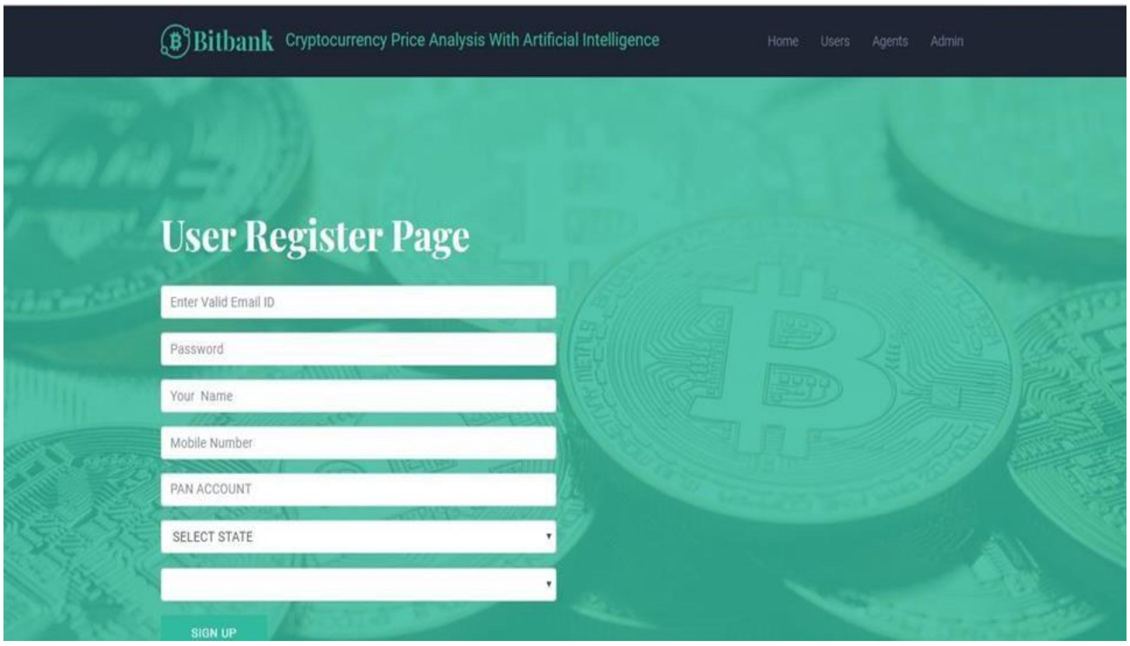


Fig 9.3: User Registration Form

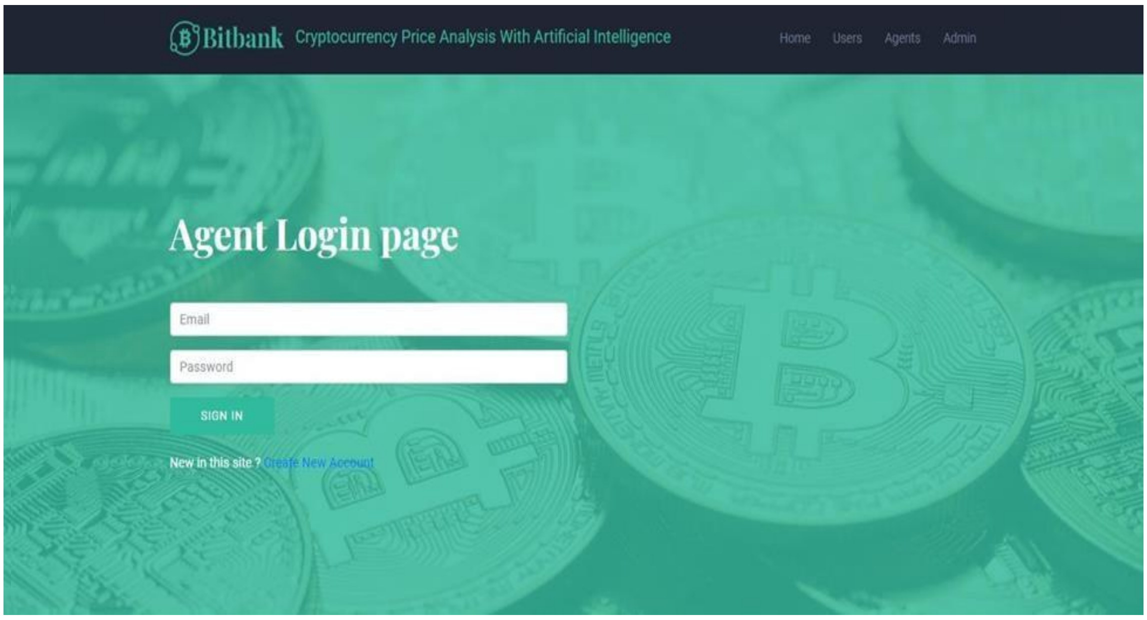

Bitbank Cryptocurrency Price Analysis With Artificial Intelligence
Home
Users
Agents
Admin
Agent Login page
Email
Password
SIGN IN
New in this site ? Create New Account

Fig 9.4: Agent Login page

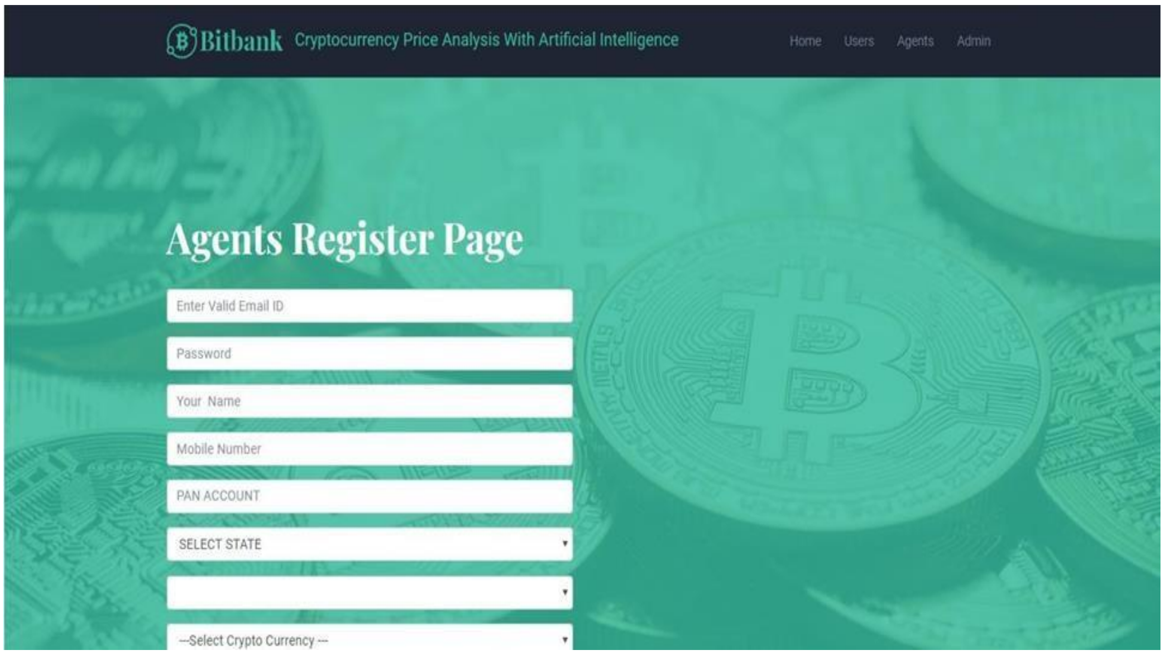


Fig 9.5: Agent Register page

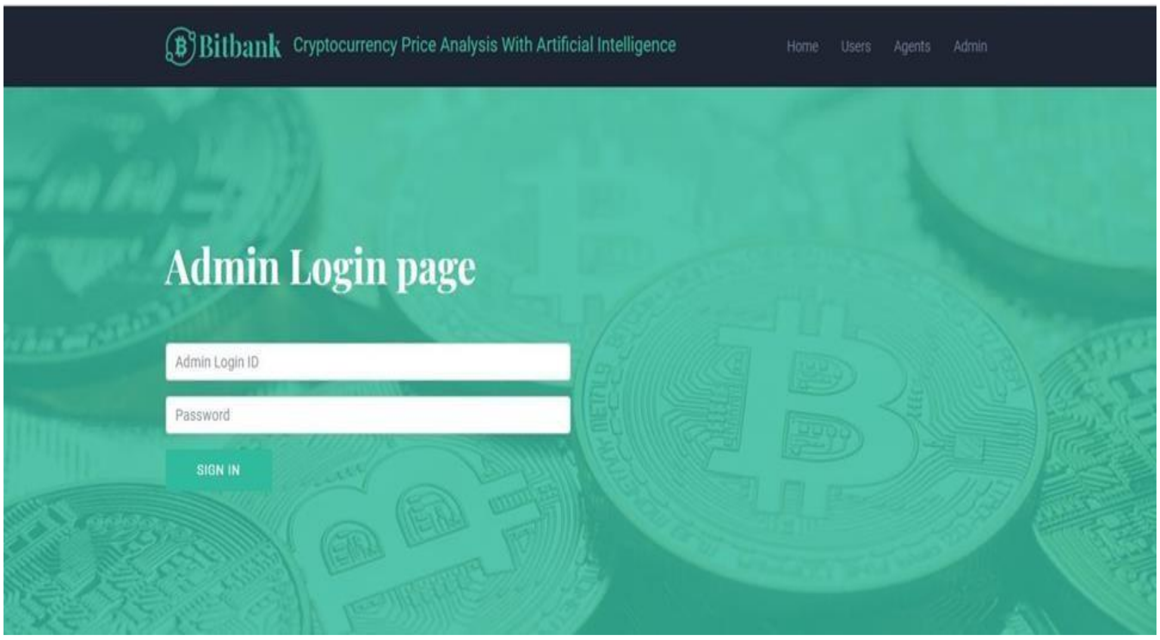
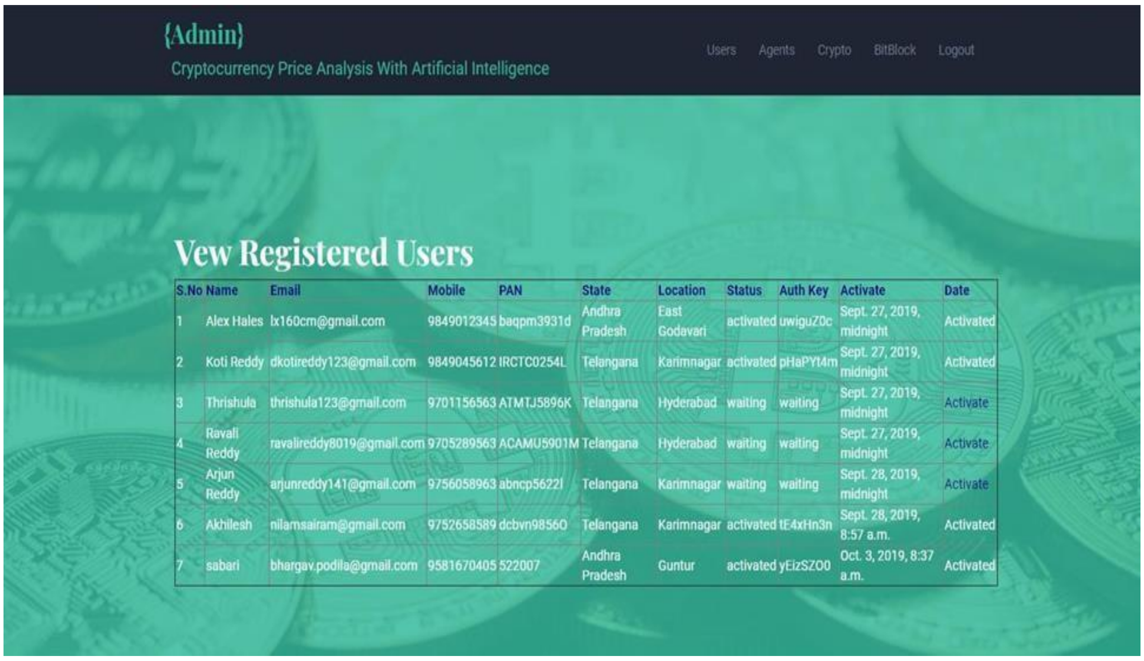

Bitbank Cryptocurrency Price Analysis With Artificial Intelligence
Home
Users
Agents
Admin
Admin Login page
Admin Login ID
Password
SIGN IN

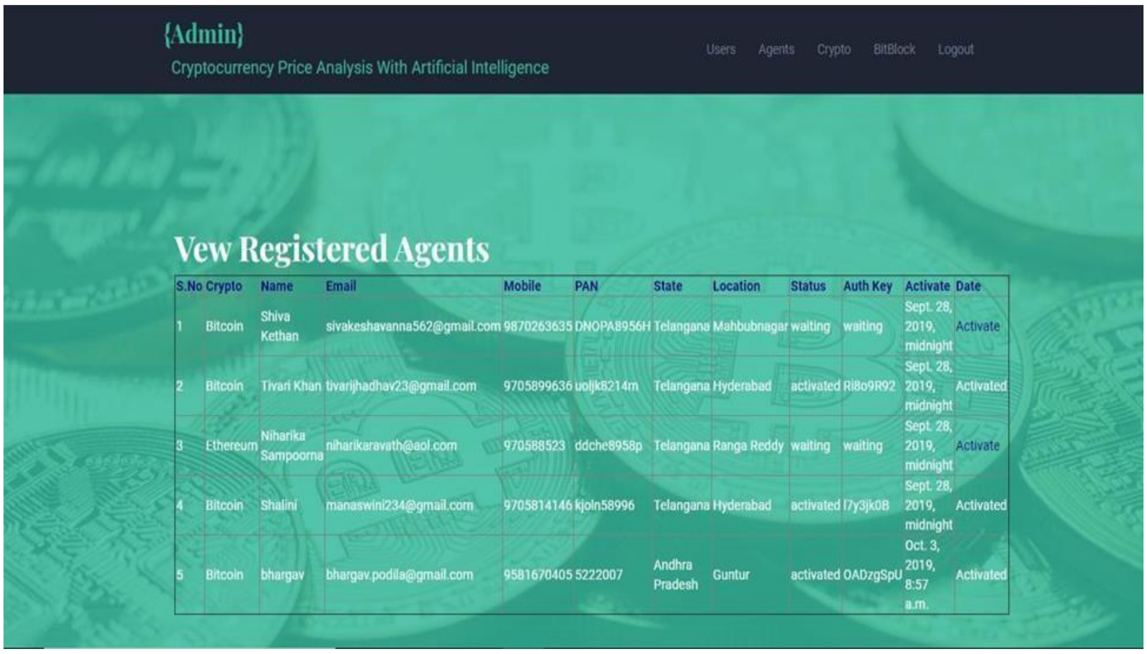
Fig 9.6: Admin Login Page



Vew Registered Users

S.No	Name	Email	Mobile	PAN	State	Location	Status	Auth Key	Activate	Date
1	Alex Hales	lx160cm@gmail.com	9849012345	baqpm3931d	Andhra Pradesh	East Godavari	activated	uwiguZ0c	Sept. 27, 2019, midnight	Activated
2	Koti Reddy	dkotireddy123@gmail.com	9849045612	IRCTC0254L	Telangana	Karimnagar	activated	pHaPYt4m	Sept. 27, 2019, midnight	Activated
3	Thrishula	thrishula123@gmail.com	9701156563	ATMTJ5896K	Telangana	Hyderabad	waiting	waiting	Sept. 27, 2019, midnight	Activate
4	Ravali Reddy	ravalireddy8019@gmail.com	9705289563	ACAMU5901M	Telangana	Hyderabad	waiting	waiting	Sept. 27, 2019, midnight	Activate
5	Arjun Reddy	arjunreddy141@gmail.com	9756058963	abncp5622l	Telangana	Karimnagar	waiting	waiting	Sept. 28, 2019, midnight	Activate
6	Akhilesh	nilamsairam@gmail.com	9752658589	dcbvn98560	Telangana	Karimnagar	activated	tE4xHn3n	Sept. 28, 2019, 8:57 a.m.	Activated
7	sabari	bhargav.podila@gmail.com	9581670405	522007	Andhra Pradesh	Guntur	activated	yEizSZO0	Oct. 3, 2019, 8:37 a.m.	Activated

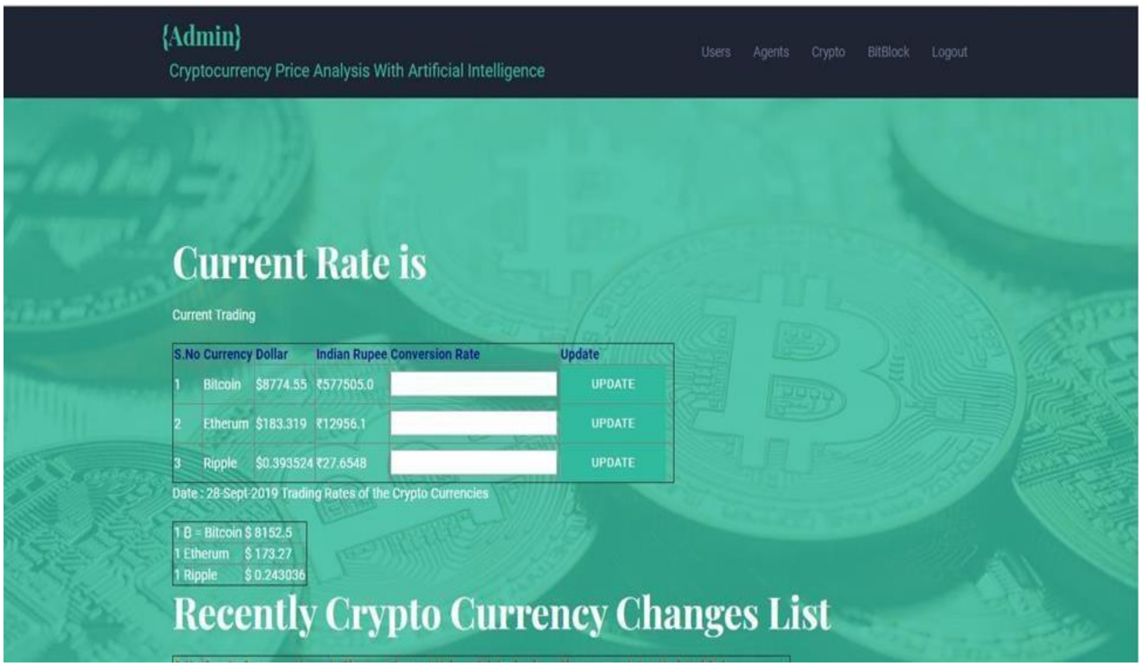
Fig 9.7: Admin Activate Users



Vew Registered Agents

S.No	Crypto	Name	Email	Mobile	PAN	State	Location	Status	Auth Key	Activate	Date
1	Bitcoin	Shiva Kethan	sivakeshavanna562@gmail.com	9870263635	DNOPA8956H	Telangana	Mahbubnagar	waiting	waiting	Sept. 28, 2019, midnight	Activate
2	Bitcoin	Tivari Khan	tivarijhadhav23@gmail.com	9705899636	uoljk8214m	Telangana	Hyderabad	activated	Ri8o9R92	Sept. 28, 2019, midnight	Activated
3	Ethereum	Niharika Sampoorna	niharikaravath@aol.com	970588523	ddche8958p	Telangana	Ranga Reddy	waiting	waiting	Sept. 28, 2019, midnight	Activate
4	Bitcoin	Shalini	manaswini234@gmail.com	9705814146	kjoln58996	Telangana	Hyderabad	activated	l7y3jk0B	Sept. 28, 2019, midnight	Activated
5	Bitcoin	bhargav	bhargav.podila@gmail.com	9581670405	5222007	Andhra Pradesh	Guntur	activated	OADzgSpU	Oct. 3, 2019, 8:57 a.m.	Activated

Fig 9.8: Admin Activate Agents



Current Rate is

Current Trading

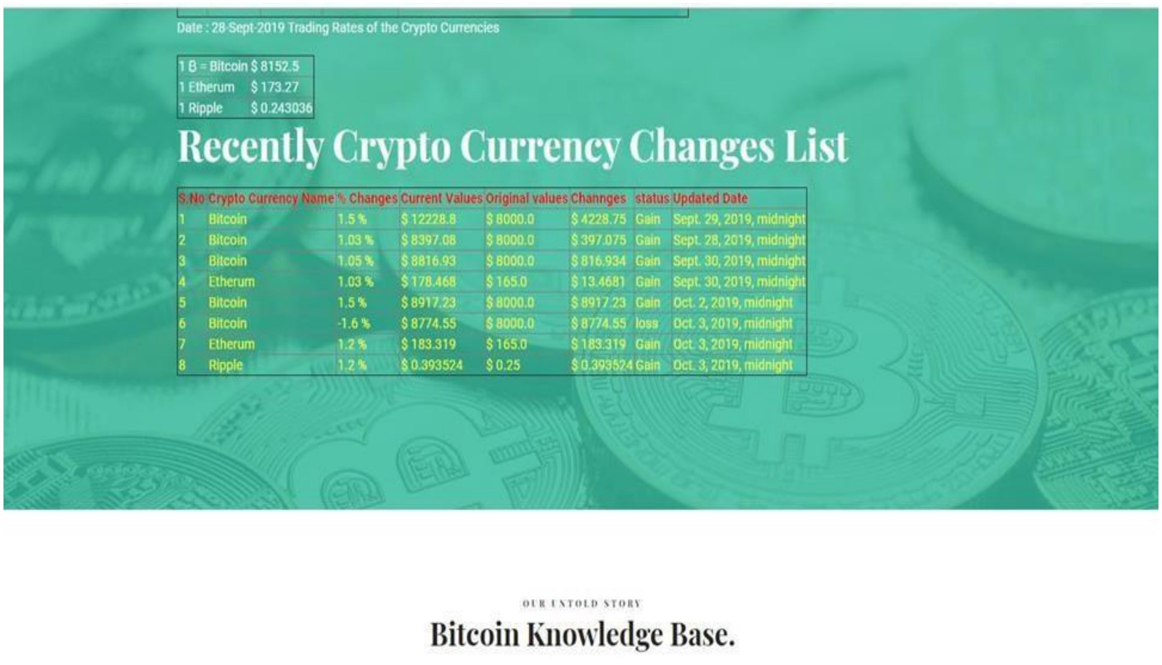
S.No	Currency	Dollar	Indian Rupee	Conversion Rate	Update
1	Bitcoin	$8774.55	₹577505.0		UPDATE
2	Etherum	$183.319	₹12956.1		UPDATE
3	Ripple	$0.393524	₹27.6548		UPDATE

Date : 28-Sept-2019 Trading Rates of the Crypto Currencies

1 B = Bitcoin	$ 8152.5
1 Etherum	$ 173.27
1 Ripple	$ 0.243036

Recently Crypto Currency Changes List

Fig 9.9: Current Price and Update



Date : 28-Sept-2019 Trading Rates of the Crypto Currencies

1 B = Bitcoin	$ 8152.5
1 Etherum	$ 173.27
1 Ripple	$ 0.243036

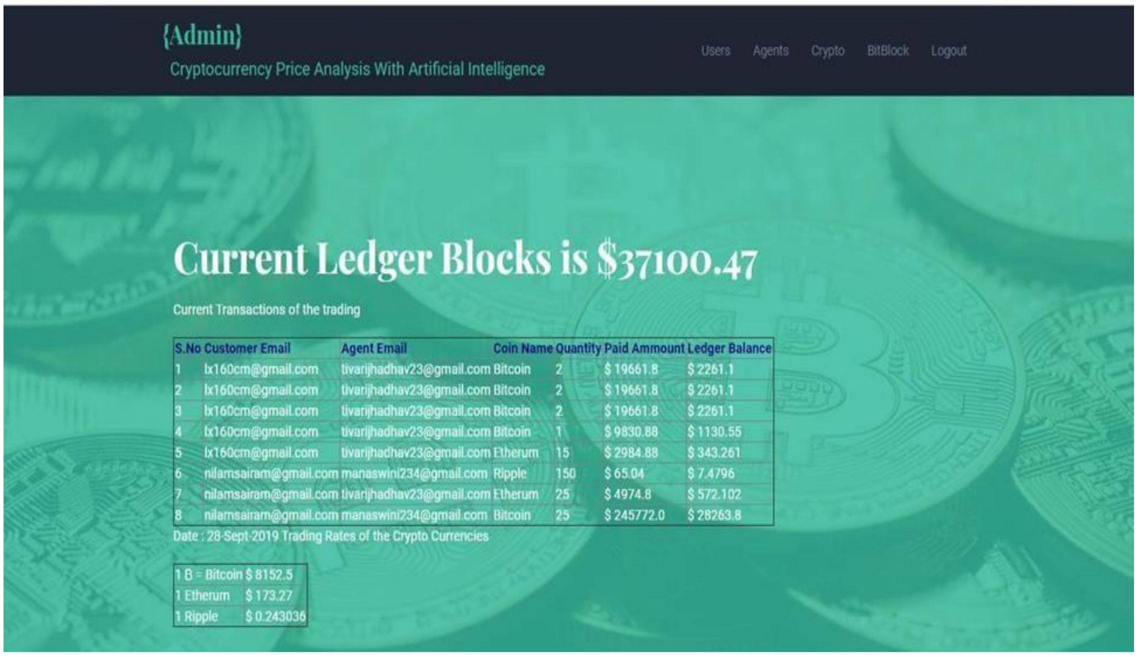
Recently Crypto Currency Changes List

S.No	Crypto Currency Name	% Changes	Current Values	Original values	Channges	status	Updated Date
1	Bitcoin	1.5 %	$ 12228.8	$ 8000.0	$ 4228.75	Gain	Sept. 29, 2019, midnight
2	Bitcoin	1.03 %	$ 8397.08	$ 8000.0	$ 397.075	Gain	Sept. 28, 2019, midnight
3	Bitcoin	1.05 %	$ 8816.93	$ 8000.0	$ 816.934	Gain	Sept. 30, 2019, midnight
4	Etherum	1.03 %	$ 178.468	$ 165.0	$ 13.4681	Gain	Sept. 30, 2019, midnight
5	Bitcoin	1.5 %	$ 8917.23	$ 8000.0	$ 8917.23	Gain	Oct. 2, 2019, midnight
6	Bitcoin	-1.6 %	$ 8774.55	$ 8000.0	$ 8774.55	loss	Oct. 3, 2019, midnight
7	Etherum	1.2 %	$ 183.319	$ 165.0	$ 183.319	Gain	Oct. 3, 2019, midnight
8	Ripple	1.2 %	$ 0.393524	$ 0.25	$ 0.393524	Gain	Oct. 3, 2019, midnight

OUR UNTOLD STORY

Bitcoin Knowledge Base.

Fig 9.10: Crypto update history



Current Ledger Blocks is $37100.47

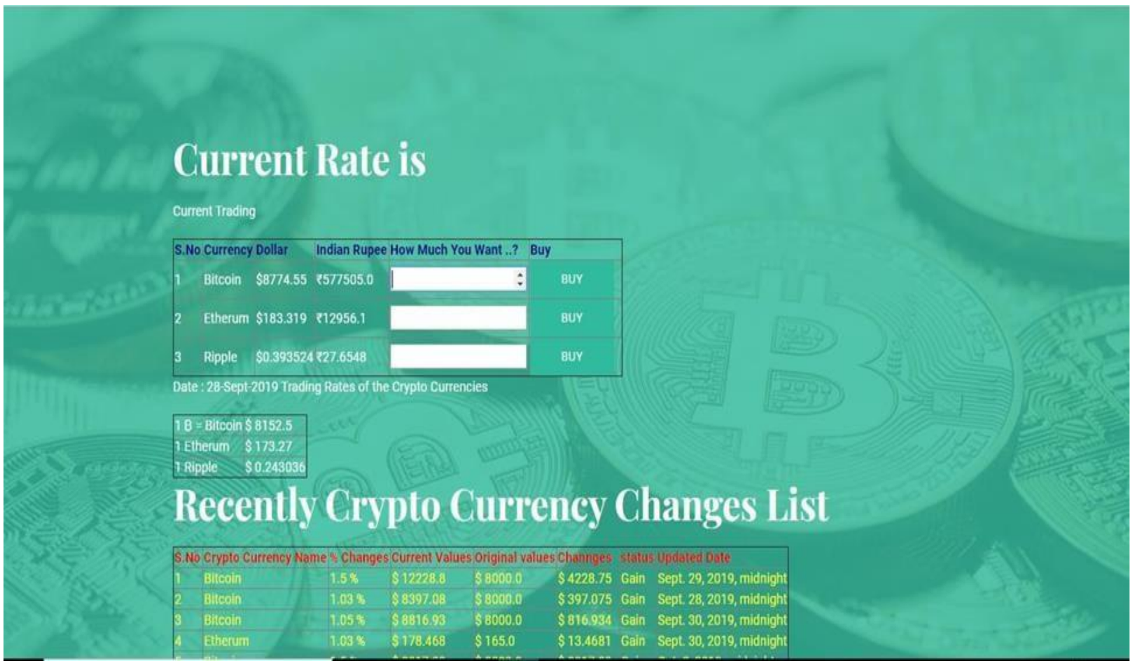
Current Transactions of the trading

S.No	Customer Email	Agent Email	Coin Name	Quantity	Paid Ammount	Ledger Balance
1	lx160cm@gmail.com	tivarijhadhav23@gmail.com	Bitcoin	2	$ 19661.8	$ 2261.1
2	lx160cm@gmail.com	tivarijhadhav23@gmail.com	Bitcoin	2	$ 19661.8	$ 2261.1
3	lx160cm@gmail.com	tivarijhadhav23@gmail.com	Bitcoin	2	$ 19661.8	$ 2261.1
4	lx160cm@gmail.com	tivarijhadhav23@gmail.com	Bitcoin	1	$ 9830.88	$ 1130.55
5	lx160cm@gmail.com	tivarijhadhav23@gmail.com	Etherum	15	$ 2984.88	$ 343.261
6	nilamsairam@gmail.com	manaswini234@gmail.com	Ripple	150	$ 65.04	$ 7.4796
7	nilamsairam@gmail.com	tivarijhadhav23@gmail.com	Etherum	25	$ 4974.8	$ 572.102
8	nilamsairam@gmail.com	manaswini234@gmail.com	Bitcoin	25	$ 245772.0	$ 28263.8

Date : 28-Sept-2019 Trading Rates of the Crypto Currencies

1 B = Bitcoin	$ 8152.5
1 Etherum	$ 173.27
1 Ripple	$ 0.243036

Fig 9.11: Blockchain ledger maintenance



Current Rate is

Current Trading

S.No	Currency	Dollar	Indian Rupee	How Much You Want ..?	Buy
1	Bitcoin	$8774.55	₹577505.0		BUY
2	Etherum	$183.319	₹12956.1		BUY
3	Ripple	$0.393524	₹27.6548		BUY

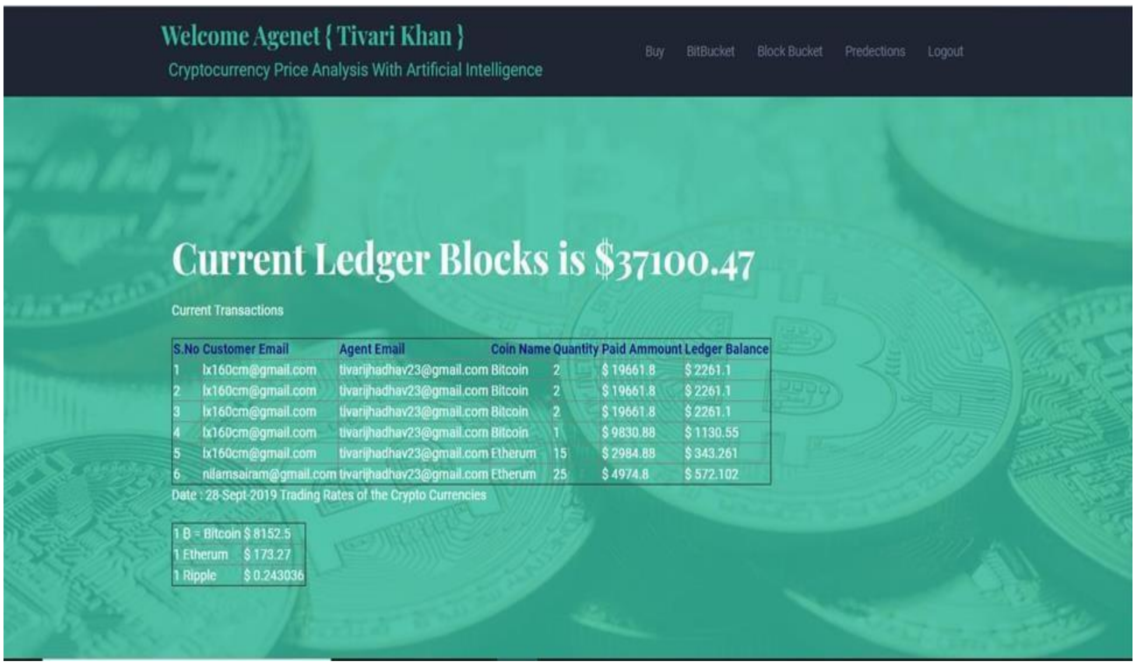
Date : 28-Sept-2019 Trading Rates of the Crypto Currencies

1 B = Bitcoin	$ 8152.5
1 Etherum	$ 173.27
1 Ripple	$ 0.243036

Recently Crypto Currency Changes List

S.No	Crypto Currency Name	% Changes	Current Values	Original values	Channges	status	Updated Date
1	Bitcoin	1.5 %	$ 12228.8	$ 8000.0	$ 4228.75	Gain	Sept. 29, 2019, midnight
2	Bitcoin	1.03 %	$ 8397.08	$ 8000.0	$ 397.075	Gain	Sept. 28, 2019, midnight
3	Bitcoin	1.05 %	$ 8816.93	$ 8000.0	$ 816.934	Gain	Sept. 30, 2019, midnight
4	Etherum	1.03 %	$ 178.468	$ 165.0	$ 13.4681	Gain	Sept. 30, 2019, midnight

Fig 9.12: Agent buying crypto coins


Welcome Agenet { Tivari Khan }
Cryptocurrency Price Analysis With Artificial Intelligence
Buy
BitBucket
Block Bucket
Predections
Logout

Current Ledger Blocks is $37100.47

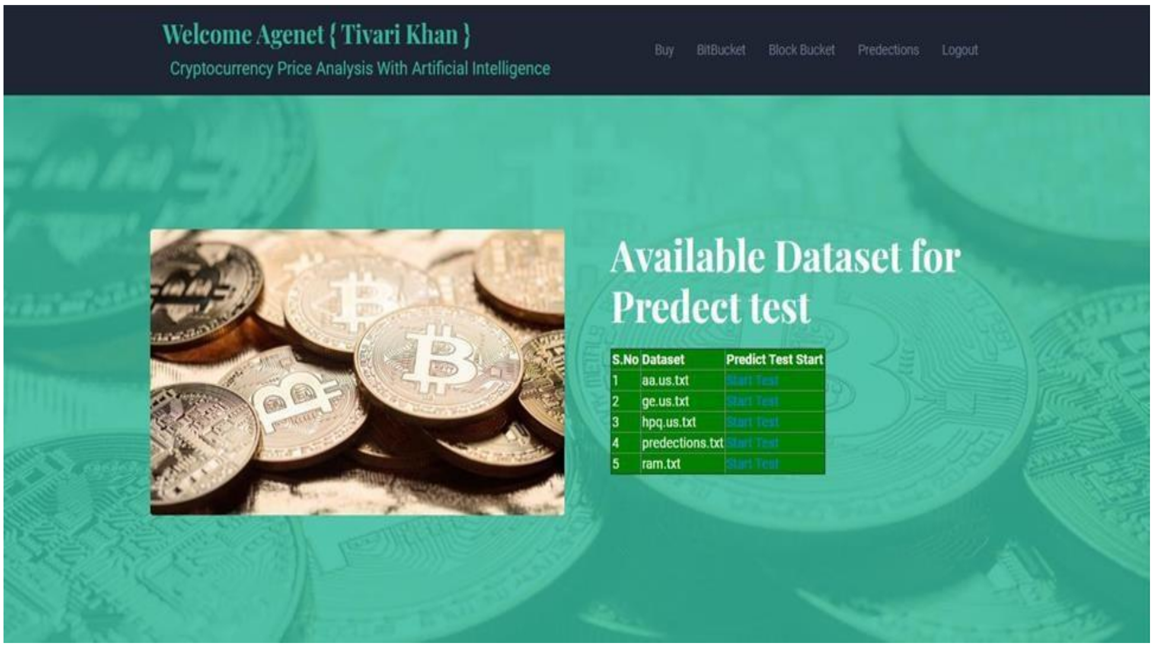
Current Transactions

S.No	Customer Email	Agent Email	Coin Name	Quantity	Paid Ammount	Ledger Balance
1	lx160cm@gmail.com	tivarijhadhav23@gmail.com	Bitcoin	2	$ 19661.8	$ 2261.1
2	lx160cm@gmail.com	tivarijhadhav23@gmail.com	Bitcoin	2	$ 19661.8	$ 2261.1
3	lx160cm@gmail.com	tivarijhadhav23@gmail.com	Bitcoin	2	$ 19661.8	$ 2261.1
4	lx160cm@gmail.com	tivarijhadhav23@gmail.com	Bitcoin	1	$ 9830.88	$ 1130.55
5	lx160cm@gmail.com	tivarijhadhav23@gmail.com	Etherum	15	$ 2984.88	$ 343.261
6	nilamsairam@gmail.com	tivarijhadhav23@gmail.com	Etherum	25	$ 4974.8	$ 572.102

Date : 28-Sept-2019 Trading Rates of the Crypto Currencies

1 B = Bitcoin	$ 8152.5
1 Etherum	$ 173.27
1 Ripple	$ 0.243036

Fig 9.13: Agent view Ledger balance


Welcome Agenet { Tivari Khan }
Cryptocurrency Price Analysis With Artificial Intelligence
Buy
BitBucket
Block Bucket
Predections
Logout

Available Dataset for Predect test

S.No	Dataset	Predict Test Start
1	aa.us.txt	Start Test
2	ge.us.txt	Start Test
3	hpq.us.txt	Start Test
4	predections.txt	Start Test
5	ram.txt	Start Test

Fig 9.14: Agent view predictions dataset for test

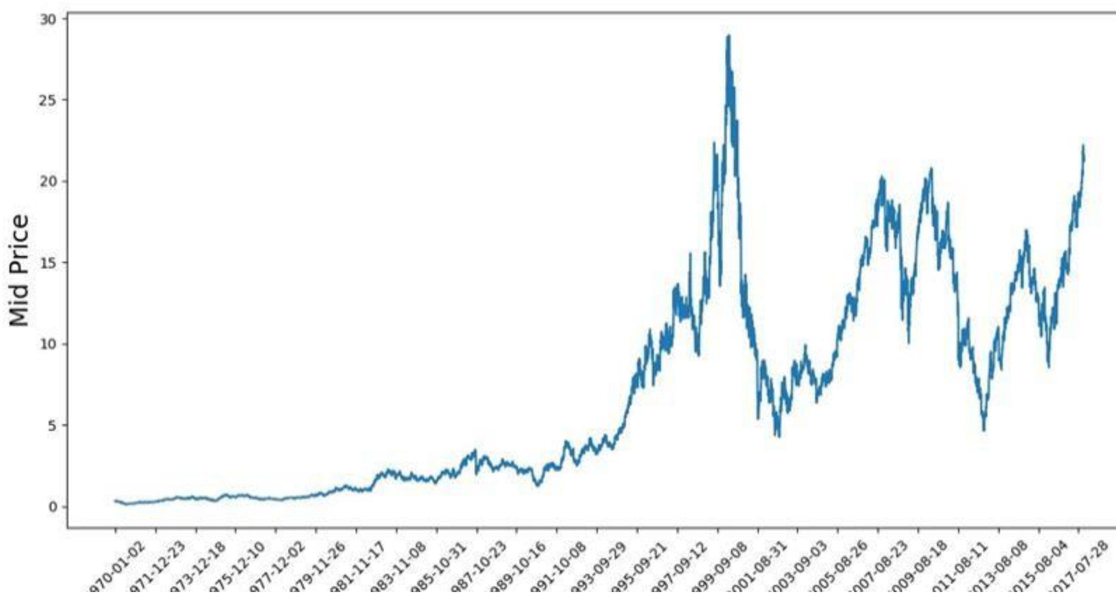


Fig 9.15: Dataset analysis

10. FUTURE ENHANCEMENT

Every development has some drawback or lack of necessary feature that emerges with the usage and need. However, given enough historical information ANN can achieve a similar accuracy, compared with LSTM. This study 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.

11.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 study, we use two distinct artificial intelligence frameworks, namely, fully-connected Artificial Neural Network (ANN) and Long-Short-Term-Memory (LSTM) to analyse and predict the price dynamics of Bitcoin, Ethereum, and Ripple. We showed that the ANN 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 ANN tends to rely more on long-term history while LSTM tends to rely more on short-term dynamics, which indicate the efficiency of LSTM to utilise useful information hidden in historical memory is stronger than ANN.

BIBLIOGRAPHY

[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.

[9] H. White, “Economic prediction using neural networks: The case of ibm daily stock returns,” in Neural Networks, 1988., IEEE International Conference on. IEEE, 1988, pp. 451–458

[10] Kaastra and M. Boyd, “Designing a neural network for forecasting financial and economic time series,” Neurocomputing, vol. 10, no. 3, pp. 215–236, 1996.

[11] H. White, “Economic prediction using neural networks: The case of ibm daily stock returns,” in Neural Networks, 1988., IEEE International Conference on. IEEE, 1988, pp. 451–458

[12] Cheung, Y. W., Chinn, M. D., & Pascual, A. G. (2005). Empirical exchange rate models of the nineties: Are any fit to survive? Journal of international money and finance, 24(7), 1150-1175.