**ALGORITHMIC TRADING BOT**

**MINI PROJECT**

By

**Vaibhav Mehta 60004180114**

**Vatsal Khandor 60004180118**

**Yash Goda 60004180124**

Guide:

**Sudhir Bagul**

Designation

|  |  |  |
| --- | --- | --- |
|  |  |  |

University of Mumbai

2020-2021

**CERTIFICATE**

This is to certify that the mini project entitled **“Algorithmic Trading Bot”** is a bonafide work of **“Vatsal Khandor” (60004180118)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of B.E. in Computer Engineering

**(Name and sign)**

**Guide**

**(Name and sign) (Name and sign)**

**Head of Department Principal**

**Mini Project Report Approval.**

This mini project report entitled ***Algorithmic Trading Bot*** by ***Vatsal Khandor*** is approved for the partial fulfillment of the degree of ***B.E. in Computer Engineering.***

Examiners

1.---------------------------------------------

2.---------------------------------------------

Date: 01/05/2021

Place: Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

-----------------------------------------

(Signature)

Vaibhav Mehta - 60004180114

Vatsal Khandor - 60004180118

Yash Goda - 60004180124

-----------------------------------------

(Name of students and Roll Nos.)

Date: 01/05/2021

**Abstract**

People are getting smarter every day and correspondingly a large chunk of people have stopped depending upon tips and correspondingly Algorithmic trading is gaining importance. Algorithmic trading is a process for executing orders utilizing automated and pre-programmed trading instructions to account for variables such as price, timing, and volume. It utilizes technical analysis of the desired stock, analyzes the resistance and support of a particular stock, and does the needful accordingly.

The basic idea of this project is to create a bot basically using Zerodha API which would automatically execute the transactions based upon a certain set of instructions or basically an algorithm on a broker account. We would fully automate and schedule our trades on a virtual server in the KiteConnect. We can automate trading, and using this API we could give calls based upon our findings automatically. Different strategies will be devised on the basis of different practical applications of our trained Reinorcement Learning models on the data either fetched through the API or the live Dataset.

Volatility forecasts and stock price forecasts play major roles in algorithmic trading. In this project, joint forecasts of volatility and stock price are first obtained and then applied to algorithmic trading. Multi-stepahead interval forecasts for nonstationary stock price series are obtained. As an application, one-step-ahead interval forecasts are used to propose a novel dynamic data-driven algorithmic trading strategy. For this project, we plan to use the Black Scholes Merton equation, which is an equation that considers both volatility and standard deviation in consideration to predict the price of a commodity. This will be dependent upon financial analysis done before and the model used to predict the value of a stock. Reinforcement Learning will be used to understand the behavior of a particular stock from its past experiences.

Apart from this, human sentiments also play a major role in deciding the stock value. Therefore, we will consider sentiments for any commodity we decide. Hence, we apply NLP techniques to see the sentiments and the reaction to the market. For example, we saw a huge increase in the value of Bitcoin and Etherum recently after Elon Musk tweeted it.

Finally, we make a model which incorporates both technical analysis and human sentiments based on news headlines or twitter and use them together to provide a robust model. The model can perform with a better efficacy and give better returns if these techniques are taken into account.

**Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Contents** | **Page No.** |
| **1** | **INTRODUCTION** |  |
|  | **1.1 Description** |  |
|  | **1.2 Problem Formulation** |  |
|  | **1.3 Motivation** |  |
|  | **1.3 Proposed Solution** |  |
|  | **1.4 Scope of the project** |  |
| **2** | **REVIEW OF LITERATURE** |  |
|  | **2.1 Previous work** |  |
|  | **2.2 Research Gap** |  |
| **3** | **SYSTEM ANALYSIS** |  |
|  | **3.1 Functional Requirements** |  |
|  | **3.2 Non Functional Requirements** |  |
|  | **3.3 Specific Requirements** |  |
|  | **3.4 Use-Case Diagrams and description** |  |
| **4** | **ANALYSIS MODELING** |  |
|  | **4.1 Data Modeling** |  |
|  | **4.2 Activity Diagrams / Class Diagram /sequence /collaboration /state** |  |
|  |  |  |
| **5** | **DESIGN** |  |
|  | **5.1 Architectural Design for proposed system** |  |
|  |  |  |
| **6** | **IMPLEMENTATION (if applicable )** |  |
|  | **6.1 Algorithms / Methods Used** |  |
|  | **6.2 Working of the project** |  |
| **7** | **TESTING(white box test cases )** |  |
| **8** | **RESULTS AND DISCUSSIONS** |  |
| **9** | **CONCLUSIONS & FUTURE SCOPE** |  |

Appendix

Literature Cited

Publications by your group (if any)

Acknowledgements

ii

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Fig. No.** | **Figure Caption** | **Page No.** |
| 3.1 | Use Case Diagram |  |
| 4.1 | Activiy Diagram |  |
| 4.2 | Class Diagram |  |
| 4.3 | Sequence Diagram |  |
| 4.4 | Collaboration Diagram |  |
| 5.1 | State Diagram |  |
| 5.2 | Architectural Diagram |  |

iii

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Title** | **Page No.** |
| 3.1 |  |  |
| 4.1 |  |  |
| 4.2 |  |  |
| 4.3 |  |  |
| 4.4 |  |  |
| 5.1 |  |  |
| 5.2 |  |  |

iv

**List of Abbreviations**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Abbreviation** | **Expanded form** |
| i | DSS | Decision Support System |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

v

1. **Introduction**
   1. **Description**

Algorithmic trading is a process for executing orders utilizing automated and pre-programmed trading instructions to account for variables such as price, timing, and volume. It utilizes technical analysis of the desired stock, analyzes the resistance and support of a particular stock, and does the needful accordingly. Trading stocks is a financial instrument developed over years to distribute the risk of

a venture and to utilize the stagnant wealth. Distributing the securities, get the company

capital for growth which in turn create more jobs, efficient manufacturing, and

cheaper goods. Trading of stocks makes the economy more flexible while delivering

benefits both for the issuer and the holder. Stock trading has gained popularity as a way of investment, but the complicated environment of trading and the costs

of expert traders are hurdles for the common public. The development of adaptive

systems that take advantage of the markets while reducing the risk can bring in more

stagnant wealth into the market.

The rise of commission free trading APIs along with cloud computing has made it possible for the average person to run their own algorithmic trading strategies.

We will discuss the concepts we use in the background section. That is followed

by the explanation of the design in the architecture section.

* 1. **Problem Formulation**

Are the current existing algorithmic trading bots capable enough of providing results considering all the factors that affect a particular stock value and are current approaches precise more accurate?

* 1. **Motivation**

The main motive behind choosing this project would probably be the fact that human sentiments and technical indicators clubbed collectively can produce phenomenal results and lead to enormous profits without any human interaction. Current dearth of knowledge in algorithmic trading especially approaches which aren’t talked about for making models and strategies which include Reinforcement Learning as well as sentimental analysis will be demonstrated in this project Report. So, we aim to leverage these techniques and implement an actual trading bot demonstrated on this.

* 1. **Proposed Solution**

Everytime the data is updated the previous live data is appended to the historical data and new live data comes into the picture. Now, for a simple algorithmic trading bot, we devise a few simple strategies and run the Python script. The script understands and follows whether to BUY/SELL/HOLD a particular stock.

This approach requires a little bit of luck since there is not much of a technical analysis and many factors are not considered.

Our approach takes all of these factors into consideration, and we plan to devise a Reinforcement Learning model for the same. Initially, the data is collected and segregated into two sections: live and historical data. RL is a learning method where the agent interacts with the environment by production actions and discovering errors or rewards. It is a type of machine learning where an agent learns to behave in an environment by performing actions and seeing the results. It is all about taking appropriate actions in order to maximise reward in a particular situation. Agent is put into an unknown environment, where it uses hit and trial method inorder to figure out the environment and come up with an outcome. We also aim to consider sentimental analysis based on Twitter or headlines of news from Money Control or other websites which provide pertinent information. A model will be trained on these things discussed above which will try to give as accurate results as possible.

* 1. **Scope Of the Project**

Algorithm Trading is fast and easier as compared to the old and traditional ways of trading. Algo Trading is automated which means you throw your own strategies to the program and it gives you the result according to your preference.  
Algo Trading is backed up with the latest technology which makes your trading hassle-free and easy to use. Due to its ease, it has become highly popular among the experienced traders and those who are new to this industry. In Algo Trading the involvement of humans gets reduced and is taken over by complex calculations and algorithms.

1. **Literature Review**
   1. **Previous Work**

The author [5] proposed an intuitive idea of combining multiple existing techniques into a much more robust prediction model which can handle various scenarios. He suggested taking into consideration both the traditional and modern approaches for stock market analysis. The traditional approach includes fundamental and technical analysis. The modern approach includes qualitative analysis ( which consists of tweets sentiments) and quantitative analysis(which consists of stock historical and current data).  They implemented an individual stock value prediction approach. Their methodology suggested consists of 3 modules: 1. Machine Learning, 2. Sentimental Analysis, 3. Fuzzy Logic. The Machine Learning model calculated the error between the predicted value of opening and closing value and concluded using the RIDGE REGRESSION model. They made the training and testing datasets from the historical and current data.  They collected data from the moneycontrol site for sentiment analysis. and along with this and also the values predicted from the machine learning algorithm were passed on to the fuzzy logic module which determined the stock faith for a particular stock. The stock faith helps in deciding for buying or selling or holding a particular stock.Stock Faith worked on the following 3 rules:

1) IF the News Sentiment was good or the Stock Prediction value was good, THEN the Stock Faith will be high. 2)IF the Stock Prediction value was average, THEN the Stock Faith will be medium.3) IF the News Sentiment was poor and the Stock Prediction value was poor, THEN the Stock Faith will be low. In this paper, the uniqueness is seen in the usage of Fuzzy Logic module which helped determine the stock faith which is strength for recommendation. If the stock faith is high, then the stocks prediction is highly accurate and vice versa.

Author[6] demonstrated sentiment analysis for the stock market by fetching Sensex and Nifty live server data values on different intervals of time that can be used for predicting the stock market status. For using the Machine Learning model, they suggested 2 approaches: the Machine Learning approach and Lexicon Based approach. For the Machine Learning approach, they recommended various models like Supervised Learning, Unsupervised Learning, and several classifiers such as Linear Classifiers, Decision Tree Classifier, Probabilistic Classifiers, and also Hybrid Fuzzy Neural Network based learning. For Lexicon Based approach, they recommended Dictionary and Corpus based approach. For the BSE and NSE stock data, they proposed using the Beautiful Soup python library for extracting live data and this tool helps to pull contents from the desired web pages. This python scripting language has a fast execution environment that provides live and accurate stock value prediction. They performed sentiment analysis on NSE and BSE stocks’ extracted data and historical and current stock value for stock value prediction. This is the only paper which solely focused on sentiment analysis of stock news and also provided a variety of machine learning algorithms which in turn helped us in exploring the best algorithm to be used for our project.

Author [12] proposed automatic swing trading using deep reinforcement learning. Swing trading is modeled as a Markov decision process. Deep Deterministic Policy Gradient (DCPG) algorithm is used for reinforcement learning and Recurrent Convolutional Neural Network(RCNN) algorithm is used for classification of news sentiments. Deep Deterministic Policy Gradient algorithm uses a stochastic behavior policy for good exploration. The DCPG algorithm contains two neural networks, actor and critic. The actor-network is updated using the DDPG algorithm and the critic network is updated using the temporal difference error signal. Recurrent Neural Network(RNN) was not chosen by them due to its failure to identify discriminating phrases in different orders. Convolutional Neural Network(CNN) can fairly determine discriminative phrases in a text with a max-pooling layer. So they considered using Recurrent Convolutional Neural Network(RCNN) as it provides the combined benefit of both RNN and CNN. RCNN accepts word embedding which is the outcome of the text preprocessing as the input. The recurrent nature of the RCNN captures the contextual information to a greater extent while learning word representations. They considered using a reinforcement learning agent(bot). The reinforcement learning system of the trading bot has two parts, agent and environment. They crafted the reinforcement learning agent to interact in an environment consisting of daily stock information, capital, stock assets. The agent interacts with this environment using three actions of buying, selling, and holding the stocks. The financial news along with the change in the stock price acts as the input for the sentiment analysis. The network has four layers: embedding, convolutional, LSTM (recurrent), and output. For the embeddings layer,

they cleaned the sentences using the various preprocessing methods and converted them into a list of indices from the list of words. The convolutional layer created a kernel that is convolved with the input over a single spatial dimension to produce a

Tensor. The LSTM layer introduces a memory element to the network. The layer is efficient in extracting sentence representations enabling our model to analyze long sentences. The final layer is an output that predicts the sentiment as a binary number. The positive sentiment is represented using one and the negative as zero. For sentiment analysis, the training was done with two epochs to avoid overfitting. They designed this architecture for the stock of only one company but they also provided a master network that trained them to leverage the predictions from individual company networks.

Author [7] proposed three baseline models: RL using price-only observation, RL using news-only observation and pretrained sentiment classifier plus manually designed policy. He implemented the sentiment classifier as a 2-layer convolutional nets using pre-trained Bert embedding. On testing his idea, it showed all the methods to be overfitted and the performance on testing data amounts to random guessing. He focused on long-term portfolio management using sentiment analysis and deep reinforcement learning. In this paper, the author extracted sentiments for certain stocks. He considered the aspect-level feature to be an important factor in stocks value prediction.  The author suggested focusing deep learning models that can learn sentiment embeddings directly instead of designing hand-crafted features. The author recommended performing optional reward shaping to facilitate the reinforcement learning algorithm by using indicators of whether the reward is positive or negative.  The author advocated using Proximal Policy Optimization as an RL algorithm. For sentence sentiment classification, Convolutional Neural Networks (CNN) is suggested for high accuracy. The 2-layer of CNN will help prevent the overfitting issue. For the dataset, he advice using the Dow Jones Industrial Average (DJIA) dataset as it contains current as well as historical data and also contains top 25 news headlines for each day.

Authors [12] propose an automated swing trading system using deep reinforcement learning based on a policy-based neural network model which decides the relevant action depending on the model evaluation whether to buy, sell or hold. They use a recurrent neural network model much similar to our implementation for the sentimental analysis model. Authors [12] suggested the usage of RCNN since it would give the benefits of both RNN and CNN for sentimental analysis classifiers. They kept on training the bot for 5 months and compared the results between the initial month and the 5-month result.

They compared the stagnant and RL- bot-based asset graph value graph which showed that the agent always maintains a higher value than the stagnant stock value. For the RL implementation, they gave the agent a binary reward representing if the action was profitable or not. Their agent executed every day observing the environment to select the action with the policy it learned on training.

In another study, various different machine learning techniques to perform algorithmic trading are observed. Author[11] proposes a novel trading approach that uses a Markov Decision Process (MDP) and to improve the performance of Q-learning, the author augments MDP with an estimate of current trend information and sentimental analysis of news articles. The implemented system performs sentimental analysis using Word2Vec and finds the trend analysis using Neural Networks and they formulate the statement as an MDP and solve this MDP using Reinforcement learning. Though, the evidence on the use of live dataset is still widely debated upon since the author uses only a 5-year historical price dataset. And for the sentimental analysis model, the author uses 10-years news articles obtained from Reuters. The author concludes the result based on the term “Sharpe ratio”. The author also captures the current trend information and formulate the augmenting trend information into an MDP. The author also uses 6 different technical indicators: SMA, Moving Average, Stochastic K, Stochastic D, RSI, and Larry William’s R% and scale them and then input them into the neural network to predict the trend for the stock.

The agent was allowed only three actions: buy, sell or hold and the agent jumped in between these based upon the transitional probability where the randomness was based upon the dynamic nature of the financial markets since each time a stock is bought sold, the price changes the next moment itself. The author implemented the Q-learning system with the number of trials as 10,000. The author suggests the usage of policy gradient methods to estimate the best policy without generating the Q-value.

The research by the author [10] seems to agree that the initial hypothesis about the market comes often from intuition or practice and hence reinforcement learning can be a viable option. The author[10] aims to optimize the agent’s performance in an unknown environment by using techniques based on least-squares temporal difference learning. The system has been implemented for foreign exchange, but the author seems to agree that a similar system can also be implemented for the stock markets. The author’s RL trading algorithm uses LSTD methods to compute an optimal policy on the provided training data which is further fed to the Markov Decision Process. This was the only paper we read where the author’s evaluated the algorithmic trading strategy on the basis of backtesting. The author places the challenge that since this is a fairly modern approach and much without a good amount of research, it still achieved a justified amount of profit.

The evidence on which reinforcement learning algorithm to use is overall mixed. For instance, researchers in [9] have used DQN, T-DQN, and Double-DQN algorithms. They insisted on using random samples since past experience being sequential would be largely correlated and such randomizing technique would help break this behavior. Tensorflow has been used to build the 4 hidden layered neural network. Normally, closing price is a parameter used in the datasets but here they used adjusted closing price because it would accurately reflect the stock value after accounting for any corporate action. The strategies used here to maximize gain with minimum risks. The authors [9] used other than standard metrics like Huber loss and Adam optimizer to make the model more robust and since adaptive methods to eliminate sensitive learning rate. FIFO methodology was used to buy and sell stocks and used checkpoints in the dataset to make sure that the data does not overfit. Their result concluded that Double-DQN performs the best than the other two algorithms because it uses two separate Q-value estimators.

The researchers in [9], [10], [11], and [12], all used historical data instead of live datasets. All of them agree that reinforcement learning could be a game-changer in the near future especially in the field of finance.

According to liang2020[2], the algorithmic trading strategy uses the Joint Forecasts of Volatility and Stock Price. One trend in the research follows the indicators of SMA (Simple Moving Average) & related technical indicators. Bollinger bands which are derived from the Moving Averages play a significant role in the trading community but Bollinger bands come with a fallback that they do not consider the price and volatility forecasting.GBM expresses the change in stock price using a constant drift μ and volatility σ as a stochastic differential equation (SDE).Also Autoregressive Integrated Moving Average (ARIMA) models and Generalized Autoregressive Conditional Heteroscedascity (GARCH) models are used to determine the volatility of the stock using the log returns provided by GBM.

The methodology and algorithms are implemented using S&P 500 (500 largest U.S. publicly traded companies) closing prices. 5 Algorithms are considered for the smooth and accurate functioning of the trading strategy. Algorithm 1, Algorithm 2, and Algorithm 3 are used to determine the optimal values for price forecast and volatility forecast, Algorithm 4 is used for Optimal cumulative profit while algorithm 5 is based on the lines of a Simulator.

Stock prices are affected by many factors, including macroeconomics and various news. However, the focus of this research[1] was solely based on the feelings of users (using their comments). The goal of this research was to stimulate a model using emotions based information in social media that could be used to predict stock market fluctuations (red or green) for the next trading day.

They’ve made use of part-of-speech from the LDA model and results have been produced on English and Persian datasets. There are two primary sources to extract the data.

1) Economic news & 2) Social Media(Twitter). A non-parametric topic model is used to learn the quotidian twitter topics.The human sentiment method is based merely on buy-sell labels, number of messages having optimistic and pessimistic outlooks are taken into account for the evaluation for the stock action.In the LDA-POS method instead of removing the stop words part-of-speech is added followed by grouping the words with same POS together. The LDA-POS method outperformed the human sentiments method by nearly 2% for the various datasets.

In the following paper[3], to address the aforementioned challenges and issues, the authors have proposed a novel deep robust reinforcement learning framework for actual algorithmic trading which can automatically trade in the financial markets. The proposed model consists of two main elements, the Environment and the Trading Agent. The Environment manages the past market data and updates the incoming data from exchanges. The Agent is composed of a data preprocessing module and the trading agent implemented by DRL (DQN-based & A3C-based) with the well-designed state, action, reward, and network structure.

Partially observable markov decision process (POMDP) is used for the reinforcement learning as it stands more practical in the real world trading environment than the Markov decision process(MDP).They have made the use of 2 algorithms- DQN & A3C for reinforcement learnings. The SDAEs-LSTM A3C can sell ahead of the LSTM. Therefore, the SDAE-LSTM A3C learns a more valuable strategy and outperform the LSTM algorithm which just predicts the accuracy. It implements the use of SDAE to remove the noisy financial data. It also makes the use of LSDM to extend the deep reinforcement learning algorithm.

* 1. **Research Gap**

Many research papers haven't used reinforcement learning for training the bot which provides more robust outcomes. Also, many research papers have not considered the factor of examining the sentiments from the tweets regarding the stocks. Almost all papers have restricted themselves to using only the historical data and not evaluating the live data as well. Ad hence, this paper aims to formulate a novel approach using RL and making a model that considers a factor of sentiment as well.

1. **System Analysis**
   1. **Functional Requirements**

Get market data - download, filter, and store structured and unstructured data.

Structured data includes real time market data from Reuters or Bloomberg transmitted using a protocol e.g. FIX. Unstructured data includes news and social media data.

Define trading strategy - specify new trading rules and strategies. Trading rule consist of an indicator, an inequality, and a numerical value e.g. "PE ratio" < 10. Trading rules are structured into a decision tree to define a trading strategy (illustrated below).

Analyze securities against trading strategy - for each security, obtain data and filter it through the trading strategy to determine which security to buy. Additionally: for each open position, determine which security to sell. Note: this requirement could vary.

* 1. **Non-functional Requirements**

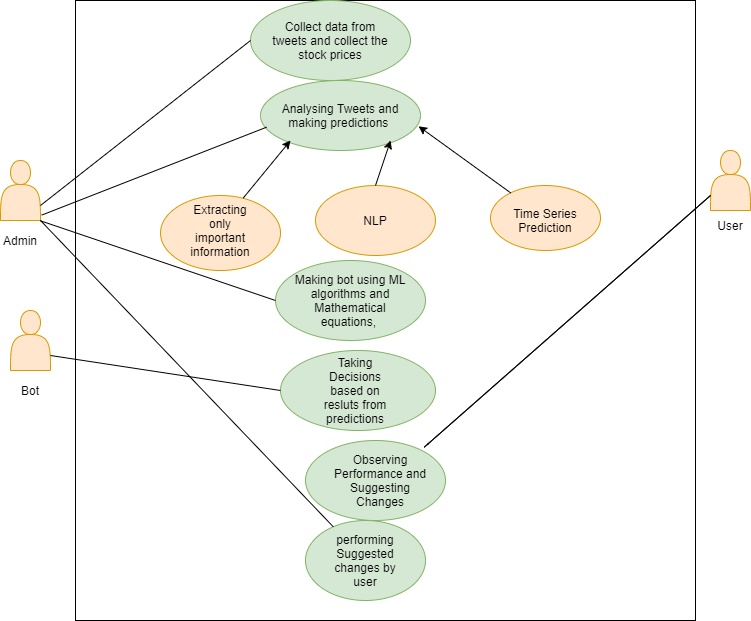
Performance - is the amount of work accomplished by a system compared to the time and resources required to do that work. An ATs should have quick response times (back to the market) and high processing and network throughput.

Modifiability - is the ease with which the system can be changed. An ATs should have easily modifiable trading strategies and data processing

Interoperability - is the ease with which the system is able to operate with a diverse range of related systems. This is important for an ATs which may be required to interface with order management systems, portfolio management systems, risk management systems, accounting systems, and even banking systems.

Reliability - is the accuracy and dependability of a system to produce correct outputs for the inputs it receives. Because errors and bugs in an ATs can result in huge losses and fines, reliability is crucial. See the Knight capital debacle for evidence of this.

* 1. **Specific Requirements**
  2. **Use Case Diagrams and Description**

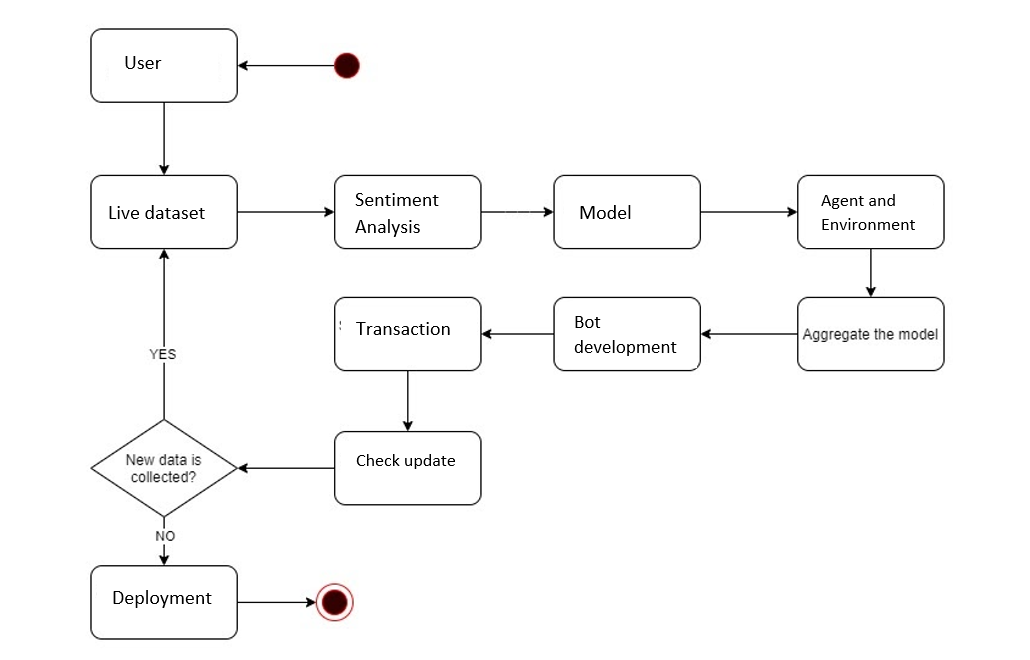
****

**Fig 3.1 Use Case Diagram**

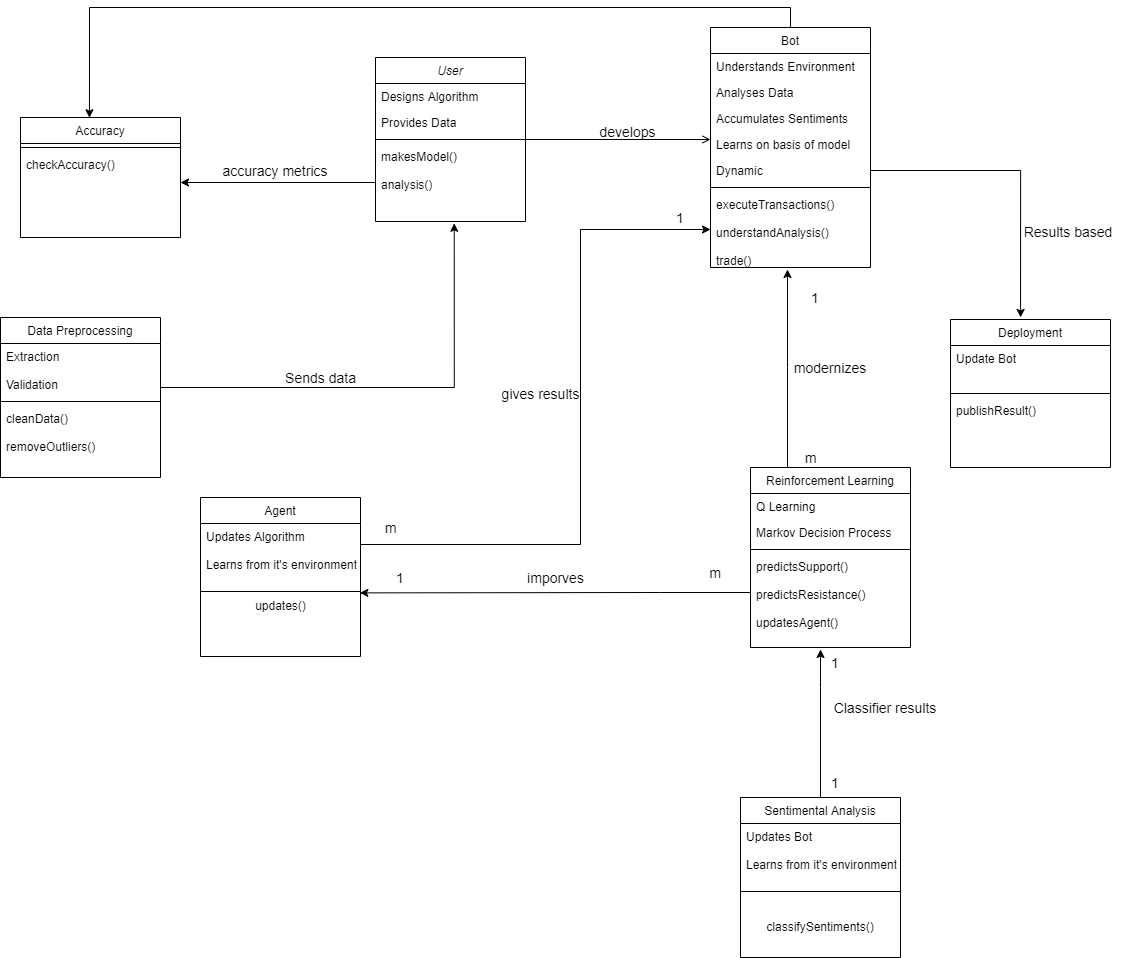
Initially, the data is collected and segregated into two sections: live and historical data. Agent is put into an unknown environment, where it uses hit and trial method in order to figure out the environment and come up with an outcome. Sentimental analysis based on Twitter or headlines of news from Money Control or other websites which provide pertinent information. A model will be trained on these things discussed above which will try to give as accurate results as possible.

After a model is trained, it will do certain actions n the open Stock market and it will subsequently learn from its actions and then the same process of RL is repeated again.

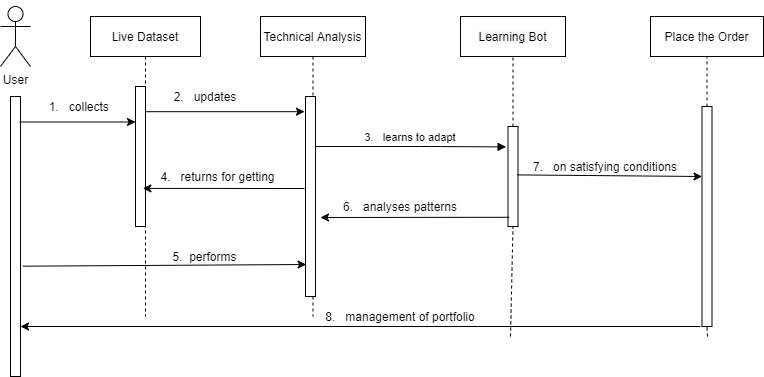
1. **Analysis Modeling**
   1. **Data Modeling**
   2. **Activity Diagrams/Class Diagram/Sequence/Collaboration/State**

****

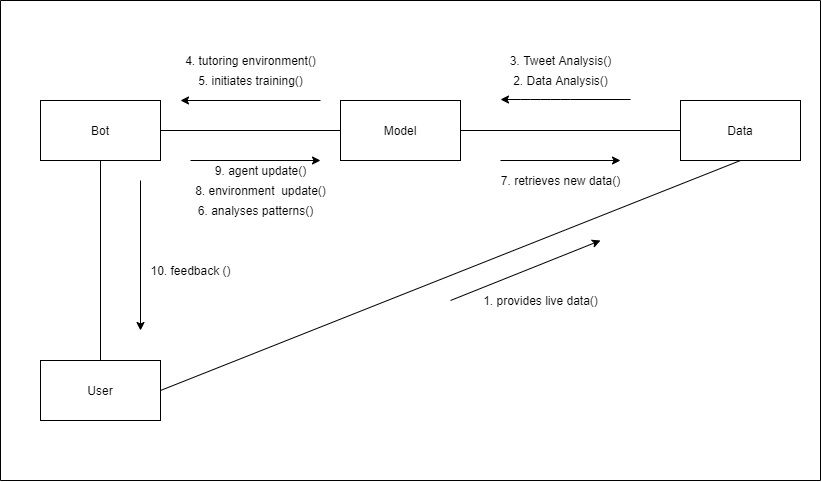
**Fig 4.1 Activity Diagram**



**Fig 4.2 Class Diagram**

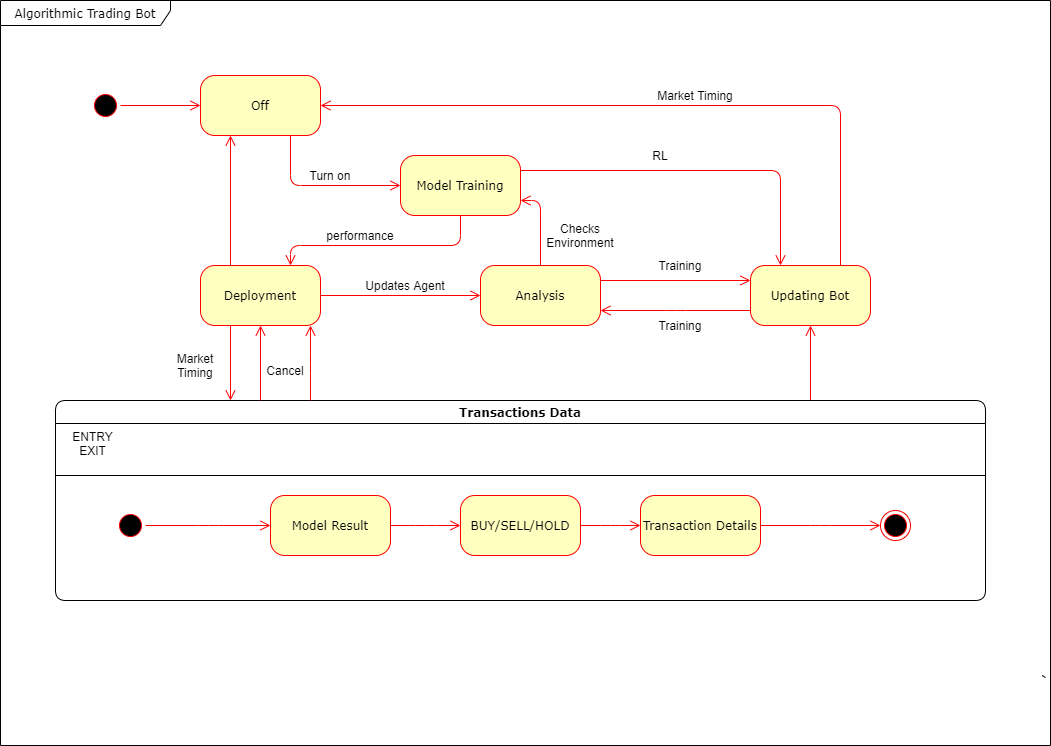


**Fig 4.3 Sequence Diagram**

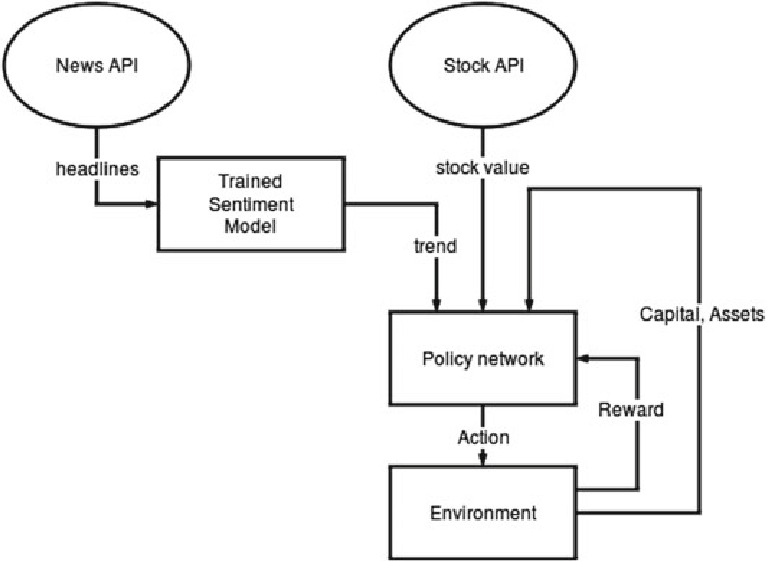
****

**Fig 4.4 Collabration Diagram**

1. **Design**
   1. **Architectural Design for proposed system**



**Fig 5.1 State Diagram**



**Fig 5.2 Architectural Design**