Automated Stock Trading Using Deep Reinforcement Learning And Portfolio Optimization

Parkavi A, Sowmya BJ, Sini Anna Alex

Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bangalore, India Email: Parkavi.a@msrit.edu

Abstract
Trading using intelligent agents has been widely researched since many years. The thought that arose is can self-learning agent take on the role of a human trader. Can Reinforcement Learning (RL) effectively trade to maximize reward? When trading, investor’s goal is to optimize reward, usually profits. Online trading is usually depicted as a two-step decision-making process: 1 Analyzing Market Condition & 2 Taking Optimal Action. In this report, we describe Deep Learning (DL) methods as a solution to automate this online trading process. We present an algorithm called Deep Reinforcement Learning (DRL) where the agent learns the environment, improves itself based on the rewards earned and takes correct action (makes trades) simultaneously. Applied DRL in stock markets to train a single stock trading agent with the goal of maximizing income in short and long term. To imitate a human trader, we also introduced technical indicators and unrealized returns as part of input states to the agent. We compared the DQN algorithm with Double DQN and Dueling Double DQN to find the best agent to maximize profits of the investment strategy. The challenges are being addressed using Deep Q-Network (DQN), Double DQN (DDQN) and Dueling Double DQN (DDDQN) agents in a simulated trading environment. Our model is inspired by off-policy learning and its application in video games. We trained and tested these agents with all S&P 500 stocks and show that just by using price and volume information deep-reinforcement learning agent can make money. Moreover, we also introduce this new workflow in algo-trading strategy development by incorporating the risk modelling and strategy optimization into the reward engineering of the agent. And also employs the Auto ARIMA Model and Holt-Winters model for prediction of stocks and Linear programming for portfolio optimization.Persistently preparing the model utilizing the most recent market information, assists the model with getting prepared on the most recent market conduct, so it tends to be conveyed further for all the more no of portfolio's in future; The pattern acquired can be broke down for something similar to foresee better speculation plans in the stock exchange. "Helping people optimally invest in the stocks, depending upon the present market analysis, which would help them fetch maximum return and suffer comparatively lesser loss!"

Keywords: Deep Learning, ARIMA model, Deep Q-Network, Deep Reinforcement Learning
1. Introduction
Deep learning consolidates counterfeit neural organizations with a support learning engineering that empowers programming characterized specialists to become familiar with the most ideal activities in virtual climate to achieve their objectives. That is, it joins work guess and target improvement, planning state-activity sets to anticipated prizes. Like a pet boosted by reproving and treats, these calculations are punished when they settle on some unacceptable choices and remunerated when they make the correct ones – this is support. Support calculations that fuse deep neural networks can beat human specialists. Reinforcement learning takes care of the troublesome issue of associating quick activities with the postponed returns they produce. Like people, reinforcement learning calculations in some cases need to stand by some time to see the product of their choices. It's sensible to expect that support learning calculations will gradually perform better a lot in more equivocal, genuine conditions while browsing a subjective number of potential activities, instead of from the restricted alternatives of a repeatable computer game. That is, with time we anticipate that they should be important to accomplish objectives in reality.

To plan a robotizing stock exchanging utilizing profound support learning, explicitly Deep Q-realizing, which can pick an activity to sell, purchase, or hold the stocks to expand the increase in resource esteem.

Exchanging stocks is a monetary instrument created over years to circulate the danger of an endeavor and to use the stale abundance. Disseminating the protections, get the organization capital for development which thus make more positions, proficient assembling, and less expensive merchandise. Exchanging of protections makes the economy more adaptable while conveying benefits both for the guarantor and the holder. Stock exchanging has acquired prevalence as a method of venture, yet the convoluted climate of exchanging and the expenses of master merchants are obstacles for the normal public. The improvement of versatile frameworks that exploit the business sectors while diminishing the danger can acquire more stale abundance into the market. The execution of the work is done as follows

- To study the dataset (S&P 500) and pre-process it.
- To implement the Deep Q Network.
- To implement the Double Deep Q Network.
- To implement the Dueling Double Deep Q Network.
- To make a comparative study of the above methods.
- To get the best method which gives the best profit or reward at the end of trading.
- To display the results using a tkinter interface.

The stock market is one of the most exciting ways for companies to raise money, along with debt markets which can be generally extra imposing but do not trade publicly. This lets in businesses to be publicly traded, and raise extra financial capital for enlargement by means of promoting shares of ownership of the business enterprise in a public market. This seems to be an appealing characteristic of investing in shares, in comparison to other less liquid investments which include property and different immovable property. This paintings uses a Model-unfastened Reinforcement Learning method called Deep Q-Learning (neural version of Q-
Learning). At any given time (episode), an agent observes its modern-day nation (n-day window stock rate illustration), selects and plays an action (purchase/sell/maintain), observes a subsequent state, receives a few reward sign (difference in portfolio function) and finally adjusts its parameters based on the Gradient of the loss computed.

2. Literature Survey
Deep reinforcement learning consolidates artificial neural networks with a reinforcement learning engineering that empowers programming characterized specialists to gain proficiency with the most ideal activities in virtual climate to accomplish their objectives. That is, it joins work guess and target advancement, planning state-activity sets to anticipated prizes. Like a pet boosted by chastening and treats, these calculations are punished when they settle on some unacceptable choices and remunerated when they make the correct ones – this is support. Support calculations that consolidate profound neural organizations can beat human specialists playing various Atari computer games, Starcraft II and Dota-2, just as the title holders of Go. It's sensible to accept that support learning calculations will gradually perform better a lot in more equivocal, genuine conditions while browsing a subjective number of potential activities, instead of from the restricted alternatives of a repeatable computer game. That is, with time we anticipate that they should be significant to accomplish objectives in reality. With the general turn of events and modern calculations being made in the fields of profound learning, the assignment of stock expectation can be mechanized, and the more profound jump can be taken into understanding the class of Deep Q Learning. We inspected the proposed strategy for anticipating stocks utilizing AI and profound learning methods, for example, Linear Regression, Multi-Layer Perceptron Model, Convolutional Neural Networks and LSTM and so on.

[1]. Utilizing deep learning to expand the advantages of a progression of dangers in the monetary business sectors is an extremely fascinating and broadly concerned issue. From the point of view of multi-source driving, this paper proposed an element mix technique dependent on earlier information and planned a profound leveled methodology model to tackle this test. This model incorporates a pre-passing judgment on module, which executes pattern judgment of the time arrangement, and an activity module that plays out the exchanging activities. For the pre-passing judgment on module, a relapse imperative Wasserstein generative ill-disposed organization is intended to finish the mission rather than the basic discriminant techniques before.

[2]. In this paper, initial, a base model utilizing LSTM cells is pre-prepared dependent on an adequately huge measure of information, which are acquired from various stocks, to enhance beginning preparing boundaries. To improve execution, the model is then prepared with target stock info highlights with boundary tuning. The test was performed with information from top-five organizations in the Korean market and the United States market from 2012 to 2018 as far as the most noteworthy market capitalization. In this paper, another structure called DTRSI is proposed to anticipate stock value developments with good execution. The model utilizing the profits of organizations with a field like the COI and accomplished the best. To the best of information, this is one of the main examinations utilizing a mix of move learning and long transient memory (LSTM) in monetary time arrangement expectation errands.
The field of AI combined with key and/or Technical Analysis yields acceptable outcomes for financial exchange forecast. In this paper, the creator has put forth an attempt to foresee the cost and value patterns of stocks by applying ideal Long Short Term Memory (LSTM) profound learning calculation and versatile Stock Technical Indicators. To advance the assignment, the creator used the idea of Correlation-Tensor worked with suitable STI's. Ideal LSTM presents choice based pointer Price-rise or Price-fall just as pattern based examination. The proposed Optimal Deep Learning Approach is a market autonomous methodology. The proposed profound learning calculation can likewise be additionally improved to streamline the exhibition.

In this paper, the creator utilized a deep learning technique dependent on Convolutional Neural Network to anticipate the stock costs development of Chinese securities exchanges. The outcomes have shown that it is somewhat solid to utilize profound learning strategies dependent on CNN to foresee the stock value development of China. In this paper, the creator utilized a conv1d capacity to handle the 1D information in the convolutional layer.

In this paper, the fundamental target is to track down the best model to anticipate the worth of the securities exchange. The creator gathered the dataset from the earlier year. At that point, in the wake of preprocessing the information, the creator surveyed the utilization of arbitrary timberland, support vector machine on the dataset and the results it produced. The fruitful expectation of the stock will be an extraordinary resource for the securities exchange organizations and will give genuine answers for the obstructions of the stock financial backers.

Monetary change gauge is basically described as attempting to choose the stock definitely worth and gift an enthusiastic idea for individuals to know and expect the market and the stock expenses. By assessing the accuracy of the various computations, the author found that the most extreme proper estimation for predicting the market cost of a stock ward on unique insights centers from the chronicled measurements is the unpredictable woods estimation. The estimation could be an awesome helpful asset for specialists and money related sponsor for putting mint pieces inside the protections trade since it is prepared on a critical assortment of recorded information and has been picked close to being endeavored on model records.

Stock value expectation is a mainstream and testing assignment and deep learning gives the ideal way to its forecast. In this paper, the undertaking is to foresee the nearby cost for 25 organizations enrolled at the Bucharest Stock Exchange, from a novel informational index presented in the paper. In light of the forecasts, an exchanging system is proposed which advises whether to purchase or sell stocks. The outcome can prompt addition more benefits. Different sorts of intermittent structures, for example, reverberation state organizations, can likewise be attempted. The profound models can be all the more extravagantly defined and gatherings like can be built with conventional Machine Learning procedures.

In this paper, the author clarified High-recurrence exchanging as a strategy for mediation on the monetary business sectors utilizing complex programming and equipment instruments to carry out high-recurrence arrangements, helped by numerical calculations that follow up on business sectors for shares, securities, items, etc. The expectation of the medium-transient pattern is one of the primary issues of the HFT frameworks (ROI) notwithstanding a diminished drawdown or misfortune which makes it monetarily more economical. The creator has precisely approved their pipeline utilizing a total and far reaching dataset.
[8]. Stock exchanging is an anon-zero-sum game with a normal objective arranged that of making more benefit with restricted capital. In this paper, the creator planned the MCN to perform appropriated stock exchanging through the specialists of support learning through joint effort. Likewise, through explore, it was affirmed that the specialists relating to everything recorded high pace of return at rapid. In any case, on account of MCN, there is no insightful portfolio choice technique that chooses the ideal stocks. Accordingly, future investigates ought to be directed to choose the ideal stock among the different stocks.

[9]. In this paper, the creator utilized the popular LSTM method to foresee the stock costs. This will give more precise outcomes when contrasted with existing stock value forecast calculations that existed previously. The point of proposed arrangement is to help financial backers and merchants in their choices when to purchase and sell resources and make more benefits. When contrasted and ARIMA calculation, it is shown that ARIMA calculation comprehends the previous information and doesn't zero in on the occasional part.

[10]. In this paper, the creator proposed a model, called the component combination long momentary memory-convolutional neural organization model, that consolidates highlights gained from various portrayals of a similar information, in particular, stock time arrangement and stock graph pictures, to anticipate stock costs. LSTM and a CNN, which are used for separating worldly highlights and picture highlights. The creator estimates the presentation of the proposed model comparative with those of single models utilizing SPDR S&P 500 ETF information. The model's exactness can be more improved if other data like news story can be utilized. The commitment of this examination is that it is proficient to decrease the forecast mistake by utilizing a blend of worldly and picture highlights from a similar information as opposed to utilizing these highlights independently.

[11]. In this paper, the creator has utilized the essential AI calculations: Linear Regression, Random Forest and Multilayer Perceptron. Securities exchange expectation is significant factor in money. It is viewed as unique in nature. The paper introduced how to anticipate stock qualities dependent on the information of NY Times of 10 years utilizing Machine Learning calculations. The creator additionally reasoned that MLP gave preferred outcomes over Linear Regression and Random Forest on account of its less mistake between the real and anticipated qualities. But since of dynamic nature of stocks, more effective methods are should have been applied to improve results.

[12]. In this paper, the writer efficiently looked into all new stock/forex expectation or exchanging articles that pre-owned support learning as their essential AI strategy. The explored articles considered had some ridiculous suppositions, for example, no exchange costs, no liquidity issues and no offer or ask spread issues. All in all, to track down the best technique, this paper tells that support learning actually requires the examination to fulfill the legitimate needs. Support learning is an expansive and developing field of revenue inside the exchanging of monetary resources on the stock and unfamiliar trade market.

[13]. Foreseeing financial exchange conduct is a space of solid interest for both scholastic specialists and industry professionals the same, as it is both a difficult errand and could prompt expanded benefits. In this work we propose a mechanized exchanging framework that, given the arrival of information in arrangement about an organization, predicts changes in stock costs. The
framework is prepared to foresee the two changes in the stock cost of the organization referenced in the news story and in the comparing stock trade record. [14]. Linear models like AR, ARMA, and ARIMA have been utilized for securities exchange estimating. The solitary issue with these models are, that they turn out just for a specific time frame arrangement information, for example the model recognized for a specific organization will not perform well for another. Because of the dubious and unforeseeable nature of financial exchange, securities exchange estimating faces higher challenge contrasted with different areas. These organizations are unmistakably intended to sidestep the drawn out reliance issue, however recollecting data for quite a while period back is their typical conduct. [15]. Quantifiable methodologies can be used to expect a financial time plan. The customary procedures are autoregressive prohibitive heteroscedastic strategies, and autoregressive move typical or an autoregressive composed move ordinary methods. This examination de-commotions information utilizing an auto encoder with a convolutional occupant neural organization. To make a 1-D convolutional neural organization for succession examination, a solitary neural organization can be joined with a convolutional neural organization with LSTM. [16]. One of different ways to deal with foreseeing future stock costs in the software engineering field is to fabricate man-made brainpower based models which use AI strategies like Neural Network or Reinforcement Learning. At the point when state portrayal is straightforward, the first Q-learning calculation is demonstrated to combine at ideal conduct.

3. Design and Implementation

Stock exchange expectation is the demonstration of attempting to decide the future worth of an organization stock or other monetary instrument exchanged on a trade. There are two costs that are basic for any financial backer to know: the current cost of the speculation the person possesses or plans to claim and its future selling cost. Regardless of this, financial backers are continually assessing past estimating history and utilizing it to impact their future investment choices. A few financial backers will not accepting a stock or record that has risen too forcefully, on the grounds that they accept that it's expected for an adjustment, while different financial backers keep away from a falling stock since they dread it will keep on disintegrating. In this work, describes Deep Learning (DL) methods as a solution to automate this online trading process. We present an algorithm called Deep Reinforcement Learning (DRL) where the agent learns the environment, improves itself based on the rewards earned and takes correct action (makes trades) simultaneously. Applied DRL in stock markets to train a single stock trading agent with the goal of maximizing income in short and long term.

Address this challenge using Deep Q-Network (DQN), Double DQN (DDQN) and Dueling Double DQN (DDDQN) agents in a simulated trading environment. Our model is inspired by off-policy learning and its application in video games. We trained and tested these agents with all S&P 500 stocks and show that just by using price and volume information deep-reinforcement learning agent can make money.

3.1 Explanation of the Algorithm
1. **Deep Q-Network (DQN):**

We define Q(s, a) function such that for given state s and action a, it returns an estimate of a total reward we can achieve should we start at this state. Using the well-known Bellman equation (3) as introduced by (Watkins 1989), a DQN agent chooses an action according to the greedy policy – maximizing the Q* function.

\[
Q^*(st, a) \rightarrow r(st , at ) + \gamma \max_a Q^*(st+1, a) \quad (1)
\]

\[
Q^* = \text{Optimal Q Value}
\]

\[
\gamma = \text{discount factor}
\]

\[
s = \text{state}, \ a = \text{action}, \ r = \text{reward}, \ t = \text{time}
\]

This equation converges to desired Q* given that there are finite number of states and each state-action pair is presented repeatedly. However, this is not possible for market conditions. All state-action pair may not be presented repeatedly or even once in the training section. Thus, the Q function is generalized and approximated with several neural networks – Deep Q-Network (DQN). However, using neural network to represent Q function is unstable (Mnih, Kavukcuoglu and David, et al. 2015). Thus, we use Experience Replay (Lin 1992) to stabilize the training. Here, a gradient descend step is performed with each memory held by the agent. Thus, more and more truth is introduced into the system and causes the system to converge.

2. **Double DQN:**

One problem faced with DQN agent is that the agent tends to overestimate the Q function (Hasselt, Double Q-learning 2010) due to the max in the formula (1). When estimating a Q function for a certain state, the estimate is noisy and differs from the true value. Due to the max function, the action with the highest positive error is selected and propagated to other states. This leads to positive bias overestimation. To solve this issue, we introduced a Double DQN (DDQN) agent. In DDQN, two separate Q functions are independently learned. One network is used to determine the maximizing action while the other estimates its value. By decoupling the maximizing action from its value, we can eliminate the maximization bias. The change in estimation is as follows:

\[
Q^*(st , a) \rightarrow r(st , at ) + \gamma Q'^*(st+1 ', \ \arg \max_a Q^*(st+1 ', \ a')) \quad (2)
\]

\[
Q^* = \text{Optimal Q Value}
\]

\[
\gamma = \text{discount factor}
\]

\[
s = \text{state}, \ a = \text{action}, \ r = \text{reward}, \ t = \text{time}, \ Q' \ & s' \ & a' = \text{target network Q value}, \ \text{state and action}
\]

This helps improve stability of the learning process to learn complicated tasks which in turn translates to improvement in performance (Hasselt, Guez and Silver, Deep Reinforcement Learning with Double Q-Learning 2016).
3. **Dueling Double DQN:**

To further improve the learning process and performance, we also compared our results with a Dueling Double DQN (DDDQN) agent. A DDDQN agent presents a change in the network structure comparing to DQN as follows:

$$Q^*(s_t,a;\beta,\alpha)=\widehat{V}(s_t;\beta)+A^*(s_t,a;\beta,\alpha)−1|A|\sum\Delta a'(s_t,a';\beta,\alpha)$$  (3)

$$Q^*=\text{Optimal Q Value}$$
$$\gamma=\text{discount factor}$$
$$V=\text{Value network, }\beta=\text{parameters specific to Value network}$$
$$A=\text{Advantage network, }\alpha=\text{parameters specific to Advantage network}$$

$s=\text{state}, a=\text{action}, r=\text{reward, } t=\text{time, } Q' & s' & a'=\text{target network Q value, state and action}$

By separating the Q Head into an advantage stream and Value stream, the agent is better able to differentiate actions from one another. This significantly improves learning. Furthermore, in DQN, on each iteration, for each state in the batch, we update the Q-Values only for the action taken in the state. This results in slower learning since Q-values for actions which are not taken are not learned. On the dueling architecture, learning is faster as we start learning the state-value even if just a single action has been taken.

Dueling architecture proves useful in a trading system where the value stream learns to pay attention to the indicator data while the advantage stream learns to pay attention only when there is an action to be taken to reap rewards. This is useful when the market is trending as we want the agent to ignore any price movements during the trend and only exit its position when the trend is over.

Deep Reinforcement learning is difficult to implement and requires a lot of knowledge of Markov Decision Process (MDP). Hence, more efficient methods of DDQN and DDDQN were used which gave nice results.

The algorithms implemented here for forecasting are ARIMA and Holt-Winters and the method used for portfolio optimization is linear programming.

### 3.2 Autoregressive Integrated Moving Average (ARIMA) Algorithm

ARIMA stands for Autoregressive Integrated Moving Average.

**Step 1:**
- Testing and Ensuring Stationarity
- Testing for stationarity
- Differencing

**Step 2:**
- Identification of p and q
- Identifying the p order of AR model
- Identifying the q order of MA model

**Step 3: Estimation and Forecasting**
The algorithm include the following steps:

- The quantity of contrasts d is resolved utilizing rehashed KPSS tests.
- The upsides of p and q are then picked by limiting the AICc in the wake of differencing the information d occasions. Maybe than thinking about each conceivable mix of p and q, the calculation utilizes a stepwise inquiry to navigate the model space.
- The best model (with littlest AICc) is chosen from the accompanying four: ARIMA(2,d,2), ARIMA(0,d,0), ARIMA(1,d,0), ARIMA(0,d,1). Assuming d=0, the steady c is incorporated; on the off chance that d≥1, the consistent c is set to nothing. This is known as the "current model".

Then applying the HOLT-WINTERS Algorithm and Linear Programming.

4. Implementation of Algorithm

The two attributes chosen from the extracted dataset here for analysis are date and open values of the selected stocks.

Create a Data Table to store MAPE values.
Create a list of all the Stocks for looping next.
The next step was creating a list of Pre-processed data frames.
Calculating and storing the MAPE values for Holt Winters and auto ARIMA models.
Selecting the Date and Stock Open values into a data frame for forecasting.
Generating the dates sequentially for the Input Stock.
Merging the sequential dates generated with the dataset.
Imputing the missing values using the Zoo package.
Splitting Data into Test and Train dataset.
Applying the Holt Winters Model on the training dataset.
Forecasting using Holt-Winters.
Then calculate MAPE values for the given stocks.
Assigning the MAPE values to a table called the MAPE table.
Then apply the ARIMA model on the training dataset.
Forecast using Arima .
Then do the MAPE Calculation.
Assigning the MAPE values to a table
Function to Plot the Components
Function to Plot ACF and PACF
Function to plot Holt Winters Forecast
Plotting the forecast using the above model.
Function to plot Arima Forecast
Plotting the forecast using the above model.
Running all the functions for all the stocks for plotting.
Then for optimization, create a table to store the yearly returns for each stock.
Making a list of all the stocks and calculate the mean yearly return for each stock and store the results in the returns table.
Finally for optimization, use linear programming to know how much to allocate/invest in each asset in a portfolio to maximize the returns and/or minimize the risk, based on the return values computed in the previous step. The algorithm thus enabled us to predict optimally which would be the best stock to maximally invest in and then invest the remaining amount in the top 4 best performing stocks based on the return values computed previously.

4.1 Algorithm Implementation

Holt-Winter’s Model Building

Holt Winters_Plot<-function (Stock_DF)
{
  Par (mfrow= c(1,1))
  Stock TS<-ts (Stock_DF[,1], frequency = 365)

  Stock TS_train=ts (Stock TS [1:4741], frequency = 365)
  Stock TS_test=ts (Stock TS [4742:4748], frequency = 365)
  ## Holt Winters model

  Stock_HW<-Holt Winters (Stock TS_train)
  #Forecasting

  Stock_forecast_HW<-forecast. Holt Winters (Stock_HW,h= 7)
  #Plotting the forecast

  plot. forecast (Stock_forecast_HW,shadecols="oldstyle")
}

4.2 Auto ARIMA Model Building

Auto Arima_Plot<-function (Stock_DF)
{
  Par (mfrow= c(1,1))
  Stock TS<-ts (Stock_DF [,1], frequency = 365)
  Stock TS_train=ts (Stock TS [1:4741], frequency = 365)

  Stock TS_test=ts (Stock TS [4742:4748], frequency = 365)
  ## Auto Arima model

  Log ARIMA<-auto.arima (log10(Stock TS_train)
  #Forecasting
Stock_forecast_Arima<-forecast.Arima(Log ARIMA,h= 7)
#Plotting the forecast
plot. Forecast (Stock_forecast_Arima,shadecols ="oldstyle")
}

Table 1: Different agents with Reward and Risk adjusted return

<table>
<thead>
<tr>
<th>Agent</th>
<th>Reward (Total PnL)</th>
<th>Risk Adjusted Return</th>
<th>Avg Holding Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>-81</td>
<td>-0.76</td>
<td>3</td>
</tr>
<tr>
<td>DQN (AAPL)</td>
<td>42</td>
<td>0.8</td>
<td>36</td>
</tr>
<tr>
<td>Double DQN(AAPL)</td>
<td>69</td>
<td>1.02</td>
<td>35</td>
</tr>
<tr>
<td>Dueling-Double DQN(AAPL)</td>
<td>100.51</td>
<td>1.21</td>
<td>16</td>
</tr>
<tr>
<td>Dueling-Double DQN(AAPL)</td>
<td>29</td>
<td>1</td>
<td>24</td>
</tr>
</tbody>
</table>

In machine learning, a cost/loss function measures the model performance. We used training loss to measure the performance of the model. Training loss measures the error on the training data which we want to minimize for optimization. Validation loss measures the loss on a subset of the training data which the model has not used. We want to make the gap between the training testing losses small. An over-fitting model is when the gap between the two is large while an under-fitting model is when the model cannot achieve a low training error (Li 2017).

In machine learning, a cost/loss function measures the model performance. We used training loss to measure the performance of the model. Training loss measures the error on the training data which we want to minimize for optimization. Validation loss measures the loss on a subset of the training data which the model has not used. We want to make the gap between the training testing losses small. An over-fitting model is when the gap between the two is large while an under-fitting model is when the model cannot achieve a low training error (Li 2017).

Comparing our charts, we can see that all three agents are learning (loss is converging, and the reward is increasing). But out of all the best results in terms of losses well reward comes from Dueling Double DQN. We can also observe that the loss in DQN is not quite stable and this is due to the positive bias overestimation.

Once we realized that Dueling Double DQN was the best performing agent, we tested the DDDQN in the second dataset without changing the parameters and training and testing on all
505 stocks for 100 Epochs. As benchmark we compared it to the buy and hold strategy of these individual stocks during the out of sample period. Taking the average across all 505 stocks and taking stocks with number of round trip trades greater than 2 (because the low number of trades makes DDDQN risk adjusted return ratio’s positively skewed).

Table 2: Different Strategies and its approach

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Mean</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy and Hold-risk adjusted return</td>
<td>0.045538</td>
<td>0.046806</td>
</tr>
<tr>
<td>DDDQN-risk adjusted return</td>
<td>1.259177</td>
<td>1.135853</td>
</tr>
</tbody>
</table>

\[
\text{riskadjustedreturn} = \frac{\text{Mean (returns)}}{\text{StandardDeviation (returns)}}
\] (4)

Figure 1: Risk Adjusted Return Distribution

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Mean</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy and Hold total return</td>
<td>17.73 %</td>
<td>17.75%</td>
</tr>
<tr>
<td>DDDQN total return</td>
<td>21.13 %</td>
<td>17.6%</td>
</tr>
</tbody>
</table>
Figure 2: The decomposition of multiplicative time series for Asian Paints

The decomposition of multiplicative time series for Asian Paints has been shown in Figure 2 above; the observed plot and trend suggests that the stock value of Asian Paints started to rise from the year 2009. Some of the key assumptions causing this behavior are:

- UPA retaining its power in 2009 to form Union Government with a very high positive note hoping to increase the infrastructure projects might have helped the Asian Paint’s stock to rise.
- Also, the scope of building constructions increase in India happened in second half of first decade of the millennium (From 2006), which could have helped increasing the stock value of Asian Paints Limited.
- There was also a significant spike in 2014 and a major drop in 2016, but the stock value again picked up rates in the later stages of 2017

ACF and PACF Plots:

Figure 3: the ACF and PACF plots for the stock values of Asian Paints

Figure 3 above portrays the ACF and PACF plots for the stock upsides of Asian Paints. The ACF plot demonstrates that the information is profoundly related with one slack, however most certainly diminishing at an exceptionally lethargic speed which shows that information is non-fixed. In PACF plot, Note that the primary slack worth is measurably huge, while halfway autocorrelations for any remaining slacks are not genuinely critical. This recommends a potential AR(1) model for these information.

Holt-Winters and ARIMA Forecasting:
HDFC BANK PLOTS

The decomposition of multiplicative time series for HDFC Bank has been shown in Figure 6 above. The observed plots and trend plots suggest that the HDFC bank stock prices significantly started to rise post year 2005.

Some of the key assumptions made here are: Plots suggest that year 2008 had increased drastically in the year 2008 as HDFC bank acquired Centurion bank in the same year which is considered as one of the largest mergers in the financial sectors of India.

Increase in the private sectors of India and HDFC Bank being a larger player in supporting those ventures for banking operations which has in turn has increased the customer base of the HDFC bank. This could possibly be one of the reasons why HDFC bank stocks have raised consistently in the last 7-9 years.

Easy allocation of credits to their customer has generated large returns to HDFC bank, which might have also increased their profits base.

ACF and PACF Plots:
Figure 7: The ACF and PACF plots for the stock values of Asian Paints

Figure 7 above portrays the ACF and PACF plots for the stock upsides of Asian Paints. The ACF plot demonstrates that the information is profoundly connected with one slack, however most certainly diminishing at an exceptionally sluggish speed which shows that information is non-fixed. In PACF plot, Note that the some slack qualities are measurably huge, though halfway autocorrelations for any remaining slacks are not genuinely critical. Since there is one negative and positive relationship, this recommends a potential AR (0) model for these information. Holt-Winters and ARIMA Forecasting:

Figure 8: Holt Winter Forecast for HDFC Bank

Figure 9: ARIMA Forecast for HDFC Bank

HINDUSTAN UNILEVER
The decomposition of multiplicative time series for Hindustan Unilever has been shown in Figure 10. The Observed and Trend plots suggest that the Hindustan Unilever Limited stock prices actually performed pretty consistently without any drastic rise or fall down between the years 2004 – 2011, except for the year 2007, when the company was renamed to “Hindustan Unilever Limited” from “Hindustan Lever Limited”.

ACF and PACF Plots:

Figure 11 above portrays the ACF and PACF plots for the stock upsides of Hindustan Unilever. The ACF plot demonstrates that the information is exceptionally related with one slack, however certainly diminishing at an extremely sluggish speed which shows that information is non-fixed. In PACF plot, Note that the some slack qualities are measurably critical, though halfway autocorrelations for any remaining slacks are not genuinely huge. Since there is one negative and positive relationship, this recommends a potential AR (0) model for these information. Holt-Winters and ARIMA Forecasting:
The decomposition of multiplicative time series for Infosys has been shown in Figure 11. The Observed and Trend plots suggest the following things:

It looks like the recession period from second half of 2007 to 2009 had larger effect on stock prices of Infosys and the stock prices were very low. It looks like the change in the leadership in the year 2014, where in for the first time a non-founder member of Infosys was made to lead the office has great impact in the increase of stock prices.

ACF and PACF Plot:
Figure 15: the ACF and PACF plots for the stock values of Infosys

Figure 15 above depicts the ACF and PACF plots for the stock values of Infosys. The ACF plot indicates that the data is highly correlated with one lag, but definitely decreasing at a very slow pace which indicates that data is non-stationary. In PACF plot, Note that the first and second lag values are statistically significant, whereas partial autocorrelations for all other lags are not statistically significant. Since there is one negative and positive correlation, this suggests a possible AR (0) model for these data.

Holt-Winters and ARIMA Forecasting:

Figure 16: Holt-Winters Forecast for Infosys

Figure 17: ARIMA forecast for Infosys

Similarly different algorithms are been tried on L&T, ITC and So on.

Stock Market Prediction:
The error percentages of the each of the stock market prices are relatively less in nature. This could have been reduced to certain extent if we could have trained our model on recent data rather than very old data, as we could have ignored unnecessary seasons, trends or randomness which could have had no significance as of today.

**Portfolio Optimization:**
The Yearly Returns that have been calculated are shown below in Figure 19:

<table>
<thead>
<tr>
<th>Stockname</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASIANPAINT.NS</td>
<td>0.3516342</td>
</tr>
<tr>
<td>HDFC BANK.NS</td>
<td>0.2954772</td>
</tr>
<tr>
<td>HINDUNI.LR.NS</td>
<td>0.1873871</td>
</tr>
<tr>
<td>INFY.NS</td>
<td>0.1644998</td>
</tr>
<tr>
<td>ITC.NS</td>
<td>0.2045008</td>
</tr>
<tr>
<td>LT.NS</td>
<td>0.3444382</td>
</tr>
<tr>
<td>MARUTI.NS</td>
<td>0.3543149</td>
</tr>
<tr>
<td>RELIANCE.NS</td>
<td>0.3037856</td>
</tr>
<tr>
<td>TCS.NS</td>
<td>0.2723172</td>
</tr>
<tr>
<td>KOTAKBANK.NS</td>
<td>0.5154164</td>
</tr>
</tbody>
</table>

**Figure 19:** Table depicting yearly returns

The Solution achieved through Linear Programming to invest say available 6000 INR into the best performing stocks is to invest 2000 INR in Kotak Mahindra Bank and rest 4000 to be invested equally in the remaining 4 best performing stocks chosen here.

5. **Conclusion**
Demonstrated that deep reinforcement learning strategies can successfully learn policies with minimal information to trade in the financial markets. Furthermore, it differentiates itself from supervised learning trading strategies, where predictions need to go through risk and execution models, whereas in reinforcement learning we can put any risk or trading constraints into the rewards structure and for production of the expectation model, different advances like information pre-preparing, order and model assessment was fused as a piece of the execution interaction. The Auto Arima and Holt-Winters model were utilized as a productive methodology for stock expectation, which assisted us with examining the patterns and irregularity from the
decay multiplicative time arrangement plots, ACF and PACF plots to comprehend the information relationship boundaries lastly the figure plots from Holt Winters and Auto Arima model, to count the anticipated and genuine stock qualities. The models were consequently found to perform well, considering the way that all the mistake rates were negligible in nature. In the long run straight writing computer programs was utilized for portfolio advancement, to figure the yearly returns for every one of the stocks subsequent to getting the mistake level of every one of the stock’s determining esteems from the past advance; The solution thus achieved through linear programming for investment of the available 6000 INR into the best performing stocks helped us suggest that 2000 INR could be invested in Kotak Mahindra Bank and the rest 4000 INR to be invested equally in the remaining 4 best performing stocks chosen here, based on the return values.

References


[15]. Yunus YETISl, Halid KAPLAN2, and Mo JAMSHID13, Fellow IEEE, Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, Texas, USA,1 yunusyetis68@hotmail.com ,2 halid88@gmail.com , 3moj@wacong.com, "Stock Market Prediction by Using Artificial Neural Network", World Automation Congress (WAC), 2014, WAC 2014 1569925719

[16]. Chang Sim Vui, Gan Kim Soon, Chin Kim On, and Rayner Alfred, Center of Excellent in Semantic Agents, Universiti Malaysia Sabah, 88400, Sabah, Malaysia, csimvui@yahoo.com, g_k_s967@yahoo.com, kimonchin@ums.edu.my, ralfred@ums.edu.my and Patricia Anthony, Department of Applied Computing, Faculty of Environment, Society and Design, Lincoln University, Christchurch, New Zealand, patricia.anthony@lincoln.ac.nz "A Review of Stock Market Prediction with Artificial Neural Network (ANN)", 2013 IEEE International Conference on Control System, Computing and Engineering, 29 Nov. - 1 Dec. 2013, Penang, Malaysia

[17]. Tao Xing, Yuan Sun, Qian Wang, Guo Yu, International School of Software, Wuhan University, Wuhan, China, e-mail:xingtao@whu.edu.cn, cjchsun@hotmail.com, "The Analysis and Prediction of Stock Price ", 2013 IEEE International Conference on Granular Computing (GrC)
[18]. Rajat Ahuja, UG, BVCOE, New Delhi, INDIA, rajat.ahuja@live.com, Harshil Rastogi, UG, BVCOE, New Delhi, INDIA, harshil9968@gmail.com, Arpita Choudhuri, UG, BVCOE, New Delhi, INDIA, choudhuriarpita@rediffmail.com, Bindu Garg, HOD, BVCOE, New Delhi, INDIA, "Stock Market Forecast Using Sentiment Analysis", 2015 2nd International Conference on Computing for Sustainable Global Development (INDIA Com)

[19]. Sunil Kumar Khatri1, Ayush Srivastava2, 1,2Amity Institute of Information Technology, Amity University, Uttar Pradesh, Noida, India, "Using Sentimental Analysis in Prediction of Stock Market Investment ",2016 5th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Sep. 7-9, 2016, AIIT, Amity University Uttar Pradesh, Noida, India

[20]. Ching-Te Wang, Department of Information Management, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C, ctwang@ncut.edu.tw and Yung-Yu Lin, Department of Information Management, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C, wfjh10602@gmail.com, "The Prediction System for Data Analysis of Stock Market by Using Genetic Algorithm ",2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)

[21]. Manna Majumder1, manna_maj@yahoo.co.in, MD Anwar Hussian2, Professor, Department of Electronics and Communication Engineering, North East Regional Institute of Science and Technology, bubuli_99@yahoo.com, "Forecasting of Indian Stock Market Index using Artificial Neural Network 

[22]. Gabriel Fiol-Roig, Margaret Miro-Julia and Andreau Pere Isern-Deya, "Applying Data Mining Techniques to Stock Market Analysis"

[23]. Robert P. Schumaker and Hsinchun Chen, Artificial Intelligence Lab, Department of Management Information Systems, The University of Arizona, Tucson, Arizona 85721, USA, {rschumak, hchen}@eller.arizona.edu, "Textual Analysis of Stock Market Prediction Using Financial News Articles"

[24]. Vivek Rajput1, Sarika Bobde2, 1,2Department of Computer Engineering, Maharashtra Institute of Technology, Pune, India, rajputvive@gmail.com, sarika.bobde@mitpune.edu.in, "Stock Market Forecasting Techniques: Literature Survey", International Journal of Computer Science and Mobile Computing, Vol.5 Issue.6, June- 2016, pg. 500-506