Tradebull - Reinforcement Learning based Automated Trading

Nisarg Vadher, Email: nisarg.vadher@sjsu.edu
Pavan Bhatt, Email: pavan.bhatt@sjsu.edu
Ujjwal Jain, Email: ujjwal.jain@sjsu.edu
and Saumil Shah, Email: saumil.shah01@sjsu.edu

***Abstract*—Automated Stock trading has always been an ongoing research due to its dynamic and non deterministic nature. Generally, a trader or an entity trading the stock/s rely on time series data and in combination of strategies based on data and past trading experiences. Recently, Machine Learning and algorithmic trading made significant progress in predicting price movements of scrips. However, these approaches have not yielded reliable results as they are based on curated data with limited parameters which results in unreliable results. The data used for predictions can be described under three categories 1. Technical Data (OHLC data and Technical Indicators) 2. Fundamental Data (Earnings Reports) and 3. News Data. Our approach combines all three types of data to analyse its correlation with price movements and to further create strategies using Reinforcement Learning. The strategies are then used to derive out recommendations and prediction on underlying scrips. The outcome provides better insights as of two reasons - Analysis based on combination of three types of data and the principles of Reinforcement Learning which differs from traditional Machine Learning approaches and performs better for this dynamic problem and non-deterministic environment.**

***Index Terms*—Algorithmic trading, Fundamental Data, News Data, Reinforcement Learning, Scrips, Stock Market, Strategy, Technical Data.**

I. Introduction

THE rapid advancement of Artificial Intelligence has spur the development of sophisticated trading strategies and are of great interest among financial institutions and individual traders for its decision making process. A strategy is traditionally created using historical time series data solely and machine learning techniques to optimize it for the risk to reward ratio. However, these approaches may work only for limited scenarios as trading is dynamic in nature and is influenced by many temporary and continuously changing parameters. These parameters are derived out of Fundamental Data comprised of earnings report and News Data. However, as these strategies fail to incorporate the factors obtained by combination of all types of data resulting in unreliable or less rewarding trade decisions.

To overcome these challenges, our solution is based on Reinforcement Learning. Reinforcement Learning is increasingly gaining popularity among researchers because of breakthroughs in Alphafold, AlphaGo and etc. highly complex problems. Deep Reinforcement Learning is used as the core engine in our project for recommendations and price predictions. DRL algorithm works as agents interacting with environment and improving the actions across different states using feedback. The intention is to create an optimized strategy to perform well across the various range of trading constraints and market conditions. The DRL algorithm we have experimented are DDPG (Deep Deteminisric Policy Gradient), PPO (Proximal Policy Optimization) and MODRL(Multi Objective Deep Reinforcement Learning.

II. Project Architecture

A. Problem Statement

The reliability of stock market strategy is low as of dynamic and continuously changing factors affecting a trend. To have a one strategy applicable to wide range of volatile situations is always in research and of a huge interest. The challenge lies in analysing changing and massively big feature set to derive out the factors affecting price change, trend reversals. Generally the dataset considered in algorithmic trading incorporates historical time series data and underlying company's balance sheet/financial reports. However, major shifts in predictions are a result of factors dependent on News Data. There are many solutions which extract out the features using Natural Language Processing but they are not effective when it comes to how they affect the price and other recommendations. Moreover, altering the datasets manually for training is an intervention which can create inconsistencies with results.

To resolve the reliability problems our TR approach works by constant adaptation of dynamic events and incorporating different aspects such as pattern recognition on technical charts, incorporating technical indicators (lagging, leading) and most importantly changing the environment for RL algorithms based on events which aids the decision making process.

The Trend Reversal event approach analysis time series data. The Trend Reversal depicts price changes using intrinsic time, an event based timing system, rather than the fixed and explicit physical time, a point-based timing system based on set time intervals. If the price change between two points meets a explicitly provided threshold value, the system recognizes sudden price movements (i.e. Trend Reversal events). Bullish events are detected when the price change is more than or equal to the defined threshold value, and Bearish events are identified when the price change

is less than or equal to the fixed threshold value[15]. The dynamic threshold definition method that replaces the fixed trend reversal threshold value has been described in [2]. The dynamic threshold applies to financial markets that have set opening and closing periods (tracked by Open High Low Close), such as stock exchanges. The dynamic threshold is a configurable parameter that can detect substantial price fluctuations (i.e., Trend Reversal events) of varying magnitudes in dynamic contexts which also are always changing. The dynamic threshold value is determined by the previous day's price behavior (i.e., the short-term price history) as well as additional data sources (e.g., news releases) [2]. Summarizing, the dynamic threshold is determined by price changes the (n - 1) day (between opening and closing price) and aftermarket (between previous day closing price and current day opening price) which is extended to even track it on month to month basis.

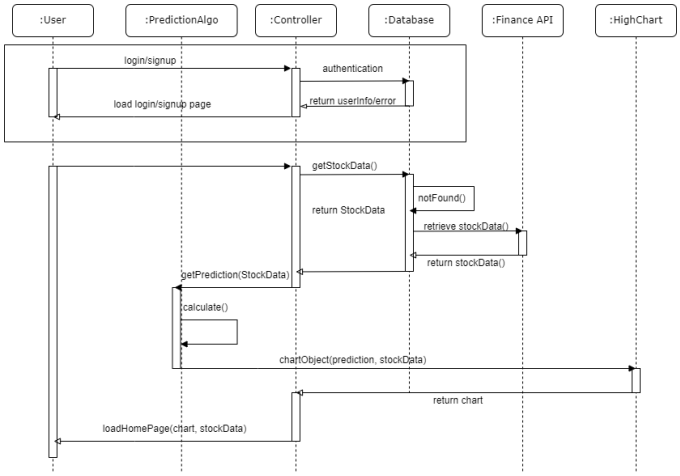


Fig. 1. A sequence diagram explains the set of sequential actions and interaction between different objects or components.

B. System Architecture

TRRL Automated Algorithmic trading consists of two major processing areas. To begin, the Trend Reversal event technique is employed with the dynamic TR threshold to detect and describe the market's environmental conditions. Second, a decision-making algorithm powered by Reinforcement Learning which is employed to make judgments and conduct suitable trading actions. The TRRL algorithmic trading approach is detailed in full in this section. In addition, this algorithmic trading method and its accompanying properties are elaborated. The TRRL algorithmic trading technique is depicted in Algorithm 1.

Algorithm 1 TRRL algorithm

Ps: Price Time-Series stream
for `each incoming Episode` E_t `in Ps` **do**
  Define TR dynamic threshold
  The TR model represents the environment state by S_t
  The RL decides trading action A_t according to environment state S_t
  Calculate reward r_t and evaluation metrics
  Observe new state s_{t+1}
  Store transition (s_t, a_t, r_t, s_{t+1}) in D
end for

III. System Design & Architecture

A. System Design

The prediction engine's primary components are DataSet and the RL algorithm strategy which are described below: The dataset comprises of three types of data extracted from multiple API sources. This data is used to derive out the featureset for model for predictions and recommendations.

1) Technical Data: Technical Data is representation of historical data using basic statistics derived from stock price. It is represented using different type of charts. The most common representation for this time series data is CandleStick chart which is plotted using OHLC candles. The OHLC candles are basic building blocks of representing data which plots Open High Low Close prices for instrument. Based on the time series data and OHLC candles, technical indicators are used over the time which supplements the analysis. Leading and lagging indicators are the two sorts of indicators. A leading indicator is one that predicts the appearance of a reversal or a new trend before it happens. While this seems intriguing, it's important to keep in mind that not all leading signs are reliable. Leading indicators have a bad reputation for sending out erroneous indications. As a result, while employing leading indicators, the trader must be extremely vigilant. In reality, as traders gain expertise, their ability to use leading indicators improves.

Because they oscillate within a restricted range, the majority of leading indicators are termed oscillators. An oscillator often oscillates between two extreme numbers i.e. between 0 and 100 and the trading meaning differs depending on the oscillator's reading (e.g., 25, 75, etc.). An example of one of the most popular leading indicator is Relative Strength Index. The RSI iindicator fluctuates between 0 and 100, indicating the security's intrinsic strength. RSI also sends out the strongest indications when the market is in a sideways or non-trending range. Whereas, a lagging indicator follows the price, signaling the occurrence of a reversal or a new trend after it has occurred. An example of lagging indicator is Moving Average Convergence Divergence. Because of their simplicity and efficiency, moving averages are commonly utilized as trend indicators and are the primary constituents of many popular indicators. Moving averages are generally calculated using closing prices with varying time periods. The time periods are usually 12, 26 or depends on a trader's strategic requirements. The convergence and divergence of the two moving averages is what MACD is all about. Convergence happens when two moving averages move in the same direction, whereas divergence occurs when they move in opposite directions. When the MACD Line crosses

the centerline from negative territory to positive territory, it indicates increasing bullish momentum for buying opportunities, and when the MACD Line crosses the centerline from positive territory to negative territory, it indicates increasing bearish momentum for selling opportunities. Hence, in a similar ways one can use technical indicators which can be overlaid on charts to supplement the trade actions on stock market instruments.

2) Fundamental Data: The potential monetary returns are the primary reason for predicting stock prices. There has been a lot of study done in this field, and various studies have proven that machine learning approaches may be used to correctly predict stocks using past data [5]. Stock market traders have been attempting to tackle the dilemma of lowering risk and maximizing rewards while also forecasting future prices for a long time. There's only a few categories in which stock prediction strategies can be classified and one of them is Fundamental analysis. Fundamental analysis is a way of establishing a stock's essential price by examining economic factors such as quarterly earnings reports, projected growth and profit margins that are made public by the firm, analyst opinions, and news stories that are related to the stock.

Research done by Jaideep et.al. [6] covering the importance of the fundamental analysis along with technical analysis for predicting the stock price accurately. The study shows that several machine learning algorithms, such as Neural Networks, Bayesian Networks, Random Forests, and hybrid models have all been applied in various scenarios to better predict the nature of the markets with a fair degree of success. Regardless of the fact that these machine learning algorithms were successful in predicting stock price patterns up to a point, they were not able to predict accurately to many different scenarios. The reason for this was that none of the machine learning models took into account fundamental analyses, such as a company's annual reports, expert opinions, and general market conditions, in order to reliably predict market dynamics and predict the future stock price for any particular public organization. In conclusion, Extensive testing has revealed that the method has the ability to outperform the SP 500's annual returns by taking technical as well as fundamental analysis into account.

A stock prediction challenge like this necessitates the use of both stock market experience and machine learning algorithm. Because of the multidisciplinary character of our study problem, it's crucial to review the terms that are relevant to quantitative fundamental analysis. The most essential things that can help anticipate the analyst stock price to outperform the market are balance sheets, EPS, P/E ratio, return on capital, income statements, and cash flow statements [7].

The Balance sheet: The Balance sheet shows the total asset of the company, that includes cash, all the equipment that company owns and liability or debt. Asset = equity + Liabilities

The Income statement: The income statement examines a company's success over a certain time period. It indicates how much money was made, how much was spent, and how much profit was made as a result of the company's operations over that time period.

Return on equity: Return on equity (ROE) is a financial measurement that determines how efficiently a firm can spend the money it receives from shareholders. It can be measured by dividing the net income by equity. It is only possible to calculate if both net income and equity are positive.

ROE = net income/equity.

Earning per Share: The earnings per share is computed by dividing the company's net profit by the total number of shares held by investors. Because earnings per share is used to assess a company's profitability, it is one of the most important metrics. It is also used to measure price-to-earnings ratio (P/E).

EPS = (net income - dividends) / Shares outstanding

Price-to-earnings ratio: Analysts, investors, and investment firms use price-to-earnings to estimate a stock's relative value, or how much the market is willing to pay for a particular stock. It determines if a stock is overvalued or undervalued. It is calculated by dividing the current price of the stock by Earning per Share. We've already discussed how to find Earning per share. The formula to get the Price-to-earning ratio is...

(P/E) = price of the stock / earnings per share

Many studies have been conducted in the past to anticipate stock prices using fundamental data taken from the company's website, news stories, and analyst comments. Yuxuan et al. [8] He compared several machine learning models, including the hybrid technique, for predicting long-term stock prices using fundamental analysis. To test and train the machine learning models, he took data from the quarterly financial reports of 70 stocks that appeared in the SP 100 between 1996 and 2017. He then ranked the result of 70 stocks using the model's predicted price and build portfolio according to the rankings. He used the actual earnings from that portfolio to compare and evaluate the accuracy of the model. They concluded that by providing enough fundamental data, all three models they used to predict the price outperformed the market without any comments from analysts.

Quah et al. [9] conducted a study where data on nearly 1600 distinct equities was retrieved from a dataset that spanned ten years, from 1995 to 2004. He has tested the data with three different machine learning algorithms FNN, general growing and pruning radial basis function (GGAP-RBF), ANFIS [4]. This research was not about predicting the stock price but it was about picking the best stock based on fundamental data. Based on Graham's book, Quah chose 11 of the most often used financial ratios as predictors. Quah classified the target variable into two classes instead of training the supervised learning models to accomplish regression. A stock was categorized as "Class 1" if its share price increased by at least 80% in one year; otherwise, it was classified as "Class 2". On the training set, oversampling was utilized solely to avoid data scooping as very few

indices would be able to appreciate more than 80% over the period of one year. The research concluded that FNN and ANFIS model were giving more accurate results compared to GGAP-RBF model.

3) News Data and Natural Language Processing: The predictions of stock prices is an active research domain, therefore in recent years, a lot of efforts have been put into the field to develop a robust and accurate model that is capable of not just predicting individual stock prices but also the trends of the overall market. Besides using technical and fundamental data, the usage of NLP and text analysis has gained a lot of traction as researchers have found a correlation between ongoing events published by journalists and media outlets and a company's stock value.

A study by Alanyali et.al. [10] covering a large dataset of the Financial Times over the period of 6 years showed that the daily references of the company in the articles enhanced the transaction volume for a company shares. This favorable link between daily news mentions of a company and increased stock transaction volume was noticed on the day the item was published as well as the day before.

H. Qu et al. [11] addresses the question of quantifying this correlation between news articles and stock prices. Two metrics are defined across all reported experiments: one to quantify the difference between two groups of media items, and the other to measure the gap between two time periods. They created two 27x27 distance matrices by assessing the correlation between the performance of 27 publicly listed equities and the news stories that accompanied them during a seven-month period. This led to a conclusion that there exists a "Clear, statistically significant, moderate level correlation" between the two.

Abdullah et al. [12] conducted a study to gather crucial information from numerous news sources and analyze it in order to predict stock prices. The researchers suggested a system that combines their text parser and analyzer technology with a free natural language processing (NLP) tool. They were able to extract important information from the news using their text analyzer algorithm (if valid information source) and natural language processing (NLP) (if unauthentic information source). After that, they used machine learning and text mining algorithms to examine the data. Finally, they used past data to forecast stock values. They assigned a ranking/weight to the retrieved data based on a comparison to historical data. They claimed that their research is unique since it incorporates text parsing methods from two independent text data sources (patterned and scattered).

The relation between social media and stock market volatility can now be investigated using data mining and natural language processing tools. Natural language processing, deep learning, and the financial field are inextricably linked. Ni et al. [13] propose a research based on historical pricing and tweet embedding This research proposes a hybrid method to stock market forecasting. Unlike traditional text embedding methods, their method considers both the internal semantic aspects and the external structural qualities of Twitter data, resulting in tweet vectors with more usable information. Through the construction of the tweet node network, they build a Tweet Node algorithm for defining probable connections in Twitter data.Additionally, this method gives Twitter representations emotional properties, which are subsequently fed into a deep learning model based on the long short - term memory and historical stock price.

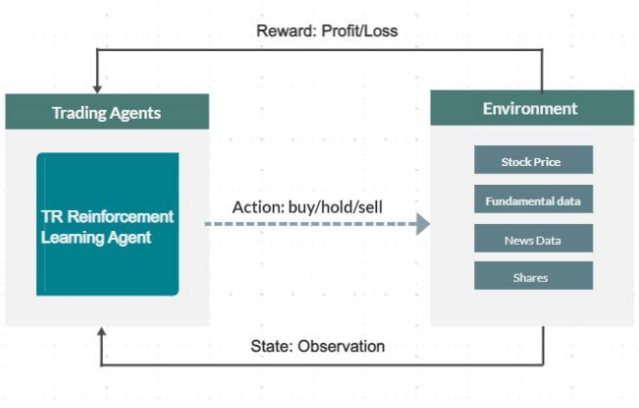


Fig. 2. Overview of reinforcement learning-based stock trading strategy.

Stock price forecasting using time series data has always been difficult.The frequency of searches (on any reputable information source) has been proven to have a considerable effect on stock price movement. Wang et al. [14] proposed a Hybrid time-series prediction neural network model that incorporates news sentiment and news affection. The attributes were retrieved from the news headline and grouped as vectors of scattered words. The dimensions of the vectors were lowered with the use of sparse automatic encoders that deleted unneeded data. By evaluating the news' arrival timeline, they were able to link daily stock data with it. Deep convolutional layers are employed to capture text characteristics, and an LSTM layer is utilized to learn stock price fluctuation in their proposed HTPNN model.

B. TRRL

Reinforcement Learning has - Agent, Environment represented by a set of states s(t) S and set of actions are represented by a(t) A. An agent performs an action at time t and receives reward r(t) which in order transforms from state s(t) to state s(t+1).

1) TRRL States: The fundamental idea behind RL is that the agent continually engages itself with surroundings and discovers the best trading strategy to enhance its investing approach. The existing stock price index and price patterns data, along with a range of basic data and fundamental, technical indicators, make up the business climate for stocks. As a result, choosing the collection of data sources is a requirement for operating agents to understand the environment of the stock market and find trading rules. Identifying market conditions at a particular moment is the fundamental issue of trading the stock market. Commonly used data in

s	a	s'	r(s, a, s')
(AfterMarket Bullish trend)	Sell	$AfterMarket_{t+1'}$ $Previous_{t+1'}$ $Neutral_{t+1}$	Rate of return
(AfterMarket Bearish trend)	Buy	$AfterMarket_{t+1'}$ $Previous_{t+1'}$ $Neutral_{t+1}$	Relative return
(Previous Day, Bullish trend)	Sell	$AfterMarket_{t+1'}$ $Previous_{t+1'}$ $Neutral_{t+1}$	Rate of return
(Previous Day, Bearish trend)	Buy	$AfterMarket_{t+1'}$ $Previous_{t+1'}$ $Neutral_{t+1}$	Relative return
(Neutral Day, Bullish trend)	Hold	$AfterMarket_{t+1'}$ $Previous_{t+1'}$ $Neutral_{t+1}$	0
(Neutral Day, Bearish trend)	Hold	$AfterMarket_{t+1'}$ $Previous_{t+1'}$ $Neutral_{t+1}$	0

the scientific literature on financial forecasting indicate the price series for price time series models at frequent intervals. We used stock index daily data for this analysis.

2) TRRL Actions: Regarding the total number of transacted stocks Qt at time t, two design of experiments restrictions are presumed. First, for the Invest transaction, the trading agent will buy a certain number of shares based on the entire sum of cash the agent has at period t The agent sells each of the shares that are outstanding at time t for something like the Sell transaction. In other sense, while purchasing, the agency invests 100% of its funds, and when disposing, 100% of its stocks. Furthermore, this model has no transaction costs. The intricacy of such investment strategy is lowered to a degree where it can be researched and analyzed inside the parameters of this research by using these oversimplified premises.

3) TRRL Reward Function: The reward function algorithm is crucial when developing trading strategies that focus on the RL approach since an agent created using this algorithm learns the best trading strategy to gain the most profit. Several studies have utilized the Rate of Return as an optimization method in the research on trading stocks. For the TRRL agent, we employed two instant reward criterion in this instance. The Relative Return (RR) is employed in the first criteria, Buy action. Here, the selling and purchasing prices are denoted by pSell and pBuy, accordingly. The RR is defined as the distinction between both the yield attained by the goal time and the overall price gain at period t The Sell action, which employs the RoR, is subject to a second instant reward criteria.

4) TRRL + Q algorithm: Q-learning is an RL algorithm that operates outside of policies and aims to maximize overall reward. In the RL method, quality refers to how successful a decision made at period t was in reaching a certain future reward. When using the Q-learning technique, we build a Q-table or grid which adheres to the policy (s, a) and begin its contents at arbitrary. The Q-values are then updated and saved in the grid for every repetition of the market run. As a result, the Q-matrix serves as a reference vector for agents to choose the best course of action with the highest possible Q-value. The Bellman solution, which the Q-function employs, requires at least two input data: a state (st) and a policy (a) (s, a).

IV. Evaluation Methodology / Materials

In this chapter, we go through a number of tests that were carried out. Using the suggested TRRL (with and without Q-learning) algorithms used in trading, the datasets utilized experimental metrics, benchmarks, and performance evaluation parameters, and the outcomes of trading performance.

We assessed the profitability and efficacy of the suggested TRRL and QTRRL trading methods, as well as the adaptability and effectiveness of the dynamic threshold TR event technique for the RL environment simulation. Finally, we verified the Q-learning algorithm's effectiveness in RL for high frequency trading.

We Used the S&P 500, Dow Jones, NASDAQ stock indices, we even conducted a number of studies to verify the efficacy and reliability of TRRL computational investing. These equities have been acquired again for a time period of July 2015 to July 2020 through Yahoo! Finance (five years) depicts the motion of the 3 stock market indices as well as the development of their respective market curves in depth, while it provides a statistical description of the researched stock market indices, including median, variance, outliers, kurtosis, lowest and highest pricing values.

To ensure that such outcomes are comparable, we performed many separate simulations utilizing the same setup variables and lots of random seeds. This enabled us to confirm the efficacy and precision of the findings.

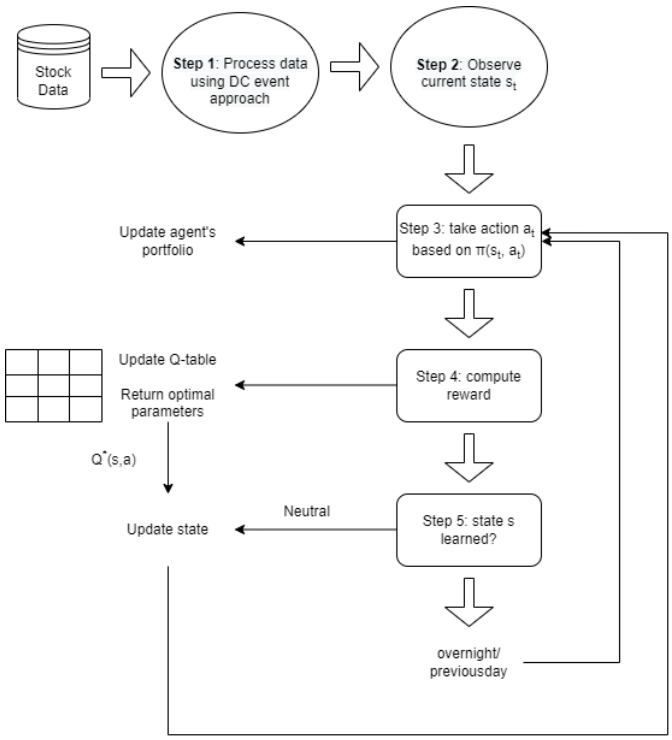
The experimental outcomes are therefore averaged across 20 independent simulation iterations. For the discount factor and learning rate parameters We performed a series of research regarding the procedure in QTRRL to investigate the impact of these two factors on the Q-learning algorithm and QTRRL algorithmic trading's profitability. Here, we looked at discount factor values in the range of 0.85-0.99 and learning rate values of "0.0001, 0.001, 0.05, and 0.1" (those values were in line with other studies). The outcomes of the various investments' Return on Investment (ROI) and Sharpe Ratio (SR).

A. Analysis Metrics

Profit curve, ROI, SR, and the quantity of alerts were the four measures that were utilized to assess the performance and durability of the derivatives market. Stock returns and differences are two performance metrics that are often used in the industry.

The following definitions apply to the four evaluation metrics.

1) The effectiveness of trading methods may be assessed qualitatively using the profit curve. The objective is to



show how the agent's capital profit has changed over time, which might represent the total gain (profit or loss) from the profit margin at each moment in time.

2) The ROI is the inverse of the ratio of net trading gains as well as the expense of trading. It is a statistical performance measure that assesses the profitability of a portfolio. It is straightforward to maximize ROI, which indicates a risk-neutral objective functions.
3) SR is a numerical assessment indicator that quantifies the portfolio's risk-adjusted return for the agent. The SR takes into account an investment's advantages and hazards. As a result, it eliminates the negative impacts of risk variables on the evaluation of trading performance. The SR demonstrates how to match returns to assumed risk as a consequence. An increased SR value in this case denotes an increased risk-adjusted RoR. SRs have been utilized as a performance metric in several research on RL algorithmic trading. The SR, however, penalizes pricing.
4) The quantity of trading signals denotes the quantity of Buy or Sell transaction actions that were carried out. the market session, which is timed to prevent too frequent trading; that would carry a very high risk.

V. Results

Conventional performance measures were performed on 3 stock market indices because the profitability, effectiveness, and efficiency of the suggested TRRL and QTRRL high - frequency trading techniques are our primary concerns. It provides an overview of the outcomes of the quantitative analysis utilizing the ROI and SR.

We extract three key conclusions from the data. First, as opposed to a Direct RL, ZI, and traditional DC predefined threshold trading methods, the "QTRRL and TRRL" investment strategies provide significant returns for the 3 stock indexes. This finding demonstrates the dynamic TR threshold's valuable contribution to the automated trading architecture. This is due to the dynamic TR occurrence approach's summary of price time series trends. For the RL algorithm, these patterns indicate environmental conditions.

The QTRRL analysis results showed performance better than the TRRL trading method without Q-learning, on average. According to ROI and SR, the QTRRL investing strategy typically beat TRRL on the SP500 and Sp Bse stock indices, which suggests the Q-learning algorithm may be able to increase the trading performance that yields good returns while keeping risk at a manageable leve.

Third, the SR values for the QTRRL investment strategy were much higher compared to those of the TRRL, ZI, Direct RL, and the traditional DC, hence validating QTRRL's trading decisions, which secure profit while avoiding danger, particularly whenever the price curve rises quickly. so validating QTRRL's trading decisions, which secure profit while avoiding danger, particularly whenever the price curve rises quickly.

It shows the distribution of trading signals produced by the QTRRL, TRRL, ZI, Direct RL, and traditional DC trading agents for the various stock indexes. As a consequence of the learning process and ability to adapt to market changes, the total amount of trading signal produced by the QTRRL, TRRL, and classic DC trader agents for the 3 stock indices was substantially lower than that for the ZI and Dir. This indicates that the three trading systems developed based just on DC event strategy (TRRL, QTRRL, and classic DC stock trading) are sensitive to the changes in the market.

The TRRL and ZI originally exceeded the QTRRL for the SP500. The learning is then evidently well represented in the QTRRL output, and as a result, it is obvious that the QTRRL has greatly outperformed both TRRL and ZI. The Dow Jones (inside the third chart) exhibits a similar pattern, with ZI and TRRL initially outperforming QTRRL.

The QTRRL, however, greatly outperformed all TRRL and ZI as learning progressed. The same is true for NASDAQ, where training has been shown to work well when combined both RL and TR. Finally, learning significantly improved QTRRL performance, which typically beat that of ZI and TRRL.

VI. Related Works

This section aims to highlight and describe the remarkable work done in the domain of reinforcement learning based stock traders. It also presents the purpose of their studies as well as any shortcomings. Utilizing automated trading based on information gathered from algorithms is widely considered the best method of navigating the stock market [15] and thus the space has become exponentially diverse with academics and enthusiasts alike working towards creating the next generation of agents that provide the best

returns at the lowest possible risk. As such, the space is saturated with price-prediction based methods which try to predict the market trend by using different classifiers [16], [3]. Badr et al. [4] explore the possibility of utilizing a new deep reinforcement learning (DRL) method to build an automated trading agent. Based on the encouragement window policy for automatic stock trading, the advantage function estimates the relative value of state's selected actions. The encouragement window is also based purely on the last rewards, allowing for a less noisy signal. This approach netted good results during the experimental phase over 4 real world stocks, but failed to model the random behaivior generated in high-frequency trading. The concept of utilizing Reinforcement learning to build algorithmic trading agents was also discussed by Aloud et al. [5]. By utilizing a dynamic Directional Change (DC) threshold, they aim to obtain an accurate representation of the environment states, thus introducing the DCRL trading strategy. Furthermore, the DCRL strategy was built upon the Q-learning algorithm to optimize the trading rules (later referred to as QDCRL). By using this approach, they tested the DCRL trading strategy on data from S&P500, NASDAQ, and Dow Jones, for a five years period from 2015-2020. The results showed the DCRL trading strategy generating profits across the 3 stock indices, thus confirming that the dynamic DC threshold was effective in contributing to the RL models performance. They also concluded that the QDCRL trading algorithm performed better than DCRL due to a faster learning process and that the "Q-learning could effectively model the long-term discounted returns".

VII. Conclusion

We have proposed and discussed two algorithm based trading strategies inspired by the TRRL model. With the dynamic TR threshold, we were able to accurately represent the environment states. In addition, the model was able to capture the state of the market efficiently, leading to obtaining positive trading returns in several stock indices.The robustness of the TRRL model was verified using real stock market data, and the results of the experiments demonstrated that the proposed trading model outperformed the ZI and classic RL trading strategies with higher SR values, as well as consistent profit curves. The following provides a summary of our main contributions. We created a straightforward lookup table for RL algorithmic stock trading, we used the dynamic TR threshold event approach to define the environment states in the RL algorithm, and we used the Q-learning algorithm to choose the best course of action in the Natural market state. Due to the dynamic movement of the data in a price time-series, the trained and responsive algorithmic trading need to be retained whenever the environment state changes based on stated preconditions. The dynamic TR threshold event approach-based learning mechanism is efficient in terms of enhancing the states' representation in the market. With the results of the TRRL agents' trading performance (including and non-including QTRRL), we can infer that learning about the environment states is needed to create accurate trading rules and generate high returns. For the TRRL model, we utilized two reward functions. Each of the reward functions is associated to a particular action (buy or sell). The Q-learning matrix's performance showed improved results by utilizing a relative return function for buy actions and a rate of return function for the sell actions. The QTRRL model performed better than the TRRL model for 2 main reasons. The QTRRL model's performance is influenced by the selection of the best trading policy, the learning process for the trading model has a critical role in impacting the performance. Despite this, our results showed that a higher learning rate value is not always the most optimal rate. The difference of the reward and loss in the base state is considered to be an artifact of Q-learning effectively modeling the returns over a long period of time. The agent was also limited to a selected action set based on it's selected policy.This allowed the agent to turn in more trading signals. The study results show that adaptive QTTRL agents display the best results based on profitibility and risk, and could be leveraged in more practical scenarios.

References

[1] M. Aloud, E. Tsang, R. Olsen, and A. Dupuis, "A directional-change events approach for studying financial time series," Econ. Open Access Open Assess. E-J., vol. 6, pp. 1–18, Dec. 2012.

[2] N. Alkhamees and M. Fasli, "Event detection from time-series streams using directional change and dynamic thresholds," in Proc. IEEE Int. Conf. Big Data (Big Data), Boston, MA, USA, Dec. 2017, pp. 1882–1891.

[3] M. Ballings, D. Van den Poel, N. Hespeels, and R. Gryp, "Evaluating multiple classifiers for stock price direction prediction," Expert Systems with Applications, vol. 42, no. 20, pp. 7046–7056, 2015.

[4] B. Hirchoua, B. Ouhbi, and B. Frikh, "Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy," Expert Systems with Applications, vol. 170, p. 114553, 2021.

[5] M. Alanyali, H. S. Moat, and T. Preis, "Quantifying the relationship between financial news and the stock market," Scientific Reports, vol. 3, no. 1, 2013.

[6] H. Qu and D. Kazakov, "Quantifying correlation between financial news and stocks," 2016 IEEE Symposium Series on Computational Intelligence (SSCI), 2016.

[7] S. S. Abdullah, M. S. Rahaman, M. S. Rahman, "Analysis of stock market using text mining and natural language processing," in 2013 International Conference on Informatics, Electronics and Vision (ICIEV), pp. 1–6. IEEE, May 2013.

[8] H. Ni, S. Wang, and P. Cheng, "A hybrid approach for stock trend prediction based on tweets embedding and historical prices," World Wide Web, vol. 24, no. 3, pp. 849–868, 2021.

[9] Y. Wang, H. Liu, Q. Guo, S. Xie and X. Zhang, "Stock Volatility Prediction by Hybrid Neural Network," in IEEE Access, vol. 7, pp. 154524-154534, 2019.

[10] Shen, J., Shafiq, M.O. Short-term stock market price trend prediction using a comprehensive deep learning system. J Big Data 7, 66 (2020). https://doi.org/10.1186/s40537-020-00333-6

[11] M. Alanyali, H. S. Moat, and T. Preis, "Quantifying the relationship between financial news and the stock market," Scientific Reports, vol. 3, no. 1, 2013.

[12] H. Qu and D. Kazakov, "Quantifying correlation between financial news and stocks," 2016 IEEE Symposium Series on Computational Intelligence (SSCI), 2016.

[13] S. S. Abdullah, M. S. Rahaman, M. S. Rahman, "Analysis of stock market using text mining and natural language processing," in 2013 International Conference on Informatics, Electronics and Vision (ICIEV), pp. 1–6. IEEE, May 2013.

[14] H. Ni, S. Wang, and P. Cheng, "A hybrid approach for stock trend prediction based on tweets embedding and historical prices," World Wide Web, vol. 24, no. 3, pp. 849–868, 2021. [15] M. Kearns and Y. Nevmyvaka, "Machine learning for market microstructure and high frequency trading," High Frequency Trading: New Realities for Traders, Markets, and Regulators, 2013.

[15] Y. Deng, Y. Kong, F. Bao, and Q. Dai, "Sparse coding-inspired optimal trading system for hft industry," IEEE Transactions on Industrial Informatics, vol. 11, no. 2, pp. 467–475, 2015.

[16] Baumeister, Roy F. and Mark R. Leary (1995). "The Need to Belong: Desire for Interpersonal Attachments as a Fundamental Human Motivation". *Psychological Bulletin*, 117:497-529.