**Creation of a Stock Trading Bot Based on**

**Q-Learning Algorithm**

Saurabh Jaiswal   
 *School of Computer and Engineering  
Lovely Professional University*Phagwara, Punjab  
saurabhjaiswal151@gmail.com

***Abstract*—** A stock trading bot that can learn to make the best trading decisions based on a Q-learning algorithm is designed and deployed in this research. Historical stock data are incorporated into the equation in order to teach the bot optimal approach to entering, holding, and exiting equity investments through practice trades. The performance of the approach is evaluated in terms of overall profit earned during a testing period together with the total rewards accrued throughout the training episodes. Consequently, the result, based on the Q-learning-based bot, can adopt itself to the alteration in the market, which means the possibility of being a valuable tool for the average trader to increase their profitability in trading activity. It also has strong implications to conclude that the inclusion of complicated methods and approaches can boost the capabilities of the bot in practical trading conditions.

I. INTRODUCTION

A. Background and Motivation

Due to the high instabilities, unpredictability, and sensitivity of feelings and emotions which can influence the decisions of traders the stock market proves to be a special challenge. Such aspects as trader attitude, company’s earnings releases, geopolitical events and economic data influence the market. These factors makes it difficult for individual investors to make rational choices as it is compounded by some of its complexities.

This kind of decision making has been associated with reckless behaviours such over trading and holding a loss making position for long time which can adversely affect the capital says research. Moreover, information asymmetry is often due to a high level of competition that is observed in the stock market: novices are not able to compete with institutional investors who have additional resources and tools for analysis. For this reason, most individual traders always find it hard to remain viable within this trading system.

1. Moreover, discrete shareholders may be overpowered by the available number of observations open for analysis, a condition that may lead to either paralysis or reliance on the help of less than ideal rules of thumb. Solutions that can assist the traders to deals with these problems most efficiently must be automated thus making it essential more and more. It means that large amounts of data have to be analyzed rapidly by such solutions for them to make justifiable decisions.A. Stock Trading Reinforcement Learningbecome overwhelmed by the vast amount of data accessible for examination, which could result in indecision or a dependence on suboptimal heuristics. Automated solutions that can help traders efficiently navigate these issues are therefore becoming more and more necessary. Large volumes of data must be processed quickly by these solutions for them to make defensible conclusions.

A. Stock Trading Reinforcement Learning

One of the popular techniques in almost all sectors including finance is reinforcement learning (RL). When an agent plays some action on its environment and receives the outcome in the form of rewards or punishments, it can learn about its best policies using the Q-learning technique that is a model free method of reinforcement learning. Q-learning is well applied in stock trading because it can learn form experience and develop strategies based on experience with stock trading environment.

For Q-learning to be applied to stock trading as was demonstrated above, it has to be presented in the form of an MDP and where the state is a representation of the status of the market, the actions are the trade actions (to buy, to hold, or to sell) and the reward is the outcomes of the actions. Incorporating previous statistics, Q-learning manages to fine-tune its trading approach in a way that options are made dependent on previous results and expected futures. The reinforcement learning framework is a feasible approach to dealing with the complexities of stock trading because it allows the agent to look for various strategies and adjust their course when necessary.

The Reasons Behind Automated Trading Automation

In fact, automated trading system has numerous benefits for individual investors as shown below. They allow the testing of strategies with historical data, increase the rate of making decisions, and reduce the decisions’ emotional component. One of the biggest advantages of trading is that investors eliminate the need for having to monitor markets very frequently, thus being time saving and reduces stress.

Also, it is getting possible to compromise trading systems with artificial intelligence. It was also established that by applying machine learning techniques, investors may also be closer to improving their procedures for decision making and, at the same time, their financial performance. A bilateral trading that involves the third party is that automated trading systems that incorporate intelligent contracts and numerical predictors such as Q-learning, can enhance the performance of the investors in the financial markets in the future as the advancement in technology continues to rise.

# In addition, there is probably the necessity of enhanced trading strategies due to a rapid increase in the role of computer programs for trading. Q-learning can be used to develop systems whereby traders do not solely wait for market conditions to develop then make their trading decisions. The purpose of developing such systems is to provide equal opportunities in the stock market by providing it to common investors which would have been available only to certain institutional users.

# **APPROACH**

1. Configuring the environment

Overview of the StockTradingEnv Class

The Q-learning bot uses StockTradingEnv class to simulate trading environment where the bot will operate. Everything which the trading agent will require to run optimally is provided for within this class. To enable the agent to learn and modify the approach, the architecture of this environment is perfect. Thus, by basing it on a number of factors that define behavior in the market, it emulates trading.

Representation of the State: One must pay an equal attention to the variables used to describe the context of environment and the most significant of these variables are stock closing price and trading volume. All the above are good indicators that help to determine the current state of the market. Because the continuous state space cannot be learned efficiently when effective meaningfully as just discussed the variables need to be discretized into bins.

For example, the agent can easily sort its observations by partitioning the closing price into quantiles compatible with different price levels.

Visual: Finally, there is a knowledge representation diagram illustrating how discretization operates to transform continuous data into exact states: bins for prices and volumes. This figure makes the structure of state space clear Moreover it explains why the continuous variables have been binned.

Three actions are available to the agent in the action space: purchase, hold, and sell. In particular, every move involves different impacts on the trading plan. For instance, purchase increases the share owned while sale disburses those shares. With the help of the hold action it is made possible to let the agent wait for specific conditions of the market before having decided upon some particular action.

This is because the action space determines the possible results of the actions the agent can take, and, in particular depends on how it is designed to shape the trading behavior of the agent.

Incentive Structure: The training of the agent varies with the incentives system to be implemented. The rewards are calculated using the trading account balance together with the value of the cash and the stock owned. Hence, the cumulative amount of reward over the course of the game is to be optimized. Moreover, it is also constructive to define the incentive function to be negative for unwanted behavior such as selling at a lost or incure high transaction cost. The signals that the reward function sends out must ensure that both, the consistent and high-value roll claims, as well as the excessive risks, are outweighed. The Algorithm for Q-learning

Important Details Described

The Q-learning algorithm uses a number of crucial parameters that have a big impact on how well it works:

• Alpha (rate of learning): The level of Pent up new information that displaces old knowledge is governed by this parameter. Higher learning rate could be obtained if a larger value of alpha is used; however, the agent might developed instability due to over reliance on last observations. But a smaller alpha term means slower learning which results to slower convergence however it might converge than the previous solution. The learning rate is a very important parameter since it determines how successful the Liepinė agent is when adapting to the change in the market environment.

Gamma (Discount Factor): This measure demonstrates the extent to which future increases are of great importance. These advantages are prioritized by a gamma value about 1 so as to help the agent consider long-term impacts of the decision to be made. A gamma value close to zero, on the other hand, points the agent’s attention towards rewards in the current period and thus can lead to short-sighted behaviour. Gamma decision represents the agent’s attitude to the relationship between risk and reward.

Exploration Rate (Epsilon): This option is in the middle of the exploitation and exploration. A high epsilon ensures the optimization of the action space and therefore provides a chance to search for more profitable strategies. However, epsilon is continually reduced in training to allow the agent to make optimum use of recognized rewards, where the acquisition densities could make substantial profits out of acquired tactics.

Probability is used in the beginning phases of training so that the agent can test its different variables but must reduce over time to ensure the agent comes to choose the best way.

2.2.2 Algorithm for Action Selection   
Exploration and exploitation are involved in the tradeoff on actions to make by the agent. In the case that a random value is lower than epsilon, the agent randomly selects an action to examine. If not, it selects the action with the greatest Q value of the present state, making use of what it currently is aware. The described method helps the agent to be adaptive to the market conditions and learn as well.

In this case the agent achieves the right balance of exploration and exploitation in order for him to be successful.

Exploration ensures the agent is able to search for new strategies, and or create techniques to fit the new market whereas, exploitation concerns itself with ensuring that the agent maximizes on the information that it has accumulated. Based on these two components, the following element constitutes the base of the Q-learning process.

**Implementation**

1. Code Structure Overview

The proposed Q-learning based stock trading bot can be decomposed into several components where each component handles the different aspects of the system. Due to its flexibility, modularity is used as the primary approach to the design of the application architecture.

Environment Initialization: The trading environment is created with an object of the StockTradingEnv class which defines the state space, action space and reward. Finally, the historical stock data is loaded and pre-processing environment is set in such a way that it permits training and testing phase.

Training Loop: The training loop is also the most fundamental concept of the learning process since the agent works with the environment. In each case, the agent resets its state and loops through action choices until the episode is over (e.g. when all the data points are through). The agent gets the award and update the Q-values by using the update rule that is explained in the previous part.

Testing Phase: In the study, after training the agent for the task, it undergoes a testing phase in order to; measure its efficiency. In this stage, the agent applies the acquired policy for trading consequent to the historical data. Employment decisions are tested on the autosomes with a view of assessing the efficiency of trained policy.B. Important aspects of Q-learning Process stock trading bot is organized into several key components, each responsible for different aspects of the overall system. A modular design approach is employed to facilitate maintainability and scalability.

Environment Initialization: The trading environment is initialized by creating an instance of the StockTradingEnv class, which sets up the state space, action space, and reward structure. Historical stock data is loaded, and the environment is configured to allow for training and testing phases.

Training Loop: The training loop is the core of the learning process, where the agent interacts with the environment. For each episode, the agent initializes its state and continues to select actions until the episode ends (e.g., when all data points have been processed). The agent receives rewards and updates its Q-values according to the update rule discussed previously.

Testing Phase: After training, the agent is evaluated in a separate testing phase to assess its performance. During this phase, the agent employs its learned policy to make trading decisions based on the same historical data. The autosomes of these decisions are analyzed to determine the effectiveness of the trained policy.

B. Key Highlights of the Q-learning Process

The key highlights of the Q-learning process include the following:

Dynamic Adaptation: One of the most important of them is the possibility for the agent to change the strategy of trade according to the current conditions on the stock exchange. Over time, the agent gain knowledge more and more from the experience with environment and thus, gets equally good at finding the best trading signals.

Backtesting Capabilities: The use of data to test the efficiency of the agent means that there is genuinely quantifiable assessment of how effective the agent is. This feature plays an important role of evaluating the feasibility of the trading strategy before putting it to practical use in the trading floor.

Continuous Improvement:The reinforcement learning framework is useful for implementation in view of this consideration as the agent refines strategy over time. In comparison to similar products, this agent can improve its decision-making process in response to the changing market conditions and with the incoming data.

# **Results**

## Performance metrics Overview

The evaluation of the Q-learning-based stock trading bot relies on several key performance metrics that provide insights into its trading effectiveness:

Cumulative Rewards: Accumulated reward that the agent gathers during the training episodes reveals the exploitation of the returns in the long run. They signal that the agent effectively responded to market characteristics and traded profitably as there are more numerous and bigger rewards over time.

Final Profit: The profit value determined from the final simulation run during testing phase provides direct means of assessing the performance of the agent. This metric can be an actual result of this agent’s trading strategy and determine its viability in practice.

Win Rate: The parameter based on which successful trades are measured through the win rate enables the determination of the competency level of the agent. A high win rate implies that the agent is making the right assumption about movement of the market.

Sharpe Ratio: The Sharpe Ratio is applied to the trading strategy to identify the level of risk-adjusted returns; helps understand the proportion of returns relating to the risk. The Sharpe Ratio, unlike the Growth-Value model, provides information about the risk-return trade-off, that is a higher Sharpe Ratio means that an agent is obtaining better returns given the risks taken.

B. Analysis of the outcomes attested during the training phase

The outcomes of the training phase show the capacity of the agent in correctly acquiring and applying an adequate trading strategy. These cases are demonstrated in the cumulative rewards where the number of the agent’s rewards raises gradually over the time, providing evidence of learning. After analyzing the records, the agent becomes aware of the beneficial patterns and alters its behavior.

The win rate analysis shows that the agent was profitable with many trades as anticipated which strengthens the argument about the agent’s trading ability. Similarly, the Sharpe Ratio also corroborates the effectiveness of the employed Q-learning strategy by showing that the agent obtains high returns with the amount of risk taken.

C. Evaluation of Testing Phase Performance

At this stage performance is compared with previous data for the purpose of analysis in real life situations to assess the effectiveness of the agent. The last profit made by the bot proves how it can earn profit in the market, thus proving the potential of automated trading system to improved individual investor performance.

This has a win rate like that obtained during the training phase and shows that the agent trades accurately in unseen data. The Sharpe Ratio has provided an understanding of how much better the Q-learning bot is than regular trading strategies, the effectiveness of the reinforcement learning for applying in trading scenarios.

# **Discussion**

1. Effectiveness of the Q-learning Approach

This is evident based on the outcomes of the observed Q-learning approach necessary in enhancing trading decisions of the system within the environment. This is a strength of the agent because applying the learning method from historical data and amending techniques are beneficial in the rapidly changing environment of stock trading. It further optimizes the bot’s decision-making because it is constantly learning from the marketplace thus improving the trading result.

Also, the Q-learning bot that is applied at the given task does not possess any emotions that could be overwhelming a trader, unlike human emotions that affect traders in their work. This aspect is mostly important especially during unstable market environments where factors such as phobia and euphoria play large roles.

1. Limitations and Challenges

Despite the encouraging outcomes, a few restrictions and difficulties need to be resolved:

Transaction Costs: Obviously, omitted currently from the model are transaction costs which significantly affect the profitability, especially for the high-frequency trading strategies. Future research should incorporate the transaction cost factors that were reviewed in this paper to provide more refined evaluation of the performance.

Hyperparameter Selection: The agent learning process must also have suitable hyperparameters choices which include the discount factor and the learning rate. It can be hypothesized that these parameters should be fine-tuned and optimized to improve performance.

Market Shifts: The stock market is a dynamic field because its condition changes over time. There is still a problem in transferring the methods learned to other conditions in the market. The immediate supervisor’s ability to adapt to changes in the marketplace may be the area of focus in future research.

Data Limitations: A potential limitation of the agent is related to the quality and quantity of previous data used for learning. All these mean that for successful development of trading techniques, the training data needs to present a wide range of market conditions.

Future Research Directions

The following directions might be investigated in future research to improve the Q-learning bot's functionality:

Integration of Advanced Algorithms: It can be suggested that application of other deep reinforcement learning techniques like Deep Q-Network (DQN) might be beneficial for the agent to improve the decision making and deal with more complex state representations. To optimize learning in the high inventorial state spaces DQNs employ Deep Neural Networks for Q-values approximation,

Sentiment analysis integration: Computing feelings of traders and reading trends on the market would be possible if news article analysis for sentimental analysis with the help of NLP could be used. Originally, moving an extra layer of data may complement the trading strategy as the decision maker incorporates the real-time market sentiment.

Investigation of Multi-Asset Trading: Expanding on the ability of the bot to execute many assets at once can improve the return on the risk and diversify the portfolio. The agent may be able to identify asset pair relationships and capitalize on them in ways wherein it would be difficult for a single asset trader to do so because of using a multi-asset trading strategy.

[1] Mnih, V., Kavukcuoglu, The theory of reinforcement learning provides a normative account[1](https://www.nature.com/articles/nature14236#ref-CR1), deeply rooted in psychological[2](https://www.nature.com/articles/nature14236#ref-CR2) and neuroscientific[3](https://www.nature.com/articles/nature14236#ref-CR3) perspectives on animal behaviour, of how agents may optimize their control of an environment. To use reinforcement learning successfully in situations approaching real-world complexity, however, agents are confronted with a difficult task: they must derive efficient representations of the environment from high-dimensional sensory inputs, and use these to generalize past experience to new situations. Remarkably, humans and other animals seem to solve this problem through a harmonious combination of reinforcement learning and hierarchical sensory processing systems , the former evidenced by a wealth of neural data revealing notable parallels between the phasic signals emitted by dopaminergic neurons and temporal difference reinforcement learning algorithms.

[2] Li, X., Jiang, P., Chen, The discretisation of an attribute refers to partitioning its continuous numerical values into intervals, each of which is associated a categorical label. The amount of such different categorical labels is called as target discretisation level of the continuous attribute. For data mining algorithms that can only work on discrete data, the discretisation will be necessary. At the same time, the discretisation can also make the original data more concise and interpretable. However, it is challenging to balance the target discretisation level and the information loss during the discretisation process.

[3] Moody, J., & Saffell, M., We present methods for optimizing portfolios, asset allocations, and trading systems based on direct reinforcement (DR). In this approach, investment decision-making is viewed as a stochastic control problem, and strategies are discovered directly. We present an adaptive algorithm called recurrent reinforcement learning (RRL) for discovering investment policies. The need to build forecasting models is eliminated, and better trading performance is obtained. The direct reinforcement approach differs from dynamic programming and reinforcement algorithms such as TD-learning and Q-learning, which attempt to estimate a value function for the control problem. We find that the RRL direct reinforcement framework enables a simpler problem representation, avoids Bellman's curse of dimensionality and offers compelling advantages in efficiency.

[4] Deng, Y., Bao, F., Kong, Y., Can we train the computer to beat experienced traders for financial assert trading? In this paper, we try to address this challenge by introducing a recurrent deep neural network (NN) for real-time financial signal representation and trading. Our model is inspired by two biological-related learning concepts of deep learning (DL) and reinforcement learning (RL). In the framework, the DL part automatically senses the dynamic market condition for informative feature learning. Then, the RL module interacts with deep representations and makes trading decisions to accumulate the ultimate rewards in an unknown environment. The learning system is implemented in a complex NN that exhibits both the deep and recurrent structures.

[5] Zhang, Y., Zohren, S., We adopt Deep Reinforcement Learning algorithms to design trading strategies for continuous futures contracts. Both discrete and continuous action spaces are considered and volatility scaling is incorporated to create reward functions which scale trade positions based on market volatility. We test our algorithms on the 50 most liquid futures contracts from 2011 to 2019, and investigate how performance varies across different asset classes including commodities, equity indices, fixed income and FX markets. We compare our algorithms against classical time series momentum strategies, and show that our method outperforms such baseline models, delivering positive profits despite heavy transaction costs. The experiments show that the proposed algorithms can follow large market trends without changing positions and can also scale down, or hold, through consolidation periods.

[6] Xiong, Z., Liu, X. Y., Zhong, Stock trading strategy plays a crucial role in investment companies. However, it is challenging to obtain optimal strategy in the complex and dynamic stock market. We explore the potential of deep reinforcement learning to optimize stock trading strategy and thus maximize investment return. 30 stocks are selected as our trading stocks and their daily prices are used as the training and trading market environment. We train a deep reinforcement learning agent and obtain an adaptive trading strategy. The agent's performance is evaluated and compared with Dow Jones Industrial Average and the traditional min-variance portfolio allocation strategy. The proposed deep reinforcement learning approach is shown to outperform the two baselines in terms of both the Sharpe ratio and cumulative returns.

[7] Wang, X., Zhou, Y., Zhang, Lung nodule proposals generation is the primary step of lung nodule detection and has received much attention in recent years. In this paper, we first construct a model of 3-dimension Convolutional Neural Network (3D CNN) to generate lung nodule proposals, which can achieve the state-of-the-art performance. Then, we analyze a series of key problems concerning the training performance and efficiency. Firstly, we train the 3D CNN model with data in different resolutions and find out that models trained by high resolution input data achieve better lung nodule proposals generation performances especially for nodules in too small sizes, while consumes much more memory at the same time. Then, we analyze the memory consumptions on different platforms and the experimental results indicate that CPU architecture can provide us with larger memory and enables us to explore more possibilities of 3D applications. We implement the 3D CNN model on CPU platform and propose an Intel Extended-Caffe framework which supports many highly-efficient 3D computations, which is opened source at https://github.com/extendedcaffe/extended-caffe.

[8] Hu, D., & Wellman, M. P. Reinforcement learning has gained attention and extensive study in recent years [5, 12]. As a learning method that does not need a model of its environment and can be used online, reinforcement learning is wellsuited for multiagent systems, where agents know little about other agents, and the environment changes during learning. Applications of reinforcement learning in multiagent systems include soccer [1], pursuit games [14, 3] and coordination games [2]. In most of these systems, single-agent reinforcement learning methods are applied without much modication. Such approach treats other agents in the system as a part of the environment, ignoring the di  
erence between responsive agents and passive environment. In this paper, we propose that a multiagent reinforcement learning method should explicitly take other agents into account. We also propose that a new framework is needed for multiagent reinforcement learning.

[9] Tsantekidis, A., Passalis, Along with the development of asset pricing model, there has been a debate that nobody can defeat market. Efficient Market Hypothesis argues that stock prices havefully reflected all information, so trying to beat market is futile in an efficient market(Fama, 1970). Then, a lot of market anomalies had been discovered on empirical studyof efficient market. Based on Fama’s hypothesis, it is observed that most of market arenot enough efficient, it means that investors can get excess return by trading, and marketalso can be defeated. Then Adaptive Market Hypothesis had been proposed to refinethe theoretical divergence in financial markets (Lo, 2005), based on this hypothesis, theopportunities for investors to get excess returns had been found in Chinese stock market(ZHOU and SONG, 2017)   
*(PDF) Deep reinforcement learning for portfolio management*. Available from: <https://www.researchgate.net/publication/347965298_Deep_reinforcement_learning_for_portfolio_management> [accessed Nov 16 2024].

[10] Yang, Y., Huang, K., He, Portfolio optimization is an important financial task that has received widespread attention in the field of artificial intelligence. In this paper, a novel deep portfolio optimization (DPO) framework was proposed, combining deep learning and reinforcement learning with modern portfolio theory. DPO not only has the advantages of machine learning methods in investment decision-making, but also retains the essence of modern portfolio theory in portfolio optimization. Additionaly, it was crucial to simultaneously consider the time series and complex asset correlations of financial market information. Therefore, in order to improve DPO performance, features of assets information were extracted and fused.

CONCLUSION AND FUTURE WORK

This paper demonstrates how reinforcement learning, in this case through the creation of a stock trading bot that uses Q-learning, may be utilized to optimize trading options. Specifically for individual investors the ability to learn from past data and adapt to specific market situations of the bot offers valuable information. The results obtained for the training and testing phases indicate the effectiveness and automatability of the Q-learning based system as the solution for improving the trading performance.

An increasing amount of trades’ strategies will be changed by using technological advancement, including artificial intelligence and machine learning. How these technologies might change the specific trading strategies is an example of Q-learning bot.

There are still areas to enhance the performance of the bot; for the next research, it is making necessary alterations in the incentive structure, including more relevant market parameters, and analyzing advanced reinforcement learning techniques. The ever-evolving systems of automated trading in the context of an ever fluctuating stock market can further the cause of the everyday investor and enhance their financial results.

##### **References**

[1] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533. https://doi.org/10.1038/nature14236

[2] Li, X., Jiang, P., Chen, T., & Zhu, S. (2019). A deep reinforcement learning approach for automated stock trading. *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, 5985–5992. https://doi.org/10.1109/BigData47090.2019.9006513

[3] Moody, J., & Saffell, M. (2001). Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks, 12*(4), 875–889. https://doi.org/10.1109/72.935097

[4] Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2017). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems, 28*(3), 653–664. https://doi.org/10.1109/TNNLS.2016.2522401

[5] Zhang, Y., Zohren, S., & Roberts, S. J. (2020). Deep reinforcement learning for trading. *The Journal of Financial Data Science, 2*(2), 25–40. https://doi.org/10.3905/jfds.2020.1.017

[6] Xiong, Z., Liu, X. Y., Zhong, S., Walid, A., Hooi, B., & Cheng, P. (2018). Practical deep reinforcement learning approach for stock trading. *arXiv preprint arXiv:1811.07522.*

[7] Wang, X., Zhou, Y., Zhang, Y., Wang, Q., & Yang, J. (2018). Stock trading strategy design based on deep reinforcement learning and multi-agent system. *Procedia Computer Science, 131*, 155–162. https://doi.org/10.1016/j.procs.2018.04.168

[8] Hu, D., & Wellman, M. P. (1998). Multiagent reinforcement learning: Theoretical framework and an algorithm. *Proceedings of the Fifteenth International Conference on Machine Learning*, 242–250.

[9] Tsantekidis, A., Passalis, N., Tefas, A., Kalliatakis, G., & Dimoulas, C. (2020). Using deep reinforcement learning for financial portfolio management. *Machine Learning and Applications: An International Journal, 7*(1), 29–43. https://doi.org/10.5121/mlaij.2020.7103

[10] Yang, Y., Huang, K., He, Q., & Jin, J. (2021). Reinforcement learning-based stock portfolio optimization. *Journal of Computational Finance, 25*(1), 1–24. https://doi.org/10.21314/JCF.2021.408