How Machine Learning Can Drive High Frequency Algorithmic Trading for Technology Stocks

Mesbaul Haque Sazu
Case Western Reserve University, Cleveland, Ohio, USA.
Email: mesbasazu@gmail.com
Corresponding author email: mesbasazu@gmail.com

Abstract- The objective of this paper is to present an innovative method, based on deep machine learning (DRL), to resolve the algorithmic trading issue of figuring out the perfect trading place at any time during a trading activity on the stock market. It presents a new DRL trading policy to maximize the Sharpe ratio performance indicator over a wide range of stock markets. Named the Trading Deep Q-Network algorithm (TDN), the famous DQN algorithm influences this new DRL approach and considerably adapts to the particular algorithmic trading issue in front of us. Training of the ensuing machine learning (RL) agent is completely based on the development of artificial trajectories from a small set of historical data on the stock market. The paper additionally proposes a new, much more rigorous performance assessment method to objectively evaluate the performance of trading methods. Promising results for the TDN algorithm are reported adhering to this new approach to performance evaluation.

Keywords: **machine learning; high frequency; algorithmic trading; finance; tech stocks**

1. Introduction

Interest in artificial intelligence (AI) has grown exponentially over the past couple of years, with new research papers published each year. Key to this increasing interest are the impressive accomplishments of deep learning (DL) techniques based on serious neural networks (DNN) mathematical models inspired directly by human brain structure. Today, these specific techniques are at the top of the line in many applications, including speech recognition, natural language or image classification processing. Learning by deep machine (DRL) has recently received significant interest in another field of research closely related to DL. This particular group of techniques is focused on an intelligent agent's learning process (i) interacting sequentially with an unknown environment (ii) aiming to maximize its snowball rewards, and (iii) using DL strategies to generalize the information gathered from the interaction with environment. Recent accomplishments of DRL methods highlight their power to resolve complex sequential decision problems.

Financial technology industry is now referred to as FinTech, an emerging industry that has been growing rapidly. The goal of FinTech is very simple: To make the most of the technology to innovate and enhance finance activities. The FinTech industry is likely to revolutionize the way many decision-making issues associated with the financial sector are dealt with in the coming years, including trading, portfolio management, risk management, investment, fraud detection and financial advising. Solving such complex decision-making problems is extremely difficult, because they are generally sequential and stochastic, with an environment that is partly observable and potentially adversarial. The challenge of algorithmic trading is particularly interesting in the fin-tech industry. Algorithmic trading, also known as quantitative trading, is the technique to trade with computers and mathematical rules.

The main objective of this research paper is to answer the following question: How can we design an algorithmic trading policy based on AI methods, which can compete with the well known algorithmic trading strategies commonly used in practice? This article presents and analyzes a new DRL solution to address the algorithmic trading issue of determining the perfect trading position (long or short) at any time during a trading activity on the stock market. The algorithmic solution provided in this research paper is influenced by the well known Deep Q-Network (DQN) algorithm adapted to the specific sequential decision-making issue at hand. The relevant research question becomes even more relevant because the trading environment offers different characteristics from those effectively solved by DRL methods, mainly due to extremely poor observability and significant stochasticity.

This research paper is organized in the following way. To begin with, in Section 2, a brief survey of the scientific literature surrounding the algorithmic trading area and its main AI contributions is presented. Section 3 then introduces the specific algorithmic trading issue considered, and rigorously formalizes it. This particular section additionally links with the machine learning (RL) approach. Section 4 then focuses on the full design of TDN's trading approach based on DRL concepts. Section 5 provides a new method to objectively evaluate the performance of trading techniques. The section 6 deals with the presentation, as well as discussion of the outcomes achieved by the TDN trading approach. Section 7 talks about intriguing leads in future work and draws significant conclusions to conclude this research paper.

2. Literature Review

To start this short literature review, 2 facts should be emphasized. For starters, it's essential to be mindful that lots of sound medical works in algorithmic trading aren't publicly accessible. As defined in Li (2017), personal FinTech firms are unlikely to build their newest investigation results public. Secondly, it must be acknowledged that creating a good comparison between trading techniques is a difficult task, due to the absence of the same, well established framework to effectively evaluate their performance. Instead, the authors usually define their own framework with the evident bias of theirs. Another serious problem is the trading costs, which are variously defined and even omitted.

To begin with, almost all the works in algorithmic trading are methods created by mathematicians, traders and economists that don't exploit AI. Typical examples of classical trading techniques are definitely the trend observing and mean reversion methods, which are discussed in detail in Chan (2009), Chan (2013 Narang and) (2009). Next, most works using machine learning (ML) methods in the algorithmic trading field focus on forecasting. If the financial industry evolution is thought ahead of time thanks to a fair degree of self-confidence, the perfect trading decisions can effortlessly be computed. Following the approach, DL methods have actually been examined with good results, see e.g. Ar´evalo et al. (2016) introducing a trading approach based on a DNN, and specifically Bao et al. (2017) using wavelet transforms, stacked auto encoders and also extended short-term memory (LSTM). Alternatively, a few writers have investigated RL strategies to resolve this particular algorithmic trading condition. For example, Saffell and Moody (2001) created a recurrent RL algorithm for finding new investment policies without the want to construct forecasting Leemans, Dempster, and models (2006) used adaptive RL to exchange international exchange marketplaces. Much more recently, several works investigated DRL methods in a scientifically sound method to fix this specific algorithmic trading problem. For example, one will primarily point out that unveiled the fuzzy recurrent deep neural network structure to attain a technical indicator free trading system using fuzzy learning to lessen time series uncertainty. One can also mention Carapu¸co et al. (2018), which analyzed the use of the full Q learning algorithm for trading in international exchange marketplaces. Lastly, there are several interesting works checking out the use of DRL strategies to algorithmic trading in certain markets, like in the area of electricity, see e.g. the content Boukas et al. (2020).

To end with this brief literature review, a sensitive issue in the medical literature will be the habit to prioritize the interaction of good or maybe results, occasionally in the price of a good logical approach with unbiased criticism. Going a lot more, Ioannidis (2005) actually says most published research findings in a few vulnerable fields are likely false. Such problem seems even more pertinent in the field of monetary sciences, particularly once the topic directly relates to trading activities. Certainly, Bailey et al. (2014) claim that many medical publications in finance lack a good medical approach, rather getting closer to financial charlatanism and pseudo-mathematics than strenuous sciences. Aware of these concerning tendencies, the existing investigation paper intends to provide an impartial systematic analysis of the novel DRL algorithm proposed.

3. Algorithmic Trading Problem Formalization

In this particular area, the sequential decision making algorithmic trading problem studied in this research paper is offered in detail. Additionally, a rigorous formalization of this specific issue is performed. Additionally, the website link with the RL formalism is highlighted.

3.1. Algorithmic Trading

Algorithmic trading, also known as quantitative trading, is a subfield of financing that could be considered the technique of instantly making trading choices based on a pair of mathematical rules computed by a machine. This generally recognized definition is used to this research paper, though other definitions can be found in the literature. Certainly, a few authors differentiate the trading decisions (quantitative trading) from the particular trading execution (algorithmic trading). For the benefit of generality, quantitative trading and algorithmic trading are believed to be synonyms in this particular research paper, defining the whole automated trading process. Algorithmic trading has shown to be useful to markets, the primary advantage being the substantial enhancement in liquidity, as reviewed in Hendershott et al. (2011). For even more info relating to this particular area, please mention Treleaven et al. (2013) Nuti and et al. (2011).

You will find numerous markets suitable for algorithmic trading methods. Shares and stocks may be traded in the stock markets, FOREX trading is about international currencies, or maybe a trader might purchase commodity futures, to just cite a couple. The latest rise of cryptocurrencies, such as Bitcoin, also provides fascinating choices. Preferably, the DRL algorithms created in this particular research paper must be appropriate to several markets. Nevertheless, the main objective will be set on stock markets for the time being, with an extension to some other markets designed down the road.

In reality, a trading activity could be considered the control of a portfolio, which is a pair of assets, including several stocks, currencies, commodities, bonds, and more. In the range of the research paper, the profile considered comprises a single stock combined with the agent cash. The portfolio value vt will be the trading agent cash value vc as well as the share value vs, which constantly evolves over time t. Selling and buying operations are simply money and share exchanges. The trading agent interacts with the stock market with an order book, which contains the whole range of purchasing orders (selling orders and bids) (asks). An order signifies the determination of a sector participant to exchange, and it is made up of a cost p, a quantity q, along with a side s (bid or even ask). For a trade to take place, a fight between bid and get orders is needed, an event which could just come about whether price (bid) > price (ask) gets the maximum (minimum) cost of any bid (ask) order. Next, a trading agent faces a hard job to produce profit: what, how, when, at which quantity and which price to trade. This is the algorithmic trading complicated sequential decision making issue studied in this particular scientific research paper.

3.2. Timeline Discretization

Since trading choices could be given at any moment, the trading activity is a consistent procedure. To learn the algorithmic trading issue discussed in this particular research paper, a discretization functioning of the steady schedule is performed. The trading timeline is discretized into a top amount of discrete trading time steps t of frequent duration Δt. In this particular research paper, for clarity, the increment (decrement) operations t one (t one) are used to model the discrete switch from time step t to time step t Δt (t' Δt).

The duration Δt is strongly connected to the trading frequency highly targeted by the trading agent (very high trading frequency, monthly, daily, intraday, etc.). Such discretization operation predictably imposes a constraint on this particular trading frequency. Certainly, as the duration Δt between 2 time steps can't be picked as little as possible because of technical constraints, the optimum trading frequency achievable, the same as 1/Δt, is restricted. In the range of the research paper, this particular constraint is greeted when the trading frequency targeted is regular, meaning the trading agent makes a brand new choice once each day.

3.3. Trading Strategy

The algorithmic trading strategy is rule-based, which means the trading choices are produced based on a set of rules: a trading strategy. In complex terminology, a trading method could be considered a programmed policy either stochastic or deterministic, and that outputs a trading action based on the info available on the trading agent at time step t. Furthermore, a vital characteristic of a trading strategy is the sequential aspect of its. An agent performing its trading strategy sequentially is true the following steps:

1. Update of the accessible market info it.
2. Execution of the policy i) getting action at.
3. Application of the designated trading action a.
4. Next time step t' t one, loop to step one.

In the next subsection, the algorithmic trading sequential decision making issue, which shares parallels with other issues effectively tackled by the RL group, is casted as an RL issue.

3.4. Machine Mastering Problem Formalization

Machine learning is worried about the sequential interaction of an agent with its environment. Within every time step t, the RL agent first observes the RL environment of inner state st, and also retrieves an observation ot. After that, it executes the action at resulting from its RL policy h), where ht is definitely the RL agent history, and also gets a reward rt due to its action. In this RL context, the agent history could be conveyed as ht = (oτ, aτ, rτ).

Machine learning methods are worried about the style of policies π maximizing an optimality criterion, which right is determined by the immediate rewards rt found over a particular time horizon. The most favored optimality criterion is the expected affordable amount of incentives over an infinite time horizon. Mathematically, the maximum policy π∗ is expressed when the following:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}[R|\pi]$$

$$R = \sum_{t=0}^{\infty} \gamma^t r_t$$

The parameter γ is the discount factor (γ [zero, 1]). It determines the benefits of future incentives. For example, if γ = zero, the RL agent is believed to be myopic, as it just considers today's incentive and completely discards the future rewards. If the discount factor improves, the RL agent is likely to be much more long-term oriented. In the intense situation wherein γ = one, the RL agent considers each reward equally. This particular key parameter must be tuned based on the preferred behavior.

3.4.1. RL Observations

In the range of the algorithmic trading issue, the RL planet is the whole complex trading world gravitating around the RL agent. In reality, this particular trading atmosphere could be considered an abstraction, like the trading mechanisms combined with each piece of info, able to have an impact on the trading activity of the representative. A significant struggle of the algorithmic trading issue is the poor observability of this particular ecosystem. Certainly, a considerable amount of info is just hidden towards the trading agent, which ranges from certain companies' information that is confidential to the various other industry participants' strategies. In reality, the info available on the RL agent is very limited compared to the intricacy of the ecosystem. Additionally, this info can take different forms, both qualitative and quantitative. Properly processing such info and reexpressing it using related quantitative figures, while minimizing the subjective bias is capital. Lastly, you will find important time correlation complexities to cope with. Thus, the info retrieved by the RL agent at every time action must be viewed sequentially as many info instead of separately.

In this particular research paper, the RL agent observations are usually mathematically conveyed when the following:

ot = S(t′), D(t′), T(t′), I(t′), M(t′), N(t′), E(t′)t′=t'τ

where:

S(t) belongs to the state info of the RL agent at precious time step t (current trading place, quantity of shares owned by the representative, accessible cash).
D(t) could be the info gathered by the representative at a precious time step t concerning the OHLCV (Open-High-Low-CloseVolume) data characterizing the stock market.
I(t) will be the agent info concerning several specialized indications about the stock market targeted at precious time step t. But there are numerous specialized indicators that give additional insights about different monetary phenomena, like moving average convergence divergence (MACD), relative strength index (RSI) or perhaps average directional index (ADX), to just cite a couple.

4. Deep Machine Learning Algorithm Design

In this particular area, a novel DRL algorithm is created to resolve the algorithmic trading issue earlier introduced. The resulting trading strategy, denominated the Trading Deep Q Network algorithm (TDN), is influenced by the excellent DQN algorithm provided in Mnih et al. (2013), and it is appreciably adapted to the particular decision-making issue at hand. To concern the instruction of the RL agent, artificial trajectories are produced from a small range of stock market historic information.

4.1. Deep Q Network Algorithm

The Deep Q Network algorithm, usually called DQN, is a DRL algorithm capable of effectively mastering control policies from high dimensional sensory inputs. It's the successor of the famous Q learning algorithm introduced in Dayan and Watkins (1992). This DRL algorithm is believed to be model free, which means an extensive model of the planet isn't needed and that trajectories are enough. In addition to the Q learning family of algorithms, it depends on the approximation of the state action value feature, represented by a DNN. In context that is such, studying the Q function requires mastering the parameters θ of this DNN. Finally, the DQN algorithm is believed to be off policy because it exploits in batch mode earlier experiences et = (st, rt, at, st one) collected at any time during instruction.

For the benefit of brevity, the DQN algorithm isn't thoroughly provided in this paper. Aside from the first publications (Mnih et al. (2013), Mnih and) et al. (2015)), there is excellent medical literature around this particular algorithm, see for example van Hasselt et al. (2015), Wang et al. (2015), Schaul et al (2016), Bellemare et al. (2017), Fortunato et al. (2018), Hessel and) et al. (2017). Concerning DL strategies, exciting information is LeCun et al. (2015), Goodfellow et al. (2015), Goodfellow and et al. (2016). For even more info about RL, the person can relate to the next books and also surveys: Barto and Sutton (2018), Szepesvari (2010), Busoniu et al. (2010), Arulkumaran et al. (2017 Shao and) et al. (2019).

$$\tau = \Big(\{o_0, a_0, r_0\}, \{o_1, a_1, r_1\}, ..., \{o_T, a_T, r_T\}\Big)$$

4.2. Artificial Trajectories Generation

In the range of the algorithmic trading issue, a total model of the planet E isn't available. The teaching of the TDN algorithm depends entirely on the development of artificial trajectories from a small range of stock market historic day OHLCV data. A trajectory τ is described as a sequence of observations ot ∈ O, behavior at ∈ A, and rewards rt from an RL agent for a particular amount of trading time steps t:

At first, though the planet E is undiscovered, one particular disposes of an individual genuine trajectory, corresponding to the historic behavior of the stock market, i.e. the specific situation of the RL agent becoming sedentary. This particular initial trajectory is made up of the historic volumes and prices combined with long actions performed by the RL agent without any cash at the disposal of its, to symbolize the point that no shares are actually traded. Because of this algorithmic trading issue, brand new fictive trajectories are next artificially produced because of this special real trajectory to simulate interactions with the planet the brand new actions just regard unaffected as E. The historic stock market behavior done by the trading agent.

During every trading time step t, the chosen activity at is performed on the trading environment E, as well as the contrary action at is executed on a copy of the setting E'. Even though this technique doesn't completely resolve the difficult exploration/exploitation trade off, it helps the RL agent continually examine at a little additional computational cost.

4.3. Varied Alterations and Improvements

The DQN algorithm was selected as a kick off point for the novel DRL trading strategy created, but was appreciably adapted to the particular algorithmic trading decision-making issue at hand. The several modifications & improvements, which are generally based upon the many simulations done, are summarized hereafter:

• Deep neural network architecture: The first variation of the classical DQN algorithm will be the structure of the DNN approximating the action value function Q(s, a). Because of the various dynamics of the input (time series rather than raw images), the convolutional neural network (CNN) is supplanted by a classical feed-forward DNN with a few leaking rectified linear unit (Leaky ReLU) activation functions.

• Double DQN: The DQN algorithm suffers from considerable overestimations, this particular overoptimism damaging the algorithm performance. To lower the effect of the undesired phenomenon, the content van Hasselt et al. (2015) provides the double DQN algorithm, which depends on the decomposition of the target max operation into both action choice and also action evaluation.

5. Performance Assessment

A precise performance evaluation approach is capital in an effort to create significant outcomes. As previously hinted, this particular treatment is even more crucial, because there is a genuine absence of a good performance assessment methodology in the algorithmic trading field. In this particular area, a novel, much more reliable methodology is provided to fairly look at the functionality of algorithmic trading methods, including the TDN algorithm.

5.1. Test Bench

In the literature, the functionality of a trading strategy is often assessed on an individual instrument (stock market or maybe others) for a particular time. Nevertheless, the analysis resulting from such a fundamental strategy shouldn't be trusted, as the trading information might have been selected so that a trading strategy appears lucrative, though it's not the situation on the whole. To avoid such bias, the performance should be evaluated on multiple instruments presenting several patterns.

About the trading horizon, the 8 years preceding the publication year of the study paper are selected to be symbolic of the present market conditions. Such a short time period might be criticized, since it might be very limited to be symbolic of the whole set of fiscal phenomena. For example, the financial problem of 2008 is rejected, although it might be fascinating to look at the robustness of trading techniques with regard to such an exceptional event. Nevertheless, the simple fact drove this particular option that a smaller trading horizon is much less likely to have substantial market regime shifts, which could significantly damage the training balance of the trading methods. Lastly, the trading horizon of 8 years is divided into each knowledge and also test sets as follows:
• Training set: 01/01/2015' 31/12/2020.
• Test set: 01/01/2021' 31/12/2021.

A validation set is considered a subset of the instruction set for the tuning of the many TDN algorithm parameters. Remember that the RL policy DNN parameters θ are repaired throughout the trading program on the whole test set, which means the brand new experiences acquired aren't appreciated for added instruction. Nevertheless, such practice constitutes a fascinating potential research direction. To stop this subsection, it must be mentioned that the proposed test bench may be enhanced because of a lot more diversification. The apparent addition will include many more stocks with various financial circumstances and properties. Another intriguing inclusion will be considering various training/testing time periods, while excluding substantial market program shifts. Nevertheless, this last plan was dumped in this systematic post because of the crucial time currently forced to create results for the proposed test bench.

5.2. Benchmark Trading Strategies

To correctly evaluate the pros and cons of the TDN algorithm, various benchmark algorithmic trading strategies have been selected for comparison purposes. Just the classical trading strategies commonly used in training were considered, excluding for example strategies based on DL methods and any other complex techniques. Despite the point that the TDN algorithm is an energetic trading strategy, both active and passive strategies are looked at. For the benefit of fairness, the techniques reveal exactly the same output and input spaces provided in Section 3.4.2 (A). and O The next list summarizes the benchmark techniques selected:
• Buy and hold (B&H).
• Sell and hold (S&H).
• Trend following with moving averages (TF).
• Mean reversion with going averages (MR).

For the benefit of brevity, a comprehensive description of each approach isn't provided in this research paper. The person can relate to Chan (2009), Chan (2013 Narang or) (2009) for more info. The very first 2 benchmark trading strategies (S&H and B&H) are believed to be passive, as there are no changes in trading place over the trading horizon. On the other hand, the additional 2 benchmark strategies (MR and TF) are established trading methods, issuing several variations in trading positions over the trading horizon. On the one hand, a trend following strategy is worried about the follow-up and identification of substantial market trends. On the other hand, a hostile reversion strategy, depends on the inclination of any stock market to return to the previous average price, in the absence of distinct trends. By design, a trend following strategy typically would make an income every time a hostile reversion tactic doesn't, the complete opposite being true too. This's mainly because these 2 families of trading techniques adopt reverse positions: a mean reversion strategy usually denies and also moves contrary to the trends, while a trend following strategy follows the movements.

5.3. Quantitative Performance Assessment

Quantitative performance assessment consists of defining one or more performance indicators, or even more, to quantify the performance of an algorithmic trading technique numerically. The reason a trading technique is so successful is that it must be profitable. Therefore, the strategy must pay off at some point. Such reasoning, however, does not take into account the risk related to trading activity that ought to be mitigated efficiently. In general, traders prefer trading strategies that result in a small but steady profit than long-term strategies.

6. Results and Discussion

The TDN trading technique is examined in this section according to the previously described performance assessment methodology. First of all, a thorough analysis is carried out for both a case that gives excellent results, along with a case that the results were mitigated. This exposes the strengths and weaknesses of the TDN algorithm. Then again, the performance of the DRL trading method on the entire test bench is summarized and analyzed. Some additional discussions are provided regarding the discount factor parameter, trading costs effect, and the primary problems faced by the TDN algorithm.

6.1. Excellent Results Google Stock

The first comprehensive analysis relates to the execution of the TDN trading method on the Google stock, leading to promising outcomes. The TDN algorithm, like many DRL algorithms, is subjected to a non-negligible variance. Trading strategies of various performance could be the result of multiple training experiments with the same initial conditions. Therefore, both a typical execution of the TDN algorithm and its anticipated performance are discussed later.

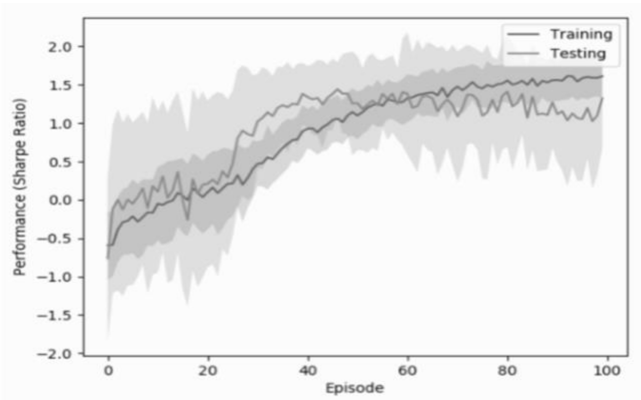

Figure 1: Google stock expected result by using TDN

Typical run: Firstly, the performance achieved by each trading strategy, the initial amount of money being $100,000. The TDN algorithm achieves good results from both an earnings and a risk mitigation perspective, clearly outperforming all benchmark active and passive trading strategies. Secondly, Figure 3 plots both the stock market price pt and RL agent portfolio value vt evolutions, together with the actions outputted by the TDN algorithm. It can be observed that the DRL trading strategy accurately detects and benefits from major trends, while being more hesitant during market behavioral shifts when volatility increases. It can also be seen that the trading agent generally lags slightly behind market trends, meaning the TDN algorithm learned to be more reactive than proactive for this particular stock. This behavior is expected with such a limited observation space not including the reasons for future market directions (new product announcement, financial report, macroeconomics, etc.). However, this does not mean that the policies learned are purely reactive. Indeed, it was observed that the RL agent may decide to adapt its trading position before a trend inversion by noticing an increase in volatility, anticipating and being proactive.

Expected performance: To estimate the expected performance and variance of the TDN algorithm, the same RL trading agent is trained multiple times. The averaged (over 50 iterations) performance of the TDN algorithm for both training and test sets, compared to the number of training episodes. This expected performance is comparable to the performance achieved during the typical run of the algorithm. It can also be noticed that the overriding tendency of the RL agent seems properly handled for this specific market. Please note that the test set performance temporarily superior to the training set performance is not a mistake. It simply indicates an easier to trade and more profitable market for the test set trading period for the Google stock. This example perfectly illustrates a major difficulty of the algorithmic trading problem: the training and test sets do not share the same distributions. Indeed, the distribution of daily returns is continuously changing, which complicates both the training of the DRL trading strategy and its performance evaluation.

6.2. Mitigated Results Microsoft Stock

The same detailed analysis is performed on the Microsoft stock, which presents very different characteristics compared to the Google stock, such as pronounced volatility. Contrary to the promising performance achieved on the previous stock, this case was specifically selected to highlight the limitations of the TDN algorithm.

Regular run: Firstly, \ the performance achieved by each trading strategy, the original sum of money being the same as 1dolar1 100,000. The TDN algorithm achieves results that are good from both an earnings and a threat mitigation perspective, clearly outperforming all benchmark active and also passive trading methods. Next, the stock market price pt and RL agent portfolio value vt evolutions, combined with the activities outputted by the TDN algorithm. It may be found that the DRL trading strategy can effectively detect and benefit from major trends, while being much more uncertain during market behavioral shifts if volatility increases. It can additionally be observed that the trading agent typically lags somewhat behind market trends, indicating that the TDN algorithm was reactive than assertive for this specific stock. This particular behavior is anticipated with such a restricted observation space, not like the causes for the upcoming industry directions (new product announcement, macroeconomics, financial report, etc.). Nevertheless, that doesn't mean the policies mastered are purely reactive. Certainly, it was noticed that the RL agent might want to adjust the trading position of its prior to a trend inversion by seeing increased volatility, thus anticipating and being practical.

Expected effectiveness: To calculate the anticipated performance along with the variance of the TDN algorithm, exactly the same RL trading agent is qualified many times. Figure three plots the averaged (more than fifty iterations) performance of the TDN algorithm for both knowledge and test sets regarding the variety of instruction episodes. This particular anticipated performance is much like the functionality achieved in the regular run of the algorithm. It can additionally be realized that the overriding tendency of the RL agent appears appropriately managed because of this particular store. Please remember that the test set performance being temporarily better on the instruction set isn't a huge mistake. It just suggests a simpler to trade and much more lucrative market for the test set trading period for the Google stock. This particular example completely illustrates a significant challenge of the algorithmic trading problem: the knowledge and test sets don't discuss exactly the same distributions. Certainly, the division of the day returns is constantly changing, which complicates both the instruction of the DRL trading approach and its performance evaluation.

6.2. Mitigated Results Microsoft Stock

The exact same comprehensive examination is carried out on the Microsoft stock, which provides different characteristics compared with the Google stock, like a pronounced volatility. In comparison to the promising performance achieved on the prior stock, this particular situation was particularly selected to spotlight the limits of the TDN algorithm.

Regular run: Just like the prior analysis, the performance attained by each trading technique considered, the original sum of money being the same as 1dolar1 100,000. The mitigated outcomes attained by the benchmark energetic strategies claim the Microsoft stock is hard to trade, partly due to its significant volatility. Though the TDN algorithm achieves an optimistic Sharpe ratio, very little profit is produced. Furthermore, the chance level associated with this particular trading activity can't be considered acceptable. For example, the optimum drawdown duration is especially big, which could lead to a tense circumstance for the operator responsible for any trading strategy. The stock market price pt and RL agent portfolio value vt evolutions combined with the activities outputted by the TDN algorithm, confirms the observation. Furthermore, it could be certainly seen that the pronounced volatility of the Microsoft stock induces a greater trading frequency (changes in trading positions, which match the circumstances regardless of the non negligible trading costs, which increases the riskiness of the DRL trading strategy.

Expected performance: the anticipated performance of the TDN algorithm for both knowledge and test sets, as a characteristic of the variety of instruction episodes (more than fifty iterations). It can easily be exclusively observed that this anticipated performance is drastically better than the functionality achieved by the standard earlier analyzed, which may therefore be considered not necessarily symbolic of the common behavior. This highlights a major limitation of the TDN algorithm: the sizable variance, which could lead to selecting poorly performing policies compared with the anticipated performance. The significantly higher performance attained on the training set additionally implies that the DRL algorithm is governed by overfitting in this particular case, despite the several regularization methods implemented. The observation area could partly clarify this particular overfitting phenomenon that is way too small to effectively apprehend the Microsoft stock. Although this overfitting phenomenon doesn't appear to be harmful in this particular case, it could lead to poor performance for many other stocks.

6.3. Global Results Testbench

As previously suggested in this particular research paper, the TDN algorithm is examined on the test bench created in Section 5.1, to bring trustful and robust more conclusions. The expected Sharpe ratio achieved by both TDN plus benchmark trading techniques on the whole range of stocks contained in this specific test bench.

About the functionality achieved by the benchmark trading methods, it's essential to distinguish the passive strategies (S&H and B&H) from the established people (MR). and TF Indeed, this next family of trading techniques has much more capability in the price of an additional non negligible risk: continuous speculation. As the stock markets were generally bullish (price pt generally increasing over time) with a few instabilities during the test set trading period, it's not shocking to find out that the buy and hold strategy outperformed another benchmark trading method. In reality, neither the trend observing nor the hostile reversion strategy managed to produce rewarding outcomes typically on this particular test bench. It clearly suggests there's a significant difficulty to definitely trade in that market conditions. The simple fact can additionally clarify this particular poorer performance that such strategies are usually well suited to exploit particular fiscal patterns, though they lack versatility and therefore typically fail to attain excellent typical performance on a large set of stocks presenting several characteristics. Additionally, such strategies are commonly somewhat more influenced by the trading costs due to their better trading frequency (for fairly light moving average durations, as it's the situation in this particular research paper).

About the revolutionary trading strategy, the TDN algorithm achieves results promising on the test bench, surpassing the benchmark busy trading techniques on average. Nevertheless, the DRL trading strategy just slightly surpasses the buy and also hold technique on these specific bullish markets, which are extremely favorable to this straightforward passive strategy. Surprisingly, it must be mentioned that the functionality of the TDN algorithm is the same or perhaps extremely near the performance of the passive trading strategies (S&H and B&H) for several stocks. This's clarified by the simple fact that the DRL strategy effectively learns to tend to a passive trading strategy whenever the anxiety related to energetic trading increases. It must also be stressed that the TDN algorithm is neither a trend following nor a mean reversion trading program, as both economic patterns could be well managed in practice. Therefore, the primary benefit of the DRL trading strategy is unquestionably the versatility of its and the ability of its to effectively handle various markets, presenting several characteristics.

6.4. Discount Factor Discussion

As previously defined in Section 3.4, the discount factor γ is worried about the benefits of future incentives. In the range of the algorithmic trading issue, the appropriate tuning of this particular parameter isn't trivial due to the substantial uncertainty of the long term. On the one hand, the sought-after trading policy must be extended oriented (γ1), to stay away from an absurdly high trading frequency and also being subjected to significant trading costs. On the other hand, it will be unwise to put far too much value in a stock market future, and that is especially uncertain (γ0). Thus, a trade off intuitively is present for the discount factor parameter.

The multiple tests validate this particular reason carried out to tune the parameter γ. Certainly, it was noticed that there's an optimum worth for the discount factor, and that is neither too little nor insanely big. Also, these experiments highlighted the secret link between the discount factor and the trading frequency, due to the trading costs. From the perspective of the RL agent, these costs represent an obstacle to get over for an alteration of trading place, due to the quick reduced (and usually negative) reward received. It models the point that the trading agent should be completely confident about the future to conquer the additional risk regarding the trading costs. The discount factor determining the value given to the long term, a tiny value for the parameter γ, will undoubtedly decrease the inclination of the RL agent to modify its trading position, which reduces the trading frequency of the TDN algorithm.

6.5. Trading Costs Discussion

The evaluation of the trading costs impact holding a trading strategy behavior and efficiency is capital, as such costs symbolize an additional threat to mitigate. A significant inspiration for learning DRL solutions rather compared to clean prediction methods, which might be also based on DL architectures, is connected to the trading costs. As previously defined in Section three, the RL formalism allows the consideration of these extra costs directly to the decision-making process. The perfect policy is discovered based on the trading costs value. On the other hand, a purely predictive strategy would only output predictions about the upcoming market direction or maybe prices, with no indications about a suitable trading strategy considering the trading costs. Even though this last approach offers much more flexibility and could definitely lead to well performing trading strategies, it's less effective by design.

To illustrate the ability of the TDN algorithm to efficiently and automatically adjust to various trading costs, the behavior of the DRL trading program for 3 diverse costs values, any other details remaining unchanged. It can obviously be found that the TDN algorithm effectively minimizes its trading frequency whenever the trading fees increase, as expected. When these expenses start to be too much, the DRL algorithm just prevents definitely trading and also uses a passive approach (buy plus sell or hold and also maintain strategies).

6.6. Core Challenges

Today, the primary DRL solutions effectively put on to real life problems concern certain spaces with certain properties, like video games (see e.g. the popular AlphaGo algorithm created by Google Deepmind Silver et al. (2016)). In this particular research paper, a completely different atmosphere characterized by a considerable complexity and significant uncertainty is analyzed with the algorithmic trading condition. Clearly, several challenges have been faced throughout the study within the TDN algorithm, the main ones being summarized hereafter.

For starters, the incredibly terrible observability of the trading atmosphere is a characteristic that substantially limits the functionality of the TDN algorithm. Certainly, the quantity of info in the fingertips of the RL agent is actually inadequate to effectively describe the financial phenomena happening during training, which is essential to effectively find out to trade. Next, though the distribution of the day returns is constantly changing, yesteryear is necessary to be symbolic of the potential future for the TDN algorithm to achieve good results. This will make the DRL trading strategy especially vulnerable to considerable market regime shifts. Thirdly, the TDN algorithm overfitting tendency must be appropriately managed to get a well-performing trading strategy. As recommended in Zhang et al. (2018), much more rigorous evaluation protocols are needed in RL due to the strong inclination of typical DRL strategies to overfit. Far more research on this matter is necessary for DRL strategies to slip a broader range of real-life applications. Finally, the sizable variance of DRL algorithms like DQN causes it to be somewhat hard to effectively put on these algorithms to particular issues, particularly once the training and test sets differ considerably. This is a major limitation of the TDN algorithm, which was previously highlighted for the Microsoft stock.

7. Conclusion

This particular scientific research paper presents the Trading Deep Q Network algorithm (TDN), a full machine learning (DRL) means to fix the algorithmic trading issue of figuring out the perfect trading place at any time during a trading activity available in markets. Carrying out a rigorous performance evaluation, this revolutionary trading strategy achieves results promising, surpassing typically the benchmark trading methods. Furthermore, the TDN algorithm demonstrates numerous advantages compared to much more classical methods, like an appreciable versatility and amazing robustness to several trading costs. Additionally, such data driven approach offers the main benefit of suppressing the complicated job of defining explicit rules suitable for the specific financial markets considered.

Nevertheless, the functionality of the TDN algorithm might remain improved, from both a generalization along with a reproducibility point of view, to cite a couple. Many research directions are suggested upgrading the DRL solution, like the use of LSTM layers to the deep neural system, that should help with much better process of the fiscal time series information, see e.g. Stone and Hausknecht (2015). An additional example is the consideration of the many improvements implemented in the Rainbow algorithm, that are thorough in Barto and Sutton (2018), van Hasselt et al. (2015), Wang et al. (2015), Schaul et al. (2016), Bellemare et al. (2017), Fortunato et al. (2018) and Hessel et al.
(2017). One more intriguing research path is definitely the comparability of the TDN algorithm with Policy Optimisation DRL algorithms, including the Proximal Policy Optimisation (PPO Schulman et al. (2017) algorithm. The final significant research direction suggested concerns the formalization of the algorithmic trading issue into machine learning. For starters, the observation room must be extended to improve the observability of the trading environment. Likewise, many constraints about the action room might be calm to allow new trading possibilities. Secondly, advanced RL reward engineering must be done to narrow the gap between the RL goal and also the Sharpe ratio maximization objective. Lastly, a promising and interesting research path would be the consideration of distributions rather than likely values in the TDN algorithm to cover the idea of risk and also to much better deal with uncertainty.

References

[1]. Bailey, D. H., Borwein, J. M., de Prado, M. L., and Zhu, Q. J. (2014). Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance. Notice of the American Mathematical Society, pages 458–471.

[2]. Bao, W. N., Yue, J., and Rao, Y. (2017). A Deep Learning Frame-work for Financial Time Series using Stacked Autoencoders and Long-Short Term Memory. PloS one, 12.

[3]. Bellemare, M. G., Dabney, W., and Munos, R. (2017). A Dis-tributional Perspective on Machine Learning. CoRR, abs/1707.06887.

[4]. Bollen, J., Mao, H., and jun Zeng, X. (2011). Twitter Mood Predicts the Stock Market. J. Comput. Science, 2:1–8.

[5]. Boukas, I., Ernst, D., Th´eate, T., Bolland, A., Huynen, A., Buch- wald, M., Wynants, C., and Corn´elusse, B. (2020). A Deep Rein- forcement Learning Framework for Continuous Intraday Market Bidding. ArXiv, abs/2004.05940.

[6]. Busoniu, L., Babuska, R., De Schutter, B., and Ernst, D. (2010). Re- inforcement Learning and Dynamic Programming using Function Approximators. CRC Press.

[7]. Carapu¸co, J., Neves, R. F., and Horta, N. (2018). Machine Learning applied to Forex Trading. Appl. Soft Comput., 73:783–794.

[8]. Chan, E. P. (2009). Quantitative Trading: How to Build Your Own Algorithmic Trading Business. Wiley.

[9]. Chan, E. P. (2013). Algorithmic Trading: Winning Strategies and Their Rationale. Wiley.

[10]. Dempster, M. A. H. and Leemans, V. (2006). An Automated FX Trading System using Adaptive Machine Learning. Expert Syst. Appl., 30:543–552.

[11]. Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q. (2017). Deep Direct Machine Learning for Financial Signal Representa- tion and Trading. IEEE Transactions on Neural Networks and Learning Systems, 28:653–664.

[12]. Fortunato, M., Azar, M. G., Piot, B., Menick, J., Hessel, M., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., and Legg, S. (2018). Noisy Networks for Exploration. CoRR, abs/1706.10295.

[13]. Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press.

[14]. Goodfellow, I. J., Bengio, Y., and Courville, A. C. (2015). Deep Learning. Nature, 521:436–444.

[15]. Hausknecht, M. J. and Stone, P. (2015). Deep Recurrent Q-Learning for Partially Observable MDPs. CoRR, abs/1507.06527.

[16]. Hendershott, T., Jones, C. M., and Menkveld, A. J. (2011). Does Algorithmic Trading Improve Liquidity? Journal of Finance, 66:1–33.

[17]. Hessel, M., Modayil, J., van Hasselt, H. P., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M. G., and Silver, D. (2017). Rainbow: Combining Improvements in Deep Reinforce-ment Learning. CoRR, abs/1710.02298.

[18]. Ioannidis, J. P. A. (2005). Why Most Published Research Findings Are False. PLoS Med, 2:124.

[19]. Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerat- ing Deep Network Training by Reducing Internal Covariate Shift. CoRR, abs/1502.03167.

[20]. Kingma, D. P. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. CoRR, abs/1412.6980.

[21]. LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep Learning. Na- ture, 521.

[22]. Leinweber, D. and Sisk, J. (2011). Event-Driven Trading and the “New News”. The Journal of Portfolio Management, 38:110–124. Li, Y. (2017). Deep Machine Learning: An Overview. CoRR, abs/1701.07274.

[23]. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. A. (2013). Playing Atari with Deep Machine Learning. CoRR, abs/1312.5602.

[24]. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis,

[25]. D. (2015). Human-Level Control through Deep Machine Learning. Nature, 518:529–533.

[26]. Moody, J. E. and Saffell, M. (2001). Learning to Trade via Direct Machine. IEEE transactions on neural networks, 12 4:875– 89.

[27]. Narang, R. K. (2009). Inside the Black Box. Wiley.

[28]. Nuij, W., Milea, V., Hogenboom, F., Frasincar, F., and Kaymak, U. (2014). An Automated Framework for Incorporating News into Stock Trading Strategies. IEEE Transactions on Knowledge and Data Engineering, 26:823–835.

[29]. Nuti, G., Mirghaemi, M., Treleaven, P. C., and Yingsaeree, C. (2011). Algorithmic Trading. Computer, 44:61–69.

[30]. Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2016). Prioritized Experience Replay. CoRR, abs/1511.05952.

[31]. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal Policy Optimization Algorithms. CoRR, abs/1707.06347.

[32]. Shao, K., Tang, Z., Zhu, Y., Li, N., and Zhao, D. (2019). A Survey of Deep Machine Learning in Video Games. ArXiv, abs/1912.10944.

[33]. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbren- ner, N., Sutskever, I., Lillicrap, T. P., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the Game of Go with Deep Neural Networks and Tree Search. Nature, 529:484– 489.

[34]. Sutton, R. S. and Barto, A. G. (2018). Machine Learning: An Introduction. The MIT Press, second edition.

[35]. Szepesvari, C. (2010). Algorithms for Machine Learning. Morgan and Claypool Publishers.

[36]. Treleaven, P. C., Galas, M., and Lalchand, V. (2013). Algorithmic Trading Review. Commun. ACM, 56:76–85.

[37]. van Hasselt, H. P., Guez, A., and Silver, D. (2015). Deep Reinforce- ment Learning with Double Q-Learning. CoRR, abs/1509.06461. Wang, Z., de Freitas, N., and Lanctot, M. (2015). Dueling Net- work Architectures for Deep Machine Learning. CoRR, abs/1511.06581.

[38]. Watkins, C. J. C. H. and Dayan, P. (1992). Technical Note: Q-Learning. Machine Learning, 8:279–292.

[39]. Zhang, C., Vinyals, O., Munos, R., and Bengio, S. (2018). A Study on Overfitting in Deep Machine Learning. CoRR, abs/1804.06893.