# Beating the Stock Market with a Deep Reinforcement Learning Day Trading System

1st Leonardo Conegundes
*Department of Computer Science*
*Universidade Federal de Minas Gerais - DCC/UFMG*
*Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG*
Belo Horizonte, Brazil
leocm@dcc.ufmg.br

2nd Adriano C. Machado Pereira
*Department of Computer Science*
*Universidade Federal de Minas Gerais*
DCC / UFMG
Belo Horizonte, Brazil
adrianoc@dcc.ufmg.br

***Abstract*—In this study we investigate the potential of using Deep Reinforcement Learning (DRL) to day trade stocks, taking into account the constraints imposed by the stock market, such as liquidity, latency, slippage and transaction costs. More specifically, we use a Deep Deterministic Policy Gradient (DDPG) algorithm to solve a series of asset allocation problems in order to define the percentage of capital that must be invested in each asset at each period, executing exclusively day trade operations. DDPG is a model-free, off-policy actor-critic method that can learn policies in high-dimensional and continuous action and state spaces, like the ones normally found in financial market environments. The proposed day trading system was tested in B3 - Brazil Stock Exchange, an important and understudied market, especially considering the application of DRL techniques to alpha generation. A series of experiments were performed from the beginning of 2017 until the end of 2019 and compared with ten benchmarks, including Ibovespa, the most important Brazilian market index, and the stock portfolios suggested by the main Brazilian banks and brokers during these years. The results were evaluated considering return and risk metrics and showed that the proposed method outperformed the benchmarks by a huge margin. The best results obtained by the algorithm had a cumulative percentage return of 311% in three years, with an annual average maximum drawdown around 19%.**

***Index Terms*—Deep Reinforcement Learning, Deep Deterministic Policy Gradient, Machine Learning, Neural Networks, Algorithmic Trading, Stock Trading, Asset Allocation Problem, Intraday Trading, Financial Markets.**

## I. Introduction

Predicting prices or trends in the stock market is a subject of major interest for both academics and practitioners in the financial market. The subject has been extensively studied in the last decades [1] [2] [3], but predicting future stock prices accurately and mainly developing profitable trading strategies based on these predictions are still big challenges.

Powerful techniques from different areas have been widely used in financial market applications, including Optimization, Signal Processing, Time Series Analysis, Control Theory, Advanced Statistical Methods and Machine Learning (ML), with special and increasing interest from the financial community in the last one. According to the Refinitiv 2019 Artificial Intelligence / Machine Learning Global Study [4], more than 90% of financial firms are using ML in their businesses, with 63% of the initiatives associated with trading investment idea generation (alpha generation).

The most classic use of ML to trade in the stock market is to apply supervised learning techniques to predict future prices in a regression problem. These predictions are then used by a trading system that buys and sells stocks based on predefined rules to enter and exit trades. The main problems with this approach are: *(i)* it highly depends on the quality of predictions, but it turns out that future prices are very hard to predict, *(ii)* price predictions are not directly translated into buy and sell signals, requiring an additional layer of logic to convert them into market actions, *(iii)* we only have a complete automated trading system if we have at least two subsystems, where the output of the first (the supervised learning model) is the input of the second (the trading system), *(iv)* the purpose of the learning model (minimize prediction error) is different from that of the complete trading system (maximize cumulative return or Sharpe ratio, minimize drawdown or volatility, etc.) and, finally, *(v)* as the trading rules are not incorporated in the definition of the learning algorithm, several important real world constraints are disregarded in this "smart" first subsystem, such as: liquidity, latency, slippage and transaction costs.

In addition, as illustrated in [5], the typical development process based on this supervised approach has several stages, such as: *(i)* data analysis, *(ii)* training of supervised models, *(iii)* development of the trading system, *(iv)* backtesting (simulation in the past), *(v)* parameter optimization, *(vi)* paper trading (simulation in real time) and *(vii)* live trading (trading account in a broker). This process is complex and not ideal for several reasons, among which we highlight: *(i)* slow iteration cycles, *(ii)* real world factors disregarded in the first stages, *(iii)* simple trading rules made by human and *(iv)* inefficient trading system optimization.

Previous successful attempts of fully machine learning schemes to alpha generation, without predicting prices, are treating the problem as a Reinforcement Learning (RL) one, where we optimally solve complex sequential decision-making problems under uncertainty via direct interaction with the environment and learning through trial and error [6] [7].

The use of RL to address these issues is justified for

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

a few reasons: *(i)* end-to-end optimization, *(ii)* automatic and intelligent creation of the trading system, where the RL directly learns a policy (trading system), being able to learn policies that are more complex and powerful than any rule that a human can create and *(iii)* training directly in simulation environments, considering factors like liquidity, latency, slippage and transaction costs.

Rapid advances in Deep Neural Networks (DNNs) over the past several years have allowed RL to solve decision-making problems with high-dimensional state-action spaces, as the one discussed here, where states and actions are continuous, thus establishing the Deep Reinforcement Learning (DRL) field, which has been widely successful in playing board games and video games and recently also being applied to alpha generation.

In this work, we developed a day trading system based on a DRL algorithm to solve a series of asset allocation problems, buying up to 10 stocks in the opening auction of each trading session and keeping the stocks purchased until the end of the day, when they are sold during the closing auction. In this way, the strategy only executes day trade operations, never being positioned overnight. The day trading system was tested in B3, the Brazilian Stock Exchange [8]. The main motivation to use this scenario is the lack of studies applying DRL in this market and the easy of using the trading system proposed here in a real account, through a partner trading platform [9].

The proposed trading system was able to generate alpha in the last 3 years in simulation account, from 2017 to 2019, with results significantly better than those obtained by all the benchmarks evaluated in this work, including Ibovespa, the main index of the Brazilian stock market, and the best recommended stock portfolios of brokers and banks from Brazil during this same period.

The remainder of this paper is organized as follows. Section II briefly discusses some related work. Section III introduces the proposed DRL day trading system. Section IV presents the experiments and results achieved with real market data from B3. Finally, Section V presents some conclusions and directions for future work.

## II. Related Work

Machine learning is disrupting decision making in almost any area of finance. In the literature, there is a vast number of papers that study alpha generation methods, and most of them use supervised learning techniques to predict future stock prices [2] [3] [10] [11].

Reinforcement Learning has become increasingly popular over the last few years due to its success in tackling challenging sequential decision-making problems. The combination of RL with deep learning techniques is responsible for several of these achievements [12] [13] [14] [15] [16], and this combination, called Deep Reinforcement Learning, is most useful in high-dimensional problems like asset allocation, an investment strategy that aims to balance risk and reward by defining the percentage of capital that must be invested in each asset at each period, according to an individual's goals, risk tolerance, and investment horizon.

Inspired by these successful DRL algorithms, an increasing number of papers have been published in recent years applying RL to financial decision-making and execution problems [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32].

For instance, [21] [26] [33] [34] are some of the previous model-free successful attempts to tackle alpha generation problems without predicting future prices, but these RL algorithms are limited to single-asset and do not apply to general asset allocation problems.

Recently, the model-free Deep Deterministic Policy Gradient (DDPG) algorithm [35] has been used to tackle the the general problem with multiple assets [25], and appear to get good results. They introduced various DNN structures and techniques to trade a portfolio consisting of cash and 10 cryptocurrencies. Similarly, [36] tries the same approach trading the S&P500 and Euro Stoxx 50 and [37] optimize stock portfolios by using the DDPG as well as the Proximal Policy Optimization (PPO) [38].

All these works proposed trading systems that operate by making swing trade (when buying and selling are made on different days, as opposed to day trade) or in markets that operate uninterrupted for 24 hours, such as cryptocurrencies. Thus, to the best of our knowledge, the trading system proposed in this work is the first one to make day trade operations using the DDPG algorithm in traditional markets such as the stock market, which have opening and closing times for each trading session.

By allowing only day trade operations, we have at least three major advantages: *(i)* lower trading costs, *(ii)* risk minimization and *(iii)* intraday leverage. On virtually every stock exchange in the world, the transaction costs are much lower in day trade comparing to swing trade. In Brazil, the reduction is approximately 25% [39]. Limited to only day trade operations, the agent is less exposed to some market risks, since by always closing the operations within the same trading session, it avoids the risk of an impacting event happening outside of trading hours that could negatively influence their positions at the beginning of the next trading session. Finally, leverage is an option that brokers offer, mostly free of charge, so that the agent can trade with more capital than it really has in its account. Thus, the agent does not need to have all the capital to buy a certain number of shares and can increase their percentage returns (at the same time increasing the risk).

## III. The Proposed Deep Reinforcement Learning Day Trading System

In this section we describe the proposed approach of Deep Reinforcement Learning, presenting first the problem definition (A). Then, the Markov Decision Process (MDP) is explained (B), followed by the algorithm for portfolio optimization (C) and some assumptions (D).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

### A. Problem Definition

The problem of assigning optimal weights to a portfolio at each instant of time (percentage of capital that must be allocated to each asset), with the aim to maximize its expected long-term return, can be reformulated as a Markov Decision Process (MDP). The uncertainty about future market states, given the difficulty of predicting investment returns with sufficient accuracy, makes it an optimal stochastic control problem with continuous action and state spaces, a problem that can be solved by model-free Reinforcement Learning, which does not have the need to know the dynamics of the system and defines the optimal policy based on samples.

Let's consider a standard configuration for the asset allocation problem through RL, with an agent interacting with an environment in discrete timesteps. At each timestep $t$, the agent receives the current state $s_t \in \mathbb{S}$ and takes an action $a_t \in \mathbb{A}$ according to its policy $\mu$, which maps states to a probability distribution over the actions $\mu : \mathbb{S} \to P(\mathbb{A})$. The agent then receives a reward $r_t = r(s_t, a_t)$, which can be interpreted as the percentage return of the portfolio between the beginning of time periods $t$ and $t+1$, and receives the next observation $s_{t+1}$.

In this work, we defined our environment similar to the one proposed in [25]. Let $N$ be the number of stocks we have available to invest. Then, our portfolio will be composed by $N+1$ assets, with the first being a special one, the Brazilian Real (R$ / BRL), the currency in which the stocks are quoted, simply referred as cash for the rest of this article.

Since our day trading system allocates capital during the opening auction of every trading session, we always buy the stocks at their opening prices, according to the percentage of capital defined to be invested in each of them. During the entire session of each day, the stocks remain purchased, being sold at the auction closing price of the same day, thus ensuring that all the operations are day trades.

### B. MDP Formulation

*1) Action Space:* In the asset allocation problem, the trading agent has to define the portfolio vector $w_t$ in the beginning of every time step $t$ (every business day). Therefore, the action $a_{t-1}$ at the end of timestep $t-1$ is the portfolio vector $w_t$ at the beginning of timestep $t$:

$$\boldsymbol{a_{t-1}} = \boldsymbol{w_t} = [w_{0,t}, w_{1,t}, \cdots, w_{N,t}]^T, \tag{1}$$

where $w_{i,t}$ represents the fraction of investment on stock $i$ at the beginning of timestep $t$, for $i \geq 1$, and $w_{0,t}$ represents the fraction of cash that we maintain in our portfolio at timestep $t$. As short-selling is prohibited in our trading system, the portfolio weights must be strictly non-negative, or:

$$a_t \in \mathbb{A} \subseteq [0,1]^{N+1}, \ \forall t \geq 0 \text{ subject to} \sum_{i=0}^{N} a_{i,t} = 1. \tag{2}$$

*2) State & Observation Space:* The stock market cannot be fully observable, since we are dealing with other agents (traders), which we can't observe (their balances, positions, hidden orders, trading strategies, etc.). This means the agent is not able to observe the full state of the system, but a noisy state instead, known as an observation, and that asset allocation should be modeled as a partially observable MDP.

Nonetheless, considering the technical analysis theory, all relevant information is believed to be reflected in the stock prices [40] [41], which are publicly available to the agent. Under this point of view, an environmental state can be roughly represented by the history of stock prices up to the moment where the state is at.

As we always buy the stocks at the opening prices and sell them at the closing prices of the same day, we define *close/open relative price vector* as:

$$\boldsymbol{y_t} = \left[1, \frac{c_{1,t}}{o_{1,t}}, \frac{c_{2,t}}{o_{2,t}}, \cdots, \frac{c_{N,t}}{o_{N,t}}\right], \ \forall t \geq 0, \tag{3}$$

where $c_{i,t}$ and $o_{i,t}$ are, respectively, the closing and opening prices of stock $i$ at timestep $t$ and $y_{i,t} = \frac{c_{i,t}}{o_{i,t}}$ is the *relative price* of stock $i$ at timestep $t$, with $y_{0,t}$ representing the relative price of cash at timestep $t$, which is equal to 1 for all $t$.

We define our observation space $\mathbb{S}$ as a subset of the continuous $(W \times (N+1))$-dimensional positive real space $\mathbb{R}_+^{W \times (N+1)}$, where $W$ is the number of past days we consider to define an observation. We limited the number of days to a fixed window length $W$ since the impact of historical data decreases as the time goes by.

Then, the observation $s_t$ at timestep $t$ is defined as:

$$s_t = \begin{bmatrix} \boldsymbol{y_{t-W}} \\ \boldsymbol{y_{t-W+1}} \\ \vdots \\ \boldsymbol{y_{t-1}} \end{bmatrix} = \begin{bmatrix} 1, \frac{c_{1,t-W}}{o_{1,t-W}}, \cdots, \frac{c_{N,t-W}}{o_{N,t-W}} \\ 1, \frac{c_{1,t-W+1}}{o_{1,t-W+1}}, \cdots, \frac{c_{N,t-W+1}}{o_{N,t-W+1}} \\ \vdots \\ 1, \frac{c_{1,t-1}}{o_{1,t-1}}, \cdots, \frac{c_{N,t-1}}{o_{N,t-1}} \end{bmatrix} \tag{4}$$

The state transitions are completely defined by the market, and we do not have any control of them. What we can get is the new observation, defined by the last relative prices.

*3) Reward:* The relative price vector $y_t$ can be used to calculate the change in total portfolio value at timestep $t$. If $p_t$ is the portfolio value at the beginning of timestep $t$, ignoring transaction costs, we have

$$p_{t+1} = p_t \boldsymbol{y_t} \cdot \boldsymbol{w_t}. \tag{5}$$

Then, without loss of generality, assuming the initial portfolio value $p_1$ is equal to 1, the portfolio value at the beginning of timestep $T$ is

$$p_T = p_1 \prod_{t=1}^{T-1} \boldsymbol{y_t} \cdot \boldsymbol{w_t} = \prod_{t=1}^{T-1} \boldsymbol{y_t} \cdot \boldsymbol{w_t}, \tag{6}$$

and if we consider a constant commission rate $\alpha \in [0,1)$ for each timestep, we can consider a transaction remainder factor $\beta \in (0,1]$, such that $\beta = 1 - \alpha$ and the equation 6 becomes

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

$$p_T = \prod_{t=1}^{T-1} \beta \boldsymbol{y_t} \cdot \boldsymbol{w_t}. \quad (7)$$

Although we want to maximize $p_T$, instead of having a 0 reward at each timestep and a $p_T$ reward at the end, we take the logarithm of equation 7:

$$\log p_T = \log \prod_{t=1}^{T-1} \beta \boldsymbol{y_t} \cdot \boldsymbol{w_t} = \sum_{t=1}^{T-1} \log(\beta \boldsymbol{y_t} \cdot \boldsymbol{w_t}). \quad (8)$$

Thus, at each timestep $t$ we have a $\log(\beta \boldsymbol{y_t} \cdot \boldsymbol{w_t})$ reward, avoiding the sparsity of the reward problem.

### C. The Actor-Critic Deep Deterministic Policy Gradient Algorithm

In order to dynamically optimize stock portfolios, we used the Deep Deterministic Policy Gradient (DDPG) algorithm proposed in [35], which is a model-free, off-policy actor-critic method using deep function approximators that can learn policies in high-dimensional and continuous action and state spaces, like those in the asset allocation problem.

The algorithm is based on the Deterministic Policy Gradient (DPG) method [42], combining the actor-critic approach with insights from the recent success of Deep Q Network (DQN) [43] [44], resulting in a general-purpose continuous DRL framework.

The DDPG combines the value-based DQN and the policy-based DPG for large continuous domains, where the actor, which is a parameterized deterministic policy $\mu(s|\theta^\mu)$, learns using the Bellman equation as in Q-Learning based on feedback from the critic, which is the state-action value function $Q(s, a)$.

As discussed in [35], one key advantage of DDPG is that with the same hyper-parameters and network structure it can learn competitive policies using low-dimensional observations, like the ones used in this work, resulting in a simple approach that requires only a straightforward actor-critic architecture and learning algorithm. We reproduce the algorithm described in [35] in Figure 1.

Considering the results presented in [45], we used a very straightforward implementation of a Long Short-Term Memory (LSTM) network as our close/open relative price predictor, based on historical prices, and all the stocks shared the same LSTM for each year of the testing set (from 2017 to 2019).

### D. Assumptions

We only consider backtest tradings, where the trading agent simulates at a past point in the market data history, not knowing any "future" value. The following five assumptions must apply as a requirement for the experiments: *(i)* sufficient liquidity, *(ii)* zero slippage, *(iii)* zero market impact, *(iv)* zero latency impact and *(v)* odd lot equivalent to round lot prices, all of which are realistic in the scenario of the proposed trading system, as explained below.

*Assumption 1 (Sufficient Liquidity)*: All stocks are liquid and each trade can be executed under the same conditions. An asset is considered liquid if it can be rapidly converted to cash, with little or no value loss [46]. As we only trade the 10 most traded stocks from the previous year (the 10 stocks with highest weight in Ibovespa composition), we are dealing with the most liquid stocks from B3.

*Assumption 2 (Zero Slippage)*: Each trade can be executed with zero slippage, exactly at open or close prices. Slippage refers to the difference between a trade's expected execution price and the price at which the trade is actually executed [47]. There is no slippage in our orders, as we send market orders during the auctions and then always trade at the opening (to buy) or closing (to sell) prices.

*Assumption 3 (Zero Market Impact)*: The amount of cash invested by the trading agent is so insignificant that it has no influence on the opening and closing auction prices.

*Assumption 4 (Zero Latency Impact)* The delay to send orders has no influence on the execution prices. As the opening and closing auctions last at least 5 minutes, the agent has all this period to send an order for each stock to be traded.

*Assumption 5 (Odd lot equivalent to round lot prices)* To trade the financial volume as close as possible to the theoretical volume defined by the weight of each share in the portfolio, it is necessary to trade in the odd lot market [48] and in the round lot market [49]. As we invest only in the 10 most liquid stocks on the market, we can assume that the prices are equivalent or have insignificant differences.

## IV. Experiments and Discussion

This section presents experimental results performed to test the ability of the proposed DRL Day Trading System to generate alpha in real world scenarios. We tested three different values for the window length $W$ (2, 3 and 5), resulting in the methods named DRL-2, DRL-3 and DRL-5, respectively. The results obtained were compared with ten important benchmarks and showed that our methods had outperformed the benchmarks by a huge margin.

### A. Dataset

The dataset used throughout this work corresponds to a historical series of data about the daily opening and closing prices of the thirteen different stocks that were available to invest according to our methodology to daily solve the asset allocation problem. The series represents a period of 1824 days, varying from 2nd January 2015 to 30th December 2019, and were obtained from the software SmarttBot [9].

In order to select the stocks that can be invested by the DRLs, we consider, for each year, only the ten stocks with the greatest weight in the first trading session of that year in the composition of the iShares Ibovespa (BOVA11), an ETF managed by BlackRock [50], the world's largest asset manager.

BOVA11 is the ETF with the most traded financial volume at B3, and seeks to obtain returns that generally correspond to the performance of the Bovespa Index (Ibovespa), the most

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

**Algorithm 1** DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights $\theta^Q$ and $\theta^\mu$.
Initialize target network $Q'$ and $\mu'$ with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer $R$
**for** episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state $s_1$
  **for** t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action $a_t$ and observe reward $r_t$ and observe new state $s_{t+1}$
    Store transition $(s_t, a_t, r_t, s_{t+1})$ in $R$
    Sample a random minibatch of $N$ transitions $(s_i, a_i, r_i, s_{i+1})$ from $R$
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  **end for**
**end for**

Fig. 1. Deep Deterministic Policy Gradient Algorithm [35]

important Brazilian market index, comprised of stocks issued by companies that represent more than $80\%$ of the number of trades and of the financial volume recorded on the spot stock market.

Then, for each trading session of an year, the DRL method should define the percentage of capital to be invested in each of the ten stocks selected for that year, and can change these values every day, always distributing up to the total available capital between these ten stocks. It is important to remember that part or even the total capital can be allocated to cash in a given day, as explained in Section III.

Table I shows the ten stocks with the greatest weight in the composition of BOVA11 at the first trading session of each year between 2017 and 2019, and so defines which stocks could be considered by the DRL method to invest in each year.

TABLE I
THE TOP 10 STOCKS OF BOVA11 AT THE FIRST DAY OF EACH YEAR

| 2017 | 2018 | 2019 |
|---|---|---|
| ITUB4 | ITUB4 | ITUB4 |
| BBDC4 | VALE3 | VALE3 |
| ABEV3 | BBDC4 | BBDC4 |
| PETR4 | ABEV3 | PETR4 |
| VALE3 | PETR4 | ABEV3 |
| BRFS3 | B3SA3 | BBAS3 |
| BBAS3 | ITSA4 | B3SA3 |
| ITSA4 | BBAS3 | ITSA4 |
| B3SA3 | UGPA3 | LREN3 |
| UGPA3 | BRFS3 | UGPA3 |

### B. Benchmarks

We divided the ten benchmarks used into two categories. The first contains two classic benchmarks defined from a holistic view of the Brazilian market. In the second, we use the stock portfolios recommended by the main Brazilian banks and brokers during the analysis period.

The first benchmark of the first category is the classical Ibovespa, the market index used as the principal baseline for any investment with stocks in Brazil. We defined the average annual return of the ten stocks selected for each year as our second benchmark, named Top 10 Stocks. This is the same as create a portfolio in which each of the ten stocks would have a weight of $10\%$ in the portfolio composition, and can be used to measure if the daily weights calculated by the DRL method are better then the most naive approach to buy and hold the same financial volume of each stock for the entire year.

In order to compare our method with the best stock portfolios suggested by the analysts from the main Brazilian banks and brokers, we used the database available at Carteira Valor [51], a web site from the journal Valor Econômico [52] that aggregates since 2012 information about the top $5$ stocks recommended by each financial institution every month.

We consider only the banks and brokers that suggested their stock portfolios for all the $36$ months of the test period, from 2017 to 2019, resulting in a total of $8$ financial institutions: Ágora, Ativa, Banco do Brasil, Genial, Guide, Planner, Santander and XP.

Just to illustrate the relevance of this group of $8$ financial

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

institutions, they comprise 3 of the 5 largest banks in Brazil and the main independent broker of the country, XP Inc., which made its IPO at Nasdaq in December 2019.

### C. Configurations

For each year of the test period, we train the models using 100 episodes, considering always the data history of the two last years. So, first we trained the agent from 1st January 2015 to 31th December 2016 in 100 episodes, and tested in 2017. Similarly, to test in 2018 we trained with data from years 2016 and 2017 and to test in 2019 we trained with data from years 2017 and 2018.

We set the transaction cost fee $\alpha$ to be equal to $0.00023660$ of the total trading financial volume, since this is the cost to individual investors day trade in the spot market, according to the current fees of B3 [39]. It is important to note that several brokers in Brazil (e.g., Toro Investimentos [53], Warren [54], Clear [55], Inter [56] and CM Capital Markets [57]) are commission-free and therefore we do not need to consider these costs in our experiments.

### D. Results and Analysis

Profit and loss are reported in terms of cumulative percentage annual returns. Table II presents the returns obtained by the three DRL methods and the ten benchmarks analyzed for each year, and the cumulative return considering the compounding results of the three years. As can be observed, the DRL-2 method obtained the best cumulative return and the best annual return for the years of 2018 and 2019, with a cumulative return of $311\%$ in three years, with a profit of more than $134\%$ compared with the best benchmark, Santander, that achieved a total return of $177\%$. This means that if an investor had invested $R\$100.000,00$ in the beginning of 2017 applying the DRL-2, his capital would have increased to $R\$411.000,00$ at the beginning of 2020, while following the best portfolio recommendation from the Brazilian financial institutions, he would have $R\$276.600,00$, a difference greater than $R\$134.000,00$.

TABLE II
ANNUAL FINANCIAL RETURNS FROM 2017 TO 2019 OF 3 DRLS METHODS AND 10 BENCHMARKS

| Method / Benchmark | Annual Return (%) | | | Cumulative Return (%) |
|---|---|---|---|---|
| | 2017 | 2018 | 2019 | |
| DRL-2 | 31.8 | **62.2** | **92.3** | **311.0** |
| DRL-3 | 29.5 | 39.0 | **50.1** | 170.1 |
| DRL-5 | 37.4 | **117.4** | 8.1 | **223.0** |
| Ibovespa | 26.9 | 15.0 | 31.6 | 92.1 |
| Top 10 Stocks | 25.2 | 13.1 | 23.3 | 74.6 |
| Ágora | 34.9 | 4.4 | 30.9 | 84.2 |
| Ativa | 30.7 | -3.7 | 22.2 | 53.9 |
| Banco do Brasil | 24.2 | 33.2 | 41.7 | 134.4 |
| Genial | 44.6 | 11.2 | 38.1 | 122.0 |
| Guide | 23.5 | 36.2 | 31.1 | 120.5 |
| Planner | 31.1 | 21.9 | 32.4 | 111.6 |
| Santander | 38.7 | 44.7 | 37.9 | 176.6 |
| XP | 21.6 | 3.0 | 39.1 | 74.4 |

Figures 2-4 show the cumulative performance along the years for each of the DRL methods, comparing it with Ibovespa.

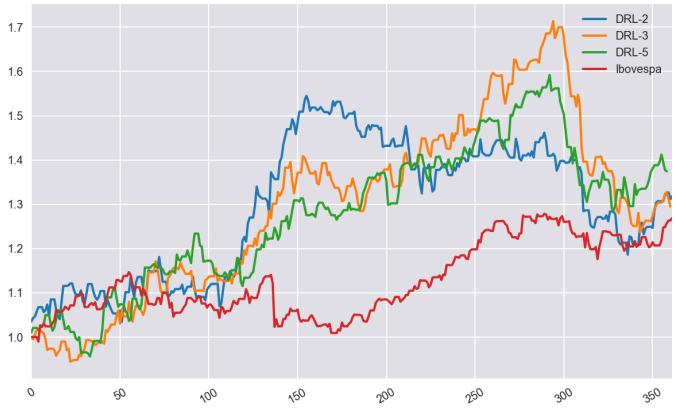


Fig. 2. Cumulative performance in 2017

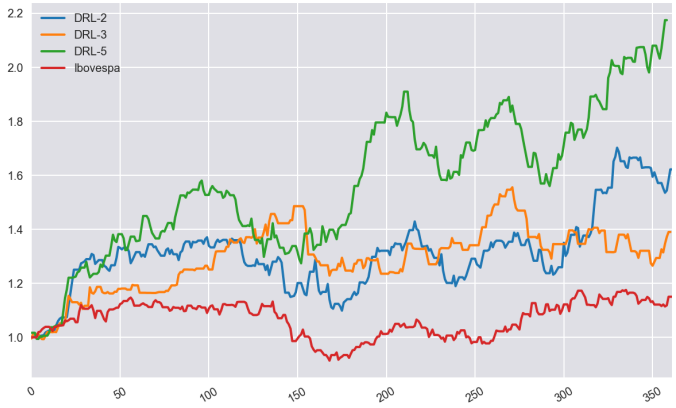


Fig. 3. Cumulative performance in 2018

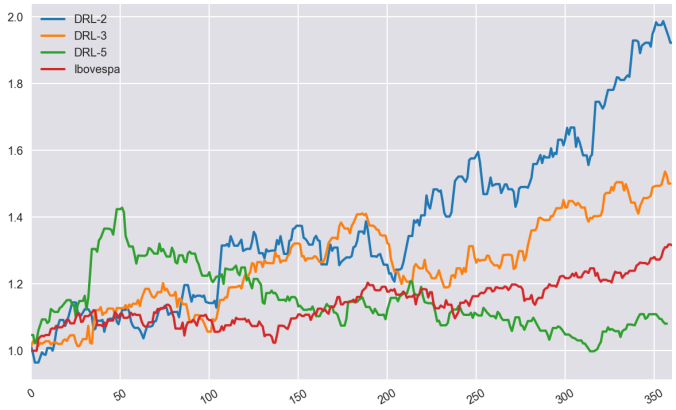


Fig. 4. Cumulative performance in 2019

Analyzing these graphs, we can calculate a classical risk metric for each method, called percentual maximum drawdown. A drawdown represents the total percentage loss of

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

capital experienced by the system before it starts winning again, considering the close prices of each day. The maximum drawdown is the highest drawdown occurred during the considered period, and represents a way to evaluate the risk associated with accepting the decisions of the trading system. Then, a low maximum drawdown is preferred as this indicates that losses from investment were small.

Table III shows the maximum drawdowns obtained by the three DRL methods and Ibovespa for each year and the average maximum drawdown in the last column. We do not present the drawdown of the benchmarks since only the monthly returns of each one are available [51], and not the daily returns. As can be noted, in 2017 and 2019 the DRL methods maximum drawdown values were higher than that of Ibovespa, while in 2018 the maximum drawdown of Ibovespa was higher than all the others, with the DRL-3 method presenting the lowest of them, with value equal to $18.7\%$, very close to the others.

It is interesting to note that although method DRL-2 was the one with the highest accumulated return, as shown in Table II, it was also the one with the lowest average drawdown of the three methods, thus having obtained a better performance in risk and reward criteria compared to the other methods with length window of 3 and 5. This can be explained by the fact that prices further back in the history have much less correlation to the current moment than that of recent ones.

TABLE III
MAXIMUM DRAWDOWN OF DRL AND IBOVESPA

| Method / Benchmark | Maximum Drawdown (%) | | | |
|---|---|---|---|---|
| | 2017 | 2018 | 2019 | Average |
| DRL-2 | 23.2 | 19.8 | 13.0 | 18.7 |
| DRL-3 | 27.9 | 18.7 | 15.7 | 20.7 |
| DRL-5 | 19.2 | 19.4 | 30.1 | 22.9 |
| Ibovespa | 12.0 | 20.4 | 10.0 | 14.1 |

## V. CONCLUSION

This paper proposed a Deep Reinforcement Learning day trading system, which considers the constraints imposed by the stock market, such as liquidity, latency, slippage and transaction costs. The proposed trading system uses a Deep Deterministic Policy Gradient (DDPG) algorithm to solve a series of asset allocation problems in order to define the percentage of capital that must be invested in each asset at each period. DDPG is a model-free, off-policy actor-critic method that can learn policies in high-dimensional and continuous action and state spaces, like the ones normally found in financial market environments.

Compared with previous works of asset allocation using reinforcement learning, we developed a trading system that makes exclusively day trade operations, which has at least three major advantages compared to swing trade: (i) lower trading costs, (ii) risk controlled outside trading hours and (iii) intraday leverage. The system buys up to 10 stocks in the opening auction of each trading session and keeps the stocks purchased until the end of the day, when they are sold during the closing auction.

The proposed day trading system was tested in B3 (Brazilian Stock Exchange) and a series of experiments were performed from 2017 to 2019 and compared with ten benchmarks, including Ibovespa and the stock portfolios suggested by the main Brazilian banks and brokers during these years. The results were evaluated considering return and risk performance metrics and showed that the proposed DRL method outperformed the benchmarks by a huge margin. The best results obtained by the algorithm had a cumulative percentage return of $311\%$ in three years, with an annual average maximum drawdown of $19\%$.

Despite the promising results obtained so far, as future work directions we want to explore larger observation spaces, with other data than just the close/open relative prices, including technical and also fundamentalist indicators as inputs to the predictor network. We intend to do it using a more systematic approach, and using feature selection methods to discover the features that would give the network more predictive power.

As we are only executing day trade operations, it is possible to obtain better returns by making purchases at prices closer to the price of the day's minimum and sales at prices closer to the prices of the day's maximum, instead of always buy at the opening and sell at the closing prices. Therefore, we intend to adapt the trading system to a more dynamic one capable of executes operations at anytime within the trading session and not only at auctions.

The main weakness of the current trading system is that it makes long-only trades. Therefore, in a bear market, it is likely to lose money unless it leaves all capital in cash. In this way, we intend to adapt it to be able to short selling.

## ACKNOWLEDGMENTS

This work was partially funded by the Brazilian National Institute of Science and Technology for the Web (grant no. 573871/2008-6), MASWeb (grant FAPEMIG/PRONEX APQ-01400-14), CAPES, CNPq (grant 459301/2014-4), Finep, and Fapemig.

## REFERENCES

[1] M. de Prado, "Advances in Financial Machine Learning," Wiley Publishin, 1st. edition, 2018.
[2] J. Heaton, N. Polson, and Jan Witte, "Deep learning for finance: deep portfolios," Applied Stochastic Models in Business and Industry, 2016.
[3] G. Atsalakis and K. Valavanis, "Surveying stock market forecasting techniques - part ii: Soft computing methods," Expert Systems with Applications, vol. In Press, Corrected Proof, 2008.
[4] Refinitiv, "Smarter Humans. Smarter Machines. Insights from the Refinitiv 2019 Artificial Intelligence / Machine Learning Global Study," 2019, https://www.refinitiv.com/content/dam/marketing/en_us/documents/reports/refinitiv-ai-ml-survey-report.pdf, visited in January 2020.
[5] D. Britz, "Introduction to Learning to Trade with Reinforcement Learning," 2018, https://www.cnblogs.com/DjangoBlog/p/9285956.html, visited in January 2020.
[6] M. Morales, "Grokking Deep Reinforcement Learning," Manning Publications, 2020.
[7] R. Sutton and A. Barto, "Introduction to Reinforcement Learning," MIT Press, 2nd edition, 2018.
[8] B3, http://www.b3.com.br, visited in January 2020.
[9] SmarttBot, www.smarttbot.com, in Portuguese, visited in January 2020.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

[10] L. Martinez, D. da Hora, J. Palotti, W. Meira Jr. and G. Pappa, "From an artificial neural network to a stock market day-trading system: a case study on the BM&F BOVESPA", International Joint Conference on Neural Networks (IJCNN), 2009.

[11] S. Niaki and S. Hoseinzade, "Forecasting SP 500 index using artificial neural networks and design of experiments", Journal of Industrial Engineering International, Vol. 9(1), pp. 1, 2013.

[12] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", Nature, Vol. 521, pp.436—444, 2015. Lee,

[13] J. Schmidhuber, "Deep learning in neural networks: An overview", Neural Networks, Vol. 61, pp. 85—117, 2015.

[14] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning", The MIT Press, Vol. 1, 2016.

[15] D. Silver and D. Hassabis, "AlphaGo: mastering the ancient game of Go with Machine Learning", Research Blog, 2016.

[16] D Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play", Science, Vol. 362, pp. 1140—1144, 2018.

[17] S. Almahdi and S. Yang, "An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown," Expert Systems With Applications, vol. 87, pp. 267–279, 2017.

[18] S. Almahdi and S. Yang, "A constrained portfolio trading system us- ing particle swarm algorithm and recurrent reinforcement learning", Expert Systems With Applications, Vol. 130, pp. 145–156, 2019.

[19] F. Bertoluzzo and M. Corazza, "Testing different Reinforcement Learning configurations for financial trading: Introduction and applications", Procedia Economics and Finance, Vol. 3, pp. 68–77, 2012.

[20] P. Casqueiro and A. Rodrigues, "Neuro-dynamic trading methods", European Journal of Operational Research, Vol. 175, pp. 1400–1412, 2006.

[21] M. Dempster and V. Leemans, "An automated FX trading system using adaptive reinforcement learning", Expert Systems With Applications, Vol. 30, pp. 543–552, 2006.

[22] Y. Deng, B. Feng, Y. Kong, Z. Ren and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading", IEEE Transactions on Neural Networks and Learning Systems, Vol. 28 pp. 653—664, 2016.

[23] D. Eilers, C. Dunis, H. Mettenheim, and M. Breitner, "Intelligent trading of seasonal effects: A decision support algorithm based on reinforcement learning", Decision Support Systems, Vol. 64, pp. 100–108, 2014.

[24] G. Jeong and H. Kim, "Improving financial trading decisions using deep Q-learning: Predicting the number of shares, action strategies, and transfer learning", Expert system with applications, Vol. 117, pp.125—138, 2019.

[25] Z. Jiang, D. Xu, and J. Liang. A deep reinforcement learning framework for the financial portfolio management problem. arXiv preprint arXiv:1706.10059, 2017.

[26] J. Moody and M. Saffell, "Learning to trade via Direct Reinforcement", IEEE Transactions On Neural Networks, Vol. 12, pp.875–889, 2001.

[27] J. Moody, L. Wu, Y. Liao and M. Saffell, "Performance Functions and Reinforcement Learning for Trading Systems and Portfolios", Journal of Forecasting, Vol. 17, pp. 441–470, 1998.

[28] R. Neuneier, "Optimal Asset Allocation using Adaptive Dynamic Program- ming", Advances in Neural Information Processing Systems, pp. 952–958, 1996.

[29] R. Neuneier, "Enhancing Q-Learning for Optimal Asset Allocation", Advances in Neural Information Processing Systems, pp. 936–942, 1998.

[30] J. O, J. Lee, J. Lee, and B. Zhang, "Adaptive stock trading with dy- namic asset allocation using reinforcement learning", Information Sciences, Vol. 176, pp. 2121–2147, 2006.

[31] P. Pendharkar and P. Cusatis, "Trading financial indices with reinforcement learning agents", Expert Systems with Applications, Vol. 103, pp.1–13, 2018.

[32] X. Zhang, Y. Hu, K. Xie, W. Zhang, L. Su, and M. Liu, "An evolutionary trend reversion model for stock trading rule discovery. Knowledge-Based Systems", Vol. 79, pp. 27–5, 2015.

[33] J. Cumming, "An investigation into the use of reinforcement learning techniques within the algorithmic trading domain", Master's thesis, Imperial College London, United Kiongdoms, 2015. http://www.doc.ic.ac.uk/teaching/distinguished-projects/2015/j.cumming.pdf, visited in January 2020.

[34] Y. Deng, F. Bao, Y. Kong, Z. Ren and Q. Dai, "Deep Direct Reinforcement Learning for Financial Signal Representation and Trading", IEEE Transactions on Neural Networks and Learning Systems, Vol. 28, no. 3, pp. 653–664, 2017.

[35] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, "Continuous control with deep reinforcement learning". arXiv preprint arXiv:1509.02971, 2015.

[36] A. Filos, "Reinforcement Learning for Portfolio Management", MEng Dissertation, 2018.

[37] Z. Liang, H. Chen, J. Zhu, K. Jiang and Y. Li, "Adversarial deep reinforcement learning in portfolio management", arXiv preprint arXiv:1808.09940, 2018.

[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O Klimov, "Proximal Policy Optimization Algorithms", ArXiv:1707.06347, 2017.

[39] B3, "Equities and Investment Funds Fees". http://www.b3.com.br/en_us/products-and-services/fee-schedules/listed-equities-and-derivatives/equities/equities-and-investment-funds-fees/spot/, visited in January 2020.

[40] C. Kirkpatrick II and J. Dahlquist, "Technical analysis: The complete resource for financial market technician", FT Press, 2010.

[41] A. Lo, H. Mamaysky and J. Wang, "Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation", The Journal of Finance, Vol. 55(4), pp. 1705-–1770, 2000.

[42] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra and M. Ried-miller, "Deterministic Policy Gradient Algorithms", Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 387–395, 2014.

[43] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M Riedmiller, "Playing Atari with deep reinforcement learning", pp. 1–9, 2013.

[44] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg and D. Hassabis, "Human-level control through deep reinforcement learning", Nature, Vol. 518, no. 7540, pp. 529–533, 2015.

[45] C. Zhang, L. Zhang and C. Chen, "Deep Reinforcement Learning for Portfolio Management", 2017.

[46] J. Chen (Investopedia), "Liquidity", https://www.investopedia.com/terms/l/liquidity.asp, visited in January 2020.

[47] J. Chen (Investopedia), "Slippage", https://www.investopedia.com/terms/s/slippage.asp, visited in January 2020.

[48] J. Chen (Investopedia), "Odd Lot", https://www.investopedia.com/terms/o/oddlot.asp, visited in January 2020.

[49] W. Kenton (Investopedia), "Round Lot", https://www.investopedia.com/terms/r/roundlot.asp, visited in January 2020.

[50] Black Rock, www.blackrock.com, visited in January 2020.

[51] Carteira Valor, https://valor.globo.com/valor-data/carteira-valor/, in Portuguese, visited in January 2020.

[52] Valor Economico, https://valor.globo.com/, in Portuguese, visited in January 2020.

[53] Corretora Toro Investimentos, https://www.toroinvestimentos.com.br/, in Portuguese, visited in January 2020.

[54] Corretora Warren, https://warrenbrasil.com.br/, in Portuguese, visited in January 2020.

[55] Corretora Clear, https://clear.com.br/, in Portuguese, visited in January 2020.

[56] Inter DTVM, https://www.bancointer.com.br/inter-dtvm/, in Portuguese, visited in January 2020.

[57] CM Capital Markets, https://www.cmcapital.com.br/, in Portuguese, visited in January 2020.


978-1-7281-6926-2/20/$31.00 ©2020 IEEE