Reinforcement Learning for Automated Investment in the Brazilian Stock Market: A Comparative Study of DQN, PPO, and Their Recurrent Versions

Paulo R. Sturion[1], **André Carlos P. de L. F. de Carvalho**[2]

[1]Instituto de Física de São Carlos – Universidade de São Paulo (USP)

[2]Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo (USP)

São Carlos - SP, Brazil

prsturion@usp.br, andre@icmc.usp.br

Abstract. *This study investigates the application of Reinforcement Learning (RL) in the development of automated investment agents in the Brazilian stock market (B3), focusing on short-term strategies (swing trading). The DQN and PPO algorithms and their recurrent versions (R-DQN and R-PPO) were compared. The simulation environment used historical data from five Brazilian companies, with technical indicators as input features. To ensure a fair comparison, data exposure was controlled and a standardized reward function was adopted. The results showed that DQN-based algorithms outperformed those based on PPO, and that the recurrent versions performed better than the traditional ones.*

Resumo. *Este trabalho investiga a aplicação de Aprendizado por Reforço no desenvolvimento de agentes de investimento automatizado no mercado de ações brasileiro (B3), com foco em estratégias de curto prazo (swing trading). Foram comparados os algoritmos DQN, PPO e suas versões recorrentes (R-DQN e R-PPO). O ambiente de simulação utilizou dados históricos de cinco empresas brasileiras, com indicadores técnicos como atributos de entrada. Para garantir comparação justa, controlou-se a exposição aos dados e adotou-se uma função recompensa padronizada. Os resultados mostraram que algoritmos baseados em DQN superaram os de PPO, e que as versões recorrentes tiveram desempenho superior às tradicionais.*

1. Introdução

Mudanças muito rápidas nas cotações de ações em bolsas de valores dificultam decisões ágeis por operadores humanos. Para lidar com esse desafio, tecnologias capazes de gerenciar e otimizar carteiras de investimento sem intervenção humana — conhecidas como investimento automatizado — têm se popularizado no Brasil e no mundo, especialmente com o avanço de técnicas baseadas em Inteligência Artificial (IA).

Segundo pesquisa da IBM (2024), 41% das empresas instaladas no Brasil já implementaram IA ativamente em seus negócios, e 73% dos profissionais de TI aceleraram sua adoção nos dois anos anteriores [IBM 2024].

Dentre as tecnologias de IA, o Aprendizado de Máquina (AM) se destaca como uma das principais formas de suporte à tomada de decisão, com aplicações significativas

no mercado financeiro, em particular o Aprendizado por Reforço (RL) (seção 3), seja no desenvolvimento de novas arquiteturas, seja na escolha do algoritmo mais adequado — como investigado neste trabalho.

Entre os algoritmos amplamente utilizados na literatura de RL, o Proximal Policy Optimization (PPO), introduzido por [Schulman et al. 2017], ganhou notoriedade por sua eficiência e simplicidade. Já os algoritmos baseados em redes neurais recorrentes — Recurrent Reinforcement Learning (RRL) — têm mostrado resultados promissores no mercado financeiro, superando métodos tradicionais em trabalhos comparativos [Zhang et al. 2020, Moody and Saffell 2001, Du et al. 2016]. Assim, este estudo propõe uma comparação entre o PPO e sua versão recorrente, juntamente com o Deep Q-Learning (DQN) — um dos algoritmos de RL mais tradicionais da literatura — e sua contraparte recorrente, aplicados ao *trading*.

A principal contribuição deste trabalho está, portanto, em estabelecer uma comparação clara e sistemática da eficiência desses métodos no contexto do investimento automatizado em ações na bolsa brasileira, com foco em operações do tipo *swing trade*.

2. Referencial Teórico

Nesta seção, são apresentados os principais conceitos necessários para este trabalho: i) Aprendizado por Reforço; ii) Estratégias de investimento e *swing trade*; iii) O mercado de ações brasileiro (B3).

2.1. Aprendizado por Reforço

RL é uma área de AM voltada para a tomada de decisões sequenciais por meio da interação de um agente com um ambiente. A cada ação tomada, o agente recebe uma recompensa e observa um novo estado, ajustando seu comportamento com o objetivo de maximizar as recompensas acumuladas ao longo do tempo.

Diferentemente de abordagens supervisionadas, o RL não requer rótulos explícitos, aprendendo com a própria experiência. Entre os algoritmos clássicos estão o Q-Learning e sua versão com redes neurais, o DQN, que estimam o valor de uma ação em um estado com base na equação de Bellman. Outra abordagem importante são os métodos de *Policy Gradient*, como o REINFORCE e o Proximal Policy Optimization (PPO), que aprendem diretamente uma política estocástica para maximizar a recompensa esperada.

2.2. Estratégias de Investimento - *Swing Trade*

O *Swing Trade* (ST) é uma estratégia de negociação de ativos com duração de alguns dias, intermediando entre o *Day Trade* e o *Position Trade*. Essa abordagem apresenta vantagens para aplicações em RL: dados diários são amplamente disponíveis e consistentes, os custos de transação são menores que em operações de alta frequência, e a demanda computacional é mais moderada, o que torna o ST mais viável para testes e aplicações automatizadas.

2.3. Mercado de Ações Brasileiro - B3

A B3 é a bolsa de valores oficial do Brasil, onde são negociados ativos como ações, títulos e derivativos. É um ambiente regulamentado e transparente, com ampla disponibilidade de dados históricos, o que favorece aplicações com aprendizado de máquina.

Algoritmos de RL já são utilizados há algum tempo no mercado brasileiro de ações. [Conegundes and Pereira 2020], por exemplo, utilizaram o algoritmo DDPG para alocação de capital na B3 e obtiveram retorno cumulativo três vezes superior ao investimento inicial em três anos. Este trabalho segue essa linha, explorando a aplicação de RL em um ambiente real e desafiador, ainda pouco estudado academicamente, como dito pelo próprio autor.

3. Trabalhos Relacionados

Com o advento do *Deep Learning* e, consequentemente, a possibilidade de lidar com grandes volumes de dados complexos, a maioria dos trabalhos envolvendo RL para *trading* concentra-se no uso de redes profundas [Sattarov et al. 2020, Conegundes and Pereira 2020, Azhikodan et al. 2019, Zhang and Maringer 2016, Dempster and Leemans 2006, Théate and Ernst 2021, Deng et al. 2016]. Vários deles demonstram resultados bastante promissores quanto à lucratividade, superando métodos clássicos [Théate and Ernst 2021] ou mesmo *benchmarks* de referência [Conegundes and Pereira 2020]. Os artigos [Azhikodan et al. 2019] e [Zhang and Maringer 2016] propõem alterações na estrutura tradicional dos algoritmos de RL: o primeiro incorpora uma CNN para análise de sentimentos em notícias financeiras, enquanto o segundo acopla um algoritmo genético para seleção de *features*, ambos com resultados animadores. Apesar da predominância de estudos focados no mercado estadunidense, há também abordagens voltadas a outros contextos. [Sattarov et al. 2020] explora o mercado de criptomoedas, com lucros mensais de 14,4% a 74%; [Conegundes and Pereira 2020], por sua vez, trabalha com a B3, apresentando um retorno acumulado de 311% em 3 anos — cenário que reforça o potencial do mercado brasileiro sob o uso de RL.

Outra vertente se dedica à comparação de diferentes métodos de RL [Zhang et al. 2020, Pendharkar and Cusatis 2018, Moody and Saffell 2001, Du et al. 2016], propósito também deste artigo. [Pendharkar and Cusatis 2018], por exemplo, compara três métodos na alocação de ativos em portfólio — tarefa também presente em [Conegundes and Pereira 2020] e [Du et al. 2016], enquanto os demais focam no *trading* de ativos isolados. Em [Zhang et al. 2020, Moody and Saffell 2001, Du et al. 2016, Zhang and Maringer 2016, Dempster and Leemans 2006] os autores adotam especificamente a abordagem RRL, destacando sua superioridade no tratamento de séries temporais, como as do mercado financeiro.

Complementando os trabalhos específicos, [Sun et al. 2023] apresenta uma *survey* abrangente e recente sobre RL em *Quantitative Trading*, com um panorama completo sobre a literatura da área, incluindo direções futuras.

Dessa forma, observa-se que o uso de RL em finanças, especialmente em operações automatizadas de *trading*, tem evoluído com bons resultados. Ainda assim, nota-se uma escassez de estudos comparativos que avaliem de forma sistemática o desempenho de diferentes algoritmos tradicionais e recorrentes, especialmente no contexto da B3 e com foco em estratégias como *swing trade*. É nesse espaço que este trabalho se insere, contribuindo para preencher essa lacuna.

4. Métodos e Desenvolvimento

Nesta seção serão tratados os principais aspectos dos métodos para o desenvolvimento do trabalho e produção dos resultados. São eles: i) a coleta e pré-processamento dos dados; ii) o ambiente desenvolvido para simulação do mercado; iii) detalhes sobre os algoritmos utilizados; iv) detalhes sobre as redes neurais implementadas; v) e os métodos para a análise de desempenho dos agentes.

4.1. Conjunto de Dados

Para a coleta dos dados, foram escolhidas cinco ações de alta relevância no mercado de ações brasileiro: PETR4, VALE3, ABEV3, BBAS3 e ELET3 — respectivamente, Petrobras, Vale, Ambev, Banco do Brasil e Eletrobras. Essa seleção também garantiu uma boa diversidade setorial, abrangendo os setores de combustíveis fósseis, mineração, bebidas, bancário e energia elétrica.

Os dados históricos foram obtidos por meio da plataforma *Yahoo Finance*, compreendendo o período de 01/01/2000 até 31/12/2024, totalizando 24 anos. As informações diárias coletadas incluem: preço de abertura (Open), fechamento (Close), máxima (High), mínima (Low) e volume negociado (Volume).

Apesar de úteis, essas cinco variáveis básicas são altamente correlacionadas entre si e estão em número limitado. Por isso, foram derivadas novas *features*, amplamente utilizadas no mercado financeiro, com o objetivo de enriquecer o conjunto de entrada e fornecer ao modelo informações mais diversas e informativas. As onze novas variáveis criadas se dividem em três grupos: indicadores de preço (amplitude diária, spread intradiário, mudança percentual no fechamento, médias móveis de 50 e 200 dias, RSI, ATR e Stochastic Oscillator), indicadores de volume (OBV e ADL) e um indicador de risco (VaR).

4.2. Ambiente Simulado

O ambiente simulado representa o mercado financeiro com o qual o agente interage, incluindo suas ações possíveis, função recompensa e outras configurações gerais. Optou-se por treinar um agente especializado para cada ativo, dado que as diferenças setoriais entre empresas geravam padrões distintos e levavam à *catástrofe do esquecimento* em um agente generalista.

Inicialmente, o espaço de ações era composto por 11 possibilidades (vender, segurar ou comprar em 5 níveis), mas foi simplificado, pois o agente naturalmente binarizava suas escolhas aos extremos; afinal, o agente quer sempre maximizar sua recompensa. Após análises, reduziu-se o espaço final a duas ações: comprar tudo ou vender tudo, com o ambiente rejeitando ações inválidas, o que implica que manter a posição é implicitamente representado.

A função recompensa também passou por revisões. A versão inicial combinava múltiplos fatores — incluindo risco e transações — mas termos dependentes de ações passadas, como volatilidade e *drawdown*, mostraram-se ineficazes, possivelmente por violar a suposição markoviana da teoria de RL. A função final adota apenas a variação de preço multiplicada pela ação do agente:

$$r_t = \Delta p_t \cdot \textit{ação}_t \tag{1}$$

com $ação_t \in \{-1, 1\}$ representando venda ou compra. Isso torna o aprendizado mais direto e eficiente, desvinculando a recompensa da posição acumulada do agente (que implicava não-markovianidade).

Em termos de configurações gerais do ambiente, o agente começa com R$10.000 (valor escolhido arbitrariamente) e a taxa de transação foi fixada em 0,001%, alinhando-se à prática da B3. A divisão dos dados seguiu a proporção tradicional de 70% treino, 15% validação e 15% teste, com os dados mais recentes alocados ao teste, refletindo aplicabilidade do agente no mercado atual. A recompensa foi escalada por um fator de 20, determinado empiricamente, para favorecer o aprendizado.

4.3. Algoritmos Utilizados

Este trabalho utilizou quatro variantes de algoritmos de aprendizado por reforço profundo: DQN, R-DQN, PPO e R-PPO. Em todos os casos, buscou-se seguir abordagens típicas da literatura, com técnicas consagradas e hiperparâmetros padronizados.

DQN e R-DQN seguem a estrutura do *Q-Learning* com redes neurais, utilizando *Replay Buffer* e *Target-Network* para maior estabilidade no treinamento [Mnih et al. 2013]. A versão recorrente (R-DQN) incorpora camadas LSTM, adaptando o buffer para fornecer sequências respeitando os limites dos episódios e permitir o aprendizado de dependências temporais (vide [Goodfellow et al. 2016] para teoria de redes recorrentes).

PPO e R-PPO utilizam a versão *clip*, com atualização da política restringida por uma margem ϵ para garantir estabilidade [Schulman et al. 2017]. Foram incorporados o *Generalized Advantage Estimation* (GAE) para reduzir viés/variância e um termo de entropia na função custo da rede para incentivar a exploração. A rede apresenta duas saídas (política e valor), e a coleta é feita via *Rollout Buffer*. A versão recorrente (R-PPO) também emprega LSTMs com sequência temporal consistente.

A Tabela 1 resume os hiperparâmetros utilizados. Quando o parâmetro não se aplica a um algoritmo, é indicado com "–".

Tabela 1. Hiperparâmetros utilizados por algoritmo.

Parâmetro	**DQN**	**R-DQN**	**PPO**	**R-PPO**
Episódios	20.000	20.000	20.000	20.000
γ (fator de desconto)	0,99	0,99	0,99	0,99
ϵ (exploração)	$1 \rightarrow 0{,}01$	$1 \rightarrow 0{,}01$	–	–
λ (GAE)	–	–	0,95	0,95
ϵ-*clip*	–	–	0,3	0,3
Atualização da *Target-Network*	100 transições	100 transições	–	–
Épocas de atualização	–	–	16	16
Batch size	128	4	128	4
Tamanho da sequência	–	32	–	32
Treinamento a cada	8 transições	8 transições	–	–

Para leitores menos familiarizados, vale lembrar que ϵ (exploração) controla a aleatoriedade na política de exploração, enquanto γ determina o peso das recompensas futuras. A técnica GAE (controlada por λ) busca um equilíbrio entre viés e variância na estimativa da vantagem.

4.3.1. Equiparação dos algoritmos

Para permitir uma comparação justa entre algoritmos distintos, igualou-se o número total de transições vistas durante o treinamento. A rede neural foi mantida idêntica em todos os métodos, e hiperparâmetros como frequência de atualização (DQN), número de épocas (PPO) e *batch size* foram ajustados para garantir volumes equivalentes de dados.

Quando parâmetros não eram diretamente comparáveis, buscou-se coerência ou escolheu-se bons valores empiricamente. O número de episódios foi fixado em todos os casos, por ser uma unidade natural de comparação.

Apesar disso, uma comparação perfeitamente justa é inatingível. Pequenas escolhas, como o valor ideal de γ, podem favorecer um método em detrimento de outro. Ainda assim, a sistematização adotada oferece uma base sólida para avaliar o desempenho relativo.

4.4. Redes Neurais

As redes neurais são componentes fundamentais nos algoritmos utilizados, permitindo o aprendizado em um ambiente tão complexo quanto o mercado financeiro. Neste trabalho, as arquiteturas das redes foram divididas em dois grupos: densas e recorrentes, associadas respectivamente aos algoritmos não recorrentes (DQN e PPO) e recorrentes (R-DQN e R-PPO).

Para os modelos DQN e PPO, foram utilizadas redes densas com duas camadas ocultas e função de ativação *Leaky ReLU*. Diversos tamanhos foram testados, mas uma configuração intermediária foi a mais eficiente: redes pequenas não aprendiam os padrões do mercado, enquanto redes grandes levavam ao colapso da política por *overfitting*. Adicionalmente, camadas *Dropout* foram incluídas para melhorar a generalização. Para o PPO, a saída da rede possui duas cabeças: uma para as ações e outra para o *state value*.

Nos modelos R-DQN e R-PPO, a arquitetura foi mantida próxima da versão densa, somente substituindo a segunda camada oculta por uma camada LSTM. Por construção, a camada LSTM já possui funções de ativação internas, dispensando ativação externa.

A Tabela 2 resume todas as arquiteturas utilizadas:

Tabela 2. Arquitetura das redes neurais utilizadas.

	DQN	**PPO**	**R-DQN**	**R-PPO**
Input	16	16	16	16
Hidden 1	128 (Dense)	128 (Dense)	128 (Dense)	128 (Dense)
Activation 1	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU
Dropout 1	0.1	0.1	0.1	0.1
Hidden 2	128 (Dense)	128 (Dense)	128 (LSTM)	128 (LSTM)
Activation 2	Leaky ReLU	Leaky ReLU	–	–
Dropout 2	0.1	0.1	0.1	0.1
Output	2	2 (ações), 1 (*value*)	2	2 (ações), 1 (*value*)

Todos os modelos adotaram o otimizador Adam com taxa de aprendizado de 5e-4. Para DQN e R-DQN, a função de perda utilizada foi o erro quadrático médio (*MSE loss*).

Já para PPO e R-PPO, a função de custo combinou três termos: a *PPO loss*, que é parte integrante da definição do algoritmo e treina a "cabeça"de ações; a *state value MSE*, que treina a saída correspondente ao valor do estado; e o termo de *Entropy*, que estimula a exploração durante o treinamento.

4.5. Métodos para Análise de Desempenho

Antes de analisarmos os resultados, é necessário definir os métodos empregados para este fim. Nesta seção, portanto, serão descritos os algoritmos de *baseline* e as métricas de desempenho utilizados.

4.5.1. Algoritmos de *baseline*

Nos experimentos, foram utilizados como *baseline* um conjunto de algoritmos que representam estratégias de negociação básicas que o agente treinado deve superar para que faça sentido aplicá-lo no mundo real. A seguir, os três algoritmos de *baseline* implementados serão listados e descritos.

- **Aleatório:** estratégia baseada em escolher uma ação aleatória a cada instante (dia) do episódio.
- ***Momentum*:** baseia-se na variação de preço do dia anterior para o dia atual: se o preço subiu de ontem para hoje, compra-se ações; se caiu, vende-se. A ideia é seguir a tendência recente, esperando que ela continue.
- ***Buy and Hold*:** consiste em comprar todas as ações possíveis no início do episódio e manter até o final. Essa estratégia revela a tendência natural de valorização da empresa. Espera-se que o agente supere essa abordagem ao vender nos momentos de queda e voltar a comprar antes da alta.

4.5.2. Métricas de desempenho

As métricas de desempenho são calculadas com base no histórico do agente no conjunto de teste. Elas podem refletir o retorno da estratégia, o risco assumido ou o desempenho ajustado ao risco (que considera ambos). As métricas escolhidas para avaliar os agentes (tanto de RL quanto os *baseline*) foram as seguintes:

- **Retorno acumulado percentual:** indica quanto o valor do portfólio aumentou ou diminuiu ao longo do tempo, em termos percentuais. Reflete o lucro ou prejuízo total obtido. Na fórmula abaixo, P é o valor do portfólio.

$$\text{Retorno total} = \left(\frac{P_{\text{final}}}{P_{\text{inicial}}} - 1 \right) \times 100$$

- ***Sharpe Ratio*:** mede o retorno excedente por unidade de risco, comparando a média dos retornos com sua volatilidade. Valores maiores indicam melhor desempenho ajustado ao risco. Na fórmula, R é o retorno do portfólio (variação percentual de P entre dois dias consecutivos); $\mathbb{E}$ representa a média, e σ o desvio padrão.

$$\text{Sharpe Ratio} = \frac{\mathbb{E}[R]}{\sigma[R]} \cdot \sqrt{252}$$

- ***Maximum Drawdown*:** representa a maior perda percentual do portfólio em relação ao pico anterior. É uma medida de risco extremo, que indica vulnerabilidade a quedas abruptas. É expressa em porcentagem, e quanto mais negativa, maior a perda.

$$\text{MDD} = \min_t \left(\frac{P_t - \max_{s \leq t} P_s}{\max_{s \leq t} P_s} \right) \times 100$$

5. Resultados e Discussão

Apresentam-se, a seguir, os resultados obtidos. A Figura 1 exibe a recompensa acumulada por episódio durante o treinamento dos algoritmos, considerando apenas o ativo BBAS3 (Banco do Brasil). A escolha desse ativo se deve à semelhança geral entre as curvas de aprendizado nos diferentes ativos, tornando-o representativo.

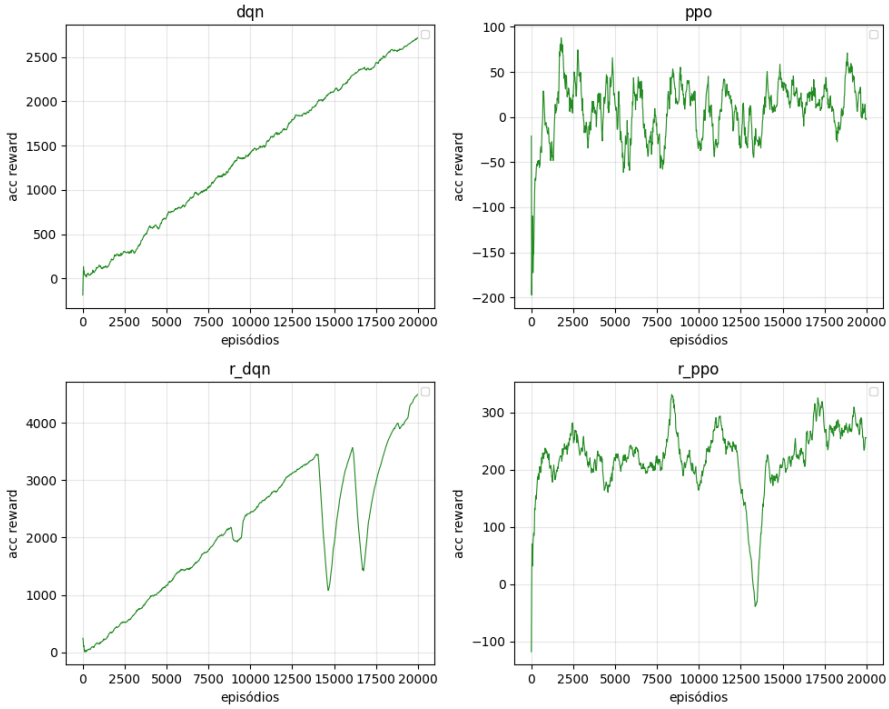

Figura 1. Recompensa acumulada por episódio de treinamento para cada algoritmo no ativo BBAS3 (suavizada com média móvel simples de janela 30).

Os algoritmos baseados em DQN e PPO apresentaram comportamentos similares dentro de seus respectivos grupos. DQN exibiu crescimento quase linear da recompensa, embora o R-DQN tenha sofrido quedas abruptas na segunda metade do treinamento, retornando depois ao padrão. Nenhuma abordagem atingiu um *plateau*, típico da convergência em RL. Investigou-se a causa, sem conclusões definitivas — o *plateau* pode ter ocorrido além do horizonte de treinamento. Já os métodos PPO mostraram maior variância e crescimento mais suave, também sem sinais claros de convergência.

Passa-se agora à análise qualitativa: o valor do portfólio ao longo do tempo no conjunto de teste. A Figura 2 mostra os gráficos para cada ativo, comparando as políticas aprendidas com as estratégias de *baseline*.

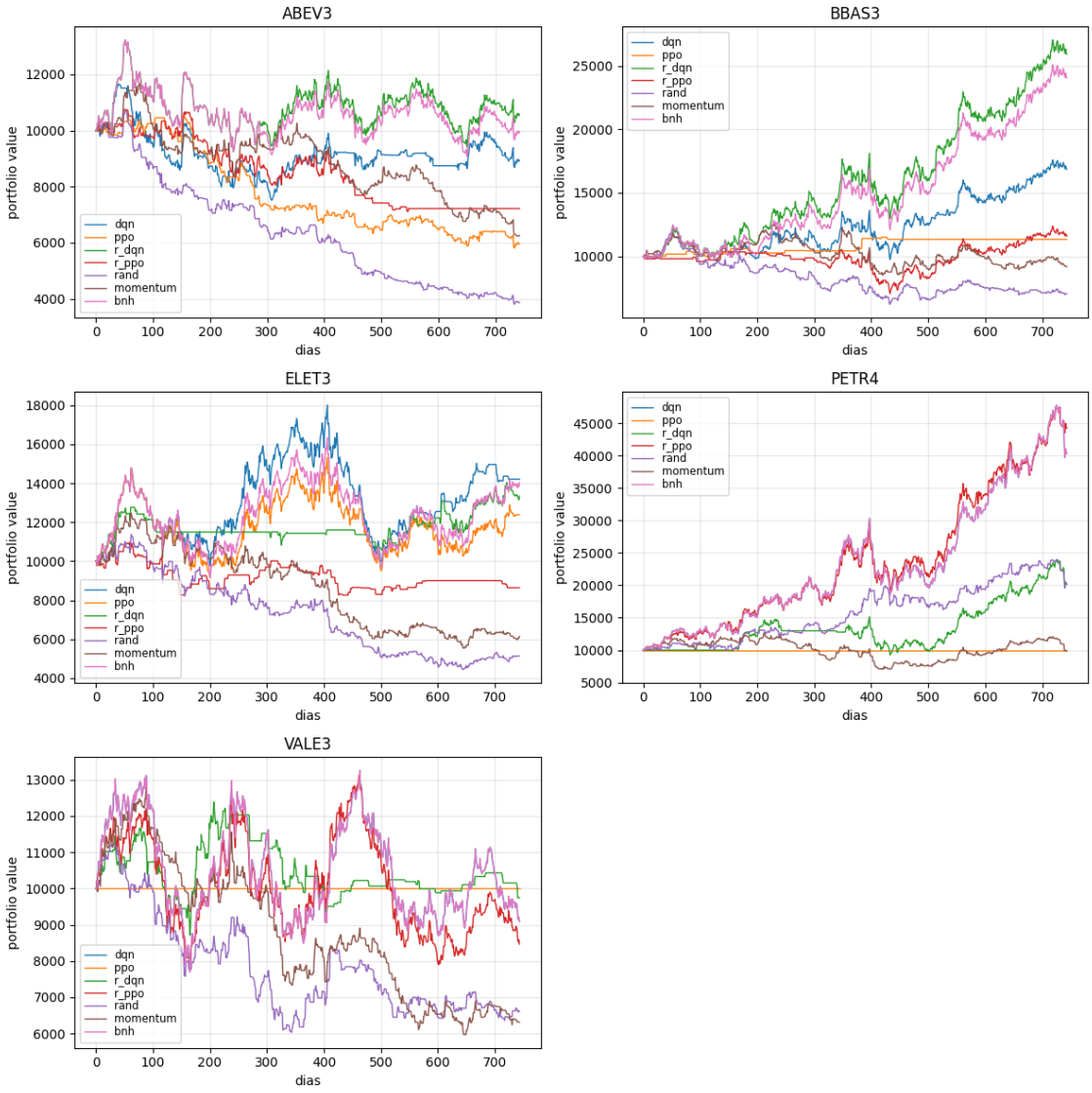

Figura 2. Valor do portfólio de cada política ao longo do tempo no conjunto de teste para diferentes ativos.

A estratégia *Buy and Hold* (bnh) mostrou-se difícil de superar, enquanto as estratégias Randômica (rand) e *Momentum* (momentum) tiveram desempenho fraco. Em geral, os agentes treinados superaram rand e momentum, sugerindo aprendizado de padrões não triviais — exceto nos casos de colapso, discutidos mais adiante. Em relação à bnh, superá-la foi raro, mas ocorreu: R-DQN foi superior em ABEV3 e BBAS3, por exemplo.

Contabilizando os desempenhos qualitativos frente às *baselines*: DQN superou todas as estratégias uma vez (ELET3) e empatou entre as melhores duas vezes (PETR4 e VALE3); PPO teve desempenho modesto, com leve relevância apenas em ELET3; R-DQN superou todas duas vezes (ABEV3 e BBAS3) e empatou entre as melhores uma vez (VALE3); R-PPO empatou duas vezes entre as melhores (PETR4 e VALE3). Todos os empates foram com bnh entre as melhores, o que, embora não reflita aplicabilidade prática, permite avaliar o potencial dos algoritmos.

De modo geral, os métodos baseados em DQN superaram os baseados em PPO

e, dentro de cada grupo, os modelos recorrentes tiveram melhor desempenho. A ordem qualitativa de desempenho final foi: R-DQN, DQN, R-PPO, PPO.

Seguem algumas observações sobre o colapso de determinadas políticas. Em PETR4 e VALE3 (e parcialmente em BBAS3), PPO convergiu para uma política que nunca compra ações, mantendo todo o capital em dinheiro, tanto no conjunto de teste quanto no de treinamento. Isso contrasta com o comportamento visto em outros ativos, sugerindo alta sensibilidade do PPO às características particulares de cada um. DQN também colapsou em PETR4 e VALE3, adotando uma política de compra total, equivalente à bnh; mas diferentemente do PPO, o comportamento do algoritmo no conjunto de treino foi diverso, sugerindo um possível *overfitting*. Técnicas como *Dropout*, que aumentam a capacidade de generalização dos modelos, foram utilizadas, mas não impediram o colapso.

Para a análise quantitativa, com base nas métricas definidas anteriormente, a Tabela 3 apresenta o *Retorno Acumulado Percentual* para cada ativo e política. Os destaques em negrito indicam que R-DQN liderou em dois ativos, sendo o algoritmo mais eficaz em retorno acumulado.

Tabela 3. Retorno acumulado percentual por política por ativo.

	ABEV3	BBAS3	ELET3	PETR4	VALE3
DQN	-10.63	68.53	**42.08**	303.86	-9.10
PPO	-40.48	13.30	23.83	0.0	**0.0**
R-DQN	**5.73**	**159.12**	33.33	101.87	-2.55
R-PPO	-27.82	15.90	-13.61	**342.79**	-15.32
rand	-61.34	-29.50	-48.61	101.87	-33.79
momentum	-37.37	-8.29	-38.60	-1.47	-36.85
bnh	-0.56	140.37	40.35	303.86	-9.10

A Tabela 4 apresenta o *Sharpe Ratio*, métrica que combina retorno e risco. Valores maiores indicam maior eficiência risco-retorno. Mais uma vez, o R-DQN se destaca, liderando em três ativos e demonstrando políticas mais eficientes e menos voláteis.

Tabela 4. *Sharpe Ratio* por política por ativo.

	ABEV3	BBAS3	ELET3	PETR4	VALE3
DQN	-0.072	0.801	0.543	1.498	**0.065**
PPO	-1.04	0.729	0.385	0.0	0.0
R-DQN	**0.199**	**1.312**	**0.678**	0.967	0.042
R-PPO	-0.601	0.347	-0.251	**1.625**	-0.034
rand	-1.827	-0.522	-0.793	1.036	-0.479
momentum	-0.863	-0.027	-0.551	0.114	-0.595
bnh	0.114	1.211	0.510	1.498	**0.065**

A Tabela 5 apresenta o *Maximum Drawdown*, que mede a maior queda observada no valor do portfólio. Entradas com * indicam políticas colapsadas, tornando a métrica inválida. De novo, R-DQN lidera em três ativos.

Com base nessas três métricas, R-DQN se consolida como a melhor abordagem. Por fim, considerando-se cada um dos algoritmos, e somando todas as lideranças de cada um: R-DQN obteve 8, DQN 3 (uma empatada com bnh), R-PPO 2 e PPO 1. Reforça-se, assim, a ordem de desempenho: R-DQN, DQN, R-PPO, PPO.

Tabela 5. *Maximum Drawdown* por política por ativo. (*: política colapsada — métrica inválida.)

	ABEV3	BBAS3	ELET3	PETR4	VALE3
DQN	-35.56	**-28.43**	-44.74	-39.09	-41.3
PPO	-45.21	**-4.06***	-38.06	**0.0***	**0.0***
R-DQN	**-30.13**	-28.44	**-17.24**	-39.07	**-25.42**
R-PPO	-33.79	-32.73	-24.71	-34.78	-39.83
rand	-64.42	-47.65	-60.88	**-18.63**	-46.43
momentum	-46.63	-32.99	-55.53	-44.63	-53.32
bnh	-31.41	-28.44	-40.74	-39.09	-41.3

6. Conclusão

Este trabalho mostrou que, apesar do uso de técnicas sofisticadas de Aprendizado por Reforço (RL), superar estratégias básicas de *baseline* no mercado financeiro não é trivial. A complexidade e aleatoriedade do mercado, influenciado por variáveis externas não modeladas (como fatores geopolíticos), impõem limites ao desempenho dos agentes treinados.

Em relação aos algoritmos utilizados nos experimentos, os resultados indicam clara vantagem dos métodos baseados em DQN sobre os baseados em PPO. Ainda assim, essa conclusão deve ser vista com cautela, pois a equiparação de parâmetros (seção 4.3.1) pode ter limitado o potencial máximo de alguns modelos. Ademais, o PPO possui a vantagem de poder operar em espaços de ação contínuos, o que não foi explorado neste estudo.

Outro achado importante foi a superioridade consistente das versões recorrentes (R-DQN e R-PPO) sobre as respectivas versões tradicionais. Esse desempenho reforça a adequação de redes recorrentes para lidar com a natureza sequencial dos dados financeiros.

Como direções futuras, propõe-se: (i) investigar o comportamento anômalo das curvas de aprendizado; (ii) aplicar PPO com ações contínuas; (iii) incorporar técnicas mais avançadas como *Prioritized Experience Replay* ao DQN; e (iv) desenvolver estratégias mais eficazes de combate ao *overfitting* e ao colapso de políticas.

7. Agradecimentos

Pesquisa desenvolvida com utilização dos recursos computacionais do Centro de Ciências Matemáticas Aplicadas à Indústria (CeMEAI), financiados pela Fundação de Amparo à Pesquisa do Estado de São Paulo – FAPESP (proc. 2013/07375-0), e com apoio de bolsa da FAPESP (proc. 2024/04827-1) concedida ao primeiro autor.

Referências

Azhikodan, A. R., Bhat, A. G., and Jadhav, M. V. (2019). Stock trading bot using deep reinforcement learning. In *Innovations in Computer Science and Engineering: Proceedings of the Fifth ICICSE 2017*. Springer Singapore.

Bilgin, E. (2020). *Mastering Reinforcement Learning with Python*. Packt Publishing.

Conegundes, L. and Pereira, A. C. M. (2020). Beating the stock market with a deep reinforcement learning day trading system. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE.

Dempster, M. A. and Leemans, V. (2006). An automated fx trading system using adaptive reinforcement learning. *Expert Systems with Applications*, 30(3):543–552.

Deng, Y. et al. (2016). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3):653–664.

Drori, I. (2023). *The Science of Deep Learning*. Cambridge University Press.

Du, X., Zhai, J., and Lv, K. (2016). Algorithm trading using q-learning and recurrent reinforcement learning. *Positions*, 1(1):1–7.

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

IBM (2024). Estudo ibm: 41% das empresas no brasil já implementaram ativamente ia. `https://www.ibm.com/blogs/ibm-comunica/estudo-ibm-41-das-empresas-no-brasil-jaimplementaram-\\ativamente-inteligencia-artificial-em-seus-negocios/`. Acessado em 15 de fevereiro de 2024.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Moody, J. and Saffell, M. (2001). Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, 12(4):875–889.

Morales, M. (2020). *Grokking Deep Reinforcement Learning*. Manning Publications.

Pendharkar, P. C. and Cusatis, P. (2018). Trading financial indices with reinforcement learning agents. *Expert Systems with Applications*, 103:1–13.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229.

Sattarov, O. et al. (2020). Recommending cryptocurrency trading points with deep reinforcement learning approach. *Applied Sciences*, 10(4):1506.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sun, S., Wang, R., and An, B. (2023). Reinforcement learning for quantitative trading. *ACM Transactions on Intelligent Systems and Technology*, 14(3):1–29.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition.

Théate, T. and Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications*, 173:114632.

Zhang, J. and Maringer, D. (2016). Using a genetic algorithm to improve recurrent reinforcement learning for equity trading. *Computational Economics*, 47:551–567.

Zhang, Z., Zohren, S., and Roberts, S. (2020). Deep reinforcement learning for trading. *The Journal of Financial Data Science*.