BOLETIM
ECONOMIA EMPÍRICA
v. 6, n. 1 (2026) | ISSN: 2675-3391

Machine Learning: Inteligência Artificial (IA) Aplicada em Precificação de Ativos Financeiros por meio de Reinforcement Learning e Deep Q-Networks (DQN)

Leandro Pereira da Silva[†]
[†]Autor correspondente: leandrosilvabrasil@gmail.com

INFORMAÇÕES DO ARTIGO

Palavras-chave:

Reinforcement Learning
Redes Neurais
Taxa de Câmbio
Deep Q-Networks
Inteligência Artificial

RESUMO

O uso de Inteligência Artificial vem se tornando cada vez mais intensivo devido às suas características de inovação tecnológica. Entre os meios teóricos procura-se saber se o algoritmo tem a capacidade por si só de aprender sem a direta interferência humana. Assim, a metodologia *Reinforcement Learning* é utilizada em problemas relacionados ao uso de IA autônomas. O problema dessa pesquisa foi desenvolver um processo de precificação de ativos financeiros por meio desse instrumento. O objetivo foi gerar o valor futuro do ativo, com nível de acurácia satisfatório. E em prévia, os resultados demonstraram bom desempenho na precificação do dólar americano.

1. Introdução

Com o desenvolvimento de recursos tecnológicos na era contemporânea é possível utilizar instrumentos de IA, que antes não era possível devido a restrições tecnológicas. No entanto, uma pergunta relevante que pode surgir diante de tudo isso é: o que é considerado um sistema de inteligência artificial? Max Tegmark (2017 *apud* Hilpisch, 2020) nos ajuda a responder essa questão ao definir inteligência como a "capacidade para atingir objetivos complexos". Por exemplo, ao considerarmos a trajetória de uma variável em uma situação considerada complexa, e a partir desse, se for possível desenvolver uma tecnologia (algoritmo de máquina) que atinja o sucesso do objetivo proposto, cria-se o que é chamado de inteligência artificial.

E especificamente, ao se tratar desta pesquisa, a precificação e as análises de predição de ativos financeiros, em especial — moedas em termos de valores futuros, é um grande problema e desafio acadêmico. A complexidade, não linearidade e a falta de padrões comportamentais das séries temporais desses ativos levou a academia, ao longo da história, ao desenvolvimento de diversos modelos matemáticos e estatísticos para tentar contornar esse problema. Porém, com o avanço de algoritmos de Inteligência Artificial (IA), em especial técnicas metodológicas como o Aprendizado por Reforço (AR), aliado às Redes Neurais (RN) e o algoritmo *Q-Learning* têm se mostrado promissor no desafio de previsão em ambientes financeiros dinâmicos. Sendo que atualmente a possibilidade do uso de grandes volumes de dados nesses modelos vem permitindo melhorar a robustez e estimar melhor potenciais padrões nessas séries de dados, que antes não era possível de detecção (Deng et al., 2017; Géron, 2019; Mnih et al., 2015; Rao e Jelvis, 2023; Sutton e Barto, 2018; Yang et al., 2020).

Para esses instrumentos serem desenvolvidos, tem-se que os tipos mais utilizados em metodologias aplicadas em IA são: Aprendizagem Supervisionada, Aprendizagem não Supervisionada, e Aprendizagem por Reforço (AR). Este trabalho, em especial, foca e limita os processos metodológicos no algoritmo de AR. Isso em função do avanço promissor desse processo computacional e pela possibilidade de que tal técnica permite, em um futuro possível, gerar a chamada Inteligência Artificial Geral, ou especificamente a também chamada Singularidade Tecnológica em IA financeira (Barrat, 2013; Hilpisch, 2020; Kissinger et al., 2021; Kurzweil, 2005; Shah et al., 2025; Vinge, 1993). Dessa forma, existe uma "corrida" no mundo para se obter esse ganho tecnológico, e o prêmio pelo primeiro que atingir esse sucesso vai ser crucial em termos de competitividade.

Na última década, a pesquisa acadêmica e profissional aplicada em algoritmos de IA teve um significativo avanço. A metodologia de AR, geralmente combinada com Redes Neurais (RN), se estabeleceu entre as técnicas mais bem sucedidas em testes empíricos realizados na área financeira (Hilpisch, 2020, p. 42). Em que, segundo Ahlawat (2023, p. 1), o campo de finanças foi o primeiro na aplicação de algoritmos de IA, aliados a RN, no objetivo de criar estratégias de negociações já na década de 1980.

Desde então, a utilização da metodologia AR com algoritmos de IA começou a ganhar mais impulso a partir dos avanços de tal técnica nos conhecidos jogos de Xadrez e Go, a partir da segunda metade da década de 2010. A empresa DeepMind (pertencente à Alphabet-Google), por meio dos seus algoritmos AlphaZero e AlphaGo, precisou de apenas um tempo aproximado de nove horas no caso do jogo de xadrez e treze dias no jogo Go, para treinar sua IA, e deixá-la praticamente imbatível (Hilpisch, 2024, p. 16). Antes dessa época, esses jogos de computadores eram dominados pela IBM, que tinham também modelos computacionais complexos aliados à experiência de grandes jogadores, como, por

exemplo, mestres e campeões mundiais de Xadrez. No entanto, a DeepMind revolucionou essa área ao não levar em consideração a experiência humana no aprendizado direto dos jogos. Pois, uma característica do algoritmo AR é que ele tem a capacidade de aprender sozinho e ganhar experiência por conta própria. A partir desse feito, nesse caso se tornou impossível um humano ganhar da máquina, ao jogar uma partida de Xadrez ou Go (Kissinger et al., 2021, p. 44-45). Ou seja, a partir destes processos metodológicos, o algoritmo treinado (geralmente por meio da técnica de redes neurais) se torna seu "próprio professor", sem precisar de instruções humanas em termos de modelagem e estratégias para atingir o objetivo (Hilpisch, 2020; Géron, 2019). A DeepMind ainda trabalhou com IA, por meio de AR nos famosos jogos Atari 2600 Computer System da década de 70 e 80. Em que o algoritmo utilizado foi uma variante do *Q-learning* aplicado a uma rede neural convolucional, no qual os resultados alcançados em alguns jogos ficaram acima do desempenho humano (Deng et al., 2017; Mnih et al., 2015; Rao e Jelvis, 2023; Sutton e Barto, 2018; Yang et al., 2020).

Assim, o *insight* para esse trabalho surgiu com a possibilidade de se trabalhar com algoritmos que tenham a capacidade de evoluir por si só. No modelo de AR, o agente aprende e se adapta ao ambiente dinâmico que está inserido. A base metodológica ocorre a partir do princípio de receber estímulos externos de acordo com as escolhas tomadas pelo próprio algoritmo. Ou seja, quando o algoritmo toma uma decisão correta, este é premiado com uma recompensa, já uma tomada de decisão incorreta retorna uma penalidade. Já a finalidade desse tipo de sistema é alcançar o melhor resultado possível. E isso é considerado um dos principais conceitos de aprendizado por máquina, que é aprender pela experiência com a finalidade de simular o raciocínio humano (Ciaburro, 2019, p. 68). Ou seja, a IA extrai conclusões baseadas em resultados experimentais, e não necessariamente é totalmente alicerçada em modelos derivados da compreensão teórica humana (Kissinger et al., 2021, p. 126).

O problema de pesquisa foi entender como desenvolver um processo metodológico de precificação e previsão da taxa de câmbio (Dólar/Real), por meio de tecnologia de IA, e do modelo RL-*Q-Learning* e por meio de redes neurais Deep Q-Networks (DQN). O objetivo deste artigo foi gerar o valor futuro do ativo observado, com um nível de desempenho (acurácia) satisfatório, mensurado por meio de erros quadrados médios (MSE — *Mean Squared Error*). Espera-se ainda que o uso dessa metodologia apresente resultados regulares de previsibilidade dos ativos no curto e no médio prazo. A justificativa da utilização dessa técnica se deve ao fato de procurarmos aplicar um algoritmo que seja capaz de criar um agente tecnológico que construa e aprenda seu próprio caminho para atingir o objetivo estipulado em ambientes financeiros. E em prévia, os resultados encontrados por meio da avaliação da modelagem indicam que o modelo utilizado conseguiu prever com relativa precisão a volatilidade da taxa cambial Dólar/Real no período de médio prazo (período entre 30 e 150 dias), no sentido de estratégia de *hedge* cambial.

2. Referencial Teórico

O diferencial e o avanço científico da abordagem RL é que este tipo de algoritmo funciona sem um modelo teórico ou prático já observado empiricamente no mundo real. Ou seja, inicia-se o processo de treino e aprendizagem por meio de um sistema aleatório e sem nenhum conhecimento de domínio por parte da máquina, exceto as regras e restrições impostas à situação objetiva. Por exemplo, na metodologia IA *Machine Learning* e redes neurais aplicada em precificação de ativos financeiros, podemos utilizar a modelagem Black-Scholes-Merton como um meio para treinar a máquina a aprender a precificar o preço de um derivativo financeiro (opções, por exemplo). Já na abordagem AR, a ideia é que essa modelagem (Black-Scholes-Merton) não seja mais necessária. O algoritmo em questão procura aprender o preço do derivativo por meio dele próprio. Ou seja, teorias financeiras e suas respeitadas modelagens estatísticas e matemáticas não são mais necessárias para ensinar a máquina, nesses casos. Em outras palavras, como no caso do algoritmo AlphaZero da empresa DeepMind, em que a IA não utilizou aprendizado humano, como técnicas e estratégias de jogadores experientes de xadrez para poder vencer os seus oponentes — aqui, neste trabalho, as técnicas RL procuram aprender sem utilizar das modelagens financeiras conhecidas, para poder prever o preço de um ativo financeiro (Ahlawat, 2023, p. 2-3).

O modelo RL é conhecido por ser um agente do tipo "aprendizado de reforço livre de modelo", pois não necessita de um modelo pré-determinístico para aprender. Esse modelo de máquina utiliza programação dinâmica, que aprende por meio de recompensa a partir de observações, para na sequência processual obter uma política ótima de aprendizado (Deng et al., 2017; Mnih et al., 2015; Rao e Jelvis, 2023; Russell e Norvig, 2022; Yang et al., 2020). Segundo Rao e Jelvis (2023) e Russell e Norvig (2022), os conceitos básicos para o entendimento teórico dos algoritmos RL são: a teoria de otimização na equação de Bellman, a Iteração de Política Generalizada, o Processo de Decisão de Markov (MDPs), e a Programação Dinâmica. Sendo que essas metodologias são adicionadas a vários tipos de algoritmos, como, por exemplo, SARSA, *Q-Learning*, Redes Deep-Q, *Least-Squares Policy Iteration*, *Policy Gradient*, e redes neurais profundas para a aproximação de função.

A base conceitual de treinamento do algoritmo RL parte do que é chamado de comportamento adaptativo, considerando que isso é um tipo de estrategista de planejar movimentos analisando possíveis contramedidas opostas. Por exemplo, um jogador de basquete, para melhorar a sua performance como um bom pontuador, aprende ao treinar exaustivamente arremessos de bolas na cesta (Ciaburro, 2019, p. 34). Da mesma forma, a melhoria do desempenho do aprendizado de máquina é medida pela experiência/retorno do algoritmo em relação à tarefa dada. Essa tarefa pode ser, por exemplo, a realização de uma estimativa, classificação ou um exercício de previsão de uma série temporal. Já uma métrica estatística de desempenho (acurácia) utilizada para mensurar tal performance pode ser o erro quadrático médio (MSE), utilizando geralmente um grande volume de dados (*Big Data*) (Géron, 2019, p. 4).

Dessa forma, o modelo RL procura aprender por experiência própria, o que é observado após uma ação ser realizada

e premiada por uma recompensa pré-estipulada. Ou seja, o algoritmo mantém o registro dos estados observados em cada etapa, aprendendo o viés da variável observada. Um ponto importante nesses casos é que a máquina treina seu aprendizado em uma quantidade humanamente impossível de repetições. Se voltarmos para o caso do jogo de xadrez, mesmo para um jogador profissional, este consegue treinar suas estratégias e técnicas do jogo em uma quantidade limitada de vezes (Hilpisch, 2020; Russell e Norvig, 2022).

Para implementar tal processo, tem-se o método Deep Q-Learning que pode ser considerado uma evolução do método básico de aprendizagem (Ciaburro, 2019, p. 48-49). Assim, dado o objetivo estratégico que se busca (por exemplo, mensurar o valor de um ativo financeiro no tempo $t+1$), uma política de ação Q atribui valor a cada combinação de um estado em combinação com esta ação (Sutton e Barto, 2018; Watkins, 1989; Watkins e Dayan, 1992). Corroborando, Ciaburro (2019, p. 34) cita que:

> O objetivo da aprendizagem por reforço é aprender uma política para cada estado, s, no qual o sistema é que especifica uma ação para o agente, para que esse possa maximizar o reforço total que recebe durante a sequência de ação inteira. Essa técnica atualiza as funções de valor usando a recompensa imediata e o valor (descontado) do próximo estado em um processo chamado *bootstrap*. A iteração de política é geralmente aplicada a processos de decisão discretos de Markov, onde tanto o espaço de estado S quanto o espaço de ação A são conjuntos discretos e finitos.

A expressão (1) a seguir, apoiada em Hilpisch (2020, p. 260), Russell e Norvig (2022) e em Wu e Mao (2021, p. 82-84), formaliza esse contexto.

$$Q(S_t, A_t) = R_{t+1} + \max_a Q(S_{t+1}, a) \tag{1}$$

Em que S_t é o estado na etapa (tempo) t, A_t é a ação realizada no estado S_t, R_{t+1} é a recompensa direta da ação A_t, $0 < \gamma < 1$ é um fator de desconto, e $\max_a Q(S_{t+1}, a)$ é a máxima recompensa atrasada dada a ação ideal da política atual Q (Hilpisch, 2020, p. 260). Nesse sentido, Géron (2019, p. 476) cita que:

> O valor ótimo (Q-Value) do estado-ação (s, a), identificado como $Q^*(s, a)$, é a soma das recompensas futuras descontadas que o agente pode esperar, em média, depois de atingir o estado s e escolher a ação a, mas antes de ver o resultado desta ação, supondo que ela funcione de maneira ideal após essa ação.

O processo de aprendizagem inicia-se com todas as estimativas de Q-Value em zero e, em seguida, as atualiza com o algoritmo de Iteração Q-Value, como demonstra a Equação (2), a seguir (Géron, 2019, p. 476).

$$Q_{k+1}(s, a) \leftarrow \sum_{s'} T(s, a, s') \left[R(s, a, s') + \gamma \cdot \max_{a'} Q_k(s', a') \right] \quad \text{para todo } (s, a) \tag{2}$$

Em que $T(s, a, s')$ é a probabilidade de transição do estado s para o estado s' dado que o agente escolheu a ação a; $R(s, a, s')$ é a recompensa que o agente obtém quando passa do estado s para o estado s', dado que tenha escolhido a ação a; γ é a taxa de desconto (Géron, 2019, p. 475).

Então, define-se a política ótima, notada como $\pi^*(s)$ (Géron, 2019, p. 476):

$$\pi^*(s) = \arg\max_a Q^*(s, a) \tag{3}$$

E em seguida aplica-se este algoritmo ao processo de decisão de Markov, e depois executa-se o algoritmo de Iteração Q-Value (Géron, 2019, p. 476-477).

Muitos processos RL utilizam dados sintéticos para treinar seus modelos. Esses dados não reais gerados ajudam a evitar viés no resultado da precificação. De acordo com Hilpisch (2020, p. 71-72), a adição de ruído branco em uma série financeira permite gerar um volume ilimitado de dados para treinar o agente DQL.

O modelo RL geralmente utiliza redes neurais, e que segundo Géron (2019, p. 260) essas redes neurais foram introduzidas pela primeira vez em 1943, no artigo intitulado "Logical Calculus of Ideas Immanent in Nervous Activity", escrito pelo neurofisiologista Warren McCulloch e pelo matemático Walter Pitts. Já a técnica de Aprendizado Profundo (*Deep Learning* — DL) abrange todos os tipos de algoritmos baseados em redes neurais. Em que o termo profundo é geralmente usado apenas quando a rede neural possui mais de uma camada oculta (Hilpisch, 2020, p. 8).

Exemplificando, Hilpisch (2020, p. 257; p. 31-32) cita que um agente de rede neural interage com o ambiente, assim, este pode coletar dados que significam combinações de valores de recursos e rótulos. Esse conjunto de dados ampliado pode ser conceitualmente entendido como uma rede neural, que geralmente é treinada para aprender a correta ação, considerando um ambiente dado. Ou seja, a rede neural representa a política nesse estado, e o agente atualiza a política com base nas novas experiências aprendidas.

Em geral, a política de ação ótima não pode ser especificada na forma analítica, ou seja, na forma de uma tabela ou função matemática. Portanto, QL depende em geral de representações aproximadas para a política ótima. O uso de redes neurais profundas em agentes do tipo *Q-Learning* se deve ao fato de que essas RN têm aproximações bastante relevantes, e em geral, para análises de variáveis com características aleatórias, ou quase aleatórias, seus desempenhos são melhores do que se comparado a métodos mais tradicionais, como, por exemplo, o método de Mínimos Quadrados Ordinários (MQO) (Hilpisch, 2024, p. 31-33).

Para mensurar o desempenho da metodologia utilizada nesta pesquisa, utiliza-se a métrica de erro quadrado médio (Gujarati e Porter, 2011, p. 817). Em que esse tipo de mensuração abordado resulta em uma medida de quão bem o modelo se ajusta aos dados. Dessa forma, o MSE de um estimador $\hat{\theta}$ pode ser definido como:

$$MSE(\hat{\theta}) = (\hat{\theta} - \theta)^2 \tag{4}$$

E de acordo com Gujarati e Porter (2011, p. 824):

$$MSE(\hat{\theta}) = \text{variância de } \hat{\theta} \text{ mais o quadrado do viés} \tag{5}$$

Já a validação de um modelo de IA é diferente da validação de modelos financeiros tradicionais. Em modelos de IA, Hilpisch (2020, p. 390) corrobora ao citar que "dificilmente há um modelo que, com base em um conjunto parcimonioso de parâmetros, pode validar os resultados de um algoritmo complexo de IA".

2.1. Críticas e limitações do modelo

Uma crítica quanto à metodologia RL é a falta de critérios de escolhas que este modelo assume automaticamente no início do processamento de dados. Assim, o início do processamento computacional é lento, pois o algoritmo não tem ideia de quais são as probabilidades de transição do estado-ação ($T(s, a, s')$), e também não sabe quais serão as recompensas utilizadas ($R(s, a, s')$). Isso pode complicar demasiadamente a otimização da modelagem, e elevar excessivamente o tempo de otimização (Géron, 2019, p. 477-479).

Uma outra complicação para o sucesso na abordagem RL se deve ao fato de testarmos esse modelo de IA em um ambiente aberto, que é o caso do ambiente financeiro. Nesse caso tem-se duas desvantagens: dados limitados (minimizado por adição de dados sintéticos) e nenhum impacto das ações (o teste é local, e não no ambiente real) (Hilpisch, 2024, p. 52-53). Ainda outra limitação da abordagem RL aplicado em ambiente financeiro é que o modelo se baseia na busca de padrões de trajetórias das variáveis (Hilpisch, 2024, p. 57). Por exemplo, nesta pesquisa captamos os dados diários da cotação do dólar, em moeda brasileira, de janeiro de 2010 a dezembro de 2024, o que significa que temos uma amostra n aproximadamente igual a 4.000. E em termos quantitativos essa amostra não é considerada grande o suficiente para esse tipo de algoritmo.

3. Metodologia

A base metodológica de AR é o sistema de aprendizagem chamado de agente (Géron, 2019, p. 14). No qual o algoritmo aprende por tentativa e erro, recebendo uma recompensa por realizar uma determinada ação. A metodologia AR ainda utiliza programação dinâmica, que representa um conjunto de algoritmos que são usados para calcular uma política ótima dado um modelo do ambiente na forma de um Processo de Decisão de Markov (Ahlawat, 2023; Ciaburro, 2019; Deng et al., 2017; Géron, 2019; Hilpisch, 2020; Mnih et al., 2015; Rao e Jelvis, 2023; Yang et al., 2020).

Dessa forma, foi desenvolvido nesta pesquisa um modelo Deep Q-Network (DQN) usando uma camada convolucional, adicionado na sequência por uma camada de *pooling* máximo, mais uma camada de achatamento e também por duas camadas densas. O modelo é treinado usando a função de perda do erro quadrático médio (MSE) e o otimizador Adam. Sendo que, após o treinamento, o modelo prevê a taxa de câmbio USD/BRL para cada sequência de entrada dada (cotação diária da taxa de câmbio).

Uma característica metodológica de AR é que este necessita de alguns aspectos fundamentais em sua construção. Sendo eles: ambiente, estado, agente, ação, etapa, recompensa, alvo, política e episódio. E como esta pesquisa aplica AR na abordagem da disciplina de finanças, tem-se que: i) o ambiente é o de negociação em mercado financeiro; ii) o estado são os níveis de preços atuais e históricos dos ativos; iii) o agente é o elemento humano que está utilizando a IA para negociar o ativo financeiro no mercado; iv) a ação é a própria operação de aquisição do ativo financeiro ("comprado ou vendido" no ativo); v) a etapa se diz ao tempo ou intervalo de tempo utilizado na operação (*daytrade*, *swing trade*, etc.); vi) a recompensa é o lucro (positivo ou negativo) da operação; vii) o alvo é o nível de lucro estipulado na estratégia da negociação; viii) a política faz parte da estratégia estabelecida em que se estabelece nível de risco, lucro alvo e tipos de concepções de operações permitidas para a IA; e ix) o episódio é o conjunto de etapas (tempo) estabelecidas até que o sucesso da operação seja alcançado (Deng et al., 2017; Hilpisch, 2020; Mnih et al., 2015; Rao e Jelvis, 2023; Yang et al., 2020).

3.1. Aprendizado por reforço e Deep Q-Networks (DQN)

Como já mencionado anteriormente, a abordagem de Deep Q-Networks (DQN) é baseada no AR, onde um agente interage com um ambiente, observa os estados e toma ações para maximizar uma recompensa cumulativa ao longo do tempo (Ciaburro, 2019, p. 65). De acordo com Hilpisch (2020), Sutton e Barto (2018) e Wu e Mao (2021, p. 82-84), a função valor de ação $Q(s, a)$ mapeia o estado s e a ação a para um valor que representa a recompensa esperada. Assim, a função Q é atualizada de acordo com a Equação (6) de Bellman:

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a') \tag{6}$$

Em que r é a recompensa imediata recebida após tomar a ação a no estado s; γ é o fator de desconto, que controla a importância das recompensas futuras ($0 \leq \gamma < 1$); s' é o próximo estado após a ação a; e a' é a próxima ação a ser tomada no estado s'.

3.2. Rede neural profunda para aproximação de Q

A rede neural profunda usada no DQN tem três principais parametrizações de arquitetura: (i) Entrada: o vetor de estado s, que representa a composição atual da carteira (pesos dos ativos); (ii) Camadas Ocultas: múltiplas camadas densas com funções de ativação ReLU; (iii) Saída: o valor $Q(s, a)$ para cada ação possível a, que corresponde a diferentes alocações de ativos. Já a função de perda usada para treinar a rede é a média do erro quadrático entre a predição atual $Q(s, a; \theta)$ e o valor-alvo, como demonstra a Equação (7) abaixo (Hilpisch, 2020; Gu et al., 2020; Mnih et al., 2015; Sutton e Barto, 2018; Watkins, 1989; Watkins e Dayan, 1992; Wu e Mao, 2021).

$$L(\Theta) = E\left[\left(r + \gamma\, Q(s', a', \Theta') - Q(s, a, \Theta)\right)^2\right] \tag{7}$$

Em que θ são os parâmetros da rede neural.

Em síntese, a IA-DQN segue uma política ϵ-*greedy* durante o treinamento, onde o agente escolhe uma ação aleatória com probabilidade ϵ (exploração) e escolhe a melhor ação com base na função Q com probabilidade $1 - \epsilon$ (explotação). O parâmetro ϵ decai ao longo do tempo para favorecer a exploração inicial e a explotação à medida que o aprendizado avança (Gu et al., 2020; Mnih et al., 2015).

3.3. Coleta e preparação dos dados

A série temporal utilizada foi a taxa de câmbio diária do Dólar/Real (USD/BRL) captada por meio da API (*Application Programming Interface*) do site Yahoo Finance. O período observado foi de janeiro de 2010 a dezembro de 2024. Foi adotada uma *window size* (média móvel) de 10 dias consecutivos, tendo o objetivo de estimar o valor da taxa de câmbio para o dia $t + 1$. Foi também realizada uma normalização dos dados com a finalidade de acelerar a convergência do modelo (Scheuch et al., 2023).

3.4. Normalização de dados, criação das sequências e arquitetura do modelo

Dada uma série temporal financeira $x_1, \ldots, x_n$, tem-se que os retornos podem ser obtidos de forma aritmética ou logarítmica. Considerando que dada uma variação positiva (negativa) em x_t implica retornos positivos (negativos) (Morettin, 2017, p. 8).

Tipo aritmético:

$$y_{t(norm)} = \frac{x_t - x_{t-1}}{x_{t-1}} \tag{8}$$

Tipo logarítmico (log-retornos):

$$y_{t(norm)} = \log \frac{x_t}{x_{t-1}} \tag{9}$$

A criação das sequências se deu a partir de 10 valores de dias consecutivos (`window_size = 10`), no qual buscou-se prever o valor do dia subsequente $(t + 1)$ (Ang, 2021; Barrau e Douady, 2022; Hainaut, 2022; Hua, 2024; Kapoor et al., 2022; Liu, 2023; Scheuch et al., 2023). Ou seja,

$$x_i = [\, x_i,\ x_{i+1},\ \ldots,\ x_{i+9} \,] \tag{10}$$

$$y_i = x_{i+10} \tag{11}$$

A implantação do modelo utilizado na pesquisa se deu por meio do software de programação Python e basicamente por meio das bibliotecas Keras e TensorFlow. O modelo RL-DQN foi composto pelas seguintes camadas (Ng e Shah, 2020; Ceja, 2022; Gridin, 2022; Lange, 2024; Wu e Mao, 2021):

- **Camada convolucional:** responsável por capturar padrões locais na série temporal (composta por 32 filtros, *Kernel* = 3, e ativação ReLU). A saída para cada filtro k é dada por:

$$h_j^k = \text{ReLU}\left(\sum_{i=0}^{f-1} w_i^{(k)} \cdot x_{j+i} + b^k\right) \tag{12}$$

  em que f é o tamanho do filtro, w são os pesos e b é o viés.
- **Camada de *Pooling* Máximo** (MaxPooling 1D; `pool_size = 2`): responsável por reduzir a dimensionalidade e destacar as principais ativações (Powell, 2022).

$$h_j^{pool} = \max\left(h_{pj},\ h_{pj+1},\ \ldots,\ h_{pj+p-1}\right) \tag{13}$$

  em que p é o tamanho da janela de *pooling*.
- **Camada *Flatten*:** responsável por transformar o tensor resultante em um vetor unidimensional (Gridin, 2022).
- **Camada Densa:** responsável por realizar a transformação linear seguida de ativação não linear (Ceja, 2022; Gridin, 2022; Lange, 2024; Ng e Shah, 2020).

$$z = \text{ReLU}(Wx - b) \tag{14}$$

- **Camada de Saída:** essa camada tem um neurônio linear que tem a função de prever o valor do próximo dia da série cambial (Ceja, 2022; Gridin, 2022; Lange, 2024; Ng e Shah, 2020).

$$y = w_{out}\, x + b_{out} \tag{15}$$

Já a função de perda utilizada foi o MSE — Erro Quadrático Médio (Equação (16), a seguir) (Gujarati e Porter, 2011; Neto, 2013; Wooldridge, 2023).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{16}$$

Em que y_i é o valor real e $\hat{y}_i$ é o valor previsto para a amostra i.

O otimizador utilizado foi o Adam (*Adaptive Moment Estimation*), que é um otimizador estocástico que se baseia em médias móveis dos gradientes e dos quadrados dos gradientes. Os parâmetros principais são α (taxa de aprendizado), β_1 e β_2 (coeficientes de decaimento), e ϵ (termo para a estabilidade numérica) (Dangeti, 2017; Dong et al., 2020; Keydana, 2023; Liu, 2023; Powell, 2022; Sanghi, 2024; Zaznov et al., 2024). De acordo com Kingma e Ba (2014, p. 3):

> Seja g_t o gradiente do objetivo estocástico f, estima-se o segundo momento usando média móvel exponencial do gradiente ao quadrado, com taxa de decaimento β_2. E, sejam $g_1, \ldots, g_T$ os gradientes em tempos subsequentes, cada um extraído de uma distribuição de gradiente subjacente $g_t \sim p(g_t)$; inicia-se a média móvel exponencial $v_0 = 0$. A atualização no tempo t da média móvel exponencial $v_t = \beta_2 \cdot v_{t-1} + (1-\beta_2)\cdot g_t^2$ (onde g_t^2 indica o quadrado elemento a elemento $g_t \odot g_t$) pode ser escrita como uma função dos gradientes em todos os passos de tempo anteriores.

$$v_t = (1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}\cdot g_i^2 \tag{17}$$

Já $E[v_t]$ é o valor esperado da média móvel exponencial no tempo t, que se relaciona com o segundo momento real $E[g_t^2]$ (Kingma e Ba, 2014, p. 3). Rearranjando a equação acima, tem-se:

$$\begin{aligned} E[v_t] &= E\left[(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}\, g_i^2\right] \\ &= E[g_t^2]\cdot(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i} + \zeta \\ &= E[g_t^2]\cdot(1-\beta_2^t) + \zeta \end{aligned} \tag{18}$$

Em que $\zeta = 0$ se o segundo momento verdadeiro $E[g_t^2]$ for estacionário; caso contrário, pode ser mantido pequeno. O termo $(1-\beta_2^t)$ é causado pela inicialização da média móvel com zeros (Kingma e Ba, 2014, p. 3-4).

Ainda de acordo com Dereich et al. (2025), Zinkevich (2003) e Wang et al. (2024), a análise de convergência do otimizador Adam utiliza uma sequência arbitrária e desconhecida de funções de custo convexas $f_1(\theta), f_2(\theta), \ldots, f_T(\theta)$, e especificamente nas palavras de Kingma e Ba (2014, p. 4), o conceito de "arrependimento", dado na equação a seguir, é conhecido:

> "a cada instante t, o objetivo é prever o parâmetro t e avaliá-lo em uma função de custo previamente desconhecida f_t". Dada a sequência desconhecida da modelagem, avalia-se o algoritmo usando o arrependimento (*regret*), que é a soma de todas as diferenças anteriores entre a previsão (*online*) $f_t(\theta_t)$ e o melhor parâmetro de ponto fixo $f_t(\theta^*)$ de um conjunto viável χ para todas as etapas anteriores".

$$R(T) = \sum_{t=1}^{T}\left[f_t(\theta_t) - f_t(\theta^*)\right] \tag{19}$$

Em que, $\theta^* = \arg\min_{\theta\in\chi}\sum_{t=1}^{T} f_t(\theta)$.

A última parte da etapa metodológica trata de demonstrar a avaliação do modelo, que envolveu as métricas de mensuração a seguir. Para isso, a diferença entre os valores reais da taxa de câmbio e seus respectivos valores previstos é calculada e analisada por meio das métricas estatísticas de erro médio absoluto e desvio padrão do erro.

Erro Médio Absoluto (MAE) (Ramasubramanian e Moolayil, 2019; Wooldridge, 2023):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{20}$$

Coeficiente de determinação (R^2) (Ramasubramanian e Moolayil, 2019; Wooldridge, 2023):

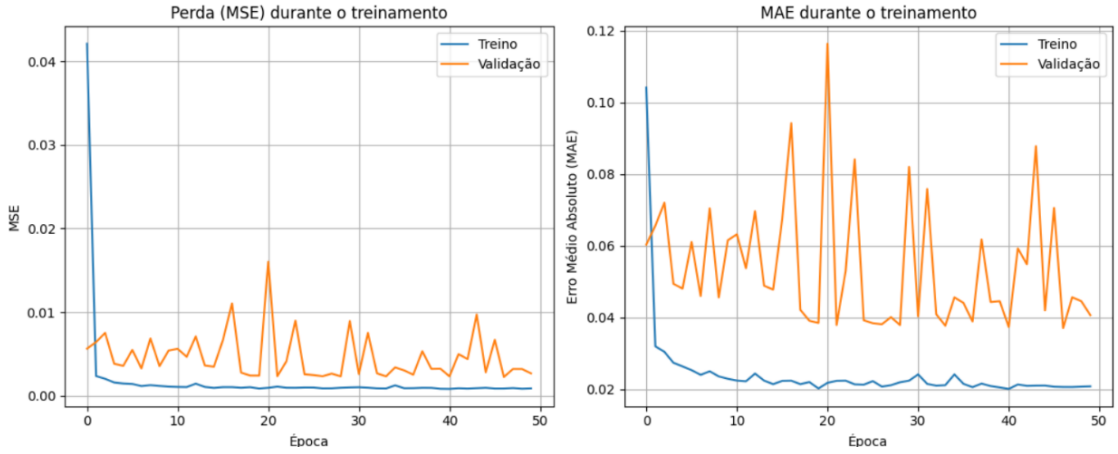
$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \tag{21}$$

4. Resultados

Este artigo procurou aplicar técnicas do algoritmo de IA Deep Q-Networks (DQN) na precificação da taxa de câmbio (Dólar/Real) por meio do instrumento RL, utilizando dados históricos e arquitetura de rede neural para séries temporais financeiras. Os dados históricos da taxa de câmbio foram captados por meio de API do Yahoo Finance e tratados por meio de programação Python, e preparados por sequências de entrada de tamanho fixo (10 dias) para prever o valor da variável no dia seguinte ($t+1$). Foi implementado um modelo baseado em arquitetura DQN, composta por camadas convolucionais, *pooling* e densas, e treinado com função de perda de erro quadrático médio (MSE) e otimização Adam. O conjunto de treinamento foi composto por 80% dos dados, reservando-se 20% para testes. O treinamento foi realizado por uma sequência de 50 épocas.

Os valores do MSE (0,0021) e do MAE (0,027) baixos indicam que o modelo utilizado nesta pesquisa conseguiu prever com precisão relevante a volatilidade da taxa cambial Dólar/Real na maior parte do tempo observado. Em que o valor ideal para o MSE e para o MAE seria zero; e caso isso acontecesse a trajetória da taxa de câmbio seria perfeitamente previsível. No entanto, devido à falta de linearidade e alta complexidade dos dados não é possível (no momento) que o processo de redes neurais identifique totalmente a trajetória da variável no tempo. Já o coeficiente R^2, surpreendentemente, apresentou um valor alto e próximo de 1 (0,89), confirmando a capacidade preditiva do modelo. Isso indica que a modelagem explica a maior parte da variabilidade dos dados por meio da correlação mensurável.

Gráfico 1. MSE e MAE durante o período de treinamento

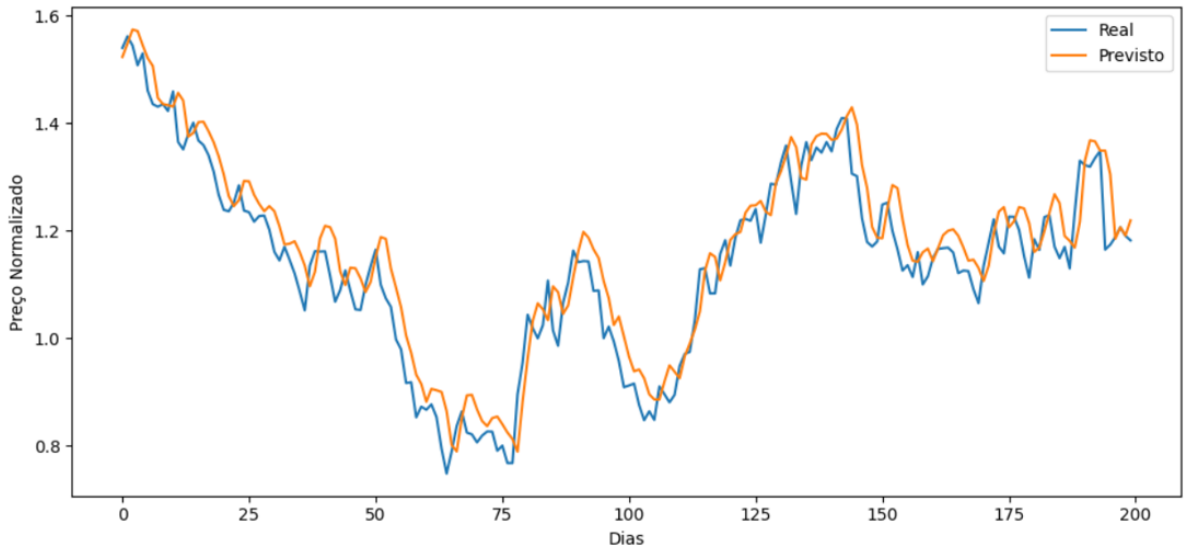


Fonte: elaboração própria por meio de dados captados do API Yahoo Finance/Python.

O modelo RL-DQN utilizado nesta pesquisa capturou relativamente bem a dinâmica da série temporal da taxa de câmbio (USD/BRL) no período observado nesta pesquisa, tendo ainda destaque para as camadas convolucionais que detectaram padrões de curto prazo. A estabilidade e bom desempenho da fase de treinamento (gráfico de perda e MAE) indica que a arquitetura programada dos hiperparâmetros foi adequada ao modelo. A otimização do modelo realizada pelo otimizador Adam também foi de fundamental importância para a estabilidade e convergência dos valores estimados, isso devido ao seu processo de ajuste adaptativo da taxa de aprendizado.

O Gráfico 2 faz uma comparação entre os valores da taxa de câmbio Real/Dólar em termos de valores reais e seus respectivos valores previstos para os primeiros duzentos períodos (dias) de treinamento do modelo. Também é possível, por meio da análise visual, perceber que mesmo em situações de maior instabilidade e volatilidade, o modelo manteve um desempenho razoável, embora também seja possível notar que os erros aumentam nestes momentos. Ainda a partir da análise visual do Gráfico 2, observa-se que a perda de validação acompanha bem o treinamento, em um estado de convergência relativamente estável, sem sinais de *overfitting* relevante.

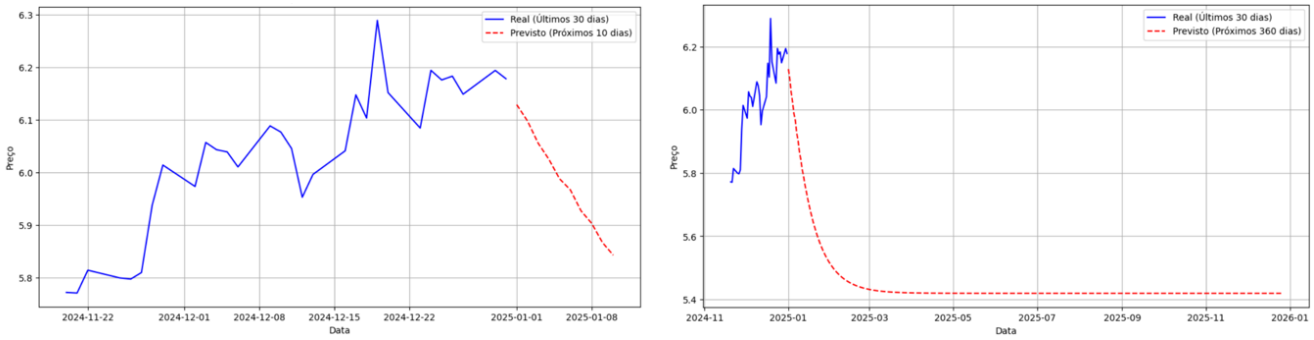
Gráfico 2. Taxa de câmbio USD/BRL — Real vs. Previsto (200 primeiros dias do teste)



Fonte: elaboração própria por meio de dados captados do API Yahoo Finance/Python.

O Gráfico 3 plota a previsão de taxa de câmbio para um período curto de 10 dias (à esquerda) e um período médio de previsão de 360 dias (à direita). Por meio desse gráfico é fácil perceber a tendência de queda na taxa de câmbio esperada no curto prazo e sua estabilidade estimada no modelo no médio prazo. Vale ainda reforçar que nenhuma outra variável foi utilizada no modelo para realizar este exercício de previsibilidade da trajetória da variável.

Gráfico 3. Taxa de câmbio USD/BRL — Previsto para os próximos 10 e 360 períodos (dias) após o último dia da série captada



Fonte: elaboração própria por meio de dados captados do API Yahoo Finance/Python.

O Gráfico 3 acima pode parecer que a estabilidade encontrada pode não trazer muitos avanços para a questão de precificação. Em situações de negociações do tipo *daytrade*, no entanto, fica claro por meio desse gráfico que o valor mínimo atingido pela estabilidade pode ser de fundamental importância para alguns tipos de negociações no mercado financeiro. Por exemplo, Hilpisch (2020, p. 380) cita que "*Deep Hedge* é um termo utilizado para algoritmos de IA, que podem ser treinados para executar transações de *hedge* de maneira otimizada". O mesmo autor também cita Buehler et al. (2019), que aplica modelagem RL para implementar *hedge* em mercados financeiros.

Por meio do Quadro 1, podemos comparar os valores de previsão do modelo RL-DQN dessa pesquisa para com os dias após a série de dados reais (*ex-post*) da taxa de câmbio do dólar no Brasil. Dessa forma, podemos verificar de maneira prática o bom desempenho do modelo no médio prazo.

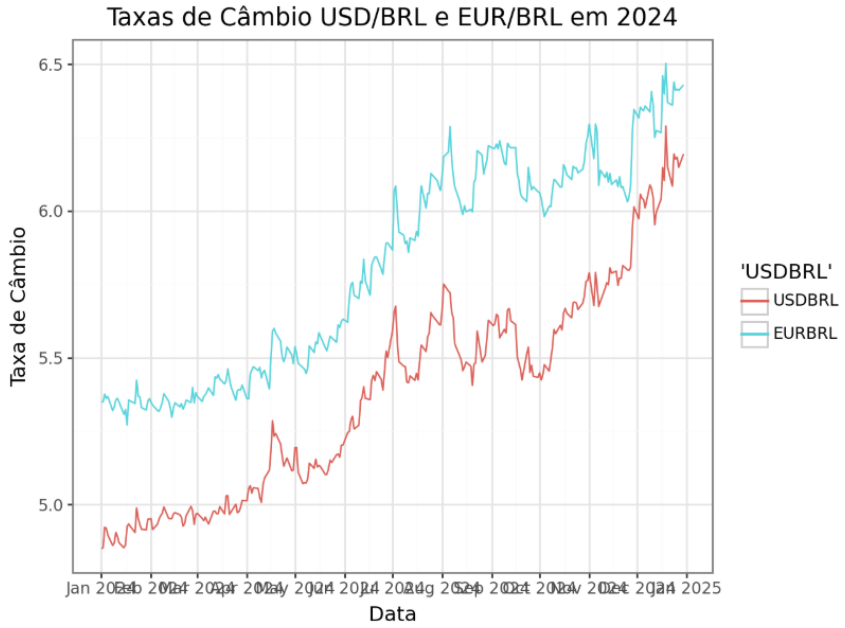
Quadro 1. *Benchmark* entre valores previstos do modelo RL-DQN e os valores de cotação de dólar no Brasil para o período observado

Data	Valores Previstos no modelo RL-DQN	Taxa de câmbio - Livre - Dólar americano (venda) - diário - u.m.c./US$
1/2/2025	6,1286	6,2086
1/3/2025	6,0978	6,1563
1/6/2025	6,0554	6,1119
1/7/2025	6,0249	6,0741
1/8/2025	5,9668	6,1321
1/9/2025	5,9268	6,0896
1/10/2025	5,903	6,0965
1/13/2025	5,8666	6,1079
1/14/2025	5,8427	6,0671
1/29/2025		5,8596
2/28/2025		5,8488
4/23/2025		5,688

Fonte: elaboração própria por meio de dados captados do API Yahoo Finance/Python, e também por meio de dados captados do Sistema de Gerenciamento de Séries Temporais do Banco Central do Brasil.

Retrocedendo ao conceito de volatilidade de série temporal, temos que este é bastante importante para modelagem de forma geral, e em especial para séries de dados financeiros. Por meio do Gráfico 4 observamos a trajetória volátil do preço do Dólar comparada com o também volátil preço do Euro no ano de 2024 no Brasil. Em que, apesar, é claro, do movimento dessas duas moedas não serem idênticos, podemos perceber visualmente que os movimentos dos preços no tempo são bem próximos. E ao calcularmos a volatilidade anualizada das duas moedas, temos que os valores também são quase idênticos (a volatilidade anualizada do Dólar e do Euro — frente ao Real — respectivamente foram de 0,1258 e 0,1274; e em termos comparativos a volatilidade anualizada entre o Dólar e o Euro foi de 0,0586).

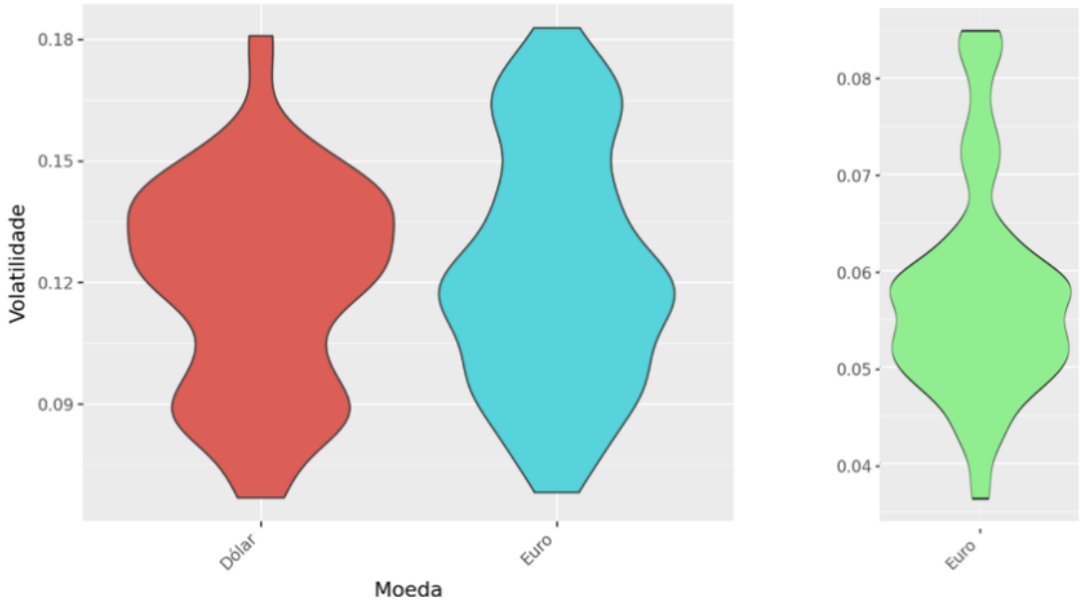
Gráfico 4. Taxa de câmbio USD/BRL e EUR/BRL (Ano de 2024)



Fonte: elaboração própria por meio de dados captados do API Yahoo Finance/Python.

E com o objetivo de aprofundar a análise do entendimento da volatilidade do Dólar e do Euro no Brasil, foi plotado um gráfico do tipo violino (Gráfico 5). Por meio dessa figura, vemos que o Dólar (cor vermelha) e o Euro (na cor azul), apesar de terem praticamente o mesmo valor numérico para a volatilidade das moedas, é perceptível visualizarmos diferenças ao olharmos a distribuição dos dados durante o período de um ano. Uma possível explicação para isso é potenciais arbitragens que os agentes econômicos realizam no mercado de câmbio no Brasil. O Gráfico 5 ainda, por meio da figura da cor verde, aponta a distribuição da volatilidade entre a moeda americana e a moeda europeia, para o mesmo ano observado.

Gráfico 5. Volatilidade Dólar (vermelho) e Euro (azul) a partir da taxa de câmbio com o Real; volatilidade da taxa de câmbio do Dólar correlacionado ao Euro (verde) (Ano de 2024)



Fonte: elaboração própria por meio de dados captados do API Yahoo Finance/Python.

O Quadro 2 traz uma síntese dos principais resultados da pesquisa, no qual podemos verificar o benefício total do conjunto de treinamento (4418,79) e também o benefício total do conjunto de testes (140,47). Além de apontar os valores encontrados de parâmetros das camadas e do modelo Adam de otimização utilizado.

Quadro 2. Síntese dos valores e resultados numéricos dos hiperparâmetros da pesquisa

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 8, 32)	128
max_poolingld (MAxPooling1D)	(None, 4, 32)	0
flatten (Flatten)	(None, 128)	0
dense (Dense)	(None, 64)	8,256
dense_1 (Dense)	(None, 1)	65
Total params: 29,349 (99.02 KB)		
Trainable params: 8,449 (33.00 KB)		
Won-trainable params: 0 (0.00 B)		
Optimizer params: 16,900 (66.02 KB)		
Função de Perda: mse		
Otimizador: <keras.src.opyimizers.adam.Adam object at 0x7fc9f48d8910>		
Benefício total do conjunto de treinamento: 4418.79		
Benefício total do conjunto de teste: 140.47		

Fonte: elaboração própria por meio de dados captados do API Yahoo Finance/Python.

Por fim, acreditamos que o objetivo do artigo foi atingido, pois, a partir da metodologia proposta (RL-DQN), foram gerados os valores preditivos da taxa de câmbio (Dólar/Real), com um nível de desempenho (acurácia) satisfatório. Ainda foi possível verificar vantagens do uso de algoritmos de IA em nível de aprendizado profundo para extrair padrões complexos dos dados avaliados. No entanto, também podemos verificar algumas limitações do modelo, como, por exemplo, problemas de previsibilidade em períodos de alta volatilidade. A escolha de janela de 10 dias foi suficiente para capturar tendências de médio prazo, mas estudos futuros também podem utilizar diferentes parametrizações com o objetivo de melhora no modelo.

5. Considerações Finais

Dos celulares aos automóveis autônomos, a sociedade de consumo começou a considerar a utilização de modelos de IA, por meio de metodologias RL. O avanço de equipamentos de *hardware* moldou o uso de redes neurais profundas que simulam as redes neurais do cérebro humano (Ciaburro, 2019, p. 58). E de acordo com Hilpisch (2020, p. 29), a aplicação de redes neurais já é um fator central em finanças, como, por exemplo, prever a direção do movimento de um índice de

ações ou a taxa de câmbio de uma moeda. Dessa forma, a corrida para se alcançar a chamada singularidade financeira (capacidade de prever movimentos nos mercados financeiros) é cada vez mais discutida hoje em dia, e isso se torna cada vez mais possível por meio de agentes de IA (Hilpisch, 2020, p. 396).

A precificação de ativos financeiros, em especial, é um dos principais desafios em finanças quantitativas, dada a complexidade e a não linearidade dos mercados. Com o avanço da Inteligência Artificial (IA), técnicas como Aprendizado por Reforço Profundo têm se mostrado promissoras para tarefas de previsão e tomada de decisão em ambientes dinâmicos, como o mercado de câmbio. Dentre esses algoritmos em AR, destaca-se o RL-DQN por unir o *Q-Learning* clássico com redes neurais profundas, permitindo o tratamento de grandes volumes de dados e extração de padrões.

Dessa forma, este artigo explorou a construção, treinamento e avaliação de um modelo RL-DQN para previsão da taxa de câmbio USD/BRL, utilizando dados históricos e arquitetura de rede neural adaptada para séries temporais para o período de janeiro de 2010 a dezembro de 2024. O modelo foi treinado utilizando a função de perda por erro quadrado médio (MSE) e o otimizador Adam. O conjunto de treinamento foi composto por 80% dos dados, reservando-se 20% para testes. Os valores encontrados de MSE (0,0021) e do MAE (0,027) foram baixos, o que indica que o modelo utilizado conseguiu prever com relativa precisão a volatilidade da taxa cambial Dólar/Real na maior parte do tempo observado.

Os resultados obtidos nesta pesquisa ainda permitiram analisar com mais profundidade a volatilidade entre o Dólar americano e o Euro, analisadas a partir da taxa de câmbio da moeda brasileira (Real). Em que, apesar dos resultados numéricos de volatilidade anualizada serem quase idênticos (0,1258 e 0,1274, respectivamente para o dólar e o euro), observou-se por meio da análise visual que existe uma diferença significativa entre as distribuições dos dados ao longo da série observada.

O coeficiente R^2, surpreendentemente, apresentou um valor alto e próximo de 1 (0,89), o que confirma a capacidade preditiva do modelo. Porém, o modelo ainda pode ser melhorado em trabalhos futuros, pois a capacidade preditiva do modelo alcançou bom desempenho apenas no médio prazo (tempo maior do que 30 dias). Esse fato impede trabalhar com esse modelo em estratégias de curtíssimo prazo (*daytrade*). No entanto, a utilização do modelo RL-DQN em situações estratégicas de *hedge* cambial parece ser bem promissora, ao ter uma grande assertividade de valores no médio prazo.

Referências

Ahlawat, S. (2023). *Reinforcement Learning for Finance: Solve Problems in Finance with CNN and RNN Using the TensorFlow Library*. Apress.

Ang, C. S. (2021). *Analyzing Financial Data and Implementing Financial Models Using R*. Springer, 2 edition.

Barrat, J. (2013). *Our Final Invention: Artificial Intelligence and the End of the Human Era*. St. Martin's Press, New York.

Barrau, T. e Douady, R. (2022). *Artificial Intelligence for Financial Markets: The Polymodel Approach*. Springer.

Buehler, H., Gonon, L., Teichmann, J., Wood, B., Mohan, B., e Kochems, J. (2019). Deep hedging: hedging derivatives under generic market frictions using reinforcement learning. *SSRN Electronic Journal*.

Ceja, E. G. (2022). *Behavior Analysis with Machine Learning Using R*. CRC Press.

Cheng, Y.-H. e Wang, H.-C. (2023). Decision support from financial disclosures with deep reinforcement learning considering different countries and exchange rates. *Engineering Proceedings*, 55:63.

Ciaburro, G. (2019). *Hands-On Reinforcement Learning with R*. Packt Publishing.

Dangeti, P. (2017). *Statistics for Machine Learning*. Packt Publishing.

Deng, Y., Bao, F., Kong, Y., Ren, Z., e Dai, Q. (2017). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3):653–664.

Dereich, S., Jentzen, A., e Riekert, A. (2025). Averaged adam accelerates stochastic optimization in the training of deep neural network approximations for partial differential equation and optimal control problems. *arXiv preprint arXiv:2501.06081*.

Dong, H., Ding, Z., e Zhang, S. (2020). *Deep Reinforcement Learning: Fundamentals, Research and Applications*. Springer.

Gridin, I. (2022). *Practical Deep Reinforcement Learning with Python*. BPB Online.

Gu, S., Kelly, B., e Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273.

Gujarati, D. N. e Porter, D. C. (2011). *Econometria básica*. Grupo A, Porto Alegre.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media, 2 edition.

Hainaut, D. (2022). *Continuous Time Processes for Finance: Switching, Self-exciting, Fractional and Other Recent Dynamics*. Bocconi & Springer.

Hilpisch, Y. (2015). *Derivatives Analytics with Python: Data Analysis, Models, Simulation, Calibration and Hedging*. John Wiley & Sons, Chichester.

Hilpisch, Y. (2020). *Artificial Intelligence in Finance: A Python-based Guide*. O'Reilly Media.

Hilpisch, Y. (2024). *Reinforcement Learning for Finance: A Python-Based Introduction*. O'Reilly Media.

Hua, P. (2024). *Neural Networks with TensorFlow and Keras: Training, Generative Models, and Reinforcement Learning*. Springer.

Kapoor, A., Gulli, A., e Pal, S. (2022). *Deep Learning with TensorFlow and Keras*. Packt Publishing, 3 edition.

Keydana, S. (2023). *Deep Learning and Scientific Computing with R torch*. CRC Press.

Kingma, D. P. e Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kissinger, H. A., Schmidt, E., e Huttenlocher, D. (2021). *The Age of AI: And Our Human Future*. Little, Brown and Company, New York.

Kurzweil, R. (2005). *The Singularity Is Near: When Humans Transcend Biology*. Penguin Group, New York.

Lange, C. (2024). *Practical Machine Learning with R: Tutorials and Case Studies*. CRC Press.

Liu, G. R. (2023). *Machine Learning with Python: Theory and Applications*. World Scientific Publishing.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill, New York.

Mnih, V. et al. (2013). Playing atari with deep reinforcement learning. *arXiv preprint*.

Mnih, V. et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Morettin, P. A. (2017). *Econometria Financeira: Um Curso em Séries Temporais Financeiras*. Blucher, São Paulo, 3 edition.

Morettin, P. A. (2020). *Análise de Séries Temporais — volume 2*. Blucher, São Paulo.

Neto, A. S. (2013). *Estatística e Introdução à Econometria*. SRV Editora, São Paulo, 2 edition.

Ng, J. e Shah, S. (2020). *Hands-On Artificial Intelligence for Banking*. Packt Publishing.

Plakandaras, V., Papadimitriou, T., e Gogas, P. (2015). Forecasting daily and monthly exchange rates with machine learning techniques. *Journal of Forecasting*.

Powell, W. B. (2022). *Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions*. Wiley.

Ramasubramanian, K. e Moolayil, J. (2019). *Applied Supervised Learning with R*. Packt Publishing.

Rao, A. e Jelvis, T. (2023). *Foundations of Reinforcement Learning with Applications in Finance*. CRC Press.

Russell, S. J. e Norvig, P. (2022). *Inteligência Artificial: Uma Abordagem Moderna*. GEN LTC, Rio de Janeiro, 4 edition.

Sanghi, N. (2024). *Deep Reinforcement Learning with Python: RLHF for Chatbots and Large Language Models*. Apress, 2 edition.

Scheuch, C., Voigt, S., e Weiss, P. (2023). *Tidy Finance with R*. CRC Press.

Shah, R. et al. (2025). An approach to technical agi safety and security. Acesso em: 20 abr. 2025.

Silver, D. (2016). Deep reinforcement learning. Acesso em: 14 jul. 2024.

Sutton, R. S. e Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge.

Vinge, V. (1993). Vernor vinge on the singularity. Acesso em: 20 jan. 2025.

Wang, S. et al. (2024). Cadam: confidence-based optimization for online learning. *arXiv preprint arXiv:2411.19647*.

Watkins, C. (1989). *Learning from Delayed Rewards*. PhD thesis, University of Cambridge.

Watkins, C. e Dayan, P. (1992). Q-learning. *Machine Learning*, 8:279–292.

Wooldridge, J. M. (2023). *Introdução à Econometria: Uma Abordagem Moderna*. Cengage Learning, São Paulo.

Wu, S. e Mao, P. (2021). Analysis and forecast of exchange rate based on reinforcement learning. *Advances in Educational Technology and Psychology*, 5:81–92.

Yahoo Finance (2024). Yahoo finance api. Acesso em: 10 abr. 2024.

Yang, H., Liu, X.-Y., e Zhong, S. (2020). Deep reinforcement learning for automated stock trading: an ensemble strategy. In *ICAIF 2020*, pages 1–8.

Zaznov, I. et al. (2024). Adamz: an enhanced optimisation method for neural network training. *arXiv preprint arXiv:2411.15375*.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. *Proceedings of the 20th International Conference on Machine Learning*.