AGENTE DE APRENDIZADO POR REFORÇO TABULAR PARA NEGOCIAÇÃO DE AÇÕES

RENATO ARANTES DE OLIVEIRA

AGENTE DE APRENDIZADO POR REFORÇO TABULAR PARA NEGOCIAÇÃO DE AÇÕES

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: ADRIANO CÉSAR MACHADO PEREIRA

Belo Horizonte

Fevereiro de 2020

Oliveira, Renato Arantes de.

O48a Agente de aprendizado por reforço tabular para negociação de ações [manuscrito] / Renato Arantes de Oliveira. — 2020.
xxii, 73 f.; il.; 29cm.

Orientador: Adriano César Machado Pereira.
Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f. 71-73

1. Computação – Teses. 2. Inteligência Artificial - Teses 3. Aprendizado por reforço. – Teses. I. Pereira, Adriano César Machado. II. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. III. Título.

CDU 519.6*82(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz CRB 6ª Região nº 819.

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Agente de Aprendizado por Reforço Tabular para Negociação de Ações

RENATO ARANTES DE OLIVEIRA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. ADRIANO CÉSAR MACHADO PEREIRA - Orientador
Departamento de Ciência da Computação - UFMG

PROFESSOR ANISIO MENDES LACERDA
Departamento de Ciência da Computação - UFMG

PROF. DANIEL HASAN DALIP
Departamento de Computação - CEFET/MG

PROF. PAULO ANDRE LIMA DE CASTRO
Divisão de Ciência da Computação - ITA

Belo Horizonte, 28 de Fevereiro de 2020.

Dedico esse trabalho a todos aqueles que um dia olharam para o céu profundo na noite e ousaram sonhar ir aonde nenhum homem jamais esteve.

Agradecimentos

Agradeço esse trabalho de mestrado em primeiro lugar a Deus que vem nos dando forças nos momentos de dificuldade e não permitindo o acomodamento nos momentos de tranquilidade. Agradeço também aos meus pais por todo apoio durante esse período e meus irmãos que sempre me motivaram. Não poderia deixar de agradecer a todos os amigos (professores, alunos e colaboradores) do grupo de Finanças Computacionais do DCC (FICO) pelas ideias, sugestões, críticas e apoio que foram muito importantes na elaboração dessa pesquisa. Em especial, agradeço ao Professor Adriano pela paciência e pelo incansável apoio ao longo desse mestrado que culminou nesta dissertação. A todos o meu mais sincero e singelo, Muito Obrigado!

"And on its journey back, it amassed so much knowledge,
it achieved consciousness itself."
(Captain Kirk, in the movie *Star Trek - The Motion Picture*, 1979)

Resumo

Modelos de aprendizado supervisionado aplicados no contexto de negociação de ativos financeiros têm sido propostos e estudados há mais de duas décadas. Embora tenham alcançado bons resultados em termos de rendimento financeiro e risco, essa abordagem padece de limitações importantes, tais como a necessidade de retreinamentos constantes sobretudo nas grandes oscilações do mercado, além da dificuldade em converter um modelo com boa taxa de acertos nas previsões em um sistema de negociação que gere altos rendimentos. Essas limitações podem ser contornadas com a utilização de técnicas de Aprendizado por Reforço. Nessa abordagem, um agente pode aprender a negociar ativos financeiros para maximizar o ganho total ou minimizar o risco através de sua própria interação com o mercado. Além disso, também é capaz de manter-se atualizado a cada modificação do ambiente dispensando a necessidade de retreinamento uma vez que o agente está sempre em aprendizado. Para obter evidências dessas propriedades, desenvolveu-se um agente de aprendizado por reforço utilizando uma modelagem tabular com o algoritmo SARSA e aplicou-se esse agente em um conjunto de ações com variados padrões de tendência com o objetivo de observar como esse agente muda sua estratégia de negociação em cada situação de tendência. Além disso, desenvolveu-se um agente de negociação baseado em aprendizado supervisionado utilizando uma rede neural LSTM para comparar o seu desempenho com o do agente de aprendizado por reforço proposto. Aplicou-se ambos os agentes em um conjunto de 10 ações da Bovespa no ano de 2018, comparando métricas de rendimento financeiro, risco e taxas de acertos. Os resultados experimentais apresentaram evidências não só das limitações do agente de aprendizado supervisionado proposto, como também das aludidas propriedades do agente de aprendizado por reforço em se adaptar às mudanças no mercado de modo a produzir ganhos financeiros com menores perdas financeiras acumuladas.

Palavras-chave: Aprendizado por Reforço, Finanças, Negociação, Mercado de Capitais, Volatilidade, Tendências, Estratégias Dinâmicas.

Abstract

Supervised learning models applied in the context of financial asset trading have been proposed and studied for more than two decades. Although many studies have succeeded in demonstrating good results in terms of financial yields and risk, this approach suffers from important limitations such as the need for constant retraining, especially in the presence of large market fluctuations and the difficulty in converting a good model in terms of prediction accuracy into a system that generates high financial yields. These limitations can be overcome with the use of Reinforcement Learning techniques. In this approach, an agent can learn to trade financial assets so as to maximize total gain or minimize risk through its own interaction with the market. In addition, it is also able to keep itself updated with each modification of the environment, eliminating the need for retraining since the agent is always learning. To obtain evidence of these properties, a reinforcement learning agent was proposed and developed using a tabular SARSA algorithm modeling. Afterwards the agent was applied to a set of stocks with varying trend patterns in order to observe how the agent behaves in terms of its strategy in each trend situation. In addition, a financial trading agent based on supervised learning was also developed using an LSTM neural network to compare its performance with that of the proposed reinforcement learning agent. Both agents were applied to a set of 10 stocks from the brazilian stock market Bovespa in the year 2018 and its performance were assessed in terms of financial yield, risk and accuracy. The experimental results provided evidence not only of the limitations of the proposed supervised learning agent, but also of the aforementioned properties of the reinforcement learning agent in adapting to changes in the market in order to produce financial gains with less accumulated financial losses.

Keywords: Reinforcement Learning, Finance, Trading, Stock Market, Volatility, Trends, Dynamic Strategies.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

O sucesso de uma estratégia de investimento no mercado de ações negociadas em bolsas de valores é determinado pela sequência de decisões tomadas pelo investidor. Por isso, os investidores estão sempre atentos às cotações de preços, notícias, cenário político e econômico com o objetivo de detectar padrões que os permitam tomar as melhores decisões em cada situação.

Sabe-se que atualmente grande parte das negociações que ocorrem nas principais bolsas de valores do mundo são executadas por sistemas automatizados de negociação (*Financial Trading Systems* em inglês)[1] também chamados de robôs ou agentes de negociação. Esses sistemas utilizam técnicas de Inteligência Artificial para detectar padrões ocultos em tempo real a partir de dados de preços, volume, notícias e outras informações. Uma vez detectado um determinado padrão pelo agente, o sistema executa automaticamente a operação mais adequada para os objetivos do investimento naquele momento.

Esses sistemas geralmente operam alta frequência e disputam por negócios que ofereçam ganhos muito pequenos. Porém, se essas oportunidades de negócios forem numerosas o suficiente eles podem acumular muitos ganhos [Bodie et al., 2008]. Uma vantagem desses sistemas é que eles podem detectar padrões e executar ordens em frações de segundo possibilitando ao investidor um melhor aproveitamento das oportunidades de negócios no mercado de bolsa de valores.

Uma das abordagens em Inteligência Artificial utilizadas é o Aprendizado por Reforço (*Reinforcement Learning* em inglês). Nessa abordagem um agente é capaz de aprender a associar ações (decisões) a situações (estados) através da sua própria interação com o ambiente de modo a maximizar uma medida de desempenho em uma determinada tarefa [Sutton & Barto, 2018].

[1] Vide em url: https://www.cnbc.com/2017/06/13/death-of-the-human-investor-just-10-percent-of-trading-is-regular-stock-picking-jpmorgan-estimates.html

Isto é semelhante a maneira como humanos aprendem várias tarefas durante a vida. Por exemplo, uma criança aprende a andar de bicicleta, jogar futebol ou andar de *skate* interagindo com esses objetos. É através da interação com esses objetos que uma criança aprende a relação entre uma ação tomada e a consequência da respectiva ação em cada situação. Desse modo, por tentativa e erro a criança adquire conhecimento e experiência e permanece sempre melhorando seu desempenho em cada tarefa.

Sistemas que empregam aprendizado por reforço têm obtido sucesso em diversas aplicações tais como jogos RTS (*Real-Time Strategy* em inglês), roteamento de veículos autônomos, jogos de *Atari*, jogos de tabuleiro (e.g. Go) e até aceleração de descobrimento de medicamentos [Tavares & Chaimowicz, 2018; Reddy et al., 2018; Mnih et al., 2015; Silver et al., 2016; Serrano et al., 2018].

No contexto de negociação de ações um sistema baseado em aprendizado por reforço é capaz de aprender através da sua própria interação com o mercado a associar a melhor decisão (comprar, vender, não operar) a cada estado (situação) de modo a otimizar uma medida de desempenho (e.g., rendimento financeiro, *Sharpe ratio*, *drawdown*). Além disso, esse tipo de sistema possui uma importante característica adaptativa: o agente de aprendizado por reforço é capaz de modificar o que aprendeu anteriormente de forma dinâmica (*on-line*) à medida que as condições do mercado se modificam.

Essas propriedades sugerem que um sistema de negociação baseado em aprendizado por reforço pode ser uma alternativa competitiva em termos de rendimento financeiro e risco perante outros tipos de modelagens de sistemas de negociação baseados em modelos de aprendizado supervisionado ou indicadores de Análise Técnica, por exemplo.

Portanto, pretende-se nesse trabalho de dissertação de mestrado conceber, propor, desenvolver e testar um agente de aprendizado por reforço para negociação de ações. Pretende-se ainda compará-lo com outro sistema baseado em aprendizado supervisionado. Para tanto, serão consideradas métricas de rendimento financeiro, risco e taxas de acertos.

Ao final, pretende-se obter evidências experimentais das propriedades adaptativas de um sistema de aprendizado por reforço aplicado ao contexto de negociação de ações bem como evidências de sua superioridade em termos de rendimento financeiro e risco comparado a um sistema baseado em aprendizado supervisionado.

1.1 Objetivos

O **objetivo geral** do presente trabalho é propor, desenvolver e analisar em termos de viabilidade técnica e financeira um sistema de negociação de ações utilizando aprendizado por reforço.

1.1.1 Objetivos Específicos

Para alcançar o referido objetivo geral, propõe-se os seguintes objetivos específicos:

1. Coletar dados históricos de ações da Bolsa de Valores de São Paulo (B3 - Bolsa Brasil Balcão).
2. Identificar padrões de tendência nas séries temporais consideradas (tendências de alta, baixa, pouca tendência).
3. Implementar um agente de negociação utilizando um algoritmo de aprendizado por reforço.
4. Treinar e simular o agente em dados históricos das ações.
5. Analisar os resultados produzidos comparando com estratégias comuns e modelagens que empregam aprendizado supervisionado.

1.2 Contribuições

Espera-se ao final desse trabalho obter evidências experimentais que possam sugerir a viabilidade técnica e financeira do agente de aprendizado por reforço proposto. Além isso, outras contribuições importantes são:

- Comparação do sistema de aprendizado por reforço implementado com um agente de negociação baseado em aprendizado supervisionado (e.g., redes neurais LSTM).
- Análise do sistema implementado em diferentes condições de tendências de uma ação.
- Análise do sistema implementado em um contexto de instabilidade do mercado de ações.

1.3 Organização do Trabalho

Esta dissertação está organizada da seguinte forma: além do presente Capítulo, no Capítulo 2 apresenta-se os conceitos fundamentais do mercado de ações para compreensão do restante do trabalho como também os conceitos básicos de aprendizado supervisionado e aprendizado por reforço. Em seguida, no Capítulo 3 os principais estudos empregando aprendizado por reforço em finanças são elencados destacando-se a divisão entre aprendizado por reforço "tradicional" e o aprendizado por reforço profundo bem como a necessidade de se estudar

e comparar modelagens de aprendizado por reforço chamadas tabulares com as modelagens mais recentes. No Capítulo 4 é apresentada a proposta de modelagem do agente de aprendizado por reforço para negociação de ações e suas propriedades. Adiante, no Capítulo 5 são apresentados os objetivos da metodologia, suas fases, dados utilizados, as hipóteses subjacentes aos experimentos propostos bem como a modelagem do agente de aprendizado supervisionado utilizado como *baseline* na comparação com o agente de aprendizado por reforço proposto. No Capítulo 6 são mostrados os resultados de cada experimento e as respectivas análises tendo em vista as hipóteses e resultados esperados levantados no capítulo anterior. No Capítulo 7 é realizada uma síntese do trabalho e apresentados os resultados obtidos à luz do objetivo geral e das contribuições esperadas mencionadas no Capítulo 1. Além disso, são apresentados o escopo do trabalho e suas limitações bem como propostas de modelagens e abordagens para aprofundamento da utilização de técnicas de Aprendizado por Reforço no contexto de negociação de ativos financeiros como trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste Capítulo, são apresentados os conceitos básicos de mercados de ações, métricas de rendimento financeiro e risco que são essenciais para compreensão do restante do trabalho. São apresentados também os conceitos básicos de aprendizado supervisionado e aprendizado por reforço no contexto de mercado financeiro fundamentais para realização desse trabalho.

2.1 Mercado de Ações

O mercado de ações negociadas em bolsas de valores é um dos pilares no desenvolvimento de economias capitalistas modernas. É através do mercado de ações, pela venda direta de participação no seu patrimônio líquido representado pelas ações, que as sociedades anônimas captam os recursos necessários ao seu desenvolvimento negocial e patrimonial, assumindo o compromisso de remunerar os seus acionistas em função do capital nela aplicado e de seus resultados futuros [Fortuna, 2015].

É também através do mercado de ações e outros títulos financeiros negociados em bolsa que investidores tem a oportunidade de auferir rendimentos acima de outras aplicações menos arriscadas (e.g. Certificado de Depósito Interbancário também chamado de CDI , títulos do tesouro e poupança) através de livre negociação especulativa de ativos financeiros. Naturalmente, os preços das ações, como qualquer bem livremente negociado, é determinado pela lei de oferta e procura. Por conseguinte, esses preços são afetados por diversos fatores tais como as expectativas dos investidores em relação a empresa, as condições econômicas e políticas do país, taxas de juros, inflação, câmbio, etc.

A variação relativa dos preços de uma ação entre os instantes $t-1$ e t é chamada *taxa de retorno* e é definida por (Equação 2.1):

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}} \tag{2.1}$$

onde r_t denota a taxa de retorno no tempo t, P_t e P_{t-1} denotam respectivamente os preços da ação nos tempos t e $t-1$.

A variação média dos retornos ao longo tempo é geralmente chamada de *volatilidade* e pode ser medida de diversas formas sendo a mais comum o desvio padrão dos retornos [Wilmott, 2013]. Momentos de incerteza e indefinições no mercado costumam apresentar alta volatilidade o que dificulta a tarefa de prever qual o próximo movimento da série de preços de uma ação para identificar uma tendência. Investidores buscam detectar o início ou final de uma tendência para tomar uma decisão de comprar ou vender um ativo. Em geral, os investidores compram uma ação no início de uma tendência de alta por um preço baixo e vendem a ação quando a tendência termina e começa a cair por um preço mais alto [Kirkpatrick II & Dahlquist, 2010] auferindo o respectivo rendimento.

Porem, detectar quando começa e quando termina uma tendência não é uma tarefa simples. Por isso, costuma-se utilizar indicadores de Análise Técnica buscando caracterizar e identificar tendências ou reversões de tendências. Embora, os indicadores de Análise Técnica sejam bastante utilizados por investidores e analistas do mercado, nem sempre eles conseguem prever precisamente o início ou o fim de uma tendência. Apesar disso, uma vez detectado ou previsto o início de uma tendência de alta ou de baixa o investidor pode iniciar uma posição buscando obter ganhos em ambas as situações.

Diz-se que o investidor está posicionado em uma *posição comprada* (*long* em inglês) quando o investidor compra uma ação esperando vendê-la futuramente por preço maior que quando comprou. O retorno r_{long} de uma posição *long* pode ser calculado como (Equação 2.2):

$$r_{long} = \frac{P_{venda} + D - P_{compra}}{P_{compra}} \tag{2.2}$$

sendo P_{venda} o preço posterior pelo qual o investidor vendeu o ativo, D os dividendos distribuídos durante a posição comprada e P_{compra} o preço anterior pelo qual o investidor comprou o ativo.

É também possível aproveitar as tendências de baixa para obter ganhos. Considerando que uma ação está em tendência de baixa e que o investidor não possui essa ação no momento, ele pode tomar emprestado essa ação de outro investidor por meio de uma corretora e vendê-la em seguida esperando recomprá-la futuramente por um preço ainda menor para então devolvê-la ao seu titular original e embolsar a diferença dos preços. Essa operação é chamada *venda a descoberto* ou *shorting* em inglês. O investidor inicia uma posição vendida,

a descoberto ou *short* vendendo uma ação e termina essa posição recomprando essa ação e devolvendo-a ao seu titular. O retorno r_{short} de uma posição vendida é dado por (Equação 2.3):

$$r_{short} = \frac{P_{compra} + D - P_{venda}}{P_{venda}} \tag{2.3}$$

em que P_{compra} é o preço posterior pelo qual o investidor recomprou a ação, D os dividendos distribuídos durante a posição vendida e P_{venda} o preço anterior pelo qual o investidor vendeu a ação e que iniciou a posição vendida.

Contudo, a posição vendida apresenta um risco maior comparado ao risco de uma posição comprada. Em uma posição vendida, caso o preço da ação suba em uma tendência de alta o investidor deverá recomprá-la para devolvê-la ao seu titular original por um preço arbitrariamente maior do que preço de quando vendeu a ação para iniciar o *shorting*. E como o preço da ação pode subir infinitamente, o prejuízo de uma posição vendida também pode ser muito grande a tal ponto do investidor necessitar de retirar do próprio patrimônio para recomprar a ação e devolvê-la a seu titular original o que pode levá-lo à insolvência.

Para evitar que isso ocorra, as bolsas ou as corretoras costumam exigir do investidor uma garantia ao iniciar uma posição vendida. Essa garantia, que pode ser dada em dinheiro, é chamada *margem de garantia* e serve para resguardar o investidor de uma perda significativa decorrente de uma posição vendida. As corretoras costumam terminar automaticamente uma posição vendida caso o prejuízo alcance um determinado valor da margem de garantia (e.g. caso prejuízo alcance mais 70% da margem de garantia).

Já na posição comprada o risco decorre do fato do preço da ação cair. Nesse caso, o prejuízo ficará restrito ao valor que o investidor gastou ao comprar a ação uma vez que o preço de uma ação não pode ser nulo.

Para limitar as perdas costuma-se utilizar as chamadas travas ou *stops*. Uma trava de perda ou *stop-loss* é estabelecida ao enviar uma ordem de compra para iniciar uma posição *long* ou ao enviar uma ordem de venda para iniciar uma posição *short*. Caso a perda em relação o início da posição ultrapasse o valor estabelecido no *stop-loss* a posição é automaticamente terminada comprando-se ou vendendo-se o ativo pelo preço atual (ordem *a mercado*).

O investidor pode também limitar os ganhos já obtidos para se resguardar de uma eventual perda que venha a reduzir os ganhos já alcançados. Isso é feito através uma trava de ganho ou *take-profit* que é estabelecida ao iniciar uma posição comprada ou vendida. Uma vez alcançado o nível de lucro na posição igual ou superior ao estabelecido no *take-profit* a posição é automaticamente encerrada pelo preço atual da ação.

O desempenho de um investimento ou estratégia pode ser medido de várias maneiras.

A principal métrica é o rendimento financeiro total ou final R_{total} que pode ser calculado pela diferença entre o valor inicial investido $V_{inicial}$ e o valor final do montante alcançado V_{final} (Equação 2.4):

$$R_{total} = \frac{V_{final} - V_{inicial}}{V_{inicial}} \tag{2.4}$$

Outra medida de desempenho é o máximo *drawdown* (Equação 2.5). O máximo *drawdown* MDD é a maior perda cumulativa a partir de um pico de capital alcançado até um vale posterior no tempo ao referido pico [Colby & Meyers, 1988]. Se o máximo *drawdown* de uma estratégia for muito elevado comparado com outras opções de investimento isso pode indicar que a estratégia não é adequada para a ação. O máximo *drawdown* é calculado por (Equação 2.5):

$$MDD = \max_{\tau \in (0,T)} \left[\max_{t \in (0,\tau)} C_t - C_\tau \right] \tag{2.5}$$

em que C_t denota o valor do capital acumulado em um pico e C_τ denota o capital acumulado no vale e T é o tempo no final da estratégia. A maior das diferenças $C_t - C_\tau$ é o máximo *drawdown*.

Investidores costumam preferir dentre várias estratégias de investimento aquela que proporcione o menor risco ou variabilidade [Bacon, 2008]. Assim, uma importante medida de comparação entre estratégias é o chamado Índice Sharpe (*Sharpe Ratio* em inglês). Considerando uma estratégia de investimento, seja r_p o retorno total dessa estratégia e σ_p o desvio padrão dos retornos ao longo dessa estratégia. Seja ainda r_f o retorno de uma outra estratégia chamada *livre-de-risco* como por exemplo um título do tesouro. O índice Sharpe é calculado como (Equação 2.6):

$$\text{sharpe ratio} = \frac{\mathbb{E}(r_p) - r_f}{\sigma_p} \tag{2.6}$$

Assim, quanto maior o índice Sharpe melhor é o desempenho combinado entre risco e retorno de uma estratégia. O índice Sharpe pode ser interpretado como o retorno ou recompensa em excesso (em relação a taxa *livre-de-risco*) por unidade de risco (variabilidade) por utilizar o investidor a estratégia considerada e não o investimento da taxa *livre-de-risco*. É a medida da *"recompensa"* que o investidor ganha por adotar a estratégia mais arriscada que a da taxa *livre-de-risco*.

O índice Sharpe é obtido através do coeficiente angular da chamada linha de alocação de capital (*Capital Allocation Line*) definida pela Equação 2.7.

$$\mathbb{E}(r_c) = r_f + \frac{\mathbb{E}(r_p) - r_f}{\sigma_p} \cdot \sigma_c \tag{2.7}$$

onde $\mathbb{E}(r_c)$ é o retorno esperado de um portfólio c contendo um ativo livre de risco com taxa de retorno r_f e um ativo de risco com retorno esperado denotado por $\mathbb{E}(r_p)$ e desvio padrão σ_p. O termo σ_c denota o desvio padrão dos retornos do portfólio.

O conjunto de todas combinações possíveis de portfólios da carteira c com retornos esperados $\mathbb{E}(r_c)$ e desvios padrões σ_c é delimitado por uma curva chamada fronteira eficiente em um gráfico (Vide gráfico da Figura 2.1) onde o eixo x contém os valores de risco σ_c e o eixo y contém os valores de retornos esperados $\mathbb{E}(r_c)$ da carteira c. O ponto da linha de alocação de capital que toca a curva da fronteira eficiente contém o portfólio possível com maior retorno esperado e menor risco.

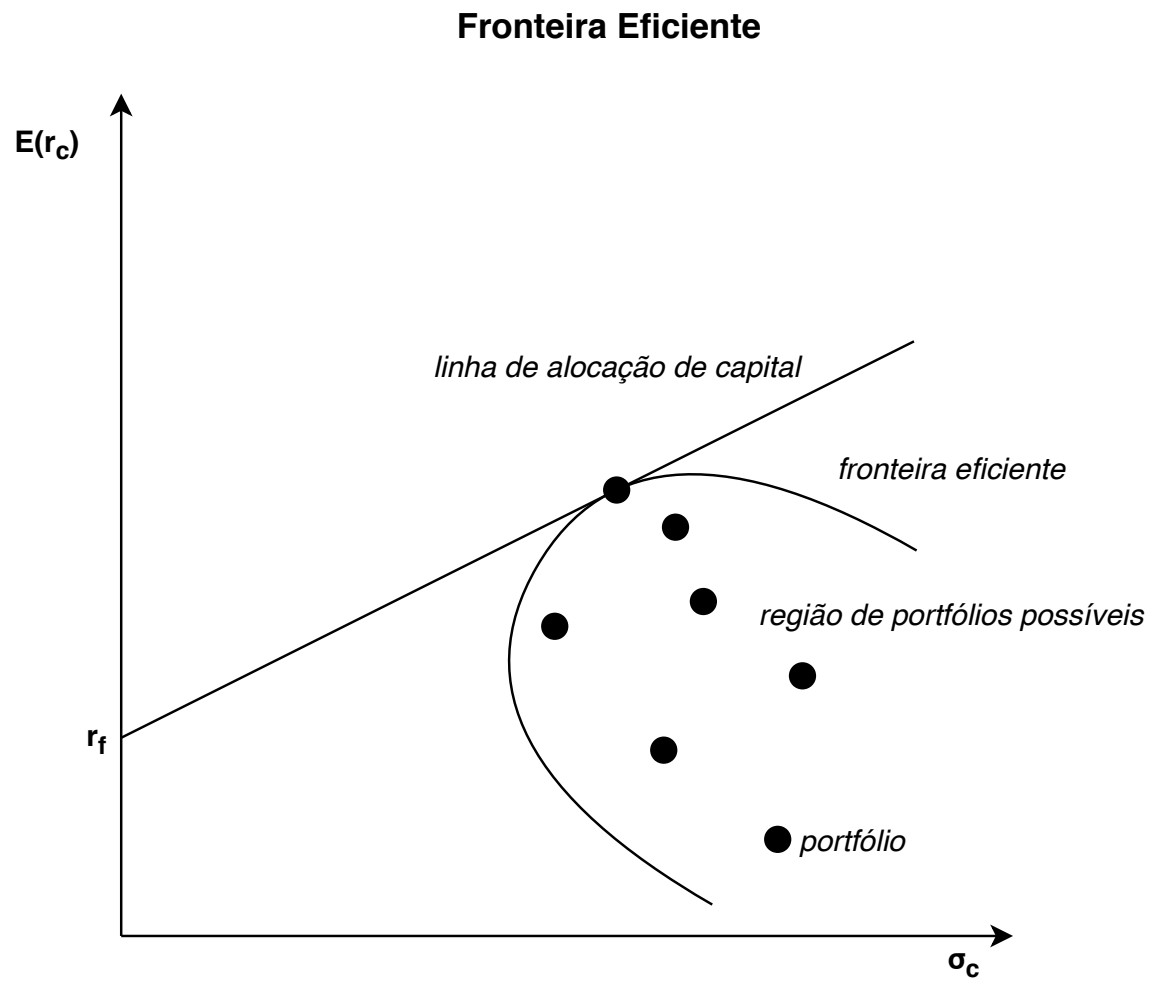

Figura 2.1: Gráfico da Fronteira Eficiente.

Nota-se, porém, que o índice Sharpe não distingue entre retornos positivos e negativos.

Se a série de retornos de uma estratégia apresenta retornos positivos significativos (o que é desejável por investidores) isso pode aumentar o denominador na Equação 2.6 penalizando os retornos positivos e diminuindo o valor do índice Sharpe fornecendo, portanto, uma falsa impressão a respeito da relação risco-retorno da estratégia.

Para contornar essa limitação do índice Sharpe costuma-se utilizar o chamado índice Sortino (*Sortino Ratio* em inglês). Nesse caso, considera-se apenas os retornos não positivos da série de retornos da estratégia para o cálculo do desvio padrão. O índice Sortino é calculado pela equação (Equação 2.8):

$$\text{sortino ratio} = \frac{r_p - r_d}{\sigma_p^-} \tag{2.8}$$

em que r_p é o retorno devido a estratégia de investimento e r_d é o chamado mínimo retorno aceitável (e.g. título de tesouro, taxa CDI). O termo no denominador σ_p^- é o desvio padrão dos retornos não positivos que pode ser calculado através da raiz quadrada da semi-variância (Equação 2.9):

$$\sigma_p^- = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\min(0, r_i))^2} \tag{2.9}$$

onde N é o número total de retornos ao longo da estratégia e r_i é o retorno obtido no tempo $i \in \{1, 2, 3, ..., N\}$.

2.2 Aprendizado Supervisionado

Um modelo de aprendizado supervisionado (*Supervised Learning* em inglês) relaciona as respostas às instâncias de um problema com o objetivo de prever as respostas em futuras observações (predição) ou entender melhor o relacionamento entre as respostas e as entradas (inferência) [James et al., 2013]. Cada instância é composta por uma ou mais variáveis chamadas preditores ou atributos (*features* em inglês).

Uma vez treinado, a capacidade do modelo de prever corretamente as respostas em instâncias que não estão presentes nos dados de treinamento é chamada de *generalização* [Bishop, 2006]. Para alcançar boa capacidade de generalização esses modelos tipicamente necessitam de grandes volumes de dados para treinamento.

Dentre os modelos de aprendizado supervisionado utilizados em sistemas de negociação de ativos financeiros estão as redes neurais MLP(*multi-layer perceptron*) [Naeini et al., 2010], SVM (*support vector machine*) [Fan & Palaniswami, 2001], árvores de decisão [Wu

et al., 2006], redes de aprendizado profundo (*deep learning*) [Chong et al., 2017], redes neurais LSTM (*long-short term memory*) [Nelson et al., 2017].

Nesse tipo de aplicação, esses modelos costumam utilizar como dados de treinamento as séries de preços e volumes em diversas periodicidades (e.g. 1 mês, 1 semana, 1 dia, 1 hora, 15 minutos, 1 minuto, milissegundos), dados de negociações (*ticks* em inglês), posições no livro de ofertas (*booking* em inglês), dados de análise de sentimento, indicadores de análise técnica, etc.

Pode-se ainda utilizar dados de análise fundamentalista os quais buscam determinar o preço adequado para uma ação com base na análise da situação financeira de uma empresa. Caso esse preço ultrapasse o valor atual da ação a análise fundamentalista pode indicar uma oportunidade de compra dessa ação Bodie et al. [2008]. Contudo, dados e indicadores de análise fundamentalista são calculados com base no balanço das empresas e são divulgados com pouca frequência o que torna a utilização desses dados inadequada para operações de negociação realizadas durante o dia (intra-day). Por isso, utilizou-se nesse trabalho somente dados de análise técnica os quais podem ser calculados diretamente a partir dos preços das ações em qualquer periodicidade.

Uma vez treinados, esses modelos podem ser combinados com estratégias de negociação preestabelecidas (Figura 2.2). Essas estratégias geralmente são baseadas em indicadores de análise de técnica e provêm tanto da experiência do próprio investidor no mercado como também da experiência de outros investidores.

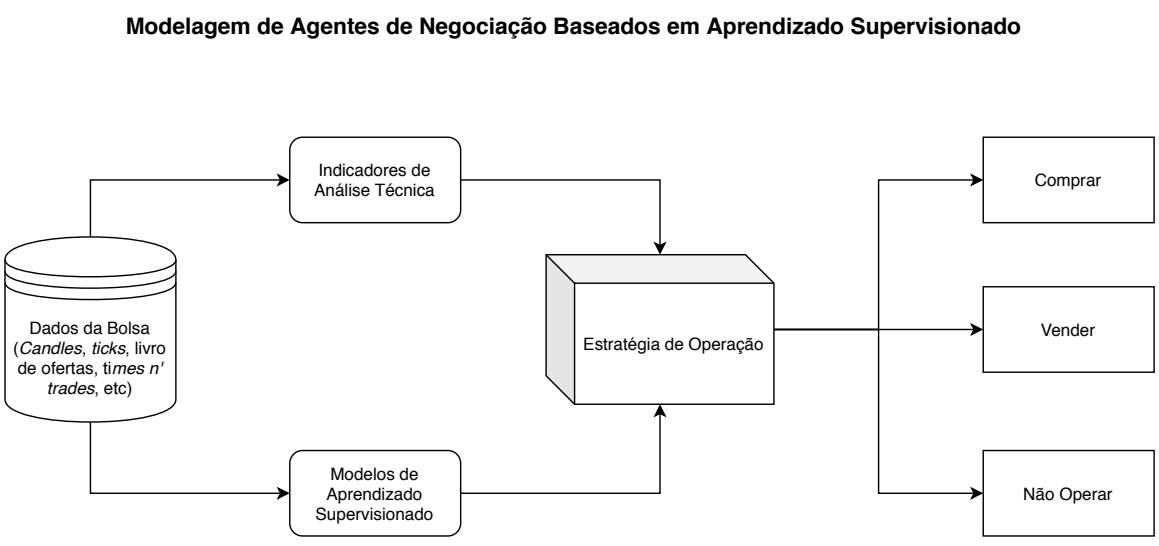

Figura 2.2: Estrutura de agentes de negociação baseados em aprendizado supervisionado.

Combinando-se modelos de aprendizado supervisionado e indicadores de análise técnica obtém-se uma estratégia de operação que produz sinais representando ordens de opera-

ção no mercado as quais constituem a estratégia de operação do robô.

Contudo, esse tipo de modelagem de sistemas de negociação apresenta algumas limitações.

A primeira delas advém da estratégia baseada em indicadores de análise de técnica. A literatura de análise técnica (Vide Kirkpatrick II & Dahlquist [2010]; Colby & Meyers [1988]) apresenta diversos indicadores técnicos divididos em várias categorias como indicadores de tendência, volatilidade, volume, *momentum*, osciladores, etc. Selecionar quais indicadores usar em uma estratégia assim como quais valores de parâmetros para cada indicador não é uma tarefa simples. Embora possa-se utilizar algoritmos evolucionários (e.g. programação genética) para selecionar e ajustar indicadores, essa abordagem fica ainda limitada às condições momentâneas dos ativos em que foi aplicada [Lohpetch & Corne, 2009; Iskrich & Grigoriev, 2017]. Ademais, uma estratégia que é lucrativa em um determinado momento e para um determinado ativo financeiro pode não ser mais no instante seguinte devido as variações de tendência, volatilidade e liquidez a que o ativo está sujeito.

Outra limitação se deve a necessidade de retreinamentos cada vez que o mercado apresenta condições significativamente distintas àquelas em que o robô foi treinado. Considerando-se que o mercado de ações pode estar sujeito a oscilações de volatilidade, tendência, volume e liquidez em intervalos tão curtos quanto um dia, uma hora ou minutos, a necessidade de retreinamentos para se adaptar as tais mudanças momentâneas do mercado pode implicar na perda de oportunidades de negócios para investidor. Isso é ainda mais claro quando se atenta ao fato de que um ciclo de retreinamento implica em treinar o robô em dados passados, validar o modelo gerado, testar via *backtesting*, otimizar parâmetros, simular em dados de tempo real para só então colocar o sistema em produção. Se todo esse ciclo for suficientemente longo as perdas de oportunidades podem até mesmo inviabilizar a utilização do sistema.

Outro fato importante é que modelos de aprendizado supervisionado otimizam funções objetivo (acurácia, precisão, erro quadrático médio) diferentes ou não relacionadas a função objetivo do sistema de negociação (maximizar o acúmulo de lucros, reduzir o risco, maximizar *Sharpe Ratio*, etc.) em que o modelo será aplicado o que também constitui outra limitação. Um modelo de aprendizado supervisionado que apresente uma alta taxa de acertos (acurácia, precisão, f-score) não necessariamente implica que o sistema de negociação em que será utilizado gerará altos rendimentos financeiros porque as perdas financeiras obtidas quando o modelo de aprendizado supervisionado erra (ainda que com pouca frequência) podem superar os ganhos quando o modelo acerta.

São essas limitações que sugerem a concepção, o estudo e experimentação de sistemas de negociação baseados em aprendizado por reforço.

2.2.1 Redes Neurais LSTM

As redes neurais LSTM (Long-Short Term Memory) são um tipo de rede neural recorrente, ou seja, uma rede capaz de processar dados sequenciais no tempo. Redes neurais recorrentes (RNN Recurrent Neural Network) implementam mecanismos de memória por meio de laços de retroalimentação entre a saída da rede e a sua entrada. A presença desses laços de retroalimentação é que possibilita esse tipo de rede neural utilizar a dimensão do tempo para associar uma determinada entrada em um tempo qualquer t à uma saída correspondente em um tempo k posterior a t . As redes neurais LSTM são bastante utilizadas atualmente em aplicações de tradução, reconhecimento de fala e escrita e análise de sentimento, o que também tem suscitado a pesquisa para sua utilização no processamento de séries temporais financeiras.

As redes neurais recorrentes comuns utilizam o método de treinamento de retropropagação no tempo *backpropagation through time* o qual geralmente sofre o efeito da explosão ou perda do gradiente do erro a medida em que dados atuais dependem de valores defasados em um passado distante. Isso é chamado dependência de longo prazo e torna a tarefa de aprendizado da rede muito custosa ou impraticável computacionalmente.

As redes neurais LSTM, contudo, não sofrem desse problema uma vez que elas implementam mecanismos de portas *gates* capazes de descartar, manter, adicionar ou atualizar informações no tempo de modo a melhor prever o próximo estado e evitar mudanças bruscas da memória o que poderia acarretar a explosão ou perda do gradiente do erro.

Essa característica das redes neurais LSTM as torna ideais para o processamento de dados sequenciais no tempo tais como linguagem natural e tradução. Esse tipo de dado caracteriza-se pelo fato de que a previsão de um estado seguinte depender do estado atual ou de um estado da rede em um momento anterior. Por isso é fundamental que o modelo de rede neural seja capaz de associar de forma eficiente dados atuais a dados remotos no tempo sem os inconvenientes da perda ou explosão do gradiente do erro.

2.3 Aprendizado Por Reforço

Segundo Russell & Norvig [2016], é possível considerar que o Aprendizado por Reforço abrange todos os elementos da Inteligência Artificial: um agente é colocado em um ambiente e deve aprender a agir satisfatoriamente nesse ambiente daí pra frente. Ou seja, no aprendizado por reforço um agente deve aprender a executar uma tarefa em um ambiente através da sua própria experiência nesse ambiente. Para Szepesvári [2010], o aprendizado por reforço enquanto problema de aprendizado, consiste em aprender a controlar um sistema para maximizar algum valor numérico que representa um objetivo no longo prazo.

Enquanto no aprendizado supervisionado um modelo recebe um conjunto de dados treinamento que associa instâncias de um problema às respostas corretas, no aprendizado por reforço o agente não dispõe desse tipo de dados de treinamento e, portanto, deve aprender através de sua própria interação com um ambiente a associar as ações(respostas) corretas aos estados(situações) do ambiente.

Essa interação consiste essencialmente em escolher e executar uma ação em cada estado do ambiente e avaliar a consequência da ação tomada naquele estado. Ao executar uma ação, o ambiente responde ao agente no instante seguinte mudando de estado e enviando uma resposta numérica chamada recompensa (*reward* em inglês) ou reforço (*reinforcement* em inglês). É por meio do valor desse reforço que o agente pode avaliar se a ação executada naquele estado foi boa ou ruim e com isso pode aprender a melhor ação em cada estado do ambiente, razão pela qual esse tipo de aprendizado é chamado de *aprendizado por reforço*.

Considerando o acúmulo de recompensas (ou reforços) como a medida de desempenho do agente, ao maximizar essa medida o agente pode executar a tarefa de maneira ótima. Portanto, o objetivo do agente de aprendizado por reforço é estimar as ações que maximizam a soma de recompensas no longo prazo. Essas ações constituem a chamada política ótima do agente.

Em geral, um problema de aprendizado por reforço pode ser descrito formalmente como um Processo de Decisão de Markov (*Markov Decision Process* em inglês). Considerando uma sequência discreta de tempo $t \in \{0, 1, 2, 3, ...\}$, um MDP pode ser definido por:

- Um conjunto finito $\mathcal{S}$ de estados sendo S_t e S_{t+1} variáveis aleatórias que denotam estados $s_t, s_{t+1} \in \mathcal{S}$ nos instantes t e $t + 1$.
- Um conjunto finito $\mathcal{A}$ de ações sendo $A(s_t) \subset \mathcal{A}$ o conjunto de ações possíveis em cada estado s_t e A uma variável aleatória denotando a ação $a_t \in A(s_t)$ executada no estado s_t.
- Um conjunto finito de recompensas $\mathcal{R} \subset \mathbb{R}$ sendo R_{t+1} uma variável aleatória denotando a recompensa $r_{t+1} \in \mathcal{R}$ obtida no instante $t + 1$.
- Uma função de recompensa $R(s_t, a_t) = \mathbb{E}[R_{t+1}|S_t = s_t, A_t = a_t]$.
- Um modelo probabilístico da dinâmica de estados e recompensas do ambiente $p(s_{t+1}, r_{t+1}|s_t, a_t) = \mathbb{P}(S_{t+1} = s_{t+1}, R_{t+1} = r_{t+1}|S_t = s_t, A_t = a_t)$.

O comportamento de um agente de aprendizado por reforço (Figura 2.3) pode ser descrito da seguinte forma: seja um espaço de estados qualquer, por exemplo, $\mathcal{S} = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. Seja também um espaço de ações para qualquer estado $s_t \in \mathcal{S}$

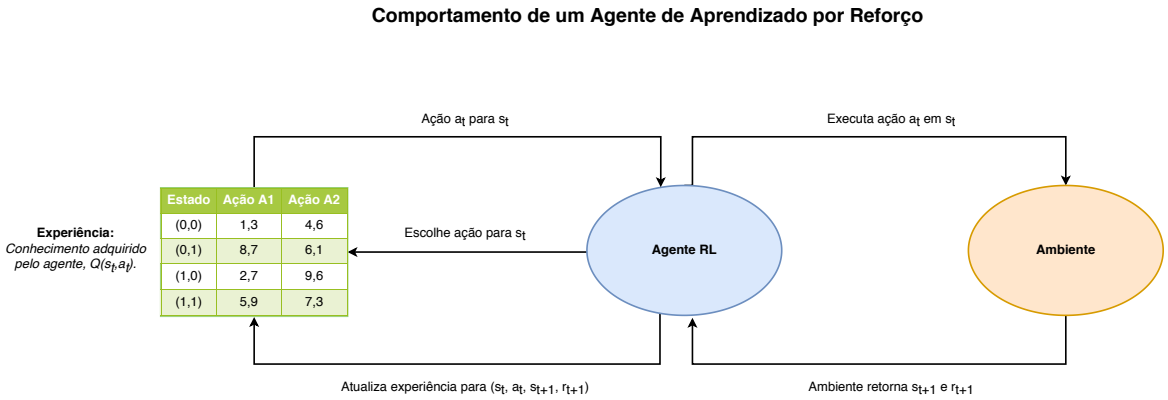

Figura 2.3: Comportamento de um agente de aprendizado por reforço

dado por $A(s_t) \in \{A1, A2\}$, por exemplo. Em um estado $s_t \in \mathcal{S}$ no instante de tempo $t \in \{0, 1, 2, 3, ...\}$ o agente escolhe uma ação $a_t \in A(s_t)$ de maior valor para o respectivo estado s_t na tabela, isto é, a que apresenta o maior valor esperado de soma de recompensas no longo prazo.

Escolhida a ação a_t, o agente executa essa ação fazendo com que o ambiente mude para o estado $s_{t+1} \in \mathcal{S}$ e retorne para o agente uma recompensa $r_{t+1} \in \mathcal{R}$. O agente, então, atualiza sua experiência (na tabela) a respeito da consequência de tomar a ação a_t no estado s_t e receber a recompensa r_{t+1}. Em seguida, prossegue escolhendo uma nova ação a_{t+1} a ser executada no estado s_{t+1} e segue dessa forma a cada novo estado até encontrar o estado final.

$$V(s_t) = \mathbb{E}\{G_t | S_t = s_t\} \tag{2.10}$$

$$Q(s_t, a_t) = \mathbb{E}\{G_t | S_t = s_t, A_t = a_t\} \tag{2.11}$$

O conhecimento adquirido pelo agente através de sua experiência é modelado na forma de uma função chamada *função-valor*. Segundo Sutton & Barto [2018], a maioria dos algoritmos de aprendizado por reforço consistem em estimar uma função-valor. Um função-valor pode ser uma função de estado-valor $V(s_t)$ (Equação 2.10) ou uma função de estado-ação $Q(s_t, a_t)$ (Equação 2.11) que, por exemplo, pode ser modelada como uma tabela tal como na Figura 2.3.

O valor da função de estado-valor $V(s_t)$ indica o valor esperado das recompensas acumuladas que um agente pode obter se começar pelo estado s_t. Por sua vez, valor da função de estado-acão $Q(s_t, a_t)$ para cada par (s_t, a_t) indica o valor esperado de recompensas acumuladas que um agente pode obter se começar no estado s_t e tomar a ação a_t. O processo de estimação dessa função utiliza métodos de programação dinâmica baseados na equação

de otimalidade de Bellman (Equação 2.12).

$$Q^*(s_t, a_t) = \sum_{s \in \mathcal{S}, r \in \mathcal{R}} p(s, r|s_t, a_t)[r + \gamma \max_{a \in A(s)} Q_*(s, a)] \tag{2.12}$$

Após cada instante t o agente recebe uma sequencia de recompensas r_{t+1},r_{t+2}, r_{t+3},..., nos instantes seguintes cuja a soma G_t é chamada de *retorno*. Se a tarefa em questão possui um estado final definido então a sequencia de tempo $t \in \{0, 1, 2, 3, ..., T\}$ é finita e assim também o retorno G_t. Porém, se a tarefa for contínua, isto é, do tipo que não possui um estado final definido, a sequencia de tempo será infinita como também o retorno. Para lidar com esse problema utiliza-se o conceito de *desconto*. Nessa abordagem o agente seleciona ações de modo que a soma das recompensas descontadas no futuro sejam maximizadas [Sutton & Barto, 2018]. O agente então busca selecionar $a_t \in A(s_t)$ que possa maximizar o retorno descontado (Equação 2.13).

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{2.13}$$

Por isso, o fator γ, $0 \leq \gamma \leq 1$ na Equação 2.12 é chamado *taxa de desconto* e permite ponderar a importância dos retornos imediatos em relação as recompensas futuras. O objetivo desse fator é fazer com que o somatório de recompensas seja um valor finito. Assim, quanto mais próximo de 0 for o fator γ, maior será o peso dado as recompensas imediatas mais próximas do momento presente. Por outro lado, quanto mais próximo de 1, maior será o peso das recompensas futuras.

Uma vez estimada a função-valor, o agente pode escolher as ações que maximizam $Q(s, a)$ para cada estado que encontrar. Esse mapeamento de estados em ações que levam ao máximo acúmulo de recompensas é chamado de política (*policy*) ótima π^*. A política ótima pode ser extraída da função de estado-ação ótima $Q^*(s_t, a_t)$ (Equação 2.14).

$$\pi^*(s_t) = \arg_{a_t \in A(s_t)} \max Q^*(s_t, a_t) \tag{2.14}$$

Para estimar essa função, o agente necessita experimentar (*exploration* em inglês) todas as ações possíveis em cada estado para descobrir qual delas leva a estados de mais alto valor na função de estado-ação $Q(s_t, a_t)$. Porém, se o agente sempre escolher a ação que maximiza $Q(s_t, a_t)$ em todo estado s_t (*exploitation* em inglês) ele poderá deixar de conhecer estados de maior valor e que levam a um maior acúmulo de recompensas. Esse dilema, chamado *exploration* v. *exploitation*, é comum em aprendizado por reforço e existem várias técnicas que buscam balancear esses dois aspectos do aprendizado. Uma delas é chamada $\epsilon - greedy$ e consiste em selecionar arbitrariamente com probabilidade pequena ϵ uma ação

a_t que não necessariamente maximiza a função de estado-ação $Q(s_t, a_t)$ em um estado s_t ou então selecionar de maneira gulosa com probabilidade $(1 - \epsilon)$ a ação que maximiza a função de estado-ação.

Os algoritmos de programação dinâmica *Policy Iteration* e *Value Iteration* utilizam a equação de otimalidade de Bellman para resolver um problema de aprendizado por reforço modelado como um MDP. Por isso são chamados algoritmos *baseados em modelo* (*model based* em inglês) porque necessitam do modelo probabilístico da dinâmica de transição de estados e recompensas do ambiente.

Porém, na prática é muito raro senão impossível obter um modelo da dinâmica do ambiente. Por exemplo, é inviável estimar um modelo de transição de estados e recompensas de um determinado jogador de xadrez dada a enorme quantidade de estados possíveis no jogo bem como outras particularidades do próprio jogo e do adversário. Por isso, utiliza-se algoritmos chamados *model-free* que dispensam um modelo do ambiente.

Dentre os algoritmos *model-free* estão os algoritmos de diferença temporal (*temporal difference* em inglês) dos quais *SARSA* e *Q-Learning* são exemplos. O que esses algoritmos fazem é observar a diferença entre a estimativa atual da função de estado-ação $Q_t(s_t, a_t)$, o valor descontado da função de estado-ação para o próximo estado s_{t+1} e a recompensa obtida r_{t+1} para então corrigir a estimativa anterior [Alpaydin, 2014]. Assim, quando o agente no estado s_t escolhe e executa uma ação a_t, o ambiente muda para para o estado s_{t+1} e retorna a recompensa r_{t+1} com as quais a função de estado-ação é atualizada (Equação 2.15).

$$Q_t(s_t, a_t) = r_{t+1} + \gamma \cdot \max_{a_{t+1}} Q_t(s_{t+1}, a_{t+1}) \tag{2.15}$$

Essa técnica baseia-se na ideia de que como o valor da função de estado-ação $Q_t(s_{t+1}, a_{t+1})$ corresponde ao instante posterior, ela tem mais chance de estar correta. Esse valor pode ser descontado pelo fator de desconto $\gamma \in (0, 1]$ e somado à recompensa obtida tornando-se o novo valor para a estimativa $Q_t(s_{t+1}, a_{t+1})$ [Alpaydin, 2014].

Os algoritmos *SARSA* e *Q-Learning* utilizam regras diferentes para atualização da função estado-ação mas ambos utilizam o conceito de diferença temporal. No algoritmo *Q-Learning* a função de estado-ação é atualizada na forma da Equação 2.16

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \tag{2.16}$$

onde o parâmetro $\alpha \in (0, 1]$ é chamado de *taxa de aprendizado*.

O algoritmo *SARSA* (Algoritmo 1), por sua vez, utiliza uma regra de atualização diferente considerando o estado s_t, a ação executada a_t, a recompensa obtida r_{t+1} e o estado alcançado s_{t+1} e a próxima ação a ser executada a_{t+1} o que dá origem ao seu nome *SARSA*.

Algoritmo 1: Algoritmo *SARSA* conforme Alpaydin [2014]

Inicializa todos os valores de Q(s,a) arbitrariamente;
foreach *episodio* **do**
  Inicializa estado inicial s;
  Escolhe ação a em Q usando $\epsilon - greedy$;
  repeat
    Executa ação a;
    Obtém recompensa r e próximo estado s';
    Escolhe próxima ação a' em Q com $\epsilon - greedy$;
    `// Atualiza Q(s,a)`
    $Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma Q(s',a') - Q(s,a)\right]$;
    $s \leftarrow s'$;
    $a \leftarrow a'$;
  until s *é estado terminal*;
end

Enquanto os espaços de estados de um problema de aprendizado por reforço forem pequenos e discretos as funções de valor podem ser modeladas na forma de uma tabela. Denomina-se esse tipo de modelagem de *modelagem tabular*. Porém, muitos problemas práticos apresentam espaços de estados ou ações contínuos o que inviabiliza a modelagem na forma tabular.

Com um espaço de estados tão grande o aprendizado só é possível por meio de amostras desse espaço utilizando técnicas de aproximação de funções que buscam generalizar as saídas da função para as demais instâncias do espaço de estados. Esse é um problema comum em *aprendizado supervisionado* o qual oferece várias técnicas como as redes neurais, regressão linear, árvores de decisão, redes neurais profundas e outros para aproximação de funções.

Combinando as técnicas existentes de aproximação de funções com os algoritmos de aprendizado por reforço pode-se estimar uma aproximação para as funções de valor em problemas com espaços de estados muito grandes.

Nesse caso, a função de estado-valor $V(s, \boldsymbol{w})$ é parametrizada por um vetor de pesos $\boldsymbol{w} \in \mathbb{R}^{\mathrm{d}}$. Considerando um estado qualquer $s \in \mathcal{S}$ como um vetor de dimensão $d \in \mathbb{N}^*$ dado por $s = (s_1, s_2, s_3, ..., s_d)^\intercal$, uma formulação possível para a função de estado-valor utiliza uma função linear nos pesos tal como na Equação 2.17.

$$V(s, \boldsymbol{w}) = \boldsymbol{w}^\intercal s = \sum_{i=1}^{d} w_i s_i \tag{2.17}$$

Derivando a função $V(s, \boldsymbol{w})$ em relação a $\boldsymbol{w}$ pode-se obter uma regra de atualização

(Equação 2.18) que utiliza a técnica do gradiente descendente estocástico (*Stochastic Gradient Descent* em inglês ou SGD) para estimar a função de valor.

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha \left[r_{t+1} + \gamma V(s_{t+1}, \boldsymbol{w}_t) - V(s_t, \boldsymbol{w}_t) \right] \nabla V(s_t, \boldsymbol{w}_t) \tag{2.18}$$

Além da modelagem através de uma função linear, pode-se ainda utilizar uma rede neural aproveitando-se a sua propriedade de aproximador universal de funções [Haykin, 1994]. Mais recentemente, valendo-se dos avanços proporcionados pelas técnicas de Aprendizado Profundo (*Deep Learning* em inglês) começou-se a utilizar as redes neurais profundas em aprendizado por reforço dando origem ao chamado Aprendizado por Reforço Profundo (*Deep Reinforcement Learning* em inglês).

Um dos primeiros e principais trabalhos utilizando aprendizado por reforço foram o de Mnih et al. [2015] que utilizou uma rede neural profunda com o algoritmo *Q-Learning* para melhorar o desempenho de um agente em jogos de Atari através do algoritmo *Deep-Q Network* ou DQN e o trabalho de Silver et al. [2016] que utilizando o algoritmo *Asynchronous Advantage Actor-Critic* ou A3C criou um agente chamado *AlphaGo* que derrotou o campeão mundial do jogo de tabuleiro *Go*.

A partir de então tem-se utilizado as redes neurais profundas (e.g. redes de convolução, LSTM, GRU, etc.) em aprendizado por reforço aproveitando a propriedade dessas redes de descobrir representações ocultas em dados de alta dimensão tais como imagens, texto, som, vídeo e até séries temporais financeiras. Desse modo, consegue-se obter melhores representações dos estados do ambiente em dados de alta complexidade o que é fundamental na modelagem de um problema de aprendizado por reforço.

Capítulo 3

Trabalhos Relacionados

Dentre os diversas modelos de aprendizado de máquina utilizados em agentes automatizados de negociação de ativos financeiros destacam-se as redes neurais artificiais. Dessas redes, uma das mais utilizadas para previsão de retornos em série temporais financeiras é a rede neural LSTM (*Long-Short Term Memory*) [Hochreiter & Schmidhuber, 1997].

A rede LSTM é um tipo de rede neural recorrente (*RNN - Recurrent Neural Network*) o que significa que esse tipo de rede neural implementa mecanismos de memória através de laços de retroalimentação possibilitando o processamento de dados sequenciais no tempo [Padua Braga, 2007].

A rede LSTM também é capaz de associar dados remotos no tempo a dados atuais para melhor prever o valor de uma série sem o inconveniente da perda do gradiente do erro durante o treinamento, um problema comum nas redes RNN tradicionais [Chen et al., 2015]. Para isso, esse modelo de rede neural pode processar dados de séries temporais descartando, atualizando, mantendo e adicionando informações de modo a melhor prever o estado (memória) da rede e produzir a saída mais provável.

Partindo dessas propriedades, as redes LSTM tem sido aplicadas frequentemente em contextos de finanças.Por exemplo, Nelson et al. [2017] desenvolveu rede neural LSTM para um sistema de previsão da direção da variação de preços de ações. Considerando que a rede LSTM obteve uma acurácia média de 55,9% nas previsões, foi possível desenvolver um sistema de negociação de ações que proporcionou retornos financeiros superiores ao um *baseline Buy-and-Hold* nas ações testadas e outros tipos de abordagens como *Random Forest*.

No trabalho de Faustryjak et al. [2018], o autor combinou as previsões de preços de ações a partir de uma rede neural LSTM com dados de notícias das ações no *Google Trends* para fornecer recomendações de compras das ações. A utilização da rede LSTM gerou uma melhora de 51,9% até 58,2% nas previsões comparado com um modelo de rede neural co-

mum MLP (*Multi-Layer Perceptron*).

Kim & Won [2018] também combinaram rede neural LSTM com modelos de séries temporais GARCH (*Generalized Autoregressive Conditional Heteroscedasticity*) para prever a volatilidade (variância) do índice KOSPI200 da bolsa de valores de Seul na Coreia do Sul. Esse trabalho mostrou que o modelo combinado foi capaz de gerar previsões de volatilidade com erro absoluto médio de até 0,0107.

Um resultado semelhante para previsão de volatilidade foi obtido por Liu et al. [2018]. Esse autores testaram uma rede neural LSTM com 3 camadas alcançando uma acurácia de até 78% nas previsões de variação da volatilidade do índice CSI300 da Bolsa de Xangai na China . Outro trabalho com resultados semelhantes utilizando rede neural LSTM para prever dados de alta frequência de ações foi de Yao et al. [2018]. Outros trabalhos mais recentes e seguindo a mesma linha são os de Ghosh et al. [2019]; JuHyok et al. [2020].

Esses e outros trabalhos apresentam evidências da viabilidade da utilização de rede neural LSTM em contextos de séries temporais financeiras sobretudo no desenvolvimento de agente para negociação de ações.

Porém, por se tratar de um modelo de aprendizado supervisionado, esse tipo de modelo depende de constantes retreinamentos devido as constantes mudanças nas condições do mercado de bolsas de valores. Nesse sentido, Li et al. [2019a] afirmam que embora os modelos de aprendizado supervisionado apresentem boa acurácia na previsão de tendências e retornos no mercado, esses modelos não são robustos à dinâmica real do mercado e portanto não podem ser aplicados diretamente na tarefa de negociação (*algotrading*). Seguindo mesmo entendimento Hu & Lin [2019]; Lei et al. [2020].

Por isso, surge a necessidade de se propor, estudar e desenvolver sistemas capazes de se adaptarem *dinamicamente* às condições variáveis do mercado. Segundo os autores acima mencionados, a principal alternativa disponível é o aprendizado por reforço.

A revisão da literatura a respeito do emprego de técnicas de aprendizado por reforço no contexto do mercado financeiro demonstra que a utilização dessa abordagem começou há pelo menos 2 décadas dividindo-se entre dois ramos principais: o aprendizado por reforço "tradicional" e o aprendizado por reforço profundo (Figura 3.1). Essa divisão se estabeleceu sobretudo a partir de 2015 com o advento do aprendizado por reforço profundo através dos trabalhos de Mnih et al. [2015]; Silver et al. [2016]. Desde então, tem crescido o interesse acadêmico no estudo da utilização do aprendizado por reforço em contexto de mercado financeiro.

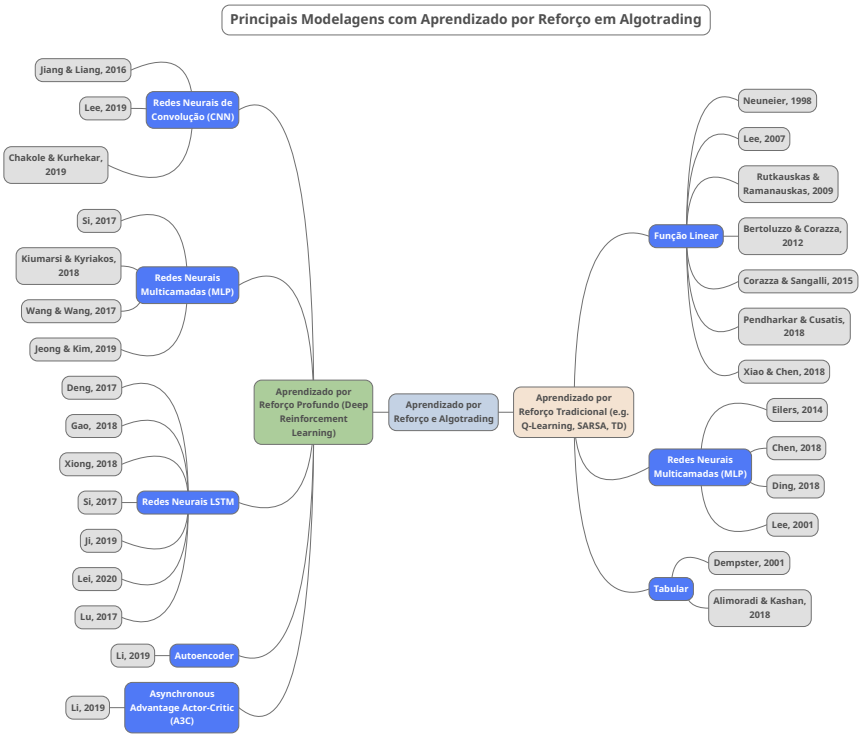

Figura 3.1: Principais abordagens de aprendizado por reforço em *algotrading* e respectivos trabalhos. Fonte: elaboração própria

3.1 Aprendizado por Reforço Tradicional

A nomenclatura *aprendizado por reforço tradicional* se deve a Jia et al. [2019] e foi usada também nesse trabalho. No chamado aprendizado por reforço tradicional utiliza-se em geral os algoritmos Q-Learning, SARSA, TD(λ) variando a modelagem em termos da função de valor, atributos de estados e função de recompensa.

Observa-se na Figura 3.1 que vários trabalhos nessa linha utilizaram função linear como aproximação para função de valor. Isso se deve pela simplicidade de implementação como também pela dificuldade na época de implementar e treinar redes neurais muito complexas. Além, segundo Sutton & Barto [2018], a modelagem com função de aproximação linear produz soluções que aproximam-se bastante do máximo global a medida que a taxa de aprendizado (parâmetro α na Equação 2.16) decai no tempo.

Um dos primeiros trabalhos nessa linha foi de Neuneier [1998] utilizou o algoritmo Q-Learning para alocação de portfólio enquanto Moody & Saffell [2001] utilizaram um sistema implementado com Q-Learning para negociar um portfólio contendo o índice S&P500 e um

título do tesouro americano T-Bill. Dessa vez, utilizaram como medida de recompensa o índice Sharpe que mede o quanto o portfólio de ações pode render em relação ao risco do investimento.

Na mesma época, Dempster et al. [2001] e outros compararam agentes que utilizaram algoritmo genético e o algoritmo Q-Learning para gerar estratégias de negociação para câmbio Euro/Dólar americano. No espaço de estados utilizaram 16 sinais de compra e venda de indicadores técnicos e o retorno como métrica de recompensa. Esse trabalho demonstrou que ambos as abordagens foram capazes de gerar estratégias lucrativas sendo que o sistema que utilizou algoritmo genético foi menos suscetível a *overfitting*.

Lee et al. [2007] criaram um sistema multi-agente cooperativo para negociação de ações utilizando o algoritmo Q-Learning e função linear para aproximação de função de valor-estado de cada agente. Esse sistema constitui-se de quatro agentes sendo dois deles responsáveis por gerar sinais de comprar e vender e os outros dois agentes responsáveis por gerar o melhor valor para a compra ou para a venda. Para modelagem de estados utilizaram uma matriz binária contendo os sinais gerados pelos agentes e sinais de indicadores técnicos. Os resultados dos experimentos desse sistema em dados da bolsa de valores da Coreia do Sul superaram outras abordagens baseadas em aprendizado supervisionado principalmente em relação a redução de custo de operação.

Rutkauskas & Ramanauskas [2009] também utilizaram uma abordagem semelhante para simular um mercado de ações e estudar seu comportamento quanto aos fundamentos do mercado tais como auto-regulação dos preços, importância do comportamento individual e da população de agentes para a eficiência do mercado e a relação entre os preços das ações e a liquidez.

Bertoluzzo & Corazza [2012] testaram diferentes configurações de agentes utilizando Q-Learning associado a uma função linear para modelagem da função de estado ação e *Kernel-Based Reinforcement Learning* que utiliza um método de regressão baseado em um kernel. Para a modelagem de estados utilizaram os 5 últimos retornos e a função de recompensa foi o Sharpe Ratio.

Corazza & Sangalli [2015] compararam a performance de dois agentes, um implementando o algoritmo SARSA e outro Q-Learning para negociar um conjunto de ações da bolsa de valores de Milão na Itália. Em ambos os agente usaram função linear para modelagem da função de estado-ação. Ambos os agentes superaram métricas de *baseline* sendo o algoritmo SARSA mais sensível a mudanças bruscas no mercado enquanto o algoritmo Q-Learning foi melhor ao explorar o mercado gerando mais ordens de compra e venda.

Almahdi & Yang [2017] utilizaram o algoritmo RRL de Moody e Saffell para desenvolver um sistema adaptativo de otimização de portfólio de ações utilizando como métrica o *máximo drawdown*. Esse métrica mede a maior perda percentual do portfólio de ações a

partir de um pico no valor alcançado no investimento. Os experimentos realizados demonstraram que o sistema foi capaz de reduzir o número de operações e produzir rendimentos superiores aos de fundos de pensões comparado com o mesmo sistema utilizando a métrica de risco índice Sharpe.

Chen et al. [2018] desenvolveram um sistema de aprendizado por reforço para clonar estratégias de investimentos de investidores experientes. O sistema desenvolvido foi testado com dados do índice futuro TAIFEX da bolsa de valores de Taiwan. Experimentos realizados pelos autores demonstraram que o sistema foi capaz de acertar em até 80% as ações tomadas por um investidor experiente. Ding et al. [2018] também usou essa abordagem para extrair conhecimento de estratégias de investimento a partir de dados históricos de negociação de ações empregando aprendizado por reforço. O sistema desenvolvido pelos autores mostrou-se eficiente em extrair conhecimento de 3 investidores modelos e gerar estratégias que superaram *baselines* como *Buy-and-Hold* utilizando o conhecimento adquirido.

Pendharkar & Cusatis [2018] também compararam a performance de agentes utilizando os algoritmos SARSA, Q-Learning e $TD\left(\lambda\right)$ para negociar um portfólio contendo o índice S&P500 e um título do tesouro americano tendo os agentes apresentado resultados semelhantes em termos de rendimentos financeiros durante os testes. Além concluíram que sistemas implementados produzem melhores resultados quando operam anualmente comparados com experimentos realizados em conjuntos de testes semestrais e trimestrais.

Alimoradi & Kashan [2018] combinaram o algoritmo de otimização global LCA (*League Championship Algorithm*) com os algoritmos SARSA e Q-Learning para obter estratégias de negociação utilizando indicadores técnicos e testá-las com dados de ações da bolsa de valores de Teerã. O modelo criado pelos autores apresentou desempenho superior a estratégia *Buy-and-Hold* além de desempenho superior nos casos em que as ações apresentaram tendências de altas.

Xiao & Chen [2018] usaram o algoritmo Q-Learning com função linear e dados de análise de sentimento, retornos passados, volume e volatilidade para criar um agente de negociação. Usando o retorno como função de recompensa e testado nas ações da Ford e Tesla da bolsa de valores de Nova Iorque o agente superou *baselines* baseados em SVM e regressão linear.

3.2 Aprendizado por Reforço Profundo

Embora o trabalho de Lee [2001] tenha utilizado uma rede neural simples foi só após a popularização das linguagens e bibliotecas de programação em placas gráficas aceleradores (GPU - *Graphics Processing Unit*) (e.g Cuda, OpenCL, Keras, Tensorflow, PyTorch, Theano) por

volta de 2013 que os trabalhos com redes neurais mais complexas começaram a serem desenvolvidos culminando por volta de 2015 nos primeiros trabalhos empregando redes neurais profundas e aprendizado por reforço.

Abordagens combinando aprendizado profundo (*Deep Learning*) e aprendizado por reforço tem surgido nos últimos anos sendo chamada de *Deep Reinforcement Learning*. Nessa abordagem utiliza-se redes neurais profundas para a modelagem da função de valor tais como redes de convolução, redes LSTM, GRU (*Gated Recurrent Unit* em inglês) e outras.

Seguindo os avanços da rede neural LSTM em aplicações de processamento de sinais e linguagem natural observa-se da Figura 3.1 que vários trabalhos utilizam esse tipo de rede neural valendo-se de sua propriedade de associar estados de memória longa e curta sem perder o gradiente do erro (*gradient vanishing* em inglês) o que permite explorar associações de padrões temporais em series financeiras [Jia et al., 2019].

Outro tipo de rede utilizada nesse contexto são as redes de convolução seguindo a linha proposta pelo modelo de *Deep Q-Network* [Mnih et al., 2015]. O objetivo é utilizar essas redes para extrair padrões complexos nas séries de dados financeiros o que é útil na modelagem de estados de um agente de aprendizado por reforço. Com o mesmo objetivo, já utilizou-se também as redes neurais multicamadas e *autoencoder*.

Um dos primeiros trabalhos foi o Jiang & Liang [2016] que utilizou aprendizado por reforço profundo associado a uma rede de convolução para a negociação de um portfólio de cripto moedas.

Deng et al. [2017] utilizaram *Deep Reinforcement Learning* para criar um agente para negociação de ações e títulos futuros das bolsas de valores da China, Japão e Estados Unidos.O sistema desenvolvido utilizou redes neurais de convolução e LSTM. Outros trabalhos semelhantes como os de Gao [2018] e o de Xiong et al. [2018] utilizaram redes neurais LSTM e o algoritmo de aprendizado por reforço Q-Learning apresentando resultados superiores a *baselines* baseados em aprendizado supervisionado e Buy-and-Hold.

Si et al. [2017] combinaram o modelo *Recurrent Reiforcement Learning* proposta por Moody & Saffell [1998] com *Deep Reinforcement Learning* para criar um agente multi-objetivo que otimiza risco e ganho financeiro. Esse agente superou a estratégia *Buy-and-Hold* nas simulações tendo apresentado também um desempenho superior ao modelo RRL quando as séries de preços apresentaram tendências. Utilizaram ainda 4 camadas de uma rede neural densa para extrair características das séries de preços de ativos de mercado futuro da China e também uma rede LSTM para modelar a função de estado-ação.

Jia et al. [2019] usou uma rede LSTM para detectar padrões temporais nas séries de preços, volume e indicadores técnicos de ativos de mercado futuro da China e combinou com um agente de aprendizado por reforço profundo. O agente produziu resultados positivos na maioria dos ativos testados embora tenha sido observado que o agente demora para mudar

de estratégia quando ocorrem grandes oscilações de preços. Na mesma linha, o trabalho de Lei et al. [2020].

Lee et al. [2019] utilizou uma abordagem peculiar baseado em *Deep Q-Network* para criar uma agente de negociação utilizando redes de convolução alimentadas por uma sequencia de imagens dos últimos 5 dias das séries de preços e volume.

Li et al. [2019b] desenvolveu uma modelagem complexa combinando rede neural LSTM para modelar padrões temporais das series de dados financeiros e autoencoder para a modelagem de estados e o algoritmo *Asynchronouns Advantage Actor-Critic*, também conhecido como A3C, que combina aprendizado por reforço utilizando função de valor e *Policy Search* de modo distribuído.

3.3 Análise

Dos trabalhos acima relacionados nota-se um interesse no estudo e uso de redes neurais profundas para modelagem de estados em aprendizado por reforço baseando-se na propriedade dessas redes de descobrir representações ocultas em dados de alta complexidade.

Em que pese os estudos utilizando aprendizado por reforço profundo tenham apresentado bons resultados em relação aos seus respectivos *baselines* essa abordagem tem a desvantagem de demandar alto custo computacional para o processamento das redes de aprendizado profundo além de também de sofrer de problemas de convergência devido a auto-correlação entre os dados de entrada e não-estacionariedade das condições do ambiente conforme foi observado por Hu & Lin [2019], Meng & Khushi [2019]. Essa abordagem ainda padece do problema do mal da dimensionalidade (*dimensionality curse* em inglês) quando aplicada em dados de alta dimensionalidade e complexidade como também é pouco responsiva na presença de *outliers* em séries de preços e portanto pode falhar em grandes oscilações no mercado [Jia et al., 2019].

Nota-se também dos trabalhos relacionados que pouco foi explorado com relação ao algoritmo SARSA empregando uma modelagem de estados tabular. Esse tipo de modelagem tem a vantagem de apresentar convergência para a política ótima desde que os estados sejam visitados um número grande de vezes e que a taxa de exploração decaia ao longo da execução [Sutton & Barto, 2018]. Por isso, apesar do crescente interesse nas redes neurais profundas também tem surgido nos últimos anos modelagens apresentando espaços de estados discretos tal como o trabalho de Pendharkar & Cusatis [2018] que utiliza somente 4 estados para modelar o estado de um portfólio de 2 ativos.

A modelagem proposta nesse trabalho segue essa mesma linha ao apresentar um agente de negociação de ações empregando o algoritmo SARSA, com espaço de estados de estados

discreto e finito, além de propor uma metodologia para testar as propriedades dinâmicas do agente em contextos de tendências variados bem como analisar o desempenho financeiro do agente através de testes do agente para várias ações em um contexto de instabilidade no mercado comparando também com um agente de negociação baseado em aprendizado supervisionado.

Capítulo 4

Modelagem do Problema

Pretende-se com a modelagem a seguir obter um agente de aprendizado por reforço para negociação de ações com o objetivo de maximizar o retorno financeiro negociando uma ação por vez na bolsa de valores. Para isso o agente interage com o mercado comprando, vendendo ou não operando de modo a aprender dinamicamente a melhor decisão (comprar, vender, não operar) a ser tomada em cada estado do mercado. Ao longo da metodologia e dos experimentos o agente de aprendizado por reforço modelado adiante será referido como *Agente RL*.

Ressalte-se que não foram levados em conta aspectos como custos de transação, taxas de corretagem, alavancagem, emolumentos e tributos como forma de simplificar a análise dos resultados financeiros e torná-los menos dependentes dessas variáveis.

Como em todo problema de aprendizado por reforço, isso envolve a definição de um espaço de estados, um conjunto de ações e uma função de recompensa.

4.1 Espaço de Estados

Utilizou-se para esse sistema um espaço de estados discreto contendo 4 variáveis categóricas. Logo, para cada instante de tempo $t \in \{0, 1, 2, 3, ...\}$ tem-se um estado s_t definido como uma tupla de 4 variáveis descritas abaixo:

1. Tipo de posição: (`LONG`, `SHORT`, `NPOS`). Essa variável descreve o tipo de posição do agente na ação. O valor `LONG` refere-se a uma posição comprada, `SHORT` refere-se a uma posição vendida e `NPOS` denota que o agente não está posicionado e portando não possui nenhuma ação no momento.

2. Ação tomada em $t - 1$: (`BUY`, `SELL`, `NOP`). Essa variável descreve a decisão que foi tomada pelo agente no tempo anterior. `BUY` denota que o agente comprou a ação,

SELL denota que o agente vendeu a ação e NOP significa que o agente não operou.

3. Extremo mais próximo do preço de fechamento em $t-2$: (MAX, MIN). Essa variável indica se o preço de fechamento da ação no tempo $t-2$ estava mais próximo do preço máximo (MAX) da ação ou do preço mínimo (MIN).

4. Extremo mais próximo do preço de fechamento em $t-1$: (MAX, MIN). Essa variável descreve o mesmo comportamento da variável anterior mas no tempo $t-1$.

Dessa forma, o espaço de estados modelado com as 4 variáveis acima contém 36 estados possíveis.

4.2 Conjunto de Ações do Agente

Em cada estado s_t o agente pode tomar as seguintes ações $a_t \in A(s_t)$ que vão determinar ordens diferentes dependendo do tipo de posição financeira em que o agente estiver:

- COMPRAR: $\{$ENTER_LONG, EXIT_SHORT$\}$. A ação ENTER_LONG inicia uma posição comprada (LONG) e a ação EXIT_SHORT sai de uma posição vendida (SHORT).

- VENDER: $\{$ENTER_SHORT, EXIT_LONG$\}$. A ação ENTER_SHORT inicia uma posição vendida (SHORT) e a ação EXIT_LONG sai de uma posição comprada (LONG).

- NOP: $\{$STAY_LONG, STAY_SHORT, NOP$\}$. A ação STAY_LONG determina que o agente permaneça em uma posição comprada (LONG). A ação STAY_SHORT determina que o agente permaneça em uma posição vendida (SHORT) e a ação NOP indica que o agente permaneça não posicionado (NPOS).

O conjunto de ações acima descreve portanto uma máquina de estados, conforme ilustra a Figura 4.1.

Note que se o agente estiver posicionado em LONG ou SHORT as ações que ele pode executar correspondem a permanecer na respectiva posição $\{$STAY_LONG, STAY_SHORT$\}$ ou sair da posição $\{$EXIT_LONG, EXIT_SHORT$\}$. Portanto, uma vez assumida uma posição financeira em um ativo o agente não pode aumentar o volume financeiro nessa posição comprando mais ações em uma posição comprada (LONG) ou vendendo mais ações em uma posição vendida (SHORT). Optou-se por um volume fixo a cada negociação para simplificar a análise dos resultados financeiros. Dessa forma, uma posição comprada pode ser interpretada como simétrica a uma posição vendida e vice-versa o que facilita também o aprendizado do agente.

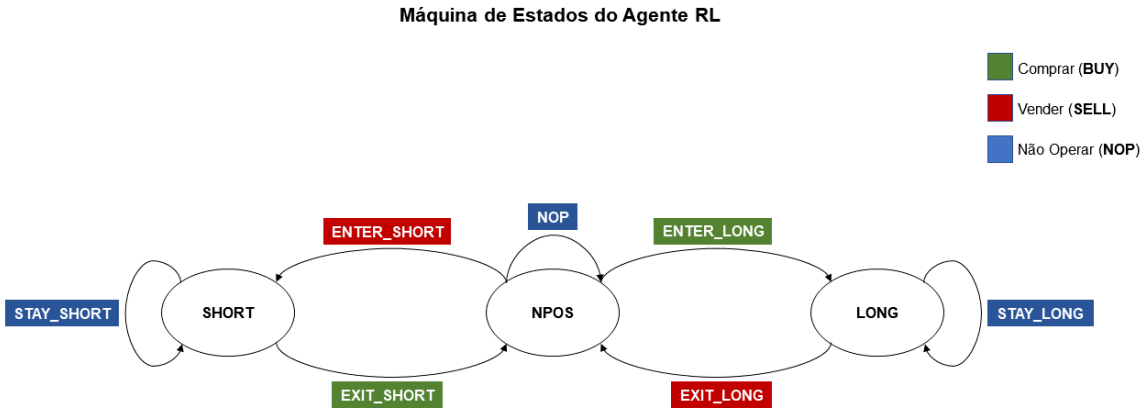

Figura 4.1: Máquina de estados do agente

4.3 Função de Recompensa

Para cada ação $a_t \in A(s_t)$ tomada pelo agente no instante t no estado atual s_t, o ambiente retorna ao agente no instante seguinte $t+1$ uma recompensa $r_{t+1} \in \mathbb{R}$ que vai depender do tipo de posição assumida pelo agente no instante atual.

Para todas as ações que denotam entrada ou permanência em uma posição $\{\texttt{ENTER\_LONG}, \texttt{ENTER\_SHORT}, \texttt{STAY\_LONG}, \texttt{STAY\_SHORT}\}$ a recompensa r_{t+1} será 0.

Para as ações que denotam saída de uma posição $\{\texttt{EXIT\_LONG}, \texttt{EXIT\_SHORT}\}$ a recompensa r_{t+1} será dada pela Equação 4.1 em que P_{enter_long} e P_{enter_short} denotam o valor da compra quando o agente entrou em uma posição comprada ou o valor da venda quando o agente entrou em uma posição vendida, respectivamente. Por sua vez, P_{exit_long} e P_{exit_short} denotam o valor da venda quando o agente saiu de uma posição comprada ou o valor da compra quando o agente saiu de uma posição vendida, respectivamente.

$$r_{t+1} = \begin{cases} P_{\text{exit-long}} - P_{\text{enter-long}} & \text{se posição for } \texttt{LONG} \\ P_{\text{enter-short}} - P_{\text{exit-short}} & \text{se posição for } \texttt{SHORT} \\ 0 & \text{se posição for } \texttt{NPOS} \end{cases} \tag{4.1}$$

4.4 Estratégia de Exploração

Uma vez que um algoritmo de aprendizado por reforço não é instruído por um conjunto de treinamento a tomar decisão correta em cada estado tal como ocorre no aprendizado supervisionado, ele deve então descobrir através de sua própria experiência qual a melhor ação a

tomar em cada estado. Para isso, ele deve experimentar (*exploration*) todas as ações possíveis para cada estado para descobrir qual delas é a melhor. Por outro lado, se o agente sempre escolhe a ação que maximiza a função de estado-ação (*exploitation*) ele pode deixar de conhecer estados que levam a uma maior soma de recompensas. Para balancear *exploration–exploitation* utilizou-se a estratégia ϵ – *greedy* com decaimento exponencial ao longo do treinamento do agente segundo a equação (Equação 4.2)

$$\epsilon_t = e^{\ln(p)+c \cdot t \cdot \frac{\ln(z)-\ln(p)}{T}} \tag{4.2}$$

em que para cada iteração $t = \{0, 1, 2, 3, ..., T\}$ do algoritmo SARSA durante o treinamento, a taxa de exploração ϵ_t decai do valor $\epsilon_0 = p$ para o valor $\epsilon_T = z$ sendo as constantes z um valor próximo de zero e c uma taxa de decaimento. Portanto, a constante p designa o valor inicial da probabilidade de exploração da estratégia ϵ – *greedy* e a constante T é o número total de iterações do algoritmo SARSA durante o treinamento. Desse modo o agente começa o treinamento explorando as ações de cada estado com probabilidade $\epsilon_0 = p$ e termina *exploiting* com probabilidade $\epsilon_T = z \cong 0$. A Equação 4.2 é baseada nas equações de decaimento exponencial na forma $N(t) = N_0 e^{-\lambda t}$.

Por exemplo, para os valores de constantes $p = 0,5$, $T = 5000$, $c = 0,18$ e $z = 2 \cdot 10^{-16}$ os valores da taxa de exploração ϵ_t descrevem a curva do gráfico da Figura 4.2 ao longo de 5000 iterações (ou episódios) no treinamento.

4.5 Fluxo de Execução do Agente

A partir da modelagem apresentada acima, considere o diagrama de fluxo de execução apresentado na Figura 4.3 que ilustra os principais aspectos da referida modelagem para um algoritmo de aprendizado por reforço no contexto de negociação de ações.

A raia inferior apresenta os instantes de tempo quando os dados de preços (Abertura, Máximo, Mínimo, Fechamento) estão disponíveis. A raia denominada *Ambiente* apresenta os estados enquanto na raia denominada *Agente* são apresentadas as decisões (ações) do agente. A raia superior mostra os retornos financeiros em razão das respectivas ações do agente.

Considere o primeiro instante de tempo `2015-01-02 11:00:00`. Nesse momento os dados de preços são apresentados ao ambiente. Nesse instante, como o agente não está posicionado (Posição atual é `NPOS`) seu retorno financeiro é nulo e o estado atual corresponde a tupla `(NPOS,NOP,MAX,MIN)` assumindo que no instante anterior ao atual o preço de fechamento ficou mais próximo do preço de máximo e que o agente executou a ação de não

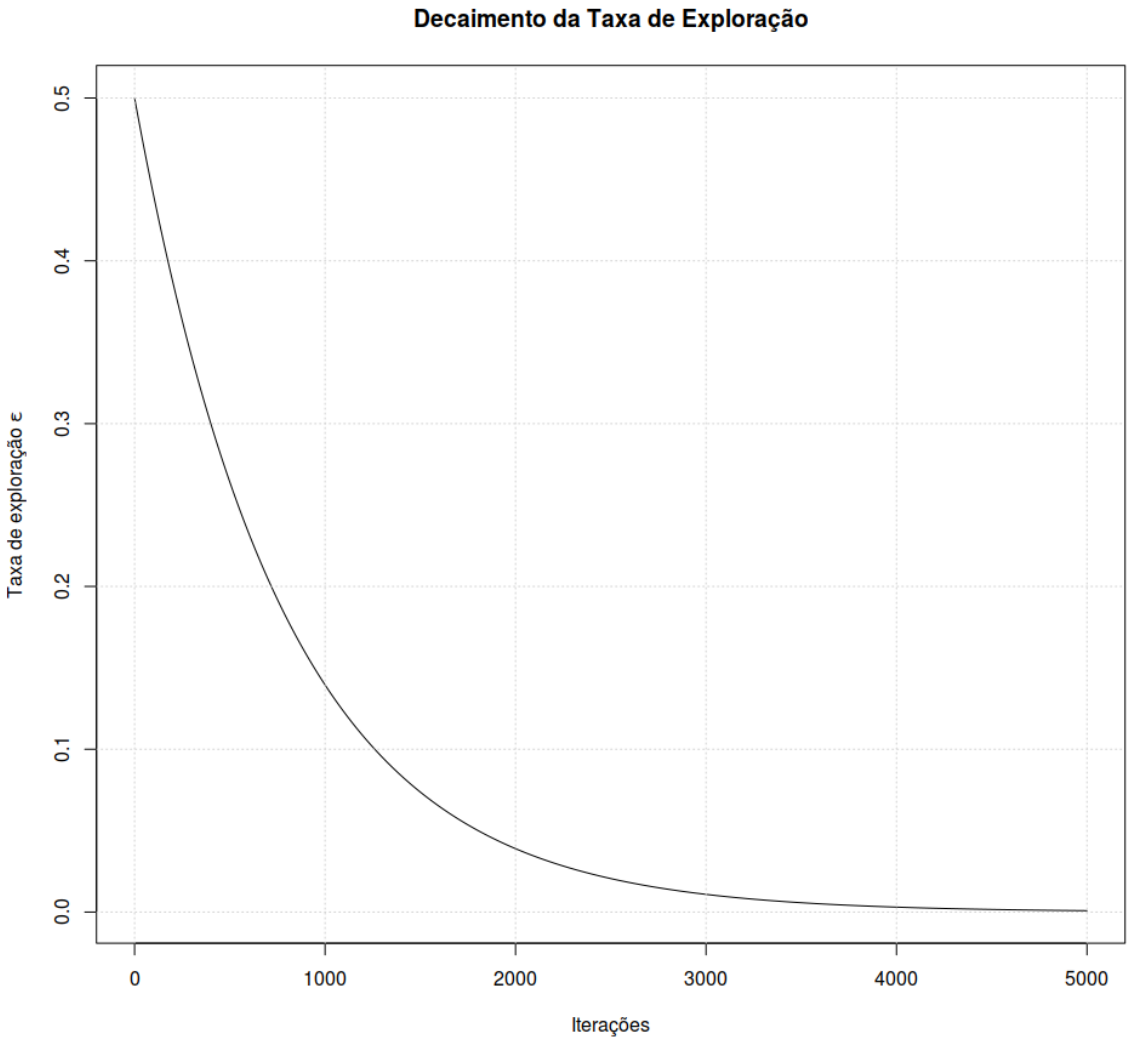

Figura 4.2: Taxa de exploração ϵ_t ao longo do treinamento.

operar, isto é, `NOP`. No instante atual, observa-se que o preço de fechamento R$11,4 está mais próximo do preço mínimo R$11,26 e portanto seu valor na tupla é `MIN`.

Nesse estado o agente deve escolher uma ação para executar. Ele então procura na sua tabela de ação-valor na entrada correspondente ao estado `(NPOS,NOP,MAX,MIN)`, a ação dentre as ações possíveis nesse estado $\{$`BUY`, `SELL`, `STILL`$\}$ que apresenta o maior valor esperado de recompensa acumulada. Supondo que essa seja a ação `BUY`, o agente então executa uma ordem de compra no valor atual da ação considerando nesse caso o preço de fechamento R$11,24. Essa ação corresponde a entrar em uma posição comprada, isto é, executar a ação do agente `ENTER_LONG` na qual o valor da posição comprada é R$11,24.

Executada essa ação e avançando para o instante seguinte no tempo `2015-01-02 11:15:00` tem-se novos valores de preços da ação. Com o preço

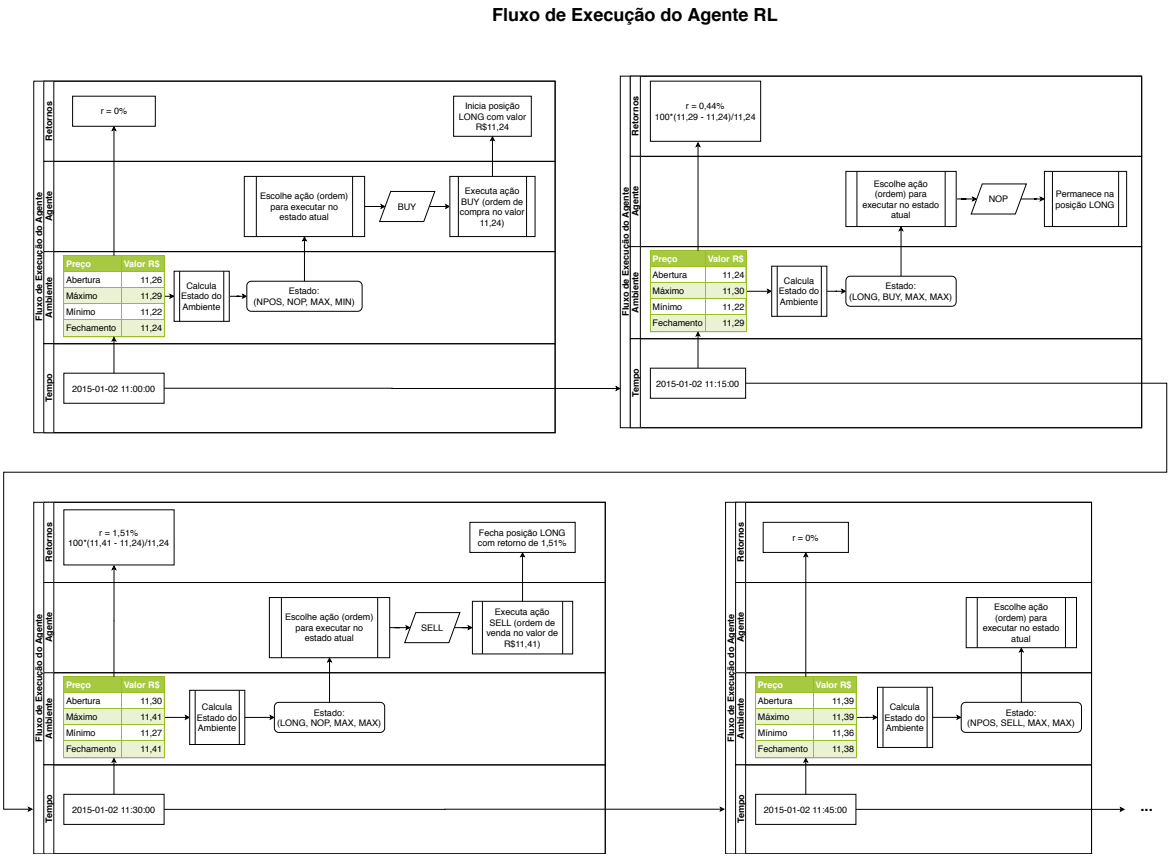

Figura 4.3: Exemplo de execução do agente.

de fechamento valendo agora R$11,29 o retorno nesse instante em relação ao valor da posição comprada R$11,24 é igual a 0,44%. O estado nesse instante corresponde a tupla `(LONG,BUY,MAX,MAX)`. O primeiro termo da tupla indica que agora o agente está em uma posição comprada `LONG` e que a ação executada no instante anterior decorreu de uma ordem de compra `BUY`. No estado anterior o preço de fechamento estava próximo do preço máximo `MAX` assim como no instante atual.

Nesse estado o agente novamente procura na sua tabela de estado-ação a entrada correspondente ao estado `(LONG,BUY,MAX,MAX)` onde escolhe a ação não operar `NOP`. Sendo assim, o agente permanece nesse estado na posição comprada o que corresponde a ação do agente `STAY_LONG`.

Passando para o próximo instante de tempo `2015-01-02 11:30:00` o retorno em relação a posição comprada corresponde agora ao valor de 1,51% uma vez que o preço de fechamento agora vale R$11,41. O estado agora é a tupla `(LONG,NOP,MAX,MAX)` quando então o agente escolhe na sua tabela de estado-ação executar a ação `SELL`. Essa ação determina ao agente fechar a posição comprada `LONG` vendendo pelo valor atual de R$11,41 todo volume comprado no inicio da posição e auferindo respectivo retorno de 1,51%.

Executada a ação do agente `EXIT_LONG` e avançado para o próximo instante de tempo `2015-01-02 11:45:00` o agente agora está não posicionado e o estado atual é a tupla `(NOP,SELL,MAX,MAX)`.

E assim prossegue o agente nos instantes de tempo seguintes, escolhendo e executando as respectivas ações para cada estado corresponde a cada instante de tempo.

4.6 Propriedades do Agente

A modelagem proposta apresenta propriedades importantes do ponto de vista do aprendizado por reforço e da aplicação como um agente de negociação de ações.

A primeira propriedade diz respeito a *convergência* da política aprendida pelo agente. Diferentemente dos algoritmos de aprendizado por reforço que empregam uma função aproximadora (e.g. *Deep Reinforcement Learning*), a modelagem utilizando o algoritmo SARSA com um espaço de estados e ações discretos e finitos converge com probabilidade 1 para a *política ótima* assim como a sua respectiva função de ação-valor [Sutton & Barto, 2018]. Isso é possível graças também ao decaimento da taxa de exploração ϵ_t ao longo do treinamento, razão pela qual foi escolhida essa estratégia de exploração.

Uma das razões para modelagem do espaço de estados em apenas 4 variáveis categóricas é evitar que o agente fique suscetível aos movimentos muitas vezes aleatórios do mercado. Embora essa modelagem seja bastante simplificada acerda das condições do mercado, uma modelagem mais complexa exigiria a utilização de funções aproximadoras o que também implica a utilização de modelos de aprendizado por reforço profundo. Como salientado anteriormente, tal tipo de modelagem é mais suscetível a *overfiting* o que é difícil de tratar em aplicações onde a aprendizado ocorre de forma online. Em termos do dilema *viés e variância* que é comum em aprendizado de máquina, a modelagem discreta e tabular favorece mais o viés enquanto a modelagem com função aproximadora favorece a variância. Em uma aplicação em que os dados apresentam muitas vezes comportamento de processos de passeio aleatório, tal como em dados de séries de preços de ações, convém utilizar uma modelagem menos complexa e menos suscetível a variância dos dados. Dessa forma, a modelagem discreta e tabular mostra-se mais robusta e embora possa também incorrer em *underfiting*.

Uma consequência dessa propriedade é que o Agente RL gera ao final do treinamento uma política determinística. Essa vantagem do Agente RL tabular facilita sobremaneira a análise dos seus resultados comparados a agentes baseados em aprendizado supervisionado que gerar resultados probabilísticos.

Assim, cada vez que o Agente RL é treinado em um mesmo conjunto de treinamento

ele também produz o mesmo resultado no respectivo conjunto de testes diferentemente de alguns agentes baseados em aprendizado supervisionado (e.g. redes neurais) os quais podem produzir resultados diferentes cada vez que treinados e testados em um mesmo conjunto de treinamento e teste.

Outra propriedade importante do algoritmo SARSA é que ele é do tipo *on-policy* o que significa que a política que o agente aprende é a mesma que ele utiliza para gerar seu comportamento. O efeito disso, conforme já observou Corazza & Sangalli [2015], é que o algoritmo SARSA é mais sensível a nova informação do que o algoritmo *Q-Learning* que é do tipo *off-policy* e também sua performance *on-line* é melhor comparado a outros algoritmos que empregam diferença temporal [Sutton & Barto, 2018]. A vantagem disso para um agente de negociação de ações é que ele pode mudar de estratégia de modo mais eficiente a cada vez que as condições do mercado se alteram permitindo a ele aproveitar melhor as variações de tendência de uma ação para obter ganhos e evitar perdas significativas.

Outras propriedades importantes decorrentes da modelagem proposta são o baixo computacional comparado aos agentes baseados em *deep reinforcement learning* que dependem de placas gráficas aceleradoras e a interpretabilidade das estratégias a partir dos valores das ações para cada estado na tabela de estado-ação.

Capítulo 5

Metodologia

A metodologia definida a seguir (ver Figura 5.1) pretende, em linhas gerais, testar as propriedades dinâmicas do Agente RL em mudar seu comportamento a partir de mudanças de tendência no ambiente e também constatar, avaliar e comparar o comportamento do Agente RL com um agente baseado em aprendizado supervisionado no contexto de instabilidade do mercado de ações.

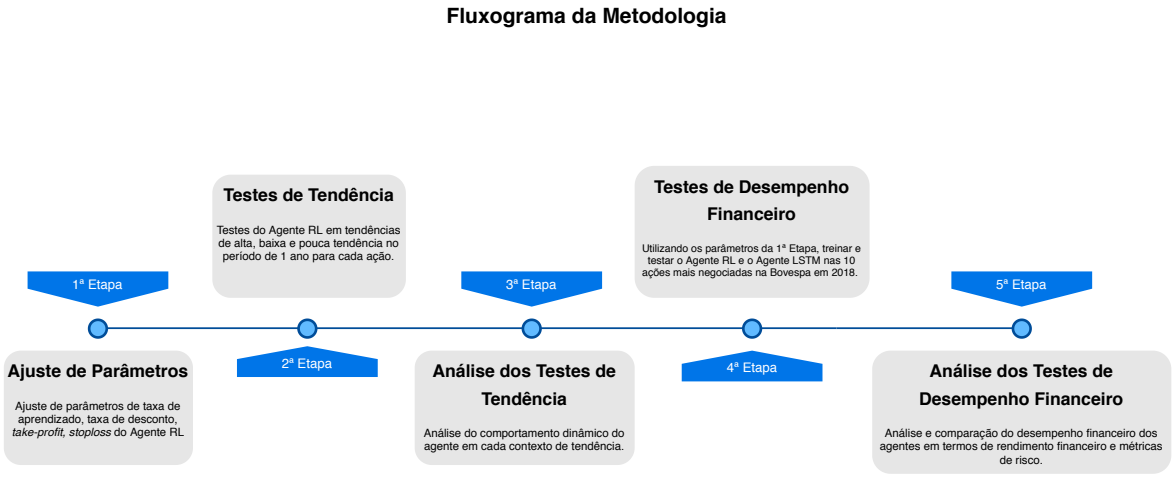

Figura 5.1: Etapas da metodologia do trabalho

Para tanto, inicialmente ajustou-se os parâmetros do Agente RL utilizando-se os dados históricos do ativo BOVA11, que representa uma variação próximo do Índice Bovespa, do ano de 2010 e uma vez obtidos os melhores parâmetros utilizou-se esses valores ao longo de toda a metodologia. Essa medida também ajuda a evitar o chamado *look-ahead bias* uma vez que os parâmetros do agente foram ajustados em dados anteriores à utilização do agente. Em seguida, escolheu-se 6 ações para testar o Agente RL em contextos diferentes de tendências de preços. Analisados os resultados nesses testes de tendência, o próximo passo foi aplicar

o Agente RL e o agente baseado em aprendizado supervisionado em um conjunto de 10 (Veja Tabela 5.5) ações do ano de 2018 para observar e comparar o desempenho de cada abordagem no contexto de elevada instabilidade no mercado naquele ano devido as eleições nacionais.

5.1 Dados Utilizados

Os dados utilizados nesse trabalho consistem nas séries históricas de preços e volumes de ações negociadas na Bolsa de Valores de São Paulo (B3 - Bolsa Brasil Balcão) no período de 1 de janeiro de 2009 até 31 de dezembro de 2018 com periodicidade de 15 minutos. Cada entrada está no formato *OHLCV*, isto é:

- *Open*: preço de abertura na respectiva entrada.
- *High*: preço máximo negociado.
- *Low*: preço mínimo negociado.
- *Close*: preço de fechamento.
- *Volume*: volume financeiro negociado no período.

Além disso cada entrada contém o campo *Datetime* designando a data e horário da entrada e o campo *Indice* que designa o preço de fechamento do ativo BOVA11 no respectivo período. A Tabela 5.1 mostra exemplos de entradas contidas nos dados utilizados nesse trabalho.

Datetime	Open	High	Low	Close	Volume	Indice(BOVA11)
2015-01-02 10:00:00	11,2	11,27	11,13	11,21	154301	48,13
2015-01-02 10:15:00	11,21	11,25	11,19	11,23	154587	48,2
2015-01-02 10:30:00	11,23	11,33	11,22	11,33	483135	48,24

Tabela 5.1: Exemplo de formato de dados utilizados.

Para cada ação do conjunto de dados, considerou-se nos experimentos o período de 1 ano em que foram utilizadas como treinamento e validação (*in-sample*) as entradas referentes ao primeiro semestre e como teste (*out-of-sample*) as respectivas entradas do segundo semestre. A escolha pelo período de 1 ano permite comparar diretamente o rendimento obtido por cada agente com outras opções de investimento menos arriscadas como CDI, por exemplo.

5.2 Agente LSTM

Para comparar o desempenho do Agente RL com um agente baseado em aprendizado supervisionado, criou-se uma rede neural LSTM com a arquitetura descrita na Figura 5.2. O agente que utiliza essa rede será referido desse ponto em diante como *Agente LSTM*. A escolha pela rede neural LSTM se deve aos vários trabalhos que utilizam essa rede no contexto financeiro o que permite, portanto, comparar o Agente RL com uma abordagem já várias vezes testada. Particularmente, essa rede é uma evolução do trabalho apresentado em Nelson et al. [2017].

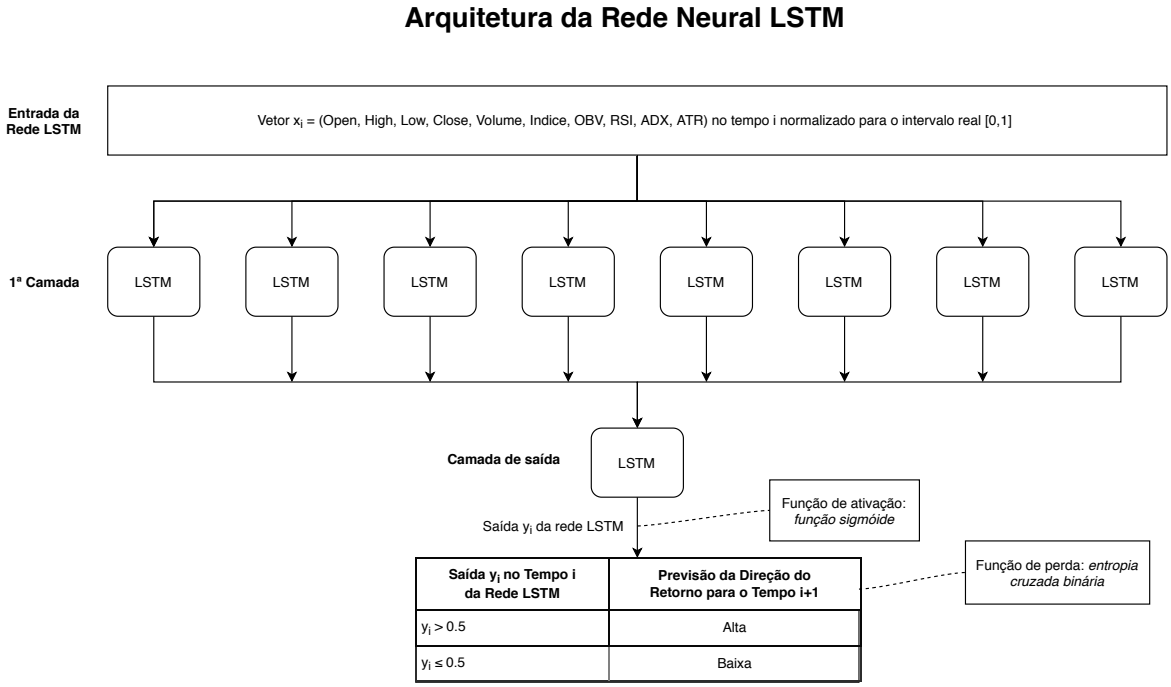

Figura 5.2: Arquitetura da Rede LSTM

A primeira camada da rede neural LSTM compõe-se de 8 unidades LSTM e a camada de saída contém 1 unidade LSTM. A camada de saída utiliza como função de ativação a função sigmóide e como função de perda a função entropia cruzada binária.

Essa rede toma como entrada um vetor de 10 atributos (*features*) normalizados para valores no intervalo $[0, 1]$ segundo a equação (Equação 5.1)

$$\bar{z}_i = \frac{z_i - min(z)}{max(z) - min(z)} \tag{5.1}$$

onde para um atributo qualquer z a instância z_i desse atributo assume o valor normalizado $\bar{z}_i \in [0, 1]$. Os termos $max(z)$ e $min(z)$ denotam respectivamente o maior e o menor valor do atributo z no conjunto de treinamento.

Para uma entrada qualquer x_i de um conjunto de dados de tamanho N onde $i \in 1, 2, 3, 4, ..., N$, utilizou-se como atributos de entrada:

1. Open: preço de abertura
2. High: preço de máximo
3. Low: preço de mínimo
4. Close: preço de fechamento
5. Volume: volume financeiro negociado
6. Indice: valor do ativo BOVA11
7. OBV: indicador técnico OBV(*On-Balance Volume*)
8. RSI: indicador técnivo RSI (*Relative Strength Index*)
9. ADX: indicador técnico ADX (*Average Directional Movement Index*)
10. ATR: indicador técnico ATR (*Average True Range*)

A classe y_i referente a cada entrada x_i assume um dos valores inteiros $y_i \in \{0, 1\}$ dados pela fórmula na Equação 5.2

$$y_i = \begin{cases} 1, & Close_i > Close_{i+1} \\ 0, & c.c \end{cases} \tag{5.2}$$

em que o termo $Close_i$ e $Close_{i+1}$ referem-se aos preços de fechamento nas entradas i e $i + 1$, respectivamente. Cada entrada tem como classe, portanto, a direção da variação de preços de fechamento no instante seguinte. Assim, dada um entrada qualquer x_i no instante i a rede LSTM deverá prever a direção da variação dos preços de fechamento no instante $i + 1$.

Associada à rede neural LSTM foi utilizada uma estratégia de operação descrita na Figura 5.3.

Se o Agente LSTM estiver não posicionado (`NPOS`) no instante de tempo i e a rede LSTM prever “Alta”, então o agente deverá executar uma ordem de compra da ação pelo preço de fechamento no tempo i. Se estiver em posição comprada (`LONG`) e a rede neural prever “Alta” então o agente deverá permanecer na posição comprada. O Agente LSTM só sairá da posição comprada caso a rede neural LSTM preveja "Baixa"quando então o agente deverá vender a ação comprada anteriormente pelo preço atual. Essa operação faz o agente deixar a posição comprada e transitar para o estado não posicionado (`NPOS`).

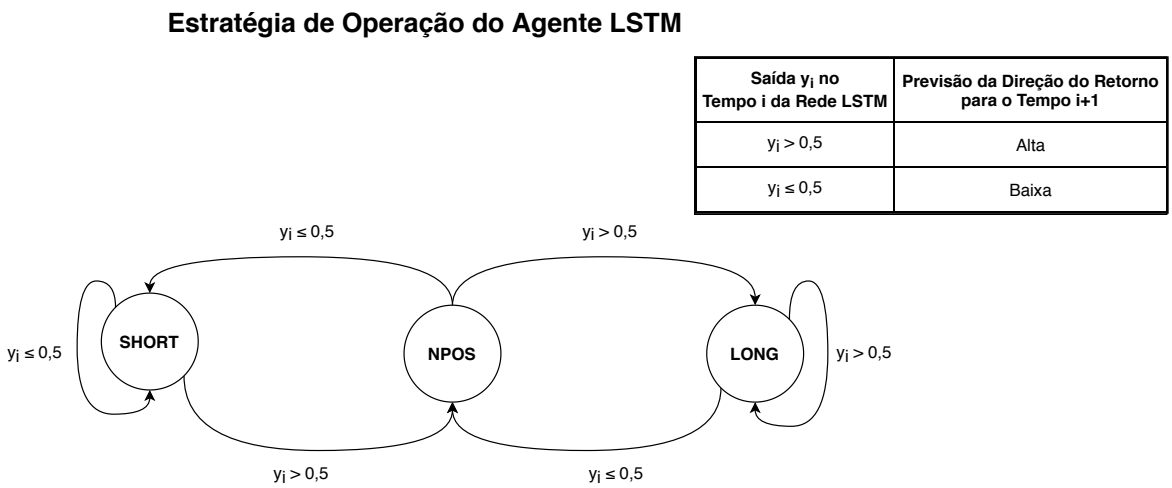

Figura 5.3: Máquina de estados da estratégia de operação do Agente LSTM.

Parâmetro	Valor	Intervalo de Valores Testados	Variação
Taxa inicial de exploração ϵ_0	$0,5$	$1,0$ até $0,1$	$0,1$
Taxa de aprendizado α	$2 \cdot 10^{-5}$	$0,2$ até $2 \cdot 10^{-5}$	$0,1$
Fator de desconto γ	$0,97$	$1,00$ até $0,10$	$0,01$
Número de episódios T	5000	1000 até 10000	1000

Tabela 5.2: Parâmetros do Agente RL implementado

De modo semelhante ocorre se o agente estiver não posicionado em um instante qualquer i e a rede neural prever "Baixa". O Agente LSTM deverá vender a ação pelo preço atual de fechamento no tempo i indo para a posição vendida (`SHORT`). Nesse estado, se a rede neural prever novamente "Baixa" o agente deverá permanecer nessa posição. Porém, caso a rede preveja "Alta" o agente deverá sair da posição vendida recomprando a ação pelo preço de fechamento atual transitando no instante seguinte para o estado não posicionado.

5.3 Parâmetros

Para ajustar os parâmetros do Agente RL utilizou-se os dados do ativo BOVA11 do ano de 2010 (Figura 5.4). Treinando (70% primeiras entradas) e validando (30% últimas entradas) no primeiro semestre (dados *in-sample*) e testando no segundo semestre (dados *out-of-sample*) escolheu-se os parâmetros do Agente RL que obteve o melhor rendimento financeiro total na validação. Esses parâmetros foram usados em todos os experimentos propostos nessa metodologia. Os valores de parâmetros alcançados estão na Tabela 5.2.

A convergência no treino ao longo das 5000 iterações também pode ser observada por meio da curva do gráfico da Figura 5.5.

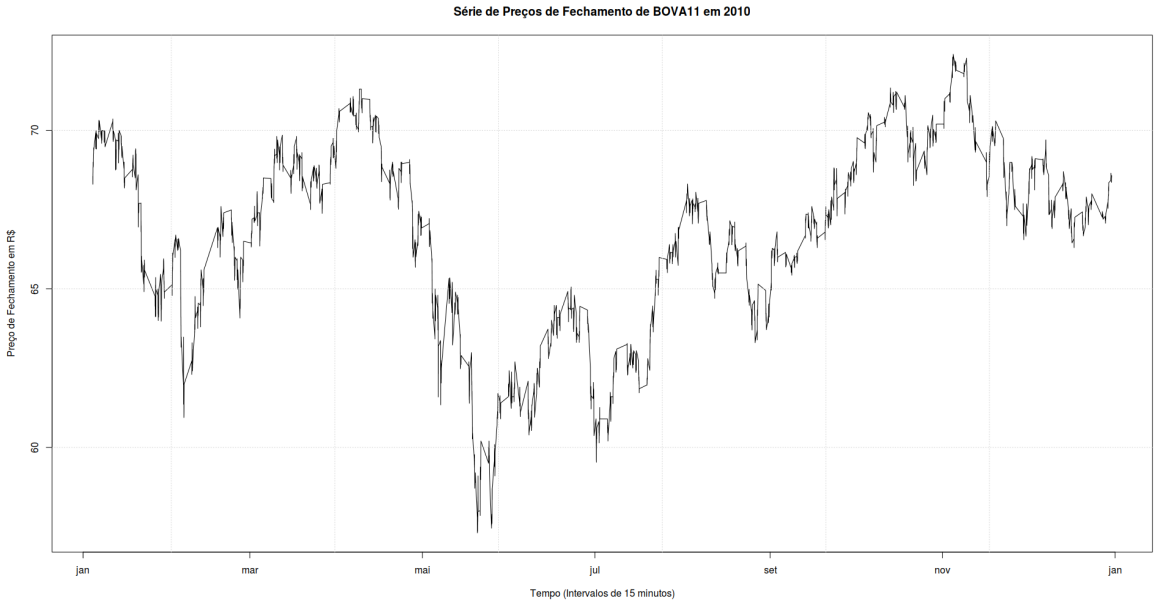

Figura 5.4: Preços de fechamento do ativo BOVA11 no ano de 2010.

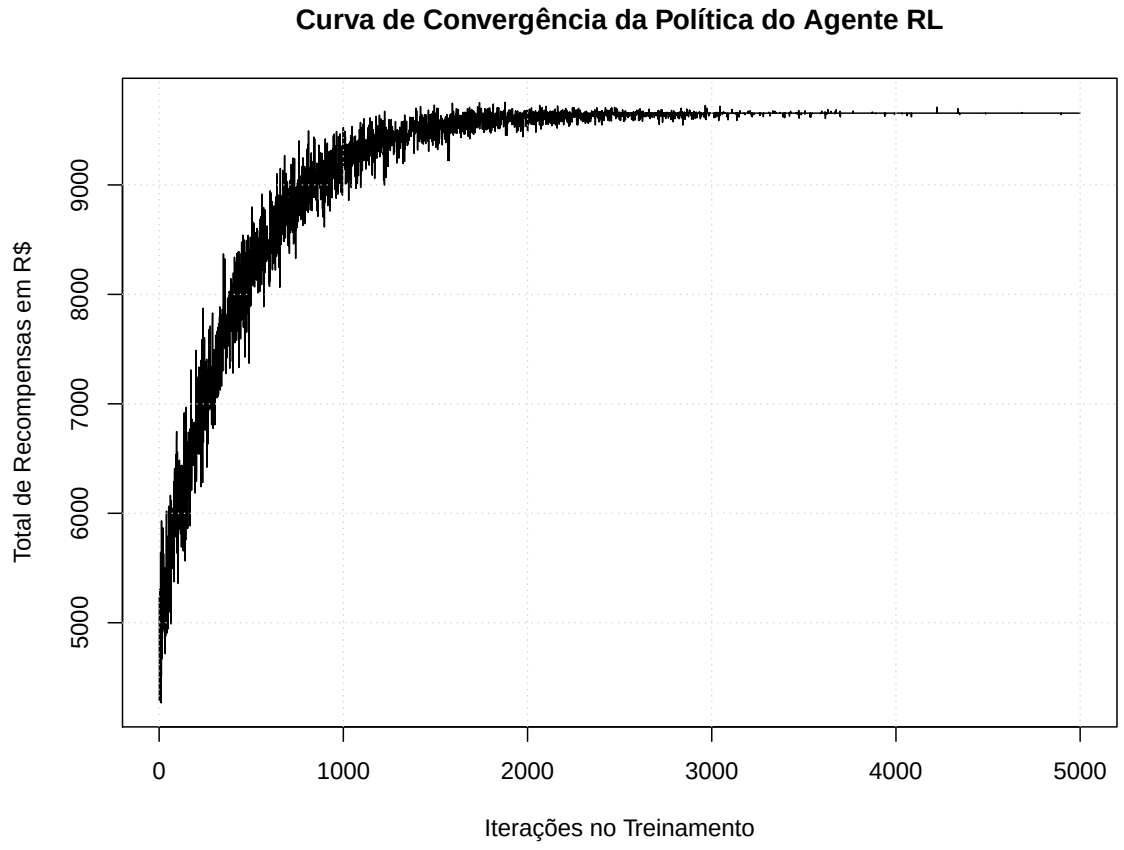

Figura 5.5: Curva de convergência do Agente RL no treinamento.

Ressalte-se que uma vez treinado o Agente RL em uma ação, a execução do agente nos dados de teste da respectiva ação ocorre com a taxa de exploração ajustada para 0 de modo que o agente de aprendizado por reforço não escolha nenhuma ação aleatoriamente (seguindo política de exploração ϵ – *greedy*) durante os testes.

Com relação ao Agente LSTM, utilizou-se os parâmetros da Tabela 5.3.

Parâmetro	**Valor**
Épocas	5000
Tamanho do lote	2000
Validação	30% últimas entradas do treinamento
Otimizador	Adamax

Tabela 5.3: Parâmetros do Agente LSTM implementado

Da mesma forma que o Agente RL, treinou-se o Agente LSTM (dados *in-sample*) nas primeiras 70% primeiras entradas do primeiro semestre de cada ação e validou-se nas últimas 30% últimas entradas do primeiro semestre. Para cada ação em que o Agente LSTM foi treinado adotou-se a política de interrupção prematura (*Early Stopping*) no treinamento tão logo o crescimento da métrica de acurácia no conjunto de validação estabilizasse por até 5 épocas.

Para tanto o Agente RL quanto o Agente LSTM, cada ordem de compra ou venda é feita a mercado considerando o preço de fechamento naquele instante e assumindo liquidez suficiente para o volume de 1 lote de ações o que equivale geralmente a 100 ações na B3.

5.4 Testes de Tendência

Com o **objetivo** de testar e analisar a capacidade do Agente RL em modificar dinamicamente sua estratégia de negociação em diferentes situações de tendência (alta, baixa e pouca tendência) em um período de 1 ano, selecionou-se as ações da Tabela 5.4. Tal escolha baseou-se tão somente no aspecto gráfico das séries de preços de fechamento das respectivas ações (Figura 5.6).

A **hipótese** subjacente a esse teste de tendência é de que o Agente RL é capaz de adaptar sua estratégia de negociação à medida que as condições de tendência da ação se modificam tal como um agente qualquer de aprendizado por reforço deve modificar dinamicamente seu comportamento a partir das mudanças no ambiente.

Os **resultados esperados** para nesses testes de tendências são:

- Nas ações que apresentam tendência de alta, espera-se que o agente execute na maior parte do tempo posições compradas (*Long*).

Ação	Tendência	Ano
ABEV3	Lateral	2014
BBDC3	Lateral	2011
CIEL3	Alta	2011
NATU3	Alta	2012
USIM5	Baixa	2014
TIMP3	Baixa	2015

Tabela 5.4: Ações classificadas por tendência anual

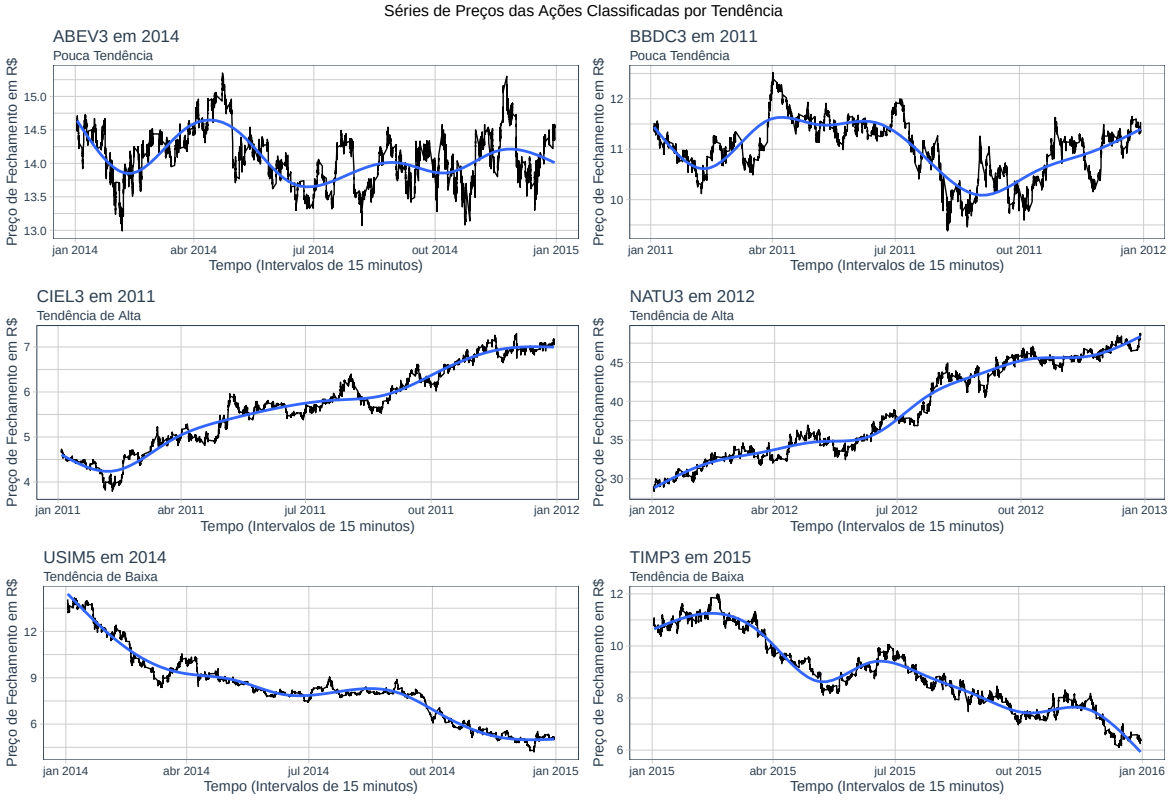

Figura 5.6: Ações classificadas por tendência.

- Nas ações que apresentam tendência de baixa, espera-se que o agente execute na maior parte do tempo posições vendidas (*Short*).

- Nas ações de pouca tendência, espera-se que o agente execute tanto posições compradas quanto vendidas variando conforme as condições momentâneas de tendência do preço da ação.

Para auxiliar a observação dos resultados nesse teste foi utilizado o gráfico de barras de retornos financeiros do agente ao longo da sua execução (Figura 5.7). As barras desse gráfico indicam a dimensão dos retornos financeiros em porcentagem para cada posição assumida

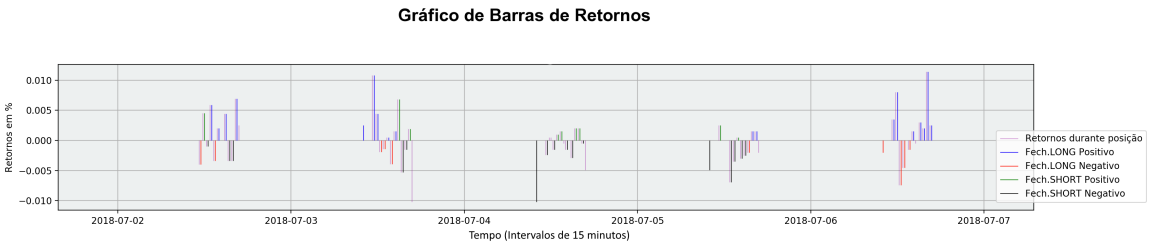

Figura 5.7: Exemplo de gráfico de barras de retornos

pelo agente. As barras roxas designam os retornos em cada instante em que o agente está posicionado e ainda não saiu de uma posição comprada (*Long*) ou vendida (*Short*).

Se a saída de uma posição comprada resultou em retorno positivo a sua cor será azul e sua altura indica a dimensão em porcentagem do retorno. Se porém, a saída da posição comprada resultou em retorno financeiro não positivo a barra será vermelha.

Para posições vendidas, as barras verdes indicam saídas da posição *short* com retorno positivo e as barras pretas indicam saídas da posição com retorno não positivo.

5.5 Testes de Desempenho Financeiro

O **objetivo** desse teste é analisar o comportamento do Agente RL em termos de rendimento financeiro, métricas de risco e taxas de acertos no contexto de alta volatilidade e instabilidade do mercado de ações no ano de 2018 em virtude das eleições nacionais naquele ano. Pretende-se também compará-lo com o desempenho do agente aprendizado supervisionado, o Agente LSTM. Para tanto, selecionou-se as 10 ações mais negociadas na B3 no ano de 2018[1] (Figura 5.8).

A **hipótese** subjacente a esses testes é de que uma vez que o Agente RL pode se adaptar as condições de instabilidade do mercado então ele pode apresentar desempenho superior ao Agente LSTM nos momentos de variações significativas de tendência em termos de rendimento financeiro e métricas de risco. Isso é o que se espera dos resultados nesse teste.

Para observar os resultados nesse teste serão consideradas as variáveis de rendimento financeiro total, as métricas máximo *drawdown*, *Sharpe Ratio* e *Sortino Ratio*. Serão ainda comparados os ganhos médios e perdas médias com relação aos rendimentos nos fechamentos de posições com ganhos e com perdas, respectivamente, nos testes de cada agente.

O rendimento financeiro ao longo da aplicação dos agentes em cada ação assim como o máximo *drawdown* podem ser observados no gráfico de evolução do capital (Figura 5.9)

[1] Veja em url: https://www.moneytimes.com.br/fast/as-20-acoes-mais-negociadas-em-2018/

Ações	Companhia	Setor
ABEV3	Ambev	Bebidas
B3SA3	B3 - Bolsa Brasil Balcão	Financeiro
BBAS3	Banco do Brasil	Financeiro
BBDC4	Banco Bradesco	Financeiro
ITSA4	Itausa Holding	Financeiro
ITUB4	Banco Itaú	Financeiro
PETR3	Petrobrás	Exploração de petróleo
PETR4	Petrobrás	Exploração de petróleo
SUZB3	Suzano	Papel e celulose
VALE3	Companhia Vale	Mineração

Tabela 5.5: Ações selecionadas para o teste de desempenho.

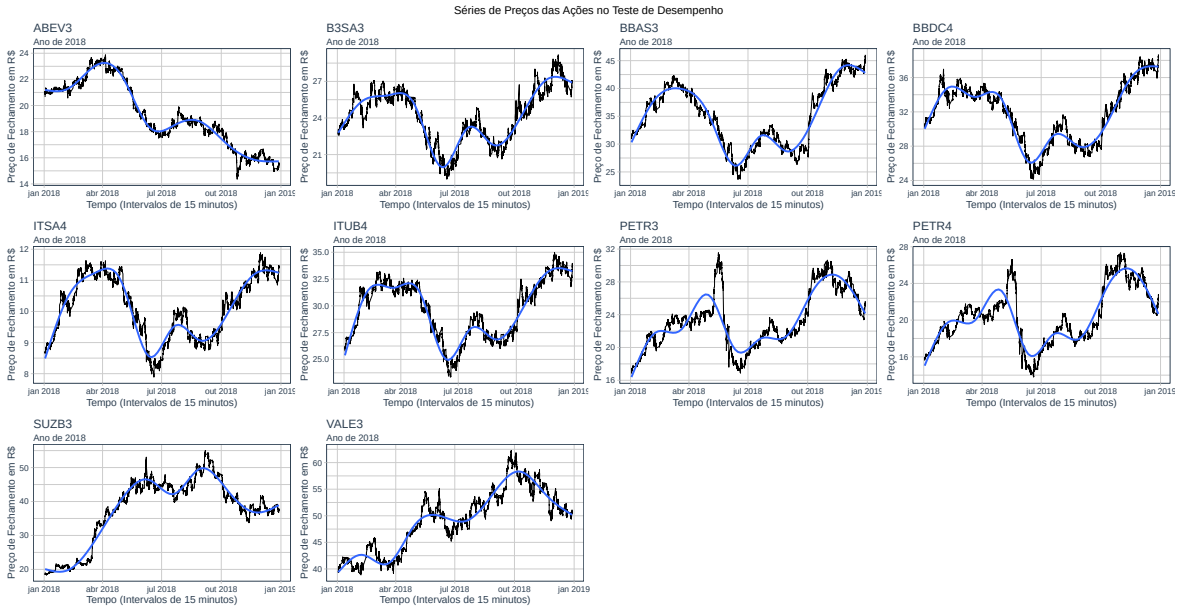

Figura 5.8: Ações utilizadas nos testes de desempenho.

ao longo do teste.

Cada ponto da curva representa o rendimento financeiro em porcentagem acumulado pelo agente no respectivo instante no tempo. Esse rendimento é relativo ao primeiro negócio executado pelo agente e assumindo a negociação de um volume fixo de 1 lote de ações que equivale a 100 ações.

O trecho destacado em vermelho na curva representa a máxima perda acumulada pelo agente e permite determinar o início e o fim do máximo *drawdown*.

Outra métrica que será observada nesse teste será a porcentagem de fechamentos de posições com resultado positivo. Essa métrica é semelhante a uma "taxa de acertos" de cada agente e permite comparar de modo mais próximo um agente de aprendizado supervisionado

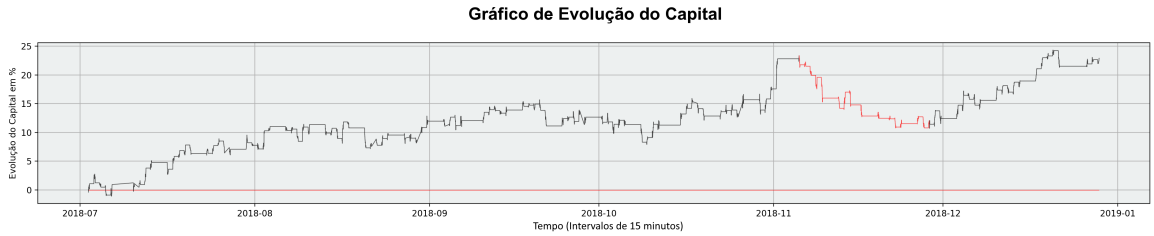

Figura 5.9: Exemplo de gráfico de evolução do capital com máximo *drawdown* destacado em vermelho.

(que procura otimizar taxa de acertos como acurácia, precisão, f-score) com um agente e aprendizado por reforço que procura otimizar o acúmulo de ganhos financeiros.

Serão ainda observadas e comparadas as taxas de acertos nos fechamentos de posições compradas (LONG) e nos fechamentos de posições vendidas (SHORT) de cada agente em cada ação testada.

Capítulo 6

Experimentos: Resultados & Análise

O Agente RL foi implementado em linguagem C++ enquanto o Agente LSTM foi implementado utilizando a linguagem Python 3.6 com as bibliotecas *Tensorflow 2.1.0* e *Keras*. Ambos foram treinados e testados em um computador PC com processador AMD FX-8320E de 64 bits, 8 núcleos de processamento e 23,4 gigabytes de memória principal. O Agente LSTM foi treinado e testado aproveitando os recursos da placa gráfica aceleradora (GPU) NVidia GTX-1080 Ti.

Para gerar os gráficos apresentados foram utilizadas as bibliotecas *ggplot2* do software estatístico R e *matplotlib* da linguagem de programação Python. Para calcular as métricas de máximo *drawdown*, *Sharpe Ratio*, *Sortino Ratio* foi utilizada a biblioteca *PerformanceAnalytics* do software estatístico R.

6.1 Testes em Tendência

6.1.1 Testes para ações de pouca tendência

Ação	Rendimento Total	Máximo Drawdown	Buy-and-Hold (Rendimento/Máximo Drawdown)	BOVA11 (Rendimento/Máximo Drawdown)
ABEV3	31,42%	-4,13%	7,37% / -13,05%	-4,68% / -26,08%
BBDC3	30,25%	-9,98%	-3,13% / -21,52%	-9,42% / -24,84%

Tabela 6.1: Resultados de testes para ações com pouca tendência.

Nos testes em tendência (Tabela 6.1) para as ações ABEV3 no ano de 2014 e BBDC3 no ano de 2011 o Agente RL apresentou desempenho superior a estratégia *Buy-and-Hold* e o BOVA11 tanto em rendimento financeiro quanto em máximo *drawdown*. A opção pela estratégia buy-and-hold se deve ao fato de que é bastante utilizada como benckmark em

trabalhos de aplicações de técnicas de inteligência artificial em finanças embora tal estratégia seja mais apropriada no contexto de negociação em portfólios de ações.

No teste na ação ABEV3 (Figura 6.1) o agente utilizou na maior parte do segundo semestre as operações compradas (865 posições compradas) ao mesmo tempo que fez uso de poucas operações vendidas (177 posições vendidas) gerando o rendimento de 31.42%. Por isso, não se percebe uma considerável mudança de estratégia pelo agente dado o grande número de operação compradas em relação as operações vendidas. Isso se explica ainda pelas várias reversões de tendência apresentadas pela ação ao longo do semestre de 2014 fazendo com que os poucos acertos na vezes que o agente tentou executar posições vendidas fossem superados em magnitude pelos vários acertos quando o agente executou posições compradas o que fez o agente preferir esse tipo de posição ao longo do teste.

Isso também é decorrência do baixo valor utilizado como taxa de aprendizado $\alpha = 2 \cdot 10^{-5}$ e também a preferência pelos retornos no longo prazo (fator de desconto $\gamma = 0,97$) o que tornou o agente insensível às pequenas variações de tendência no curto prazo.

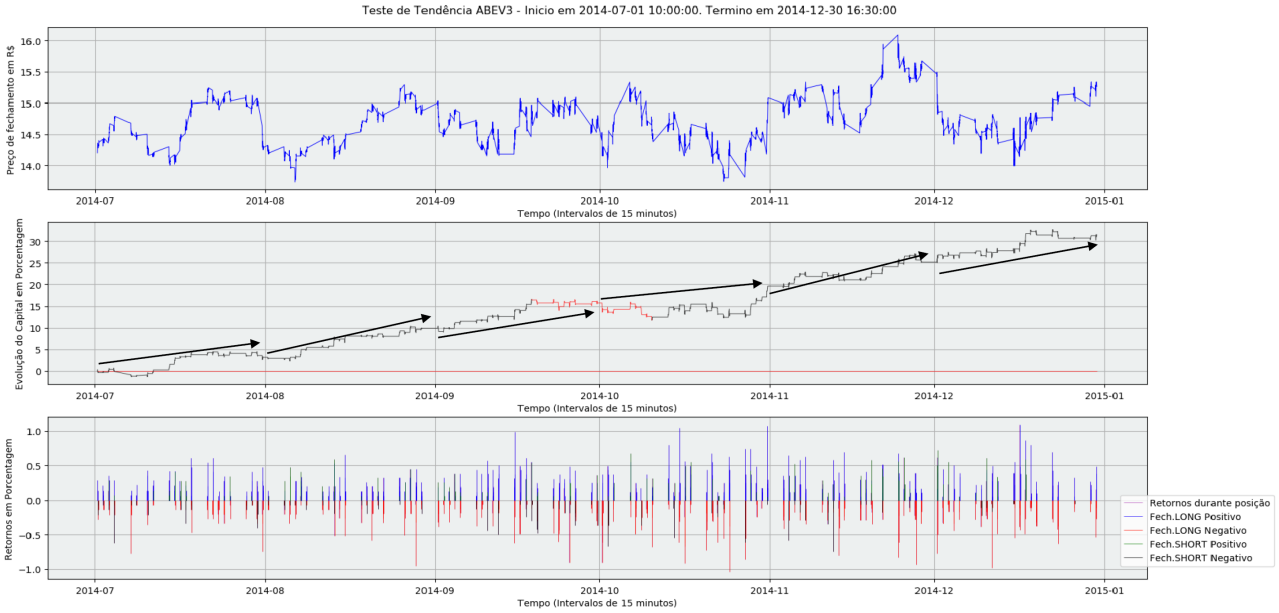

Figura 6.1: Teste em tendência na ação ABEV3 em 2014

Já no teste da ação BBDC3 (Figura 6.2) percebe-se que ao final do mês de agosto (primeiro trecho destacado em amarelo) e início do mês de setembro o agente inicia a perda do máximo *drawdown*. Nota-se que nesse período a ação apresenta uma alta considerável (segundo trecho destacado em amarelo) que ocasionou uma perda em torno de 3% na estratégia do agente que estava posicionado em *short*. A partir dessa perda o agente tenta mudar a estratégia operando comprado, mas a ação volta a cair o que gera outra perda em torno 3% fazendo o agente mudar novamente a estratégia para operar vendido voltando a lucrar com

as quedas da ação durante o mês de setembro. Observa-se ainda outra mudança dinâmica de estratégia logo ao final do mês quando a ação volta majoritariamente a subir e o agente começa a operar comprado na maior parte das vezes até o final do ano de 2011 gerando proporcionado um rendimento de 30,25%.

Nesse teste, portanto, observou-se que pequenas oscilações não foram bastante para mudar a estratégia do agente necessitando portanto de variações significativas para que ocorresse tal efeito.

Figura 6.2: Teste em tendência na ação BBDC3 em 2011

6.1.2 Testes para ações com tendência de alta

Ação	Rendimento Total	Máximo Drawdown	Buy-and-Hold (Rendimento/Máximo Drawdown)	BOVA11 (Rendimento/Máximo Drawdown)
CIEL3	24,08%	-6,63%	26,05% / -13,61%	-9,42% / -24,84%
NATU3	24,61%	-9,02%	28,16% / -9,83%	12,15% / -13,04%

Tabela 6.2: Resultados de testes para ações com tendência de alta.

Nas ações com tendência de alta (Tabela 6.2) o Agente RL apresentou desempenho superior aos *baselines* apenas em máximo *drawdown* embora o rendimento total financeiro nas duas ações tenha ficado próximo dos rendimentos da estratégia *Buy-and-Hold* e do BOVA11.

No teste na ação CIEL3 (Figura 6.3) observa-se a partir do gráfico de retornos que o agente modificou sua estratégia 4 vezes (trechos destacados em amarelo) operando conforme os resultados esperados (Vide resultados esperados na página 44) para as condições locais tendência.

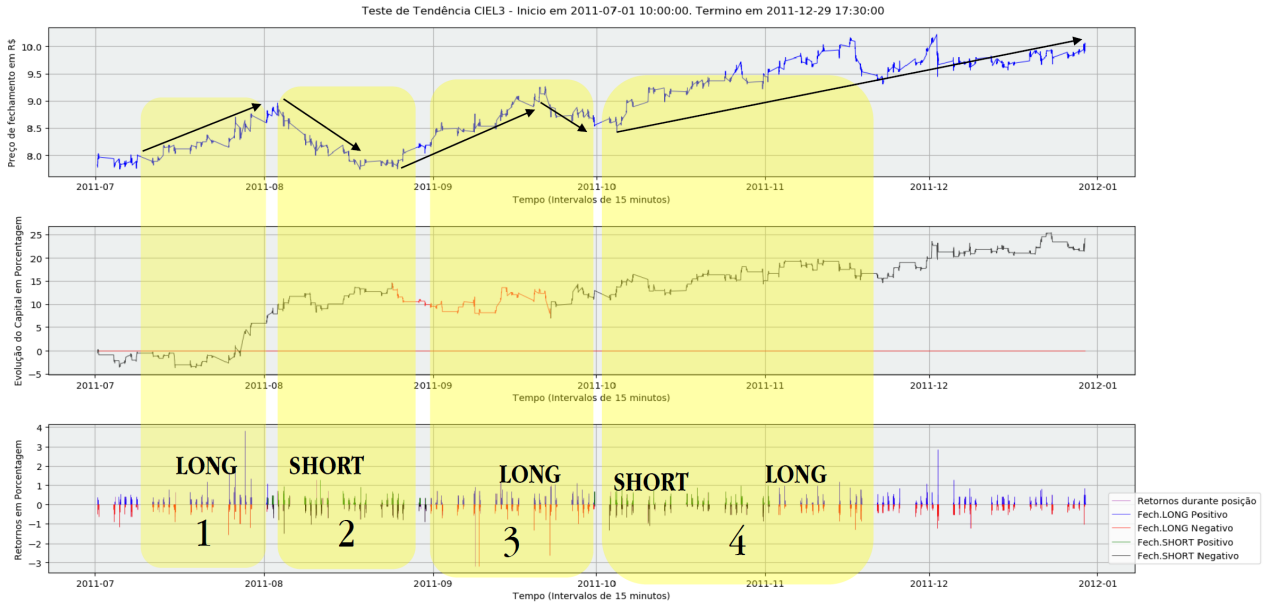

Figura 6.3: Teste em tendência na ação CIEL3 em 2011

No mês de julho a ação apresentou uma tendência de alta na segunda metade do mês o que fez o agente operar posições compradas embora as perdas nesse mês tenham sido maiores que os ganhos gerando um rendimento financeiro negativo na maior parte do período. Porém, foi somente no final do mês com o fortalecimento da tendência de alta que o agente obteve maiores ganhos que geraram um montante acumulado do capital em torno de 6% fortalecendo a estratégia de operar comprado nesse período.

Em seguida, no início do mês de agosto a ação apresentou uma queda na tendência que durou até quase o final do mês o que fez o agente, conforme esperado, a mudar sua estratégia e começar a operar com posições vendidas proporcionando um acréscimo no capital de aproximadamente 5% em relação ao acumulado no início do mês. Com o retorno à tendência de alta dos preços no final desse mês o agente mudou novamente sua estratégia para operar posições compradas iniciando o mês de setembro com essa estratégia que gerou mais perdas significativas do que ganhos razão pela ocorreu a maior parte do período do máximo *drawdown*.

Com a queda na tendência dos preços da ação no final do mês de setembro o Agente RL mudou sua estratégia para operar vendido e assim iniciou o mês de outubro. Porém, a ação voltou a subir no início daquele mês o que gerou um rendimento financeiro de em torno de 2,5% em relação ao inicio do mês.

A partir do mês de novembro as constantes mudanças de tendência começaram a diminuir e a ação apresentou uma tendência majoritariamente de alta fazendo o Agente RL operar comprado até o final de dezembro produzindo um capital acumulado de 24,08%.

No teste na ação NATU3 no segundo semestre do ano de 2012 (Figura 6.4) o agente novamente operou conforme o esperado a partir da tendência majoritariamente de alta da ação ao longo do período do teste. O agente executou 882 posições compradas ao lado 279 posições vendidas.

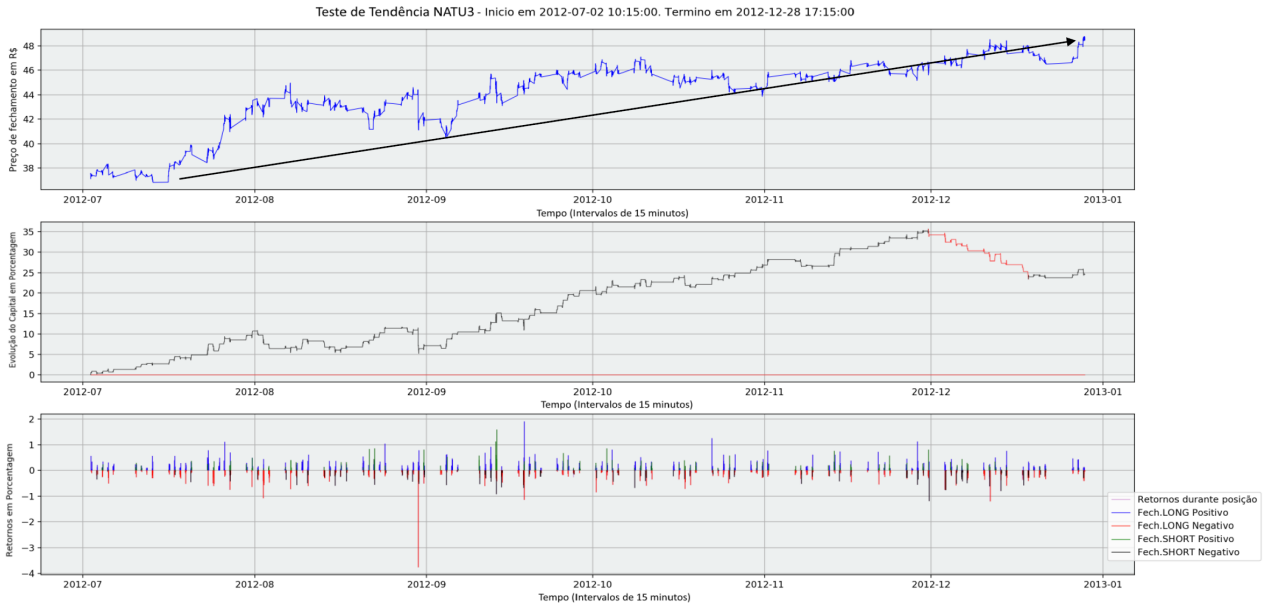

Figura 6.4: Teste em tendência na ação NATU3 em 2012

Nesse teste as mudanças de estratégias foram bem mais curtas como pode-se observar a partir gráfico de retornos. Uma dessas mudanças ocorreu no final do mês de agosto quando o agente estava operando comprado mas ação apresentou uma queda abrupta gerando a perda máxima de -3,76%. Isso fez o Agente RL mudar sua estratégia para operar vendido logo em seguida gerando um acréscimo de aproximadamente 0,6% na evolução capital.

Outro instante de destaque na mudança de estratégia ocorre no final do mês de novembro quando o agente alternava posições compradas e vendidas produzindo um rendimento total até então em torno de 35%. Essa estratégia passou a não mais gerar ganhos a partir do início do mês de dezembro o que ocasionou o máximo *drawdown*.

A partir das perdas acumuladas ao longo de quase todo o mês dezembro o Agente RL foi diminuindo o número de posições vendidas conforme se observa pela diminuição de barras verdes e pretas no gráfico de retornos e passou a operar majoritariamente comprado a partir da segunda quinzena desse mês estabilizando as perdas acumuladas e apresentando uma leve alta no capital acumulado nos últimos dias do mês aproveitando a alta da ação.

Novamente, observou-se como esperado nesses testes com ações com tendência de alta que o Agente RL foi capaz de mudar sua estratégia a partir de mudanças no comportamento de preços da ação mas executando majoritariamente posições compradas.

6.1.3 Testes para ações com tendência de baixa

Ação	Rendimento Total	Máximo Drawdown	Buy-and-Hold (Rendimento/Máximo Drawdown)	BOVA11 (Rendimento/Máximo Drawdown)
USIM5	29,21%	-8,76%	-33,11% / -53,64%	-4,68% / -26,08%
TIMP3	15,52%	-4,68%	-33,76% / -37,58%	-18,88% / -19,79%

Tabela 6.3: Resultados de testes para ações com tendência de baixa.

Nos testes com ações com tendência de baixa (Tabela 6.3) o Agente RL apresentou desempenho superior aos *baselines* *Buy-and-Hold* e BOVA11 tanto em relação ao rendimento financeiro quanto ao máximo *drawdown*. O agente também operou como esperado (Vide resultados esperados na página 44) para ações com tendência de baixa executando na maior parte do tempo posições vendidas.

No teste da ação USIM5 (Figura 6.5) o Agente RL operou quase que todo o segundo semestre de 2014 com posições vendidas. Foram 1176 posições vendidas ao lado de somente 72 posições compradas.

Figura 6.5: Teste em tendência na ação USIM5 em 2014

E como esperado, começou operando no início do mês julho com posições compradas (primeira parte do trecho destacado em amarelo) aproveitando a tendência de alta do preço da ação no início daquele mês o que proporcionou um ganho acumulado em torno de 14%. Porém, a tendência da ação começou a cair a partir da segunda quinzena de julho o que fez o Agente RL começar operar vendido (segunda parte do trecho destacado em amarelo) gerando um ganho nesse mês de aproximadamente 15%. Essa queda na tendência do preço

da ação se estabilizou em torno de R$8,00 e permaneceu assim até inicio do mês de setembro, razão pela qual o agente permaneceu operando vendido gerando pouco aumento no capital acumulado até esse período. Tal como se observou nos testes de ações com pouca tendência, as pequenas oscilações não foram suficientes para fazer o agente mudar sua estratégia tendo ele permanecido operando com posições *short*.

A partir da queda na tendência de preços da ação em meados do início de setembro e que durou até aproximadamente o final da primeira quinzena de dezembro o Agente RL apresentou o comportamento esperado para a referida condição de tendência: permaneceu executando posições *short* que geraram o rendimento total de 29,21% no final do período do teste.

Comportamento semelhante e no mesmo sentido dos resultados esperados para ações com tendência de baixa foi apresentado pelo agente no teste com a ação TIMP3 (Figura 6.6).

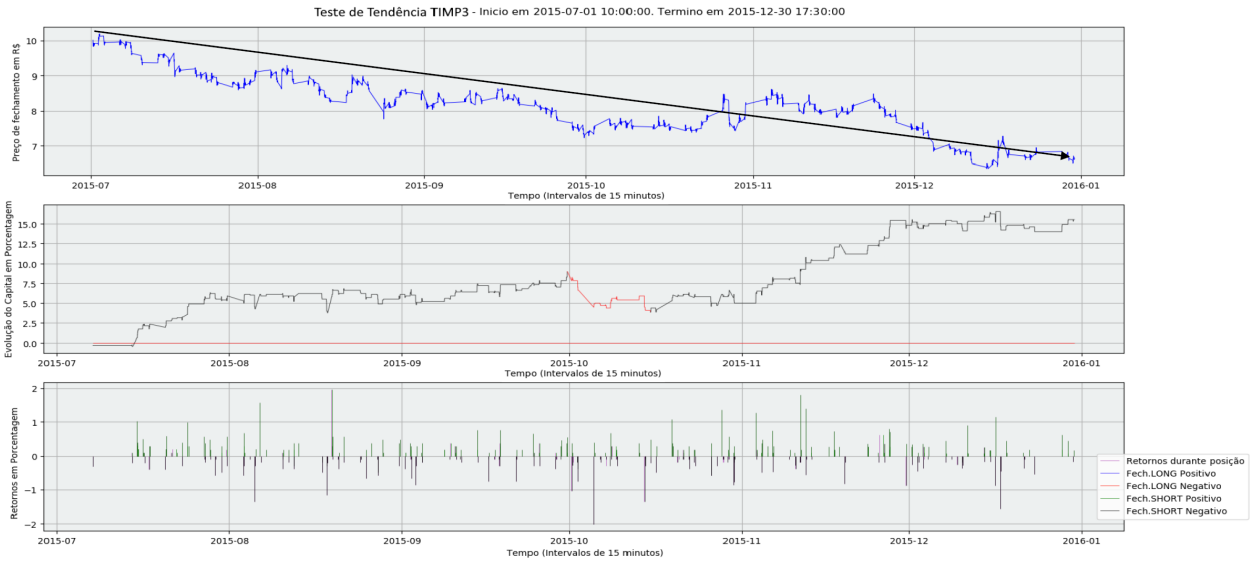

Figura 6.6: Teste em tendência na ação TIMP3 em 2015

Nesse caso, o agente executou por todo o período 317 posições vendidas aproveitando o fato de que a tendência dos preços ação foi majoritariamente de queda ao longo de todo o segundo semestre de 2015. Note-que a partir do início do mês de outubro a tendência dos preços da ação começa a lateralizar o que faz o agente gerar perdas acumuladas correspondentes ao máximo *drawdown* de -4,68%. No entanto, com a volta da tendência de queda no início de novembro o agente permaneceu operando vendido gerando um ganho de aproximadamente 10% em relação ao início do mês e assim permanecendo até o final de dezembro.

6.2 Testes de Desempenho Financeiro

Nos testes de desempenho financeiro no contexto de instabilidade no mercado de ações do ano de 2018 observou-se, como esperado, a partir dos resultados dos testes apresentados a seguir que o Agente RL superou na maioria das ações o Agente LSTM em termos de rendimento financeiro, máximo *drawdown* e nas métricas de risco.

Em termos de rendimento financeiro (Vide Tabela 6.4) o Agente RL foi melhor que o Agente LSTM em quase todas as ações testadas, exceto na ação BBDC4. Na referida tabela, os valores destacados em *verde* referem-se ao melhor rendimento financeiro final entre o Agente RL e o Agente LSTM na respectiva ação. Os valores destacados em negrito referem-se aos maiores valores de rendimento financeiro final entre o Agente RL, Agente LSTM, estratégia *Buy-and-Hold* e o índice BOVA11.

Ação	Agente RL Rendimento Final	Agente LSTM Rendimento Final	Buy-and-Hold Rendimento Final	BOVA11 Rendimento Final
ABEV3	19,32%	9,85%	-11,96%	**22,00%**
B3SA3	21,72%	12,46%	**33,11%**	22,00%
BBAS3	18,95%	12,95%	**64,69%**	22,00%
BBDC4	13,52%	30,37%	**49,22%**	22,00%
ITSA4	**41,45%**	18,05%	36,01%	22,00%
ITUB4	15,11%	8,0%	**36,31%**	22,00%
PETR3	24,80%	7,25%	**34,31%**	22,00%
PETR4	20,99%	19,90%	**38,48%**	22,00%
SUZB3	**24,83%**	9,95%	-15,15%	22,00%
VALE3	**25,77%**	14,09%	7,78%	22,00%

Tabela 6.4: Resultados de rendimento financeiro nos testes de desempenho.

Com relação à métrica de máximo *drawdown* (Vide Tabela 6.5) o Agente RL também obteve desempenho superior ao agente de aprendizado supervisionado, o Agente LSTM, exceto no teste com a ação PETR4. Nessa tabela, os valores em negrito denotam os melhores valores de máximo *drawdown* em cada ação para o Agente RL, Agente LSTM, a estratégia *Buy-and-Hold* e o índice BOVA11.

Nas métricas de risco (Vide Tabela 6.6) o Agente RL também foi melhor que o Agente LSTM tanto em *Sharpe Ratio* quanto no índice Sortino na maioria das ações. Os valores em azul nessa tabela denotam os melhores valores em cada ação para a métrica *Sharpe Ratio* entre o Agente RL e o Agente LSTM, par-a-par. Os melhores valores para a métrica *Sortino Ratio* entre o Agente RL e o Agente LSTM, par-a-par, estão destacados em cor laranja.

Quanto a porcentagem de fechamentos de posições com resultado positivo, a Tabela 6.7 apresenta evidências do que já foi afirmado anteriormente a respeito da limitação de agen-

Ação	Agente RL Máx. Drawdown	Agente LSTM Máx. Drawdown	Buy-and-Hold Máx. Drawdown	BOVA11 Máx. Drawdown
ABEV3	**-7,18%**	-8,13%	-27,80%	-9,12%
B3SA3	-11,09%	-14,01%	-16,01%	**-9,12%**
BBAS3	-9,17%	-9,62%	-21,43%	**-9,12%**
BBDC4	**-7,87%**	-8,03%	-15,60%	-9,12%
ITSA4	**-5,78%**	-7,38%	-13,81%	-9,12%
ITUB4	**-6,69%**	-8,48%	-11,72%	-9,12%
PETR3	**-5,78%**	-11,12%	-23,74%	-9,12%
PETR4	-10,15%	**-8,98%**	-24,97%	-9,12%
SUZB3	**-6,01%**	-17,18%	-38,69%	-9,12%
VALE3	**-4,97%**	-6,09%	-21,23%	-9,12%

Tabela 6.5: Resultados de máximo drawdown nos testes de desempenho.

Ação	Agente RL Sharpe Ratio	Agente LSTM Sharpe Ratio	Agente RL Sortino Ratio	Agente LSTM Sortino Ratio
ABEV3	**0,031**	-0,019	**0,050**	-0,152
B3SA3	**0,040**	0,019	**0,061**	0,028
BBAS3	**0,036**	0,012	**0,055**	0,017
BBDC4	0,031	**0,046**	0,045	**0,068**
ITSA4	**0,024**	-0,020	**0,043**	-0,028
ITUB4	**0,038**	0,035	**0,056**	0,052
PETR3	**0,040**	-0,017	**0,060**	-0,024
PETR4	**0,039**	-0,024	**0,059**	-0,034
SUZB3	0,058	**0,081**	0,084	**0,126**
VALE3	0,016	**0,019**	0,023	**0,027**

Tabela 6.6: Resultados de métricas de risco nos testes de desempenho.

tes de aprendizado supervisionado que procuram otimizar métricas (e.g. acurácia, precisão, f-score, desvio médio quadrático) que não necessariamente implicam em ganhos financeiros quando o modelo é utilizado em um agente de negociação. Na referida tabela, os maiores valores de fechamentos de posições com resultado positivo estão destacados em azul.

Os resultados dessa tabela apontam que embora o Agente LSTM tenha alcançado uma maior taxa de fechamentos de posições com ganhos na maioria das ações testadas, esses ganhos não foram suficientes para superar as vezes em que o Agente RL acertou em fechar posições garantindo a este um rendimento final superior (Vide Tabela 6.4) ao do Agente LSTM na maioria das ações, mesmo tendo o Agente RL apresentado uma taxa de fechamentos positivos menor na maioria das ações testadas.

Essa resultado também pode ser observado na Tabela 6.8 de ganhos e perdas médias de cada agente em cada ação. Nessa tabela os maiores valores de ganho médio estão destacados

Ação	Agente RL Fechs. Positivos	Agente LSTM Fechs. Positivos
ABEV3	50,1%	**62,24%**
B3SA3	49,21%	**51,45%**
BBAS3	50,53%	**53,10%**
BBDC4	51,64%	**54,09%**
ITSA4	46,37%	**56,29%**
ITUB4	**52,09%**	48,33%
PETR3	49,63%	**53,65%**
PETR4	49,58%	**58,26%**
SUZB3	**54,56%**	42,30%
VALE3	**57,13%**	53,72%

Tabela 6.7: Resultados de fechamentos positivos em cada ação nos testes de desempenho.

em verde e os melhores valores de perdas médias estão destacados em vermelho. Observa-se nessa tabela que o Agente LSTM obteve ganhos médios superiores aos do Agente RL na maioria das ações uma vez que o agente de aprendizado supervisionado foi treinado para otimizar sua taxa de acertos.

Ação	Agente RL Ganho Médio	Agente LSTM Ganho Médio	Agente RL Perda Média	Agente LSTM Perda Média
ABEV3	**0,28%**	0,21%	**-0,25%**	-0,32%
B3SA3	0,28%	**0,30%**	**-0,24%**	-0,30%
BBAS3	0,23%	**0,27%**	**-0,22%**	-0,28%
BBDC4	0,23%	**0,26%**	**-0,22%**	-0,26%
ITSA4	**0,30%**	0,23%	**-0,21%**	-0,27%
ITUB4	0,20%	**0,24%**	**-0,19%**	-0,22%
PETR3	0,28%	**0,29%**	**-0,25%**	-0,32%
PETR4	0,26%	**0,27%**	**-0,23%**	-0,35%
SUZB3	0,26%	**0,36%**	-0,27%	**-0,25%**
VALE3	0,20%	**0,22%**	**-0,22%**	-0,23%

Tabela 6.8: Resultados de ganhos médios e perdas médias nos testes de desempenho.

Porém, o Agente LSTM também apresentou as maiores perdas médias em comparação com as do Agente RL. Por isso, mesmo que o Agente LSTM tenha apresentado melhores taxas de acertos nos fechamentos de posições (Vide Tabela 6.7), essa superioridade não foi capaz de produzir rendimentos financeiros superiores aos do Agente RL uma vez que, na média, o Agente LSTM apresentou perdas médias superiores que as do Agente RL.

Esse último, embora tenha apresentado uma taxa de acertos em fechamentos positivos menor do que as do Agente LSTM (Vide Tabela 6.7), apresentou também melhores valores de perdas médias (Vide Tabela 6.8) o que garantiu o maior rendimento financeiro em geral

comparado ao Agente LSTM.

Assim, os resultados experimentais apresentaram evidências de que Agente RL foi melhor em relação ao Agente LSTM em gerar melhores rendimentos financeiros e com menos risco o que se deve à capacidade do agente de aprendizado por reforço em reagir melhor às mudanças no mercado tanto para evitar perdas acumuladas significativas como também para obter ganhos nesses momentos.

É o que se observa também a partir dos gráficos de barras de retornos das figuras dos testes a seguir.

Em geral, o Agente LSTM não foi capaz de detectar tendências mais longas e por isso na maioria das ações executou ao longo dos testes tanto posições compradas quanto vendidas procurando obter ganhos com as oscilações de curto prazo (Veja os gráficos de barras de retornos do Agente LSTM nas Figuras 6.8, 6.9, 6.10,6.12, 6.13, 6.15, 6.16).

Por outro lado, o Agente RL foi mais robusto às variações curtas executando posições compradas seguidas e mudando de estratégia para posições vendidas, ou vice-versa, apenas quando houve significativa mudança de tendência dos preços das ações. É o que se observou nos testes das ações BBAS3 (Vide Figura 6.9), ITUB4 (Vide Figura 6.12), PETR3 (Vide Figura 6.13), SUZB3 (Vide Figura 6.15) e VALE3 (Vide Figura 6.16). Ressalte-se ainda que esse comportamento do Agente RL também foi observado nos testes de tendência.

Note-se que a maioria das ações testadas no ano de 2018 apresentou uma tendência majoritariamente de alta no segundo semestre daquele ano (Vide gráficos na Figura 5.8). Dessa forma, o Agente RL apresentou uma maior taxa de acertos nos fechamentos de posições compradas (LONG) como pode ser observado nos dados da Tabela 6.9. Isso indica que o Agente RL conseguiu detectar a tendência de alta de longo prazo e por isso apresentou uma maior taxa de acertos nos fechamentos das posições compradas o que também reforça a propriedade de robustez do Agente RL diante de variações curtas de tendência.

Por outro lado, o Agente LSTM não foi capaz de detectar as tendências de alta no longo prazo tendo apresentado desempenho inferior ao Agente RL nos fechamentos positivos de posições compradas (Vide Tabela 6.9) embora tenha apresentado desempenho superior nos fechamento de posições vendidas (Vide Tabela 6.10). Foi isso que prejudicou o rendimento financeiro do Agente LSTM pois os movimentos de alta apresentaram maior valor financeiro provocando a tendência de alta no longo prazo o que foi captado e aproveitado pelo Agente RL para obter maiores ganhos.

Essas diferenças de percepções de curto e longo prazo entre o Agente RL e o Agente LSTM também refletiram na suavidade das curvas de evolução do capital dos teste de cada agente. Em geral, observou-se que as curvas de evolução capital do Agente RL foram menos irregulares que as curvas do Agente LSTM. Isso afetou as métricas de risco de cada agente (Veja Tabela 6.6) tendo o Agente RL apresentado um melhor desempenho nas métricas de

Ações	Agente RL Fech. Pos. LONG Positivos	Agente LSTM Fech. Pos. LONG Positivos
ABEV3	-	-
B3SA3	**48,55%**	46,38%
BBAS3	**51,04%**	46,96%
BBDC4	**51,18%**	46,38%
ITSA4	46,37%	**47,05%**
ITUB4	**57,79%**	43,99%
PETR3	**50,29%**	46,57%
PETR4	**47,59%**	43,90%
SUZB3	**54,10%**	39,64%
VALE3	**55,09%**	41,79%

Tabela 6.9: Resultados de fechamentos positivos em posições compradas (LONG) nos testes de desempenho.

Ações	Agente RL Fech. Pos. SHORT Positivos	Agente LSTM Fech. Pos. SHORT Positivos
ABEV3	50,09%	**62,24%**
B3SA3	**61,03%**	57,36%
BBAS3	49,80%	**58,02%**
BBDC4	52,90%	**59,09%**
ITSA4	-	57,82%
ITUB4	49,84%	**53,72%**
PETR3	49,37%	**55,89%**
PETR4	50,40%	**58,82%**
SUZB3	54,93%	**58,22%**
VALE3	57,59%	**60,57%**

Tabela 6.10: Resultados de fechamentos positivos em posições vendidas (SHORT) nos testes de desempenho.

risco exceto nas ações BBDC4 (Vide Figura 6.10), SUZB3 (Vide Figura 6.15), VALE3 (Vide Figura 6.16) onde o agente LSTM soube aproveitar melhor as variações de curto prazo.

6.3 Síntese dos Resultados

Os resultados dos testes de tendência evidenciaram portanto a propriedade do agente de aprendizado por reforço (Agente RL) em modificar dinamicamente (*on-line*) sua estratégia de operação no mercado à medida que as condições atuais de tendência de preços da ação de alteraram. Isso ficou evidente nos testes da ação BBDC3 (Vide Figura 6.2), CIEL3 (Vide Figura 6.3), NATU3 (Vide Figura 6.4), USIM5 (Vide Figura 6.5) e TIMP3 (Vide Figura 6.6).

Os resultado do testes de desempenho também evidenciaram a propriedade do Agente

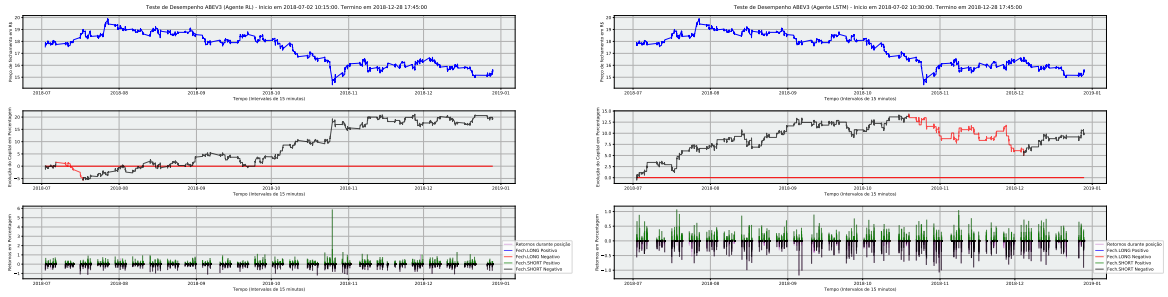

Figura 6.7: Teste de desempenho na ação ABEV3 em 2018.

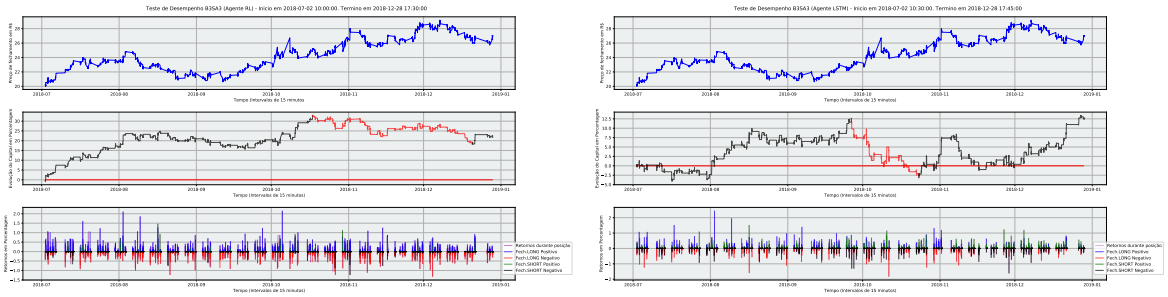

Figura 6.8: Teste de desempenho na ação B3SA3 em 2018.

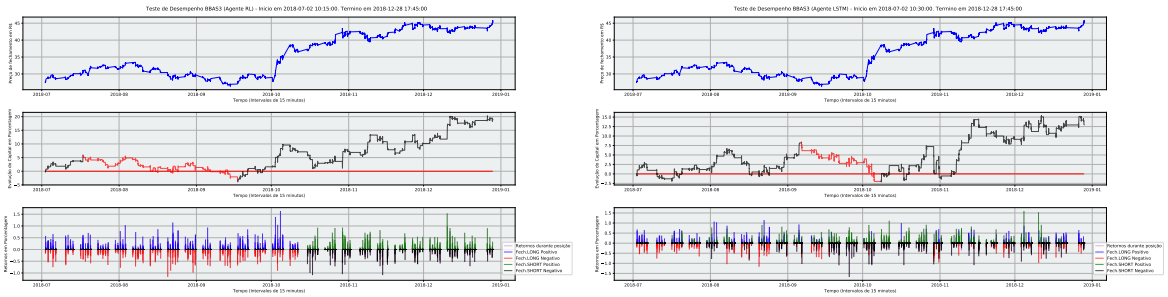

Figura 6.9: Teste de desempenho na ação BBAS3 em 2018.

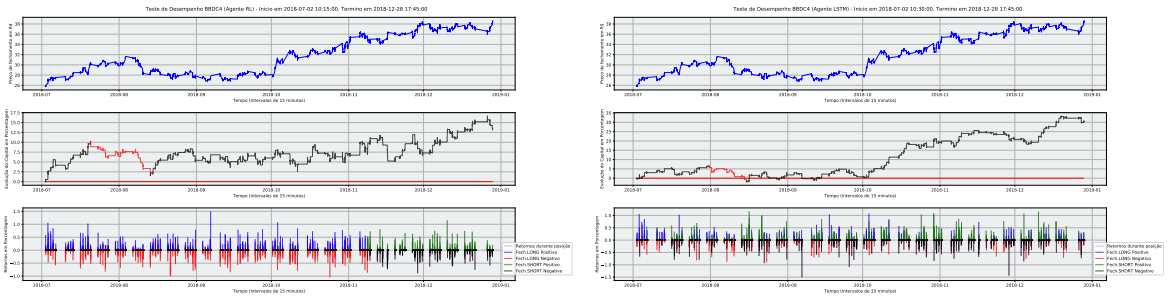

Figura 6.10: Teste de desempenho na ação BBDC4 em 2018.

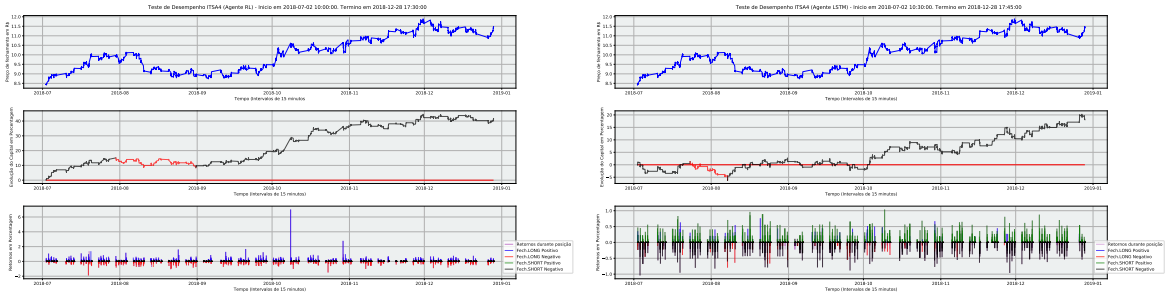

Figura 6.11: Teste de desempenho na ação ITSA4 em 2018.

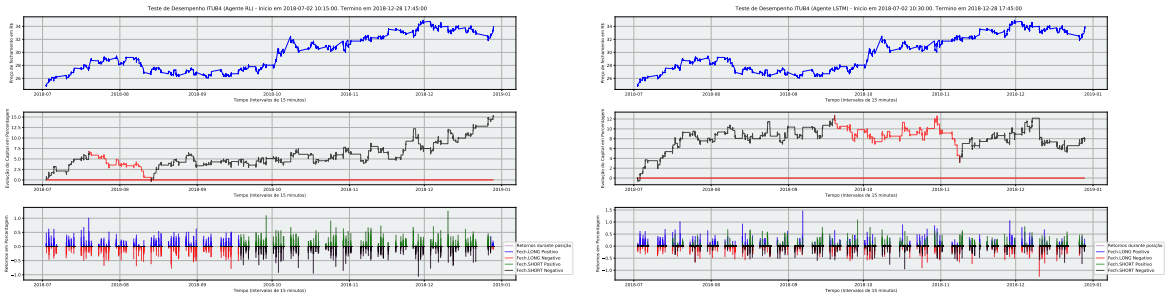

Figura 6.12: Teste de desempenho na ação ITUB4 em 2018.

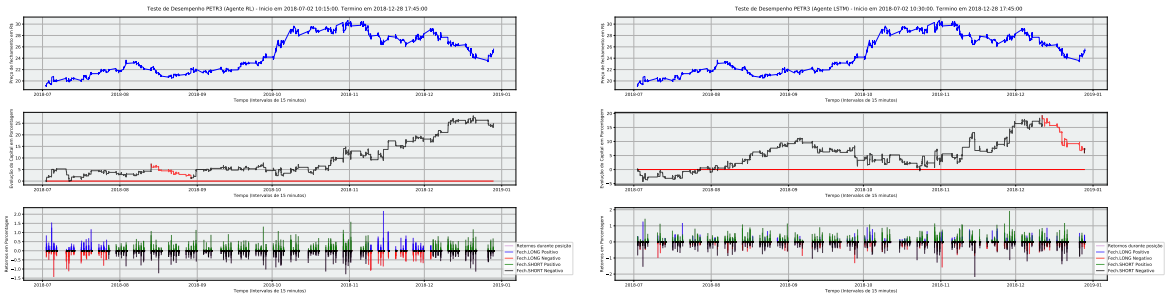

Figura 6.13: Teste de desempenho na ação PETR3 em 2018.

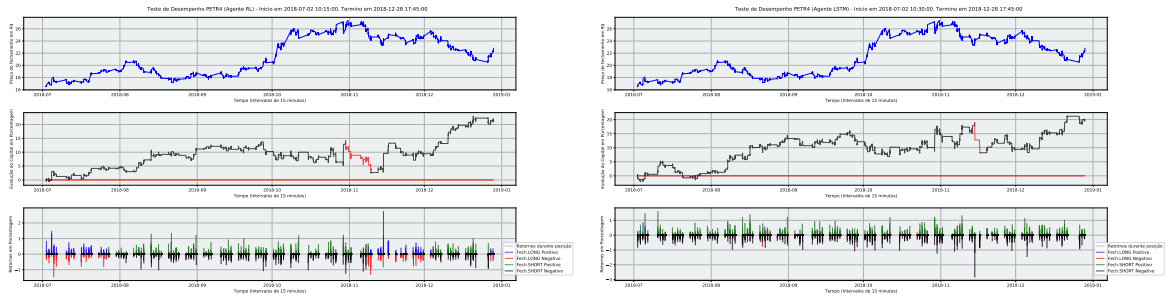

Figura 6.14: Teste de desempenho na ação PETR4 em 2018.

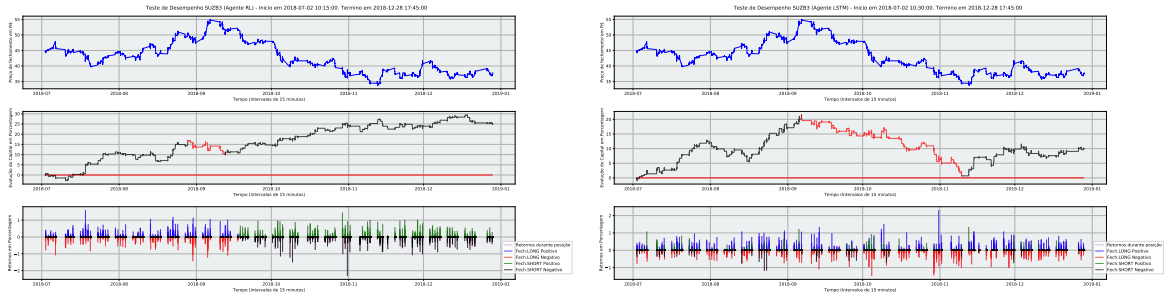

Figura 6.15: Teste de desempenho na ação SUZB3 em 2018.

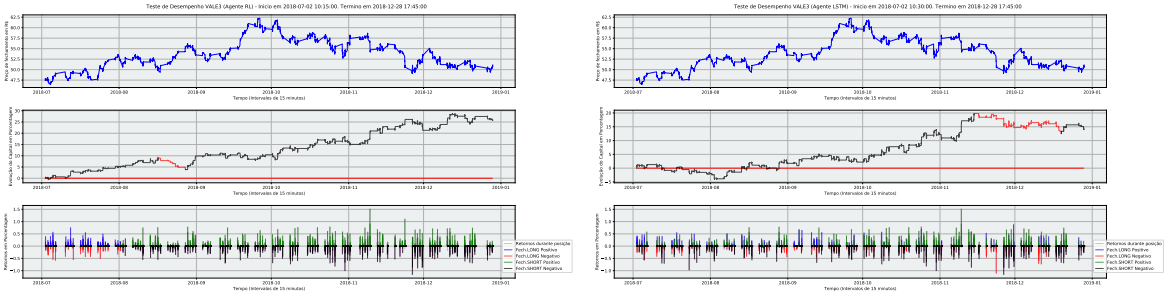

Figura 6.16: Teste de desempenho na ação VALE3 em 2018.

RL em produzir ganhos financeiros superiores superiores a um agente baseado em aprendizado supervisionado (Agente LSTM) tanto em termos de magnitude do rendimento financeiro como em termos de risco. É o que se observou sobretudo nos resultados das Tabelas 6.4, 6.6 e 6.5.

Ficou ainda evidenciada a deficiência do agente de aprendizado supervisionado no contexto de negociação de ações. Embora tenha o Agente LSTM alcançado uma taxa de acertos nos fechamentos de posições com resultados positivos (Vide Tabela 6.7) superior ao Agente RL, o que decorre do fato de que o agente de aprendizado supervisionado busca otimizar uma taxa de acertos (e.g. acurácia nas previsões), essa superioridade não se refletiu nos resultados financeiros desse agente (Vide Tabela 6.4). Isso ficou ainda mais evidente nos resultados de perda média da Tabela 6.8 em que o Agente LSTM apresentou as maiores perdas médias e também não foi capaz de detectar as tendências de longo prazo das ações no segundo semestre de 2018 apresentado uma taxa de acertos em posições compradas (Vide Tabela 6.9) inferior as do Agente RL.

Capítulo 7

Conclusões e Trabalhos Futuros

Os sistemas automatizados de negociação de ações baseados em modelos de aprendizado supervisionado podem apresentar bons valores em termos de acurácia na previsão de tendências ou retornos de ações. Contudo, esse desempenho não é suficiente para se produzir um agente de negociação que gere também bons ganhos financeiros. Além, disso esses modelos não são capazes de se adaptarem dinamicamente as mudanças repentinas de tendências de preços das ações necessitando de constantes retreinamentos.

Uma alternativa a essas deficiências de agente baseados em modelos de aprendizado supervisionado reside nos agentes de aprendizado por reforço. Nessa abordagem, um agente é capaz de aprender a agir em um determinado ambiente a partir de sua própria experiência nesse ambiente. Além disso, também é capaz de se manter atualizado diante das mudanças do ambiente de modo a buscar sempre as ações que maximizem o acúmulo de recompensas.

Essas propriedades seriam adequadas a aplicação em negociação de ativos negociados em bolsas de valores uma vez que um agente que não precisasse ser retreinado a cada mudança de condições do mercado aproveitaria melhor as oportunidades de negócios nesses instantes.

Partindo das referidas premissas, modelou-se um agente de negociação de ações com o algoritmo de aprendizado de reforço SARSA, denominado Agente RL, utilizando um espaço de estados e ações discretos e finitos o que favorece a convergência da politica aprendida no treinamento. O objetivo do Agente RL é maximizar o rendimento financeiro total ao longo de sua execução operando com posições compradas ou vendidas com uma ação.

Para testar as propriedades do Agente RL estabeleceu-se duas hipóteses: a) O Agente RL é capaz de mudar de estratégia diante de mudanças significativas nas condições de tendência da ação e b) O agente também pode apresentar desempenho superior a um agente de aprendizado supervisionado em termos de rendimento financeiro, métricas de risco e taxas de acerto.

Para testar a primeira hipóteses aplicou-se o Agente RL proposto em um conjunto de 6 ações com diferentes condições de tendência ao longo de 1 ano. Os resultados desses testes apresentaram evidências no sentido da hipótese uma que o agente foi capaz de mudar de estratégia nos momentos de inversões significativas de tendência nas ações testadas.

Aproveitando essa propriedade para testar a segunda hipótese, escolheu-se um conjunto de 10 ações da B3 no ano de 2018 esperando-se que o Agente RL apresentasse desempenho superior ao Agente LSTM em rendimento financeiro e risco no contexto de instabilidade no mercado naquele ano devido as eleições nacionais.

Os resultados nos testes de desempenho apresentaram evidências que sugerem que o Agente RL proposto foi capaz de não só produzir rendimento financeiro superior ao Agente LSTM na maioria das ações como também gerar um rendimento final com menos risco.

Além disso, os resultados nos testes de desempenho evidenciaram as limitações de agentes baseados em aprendizado supervisionado uma vez que o Agente LSTM embora tenha apresentado maiores taxas de acertos nos fechamentos de posições, essa superioridade não foi capaz de se converter em rendimentos financeiros superiores uma vez que as perdas médias desse agente também foram de mesma dimensão ou superior afetando negativamente seu desempenho financeiro.

Por outro lado, o Agente RL, mesmo tendo apresentado menores taxas de acertos em relação ao Agente RL, apresentou também perdas médias inferiores às do Agente LSTM, o que garantiu ao agente de aprendizado por reforço um melhor rendimento financeiro e melhores taxas de risco.

Em suma, os referidos resultados apresentados nos testes apresentaram evidências no sentido de que um agente de aprendizado por reforço pode ser uma alternativa viável aos sistemas de negociação baseados em aprendizado supervisionado podendo produzir ganhos financeiros em contextos de instabilidade com menores perdas acumuladas, menor risco e dispensando ciclos longos de retreinamentos comuns aos sistemas baseados em aprendizado supervisionado.

7.1 Escopo e Limitações

Este trabalho teve como escopo apresentar evidências experimentais das seguintes propriedades de um agente de aprendizado por reforço proposto para negociação de ações:

- Mudança dinâmica de estratégia de negociação a partir de mudanças nas condições de tendência de preço de uma ação.
- Desempenho superior em comparação a um agente baseado em aprendizado supervisionado em termos de rendimento financeiro e risco.

Assim, o trabalho limitou-se com relação à primeira propriedade a aplicar o agente de aprendizado por reforço, Agente RL, nas condições de tendência (alta, baixa e pouca tendência) das ações selecionadas nos anos selecionados. Não foram selecionados outros períodos nem outras ações devido a indisponibilidade de dados de boa qualidade (preços ajustados para dividendos e *splits*, ausência de entradas faltantes, maior número de entradas).

Com relação a periodicidade dos dados, utilizou-se a de 15 minutos porque era a que apresentava a maior quantidade de entradas nos dados disponíveis. Caso fosse feita uma reamostragem para periodicidades diárias, semanais, mensais isso implicaria na diminuição dos dados disponíveis para treino e teste o que comprometeria o aprendizado do Agente RL e do Agente LSTM e as respectivas análises dos resultados.

Para a segunda propriedade, o trabalho limitou-se a aplicar o Agente RL e o Agente LSTM nas 10 ações mais negociadas do ano de 2018 porque o objetivo foi determinar o quão robusto poderia ser o Agente RL comparado ao Agente LSTM em termos de rendimento financeiro e risco em um contexto de instabilidade no mercado de ações naquele ano.

Não foram considerados aspectos de custos de operação (e.g. taxas de corretagem, emolumentos, tributos), liquidez para execução das ordens e desvio entre o valor da ordem emitida pelo agente e o valor da ordem efetivamente executada na bolsa (*slippage* em inglês) uma vez que tais aspectos implicariam na adição de complexidades práticas em termos de simulação que extrapolariam os objetivos do trabalho e dificultariam a análise dos resultados.

Como o objetivo era comparar um agente de aprendizado por reforço com um agente baseado em aprendizado supervisionado não foram também analisadas outras modelagens baseadas exclusivamente em indicadores técnicos, modelos de séries temporais (e.g. ARIMA - *Auto Regressive Integrated Moving Average*, GARCH - *Generalized Auto Regressive Conditional Heteroskedastic* ,VAR - *Vector Auto Regressive*), agentes baseados em aprendizado por reforço profundo e outros modelos de aprendizado supervisionado como árvores de decisão, SVM (*Support-Vector Machine* em inglês), KNN (*K-Nearest Neighbors*) e outros tipos de redes neurais (autoencoder, redes de convolução, rede recorrentes simples).

Com relação às restrições de operação dos agentes não foram consideradas operações com *stop loss* e *take-profit* pois o objetivo era de que ambos os agentes aprendessem quando entrar em uma posição e quando sair independentemente de qualquer restrição quanto a perdas e ganhos. Essa escolha também favoreceu a análise das métricas de risco de cada agente uma vez que caso fossem escolhidos valores de *stop loss* e *take-profit* a análise ficaria limitada aos respectivos valores. O mesmo pode ser dito com relação a operações com alavancagem.

Também limitou-se as operações dos agentes para somente as ordens intra-diário (*intraday*) para evitar custos com aluguel de ações em operações vendidas (*shorting*) e evitar exposições excessivas nas viradas de dias e finais de semana.

Não foram analisados outros ativos financeiros tais como opções, títulos de mercado futuro (e.g. mini-índice, mini-dólar) ou mercado de câmbio em virtude da indisponibilidade de dados de boa qualidade desses ativos durante a execução do trabalho.

7.2 Trabalhos Futuros

Os resultados apresentados com a modelagem proposta nesse trabalho além de outras propostas de modelagens e resultados observados na revisão da literatura estimulam o aprofundamento no estudo de agentes de aprendizado por reforço em negociação de ativos financeiros.

Esse parece ser uma linha de pesquisa ainda pouco explorada comparada aos que já se produziu de resultados e modelagens utilizando técnicas de aprendizado supervisionado ou não-supervisionado o que, portanto, pode ser uma oportunidade em termos de contribuições tanto para a academia como para o mercado.

Das oportunidades de aprofundamento em aprendizado por reforço no mercado financeiro e sugere-se:

- Uma implementação em que o agente opere não só com 1 ativo mas com um portfólio de ações, títulos futuros, *commodities*, cripto moedas e até fundos de investimentos.
- Uma modelagem em que o agente opere orientado a um determinado nível de retorno pré-especificado pelo investidor. Uma vez atingido o ganho desejado o agente termina sua execução no estado final. Dessa forma, obtém uma modelagem de estados em aprendizado por reforço em que exite um estado final definido. Isso permite determinar durante a execução do agente o quanto ele está distante do objetivo a ser alcançado.
- Modelagens considerando custos de transação e operações com volumes variáveis.
- Em uma modelagem multiagentes poderia-se criar agentes para operar em estados específicos, isto é, se o agente está em posição comprada haveria um agente para decidir permanecer posicionado e outro agente para decidir finalizar a ação. Assim, também poderia haver os respectivos agentes para estados em que o agente está posicionado em short ou não posicionado. Dessa forma, para cada tipo de posição haveria um comitê para decidir qual a próxima ação a executada.
- A operação com cripto moedas também apresenta desafios importantes dada a facilidade de obtenção de dados de negociação desses ativos bem como as peculiaridades desse mercado (e.g. facilidade de negociação, baixo custo de operação e até implementação real) o que viabilizam até mesmo testes e experimentos em contas proporcionado resultados e análises mais robustas.

Referências Bibliográficas

Alimoradi, M. R. & Kashan, A. H. (2018). A league championship algorithm equipped with network structure and backward q-learning for extracting stock trading rules. *Applied Soft Computing*, 68:478–493. ISSN 1568-4946.

Almahdi, S. & Yang, S. Y. (2017). An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Systems with Applications*, 87:267–279. ISSN 0957-4174.

Alpaydin, E. (2014). *Introduction to Machine Learning*. The MIT Press, 3rd edition edição. ISBN 0262028182.

Bacon, C. R. (2008). *Practical portfolio performance measurement and attribution*, volume 546. John Wiley & Sons.

Bertoluzzo, F. & Corazza, M. (2012). Testing different reinforcement learning configurations for financial trading: Introduction and applications. *Procedia Economics and Finance*, 3:68--77.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Bodie, Z.; Kane, A.; Marcus, A. J. & Mohanty, P. (2008). *Investments (SIE)*. McGraw-Hill Education.

Chen, C. T.; Chen, A. & Huang, S. (2018). Cloning strategies from trading records using agent-based reinforcement learning algorithm. pp. 34--37.

Chen, K.; Zhou, Y. & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of china stock market.

Chong, E.; Han, C. & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, 83:187--205.

Colby, R. W. & Meyers, T. A. (1988). *The encyclopedia of technical market indicators*. Dow Jones-Irwin Homewood, IL.

Corazza, M. & Sangalli, A. (2015). Q-learning and SARSA: A comparison between two intelligent stochastic control approaches for financial trading. *SSRN Electronic Journal*.

Dempster, M.; Payne, T.; Romahi, Y. & Thompson, G. (2001). Computational learning techniques for intraday FX trading using popular technical indicators. *IEEE Transactions on Neural Networks*, 12(4):744--754.

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

Ding, Y.; Liu, W.; Bian, J.; Zhang, D. & Liu, T.-Y. (2018). Investor-imitator: A framework for trading knowledge extraction. pp. 1310--1319.

Fan, A. & Palaniswami, M. (2001). Stock selection using support vector machines. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, volume 3, pp. 1793--1798. IEEE.

Faustryjak, D.; Jackowska-Strumillo, L. & Majchrowicz, M. (2018). Forward forecast of stock prices using lstm neural networks with statistical analysis of published messages. pp. 288--292.

Fortuna, E. (2015). *Mercado financeiro: produtos e servicos. rev. atual. e ampl*. Qualitymark, 20 edição. ISBN 9788541401890.

Gao, X. (2018). Deep reinforcement learning for time series: playing idealized trading games. *http://arxiv.org/abs/1803.03916v1*.

Ghosh, A.; Bose, S.; Maji, G.; Debnath, N. & Sen, S. (2019). Stock price prediction using lstm on indian share market. In *Proceedings of 32nd International Conference on*, volume 63, pp. 101--110.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735--1780. ISSN 0899-7667.

Hu, Y.-J. & Lin, S.-J. (2019). Deep reinforcement learning for optimizing finance portfolio management. In *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pp. 14--20. IEEE.

Iskrich, D. & Grigoriev, D. (2017). Generating long-term trading system rules using a genetic algorithm based on analyzing historical data. In *2017 20th Conference of Open Innovations Association (FRUCT)*, pp. 91--97. IEEE.

James, G.; Witten, D.; Hastie, T. & Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Jia, W.; Chen, W.; XIONG, L. & Hongyong, S. (2019). Quantitative trading on stock market based on deep reinforcement learning. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1--8. IEEE.

Jiang, Z. & Liang, J. (2016). Cryptocurrency portfolio management with deep reinforcement learning. *Intelligent Systems Conference 2017*.

JuHyok, U.; Lu, P.; Kim, C.; Ryu, U. & Pak, K. (2020). A new lstm based reversal point prediction method using upward/downward reversal point feature sets. *Chaos, Solitons & Fractals*, 132:109559.

Kim, H. Y. & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models. *Expert Systems with Applications*, 103:25 – 37. ISSN 0957-4174.

Kirkpatrick II, C. D. & Dahlquist, J. A. (2010). *Technical analysis: the complete resource for financial market technicians*. FT press.

Lee, J.; Kim, R.; Koh, Y. & Kang, J. (2019). Global stock market prediction based on stock chart images using deep q-network. *IEEE Access*, 7:167260--167277.

Lee, J. W. (2001). Stock price prediction using reinforcement learning. 1:690--695 vol.1.

Lee, J. W.; Park, J.; O, J.; Lee, J. & Hong, E. (2007). A multiagent approach to q-learning for daily stock trading. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 37(6):864--877.

Lei, K.; Zhang, B.; Li, Y.; Yang, M. & Shen, Y. (2020). Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading. *Expert Systems with Applications*, 140:112872.

Li, Y.; Zheng, W. & Zheng, Z. (2019a). Deep robust reinforcement learning for practical algorithmic trading. *IEEE Access*, 7:108014--108022.

Li, Y.; Zheng, W. & Zheng, Z. (2019b). Deep robust reinforcement learning for practical algorithmic trading. *IEEE Access*, 7:108014--108022.

Liu, S.; Liao, G. & Ding, Y. (2018). Stock transaction prediction modeling and analysis based on lstm. pp. 2787--2790. ISSN 2158-2297.

Lohpetch, D. & Corne, D. (2009). Discovering effective technical trading rules with genetic programming: Towards robustly outperforming buy-and-hold. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 439--444. IEEE.

Meng, T. L. & Khushi, M. (2019). Reinforcement learning in financial markets. *Data*, 4(3):110.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G. et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

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

Moody, J. E. & Saffell, M. (1998). Reinforcement learning for trading. pp. 917--923.

Naeini, M. P.; Taremian, H. & Hashemi, H. B. (2010). Stock market value prediction using neural networks. In *2010 international conference on computer information systems and industrial management applications (CISIM)*, pp. 132--136. IEEE.

Nelson, D. M. Q.; Pereira, A. C. M. & de Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. *2017 International Joint Conference on Neural Networks (IJCNN)*.

Neuneier, R. (1998). Enhancing q-learning for optimal asset allocation. In *Advances in neural information processing systems*, pp. 936--942.

Padua Braga, A. (2007). *Redes neurais artificiais: teoria e aplicacoes*. LTC Editora. ISBN 9788521615644.

Pendharkar, P. C. & Cusatis, P. (2018). Trading financial indices with reinforcement learning agents. *Expert Systems with Applications*, 103:1 – 13. ISSN 0957-4174.

Reddy, G.; Wong-Ng, J.; Celani, A.; Sejnowski, T. J. & Vergassola, M. (2018). Glider soaring via reinforcement learning in the field. *Nature*, 562(7726):236--239. ISSN 1476-4687.

Russell, S. J. & Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

Rutkauskas, A. V. & Ramanauskas, T. (2009). Building an artificial stock market populated by reinforcement learning agents. *Journal of Business Economics and Management*, 10(4):329--341.

Serrano, A.; Imbernón, B.; Pérez-Sánchez, H.; Cecilia, J. M.; Bueno-Crespo, A. & Abellán, J. L. (2018). Accelerating drugs discovery with deep reinforcement learning: An early approach. pp. 6:1--6:8.

Si, W.; Li, J.; Ding, P. & Rao, R. (2017). A multi-objective deep reinforcement learning approach for stock index future's intraday trading.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M. et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484.

Sutton, R. S. & Barto, A. G. (2018). *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 2 nd edição. ISBN 0262193981.

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1--103.

Tavares, A. R. & Chaimowicz, L. (2018). Tabular reinforcement learning in real-time strategy games via options.

Wilmott, P. (2013). *Paul Wilmott on quantitative finance*. John Wiley & Sons.

Wu, M.-C.; Lin, S.-Y. & Lin, C.-H. (2006). An effective application of decision tree to stock trading. *Expert Systems with Applications*, 31(2):270--274.

Xiao, C. & Chen, W. (2018). Trading the twitter sentiment with reinforcement learning. *arXiv preprint arXiv:1801.02243*.

Xiong, Z.; Liu, X.-Y.; Zhong, S.; Hongyang; Yang & Walid, A. (2018). Practical deep reinforcement learning approach for stock trading. *32nd Conference on Neural Information Processing Systems (NIPS 2018), Montreal, Canada.*

Yao, S.; Luo, L. & Peng, H. (2018). High-frequency stock trend forecast using lstm model. pp. 1--4. ISSN 2473-9464.