



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

**Hare: Serviço de investimento baseado em agentes autônomos para operar na B<sup>3</sup>**

Khalil C. do Nascimento  
Renato A. Nobre

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Ciência da Computação

Orientador  
Prof. Dr. Geraldo P. Rocha Filho

Brasília  
2020



# Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

## **Hare: Serviço de investimento baseado em agentes autônomos para operar na B<sup>3</sup>**

Khalil C. do Nascimento  
Renato A. Nobre

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Geraldo P. Rocha Filho (Orientador)  
CIC/UnB

Prof. Dr. Donald Knuth    Dr. Leslie Lamport  
Stanford University       Microsoft Research

Prof. Dr. Edison Ishikawa  
Coordenador do Bacharelado em Ciência da Computação

Brasília, 24 de dezembro de 2020

# Dedicatória

Na *dedicatória* o autor presta homenagem a alguma pessoa (ou grupo de pessoas) que têm significado especial na vida pessoal ou profissional. Por exemplo (e citando o poeta): *Eu dedico essa música a primeira garota que tá sentada ali na fila. Brigado!*

# Agradecimentos

Os autores gostariam de começar agradecendo em conjunto a todos aqueles que nos ajudaram, sendo na confecção desse trabalho, ou sendo no nosso dia-a-dia. Ao Prof. Dr. Geraldo Pereira por acreditar no potencial do nosso projeto e se dispor a nos ajudar mesmo fora de sua área de pesquisa. A OpenAI por fornecer bibliotecas de ponta para o desenvolvimento de um futuro com inteligência artificial geral que beneficiará toda a humanidade. A todas a rádios de Jazz e Lo-fi do Youtube, com suas músicas tranquilas que nos forneceram paciência para conseguir fazer os modelos funcionarem. Aos nossos amigos que se fazem presente para nos apoiar diariamente e levaremos para a vida: André Marques, Augusto Brandão, Bernardo Carsten, Camila Sidersky, Claudio Segala, Daniel Bemerguy, Johannes Peter, Léo Akira, Marcelo Araújo, Mateus Bittencourt, Pedro Saman, Ricardo Nunes e Rodrigo Navarro.

O autor Khalil Carsten agradece com as seguinte mensagem:

“Direciono, prioritariamente, ao meu colega de curso e amigo, Renato Avellar Nobre, meus agradecimentos pela sua incrível lógica científica e capacidade de escrita que me conduziram em momentos de confusão, além de sua determinação e disciplina nas quais me mantiveram focado e disposto durante todo o desenvolvimento deste trabalho.

Com muito carinho também agradeço aos meus pais Benivaldo do Nascimento Junior e Germana Magalhães Carsten por todo conforto emocional e financeiro que me proveram para que esse trabalho fosse concluído, sem eles nada disso teria sido possível.

Também aos meus irmãos Caio Lemos e Carmel Carsten por serem meu parceiros de vida e me proporcionarem momentos de alegria e divertimento.

Da mesma forma agradeço aos meus Tios Carla Miranda e Alexandre Miranda por sempre me receberem como um filho, juntamente com meus primos Gabriel Miranda e Bernardo Miranda por se disporem a me ouvir em tranquilos cafés da manhã em sua casa.”

O autor Renato Nobre agradece com as seguinte mensagem:

“Gostaria de começar agradecendo ao Khalil Carsten por toda a amizade e parceria desenvolvida durante esses 5 anos de graduação, pela paciência para me aguentar, pelos códigos incríveis, e por aceitar o desenvolvimento deste trabalho megalomaníaco.

A meus pais Ademar Thadeu Murta Nobre e Martha Maria Nobre Avellar, e meus avós Nair Renata Nobre Avellar e Américo José Avellar, pela confiança, apoio, orientação, e suporte para a realização desse curso e deste trabalho, nada do que eu sou hoje seria possível sem a contribuição de vocês. O mesmo se estende também para toda a minha família, meus irmãos e tios, que também sempre estiveram ao meu lado e criaram essa família acolhedora e presente a qual eu pertenço.

Gostaria de agradecer também, com todo meu coração, a Giovanna Mundstock, você é o amor da minha vida. Obrigado por todos os momentos de apoio, compreensão e incentivos. Cada dia ao seu lado eu cresço mais um pouco e você me motiva a ir mais longe.

Também não pode faltar agradecimentos aos meus amigos de São Paulo, pelos conselhos de investimento e pelas conversas de bar que tivemos em 2017 que me motivaram na criação desse projeto: Daniel Miranda, Rodrigo Nogueira e Guilherme Carvalho.

Finalmente, agradeço ao Prof. Dr. Guilherme Novaes Ramos por fornecer uma oportunidade de pesquisa para mim no meu segundo semestre de graduação, e me orientar em diversos outros momentos ao longo desses 5 anos. Essa oportunidade me apresentou a aprendizagem de máquina e o aprendizado por reforço, o ramo no qual viria a dedicar todo o resto da minha graduação.”

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

# Resumo

Os mercados de ações desempenham um papel crucial na economia, permitindo o crescimento de empresas e possibilitando uma geração de rendimento para seus investidores. Na literatura de aprendizado de máquina para mercados financeiros, múltiplas ferramentas e técnicas foram propostas e aplicadas para analisar o comportamento geral do mercado. No entanto, entender as regras intrínsecas do funcionamento da bolsa de valores, com a possibilidade de gerar lucros, está longe de ser uma tarefa trivial. Abordando esse desafio, este trabalho propõe o Hare: um serviço de investimento com técnicas híbridas, orientado a agentes racionais autônomos, para negociar ativos no mercado de ações. O Hare oferece um serviço confiável com alta precisão e estabilidade no processo de tomadas de decisão, se baseando em análises técnicas e fundamentais. O cerne de funcionamento do Hare é sua modelagem utilizando um agente racional capaz de perceber o mercado e agir de forma autônoma com base em suas decisões. Para tal dois módulos principais foram implementados com o objetivo de fornecer uma racionalidade ao agente: (i) o Módulo Preditor de Movimento (MPM), responsável por prever a movimentação de um ativo; e (ii) o Modelo de Alocação de Recursos (MAR), responsável por utilizar as informações de predição ao seu favor, distribuindo seus recursos entre os ativos disponíveis para tentar gerar o maior lucro possível com o menor risco. Como prova de conceito o Hare foi projetado para operar gerenciando um portfólio no mercado de ações da [B]<sup>3</sup>. Os resultados avaliados do MPM demonstraram que o serviço é capaz de prever o ganho ou perda de valor no preço de uma ação, quando comparado com sua média da janela de tempo analisada, com uma acurácia de 82% no pior caso e 94% na melhor situação. Ademais, o MAR foi capaz de obter uma rentabilidade de 11,74% gerenciando um portfólio com 3 ativos no período de tempo analisado, demonstrando que o serviço Hare é uma opção de investimento viável capaz de superar a rentabilidade de investimentos de renda fixa e de portfólios construídos com base na Variância Média de Markowitz.

**Palavras-chave:** LaTeX, metodologia científica, trabalho de conclusão de curso

# Abstract

Stock markets play an essential role in the economy and offer companies opportunities to grow and investors to make profits. Many tools and techniques have been proposed and applied for the analysis of the overall market behavior. However, understanding the intrinsic rules of the stock exchange and thus seizing opportunities are not trivial tasks. To address this challenge, this work proposes Hare: a new hybrid, autonomous, agent-orientated service created to trade equities in the stock market successfully. It offers a reliable service based on technical and fundamental analysis with high precision and stability in the decision-making process. Hare's intelligence core is modeled using a rational agent capable of perceiving the market and acting upon its perception autonomously. Two main modules were implemented to provide the agent's rationality: (i) a Predictor Module, which is responsible for forecasting an asset's movement; and (ii) a Resource Allocation Module, that given the predictions made by the agent, distributes its resources trying to generate the maximum profit with the lowest risk. As proof of concept, Hare was designed to operate on the [B]<sup>3</sup> stock exchange, considering different stocks in its portfolio. The Predictor Module results showed that the proposed service could predict the rise or fall of a price compared to its time-steps' mean with an accuracy of 82% in the worst case and 94% in the best case. Furthermore, the Resource Allocation Module was capable of achieving an 11,74% rentability managing a portfolio in the evaluated period, demonstrating that the Hare service is a viable investment system able to overcome fixed income investments and portfolios built with the Markowitz Mean-Variance model.

**Keywords:** LaTeX, scientific method, thesis

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	3
1.2	Contribuição . . . . .	3
1.3	Estrutura do Trabalho . . . . .	4
<b>2</b>	<b>Fundamentação Teórica</b>	<b>5</b>
2.1	Economia e Mercado . . . . .	5
2.1.1	O Mercado de Ações . . . . .	6
2.1.2	Teoria Moderna do Portfólio . . . . .	7
2.2	Aprendizado de Máquina . . . . .	9
2.2.1	k-Vizinhos Mais Próximos . . . . .	11
2.2.2	Máquina de Vetores de Suporte . . . . .	12
2.3	Aprendizado Profundo . . . . .	14
2.3.1	Redes Neurais Recorrentes . . . . .	16
2.3.2	Long Short-Term Memory . . . . .	17
2.3.3	Gated Recurrent Unit . . . . .	19
2.4	Aprendizado Por Reforço . . . . .	20
2.4.1	Gradiente de Política Determinística Profunda . . . . .	23
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>27</b>
3.1	Análise Fundamental . . . . .	27
3.2	Análise Técnica . . . . .	28
3.3	Considerações Finais . . . . .	29
<b>4</b>	<b>Hare: Um Serviço Autônomo de Investimentos</b>	<b>31</b>
4.1	O Serviço Hare . . . . .	31
4.2	Visão Geral . . . . .	32
4.3	Módulo de Predição . . . . .	33
4.4	Módulo de Gerenciamento de Riscos . . . . .	35
4.5	Módulo do Agente . . . . .	37

4.5.1	Modelo de Alocação de Recursos . . . . .	37
4.6	Considerações Finais . . . . .	40
<b>5</b>	<b>Resultados Experimentais</b>	<b>41</b>
5.1	Metodologia . . . . .	41
5.1.1	Base de Dados . . . . .	42
5.2	Experimentos do Módulo Preditor . . . . .	42
5.2.1	Pré Processamento da Base . . . . .	43
5.2.2	Hiper-parametrização do Modelo . . . . .	44
5.2.3	Treinamento e Desempenho do Modelo . . . . .	48
5.2.4	Comparação com Modelos de Base . . . . .	52
5.3	Experimentos de Alocação de Recursos . . . . .	53
5.3.1	Alocação dos Ativo Individuais . . . . .	55
5.3.2	Alocação do Portfólio Completo . . . . .	56
5.4	Considerações Finais . . . . .	61
<b>6</b>	<b>Conclusão</b>	<b>63</b>
6.1	Trabalhos Futuros . . . . .	64
<b>Apêndice</b>		<b>69</b>
<b>A</b>	<b>Análise detalhada dos experimentos individuais de cada ativo</b>	<b>70</b>
A.1	Alocação do Ativo PETR3 . . . . .	70
A.2	Alocação do Ativo VALE3 . . . . .	72
A.3	Alocação do Ativo ABEV3 . . . . .	75
<b>B</b>	<b>Experimento extra com o Módulo de Gerenciamento de Riscos</b>	<b>78</b>

# Listas de Figuras

2.1	Diagrama do processo de compra e venda de ações na bolsa de valores. . . . .	7
2.2	Fronteira Eficiente de um conjunto de ativos. Adaptado de [3]. . . . .	8
2.3	Classificação do elemento $x_t$ através do algoritmo KNN. . . . .	12
2.4	Separação das classes num espaço de características de maior dimensão utilizando um hiperplano separador.. . . . .	13
2.5	Perceptron multicamadas classificando uma imagem de cachorro. Cada cor de camada escondida representa uma conceito da imagem sendo observada pelo modelo. Adaptado de [25]. . . . .	15
2.6	Topologia de uma unidade RNN simples. . . . .	16
2.7	Topologia de uma unidade LSTM. . . . .	18
2.8	Topologia de uma unidade GRU. . . . .	20
2.9	Interação básica agente-ambiente no aprendizado por reforço. . . . .	21
2.10	Diagrama de funcionamento do DDPG. Adaptado: . . . . .	24
4.1	Cenário Operacional do Serviço Hare. . . . .	32
4.2	Visão Geral do Módulo Preditor de Movimento. O modelo preditor escolhido pode ser alterado por qualquer modelo de aprendizagem desejado. . .	34
4.3	Visão Geral do Módulo de Gerenciamento de Riscos. O modelo preditor escolhido pode ser alterado por qualquer modelo de aprendizagem desejado.	37
4.4	Visão Geral do Modelo de Alocação de Recursos. O modelo preditor escolhido pode ser alterado por qualquer modelo de aprendizagem desejado. . .	38
5.1	Histograma de parâmetros selecionados e progressão do <i>F1 Score</i> no ativo PETR3. . . . .	47
5.2	Histograma de parâmetros selecionados e progressão do <i>F1 Score</i> no ativo VALE3. . . . .	47
5.3	Histograma de parâmetros selecionados e progressão do <i>F1 Score</i> no ativo ABEV3. . . . .	48
5.4	Função de perda e matriz de confusão do ativo PETR3. . . . .	49
5.5	Função de perda e matriz de confusão do ativo VALE3. . . . .	49

5.6	Função de perda e matriz de confusão do ativo ABEV3. . . . .	50
5.7	Predições do ativo PETR3. . . . .	51
5.8	Predições do ativo VALE3. . . . .	51
5.9	Predições do ativo ABEV3. . . . .	52
5.10	Portfólio - Retorno médio. . . . .	56
5.11	Portfólio - Comportamento do QValor. . . . .	57
5.12	Portfólio - Comportamento das funções de perda. . . . .	57
5.13	Portfólio - Lucro em treino. . . . .	57
5.14	Portfólio - Lucro médio em teste. . . . .	58
5.15	Portfólio - Quantidade de ações selecionadas em teste. . . . .	58
5.16	Fronteira eficiente do portfólio PETR3, VALE3, ABEV3. . . . .	59
A.1	PETR3 - Retorno médio. . . . .	70
A.2	PETR3 - Comportamento do QValor . . . . .	71
A.3	PETR3 - Comportamento das funções de perda . . . . .	71
A.4	PETR3 - Lucro em treino. . . . .	71
A.5	PETR3 - Lucro médio em teste. . . . .	72
A.6	PETR3 - Quantidade de ações selecionadas em teste. . . . .	72
A.7	VALE3 - Retorno médio. . . . .	73
A.8	VALE3 - Comportamento do QValor. . . . .	73
A.9	VALE3 - Comportamento das funções de perda. . . . .	73
A.10	VALE3 - Lucro em treino. . . . .	74
A.11	VALE3 - Lucro médio em teste. . . . .	74
A.12	VALE3 - Quantidade de ações selecionadas em teste. . . . .	74
A.13	ABEV3 - Retorno médio. . . . .	75
A.14	ABEV3 - Comportamento do QValor. . . . .	76
A.15	ABEV3 - Comportamento das funções de perda. . . . .	76
A.16	ABEV3 - Lucro em treino. . . . .	76
A.17	ABEV3 - Lucro médio em teste. . . . .	76
A.18	ABEV3 - Quantidade de ações selecionadas em teste. . . . .	76
B.1	Portfólio com MGR - Lucro médio em teste. . . . .	79
B.2	Portfólio com MGR - Quantidade de ações selecionadas. . . . .	79

# **Lista de Tabelas**

5.1	Parâmetros utilizados nos experimentos. . . . .	45
5.2	Melhor conjunto de parâmetros encontrados para cada ativo. . . . .	46
5.3	Resumo dos resultados obtidos. . . . .	51
5.4	Resumo comparativo dos métodos. . . . .	53
5.5	Parâmetros utilizados na DDPG. . . . .	54
5.6	Resultados obtidos nos investimentos realizados pelo Hare em comparação com popuança, CDI e estratégia <i>buy and hold</i> . . . . .	56
5.7	Comparação dos resultados obtidos pelo portfólio Hare com portfólios de fronteira eficiente. . . . .	60

# Listas de Abreviaturas e Siglas

**AG** Algoritmo Genético.

**CDI** Certificado de Depósito Interbancário.

**DDPG** Gradiente de Política Determinística Profunda.

**DTW** Distorção Dinâmica de Tempo.

**EQMB** Erro Quadrático Médio de Bellman.

**GRU** *Gated Recurrent Unit.*

**IA** Inteligência Artificial.

**KNN** k-Vizinhos Mais Próximos.

**LSTM** *Long Short-Term Memory.*

**MA** Módulo do Agente.

**MAR** Modelo de Alocação de Recursos.

**MGR** Módulo de Gerenciamento de Riscos.

**ML** Aprendizado de Máquina.

**MPM** Módulo Preditor de Movimento.

**RNA** Rede Neural Artificial.

**RNN** Rede Neural Recorrente.

**RNNs** Redes Neurais Recorrentes.

**SMBO** Otimização Sequencial Baseada em Modelo.

**SVM** Máquina de Vetores de Suporte.

**TPE** Tree Parzen Estimator.

**VMM** Variância Média de Markowitz.

# Listas de Símbolos

## Símbolos Latinos Minúsculos

<i>a</i>	Ação
<i>c</i>	Estado da Célula
<i>e</i>	Número de Euler
<i>h</i>	Estados Ocultos
<i>r</i>	Recompensa
<i>s</i>	Estado do Ambiente
<i>t</i>	Tempo
<i>w</i>	Peso ou Proporção

## Símbolos Latinos Maiúsculos

<i>A</i>	Espaço de Ações
<i>D</i>	Conjunto de Experiências
<i>E</i>	Valor Esperado
<i>L</i>	Função de Perda
<i>N</i>	Ruído de Exploração
<i>Q</i>	Valor do par estado-ação ou estado-politica
<i>R</i>	Retorno
<i>S</i>	Espaço de Estados
<i>T</i>	Intervalo de Tempo

## Símbolos Gregos

$\alpha$	Fator de aprendizado
$\gamma$	Fator de desconto
$\phi$	Função de Mapeamento
$\delta$	Função de Distância
$\sigma$	Função Sigmoide
$\kappa$	Função Kernel
$\lambda$	Multiplificador de Lagrange
$\tau$	Trajetória
$\mu$	Política Determinística
$\pi$	Política Estocástica
$\varphi$	Valor do Risco (Desvio Padrão)
$\rho$	Coeficiente de Correlação
$\theta$	Conjunto de Parâmetros

## Subscritos

$t$	Instante de Tempo
$P$	Portfólio
$\theta$	Conjunto de Parâmetros

## Sobrescritos

*	Valor Ótimo
'	Novo Valor

# Capítulo 1

## Introdução

Mercados financeiros assumem um papel essencial no comportamento da economia de um país [1, 2, 3]. Tais mercados tornam fácil para acionistas negociarem seus ativos financeiros, contribuindo para o crescimento das empresas e corporações, e gerando oportunidades para investidores possuírem rentabilidades sobre seus ativos [1]. O estudo desses mercados com o objetivo de realizar previsões sobre seu movimento, pode ser capaz de aumentar o lucro e a rentabilidade dos investidores, além de prover um melhor entendimento das movimentações e valores de um determinado ativo em avaliação [4]. Sendo assim, serviços de negociações baseados em modelos de decisão estão ganhando espaço no mercado financeiro nacional e internacional [4].

Entre os vários tipos de mercados financeiros em que o investidor pode alocar os ativos, o mercado de ações se destaca em termos de popularidade [1]. Esse tipo de mercado negocia frações das empresas e corporações, denominadas ações<sup>1</sup>, ou diversos outros tipos de instrumentos financeiros, tais como: câmbio, ouro e commodities [3]. Os investidores que negociam ativos no mercado normalmente são guiados por alguma forma de previsão gerada por uma análise oportunista do meio em que estão inserido. Alguns tipos comuns de análises realizadas são: análise da situação do governo, da situação do país, da situação da empresa detentora dos ativos, ou até mesmo das variações de preço durante um intervalo de tempo [4].

Considerando o potencial que as técnicas de Aprendizado de Máquina (ML) podem oferecer, seu uso é um dos caminhos viáveis para prover serviços relacionados com a previsão de um ativo no mercado financeiro. O processo de prever movimento dos mercados é uma área de pesquisa em crescimento dentro dos estudos de aprendizado de máquina

---

<sup>1</sup>Quando um número específico de ações estão sendo negociadas elas são denominadas de posições, é dito que o investidor está comprando ou vendendo posições das ações de uma empresa. Por outro lado, quando um número indeterminado de ações estão sendo negociadas, a terminologia utilizada é de nomear essas ações de ativos [1].

[4]. Esse crescimento pode oferecer possibilidades para todos que buscam obter lucro e rentabilidade com suas finanças.

Estudos recentes vêm tentando prever os mercados financeiros utilizando diversas abordagens e modelos de aprendizagem de máquina [5, 6, 7, 8, 9, 10, 11]. Tais abordagens são divididas em duas categorias principais: (i) *análise fundamental*, na qual o valor da companhia que é responsável por definir o preço da ação, e não a ação em si; e (ii) *análise técnica* na qual a predição do preço futuro de uma ação é realizada se baseando no estudo de seus preços e indicadores passados e presentes [12].

Nos trabalhos analisados, percebe-se que alguns estudos são baseado em técnicas de *análise de sentimento* com o objetivo de prever o valor de um ativo [5, 4]. Outros trabalhos utilizam o comportamento de variáveis macroeconômicas<sup>2</sup> ou índices de mercados ao redor do mundo [13, 14, 4]. Entretanto, em grande maioria os trabalhos utilizam dados técnicos dos ativos em formato de séries temporais para criar modelos ou indicadores técnicos [15, 6, 16, 17]. Ainda, uma grande parcela dos trabalhos encontrados na literatura não utilizam predições em cima de ações de empresas e corporações disponíveis no mercado, preferindo outros tipos de ativos financeiros, tais como: cripto-moedas[18], moedas estrangeiras [19, 16], índices de mercados [14, 20, 17] e commodities [19]. Além disso, quando um estudo é de fato focado no mercado de ações é, usualmente, baseado nos mercados Asiáticos e Europeus [4]. Não obstante, todos os trabalhos mencionados não exploram serviços híbridos baseados em análise técnica em conjunto com a análise fundamental, além de não prover um mecanismo de agir em cima das decisões (mesmo fazendo um esforço considerável pra prever os mercados financeiros).

Enquanto a maioria das pessoas na área acreditam que os mercados de alguma maneira são previsíveis, também existem opiniões contrárias [4]. Alguns pesquisadores acreditam na hipótese do mercado eficiente, que afirma que o preço dos ativos é um reflexo de toda informação acessível daquele ativo naquele instante de tempo. Dessa forma, as informações do presente momento não alterariam o futuro de um ativo e, por consequência, a previsão de retornos futuros não seria possível [21]. Portanto, para o desenvolvimento deste trabalho, não será considerado tal hipótese, visto que se ela fosse verdade, não faria sentido realizá-lo. Sendo assim, superando os desafios e limitações supracitadas, este trabalho se sustenta na hipótese de que é possível criar um serviço confiável baseado em análises técnicas e fundamentais para negociar ativos no mercado de ações com alta precisão e estabilidade no serviço proposto.

---

<sup>2</sup>Medidas que indicam as variáveis agregadas de todo o país

## 1.1 Objetivos

Com isso em mente, este trabalho vai além e propõe o Hare<sup>3</sup> um serviço híbrido, autônomo, orientado a agente para negociar ativos no mercado financeiro desejado. O Hare é baseado em agentes racionais, entidades computacionais capazes de observar o mercado e agir de forma autônoma em cima de suas percepções. Racionalidade para o agente, é definida neste escopo, como a capacidade do agente escolher as ações que maximizarão seu lucro e diminuirão seus riscos. Para atingir esse objetivo, o Hare foi projetado em uma estrutura modular que contém: (i) um módulo preditor utilizando uma unidade de redes neurais recorrentes baseada em portões chamada *Long Short-Term Memory* (LSTM), juntamente com uma biblioteca de hiper-parametrização denominada Hyperopt; (ii) um módulo de controle de risco utilizando informações de busca do *Google Trends*<sup>4</sup>, informações de notícias e relatórios financeiros; e (iii) um módulo atuador racional treinado utilizando um algoritmo de aprendizado por reforço denominado Gradiente de Política Determinística Profunda, que tem como objetivo gerenciar os recursos disponíveis ao agente. Portanto, o Hare provê um serviço de ponta-a-ponta para investimento em ações na bolsa de valores, da análise dos dados até as ordens de negociação na corretora.

## 1.2 Contribuição

Com os objetivos do Hare bem definidos, as principais contribuições deste trabalho, em comparação com outros trabalhos na literatura, são destacadas a seguir:

1. O modelo dos agentes possui módulos preditores especializados: cada um desses módulos são treinados em uma ação específica, do conjunto de ações escolhidos pelo usuário em seu portfólio.
2. As análises de movimento do ativo são híbridas: utilizando análise fundamental para administrar os riscos de um ativo e análise técnica, baseada em séries históricas, para prever a direção do ativo.
3. Os módulos são adaptáveis: podem trocar os seus modelos de aprendizado e provendo alta versatilidade nos experimentos dos modelos.
4. O serviço aloca recursos no mercado: a estrutura do serviço não para na predição, sendo também responsável por gerenciar recursos com conhecimento aprendido por métodos de aprendizagem por reforço

---

<sup>3</sup>O nome escolhido significa Lebre em inglês. O motivo de escolha desse animal para nomear o sistema é dado pelo senso comum de que a Lebre é um animal rápido e ágil em suas movimentações e decisões, características que o serviço proposto busca promover.

<sup>4</sup><https://trends.google.com>

### **1.3 Estrutura do Trabalho**

A estrutura deste trabalho está organizada do seguinte modo. O Capítulo 2 estabelece a base teórica necessária para a compreensão do desenvolvimento do trabalho, apresentando os conceitos básicos de economia e mercado, e conceitos da área de aprendizado de máquina inerentes a aplicação do serviço. O Capítulo 3 apresenta os trabalhos relacionados no campo de predição de mercados financeiros, que utilizam análises fundamentais e técnicas em suas abordagens. No Capítulo 4 o Hare é apresentado, com foco em sua implementação como serviço de investimento autônomo, racional e baseado em agentes. Posteriormente, no Capítulo 5 é apresentada a metodologia para experimentação dos módulos do Hare, responsáveis pela sua validação. Por fim, o Capítulo 6 apresenta as conclusões e as oportunidades para trabalhos futuros.

# **Capítulo 2**

## **Fundamentação Teórica**

Neste capítulo, apresentam-se os conceitos básicos de economia e mercado, discorrendo sobre o mercado de ações e a teoria moderna de portfólio. Em seguida, discutem-se conceitos relacionados a área de aprendizado de máquina, apresentando juntamente algoritmos relevantes para este trabalho. Posteriormente é apresentado o conceito de aprendizado profundo e seus algoritmos utilizados. Por fim, discorre-se sobre o aprendizado por reforço com técnicas de aprendizado profundo.

### **2.1 Economia e Mercado**

A riqueza material de uma sociedade é determinada pela capacidade produtiva de sua economia, representada pelos bens e serviços que fornece aos membros da mesma [1]. Essa capacidade produtiva, ou produtividade, é definida diretamente pelos chamados ativos reais, tais como construções, conhecimento, terras, máquinas e qualquer elemento da sociedade que são necessários para gerar esses bens [1].

Em contraste aos ativos reais existem os ativos financeiros. Tais ativos não possuem a capacidade de representar a riqueza de uma sociedade, contudo conseguem afetar a produtividade de maneira indireta [1]. Ações e títulos são os exemplos mais simples de ativos financeiros. Esses conseguem definir os donos de propriedades e frações de empresas, gerando assim uma facilidade na transação de recursos entre os proprietários de um ativo real [1].

Em suma, ativos reais produzem bens e serviços, enquanto ativos financeiros definem alocação de riqueza e lucro entre investidores [1]. No entanto, ativos financeiros representam posses de frações de ativos reais. Com exemplo, quando o dinheiro que é investido em uma firma, ele é convertido em ativos reais. Quando tal investimento oferecer um retorno, ele pode ser retribuído em ativos financeiros através de títulos que representam parte da empresa. Os títulos, consequentemente, representam frações de ativos reais. Portanto,

ativos financeiros e os mercados nos quais são negociados preenchem um papel importante no desenvolvimento da economia [1].

### 2.1.1 O Mercado de Ações

Assim como as instituições financeiras e seguradoras surgiram naturalmente a partir das necessidades de investidores, o mercado financeiro não é diferente [1]. A necessidade de uma instituição para unir, em um mesmo lugar, interessados em negociar ativos financeiros fez surgir os mercados de ações e a bolsas de valores [1].

Chama-se de mercado toda interação entre dois ou mais interessados que se identifiquem como compradores e vendedores. Dentre os tipos de mercado podemos dividi-los em quatro categorias: (i) o mercado de pesquisa direta, em que os compradores e vendedores precisam se procurar por conta própria; (ii) o mercado intermediado, no qual existe um intermediador, entre o comprador e o vendedor, que obterá algum lucro na negociação; (iii) o mercado de revendedores em que o intermediador compra os ativos e revende para os vendedores; e por último (iv) o mercado de leilão, no qual todos os transacionistas se reúnem em um mesmo local para fazerem ofertas [1]. O mercado de leilão é o melhor exemplo de mercado financeiro moderno, criando a vantagem de reunir todas as ofertas em um mesmo lugar sem a necessidade de pesquisas por preços mais baratos [1].

Dentre os mercados de leilão, existem os mercados de ações, comumente chamados de bolsa de valores. Em geral, a bolsa de valores é um mercado no qual são efetuadas ações de compra e venda de valores mobiliários (ações, títulos, derivativos, e entre outros) [2]. Tais valores podem ser vistos como contratos legais que representam o direito de receber um benefício futuro [3]. Neste trabalho destacaremos dois principais valores mobiliários: ações preferenciais e ações ordinárias [3]. Ações ordinárias representam o direito de um investidor sobre os rendimentos ou ativos de uma corporação, possuindo como principal característica a responsabilidade limitada de seus detentores [1]. Se a corporação for a falência o máximo que o investidor de ações ordinárias pode perder é o seu investimento inicial, não tendo acesso aos ativos gerais da empresa. Já as ações preferenciais são semelhantes com as ações ordinárias exceto pela sua prioridade nos pagamentos dos dividendos<sup>1</sup> da empresa. Caso a empresa vá a falência todos os acionistas com ações preferenciais devem receber os dividendos antes dos pagamentos de dividendos de ações ordinárias [3].

As ações, tanto ordinárias como preferenciais, possuem suas cotas negociadas na bolsa de valores seguindo uma mecânica bem definida, como apresentada na Figura 2.1. Um indivíduo que deseja comprar ou vender um ação necessita antes contratar uma corretora, uma entidade que possui permissão para emitir ordens em uma bolsa de valores. Uma

---

<sup>1</sup>Parte dos lucros de uma empresa que são distribuídos aos seus acionistas como forma de remuneração.

ordem é responsável por anexar as informações de compra ou venda de uma ação. Nas informações da ordem há: nome do negociante e qual ativo deseja negociar, se é uma compra ou uma venda, a quantidade de ativos a serem negociados e o prazo que a ordem permanecerá válida. Assim que a ordem é emitida e executada a corretora é responsável por transacionar o dinheiro entre o indivíduo e a bolsa de valores [3].

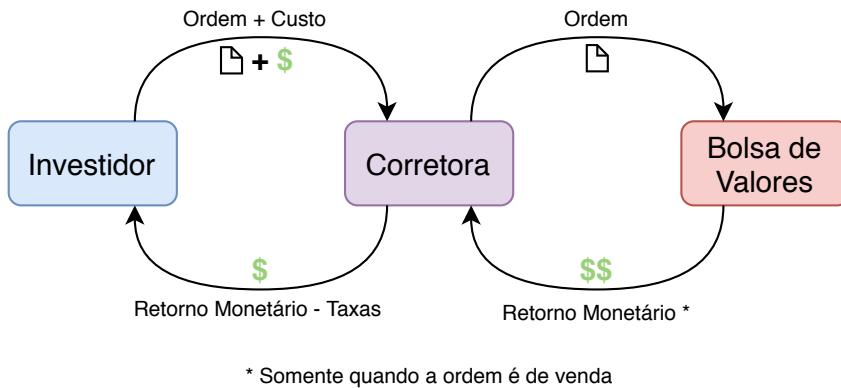


Figura 2.1: Diagrama do processo de compra e venda de ações na bolsa de valores.

### 2.1.2 Teoria Moderna do Portfólio

Um indivíduo que possui riquezas, pode ter seus bens referidos como um conjunto de ativos que possui, seja eles reais ou financeiros. Por exemplo, um investidor que possui imóveis e títulos do tesouro direto, os imóveis representam seus ativos reais e os títulos seus ativos financeiros, juntos correspondem a um total de riqueza. Esse conjunto também pode ser chamado de portfólio ou carteira [3]. A composição dos portfólios são resultados de decisões casuais ou podem ser fim de um planejamento pré-estabelecido [3].

A criação de portfólios planejados com premissa nos conceitos de otimização e diversificação da carteira, foram pilares no desenvolvimento e entendimento dos mercados financeiros [22]. Em 1952 Markowitz [23], publicou o artigo *Portfolio Selection* que viria a ser um dos maiores avanços na criação de portfólios [22]. Sua proposta ficou conhecida como a Teoria Moderna do Portfólio, ou Variância Média de Markowitz (VMM) [23]. A teoria busca responder a questão fundamental de como um investidor deve alocar seus fundos dentre as diversas possibilidades de escolha de investimentos.

Markowitz aderiu essa questão com duas etapas. Primeiro ele quantificou o retorno e risco de um valor mobiliário [22]. Depois, propôs que os investidores considerassem o retorno e risco de forma conjunta, determinando a alocação dos fundos com base no balanceamento entre os fatores [22]. Portanto, os fatores devem ser calculados em relação ao portfólio como um todo, e não somente ao ativo específico. O retorno do portfólio

$R_p$  foi quantificado de acordo com a medida estatística de retorno esperado demonstrada na Equação 2.1, onde  $R_i$  é o retorno do ativo  $i$ , e  $w_i$  a proporção do ativo  $i$  na carteira [3]. Enquanto o risco da carteira  $\varphi_p$  foi medido pelo seu desvio padrão, demonstrado na Equação 2.2, onde  $\varphi_i^2$  é a variância do retorno do ativo  $i$ ,  $\varphi_j^2$  a variância do retorno do ativo  $j$ , e  $\rho_{ij}$  o coeficiente de correlação entre os ativos  $i$  e  $j$  [3]. O coeficiente  $\rho$  é limitado entre  $[-1, 1]$ , onde 1 significa que os dois ativos vão se mover sempre em equilíbrio, enquanto o valor  $-1$  significa que seus movimentos são exatamente opostos.

$$E(R_p) = \sum_i w_i E(R_i) \quad (2.1)$$

$$\varphi_p = \sqrt{\sum_i w_i^2 \varphi_i^2 + \sum_i \sum_{j \neq i} w_i w_j \varphi_i \varphi_j \rho_{ij}} \quad (2.2)$$

O entendimento de que o processo de tomada de decisão financeira sólida é realizada como um balanceamento entre retorno e risco foi revolucionária [22]. O princípio mais inovador foi o da diversificação do portfólio [22]. Com esse princípio, o risco de um portfólio pode passar a ser analisado de acordo com a correlação dos seus constituintes, e não apenas o risco do ativo independentemente. Esse conceito era estranho à análise financeira clássica, que girava em torno da crença de que os investidores deveriam investir naqueles ativos que oferecem o maior valor futuro, dado seu preço atual [22]. A ideia de Markowitz, foi inovadora também por passar a formular o problema de tomadas de decisões financeiras, em um problema de otimização [22]. Em particular, o modelo de otimização da VMM propõe que entre os números infinitos de portfólio que alcançam um retorno pré-definido, o investidor deve escolher aquele que tem o menor risco.

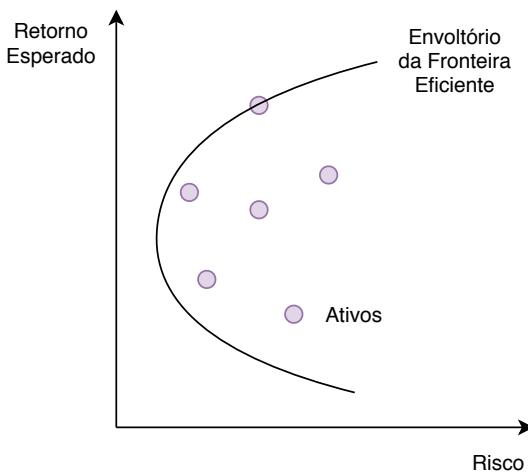


Figura 2.2: Fronteira Eficiente de um conjunto de ativos. Adaptado de [3].

O problema de otimização da carteira pode ser visto com uma interpretação geométrica das combinações de ativos [3]. Ao realizar a análise, pode-se desenhar o subconjunto de carteiras que serão as preferidas de todos os investidores com aversão a risco [3]. Tal conjunto, denominado de fronteira eficiente, é o envoltório externo de todas as possíveis carteiras, entre a carteira de menor risco (ou de variância mínima global) e a carteira de máximo retorno [3]. A Figura 2.2 ilustra a interpretação geométrica da otimização da carteira, e a fronteira eficiente. Cabe então ao investidor decidir, dado o balançamento entre retorno e risco que ele deseja assumir, em que ponto da fronteira seu portfólio será montado.

## 2.2 Aprendizado de Máquina

Durante milhares de anos os seres humanos vêm procurando desvendar como funcionam seus pensamentos, como um punhado de matéria pode perceber, compreender, prever e manipular um mundo muito maior que ele mesmo [24]. O campo da Inteligência Artificial (IA) estende o desejo de desvendar e entender o pensamento humano, indo além, construindo entidades inteligentes [24]. O desejo de construir tais entidades capazes de pensar vem desde a Grécia antiga, que apresenta diversos inventores e vidas artificiais em sua mitologia [25]. Até o momento de escrita deste trabalho a IA vem se mostrando uma subárea da ciência da computação próspera, com diversas aplicações práticas e tópicos de pesquisa [25, 17, 13, 16, 26, 8, 27, 11]. Algumas de suas aplicações são realizações de tarefas, tais como: jogos eletrônicos [27], demonstração de teoremas matemáticos [28], criação de música [29], controle robótico [30], e diagnóstico de doenças [31].

Pode-se definir a IA de diversas maneiras: em termos de processos de pensamento e raciocínio, comportamento, fidelidade ao desempenho humano, ou racionalidade [24]. Para o escopo deste trabalho, a IA foi conceituada como o processo de agir com racionalidade, definido por Poole, Mackworth e Goebel [32], “Inteligência computacional é o estudo do projeto de agentes inteligentes”. Segundo Russell e Norvig [24], um agente é tudo que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre o ambiente por meio de atuadores. Para um agente humano, os olhos, ouvidos e outros sentidos são os sensores que percebem o mundo, já suas pernas, mãos bocas e outras partes do corpo são atuadores que interagem com ele. Um agente é dito inteligente, ou racional, quando realizam as melhores ações para maximizar alguma medida de desempenho pré-definida para o problema sendo tratado [24]. Como exemplo, considere um agente investindo na bolsa de valores, sua medida de desempenho pode ser considerada o lucro ganho com seus investimentos. Esse agente é considerado racional caso a ação realizada possa vir a maximizar o seu lucro.

Em diversos cenários, como o exemplo citado, um comportamento considerado bom em determinado momento pode deixar de ser em um outro contexto ou situação. Sendo assim, um agente racional necessita aprender a se adaptar em situações adversas [24]. Dizemos que o agente aprende, quando melhora o seu desempenho nas tarefas futuras após observar informações atuais disponíveis a ele [24]. Ainda, no contexto de agente inteligente, o aprendizado apresenta duas vantagens, sendo elas: (i) um programador não consegue antecipar todas as situações possíveis que um agente pode encontrar; e (ii) as vezes o programador não tem ideia de como resolver o problema, e deixa a tarefa de descoberta para o agente [24].

Diante do exposto anteriormente, os algoritmos de Aprendizado de Máquina (ML) surgem como um meio para auxiliar o agente a adquirir o próprio conhecimento para aprender [33]. O surgimento de algoritmos de ML permitiu computadores abordarem problemas envolvendo conhecimento do mundo real e tomar decisões que pareçam subjetivas [25]. De maneira direta, aprender é o processo de converter experiências em conhecimento. A entrada de um algoritmo de aprendizado são dados de treinamento, representando a experiência, e a saída é um conhecimento sobre a tarefa que está sendo aprendida [34].

Em ML, existem diversos tipos de tarefas para a entidade aprendiz. Essas tarefas podem ser agrupadas de acordo com o paradigma de aprendizado para lidar com ela, podendo ser preditiva ou descriptiva [35]. Uma tarefa preditiva busca encontrar uma função que faz o mapeamento dos dados de entrada para obter uma predição como saída [24]. Algoritmos que seguem o modelo preditivo são do paradigma de aprendizado supervisionado. O termo supervisionado, se relaciona com a necessidade de existir um “supervisor externo”, também conhecido como especialista, que conhece a resposta da tarefa (rótulo), para analisar o desempenho do aprendiz [35]. Por outro lado, em uma tarefa descriptiva o objetivo é explorar ou descrever um conjunto de dados. Algoritmos que se encaixam nesse tipo de classificação de tarefa são do paradigma de aprendizado não supervisionado [35]. Esses algoritmo não são fornecidos um rótulo [24], e portanto, não necessitam de uma supervisão externa. Outro paradigma de aprendizado, é o de aprendizado por reforço. Em tal paradigma a meta é reforçar ou recompensar de forma positiva ou negativa uma ação realizada pela entidade que está aprendendo [35]. Essa categoria de aprendizado será apresentado na Subseção 2.4.

Tem-se como escopo deste trabalho resolver tarefas de predição, logo discorre-se majoritariamente sobre o aprendizado supervisionado. Um algoritmo supervisionado é uma função que dado um conjunto de dados de entrada rotulado, constrói um estimador [35]. Os rótulos tomam valores em um domínio conhecido. Se o domínio conter valores nominais, subclassificamos a tarefa como classificação [24]. Caso o contrário, os rótulos possuírem um conjunto infinito e ordenado de valores tem-se um problema de regres-

são [35]. Com um exemplo prático, uma predição na bolsa de valores pode se tomar de ambas as formas. Se deseja-se descobrir se uma determinada ação vai ganhar ou perder valor, temos um problema de classificação. Se deseja descobrir uma estimativa do preço de uma ação, o problema toma a forma de regressão, porque o domínio do valor da ação é de qualquer valor real positivo. Com o objetivo de realizar uma tarefa de predição, diversos algoritmos de aprendizado supervisionado foram criados. Para este trabalho, foram explorados os algoritmos k-Vizinhos Mais Próximos e Máquina de Vetores de Suporte que são discutidos nas Subseções 2.2.1 e 2.2.2.

### 2.2.1 k-Vizinhos Mais Próximos

No aprendizado supervisionado, a simplicidade do k-Vizinhos Mais Próximos (KNN) [36] se destaca entre os modelos, sendo um dos algoritmos mais usados no problema de classificação [37]. O KNN classifica objetos desconhecidos com base na similaridade com outros objetos já conhecidos [37]. Tais objetos são definidos pelos seus atributos e representados como pontos em um espaço, nos quais suas similaridades são comumente definidas como a distância euclidiana entre os pontos [37]. Portanto, quando o algoritmo recebe uma instância não conhecida  $x_t$ , o KNN calcula a distância entre  $x_t$  e todos os outros pontos presentes no espaço. O rótulo selecionado para  $x_t$  será o mesmo do objeto  $x_i$  onde  $x_i$  possui a menor distância euclidiana até  $x_t$  [35].

O KNN não precisa necessariamente definir a similaridade com a distância euclidiana [37]. Outras abordagens podem apresentar melhor desempenho alterando a forma de medir essa similaridade. Como por exemplo, o Distorção Dinâmica de Tempo (DTW)[38], utilizada principalmente pela sua eficiência como séries temporais pela sua capacidade de mover e distorcer os dados a fim de facilitar o encontro de padrões similares [39]. Tendo isso em vista pode-se definir qualquer métrica de similaridade que seja uma função real  $\delta$  de modo que para qualquer coordenadas  $x$ ,  $y$  e  $z$  as propriedades definidas pela Equação 2.3, 2.4 e 2.5 se satisfazem [37].

$$\begin{aligned} \delta(x, y) &\geq 0 \\ \delta(x, y) = 0 &\iff x = y \end{aligned} \tag{2.3}$$

$$\delta(x, y) = \delta(y, x) \tag{2.4}$$

$$\delta(x, z) \leq \delta(x, y) + \delta(y, z) \tag{2.5}$$

A propriedade da Equação 2.3 garante que a distância sempre será não negativa e que a única maneira de ser zero é caso as coordenadas sejam as mesmas. Já a propriedade da Equação 2.4 indica que a distância de  $x$  para  $y$  deve ser a mesma de  $y$  para  $x$ . Por

último, a Equação 2.5 representa a desigualdade triangular, que define que a introdução de um terceiro ponto não pode reduzir a distância entre outros dois pontos.

Salienta-se, entretanto, que o KNN considera  $k$  vizinhos próximos como base para rotular a instância nunca vista, gerando uma votação entre os vizinhos mais próximos para decidir o rótulo da entrada teste [35]. Considere a Figura 2.3 como exemplo, dado um valor de entrada  $x_t$  o algoritmo seleciona os  $k$  vizinhos (no exemplo,  $k = 5$ ) mais próximos de  $x_t$ . Com isso, cada vizinho vota em uma classe e  $x_t$  é classificado segundo a classe mais votada. Tal votação pode ser formalmente definida pela Equação 2.6, em que a função de moda retorna o elemento que mais se repete, representando a classe mais votada.

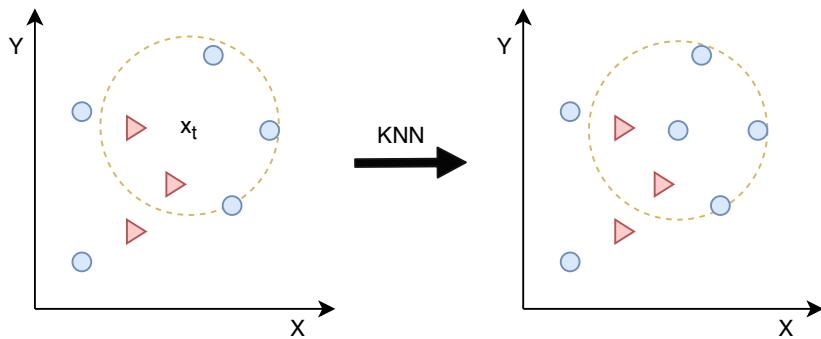


Figura 2.3: Classificação do elemento  $x_t$  através do algoritmo KNN.

O método de votação para classificar uma entrada não se restringe somente a moda, o KNN também pode ser usado em problemas de regressão, o que trás duas outras maneiras de se abordar a classe mais votada, em relação a função de perda que se deseja minimizar [35]. Caso a função a ser minimizada seja o erro quadrático o valor do rótulo de  $x_t$  torna-se a média dos  $k$  vizinhos mais próximos, formalmente dado pela Equação 2.7 [37]. Para o caso de que a função a ser minimizada é o desvio padrão utiliza-se a mediana como métrica de votação, sendo formalizada pela Equação 2.8.

$$f'(x_t) = \text{moda}(f(x_1), f(x_2), \dots, f(x_k)) \quad (2.6)$$

$$f'(x_t) = \text{média}(f(x_1), f(x_2), \dots, f(x_k)) \quad (2.7)$$

$$f'(x_t) = \text{mediana}(f(x_1), f(x_2), \dots, f(x_k)) \quad (2.8)$$

## 2.2.2 Máquina de Vetores de Suporte

Uma Máquina de Vetores de Suporte (SVM) [40] é um modelo de aprendizado supervisionado que mapeia pontos de um espaço de entrada, para um espaço de características.

O SVM realiza o aprendizado gerando um plano para separar classes de dados e realizar a tomada de decisões, denominado hiperplano separador [41]. A ideia central, é separar classes de dados utilizando o hiperplano separador, de tal forma que os elementos de classes opostas, mais próximos entre si, fiquem o mais distante possível dos planos gerados [42].

Dependo da dimensão dos dados de entrada o hiperplano irá assumir uma dimensão a menos do que a dimensão dos dados. Por exemplo, em dados com duas dimensões o hiperplano separador terá uma única dimensão, sendo representado por uma linha. A lógica se mantém para dimensões superiores, se o espaço for tridimensional o hiperplano separador será um plano de duas dimensões [42]. Entretanto, não é sempre que o problema apresenta dados linearmente separáveis sendo necessário mapear o espaço de entrada para dimensões superiores a fim de encontrar melhores hiperplanos.

Na Figura 2.4, é mostrado o processo de encontro do hiperplano ótimo para uma entrada de dados que não é linearmente separável. Os dados são mapeados para um espaço de características de maior dimensão através da função  $z = x^2 + y^2$ . Na Figura 2.4 torna-se visível que o mapeamento do espaço de entrada para espaço de característica cria uma melhor distribuição dos dados para traçar o hiperplano.

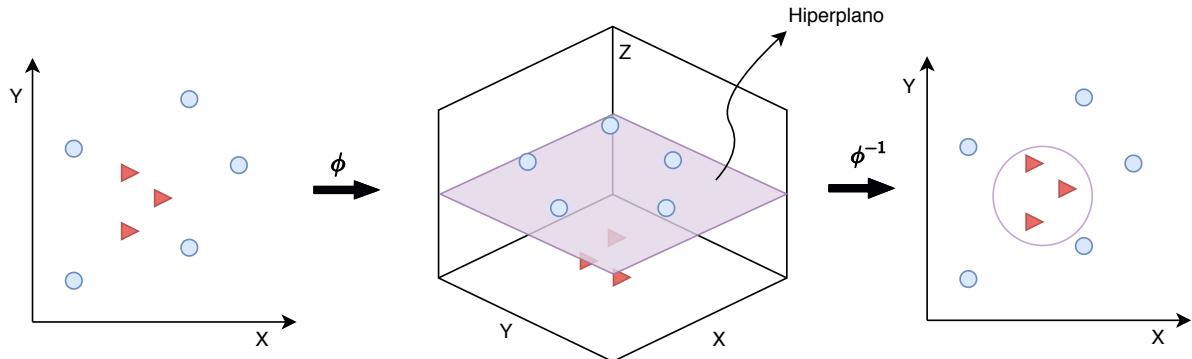


Figura 2.4: Separação das classes num espaço de características de maior dimensão utilizando um hiperplano separador..

Entretanto, calcular o mapeamento dos dados para maiores dimensões pode ser extremamente custoso em termos computacionais. A complexidade de tal transformação é quadrática,  $O(n^2)$ , tornando o algoritmo pouco eficiente para dados de grandes dimensões [42]. Como solução pra essa limitação, Boser, Guyon e Vapnik [40] apresentam o “truque” do *kernel*, uma técnica que faz uso de funções *kernel* que permitem descobrir o produto interno entre dois vetores em dimensões maiores, sem a necessidade de utilizar a função de mapeamento  $\phi(x)$ , para o espaço de características. Formalmente o *kernel* é dado por  $\kappa(x_i, x_j)$ , onde  $x_i$  e  $x_j$  são vetores do espaço de entrada [41]. Tal função é escolhida a *priori* e define como a classificação se comportará, entre eles os mais comuns são: (i)

kernel linear, Equação 2.9; (ii) kernel polinomial Equação 2.10, onde  $d$  é a dimensão da entrada; e (iii) kernel RBF gaussiano, Equação 2.11.

$$\kappa(x_i, x_j) = x_j \cdot x_i \quad (2.9)$$

$$\kappa(x_i, x_j) = (1 + x_i \cdot x_j)^d \quad (2.10)$$

$$\kappa(x_i, x_j) = e^{(-\|x_i - x_j\|^2)} \quad (2.11)$$

A SVM utiliza uma função chamada função de decisão para efetuar a classificação, entre os rótulos  $-1$  e  $1$ , de dados não rotulados. A definição da função se dá pela Equação 2.12, onde  $m$  é o tamanho do vetor de entrada,  $x_t$  é o vetor de entrada,  $x_i$  é o iésimo vetor de suporte,  $y = \pm 1$  é o rótulo padrão,  $b$  é o parâmetro inicial do hiperplano,  $\lambda_i$  é o iésimo multiplicador de Lagrange para o hiperplano ótimo,  $\kappa(x_t, x_i)$  é a função kernel e a função *sinal* retorna  $-1$  para valores abaixo de  $0$ ,  $1$  para valores maiores que  $0$  e zero para o valor  $0$ .

$$f(x_t) = \text{sinal} \left( \sum_{i=1}^m y_i \lambda_i \kappa(x_t, x_i) + b \right) \quad (2.12)$$

## 2.3 Aprendizado Profundo

A solução de diversas classes de problemas vem do entendimento e da experiência da adversidade em termos de uma hierarquia de conceitos. Cada conceito é definido em uma relação com conceitos mais simples, permitindo assim se aprender conceitos complexos [43]. Essa abordagem de transformar conceitos complexos em simples, formando assim profundas camadas conceituais hierárquicas, é característica do aprendizado profundo [25].

O problema central de representação do conhecimento, é resolvido no aprendizado profundo com a introdução de representações expressas em termos de outras mais simples [43]. Seu exemplo típico é o Perceptron Multicamadas, uma função matemática que mapeia um conjunto de valores de entrada para valores de saída. Tal função é formada por diversas outras funções mais simples, podendo se pensar como novas representações do valor de entrada. Considerando um exemplo prático, a Figura 2.5 representa um perceptron multicamadas para classificar uma imagem em cachorro, gato ou humano.

Na Figura 2.5, o objetivo é mapear uma imagem em uma das três categorias apresentadas. O aprendizado profundo realiza essa tarefa quebrando a imagem complexa e avaliando mapeamentos aninhados mais simples. Cada um dos mapeamentos é denominado de camada. A primeira camada é a de entrada, ou camada visível, é nela que os dados são fornecidos ao modelo. As camadas subsequentes são denominadas camadas es-

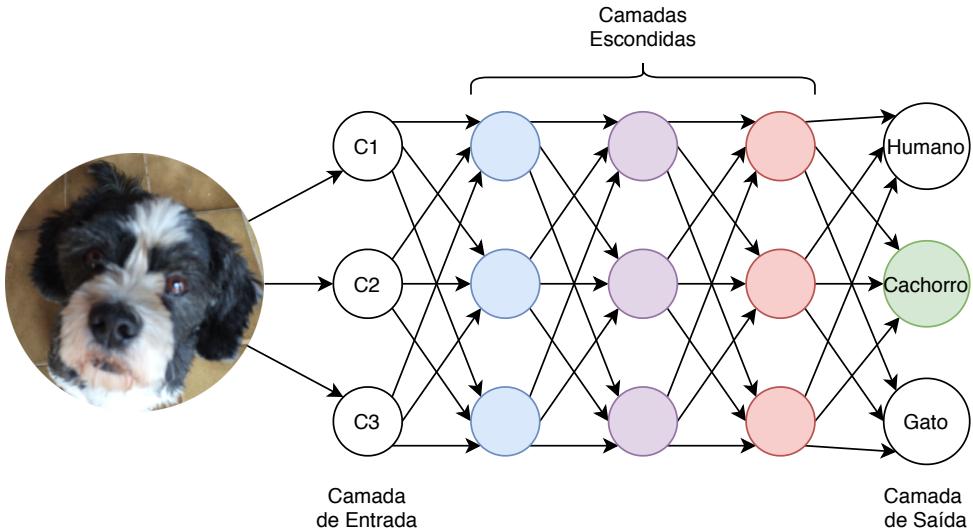


Figura 2.5: Perceptron multicamadas classificando uma imagem de cachorro. Cada cor de camada escondida representa uma conceito da imagem sendo observada pelo modelo. Adaptado de [25].

condidas, cada uma extraiendo conceitos cada vez mais abstratos da entrada (no exemplo proposto, da imagem). Essas camadas são denominadas escondidas, porque seu valor não é fornecido pelo dado de entrada, sendo responsabilidade do modelo determiná-las. Após todas as camadas calcularem seus valores, a camada de saída vai ativar o resultado do que está sendo avaliado.

Esse processo de ativação acontece devido a um mecanismo denominado função de ativação. Tais funções são responsáveis por determinar quando uma camada vai ativar ou não cada elemento da rede, dependendo da importância dele para avaliação da entrada atual. As funções de ativação são computacionalmente eficientes, e ajudam a normalizar a saída em um intervalo de  $-1$  a  $1$  ou  $0$  a  $1$ . Em aplicações modernas de aprendizagem profunda, as funções de ativação assumem formas não lineares e suas equações variam dependendo da aplicação.

O aprendizado profundo, auxilia a resolver problemas de diversas áreas, e de diversos formatos de dados. Uma das principais áreas abordada pelo aprendizado profundo, é o estudo de dados que foram observados em diferentes momentos no tempo [44]. A análises de séries temporais estudam a matemática e estatística das correlações temporais dos dados, se concentrando na modelagem de algum valor futuro como uma função paramétrica dos valores atuais e passados [44]. Gera-se então, resultados no domínio do tempo como uma ferramenta de previsão [44]. Alguns exemplos de problemas que podem ser estudados com séries temporais são: aquecimento global [45], reconhecimento de texto [46], e a bolsa de valores [10].

No entanto, o estudo de séries necessita de um conceito chave de aprendizado de máquina: a possibilidade de compartilhar parâmetros entre diferentes partes do modelo [25]. Em abordagens com parâmetros não compartilhados, não conseguiria-se generalizar sequências que não foram observadas durante o processo de aprendizagem para cada valor de tempo [25]. Quando um parâmetro é compartilhado, torna-se possível a extensão e aplicação de um modelo em dados de diferentes formas, e realizar generalizações através deles [25]. Tal compartilhamento é particularmente importante quando um pedaço específico de informação pode acontecer em múltiplas posições dentro de uma sequência. Do ponto de vista de aprendizagem de máquina, a abordagem do compartilhamento de parâmetros é aplicada com o auxílio de redes recorrentes e recursivas. Particularmente com uma classe das redes neurais para processar dados sequenciais denominada Rede Neural Recorrente.

### 2.3.1 Redes Neurais Recorrentes

O processo de aprendizado, seja humano ou de máquina, necessita da capacidade de processar dados de forma sequencial. Sem esse processamento, o ser humano não seria capaz de entender um texto nem assistir um filme sem perder o contexto da situação [25]. Redes neurais tradicionais, como o Perceptron, não são capazes de processar sequências de dados [43]. No entanto, com a ideia do compartilhamento de parâmetros, a classe das Redes Neurais Recorrentes (RNNs) [47] busca resolver essa limitação. A Figura 2.6 ilustra uma estrutura básica de um dos modelos mais simples da classe, a topologia de uma unidade RNN.

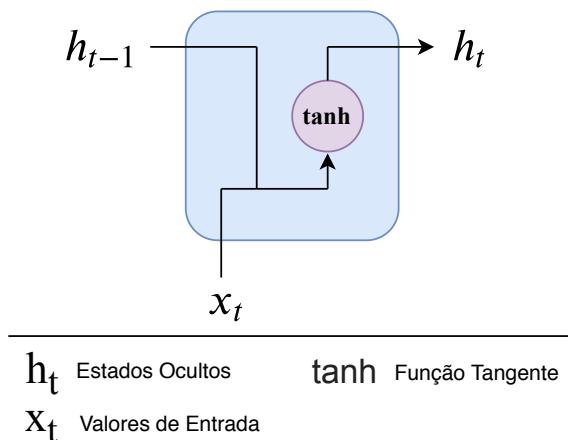


Figura 2.6: Topologia de uma unidade RNN simples.

Uma RNN possui um mecanismo de ciclo, que permite a comunicação de informações de um passo de tempo  $t$  para o próximo passo  $t + 1$ . Essas informações são denominadas de estados escondidos  $h_t$ , e representam a entrada dos dados ao longo do tempo. Na Figura 2.6, o modelo observa uma entrada  $x_t$  e calcula uma saída  $h_t$ , e com o ciclo presente passa a informação para o próximo passo da rede, no qual  $h_t$  se torna  $h_{t-1}$ . Assim, a RNN da Figura 2.6 pode ser pensada como múltiplas cópias da mesma rede, cada qual comunicando uma mensagem para seu sucessor.

No entanto, as RNNs simples possuem dificuldades para aprender sequências de informações longas [48]. Tal dificuldade acontece porque as RNNs possuem um valor denominado gradiente, que é responsável pela permanência da informação. Quando muita informação é alimentada no modelo, o gradiente começa a diminuir de forma significativa, causando com que as informações prévias percam valor. Essa limitação é denominada de dissipação do gradiente [43]. Entender as dependências de longo prazo ainda é um dos maiores desafios em aprendizado profundo [25].

Até a escrita deste trabalho, a principal forma de tratamento para a limitação da dissipação do gradiente foi a ideia de incluir caminhos pelo tempo que possuem valores de gradiente que não desaparecem, nem crescem de forma descontrolada [25]. Modelos baseados nesse mecanismo são denominados de RNNs baseada em portões, e são capazes de decidir que informações desejam guardar ou esquecer para melhorar o desempenho do processo de decisão. Essas redes incluem a *Long Short-Term Memory* e o *Gated Recurrent Unit*, apresentados nas Subseções 2.3.2 e 2.3.3.

### 2.3.2 Long Short-Term Memory

Uma rede *Long Short-Term Memory* (LSTM) [48] tem como característica uma arquitetura de RNN baseada em portões. Sendo possível ser aplicada à previsão de dados temporais [25]. Além de ser o modelo de sequência mais eficaz, a LSTM mostrou resultados substanciais ao explorar inter-correlações de longo prazo [48, 25]. O uso de um modelo baseado em portões supera a dificuldade da RNN simples em aprender dependências de longo prazo com mais de alguns intervalos de tempo [48], aderindo o problema da série temporal com uma abordagem viável.

A Figura 2.7 apresenta uma topologia para uma célula LSTM. A célula é composta de um estado  $c_t$ , que transmite informação através da cadeia, implicando em uma redução dos efeitos da memória de curto prazo; e portões, mecanismos internos que regulam o fluxo de informação. Os mecanismos de portões (veja novamente na Figura 2.7) permitem informações serem adicionadas ou removidas no estado da célula através do processo de aprendizado por funções *sigmoide*. Essas funções são representadas pela Equação 2.13, onde  $e$  representa o número de Euler, e  $x$  o número real a ser escalado.

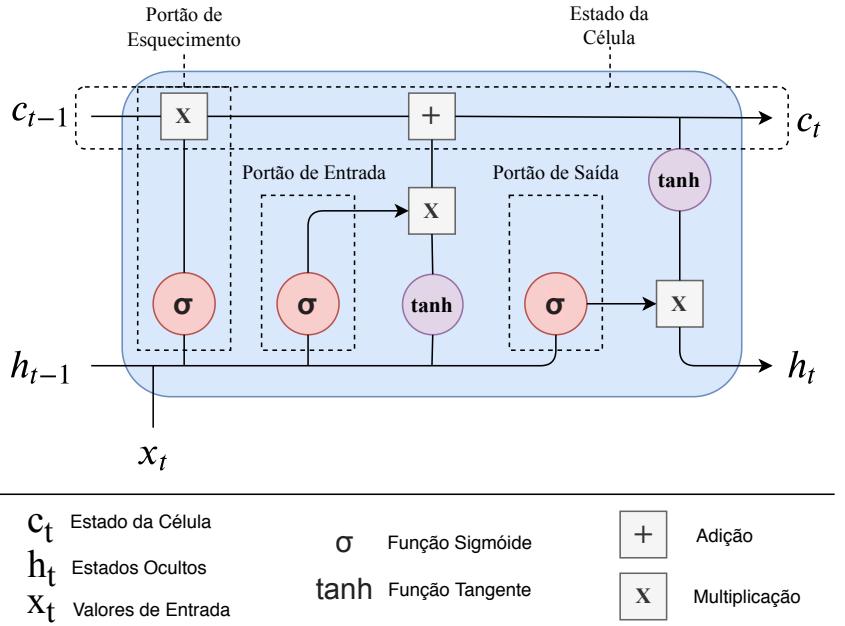


Figura 2.7: Topologia de uma unidade LSTM.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.13)$$

A estrutura da LSTM possui três portões principais, cada qual com uma tarefa específica, como visto na Figura 2.7. O portão do esquecimento é o que decide a permanência de informações. Valores de estados prévios  $c_{t-1}$  e valores de entrada atual  $x_t$  são passados para a função *sigmoide*, gerando um fator de esquecimento. O portão de entrada é responsável por atualizar o estado da célula com informações da entrada atual  $x_t$  e de estados prévios  $c_{t-1}$ . Esses valores são passados para a função *sigmoide* que decide quais informações serão atualizadas, e posteriormente para a função *tangente* para regular a rede. A saída da função de ativação *sigmoide* decide quais informações manter da função tangente, por uma operação de multiplicação. Por último, o portão de saída decide qual deve ser o valor do próximo estado oculto  $h_t$ . Esse estado contém informações de entradas prévias e é responsável pela predição do modelo. Seu valor é calculado com o estado oculto prévio  $h_{t-1}$  e com a entrada atual  $x_t$  em uma função *sigmoide* que calcula o valor  $out_t$ . O valor  $out_t$  é posteriormente multiplicado pela *tangente* do estado atual  $c_t$ , para decidir que informação o novo estado oculto  $h_t$  deve conter.

Após a nova entrada  $x_t$  ser processada nos portões, o valor do estado da célula é atualizado. Seu valor é obtido primeiramente multiplicando o estado prévio  $c_{t-1}$  pelo fator de esquecimento, removendo informações desnecessárias. Posteriormente, é realizada uma operação de adição com o valor resultante do portão de entrada, atualizando assim o valor

de estado  $c_{t-1}$  para um novo valor  $c_t$ . O novo valor  $c_t$  e o valor oculto  $h_t$  são propagados pela cadeia para o próximo passo até que todos os passos sejam realizados, e assim o modelo realiza o aprendizado profundo.

Como entrada a célula LSTM recebe um vetor de três dimensões com a forma:  $x_t = \langle \text{batch size}, \text{passos}, \text{características} \rangle$ . O *batch size* define o número de amostras que vão ser propagadas pela rede; os *passos* representam qual o período de tempo de cada uma das amostras são; e as *características* são o número de dimensões que são alimentadas a cada *passo*, no qual cada dimensão é um dado de aprendizado. Como um exemplo, uma aplicação no mercado de ações poderia considerar um *passo* sendo um dia útil de mercado. Com isso, se a entrada contém 12 *passos* com 32 de *batch size* e o preço de fechamento sua única *característica*, uma entrada teria o formato de 32 amostras de preço de fechamento de cada um dos 12 dias.

Uma camada LSTM também pode considerar múltiplas unidades. Quando tal configuração acontece, a LSTM consistirá de  $n$  cópias independentes dela mesma, cada cópia contendo uma estrutura idêntica, mas inicializada com pesos diferentes, e portanto, computando diferentemente. Com isso, ao usar  $n$  unidades, a camada LSTM vai produzir  $n$  saídas.

### 2.3.3 Gated Recurrent Unit

Outro tipo de RNN baseada em portões é a *Gated Recurrent Unit* (GRU) [49]. A GRU foi motivada pela LSTM, mas sua implementação e computação são muito mais simples [49]. Com base no conhecimento da LSTM, pode-se analisar a GRU de forma similar. Sua topologia, como vista na Figura 2.8, remove o estado da célula  $c_t$  e usa um estado escondido  $h_t$  para transferir informações no seu lugar. Além disso a GRU contém outros portões, o de redefinição e o de atualização, como apresentado na Figura 2.8.

O portão de atualização funciona de forma similar aos portões de esquecimento e de entrada de uma LSTM, cabe a ele decidir quais informações são úteis [25]. Logo, o portão é responsável por decidir quanto de informação é passado do estado oculto  $h_t$  atual para o próximo  $h_{t+1}$  [49]. Já o portão de redefinição é utilizado para decidir quanto de informação passada vai ser esquecida, introduzindo um efeito adicional não linear na relação entre estados passados e futuros [25]. Quando o portão de redefinição chega próximo a zero, o estado oculto  $h_t$  é forçado a esquecer informações prévias e redefinir-se com apenas a entrada atual  $x_t$  [49]. Portanto, o portão permite que estados ocultos esqueçam informações irrelevantes, permitindo uma representação mais compacta dos dados [49].

Assim como a LSTM, a GRU tem sua entrada  $x_t$  do mesmo formato, e também pode considerar múltiplas unidades, cada unidade computando diferentemente. Como cada

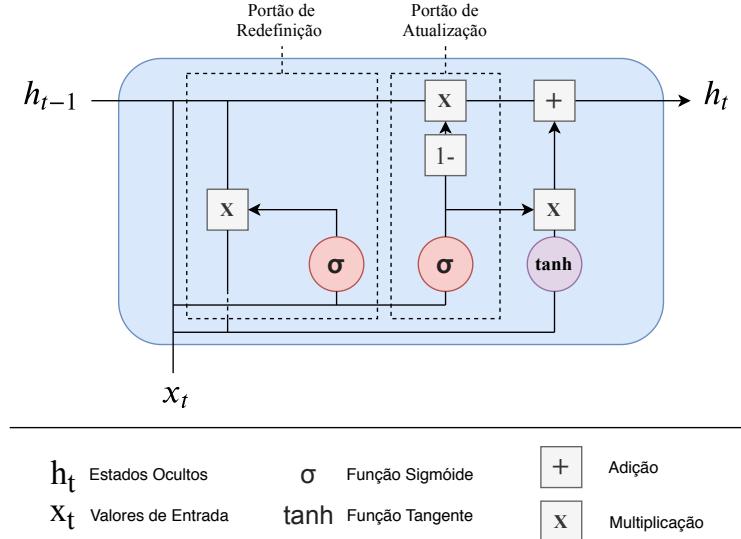


Figura 2.8: Topologia de uma unidade GRU.

unidade GRU possui portões de atualização e redefinição independentes, cada unidade aprenderá a capturar informações em diferentes escalas de tempo [49]. Aquelas unidades que aprenderam informações de curto prazo, tenderão a ter seus portões de redefinição frequentemente ativos. Por outro lado, as unidades que aprenderam informações de longo prazo, vão ter seus portões de atualização mais ativos.

## 2.4 Aprendizado Por Reforço

Situações em que uma entidade não conhece nada sobre o ambiente inserido e precisa descobrir o que fazer são características do Aprendizado por Reforço. Nos ambientes a entidade aprendiz precisa descobrir como mapear situações com ações, de forma a maximizar um sinal de recompensa numérico [50]. A entidade aprendiz, não sabe quais ações realizar, nem como o ambiente funciona, assim precisando explorar o custo de suas ações como forma de aprendizado [24]. Essa forma de aprendizado pode ser comparada com muitos ambientes não computacionais. Por exemplo, um bebê aprendendo a andar, aprende a se equilibrar e a se movimentar com seus próprios erros, tomando ações exploratórias para conseguir ficar mais tempo em pé e começar a andar.

As principais características do aprendizado por reforço são o agente e o ambiente [33]. A Figura 2.9 apresenta a interação, em um espaço de tempo (ou passo), entre ambos. O ambiente é o meio em que o agente está inserido, podendo ser modificado pelo agente, ou sofrer mudanças por si só [50]. O agente por sua vez, interage com o ambiente em cada passo da simulação, realizando uma ação com base em sua observação feita

do estado do ambiente [50]. O agente recebe também do ambiente uma realimentação numérica do impacto de suas ações em cada passo, chamada de recompensa ou reforço [24]. O objetivo do agente é maximizar sua recompensa acumulada durante a simulação, denominada retorno. Métodos de aprendizagem por reforço são meios de um agente aprender comportamentos para alcançar um objetivo.

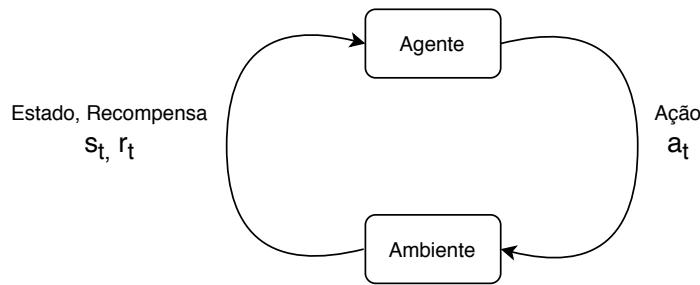


Figura 2.9: Interação básica agente-ambiente no aprendizado por reforço.

O aprendizado por reforço possui conceitos intrínsecos que são necessários para entender o seu funcionamento, sendo os principais:

- **Estado e Observação:** Um estado  $s$  é a descrição completa do ambiente, não existe informação sobre o ambiente que não é refletida pelo estado [50]. Quando a informação não reflete o ambiente de forma completa, descrevendo-o de forma parcial, ela recebe o nome de observação [33]. Quando o agente consegue observar o estado completo do ambiente, o ambiente é denominado totalmente observável. Por outro lado, quando apenas uma parte é percebida, denominamos o ambiente de parcialmente observável.
- **Espaço de Ações:** Ambientes diferentes permitem diferentes tipos de ações  $a$ . O conjunto de todas as ações validas em dado ambiente é denominado espaço de ações [50]. Há ambientes que possuem espaço de ação discretos, nos quais um número finito de movimentos está disponível para o agente. Outros ambientes, como ambientes robóticos, possuem espaço de ações contínuos, nos quais ações são vetores de valores reais. A distinção do espaço de ações implica consequências para métodos em aprendizagem por reforço profunda [51]. Alguns algoritmos podem ser aplicados somente em um dos casos, e precisaria de uma remodelagem substancial para trocar [51]. As ações são realizadas por um agente de acordo com sua política.
- **Política:** Pode-se pensar na política como o cérebro de um agente racional. A política é uma regra usada por um agente para decidir quais ações realizar [33]. Ela

pode ser determinística, denotado na Equação 2.14 por  $\mu$  ou pode ser estocástica, denotada na Equação 2.15 por  $\pi$ .

$$a_t = \mu(s_t) \quad (2.14)$$

$$a_t \sim \pi(\cdot|s_t) \quad (2.15)$$

Em aprendizado por reforço profundo, políticas são parametrizadas [52]. Logo, a saída de uma política parametrizada são funções dependentes de um conjunto de parâmetros ajustáveis por algum algoritmo de otimização [52]. Neste trabalho, parâmetros de tais políticas serão representados por  $\theta$  de forma subscrita no símbolo da política:

$$\begin{aligned} a_t &= \mu_\theta(s_t) \\ a_t &\sim \pi_\theta(\cdot|s_t) \end{aligned} \quad (2.16)$$

- **Trajetória:** Uma trajetória, ou episódio,  $\tau$  é uma sequência de estados e ações no ambiente.

$$\tau = (s_0, a_0, s_1, a_1, \dots) \quad (2.17)$$

Transições de estados, são mudanças que acontecem no ambiente entre o estado atual  $s_t$  e o estado futuro  $s_{t+1}$  [50]. Tais transições são regidas por leis naturais do ambiente, e dependem apenas da ação mais recente  $a_t$ , podendo ser determinísticas  $s_{t+1} = f(s_t, a_t)$  ou estocásticas  $s_{t+1} \sim P(\cdot|s_t, a_t)$  [50].

- **Recompensa:** A função de recompensa é um conceito de extrema importância para o aprendizado por reforço. Seu valor  $r_t$  é descrito pela Equação 2.18 e depende do estado atual do ambiente  $s_t$ , da ação tomada  $a_t$  e do próximo estado do meio  $s_{t+1}$  [50].

$$r_t = R(s_t, a_t, s_{t+1}) \quad (2.18)$$

O objetivo do agente é maximizar o valor acumulado da recompensa ao longo de uma trajetória [50]. Esse valor é denominado retorno  $R(\tau)$ , mas pode representar noções diferentes dependendo da aplicação. Uma das possíveis formas do retorno, representado pela Equação 2.19, é o não descontado de horizonte finito, a soma de todos as recompensas obtidas  $r_t$  em um intervalo de tempo  $T$  [52]. A outra abordagem é remover o intervalo de tempo  $T$  e realizar uma soma de todas as recompensas

obtidas, descontadas em quanto tempo no futuro cada uma foi obtida [50]. Tal abordagem é dominada retorno descontado de horizonte infinito, representada pela Equação 2.20, e requer um termo adicional denominado fator de desconto, denotado por  $\gamma \in (0, 1)$  [33].

$$R(\tau) = \sum_{t=0}^T r_t \quad (2.19)$$

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t \quad (2.20)$$

A necessidade de descontar uma recompensa em vez de receber-las por completo é visado por dois motivos. Primeiro que, a recompensa imediata é geralmente mais atraente que recompensas futuras [51]. E segundo, que matematicamente, uma soma de horizonte infinito de recompensas pode não convergir para um valor finito, criando dificuldades matemáticas. Com o fator de desconto e sob circunstâncias sensatas a soma infinita converge [51].

- **Funções de valor:** As funções de valor são responsáveis por conhecer o valor de um estado (ou par estado-ação), no qual o valor é a métrica de retorno esperado se você iniciar em um determinado estado (ou par estado-ação) e agir de acordo com uma política específica para sempre [50]. Tais funções obedecem equações de auto-consistência denominadas Equações de Bellman [53, 50]. A ideia básica por trás dessas equações é: O valor do seu ponto de partida é a recompensa que você espera receber por estar lá, mais o valor de onde quer que você for em seguida [53].

#### 2.4.1 Gradiente de Política Determinística Profunda

No aprendizado por reforço profundo, o algoritmo do Gradiente de Política Determinística Profunda (DDPG) [54, 55], combina ideias dos recentes avanços do aprendizado profundo e do aprendizado por reforço, resultando em um algoritmo que resolve de maneira robusta problemas desafiadores em domínios com espaços de ação contínuos [55].

O DDPG é um algoritmo que aprende simultaneamente uma função ação-valor  $Q$  e uma política, por meio de um mecanismo de redes neurais profundas denominado de ator-critico. A Figura 2.10 apresenta um diagrama do funcionamento do DDPG. O ator (cor azul na Figura 2.10) é o responsável por utilizar dados fora da política e uma equação de Bellman para aprender a função  $Q$ . Posteriormente, o crítico (cor verde) utiliza a função  $Q$  aprendida pra aprender a política. A função ação-valor  $Q$  é baseada em um algoritmo clássico de aprendizagem por reforço, o *Q-Learning*, e é motivado da mesma forma: se

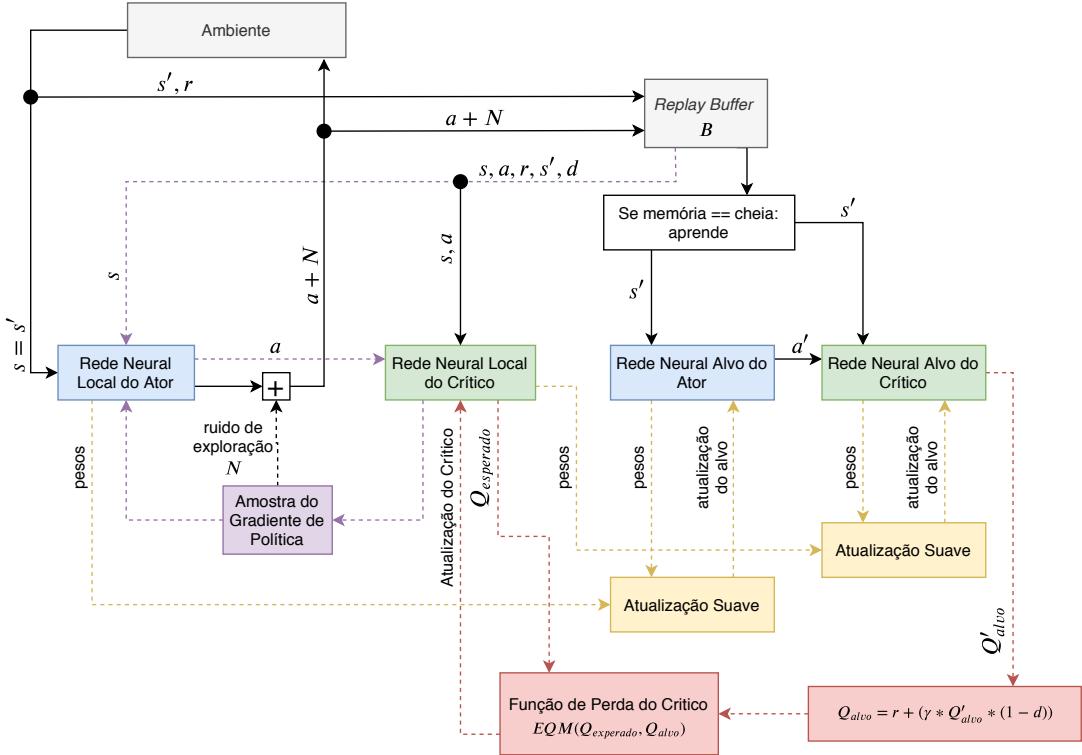


Figura 2.10: Diagrama de funcionamento do DDPG. Adaptado:.

a função ação-valor ótima  $Q^*(s, a)$  é conhecida, então em qualquer estado, a ação ótima  $a^*(s)$  pode ser encontrada resolvendo a Equação 2.21.

$$a^*(s) = \arg \max_a Q^*(s, a) \quad (2.21)$$

Portanto, o algoritmo do DDPG alterna entre aprender um approximador para  $Q^*(s, a)$  (ator) e aprender um approximador para  $a^*(s)$  (crítico), realizando a tarefa de forma especificamente adaptada para ambientes de com espaço de ações contínuos.

Matematicamente, o DDPG se adapta em ambientes contínuos de acordo com a forma que computa o valor máximo das suas ações em  $\max_a Q^*(s, a)$ . Descobrir o valor máximo não seria uma tarefa problemática em espaço de ações discretos, quando se existe números finitos de ações pode-se comparar todos os valores diretamente um com o outro. No entanto, ao adaptar o problema para espaços contínuos, avaliar de forma bruta todos os espaços possíveis e resolver o problema de otimização, significaria calcular o máximo de infinitos valores em todo passo da simulação. Esse custo temporal não é aceitável para algoritmos de aprendizado. Logo, por o espaço de ação ser contínuo, a função  $Q^*(s, a)$  é assumida ser derivável em relação a ação. Isso permite estabelecer uma regra de aprendizado eficiente, baseada em gradiente, para uma política  $\mu(s)$  que explora esse fato. Assim, em vez de resolver uma tarefa custosa quando  $\max_a Q(s, a)$  precisa ser

calculado, realiza-se uma aproximação do valor com  $\max_a Q(s, a) \approx Q(s, \mu(s))$ .

A matemática básica por traz do funcionamento do DDPG é apresentada a seguir em duas partes: aprender a função  $Q$ , e aprender uma política.

## Aprendizado da Função $Q$

No algoritmo do DDPG existe uma equação de Bellman que descreve a função de ação-valor ótima  $Q^*(s, a)$ . Essa equação é descrita pela Equação 2.22, onde  $s' \sim P$  significa que o próximo estado  $s'$ , é uma amostra do ambiente dado pela distribuição normal  $P(\cdot|s, a)$ .

$$Q^*(s, a) = \underset{s' \sim P}{\text{E}} \left[ r(s, a) + \gamma \max_{a'} Q^*(s', a') \right] \quad (2.22)$$

Essa equação de Bellman é o ponto de partida para aprender um ator para aproximar  $Q^*(s, a)$ . Esse ator será descrito como uma rede neural  $Q_\theta(s, a)$ , com parâmetros  $\theta$ . Com o propósito de avaliar o desempenho do ator, precisa-se construir uma função de perda que indique o quanto  $Q_\theta$  chega a satisfazer a Equação 2.22. Tal função é denominada Erro Quadrático Médio de Bellman (EQMB) (representada pela Equação 2.23), e pressupõe a existência de um conjunto de  $\mathcal{D}$  de transições  $(s, a, r, s', d)$  passadas (onde  $d$  indica se  $s'$  é um estado final).

$$L(\theta, \mathcal{D}) = \underset{(s, a, r, s', d) \sim \mathcal{D}}{\text{E}} \left[ \left( Q_\theta(s, a) - \left( r + \gamma(1-d) \max_{a'} Q_\theta(s', a') \right) \right)^2 \right] \quad (2.23)$$

Algoritmos de *Q-Learning* para aproximação de funções, como o DDPG, são baseados na minimização da função EQMB. Para tal minimização, a implementação do algoritmo neste trabalho conta com duas técnicas apresentadas na Figura 2.10:

1. **Replay Buffer:** Um mecanismo de armazenamento do conjunto  $\mathcal{D}$  de experiências prévias. Para o algoritmo, ter um comportamento estável, o conjunto tem que ser grande o suficiente para conter um grande números de experiências, mas não deve ser grande a ponto de conter tudo. Se apenas for usado dados muito recentes, o modelo vai realizar um sobreajuste e não irá funcionar; se muitas experiências forem usadas o processo de aprendizado vai ficar muito custoso.
2. **Redes Neurais Alvo:** Uma técnica que utiliza o termo alvo descrito na Equação 2.24 com o objetivo de aproximar a função  $Q$  desse valor. Realizando assim, a minimização da função EQMB.

$$r + \gamma(1-d) \max_{a'} Q_\theta(s', a') \quad (2.24)$$

No entanto, o problema de minimização da EQMB é instável, devido ao fato do alvo depender dos mesmos parâmetros  $\theta$  que será treinado. Para adereçar esse problema é necessário a utilização de um conjunto de parâmetros que se aproximam de  $\theta$ , mas com um atraso de tempo. Para implementar o atraso, utiliza-se redes neurais adicionais, denominadas redes neurais alvo, uma para o ator e outra para o crítico, com o objetivo de atrasar suas redes locais respectivas (observe novamente a Figura 2.10). Os parâmetros das redes secundárias são denominados  $\theta_{\text{targ}}$ , e as redes são atualizadas a cada atualização da sua rede primária respectiva.

Portanto, juntando ambas as técnicas, o aprendizado da função  $Q$  no DDPG é realizado minimizando a Equação 2.25 com gradiente descendente estocástico, onde  $\mu_{\theta_{\text{targ}}}$  é a política alvo.

$$L(\theta, \mathcal{D}) = \underset{(s, a, r, s', d) \sim \mathcal{D}}{\mathbb{E}} \left[ \left( Q_\theta(s, a) - (r + \gamma(1-d)Q_{\theta_{\text{targ}}}(s', \mu_{\theta_{\text{targ}}}(s')) \right)^2 \right] \quad (2.25)$$

## Aprendizado de Política

Após o aprendizado da função  $Q$ , o algoritmo precisa aprender uma política determinística  $\mu_\theta(s)$ , que retorna a ação que maximiza o valor  $Q_\theta(s, a)$ . Assim, pelo espaço de ações ser contínuo, e pela suposição de que a função  $Q$  é derivável em relação a ação, pode-se realizar um cálculo de gradiente ascendente, apenas com os parâmetros de política, para resolver a Equação 2.26.

$$\max_{\theta} \underset{s \sim \mathcal{D}}{\mathbb{E}} [Q_\theta(s, \mu_\theta(s))] \quad (2.26)$$

Devido a política ser determinística, se o agente decidisse explorar somente com dados da política, no começo provavelmente não teria uma variedade de ações ampla o suficiente para o agente receber sinais de recompensa significativos. Portanto, o algoritmo do DDPG treina uma política determinística com dados fora de sua política. Adicionalmente, para o melhorar a exploração, é necessário adicionar um ruído nas ações do agente enquanto treina. O ruído é retirado na fase de teste para avaliação significativa da qualidade em que a política aproveita o que aprendeu.

# Capítulo 3

## Trabalhos Relacionados

Nos últimos anos, vários trabalhos foram publicados no campo de aprendizagem de máquina com foco em predição do mercado de ações. A literatura contem trabalhos mais recentes [15, 17, 13, 4, 16, 14, 26] e outros mais clássicos ao decorrer dos anos [5, 6, 7, 8, 9, 10, 11]. Nesta seção serão apresentados os desafios da área em questão em duas frentes: analise técnica e analise fundamental. Apesar dos avanços alcançados na área, até o momento não foram encontrados pesquisas propondo serviços híbridos que fazem uso da ambas as análises para negociação de ativos nos mercados.

### 3.1 Análise Fundamental

Considerando as abordagens que utilizam análise fundamental, Malagrino, Roman e Monteiro [14] investigam o uso de redes Bayesianas como um meio de verificar até que ponto os índices de mercados internacionais influenciam no índice principal do mercado de ações brasileiro, a [B]<sup>3</sup>. Com esse intuito, a direção, subida ou ou decida, do valor dos índices foram usadas como entrada para a rede, sendo passadas em ciclos de 24 e 48 horas, e produzindo como saída a direção do índice [B]<sup>3</sup> no próximo dia. A rede Bayesiana modelada permite a vantagem adicional de permitir um uso mais direto com maior capacidade de tratamentos para seus usuários quando comparada com outros modelos analisados pelo trabalho. No entanto, um aprendizado Bayesiano é computacionalmente muito custoso, o que leva as redes a tenderem para um desempenho pior em dados de grande dimensão.

Nti, Adekoya e Weyori [13] propõem um modelo baseado em florestas aleatórias para seleção de características de variáveis macroeconômicas e, em seguida, uma LSTM, apropriada para predição do mercado. O propósito era examinar o grau de significância entre os preços históricos, de diferentes setores, e variáveis macroeconômicas para predizer o preço mensal de um ativo. Porém, o modelo proposto fornece pouco controle sobre o

processo de decisão para predição do mercado, visto que floresta aleatória, geralmente, é considerada como uma técnica de caixa-preta.

Ainda com foco em análises fundamentais, a abordagem primária realizada pelos pesquisadores é a análise de sentimentos de redes sociais, tais como *Twitter* e *Facebook*. Nesse contexto, Preis, Moat e Stanley [5] sugerem que a base de dados massivas resultantes da interação de humanos com a internet pode oferecer uma perspectiva do comportamento do mercado. O estudo identificou padrões que podem ser interpretados como “sinais de alerta” de um alto movimento no mercado de ações, analisando as mudanças no volume de buscas de termos financeiros no Google. Contudo, os pesquisadores escolheram uma premissa de que um aumento nas buscas pode indicar uma baixa do valor do ativo. Essa suposição nem sempre é verdadeira para todos os casos.

## 3.2 Análise Técnica

Outra frente de pesquisa utiliza a análise técnica como solução para prever os preços futuros de ativos. Paiva et al. [15] propõem um modelo de decisão para seleção de portfólio de investimentos do tipo *day-trading* usando uma abordagem de classificador baseado na fusão de uma Máquina de Vetores de Suporte (SVM) e da Variância Média de Markowitz (VMM). Para isso o modelo proposto foi dividido em dois estágios: (i) a SVM seleciona os ativos com maior potencial de retorno; e (ii) o modelo VMM define a proporção dos recursos para cada ativo do portfólio. No entanto, mesmo VMM sendo amplamente reconhecido como um dos pilares da teoria moderna de portfólio, o modelo apresenta muitas críticas às simplificações realizadas. Algumas das mais criticadas são: os investidores agem sempre de forma racional, não existe informação privilegiada, e ativos são infinitamente divisíveis [6]. Tais críticas motivam a busca de alternativas mais refinadas para seleção de portfólio.

Uma outra abordagem técnica é utilizada por Chandrinos, Sakkas e Lagaros [16], que propõem uma ferramenta de gerenciamento de risco, nomeado ARIMS. O principal objetivo do AIRMS é melhorar o desempenho de dois portfólios, prevenindo-os de qualquer perda. No estudo, a ferramenta é modelada usando dois modelos: uma Rede Neural Artificial (RNA) e uma árvore de decisão. O experimento foi aplicado aos cinco maiores pares de moedas entre os anos de 2010 e 2016. Foi observado que AIRMS com a árvore de decisão, e AIRMS com redes neurais obtiveram sucesso em aumentar o retorno total dos portfólios. Isso evidencia que o AIRMS pode transformar anos em que houveram perda em anos mais lucrativos, diminuindo os retornos negativos. Entretanto, deve-se notar que a ferramenta AIRMS foi aplicada somente a portfólios lucrativos, o mesmo comportamento não necessariamente seria observado em portfólios não lucrativos.

Chung e Shin [17] propõem um Algoritmo Genético (AG) para otimizar um modelo de LSTM, com base em indicadores técnicos do índice do mercado Coreano. Para avaliar o modelo, diferentes tamanhos de janelas e unidades LSTM foram configurados na função objetivo da AG. Resultados mostraram que a AG-LSTM apresentou um desempenho melhor que as configurações de referências em todas as medidas de erro. Isso sugere que a hiper-parametrização apropriada da LSTM é uma condição essencial para uma melhora no desempenho. Entretanto, AGs são usadas para otimização de problemas que a qualidade da solução depende do tempo de processamento. Em outras palavras, soluções baseadas em AG são mais lentas que os métodos tradicionais e, portanto, podem influenciar na predição dos ativos do mercado.

Além dos diversos trabalhos mencionados, pesquisadores vêm estudando majoritariamente análises técnicas nos últimos anos. O uso de métodos tradicionais de aprendizado de máquina pode ser encontrado no *survey* [4]. A SVM é amplamente usada, cada uma com suas particularidades [7, 8, 9, 26]; outras abordagens usam redes neurais clássicas e suas variações [10, 11]. Conduzindo para uma abordagem mais elegante, RNA profundas são amplamente usadas em outros estudos [56, 20, 19, 57]; redes neurais adversárias e co-evolutivas também são encontradas [58, 59]. Por último, alguns dos trabalhos encontrados conduz estudos com uma abordagem baseada em agentes e algumas das vezes combinada com aprendizado por reforço. Cocco, Concias e Marchesi [18] sugerem uma solução para negociação de cripto-moedas utilizando um modelo baseado em agente, enquanto Huang [60] aborda o problema com uma rede neural profunda e um modelo baseado em agente guiado por tal rede. Portanto, observa-se na literatura um conjunto vasto de métodos e técnicas para análise de diversos mercados, mas ainda se mostra um ambiente repleto de oportunidades de pesquisa.

### 3.3 Considerações Finais

Apesar dos avanços realizados no campo de aprendizagem de máquina para o mercado de ações, ainda há inúmeros desafios e problemas na área que essa pesquisa soluciona, diferenciando-se dos trabalhos relacionados nos seguintes aspectos:

1. Um sistema de investimento autônomo, racional e baseado em agentes é apresentado para substituir o lugar de um humano no mercado de investimentos, lidando com previsões e alocações apropriadas de recursos no gerenciamento de portfólios.
2. Metodologia investigativa para melhorar as previsões de mercado. Tipicamente, parâmetros são empiricamente definidos e aplicados independentemente do mercado e do ativo. Para tal propósito, este trabalho recorre para otimização de hiper-

parâmetros focada em: maior possibilidades de configuração para o usuário, desenvolvimento de algoritmos bem calibrados, e *fuzz testing*; a biblioteca Hyperopt [61] provê tais características.

3. Um mecanismo de predição que utiliza um dos estados da arte de redes neurais recorrentes, a LSTM [17].
4. Atenção direcionada individualmente para cada ação selecionada da bolsa de valores [B]<sup>3</sup>, em outras palavras, cada ativo do portfólio tem seu próprio modelo especializado com seus próprios parâmetros.

Portanto, este trabalho se destaca em comparação aos demais da literatura recente, tendo em vista os distintos aspectos apresentados. O Aspecto 1 propõe um sistema de investimento autônomo, diferentemente da maioria dos outros trabalhos analisados [13, 15, 17, 14], que geralmente propõem estudos focados apenas em um certo modelo. É diferente também de Chandrinos, Sakkas e Lagaros [16], que propõem uma ferramenta projetada para controle de risco. Já o Aspecto 2 aborda o refinamento de hiper-parâmetros através de um método com etapas bem definidas. Na literatura, Chung e Shin [17] usam uma AG para encontrar as melhores janelas e unidades de LSTM, mas não considera outros parâmetros como otimizadores e *batch size*. Malagrino, Roman e Monteiro [14] efetuam experimentos com somente dois tamanhos de janela, 24 e 48 horas. Nti, Adekoya e Weyori [13] e Paiva et al. [15] escolhem otimizadores e *kernels*, respectivamente, através de vantagens teóricas, porém não realizam outras experimentações exploratórias. Finalmente, Chandrinos, Sakkas e Lagaros [16] calibram os hiper-parâmetros da rede neural artificial experimentando somente com as camadas escondidas e quantidade de épocas. No entanto, todos esses trabalhos apresentam perspectivas mistas e incompletas para a calibração dos hiper-parâmetros, díspar desta pesquisa. O Aspecto 3 se mostra distintivo quando comparado com [15, 16, 14], nos quais usam SVM, RNA, e rede Bayesiana, respectivamente. Por último, o Aspecto 4 se distingue quando relacionado com Malagrino, Roman e Monteiro [14], Chung e Shin [17] e Paiva et al. [15] que foca nos estudos do mercado de moedas. Portanto, nenhum dos trabalhos citados utilizam ativos de empresas fornecidas pelo mercado. Por esses motivos, este trabalho propõe o Hare, um serviço de investimento baseado em um sistema de agentes autônomos e racionais que será apresentado no Capítulo 4.

# Capítulo 4

## Hare: Um Serviço Autônomo de Investimentos

Neste capítulo, é apresentado o serviço proposto para negociar na bolsa de valores, sendo o foco da implementação o investimento autônomo, racional e baseado em agentes. Denominado de Hare, o serviço trata diversos aspectos do processo de investimento de forma modular, em que cada módulo é responsável por uma tarefa específica, sendo as principais: predição de movimento do valor de ativos, controle de riscos de ativos, e gerenciamento de recursos de um portfólio.

### 4.1 O Serviço Hare

Seguindo a lógica de raciocínio apresentada no Capítulo 3, buscou-se uma abordagem que preenche os aspectos citados como lacunas existentes nos trabalhos relacionados. Portanto, o Hare utiliza séries temporais de dados históricos de ativos da [B]<sup>3</sup>, e múltiplas variáveis de análise fundamental em seu processo de investimento. As séries temporais utilizam ativos, tais como PETR3, VALE3, e ABEV3, para predizer o valor deste ativo no dia seguinte. Enquanto as variáveis fundamentais (como análise de volume de pesquisa no *Google*, variáveis macroeconômicas, e índices de mercados internacionais), observam pela existência de possíveis riscos no ativo. Todas essas informações são posteriormente utilizadas para gerenciar os recursos do investidor em seu portfólio.

O Hare é descrito utilizando uma estratégia, na qual apresenta-se, primeiramente, os conceitos relativos aos módulos internos que servirão de base para o entendimento dos módulos externos e da estrutura do Hare como um todo. Com esse propósito, uma visão geral do funcionamento do Hare e de suas tecnologias inerentes é apresentado na Seção 4.2. Seguindo com o desenvolvimento de suas estruturas internas: o Módulo Preditor de Movimento, na Seção 4.3; e o Módulo de Gerenciamento de Riscos na Seção 4.4. E

finalizando no Módulo do Agente na Seção 4.5, no qual é pormenorizado como o agente é responsável na realização de ações racionais para gerenciar os recursos do seu portfólio.

## 4.2 Visão Geral

A visão geral do funcionamento do serviço disponibilizado pelo Hare é ilustrado na Figura 4.1. O Hare possui dois módulos racionais principais: o módulo preditor e o agente. O preditor, especializado para um dado ativo, é responsável por indicar se o valor do ativo estudado vai aumentar ou diminuir no próximo dia, dada a série histórica de valores do mesmo. Essa predição é então utilizada pelo agente, junto com a probabilidade de um possível risco para decidir se é mais rentável manter o ativo, comprar mais cotas, ou liquidá-lo. Este processo é realizado para cada ativo que o agente possui em seu portfólio, visando assim, não somente o maior lucro possível de uma só ação, como do portfólio como um todo

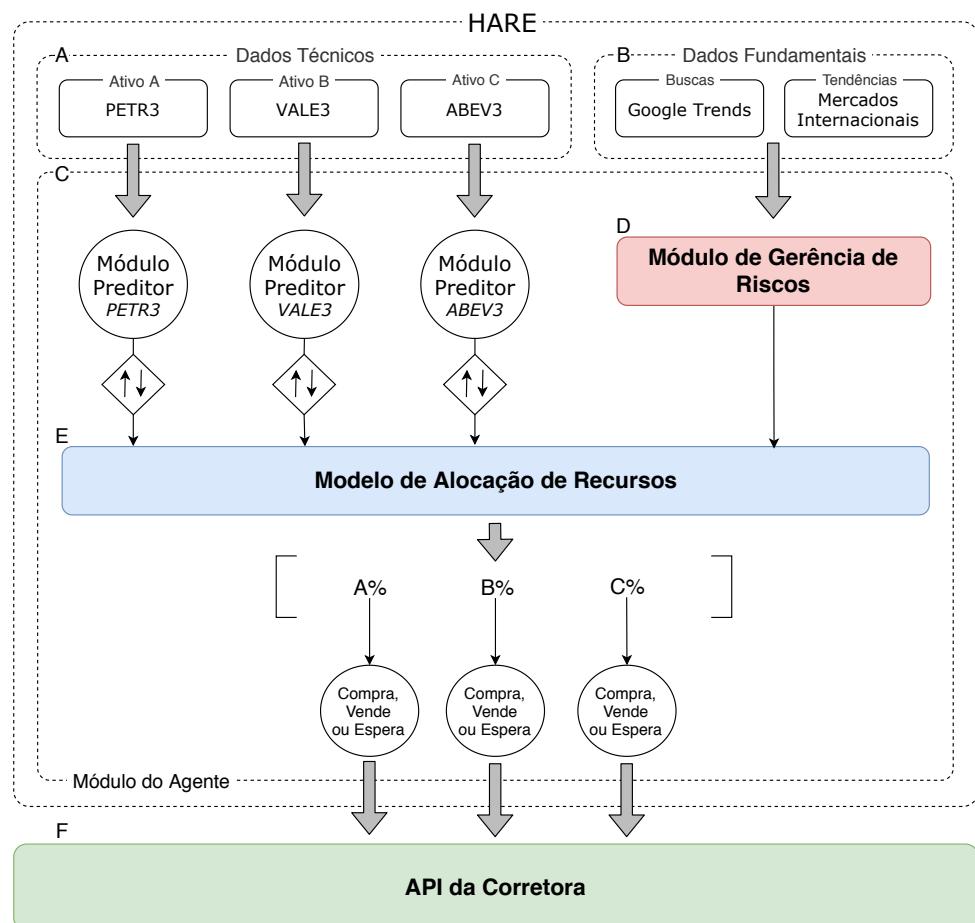


Figura 4.1: Cenário Operacional do Serviço Hare.

O Hare é projetado para funcionar com bases de dados técnicas e fundamentais, representados respectivamente pelos rótulos A e B da Figura 4.1. As informações de dados técnicos, tais como o preço do ativo, são utilizadas como entradas para o Módulo Preditor de Movimento, o qual é responsável por realizar as previsões de movimento para seu ativo especializado. Adicionalmente, os dados de análises fundamentais, como informações do *Google Trends* ou dados de relatórios trimestrais da empresa, são responsáveis por alimentar o Módulo de Gerenciamento de Riscos (MGR) (como visto na Figura 4.1 rótulo D). O MGR tem a função de descobrir, com os dados das variáveis fundamentais, problemas e riscos em tempo real com as ações no portfólio do usuário. A saída de ambos esses módulos, Preditor e MGR, são utilizadas como entrada para o Modelo de Alocação de Recursos (MAR) (Figura 4.1 rótulo E). O Modelo de Alocação de Recursos (MAR) é responsável por decidir que ações vão ser vendidas, mantidas ou compradas, além de decidir também as quantidades dos recursos que serão alocados no caso de uma compra. Após o MAR tomar as decisões, cabe ao agente (Figura 4.1 rótulo C) agir com base nelas. As ações são realizadas então por uma série de requisições para a API da corretora desejada (Figura 4.1 rótulo F).

Para realizar esse processo na prática, o Hare foi projetado<sup>1</sup> usando a linguagem de programação Python <sup>2</sup> em conjunto com os pacotes *Pandas*<sup>3</sup>, *Keras*<sup>4</sup>, *Gym*<sup>5</sup> e *Spinning Up*<sup>6</sup>. O *Pandas* foi utilizado para leitura e processamento de dados; o *Keras*, com seu processo interno do *Tensor Flow*, foi utilizado para projetar os modelos de aprendizado supervisionado; o *Gym* utilizado para criar o ambiente o qual o agente interage durante o aprendizado por reforço; e por fim, o *Spinning Up* para implementar os algoritmos de aprendizado por reforço profundo. Para entender melhor cada aspecto do Hare, os seus três módulos são detalhados nas Seções seguintes: Módulo Preditor de Movimento na Seção 4.3, MGR na Seção 4.4 e Módulo do Agente na Seção 4.5.

## 4.3 Módulo de Previsão

Um dos principais módulos racionais do Hare é o Módulo Preditor de Movimento. Sua tarefa é prever informações de movimento de um determinado ativo, baseando-se nos dados históricos do mesmo. Além de dar suporte para que módulo do agente possa realizar suas decisões. Cada módulo preditor é uma entidade individual especializada em uma dada ação, programável para utilizar qualquer modelo de aprendizado desejado.

<sup>1</sup>Disponível em: <https://github.com/EmpyreanAI/AURORA>

<sup>2</sup>Disponível em: <https://www.python.org>

<sup>3</sup>Disponível em: <https://pandas.pydata.org>

<sup>4</sup>Disponível em: <https://keras.io>

<sup>5</sup>Disponível em: <https://gym.openai.com>

<sup>6</sup>Disponível em: <https://spinningup.openai.com>

Unidades de predição, apresentadas na Figura 4.2, coletam seus dados da base da bolsa de valores, com diversos campos de informações sobre as ações. Nestes dados, múltiplas informações estão disponíveis para serem usadas como entrada do modelo, tais como: preço de abertura, preço de fechamento, e volume. Qualquer um desses pode ser utilizado como valor de entrada para alimentar o modelo de aprendizado, e uma predição é criada como resultado. Por uma visão holística, este trabalho utiliza o preço de fechamento, que é condizente com o final do pregão<sup>7</sup>. Tal predição toma a forma de um valor binário, onde 1 indica um aumento e 0 uma diminuição do valor da ação.

Com o objetivo de realizar as predições das ações, os módulos de predição do Hare foram modelados por meio de uma rede LSTM, uma variação da RNN utilizando técnicas de portões. As redes RNNs baseadas em portões foram modeladas no Hare com o propósito de conseguir criar caminhos pelo tempo que possuem deriváveis que não somem nem explodem ao longo do tempo [25]. Graças a esse mecanismo, redes LSTM são capazes de decidir quais informações vão ser mantidas ou esquecidas, visando aumentar a acurácia do processo de tomada de decisão. A Figura 4.2 apresenta o Módulo Predictor de Movimento, e sua versão implementada.

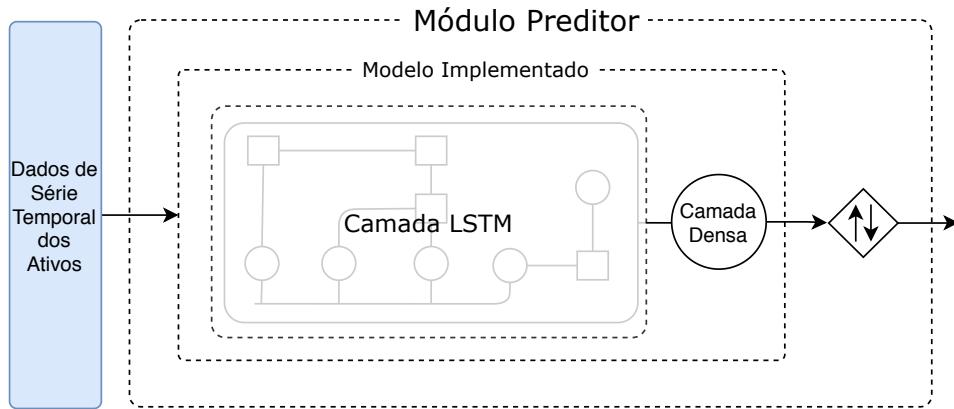


Figura 4.2: Visão Geral do Módulo Predictor de Movimento. O modelo preditor escolhido pode ser alterado por qualquer modelo de aprendizagem desejado.

A rede LSTM foi escolhida por possuir uma arquitetura distinta, que permite ser utilizada para predição de dados temporais. Além de ser o modelo de processamento sequencial mais efetivo [25], a LSTM demonstrou resultados substanciais ao analisar correlações de longo prazo. O uso de tal modelo baseado em portões no Hare supera as dificuldades da RNN de aprender dependências de longo prazo [48], aderêçando o problema de predição de movimento de ações da bolsa de valores com uma abordagem viável.

<sup>7</sup>Nos pregões, tipicamente, os compradores e vendedores possuem as melhores oportunidades nos preços dos ativos

Os dados de entrada que alimentam o modelo são a série histórica de um determinado ativo. A entrada  $x_t$  de uma célula LSTM é um vetor de três dimensões, com o formato  $x_t = \langle \text{batch size}, \text{passo}, \text{características} \rangle$ . O *batch size* define o número de amostrar que vão ser propagadas pela rede; os passos representam quanto tempo cada uma das amostras representam; e finalmente, as características são os números de dimensões de informações que são passados para o modelo a cada passo. Tais informações são cruciais para o aprendizado. O Hare, representa cada passo como um dia útil de mercado. Exemplificando, se o tamanho da entrada consiste de 12 passos com um *batch size* de 32 e uma única característica sendo o preço de fechamento, então a entrada consiste de 32 amostras de preço de fechamento dos últimos 12 dias.

O serviço do Hare utiliza uma LSTM considerando múltiplas unidades (revisite a Figura 4.2). Realizando isso, a LSTM consiste de  $n$  cópias dela mesma, cada cópia terá uma estrutura idêntica, mas inicializada com valores de pesos diferentes, e portanto computando de forma diferente. Ao utilizar  $n$  unidades, a camada de LSTM vai produzir  $n$  saídas, necessitando assim de mecanismos para tradução dessas  $n$  saídas para um valor preditivo. Aderençando essa situação, uma camada de rede neural densamente conectada, ou camada densa, foi adicionada no final do modelo.

Realizando-se o cálculo da camada densa, um valor binário é informado como saída pra cada passo de tempo, este valor é a predição. Após essa predição ser feita, seu valor é utilizado como entrada para o MGR, o qual é então responsável por realizar as decisões mais lucrativas.

No entanto, dada a complexidade do mercado e sua volatilidade à eventos externos, a análise das séries temporais por si só, pode não ser o suficiente para prever algumas anomalias. Para superar essa limitação, o agente é providenciado com um módulo de detecção de distúrbios.

## 4.4 Módulo de Gerenciamento de Riscos

O uso de séries temporais de ações no mercado pode ser capaz de identificar previsões promissoras, mas não é o suficiente para se criar modelos robustos. Com o objetivo de tratar a volatilidade do mercado em relação a influências externas, adicionou-se o Módulo de Gerenciamento de Riscos (MGR) no sistema Hare. O MGR tem como tarefa principal estudar múltiplas fontes de variáveis fundamentais, com o objetivo de refletir a situação atual de um ativo no mercado.

A combinação de diversas técnicas de análise do mercado, utilizando diversas base de dados, tais como as séries temporais de preço das ações e o volume de consultas de pesquisa no *Google*, pode abrir novos horizontes sobre diferentes estágios do processo de tomada

de decisão [5]. O uso das consultas de pesquisa permite, por exemplo, a possibilidade de realizar uma análise de sentimento de um dado ativo, ou de áreas relacionadas a ele, visto que o volume de pesquisa reflete o sentimento de um grupo em relação à aquilo sendo pesquisado. A abordagem de Preis, Moat e Stanley [5] tem como premissa que um aumento no volume de consultas é relacionado com um risco no ativo. Ademais, Bordino et al. [62] afirmam que os volumes de consultas antecipam, em muitos casos, picos de negociação por um dia ou mais. Esta informação pode ser usada no Hare como uma das entradas do modelo, como um possível indicativo de um risco.

Não obstante, outra entrada que pode ser inserida no Hare, é a de dados extraídos de notícias e relatórios de desempenho das empresas dos ativos. Pelas notícias afetarem as decisões humanas e a volatilidade dos preços das ações ser resultado das transações financeiras, é razoável assumir que tais eventos podem influenciar o mercado financeiro [63]. Hagenau, Liebmann e Neumann [64], Ding et al. [63] e Babu, Geethanjali e Satyanarayana [65] estudam diferentes métodos de extração de características que podem ser utilizadas para extrair dados de notícias e relatórios. O serviço do Hare não realiza tal extração de características, e espera em sua entrada somente dados tratados. Características como eficiência da empresa, melhorias, crescimento, lucro, fluxo de caixa, podem ser utilizadas para alimentar o módulo.

As entradas apresentadas podem ser utilizadas então para alimentar um algoritmo ou modelo de aprendizagem que estuda a probabilidade de haver um risco. Dado as entradas, o modelo deve apresentar como resultado um vetor de probabilidades de riscos, sendo cada posição do vetor a probabilidade de um ativo em específico estar em risco. O MGR pode ser customizado, e o modelo é deixado a escolha do usuário. As informações geradas são posteriormente utilizadas como uma das entradas do MAR.

O MGR é modelado como uma função predefinida, não utilizando modelos de aprendizagem de máquina. A abordagem sintética criada é demonstrada na Figura 4.3. O Hare utiliza, para este trabalho, o volume de consultas da razão social de um ativo no *Google Trends*, não considerando as entradas propostas de notícias e relatórios. Os dados utilizados do *Google Trends* são fornecidos em valores semanais de interesse do termo desejado. Tal valor vem como um valor inteiro de 0 a 100, onde 0 representa nenhum interesse e 100 representa um interesse extremo. O modelo sintético implementado, da Figura 4.3, usa um valor limite escolhido pelo usuário. Quando o interesse de pesquisa é abaixo desse valor, o modelo não acusa um risco no ativo, quando o interesse é maior um risco é indicado. A escolha dessa abordagem parte da premissa que todo aumento de pesquisa indica um risco no ativo. Essa premissa pode não ser necessariamente verdadeira, o que pode inserir um ruído no módulo alocador de recursos. Soluções de análise de riscos mais otimizadas serão consideradas em trabalhos futuros.

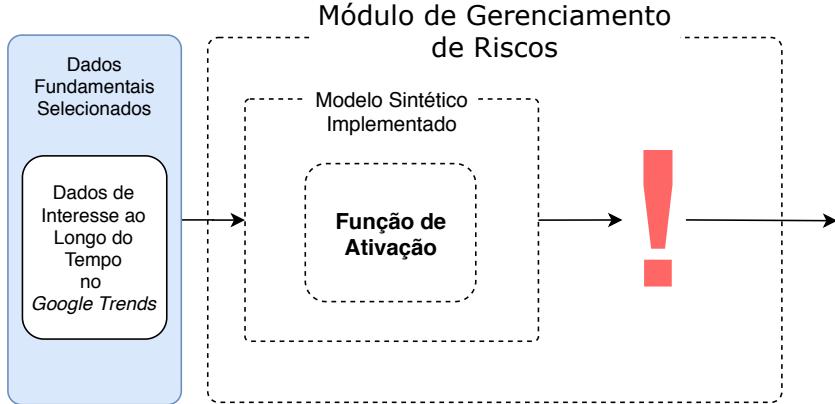


Figura 4.3: Visão Geral do Módulo de Gerenciamento de Riscos. O modelo preditor escolhido pode ser alterado por qualquer modelo de aprendizagem desejado.

## 4.5 Módulo do Agente

O Hare utiliza uma estrutura baseada em agentes, na qual cada agente pode ser visto como um investidor em ações, que tem como objetivo maximizar lucro e reduzir riscos, realizando decisões no portfólio do usuário. Por agente entende-se como uma entidade computacional capaz de perceber o meio por sensores e agir de forma autônoma sobre esse por meio de atuadores [24]. O núcleo de funcionamento do Hare é baseado em um agente que percebe o ambiente (o mercado de ações escolhido), pelas informações de seus ativos, e age neste ambiente de forma racional.

O agente possui uma conta, em que seu dinheiro não alocado em recursos financeiros é guardado, e um portfólio, um conjunto de diferentes ativos escolhidos pelo usuário do serviço, no qual o agente investe seu dinheiro disponível em sua conta. Um agente considerado racional, é aquele que busca maximizar o valor no seu portfólio e minimizar o dinheiro desalocado em sua conta, em outras palavras buscar o investimento com maior rentabilidade. A racionalidade do agente é definida pelo Modelo de Alocação de Recursos.

### 4.5.1 Modelo de Alocação de Recursos

Um agente que busca o investimento com maior rentabilidade utiliza o Modelo de Alocação de Recursos (MAR) como visto na Figura 4.1. O MAR recebe como entrada, informações do movimento das ações, providenciadas pelo Módulo Predictor de Movimento, e informações providenciadas pelo MGR. A saída é então dada como um vetor de porcentagens (revisite a Figura 4.1), sendo cada elemento do vetor respectivo a um ativo.

A implementação do MAR no Hare foi baseado no DDPG, como apresentado na Figura 4.4. Portanto, utilizou-se a biblioteca *Spinning Up* da *OpenAI*, que oferece suporte

aos algoritmos de aprendizagem por reforço implementados. O DDPG foi escolhido por ser livre de modelo, fora de política, e ator-crítico que utiliza aproximação de funções profundas para aprender políticas em espaços de ações contínuos de altas dimensões [55]. O algoritmo tem em sua base o gradiente de política determinista proposto por Silver et al. [54], combinado com ideias de sucesso do *Deep Q Network* de Mnih et al. [66]. Nota-se que este algoritmo e o aprendizado por reforço são estratégias viáveis ao problema, dado que o programador não precisa definir as melhores ações a serem realizadas, e o algoritmo aprenderá a comprar, esperar ou vender, por meio de sua própria exploração.

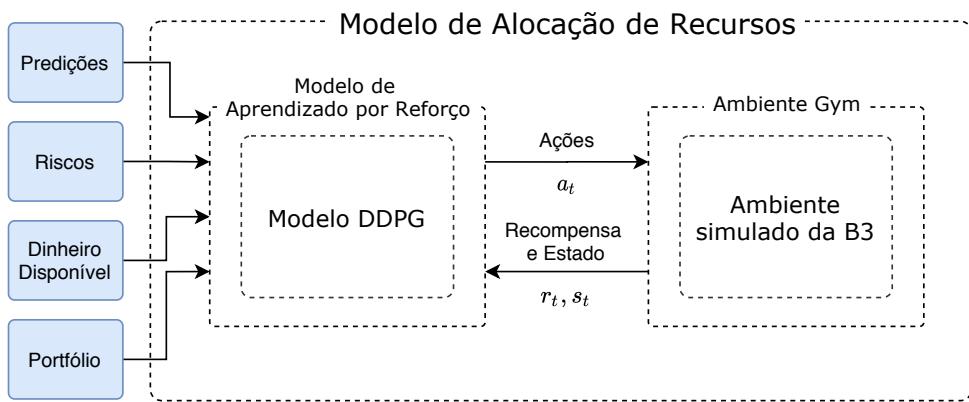


Figura 4.4: Visão Geral do Modelo de Alocação de Recursos. O modelo preditor escolhido pode ser alterado por qualquer modelo de aprendizagem desejado.

No entanto, algoritmos de aprendizagem por reforço possuem características base intrínseca ao funcionamento do modelo, dentre essas, as que precisam ser definidas são: espaço de estados, espaço de ações e recompensa. Quando defini-se essas características, pode-se dizer que foi criado um ambiente de aprendizagem na qual o agente vai interagir. O agente realizará uma ação  $a_t$  dentro do espaço de ações do ambiente, e receberá um estado  $s_t$  e uma recompensa  $r_t$  pela ação realizada. A Figura 4.4 apresenta como é realizada a interação do agente com o ambiente desenvolvido, bem como as ferramentas utilizadas. A criação deste ambiente foi realizada com a biblioteca *Gym* da *OpenAI*, que fornece um conjunto de ferramentas para desenvolver e comparar algoritmos de aprendizado por reforço.

## O Ambiente da [B]<sup>3</sup>

Utilizando o *Gym*, desenvolveu-se então um ambiente simulado da bolsa de valores, no qual o agente do Hare interage. Este ambiente tem como propósito treinar e testar o modelo do DDPG no MAR. Como supracitado, o ambiente precisa ter um espaço de estado, um espaço de ações e uma função de recompensa, os quais são definidos a seguir:

- O **Espaço de Estado** foi desenvolvido pensando nas variáveis do agente, junto com os valores de entrada do MAR. Portanto, o espaço contém a quantidade de dinheiro que o agente tem desalocado, e o quanto seu portfólio está rendendo em relação ao dinheiro inicial, como variáveis do agente. Complementando também o espaço, informações do módulo preditivo e do MGR são adicionadas. Para cada ativo no portfólio, há a possibilidade de conter a previsão, preço, lucro ou prejuízo do ativo, quantidade de ativos na carteira, e risco do ativo no tempo  $t$ . É dito que há a possibilidade, pois o ambiente foi programado de forma modular para permitir a experimentação com diversas modificações do espaço de estado.
- O **Espaço de Ações**, assume a forma definida pela Equação 4.1. Onde cada elemento  $a_{i,t}$  representa a ação que o agente deve fazer para o ativo  $i$  no tempo  $t$

$$A_t = [a_{0,t}, \dots, a_{n,t}] \quad (4.1)$$

O valor de uma ação  $a_{i,t}$  foi definido entre o intervalo de  $[-1, 1]$ , onde: entre  $[-1, 0)$  representa uma porcentagem do dinheiro em conta que será alocado na compra do ativo; 0 representa uma espera; e  $(0, 1]$  a venda de todas as posições que o agente possui do ativo.

- A **Recompensa** é atribuída para cada uma das possíveis ações realizadas pelo agente. Com o propósito de experimentar diversos cenários para aprimorar o aprendizado da DDPG, implementou-se as seguintes recompensas:
  1. **Ação de Venda Lucrativa:** Recompensa relativa a ganho ou perda na venda realizada. Para cada passo de tempo  $t$ , calcula-se quanto foi ganho ou perdido na venda daquele ativo em relação ao preço comprado.
  2. **Ação de Espera Lucrativa:** Diferença da quantidade de dinheiro na carteira no tempo  $t$  em relação a quantidade inicial.
  3. **Ação de Compra:** Recompensa constante negativa.
  4. **Ações Indisponíveis ou Prejudiciais:** Recompensa negativa atribuída toda vez que o agente falha a realizar uma ação selecionada. Essa recompensa leva em consideração a posição do agente no tempo da simulação, quanto mais para o fim dos seis meses, maior vai ser seu valor.
  5. **Fim do Episódio:** Diferença final da quantidade de dinheiro na carteira em relação a quantidade inicial.

Juntando todas essas recompensas, podemos descrever a recompensa final para o passo  $t$ . Para cada ativo no portfólio o agente receberá a recompensa respectiva a

ação realizada para aquele ativo. No final, o valor é dividido por uma constante  $c$ , evitando recompensas muito grande e facilitando os ajustes no modelo. No final do episódio, a diferença da quantidade de dinheiro final com a quantidade inicial é atribuída como recompensa final da simulação, indicando se houve um lucro ou uma perda naquele período.

## 4.6 Considerações Finais

Este capítulo propôs o Hare, um serviço autônomos de investimentos para negociar na bolsa de valores.

Neste capítulo, foi apresentado um serviço autônomos de investimentos para negociar na bolsa de valores, nomeado de Hare. Para isto, o Hare se favorece do uso de séries temporais de dados históricos de ativos, bem como o seu volume de consulta no *Google Trends* para prever e gerenciar o risco de tais ativos. Esses fatores servem como entrada para a inteligencia baseada em um agente racional do Hare, que busca maximizar a rentabilidade de um portfólio.

Resta-se então, avaliar o desempenho do Hare através de métricas e resultados. Os métodos de experimentação para a avaliação de desempenho são descritos no capítulo seguinte, bem como os resultados obtidos.

# Capítulo 5

## Resultados Experimentais

Neste capítulo, descreve-se a metodologia adotada para realizar os experimentos do Módulo Preditor de Movimento e do Modelo de Alocação de Recursos para validação do Hare. Para tanto, os experimentos realizados buscam clarificar as seguintes perguntas:

1. É possível prever o próximo movimento do ativo utilizando uma rede LSTM Hiper-parametrizada?
2. A rede LSTM Hiper-parametrizada é capaz de superar outros modelos presentes na literatura?
3. O algoritmo DDPG é capaz de aprender a gerenciar um portfólio, com um ou mais ativos, para gerar lucros no ambiente de mercado proposto?
4. O Hare consegue gerar um rendimento maior que opções de investimento de renda fixa mais seguros?
5. O Hare consegue alocar recursos em um portfólio de forma mais lucrativa que um portfólio alocado utilizando a Variância Média de Markowitz?

### 5.1 Metodologia

Cada solução possui suas particularidades, e consequentemente, a avaliação de desempenho para cada solução se torna individual para cada sistema distinto [67]. Com o propósito de avaliar os módulos presentes no Hare e responder as perguntas propostas, avaliações qualitativas e quantitativas foram realizadas.

O roteiro experimental validam o Hare em duas etapas: (i) validação do módulo preditor na Seção 5.2; e (ii) validação do MAR na Seção 5.3. Para validar o MPM realizou-se uma metodologia em três etapas: rotina de exploração de hiper-parâmetros, avaliação do desempenho dos resultados de treino e teste, e comparação com outros métodos para a

resolução do problema. Posteriormente validou-se o MAR analisando os modelos de cada um dos ativos e suas combinações, e os comparando com outras formas de investimento.

### 5.1.1 Base de Dados

Para realizar os experimentos supracitados, foi necessário a construção de uma base de dados com as informações necessárias. A base utilizada pelo Hare foi modelada manualmente através da plataforma oficial<sup>1</sup> da [B]<sup>3</sup>. A plataforma provê informações, desde 1986, sobre os ativos, tais como: nome da companhia, código de negociação, tipo de mercado, preço (abertura, fechamento, mínimo, máximo), e número de negociações feitas. Os diversos dados disponíveis na base foram tratados de acordo com a necessidade do experimento para adequá-lo a aplicação proposta. Seus tratamentos, quando realizados, são descritos dentro da seção do experimento respectivo.

## 5.2 Experimentos do Módulo Predictor

O primeiro conjunto de experimentos realizados, busca avaliar o desempenho o MPM e responder as perguntas 1 e 2 desta pesquisa. O processo de experimentação segue uma rotina de três passos. Começa-se explorando os hiper-parâmetros que melhor se adaptam a proposta da LSTM, utilizando métodos de otimização Bayesiana fornecidos pela biblioteca Hyperopt. Posteriormente, a rede LSTM calibrada com o Hyperopt, é colocada em processo de treino, avaliando sua consistência com uma série de experimentos. A rede treinada é então validada e então comparada com outros modelos de aprendizado.

Para a análise dos resultados experimentais foram utilizadas as seguintes métricas:

$$\text{sensibilidade} = \frac{\text{VP}}{\text{VP} + \text{FN}} \quad (5.1)$$

$$\text{precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} \quad (5.2)$$

$$\text{especificidade} = \frac{\text{VN}}{\text{VN} + \text{FP}} \quad (5.3)$$

$$\text{acurácia} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}} \quad (5.4)$$

$$\text{F1 Score} = 2 * \frac{\text{precisão} * \text{sensibilidade}}{\text{precisão} + \text{sensibilidade}} \quad (5.5)$$

---

<sup>1</sup>Disponível em: <http://www.b3.com.br/>

Cada métrica utilizada avalia um aspecto diferente dos resultados. A Equação 5.1, descreve a sensibilidade, o quão completo os resultados estão. A precisão, definida pela Equação 5.2, representa o quanto os resultados da pesquisa são úteis para reproduzibilidade. A Equação 5.3, descreve a especificidade, a proporção de resultados negativos corretamente identificados. A acurácia, definida pela Equação 5.4, é o quão aproximado os resultados estão do valor específico da realidade. Por último, a Equação 5.5, descreve a métrica *F1 Score*. Tal métrica leva em consideração a precisão e a sensibilidade para computar seu resultado. Seu valor representa a média harmônica entre essas métricas, onde o *F1 Score* alcança a melhor precisão e sensibilidade no valor 1 e a pior no valor 0.

Para gerar os modelos, a base de dados foi dividida em treino, validação e teste. O treino e validação correspondem a 70% dos seis meses contidos na base de dados. Os outros 30% foram utilizado para a fase de teste. Todos os experimentos utilizaram uma técnica para avaliar a capacidade de generalização do modelo, denominada validação cruzada. O método escolhido para essa validação foi o *Nested K-Fold*. Esse método consiste em dividir o conjunto total de dados em  $k$  subconjuntos. Realizado então o treino e teste  $k$  vezes. Cada vez que o  $k$  é incrementado a base de treino é aumentada e treina-se no subconjunto seguinte. Neste trabalho realizou-se a validação cruzada com  $k = 6$ , para a base de dados de seis meses.

Os experimentos foram conduzidos usando uma máquina virtual Linux hospedada na plataforma *Google Cloud*. As máquinas foram fornecidas com 4 CPUs e 3.8 GB de memória principal.

### 5.2.1 Pré Processamento da Base

O MPM do Hare utiliza uma base de dados de um semestre para treino, validação e teste. Foram usados os seguintes ativos para compor o portfólio do agente, sendo estes: VALE3, PETR3 e ABEV3. Além disso, utilizou-se o preço de fechamento da base de dados por ser uma reflexão de todos os movimentos do dia no mercado. Realizou-se então uma filtragem da base de dados com apenas o preço de fechamento nos seis primeiros meses de 2014 para cada um dos ativos no portfólio. No entanto, o preço de fechamento foi submetido a um processo de normalização, utilizando uma abordagem *Min-Max*, colocando os preços em intervalo de 0 a 1.

Em experimentos inciais observou-se que a quantidade de dados não era o suficiente para que a predição do modelo não fosse afetada pelos *outliers*<sup>2</sup>. Como solução para tratar esses “pontos fora da curva”, uma abordagem de duplicação dos dados foi efetuada, introduzindo a média entre dois valores consecutivos nos dados históricos. Sendo assim, a série

---

<sup>2</sup>Dados que se diferenciam drasticamente de todos os outros da base, pontos fora da curva

histórica apresenta mais dados, sem variações bruscas e sem perder seu comportamento original.

Para completar a base de dados para os experimentos do modelo preditor, a criação de rótulos indicadores de movimento do mercado, que atendam as necessidades dos modelos de treinamento. Os rótulos foram então criados utilizando uma abordagem discreta para cada ativo, em cada passo de tempo. Tal abordagem foi escolhida pois prever se um ativo vai ganhar ou perder valor é uma tarefa com maior probabilidade de acurácia do que prever o seu valor real. Sendo assim, os rótulos são determinados como 0 caso o preço de fechamento do dia foi menor que a média da janela de dias anteriores, ou como 1 caso o contrário.

### 5.2.2 Hiper-parametrização do Modelo

Com a base de dados preparada, pode-se começar a primeira fase experimental, a exploração dos hiper-parâmetros. A maioria dos algoritmos de aprendizado possuem um conjunto de variáveis previamente definidas antes do início do processo de treinamento, tais variáveis são os chamados de hiper-parâmetros. A escolha correta dos mesmos pode alterar significativamente o desempenho de um modelo. Portanto, otimizá-los de forma correta se torna um ponto essencial no processo de modelagem.

A metodologia de calibragem dos hiper-parâmetros é realizada em duas etapas. Começando com uma rotina de exploração de hiper-parâmetros utilizando métodos de otimização Bayesiana.

#### Metodologia de Hiper-parametrização

A rede LSTM possui diferentes valores de hiper-parâmetros para cada processo de modelagem. Nesse trabalho, foram selecionados quatro parâmetros da LSTM baseado nas calibrações que Nti, Adekoya e Weyori [13] e Chung e Shin [17] realizaram. Chung e Shin [17] calibraram a janela de tempo e número de unidades LSTM que seus modelos possuem. Por outro lado, Nti, Adekoya e Weyori [13] clarificam a importância dos algoritmos de otimização durante o processo de treinamento. Baseando-se nesses dois trabalhos foram escolhidos então os seguintes parâmetros a serem calibrados: janela de tempo, unidades LSTM, algoritmos de otimização e o *batch-size*.

Tendo selecionada o conjunto de hiper-parâmetros a serem explorados, a biblioteca Hyperopt<sup>3</sup> foi utilizada na aplicação dos métodos de otimização. Hyperopt é uma biblioteca em *Python* que implementa um algoritmo denominado, Otimização Sequencial Baseada em Modelo (SMBO), também conhecido como otimização Bayesiana. O SMBO

---

<sup>3</sup>Disponível em <https://github.com/hyperopt/hyperopt>

Tabela 5.1: Parâmetros utilizados nos experimentos.

Batch size	1	2	32	64	128	256
Unidades LSTM	1	50	80	100	150	200
Janelas	1	3	6	9	12	–
Otimizadores	Adam	SGD	RMSprop	–	–	–

é aplicável em situações que a minimização do valor para alguma função  $f(x)$  possui um alto custo devido a complexidade do método de avaliação. Tal método utiliza um algoritmo de busca, para determinar o conjunto de hiper-parâmetros, através de interações denominadas ensaios.

A biblioteca Hyperopt realiza sua otimização, baseando-se em três características principais a serem definidas pelo usuário: (i) uma função objetivo; (ii) um algoritmo de busca<sup>4</sup>; e (iii) o espaço de busca utilizado pelo o algoritmo. Definido estas características, a seleção de valores é utilizada para encontrar o melhor conjunto de hiper-parâmetros para o modelo.

## Resultados da Hiper-parametrização

A condução dos experimentos começou pela utilização da biblioteca Hyperopt, com o intuito de descobrir os melhores hiper-parâmetros para o modelo LSTM proposto. A realização deste experimento necessita, no entanto, das definições das características de funcionamento do Hyperopt. Neste trabalho, definiu-se as características da seguinte forma: (i) a função objetivo selecionada foi o *F1 Score* retornado pelo modelo LSTM, previamente apresentado na Equação 5.5; (ii) o algoritmo de busca escolhido foi o TPE; e (iii) o espaço de busca selecionado são os valores escolhidos para serem explorados para cada um dos parâmetros previamente apresentados, definidos na Tabela 5.1.

Foram realizados três experimentos do Hyperopt, um para cada ativo no portfólio (PETR3, VALE3, ABEV3). Cada experimento foi realizado com 1000 ensaios para um ativo específico, utilizando como dado de entrada o preço de fechamento de cada dia útil do primeiro semestre de 2014. O processo de treinamento foi feito com processos paralelos, e um histórico foi gerado para permitir futuras análises das escolhas de parâmetros feita pelo algoritmo. Os resultados de cada experimento, com os melhores parâmetros encontrados para cada ativo, é apresentado na Tabela 5.2.

Com os resultados obtidos, foi observado a necessidade de existir um modelo específico para cada ativo, dado que seus hiper-parâmetros são diferentes e possivelmente a forma que o modelo analisa os padrões da série temporal também é diferente. Os resultados

---

<sup>4</sup>No período em que esse trabalho foi realizado, a biblioteca oferecia apenas duas opções: Busca aleatória e o algoritmo Tree Parzen Estimator (TPE).

Tabela 5.2: Melhor conjunto de parâmetros encontrados para cada ativo.

	VALE3	PETR3	ABEV3
Batch Size	2	2	128
Unidades LSTM	200	80	1
Janelas	6	9	6
Otimizadores	RMSprop	RMSprop	Adam
Melhor F1 Score	0.78781	0.83006	0.65109

obtidos no estudo dos hiper-parâmetros para o ativo da VALE3 mostram que sua melhor pontuação de 0.7878, é resultado da combinação de um *batch size* de 2, 200 unidades LSTM, 6 dias de janela e a utilização do otimizador RMSprop. Para o caso do ativo PETR3 a maior pontuação obtida foi de 0.83006 sendo resultante da combinação de 2 de *batch size*, 80 unidades LSTM, 9 dias de janela e *RMSprop* como otimizador. Por último, o ativo ABEV3, obteve um *F1 Score* de apenas 0.65109, com 128 de *batch size*, 1 unidade LSTM, 6 dias de janela, e Adam como otimizador.

Além de observar os melhores resultados obtidos através do Hyperopt, uma análise mais aprofundada foi realizada para entender o comportamento do algoritmo para cada ativo, os resultados são apresentados nas Figuras 5.1 a 5.3. A figura da esquerda apresenta quatro histogramas, um para cada hiper-parâmetro, com os valores do parâmetro e quantidade de vezes que cada valor foi selecionado pelo algoritmo do Hyperopt. Na figura da direita é apresentado o valor da função objetivo escolhida, *F1 Score*, durante os 1000 ensaios do algoritmo. Para auxiliar a visualização é adicionado um ajuste baseado em um polinômio de primeiro grau, para traçar uma reta capaz de representar a tendência dos dados.

A Figura 5.1, representa os resultados do experimento para os ativos PETR3. Quando analisado seu histograma, pode-se observar que os parâmetros mais selecionados, com exceção da janela, não correspondem aos melhores parâmetros escolhidos pelo Hyperopt demonstrados na Tabela 5.2. O que pode indicar que o algoritmo de seleção dos hiper-parâmetros busca priorizar a exploração de valores que obtiveram resultados piores, do que aproveitar de primeira os melhores resultados encontrados. No entanto, mesmo os melhores parâmetros escolhidos não corresponderem a quantidade, observa-se através do ajuste de polinômio que o *F1 Score* converge para resultados melhores.

Por outro lado, ao observar a Figura 5.2, nota-se no histograma, que o *batch size*, unidades LSTM e otimizadores mais selecionados correspondem as escolhas de melhores parâmetros do Hyperopt para o modelo da VALE3, divergindo apenas com a janela. Tal comportamento pode ser justificado pelo algoritmo de exploração de parâmetros, encontrar em sua maioria das vezes, valores de *F1 Score* satisfatórios. Observa-se também que o ativo VALE3 provou possuir o modelo mais consistente, com base no *F1 Score* em

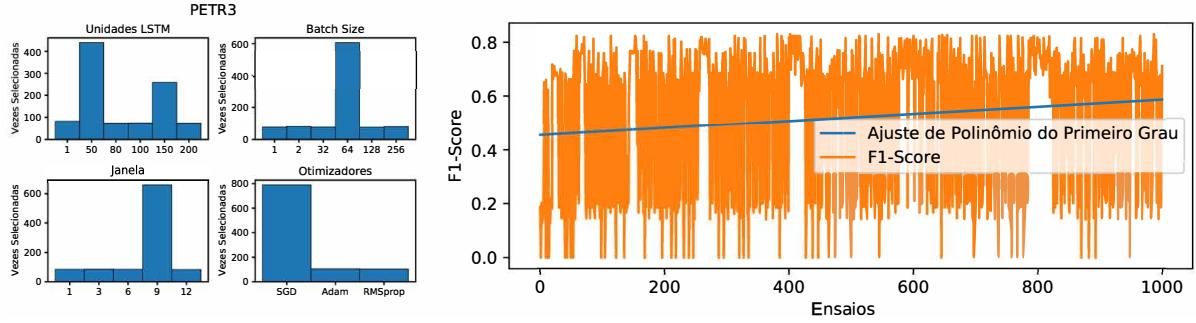


Figura 5.1: Histograma de parâmetros selecionados e progressão do *F1 Score* no ativo PETR3.

comparação com os ativos PETR3 e ABEV3.

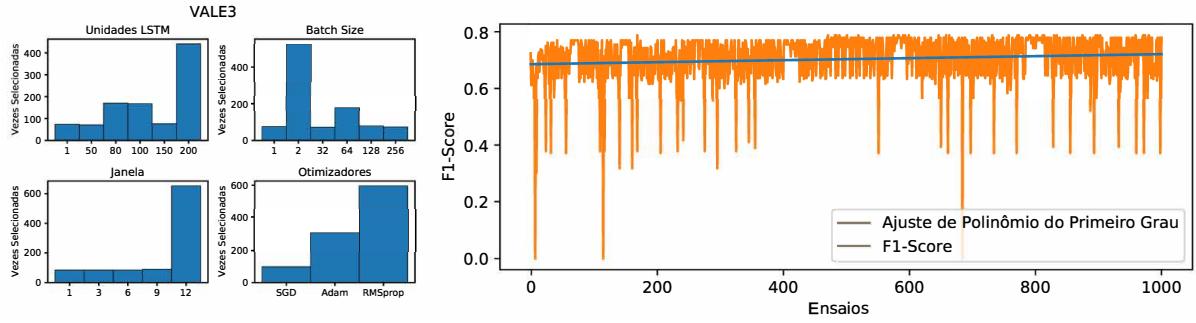


Figura 5.2: Histograma de parâmetros selecionados e progressão do *F1 Score* no ativo VALE3.

Similarmente, o ativo ABEV3, apresentado na Figura 5.3, difere os valores mais selecionados dos melhores valores encontrados pelo tamanho da janela e adicionalmente pelo *batch size*. Ao analisar o *F1 Score* deste experimento, observa-se que não somente foi o pior obtido, como também foi bem abaixo dos outros valores. Este comportamento ocorre por ser o primeiro ano do ativo na bolsa, que haveria trocado seu código e seus papéis após uma reestruturação societária no final de 2013.

Portanto, nota-se que os resultados obtidos sempre demonstram um aumento da inclinação na função objetivo dos três ativos. Sendo assim, observando a representação do *F1 Score* pelo polinômio de primeiro grau apresentado, nota-se que de fato há um aumento no valor com o passar dos ensaios, demonstrando que o processo de hiper-parametrização apresenta melhorias. Conclui-se também que os parâmetros mais selecionados não necessariamente indicam que farão parte do grupo de melhores parâmetros ao final do experimento, o que pode indicar a tentativa do algorítimo de explorar novas combinações.

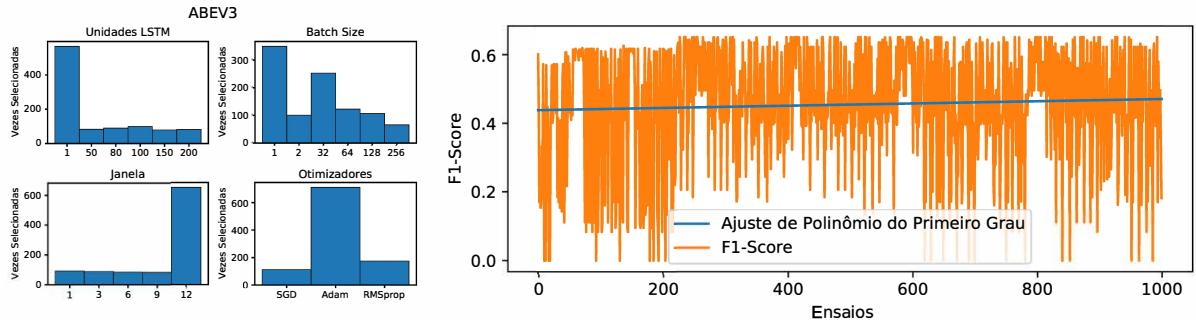


Figura 5.3: Histograma de parâmetros selecionados e progressão do *F1 Score* no ativo ABEV3.

### 5.2.3 Treinamento e Desempenho do Modelo

Após a hiper-parametrização dos modelos propostos, conduziu-se experimentos para treinar os melhores modelos. Tais experimentos de treino levam em consideração o comportamento da função de perda e da consistência de acurácia para diferentes janelas de tempo.

Para cada ativo, um novo experimento foi executado utilizando os melhores resultados de hiper-parâmetros fornecidos pelos testes realizados anteriormente. O experimento consiste em um K-Fold de validação cruzada, com  $k = 6$ , para cada janela de tempo apresentada na Tabela 5.1. Os resultados são apresentados nas Figuras 5.4 a 5.6.

A Figura 5.4 demonstra o treinamento do ativo da PETR3 que apresentou o maior *F1 Score*, de 0.83006, na hiper-parametrização. Os resultados obtidos demonstram que a janela pode impactar de forma significante o modelo. Observa-se que janelas de tamanho 1 e 3 são as de piores desempenho. Já as janelas 6, 9 e 12 apresentam resultados melhores. Dentre estes, a janela de tamanho 6 foi escolhida para ser utilizada. Apesar da dispersão do modelo ser maior que o de tamanho 12, seu limite superior é bem maior, chegando inclusive a 100% em algumas situações. Além de apresentar também um limite inferior bem maior que a janela de tamanho 9. Outra característica também importante para a seleção deste tamanho de janela foi a mediana, que é a mais simétrica entre as janelas. Considerando então a janela de tamanho 6, seus resultados de predição são apresentados na matriz de confusão. Com as previsões realizadas pelo modelo, se obteve as seguintes métricas: acurácia de 0.803, especificidade de 0.7939, precisão de 0.8 e sensibilidade de 0.812. Observa-se que todas as métricas apresentam valores altos, portanto o modelo gerado possui 81,2% de resultados completos, é 80% reproduzível, possui uma taxa de acerto das previsões de 80,3% e identifica corretamente 79% dos decréscimos de valor no preço de um ativo.

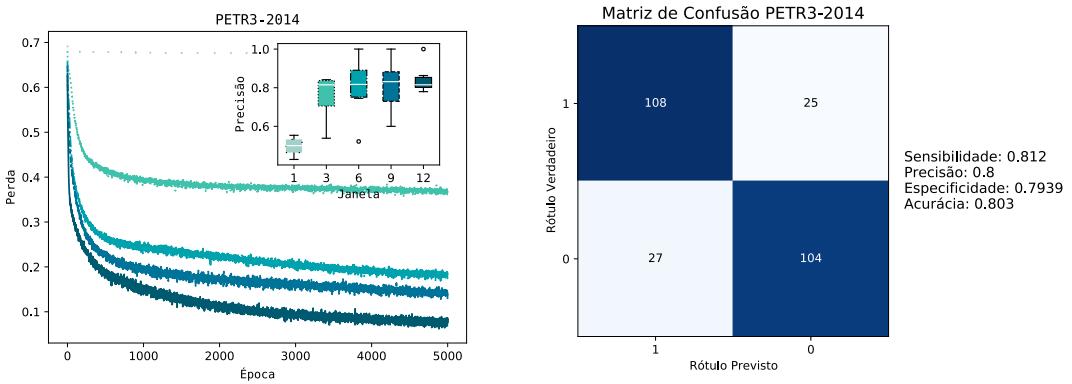


Figura 5.4: Função de perda e matriz de confusão do ativo PETR3.

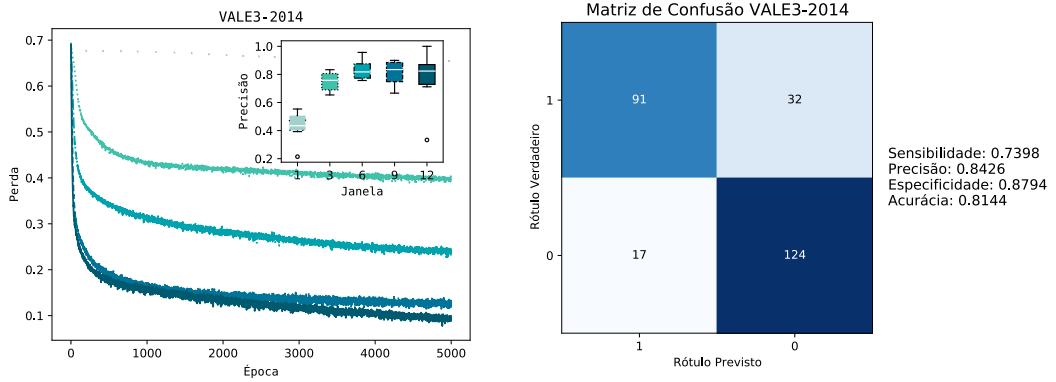


Figura 5.5: Função de perda e matriz de confusão do ativo VALE3.

Finalizando a análise do processo de treinamento, as Figura 5.5 e Figura 5.6 demonstram os resultados para os modelos da VALE3 e ABEV3, respectivamente. Os modelos para esses ativos foram os que obtiveram as menores métricas F1 Score no processo de hiper-parametrização. Entretanto, a concisão de ambos os modelos se apresentaram superiores aos modelos da PETR3, possuindo uma menor dispersão se compararmos janelas de tamanhos iguais. Assim como na PETR3, as janelas de tamanho 1, 3 podem ser desconsideradas para os modelos da VALE3, e para a ABEV3, os tamanhos 1, 3 e 12 por apresentarem resultados piores. Dentre as janelas restantes o tamanho 6 foi escolhido para o modelo da VALE3, por apresentar um limite inferior menor e para o modelo da ABEV3 foi escolhido o tamanho de janela 9, por ser bastante conciso permitindo uma maior chance de reproduzibilidade. Observando-se particularmente as previsões realizadas pelo modelo da VALE3 com janela de tamanho 6 obteve-se os valores das seguintes métricas: acurácia de 0.8144, especificidade de 0.8794, precisão de 0.8426 e sensibilidade de 0.7398. Observa-se que quase todas as métricas obtiveram resultados melhores que

o modelo da PETR3, com exceção da sensibilidade. Portanto, este modelo apresenta um resultado menos completo (73,9%), mas uma maior reproduzibilidade (84,2%), maior taxa de acerto (81,4%) e maior taxa de identificação de decréscimos (87,94%). Por fim com os resultados então obtidos na matriz de confusão para o modelo com o tamanho de janela 9 da ABEV3, obteve-se as métricas de análise. Os valores obtidos para acurácia (0.8333) e sensibilidade (0.8871) foram os maiores em comparação com os outros ativos. Indicando a maior taxa de acerto (83,3%) e a maior completude dos resultados (88,71%), resultado provavelmente proveniente da escolha do modelo mais conciso. No entanto, sua especificidade é a menor obtida com o valor de 0.7787, o que indica que a capacidade do modelo prever uma queda do preço do ativo é de 77,8%. Finalizando, a precisão de 0.8029, se apresenta com um valor parecido do modelo da PETR3, indicando a capacidade do modelo de reproduzibilidade em 80,2%.

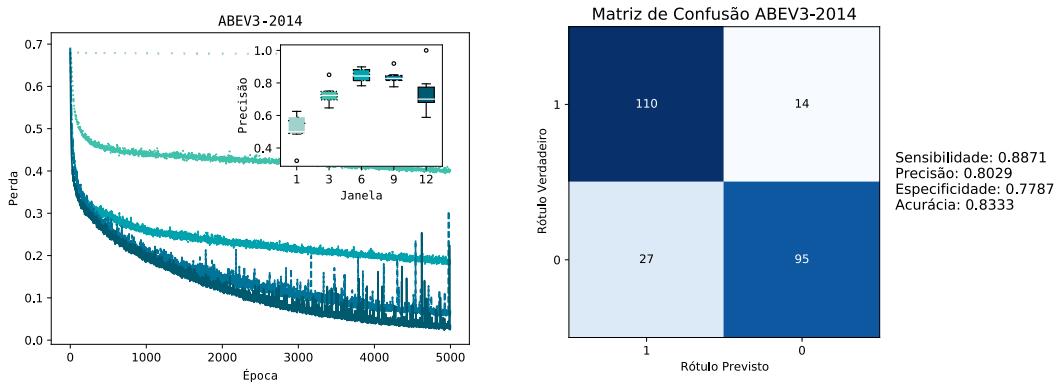


Figura 5.6: Função de perda e matriz de confusão do ativo ABEV3.

Com a análise dos resultados das Figuras 5.4 a 5.6 pode ser notado a importância do parâmetro da janela de tempo, podendo impactar significativamente a acurácia do modelo. As janelas com melhor desempenho apresentaram valores altos nas matrizes de confusão mostrando a eficiência na escolha dos hiper-parâmetros. Avaliando as matrizes de confusão, percebe-se que todos os modelos apresentaram resultados de predição satisfatórios. As métricas e o tamanho de janela resultantes para os modelos são resumidos na Tabela 5.3. Portanto, esses resultados demonstram a viabilidade do uso de modelos LSTM hiper-parametrizados na predição do movimento de ativos no Hare.

Em seguida, utilizando os melhores modelos, uma análise real de mercado foi realizada. O objetivo é entender o comportamento das previsões de cada um dos ativos, validando as métricas encontradas nos experimentos anteriores. Os resultados são apresentados nas Figuras 5.7 a 5.9. Os gráficos apresentam a variação do preço de fechamento do ativo ao longo do tempo (linha rosa) e a variação do preço médio na janela de tempo selecionada para o ativo (linha azul). Os marcadores presentes na linha azul representam as previsões

Tabela 5.3: Resumo dos resultados obtidos.

	PETR3	VALE3	ABEV3
Janela	6	6	9
Acurácia	80,30%	81,44%	<b>83,33%</b>
Precisão	80%	<b>84,26%</b>	80,29%
Sensibilidade	81,20%	73,98%	<b>88,71%</b>
Especificidade	79,39%	<b>87,94%</b>	77,87%

de preço em relação a média de dias anteriores, a seta verde para cima representa que a predição de aumento foi correta, enquanto a seta vermelha para baixo representa a predição correta. Predições incorretas são representadas por um  $x$  verde ou vermelho para uma aumento ou decréscimo respectivamente.

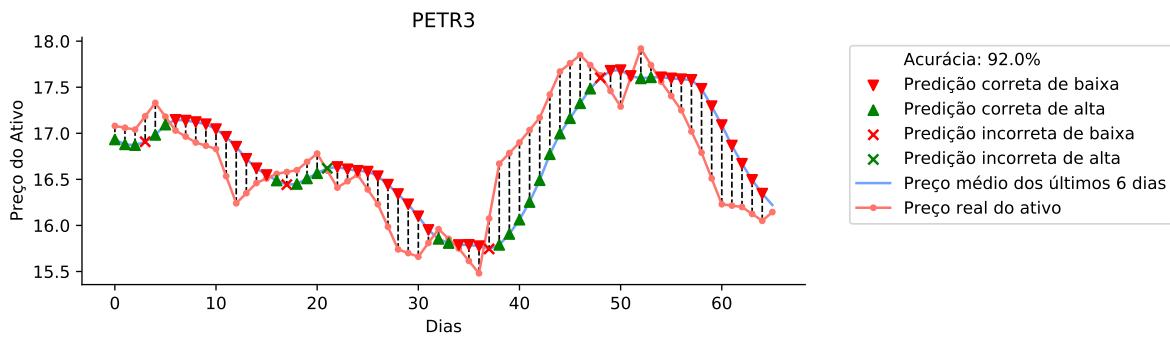


Figura 5.7: Predições do ativo PETR3.

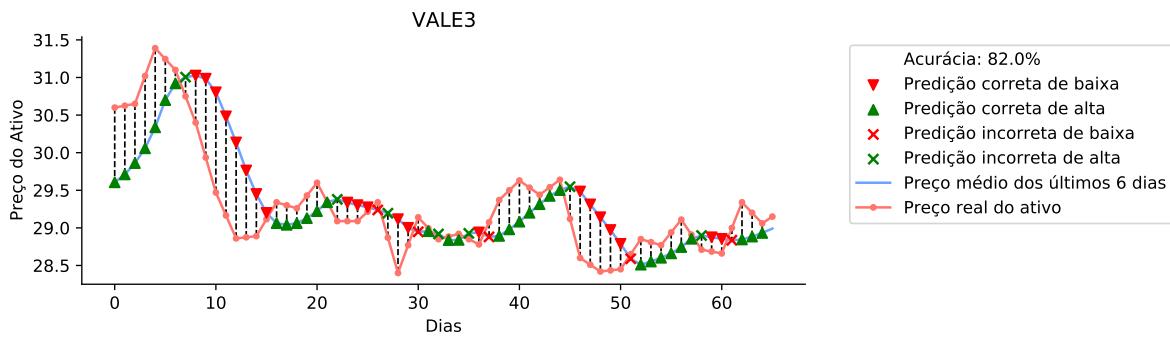


Figura 5.8: Predições do ativo VALE3.

Como se pode observar a acurácia foi satisfatória para todos os ativos avaliados. No pior caso, a acurácia foi de 82% para o ativo da VALE3 (Figura 5.8). Em contraste, o ativo PETR3 obteve 92% (Figura 5.7) de acerto, seguindo pelo ABEV3 com 94% (Figura 5.9). Tais resultados demonstram a eficiência do modelo LSTM hiper-parametrizado proposto,

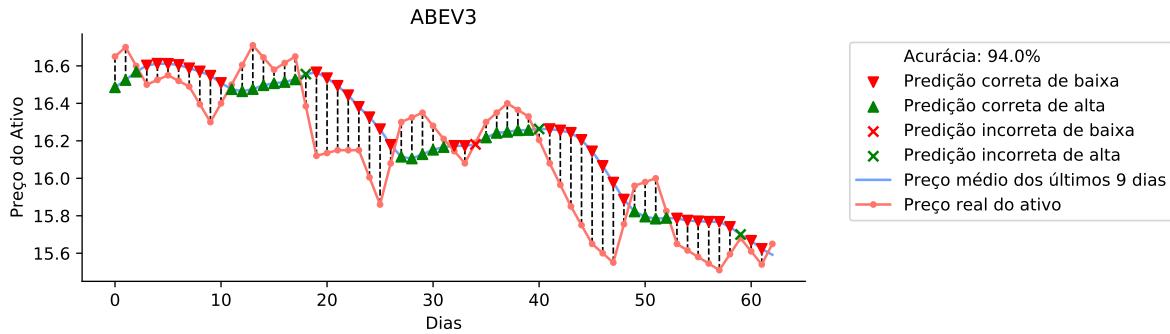


Figura 5.9: Previsões do ativo ABEV3.

e respondem a Pergunta 1 proposta. Observa-se também nos resultados que a maioria dos erros preditivos acontecem em transições de movimento, o que pode ser considerado uma possível limitação da LSTM. Portanto, modelos aplicados a ativos de alta volatilidade, podem se tornar um desafio extra para o módulo preditor, reforçando assim a necessidade de um modulo que estude as características fundamentais para analisar riscos.

#### 5.2.4 Comparação com Modelos de Base

Mesmo com o modelo LSTM proposto apresentando resultados promissores, foi realizada uma análise comparativa com outros modelos tradicionais de aprendizado de máquina, sendo eles: (i) KNN; (ii) SVM; (iii) RNN; e (iv) GRU. Além disso, é realizado uma análise comparativa com uma rede LSTM sem o processo de hiper-parametrização para identificar a diferença entre os modelos. Os modelos selecionados foram submetidos a um experimento similar ao realizado para a LSTM hiper-parametrizada, utilizando a série histórica dos seis primeiros meses de 2014 do ativo PETR3. O ativo PETR3 foi selecionado para o método comparativo entre os modelos, pois obteve o pior desempenho em relação aos outros ativos, como apresentado na Tabela 5.3.

Tendo em vista que o experimento é uma comparação de algoritmos, não seria razoável comparar um modelo hiper-parametrizado, com modelos não parametrizados. Os modelos comparativos precisam possuir também algum tipo de parametrização para se mostrarem viáveis a aplicação. O modelo SVM foi parametrizado em relação ao *kernel* RBF, calibrando o parâmetro de regularização C para permitir que hiperplano fica mais suave evitando situações de sobreajuste, e um parâmetro gamma que define até que ponto a influência de um único exemplo de treinamento atinge. No caso do KNN os experimentos foram realizados com valor de  $k$  fixado em 1. No entanto, para adaptá-lo ao problema de séries temporais, o modelo foi parametrizado com a métrica de distância Distorção Dinâmica de Tempo (DTW). Para as redes neurais profundas, RNN, GRU e LSTM a

Tabela 5.4: Resumo comparativo dos métodos.

	SVM	KNN	RNN	LSTM	GRU	LSTM Proposta
Acurácia	55,17%	55,17%	67,74%	70,97%	41,94%	<b>80,30%</b>
Precisão	47,73%	47,73%	87,50%	<b>93,33%</b>	83,33%	80%
Sensibilidade	<b>87,50%</b>	<b>87,50%</b>	63,64%	63,64%	22,73%	81,20%
Especificidade	32,35%	32,35%	77,78%	<b>88,89%</b>	<b>88,89%</b>	79,39%
Média das Métricas	55,68%	55,68%	74,78%	79,20	59,22%	<b>80,22%</b>

janela de tempo de tamanho 6, igual a do modelo proposto, foi utilizada. As redes foram todas treinadas com uma unidade por um total de 5000 épocas. A Tabela 5.4 resume os resultados encontrados por todos os modelos comparativos e o resultado do modelo proposto, para o ativo PETR3.

Iniciando a análise pelos métodos clássicos, a SVM<sup>5</sup> e o KNN<sup>6</sup> obtiveram resultados semelhantes, com uma acurácia de 55%, muito abaixo do modelo proposto. Acredita-se que tal resultado pode ser fruto do desequilíbrio entre as classes de alta e baixa na série temporal. Em relação aos os resultados das rede neurais, pode-se observar que o modelo da LSTM, mesmo sem hiper-parametrização, se sobressai perante os demais em todas as métricas, empatando somente na sensibilidade com o modelo da RNN. Em relação ao modelo LSTM proposto, apesar de não apresentar o melhor resultado para todos as métricas, é o que apresenta a melhor acurácia, com vantagem de aproximadamente 10%, além da média de suas métricas ser relativamente melhor que a dos outros modelos.

Portanto, é visível que o modelo proposto, excede o desempenho de outros modelos de aprendizagem de máquina. Observa-se também que a hiper-parametrização aumenta consideravelmente o resultado do modelo LSTM gerado, respondendo a Pergunta 2 deste trabalho. Com esses resultados encerra-se os experimentos relacionados ao módulo preditivo. Faltando então analisar o desempenho de como o módulo alocador de recursos trata as informações previstas.

### 5.3 Experimentos de Alocação de Recursos

Após a realização dos experimentos do módulo preditor, para responder as perguntas 3, 4 e 5, realizou-se os experimentos do módulo de alocação de recursos. Finalizando assim a validação da proposta de um sistema de investimento autônomo, racional baseado em agentes, capaz de lidar com previsões e alocações de recursos de forma apropriada.

A proposta definida para a criação deste módulo, envolve a utilização do algoritmo DDPG de aprendizagem com reforço, junto com o ambiente simulado da [B]<sup>3</sup> criado.

---

<sup>5</sup>Parametrizado com  $C = 0.1$  e  $\gamma = 1$

<sup>6</sup>Parametrizado utilizando a métrica DTW

Tabela 5.5: Parâmetros utilizados na DDPG.

Parâmetro	Valor Selecionado
Épocas	200
Passos por Época	1000
Taxa de Aprendizagem da Política	0.0005
Taxa de Aprendizagem do Q-Valor	0.0001
Buffer de Repetição	500.000
Batch Size	100
Passos Iniciais	10
Ruído	1.0
Tamanho da Camada Escondida	(16, 16)
Função de Ativação da Camada Escondida	ReLU
Função de Ativação da Saída	Tanh
Episódios de Teste	10

A versão implementada do algoritmo da DDPG possui uma série de hiper-parâmetros a serem adaptados para a proposta. Dentre esses, citam-se os que foram utilizados: tamanho do buffer de repetição, fator de desconto, taxa de aprendizagem da política, taxa de aprendizagem do Q-Valor, *batch size*, passos iniciais, ruído, tamanho da camada escondida da rede ator-critico, e camadas de ativação da rede ator crítico.

O algoritmo do DDPG foi então calibrado de forma empírica, com valores que apresentaram resultados satisfatórios em experimentos prévios realizados. Os valores encontrados na calibração são resumidos na Tabela 5.5

Assim como nos experimentos do módulo preditor, os experimentos realizados no modelo de gerenciamento de recursos utilizaram dados do primeiro semestre de 2014. Para evitar a propagação do erro no modelo de treinamento, os rótulos de predição verdadeiros foram alimentados, não utilizando a predição realizada pelo módulo preditor, nem pelo MGR. Os experimentos foram realizados de forma crescente em complexidade, realizando primeiro experimentos individuais, e finalizando com o experimento do portfólio. Como os dados do módulo de gerenciamento de risco são fontes de uma abordagem sintética, com o propósito primário de auxiliar no entendimento do funcionamento do Hare, os experimentos aqui realizados não os levam em consideração. No entanto, um experimento extra, foi realizado e está descrito no Apêndice B deste trabalho.

A análise dos resultados é realizada de forma comparativa com a rentabilidade da poupança<sup>7</sup> e do Certificado de Depósito Interbancário (CDI)<sup>8</sup>, os quais renderam 3,35% e 4,87% respectivamente no período avaliado. Outra comparação realizada foi com o rendimento do portfólio caso o investidor tenha realizado uma estratégia *buy and hold*,

---

<sup>7</sup>Informações retiradas de: <http://www.yahii.com.br/poupanca.html>

<sup>8</sup>Informações retiradas de: <http://www.yahii.com.br/cetip13a21.html>

onde o investidor decide comprar um ativo e “segura-lo” para o longo prazo, de forma a se beneficiar com os rendimentos e valorizações que o papel por ventura apresentará no futuro.

### 5.3.1 Alocação dos Ativo Individuais

O primeiro tipo de experimento realizado pelo MAR foi utilizando ativos individuais, para analisar como a DDPG funcionaria em ambientes mais simples. Desta forma três experimentos foram realizados, um para cada ativo. Cada experimento utilizou o melhor parâmetro de janela encontrado pelo módulo preditor e considerou R\$10.000 de investimentos iniciais. Os modelos foram treinados com 200 épocas e 10 episódios de teste por época. As métricas utilizadas foram o retorno e o lucro obtido no treino e no teste, as ações tomadas, o comportamento das funções de perda, e o comportamento do Q-Valor. Nesta seção descreve-se o resumo dos resultados encontrados, uma análise completa contendo todos os gráficos obtidos em treino se encontra disponível no Apêndice A.

Em relação ao retorno obtido no treinamento e no teste, observou-se que mesmo tendo lucros o retorno se mantinha majoritariamente negativo, tal resultado pode ser reflexo da formulação da função de recompensa. O valor de retorno final mais alto em teste foi de 82.272 para Petrobras, enquanto o menor foi da Ambev de -68.70. Continuando a análise do treinamento, observou-se que as funções de perda da política e do Q-Valor convergiram, mas não chegaram próximo de 0, o que pode indicar que a DDPG convergiu para um mínimo local ou para uma situação de *deadlock* [68]. Partindo então para a análise do teste, percebeu-se que muitas vezes o agente realizaria uma ação específica, ao invés de alternar entre compras, vendas e esperas<sup>9</sup>. No entanto, mesmo com esse comportamento, resultados satisfatórios ainda foram encontrados, obteve-se lucros de R\$1377,00, R\$603,00, e R\$0,00 para os ativos PETR3, VALE3 e ABEV3 respectivamente. A Tabela 5.6 apresenta os lucros, rentabilidades e rendimentos comparativos em relação a poupança e ao CDI.

Observa-se que os modelos da PETR3 e VALE3 encontraram resultados melhores, com uma rentabilidade de 13,77% e 6,03% respectivamente. Tais rentabilidades são maiores que a da poupança do CDI e de um investimento seguindo uma estratégia *buy and hold*, chegando a render mais que o dobro desses investimentos para o ativo da Petrobras. Por outro lado, o modelo treinado para ABEV3 não apresenta nenhum tipo de lucro, consequentemente, não tendo rentabilidade. No entanto, observa-se que o ativo desvalorizaria 9,3% no período analisado, indicando que a estratégia aprendida pelo agente, apesar de não ser ótima, consegue evitar prejuízos ao investidor.

---

<sup>9</sup>Vale enfatizar que nem toda ação selecionada é de fato executada. Se uma ação de compra é requerida, e o agente não possui a quantidade de dinheiro para a comprar o ativo, a mesma não vai ser executada, e o agente recebe uma punição pela ação realizada de forma incorreta. A mesma lógica se aplica de forma similar para as vendas

Tabela 5.6: Resultados obtidos nos investimentos realizados pelo Hare em comparação com poupança, CDI e estratégia *buy and hold*.

		PETR3	VALE3	ABEV3
Análise da Estratégia Buy and Hold	Lucro	R\$624,00	-R\$957	-R\$730
	Rentabilidade	6,24%	-9,57%	-7,30%
Análise de Desempenho do Hare	Lucro	R\$1377,00	R\$603,00	R\$0,00
	Rentabilidade	13,77%	13,77%	0%
	Rendimento Comparado à Poupança	311,04%	80%	-
	Rendimento Comparado ao CDI	182,75%	23,81%	-

### 5.3.2 Alocação do Portfólio Completo

Todos os experimentos analisados na seção anterior utilizam apenas um ativo específico, uma contradição a Teoria Moderna do Portfólio, que diz que os fatores devem ser calculados e analisados em relação ao conjunto de ativos [23]. Esse experimento busca então analisar o desempenho do agente com um portfólio diversificado de ativos. O portfólio foi então construído com os ativos utilizados nos experimentos anteriores (PETR3, VALE3 e ABEV3) e suas janelas respectivas. Com o propósito de aumentar a quantidade de ativos que o agente consegue investir, a quantidade de dinheiro inicial foi definida como R\$50.000. As Figuras 5.11 a 5.15, demonstram os dados obtidos em 200 épocas de experimentos. A Figura 5.11 apresenta o comportamento do retorno médio obtido por episódios de treino e teste. Observe que diferente dos ativos individuais o retorno chega a assumir valores bem maiores, resultado consequente da maior quantidade de recursos inserida no sistema. Nota-se também que é visível a convergência do retorno médio de treinamento ao longo do treino e teste, obtendo um valor final de 404.24 para ambos treino e teste.

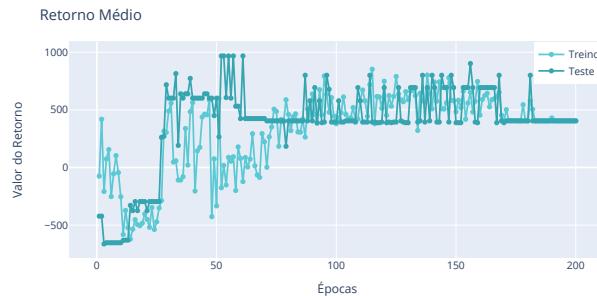


Figura 5.10: Portfólio - Retorno médio.

Analisa-se também o Q-Valor médio e as funções de perda do treinamento, com base nas Figuras A.2 a A.3. O Q-Valor alcança uma média de 230,84, variando entre 1633,95 a -594,62, enquanto as funções de perda assumem valores de 2.357,02 e -232,13 para perda

$Q$  (função de aproximação do  $Q$ -Valor) e perda  $\pi$  (função de aproximação da política) respectivamente. Nota-se que os valores de perda são maiores que dos ativos individuais, o que representa uma correlação entre a recompensa maior obtida, devido a maior quantidade de recursos no sistema. Apesar desses valores ainda serem extremamente altos para funções de perda, experimentos com maiores números de época não apresentaram resultados diferentes ao comportamento.

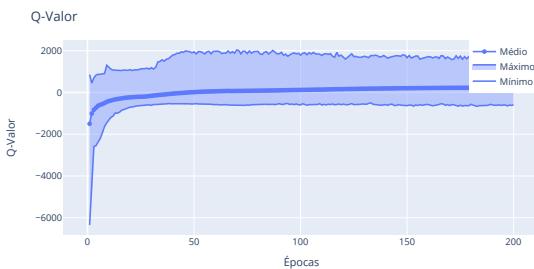


Figura 5.11: Portfólio - Comportamento do Q-Valor.

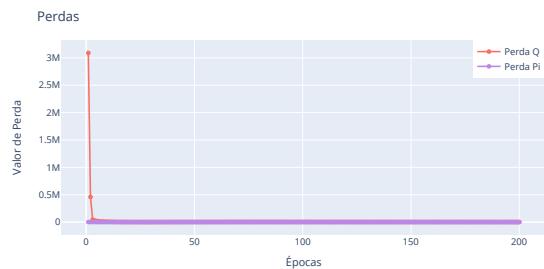


Figura 5.12: Portfólio - Comportamento das funções de perda.

Em relação ao lucro obtido, nota-se na Figura 5.13, que durante o treino se obteve uma alta dispersão de valores, incluindo um lucro de R\$23.860 em um dos episódios. Novamente, no entanto, o algoritmo não explora essas recompensas maiores. Esse comportamento também foi observado nos outros experimentos, o estudo do mesmo abre possibilidades para trabalhos futuros que podem ser capazes de melhorar显著mente os modelos. Se o agente tivesse explorado o melhor lucro encontrado, teria obtido um rendimento de aproximadamente 47% no período, quase 8% ao mês.

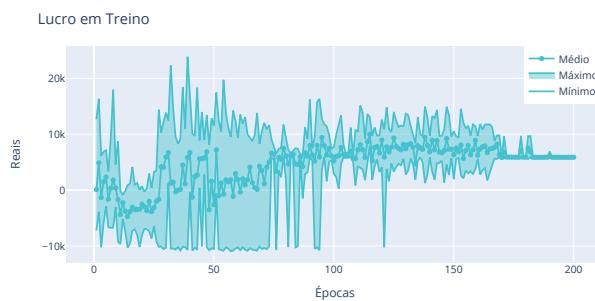


Figura 5.13: Portfólio - Lucro em treino.

Quando o resultado do lucro é observado em teste (Figura A.5), percebe-se oscilações menores (por não conter ruído em suas ações), e nota-se que também não é explorada a política com maior lucro encontrada. No entanto, mesmo com essas limitações presentes no modelo proposto, ao final do treinamento obteve-se um lucro médio de 5.874 reais, realizando ações majoritariamente de venda, comprando algumas vezes, e nunca realizando a ação de espera (veja a Figura 5.15). Observa-se que a ação de espera raramente

é selecionada em experimentos, sendo de ativos individuais ou do portfólio completo, esse comportamento provavelmente é fruto da função de recompensa, cabendo a trabalhos futuros investigar como adaptar a função para melhorar o sistema.

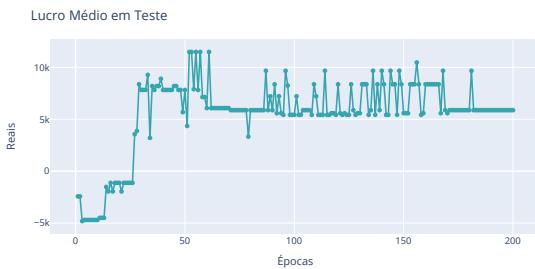


Figura 5.14: Portfólio - Lucro médio em teste.

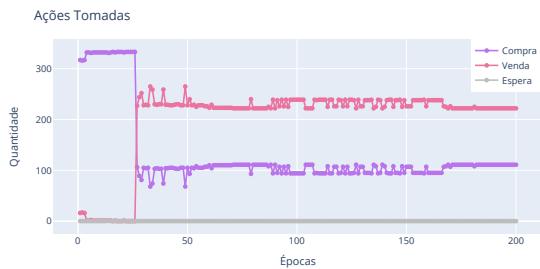


Figura 5.15: Portfólio - Quantidade de ações selecionadas em teste.

Portanto, nota-se que o modelo, apesar de todas as adversidades encontradas, obteve um lucro de R\$5.874,00 no final do teste, uma rentabilidade de 11,74% no total dos 6 meses. A rentabilidade encontrada é 250,44% maior que da poupança e 141,06% maior que o CDI. Nota-se que a rentabilidade encontrada é um pouco menor que a rentabilidade do ativo PETR3. Essa diferença pode ser justificada pelo fato do ativo da Petrobras ser o único que valoriza no período avaliado, sendo possível que o modelo entenda o risco de investir nos outros ativos. Com os resultados apresentados, responde-se positivamente as Perguntas 3 e 4 desse trabalho.

A análise do investidor seguindo uma estratégia *buy and hold*, para essa situação demanda um estudo um pouco mais complicado. Essa complicaçāo é devido ao fato de existir infinitas maneiras no qual um investidor pode montar seu portfólio com três ativos. Para o propósito desse trabalho, o construiu-se um portfólio comparativo considerado ótimo, seguindo a teoria de Markowitz. O processo em construção desse portfólio leva em consideração a quantificação do retorno e o risco individual de cada ativo para posteriormente achar uma fronteira eficiente que tenha a maior possibilidade de retorno para a quantidade de risco que se deseja assumir.

### Análise Comparativa Com Fronteira Eficiente

A Figura 5.16 apresenta as combinações de portfólios existentes em relação ao retorno esperado para o primeiro semestre de 2014 e o risco assumido pelo mesmo. Para calcular esses valores, é utilizado como base os preços de fechamento mensais ajustado de cada ativo<sup>10</sup>, de 2011 a 2013. Com os preços, calcula-se então os retornos e riscos mensais, informações utilizadas para compor a carteira. A cor de cada combinação de portfólio é respectiva a um indicador que ajuda a descobrir um ponto de equilíbrio entre risco e

<sup>10</sup>Dados retirados do site: <http://finance.yahoo.com>

retorno, denominado Índice de Sharpe. Esse indicador mede o retorno excedente de uma aplicação financeira em relação a outra aplicação livre de risco. A aplicação livre de risco escolhida para calcular o índice, foi o retorno do CDI no ano de 2013 (8,06%)<sup>11</sup>, ano anterior ao que queremos estudar.

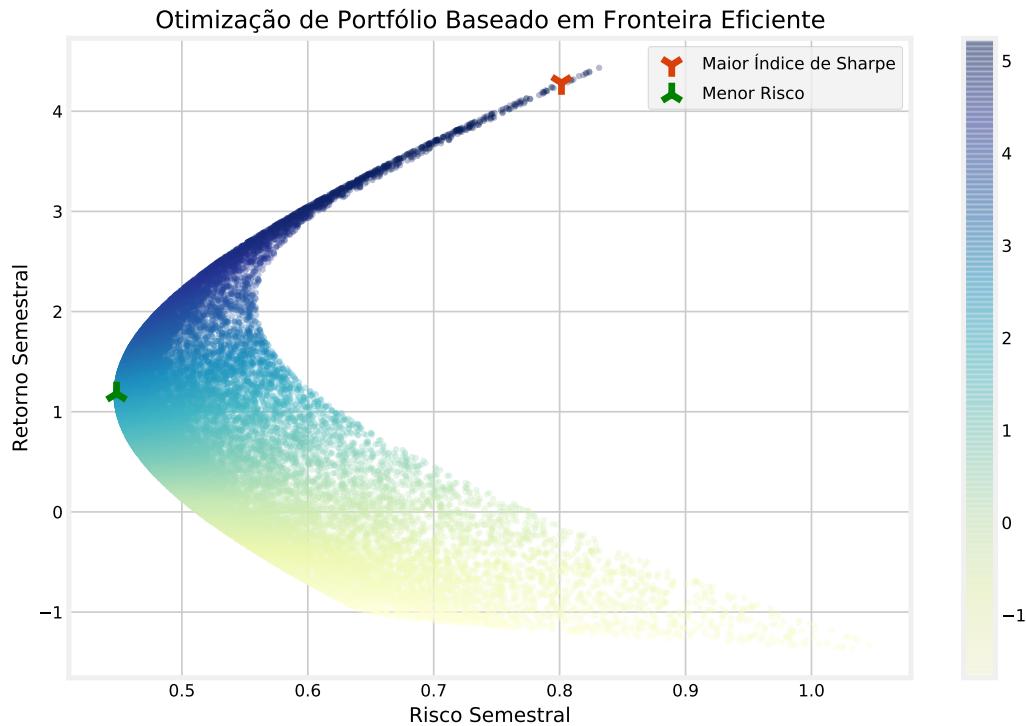


Figura 5.16: Fronteira eficiente do portfólio PETR3, VALE3, ABEV3.

Os diversos portfólios que foram gerados na Figura 5.16 são frutos da aplicação de um método que tem como objetivo gerar simulações aleatórias de forma massiva para encontrar um resultado aproximado da realidade, denominado método de Monte Carlo. A fronteira eficiente foi então encontrada com um conjunto de 25.000 simulações aleatórias. Dentro da fronteira, qualquer portfólio selecionado será considerado ótimo (de maior retorno e menor risco). No entanto, para esse trabalho consideramos dois portfólios apenas, o de menor risco, e o de maior índice de Sharpe, representados respectivamente pelos marcadores verde e vermelho. A carteira de menor risco estima um retorno semestral de 1.18%, com um risco de 45%, sendo construída de 17.86% PETR3, 42.91% VALE3 e 39.22% ABEV3. Por outro lado, a carteira de maior índice de Sharpe, possui um poten-

<sup>11</sup>Dados retirados do site: <https://www.totoradar.com.br/>

Tabela 5.7: Comparação dos resultados obtidos pelo portfólio Hare com portfólios de fronteira eficiente.

		Valores Estimados	Valores Reais
Portfólio de Maior Índice de Sharpe	Lucro	R2140,00\$	-R3.984\$
	Rentabilidade	4,28%	-7,96%
Portfólio de Menor Risco	Lucro	R590\$	-R3.984\$
	Rentabilidade	1,18%	-6%
Portfólio do Hare	Lucro	-	R\$5.874,00
	Rentabilidade	-	\$11,74%
	Rendimento Comparado à Poupança	-	250,44%
	Rendimento Comparado ao CDI	-	141,06%

cial de retorno de 4.28% com um risco de 80%, sendo construída por 4.54% PETR3, 0.1% VALE3 e 95.36 ABEV3.

Tendo a porcentagem de cada ativo para cada portfólio podemos então criar os portfólios em que um investidor alocaria seu saldo de R\$50.000. Começando pela estratégia mais segura, seguindo o portfólio de menor risco, o investidor distribuiria seu dinheiro nos ativos da forma supracitada. Relembmando, os preços iniciais dos ativos no período inicial de 2014 são 15,01 para PETR3, 32,25 para VALE3 e 17,00 para ABEV3. Nesses valores, o portfólio de menor risco, com os pesos resultantes da fronteira eficiente, consistiria de 500 cotas da PETR3, 600 cotas da VALE3 e 1100 cotas da ABEV3, totalizando R\$45.555 investidos, e sobrando R\$4.445<sup>12</sup>. Os valores finais assumidos pelos ativos, no período analisado, foram de 16,05 para PETR3, 29,06 para VALE3 e 15,54 para ABEV3. Desta forma o investidor que segui-se o portfólio de menor risco obteria um lucro de 520 na Petrobras, mas perderia 1.914 com a Vale e 1.606 com a Ambev. Obtendo um rendimento total de -3000 reais do valor investindo e totalizando sua carteira com 47.000 reais. O investidor que escolhesse pelo portfólio com maior índice de Sharpe seguiria por um caminho similar. Sua carteira seria alocada de 100 cotas da PETR3, 0 cotas da VALE3 e 2800 cotas da ABEV3, totalizando R\$49.101 investidos, e sobrando R\$899 de saldo. Este investidor ganharia 104 reais na Petrobras, 0 na Vale, e seria abatido por uma perda de -4.088 reais. O mesmo perderia R\$3.984 do dinheiro investido, totalizando 46.016 reais na carteira. Os resultados obtidos são resumidos na Tabela 5.7.

O primeiro semestre de 2014 não viria a ser um período lucrativo para um investidor que seguisse o modelos de composição de portfólio baseados na fronteira eficiente.

<sup>12</sup>Os valores selecionados para as quantidades de cotas são calculados com base na porcentagem de dinheiro que deve ser investido naquele ativo. Neste trabalho, consideramos que o agente não pode comprar ações fracionadas, apenas em lotes, portanto, a quantidade de cotas é sempre arredondada para a centena inferior.

Os portfólios de menor risco e maior índice de Sharpe que possuíam capacidades para apresentar retornos de 1,18% e 4,28%, sucumbiram a seus riscos, fornecendo uma rentabilidade real de -6% e -7,96%. Portanto, o investidor que seguisse o modelo treinado com o DDPG proposto pelo Hare, obteria no final dos 6 meses uma rentabilidade de 11,74%, um retorno claramente maior que os retornos reais e projetados dos portfólios considerados “eficientes”. Demonstrando assim, a viabilidade do modelo proposto para investimentos na bolsa de valores, e respondendo a Pergunta 5 proposta.

## 5.4 Considerações Finais

Este capítulo realizou a validação do serviço proposto em duas etapas, primeiro validando o módulo preditor, seguindo com a validação o MAR. Para tanto, criou-se uma rotina de exploração de hiper-parâmetros e treinou-se os modelos que obtiveram os melhores desempenho. Os dados obtidos demonstraram resultados satisfatórios para os modelos gerados, com acurácia de até 92.0% no ativo PETR3, 82% em VALE3, e 94% em ABEV. Tais resultados foram comparados, avaliando o desempenho em relação aos modelos tradicionais da literatura. O modelo proposto obteve a maior acurácia e a maior média das métricas obtidas. Posteriormente, realizou-se a validação do MAR, começando pelos ativos individuais até chegar no portfólio completo. Os resultados obtidos, mostraram uma rentabilidade de 13,77% em PETR3, 6,03% em VALE3, 0% em ABEV, e 11,74% para o portfólio completo. O resultado do portfólio foi comparado com um “portfólio eficiente” criado utilizando a VMM, obtendo resultados consideravelmente melhores, visto que o “portfólio eficiente” apresentaria perdas no fim do período analisado.

Os experimentos realizados no algoritmo DDPG envolveram muitas tentativas (e erros), nos quais não foram mencionados. Diversas funções de recompensas foram testadas, estas incluem: valores simples como -1 e +1 para recompensas de perda ou lucro respectivamente, valores de recompensa somente no fim da simulação, recompensas baseadas em ganho diário, dentre diversas outras. Nenhuma das recompensas apresentou um resultado tão eficiente quanto o demonstrado, muita das vezes convergindo para uma política específica na primeira época e não explorando. Acredita-se que a DDPG possa estar apresentando dois principais problemas à formulação da função recompensa: recompensas esparsas e falta de reforços positivos. Esses dois casos podem estar levando a DDPG a uma situação de *deadlock*, onde não consegue mais aprender [68]. Além das funções de recompensa, espaço de estados diversos também foram implementados, variando o espaço de estados apresentados no desenvolvimento, incluindo situações onde o espaço de estados possuía somente o indicador de movimento. Nenhum dos quais apresentaram resultados demonstráveis. Ademais, notou-se que a DDPG é extremamente sensível a

hiper-parametrização, desde a semente aleatória que gerará o modelo até o tamanho da camada escondida, diversos parâmetros foram testados, os apresentados neste trabalho foram os que obtiveram melhores resultados.

# Capítulo 6

## Conclusão

Durante o desenvolvimento deste trabalho, ficou claro que os mercados de ações representam um papel importante na economia, e oferece oportunidades para empresas e corporações crescerem e investidores gerarem rentabilidade em seus ativos financeiros. Complementarmente, a complexidade, volatilidade, e oportunidades de gerar lucros investindo no mercado financeiro, veem criando um interesse crescente na comunidade acadêmica. No entanto, prever o movimento de um dado ativo no mercado não é uma tarefa trivial, é necessário correlacionar análises técnicas e fundamentais para conseguir previsões suficientes para negociar o ativos com segurança em um mercado de ações desejado.

Diante desse cenário, este trabalho propôs o Hare como um novo serviço de investimento. O Hare divide a complexidade do mercado em sub-tarefas e oferece um serviço confiável baseado em análises técnicas e fundamentais para negociar ativos na bolsa de valores com alta precisão e estabilidade. Sua metodologia consiste na aplicação de modelos de aprendizagem para classificar quando uma ação vai ganhar ou perder valor, e posteriormente, aprender a utilizar essa informação para descobrir qual é o momento ideal de realizar uma compra ou uma venda.

Sua metodologia é validada por meio de uma série de experimentos para avaliar cada módulo racional do Hare individualmente. Desta forma o roteiro experimental consegue avaliar todo o processo de investimento realizado pelo Hare. Seu MPM é validado com uma série de processos, cuidando da otimização de hiper-parâmetros por métodos Bayesianos, treinamento dos modelos e obtenção de métricas, e comparação com modelos de aprendizado tradicionais. Posteriormente, o modelo alocador de recursos é validado em comparativo com investimentos de renda fixa tradicionais e modelos de alocações de portfólios da teoria moderna.

Como prova de conceito, o Hare foi projetado para operar na bolsa de valores [B]<sup>3</sup>, a bolsa Brasileira, previamente conhecida como BM&FBovespa. O portfólio considerado possui ativos de 3 grandes empresas brasileiras: Petrobras (PETR3), Ambev (ABEV3)

e Vale S.A. (VALE3). O roteiro experimental criado focou em validar o Hare em duas etapas: (i) validação do módulo preditor; e (ii) validação do Modelo de Alocação de Recursos (MAR). Desta forma o roteiro experimental consegue avaliar todo o processo de investimento realizado pelo Hare, os seguintes resultados obtidos destacam-se:

- Os modelos de predição de movimento obtiveram tiveram uma acurácia de 82% para ABEV3, 92% para PETR3 e 94% para VALE3, se destacando em relação à outros modelos comparativos.
- O modelo de alocação de recursos é capaz, em certo nível, de obter lucros em ativos individuais, e em um portfólio completo, no qual obteve uma rentabilidade de 11,74% no semestre avaliado. Resultados demonstram rentabilidades maiores que investimentos de renda fixa e de portfólios criados utilizando Variância Média de Markowitz.

Esses resultados corroboram a hipótese de que é possível criar um serviço capaz de investir na bolsa de valores com precisão e estabilidade em suas previsões e rentabilidade em seus investimentos.

## 6.1 Trabalhos Futuros

No decorrer do desenvolvimento desta pesquisa, principalmente durante os resultados, surgiram novas ideias que podem ser desenvolvidas para melhorar o Hare, bem como outras pesquisar para nortear futuros projetos na área de mercado financeiro, sendo eles: utilização de mais valores de entrada para o modelo LSTM, criação de um modelo inteligente para o MGR, hiper-parametrização da DDPG utilizando o Hyperopt, estudo de reformulações do ambiente da [B]<sup>3</sup>, e estudo dos motivos nos quais a DDPG não converge para políticas mais lucrativas.

Acredita-se também que um experimento possa ser realizado discretizando o espaço de ações e transformando o modelo para uma *Deep Q-Network*. É possível que a utilização de ações continuas seja desnecessária para o ambiente formulado.

Experimentos futuros também podem levar em consideração a utilização de taxas de transações como um fator que altera o lucro e a recompensa do agente. Além de se recomendar uma experimentação no semestre ou anos seguintes.

# Bibliografia

- [1] Zvi Bodie et al. *Investments*. Tata McGraw-Hill Education, 2009.
- [2] Fátima Rocha Gomes. “A Bolsa de Valores brasileira como fonte de informações financeiras”. Em: *Perspect. cienc. inf., Belo Horizonte* 2.2 (1997), pp. 189–202.
- [3] Edwin Elton, Martin Gruber e Stephen Brown. *Moderna teoria de carteiras e análise de investimentos*. Elsevier Brasil, 2012.
- [4] Isaac Kofi Nti, Adebayo Felix Adekoya e Benjamin Asubam Weyori. “A systematic review of fundamental and technical analysis of stock market predictions”. Em: *Artificial Intelligence Review* (2019), pp. 1–51.
- [5] Tobias Preis, Helen Susannah Moat e H Eugene Stanley. “Quantifying trading behavior in financial markets using Google Trends”. Em: *Scientific reports* 3 (2013), p. 1684.
- [6] Francesco Cesarone, Andrea Scozzari e Fabio Tardella. “Portfolio selection problems in practice: a comparison between linear and quadratic optimization models”. Em: *arXiv preprint arXiv:1105.3594* (2011).
- [7] Yakup Kara, Melek Acar Boyacioglu e Ömer Kaan Baykan. “Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange”. Em: *Expert systems with Applications* 38.5 (2011), pp. 5311–5319.
- [8] Rohit Choudhry e Kumkum Garg. “A hybrid machine learning system for stock market forecasting”. Em: *World Academy of Science, Engineering and Technology* 39.3 (2008), pp. 315–318.
- [9] Lean Yu, Shouyang Wang e Kin Keung Lai. “Mining stock market tendency using GA-based support vector machines”. Em: *International Workshop on Internet and Network Economics*. Springer. 2005, pp. 336–345.
- [10] David Enke e Suraphan Thawornwong. “The use of data mining and neural networks for forecasting stock market returns”. Em: *Expert Systems with applications* 29.4 (2005), pp. 927–940.

- [11] David Enke e Suraphan Thawornwong. “The use of data mining and neural networks for forecasting stock market returns”. Em: *Expert Systems with applications* 29.4 (2005), pp. 927–940.
- [12] Monica Lam. “Neural network techniques for financial performance prediction: integrating fundamental and technical analysis”. Em: *Decision support systems* 37.4 (2004), pp. 567–581.
- [13] Kofi O Nti, Adebayo Adekoya e Benjamin Weyori. “Random Forest Based Feature Selection of Macroeconomic Variables for Stock Market Prediction”. Em: *American Journal of Applied Sciences* 16.7 (2019), pp. 200–212.
- [14] Luciana S Malagrino, Norton T Roman e Ana M Monteiro. “Forecasting stock market index daily direction: a Bayesian network approach”. Em: *Expert Systems with Applications* 105 (2018), pp. 11–22.
- [15] Felipe Dias Paiva et al. “Decision-making for financial trading: A fusion approach of machine learning and portfolio selection”. Em: *Expert Systems with Applications* 115 (2019), pp. 635–655.
- [16] Spyros K Chandrinos, Georgios Sakkas e Nikos D Lagaros. “AIRMS: A risk management tool using machine learning”. Em: *Expert Systems with Applications* 105 (2018), pp. 34–48.
- [17] Hyejung Chung e Kyung-shik Shin. “Genetic algorithm-optimized long short-term memory network for stock market prediction”. Em: *Sustainability* 10.10 (2018), p. 3765.
- [18] Luisanna Cocco, Giulio Concas e Michele Marchesi. “Using an artificial financial market for studying a cryptocurrency market”. Em: *Journal of Economic Interaction and Coordination* 12.2 (2017), pp. 345–365.
- [19] Matthew Dixon, Diego Klabjan e Jin Hoon Bang. “Classification-based financial markets prediction using deep neural networks”. Em: *Algorithmic Finance* 6.3-4 (2017), pp. 67–77.
- [20] Ricardo de A Araújo et al. “A deep increasing-decreasing-linear neural network for financial time series prediction”. Em: *Neurocomputing* 347 (2019), pp. 59–81.
- [21] Burton G Malkiel e Eugene F Fama. “Efficient capital markets: A review of theory and empirical work”. Em: *The journal of Finance* 25.2 (1970), pp. 383–417.
- [22] Petter N Kolm, Reha Tütüncü e Frank J Fabozzi. “60 Years of portfolio optimization: Practical challenges and current trends”. Em: *European Journal of Operational Research* 234.2 (2014), pp. 356–371.

- [23] Harry Markowitz. “Portfolio Selection, Journal of Finance”. Em: *Markowitz HM—1952* (1952), pp. 77–91.
- [24] Stuart J Russell e Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.
- [25] Ian Goodfellow, Yoshua Bengio e Aaron Courville. *Deep learning*. MIT press, 2016.
- [26] Lucas FS Vilela et al. “Forecasting financial series using clustering methods and support vector regression”. Em: *Artificial Intelligence Review* 52.2 (2019), pp. 743–773.
- [27] Oriol Vinyals et al. “Starcraft ii: A new challenge for reinforcement learning”. Em: *arXiv preprint arXiv:1708.04782* (2017).
- [28] Wolfgang Bibel. *Automated theorem proving*. Springer Science & Business Media, 2013.
- [29] Prafulla Dhariwal et al. *Jukebox: A Generative Model for Music*. 2020. arXiv: 2005.00341 [eess.AS].
- [30] Shixiang Gu et al. “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates”. Em: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 3389–3396.
- [31] KC Santosh. “AI-driven tools for coronavirus outbreak: need of active learning and cross-population train/test models on multitudinal/multimodal data”. Em: *Journal of Medical Systems* 44.5 (2020), pp. 1–5.
- [32] David Poole, Alan Mackworth e Randy Goebel. “Computational Intelligence”. Em: (1998).
- [33] Tom M Mitchell et al. *Machine learning*. 1997.
- [34] Shai Shalev-Shwartz e Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [35] Katti Faceli et al. “Inteligência Artificial: Uma abordagem de aprendizado de máquina”. Em: (2011).
- [36] T. Cover e P. Hart. “Nearest neighbor pattern classification”. Em: *IEEE Transactions on Information Theory* 13.1 (1967), pp. 21–27.
- [37] Daniel T Larose e Chantal D Larose. *Discovering knowledge in data: an introduction to data mining*. Vol. 4. John Wiley & Sons, 2014.
- [38] Donald J Berndt e James Clifford. “Using dynamic time warping to find patterns in time series.” Em: *KDD workshop*. Vol. 10. 16. Seattle, WA, USA: 1994, pp. 359–370.

- [39] Pavel Senin. “Dynamic time warping algorithm review”. Em: *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855.1-23 (2008), p. 40.
- [40] Bernhard E Boser, Isabelle M Guyon e Vladimir N Vapnik. “A training algorithm for optimal margin classifiers”. Em: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 144–152.
- [41] Bernhard Scholkopf e Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.
- [42] Lutz H Hamel. *Knowledge discovery with support vector machines*. Vol. 3. John Wiley & Sons, 2011.
- [43] Simon Haykin. *Redes neurais: princípios e prática*. Bookman Editora, 2007.
- [44] Robert H Shumway e David S Stoffer. *Time series analysis and its applications: with R examples*. Springer, 2017.
- [45] David Rolnick et al. “Tackling climate change with machine learning”. Em: *arXiv preprint arXiv:1906.05433* (2019).
- [46] Alec Radford et al. “Language models are unsupervised multitask learners”. Em: *OpenAI Blog* 1.8 (2019), p. 9.
- [47] David E Rumelhart, Geoffrey E Hinton e Ronald J Williams. “Learning representations by back-propagating errors”. Em: *nature* 323.6088 (1986), pp. 533–536.
- [48] Sepp Hochreiter e Jürgen Schmidhuber. “Long short-term memory”. Em: *Neural computation* 9.8 (1997), pp. 1735–1780.
- [49] Kyunghyun Cho et al. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. Em: *arXiv preprint arXiv:1406.1078* (2014).
- [50] Richard S Sutton e Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [51] Joshua Achiam. “Spinning Up in Deep Reinforcement Learning”. Em: (2018).
- [52] Yuxi Li. “Deep reinforcement learning: An overview”. Em: *arXiv preprint arXiv:1701.07274* (2017).
- [53] Richard Bellman. “Dynamic programming”. Em: *Science* 153.3731 (1966), pp. 34–37.
- [54] David Silver et al. “Deterministic policy gradient algorithms”. Em: 2014.
- [55] Timothy P Lillicrap et al. “Continuous control with deep reinforcement learning”. Em: *arXiv preprint arXiv:1509.02971* (2015).

- [56] Eunsuk Chong, Chulwoo Han e Frank C Park. “Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies”. Em: *Expert Systems with Applications* 83 (2017), pp. 187–205.
- [57] Rosdyana Mangir Irawan Kusuma et al. “Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market”. Em: *arXiv preprint arXiv:1903.12258* (2019).
- [58] Zhipeng Liang et al. “Adversarial Deep Reinforcement Learning in Portfolio Management”. Em: *arXiv preprint arXiv:1808.09940* (2018).
- [59] Omer Berat Sezer e Ahmet Murat Ozbayoglu. “Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach”. Em: *Applied Soft Computing* 70 (2018), pp. 525–538.
- [60] Chien Yi Huang. “Financial Trading as a Game: A Deep Reinforcement Learning Approach”. Em: *arXiv preprint arXiv:1807.02787* (2018).
- [61] James Bergstra et al. “Hyperopt: a python library for model selection and hyperparameter optimization”. Em: *Computational Science & Discovery* 8.1 (2015), p. 014008.
- [62] Ilaria Bordino et al. “Web search queries can predict stock market volumes”. Em: *PloS one* 7.7 (2012).
- [63] Xiao Ding et al. “Using structured events to predict stock price movement: An empirical investigation”. Em: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1415–1425.
- [64] Michael Hagenau, Michael Liebmann e Dirk Neumann. “Automated news reading: Stock price prediction based on financial news using context-capturing features”. Em: *Decision Support Systems* 55.3 (2013), pp. 685–697.
- [65] M Suresh Babu, N Geethanjali e B Satyanarayana. “Clustering approach to stock market prediction”. Em: *International Journal of Advanced Networking and Applications* 3.4 (2012), p. 1281.
- [66] Volodymyr Mnih et al. “Playing atari with deep reinforcement learning”. Em: *arXiv preprint arXiv:1312.5602* (2013).
- [67] Raj Jain. *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons, 1990.
- [68] Guillaume Matheron, Nicolas Perrin e Olivier Sigaud. “The problem with DDPG: understanding failures in deterministic environments with sparse rewards”. Em: *arXiv preprint arXiv:1911.11679* (2019).

# Apêndice A

## Análise detalhada dos experimentos individuais de cada ativo

### A.1 Alocação do Ativo PETR3

Dentre os experimentos realizados para os ativos individuais, o primeiro a ser analisado foi o PETR3. O experimento foi realizado utilizando a mesma janela de valor 6 encontrada nos experimentos do módulo preditor, e com R\$10.000 de investimento inicial. As Figuras A.2 a A.6, demonstram todos os dados obtidos nas 200 épocas de treino e em seus testes por época. A Figura A.1 apresenta o comportamento do retorno médio obtido por episódios de treino e teste. Apesar das altas oscilações apresentadas, no final da simulação o retorno médio alcança o valor de 92.21 e 82.272 em treino e teste respectivamente, o que demonstra uma devida conversão do aprendizado.

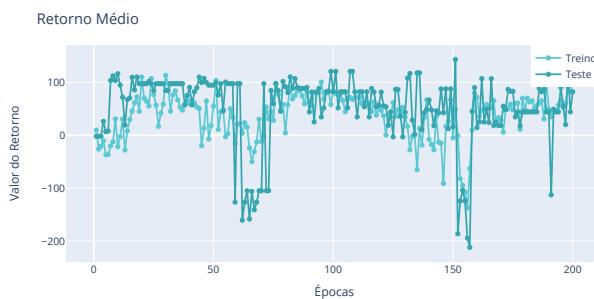


Figura A.1: PETR3 - Retorno médio.

Observa-se também, analisando a Figura A.2 e a Figura A.3 que o Q-Valor médio e as funções de perda também convergiram. O Q-Valor alcança uma média de 26.36, variando entre 203.69 a -73.92, enquanto as funções de perda assumem os valores de 62.77 e -26.43 para perda Q (função de aproximação do Q-Valor) e perda  $\pi$  (função de aproxi-

mação da política). Apesar dos valores não estarem muito próximos de 0, experimentos com mais quantidades de épocas foram conduzidos e sua variação não foi significante. Tal comportamento pode ser um indicativo de que o experimento encontrou um mínimo local, necessitando assim de uma modificação na formulação de ambiente ou na função de recompensa.

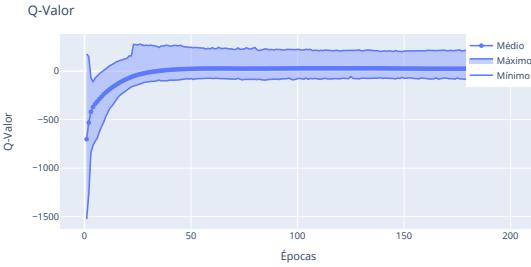


Figura A.2: PETR3 - Comportamento do QValor

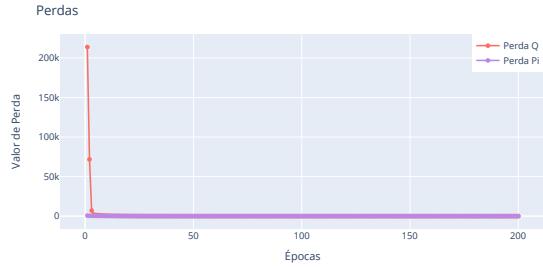


Figura A.3: PETR3 - Comportamento das funções de perda

Em relação ao lucro obtido, nota-se na Figura A.4, que durante o treino se obteve uma alta dispersão de valores, fruto do alto ruido introduzido no treinamento, incluindo encontrar um lucro de 4670 em um dos episódios. No entanto, o algoritmo não decide aproveitar o comportamento encontrado para esse lucro.

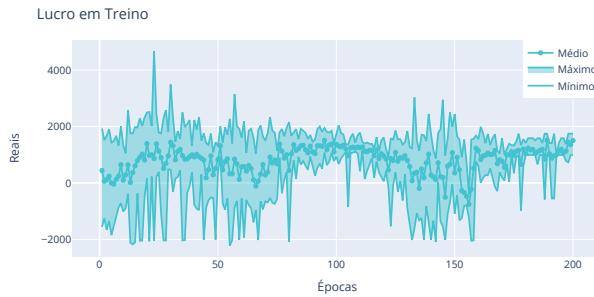


Figura A.4: PETR3 - Lucro em treino.

Quando o resultado do lucro é observado em teste (Figura A.5), percebe-se ainda as diversas oscilações apresentadas no treino, além de também não explorar a política com maior lucro encontrada. O fato do algoritmo divergir de forma significativa e não explorar necessariamente o maior lucro pode estar relacionada com a formulação da função de recompensa, que não reflete exatamente o lucro obtido. No entanto, mesmo com essas variações presentes, o modelo obteve um lucro final médio em teste de R\$1377,00, realizando majoritariamente ações de compra (veja a Figura A.6), quase não vendendo nem esperando. Vale enfatizar que nem toda ação de compra é realizada, se o agente não possui a quantidade de dinheiro para a compra ele recebe uma punição pela ação realizada

de forma incorreta, o porquê do agente preferir uma punição a uma ação de espera é algo a ser explorado.

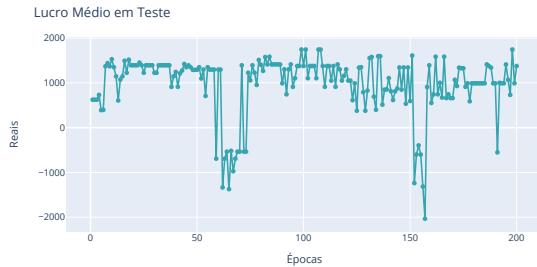


Figura A.5: PETR3 - Lucro médio em teste.

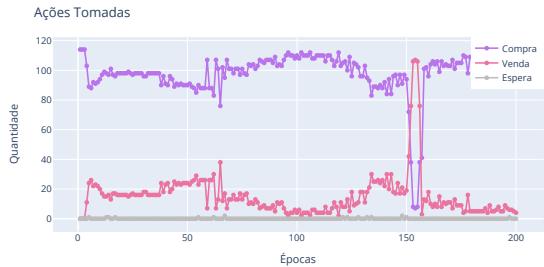


Figura A.6: PETR3 - Quantidade de ações selecionadas em teste.

Portanto, nota-se que o modelo, apesar de todas as adversidades encontradas, obteve um lucro de R\$1377,00 no final do teste, uma rentabilidade de 13,77% no total dos 6 meses. A rentabilidade encontrada é 311,04% maior que da poupança e 182,75% maior que o CDI. No caso de um investidor seguindo uma estratégia *buy and hold*, o mesmo compraria o ativo a R\$15,01, com 10.000 reais esse investidor compraria 600 cotas a R\$9.006, e sobraria 994 reais em conta. No fim do período, as ações da PETR3 estavam valendo R\$16,05, e portanto, o seu portfólio estaria valendo R\$9.6300, totalizando R\$10.624 com o dinheiro em conta, uma rentabilidade de 6,24% no final do período. A estratégia de investimento aprendida pelo agente para operar no ativo, possui uma rentabilidade 120,67% maior que o investidor seguindo um comportamento *buy and hold*.

## A.2 Alocação do Ativo VALE3

O segundo ativo individual a analisado foi o VALE3, Os experimentos realizados para este ativo possuem a mesma configuração de janela e de investimento inicial do ativo PETR3 previamente apresentado. As Figuras A.8 a A.12, demonstram todos os dados obtidos, também utilizando 200 épocas de treino e 10 testes por época. Novamente realiza-se a análise começando pelo gráfico de retorno médio por episódios de treino e teste, demonstrados na Figura A.7. Diferentemente do ativo da PETR3, o retorno apresentado nas simulações da VALE não possuem tanta oscilação, indicando uma menor exploração pelo modelo. No final da simulação, o algoritmo obtém um retorno médio de -3.327 para ambos treino e teste, mesmo tendo encontrado um retorno de 31,83 ao longo de sua trajetória.

De forma similar ao ativo anterior, analisa-se também as Figuras A.8 a A.9 para entender o comportamento do Q-Valor médio e das funções de perda. O Q-Valor obtido possui uma média de -11.31, variando entre 51.44 e -135.28. Tal valor médio negativo

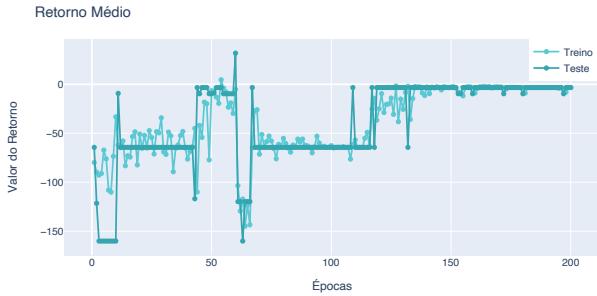


Figura A.7: VALE3 - Retorno médio.

de Q-Valor, junto com um mínimo tão discrepante da média justificam o retorno negativo médio obtido pelo modelo. Por outro lado, as funções de perda assumem os valores mais próximos de 0, com 21.13 para a aproximação do Q-Valor e 11.18 para aproximação da política, o que pode ser um indicativo de que este modelo é mais adaptável a replicações em relação ao do ativo PETR3.

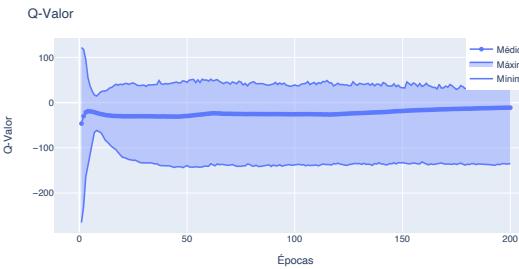


Figura A.8: VALE3 - Comportamento do QValor.

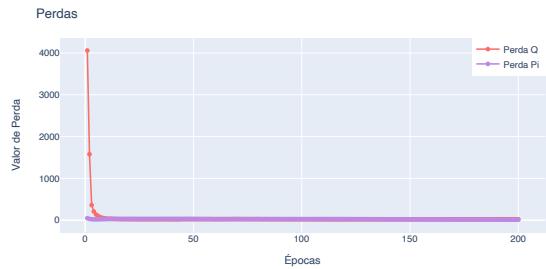


Figura A.9: VALE3 - Comportamento das funções de perda.

Com base no lucro obtido durante a fase de treino, apresentado na Figura A.10, nota-se também uma alta dispersão dos valores chegando a encontrar lucros maiores que mil reais, mas esses não foram explorados. Observa-se que a conversão para um valor de lucro constante acontece bem mais rápida, e mesmo com o ruido presente, o treino não varia a mesma quantidade de PETR3.

Quando o valor do lucro é analisado em teste (Figura A.11), percebe-se um padrão similar a média de lucro obtida no treino. O maior lucro encontrado durante a simulação foi de R\$942,00, no entanto, o valor final para qual o algoritmo converge é de apenas 603,00 reais. Este comportamento reforça a hipótese de que a função de recompensa pode ser responsável pela falta de aproveitamento do agente por lucros maiores. Em relação as ações, apresentadas na Figura A.12, o comportamento selecionado é discrepante com o realizado para o ativo da PETR3, realizando majoritariamente ações de venda, quase não vendendo e nunca esperando. De forma similar, enfatiza-se que nem toda ação de venda é realizada, se o agente não possui ativos para vender ele recebe uma punição pela

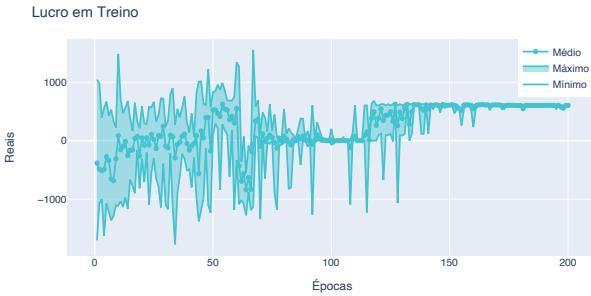


Figura A.10: VALE3 - Lucro em treino.

ação realizada de forma incorreta. Também é incerto o porquê do agente não aproveitar os comportamentos de espera.



Figura A.11: VALE3 - Lucro médio em teste.

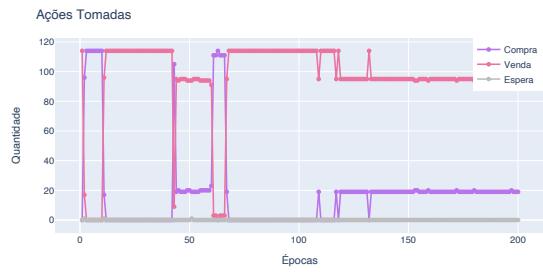


Figura A.12: VALE3 - Quantidade de ações selecionadas em teste.

Analizando o desempenho do modelo, percebe-se que as adversidades impactaram de forma significante a capacidade do agente lucrar, obtendo um lucro de apenas R\$603,00. Este lucro representa uma rentabilidade de 6,03% no total dos 6 meses. No entanto, a rentabilidade, apesar de baixa, ainda oferece lucratividade 80% e 23,81% maior que a poupança e o CDI respectivamente. Considerando a estratégia *buy and hold* de um investidor, os ativos da VALE3 seriam comprados inicialmente pelo valor de R\$32,25, o que permitira a compra de 300 cotas a R\$9.675 e sobraria R\$325 em conta. No fim do período estipulado para a simulações, as ações da VALE3 alcançariam um valor de R\$29.06. O investidor que tivesse alocado essa quantia nos ativos da Vale nesse período perderia 957 reais, terminando com 8.718 em ativos, totalizando com o saldo em carteira 9.043, uma rentabilidade de -9,57%. A estratégia de investimento aprendida pelo agente, apesar de não apresentar grandes lucros, demonstra uma aversão a perdas, sabendo parar de investir quando algum lucro foi obtido.

### A.3 Alocação do Ativo ABEV3

Finalizando, o ultimo ativo individual a ser analisado foi o ABEV3. Neste ativo também foi utilizado R\$10.000 de valor inicial, mas diferentemente da PETR3 e VALE3, a janela utilizada foi de 9 dias, similar ao resultado da LSTM. Os dados obtidos nos experimentos estão representados nas Figuras A.13 a A.18, onde cada experimento também acontece em 200 épocas de treino e 10 episódios de teste por época. A Figura A.7, apresenta o retorno médio por episódios de treino e teste. Para o ativo específico da ABEV3, o modelo não conseguiu generalizar a ponto de obter lucros. Acredita-se que este comportamento pode ser fruto da dificuldade do agente em encontrar recompensas positivas significantes no começo da simulação, entrando em uma situação de *deadlock*, onde o algoritmo da DDPG falha em aprender []. No final da simulação, o algoritmo obtém um retorno médio de  $-61.05$  para treino e  $-68.70$  para teste, mesmo tendo encontrado retornos maiores ao longo de sua trajetória.

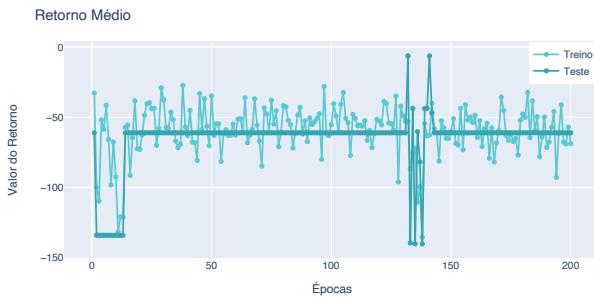


Figura A.13: ABEV3 - Retorno médio.

A busca pelo entendimento desse comportamento para o ativo da ABEV3, segue com a análise das Figuras A.14 a A.15. Essa análise busca entender o comportamento do Q-Valor médio e das funções de perda, que levaram a um desempenho ruim. O Q-Valor obtido possui uma média de  $-27.59$ , variando entre  $98.05$  e  $-154.09$ . Em relação as função de perda, aproximação do Q-Valor chega a  $13.59$ , enquanto a aproximação de  $\pi$  a ultrapassa, chegando a  $27.57$ , claramente divergindo da política buscada.

Nota-se na Figura A.16, que o lucro médio obtido durante a fase de treino, fica sempre perto de zero, mesmo apresentando dispersões de mil reais para mais ou para menos. Observa-se que a conversão para um valor de lucro constante não acontece de forma aparente, o que pode ser apenas o ruido presente nas ações, mas pode ser um indicativo de que o algoritmo entrou em seu estado de *deadlock*.

Quando o valor do lucro é analisado em teste (Figura A.17), percebe-se que a política de aversão a perdas, vista no ativo da VALE3 também foi aprendida. O agente termina a simulação com um nenhum lucro, mas também nenhuma perda, exatamente 0, mesmo

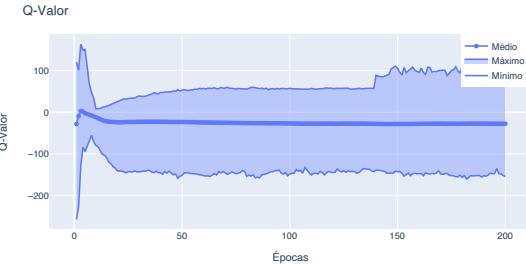


Figura A.14: ABEV3 - Comportamento do QValor.

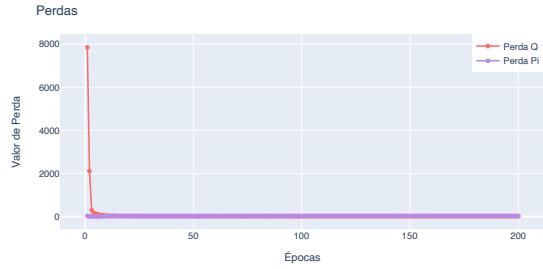


Figura A.15: ABEV3 - Comportamento das funções de perda.

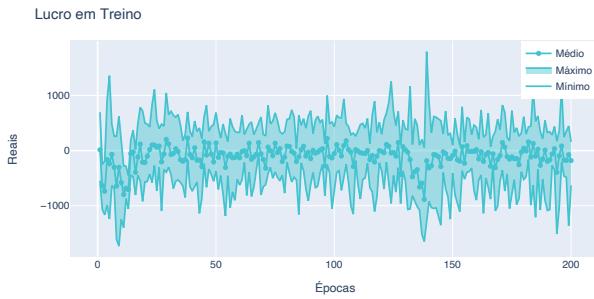


Figura A.16: ABEV3 - Lucro em treino.

eventualmente o agente encontrando alguns lucros baixos de R\$140,00. Novamente, esse comportamento impede o aproveitamento do agente por lucros maiores, o que pode ser causado pela função de recompensa. Em relação as ações, apresentadas na Figura A.18, o comportamento selecionado apresenta claramente o lucro de 0, visto que o agente decide só vender desde o inicio da simulação, se avergindo a investir. Observe também que quando o agente tentou explorar as outras ações o agente obteve perdas significantemente maiores que ganhos, o que pode ter causado uma espécie de desistência pelo lado do agente.

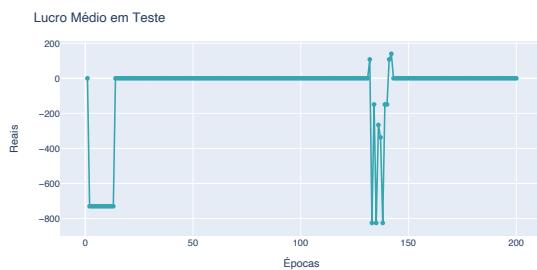


Figura A.17: ABEV3 - Lucro médio em teste.

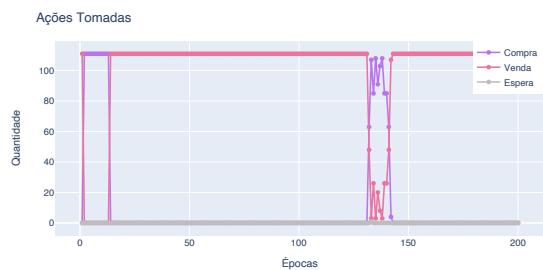


Figura A.18: ABEV3 - Quantidade de ações selecionadas em teste.

O desempenho do modelo para o ativo ABEV3 não é o esperado, no entanto, ainda assim o modelo evita a possibilidade de perdas. Este ativo, no inicio do período de investimento, estava sendo vendido a 17 reais, mas no fim do período, o ativo chegaria

a 15.54, desvalorizando em 9,3%. Um investidor seguindo a estratégia *buy and hold* possuiria grandes perdas nesse ativo. O que confirma mais uma vez, que a estratégia aprendida pelo agente, apesar de não ser ótima, consegue evitar prejuízos ao investidor.

## Apêndice B

# Experimento extra com o Módulo de Gerenciamento de Riscos

Após todos os experimentos analisados com os estados preditivos dos ativos, realizou-se então um novo experimento com informações de riscos fornecidas pelo módulo de gerenciamento de risco sintético implementado. O experimentos foram realizados com parâmetros similares ao experimento de portfólio previamente realizado. Sua única diferença foi em relação ao estado do agente, agora o agente além de ter acesso a previsão do ativo, tem acesso também ao indicador de riscos do MGR.

Para cada ativo do portfólio, utiliza-se o volume de consultas do mesmo no *Google Trends*. Esse volume representa a quantidade de pesquisas semanais entre 0 a 100 do termo pesquisado. No entanto, o MGR processa essa informação, indicando um risco com o valor 1 caso o valor esteja acima de um limite pré-determinado, ou 0 caso o contrário. O experimento realizado possui um valor limite de 90, representando que apenas pesquisas demasiadas indicariam um risco.

As Figuras B.1 a B.2, apresentam os resultados para o experimento. Como podemos ver, o experimento apresenta resultados semelhantes ao anterior (sem o MGR). No entanto apresenta menos variações em seus resultados de teste em relação os valores encontrados sem o risco. O lucro encontrado é idêntico ao do experimento anterior, R\$5.874,00 no final do teste, uma rentabilidade de 11,74% no total dos 6 meses. Sendo a única diferença aparente a impressão de uma conversão mais rápida ao resultado encontrado. Os outros gráficos obtidos durante o experimento foram omitidos por possuírem resultados muito similares ao experimento anterior.

A semelhança dos resultados encontrados pode ser um indicativo de que o módulo de risco sintético não apresenta consistência em sua informação, e seguindo essa lógica de raciocínio, invalidaria a premissa de que um alto numero de pesquisas representa

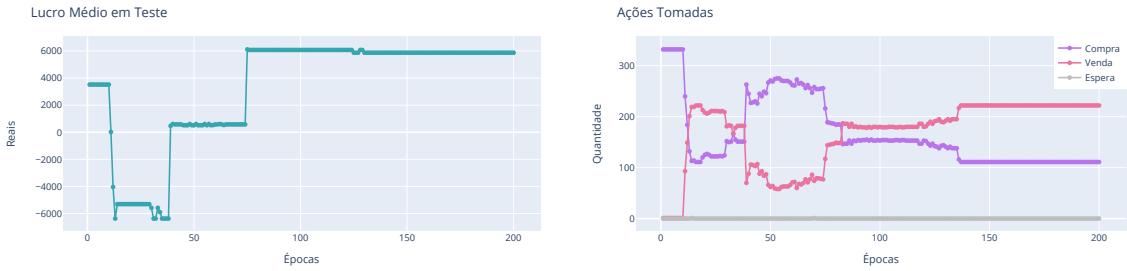


Figura B.1: Portfólio com MGR - Lucro médio em teste.

Figura B.2: Portfólio com MGR - Quantidade de ações selecionadas.

necessariamente um risco. A implementação de modelos robustos para o MGR e a análise mais profunda de seu comportamento será investigada em trabalhos futuros.

Com esses experimentos finaliza-se a análise do Módulo Alocador de Recursos, demonstrando sua eficiência e capacidade de gerar lucros investindo de forma autônoma.