

GABRIEL VINÍCIUS MOSSATO FERREIRA  
PAULINO FONSECA VELOSO JÚNIOR

Previsão do Índice S&P 500 Utilizando LSTM e  
Mecanismos de Atenção

São Paulo  
2023

**GABRIEL VINÍCIUS MOSSATO FERREIRA  
PAULINO FONSECA VELOSO JÚNIOR**

**Previsão do Índice S&P 500 Utilizando LSTM e  
Mecanismos de Atenção**

Trabalho apresentado à Escola Politécnica  
da Universidade de São Paulo para obten-  
ção do Título de Engenheiro Mecatrônico.

São Paulo  
2023

**GABRIEL VINÍCIUS MOSSATO FERREIRA  
PAULINO FONSECA VELOSO JÚNIOR**

# **Previsão do Índice S&P 500 Utilizando LSTM e Mecanismos de Atenção**

Versão 1

Trabalho apresentado à Escola Politécnica  
da Universidade de São Paulo para obten-  
ção do Título de Engenheiro Mecatrônico.

Área de Concentração:  
Engenharia Mecatrônica

Orientador:  
Oswaldo Luiz do Valle Costa

São Paulo  
2023

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

#### Catálogo-na-publicação

FERREIRA, GABRIEL

Previsão do Índice S&P 500 Utilizando LSTM e Mecanismos de Atenção /  
G. FERREIRA, P. JÚNIOR -- São Paulo, 2023.  
125 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São  
Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.REDES NEURAIS 2.PREVISÃO (ANÁLISE DE SÉRIES TEMPORAIS)  
3.MERCADO DE CAPITAIS I.Universidade de São Paulo. Escola Politécnica.  
Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t.  
III.JÚNIOR, PAULINO

# AGRADECIMENTOS

Nossos sinceros agradecimentos são dirigidos a todos os professores da Escola Politécnica da Universidade de São Paulo (EP-USP), que desempenharam papéis fundamentais em nossa jornada acadêmica no curso de Engenharia Mecatrônica. Os conhecimentos e valores compartilhados por cada um deles refletem-se, de diversas maneiras, nas páginas deste trabalho.

Expressamos nossa profunda gratidão ao Professor Oswaldo Luiz do Valle Costa, do Departamento de Engenharia de Telecomunicações e Controle da EP-USP, por seu apoio, paciência e orientação, essenciais no desenvolvimento desse estudo.

Nossos agradecimentos estendem-se também aos professores que integram a comissão de coordenação dos trabalhos de formatura do Departamento de Engenharia Mecatrônica e Sistemas Mecânicos da EP-USP. Especialmente, nós agradecemos à Professora Larissa Driemeier e ao Professor Thiago de Castro Martins cujas avaliações e sugestões contribuíram significativamente para o enriquecimento e aprimoramento desse trabalho.

*“History never repeats itself, but it does  
often rhyme.”*

Mark Twain

# RESUMO

Esse estudo desenvolveu um modelo de previsão para o preço de fechamento do índice S&P 500 empregando células LSTM integradas a mecanismos de atenção. O modelo foi otimizado através de técnicas de busca em *grid* e de busca bayesiana. Os resultados demonstraram um desempenho notavelmente promissor na previsão dos preços, embora não tenham superado completamente o *benchmark* estabelecido. Além disso, observou-se um impacto significativo do mecanismo de atenção nas previsões. Por fim, o modelo foi combinado com técnicas de gestão de banca, visando explorar suas aplicações práticas e estudar a potencial rentabilidade das previsões em um ambiente controlado, aspecto que também revelou-se promissor.

**Palavras-Chave** — REDES NEURAIIS, PREVISÃO (ANÁLISE DE SÉRIES TEMPORAIS), MERCADO DE CAPITAIS.

# ABSTRACT

This study developed a forecasting model for the closing price of the S&P 500 index employing LSTM cells integrated with attention mechanisms. The model was optimized using grid search and Bayesian search techniques. The results demonstrated a remarkably promising performance in price prediction, although they did not completely surpass the established benchmark. Furthermore, a significant impact of the attention mechanism on the predictions was observed. Finally, the model was combined with bankroll management techniques, aiming to explore its practical applications and study the potential profitability of the predictions in a controlled environment, an aspect that also proved to be promising.

**Keywords** — Neural Networks, Prediction (Time Series Analysis), Capital Markets.



# LISTA DE FIGURAS

1	Representação gráfica de alguns exemplos de padrões temporais com segmentações de três e quatro pontos. Esses padrões ilustram variações sequenciais em séries temporais, demonstrando relações entre valores consecutivos. . . . .	29
2	Esquematização da transformada wavelet com os coeficientes de aproximação ( $A$ ) e detalhamento ( $D$ ). No exemplo, aplica-se um limiar a todos os detalhamentos afim de suprimir o ruído. É esperado que a frequência decresça na cadeia até $A_3$ , sendo $D_1$ a componente de maior frequência. . .	32
3	Diagrama esquemático da arquitetura de uma rede neural artificial multicamadas. A estrutura apresenta camadas de entrada, ocultas e de saída, com $n$ neurônios em cada camada. As equações associadas ilustram as relações de transformação entre os neurônios. . . . .	34
4	Representação gráfica de funções de ativação frequentemente utilizadas em redes neurais, incluindo: Sigmóide, Tangente Hiperbólica, ReLU, Leaky ReLU, Softmax e Softplus. Cada gráfico ilustra a relação entre a entrada ( $x$ ) e a saída ( $y$ ) para a respectiva função. . . . .	36
5	Representação esquemática da arquitetura de uma rede neural recorrente e sua respectiva versão desdobrada ao longo do tempo. A diagramação destaca os componentes principais, incluindo os estados ocultos $h$ , entradas $x$ e saídas $y$ , bem como os pesos associados ( $w_{hy}$ e $w_{xh}$ ). . . . .	40
6	Diagrama da arquitetura de uma célula LSTM destacando os principais componentes e subunidades, incluindo os portões de esquecimento, entrada e saída, bem como os estados da célula e suas respectivas funções de ativação. . . . .	43
7	Representação esquemática do mecanismo de atenção do tipo “soft attention”. O diagrama demonstra como as entradas ( $h_1, h_2, \dots, h_T$ ) são ponderadas pelos pesos de atenção ( $\alpha_1, \alpha_2, \dots, \alpha_T$ ), calculados com base nos <i>scores</i> ( $s_t$ ) e normalizados pela função <i>softmax</i> , a fim de determinar a saída $y$ . . . . .	48

8	Representação esquemática do mecanismo de atenção do tipo <i>soft attention</i> , destacando as componentes <i>query</i> ( $Q$ ), <i>key</i> ( $K$ ) e <i>value</i> ( $V$ ). As entradas ( $h_1, h_2, \dots, h_T$ ) são transformadas nas triplas $Q, K$ e $V$ . Os scores ( $s_t$ ) resultantes da interação entre <i>query</i> e <i>key</i> são normalizados por uma função <i>softmax</i> para calcular os pesos de atenção ( $\alpha_1, \alpha_2, \dots, \alpha_T$ ), que influenciam a saída final $y$ por meio dos vetores de <i>value</i> . . . . .	50
9	Ilustração do processo de janela deslizante ( <i>sliding window</i> ) para a geração de entradas e rótulos no treinamento de modelos de previsão temporal. As entradas são constituídas por subsequências dos dados $X_1, X_2, \dots, X_n$ , e os rótulos correspondem ao dado subsequente na série temporal, que é o valor a ser previsto pelo modelo. A janela deslizante avança sequencialmente, um passo por vez, enfatizando a natureza contínua e sobreposta das amostras utilizadas para a previsão. . . . .	58
10	Representação esquemática do método de validação cruzada do tipo <i>k-fold</i> específico para séries temporais, com $k = 3$ . Os conjuntos de treino e teste são segmentados sequencialmente em cada iteração. Esta abordagem otimiza o uso dos dados e assegura que a sequencialidade seja mantida. . .	59
11	Curva de aprendizado do modelo ao longo de 2000 épocas. A curva azul representa o custo de treinamento, enquanto a curva vermelha indica o custo de validação. Ambas as curvas mostram uma rápida diminuição inicial, refletindo um rápido aprendizado com uma aparente estabilização já na época 500, aproximadamente. . . . .	67
12	Detalhe ampliado da curva de aprendizado também para as 2000 <i>epochs</i> . Nota-se que a curva de custo de treinamento (em azul) de fato se estabiliza rapidamente, sem decréscimos acentuados a partir da época 500, aproximadamente. Todavia o custo de validação (em vermelho) revela um claro salto de performance ao redor da época 1000. . . . .	68
13	Representação esquemática da arquitetura do modelo de aprendizado profundo implementado, consistindo em células LSTM empilhadas seguidas de um mecanismo de <i>soft attention</i> , convergindo para uma saída única ( $y$ ). A entrada é um tensor tridimensional ( $128 \times 6 \times 20$ ), representando o tamanho do lote (omitido da representação para clareza), o número de <i>features</i> e a janela de tempo, respectivamente. Uma taxa de <i>dropout</i> de 12,241% é incluída para mitigar a ocorrência de <i>overfitting</i> . . . . .	70

- 14 Efeito da aplicação de *denoising* utilizando transformada wavelet na série temporal de fechamento do índice S&P 500. A suavização resultante (em vermelho) é evidente, principalmente na atenuação de flutuações de curto prazo, mantendo as tendências gerais de mercado. A transformada wavelet permite uma análise mais clara das tendências subjacentes ao reduzir a volatilidade aparente sem descartar a volatilidade inerente ao mercado. . . . 79
- 15 Desempenho do modelo nos dados de treino para o fechamento do índice S&P 500. A linha azul representa os preços reais de fechamento, enquanto a linha vermelha representa os preços previstos pelo modelo. A sobreposição significativa das duas séries indica uma forte aderência do modelo aos dados de treino, praticamente descartando a hipótese de *underfitting*. . . . . 80
- 16 Resultados do modelo no conjunto de validação, evidenciando a capacidade de generalização com dados cujas respostas não são conhecidas. As flutuações e tendências do mercado são bem capturadas e exibem os efeitos da otimização de hiperparâmetros. . . . . 81
- 17 Avaliação do modelo nos dados de teste, que representam o cenário mais próximo de uma aplicação do mundo real. A correspondência entre as séries temporais destaca a habilidade do modelo em prever a direção do mercado, mesmo sem ter tido exposição prévia a esses dados. O conjunto em questão é o temporalmente mais recente temporalmente e foi mantido isolado de qualquer fase anterior do processo de modelagem. . . . . 82
- 18 Sobreposição entre as ocorrências reais e as previstas de movimentos de ascensão (subida) e declínio (queda) no índice. A barra azul representa a frequência real, enquanto a barra em vermelho mostra as previsões do modelo. A proximidade entre as alturas das barras para cada tendência indica a acurácia do modelo em replicar a distribuição dos resultados reais. 86
- 19 Adaptação de uma matriz de confusão para a tendência binária do índice S&P 500 frente às previsões do modelo. A matriz ilustra em verde a capacidade do modelo de prever corretamente as tendências de alta (subida) e baixa (queda) do índice. Em contrapartida, destacam-se em tons de vermelho as previsões falsas. Os valores representam o percentual de acerto ou erro e entre parênteses o número absoluto de ocorrências sobre o total de dias. . . . . 87

- 20 Retorno acumulado no período para a predição e para o índice. A linha azul representa o retorno acumulado real do S&P 500, enquanto a linha vermelha indica o retorno acumulado conforme predito pelo modelo. A estreita aderência entre as duas linhas sugere uma notável capacidade do modelo em capturar a trajetória de retorno do índice ao longo do período. 88
- 21 Diferença (distância) entre o retorno acumulado do índice e da predição — *spread*. Valores positivos do *spread* indicam que o retorno do índice superou o retorno predito pelo modelo. A linha vermelha denota a média ao longo do período analisado. Em virtude de uma volatilidade de mercado mais alta, percebe-se um descolamento mais acentuado no início do período, alimentado ainda por tentativas de correções superestimadas pelo próprio modelo. . . . . 89
- 22 Comparativo do desempenho dos modelos frente abordagens distintas no uso ou não de atenção. Na coluna da esquerda encontram-se os resultados no conjunto de dados de treino, estando na direita os de teste. Cada linha explora uma alternativa, de cima para baixo: sem mecanismo de atenção; utilizando a atenção com *tanh* conforme (QIU et al., 2020); valendo-se da abordagem clássica por *query*, *key* e *value* segundo (VASWANI et al., 2017). 90
- 23 Desempenho financeiro da estratégia Martingale. O gráfico superior exhibe a evolução do caixa (em R\$), evidenciando um acréscimo gradual ao capital inicial. O gráfico central mostra as apostas individuais (em laranja) em comparação com o caixa disponível (em azul), destacando a variação do montante apostado. O gráfico inferior apresenta os ganhos (em verde) e perdas (em vermelho) por rodada, delineando a frequência e a magnitude dos resultados de cada aposta realizada. . . . . 93
- 24 Desempenho financeiro da estratégia Paroli. O gráfico no topo mostra a progressão do caixa ao longo das rodadas, demonstrando um aumento consistente. O gráfico central detalha as apostas individuais em contraste com o caixa disponível, é notável que, em inúmeras vezes, o investidor aloca todo o seu caixa disponível. O gráfico de barras inferior ilustra os ganhos (verde) e as perdas (vermelho) por rodada, destacando a capacidade do modelo de aproveitar sequências de vitórias para construir lucro incrementalmente. . . 94

- 25 Desempenho financeiro da estratégia D'Alembert. O gráfico superior mostra a trajetória do caixa ao longo das rodadas, exibindo um aumento progressivo, porém modesto. O gráfico do centro ilustra as apostas (em laranja) e o caixa (em azul), destacando a gestão conservadora do capital. O gráfico inferior registra os ganhos (verde) e as perdas (vermelho) a cada rodada, mostrando que, apesar da abordagem cautelosa, houve uma quantidade significativa de perdas. O lucro final, comparável ao da estratégia Martingale, é modesto em relação ao período e em conformidade com o volume das apostas. . . . . 96
- 26 Desempenho financeiro da estratégia de Apostas Fixas. O gráfico superior mostra uma progressão lenta e consistente do caixa, sem grandes oscilações. O gráfico central evidencia a consistência da estratégia, com o valor de cada aposta (em laranja) mantido igual independentemente do resultado das rodadas anteriores, refletindo a natureza pouco sofisticada da abordagem. O gráfico inferior destaca os ganhos e perdas por rodada (ganhos em verde, perdas em vermelho). . . . . 97
- 27 Desempenho financeiro da estratégia de Apostas Proporcionais a 10%. O gráfico superior evidencia um acréscimo marginal no caixa, refletindo uma abordagem conservadora que resultou em ganhos limitados. O gráfico central apresenta as apostas proporcionais (em laranja) ao caixa (em azul), com um valor baixo devido ao percentual reduzido aplicado e relativamente constante, devido a resultados modestos. O gráfico inferior mostra os ganhos (em verde) e as perdas (em vermelho) de cada rodada, destacando a similaridade com os resultados obtidos pela estratégia de Apostas Fixas devido ao baixo risco e retorno. . . . . 99
- 28 Desempenho financeiro da estratégia de Apostas Proporcionais a 25%. O aumento do caixa é mais perceptível quando comparado ao modelo de 10%, como demonstrado no gráfico superior. O aumento da proporção de apostas (em laranja) no gráfico do centro é mais notável conforme o crescimento do caixa, indicando um gerenciamento de risco mais intermediário. Os ganhos e perdas por rodada são ilustrados no gráfico inferior, resultando em uma rentabilidade boa para o período, sobretudo frente ao risco. . . . . 100

- 29 Desempenho financeiro da estratégia de Apostas Proporcionais a 50%. O gráfico no topo mostra um crescimento notável do caixa, indicando uma abordagem mais apta ao risco. O gráfico central destaca o aumento significativo nas apostas (em laranja) à medida que o caixa cresce, demonstrando um comprometimento substancial de recursos. O gráfico inferior registra ganhos e perdas por rodada, com os ganhos (verde) sobressaindo mais frequentemente do que as perdas (vermelho), ilustrando um bom desempenho geral da estratégia. . . . . 101
- 30 Desempenho financeiro da estratégia de Apostas Proporcionais a 75%. Essa abordagem agressiva resultou em um aumento substancial do caixa, conforme mostrado no gráfico superior. As apostas (em laranja) no gráfico central refletem uma maior alocação de capital, em linha com o aumento do caixa, tipicamente delatando um comportamento ávido ao risco. O gráfico inferior revela uma frequência aumentada de ganhos (verde), com os valores superando as perdas (vermelho), evidenciando uma estratégia mais rentável, porém mais volátil. . . . . 102
- 31 Desempenho financeiro da estratégia de Apostas Proporcionais a 90%. O gráfico superior exibe o maior crescimento do caixa entre todas as estratégias testadas, sugerindo uma alta rentabilidade. No gráfico do meio, as apostas (em laranja) acompanham de perto o valor do caixa, evidenciando uma estratégia altamente agressiva. O gráfico de ganhos e perdas mostra uma variação maior nos resultados de cada rodada, com ganhos consideravelmente altos (verde) que compensam as perdas (vermelho) ao todo, refletindo o elevadíssimo risco e a alta rentabilidade dessa abordagem. . . . 103
- 32 Interface do *site* de previsão do índice S&P 500. O portal encontra-se operacional e com a infraestrutura planejada, incluindo a página e o banco de dados hospedados *online*, enquanto o modelo é executado localmente. Os dados da imagem são as previsões reais do modelo frente ao índice. A interface mostra a previsão de fechamento ao lado do fechamento real, fornecendo indicações de compra ou venda baseadas na tendência predita. . 106

# LISTA DE TABELAS

1	Visão geral dos dados históricos utilizados no modelo, abrangendo um intervalo de 10.252 dias úteis entre 1983 e 2023. Cada entrada inclui informações de abertura ( <i>Open</i> ), máxima ( <i>High</i> ), mínima ( <i>Low</i> ), fechamento ( <i>Close</i> ), fechamento ajustado ( <i>Adj Close</i> ) e volume ( <i>Volume</i> ) de negociações para o dia especificado. Note que a série de fechamento servirá tanto como entrada quanto como saída do modelo. . . . .	55
2	Resultados ordenados dos testes de <i>grid search</i> para a otimização inicial dos hiperparâmetros. . . . .	61
3	Dois primeiros quartis de resultados ordenados dos testes de <i>bayesian search</i> para a otimização final dos hiperparâmetros. . . . .	65
4	Dois últimos quartis de resultados ordenados dos testes de <i>bayesian search</i> para a otimização final dos hiperparâmetros. . . . .	66
5	Comparação dos conjuntos de teste para ambos os modelos, destacando diferenças que não devem ser negligenciadas em uma comparação de desempenho. . . . .	83
6	Métricas para ambos os modelos, com comparação após normalização para minimização dos efeitos das discrepâncias entre os conjuntos de teste dos modelos. . . . .	84
7	Comparativo de desempenho financeiro entre as diferentes estratégias de gestão de banca aplicadas ao modelo de previsão. As colunas concentram os critérios de comparação para cada abordagem. As cores comparam um mesmo critério (linha) entre todas as estratégias, tonalidades verdes indicam resultados melhores frente às tonalidades vermelhas, sendo o branco indicativo de valores intermediários. Além dos dados base das simulações, apresentam-se ainda valores anualizados para retorno e volatilidade, além da razão retorno por risco. . . . .	104

# LISTA DE SIGLAS

EMH – *Efficients Markets Hypothesis* (Hipótese dos Mercados Eficientes)

ANN – *Artificial Neural Network* (Rede Neural Artificial)

RNN – *Recurrent Neural Network* (Rede Neural Recorrente)

LSTM – *Long Short-Term Memory* (Memória de Curto e Longo Prazo)

CNN – *Convolutional Neural Network* (Rede Neural Convolucional)

MSE – *Mean Squared Error* (Erro Quadrático Médio)

RMSE – *Root Mean Square Error* (Raiz do Erro Quadrático Médio)

MAE – *Mean Absolute Error* (Erro Médio Absoluto)

TE – *Tracking Error* (Erro de Rastreamento)

ETF – *Exchange-Traded Fund* (Fundo Negociado em Bolsa)

BS – *Batch Size* (Tamanho do Lote)

HSS – *Hidden State Size* (Tamanho do Estado Oculto)

LR – *Learning Rate* (Taxa de Aprendizado)

DR – *Dropout Rate* (Taxa de Desativação)



# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>19</b>
1.1	Contexto . . . . .	19
1.1.1	<i>Financial Deepening</i> Brasileiro . . . . .	19
1.1.2	S&P 500 e a Economia Global . . . . .	20
1.2	Motivação . . . . .	20
1.3	Objetivo . . . . .	21
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>23</b>
<b>3</b>	<b>Padrões, Redes Neurais e Atenção</b>	<b>27</b>
3.1	Padrões e Modelagem de Séries Temporais . . . . .	27
3.2	Redução de Ruído . . . . .	30
3.2.1	Transformada Wavelet . . . . .	30
3.3	Redes Neurais Artificiais . . . . .	33
3.3.1	<i>Feedforward</i> . . . . .	33
3.3.2	Funções de Ativação . . . . .	35
3.3.3	<i>Backpropagation</i> . . . . .	37
3.3.4	Funções de Custo . . . . .	39
3.4	Redes Neurais Recorrentes . . . . .	40
3.4.1	Formulação . . . . .	40
3.4.2	Limitações e Alternativas . . . . .	41
3.5	<i>Long-Short Term Memory</i> . . . . .	42
3.5.1	Estrutura . . . . .	42
3.5.2	Portão de Esquecimento . . . . .	44

3.5.3	Portão de Entrada . . . . .	44
3.5.4	Estados da Célula . . . . .	44
3.5.5	Portão de Saída . . . . .	45
3.5.6	Estado Oculto . . . . .	45
3.5.7	Considerações Finais . . . . .	45
3.6	Mecanismos de Atenção . . . . .	46
3.6.1	Analogia Biológica . . . . .	46
3.6.2	Aplicações . . . . .	46
3.6.3	Atenção Suave e Rígida . . . . .	47
3.6.4	Atenção Suave com <i>Score</i> por Tangente Hiperbólica . . . . .	48
3.6.5	Atenção Suave com <i>Query</i> , <i>Key</i> e <i>Value</i> . . . . .	49
3.6.6	Considerações Finais . . . . .	50
<b>4</b>	<b>S&amp;P 500 e os Mercados de Capitais</b>	<b>52</b>
4.1	Composição e Relevância Global . . . . .	52
<b>5</b>	<b>Metodologia</b>	<b>54</b>
5.1	Coleta dos Dados . . . . .	54
5.2	Pré-Processamento dos Dados . . . . .	56
5.2.1	Redução de Ruído . . . . .	56
5.2.2	Normalização . . . . .	56
5.2.3	Repartição . . . . .	57
5.2.4	Alimentação . . . . .	57
5.3	Arquitetura do Modelo . . . . .	58
5.3.1	Fundamento . . . . .	58
5.3.2	Otimização por <i>Grid Search</i> . . . . .	59
5.3.3	Otimização por <i>Bayesian Search</i> . . . . .	62
5.3.4	Análise da Curva de Aprendizado . . . . .	67

5.3.5	Concepção Final . . . . .	69
5.4	Avaliação do Modelo . . . . .	71
5.4.1	Métricas . . . . .	71
5.4.1.1	MSE . . . . .	71
5.4.1.2	RMSE . . . . .	72
5.4.1.3	MAE . . . . .	72
5.4.1.4	$R^2$ . . . . .	72
5.4.1.5	TE . . . . .	72
5.4.1.6	Considerações Finais . . . . .	73
5.4.2	Retorno Acumulado e Tendências . . . . .	74
5.4.3	Influência da Atenção . . . . .	74
5.4.4	Estratégias de Gestão de Banca . . . . .	75
5.4.4.1	Martingale . . . . .	75
5.4.4.2	Paroli . . . . .	75
5.4.4.3	D'Alembert . . . . .	76
5.4.4.4	Apostas Fixas . . . . .	76
5.4.4.5	Apostas Proporcionais . . . . .	76
5.4.4.6	Considerações Finais . . . . .	77
<b>6</b>	<b>Resultados</b>	<b>78</b>
6.1	Performance Técnica do Modelo . . . . .	78
6.1.1	Efeito da Redução de Ruído . . . . .	78
6.1.2	Desempenho da Predição Frente ao Índice . . . . .	79
6.1.2.1	Comparação das Séries Temporais . . . . .	80
6.1.2.2	Comparação de Métricas Frente ao <i>Benchmark</i> . . . . .	83
6.1.2.3	Comparação Binária da Tendência: Subida e Queda . . . . .	84
6.1.2.4	Comparação do Retorno Total Acumulado . . . . .	87

6.2	Influência da Atenção . . . . .	89
6.3	Performance Financeira do Modelo . . . . .	91
6.3.1	Martingale . . . . .	93
6.3.2	Paroli . . . . .	94
6.3.3	D'Alembert . . . . .	96
6.3.4	Apostas Fixas . . . . .	97
6.3.5	Apostas Proporcionais . . . . .	98
6.3.6	Considerações Finais . . . . .	104
6.4	Compartilhamento dos Resultados . . . . .	105
<b>7</b>	<b>Conclusões</b>	<b>107</b>
	<b>Anexo A – Código</b>	<b>109</b>
A.1	run.py . . . . .	109
A.2	src/model/model.py . . . . .	112
A.3	src/model/params.py . . . . .	117
	<b>Referências</b>	<b>119</b>

# 1 INTRODUÇÃO

## 1.1 Contexto

### 1.1.1 *Financial Deepening* Brasileiro

O Brasil tem experimentado um aprofundamento financeiro (*financial deepening*) significativo nas últimas décadas, impulsionado por uma série de reformas e inovações nesse setor. A liberalização financeira, a desregulamentação e a introdução de novos produtos financeiros desempenharam um papel importante nesse processo (CHORTAREAS et al., 2011).

A expansão do crédito tem sido um componente chave do aprofundamento financeiro no Brasil. O aumento da disponibilidade de crédito para consumidores e empresas tem contribuído para o crescimento econômico, permitindo maior consumo e investimento. (LAVINAS, 2017) destaca como o desenvolvimentismo social reformulou a política social no Brasil na década de 2000, estimulando o acesso ao crédito e aos mercados financeiros e promovendo uma transição para uma sociedade de consumo em massa.

Além disso, a inclusão financeira, por meio da expansão dos serviços bancários para populações anteriormente desbancarizadas, tem sido um aspecto importante do aprofundamento financeiro no Brasil (CATÃO et al., 2009).

Desse modo, percebe-se um grande aumento de participantes do mercado financeiro nacional nos últimos anos, seja por meio de novas e pequenas gestoras de investimentos ou por pessoas físicas que têm cada vez mais acesso a serviços financeiros e diferentes opções de investimento.

Entretanto, vê-se também um crescimento acentuado da desinformação no que tange aos investimentos, conforme abordado no relatório (CVM, 2020). De tal forma que o aumento massivo de pequenos gestores e pessoas físicas no cenário do mercado de capitais brasileiro não se traduz necessariamente em um aumento da renda nacional.

### 1.1.2 S&P 500 e a Economia Global

O S&P 500 é um dos índices mais conhecidos e amplamente seguidos no mundo dos investimentos, composto por 500 das maiores empresas listadas nas bolsas de valores dos Estados Unidos. No entanto, o que muitas pessoas podem não perceber é que uma parte significativa dos lucros dessas empresas vem de fora desse país, como destaca (DE, 2013).

De acordo com o referido artigo, muitas das empresas listadas no S&P 500 são multinacionais com operações significativas no exterior. Isso implica que, embora estejam sediadas nos Estados Unidos, uma grande parte de seus lucros é gerada por suas operações internacionais, incluindo tanto vendas de produtos e serviços quanto investimentos em empresas e infraestruturas estrangeiras.

Essa exposição internacional faz do S&P 500 um indicador útil não apenas da economia dos Estados Unidos, mas também da economia global. As tendências e mudanças no índice podem refletir eventos e condições econômicas em todo o mundo. Por exemplo, se as economias estrangeiras estão prosperando, isso pode se refletir em lucros mais altos para as empresas do S&P 500 e, conseqüentemente, em um índice com retornos positivos.

Além disso, como explorado em (DE, 2013), o S&P 500 pode oferecer aos investidores uma maneira conveniente de obter exposição a mercados estrangeiros, haja vista que, em vez de investir diretamente em ações *offshore* — que podem ser menos acessíveis ou negociadas a preços mais elevados —, eles podem investir em empresas do S&P 500 que, como mencionado, fazem negócios significativos em outros países.

Em resumo, o S&P 500 é mais do que apenas um indicador da economia dos Estados Unidos. Graças à presença global das empresas que o compõem, ele também oferece uma visão valiosa da economia global. Isso é algo que os investidores podem querer considerar ao avaliar o índice e ao tomar decisões de investimento.

## 1.2 Motivação

Nesse contexto, entende-se que o mercado de ações é um mecanismo vital para o crescimento de empresas e também pode ser para o aumento da riqueza pessoal, ocupando, assim, um papel central em qualquer economia globalizada. Todavia, a complexidade e a volatilidade desse mercado, além da crescente difusão de informações de baixa confiabilidade, podem dificultar a tomada de decisão para muitos agentes ou até mesmo interromper prematuramente a jornada deles no mundo dos investimentos. Desse modo,

a criação de um modelo eficaz para prever movimentos de preços e auxiliar nas decisões tomadas representa uma contribuição social e economicamente pertinente.

Em síntese, o presente trabalho propõe a criação de um modelo computacional para a previsão do preço de fechamento diário do S&P 500. O projeto visa fornecer aos novos atores, que têm entrado no mercado financeiro brasileiro por meio do aprofundamento dos serviços financeiros no Brasil, um sistema capaz de ajudá-los na tomada de decisão diária para o *trading*.

Isso é particularmente valioso em um ambiente no qual a qualidade e a precisão das informações financeiras disponíveis podem variar significativamente, potencialmente dificultando as decisões informadas dos indivíduos e aumentando a vulnerabilidade deles às fraudes financeiras, conforme explorado por (CARVALHO, 2022).

Espera-se que, caso seja implementado de forma suficientemente eficaz, o modelo possa servir como uma fonte auxiliar de informação para prever o fechamento — e consequentemente o desempenho — desse mercado de ações no dia posterior ao de referência da negociação. Tal informação pode ser especialmente valiosa para *players* financeiros com baixa infraestrutura, isto é, capacidade de processamento limitada e análise tardia de informações financeiras, como não raramente ocorre com pequenas e/ou novas gestoras e também com investidores pessoa física.

### 1.3 Objetivo

Assim, o modelo proposto nesse trabalho visa utilizar alguns dos mais recentes desenvolvimentos no campo de previsões de séries temporais para realizar previsões diárias do índice S&P 500. Para isso, o modelo emprega uma arquitetura de rede neural recorrente composta por células LSTM, bem como um mecanismo de atenção para a ponderação dos resultados intermediários dessas.

Com a popularização e eficácia dos mecanismos de atenção em redes neurais do tipo *transformers*, evidenciada em *Large Language Models* (LLM) — como discutido por (NAVEED et al., 2023) —, optou-se por explorar o uso de tais mecanismos em um tipo diferente de sequência: séries financeiras. Assim, o trabalho se destaca por sua abordagem integradora para a tentativa de solução de problemas complexos atuais, com uma forte natureza interdisciplinar, combinando elementos de finanças, aprendizado de máquina e desenvolvimento de sistemas, alinhando-se aos princípios da Engenharia Mecatrônica discutidos ao longo do curso.

Portanto, nosso objetivo é construir um modelo focado em previsão, servindo como uma ferramenta que auxilia e busca fortalecer o processo decisório, em vez de ser um sistema que efetivamente toma as decisões. A eficácia do modelo será validada por meio de indicadores como a raiz do erro quadrático médio (RMSE), o erro absoluto médio (MAE) e o coeficiente de determinação ( $R^2$ ), além do *tracking error* (TE, erro de rastreamento, do inglês), em vez de comparações diretas com possíveis estratégias de investimento.

Todavia, ainda assim serão discutidas algumas estratégias de gestão de banca a fim de entender como o modelo poderia ser empregado na prática para que fosse possível rentabilizar com as previsões, tratando-se esse apenas de um estudo em ambiente controlado e sem a premissa de assegurar retornos consistentes.

Novamente, nosso foco principal reside na qualidade das previsões, não na tomada de decisões em si. A arquitetura da rede é baseada em um trabalho similar de (QIU et al., 2020), de modo que o modelo do artigo será usado como *benchmark* para a validação do modelo proposto nesse trabalho. A solução apresentada será considerada totalmente bem-sucedida se obtiver resultados iguais ou superiores aos relatados naquele estudo, no que se refere a algumas das métricas supracitadas.



## 2 REVISÃO BIBLIOGRÁFICA

No que se tange à previsão do preço de ativos financeiros, tarefa fundamental do presente trabalho, existe uma rica bibliografia basilar. Conforme (AL-HMOUZ et al., 2015) e (BARAK; MODARRES, 2015), a previsão de preços de ativos é uma atividade crucial no mercado financeiro, capaz, quando precisa, de gerar retornos ajustados ao risco que superam a média do mercado e podem inclusive auxiliar na proteção contra os riscos inerentes a ele.

Segundo (LAM, 2004) e (SEDIGHI et al., 2019), os estudos sobre previsão de preços de ativos de mercado podem ser classificados em duas abordagens principais: análise fundamentalista e análise técnica.

A análise fundamentalista baseia-se em informações financeiras presentes nos relatórios trimestrais divulgados pelas empresas. Nesse contexto, a previsão de preços dos ativos utiliza modelos de *valuation* (avaliação de empresas) que levam em conta tais dados para orientar a tomada de decisão sobre as ações de uma empresa, conforme descrito por (BODIE et al., 2014) e (MURPHY, 1999).

Em contrapartida, a análise técnica utiliza dados históricos de preços e volumes de negociação dos ativos para realizar previsões sobre os preços futuros, conforme mencionado por (TURNER, 2007) e (WEI et al., 2011). Esses conjuntos de dados, organizados de forma temporal, são conhecidos como “séries temporais”, e são utilizados, como explica (SINGH, 2000), para a identificação de padrões.

No entanto, no que diz respeito à previsão de séries temporais de dados financeiros, antes de tentar realizar previsões propriamente ditas, é preciso entender se é teoricamente possível fazê-las. A tentativa de prever preços futuros de ativos, em termos gerais, levanta uma questão amplamente discutida na literatura acadêmica: se os preços dos ativos de mercado são efetivamente previsíveis com base em dados históricos.

Por muitos anos, especialmente nas décadas seguintes a sua proposição, a hipótese dos mercados eficientes (EMH) foi amplamente aceita como o modelo predominante para

compreender o comportamento dos preços dos ativos. Segundo (FAMA, 1970), a EMH sustenta que, devido à natureza imprevisível das informações que chegam ao mercado, os preços dos ativos refletem todas as informações conhecidas, tornando as mudanças futuras nos preços dos ativos independentes de seus históricos e, portanto, inviabilizando retornos acima da média do mercado de forma consistente através da análise de informações passadas.

Entretanto, conforme (DANIEL et al., 1998) e mais recentemente (BOROVKOVA; TSIAMAS, 2019), desde a introdução da EMH diversas contestações surgiram a respeito da validade da teoria e de sua premissa fundamental — que os preços dos ativos de mercado são determinados racionalmente por seus participantes.

Naturalmente, diversas anomalias de mercado ocorreram ao longo dos anos, como discutido por (MALKIEL et al., 2005). Exemplos incluem a reação extrema dos mercados a novas informações (BONDT; THALER, 1985), a existência de *momentum* de curto prazo e reversões de longo prazo, e a alta volatilidade observada em diversos ativos (DANIEL et al., 1998), dentre outras. Esses eventos fundamentam argumentos contrários à validade da EMH.

Há também uma corrente de pesquisadores que defende que as anomalias de mercado estão, de fato, alinhadas com a EMH, argumentando que reações exageradas de compra e venda de ativos ocorrem de maneira aleatória e são igualmente frequentes (FAMA, 1998), e que investidores institucionais são capazes de contrabalançar as anomalias causadas no mercado por investidores menos sofisticados (SHILLER, 2003).

No entanto, muitos pesquisadores permanecem céticos quanto à capacidade de um modelo baseado na racionalidade dos investidores de explicar essas numerosas anomalias, como já apontado por (DANIEL et al., 1998).

Esse conflito entre a teoria predominante e a realidade dos mercados incentivou estudos sobre modelos que incorporam a psicologia humana nas previsões de mercado, culminando no campo das finanças comportamentais, conforme explorado por (BONDT; THALER, 1990) e (SHILLER, 2003). Essa abordagem questiona a racionalidade dos investidores e destaca vieses comportamentais como aversão a perdas, reações exageradas e o *fear of missing out* (FOMO), conforme mencionado por (LO, 2004).

Uma tentativa de reconciliar a EMH com as teorias das finanças comportamentais foi a hipótese dos mercados adaptativos de (LO, 2004), que sugere que, embora os mercados sejam em grande parte racionais, eles não o são em todos os momentos. Em períodos de crises ou extrema volatilidade, os agentes do mercado podem exibir comportamentos

predominantemente irracionais.

Assim, considerando a existência comprovada de anomalias de mercado, não é surpreendente que muitos agentes de mercado interpretem informações históricas sobre preços, desempenho e outros fatores como projeções de valores futuros, conforme observado por (PATEL; MARWALA, 2006).

Além disso, em média, investidores tendem a esperar que retornos de curto prazo obtidos com certos ativos persistam, considerando que tais ganhos anteriores possam refletir o sentimento do mercado em relação ao ativo, conforme destacado por (BUSTOS et al., 2011).

Com base nesses aspectos, utilizar informações históricas de um ativo para tentar prever seu valor futuro é uma abordagem razoável. Portanto, esse trabalho se baseia na fundamentação teórica apresentada para justificar o esforço empreendido, reconhecendo, no entanto, que se trata de um campo repleto de complexidades e nuances.

Nesse contexto, diversos estudos também têm empregado metodologias estatísticas de análise de séries temporais, baseadas em dados históricos, para prever os preços e retornos dos ativos, como mencionado por (EFENDI et al., 2018), que utilizou técnicas estatísticas tradicionais para análise de séries temporais no mercado de ações.

Contudo, o protagonismo dessas ferramentas foi atenuado pela disseminação e popularização dos algoritmos de aprendizado de máquina em variados campos da ciência. Diferentemente de alguns métodos estatísticos clássicos de análise de séries temporais, as técnicas de IA são amplamente capazes de lidar com dados não lineares e ruidosos, resultando, geralmente, em previsões mais robustas. Isso é evidenciado por (CHEN; HAO, 2017), que utilizou *Support Vector Machine* (SVM) e *K-Nearest Neighbors* (KNN) para prever preços de índices de ações.

Relativamente aos modelos de aprendizado de máquina, houve um avanço significativo no desenvolvimento de métodos de *deep learning* nos últimos anos, devido a sua gama de aplicações bem-sucedidas. Por exemplo, na classificação de imagens (MA et al., 2020), (CHEN et al., 2019b), (CHEN et al., 2019a); no reconhecimento de fala (LAM et al., 2018), (LI et al., 2018), (ZHANG et al., 2019a); e no processamento de linguagem natural (NLP) (CHAI; LI, 2019), (LIU et al., 2019), (MALYKH, 2019).

No contexto de problemas de séries temporais, a arquitetura de rede neural recorrente (RNN) tem sido amplamente utilizada. No entanto, para mitigar problemas no treinamento dessas redes, o modelo *Long Short-Term Memory* (LSTM) muitas vezes é preferido

(LECUN et al., 2015). Essa última arquitetura apresenta excelente desempenho para dados sequenciais e minimiza problemas relacionados ao desaparecimento ou explosão do gradiente, conforme demonstrado no artigo pioneiro (HOCHREITER; SCHMIDHUBER, 1997).

A notoriedade dessa arquitetura motivou diversas publicações visando a previsão do preço de ativos financeiros, incluindo modelos híbridos notáveis, como em (KIM; WON, 2018), que empregou a abordagem *generalized autoregressive conditional heteroskedasticity* (GARCH) combinada com uma rede LSTM para prever flutuações no preço de ações.

Outros exemplos incluem (LIU et al., 2017), que empregou uma rede convolucional (CNN) para desenvolver uma estratégia de *stock picking* e, posteriormente, aplicou uma LSTM para previsão do preço desses ativos; (ZHAO et al., 2017), que incorporou uma função de tempo ponderada a uma rede LSTM para prever a tendência de ativos financeiros; (JIANG et al., 2019), que explorou como LSTMs podem ser aplicadas na previsão de ativos negociados no mercado; e (JIN et al., 2020), que integrou uma análise de sentimentos, a técnica *empirical model decomposition* (EMD) e uma LSTM, visando previsões mais precisas de preços.

Por fim, a arquitetura LSTM, quando acoplada com mecanismos de atenção conforme exposto em (VASWANI et al., 2017), é comum em aplicações de processamento de linguagem natural e reconhecimento de imagens, mas ainda não é tão utilizada no campo das finanças. Assim, esse trabalho focará na aplicação dessa arquitetura no domínio financeiro, similar ao realizado por (QIU et al., 2020), que usou a transformada wavelet para reduzir o ruído dos dados históricos do S&P 500 e estabelecer um modelo de previsão do preço das ações, fornecendo, assim, um *benchmark* e a base teórica fundamental para o presente trabalho.

## 3 PADRÕES, REDES NEURAIS E ATENÇÃO

Nesta seção, discutem-se aspectos teóricos da conceituação e arquitetura de redes neurais artificiais, tanto tradicionais quanto recorrentes, além de considerações sobre padrões e séries temporais. Desenvolvem-se noções fundamentais das redes neurais como um todo, essenciais para a compreensão da arquitetura proposta nesse trabalho. Conclui-se a seção com uma abordagem sobre as vantagens que as redes LSTM oferecem em relação a outras arquiteturas e de como os mecanismos de atenção aplicados a essas podem aprimorar seu desempenho.

### 3.1 Padrões e Modelagem de Séries Temporais

Conforme relatado por (ABIODUN et al., 2019), as Redes Neurais Artificiais (ANNs) são frequentemente descritas como modelos estatísticos não-lineares que originalmente buscavam emular o funcionamento das redes neurais biológicas. Assim, as ANNs têm ganhado destaque por sua eficiência, ou pelo menos eficácia, na realização do reconhecimento de padrões.

Especificamente, o reconhecimento é um paradigma computacional utilizado para a classificação de dados brutos, englobando uma miríade de abordagens que possibilitam o desenvolvimento de diferentes aplicações em vários campos de atuação. A viabilidade dessas abordagens reside, primariamente, na tentativa de imitação inteligente do comportamento humano, no qual um padrão pode ser considerado um conjunto de itens, objetos, imagens, eventos, situações, características ou abstrações em geral, cujos aspectos dentro de cada conjunto são semelhantes em um sentido inequívoco (ABIODUN et al., 2019).

Conforme afirmou o célebre matemático estadunidense Norbert Wiener, “Padrão é um arranjo; ele é caracterizado pela sequência das características das quais é composto, mais do que pelas características inerentes subjacentes”<sup>1</sup> (ABIODUN et al., 2019). Essa definição destaca dois pontos de suma importância: (1) um padrão, sendo um arranjo, é

---

<sup>1</sup>Tradução livre dos autores

essencialmente uma organização ou configuração de elementos, não uma entidade única ou objeto singular, mas uma combinação de várias partes; (2) um padrão, sendo uma sequência, relaciona-se mais à ordem de seus elementos constituintes do que às características intrínsecas desses elementos. Por exemplo, em um padrão musical, a sequência das notas é mais importante do que a nota musical individual em si; ou em uma imagem, a cor ou forma de um único elemento não tem tanta importância quanto a disposição desses elementos no espaço.

Destarte, a modelagem de padrões em séries temporais — sequências de pontos de dados ordenados temporalmente — quando destinada à predição, refere-se ao processo de identificar relações e tendências em dados históricos para prever valores futuros. Essa abordagem é bem definida em (SINGH, 2000), no qual se propõe um novo método para a previsão de séries temporais utilizando-se também do próprio índice S&P 500, apresentando resultados superiores aos de métodos estáticos tradicionais nesse campo, como ARIMA.

Com relação ao trabalho mencionado, recorre-se à definição de padrões que, em uma série temporal, podem ser especificados em termos do gradiente dessa em um determinado intervalo, ou seja, um movimento ascendente ou descendente, sua magnitude e o número de instantes que contém.

Formalmente, considere um vetor  $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$ , em que  $n \in \mathbb{N}$  é o número de amostras da série, e a cada uma dessas associa-se univocamente um instante de tempo:  $t_1, t_2 = t_1 + 1$ , e sucessivamente até  $t_n = t_1 + (n - 1)$ . Dessa definição advém que um segmento pode ser expresso por  $\delta_i = y_{i+1} - y_i \ \forall \ 1 \leq i \leq n - 1$ , a partir da qual é possível afirmar que um padrão, no contexto de séries temporais, é então um conjunto com um ou mais segmentos, tal como ilustram os exemplos da Figura 1 (SINGH, 2000).

Conforme exposto por (SINGH, 2000), dessa definição também se infere que é possível definir padrões de distintas complexidades com base em seu tamanho estrutural: aqueles de maior complexidade apresentam tamanhos maiores, podendo, inclusive, conter outros padrões integralmente. Essa premissa norteia a modelagem do trabalho referenciado, sustentando a visão de que padrões observados no passado podem se repetir em momentos subsequentes.

Entretanto, é importante salientar que essa abordagem proposta por (SINGH, 2000), que envolve a tentativa consciente de reproduzir padrões estruturais em diferentes instantes de tempo, não está em total consonância com as aspirações do presente trabalho. Aqui, faz-se proveito primordialmente da formalização de séries temporais e das inferências

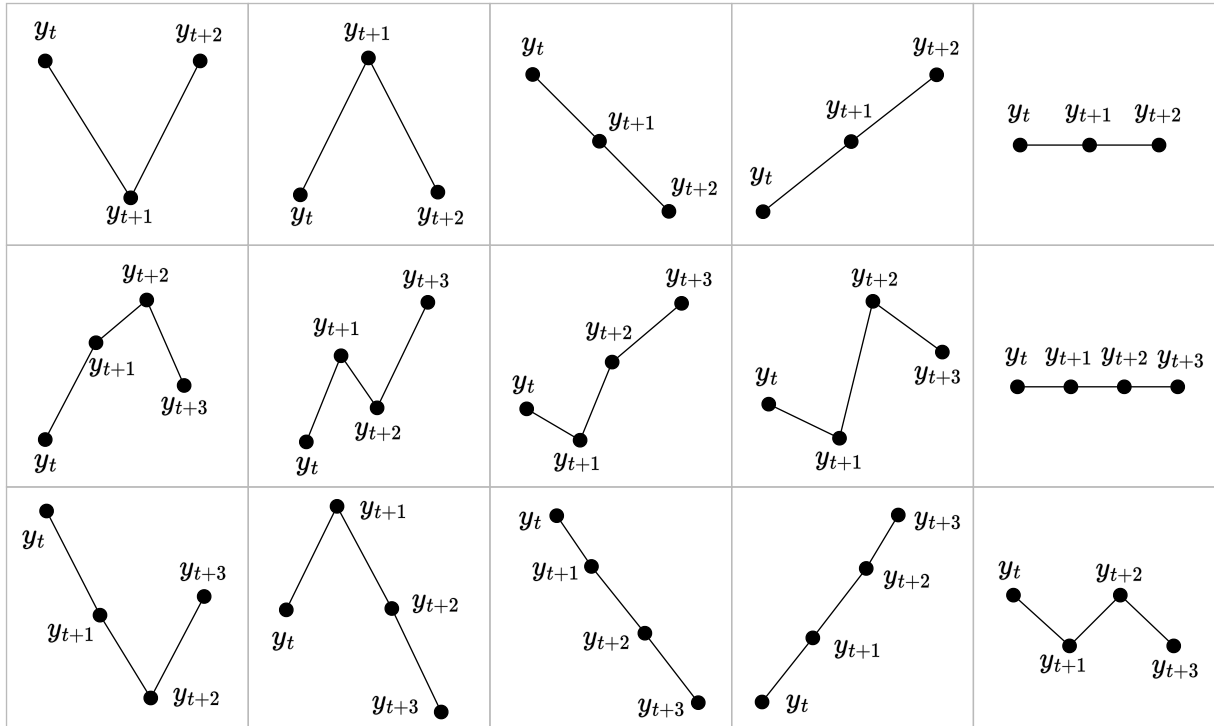


Figura 1: Representação gráfica de alguns exemplos de padrões temporais com segmentações de três e quatro pontos. Esses padrões ilustram variações sequenciais em séries temporais, demonstrando relações entre valores consecutivos.

Adaptado de (SINGH, 2000)

derivadas dessa.

## 3.2 Redução de Ruído

A redução de ruído é um processo crucial em diversas áreas de estudo, incluindo o processamento de sinais, a ciência da computação, a estatística e, até mesmo, as finanças. De forma concisa, conforme discutido por (ROSAS-OREA et al., 2012), o objetivo principal do *denoising* é extrair informações úteis de um sinal potencialmente distorcido pela presença de ruído, isto é, elementos que podem interferir na interpretação correta do sinal original. Simplificando, o processo de *denoising* envolve a remoção ou atenuação desse ruído, preservando ao máximo as características essenciais do sinal original.

Em geral, a redução de ruído em um sinal tem o propósito de aprimorar a qualidade da informação, possibilitando uma análise mais acurada. No contexto dos mercados financeiros, por exemplo, flutuações repentinas e abruptas de preços em períodos de alta volatilidade podem ser vistas como ruídos que mascaram tendências subjacentes mais significativas. Portanto, a remoção ou, ao menos, a minimização desse ruído é particularmente útil para a tomada de decisões de investimento mais informadas (KE et al., 2023).

Existem diversos métodos para a redução de ruído em um sinal, como a transformada de Fourier, que converte um sinal do domínio do tempo para o domínio da frequência, facilitando a identificação e remoção do ruído. Outro método, possivelmente menos comum mas especialmente útil para aplicação em sinais não estacionários — aqueles cujas propriedades estatísticas mudam com o tempo — é a transformada wavelet, que será explorada mais profundamente nesse trabalho. No entanto, é vital reconhecer que todos os métodos de *denoising* possuem suas limitações, e a escolha do mais apropriado deve considerar o tipo de sinal e o tipo de ruído presentes na amostra, conforme elucidado por (KE et al., 2023) e (ROSAS-OREA et al., 2012).

### 3.2.1 Transformada Wavelet

Conforme destacado por (QIU et al., 2020), a transformada wavelet tem propiciado avanços significativos em áreas como processamento de imagens e sinais, muito devido a sua habilidade de contornar algumas das limitações da análise clássica de Fourier, sendo cada vez mais adotada na economia e nas finanças.

Pode-se considerar que as séries temporais financeiras compartilham características com os sinais comumente analisados em engenharia, de modo que uma série temporal financeira possa ser interpretada como um sinal (QIU et al., 2020).



A estratégia fundamental da redução de ruído pela transformada wavelet envolve a decomposição do sinal em coeficientes, a determinação e o processamento de um limiar (*threshold*) e, por fim, a reconstrução do sinal. A aplicação desse método depende de fatores como a escolha de uma função de base (por exemplo, Haar, Daubechies, Coiflet, etc.) e a definição do número de níveis de decomposição, o que influencia diretamente a subsequente reconstrução do sinal (QIU et al., 2020) e (ROSAS-OREA et al., 2012).

Nesse contexto, os “coeficientes” referem-se aos valores obtidos ao aplicar a decomposição wavelet a um sinal. Existem dois tipos de coeficientes, a saber:

1. Coeficientes de escala/aproximação: resultantes da convolução do sinal com um filtro passa-baixa, esses coeficientes representam as tendências gerais do sinal, fornecendo, essencialmente, uma versão suavizada do mesmo.
2. Coeficientes de detalhe: resultantes da convolução do sinal com um filtro passa-alta, esses coeficientes capturam as componentes de alta frequência do sinal, isto é, os detalhes mais finos.

Os níveis, portanto, referem-se a diferentes estágios de “profundidade” da decomposição de um sinal, que é submetido repetidamente ao par de filtros mencionados, permitindo a cada iteração calcular novos coeficientes de detalhe e de aproximação.

Após a determinação dos coeficientes, aqueles de aproximação podem ser submetidos a um limiar preestabelecido. No contexto deste trabalho, optou-se por um limiar suave (Equação 3.1), em detrimento de um limiar rígido (Equação 3.2), cujo valor pode ser estabelecido por meio da Equação 3.3, conhecida como “limiar universal” (XING et al., 2015).

$$SOFT_T(x) = \begin{cases} x - T & x > T \\ x + T & x < -T \\ 0 & |x| \leq T \end{cases} \quad (3.1) \quad HARD_T(x) = \begin{cases} x & |x| \geq T \\ 0 & |x| < T \end{cases} \quad (3.2)$$

$$T = \sigma \sqrt{2 \ln(n)} \quad (3.3)$$

Nessa última,  $\sigma$  representa o desvio padrão do ruído — comumente considerado como o desvio padrão dos coeficientes de maior frequência, isto é, os coeficientes de detalhamento do primeiro nível da decomposição — e  $n$  é o tamanho da amostra.

O número de níveis escolhidos para uma transformada depende das características de interesse. Mais níveis fornecerão maior resolução para componentes de baixa frequência, pois se está, essencialmente, ampliando coeficientes de aproximação em detrimento da resolução de alta frequência. Assim, ao processar os coeficientes de detalhe em determinados níveis e em seguida utilizar a reconstrução wavelet para recuperar o sinal, é possível atenuar componentes de alta frequência que, em muitos casos práticos, estão associadas ao ruído.

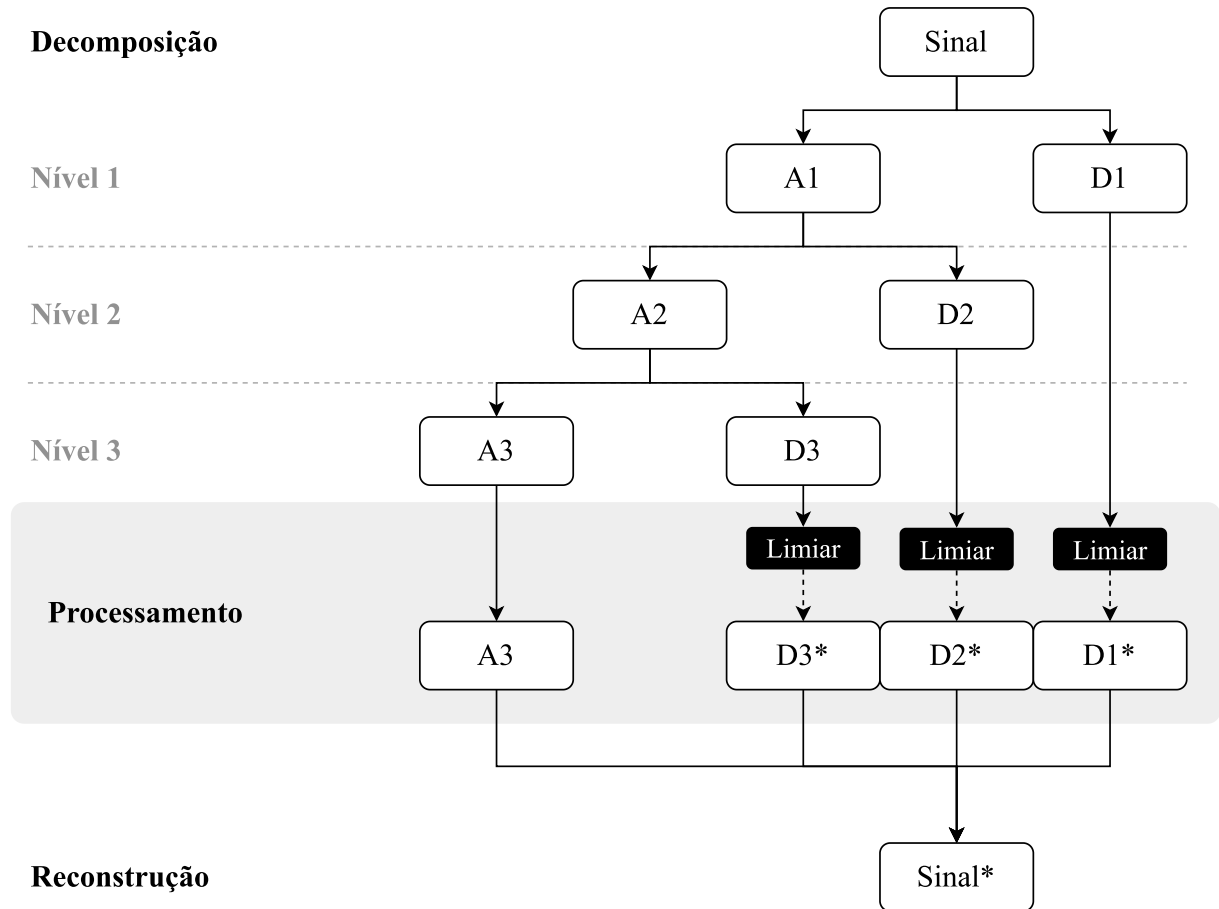


Figura 2: Esquematização da transformada wavelet com os coeficientes de aproximação ( $A$ ) e detalhamento ( $D$ ). No exemplo, aplica-se um limiar a todos os detalhamentos afim de suprimir o ruído. É esperado que a frequência decresça na cadeia até  $A3$ , sendo  $D1$  a componente de maior frequência.

Adaptado de (HEIDARY et al., 2010)

Em suma, conforme ilustrado na Figura 2, essa transformada permite uma análise em múltiplas resoluções, o que se mostra particularmente útil para diferenciar ruído de sinal. Contudo, a abordagem requer atenção especial, pois as componentes de alta frequência também podem conter informações valiosas, especialmente em séries financeiras, onde a eliminação excessiva da volatilidade pode distorcer os preços.

### 3.3 Redes Neurais Artificiais

Redes neurais artificiais são algoritmos computacionais modelados, inicialmente, com base na estrutura e no comportamento dos neurônios no cérebro humano, capazes de reconhecer e categorizar padrões complexos, conforme mencionado por (KHAN et al., 2001). Suas aplicações abrangem uma variedade de casos, incluindo a classificação de dados, a previsão de tendências e o reconhecimento de padrões, em diversos campos, tais como engenharia, medicina, agricultura, finanças e até mesmo artes (ABIODUN et al., 2018).

#### 3.3.1 *Feedforward*

Quanto à estrutura, uma ANN é organizada em camadas de neurônios — ou *neurons* — que são unidades de processamento simples. Em uma rede, cada neurônio recebe uma série de entradas e realiza algumas operações matemáticas sobre essas para produzir uma saída. Essa saída do neurônio pode então servir como entrada para outros neurônios na próxima camada da rede. Especificamente, as ANNs são geralmente compostas por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída, como descrito por (DASTRES; SOORI, 2021).

Em uma rede neural biológica, os neurônios comunicam-se através de sinapses; já em uma ANN, utiliza-se pesos para ponderar a transmissão de informações de uma camada para outra. É por meio do ajuste desses pesos durante o processo conhecido como “treinamento” que ocorre o “aprendizado” do modelo. Em outras palavras, esse treinamento consiste em uma etapa na qual se realiza um processo de minimização do erro; isto é, a rede utiliza regras pré-definidas para ajustar seus pesos com o objetivo de reduzir a diferença entre as previsões que ela produz e os valores reais esperados (KHAN et al., 2001).

Formalmente, seja  $\mathbf{x}$  um conjunto de dados composto por  $t$  amostras de dimensão  $n$ , na qual cada dimensão representa um atributo — ou seja, uma *feature* — da amostra considerada. Portanto, temos uma rede neural cuja camada de entrada deve conter necessariamente  $n$  neurônios. Cada neurônio nessa camada estará associado a cada um dos  $i$  neurônios da subsequente camada  $l$ , e a cada associação será atribuído um peso  $w_{ij}^l$ ; ou seja, tendo em vista a notação, um número real que pondera a informação recebida pelo nó  $i$  da camada  $l$  oriunda do nó  $j$  da camada  $l - 1$ . As camadas que seguem esse padrão de conexão de nós são conhecidas como densamente conectadas e, em geral, não possuem

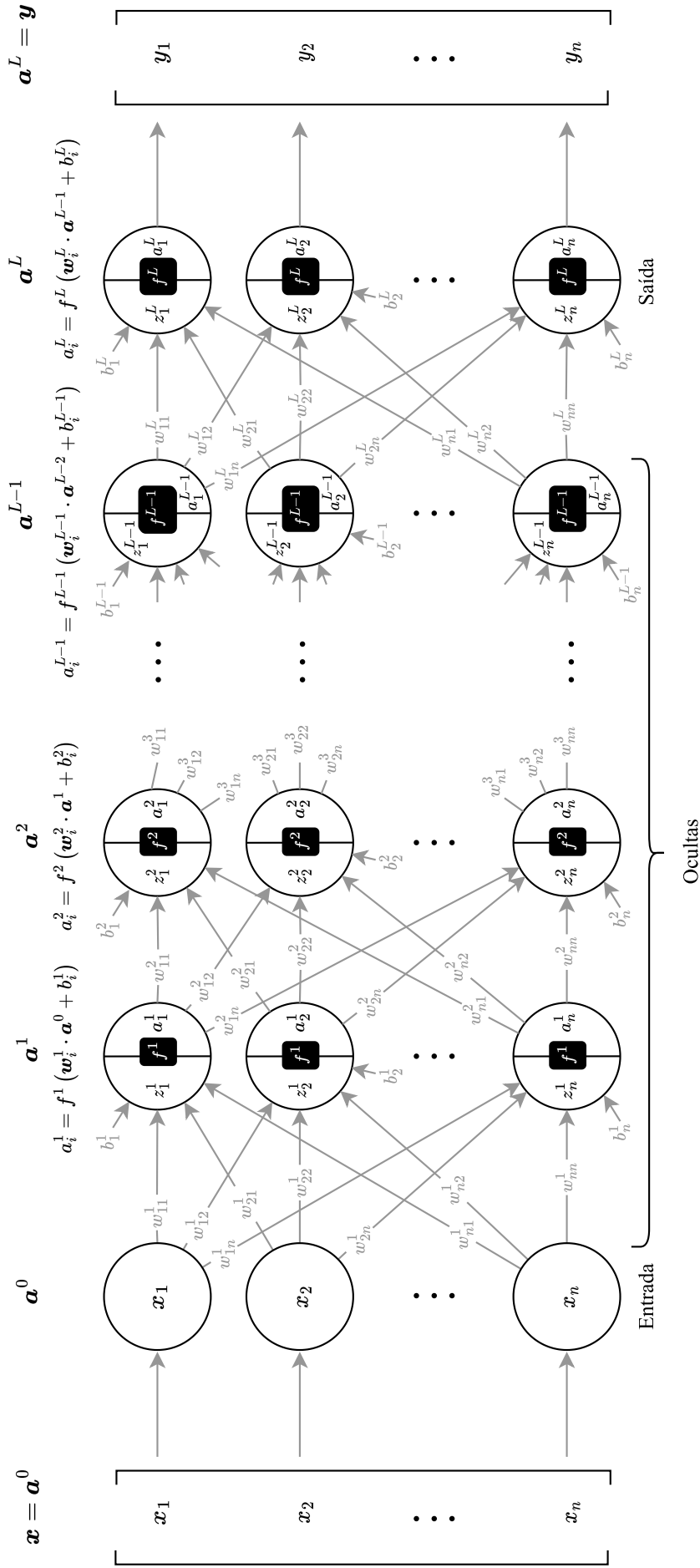


Figura 3: Diagrama esquemático da arquitetura de uma rede neural artificial multicamadas. A estrutura apresenta camadas de entrada, ocultas e de saída, com  $n$  neurônios em cada camada. As equações associadas ilustram as relações de transformação entre os neurônios.

Adaptado de (DASTRES; SOORI, 2021)

restrição estrita de dimensão (KRIEGESKORTE; GOLAN, 2019).

Em teoria, mediante essa configuração, a rede pode ser estendida indefinidamente (conforme o padrão estabelecido, como ilustra a Figura 3) por meio da inserção de camadas  $l+1$ ,  $l+2$ ,  $l+3$  e assim sucessivamente, todas análogas à camada  $l$  e denominadas “camadas ocultas”. Finalmente, ao se chegar à esperada camada final da rede, teremos a camada de saída, cuja dimensão corresponde à dimensão desejada para a resposta  $\mathbf{y}$ , como formaliza (KRIEGESKORTE; GOLAN, 2019).

Por meio dessa definição, formula-se um dos processos fundamentais das redes neurais: o *feedforward*, ou “alimentação para frente”. Semelhante ao apresentado em (KRIEGESKORTE; GOLAN, 2019) e (NWANKPA et al., 2018), o  $i$ -ésimo neurônio da camada  $l$  possui um valor de saída  $a_i^l$ , que é determinado aplicando-se uma função  $f^l$  — denominada “função de ativação”, motivo pelo qual se justifica o uso do símbolo “ $a$ ” — ao produto escalar dos pesos  $\mathbf{w}_i^l$  com a saída da camada anterior  $\mathbf{a}^{l-1}$ , além de um viés  $b_i^l$  que é adicionado ao resultado, aumentando um grau de liberdade desse. Matematicamente, para um único nó, pode-se escrever a Equação 3.4.

$$a_i^l = f^l (\mathbf{w}_i^l \cdot \mathbf{a}^{l-1} + b_i^l) \quad (3.4)$$

Isso corresponde à ativação ou não de um neurônio, um dos principais fluxos de informação de uma rede neural. A Equação 3.4 é comumente apresentada com um passo intermediário, no qual se considera primeiro uma variável auxiliar  $z_i^l = \mathbf{w}_i^l \cdot \mathbf{a}^{l-1} + b_i^l$ , e a ativação torna-se diretamente e exclusivamente uma função dessa (KRIEGESKORTE; GOLAN, 2019). Para todos os casos, os vetores apresentados são definidos como:

$$\mathbf{a}^{l-1} = \begin{bmatrix} a_1^{l-1} & a_2^{l-1} & a_3^{l-1} & \dots & a_j^{l-1} \end{bmatrix} \quad \mathbf{w}_i^l = \begin{bmatrix} w_{i1}^l \\ w_{i2}^l \\ w_{i3}^l \\ \vdots \\ w_{ij}^l \end{bmatrix}$$

### 3.3.2 Funções de Ativação

Como aborda (NWANKPA et al., 2018), as funções de ativação são um componente crucial desse processo. Para essa finalidade, são comumente utilizadas a função sigmoide (Equação 3.5), a função tangente hiperbólica (Equação 3.8) e a função linear retificada

(ReLU, Equação 3.6), além de suas variantes. As duas primeiras são bastante empregadas na obtenção de saídas binárias, apesar da natureza contínua dessas.

Atualmente, a função ReLU é uma das funções de ativação mais populares para redes neurais profundas, tanto devido a sua eficiência computacional quanto ao fato de essa não ser tão prejudicada pelo problema do desaparecimento do gradiente. Também conhecido como *vanishing gradient*, esse problema consiste na diminuição latente dos gradientes utilizados na atualização dos pesos da rede à medida que esses são propagados, resultando em pouco ou nenhum efeito no treinamento. Tal problema é particularmente acentuado nas outras funções citadas em virtude da restrição que essas impõem aos valores de entrada, como é possível destacar verificando a Figura 4 (NWANKPA et al., 2018).

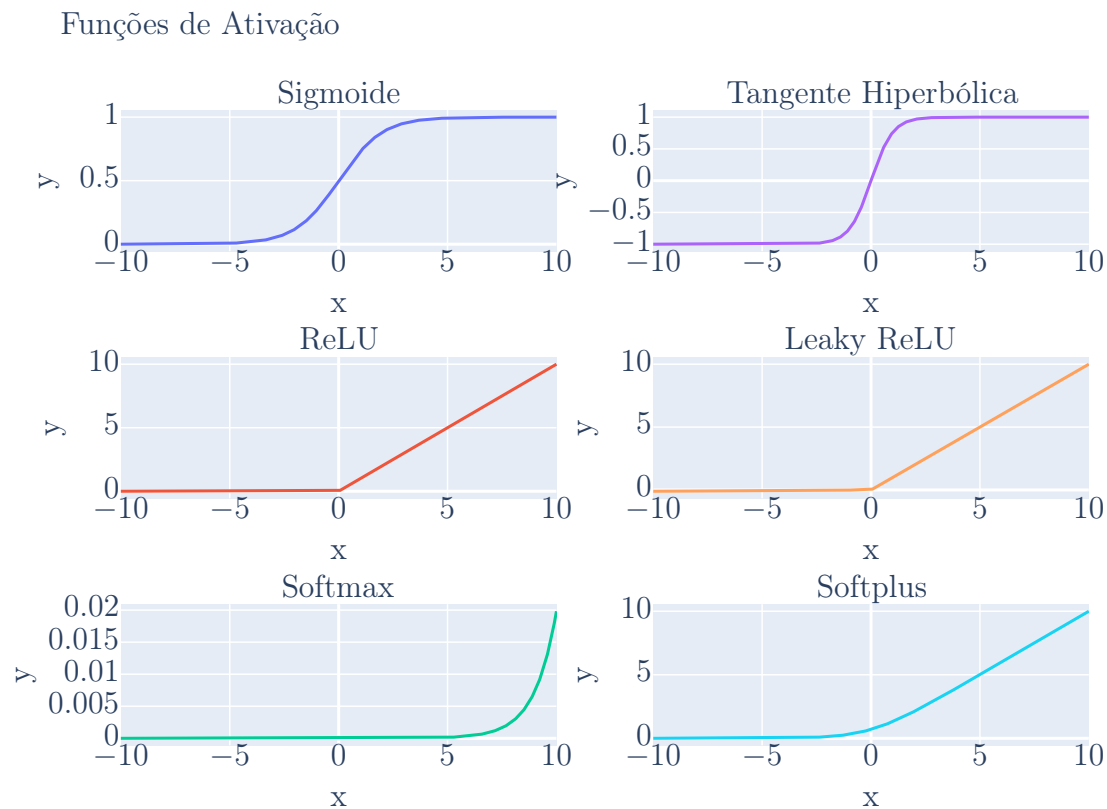


Figura 4: Representação gráfica de funções de ativação frequentemente utilizadas em redes neurais, incluindo: Sigmóide, Tangente Hiperbólica, ReLU, Leaky ReLU, Softmax e Softplus. Cada gráfico ilustra a relação entre a entrada ( $x$ ) e a saída ( $y$ ) para a respectiva função.

Autoria própria

Todavia, como explica (NWANKPA et al., 2018), é sabido que a ReLU também pode ser afetada pelo problema dos “neurônios mortos” (ou desativados), no qual alguns dos nós

da rede se tornam inativos e deixam de participar ativamente do processo de aprendizado da rede. Grande parte sob essa motivação, foram proposta diversas variantes da ReLU, como (SHAMIR et al., 2020) discute aprofundadamente, sendo a ReLU com vazamento (*Leaky ReLU*, Equação 3.9) uma das mais notáveis.

Já a função *softplus* (Equação 3.10) trata-se de uma versão suavizada da função ReLU, isto é, mais uma de suas variantes. A função *softplus* produz uma saída que é quase linear para entradas grandes, mas que se aproxima de zero para entradas negativas. O princípio disso também visa mitigar o aparecimento de neurônios desativados (SHAMIR et al., 2020).

$$f(x) = \frac{1}{1 + e^x} \quad (3.5)$$

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (3.8)$$

$$f(x) = \max(0, x) \quad (3.6)$$

$$f(x) = \max(0.1x, x) \quad (3.9)$$

$$f(x)_i = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \quad (3.7)$$

$$f(x) = \log(1 + e^x) \quad (3.10)$$

Finalmente, como relata (NWANKPA et al., 2018), a função *softmax* (apresentada na Equação 3.7) é uma função de ativação comumente utilizada na camada de saída de um classificador de múltiplas classes em redes neurais. Formalmente, ela transforma um vetor de  $K$  dimensões de números reais em um vetor de  $K$  probabilidades, onde cada elemento do vetor representa a probabilidade de que a entrada da rede pertença a uma determinada classe de saída. Decorre disso que a soma de todos os valores de um vetor produzido pela função *softmax* é sempre unitária.

Conclui-se, então, que a escolha da função de ativação depende, pelo menos, tanto do tipo específico de problema de aprendizado de máquina que está sendo resolvido quanto das dimensões da rede proposta.

### 3.3.3 *Backpropagation*

O gradiente, já citado anteriormente, é uma medida da taxa de alteração de uma função. Em redes neurais, essa ferramenta matemática é utilizada para atualizar os pesos e vieses durante o treinamento, cujo objetivo, como mencionado, é minimizar a função de custo, que quantifica o desvio entre a saída prevista pela rede neural e a saída real. Como o gradiente aponta na direção de maior aumento da função de custo, é necessário atualizar os pesos e vieses na direção oposta justamente para minimizar essa função

(KRIEGESKORTE; GOLAN, 2019).

Para tal, é preciso compreender que a função de custo  $J(w, b)$  é influenciada pelos valores dos pesos  $w$  e dos vieses  $b$  por meio das previsões da rede. Essa relação pode ser expressa considerando uma rede fictícia composta por um único neurônio com ativação linear que recebe uma entrada  $x$ . Assim, uma vez que este neurônio calcule a saída  $a = x \cdot w + b$ , a função de custo, como o erro quadrático médio (MSE), pode ser escrita incorporando essa operação:

$$J(w, b) = (y - a)^2 = [y - (x \cdot w + b)]^2$$

Esta formulação de  $J(w, b)$  demonstra que qualquer alteração em  $w$  ou  $b$  resultará em uma mudança correspondente no valor de  $J(w, b)$ , evidenciando mais diretamente a dependência da função de custo desses parâmetros. Portanto, durante o treinamento, são os valores de  $w$  e  $b$  que a rede ajusta para minimizar  $J(w, b)$  (KRIEGESKORTE; GOLAN, 2019). Com isso em mente, pode-se construir a regra de atualização, apresentada na Equação 3.11 para os pesos e na Equação 3.12 para os vieses.

$$w_{ij, novo}^l = w_{ij}^l - \alpha \frac{\partial J(w, b)}{\partial w_{ij}^l} \quad (3.11)$$

$$b_{i, novo}^l = b_i^l - \alpha \frac{\partial J(w, b)}{\partial b_i^l} \quad (3.12)$$

Nessas equações,  $\alpha$  é conhecido como *learning rate*. Esta taxa de aprendizado determina a magnitude dos ajustes corretivos do modelo ao corrigir os erros em cada atualização. Valores típicos variam numa faixa de 0.1 a 0.0001, embora a escolha ideal possa depender significativamente da aplicação específica, do tipo de modelo, do conjunto de dados e do método de otimização usado (SMITH, 2018). Uma alta taxa de aprendizado pode reduzir o tempo de treinamento, mas, possivelmente, compromete a precisão, podendo até ocasionar problemas de convergência. Por outro lado, uma taxa de aprendizado mais baixa torna o modelo mais conservador, potencialmente aumentando a eficácia, mas exigindo mais tempo para convergir (DASTRES; SOORI, 2021).

Observa-se que, após a propagação da informação por toda a rede no sentido direto, processo resumido pelo *feedforward*, a atualização de pesos e vieses é realizada de maneira mais coerente percorrendo-se o sentido inverso, conforme também discutido em (KRIEGESKORTE; GOLAN, 2019). Para tal, considere uma rede com  $L$  camadas, ao aplicar a



regra de atualização (Equação 3.11 e Equação 3.12), torna-se necessário empregar a regra da cadeia:

$$\begin{aligned}\frac{\partial J(w, b)}{\partial w_{ij}^{L-1}} &= \frac{\partial z_i^L}{\partial w_{ij}^{L-1}} \cdot \frac{\partial a_i^L}{\partial z_i^L} \cdot \frac{\partial J(w, b)}{\partial a_i^L} \\ \frac{\partial J(w, b)}{\partial b_{ij}^{L-1}} &= \frac{\partial z_i^L}{\partial b_{ij}^{L-1}} \cdot \frac{\partial a_i^L}{\partial z_i^L} \cdot \frac{\partial J(w, b)}{\partial a_i^L}\end{aligned}$$

De maneira análoga, para a atualização de pesos e vieses da camada  $L - 2$ , também é preciso calcular:

$$\begin{aligned}\frac{\partial J(w, b)}{\partial w_{ij}^{L-2}} &= \frac{\partial z_i^{L-1}}{\partial w_{ij}^{L-2}} \cdot \frac{\partial a_i^{L-1}}{\partial z_i^{L-1}} \cdot \frac{\partial J(w, b)}{\partial a_i^{L-1}} \\ \frac{\partial J(w, b)}{\partial b_{ij}^{L-2}} &= \frac{\partial z_i^{L-1}}{\partial b_{ij}^{L-2}} \cdot \frac{\partial a_i^{L-1}}{\partial z_i^{L-1}} \cdot \frac{\partial J(w, b)}{\partial a_i^{L-1}}\end{aligned}$$

Ao propagar essa lógica para todas as camadas, de  $L$  até a camada inicial, percebe-se a interdependência entre elas e, sobretudo, destaca-se o fluxo de informações que ocorre no sentido contrário ao estabelecido previamente para a rede. Em outras palavras, trata-se do mecanismo conhecido como “retropropagação”.

### 3.3.4 Funções de Custo

As funções de custo, também conhecidas como funções de perda, são componentes essenciais das redes neurais artificiais, pois fornecem uma medida de quão bem o modelo está realizando suas previsões. Conforme abordado na subseção anterior — sob a forma  $J(w, b)$  —, tais funções quantificam o erro entre o valor predito e o valor real conhecido. Diferentes funções de custo podem ser aplicadas, dependendo do problema específico que a rede neural está tentando resolver (SUDRE et al., 2017).

Dentre a vasta gama de funções de custo comumente utilizadas em soluções que envolvem redes neurais, pode-se destacar, por exemplo, a função de erro quadrático médio, expressa através da Equação 5.2. Ainda assim, existem outras funções de custo proeminentes como a do erro absoluto médio (MAE) na Equação 5.4 — que é menos sensível a *outliers* do que a MSE — e a função de entropia cruzada, frequentemente utilizada em problemas de classificação com categorias múltiplas, expressando a distância entre as distribuições de probabilidade reais e previstas.

A escolha da função de custo influencia diretamente a forma como o modelo é atu-

alizado. Se a função for excessivamente sensível a pequenas mudanças, poderá provocar atualizações que ultrapassem o ponto ótimo, fazendo o modelo oscilar sem nunca convergir para uma solução. Por outro lado, uma função de custo muito insensível pode resultar em atualizações tão ínfimas nos parâmetros que fazem o modelo convergir muito lentamente ou cair mais facilmente em mínimos locais, conforme explicam (CHRISTOFFERSEN; JACOBS, 2004) ao estudar o efeito das funções de perda em redes neurais para a precificação de opções.

### 3.4 Redes Neurais Recorrentes

Redes neurais recorrentes (RNNs) são uma extensão das redes neurais artificiais tradicionais, projetadas para lidar com dados sequenciais, como séries temporais ou linguagem natural. Diferentemente das redes neurais tradicionais, que assumem independência entre as entradas, as RNNs possuem conexões recorrentes que permitem a propagação de informações ao longo de uma sequência, conforme mostra a Figura 5. Isso as torna particularmente eficazes para tarefas que envolvem dependências temporais, como reconhecimento de fala e modelagem de linguagem natural (SHEWALKAR et al., 2019).

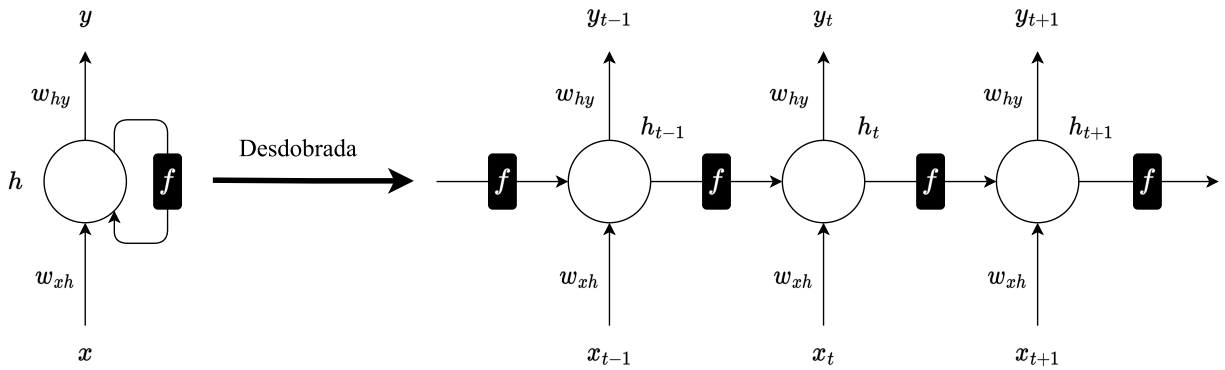


Figura 5: Representação esquemática da arquitetura de uma rede neural recorrente e sua respectiva versão desdobrada ao longo do tempo. A diagramação destaca os componentes principais, incluindo os estados ocultos  $h$ , entradas  $x$  e saídas  $y$ , bem como os pesos associados ( $w_{hy}$  e  $w_{xh}$ ).

Adaptado de (SHERSTINSKY, 2020)

#### 3.4.1 Formulação

Conforme mencionado, em uma RNN, a saída atual depende não apenas da entrada atual, mas também do estado oculto anterior. Assim, as expressões matemáticas que

regem a operação de uma rede neural recorrente são semelhantes às de uma rede neural clássica e são dadas pela Equação 3.13 e pela Equação 3.14, como mostrado por (SHEWALKAR et al., 2019).

$$\mathbf{h}_t = f(\mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{xh}\mathbf{x}_t + \mathbf{b}_h) \quad (3.13)$$

$$\mathbf{y}_t = \mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_y \quad (3.14)$$

Nas equações apresentadas, cada variável e parâmetro tem um papel específico no funcionamento da RNN, paralelo ao de uma rede neural tradicional multicamadas. No tempo  $t$ ,  $\mathbf{h}_t$  representa o estado oculto, análogo às ativações dos neurônios em uma camada oculta de uma rede neural padrão. Este estado oculto é um vetor cuja dimensão depende do número  $n_h$  de neurônios na camada recorrente e é fundamental para manter informações da sequência ao longo do tempo (HU et al., 2018).

A entrada  $\mathbf{x}_t$  é o vetor de dados de entrada no tempo  $t$ . Assim como em uma rede neural tradicional, sua dimensão depende do número  $n_x$  de características de entrada. Similarmente,  $\mathbf{y}_t$  é o vetor de saída da rede, e sua dimensão  $n_y$  é determinada pelo número de *targets* ou classes que a rede deve prever (HU et al., 2018).

Os termos  $\mathbf{W}_{hh}$ ,  $\mathbf{W}_{xh}$  e  $\mathbf{W}_{hy}$  são as matrizes de pesos que governam a transformação dos dados em cada etapa da rede. Em particular,  $\mathbf{W}_{hh}$  é a matriz de dimensão  $(n_h, n_h)$  que contém os pesos das conexões recorrentes, lidando com a contribuição do estado anterior  $\mathbf{h}_{t-1}$  para a transformação no estado atual  $\mathbf{h}_t$ .  $\mathbf{W}_{xh}$  é a matriz de pesos que transforma a entrada atual  $\mathbf{x}_t$  e sua dimensão é  $(n_h, n_x)$ . Já  $\mathbf{W}_{hy}$  transforma o estado oculto  $\mathbf{h}_t$  na saída  $\mathbf{y}_t$  e tem dimensão  $(n_y, n_h)$  (HU et al., 2018).

Finalmente, os vetores de viés  $\mathbf{b}_h$  e  $\mathbf{b}_y$  — análogos aos das redes tradicionais — adicionam um termo constante às transformações, e  $f$  denota a função de ativação que introduz a não-linearidade necessária para a rede. Funções de ativação comuns em RNNs incluem a tangente hiperbólica (Equação 3.8) e a função sigmoide (Equação 3.5), embora outras possam ser utilizadas, dependendo da especificidade da tarefa (HU et al., 2018).

### 3.4.2 Limitações e Alternativas

As RNNs, no entanto, enfrentam desafios significativos durante o treinamento, principalmente devido aos problemas conhecidos como *vanishing* e *exploding gradients*. Como

discutido anteriormente, esses problemas ocorrem quando os gradientes, que são usados para atualizar os pesos da rede durante o treinamento, se tornam muito grandes (explosão) ou muito pequenos (desaparecem), dificultando a convergência do modelo para uma solução ideal (YUE et al., 2018).

O problema do desaparecimento e explosão de gradientes é particularmente nocivo em sequências longas, onde a informação precisa ser propagada por muitos passos de tempo. Isso ocorre porque, como já expresso matematicamente, o gradiente de uma camada é calculado como o produto dos gradientes de todas as camadas subsequentes, pela recorrência da rede. Assim, se os gradientes são pequenos ( $\ll 1$ ), o produto pode se tornar insignificante, levando ao desaparecimento desse; por outro lado, se os gradientes são grandes ( $\gg 1$ ), o produto pode se tornar exorbitante, levando à explosão do gradiente (YU et al., 2019a).

Para mitigar esses problemas, foram introduzidas variantes de RNNs, como a *Long Short-Term Memory* (LSTM) e a *Gated Recurrent Unit* (GRU). Essas redes utilizam mecanismos de *gating* para controlar o fluxo de informações ao longo do tempo, permitindo que a rede “lembre” ou “esqueça” informações conforme necessário. Isso torna essas redes mais capazes de lidar com dependências de longo prazo, melhorando o desempenho delas em diversas tarefas, em comparação com as redes mais tradicionais (SHEWALKAR et al., 2019).

### 3.5 *Long-Short Term Memory*

O modelo LSTM foi introduzido por (HOCHREITER; SCHMIDHUBER, 1997) como uma arquitetura caracterizada por células de memória únicas, em substituição aos neurônios tradicionais encontrados nas camadas ocultas das RNNs.

Essa arquitetura avançada de rede neural recorrente foi desenvolvida para mitigar os problemas associados à dependência de longo prazo, uma limitação significativa das RNNs tradicionais, nas quais a rede perde a capacidade de manter informações anteriores relevantes para previsões ou análises futuras (LECUN et al., 2015).

#### 3.5.1 Estrutura

A chave para o funcionamento do modelo LSTM reside em sua estrutura. Diferentemente das RNNs, nas quais a informação é transferida apenas de um passo temporal para o próximo, a LSTM possui mecanismos denominados “portões” (ou *gates*, em inglês) que

regulam o fluxo de informações (QIU et al., 2020).

Desse modo, estruturalmente, cada célula é composta por subunidades: os *gates* de esquecimento, entrada e saída, além da unidade de estado da célula. A Figura 6 ilustra essa estrutura fundamental (QIU et al., 2020).

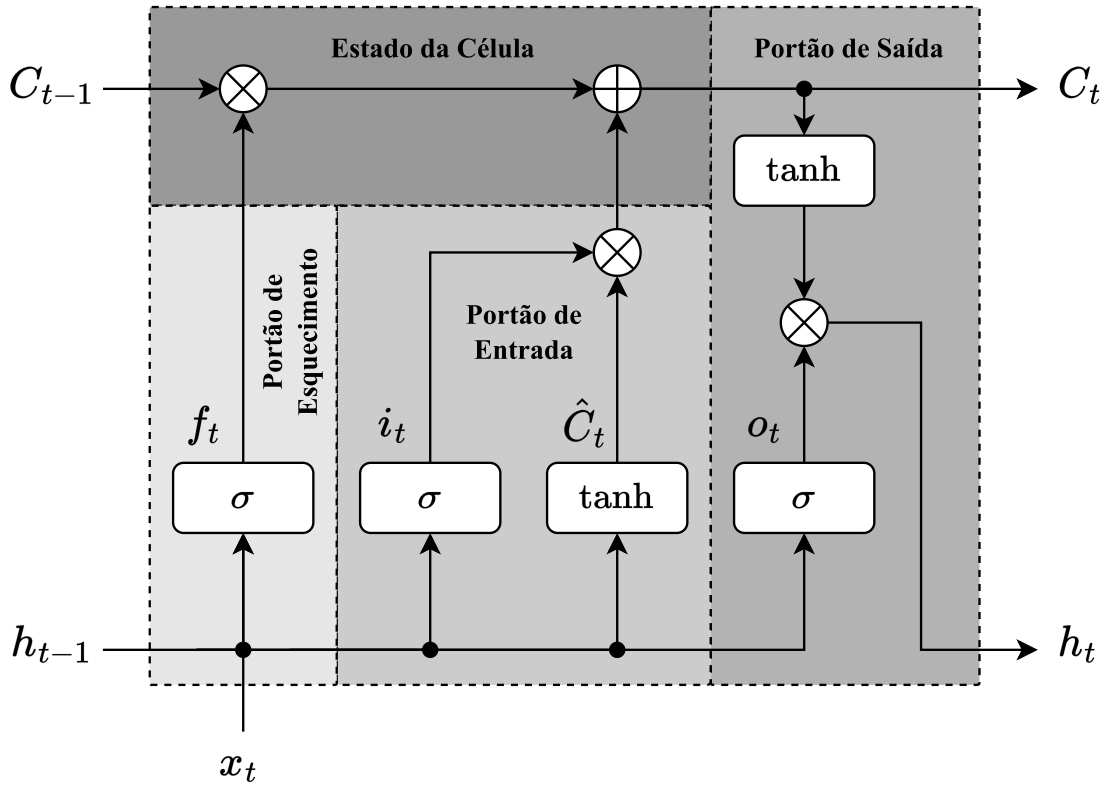


Figura 6: Diagrama da arquitetura de uma célula LSTM destacando os principais componentes e subunidades, incluindo os portões de esquecimento, entrada e saída, bem como os estados da célula e suas respectivas funções de ativação.

Adaptado de (QIU et al., 2020)

Muito em virtude da introdução desses mecanismos de portões, que alteram significativamente o funcionamento da célula, adotou-se nessa arquitetura uma simbologia que difere um tanto daquela apresentada nos modelos anteriores. Isso se deve, em parte, à quantidade de variáveis e processos envolvidos, optando-se por símbolos que remetam ao nome do processo em inglês. Aborda-se, então, na sequência, com maior detalhamento, cada subprocesso que compõe o funcionamento de uma célula característica, adotando-se  $t$  como o instante de tempo atual.

### 3.5.2 Portão de Esquecimento

Definido por  $f_t$  (*forget*), conforme a Equação 3.15, ele é empregado na determinação de quais informações devem ser mantidas ou descartadas do estado da célula anterior.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3.15)$$

Na qual  $\sigma$  é a função de ativação sigmoide,  $W_f$  é a matriz de pesos e  $b_f$  o vetor de vieses, em consonância com o que ocorre nos modelos anteriores. Esse *gate* combina na fórmula tanto a entrada atual  $x_t$  quanto a saída da célula anterior  $h_{t-1}$ . O resultado então é um vetor com valores entre 0 (completamente esquecido) e 1 (completamente mantido), justificando a analogia com um portão ou filtro (HOCHREITER; SCHMIDHUBER, 1997).

### 3.5.3 Portão de Entrada

Simbolizado por  $i_t$  (*input*) e expresso na Equação 3.16, é o responsável por controlar a atualização do estado da célula, ponderando o vetor de estado candidato.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3.16)$$

Nas quais  $W_i$  e  $b_i$  são, respectivamente, os pesos e vieses e correspondem a uma transformação da entrada. Ressalta-se ainda que a função  $\sigma$  garante que os valores permaneçam entre 0 e 1, como no *gate* anterior, indicando a intensidade das informações permitidas através do portão (HOCHREITER; SCHMIDHUBER, 1997).

### 3.5.4 Estados da Célula

Cada célula LSTM trabalha ponderando dois valores centrais na determinação de seu estado  $C_t$  (*cell*). São eles  $C_{t-1}$ , o vetor que representa o estado da célula no tempo anterior, e  $\hat{C}_t$ , denominado “estado candidato”, calculado através da Equação 3.17, na qual  $W_C$  é a matriz de pesos e  $b_C$  é o vetor de viés.

$$\hat{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3.17)$$

Uma vez determinado, a Equação 3.18 é utilizada para ponderar a influência dos referidos estados ( $C_{t-1}$  e  $\hat{C}_t$ ) na determinação do novo estado ( $C_t$ ), que é o vetor que contém as informações atuais e, conseqüentemente, mantém as informações ao longo do

processamento da sequência — em teoria, atua como a memória de longo prazo.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t \quad (3.18)$$

Assim, nota-se que o portão de esquecimento atua, justamente, ponderando o estado anterior, enquanto o portão de entrada atua sobre o estado candidato. Isso combina, como citado, a informação que a LSTM deseja reter do passado com as novas informações (HOCHREITER; SCHMIDHUBER, 1997).

### 3.5.5 Portão de Saída

Representado por  $o_t$  (*output*), o portão de saída controla quais informações do estado da célula atual (recém-calculado) serão também incorporadas na saída ( $h_t$ ). A Equação 3.19 ilustra esse processo.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3.19)$$

Analogamente aos outros portões e também aos modelos anteriores, temos novamente uma matriz de peso e um vetor de vieses, respectivamente,  $W_o$  e  $b_o$  (HOCHREITER; SCHMIDHUBER, 1997).

### 3.5.6 Estado Oculto

A saída final, ou estado oculto, representado por  $h_t$  (*hidden* — assim como nas RNNs tradicionais), é então uma versão filtrada do estado da célula, influenciada pela atuação do portão de saída, como ilustra a Equação 3.20.

$$h_t = o_t \cdot \tanh(C_t) \quad (3.20)$$

Isso permite que a célula LSTM tenha um controle mais refinado sobre o que é transmitido ao longo da cadeia de processamento, sendo essa considerada a memória de curto prazo da rede (HOCHREITER; SCHMIDHUBER, 1997).

### 3.5.7 Considerações Finais

Com isso posto, ficam evidentes algumas das vantagens que a LSTM apresenta. Apesar de sua inerente maior complexidade em comparação com os modelos de RNNs mais tradicionais, a célula se vale de mecanismos que conseguem perpetuar um estado mais

temporalmente distante,  $C_t$ , e, em paralelo, ponderá-lo para propagar uma saída mais próxima,  $h_t$  (YU et al., 2019b).

Em suma, enquanto em uma RNN a informação é passada diretamente do passo anterior para o atual sem modificações ou qualquer controle dedicado, em uma LSTM, os portões de esquecimento, entrada e saída modulam essas informações. Isso permite que, em um ramo, a rede se lembre ou se esqueça de informações mais longínquas, e, em outro, entenda relações mais imediatas, aprendendo, ao fim, como criar e manter dependências ao longo de sequências mais extensas (YU et al., 2019b).

## 3.6 Mecanismos de Atenção

Como observado nas redes neurais, muitos algoritmos e mecanismos computacionais são inspirados, a princípio, por fenômenos biológicos. Expandindo ainda mais esse campo, o conceito de atenção no aprendizado de máquina é, em certo grau, motivado por como a atenção é compreendida em seres humanos, embora de maneira bastante simplificada e abstrata.

### 3.6.1 Analogia Biológica

Na cognição humana, a atenção nos permite focar em partes específicas de um estímulo maior, por exemplo, prestando atenção em uma única conversa em meio a um ambiente ruidoso, como uma festa ou um restaurante cheio — fenômeno conhecido como “efeito coquetel” (BRONKHORST, 2000). Outro exemplo notável dessa atenção seletiva em ação é quando estamos buscando alguém em uma multidão.

É esse foco seletivo que nos permite processar grandes quantidades de informações de forma mais eficiente, onde nossa mente consegue “filtrar” ruídos de fundo ou estímulos menos relevantes, permitindo-nos concentrar em informações mais críticas para nossa interação ou sobrevivência, conforme discutido por (GERVEN; BEHRENS, 2018).

### 3.6.2 Aplicações

Demonstrou-se que o mecanismo de atenção tem bom desempenho quando aplicado a dados serializados, como no reconhecimento de fala, tradução automática e marcação de partes de um discurso. Por causa disso, as redes neurais baseadas nesses mecanismos já abrangem uma ampla gama de áreas: (MNIH et al., 2014) utilizou o mecanismo de atenção



em um modelo RNN para classificar imagens, focando nas partes mais relevantes para reduzir a complexidade da tarefa; (BAHDANAU et al., 2014) aplicou este mecanismo no campo do processamento de linguagem natural, especificamente, na tradução automática e obteve resultados próximos ao do estado da arte naquele momento.

No âmbito das finanças, mecanismos de atenção são frequentemente utilizados para previsão de preços de ações indiretamente, através da extração de informações de notícias, desempenhando assim um papel mais auxiliar na análise das flutuações do mercado (QIU et al., 2020). Por exemplo, (LIU, 2018) propôs uma rede neural recorrente baseada em atenção e treinada com notícias do mercado financeiro para inferir os preços de ativos.

### 3.6.3 Atenção Suave e Rígida

Um mecanismo de atenção pode ser caracterizado por dois passos centrais: primeiro, o cálculo de um *score*, que determina a relevância (peso) de cada segmento da entrada; em seguida, a formação do vetor de contexto, que é uma combinação das partes da entrada ponderada pelos *scores* calculados. Assim, ao final, espera-se que esse vetor represente as informações mais pertinentes, permitindo que o modelo se concentre nas partes essenciais da entrada (QIU et al., 2020).

Além disso, os mecanismos de atenção podem ser classificados como suaves (*soft attention*) ou rígidos (*hard attention*). A diferença essencial está na forma como o vetor de contexto é criado. No primeiro, o vetor é calculado através de uma ponderação de todos os valores de entrada recebidos pelo mecanismo; no segundo, são utilizadas pontuações para escolher um único dado de entrada, ao invés de se utilizar a ponderação, conforme discutido em (XU et al., 2015) e (QIU et al., 2020).

Em geral, concorda-se que a atenção rígida apresenta desafios adicionais, pois nem a seleção nem a determinação da pontuação máxima são triviais. A função *argmax*, que pode ser empregada na seleção, é não diferenciável. Além disso, o treinamento do modelo geralmente precisa ser mais extenso, em comparação à atenção suave, para se obterem resultados satisfatórios, como também apontado por (XU et al., 2015) e (QIU et al., 2020).

Dado os desafios associados ao mecanismo de atenção rígida, optamos pela utilização do modelo de *soft attention*. Mais especificamente, em vez da metodologia proposta pelo artigo base deste trabalho (QIU et al., 2020), recorreremos à abordagem tradicional de *query*, *key* e *value* para determinação da atenção (VASWANI et al., 2017). As especificidades e equações desses dois métodos da atenção suave são detalhadas a seguir.

### 3.6.4 Atenção Suave com *Score* por Tangente Hiperbólica

Na abordagem proposta por (QIU et al., 2020), adota-se uma transformação linear dos dados de entrada e posterior aplicação da função  $\tanh$  para normalização, conforme ilustrado na Equação 3.21. Considera-se  $[h_1, h_2, \dots, h_T]$  como o vetor de entrada, nesse caso, em virtude das saídas  $h_t$  das unidades LSTM que passarão pelo mecanismo de atenção.

$$s_t = \tanh(W_a \cdot [h_1, h_2, \dots, h_T] + b_a) \quad (3.21)$$

Nessa abordagem, utiliza-se uma matriz de pesos  $W_a$  e um vetor de vies  $b_a$ , ambos treináveis, para inferir o *score*. Como se observará, esta é uma abordagem mais simplificada de cálculo em comparação com a tradicionalmente proposta para mecanismos de atenção, motivo principal pelo qual optamos pela implementação do modelo convencional em detrimento deste (QIU et al., 2020).

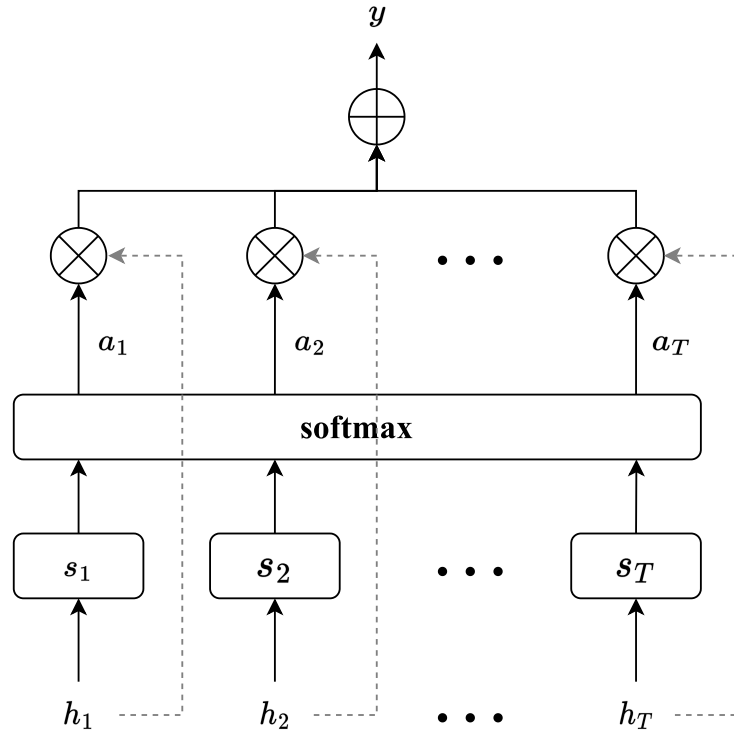


Figura 7: Representação esquemática do mecanismo de atenção do tipo “soft attention”. O diagrama demonstra como as entradas  $(h_1, h_2, \dots, h_T)$  são ponderadas pelos pesos de atenção  $(\alpha_1, \alpha_2, \dots, \alpha_T)$ , calculados com base nos *scores*  $(s_t)$  e normalizados pela função *softmax*, a fim de determinar a saída  $y$ .

Adaptado de (QIU et al., 2020)

Por fim, conforme evidenciado na Figura 7, os *scores* são ponderados por uma função

*softmax*, conforme a Equação 3.22, e empregados diretamente no cálculo do vetor de contexto, que ao ponderar as entradas  $h_t$ , produzirá a saída  $y$ .

$$a_t = \frac{\exp(s_t)}{\sum_{k=1}^T \exp(s_k)} \quad (3.22)$$

### 3.6.5 Atenção Suave com *Query*, *Key* e *Value*

Conforme delineado no estudo seminal (VASWANI et al., 2017), o *score* do mecanismo de atenção é tradicionalmente calculado por meio de uma soma ponderada envolvendo três vetores específicos: *query* (consulta), *key* (chave) e *value* (valor).

No processo, são inicializadas três matrizes —  $W_Q$ ,  $W_K$  e  $W_V$ . Essas matrizes atuam como pesos treináveis na determinação dos respectivos vetores *query* ( $Q$ ), *key* ( $K$ ) e *value* ( $V$ ), conforme ilustram as Equações 3.23, 3.24 e 3.25 (VASWANI et al., 2017).

$$Q = H \cdot W_Q \quad (3.23) \quad K = H \cdot W_K \quad (3.24) \quad V = H \cdot W_V \quad (3.25)$$

Nessas equações, as transformações lineares são aplicadas ao vetor de entrada  $H = [h_1, h_2, \dots, h_T]$ , que, nesse contexto, refere-se às saídas  $h_t$  das unidades LSTM. Uma vez definidas essas matrizes, pode-se calcular o *score*, conforme expresso na Equação 3.26 (VASWANI et al., 2017).

$$s_t = \frac{Q \cdot K^T}{\sqrt{d_k}} \quad (3.26)$$

Em que  $T$  é o tamanho da sequência, ou seja, o tamanho da janela temporal da LSTM (seu *step size*) e  $d_k$  é a dimensão dos vetores de chave, usado essencialmente para fins de escalonamento. A partir disso, é possível normalizar cada *score* por meio de uma função *softmax*, conforme a Equação 3.22. Posteriormente, determina-se o vetor de contexto multiplicando-se os scores normalizados pelo vetor de *value*, resultando na saída  $y$ , conforme a Equação 3.27 (VASWANI et al., 2017).

$$y = \sum_{t=1}^T a_t \cdot V_t \quad (3.27)$$

A Figura 8 ilustra a totalidade desse processo. No mecanismo de atenção, o vetor *query* representa a informação que se deseja decodificar ou analisar. O vetor *key*, por sua vez, atua como um indexador, auxiliando na determinação do grau de relevância de cada

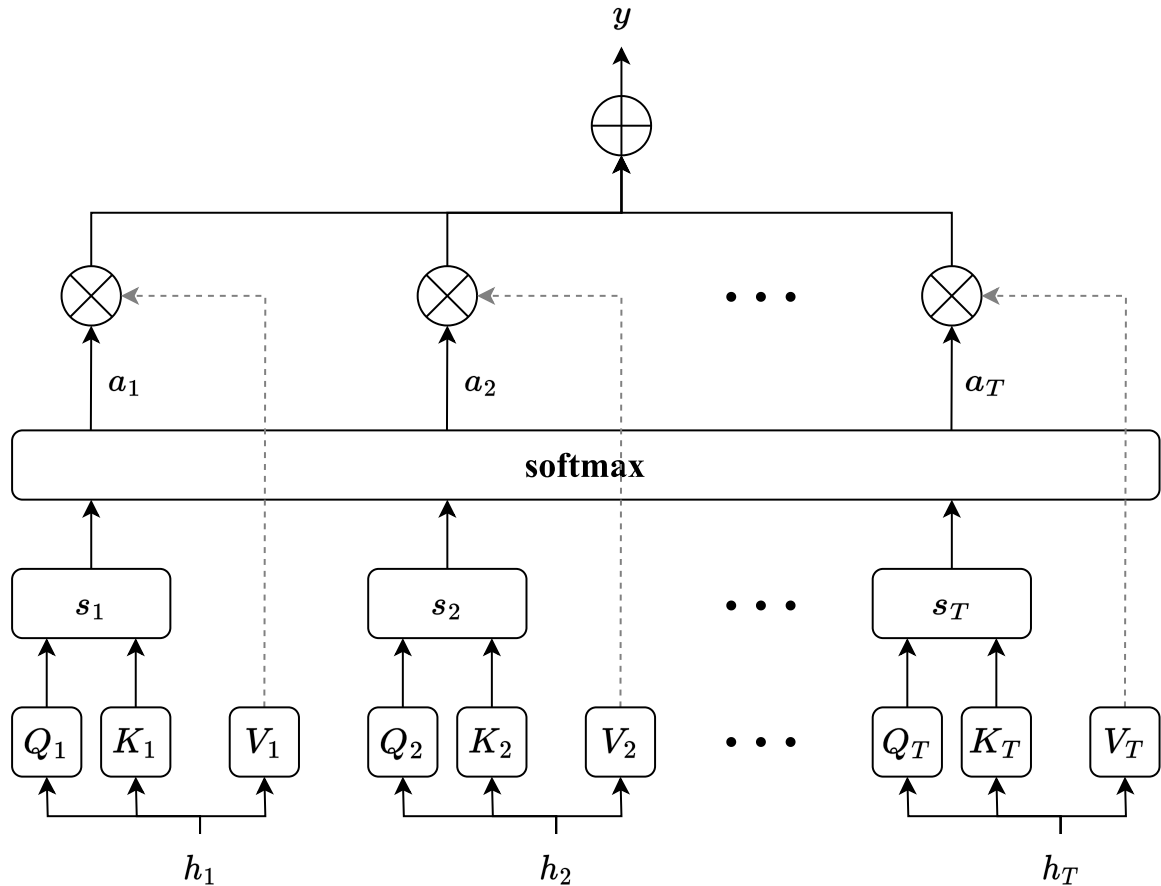


Figura 8: Representação esquemática do mecanismo de atenção do tipo *soft attention*, destacando as componentes *query* ( $Q$ ), *key* ( $K$ ) e *value* ( $V$ ). As entradas ( $h_1, h_2, \dots, h_T$ ) são transformadas nas triplas  $Q, K$  e  $V$ . Os scores ( $s_t$ ) resultantes da interação entre *query* e *key* são normalizados por uma função *softmax* para calcular os pesos de atenção ( $\alpha_1, \alpha_2, \dots, \alpha_T$ ), que influenciam a saída final  $y$  por meio dos vetores de *value*.

Autoria própria

elemento da sequência de entrada em relação à consulta. Isso ajuda a discernir quanta atenção cada parte da entrada deve receber (VASWANI et al., 2017).

O vetor *value* fornece os valores que serão ponderados com base nos *scores* de atenção normalizados. Assim, após avaliar a relevância de partes específicas da entrada, através da comparação entre *query* e *key*, essas informações são usadas para ponderar o vetor *value*, que, por sua vez, influencia a saída do mecanismo de atenção (VASWANI et al., 2017).

### 3.6.6 Considerações Finais

A capacidade de uma rede LSTM tradicional de se ater majoritariamente a sua saída mais recente sempre foi uma limitação para problemas em que etapas intermediárias da

sequência têm relevância significativa. O mecanismo de atenção surge como uma solução para esse desafio, permitindo que o modelo, ao fazer uma predição, considere não apenas a saída final, mas também estados intermediários de uma sequência, ponderando-os de acordo com sua relevância (WANG DONGJIN YU, 2021). Isso é análogo a recordar-se de partes cruciais de uma conversa, em vez de apenas a última frase.

Assim, enquanto a implementação pode variar, seja através do cálculo de *scores* por tangente hiperbólica ou pela abordagem tradicional de *query*, *key* e *value*, o princípio subjacente é o mesmo: dar ao modelo a capacidade de focar seletivamente nas informações mais pertinentes para produzir uma saída mais informada e precisa (WANG DONGJIN YU, 2021).

## 4 S&P 500 E OS MERCADOS DE CAPITAIS

O S&P 500, ou *Standard & Poor's 500*, é um índice de mercado criado em 1957 pela companhia homônima *Standard & Poor's*. Esse índice busca refletir as variações de valor de 500 das maiores corporações negociadas publicamente nas bolsas de valores dos Estados Unidos, abrangendo a *New York Stock Exchange* (NYSE) e a *National Association of Securities Dealers Automated Quotations* (NASDAQ) (CANALES, 2020).

### 4.1 Composição e Relevância Global

O índice foi concebido para ser um indicador abrangente do desempenho do mercado, cobrindo as principais indústrias de múltiplos setores e refletindo a dinâmica econômica do país. Assim, o S&P 500 é um indicador vital da saúde econômica americana e, por extensão, da economia global. Isso se deve, em parte, ao fato de que o capital de muitas dessas companhias sediadas nos EUA provém da atuação de filiais estabelecidas ao redor do mundo (SREENIVASAN et al., 2022).

Diferentemente de outros índices, como o *Dow Jones Industrial Average* (DJIA), que rastreia apenas 30 empresas e é baseado no preço das ações, e o *NASDAQ Composite*, que, apesar de vasto, concentra-se predominantemente em empresas de tecnologia, o S&P 500 abrange uma gama muito mais diversificada de setores (CANALES, 2020).

Além disso, a metodologia empregada na ponderação dos ativos do índice baseia-se na capitalização de mercado ajustada por *free-float*. Isso implica que a influência de cada empresa no índice é proporcional ao seu valor de mercado, que é o produto do número de ações disponíveis para negociação com o preço de mercado dessas ações (DANSO, 2020).

Assim, empresas com maior capitalização de mercado exercem uma influência mais significativa no índice e os movimentos nas ações delas têm um impacto direto no valor geral do S&P 500. Essa dinâmica é um reflexo das relações econômicas nas quais as empresas de maior porte e mais valorizadas têm a capacidade de influenciar as condições

de mercado, as atividades econômicas e, em muitos casos, até as dinâmicas sociais e políticas (OTCHERE; GYGAX, 2007).

Portanto, a capitalização de mercado não é apenas uma métrica para ponderação no índice. Ela também reflete a avaliação do mercado sobre o tamanho, estabilidade e potencial futuro de uma empresa. Empresas com alta capitalização de mercado são frequentemente vistas como mais estáveis e seguras para investimentos. Contudo, essa capitalização também pode ser influenciada por fatores macroeconômicos e especulação do mercado, consequentemente, uma alta capitalização nem sempre indica a saúde financeira da empresa (PAVONE, 2019).

Em virtude disso, para que uma empresa seja elegível para integrar o S&P 500 ela deve cumprir critérios que não se limitam apenas à alta capitalização de mercado, mas também incluem fatores geográficos, como ter sede nos Estados Unidos, e requisitos societários, como ter pelo menos 50% de suas ações disponíveis ao público (OTCHERE; GYGAX, 2007).

Além disso, o S&P 500 é frequentemente utilizado como um *benchmark* por investidores e gestores de portfólio para avaliar o desempenho de seus investimentos. É comum comparar o retorno de uma carteira de ativos com a performance do índice. Nesse cenário, destaca-se a estratégia de investimento passivo, onde os investidores adquirem fundos que replicam o S&P 500, aproveitando-se da diversificação inerente e, geralmente, de custos menores em comparação com fundos de gestão ativa (MOUSSAWI et al., 2022).

Dessa forma, o S&P 500 não é apenas um índice. Ele é uma ferramenta analítica multifacetada que oferece *insights* sobre a saúde do mercado de ações dos EUA; serve como um indicador da economia global; influencia decisões de investimento e políticas econômicas; e auxilia investidores na construção de portfólios diversificados. Assim, compreender sua metodologia e seus movimentos é de valor para investidores, analistas financeiros e formuladores de políticas.

## 5 METODOLOGIA

Nesse capítulo, a metodologia do modelo proposto é detalhada em algumas fases principais: a coleta e tratamento de dados, que envolve a seleção de uma fonte confiável e a limpeza e transformação para garantir a qualidade do treinamento; o desenvolvimento da arquitetura do modelo, usando células LSTM com mecanismos de atenção e a otimização dos hiperparâmetros; a avaliação do modelo por meio de métricas pré-estabelecidas e, por fim, a análise de estratégias de gestão de banca, fundamentais para entender um pouco mais as possíveis aplicações práticas.

### 5.1 Coleta dos Dados

Em relação à aquisição de dados, busca-se obter um conjunto histórico de informações sobre o índice S&P 500, com ênfase em uma ou mais séries temporais desse indicador. Para tanto, foi realizada uma análise de provedores conhecidos de dados de mercado, como Nasdaq Data Link (antigo Quandl), Alpha Vantage, Investing, entre outros.

Ao término dessa análise, assim como fizeram (QIU et al., 2020) e (ZHANG et al., 2019b), optou-se pela Yahoo! Finance como fonte das informações. A decisão baseou-se em diversos fatores como a abrangente cobertura temporal, a gratuidade, a ausência de necessidade de cadastro, a ampla aceitação no mercado e a facilidade de coleta por meio de bibliotecas já implementadas em Python. Assim, serão coletadas seis séries temporais do S&P 500 com frequência diária, a saber:

1. **Preço de abertura:** *open*, corresponde ao primeiro preço pelo qual o ativo é negociado após a abertura da bolsa de valores, influenciado principalmente por eventos ocorridos entre pregões e por ordens no *pré-market*;
2. **Preço de fechamento:** *close*, representando o último preço ao qual o ativo é negociado antes do fechamento da bolsa, essa será também a série predita pelo modelo;



3. **Alta:** *high*, indica o maior preço pelo qual o ativo foi negociado naquele dia de mercado;
4. **Baixa:** *low*, contrariamente ao item anterior, representa o menor preço pelo qual o ativo foi negociado;
5. **Preço de fechamento ajustado:** *adj close*, trata-se de uma métrica que não necessariamente reflete um preço efetivo de negociação do ativo, mas sim uma representação do preço de fechamento que leva em consideração eventos de mercado — distribuição de dividendos ou JCP, *split* ou *inplit*, etc.;
6. **Volume:** *volume*, representando a quantidade total (não em valor financeiro) de ativos transacionados no período, sendo um indicador útil para avaliar a liquidez de um ativo na bolsa.

O intervalo de datas dessa extração de dados se estende desde o primeiro dia útil de janeiro de 1983 até o primeiro dia útil de setembro de 2023, baseando-se no calendário de mercado estadunidense, de forma direta: inicia-se em 03 de janeiro de 1983 e possui término em 01 de setembro de 2023. Um recorte ilustrativo desse *dataset* de mais de 40 anos pode ser visto na Tabela 1.

<b>Data</b>	<b><i>Open</i></b>	<b><i>High</i></b>	<b><i>Low</i></b>	<b><i>Close</i></b>	<b><i>Adj Close</i></b>	<b><i>Volume</i></b>
1983-01-03	140,65	141,33	138,20	138,34	138,34	59.080.000
1983-01-04	138,33	141,36	138,08	141,36	141,36	75.530.000
1983-01-05	141,35	142,60	141,15	141,96	141,96	95.390.000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2023-08-30	4.500,34	4.521,65	4.493,59	4.514,87	4.514,87	3.064.110.000
2023-08-31	4.517,01	4.532,26	4.507,39	4.507,66	4.507,66	3.946.360.000
2023-09-01	4.530,60	4.541,25	4.501,35	4.515,77	4.515,77	3.246.260.000

Tabela 1: Visão geral dos dados históricos utilizados no modelo, abrangendo um intervalo de 10.252 dias úteis entre 1983 e 2023. Cada entrada inclui informações de abertura (*Open*), máxima (*High*), mínima (*Low*), fechamento (*Close*), fechamento ajustado (*Adj Close*) e volume (*Volume*) de negociações para o dia especificado. Note que a série de fechamento servirá tanto como entrada quanto como saída do modelo.

A princípio, o período selecionado assegura volume suficiente de informação, tendo em vista que modelos de aprendizado profundo tradicionalmente requerem dados em abundância. Nas mais de quatro décadas consideradas existiram crises econômicas, momentos de alta e baixa dos preços de ações e, sobretudo, mudanças estruturais do mercado. Portanto, com respaldo no trabalho de (QIU et al., 2020) e (ZHANG et al., 2019b), que

incorporaram, respectivamente, duas décadas e três décadas em seus modelos, avalia-se que o intervalo é extenso o suficiente para permitir o aprendizado adequado da rede.

## 5.2 Pré-Processamento dos Dados

No pré-processamento dos dados, adotar-se-ão técnicas para aprimorar a qualidade desses, tornando-os adequados para alimentação do modelo, os processos aqui descritos foram aplicados indistintamente em todas as séries temporais mencionadas.

### 5.2.1 Redução de Ruído

Inicialmente, de maneira similar ao procedimento adotado por (QIU et al., 2020), visando reduzir o ruído presente nos dados, será aplicada uma transformada wavelet. A família escolhida para a transformada é a Coiflets, em especial, de terceira ordem.

Como mencionado, essa técnica demonstra eficácia na análise de sinais não estacionários, como os preços das ações, desagregando o sinal original em diferentes componentes de frequência. Essa desagregação permite identificar e eliminar ruídos de alta frequência, preservando primordialmente as tendências e padrões subjacentes, essenciais para a previsão de séries temporais (KE et al., 2023).

### 5.2.2 Normalização

Posteriormente ao *denoising*, os dados serão normalizados utilizando a técnica de normalização Z-Score. Este procedimento visa assegurar que todas as variáveis estejam na mesma escala, evitando que algumas dominem sobre outras durante o treinamento devido às disparidades nas magnitudes dos valores. A motivação para essa abordagem decorre da necessidade de lidar com preços na ordem das centenas/milhares e volumes transacionais na ordem dos milhões/bilhões (QIU et al., 2020).

$$X_{normalizado} = \frac{X - \bar{X}}{\sigma_X} \quad (5.1)$$

Conforme apresentado na Equação 5.1, a normalização é efetuada subtraindo-se a média ( $\bar{X}$ ) de cada variável aleatória de seus respectivos valores e, subsequentemente, dividindo pelo desvio padrão dessa ( $\sigma_X$ ). O resultado são dados com média nula e desvio padrão unitário, alinhado ao procedimento adotado por (QIU et al., 2020).

### 5.2.3 Repartição

Para a segmentação dos dados em distintos conjuntos, será adotada a estratégia de particionamento sequencial, mantendo a ordem temporal dos dados, isto é, sem embaralhá-los. O conjunto de dados será dividido na proporção 95/2.5/2.5, sendo que os primeiros 95% dos registros serão utilizados para treinamento, os 2.5% subsequentes para validação, e os 2.5% finais (correspondentes ao período mais recente) para teste. Como referência, os trabalhos de (QIU et al., 2020) e (ZHANG et al., 2019b) utilizaram segmentações, respectivamente, de 99/01 e 65/35, concebendo apenas conjuntos de treino e teste (sem conjunto de validação).

Todavia, é reconhecido que o conjunto de validação é crítico tanto para o ajuste (*tuning*) de hiperparâmetros quanto para o treinamento do modelo, ainda que apenas as respostas (*labels*) do conjunto de treino sejam conhecidas por esse.

Assim, o conjunto de validação é usado para avaliar a performance do modelo durante o treinamento e ajudar a evitar o *overfitting*, esse conjunto permite que façamos ajustes sem comprometer a integridade do teste final. Já o conjunto de teste propriamente dito serve para avaliar a capacidade do modelo de generalizar utilizando informações nunca vistas antes, fornecendo uma estimativa realista do desempenho.

### 5.2.4 Alimentação

Para alimentar o modelo, os dados serão fornecidos por meio do mecanismo de janela deslizante, com um tamanho de janela de 20 dias, o qual constitui o passo temporal (*time step*) do modelo. Essa configuração segue o precedente estabelecido por (ZHANG et al., 2019b), representando aproximadamente um mês de dados, levando em consideração os dias úteis.

Durante esse período, as seis séries temporais distintas — abertura, fechamento, máxima, mínima, fechamento ajustado e volume — serão inseridas no modelo para cada um dos 20 dias. O modelo é então configurado para usar essas sequências de dados com o intuito de prever o valor de fechamento no vigésimo primeiro dia. A metodologia desse processo é ilustrada, para uma única série temporal, na Figura 9.

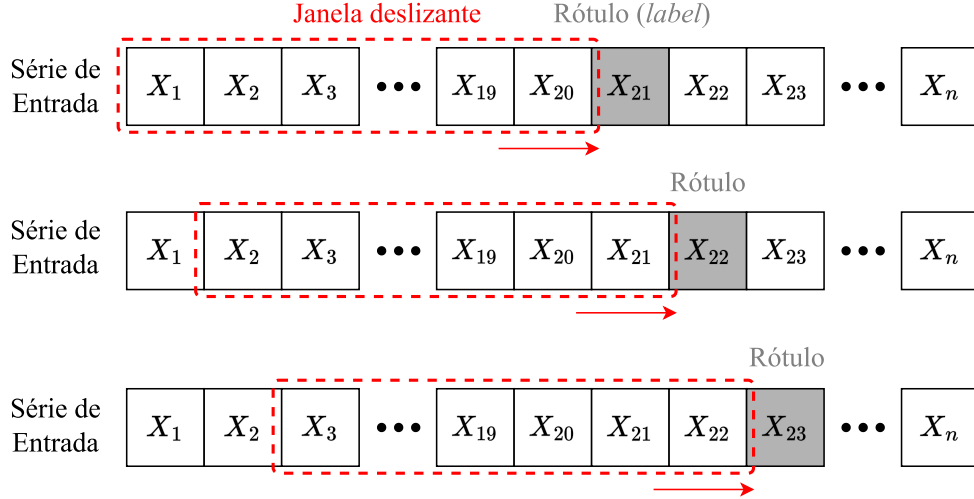


Figura 9: Ilustração do processo de janela deslizante (*sliding window*) para a geração de entradas e rótulos no treinamento de modelos de previsão temporal. As entradas são constituídas por subsequências dos dados  $X_1, X_2, \dots, X_n$ , e os rótulos correspondem ao dado subsequente na série temporal, que é o valor a ser previsto pelo modelo. A janela deslizante avança sequencialmente, um passo por vez, enfatizando a natureza contínua e sobreposta das amostras utilizadas para a previsão.

Adaptado de (BIAN et al., 2021)

## 5.3 Arquitetura do Modelo

O modelo foi desenvolvido em Python, utilizando principalmente a biblioteca TensorFlow. Uma característica notável é a implementação do mecanismo de atenção, destacando-se por não ser uma camada pré-fabricada da biblioteca e, portanto, sendo de desenvolvimento autoral. Além disso, o projeto também incorporou outras bibliotecas largamente utilizadas nos campos da ciência de dados e do aprendizado de máquina, incluindo Pandas, Numpy e Scikit-Learn.

### 5.3.1 Fundamento

A arquitetura base é fortemente inspirada no trabalho de (QIU et al., 2020). O modelo é composto por três camadas principais: a de entrada, responsável pelo recebimento dos dados já tratados; a oculta, formada pela combinação de células LSTM e do mecanismo de atenção; e a de saída, que disponibiliza os resultados ao usuário por meio de um único neurônio com ativação linear, podendo ser precedido por um *dropout* para reduzir a possibilidade de *overfitting*, cujo valor inicialmente assumido é de 20%. O número de *epochs* para o treinamento foi inicialmente definido em 1000.

### 5.3.2 Otimização por *Grid Search*

O *grid search* é uma técnica amplamente difundida no aprendizado de máquina para otimização de hiperparâmetros. Trata-se de um método que examina exaustivamente todas as combinações possíveis dos valores de hiperparâmetros especificados, com o objetivo de encontrar aqueles que maximizam a performance do modelo.

Nesta fase do projeto, optou-se pelo *grid search* para obter uma perspectiva abrangente do que seria ideal para a rede, tendo em vista a intenção de explorar variações em torno das proposições do artigo base (QIU et al., 2020). Pois, embora diretrizes pré-existentes possam fornecer um ponto de partida sólido, é crucial também experimentar variações para garantir que o modelo seja adequadamente ajustado ao contexto específico do problema em questão.

Assim, após definir a arquitetura fundamental, implementou-se o algoritmo ajustando: o tamanho do estado oculto, isto é, o tamanho da memória das células na camada LSTM, com os valores 10, 20, 50 e 100; a taxa de aprendizado, testando os valores 0,001, 0,01 e 0,1; e o tamanho do lote, com os valores 64, 128, 256 e 512 testados.

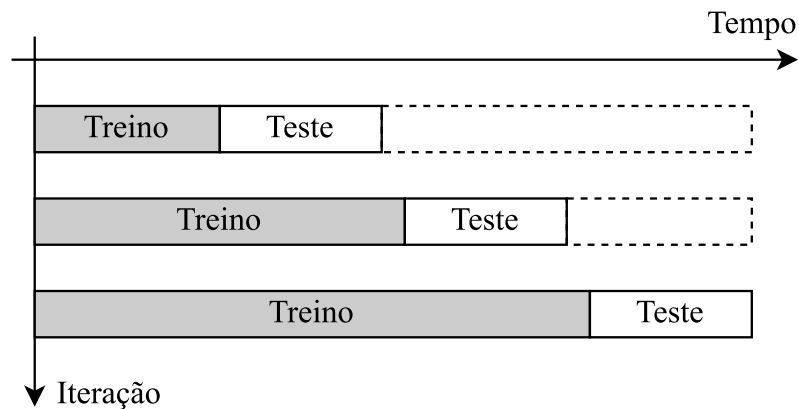


Figura 10: Representação esquemática do método de validação cruzada do tipo *k-fold* específico para séries temporais, com  $k = 3$ . Os conjuntos de treino e teste são segmentados sequencialmente em cada iteração. Esta abordagem otimiza o uso dos dados e assegura que a sequencialidade seja mantida.

Adaptado de (ASSAAD; FAYEK, 2021)

Destarte, um total de  $4 \times 3 \times 4 = 48$  configurações de rede foram avaliadas em busca da melhor arquitetura para o modelo. Além disso, os testes incorporaram o método de validação cruzada usando *k*-dobras para séries temporais, definindo-se  $k = 3$ . O processo é ilustrado na Figura 10, resultando em  $3 \times 48 = 144$  execuções. As métricas avaliativas são calculadas a cada uma das três iterações para se obter uma pontuação final que indique

o modelo mais eficaz — baseado na minimização do erro quadrático médio.

Dessa forma, a Tabela 2 reúne todos os resultados obtidos dos testes, classificando-os do melhor para o pior. A grande amplitude observada nas pontuações reflete a uma sensibilidade significativa do desempenho do modelo em relação à seleção dos hiperparâmetros. As configurações que compreendem diferentes valores para o tamanho do lote (BS), o tamanho do estado oculto (HSS) e a taxa de aprendizado (LR) demonstram que certas combinações podem resultar em uma performance substancialmente sub-ótima, conforme evidenciado pelas pontuações fortemente negativas.

# <sup>a</sup>	BS <sup>b</sup>	HSS <sup>c</sup>	LR <sup>d</sup>	Pontuação <sup>e</sup>
1	256	20	0,001	-0,0499
2	128	10	0,001	-0,0529
3	128	50	0,001	-0,0631
4	512	50	0,001	-0,0905
5	512	10	0,001	-0,1484
6	512	20	0,001	-0,1533
7	512	20	0,01	-0,1548
8	64	20	0,001	-0,1659
9	512	100	0,001	-0,1703
10	128	20	0,001	-0,2208
11	64	50	0,001	-0,2658
12	64	10	0,001	-0,2667
13	256	50	0,001	-0,3082
14	256	10	0,01	-0,3552
15	512	100	0,01	-0,5176
16	256	10	0,001	-0,5200
17	128	100	0,001	-0,5321
18	256	100	0,001	-0,5667
19	512	10	0,01	-0,7388
20	256	50	0,01	-0,7499
21	512	50	0,01	-0,9814
22	64	100	0,001	-1,4447
23	256	20	0,01	-1,9792
24	512	10	0,1	-4,4971

# <sup>a</sup>	BS <sup>b</sup>	HSS <sup>c</sup>	LR <sup>d</sup>	Pontuação <sup>e</sup>
25	256	100	0,01	-11,2637
26	512	20	0,1	-14,7155
27	256	20	0,1	-23,2697
28	128	50	0,01	-35,5459
29	256	100	0,1	-36,6156
30	128	10	0,01	-54,9971
31	128	100	0,01	-68,0543
32	512	50	0,1	-227,2002
33	512	100	0,1	-240,5054
34	256	10	0,1	-360,5586
35	128	20	0,01	-425,2916
36	64	100	0,1	-574,9194
37	64	10	0,01	-3475,3889
38	64	20	0,01	-7532,8302
39	128	100	0,1	-9017,5375
40	128	50	0,1	-9655,7233
41	128	20	0,1	-69278,5288
42	64	100	0,01	-118243,0020
43	64	10	0,1	-178437,9370
44	64	50	0,01	-474686,6320
45	128	10	0,1	-730496,4640
46	256	50	0,1	-1172530,2300
47	64	50	0,1	-30000486,3000
48	64	20	0,1	-83241145,2000

Tabela 2: Resultados ordenados dos testes de *grid search* para a otimização inicial dos hiperparâmetros.

<sup>a</sup> Classificação do modelo

<sup>b</sup> Tamanho de lote (*batch size*)

<sup>c</sup> Tamanho do estado oculto (*hidden state size*)

<sup>d</sup> Taxa de aprendizado (*learning rate*)

<sup>e</sup> Média do oposto do MSE das iterações

Há uma tendência discernível associando incrementos na taxa de aprendizado com uma degradação na performance, sugerindo que taxas elevadas de aprendizado podem ocasionar perda de convergência. Da mesma forma, extremos no tamanho do lote e no tamanho do estado oculto parecem ser prejudiciais, implicando a existência de valores ótimos para esses parâmetros a fim de maximizar a eficácia do modelo.

A incidência de pontuações altamente negativas levanta a possibilidade de problemas de convergência dentro do modelo ou a necessidade de delimitar mais estreitamente o espaço de busca dos hiperparâmetros para áreas mais promissoras. Além disso, algumas configurações de hiperparâmetros podem ter levado a instabilidades no aprendizado ou à explosão de gradientes, especialmente em contextos de taxas de aprendizado elevadas, como mencionado.

Em conclusão, a análise desses resultados fornece percepções significativas quanto à influência dos hiperparâmetros no desempenho do modelo, estabelecendo um fundamento para as iterações subsequentes de otimização.

### 5.3.3 Otimização por *Bayesian Search*

Com base ainda nessa premissa de otimização de hiperparâmetros, torna-se imperativo considerar as configurações que exibiram as mais altas métricas de desempenho para uma análise ainda mais aprofundada. Em vista disso, um exame das combinações de hiperparâmetros que resultaram nas pontuações menos negativas (maiores performances) é a etapa lógica subsequente. Tal análise não só valida a eficácia dos valores atuais, mas também serve como ponto de partida para investigar variações próximas a esses parâmetros. Assim, com a intenção de refinar ainda mais o modelo, a busca por um conjunto ótimo de hiperparâmetros agora será conduzida por meio de um *bayesian search*.

No contexto de otimização de hiperparâmetros, a busca bayesiana surge como uma alternativa mais sofisticada e eficiente quando comparada a métodos como o *grid search*. Essa técnica baseia-se em princípios estatísticos do Teorema de Bayes e busca construir um modelo probabilístico que mapeia os hiperparâmetros aos resultados esperados de desempenho. Ao invés de testar combinações aleatórias, a busca bayesiana utiliza as informações dos testes anteriores para prever quais conjuntos de hiperparâmetros têm a maior probabilidade de produzir melhores resultados, otimizando assim o processo de busca e reduzindo o número de iterações necessárias.

A eficácia do método bayesiano reside em sua habilidade de balancear a exploração



de novos espaços de hiperparâmetros com a exploração de áreas conhecidas que já demonstraram resultados promissores, em inglês, essa prática é conhecida como *exploration-exploitation trade-off*. Esse processo iterativo é particularmente útil em situações com um custo computacional elevado, pois visa maximizar a eficiência ao refinar o espaço de busca com base na probabilidade de melhoria, restringindo um espaço amostral inicial à medida que se aprende sobre ele, ao invés de explorá-lo como um todo e arcar com o custo de inúmeras soluções pouco promissoras.

Destarte, levando em conta a estratégia delineada, a próxima fase de experimentação focará em hiperparâmetros específicos para teste: o tamanho do lote (*batch size*), com valores estipulados em 64, 128, 256, 512 e 1024; a taxa de aprendizado (*learning rate*), que será restringida e variará continuamente dentro do intervalo de 0.0001 até 0.01; e a taxa de desativação (*dropout rate*), que será explorada dentro de um espectro contínuo de 0% a 30%.

Como um passo final na delimitação do espaço de hiperparâmetros, o tamanho do estado oculto (*hidden state size*) será fixado em 20, decisão esta fundamentada por duas razões cruciais: primeiro, a análise preliminar do *grid search* revelou que 5 dos 10 modelos mais eficientes possuíam esse valor, incluindo o de melhor desempenho; segundo, tal configuração harmoniza-se com a entrada de dados de 20 dias úteis que serão fornecidos ao modelo para projeções futuras.

Finalmente, a busca será feita ao longo de 100 iterações distintas, ou seja, em cada uma dessas iterações um modelo diferente será testado e então validado através do método de validação cruzada com  $k$ -dobras, análogo ao empregado durante o *grid search* — Figura 10. Semelhantemente, toma-se  $k = 3$ ; espera-se, portanto, um total de 300 execuções, garantindo uma avaliação abrangente e detalhada do espaço de hiperparâmetros. Essa estratégia intensiva de teste é projetada para maximizar as chances de identificar a combinação ótima de parâmetros, levando a uma melhoria da performance do modelo.

Tendo isso em vista, a Tabela 3 concentra os 50 melhores modelos. Em conformidade com o que foi observado no *grid search*, as taxas de aprendizado e agora também as taxas de desativação são relativamente baixas, indicando, possivelmente, um modelo mais estável, ou seja, que não requer tanta regularização (indicado por DR mais baixo) ou ajustes agressivos durante o aprendizado (indicado por LR mais baixo).

Entre esses modelos, há uma certa variedade no tamanho de lote, mas nota-se que existe uma dominância, especialmente entre os 10 melhores colocados, de tamanhos de lote menores, entre 256 e, especialmente, 128.

Já na Tabela 4 encontram-se os 50 modelos com piores desempenhos. Novamente, a tendência é para taxas de aprendizado mais altas e taxas de desativação extremas — tanto muito baixas quanto muito altas.

Em conclusão, os resultados obtidos por meio da busca bayesiana para a otimização de hiperparâmetros do modelo complementam e reforçam as observações anteriores. A análise dos melhores e piores desempenhos sugere que as estratégias mais eficazes para o ajuste fino dos hiperparâmetros se concentram em taxas de aprendizado mais conservadoras e taxas de desativação dentro de um espectro moderado.

# <sup>a</sup>	BS <sup>b</sup>	DR <sup>c</sup>	LR <sup>d</sup>	Pontuação <sup>e</sup>
1	128	0,12241	0,00018	-0,0175
2	128	0,12271	0,00033	-0,0179
3	128	0,12285	0,00033	-0,0196
4	128	0,12198	0,00013	-0,0219
5	256	0,30000	0,00047	-0,0229
6	1024	0,13285	0,00120	-0,0230
7	256	0,29494	0,00128	-0,0272
8	256	0,12912	0,00027	-0,0307
9	128	0,12291	0,00032	-0,0338
10	128	0,11679	0,00010	-0,0377
11	128	0,12506	0,00029	-0,0435
12	128	0,02473	0,00026	-0,0456
13	128	0,12324	0,00021	-0,0623
14	128	0,02832	0,00026	-0,0637
15	128	0,12290	0,00033	-0,0680
16	128	0,12511	0,00029	-0,0706
17	128	0,12267	0,00033	-0,0823
18	128	0,12553	0,00036	-0,0852
19	64	0,12512	0,00029	-0,0857
20	128	0,12169	0,00010	-0,0867
21	256	0,12291	0,00017	-0,0959
22	128	0,12287	0,00032	-0,0990
23	128	0,12509	0,00029	-0,1121
24	128	0,12512	0,00027	-0,1148
25	128	0,12508	0,00030	-0,1159

# <sup>a</sup>	BS <sup>b</sup>	DR <sup>c</sup>	LR <sup>d</sup>	Pontuação <sup>e</sup>
26	128	0,12484	0,00011	-0,1164
27	128	0,12282	0,00032	-0,1205
28	64	0,30000	0,00010	-0,1230
29	128	0,29559	0,00145	-0,1260
30	256	0,02285	0,00017	-0,1288
31	256	0,30000	0,00127	-0,1292
32	256	0,12322	0,00017	-0,1358
33	1024	0,30000	0,00269	-0,1384
34	256	0,30000	0,00010	-0,1394
35	128	0,02207	0,00027	-0,1420
36	256	0,30000	0,00136	-0,1479
37	128	0,12267	0,00032	-0,1505
38	128	0,12258	0,00032	-0,1552
39	128	0,12512	0,00029	-0,1602
40	128	0,12501	0,00027	-0,1677
41	128	0,12287	0,00032	-0,1744
42	256	0,12512	0,00029	-0,1783
43	128	0,12275	0,00033	-0,1840
44	128	0,12512	0,00029	-0,1853
45	64	0,09527	0,00031	-0,1881
46	64	0,12515	0,00027	-0,2224
47	128	0,12510	0,00029	-0,2632
48	1024	0,00029	0,00946	-0,2692
49	128	0,12509	0,00030	-0,3067
50	1024	0,30000	0,00043	-0,3081

<sup>a</sup> Classificação do modelo

<sup>b</sup> Tamanho de lote (*batch size*)

<sup>c</sup> Taxa de desativação (*dropout rate*)

<sup>d</sup> Taxa de aprendizado (*learning rate*)

<sup>e</sup> Média do oposto do MSE das iterações

Tabela 3: Dois primeiros quartis de resultados ordenados dos testes de *bayesian search* para a otimização final dos hiperparâmetros.

# <sup>a</sup>	BS <sup>b</sup>	DR <sup>c</sup>	LR <sup>d</sup>	Pontuação <sup>e</sup>
51	1024	0,11675	0,00012	-0,3408
52	1024	0,00000	0,01000	-0,3467
53	128	0,29782	0,00144	-0,3608
54	128	0,12515	0,00029	-0,3782
55	128	0,12467	0,00034	-0,3967
56	128	0,12246	0,00033	-0,4127
57	128	0,12513	0,00028	-0,4172
58	128	0,12508	0,00029	-0,4479
59	1024	0,22102	0,00102	-0,4818
60	128	0,12280	0,00033	-0,5031
61	128	0,12283	0,00032	-0,5340
62	256	0,02706	0,00467	-0,7334
63	128	0,12495	0,00031	-0,7571
64	1024	0,02400	0,00012	-1,0682
65	1024	0,03576	0,00016	-1,0757
66	64	0,03236	0,00144	-1,4186
67	128	0,30000	0,00152	-1,6855
68	128	0,30000	0,00140	-1,7320
69	128	0,29761	0,00144	-1,9349
70	64	0,00000	0,00029	-3,4956
71	64	0,02911	0,00120	-3,8066
72	128	0,29383	0,00146	-4,4769
73	256	0,03429	0,00433	-6,8047
74	64	0,01873	0,00158	-7,6722
75	256	0,04301	0,00454	-8,7194

# <sup>a</sup>	BS <sup>b</sup>	DR <sup>c</sup>	LR <sup>d</sup>	Pontuação <sup>e</sup>
76	256	0,30000	0,00547	-10,9343
77	256	0,29953	0,00697	-12,3735
78	128	0,29770	0,00145	-19,3648
79	64	0,03467	0,00163	-19,7170
80	256	0,29888	0,00505	-27,9392
81	256	0,04726	0,00403	-32,4116
82	256	0,30000	0,00669	-34,4340
83	128	0,15543	0,00260	-48,3288
84	64	0,05368	0,00188	-49,2205
85	256	0,30000	0,00559	-51,5116
86	256	0,29944	0,00687	-57,1890
87	128	0,15336	0,00267	-65,9337
88	256	0,29820	0,00846	-70,8277
89	256	0,30000	0,00668	-71,4774
90	128	0,15609	0,00258	-98,0372
91	256	0,29959	0,00885	-101,8230
92	256	0,30000	0,00800	-138,6651
93	128	0,10993	0,00312	-225,9962
94	256	0,30000	0,00716	-234,8195
95	256	0,30000	0,00719	-282,5871
96	128	0,13218	0,00346	-337,6577
97	256	0,27938	0,00946	-907,1448
98	64	0,15290	0,00268	-1242,2719
99	128	0,12151	0,01000	-9897,5787
100	128	0,29841	0,00861	-31067,1

<sup>a</sup> Classificação do modelo

<sup>b</sup> Tamanho de lote (*batch size*)

<sup>c</sup> Taxa de desativação (*dropout rate*)

<sup>d</sup> Taxa de aprendizado (*learning rate*)

<sup>e</sup> Média do oposto do MSE das iterações

Tabela 4: Dois últimos quartis de resultados ordenados dos testes de *bayesian search* para a otimização final dos hiperparâmetros.

### 5.3.4 Análise da Curva de Aprendizado

As metodologias de *grid search* e *bayesian search*, previamente discutidas, são amplamente reconhecidas por sua eficácia no ajuste de uma diversidade de hiperparâmetros em modelos de aprendizado de máquina. Contudo, é imperativo destacar que o número de épocas (*epochs*) de treinamento frequentemente requer um ajuste mais nuanciado, o qual pode ser alcançado mediante uma análise da curva de aprendizado.

A avaliação da curva de aprendizado traz benefícios notáveis. Em primeiro lugar, ela faculta uma análise visual contínua do progresso do treinamento, elucidando o padrão de convergência do modelo. Essa visualização não apenas apoia decisões acerca da continuação do treinamento, mas também fornece percepções críticas sobre a dinâmica de aprendizado do modelo.

Nesse contexto, torna-se possível a identificação de um ponto ótimo de treinamento. Tal ponto é caracterizado pelo número de *epochs* além do qual o modelo cessa de apresentar avanços significativos em termos de minimização do erro. A observação desse ponto permite evitar tanto o uso excessivo de recursos computacionais quanto a incidência de *overfitting*, garantindo uma otimização eficiente.

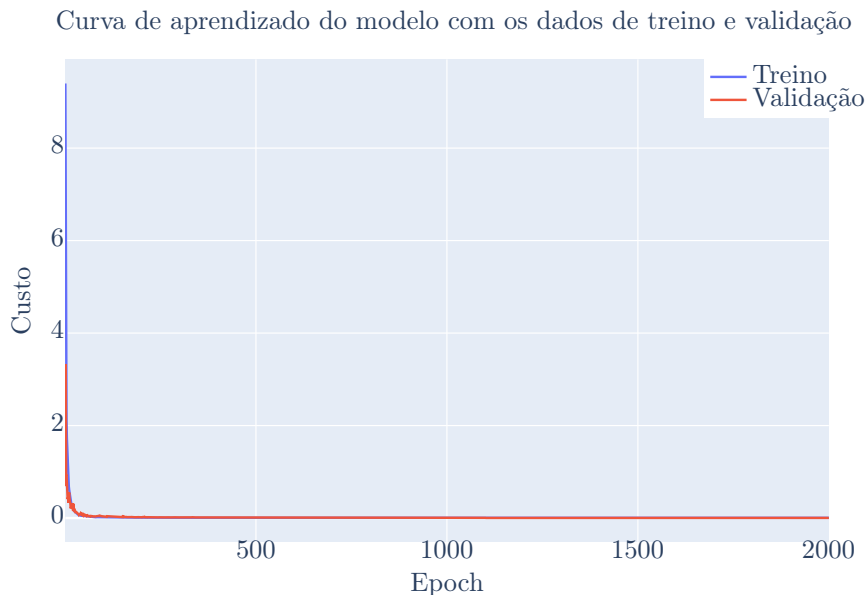


Figura 11: Curva de aprendizado do modelo ao longo de 2000 épocas. A curva azul representa o custo de treinamento, enquanto a curva vermelha indica o custo de validação. Ambas as curvas mostram uma rápida diminuição inicial, refletindo um rápido aprendizado com uma aparente estabilização já na época 500, aproximadamente.

Autoria própria

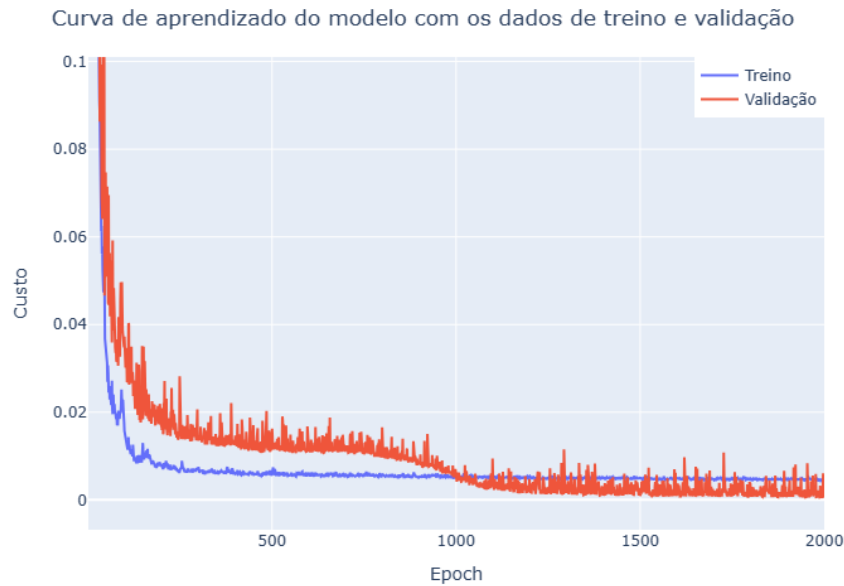


Figura 12: Detalhe ampliado da curva de aprendizado também para as 2000 *epochs*. Nota-se que a curva de custo de treinamento (em azul) de fato se estabiliza rapidamente, sem decréscimos acentuados a partir da época 500, aproximadamente. Todavia o custo de validação (em vermelho) revela um claro salto de performance ao redor da época 1000.

Autoria própria

Especificamente em relação ao modelo em questão, a decisão de modificar o número inicial de treinamento de 1000 para 2000 *epochs* foi fundamentada por essas análises. Tal decisão foi motivada pela observação de que, durante as primeiras 1000 épocas, o modelo ainda demonstrava sinais de aprimoramento contínuo, sem indícios de estagnação ou deterioração no desempenho. Isso indica que o modelo estava ainda em uma fase de beneficiar-se do treinamento adicional, sugerindo que um aumento no número de épocas poderia resultar em melhorias na precisão e na capacidade de generalização.

Assim, como mostra a Figura 11 que representa a curva de aprendizado completa do modelo ao longo de 2000 épocas, observamos que tanto o custo de treino (em azul) quanto o custo de validação (em vermelho) diminuem significativamente nas primeiras épocas, o que é comum em modelos de aprendizado profundo, e depois tendem a se estabilizar. O comportamento das curvas sugere que o modelo rapidamente aprende a generalizar a partir dos dados, mas sendo uma ferramenta visual, a escala pode fornecer uma falsa ideia dos nuances.

Na Figura 12 o detalhe das curvas com recorte em custos inferiores a 0.1, destaca um salto da curva de validação no entorno da época 1000. Esse salto, é de extrema valia para o desempenho do modelo, pois representa um momento de inflexão no treinamento, no qual

a rede, que até então poderia estar começando a apresentar sinais de convergência, recebe uma nova oportunidade de minimizar o custo, justificando mais uma vez a importância do uso de um conjunto de dados de validação.

### 5.3.5 Concepção Final

Frente ao proposto, pode-se afirmar que é possível alcançar agora uma construção mais informada e sólida da arquitetura final do modelo. A consistência dessas conclusões com os resultados precedentes da busca em *grid* e da busca bayesiana leva à consolidação do modelo finalmente utilizado e descrito a seguir.

A arquitetura proposta para o modelo apresenta então um tamanho de estado oculto de 20, em harmonia com a janela de 20 dias que o modelo é projetado para analisar. Assim, espera-se que essas 20 células LSTM utilizadas sejam capazes de auxiliar a rede a capturar padrões relevantes dentro do intervalo de tempo estipulado.

Realizada essa etapa, a saída de cada célula LSTM é repassada para o mecanismo de atenção. Esse adota, como discutido, o modelo de *soft attention*, que funciona ponderando as saídas de cada uma das células LSTM com a finalidade de atribuir maior ou menor “atenção” a elas. Como já mencionado, esse processo busca reconhecer que as saídas de células intermediárias podem ser tão importantes quanto a da última célula dessa camada e, portanto, merecem receber um determinado nível de atenção. Finalmente, essas saídas ponderadas são somadas para produzir um único vetor de contexto, que representa a saída da camada de atenção.

Por fim, tem-se a última camada da rede, que compreende uma camada densa com um único neurônio de ativação linear (para regressão) e que recebe o vetor de contexto após esse ser submetido a um *dropout* com taxa de 12,241%, resultando então na saída da rede.

Salienta-se ainda que esse processo ocorre por 2000 *epochs* para um tamanho de lote de 128 e a rede é treinada utilizando o algoritmo de otimização baseado em gradiente, denominado ADAM, com uma taxa de aprendizado de 0,00018, parâmetros todos em conformidade com o que foi encontrado na etapa de busca bayesiana e de análise da curva de aprendizado.

Assim, a Figura 13 fornece uma diagramação de alto nível da rede utilizada. Ademais, nos anexos é possível verificar uma listagem dos principais códigos desenvolvidos para a implementação do modelo. Por fim, está disponível no repositório do GitHub

gvmossato/alstm-stock-market a totalidade das implementações.

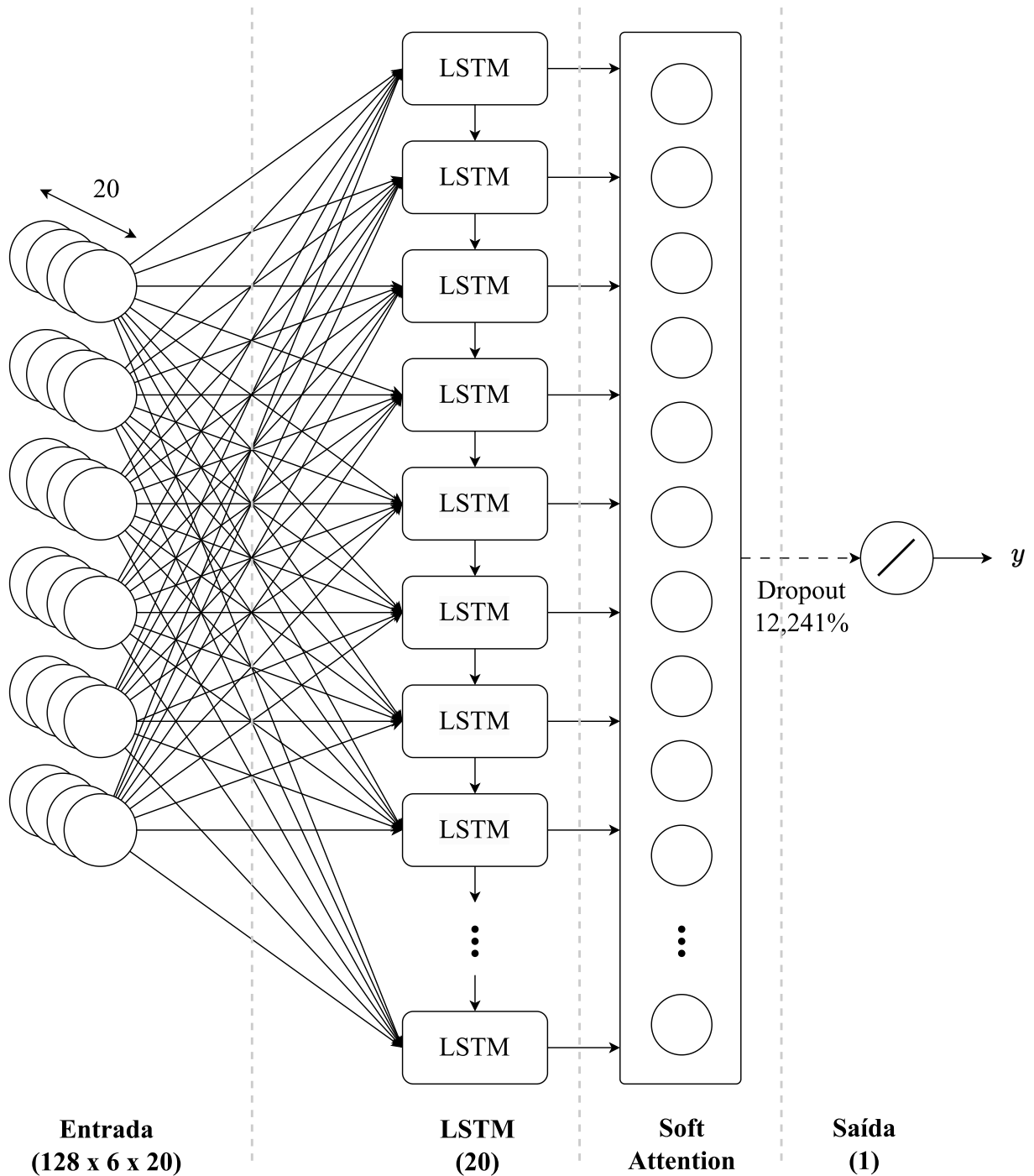


Figura 13: Representação esquemática da arquitetura do modelo de aprendizado profundo implementado, consistindo em células LSTM empilhadas seguidas de um mecanismo de *soft attention*, convergindo para uma saída única ( $y$ ). A entrada é um tensor tridimensional (128 x 6 x 20), representando o tamanho do lote (omitido da representação para clareza), o número de *features* e a janela de tempo, respectivamente. Uma taxa de *dropout* de 12,241% é incluída para mitigar a ocorrência de *overfitting*.



## 5.4 Avaliação do Modelo

A eficácia do modelo será analisada sob múltiplas perspectivas para assegurar uma compreensão abrangente de seu desempenho. Utilizaremos gráficos comparativos que contrastam as previsões do modelo com os valores reais, oferecendo uma visualização intuitiva do quão próximo as previsões estão da realidade. Seguiremos com a aplicação de métricas quantitativas específicas, incluindo o RMSE, o MAE e o coeficiente de determinação  $R^2$ , conforme o *benchmark* estabelecido por (QIU et al., 2020), assim, essas métricas são fundamentais para considerar o sucesso ou não do modelo.

Adicionalmente, examinaremos o retorno acumulado e verificaremos o comportamento das previsões para tendências de alta e baixa, afim de avaliar a aplicabilidade prática do modelo no mercado financeiro. Além disso, será dada uma ênfase na capacidade que a atenção possui de influenciar os resultados do modelo. Por fim, realizaremos um exercício teórico de gestão de banca, visando medir não apenas a validade técnica, mas também a viabilidade financeira da solução proposta. Esse conjunto diversificado de métodos de avaliação visa fornecer uma análise holística e mais rigorosa do desempenho da rede.

### 5.4.1 Métricas

Quantificação do desvio das previsões, mensurando os resíduos, para ser possível estabelecer uma comparação direta entre o modelo implementado e o *benchmark* definido por (QIU et al., 2020).

#### 5.4.1.1 MSE

O erro quadrático médio, Equação 5.2, é uma métrica de erro amplamente usada em tarefas de regressão. Ele é calculado como a média dos quadrados dos erros de previsão, essa métrica dá maior peso aos erros maiores, tornando-a particularmente útil quando grandes erros são indesejáveis e aumentando a sensibilidade aos *outliers*. Essa métrica foi tomada como a função de custo do modelo, não sendo utilizada diretamente na comparação com o *benchmark* em virtude de sua semelhança ao RMSE.

$$MSE = \frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2 \quad (5.2)$$

#### 5.4.1.2 RMSE

A raiz do erro quadrático médio, representada na Equação 5.3, é obtida pela raiz quadrada do MSE, o que busca ajustar a escala desse para a unidade original das medidas. Isso o torna uma métrica mais facilmente interpretável, especialmente quando comparada com a magnitude das variáveis dependentes.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2} \quad (5.3)$$

#### 5.4.1.3 MAE

O erro absoluto médio, Equação 5.4, representa outra métrica de erro largamente utilizada. É calculado como a média dos valores absolutos dos erros de previsão. O MAE é menos sensível a grandes erros do que o MSE tornando-o uma medida útil de erro médio.

$$MAE = \frac{1}{N} \sum_{i=1}^N |Y_i - \hat{Y}_i| \quad (5.4)$$

#### 5.4.1.4 $R^2$

O coeficiente de determinação, ou  $R^2$ , Equação 5.5, é uma métrica que indica a proporção da variância na variável dependente que é previsível a partir das variáveis independentes. Em outras palavras, o  $R^2$  indica o quão bem as variáveis independentes explicam a variabilidade da variável dependente. Um valor de  $R^2$  mais próximo de 1 indica um ajuste mais fiel do modelo aos dados.

$$R^2 = 1 - \frac{\sum_{i=1}^N (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^N (Y_i - \bar{Y})^2} \quad (5.5)$$

#### 5.4.1.5 TE

Finalmente, o *tracking error*, Equação 5.6, é uma medida da divergência entre o desempenho do modelo e o desempenho do índice S&P 500. Ele é comumente usado na gestão de portfólio para avaliar o quão bem um fundo (por exemplo, um *Exchange-Traded Fund*) segue seu *benchmark*. Essa é a única métrica não utilizada por (QIU et al., 2020),

mas decidiu-se incluí-la na avaliação do modelo, uma vez que essa mesma possui ampla adoção mercadológica na avaliação da aderência de ETFs a seus índices de referência.

$$TE = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (R_{p,i} - R_{b,i})^2} \quad (5.6)$$

#### 5.4.1.6 Considerações Finais

Ao utilizar as métricas RMSE, MAE,  $R^2$  e TE em conjunto, obtemos uma visão mais ampla e, sobretudo, multifacetada do desempenho do modelo. Como mencionado, cada uma dessas métricas traz suas percepções únicas: o RMSE enfatizando erros maiores, o MAE oferecendo uma perspectiva sobre o erro médio, o  $R^2$  avaliando a proporção da variância explicada pelo modelo e o TE medindo a aderência com a realidade. Essa abordagem abrangente é crucial para evitar o viés que pode surgir ao se confiar em uma única métrica de avaliação.

Portanto, o desempenho do modelo será considerado totalmente bem-sucedido se as métricas de erro RMSE, MAE e o coeficiente  $R^2$  forem iguais ou superiores às do estudo (QIU et al., 2020). Essa metodologia de avaliação relativa permite não apenas uma comparação direta com o modelo de referência, mas também oferece uma compreensão contextual sobre a eficácia do modelo em replicar ou superar *benchmarks* estabelecidos.

Assim, acredita-se que a comparação relativa é fundamental, pois ela situa o desempenho do modelo dentro de um quadro de referências conhecidas, facilitando a avaliação de sua adequação para aplicações práticas.

No mais, nas equações apresentadas, adotou-se que:

- $N$  é o número total de observações;
- $Y_i$  valor real para a  $i$ -ésima observação;
- $\hat{Y}_i$  valor predito para a  $i$ -ésima observação;
- $\bar{Y}$  média dos valores reais;
- $R_{p,i}$  é o retorno do portfólio (nesse caso, da predição) em  $i$ ;
- $R_{b,i}$  é o retorno do *benchmark* (nesse caso, do S&P 500) em  $i$ .

Para o retorno foi utilizada a formulação tradicional que considera a variação relativa ao último preço observado, isto é,  $R_i = (Y_i - Y_{i-1}) / Y_{i-1}$ .

### 5.4.2 Retorno Acumulado e Tendências

O retorno acumulado (Equação 5.7) é uma métrica fundamental no campo das finanças, pois reflete o ganho ou perda total obtido ao longo do tempo, sendo essencial para avaliar a eficácia de estratégias de investimento. Nesse contexto, investigaremos como o modelo se comporta em termos de geração de valor ao longo de um período definido. Essa análise não apenas fornecerá uma visão clara sobre a proximidade do modelo com o índice, mas também ajudará a estabelecer uma relevância mais prática para investidores e analistas financeiros.

$$\prod_{i=1}^N (1 + R_i) - 1 \quad (5.7)$$

Em seguida, voltaremos nossa atenção para a análise de tendências, focando especificamente nos movimentos de subida e queda indicados pelas previsões do modelo. A compreensão desses movimentos de tendência será a base para os testes subsequentes de viabilidade financeira do modelo, permitindo-nos uma avaliação um pouco mais prática de sua utilidade.

### 5.4.3 Influência da Atenção

Demos destaque especial ao mecanismo de atenção em nosso modelo LSTM pois o consideramos um elemento distintivo desse trabalho, assim, para quantificar o impacto desse mecanismo no contexto de previsões financeiras, realizaremos alguns testes comparativos.

Serão trabalhados mecanismos conforme a implementação da atenção clássica, conforme descrito na literatura de (VASWANI et al., 2017), a atenção utilizada no próprio (QIU et al., 2020) e uma versão do nosso modelo sem qualquer mecanismo de atenção.

Através desse estudo buscamos tanto validar a importância (ou não) da atenção em nosso modelo, quanto conseguir uma compreensão mais ampla de sua aplicação em modelos de previsão financeira.

### 5.4.4 Estratégias de Gestão de Banca

Entendemos que a eficácia financeira de uma estratégia de investimento não se restringe unicamente à acurácia de previsões de mercado ou a minimização de métricas frente a um *benchmark*, mas também à metodologia empregada na alocação e gestão de recursos ao longo do tempo.

Assim, distintas estratégias de gestão de banca podem influenciar de maneira significativa os resultados obtidos, particularmente em mercados de alta volatilidade. Pois então, nesse segmento, será discutida uma análise de algumas estratégias de gestão de banca, ponderando suas características intrínsecas e avaliando seu potencial impacto financeiro consoante o modelo desenvolvido.

#### 5.4.4.1 Martingale

A técnica de Martingale é uma estratégia que propõe o aumento progressivo das apostas após cada resultado desfavorável. A ideia central é que uma série contínua de resultados negativos será eventualmente contrabalanceada por um resultado positivo, acarretando recuperação das perdas antecedentes e algum ganho equivalente.

Matematicamente, considere  $B_n$  como o valor apostado no  $n$ -ésimo período e  $X_n$  o resultado desse período — no qual 1 representa um ganho e  $-1$  uma perda. Então, a metodologia de Martingale pode ser matematicamente expressa como na Equação 5.8.

$$B_{n+1} = \begin{cases} B_0 & \text{se } X_n = 1, \\ 2B_n & \text{se } X_n = -1. \end{cases} \quad (5.8)$$

#### 5.4.4.2 Paroli

Em contraste com a estratégia Martingale, a abordagem Paroli — também conhecida como Martingale Reverso ou Anti-Martingale —, sugere a duplicação da aposta após cada ganho. Esse método busca aproveitar uma sequência de resultados favoráveis, maximizando os ganhos, e minimizar as perdas durante períodos de resultados adversos.

Adotando notação idêntica à anterior, a estratégia Paroli é formalmente descrita consoante a Equação 5.9.

$$B_{n+1} = \begin{cases} 2B_n & \text{se } X_n = 1, \\ B_0 & \text{se } X_n = -1. \end{cases} \quad (5.9)$$

#### 5.4.4.3 D'Alembert

A estratégia D'Alembert é baseada na ideia de aumentar ou diminuir a aposta por uma unidade a cada rodada, dependendo do resultado anterior. Especificamente, após uma perda, a aposta é aumentada em uma unidade e após um ganho, é reduzida em uma unidade.

Definamos  $L_n$  como o total acumulado de perdas até o  $n$ -ésimo período. O valor da aposta para o próximo período,  $B_{n+1}$ , é então definido pela Equação 5.10.

$$B_{n+1} = B_0 + (L_n - G_n) \quad (5.10)$$

Na qual  $G_n$  representa o total acumulado de ganhos até o  $n$ -ésimo período, refletindo a natureza de compensação entre ganhos e perdas da estratégia.

#### 5.4.4.4 Apostas Fixas

A estratégia de apostas fixas adota um valor constante para cada aposta, independentemente dos resultados anteriores. Essa abordagem conservadora tem como objetivo principal a minimização de riscos, mantendo a estabilidade no montante investido a cada operação. Assim, em qualquer período, a aposta é invariavelmente determinada através da Equação 5.11.

$$B_{n+1} = B_0 \quad (5.11)$$

#### 5.4.4.5 Apostas Proporcionais

Na estratégia de apostas proporcionais, o investidor aplica uma fração fixa do seu capital total em cada aposta. Assim, o valor apostado varia proporcionalmente ao montante total disponível: aumenta com o crescimento do capital e diminui se o capital total se reduz.

Definindo  $M_n$  como o montante total disponível no  $n$ -ésimo período e  $\alpha$  como a fração

a ser aplicada, o valor da aposta é calculado pela Equação 5.12

$$B_{n+1} = \alpha \cdot M_n \text{ com } 0 < \alpha \leq 1 \quad (5.12)$$

#### 5.4.4.6 Considerações Finais

Como mencionado, a motivação principal dessa análise é identificar uma gestão de banca ótima que possa aumentar os rendimentos gerados pelo modelo de previsão proposto. Fundamentamos nossa análise na convicção de que a metodologia de previsão para o índice S&P 500 supera a mera especulação aleatória, tipicamente caracterizada por uma probabilidade de sucesso de 50%.

Sob essa premissa, é essencial explorar o impacto de variadas técnicas de gestão de banca nos resultados finais. Se o algoritmo realmente demonstra uma precisão acima de 50%, então deve existir uma abordagem de gestão de banca que, aliada ao nosso modelo de previsão, possa otimizar os lucros e mitigar os riscos. A correta identificação e implementação dessa combinação pode transformar a estratégia de investimentos diários no índice S&P 500, por exemplo, através de operações de *day-trade*, em um método com valor prático para investidores.

## 6 RESULTADOS

Este capítulo está organizado em três seções principais. A primeira seção é dedicada às análises técnicas, focando em avaliar a capacidade do modelo em reproduzir o comportamento do índice S&P 500 e sua performance frente ao *benchmark*. A segunda seção ocupa-se de entender como os mecanismos de atenção podem influenciar as previsões do modelo. A terceira seção, por fim, explora a aplicabilidade prática do modelo no contexto financeiro, testando e avaliando estratégias financeiras baseadas nas previsões do modelo.

### 6.1 Performance Técnica do Modelo

Nesta seção, são analisados e discutidos os seguintes aspectos: o impacto da redução de ruído na série temporal original do índice S&P 500; a acurácia das previsões do modelo em relação ao comportamento esperado do índice, utilizando as métricas mencionadas anteriormente; análises sobre a direção e o retorno acumulado do índice; e, por fim, a influência do mecanismo de atenção no desempenho geral da rede neural.

#### 6.1.1 Efeito da Redução de Ruído

A Figura 14 apresenta em comparação um recorte da série temporal de preço de fechamento antes e após o processo de remoção do ruído, pela transformada wavelet. Nota-se que essa possui, de fato, comportamento mais suave, com atenuação forte das faixas de alta frequência. Isso pois optou-se pela anulação parcial das “componentes de ruído” (coeficientes de detalhamento), uma vez que remover a alta frequência por completo é tirar a volatilidade de um mercado que historicamente é bastante afetado por ela.

O gráfico ilustra claramente a redução de flutuações extremas, o que pode ser atribuído à remoção eficaz de variações de curto prazo que são comumente consideradas como ruído. Contudo, a tendência geral e os movimentos significativos do mercado permanecem intactos, sugerindo que a técnica de *denoising* preservou as características essenciais dos



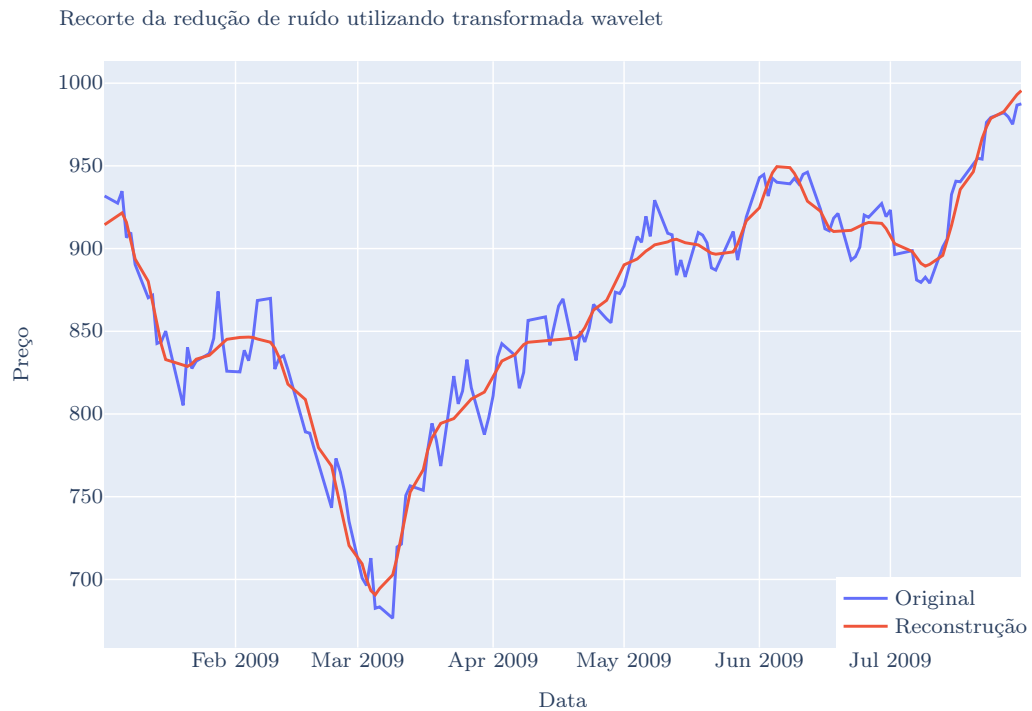


Figura 14: Efeito da aplicação de *denoising* utilizando transformada wavelet na série temporal de fechamento do índice S&P 500. A suavização resultante (em vermelho) é evidente, principalmente na atenuação de flutuações de curto prazo, mantendo as tendências gerais de mercado. A transformada wavelet permite uma análise mais clara das tendências subjacentes ao reduzir a volatilidade aparente sem descartar a volatilidade inerente ao mercado.

Autoria própria

dados enquanto minimizava as distorções transitórias. Esse equilíbrio entre a remoção de ruído e a retenção de movimentos de mercado importantes é crucial para análises subsequentes e processos de tomada de decisão baseados no modelo.

Além disso, ao analisar as áreas onde o preço de fechamento original exibe picos e vales acentuados, observa-se que a série *denoised* tende a suavizar esses pontos, apresentando um perfil mais nivelado. Essa característica é particularmente benéfica para modelos preditivos que podem ser excessivamente influenciados por variações atípicas, que não necessariamente representam o comportamento subjacente do mercado.

### 6.1.2 Desempenho da Predição Frente ao Índice

Elencam-se nessa seção a comparação direta das séries temporais original e prevista, com a quantificação direta dos resultados produzidos pelo modelo.

### 6.1.2.1 Comparação das Séries Temporais

Primeiro, ao analisar a Figura 15 que exibe o resultado da predição nos dados de treino, observa-se que o modelo demonstra uma capacidade notável de seguir a tendência dos dados reais. A congruência entre as linhas de preço real e predição é evidente pela sobreposição significativa das duas séries temporais ao longo do período considerado.

Inicialmente, há de fato alguma discrepância mais acentuada entre as séries, o que pode ser atribuído a um período do índice cujos os preços encontravam-se bastante descontados em relação ao que se tornaria o usual. Felizmente, esse desajuste inicial não perdura e o modelo se alinha com os valores reais satisfatoriamente, indicando uma adaptação bastante rápida às características do mercado.

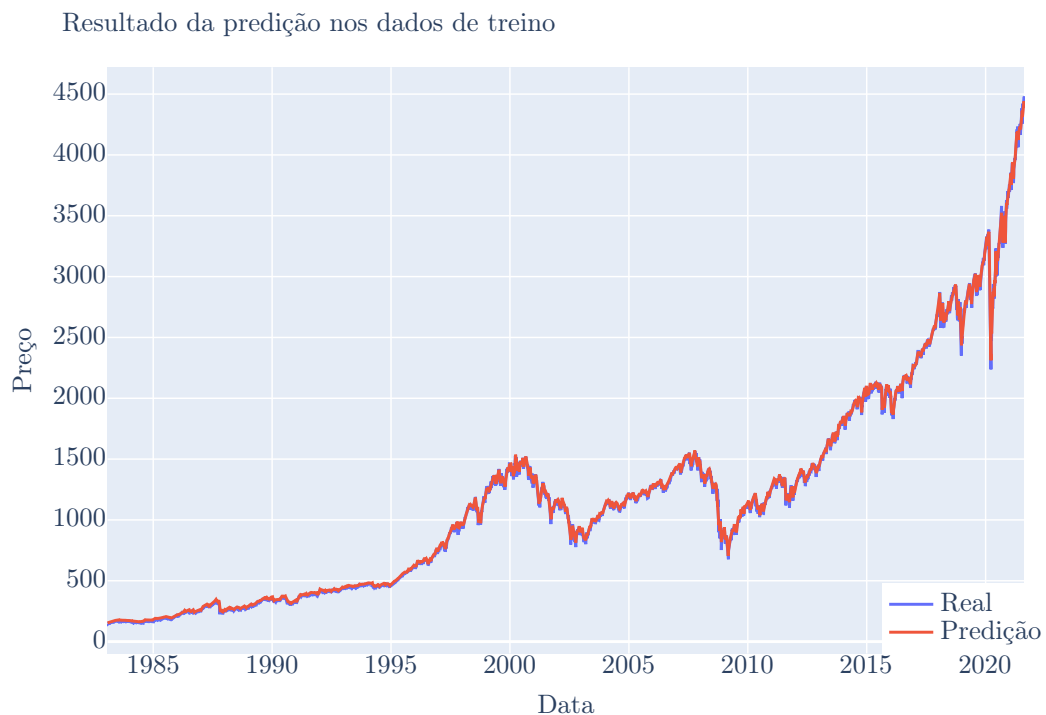


Figura 15: Desempenho do modelo nos dados de treino para o fechamento do índice S&P 500. A linha azul representa os preços reais de fechamento, enquanto a linha vermelha representa os preços previstos pelo modelo. A sobreposição significativa das duas séries indica uma forte aderência do modelo aos dados de treino, praticamente descartando a hipótese de *underfitting*.

Autoria própria

Ao longo de períodos mais estáveis e com tendências melhor definidas, o modelo captura com eficácia tanto a direção quanto a velocidade da mudança nos preços, sugerindo uma compreensão razoável das dinâmicas de mercado incorporadas nos dados de treino.

A aderência a esse conjunto de dados é grande um indicativo de que o modelo não sofre com *underfitting*.

Por fim, ao analisar picos e vales que sucedem os momentos de alta volatilidade ou de troca de tendência, cujas previsões são intrinsecamente mais desafiadoras, o modelo ainda assim mostra um alinhamento plausível com a série real, embora sejam visíveis alguns pontos de descolamento, mas que não comprometem a integridade geral da previsão.

Já ao observar a Figura 16, que exhibe o resultado da previsão nos dados de validação, nota-se a capacidade do modelo de generalizar seu aprendizado para além do conjunto de treino. A proximidade entre as linhas do preço real e da previsão demonstram que o modelo conseguiu capturar as características essenciais do índice S&P 500, que são fundamentais para realizar previsões acuradas em dados não vistos anteriormente.

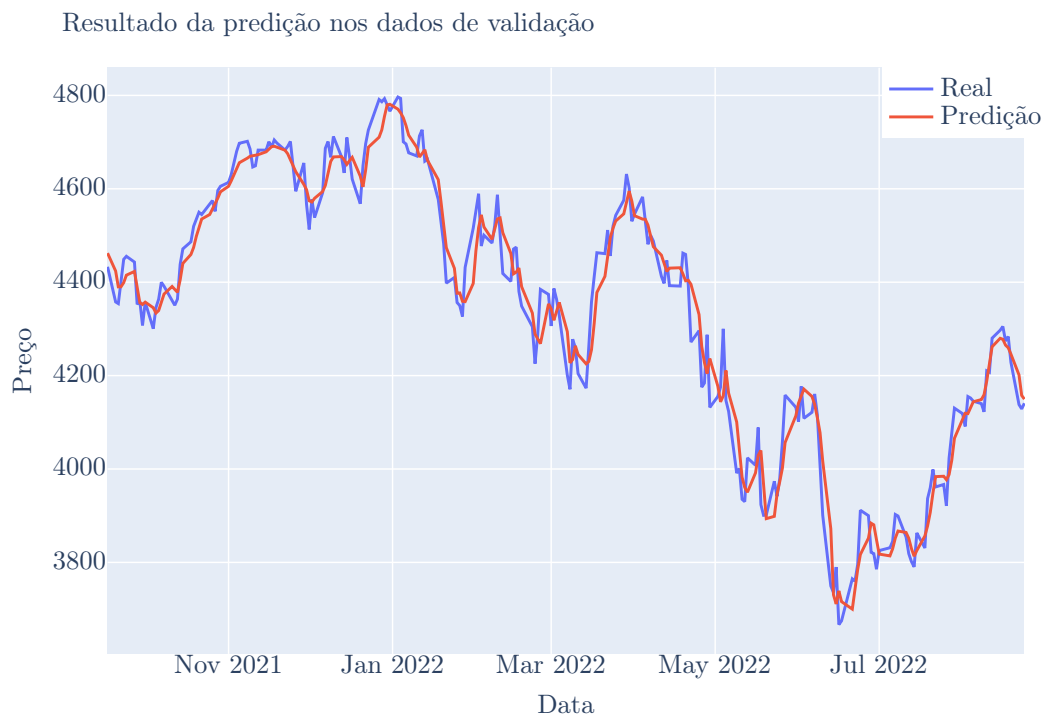


Figura 16: Resultados do modelo no conjunto de validação, evidenciando a capacidade de generalização com dados cujas respostas não são conhecidas. As flutuações e tendências do mercado são bem capturadas e exibem os efeitos da otimização de hiperparâmetros.

Autoria própria

É notável que, mesmo em um conjunto de dados de validação, o modelo consegue refletir as flutuações do mercado com uma fidelidade considerável. Esse comportamento é um indicativo de que o modelo pode ser eficaz também em dados de teste, que representam o verdadeiro teste de desempenho de um modelo preditivo.

Por fim, na Figura 17 está o resultado da predição nos dados de teste, que é fundamental para avaliar a eficácia final do modelo preditivo, pois ilustra o comportamento desse em condições semelhantes àsquelas que enfrentaria em aplicações do mundo real. A linha de predição acompanha de perto a série real do preço de fechamento do índice S&P 500, refletindo uma forte capacidade do modelo de prever a tendência subjacente do mercado.

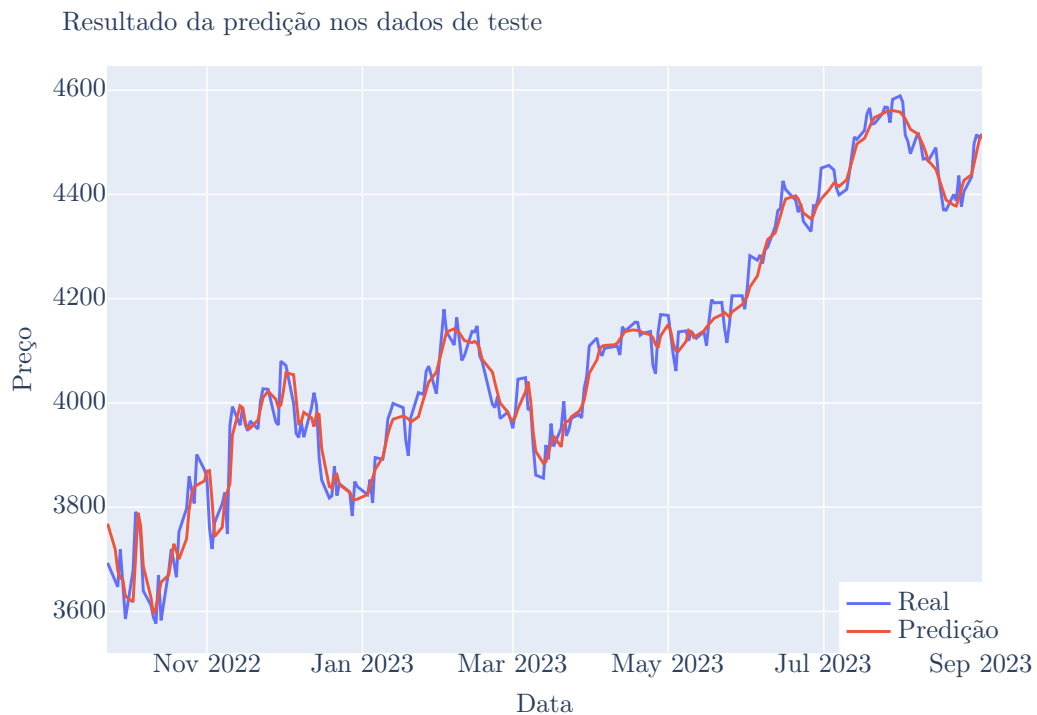


Figura 17: Avaliação do modelo nos dados de teste, que representam o cenário mais próximo de uma aplicação do mundo real. A correspondência entre as séries temporais destaca a habilidade do modelo em prever a direção do mercado, mesmo sem ter tido exposição prévia a esses dados. O conjunto em questão é o temporalmente mais recente temporalmente e foi mantido isolado de qualquer fase anterior do processo de modelagem.

Autoria própria

O desempenho do modelo nos dados de teste é bastante satisfatório, especialmente quando se considera a ausência de grandes defasagens temporais entre a série predita e os dados reais. Essa precisão temporal é crucial em aplicações financeiras, nas quais um *lag* significativo poderia reduzir a utilidade prática das previsões. Além disso, o fato de que as previsões estão em sincronia com os preços reais confirma a robustez do modelo e sugere que o processamento de *denoising* não comprometeu a integridade das previsões temporais.

Isso é relevante pois indica que o modelo não apenas aprendeu a tendência geral,

mas também preservou uma sensibilidade às mudanças de mercado, mesmo que filtradas para remover ruídos, assim, a atenuação da volatilidade não impede o modelo de seguir a trajetória dos preços.

Em resumo, a análise qualitativa e visual dos resultados indicam um desempenho satisfatório do modelo, fornecendo indícios de que é um instrumento promissor para prever preços de fechamento do S&P 500, com aplicações potenciais diversas.

#### 6.1.2.2 Comparação de Métricas Frente ao *Benchmark*

Conforme mencionado ao longo desse trabalho, tomou-se como base e métrica comparativa o proposto por (QIU et al., 2020), todavia, como compila a Tabela 5 a comparação direta das métricas de erro entre o modelo *benchmark* e modelo aqui proposto não é trivial. Devido a discrepâncias significativas nos conjuntos de teste em termos de complexidade, período temporal, escala e amplitude de preços, não é justo inferir diretamente a diferença; haja vista que modelo proposto foi testado em um conjunto mais amplo e volátil, em que os preços apresentam maior amplitude e média.

	<b>Benchmark</b>	<b>Modelo</b>
<b>Data mínima (teste)</b>	2019-05-17	2022-09-23
<b>Data máxima (teste)</b>	2019-07-01	2023-09-01
<b>Observações</b>	31	237
<b>Preço Mínimo</b>	2751,53	3577,03
<b>Preço Máximo</b>	2971,41	4588,96
<b>Amplitude de Preços</b>	219,88	1011,93
<b>Preço Médio</b>	2870,27	4095,33
<b>Volatilidade Anualizada</b>	11,28%	17,58%

Tabela 5: Comparação dos conjuntos de teste para ambos os modelos, destacando diferenças que não devem ser negligenciadas em uma comparação de desempenho.

Assim, o que a Tabela 6 mostra é não só o que ambos os modelos alcançaram em termos de RMSE, MAE e  $R^2$ , mas também a normalização das métricas de erro tanto pelo preço médio quanto pela amplitude de preços.

Posto isso, com exceção do  $R^2$  que já representa uma medida normalizada, buscou-se fornecer uma base mais justa para comparação dividindo-se o RMSE e o MAE de cada modelo por, respectivamente, o preço médio e a amplitude de preços do conjunto de teste.

Destarte, apesar de nosso modelo ter melhor ajustado um R-quadrado, sugerindo uma capacidade ótima de explicar a variação dos preços do S&P 500, ele tem um desempenho

Métricas		
	Benchmark	Modelo
RMSE	0,3475	19,5238
MAE	0,1935	13,9011
$R^2$	0,8783	0,9940

Normalização pelo Preço Médio		
	Benchmark	Modelo
RMSE	0,00012107	0,00476734
MAE	0,00006742	0,00339437

Normalização pela Amplitude de Preços		
	Benchmark	Modelo
RMSE	0,00158041	0,01929363
MAE	0,00088003	0,01373717

Tabela 6: Métricas para ambos os modelos, com comparação após normalização para minimização dos efeitos das discrepâncias entre os conjuntos de teste dos modelos.

marcadamente inferior em termos de RMSE e MAE, mesmo após a normalização.

Essa observação é pertinente pois um  $R^2$  superior de nosso modelo pode ser indicativo de que ele é capaz de capturar uma proporção maior da variação dos preços em relação ao *benchmark*, no entanto, os valores elevados de RMSE e MAE, mesmo após normalização, sugerem que o modelo pode estar produzindo previsões precisas em termos de tendência, mas não em termos de valores específicos.

Em suma, tem-se que embora o novo modelo mostre resultados bastante promissores comparativamente às séries temporais reais do S&P 500, não se pode afirmar que ele superou o *benchmark* — um dos objetivos centrais do presente trabalho. Essa descoberta demarca um ponto inicial para futuras iterações, buscando um melhor equilíbrio entre a capacidade de explicar a variação dos preços e a precisão das previsões pontuais, não invalidando, ou muito menos descartando, a aplicabilidade desse modelo.

### 6.1.2.3 Comparação Binária da Tendência: Subida e Queda

Aborda-se agora uma metodologia simplificada para avaliar a capacidade preditiva do modelo em termos de direção de movimento do índice S&P 500. Esta análise binária de tendência categoriza os retornos diários em duas classes distintas: ascensão, caso haja um aumento e declínio, se houver uma queda no valor do índice — para manter a consis-

tência, retornos nulos são considerados como ascensão, mas não ocorrem para o período considerado. Tal simplificação permite uma visão mais intuitiva do desempenho preditivo do modelo, focando na sua habilidade de detectar a direção correta da mudança de preço, independentemente da magnitude.

Para contrapor as previsões do modelo com os dados reais foi utilizado o conjunto de testes, além disso, empregamos uma estrutura semelhante à de uma matriz de confusão, comum em tarefas de classificação. Essencialmente, a matriz de confusão aqui adaptada registra a frequência de verdadeiros positivos (subidas corretamente previstas), verdadeiros negativos (quedas corretamente previstas), falsos positivos (subidas incorretamente previstas como quedas), e falsos negativos (quedas incorretamente previstas como subidas).

É crucial enfatizar que, apesar de o modelo não ter sido especificamente treinado para prever a direção do movimento como uma classificação binária, a capacidade de fazer tais previsões emerge como uma característica intrínseca desse, uma vez que ele aprendeu a capturar as dinâmicas subjacentes do índice. A aplicação desta técnica binária é um meio de inspecionar o comportamento do modelo sob uma lente diferente, sem a necessidade de reestruturação ou re-treinamento do mesmo.

Salienta-se ainda que, em um cenário binário, uma taxa de acerto superior a 50% pode ser indicativa de que o modelo possui uma capacidade preditiva melhor do que a aleatoriedade, de tal forma que taxas de acerto abaixo desse limiar seriam não melhores do que uma escolha ao acaso. Todavia, é importante antes considerar também a distribuição inicial das direções.

Isto é, se a distribuição dos retornos diários do índice S&P 500 for desequilibrada, isso pode influenciar significativamente a interpretação do desempenho do modelo. Por exemplo, se historicamente o índice apresentou uma tendência de subida em 60% dos dias, um modelo que preveja constantemente uma ascensão, independentemente das entradas, alcançará uma precisão de 60% sem qualquer aprendizado genuíno.

É verdade que, uma vez constada a aderência entre séries temporais como visto anteriormente, essa hipótese perde força, mas ainda assim, nesse contexto, o modelo será considerado válido se ele distinguir-se significativamente da taxa de acerto esperada com base na distribuição predominante dos dados.

A distribuição binária das tendências de retorno, como mostra a Figura 18, revela que o modelo conseguiu espelhar de maneira muito próxima a distribuição real dos dados, com uma pequena diferença entre os números reais e previstos. Esse é um sinal extremamente

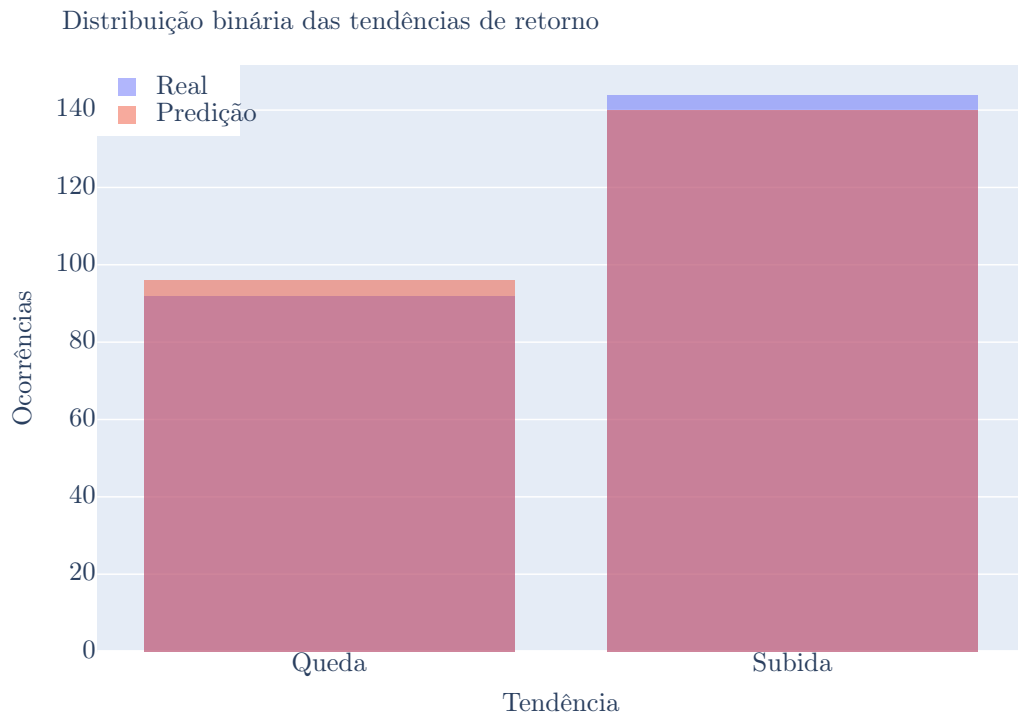


Figura 18: Sobreposição entre as ocorrências reais e as previstas de movimentos de ascensão (subida) e declínio (queda) no índice. A barra azul representa a frequência real, enquanto a barra em vermelho mostra as previsões do modelo. A proximidade entre as alturas das barras para cada tendência indica a acurácia do modelo em replicar a distribuição dos resultados reais.

Autoria própria

promissor de que o modelo não está apenas aprendendo a replicar a distribuição geral dos dados, mas está também discernindo os nuances dentro desses para fazer previsões informadas.

Já observando a matriz de confusão adaptada, conforme a Figura 19 é possível afirmar que o modelo demonstrou uma competência notável em identificar corretamente os dias em que o índice apresentou ascensão, com uma porcentagem substancial de verdadeiros positivos, o que demonstra uma boa capacidade de captar as tendências de subida do mercado.

Ao examinar os falsos positivos e negativos, ou seja, previsões equivocadas de tendência, notamos um número reduzido, mas ainda assim não insignificante: somando os dois quadrantes em vermelho temos aproximadamente dois meses (em dias úteis) de erros. Por tais previsões estarem “diluídas” ao longo das corretas o efeito desse fenômeno é mitigado, mas para todos os efeitos, trata-se de um ponto que requer cautela.

Todavia, é crucial mencionar que, ainda assim, o modelo obteve uma taxa de acerto



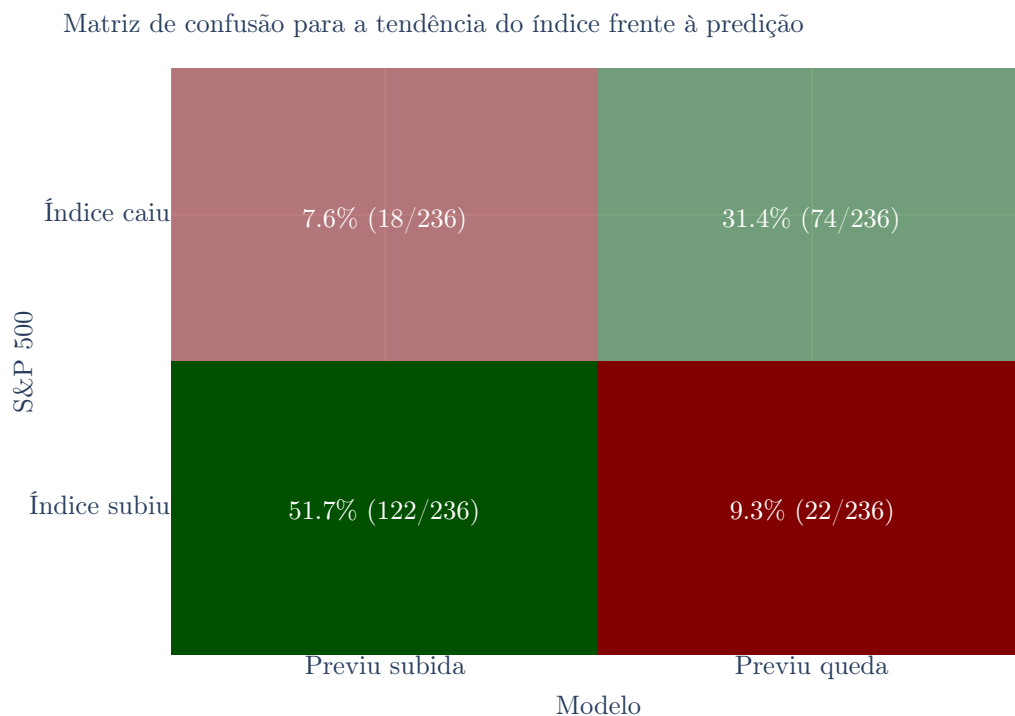


Figura 19: Adaptação de uma matriz de confusão para a tendência binária do índice S&P 500 frente às previsões do modelo. A matriz ilustra em verde a capacidade do modelo de prever corretamente as tendências de alta (subida) e baixa (queda) do índice. Em contrapartida, destacam-se em tons de vermelho as previsões falsas. Os valores representam o percentual de acerto ou erro e entre parênteses o número absoluto de ocorrências sobre o total de dias.

Autoria própria

total superior aos 80%, o que excede a expectativa de um modelo que faz escolhas aleatórias, sugerindo que há aprendizado genuíno ocorrendo.

Por fim, a validade do modelo não se resume apenas em superar os 50% na sua taxa de acerto, mas também na sua capacidade de superar a distribuição inicial predominante dos dados. Assim, o modelo demonstra uma competência na previsão da direção do movimento do índice S&P 500, o que é mais um indicativo de sua utilidade prática, ainda que demarque também um futuro ponto de partida para novas iterações e refinamentos.

#### 6.1.2.4 Comparação do Retorno Total Acumulado

Ao avaliar modelos preditivos financeiros, a aderência do retorno acumulado da previsão ao retorno acumulado real é uma métrica essencial. Assim, o principal objetivo nessa etapa não é superar o mercado, mas refletir sua performance com precisão. Isso é crítico porque um modelo eficaz para investidores passivos deve emular o comportamento

do índice de referência, permitindo estratégias que sigam o mercado de forma confiável.

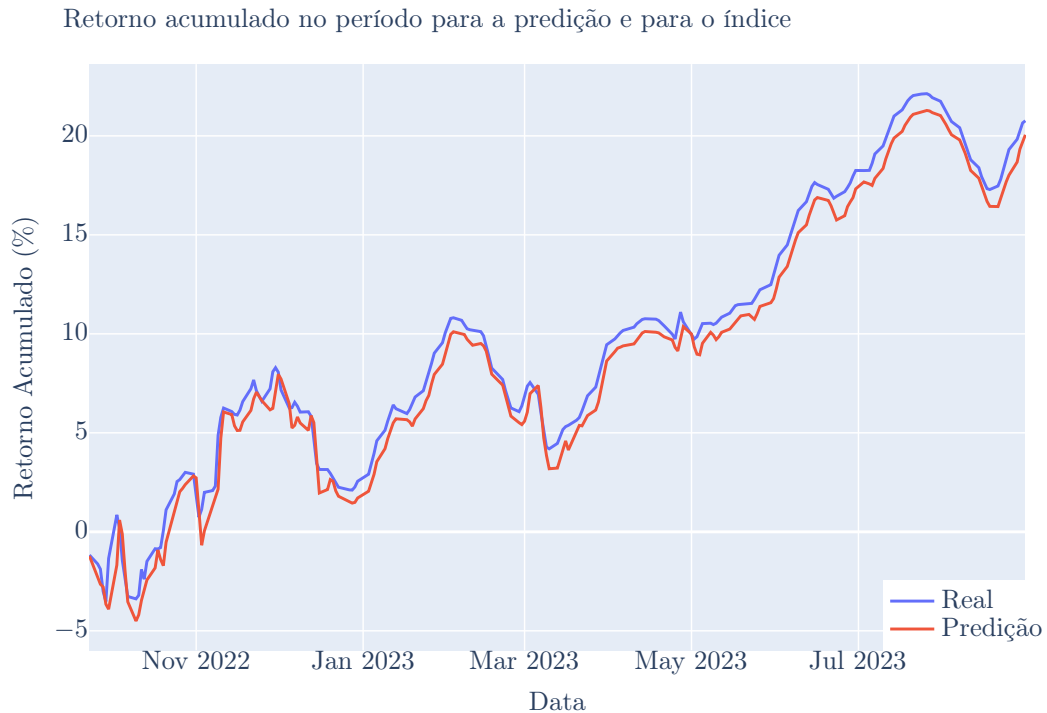


Figura 20: Retorno acumulado no período para a predição e para o índice. A linha azul representa o retorno acumulado real do S&P 500, enquanto a linha vermelha indica o retorno acumulado conforme predito pelo modelo. A estreita aderência entre as duas linhas sugere uma notável capacidade do modelo em capturar a trajetória de retorno do índice ao longo do período.

Autoria própria

Portanto, a partir do gráfico da Figura 20, pode-se observar que o retorno acumulado da previsão segue de maneira bastante alinhada o retorno acumulado real do índice. Isso sugere que o modelo captura efetivamente as tendências macroeconômicas e as dinâmicas de mercado que influenciam o S&P 500, um indicador de sua adequação para replicar a trajetória do índice com um nível de precisão substancial.

No entanto, ao focalizar na análise de *spread* da Figura 21, notamos algumas variações mais significativas, que indicam momentos de desalinhamento entre a previsão e o desempenho real. O *spread* é uma medida crucial pois revela não apenas a precisão pontual do modelo ao final do período prevista, mas também sua estabilidade ao longo do processo. Picos e vales ocasionais para além da média podem ser indicativos de volatilidade no mercado que o modelo não capturou, indicando uma possível limitação do modelo em responder a tempo às mudanças repentinas.

Apesar dessas variações, a média do *spread* abaixo de um ponto percentual indica que

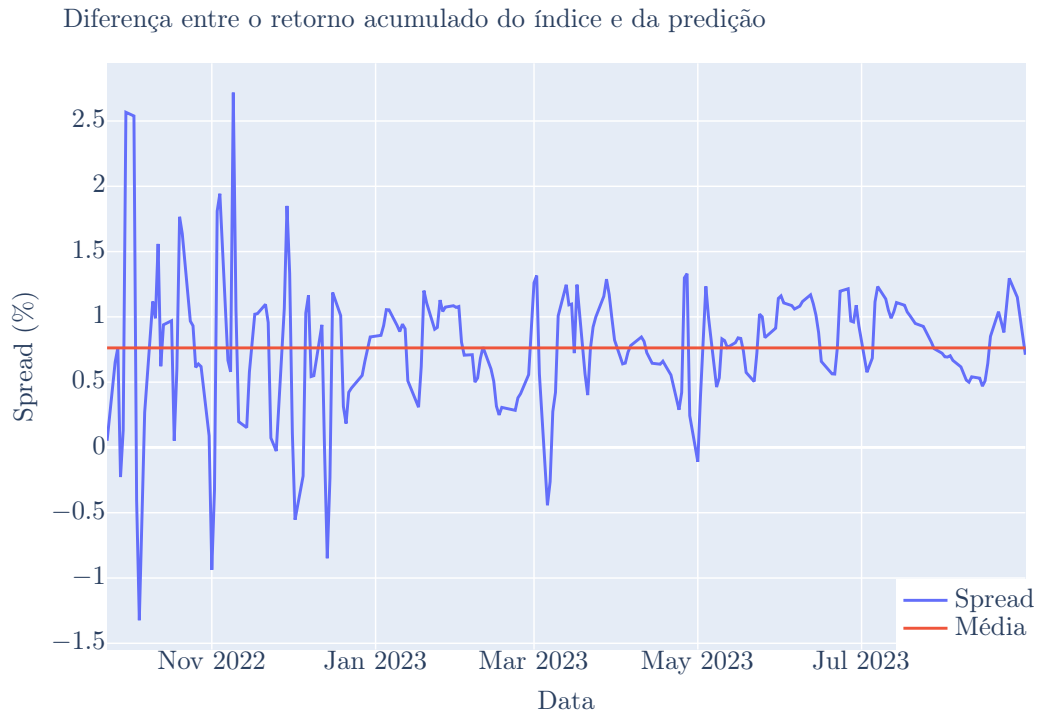


Figura 21: Diferença (distância) entre o retorno acumulado do índice e da predição — *spread*. Valores positivos do *spread* indicam que o retorno do índice superou o retorno predito pelo modelo. A linha vermelha denota a média ao longo do período analisado. Em virtude de uma volatilidade de mercado mais alta, percebe-se um descolamento mais acentuado no início do período, alimentado ainda por tentativas de correções superestimadas pelo próprio modelo.

Autoria própria

o modelo é bem-sucedido na sua tarefa principal de espelhar o índice, na verdade esse valor se apresenta em conformidade, inclusive, com o erro de rastreamento (*tracking error*) de 0,536%. Em conclusão, os resultados indicam um modelo que é, no geral, competente e fornece fortes indícios de aprender com o mercado, mas que também deixa espaço para aprimoramentos.

## 6.2 Influência da Atenção

Consideramos o mecanismo de atenção um componente crítico do nosso modelo, em virtude disso, optamos por nos aprofundar e experimentar alternativas tanto com distintos mecanismos quanto sem qualquer atenção. A Figura 22 apresenta uma comparação visual entre os modelos com as diferentes configurações.

É evidente pelo exposto que o modelo sem mecanismo de atenção apresentou o desem-



Figura 22: Comparativo do desempenho dos modelos frente abordagens distintas no uso ou não de atenção. Na coluna da esquerda encontram-se os resultados no conjunto de dados de treino, estando na direita os de teste. Cada linha explora uma alternativa, de cima para baixo: sem mecanismo de atenção; utilizando a atenção com *tanh* conforme (QIU et al., 2020); valendo-se da abordagem clássica por *query*, *key* e *value* segundo (VASWANI et al., 2017).

penho menos satisfatório, com a maior discrepância entre as previsões e os valores reais, tanto nos dados de treino quanto de teste. A introdução da atenção suave que utiliza tangente hiperbólica melhorou o desempenho do modelo; no entanto, observa-se a introdução de um atraso (*lag*) nas previsões em comparação com os valores reais, mais notadamente nos dados de teste. Isso sugere que, embora a atenção com *tanh* tenha proporcionado

alguma melhoria na capacidade do modelo de seguir a tendência dos dados, ela também pode ter comprometido a agilidade da resposta do modelo a mudanças no mercado.

Por outro lado, a atenção com *query*, *key* e *value* mostrou-se a mais promissora, com um ajuste impressionante nos dados de treino e a melhor performance nos dados de teste. As implicações disso, em geral, são significativas, pois sugerem que a estrutura de atenção adequada pode ter um impacto substancial na eficácia dos modelos preditivos, inclusive no domínio financeiro.

Cabe, por fim, a ressalva de que os hiperparâmetros foram otimizados exclusivamente para o melhor caso, portanto, uma análise mais aprofundada deveria incluir a modelagem desde a base da rede, propiciando uma otimização exclusiva para cada arquitetura, o que foge ao escopo desse trabalho.

## 6.3 Performance Financeira do Modelo

A avaliação da performance financeira de um modelo de investimento é um componente crítico para estabelecer sua efetividade no ambiente de mercado. A precisão nas previsões é um aspecto importante, mas a maneira como os recursos são geridos e alocados com base nestas previsões é igualmente essencial. Nessa seção, exploramos como o nosso modelo se comporta sob diferentes estratégias de gestão de banca.

As simulações foram conduzidas em uma janela temporal de 236 dias (referentes ao conjunto de dados de teste), um capital inicial pré-definido em R\$ 1.000,00 para todos os casos e um valor base de aposta (*stake*) de R\$ 100,00 salvo quando expresso o contrário — como no caso das apostas proporcionais. Cada simulação envolveu a execução do modelo para prever a direção do mercado (alta ou baixa) e então embasado pela previsão, o investidor entra com uma posição *long* (longa, apostando na alta) ou *short* (curta, apostando na baixa) do índice. O retorno é então determinado pelo desempenho real do índice, avaliando se a previsão foi bem-sucedida. As apostas são atualizadas de acordo com a estratégia de gestão de banca escolhida, refletindo o resultado da previsão anterior.

Salienta-se que os testes ainda que busquem aproximar o modelo da prática, incorrem em condições de contorno que delimitam apropriadamente o problema a um ambiente de controle no qual algumas hipóteses são tomadas como verdadeiras, a saber:

1. As operações excluem taxas e custos operacionais de qualquer natureza (corretagens, emolumentos, taxas de registro, etc.);

2. Consideram-se liquidez e volume suficientes no mercado para atender às ordens de compra e venda do investidor em qualquer circunstância e sem que essas exerçam força suficiente para influenciar as tendências pré-existentes;
3. Assumem-se operações próximas ao fechamento do mercado, a fim de alinhar os preços com aquilo que é trabalhado pelo modelo;
4. Exclui-se a possibilidade de alavancagem financeira, limitando o valor máximo da aposta ao capital inicial acrescido dos ganhos e decrescido das perdas anteriores;
5. Considera-se que as operações de venda a descoberto (*short*) têm a mesma complexidade e custo que as operações de compra tradicionais (*long*).

É importante ainda delinear que investimentos em bolsas de valores não devem ser tratados como apostas em jogos de azar. A premissa aqui desenvolvida é de ancorar decisões em previsões confiáveis de um modelo matemático e ainda que esse use de rebuscado fundamento estatístico, o mercado de ações possui inerentes riscos e incertezas. Os investidores devem sempre proceder com cautela, considerando a tolerância individual a riscos e sua disponibilidade de capital.

### 6.3.1 Martingale

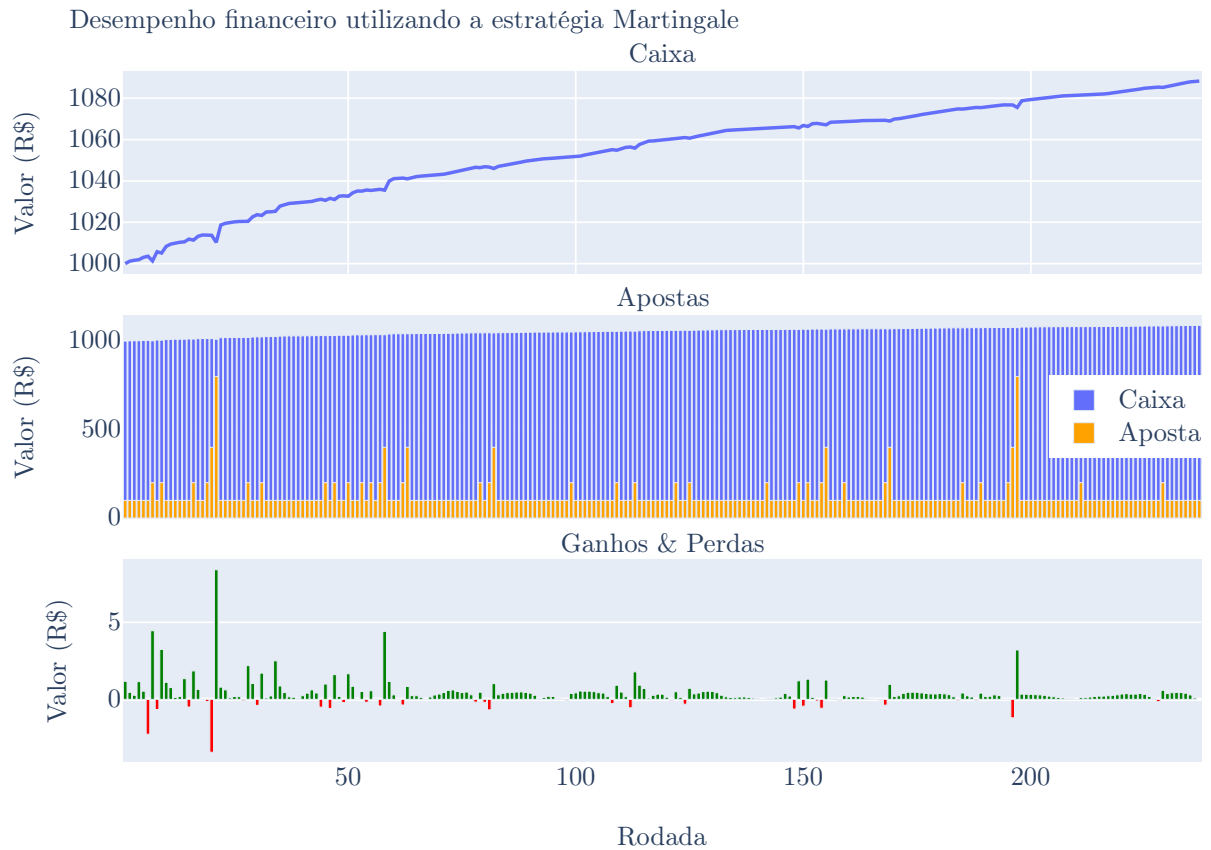


Figura 23: Desempenho financeiro da estratégia Martingale. O gráfico superior exibe a evolução do caixa (em R\$), evidenciando um acréscimo gradual ao capital inicial. O gráfico central mostra as apostas individuais (em laranja) em comparação com o caixa disponível (em azul), destacando a variação do montante apostado. O gráfico inferior apresenta os ganhos (em verde) e perdas (em vermelho) por rodada, delineando a frequência e a magnitude dos resultados de cada aposta realizada.

Autoria própria

Como mencionado, a estratégia Martingale, aplicada nesse contexto financeiro, baseia-se na premissa de dobrar a aposta após cada resultado negativo, com o intuito de recuperar as perdas e gerar lucro igual ao valor da aposta inicial quando um resultado positivo é alcançado. A Figura 23 ilustra a aplicação dessa estratégia ao longo das mais de 200 rodadas de apostas. Pode-se observar que o caixa apresenta uma tendência de crescimento gradual, sugerindo que, apesar da volatilidade e de perdas pontuais, os ganhos constantes foram suficientes para compensar as apostas anteriores e acrescentar um lucro modesto ao montante inicial.

Contudo, é importante notar que o lucro obtido é relativamente pequeno quando

comparado ao número de apostas realizadas e ao horizonte de tempo considerado.

Apesar de ser considerada uma estratégia agressiva, o desempenho modesto pode ser justificado pelo *stake* reduzido — de apenas R\$ 100,00 — e pela precisão do modelo: como os erros são bastante esporádicos, longas tendências de perda ocorrem muito pouco, então ganhos advêm, basicamente, de séries mais comedidas de acertos com baixo *stake*.

Por fim, é importante ressaltar que o desempenho do caixa indica que o modelo foi capaz de evitar perdas catastróficas — que não são raras nessa estratégia —, mas também sugere uma eficiência limitada em termos de geração de lucro em longo prazo.

### 6.3.2 Paroli

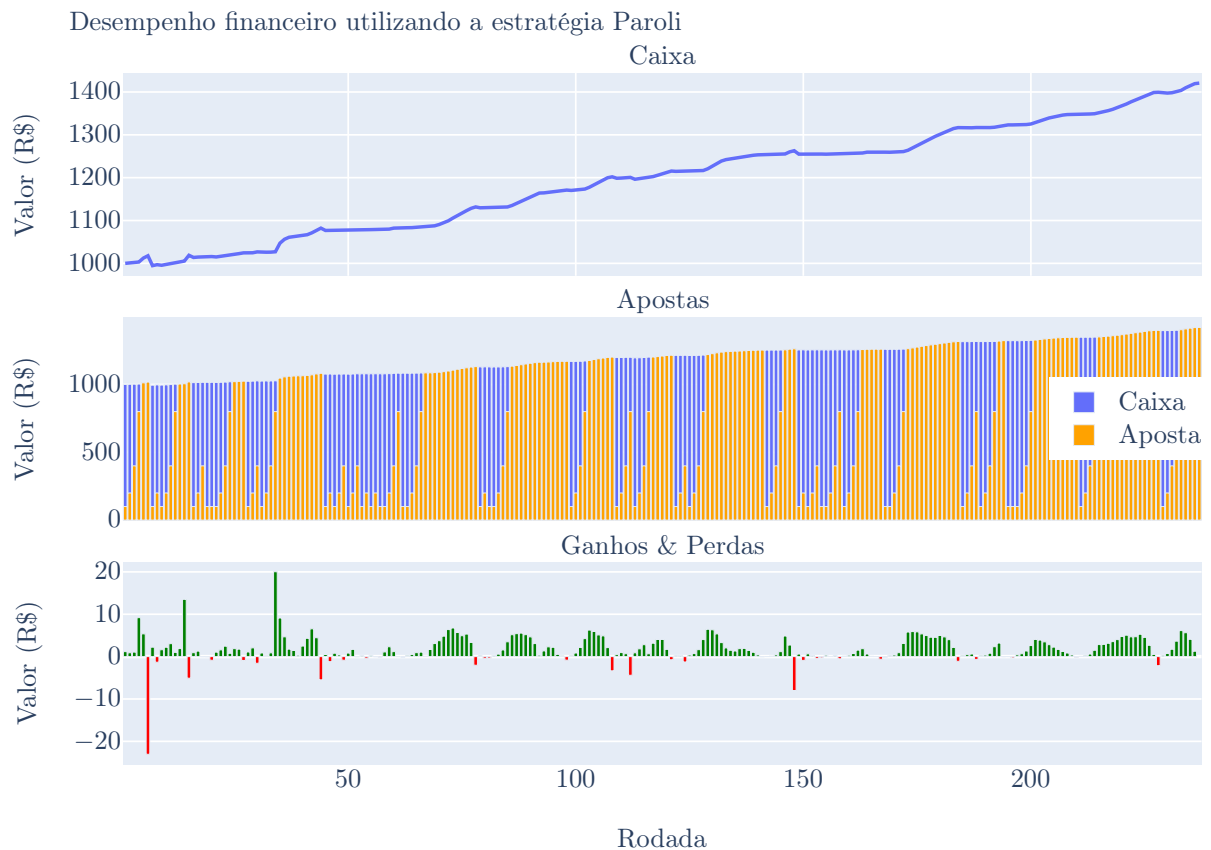


Figura 24: Desempenho financeiro da estratégia Paroli. O gráfico no topo mostra a progressão do caixa ao longo das rodadas, demonstrando um aumento consistente. O gráfico central detalha as apostas individuais em contraste com o caixa disponível, é notável que, em inúmeras vezes, o investidor aloca todo o seu caixa disponível. O gráfico de barras inferior ilustra os ganhos (verde) e as perdas (vermelho) por rodada, destacando a capacidade do modelo de aproveitar sequências de vitórias para construir lucro incrementalmente.

Autoria própria



A estratégia Paroli, também conhecida como Anti-Martingale, fundamenta-se, como sugere o nome, na duplicação da aposta após cada resultado positivo, visando capitalizar em uma tendência de ganhos enquanto minimiza perdas durante séries de resultados negativos. Observando a Figura 24, é evidente que a aplicação dessa estratégia resultou em um lucro acumulado substancial, refletido pelo aumento progressivo do caixa.

Esse crescimento sugere que o modelo foi capaz de identificar corretamente largas sequências de resultados positivos, permitindo que o investidor capitalizasse nesses momentos.

Em relação ao período testado, o lucro foi bastante substancial, sendo uma das melhores estratégias aqui apresentadas, todavia, é válido notar que o volume de apostas — e consequentemente a exposição do investidor — são bastante elevados. Esse resultado está em conformidade com o que foi visto na estratégia anterior, porque aqui o investidor não busca recuperar perdas anteriores, mas sim acumular ganhos de forma progressiva, rapidamente reduzindo sua exposição em momentos desfavoráveis.

A proibição, ou pelo menos limitação, da alavancagem é fundamental para que essa estratégia e a de Martingale forneçam resultados minimamente factíveis, haja vista que, em ambas, a capacidade do investidor apostar é diretamente limitada pelo caixa. Em Martingale, o investidor chega ao limite de alocar, no máximo, tudo o que tem em uma sequência de perdas, sem a chance de ganhos desproporcionais em relação ao seu caixa, o que também ocorre com Paroli, em uma sequência de ganhos.

### 6.3.3 D'Alembert

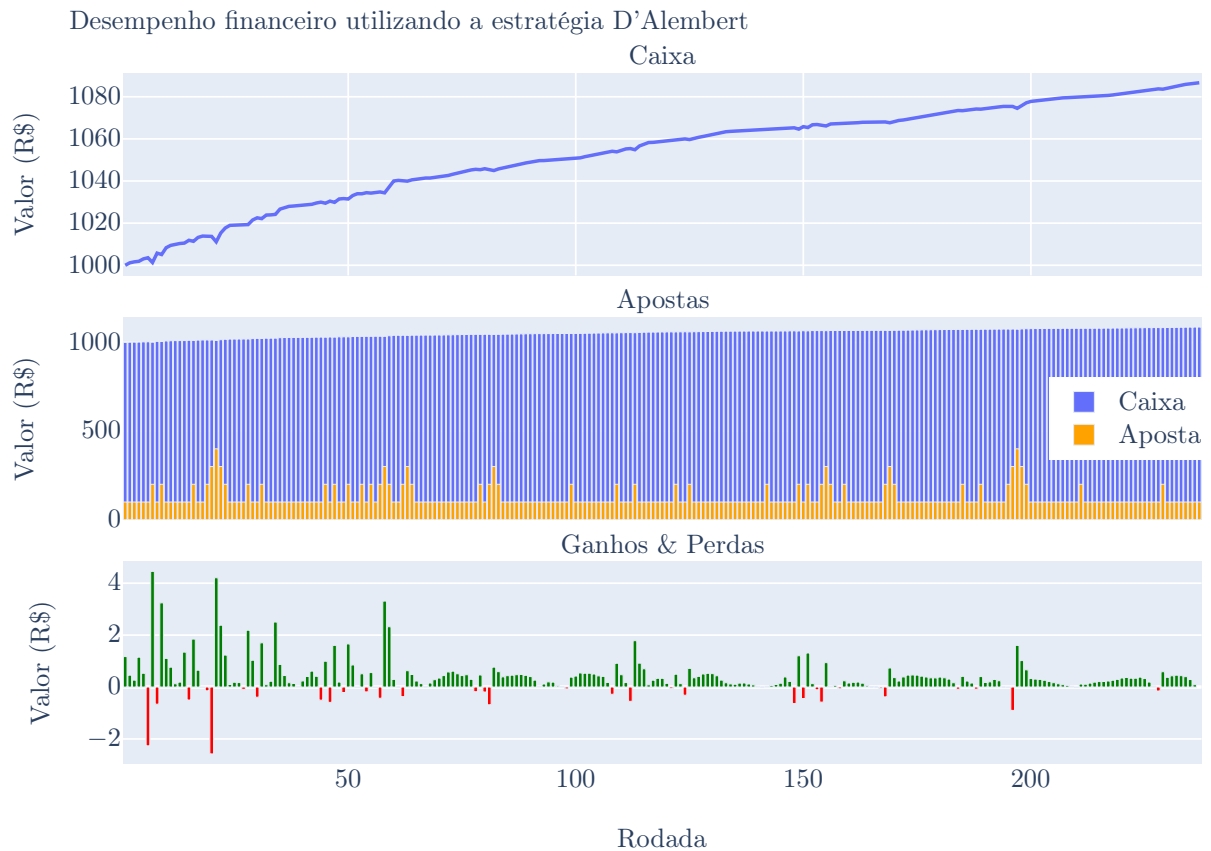


Figura 25: Desempenho financeiro da estratégia D'Alembert. O gráfico superior mostra a trajetória do caixa ao longo das rodadas, exibindo um aumento progressivo, porém modesto. O gráfico do centro ilustra as apostas (em laranja) e o caixa (em azul), destacando a gestão conservadora do capital. O gráfico inferior registra os ganhos (verde) e as perdas (vermelho) a cada rodada, mostrando que, apesar da abordagem cautelosa, houve uma quantidade significativa de perdas. O lucro final, comparável ao da estratégia Martingale, é modesto em relação ao período e em conformidade com o volume das apostas.

Autoria própria

A estratégia D'Alembert é uma abordagem bastante conservadora que aumenta a aposta em uma unidade após uma perda e a diminui em uma unidade após um ganho. O objetivo é estabelecer um equilíbrio entre ganhos e perdas, com o intuito de alcançar um crescimento gradual do caixa.

A Figura 25 exibe um crescimento lento e constante do caixa, característico dessa abordagem. Assim, o lucro obtido ao final do período é modesto — comparável ao obtido com Martingale —, especialmente considerando o número de rodadas, o que é consistente com a natureza cautelosa da estratégia.

O gráfico central revela que em pouquíssimos momentos o investidor alocou uma porção significativa do seu caixa, indicando sequências longas de acertos que o faziam reduzir o investimento a seu valor inicial.

É interessante ainda notar que, apesar de ser uma estratégia destinada a minimizar riscos, houve um número considerável de perdas, inclusive frente a estratégias mais agressivas como Martingale.

### 6.3.4 Apostas Fixas

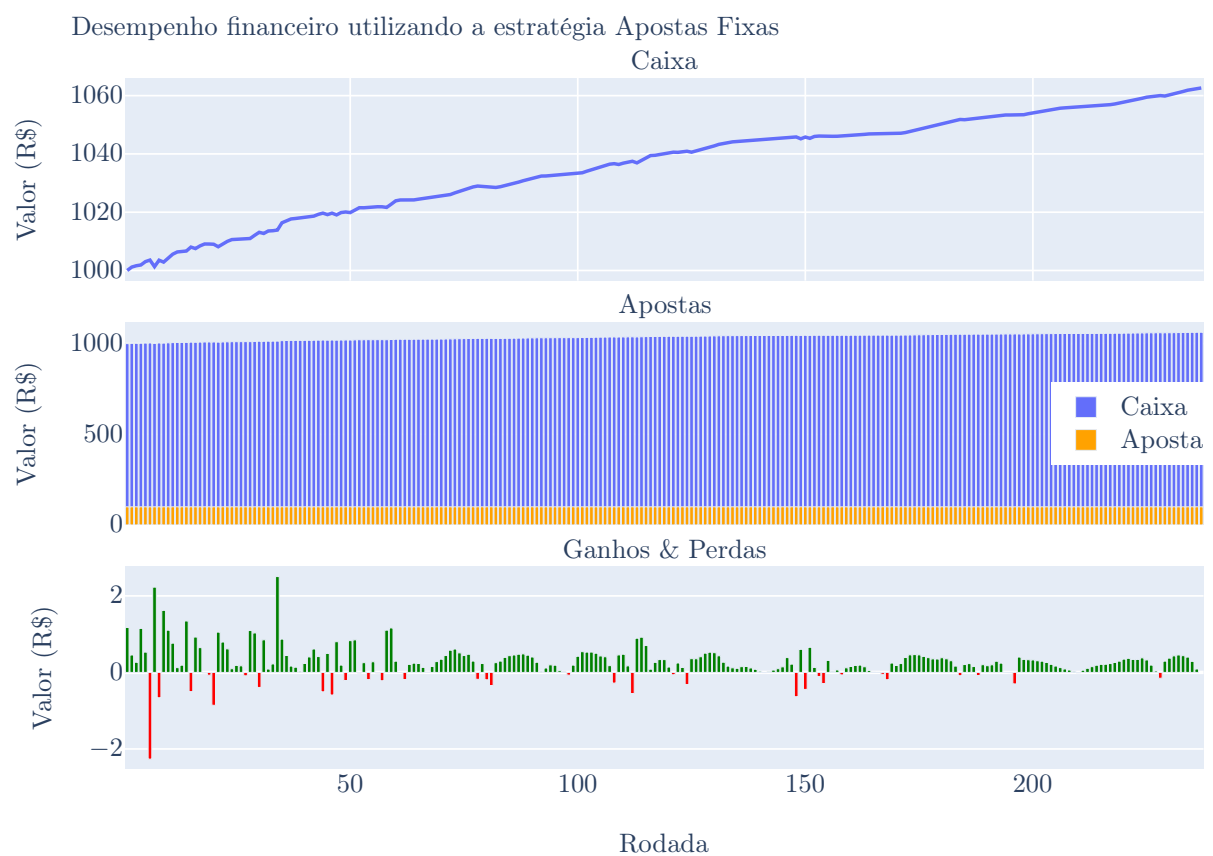


Figura 26: Desempenho financeiro da estratégia de Apostas Fixas. O gráfico superior mostra uma progressão lenta e consistente do caixa, sem grandes oscilações. O gráfico central evidencia a consistência da estratégia, com o valor de cada aposta (em laranja) mantido igual independentemente do resultado das rodadas anteriores, refletindo a natureza pouco sofisticada da abordagem. O gráfico inferior destaca os ganhos e perdas por rodada (ganhos em verde, perdas em vermelho).

Autoria própria

A estratégia de Apostas Fixas é caracterizada pela constância no valor investido em cada aposta, independentemente dos resultados anteriores. Essa técnica é considerada

uma das menos sofisticadas, pois não ajusta o valor da aposta com base em ganhos ou perdas anteriores. Ainda que busque controlar os riscos e evitar a possibilidade de perdas substanciais, essa alternativa é basicamente passiva e não reage aos resultados que alcança.

A Figura 26 mostra um incremento muito gradual do caixa ao longo das rodadas, refletindo a estabilidade forte dessa estratégia — quando considerados *stakes* pequenos. No entanto, o lucro final é bastante limitado, sendo essa a pior estratégia de gestão de banca desenvolvida em termos de rentabilidade, o que pode ser atribuído à limitação da possibilidade do investidor de capitalizar sobre séries de resultados positivos, já que o valor apostado permanece constante.

### 6.3.5 Apostas Proporcionais

Nessa abordagem, foram realizados 5 simulações, considerando apostas proporcionais de 10%, 25%, 50%, 75 e 90% do capital da banca em cada aposta. Essa é a única estratégia em que o investimento inicial não é necessariamente de R\$ 100,00.

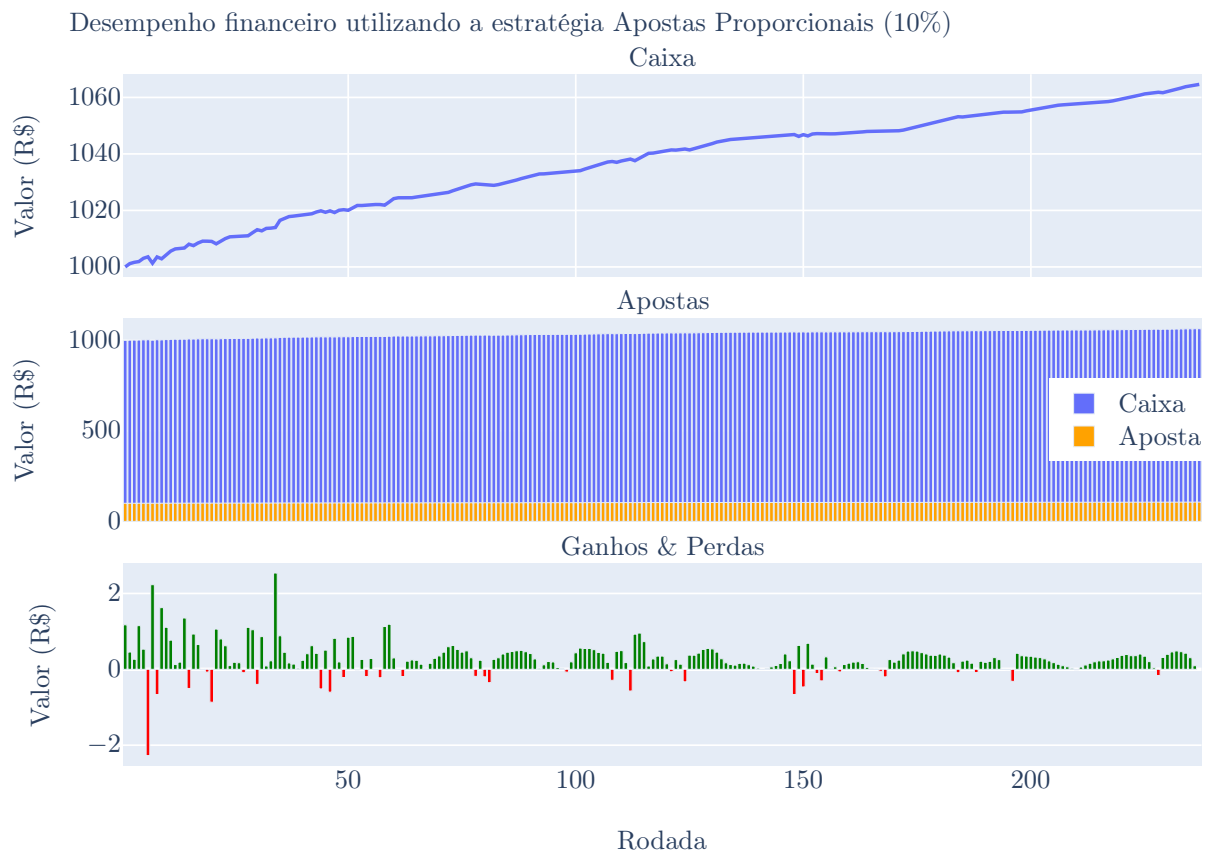


Figura 27: Desempenho financeiro da estratégia de Apostas Proporcionais a 10%. O gráfico superior evidencia um acréscimo marginal no caixa, refletindo uma abordagem conservadora que resultou em ganhos limitados. O gráfico central apresenta as apostas proporcionais (em laranja) ao caixa (em azul), com um valor baixo devido ao percentual reduzido aplicado e relativamente constante, devido a resultados modestos. O gráfico inferior mostra os ganhos (em verde) e as perdas (em vermelho) de cada rodada, destacando a similaridade com os resultados obtidos pela estratégia de Apostas Fixas devido ao baixo risco e retorno.

Autoria própria

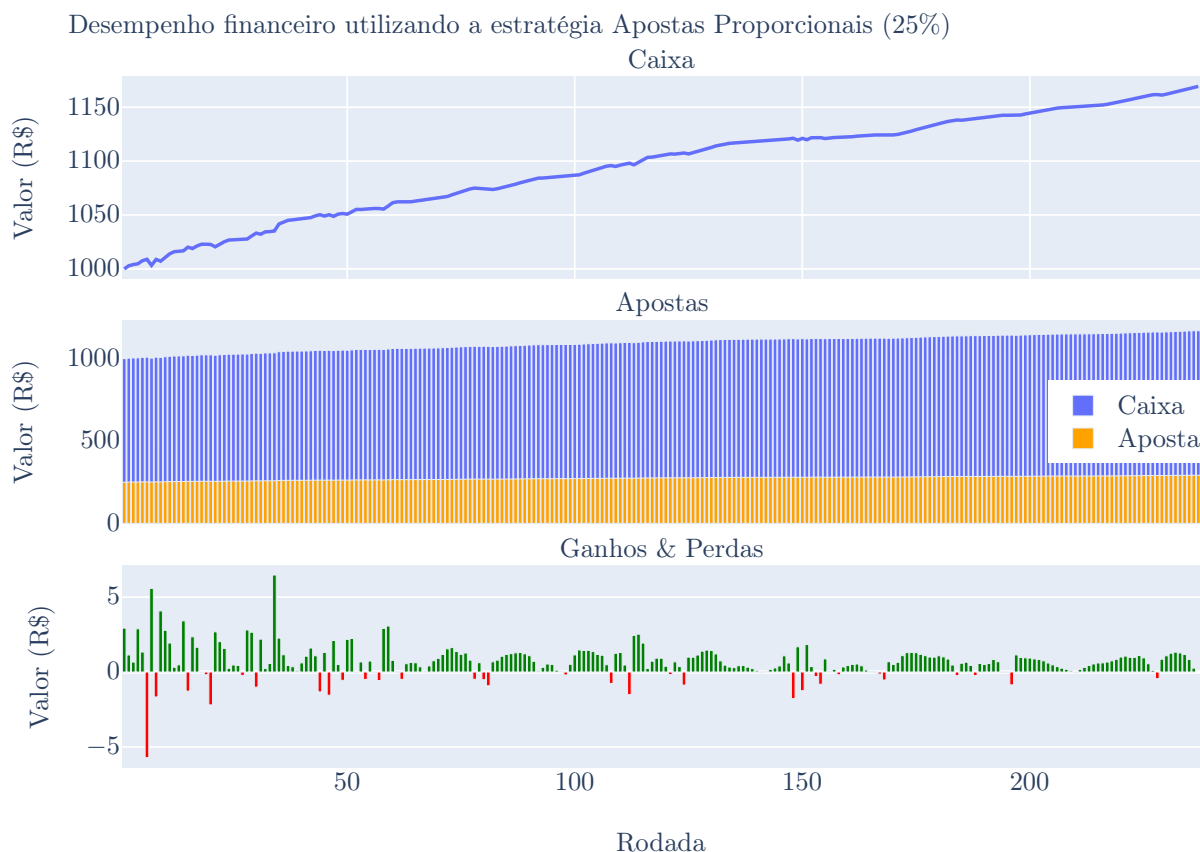


Figura 28: Desempenho financeiro da estratégia de Apostas Proporcionais a 25%. O aumento do caixa é mais perceptível quando comparado ao modelo de 10%, como demonstrado no gráfico superior. O aumento da proporção de apostas (em laranja) no gráfico do centro é mais notável conforme o crescimento do caixa, indicando um gerenciamento de risco mais intermediário. Os ganhos e perdas por rodada são ilustrados no gráfico inferior, resultando em uma rentabilidade boa para o período, sobretudo frente ao risco.

Autoria própria

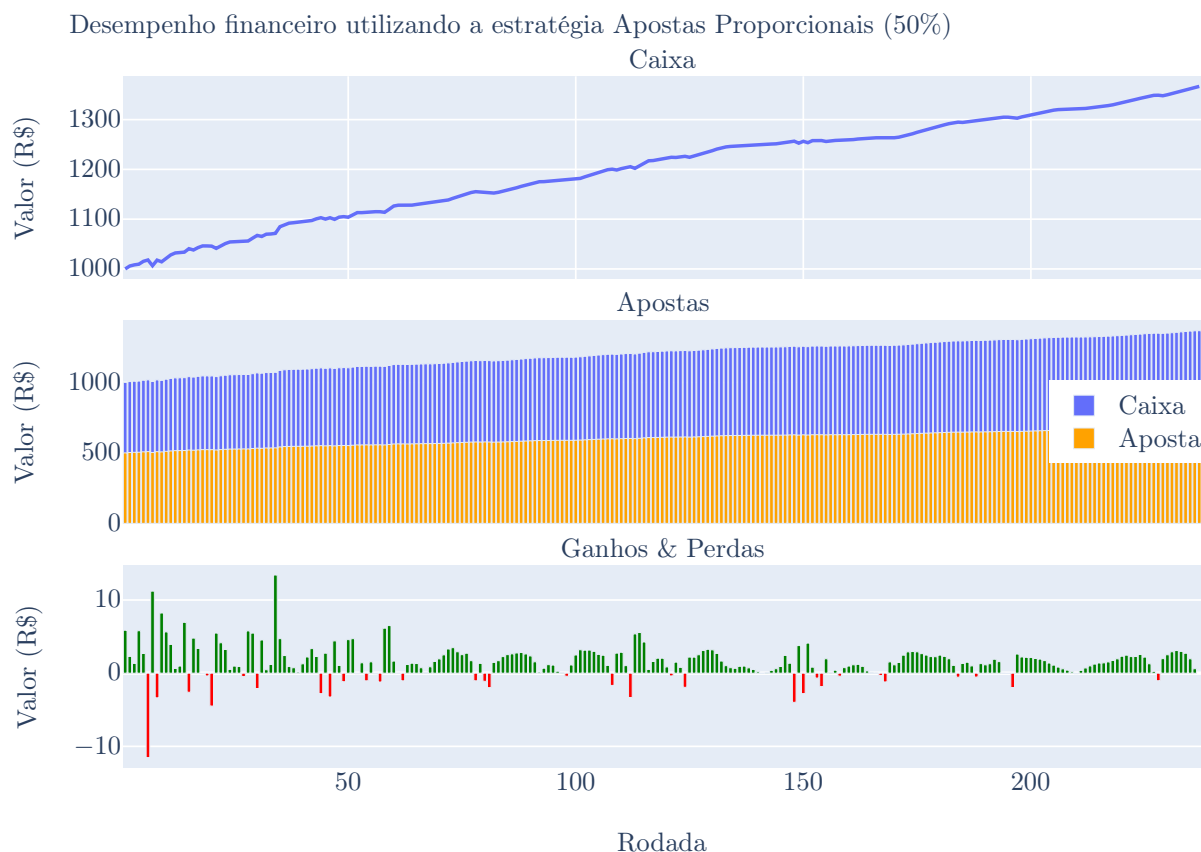


Figura 29: Desempenho financeiro da estratégia de Apostas Proporcionais a 50%. O gráfico no topo mostra um crescimento notável do caixa, indicando uma abordagem mais apta ao risco. O gráfico central destaca o aumento significativo nas apostas (em laranja) à medida que o caixa cresce, demonstrando um comprometimento substancial de recursos. O gráfico inferior registra ganhos e perdas por rodada, com os ganhos (verde) sobressaindo mais frequentemente do que as perdas (vermelho), ilustrando um bom desempenho geral da estratégia.

Autoria própria

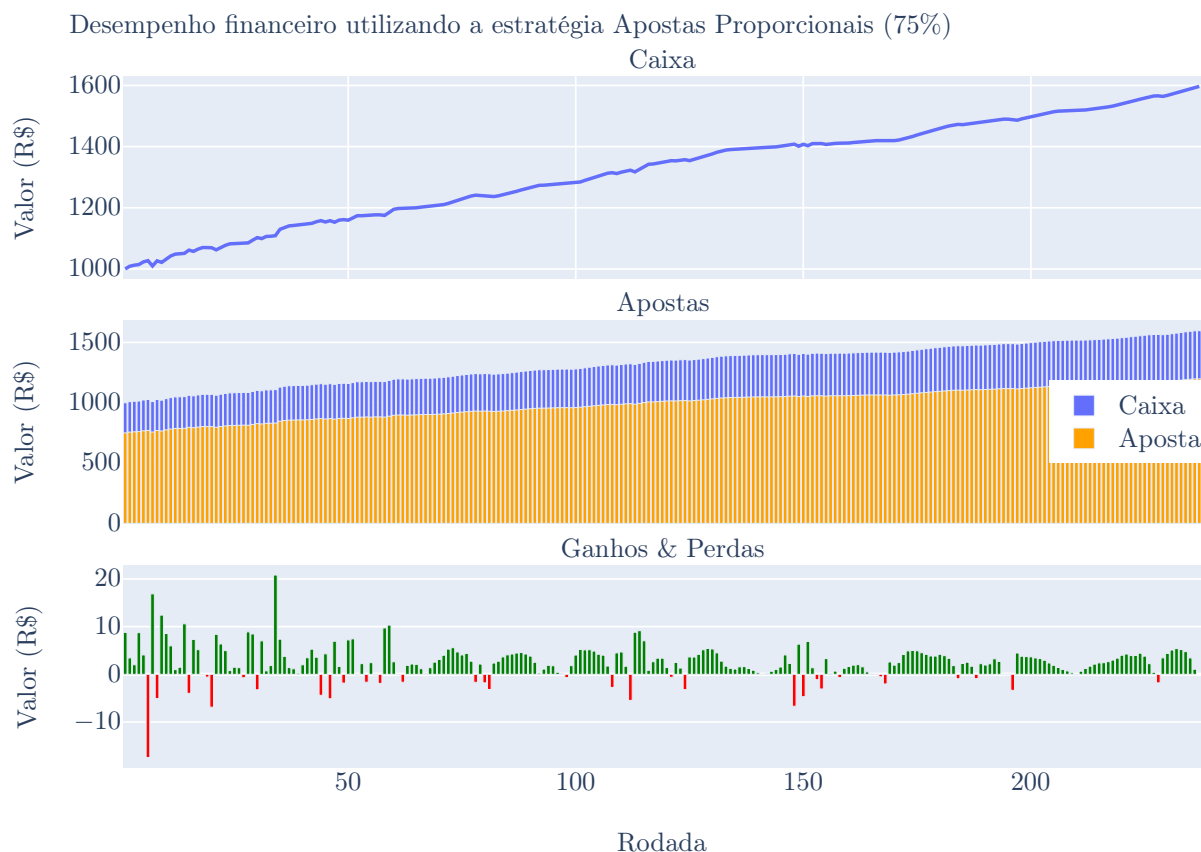


Figura 30: Desempenho financeiro da estratégia de Apostas Proporcionais a 75%. Essa abordagem agressiva resultou em um aumento substancial do caixa, conforme mostrado no gráfico superior. As apostas (em laranja) no gráfico central refletem uma maior alocação de capital, em linha com o aumento do caixa, tipicamente delatando um comportamento ávido ao risco. O gráfico inferior revela uma frequência aumentada de ganhos (verde), com os valores superando as perdas (vermelho), evidenciando uma estratégia mais rentável, porém mais volátil.

Autoria própria



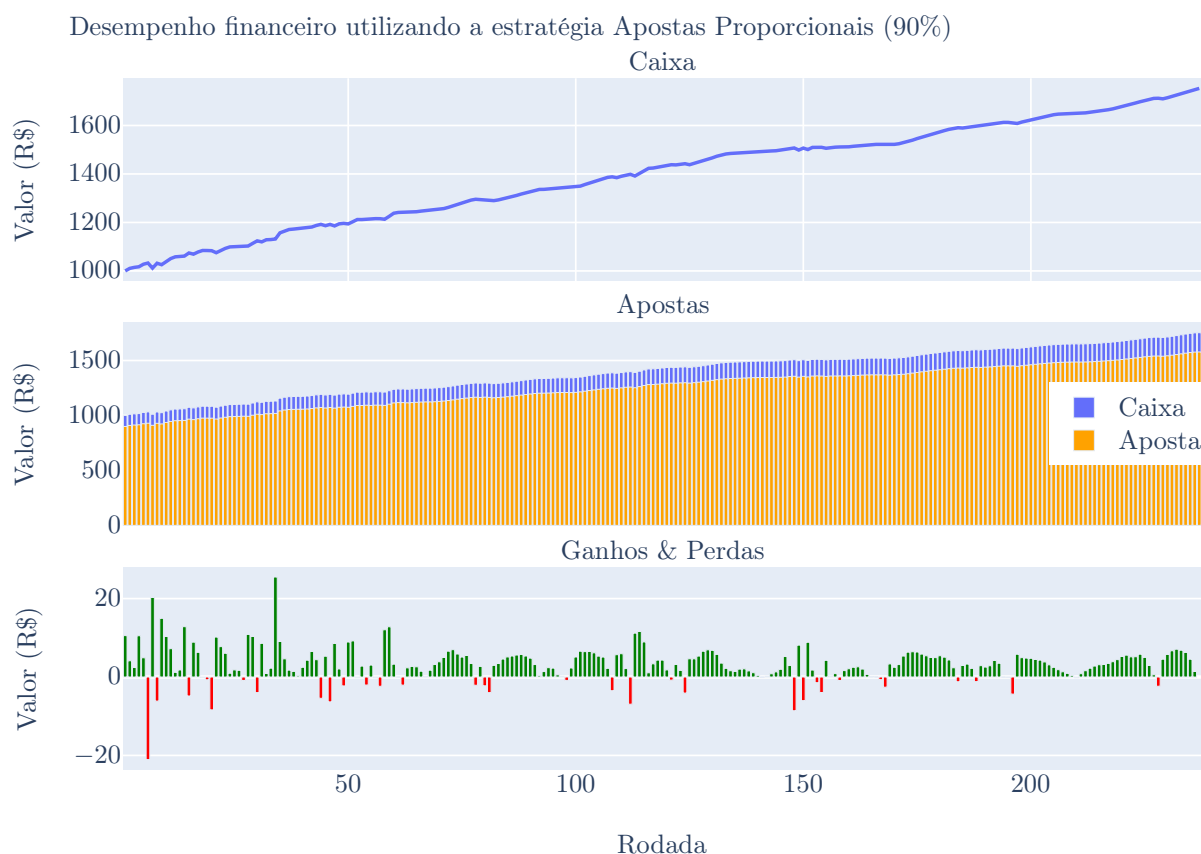


Figura 31: Desempenho financeiro da estratégia de Apostas Proporcionais a 90%. O gráfico superior exibe o maior crescimento do caixa entre todas as estratégias testadas, sugerindo uma alta rentabilidade. No gráfico do meio, as apostas (em laranja) acompanham de perto o valor do caixa, evidenciando uma estratégia altamente agressiva. O gráfico de ganhos e perdas mostra uma variação maior nos resultados de cada rodada, com ganhos consideravelmente altos (verde) que compensam as perdas (vermelho) ao todo, refletindo o elevadíssimo risco e a alta rentabilidade dessa abordagem.

Autoria própria

Na estratégia de Apostas Proporcionais, observa-se uma correlação direta entre o percentual da banca investido e o retorno financeiro: quanto maior a porcentagem apostada, maior o lucro potencial. A abordagem conservadora de 10% resultou em um desempenho modesto, assemelhando-se aos resultados da estratégia de Apostas Fixas. Em contraste, o modelo de 90% apresentou uma rentabilidade significativamente elevada, porém com um risco proporcionalmente alto, evidenciando que a maximização dos ganhos vem acompanhada de uma possibilidade aumentada de perdas substanciais.

Por outro lado, uma estratégia que aloca 50% do capital parece oferecer um ponto de equilíbrio atraente entre risco e retorno, fornecendo ganhos sólidos sem a exposição perigosa de uma estratégia mais agressiva. Esse nível de alocação permite ao investidor

aproveitar as vantagens de apostas maiores durante as tendências de ganho, sem arriscar quase a totalidade do caixa em períodos de perda.

Além disso, a natureza dessa estratégia permite uma atuação menos ativa em comparação a métodos que exigem ajustes constantes baseados nos resultados anteriores. Contudo, isso pode resultar em alguma incapacidade tática na recuperação de perdas anteriores. Investidores podem preferir essa abordagem por sua simplicidade e potencial de crescimento gradual, mas devem estar cientes de que, em face de uma série de perdas, a recuperação do capital pode ser um processo lento.

### 6.3.6 Considerações Finais

	Martingale	Paroli	D'Alembert	Fixa	Prop. 10%	Prop. 25%	Prop. 50%	Prop. 75%	Prop. 90%
<b>Caixa Final</b>	1.088,33	1.420,72	1.086,66	1.062,64	1.064,61	1.169,31	1.366,78	1.597,00	1.753,04
<b>Ganhos Tot.</b>	104,30	490,78	101,47	73,83	76,06	198,96	429,71	697,36	878,08
<b>Perdas Tot.</b>	-15,97	-70,06	-14,81	-11,19	-11,45	-29,65	-62,93	-100,36	-125,04
<b>Lucro</b>	88,33	420,72	86,66	62,64	64,61	169,31	366,78	597,00	753,04
<b>Retorno</b>	8,83%	42,07%	8,67%	6,26%	6,46%	16,93%	36,68%	59,70%	75,30%
<b>Ret. (ano)</b>	9,38%	45,04%	9,20%	6,64%	6,85%	18,01%	39,21%	64,16%	81,19%
<b>Vol. (ano)</b>	1,38%	4,65%	1,12%	0,67%	0,68%	1,70%	3,41%	5,11%	6,13%
<b>Ret./Risco</b>	6,78	9,69	8,21	9,89	10,06	10,57	11,51	12,55	13,24

Tabela 7: Comparativo de desempenho financeiro entre as diferentes estratégias de gestão de banca aplicadas ao modelo de previsão. As colunas concentram os critérios de comparação para cada abordagem. As cores comparam um mesmo critério (linha) entre todas as estratégias, tonalidades verdes indicam resultados melhores frente às tonalidades vermelhas, sendo o branco indicativo de valores intermediários. Além dos dados base das simulações, apresentam-se ainda valores anualizados para retorno e volatilidade, além da razão retorno por risco.

A Tabela 7 oferece uma visão abrangente dos resultados alcançados pelas distintas estratégias de gestão de banca. É notável que estratégias mais agressivas, como as de Apostas Proporcionais a 75% e 90%, alcançaram os maiores retornos absolutos, no entanto, essas vieram acompanhadas de uma volatilidade substancialmente maior.

Isso evidencia uma relação clássica entre risco e retorno, onde maiores riscos podem potencialmente recompensar os investidores com maiores ganhos, mas também podem resultar em perdas acentuadas. Nesse contexto, a estratégia de Apostas Proporcionais a 25% destaca-se por apresentar um equilíbrio notável entre risco e retorno, sugerindo ser uma opção viável para investidores que desejam uma abordagem mais comedida.

Por outro lado, estratégias conservadoras como Apostas Fixas e Proporcionais a 10%

mostraram-se menos voláteis, mas também menos rentáveis em termos absolutos. Esse desempenho reforça a ideia de que a cautela excessiva pode limitar o potencial de crescimento do capital.

Interessantemente, a estratégia Paroli sobressaiu com um retorno considerável, ultrapassando até mesmo algumas das estratégias proporcionais, o que demonstra a eficácia de capitalizar sobre as tendências positivas do mercado. A análise da razão de retorno pelo risco destaca as estratégias que conseguiram otimizar o equilíbrio entre os ganhos e as exposições ao risco, sendo um indicativo chave para a seleção de uma estratégia de gestão de banca. A decisão final do investidor deve, portanto, ponderar entre a tolerância ao risco e os objetivos de retorno, considerando que estratégias de alto risco podem não ser sustentáveis a longo prazo.

## 6.4 Compartilhamento dos Resultados

Para tornar as previsões do nosso modelo mais acessíveis, desenvolvemos um site que se encontra disponível através do link <https://gvmossato.github.io/>, com toda a infraestrutura necessária implementada, incluindo página *web* e banco de dados. Para demonstração, a Figura 32 contém uma visualização já utilizando dados reais.

O modelo é executado localmente, próximo ao horário de encerramento do mercado, o que visa tanto permitir a captura de preços próximos ao que seriam de fato o preço de fechamento — requeridos pelo modelo —, quanto fornecer tempo hábil necessário para o comando das operações — contornando as limitações práticas de se operar exatamente a esse preço.

O site é projetado para que, com base na previsão gerada pelo modelo (indicando alta ou baixa), e em consonância com a estratégia de gestão de banca selecionada pelo investidor, seja possível tomar decisões informadas de compra ou venda.

Adicionalmente, o treinamento do modelo é incrementado a cada 128 dias úteis (tendo em visto o *batch size* escolhido), o que corresponde a aproximadamente um semestre, permitindo que ele incorpore dados recentes que anteriormente não haviam sido vistos, esse processo é realizado com 15 *epochs*. Vale ressaltar que a eficácia do treinamento incremental ainda está sob análise, considerando o prazo e a necessidade de validação para garantir a confiabilidade das previsões em um ambiente de mercado em constante mudança.

Assim, como o treinamento integral do modelo compreendeu dados até 18/08/2021,

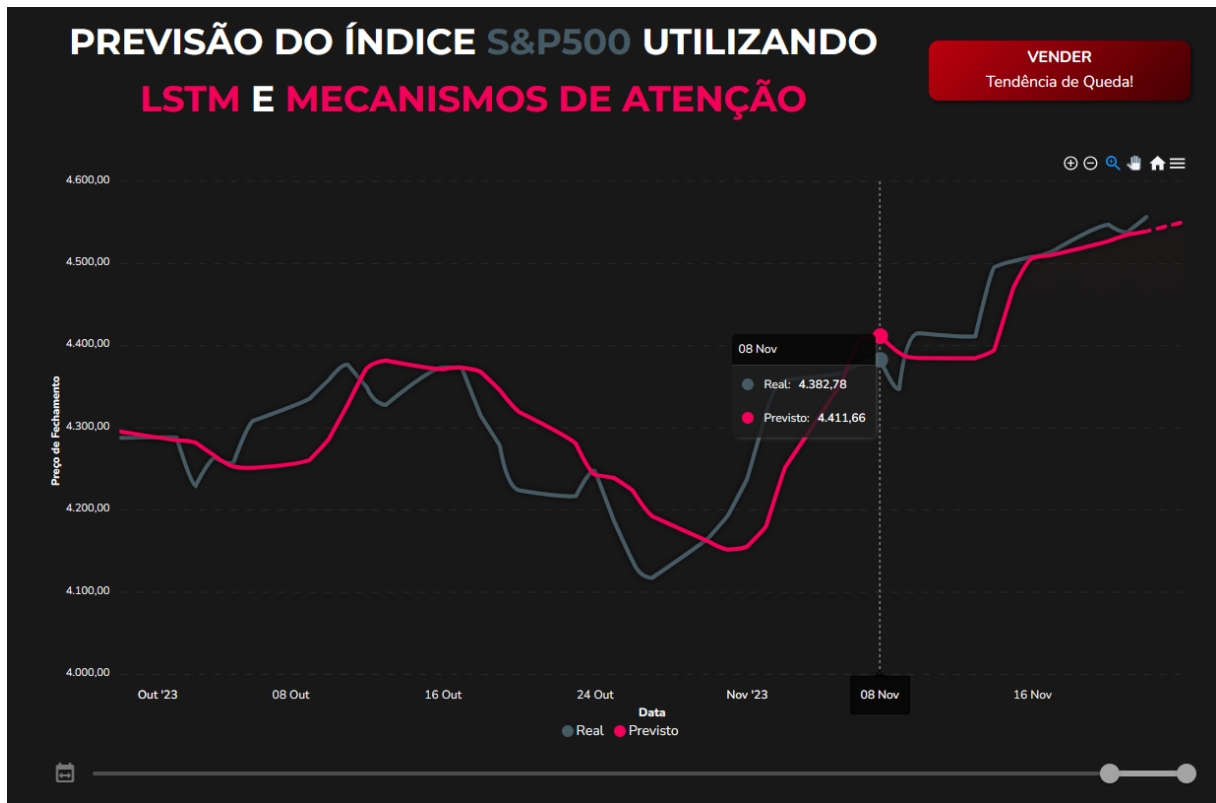


Figura 32: Interface do *site* de previsão do índice S&P 500. O portal encontra-se operacional e com a infraestrutura planejada, incluindo a página e o banco de dados hospedados *online*, enquanto o modelo é executado localmente. Os dados da imagem são as previsões reais do modelo frente ao índice. A interface mostra a previsão de fechamento ao lado do fechamento real, fornecendo indicações de compra ou venda baseadas na tendência predita.

já foram realizados treinamentos incrementais em 22/02/2022, 23/08/2022, 27/02/2023 e 28/08/2023, a fim de incorporar todas as informações mais recentes no modelo que está em produção.

Finalmente, a infraestrutura *online* conta com o *front-end* desenvolvido utilizando Vue.js, um *framework* progressivo para construir interfaces de usuário, e hospedado pelo GitHub Pages, devido a sua simplicidade e eficiência na entrega de conteúdo estático, o repositório pode ser acessado em [gvmossato/gvmossato.github.io](https://gvmossato.github.io). Em relação ao armazenamento de dados, escolhemos o Cloudant da IBM, um banco de dados distribuído como serviço (DBaaS) que utiliza o modelo não relacional e garante alta disponibilidade e durabilidade dos dados, a custos iniciais nulos. Essa combinação de tecnologias assegura que o site não só esteja fundamentado em uma base de previsão sólida, mas também que seja sustentável e escalável conforme a necessidade.

## 7 CONCLUSÕES

Esse trabalho abordou de forma abrangente a criação de um modelo de previsão para o índice S&P 500 utilizando redes neurais LSTM com mecanismos de atenção. A motivação se enraíza na compreensão de que o mercado de ações desempenha um papel vital no crescimento econômico e na geração de riqueza pessoal. Entretanto, a complexidade e a volatilidade do mercado, juntamente com a difusão de informações financeiras de baixa confiabilidade, tornam a tomada de decisão um desafio significativo para muitos investidores, especialmente os novos entrantes do mercado financeiro.

No desenvolvimento do modelo, foi empregada uma arquitetura baseada em células LSTM e mecanismos de atenção, inspirada em trabalhos anteriores como o de (QIU et al., 2020). O modelo foi otimizado usando técnicas de *grid search* e *bayesian search*, com um foco também análise da curva de aprendizado para garantir a eficácia do treinamento e evitar problemas como *overfitting*. Esse processo de otimização arrojado ajudou a refinar os hiperparâmetros e aprimorar o desempenho geral do modelo ao longo de diversas iterações.

Os resultados obtidos, apesar de não superarem completamente o *benchmark* estabelecido por (QIU et al., 2020), demonstraram um desempenho promissor na previsão dos preços de fechamento do S&P 500. A complexidade dos conjuntos de teste, que variaram em termos de escala, amplitude de preços e período temporal, tornou a comparação direta das métricas de erro entre o modelo proposto e o *benchmark* uma tarefa não trivial. No entanto, os indícios sugerem que o modelo desenvolvido é uma ferramenta promissora para prever os preços de fechamento do S&P 500, com diversificadas aplicações potenciais.

Além disso, a análise do retorno acumulado e a tendência de alta e baixa das predições foram importantes para avaliar a aplicabilidade prática do modelo no mercado financeiro.

Foi dada então uma ênfase especial ao mecanismo de atenção, considerado-o um elemento distintivo e que demonstrou considerável poder de influência nas predições. Testes comparativos foram realizados para quantificar o impacto desse mecanismo no contexto

das previsões financeiras, utilizando abordagens de atenção clássica e variações do modelo sem qualquer mecanismo de atenção.

Ademais, a eficácia financeira de uma estratégia de investimento, conforme entendida nesse trabalho, não se restringe apenas à precisão das previsões de mercado ou à minimização de métricas em relação a um *benchmark*, mas também inclui a metodologia empregada na alocação e gestão de recursos. Diferentes estratégias de gestão de banca foram analisadas para avaliar seu potencial impacto financeiro em conjunto com o modelo desenvolvido, em geral, essas possuíram resultados bastante positivos, apesar das hipóteses delineadas.

A importância desse trabalho reside na sua contribuição para a exploração do campo das previsões financeiras usando técnicas de aprendizado de máquina. Ao oferecer um sistema que pode auxiliar na tomada de decisões de *trading* diárias, o modelo tem o potencial de se tornar um recurso pertinente para investidores, especialmente aqueles com infraestrutura limitada e acesso tardio a informações financeiras. Todavia, os resultados apesar de promissores, representam um marco inicial para possíveis futuras iterações a fim de aprimorar o modelo.

# ANEXO A – CÓDIGO

## A.1 run.py

```

1 import yfinance as yf
2 from skopt.space import Categorical, Real
3
4 import alstm_stock_market.src.manager.strategies as st
5 import alstm_stock_market.src.model.params as p
6 from alstm_stock_market.src.data.preprocessor import Preprocessor
7 from alstm_stock_market.src.helpers.plotter import Plotter
8 from alstm_stock_market.src.helpers.utils import cmd_args
9 from alstm_stock_market.src.manager.manager import Manager
10 from alstm_stock_market.src.model.evaluator import Evaluator
11 from alstm_stock_market.src.model.model import Model
12
13
14 def main():
15     data = yf.download(p.ticker, start=p.start, end=p.end)
16
17     pre = Preprocessor(data, p.sets_sizes)
18     pre.run()
19
20     args = cmd_args()
21     model = Model(load_weights=args.load_weights)
22
23     if args.tuning:
24         param_grid = { # Grid search only
25             "model__learning_rate": [0.001, 0.01, 0.1],
26             "model__hidden_state_size": [10, 20, 50, 100],
27             "batch_size": [64, 128, 256, 512],
28         }
29         param_space = { # Bayesian search only
30             "model__learning_rate": Real(0.0001, 0.01, prior="log-
→ uniform"),

```

```

31         "model__dropout_rate": Real(0, 0.3),
32         "batch_size": Categorical([64, 128, 256, 1024]),
33     }
34     best = model.tune(
35         args.tuning,
36         pre.X_train,
37         pre.y_train,
38         param_grid,
39         param_space,
40     )
41     print(
42         f"Best score of {best['score']} obtained with parameters ↵
→ {best['params']}.",
43         "Overall results logs saved.",
44     )
45     return
46
47 plot = Plotter()
48
49 plot.wavelet_results(pre)
50 plot.wavelet_results_detail(pre)
51
52 model.fit(
53     pre.X_train,
54     pre.y_train,
55     pre.X_valdn,
56     pre.y_valdn,
57 )
58
59 if not args.load_weights:
60     plot.learning_curve(model)
61
62 pred_train = model.predict(pre.X_train)
63 pred_valdn = model.predict(pre.X_valdn)
64 pred_test = model.predict(pre.X_test)
65
66 plot.prediction_train(pre, pred_train)
67 plot.prediction_valdn(pre, pred_valdn)
68 plot.prediction_test(pre, pred_test)
69
70 evaluator = Evaluator(
71     pre.y_test,
72     pred_test,

```



```

73     pre.target_norm_mean,
74     pre.target_norm_std,
75 )
76 evaluator.run()
77
78 print("\nAvaliação dos Resultados:")
79 print("Raiz do Erro Quadrático Médio:", evaluator.metrics["rmse"] ↵
→ ])
80 print("Erro Absoluto Médio:", evaluator.metrics["mae"])
81 print("R-quadrado:", evaluator.metrics["r2"])
82 print("Tracking Error:", evaluator.metrics["te"])
83
84 plot.returns_trend_distribution(evaluator.y_trend, evaluator. ↵
→ y_pred_trend)
85 plot.confusion_matrix(evaluator.confusion_matrix)
86 plot.cumulative_return(pre, evaluator.cumulative_return)
87 plot.cumulative_return_spread(pre, evaluator.cumulative_return)
88
89 initial_cash = 1000
90 initital_bet = 100
91
92 manager = Manager(
93     initial_cash, initital_bet, evaluator.y_pred_trend, evaluator ↵
→ .y_return
94 )
95
96 strategies = {
97     "Martingale": st.Martingale(initital_bet),
98     "Paroli": st.Paroli(initital_bet),
99     "D'Alembert": st.DAlembert(initital_bet),
100     "Apostas Fixas": st.Fixed(initital_bet),
101     "Apostas Proporcionais (10%)": st.Proportional(initial_cash, ↵
→ proportion=0.10),
102     "Apostas Proporcionais (25%)": st.Proportional(initial_cash, ↵
→ proportion=0.25),
103     "Apostas Proporcionais (50%)": st.Proportional(initial_cash, ↵
→ proportion=0.50),
104     "Apostas Proporcionais (75%)": st.Proportional(initial_cash, ↵
→ proportion=0.75),
105     "Apostas Proporcionais (90%)": st.Proportional(initial_cash, ↵
→ proportion=0.90),
106 }
107

```

```

108     for name, strategy in strategies.items():
109         manager = Manager(
110             initial_cash,
111             strategy.initial_bet,
112             evaluator.y_pred_trend,
113             evaluator.y_return,
114         )
115         results = manager.run(strategy)
116         plot.strategy(results, name)
117
118
119 if __name__ == "__main__":
120     main()

```

## A.2 src/model/model.py

```

1 import tensorflow.keras.backend as K
2 from dotenv import load_dotenv
3 from scikeras.wrappers import KerasRegressor
4 from sklearn.model_selection import GridSearchCV, TimeSeriesSplit
5 from skopt import BayesSearchCV
6 from tensorflow.keras.layers import LSTM, Dense, Dropout, Input,
    → Layer
7 from tensorflow.keras.models import Sequential
8 from tensorflow.keras.optimizers import Adam
9
10 import alstm_stock_market.src.model.params as p
11 from alstm_stock_market.src.helpers.utils import (
12     get_latest_weights,
13     save_txt,
14     save_weights,
15 )
16
17 load_dotenv()
18
19
20 def create_model(
21     learning_rate=0.00018,
22     dropout_rate=0.12241,
23     hidden_state_size=20,
24     add_attention=False,
25 ):
26     model = Sequential()

```

```

27     model.add(Input(shape=(p.time_step, p.num_features)))
28     model.add(LSTM(hidden_state_size, return_sequences=add_attention) ↓
    → )
29     if add_attention:
30         model.add(ClassicAttention())
31     model.add(Dropout(dropout_rate))
32     model.add(Dense(1, activation="linear"))
33
34     optimizer = Adam(learning_rate=learning_rate)
35
36     model.compile(optimizer=optimizer, loss=p.loss_function)
37     return model
38
39
40 class ClassicAttention(Layer):
41     def __init__(self, **kwargs):
42         super(ClassicAttention, self).__init__(**kwargs)
43
44     def build(self, input_shape):
45         self.W_Q = self.add_weight(
46             shape=(input_shape[-1], input_shape[-1]),
47             initializer="glorot_uniform",
48             trainable=True,
49         )
50         self.W_K = self.add_weight(
51             shape=(input_shape[-1], input_shape[-1]),
52             initializer="glorot_uniform",
53             trainable=True,
54         )
55         self.W_V = self.add_weight(
56             shape=(input_shape[-1], input_shape[-1]),
57             initializer="glorot_uniform",
58             trainable=True,
59         )
60         super(ClassicAttention, self).build(input_shape)
61
62     def call(self, x):
63         Q = K.dot(x, self.W_Q)
64         K_mat = K.dot(x, self.W_K)
65         V = K.dot(x, self.W_V)
66
67         QK = K.batch_dot(Q, K.permute_dimensions(K_mat, (0, 2, 1)))
68         d_k = K.int_shape(Q)[-1]

```

```

69         scaled_attention_logits = QK / K.sqrt(K.cast(d_k, dtype=K. ↵
→ floatx()))
70
71         attention_weights = K.softmax(scaled_attention_logits, axis ↵
→ =-1)
72         weighted_sum = K.batch_dot(attention_weights, V)
73         return K.sum(weighted_sum, axis=1)
74
75     def compute_output_shape(self, input_shape):
76         return (input_shape[0], input_shape[-1])
77
78
79 class TanhAttention(Layer):
80     def __init__(self, **kwargs):
81         super(TanhAttention, self).__init__(**kwargs)
82
83     def build(self, input_shape):
84         self.W_a = self.add_weight(
85             name="W_a",
86             shape=(input_shape[-1], input_shape[-1]),
87             initializer="glorot_uniform",
88             trainable=True,
89         )
90         self.b_a = self.add_weight(
91             name="b_a",
92             shape=(input_shape[-1],),
93             initializer="zeros",
94             trainable=True,
95         )
96         super(TanhAttention, self).build(input_shape)
97
98     def call(self, x):
99         s_t = K.tanh(K.dot(x, self.W_a) + self.b_a)
100         attention_weights = K.softmax(s_t, axis=1)
101         weighted_input = attention_weights * x
102         return K.sum(weighted_input, axis=1)
103
104     def compute_output_shape(self, input_shape):
105         return (input_shape[0], input_shape[-1])
106
107
108 class Model:
109     def __init__(self, load_weights=False):

```

```

110     self.model = create_model(p.learning_rate, p.dropout_rate, p. ↵
→ hidden_state_size)
111     self.load_weights = load_weights
112
113     if self.load_weights:
114         self.model.load_weights(get_latest_weights())
115
116 def tune(self, method, X_train, y_train, param_grid, param_space) ↵
→ :
117     methods = {
118         "grid": lambda: self._grid_search(param_grid, X_train, ↵
→ y_train),
119         "bayes": lambda: self._bayesian_search(param_space, ↵
→ X_train, y_train),
120     }
121
122     try:
123         return methods[method]()
124     except KeyError:
125         raise ValueError(
126             f"Invalid tuning method. Valid methods are: {'', ' '. ↵
→ join(methods.keys())}"
127         )
128
129 def fit(self, X_train, y_train, X_valdn, y_valdn):
130     if self.load_weights:
131         return None
132
133     self.fitted = self.model.fit(
134         X_train,
135         y_train,
136         epochs=p.epochs,
137         batch_size=p.batch_size,
138         validation_data=(X_valdn, y_valdn),
139         validation_freq=1,
140         shuffle=False,
141         verbose=1,
142     )
143     save_weights(self.model)
144
145 def _grid_search(self, X_train, y_train, param_grid):
146     estimator = KerasRegressor(
147         model=create_model,

```

```

148         epochs=p.epochs,
149         loss=p.loss_function,
150         shuffle=False,
151         verbose=0,
152     )
153     grid_search = GridSearchCV(
154         estimator=estimator,
155         param_grid=param_grid,
156         scoring=f"neg_{p.loss_function}",
157         n_jobs=-1,
158         cv=TimeSeriesSplit(n_splits=3),
159     )
160     grid_search_result = grid_search.fit(X_train, y_train)
161
162     best = {
163         "score": grid_search_result.best_score_,
164         "params": grid_search_result.best_params_,
165     }
166
167     save_txt(grid_search_result, "GridSearch_results")
168     return best
169
170 def _bayesian_search(self, X_train, y_train, param_space):
171     estimator = KerasRegressor(
172         model=create_model,
173         epochs=p.epochs,
174         loss=p.loss_function,
175         shuffle=False,
176         verbose=0,
177     )
178     bayes_search = BayesSearchCV(
179         estimator=estimator,
180         search_spaces=param_space,
181         n_iter=100,
182         scoring=f"neg_{p.loss_function}",
183         n_jobs=-1,
184         cv=TimeSeriesSplit(n_splits=3),
185         refit=True,
186         verbose=2,
187     )
188     bayes_search_result = bayes_search.fit(X_train, y_train)
189
190     best = {

```

```

191         "score": bayes_search_result.best_score_,
192         "params": bayes_search_result.best_params_,
193     }
194
195     save_txt(bayes_search_result, "BayesSearch_results")
196     return best
197
198     def predict(self, X, name="y_pred"):
199         return self.model.predict(X, batch_size=p.batch_size).flatten
200     → ()
201
202     def incremental_train(self, X_train, y_train):
203         if not self.load_weights:
204             print("Model weights were not loaded. Loading latest
205     → weights.")
206             self.model.load_weights(get_latest_weights())
207
208             self.model.fit(
209                 X_train,
210                 y_train,
211                 epochs=p.incremental_epochs,
212                 batch_size=p.batch_size,
213                 shuffle=False,
214                 verbose=1,
215             )
216
217             save_weights(self.model)

```

### A.3 src/model/params.py

```

1 # Market Data
2 ticker = "^GSPC"
3 start = "1983-01-03"
4 end = "2023-09-02"
5 target = "Close"
6 num_features = 6
7
8 # Wavelet Transform
9 wavelet = "coif3"
10 wavelet_mode = "symmetric"
11 levels = 3
12 shrink_coeffs = [False, True, True, True]
13 threshold_mode = "soft"

```

```
14
15 # Model Params
16 sets_sizes = {"train": 0.95, "valdn": 0.025, "test": 0.025}
17 epochs = 2000
18 learning_rate = 0.00018
19 hidden_state_size = 20
20 batch_size = 128
21 time_step = 20
22 loss_function = "mean_squared_error"
23 dropout_rate = 0.12241
24
25 incremental_epochs = 15
```



## REFERÊNCIAS

ABIODUN, O. I.; JANTAN, A.; OMOLARA, A. E.; DADA, K. V. State-of-the-art in artificial neural network applications: A survey. **Nature Medicine**, Heliyon, v. 4, n. 11, p. 673–679, 2018. Disponível em: <<http://doi.org/10.1016/j.heliyon.2018.e00938>>.

ABIODUN, O. I.; JANTAN, A.; OMOLARA, A. E.; DADA, K. V.; UMAR, A. M.; LINUS, O. U.; ARSHAD, H.; KAZAURE, A. A.; GANA, U.; KIRU, M. U. Comprehensive review of artificial neural network applications to pattern recognition. **IEEE Access**, v. 7, p. 158820–158846, 2019. Disponível em: <<http://doi.org/10.1109/ACCESS.2019.2945545>>.

AL-HMOUZ, R.; PEDRYCZ, W.; BALAMASH, A. Description and prediction of time series: A general framework of granular computing. **Expert Systems with Applications**, Elsevier, v. 42, n. 10, p. 4830–4839, 2015. Disponível em: <<http://doi.org/10.1016/j.eswa.2015.01.060>>.

ASSAAD, R. H.; FAYEK, S. Predicting the price of crude oil and its fluctuations using computational econometrics: Deep learning, lstm, and convolutional neural networks. **Econometric Research in Finance**, v. 6, p. 119–137, 09 2021. Disponível em: <<http://doi.org/10.2478/erfin-2021-0006>>.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. **arXiv preprint arXiv:1409.0473**, 2014. Disponível em: <<http://10.48550/arXiv.1409.0473>>.

BARAK, S.; MODARRES, M. Developing an approach to evaluate stocks by forecasting effective features with data mining methods. **Expert Systems with Applications**, Elsevier, v. 42, n. 3, p. 1325–1339, 2015. Disponível em: <<http://doi.org/10.1016/j.eswa.2014.09.026>>.

BIAN, J.; WANG, L.; SCHERER, R.; WOŹNIAK, M.; ZHANG, P.; WEI, W. Abnormal detection of electricity consumption of user based on particle swarm optimization and long short term memory with the attention mechanism. **IEEE Access**, IEEE, v. 9, p. 47252–47265, 2021. Disponível em: <<http://dx.doi.org/10.1109/ACCESS.2021.3062675>>.

BODIE, Z.; KANE, A.; MARCUS, A. **EBOOK: Investments - Global Edition**. [S.l.]: McGraw Hill, 2014. ISBN 9780077161149, 0077161149.

BONDT, W. F. D.; THALER, R. Does the stock market overreact? **The Journal of Finance**, Wiley Online Library, v. 40, n. 3, p. 793–805, 1985. Disponível em: <<http://doi.org/10.1111/j.1540-6261.1985.tb05004.x>>.

BONDT, W. F. D.; THALER, R. H. Do security analysts overreact? **The American economic review**, JSTOR, p. 52–57, 1990.

- BOROVKOVA, S.; TSIAMAS, I. An ensemble of lstm neural networks for high-frequency stock market classification. **Journal of Forecasting**, Wiley Online Library, v. 38, n. 6, p. 600–619, 2019. Disponível em: <<http://doi.org/10.1002/for.2585>>.
- BRONKHORST, A. The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions. **Acta Acustica united with Acustica**, v. 86, p. 117–128, 2000. Disponível em: <[10.3758/s13414-015-0882-9](https://doi.org/10.3758/s13414-015-0882-9)>.
- BUSTOS, S. M.; ANDERSEN, J. V.; MINICONI, M.; NOWAK, A.; ROSZCZYNSKA-KURASINSKA, M.; BRÉE, D. Pricing stocks with yardsticks and sentiments. **Algorithmic Finance**, IOS Press, v. 1, n. 2, p. 183–190, 2011. Disponível em: <<http://doi.org/10.48550/arXiv.1109.6909>>.
- CANALES, M. P. Los índices bursátiles: relevancia del dow jones, s&p 500 y nasdaq en la economía. **Apuntes de Economía y Sociedad**, v. 1, n. 2, p. 11–26, 2020. Disponível em: <<https://doi.org/10.5377/AES.V1I2.11443>>.
- CARVALHO, T. A. C. Um estudo sobre golpes financeiros potencializados pela tecnologia e a relação com a alfabetização financeira no brasil. **Trabalho de Conclusão de Curso (Graduação em Administração)**, Faculdade de Economia, Administração, Contábeis e Atuariais da Pontifícia Universidade Católica de São Paulo, 2022. Disponível em: <<https://repositorio.pucsp.br/jspui/handle/handle/31986>>.
- CATÃO, L. A. V.; PAGÉS, C.; ROSALES, M. F. Financial dependence, formal credit, and informal jobs: New evidence from brazilian household data. **IZA Discussion Paper**, Institute for the Study of Labor (IZA), n. 4609, 2009. Disponível em: <<https://www.econstor.eu/bitstream/10419/36246/1/617481180.pdf>>.
- CHAI, J.; LI, A. Deep learning in natural language processing: A state-of-the-art survey. In: **IEEE. 2019 International Conference on Machine Learning and Cybernetics (ICMLC)**. 2019. p. 1–6. Disponível em: <<http://doi.org/10.1109/ICMLC48188.2019.8949185>>.
- CHEN, Q.; ZHANG, W.; YU, J.; FAN, J. Embedding complementary deep networks for image classification. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [s.n.], 2019. p. 9238–9247. Disponível em: <<http://doi.org/10.1109/CVPR.2019.00946>>.
- CHEN, W.; XIE, D.; ZHANG, Y.; PU, S. All you need is a few shifts: Designing efficient convolutional neural networks for image classification. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [s.n.], 2019. p. 7241–7250. Disponível em: <<http://doi.org/10.48550/arXiv.1903.05285>>.
- CHEN, Y.; HAO, Y. A feature weighted support vector machine and k-nearest neighbor algorithm for stock market indices prediction. **Expert Systems with Applications**, Elsevier, v. 80, p. 340–355, 2017. Disponível em: <<http://doi.org/10.1016/j.eswa.2017.02.044>>.
- CHORTAREAS, G. E.; GARZA-GARCÍA, J. G.; GIRARDONE, C. Financial deepening and bank productivity in latin america. **The European Journal of Finance**, Taylor & Francis, v. 17, n. 9, p. 811–827, 2011. Disponível em: <<https://doi.org/10.1080/1351847X.2010.538512>>.

CHRISTOFFERSEN, P.; JACOBS, K. The importance of the loss function in option valuation. **Journal of Financial Economics**, Elsevier, v. 72, n. 2, p. 291–318, 2004. Disponível em: <<http://doi.org/10.1016/j.jfineco.2003.02.001>>.

CVM. **Relatório da Pesquisa com Vítimas de Fraudes Financeiras**. 2020. Disponível em: <<https://www.gov.br/cvm/pt-br/centrais-de-conteudo/publicacoes/pesquisas/relatorio-pesquisa-fraudes-fin.pdf>>.

DANIEL, K.; HIRSHLEIFER, D.; SUBRAHMANYAM, A. Investor psychology and security market under- and overreactions. **The Journal of Finance**, Wiley Online Library, v. 53, n. 6, p. 1839–1885, 1998. Disponível em: <<http://doi.org/10.1111/0022-1082.00077>>.

DANSO, E. I. Assessing the impact of macroeconomic variables on the performance of the u.s. stock market. 07 2020. Disponível em: <<http://doi.org/10.7176/RJFA/11-14-08>>.

DASTRES, R.; SOORI, M. Artificial neural network systems. **International Journal of Imaging and Robotics (IJIR)**, v. 21, n. 2, p. 13–25, 2021. Disponível em: <<https://hal.science/hal-03349542>>.

DE, I. Understanding the s&p 500: This index offers a lot of international exposure. **Journal of Investment Consulting**, v. 14, n. 2, p. 26–37, 2013. Disponível em: <<https://ssrn.com/abstract=2371340>>.

EFENDI, R.; ARBAIY, N.; DERIS, M. M. A new procedure in stock market forecasting based on fuzzy random auto-regression time series model. **Information Sciences**, Elsevier, v. 441, p. 113–132, 2018. Disponível em: <<http://doi.org/10.1016/j.ins.2018.02.016>>.

FAMA, E. F. Efficient capital markets: A review of theory and empirical work. **The Journal of Finance**, JSTOR, v. 25, n. 2, p. 383–417, 1970. Disponível em: <<http://doi.org/10.2307/2325486>>.

\_\_\_\_\_. Market efficiency, long-term returns, and behavioral finance. **Journal of financial economics**, Elsevier, v. 49, n. 3, p. 283–306, 1998. Disponível em: <[http://doi.org/10.1016/S0304-405X\(98\)00026-9](http://doi.org/10.1016/S0304-405X(98)00026-9)>.

GERVEN, M. van; BEHRENS, T. E. How biological attention mechanisms improve task performance in a large-scale visual system model. **eLife**, National Center for Biotechnology Information, n. 7, 2018. Disponível em: <<https://doi.org/10.7554/eLife.38105>>.

HEIDARY, H.; OSKOU EI, A. R.; HAJIKHANI, M.; MOOSALOO, B.; NAJAFABADI, M. A. Acoustic emission signal analysis by wavelet method to investigate damage mechanisms during drilling of composite materials. In: . [s.n.], 2010. v. 1. Disponível em: <<http://doi.org/10.1115/ESDA2010-24389>>.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, MIT press, v. 9, n. 8, p. 1735–1780, 1997. Disponível em: <<http://doi.org/doi.org/10.1162/neco.1997.9.8.1735>>.

HU, Y.; WONG, Y.; WEI, W.; DU, Y.; KANKANHALLI, M.; GENG, W. A novel attention-based hybrid cnn-rnn architecture for semg-based gesture recognition. **PLOS**

**ONE**, Public Library of Science, v. 13, n. 10, p. 1–18, 2018. Disponível em: <<https://doi.org/10.1371/journal.pone.0206049>>.

JIANG, Q.; TANG, C.; CHEN, C.; WANG, X.; HUANG, Q. Prediction of stock price based on lstm neural network. In: SPRINGER. **Proceedings of the Twelfth International Conference on Management Science and Engineering Management**. 2019. p. 393–408. Disponível em: <<http://doi.org/10.1109/AIAM48774.2019.00113>>.

JIN, Z.; YANG, Y.; LIU, Y. Stock closing price prediction based on sentiment analysis and lstm. **Neural Computing and Applications**, Springer, v. 32, p. 9713–9729, 2020. Disponível em: <<http://doi.org/10.1007/s00521-019-04504-2>>.

KE, J.; DUAN, Y.; XU, C.; ZHANG, Y. Cross-correlation multifractal analysis of technological innovation, financial market and real economy indices. **Fractal and Fractional**, MDPI AG, v. 7, n. 3, p. 267–286, 2023. ISSN 2504-3110. Disponível em: <<http://dx.doi.org/10.3390/fractalfract7030267>>.

KHAN, J. K.; WEI, J. S.; RINGNÉR, M.; SAAL, L. H. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. **Nature Medicine**, Nature, v. 7, p. 673–679, 2001. Disponível em: <<http://doi.org/10.1038/89044>>.

KIM, H. Y.; WON, C. H. Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models. **Expert Systems with Applications**, Elsevier, v. 103, p. 25–37, 2018. Disponível em: <<http://doi.org/10.1016/j.eswa.2018.03.002>>.

KRIEGESKORTE, N.; GOLAN, T. Neural network models and deep learning. **Current Biology**, v. 29, n. 7, p. 231–236, 2019. Disponível em: <<http://doi.org/10.1016/j.cub.2019.02.034>>.

LAM, M. Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. **Decision support systems**, Elsevier, v. 37, n. 4, p. 567–581, 2004. Disponível em: <[http://doi.org/10.1016/S0167-9236\(03\)00088-5](http://doi.org/10.1016/S0167-9236(03)00088-5)>.

LAM, M. W.; HU, S.; XIE, X.; LIU, S.; YU, J.; SU, R.; LIU, X.; MENG, H. Gaussian process neural networks for speech recognition. In: **INTERSPEECH**. [s.n.], 2018. p. 1778–1782. Disponível em: <<http://doi.org/10.21437/Interspeech.2018-1823>>.

LAVINAS, L. How social developmentalism reframed social policy in brazil. **New Political Economy**, Taylor & Francis, v. 22, n. 6, p. 628–644, 2017. Disponível em: <<https://doi.org/10.1080/13563467.2017.1297392>>.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015. Disponível em: <<http://doi.org/10.1038/nature14539>>.

LI, K.; XU, H.; WANG, Y.; POVEY, D.; KHUDANPUR, S. Recurrent neural network language model adaptation for conversational speech recognition. In: **Interspeech**. [s.n.], 2018. v. 2018, p. 3373–3377. Disponível em: <<http://doi.org/10.21437/Interspeech.2018-1413>>.

LIU, H. Leveraging financial news for stock trend prediction with attention-based recurrent neural network. **arXiv preprint arXiv:1811.06173**, 2018. Disponível em: <<http://doi.org/10.48550/arXiv.1811.06173>>.

LIU, S.; ZHANG, C.; MA, J. Cnn-lstm neural network model for quantitative strategy analysis in stock markets. In: SPRINGER. **Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part II 24**. 2017. p. 198–206. Disponível em: <[http://doi.org/10.1007/978-3-319-70096-0\\_21](http://doi.org/10.1007/978-3-319-70096-0_21)>.

LIU, X.; HE, P.; CHEN, W.; GAO, J. Multi-task deep neural networks for natural language understanding. **arXiv preprint arXiv:1901.11504**, 2019. Disponível em: <<http://doi.org/10.48550/arXiv.1901.11504>>.

LO, A. W. The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. **The Journal of Portfolio Management**, v. 30, n. 5, p. 15–29, 2004. Disponível em: <<http://doi.org/10.3905/jpm.2004.442611>>.

MA, B.; LI, X.; XIA, Y.; ZHANG, Y. Autonomous deep learning: A genetic dcnn designer for image classification. **Neurocomputing**, Elsevier, v. 379, p. 152–161, 2020. Disponível em: <<http://doi.org/10.1016/j.neucom.2019.10.007>>.

MALKIEL, B.; MULLAINATHAN, S.; STANGLE, B. Market efficiency versus behavioral finance. **Journal of Applied Corporate Finance**, Wiley Online Library, v. 17, n. 3, p. 124–136, 2005. Disponível em: <<http://doi.org/10.1111/j.1745-6622.2005.00053.x>>.

MALYKH, V. Robust to noise models in natural language processing tasks. In: **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop**. [s.n.], 2019. p. 10–16. Disponível em: <<http://doi.org/10.18653/v1/P19-2002>>.

MNIH, V.; HEES, N.; GRAVES, A. et al. Recurrent models of visual attention. **Advances in neural information processing systems**, v. 27, 2014. Disponível em: <<http://doi.org/10.48550/arXiv.1406.6247>>.

MOUSSAWI, R.; SHEN, K.; VELTHUIS, R. The role of taxes in the rise of etfs. **Social Science Research Network**, 2022. Disponível em: <<http://dx.doi.org/10.2139/ssrn.3744519>>.

MURPHY, J. J. **Technical analysis of the financial markets: A comprehensive guide to trading methods and applications**. [S.l.]: Penguin, 1999. ISBN 9780735200661.

NAVEED, H.; KHAN, A.; QIU, S.; ANWAR, S.; USMAN, M.; BARNES, N.; MIAN, A. A comprehensive overview of large language models. 2023. Disponível em: <<https://doi.org/10.48550/arXiv.2307.06435>>.

NWANKPA, C.; IJOMAH, W.; GACHAGAN, A.; MARSHALL, S. Activation functions: Comparison of trends in practice and research for deep learning. 2018. Disponível em: <<http://doi.org/10.48550/arXiv.1811.03378>>.

OTCHERE, I. K.; GYGAX, A. Do index effects reflect idiosyncratic or industry effects? a re-examination of the winners and losers of s&p 500 index addition. **20th Australasian Finance & Banking Conference**, 2007. Disponível em: <<http://dx.doi.org/10.2139/ssrn.1009290>>.

PATEL, P. B.; MARWALA, T. Forecasting closing price indices using neural networks. In: IEEE. **2006 IEEE international conference on systems, man and cybernetics**. 2006. v. 3, p. 2351–2356. Disponível em: <<http://doi.org/10.1109/ICSMC.2006.385214>>.

PAVONE, P. Market capitalization and financial variables: Evidence from italian listed companies. **International Journal of Academic Research in Business and Social Sciences**, v. 9, 2019.

QIU, J.; WANG, B.; ZHOU, C. Forecasting stock prices with long-short term memory neural network based on attention mechanism. **PLoS ONE**, v. 15, 2020. Disponível em: <<http://doi.org/10.1371/journal.pone.0227222>>.

ROSAS-OREA, M.; HERNANDEZ-DIAZ, M.; ALARCON-AQUINO, V.; GUERRERO-OJEDA, L. A comparative simulation study of wavelet based denoising algorithms. In: **15th International Conference on Electronics, Communications and Computers**. [s.n.], 2012. p. 125–130. Disponível em: <<http://doi.org/10.1109/CONIEL.2005.6>>.

SEDIGHI, M.; JAHANGIRNIA, H.; GHARAKHANI, M.; FARD, S. F. A novel hybrid model for stock price forecasting based on metaheuristics and support vector machine. **Data**, MDPI, v. 4, n. 2, p. 75, 2019. Disponível em: <<http://doi.org/10.3390/data4020075>>.

SHAMIR, G. I.; LIN, D.; COVIELLO, L. Smooth activations and reproducibility in deep networks. 2020. Disponível em: <<http://doi.org/10.48550/arXiv.2010.09931>>.

SHERSTINSKY, A. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. **Physica D: Nonlinear Phenomena**, Elsevier, v. 404, p. 132306, 2020. Disponível em: <<http://doi.org/10.1016/j.physd.2019.132306>>.

SHEWALKAR, A.; NYAVANANDI, D.; LUDWIG, S. A. Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru. **Journal of Artificial Intelligence and Soft Computing Research**, v. 9, n. 4, p. 235–245, 2019. Disponível em: <<http://doi.org/10.2478/jaiscr-2019-0006>>.

SHILLER, R. J. From efficient markets theory to behavioral finance. **Journal of economic perspectives**, American Economic Association, v. 17, n. 1, p. 83–104, 2003. Disponível em: <<http://doi.org/10.1257/089533003321164967>>.

SINGH, S. Pattern modelling in time-series forecasting. **Cybernetics and Systems**, Taylor & Francis, v. 31, n. 1, p. 49–65, 2000. Disponível em: <<http://doi.org/10.1080/019697200124919>>.

SMITH, L. N. A disciplined approach to neural network hyper-parameters: Part 1 — learning rate, batch size, momentum, and weight decay. **CoRR**, 2018. Disponível em: <<http://arxiv.org/abs/1803.09820>>.

- SREENIVASAN, V.; KOLLI, H.; SIALM, C. Does the stock market anticipate economic growth? empirical evidence based on the u.s. stock market. **Journal of Student Research**, v. 11, n. 3, 2022. Disponível em: <<https://www.jsr.org/hs/index.php/path/article/view/3316>>.
- SUDRE, C. H.; LI, W.; VERCAUTEREN, T.; OURSELIN, S.; CARDOSO, M. J. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. **arXiv preprint arXiv.1707.03237**, 2017. Disponível em: <<http://doi.org/10.48550/arXiv.1707.03237>>.
- TURNER, T. **A Beginner's Guide To Day Trading Online 2nd Edition**. [S.l.]: Simon and Schuster, 2007. ISBN 9781593376864.
- VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017. Disponível em: <<http://doi.org/10.48550/arXiv.1706.03762>>.
- WANG DONGJIN YU, C. L. J. Predicting outcomes of business process executions based on lstm neural networks and attention mechanism. 07 2021. Disponível em: <<https://doi.org/10.21203/rs.3.rs-260970/v1>>.
- WEI, L.-Y.; CHEN, T.-L.; HO, T.-H. A hybrid model based on adaptive-network-based fuzzy inference system to forecast taiwan stock market. **Expert Systems with Applications**, Elsevier, v. 38, n. 11, p. 13625–13631, 2011. Disponível em: <<http://doi.org/10.1016/j.eswa.2011.04.127>>.
- XING, J.; LI, J.; YANG, Q.; WANG, R. A new wavelet threshold determination method considering interscale correlation in signal denoising. **Mathematical Problems in Engineering**, 2015. Disponível em: <<http://doi.org/10.1155/2015/280251>>.
- XU, K.; BA, J.; KIROS, R.; CHO, K.; COURVILLE, A.; SALAKHUTDINOV, R.; ZEMEL, R.; BENGIO, Y. Show, attend and tell: Neural image caption generation with visual attention. **Computing Research Repository**, 2015. Disponível em: <<https://doi.org/10.48550/arXiv.1502.03044>>.
- YU, Y.; CAO, J.; ZHU, J. An lstm short-term solar irradiance forecasting under complicated weather conditions. **IEEE Access**, IEEE, v. 7, p. 145651–145666, 2019. Disponível em: <<http://doi.org/10.3390/s23052799>>.
- YU, Y.; SI, X.; HU, C.; ZHANG, J. A review of recurrent neural networks: Lstm cells and network architectures. **Neural Computation**, v. 31, n. 7, p. 1235–1270, 2019. Disponível em: <[https://doi.org/10.1162/neco\\_a\\_01199](https://doi.org/10.1162/neco_a_01199)>.
- YUE, B.; FU, J.; LIANG, J. Residual recurrent neural networks for learning sequential representations. **Information**, MDPI, v. 9, n. 3, p. 56, 2018. Disponível em: <<http://doi.org/10.3390/info9030056>>.
- ZHANG, W.; CUI, X.; FINKLER, U.; KINGSBURY, B.; SAON, G.; KUNG, D.; PI-CHENY, M. Distributed deep learning strategies for automatic speech recognition. In: IEEE. **ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. 2019. p. 5706–5710. Disponível em: <<http://doi.org/10.48550/arXiv.1904.04956>>.

ZHANG, X.; LIANG, X.; ZHIYULI, A.; ZHANG, S.; XU, R.; WU, B. At-lstm: An attention-based lstm model for financial time series prediction. **IOP Conference Series: Materials Science and Engineering**, IOP Science, v. 569, n. 5, 2019. Disponível em: <<http://doi.org/10.1088/1757-899X/569/5/052037>>.

ZHAO, Z.; RAO, R.; TU, S.; SHI, J. Time-weighted lstm model with redefined labeling for stock trend prediction. In: IEEE. **2017 IEEE 29th international conference on tools with artificial intelligence (ICTAI)**. 2017. p. 1210–1217. Disponível em: <<http://doi.org/10.1109/ICTAI.2017.00184>>.