CESAR SCHOOL

LUÍS HENRIQUE CARVALHO DA CRUZ

PREVISÃO DA DIREÇÃO DOS RETORNOS DE AÇÕES DO IBOVESPA COM APRENDIZADO DE MÁQUINA

RECIFE
2025

LUÍS HENRIQUE CARVALHO DA CRUZ

PREVISÃO DA DIREÇÃO DOS RETORNOS DE AÇÕES DO IBOVESPA COM APRENDIZADO DE MÁQUINA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação do Centro de Estudos e Sistemas Educacionais do Recife – CESAR School, como requisito para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Anderson Tenório Sergio

Coorientador: Prof. Me. Manoel Alves de Almeida Neto

RECIFE
2025

Dedicatória

Dedico este trabalho aos meus pais, Claudia Carvalho e Carlos Eduardo Cruz, que em 2003 me presentearam a vida e passaram todos os anos subsequentes garantindo que eu tivesse o poder de escolher o que fazer com ela.

Agradecimentos

Agradeço aos meus orientadores por me guiarem na direção certa no desenvolvimento do trabalho, à minha família, meus pais e minha namorada por terem a paciência, compreensão e apoio pelos dias que me isolei para focar nos experimentos, e aos amigos que fiz durante a graduação pelos 4 anos de trabalho em equipe e troca de conhecimento.

Epígrafe

“*The best thing about being a statistician is that you get to play in everyone's backyard.*” - John Tukey

Resumo

O presente estudo investiga a viabilidade do uso de algoritmos de Aprendizado de Máquina (AM) para prever a direção dos retornos de ações do índice Bovespa. São comparados os modelos *Long Short-Term Memory* (LSTM), *Gated Recurrent Unit* (GRU), Rede Neural Recorrente simples (RNR) e Floresta Aleatória (FA), além de um comitê formado pelos três melhores modelos com base no *Sharpe Ratio*, uma métrica que avalia o retorno por unidade de risco em uma carteira de investimentos. Os modelos são comparados com uma estratégia Buy and Hold (B&H) e um modelo aleatório (MA). As acurácias observadas giram em torno de 50%, sugerindo dificuldade dos modelos em identificar padrões nos dados. No entanto, os modelos LSTM, GRU e o comitê apresentaram retornos acumulados significativamente superiores ao modelo aleatório, com o comitê demonstrando maior estabilidade. Em experimentos pontuais, os modelos também superaram o B&H em grande parte da janela temporal, indicando potencial para aplicações futuras. Conclui-se que, nas condições testadas, apesar de as técnicas utilizadas superarem uma abordagem aleatória, a estratégia B&H ainda se mostra mais vantajosa no longo prazo.

Palavras-chave: Redes Neurais Recorrentes; LSTM; GRU; Floresta Aleatória; Ibovespa; Séries Temporais Financeiras.

Abstract

This study investigates the feasibility of using Machine Learning (ML) algorithms to predict the direction of stock returns in the Bovespa index. The models compared include Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Simple Recurrent Neural Network (RNN), and Random Forest (RF), as well as an ensemble composed of the three best-performing models based on the Sharpe Ratio, a metric that evaluates return per unit of risk in an investment portfolio. The models are compared to a Buy and Hold (B&H) strategy and a random model (RM). The observed accuracies hover around 50%, suggesting that the models struggle to identify patterns in the data. However, the LSTM, GRU, and ensemble models achieved cumulative returns that were significantly higher than those of the random model, with the ensemble demonstrating greater stability. In specific experiments, the models also outperformed the B&H strategy for most of the tested time window, indicating potential for future applications. It is concluded that, under the tested conditions, although the techniques used surpass a random approach, the B&H strategy still appears more advantageous in the long run.

Keywords*: Recurrent Neural Networks; LSTM; GRU; Random Forest; Ibovespa; Financial Time Series.*

SUMÁRIO

Lista de imagens

Lista de tabelas

Abreviaturas

B&H - *Buy and Hold*

HME - Hipótese dos Mercados Eficientes

AM - Aprendizado de Máquina

RNA - Redes Neurais Artificiais

RNR - Redes Neurais Recorrentes

LSTM - *Long Short-Term Memory*

GRU - *Gated Recurrent Unit*

FA - Floresta Aleatória

MA - Modelo Aleatório

1. Introdução

De acordo com Fama (1970), a Hipótese dos Mercados Eficientes (HME) afirma que o preço de um ativo na bolsa de valores, como ações de uma empresa, em determinado ponto no tempo, reflete toda a informação disponível sobre ele mesmo. Portanto, toda negociação de tal ativo está a um preço justo, não sendo possível superar o mercado consistentemente. Em uma revisão da HME, Burton Malkiel reforça o argumento de que é impossível utilizar movimentos passados de ativos para obter retornos excessivos em operações na bolsa de valores, argumentando que não existem padrões que possam gerar oportunidades de operações lucrativas, e mesmo se houvessem, após tornados públicos para o conhecimento geral, não permitiriam mais a obtenção de lucro (Malkiel, 2003). Outros autores como Fjellstrom (2022) e Nelson, Pereira e Oliveira (2017) reforçam a dificuldade de se trabalhar com previsão de dados de ativos financeiros, caracterizando seu comportamento como não-linear, ruidoso, dinâmico e caótico.

Entretanto, conforme novas tecnologias são desenvolvidas, os acadêmicos procuram maneiras de utilizá-las para tentar impor uma contra-prova à HME. Uma das formas que se utiliza para tentar atingir esse objetivo é através do Aprendizado de Máquina (AM), que é “a ciência (e a arte) da programação de computadores de modo que eles possam aprender com os dados” (Géron, 2021, p. 3) e Redes Neurais Artificiais (RNA), uma arquitetura de AM inspirada no funcionamento de neurônios biológicos (Goodfellow, Bengio e Courville, 2016, p. 165), na tentativa de extrair padrões dos movimentos passados dos ativos. O trabalho de Tang et al. (2022) realiza uma revisão da literatura sobre a aplicação de modelos de AM a séries temporais financeiras como séries de preços de ativos no mercado, e demonstra um número crescente de artigos publicados em fontes especializadas como *Expert Systems with Applications, IEEE Access e Applied Soft Computing* no período entre 2011 e 2021, indo de menos de 5 em 2011 e passando por mais de 40 no ano de 2020. Além de Tang et al., outros autores como Chen et al. (2023) destacam o interesse crescente na literatura entre 2015 e 2023 em aplicações com AM, RNA e modelos híbridos que são compostos por mais de uma técnica de AM na tentativa de extrair informações dos dados.

De acordo com Gao, Zhang e Yang (2020) e Caride, Bariviera e Lanzarini (2018), as RNA apresentam uma melhor capacidade de explicar relacionamentos não lineares entre variáveis, e por isso vem ganhando atenção em tarefas de previsão de dados financeiros complexos. Além disso, as Redes Neurais Recorrentes (RNR), que são uma extensão das RNA especializadas em dados sequenciais (Goodfellow, Bengio e Courville, 2016) e suas variantes como a rede *Long Short-Term Memory* (LSTM), uma extensão da RNR melhorada para dependências temporais de longo prazo (Hochreiter e Schmidhuber, 1997), vem se destacando na academia. Tang et al. (2022), por exemplo, observou a LSTM como o modelo mais frequentemente utilizado na literatura revisada.

Com o desenvolvimento de novas arquiteturas de AM, a quantidade de algoritmos disponíveis para estudo e aplicação aumenta. Somente a revisão conduzida por Tang et al. (2022) menciona 13 algoritmos incluindo LSTM e a Floresta Aleatória (FA), que é um modelo de AM que combina a decisão majoritária de vários modelos “fracos” para formar um decisor “forte” (Breiman, 2001). Fjellstrom (2022) e Nelson, Pereira e Oliveira (2017) utilizaram a arquitetura LSTM com ações do índice OMX30 de Estocolmo e o índice Bovespa respectivamente, na tentativa de prever a direção dos retornos diários. Fjellstrom utilizou um comitê de LSTMs, sendo um comitê formado pela união da resposta de mais de um modelo que separadamente resolvem uma tarefa específica (Villanueva, 2006), e Nelson, Pereira e Oliveira compararam a LSTM com o Perceptron Multicamadas (*Multi-Layer Perceptron*), que é a forma mais básica de RNA (Goodfellow, Bengio e Courville, 2016), além da FA e um modelo pseudo-aleatório. Krauss, Do e Huck (2017) compararam diferentes arquiteturas de AM, incluindo FA e RNA, para prever retornos diários de ações do índice americano S&P 500, e extraíram comitês de previsores a partir dos melhores 3 modelos. Chavhan et al. (2024) realizaram uma comparação entre LSTM, RNR, e *Gated Recurrent Unit* (GRU), o último sendo uma arquitetura para dados sequenciais com dependência de longo prazo similar à LSTM (Cho et al., 2014), para prever os preços de ações no mercado americano. Na literatura internacional, costuma-se utilizar dados de bolsas historicamente

consolidadas e de alto volume, como a americana S&P 500, sobrando espaço para um aprofundamento da pesquisa na bolsa brasileira, a B3[1].

A pesquisa presente busca aplicar a LSTM, GRU, RNR simples e FA na previsão da direção dos retornos de um subconjunto de ações do índice Bovespa. A previsão diária será conduzida na forma de uma classificação binária positiva ou negativa, sendo positiva caso o modelo preveja um retorno maior que 0, e negativa no caso contrário. O desempenho dos modelos será comparado por meio de acurácia, retornos acumulados das carteiras geradas, e do *Sharpe Ratio*, uma métrica de avaliação de portfólio que mede o retorno excessivo por unidade de risco (Krauss, Do e Huck, 2017). A metodologia aplicada é similar a Fjellstrom (2022) e Krauss, Do e Huck (2017). As carteiras de investimentos hipotéticas geradas pelos modelos serão construídas a partir de um sinal de compra ou venda diária caso o modelo preveja um retorno positivo ou negativo no dia seguinte.

Os modelos serão comparados entre si, contra um Modelo Aleatório (MA) de benchmark que atribui a cada dia uma classe positiva ou negativa com probabilidades iguais, e contra uma estratégia passiva de *Buy and Hold* (B&H), na qual o investidor hipotético mantém as ações compradas no decorrer de todo o tempo. Além disso, será avaliado se um comitê dos melhores 3 modelos com base no *Sharpe Ratio* atinge uma performance mais satisfatória do que seus constituintes individuais. As observações acerca da diferença entre as performances serão verificadas através de testes de hipótese estatísticos.

1.1. Objetivos

Objetivo Geral:

- Comparar os modelos LSTM, GRU, RNR simples e FA na previsão da direção dos retornos de um subconjunto de ações do índice Bovespa.

Objetivos Específicos:

- Comparar os modelos pela acurácia, retornos acumulados das carteiras, e pelo *Sharpe Ratio*.
- Extrair um comitê a partir dos 3 modelos que atingirem as top 3 performances médias em *Sharpe Ratio* e compará-lo com os outros modelos individuais.

[1] B3 - *Brasil, Bolsa, Balcão*. Disponível em: https://www.b3.com.br/pt_br/institucional. Acesso em 03 mar. 2025.

- Avaliar a performance dos modelos treinados contra uma estratégia B&H e contra um MA que atribui uma classe positiva ou negativa a cada passo com probabilidades iguais.
- Verificar diferenças entre os retornos acumulados das carteiras geradas por cada modelo utilizando testes de hipótese estatísticos.

1.2. Estrutura do Trabalho

No tópico 2 serão apresentados os conceitos utilizados no decorrer do trabalho, incluindo as arquiteturas de AM utilizadas e métricas de avaliação. No ponto 3 será discutida a metodologia utilizada para o trabalho, como foram conduzidos os experimentos e quais escolhas foram feitas. No ponto 4, será realizada a discussão dos resultados obtidos. Por fim, no tópico 5, o trabalho será concluído e serão apresentadas as ameaças à validade da pesquisa e trabalhos futuros.

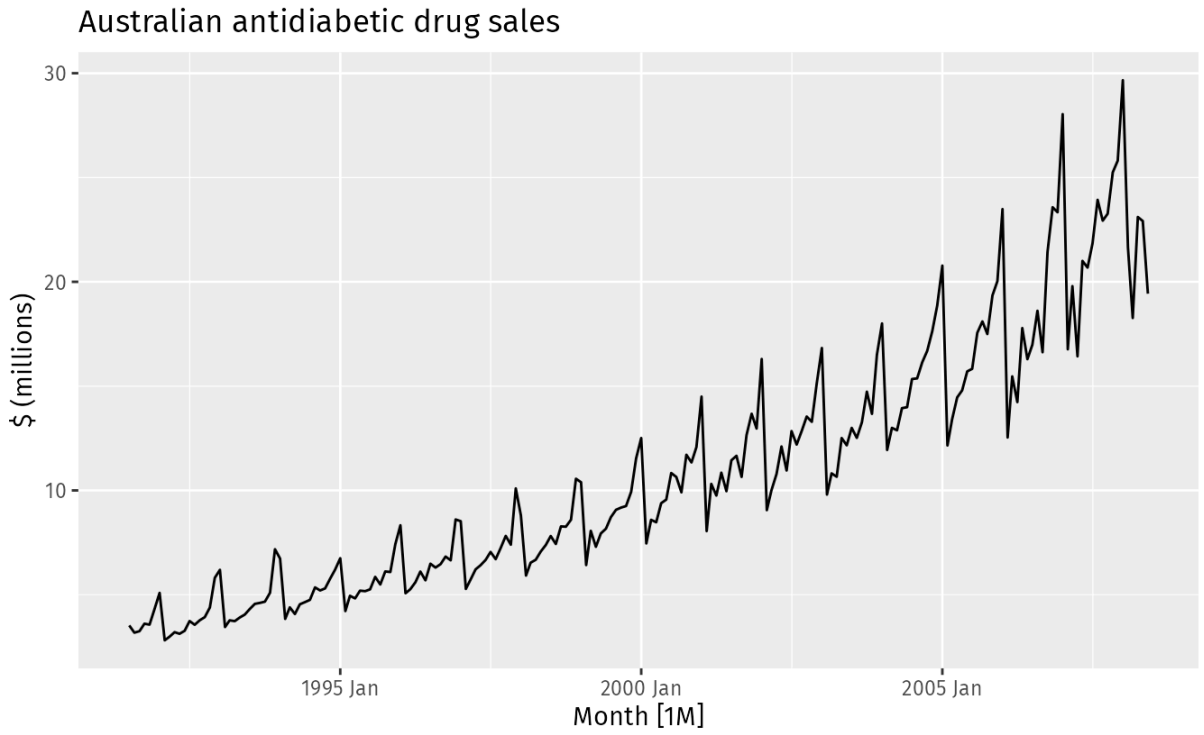
2. Fundamentação Teórica

Esse tópico irá fundamentar o essencial para a compreensão do desenvolvimento do trabalho, passando por séries temporais, AM, RNA, as arquiteturas selecionadas e métricas de performance utilizadas.

2.1 Séries Temporais

De acordo com Hyndman e Athanasopoulos (2018), séries temporais são qualquer tipo de dado observado sequencialmente. Como exemplos estão os lucros anuais da Google, resultados trimestrais da Amazon, índices de chuvas mensais e vendas de remédios para diabetes na Austrália, como demonstrado na imagem 1. Já a análise desse tipo de dado, de acordo com Nielsen (2021), se resume a responder à questão: “como o passado influencia o futuro?".

Imagem 1 - Vendas mensais de remédios para diabetes na Austrália



Fonte: Hyndman e Athanasapoulos (2018)

Para responder essa pergunta, utilizam-se técnicas de previsão que variam de modelos estatísticos de 100 anos de idade até o uso de Aprendizado de Máquina (Nielsen, 2021). Hyndman e Athanasopoulos (2018) descrevem que a tarefa de previsão de séries temporais tem como objetivo estimar como uma sequência de observações vai continuar no futuro. Os mesmos autores afirmam que a forma mais simples de previsão utiliza apenas o passado da própria variável que se busca estimar, sem incluir fatores externos que podem interferir no comportamento de uma série temporal, como em um exemplo de vendas: mudanças econômicas, iniciativas de marketing e atividades da concorrência.

De volta a Nielsen (2021), a autora afirma que no mercado financeiro, o armazenamento minucioso de registros acerca dos ativos permitiu que os participantes da bolsa de valores pudessem ganhar dinheiro nos mercados por meio da matemática e da estatística, e mais recentemente pelo AM. Além disso, ela afirma que o mercado financeiro é o avô dos dados de séries temporais, e que a análise desse tipo de dado pode acontecer em diferentes janelas de tempo, desde meses, dias e horas, como fazem as firmas tradicionais, até milissegundos com o advento do *Trading* de Alta Frequência (*High Frequency Trading*).

2.2 Aprendizado de Máquina

De acordo com Mitchell (1997), diz-se que um programa de computador aprende com uma dada experiência E em respeito a uma classe de tarefas T, e métrica de performance P, se sua performance em tarefas T, medida por P, melhora com a experiência E. Em palavras mais simples, de acordo com Géron (2021, p. 3), o Aprendizado de Máquina é uma forma de programar computadores de modo que eles possam aprender com os dados. Utiliza-se AM em problemas muito complexos para a programação tradicional ou problemas que não tem um algoritmo conhecido. Morettin e Bussab (2023) estendem a definição explicando a diferença entre programação clássica e AM: enquanto no primeiro se utiliza uma regra (programa) juntamente com os dados a serem processados e constrói-se uma resposta, o segundo constrói a regra a partir de dados e das respostas esperadas do processamento desses dados.

Uma das tarefas do AM é a de classificação, em que, de acordo com Morettin e Singer (2022), o computador recebe um conjunto de dados com n exemplos na forma (x_i, y_i), $i = 1, \ldots, n$, sendo x_i um vetor p-dimensional representando os valores de p variáveis preditoras, e y_i representa o valor de uma variável de resposta (classe) da classe a qual o i-ésimo elemento do conjunto pertence. Ainda de acordo com os autores, o objetivo é que a partir dos exemplos fornecidos, construa-se um algoritmo capaz de atribuir uma classe a um elemento identificado pelo vetor de variáveis preditoras ainda não vistos no conjunto de exemplos. Em um exemplo de classificação binária, Morettin e Singer (2022) define então um classificador como uma função g capaz de mapear um vetor de p variáveis preditoras em um elemento de um conjunto binário, no formato $g: X \rightarrow Y$ sendo X um vetor de valores pertencentes aos números reais p-dimensionais, formalmente $X \in \Re^p$, e Y um elemento pertencente a um conjunto binário de quaisquer duas possibilidades como $Y \in \{-1, 1\}$.

Um dos pontos de atenção ao treinar algoritmos de AM é o problema de *overfitting* e *underfitting*. Bashir et al. (2020) explica que o primeiro acontece quando o algoritmo reduz o erro através da memorização dos exemplos de treinamento, sem conseguir aprender a verdadeira relação geral entre as entradas e os rótulos. Já o

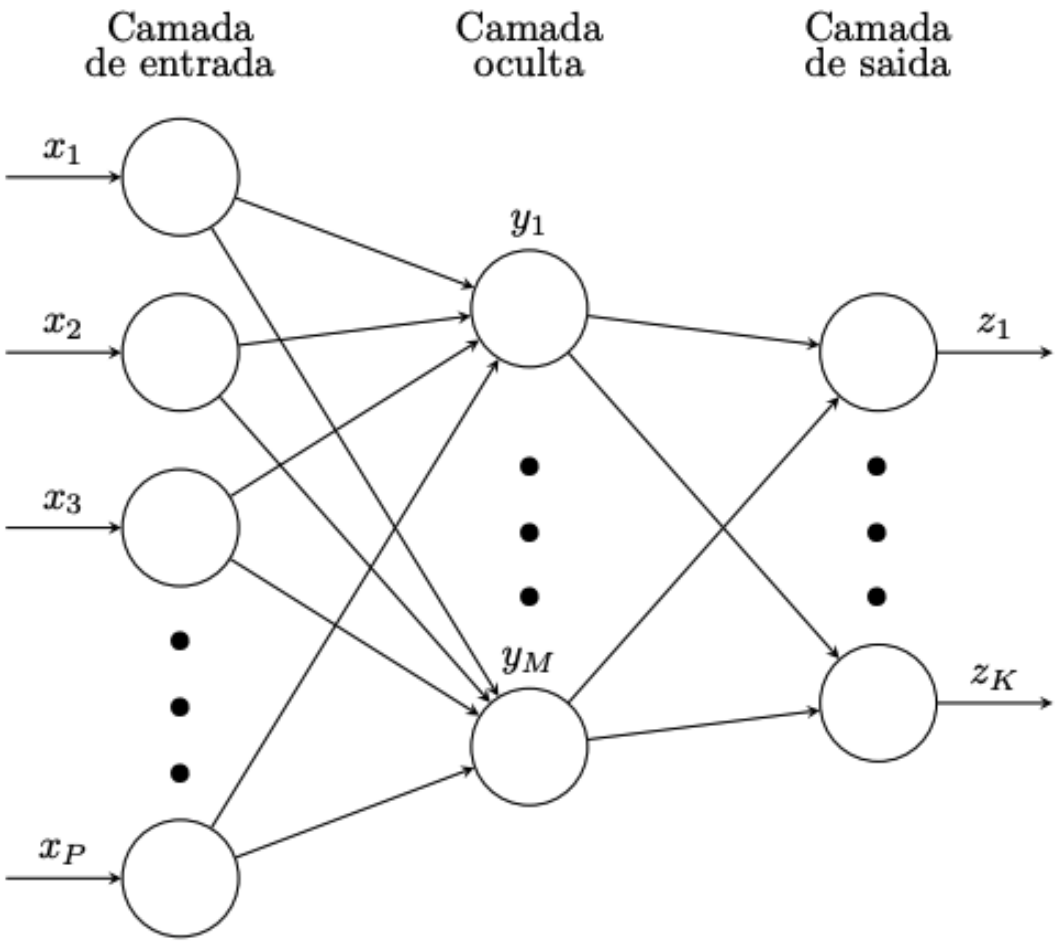
underfitting acontece quando o modelo é incapaz de aprender a relação entre as entradas e os rótulos.

2.3 Redes Neurais Artificiais

Seguindo a definição de Morettin e Singer (2022), uma Rede Neural identifica relações entre variáveis por intermédio de um processo que tenta imitar a maneira como os neurônios interagem no cérebro. De acordo com Goodfellow, Bengio e Courville (2016), as Redes Proalimentadas (*Feedforward Networks*), formam a fundação de arquiteturas de RNA. Elas se baseiam em uma camada de entrada, uma de saída e no mínimo uma camada intermediária chamada de camada escondida. O objetivo dessa arquitetura é aproximar alguma função f^* a partir da combinação de funções definidas em cada camada da rede, de modo a formar um mapeamento $y = f(x; \theta)$ em que θ são parâmetros aprendidos durante o treinamento para melhor aproximar f^*, f é a função aproximada, x são as variáveis de entrada e y são as variáveis de saída.

Cada camada de uma rede define uma função vetor-para-vetor, e é constituída de neurônios, ou unidades, em paralelo que individualmente recebem como entrada um vetor, e computam seu valor de ativação através de uma função de ativação, formando uma função vetor-para-escalar. As unidades escondidas podem conter funções de ativação diferentes, e a escolha do tipo de unidade nas camadas escondidas que irá render melhores resultados é um problema difícil de prever durante o desenho de uma rede. Uma Rede Proalimentada simples de uma camada escondida é demonstrada na imagem 1, em que x_i, $i = 1, \ldots, p$ são as entradas, y_i, $i = 1, \ldots, M$ são os valores de ativação da camada escondida e z_i, $i = 1, \ldots, K$ são os valores de ativação da camada de saída.

Imagem 2 - Rede Proalimentada



Fonte: Morettin e Singer (2022)

Segundo Morettin e Singer (2022), o treinamento de RNAs se dá pela Retropropagação, um algoritmo de otimização numérica que usa o gradiente de uma função de erro em respeito a uma escolha de pesos para decidir em que direção atualizar os pesos de modo a minimizar o erro. O uso do gradiente para atualizar os pesos se chama método do decréscimo do gradiente. A fórmula da atualização de pesos se dá na fórmula 1, em que $w^{(r)}$ é a escolha de pesos na iteração r, $Q(w)$ é uma função de erro da RNA calculada com base nos exemplos de treinamento para uma escolha de w, λ é a taxa de aprendizado que escala os valores do gradiente para atualizar w, e $\nabla\ Q(w^{(r)})$ é o gradiente de Q em respeito aos valores de $w^{(r)}$.

$$w^{(r+1)} = w^{(r)} - \lambda \nabla\ Q(w^{(r)}) \quad (1)$$

Nielsen (2015) estende a explicação, mostrando que uma passagem completa pelo conjunto de dados de treinamento se chama uma época de treino. Ao final de cada época de treino, é aplicado o algoritmo de Retropropagação com base na função de erro recém-calculada com a atual escolha de pesos. O número máximo de épocas de treinamento é uma escolha proposital no desenho de uma RNA.

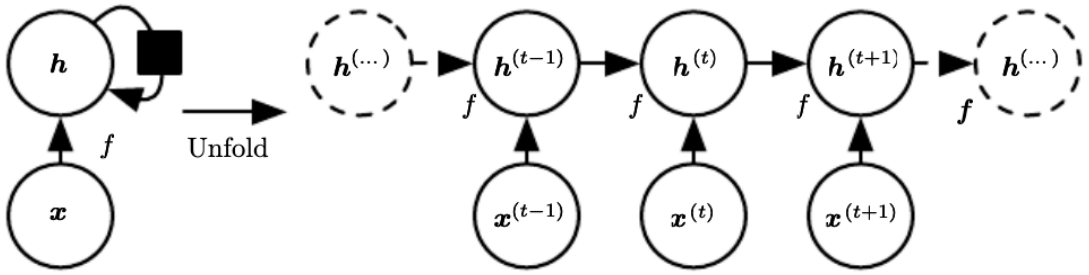
Prechelt (1997) explica que algoritmos de RNAs são suscetíveis ao *overfitting*, em que conforme o treinamento prossegue, o erro nos exemplos de treinamento decresce até que começa a aumentar nos exemplos não vistos. Existem várias formas de evitar que isso aconteça, e uma delas é através da parada antecipada. A parada antecipada essencialmente encerra o treinamento da RNA quando o erro em exemplos não vistos começa a subir. O autor ainda argumenta que essa técnica é amplamente adotada devido à simplicidade de entendimento e implementação. A seleção de um critério de parada antecipada envolve uma troca entre tempo de treinamento e erro de generalização, pois parar muito cedo pode impedir que o treinamento atinja o mínimo global da função de erro, e parar tarde demais pode aumentar consideravelmente o tempo de treinamento.

2.4. Redes Neurais Recorrentes

Seguindo a definição de Goodfellow, Bengio e Courville (2016), Redes Neurais Recorrentes são uma família de RNAs para processamento de dados sequenciais. Os autores explicam que enquanto Redes Proalimentadas tem o fluxo da informação estritamente da entrada até a saída da rede, RNRs possuem conexões que permitem o fluxo retroalimentado, em que uma unidade flui informação de volta a si própria. RNRs permitem que uma escolha de parâmetros seja compartilhada entre os valores de uma sequência, permitindo generalizar e estender a rede para exemplos de sequências de diferentes comprimentos.

Goodfellow, Bengio e Courville (2016) explicam o compartilhamento de parâmetros através do conceito do desdobramento (*unfold*) de uma unidade recorrente. A imagem 2 demonstra o desdobramento de uma unidade h que recebe uma sequência de entrada x de tamanho finito. O desdobrar de h então se entende como uma RNA de profundidade equivalente ao tamanho de x, em que h no índice temporal t recebe como entrada o valor de x^t e a saída de h^{t-1}.

Imagem 3 - Desdobramento de uma unidade recorrente



Fonte: Goodfellow, Bengio e Courville (2016)

2.5. *Long Short-Term Memory*

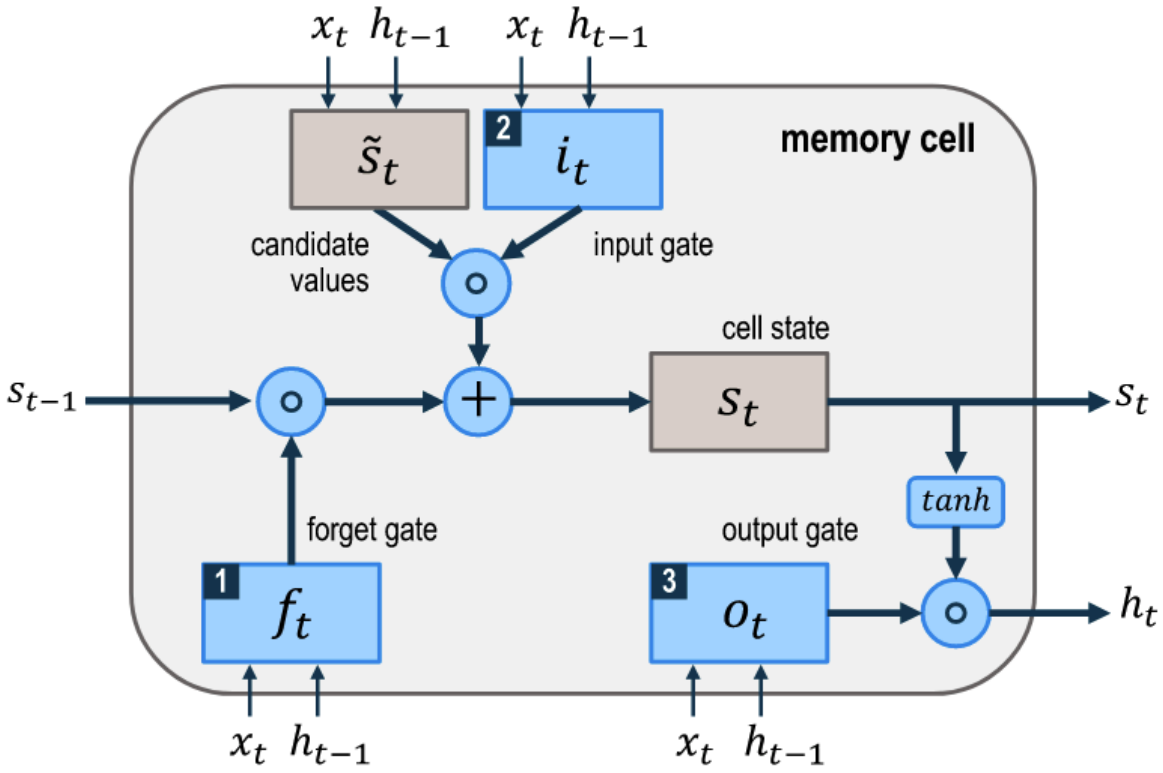
Hochreiter e Schmidhuber (1997) propuseram a LSTM (*Long Short-Term Memory*) como uma nova arquitetura para resolver um problema no treinamento de RNR baseado no gradiente. A dificuldade do treinamento de RNR acontece em exemplos de sequências muito longas, que interferem com o algoritmo de Retropropagação, causando a atualização de pesos a decrescer ou aumentar exponencialmente com o tamanho da sequência. Esse problema é chamado de gradiente que explode ou desaparece (*exploding or vanishing gradients*), em que a explosão torna o treinamento instável e o desaparecimento do gradiente torna a convergência do algoritmo impossível.

O trabalho de Hochreiter e Schmidhuber (1997) propõe uma arquitetura alternativa que reforça a propagação do erro através do tempo com um fluxo contínuo que permeia todas as unidades de memória, essencialmente impedindo que o gradiente exploda ou desapareça. Fischer e Krauss (2018) explicam o fluxo contínuo como o estado da célula (*cell state*), e a unidade de memória é uma unidade recorrente que atualiza o estado da célula e computa sua própria saída com base em 3 portas (*gates*):

- *Forget gate*: uma porta que define quanta informação remover do estado da célula.
- *Input gate*: uma porta que define quanta informação adicionar ao estado da célula.
- *Output gate*: uma porta que utiliza o estado da célula para definir qual informação será a saída recorrente da unidade de memória.

As 3 portas utilizam a informação da saída anterior da célula de memória recorrente, denotada por h_{t-1} e os valores do vetor de entrada x no tempo t, denotado por x_t. A imagem 3 representa uma unidade de memória em que s_t é o estado da célula no índice temporal t, x_t é o valor da sequência no índice temporal t, h_t é a saída da unidade de memória no tempo t, f_t, i_t e o_t representam as portas *forget*, *input*, e *output*, no tempo t, e $\tilde{s}t$ são os valores candidatos a serem incorporados no estado da célula.

Imagem 4 - Célula de memória da LSTM



Fonte: Fischer e Krauss (2018)

2.6. *Gated Recurrent Unit*

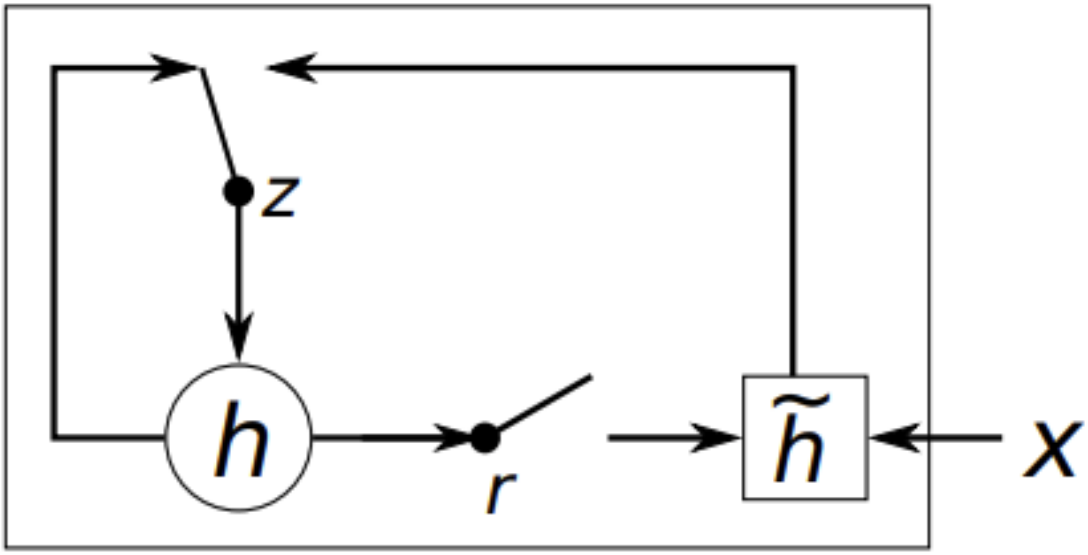
Ravanelli et al. (2018) define a arquitetura GRU (*Gated Recurrent Unit*) como uma simplificação da LSTM com menos portas (*gates*), para melhor eficiência computacional. Os autores afirmam que a LSTM, apesar da sua eficácia em resolver os problemas do treinamento de RNRs, resulta em um modelo muito complexo.

Cho et al. (2014) e Ravanelli et al. (2018) explicam que a GRU dispensa o estado da célula e utiliza apenas duas portas:

- *Reset gate*: decide quanto da saída recorrente anterior será ignorado.
- *Update gate*: decide o quanto da saída recorrente anterior será incorporada na saída da célula atual.

A imagem 4 mostra de forma simplificada o funcionamento de uma célula recorrente GRU, em que z é a função que descreve a porta *update*, r representa a porta *reset*, x são os dados de entrada, h é a saída recorrente e $\tilde{h}$ é a saída recorrente candidata a ser atualizada.

Imagem 5 - Célula recorrente GRU



Fonte: Cho et al. (2014)

2.7. Comitê de Modelos

Dietterich (2000) explica o conceito de comitês (*ensembles*) como algoritmos de aprendizado que constroem um conjunto de classificadores e tomam uma decisão a partir de um voto ponderado das classificações individuais. O autor descreve 3 razões para o sucesso de comitês em comparação a classificadores individuais:

1. Se um algoritmo de aprendizado pode ser visto como uma busca em um espaço de hipóteses H para identificar aquela que rende a melhor acurácia com os dados de treinamento, pode existir mais de uma hipótese em H que

atinge um mesmo nível de acurácia. Portanto, unir a decisão de diferentes classificadores pode mitigar o risco de escolher uma hipótese errada.

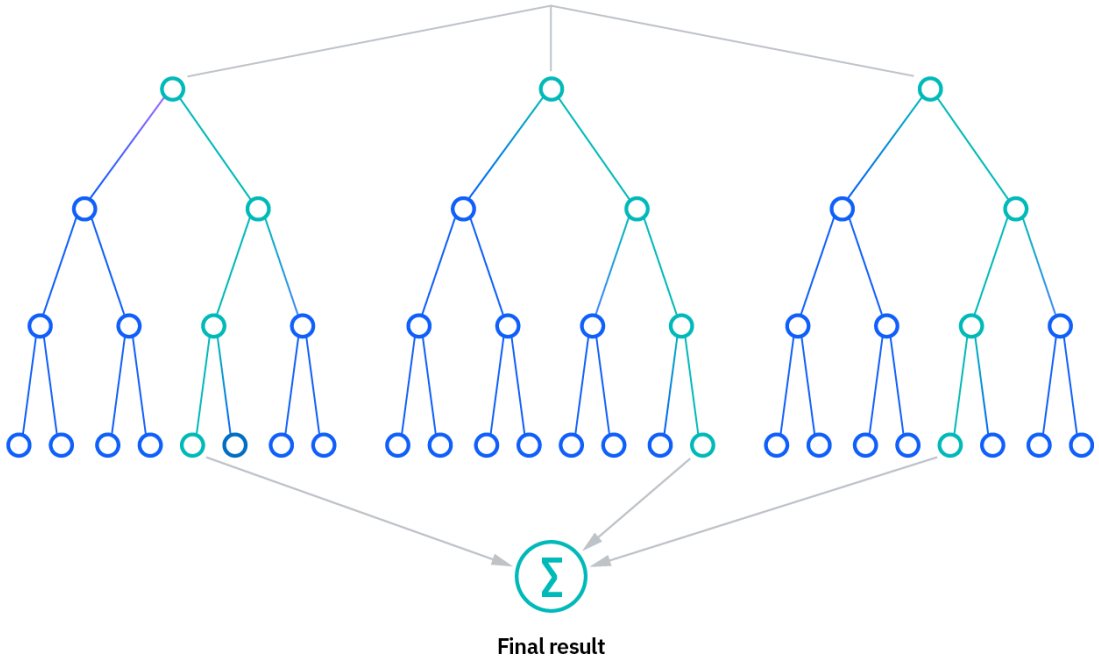
2. O ponto de partida para início de um algoritmo de aprendizado pode influenciar sua convergência através de fatores como a aleatoriedade, impedindo sua convergência, como no treinamento de uma RNA que pode estacionar em um ótimo local na otimização da função de perda. A união de vários classificadores iniciando em pontos de partida diferentes pode mitigar o problema de selecionar um único classificador que não representa bem a hipótese desejada.
3. A função real que se deseja aproximar através de um classificador pode não ser representada por nenhuma das hipóteses em H. A união ponderada de diferentes hipóteses em H pode expandir o espaço de funções representáveis para incluir a função real desejada.

Ganaïe et al. (2022) explicam o conceito de *Bagging*, uma das formas de construir comitês. A ideia é realizar amostragem aleatória do conjunto de dados de treinamento com ou sem reposição e alimentá-los aos classificadores, para então tirar um voto ponderado das suas decisões. Os autores ainda argumentam, seguindo o raciocínio de Dietterich (2000), que uma das condições necessárias para o sucesso de comitês é que seus classificadores sejam acurados e diversos. Ser acurado quer dizer que a taxa de erro é melhor do que adivinhação aleatória em novos exemplos, e ser diverso se refere aos erros dos classificadores não terem correlação.

2.8. Floresta Aleatória

O algoritmo Floresta Aleatória foi proposto por Breiman (2001), e se trata da combinação de árvores de decisão treinadas individualmente com uma amostra aleatória e independente dos exemplos de treinamento, utilizando o conceito de *Bagging*. As árvores classificam novas instâncias com base no que foi aprendido com o subconjunto de exemplos de treinamento, e os votos têm pesos iguais. A imagem a seguir demonstra, em alto nível, como as árvores colaboram paralelamente na tomada da decisão final.

Imagem 6 - Demonstração de uma Floresta Aleatória



Fonte: IBM (s.d.)

Dietterich (2000) argumenta que árvores de decisão originais são algoritmos instáveis, pois alterações nos dados influenciam facilmente a decisão do algoritmo. Breiman (2001) explica que as FAs mitigam a instabilidade de árvores de decisão ao usar *Bagging* e tirar o voto majoritário. O *Bagging* limita a influência de *outliers* às árvores que foram treinadas com o subconjunto de dados em que eles estão presentes, de modo que o voto majoritário é menos suscetível à influência de dados extremos por incluir árvores que não viram esses dados em treinamento.

2.9. Métricas de Avaliação

A presente seção da fundamentação teórica irá explicar as métricas utilizadas para comparar os modelos, nomeadamente a acurácia de classificação e a métrica *Sharpe Ratio* para avaliação de portfólio.

2.9.1. Acurácia

A definição de acurácia segue Nelson, Pereira e Oliveira (2017) de acordo com a fórmula 2, onde TP é a taxa de verdadeiros positivos, TN a de verdadeiros negativos, e FP e FN são as taxas de falso positivo e falso negativo. A métrica calcula a proporção de classificações corretas entre todas as classificações realizadas.

$$Acurácia = \frac{tp + tn}{tp+tn+fp+fn} \quad (2)$$

2.9.2. *Sharpe Ratio*

O *Sharpe Ratio* segue a definição de Sharpe (1994). A métrica desenvolvida por William Sharpe mede o retorno excessivo alcançado por um portfólio em comparação com um retorno benchmark, ajustado à volatilidade do retorno excessivo. O cálculo do *Sharpe Ratio* segue as fórmulas a seguir. O termo D_t é a diferença entre o retorno do portfólio F e do benchmark B no tempo t:

$$D_t = R_{Ft} - R_{Bt} \quad (3)$$

O termo D_μ constitui a média das diferenças dos retornos:

$$D_\mu = \frac{1}{T}\Sigma^{T}_{t=1}(D_t) \quad (4)$$

Então o *Sharpe Ratio* é definido na equação 7, em que σ_D é o desvio padrão das diferenças entre os retornos em cada ponto t:

$$Sharpe\ Ratio = \frac{D_\mu}{\sigma_D} \quad (5)$$

3. Metodologia

O tópico atual irá detalhar como os experimentos foram preparados, conduzidos e em que ambiente de hardware e software foram executados, com objetivo de promover a reprodutibilidade. O código-fonte está disponível em: https://github.com/lhckb/tcc.

3.1. Ambiente

A máquina utilizada contém uma CPU Intel Core i9 de 8 núcleos a 2.3GHz, junto a 16GB de memória RAM. Para programar os experimentos, foram utilizados o Python 3.12 (Python Software Foundation, 2023), Pandas 2.2.3 (Pandas Development Team, 2024) Scikit-learn 1.5.2 (Scikit-learn Developers, 2024), NumPy 2.1.2 (NumPy Developers, s.d.) e Tensorflow 2.16 (Tensorflow Team, 2024) no sistema operacional macOS 15.

3.2. Dados

As 5 ações escolhidas foram ABEV3 (Ambev), BBDC3 (Banco Bradesco), ITSA3 (Itaúsa), ITUB3 (Itaú Unibanco) e WEGE3 (Weg). Elas foram selecionadas com base em dois critérios: maior valor de mercado no dia da coleta (05/10/2024) e não serem empresas estatais. O critério de não-estatal foi adicionado para evitar ações que sofrem com a volatilidade política no país, como acontece com empresas como Petrobras e Vale (Julião, 2023; Machado e Gil, 2024; Yazbek, 2021), reconhecendo que a exclusão de ações pode descartar ativos que exibem padrões exploráveis por algoritmos de AM.

A coleta foi feita na plataforma Investopedia[2], uma plataforma de informações sobre finanças e investimentos, que permite o download de dados históricos de ativos de diferentes bolsas, incluindo a B3. As 5 séries de dados originais contém as características: data, preço no fim do dia, preço no começo do dia, preço máximo do dia, preço mínimo do dia, volume de transações e mudança percentual do preço de fechamento relativo ao dia anterior. Desses mencionados, apenas o preço de fechamento diário foi utilizado.

3.3. Pré-processamento

As séries de dados foram truncadas para manter-se entre as datas 04/10/2018 e 04/10/2024, totalizando um intervalo de 6 anos. Contudo, não há registros para todos os dias nesse intervalo, pois a fonte inclui apenas informações nos dias úteis, em que a bolsa está operante.

Após seleção do intervalo temporal e característica do preço de fechamento, as séries foram concatenadas no eixo temporal, formando um único Dataframe Pandas[3] e revelando as observações ausentes. A série de preços ABEV3 e WEGE3 continham apenas 1 observação ausente, e todas as outras continham 0 ausentes. Visto que os intervalos temporais sem dados eram de apenas 1 dia, foi utilizada uma interpolação linear para preservar tendências que pudessem existir na série original.

Após preencher os valores faltantes, a série de preços foi transformada em uma série de retornos seguindo a definição de Krauss, Do e Huck (2017) e Fjellstrom (2022), especificado na fórmula 6. Justifica-se essa transformação pelo

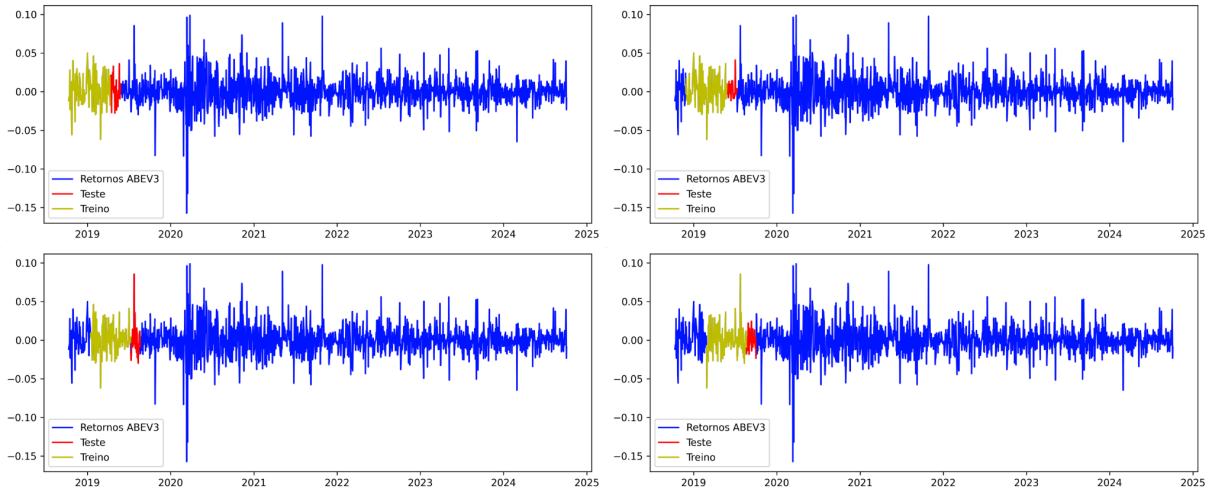
[2] Investopedia. Disponível em: https://www.investopedia.com/. Acesso em: 05 out. 2024.
[3] pandas.DataFrame. Disponível em: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html. Acesso em: 12 abr. 2025

argumento de Nielsen (2021, p. 399), em que ela demonstra que passeios aleatórios são ajustes naturais para movimentos de preços de ações, e portanto, o modelo mais simples para esse tipo de dado é simplesmente repetir o passo anterior, mas trata-se de um modelo ingênuo, sem poder preditivo real. Modelar retornos, por outro lado, não vem necessariamente atrelado a um poder preditivo maior, mas evita que o algoritmo aprenda um modelo ingênuo sem aplicabilidade. Após transformações, o conjunto de dados possui observações em 1492 dias.

$$R_t = \frac{P_t}{P_{t-1}} - 1 \qquad (6)$$

Os dados foram alimentados ao modelo na forma treino/teste em janela rolante, com treino de 120 exemplos e teste de 30, avançando a cada 30 pontos para evitar intersecção entre exemplos de teste, conforme a imagem 5. A escolha de 120 para treino e 30 para teste foi feita arbitrariamente como linha de base, porém outros trabalhos como Fjellstrom (2022), Nelson, Pereira e Oliveira (2017) e Krauss, Do e Huck (2017) costumam usar janelas maiores, como Fjellstrom, que usou 750 dias para treino. Experimentos com intervalos maiores ou menores para treino poderiam acarretar em benefícios para a performance dos modelos, porém limitações de tempo impediram a realização desses experimentos. As carteiras hipotéticas são geradas com base nas classificações nos períodos de teste. Ressalta-se que os primeiros 120 dias de cada série, que nunca são usados para teste, são perdidos e não compõem a carteira hipotética gerada pelos modelos.

Imagem 7 - Exemplificação da janela rolante 120/30



Fonte: elaborado pelo autor

Em cada instância foi atribuída uma classe 1 caso o retorno seja maior que 0, e 0 no caso contrário. Optar por uma tarefa de classificação em vez de uma de regressão segue Enke e Thawornwong (2005), em que os autores apresentam evidências empíricas de que simplificar o problema de prever o valor para prever a direção do retorno geralmente acarreta em retornos melhores ajustados ao risco, pois uma estratégia de compra e venda baseada no sinal do retorno é mais efetiva do que baseada no preço.

Para classificar cada instância correspondente ao dia t, foram utilizados 5 *lags*, ou valores passados da série de retornos, correspondentes aos dias $t-1$, $t-2$, $t-3$, $t-4$ e $t-5$. Dessa forma, cada exemplo na base de treino possui 5 características, cada uma contendo o retorno de um dos 5 dias anteriores. Em notação formal, em cada par (x_i, y_i), $i = 1, \ldots, 1492$, x_i é um vetor de 5 dimensões correspondente aos 5 lags do i-ésimo elemento e y_i é a classe que representa se o retorno do dia i é maior que zero ou se é igual ou menor que zero. A escolha desse *lag* foi feita de forma arbitrária para capturar uma semana útil de informação do mercado, e um valor maior ou menor poderia influenciar a performance dos modelos, mas restrições de tempo também restringiram a experimentação com esse parâmetro. Por fim, antes de serem alimentados aos modelos, os dados foram escalados para o intervalo [-1, 1].

3.4. Modelos e Treinamento

As 3 arquiteturas de RNR escolhidas, nomeadamente a LSTM, GRU e RNR simples, seguiram o mesmo padrão:

- 1 camada escondida.
- 8 unidades na camada escondida.
- Camada de saída com 1 unidade sigmóide, com limiar de decisão em 0,5: atribui-se 1 caso o valor seja maior que 0,5 e 0 caso seja igual ou menor.
- Função de perda Entropia Cruzada para classificação binária.
- Treinamento de 250 épocas com otimizador Adam.
- Parada antecipada com paciência de 25 épocas.

Os hiperparâmetros foram selecionados arbitrariamente com o objetivo de constituir uma linha de base comum para comparação. O algoritmo FA seguiu as

configurações padrão da biblioteca scikit-learn, através da classe RandomForestClassifier[4], também constituindo uma linha de base. Uma busca por hiperparâmetros poderia ser benéfica, mas, novamente, por restrições da pesquisa presente, os experimentos foram limitados. Por fim, o MA atribuiu uma das duas classes, com probabilidades iguais, utilizando a função np.random.choice[5].

Os modelos foram treinados e testados com toda a série de dados 30 vezes para contabilizar o papel da aleatoriedade na inicialização de pesos das redes neurais, na divisão de nós na FA e na atribuição de classes no MA. Para uma execução n, $n = 1, \ldots, 30$, a semente aleatória foi definida como n para assegurar a reprodutibilidade enquanto promove a variabilidade dos pontos iniciais dos modelos. Uma execução se refere a uma passagem completa da janela rolante pelo conjunto de dados de todas as ações, formando uma carteira baseada na decisão de investimentos diários no decorrer do período utilizado. Para cada execução n de cada técnica M, foi treinado um modelo por ação s, e as decisões individuais de cada modelo ${M^S}_n$, foram utilizadas para construir a carteira final da execução. Por exemplo: para a execução n da LSTM, foram treinadas 5 LSTMs: ${LSTM_n}^{ABEV3}$, ${LSTM_n}^{BBDC3}$, ${LSTM_n}^{ITSA3}$, ${LSTM_n}^{ITUB3}$, ${LSTM_n}^{WEGE3}$ e a decisão de compra ou venda de cada ação a cada dia com base na previsão do modelo formou a carteira hipotética da execução n. Para cada execução, foram coletadas as métricas de acurácia de cada modelo individual seguindo a definição de acurácia na fórmula 7, as acurácias médias entre os modelos de uma mesma técnica seguindo a fórmula 8, retornos acumulados e *Sharpe Ratio*, para posteriormente construir uma distribuição desses valores.

$$Acurácia({M^S}_n) = \frac{tp+tn}{tp+tn+fp+fn} \quad (7)$$

$$Acurácia_n = \frac{Acurácia({M^{ABEV3}}_n)+Acurácia({M^{BBDC3}}_n)+Acurácia({M^{ITSA3}}_n)+Acurácia({M^{ITUB3}}_n)+Acurácia({M^{WEGE3}}_n)}{5} \quad (8)$$

Os retornos acumulados são calculados como definido na fórmula 9, em que ${r_t}$ representa o retorno da carteira no dia t já considerando o efeito de cada ação na composição da carteira. Tira-se o produtório de $1 + {r_t}$ para todo o período T,

[4] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[5] https://numpy.org/doc/2.1/reference/random/generated/numpy.random.choice.html

efetivamente resultando em quanto o capital investido foi multiplicado na janela de tempo.

$$Retorno\ Acumulado = \Pi^{T}_{t=1}(1 + r_t) \quad (9)$$

Utilizando as classes preditas, foram construídos portfólios hipotéticos em que cada ação é mantida se a classe predita for 1, como se fosse comprada, e desconsiderada se a classe predita for 0, como se fosse vendida. O esquema anteriormente descrito faz com que o retorno diário em um dia t, que pode ser positivo ou negativo, seja realizado para a carteira caso o modelo decida que ação seja mantida, e desconsiderando a ação caso seja vendida. Ressalta-se que os retornos acumulados de cada carteira não consideram custos de transação, dividendos e impostos, para simplificar a interpretação e focar a análise na performance dos modelos. Cada uma das 5 ações têm peso igual de 20% na composição do portfólio. Através da carteira hipotética é extraído o *Sharpe Ratio*, considerando o retorno de baixo risco como a taxa SELIC arbitrariamente a 9,75% ao ano, que é uma média aproximada dos 6 anos do período, calculada a partir dos valores disponibilizados pelo Banco Central (Banco Central do Brasil, s.d.).

As três técnicas que atingirem os maiores valores médios de *Sharpe Ratio* na distribuição de experimentos irão constituir um comitê de previsores. O comitê será então avaliado em comparação com os modelos originais e será discutido se sua performance os supera. São selecionados 3 modelos para que seja possível obter um voto majoritário sem que haja a possibilidade de empate, como poderia acontecer se fossem utilizados os 4 modelos treinados: LSTM, GRU, RNR e FA.

4. Resultados

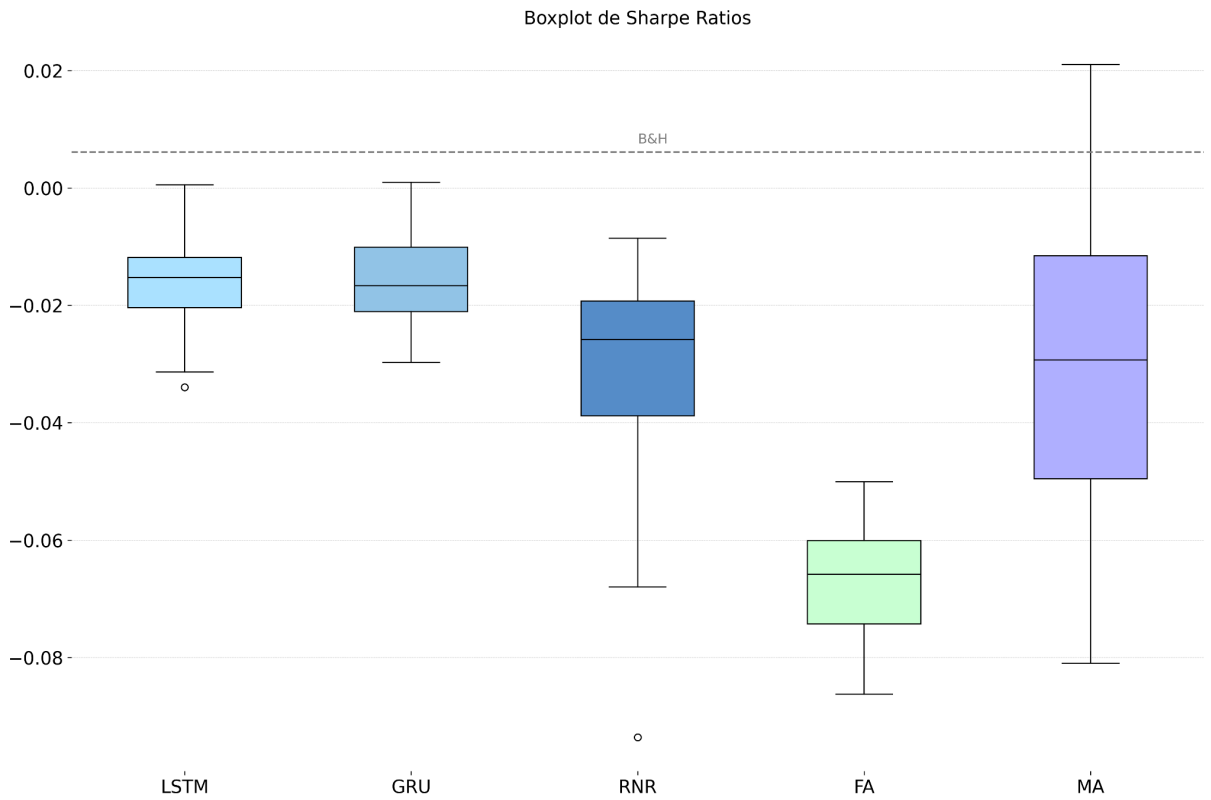
O presente tópico irá apresentar os resultados obtidos nos experimentos. As menções ao “melhor experimento” de cada modelo se referem à execução cuja carteira hipotética gerada rendeu o maior *Sharpe Ratio* entre as trinta execuções.

4.1. *Sharpe Ratio*

Os *Sharpe Ratios* dos portfólios criados por cada experimento distribuem-se majoritariamente abaixo de zero, como mostrado na imagem 6, sendo a FA o modelo com a menor média entre todos, com um valor de -0,0669. Um valor

negativo para o *Sharpe Ratio* indica que a carteira em questão não supera a taxa de retorno de baixo risco, pois para que essa métrica seja positiva, é necessário que os retornos da carteira sejam maiores que os retornos da taxa de baixo risco. Então pode-se afirmar que, em média, os modelos não foram capazes de obter retornos que superem a taxa de baixo risco. Contudo, o desvio padrão das distribuições referentes aos modelos treinados é menor do que o do MA, um fator que indica uma menor sensibilidade à aleatoriedade. É notável também o fato de o MA ter observações além do terceiro quartil distribuídas acima de 0 e acima das técnicas, atingindo uma máxima de 0,0210, mas esse fato se deve somente à aleatoriedade. Vale ressaltar que o *Sharpe Ratio* da carteira B&H é de apenas 0,0061, que é positivo mas muito próximo a zero, indicando um retorno baixo por unidade de risco. Os valores médios, máximos e os desvios padrão são demonstrados na tabela 1.

Imagem 8 - Distribuições de *Sharpe Ratios* por cada modelo



Fonte: elaborado pelo autor

Tabela 1 - Sumário de *Sharpe Ratios*

Modelo	Média	Máxima	Desvio Padrão
LSTM	-0,0162	0,0006	0,0080
GRU	-0,0163	0,010	0,0075
RNR	-0,0308	-0,0086	0,0181
FA	-0,0669	-0,0501	0,0102
MA	-0,0304	0,0210	0,0241
Benchmark (B&H)	**0,0061**	**-**	**-**

Fonte: elaborado pelo autor

4.2. Comitê

Para criar o comitê dos melhores três modelos treinados, foram utilizados os que atingiram maiores médias de *Sharpe Ratio*. É necessário verificar se há diferença significativa entre os modelos para validar que de fato se pode distinguir essas distribuições e afirmar que uma está localizada acima de outra. Para isso, foi conduzido um teste de Kruskal-Wallis para verificar diferenças entre os *Sharpes* da LSTM, GRU, RNR simples e FA. Foi verificado que há de fato diferença entre as distribuições para um nível de significância de 95%, e o teste pos-hoc de Dunn com correção de Bonferroni mostrou que a única exceção é entre a LSTM e a GRU, confirmando que a distribuição da FA está de fato localizada abaixo das outras três. Os valores-p do teste são demonstrados na tabela seguinte, em que a presença de "*" indica diferença estatisticamente significativa para um nível de significância de 95% e a ausência indica que não há diferença significativa.

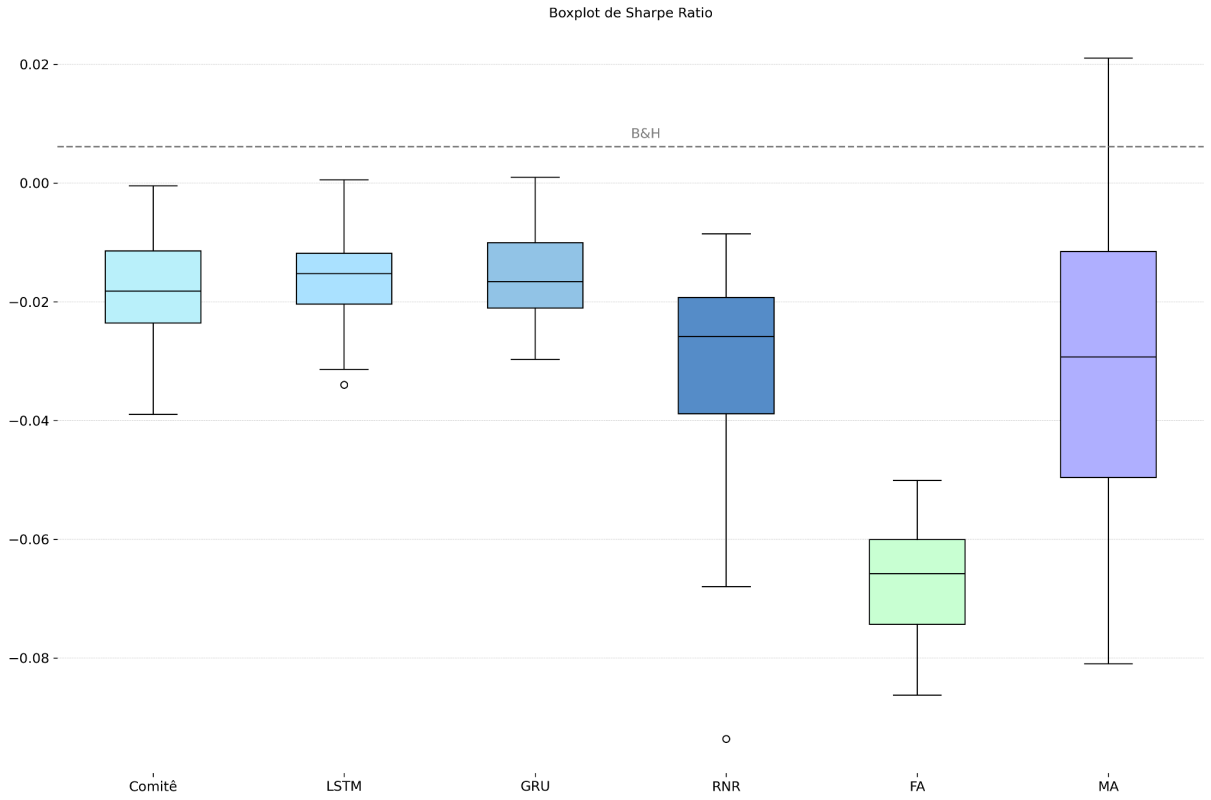
Tabela 2 - Valores-p do teste Kruskal-Wallis com correção de Bonferroni para diferença entre os Sharpe Ratios das técnicas de AM

	LSTM	GRU	RNR	FA
LSTM	-			
GRU	1,0	-		
RNR	0,008*	0,012*	-	
FA	0,0*	0,0*	0,0*	-

Fonte: elaborado pelo autor

Para cada um dos 30 experimentos executados dos três modelos, foi utilizado o voto majoritário da classificação de cada instância para formar a decisão do comitê. O modelo gerado resultou em um Sharpe médio acima da RNR simples, da FA, do MA, e muito próximo da LSTM e da GRU, com um valor de -0,0176 como mostrado na imagem 7 e tabela 3.

Imagem 9 - Distribuições de *Sharpe Ratios* incluindo o comitê



Fonte: elaborado pelo autor

Tabela 3 - Sumário de *Sharpe Ratios* entre o Comitê e seus modelos base

Modelo	Média	Máxima	Desvio Padrão
Comitê	-0,0176	-0,0004	0,0088
LSTM	-0,0162	0,0006	0,0080
GRU	-0,0163	0,010	0,0075
RNR	-0,0308	-0,0086	0,0181
FA	-0,0669	-0,0501	0,0102
MA	-0,0304	0,0210	0,0241
Benchmark (B&H)	**0,0061**	**-**	**-**

Fonte: elaborado pelo autor

Pela proximidade dessa nova distribuição às existentes, foi novamente conduzido um teste de Kruskal-Wallis com nível de significância de 95%, em que foi verificada uma diferença significativa do comitê apenas contra a FA, como demonstrado pelos valores-p na tabela 4. O resultado do comitê em questão de *Sharpe Ratio*, com distribuição indistinguível dos seus constituintes individuais, indica que para esse problema, uma união entre os melhores 3 modelos não consegue formar uma decisão que gere melhores retornos ajustados ao risco do que os seus modelos base, nem contra a estratégia aleatória.

Tabela 4 - Valores-p do teste Kruskal-Wallis com correção de Bonferroni para diferença entre os Sharpe Ratios das técnicas de AM, Comitê e MA

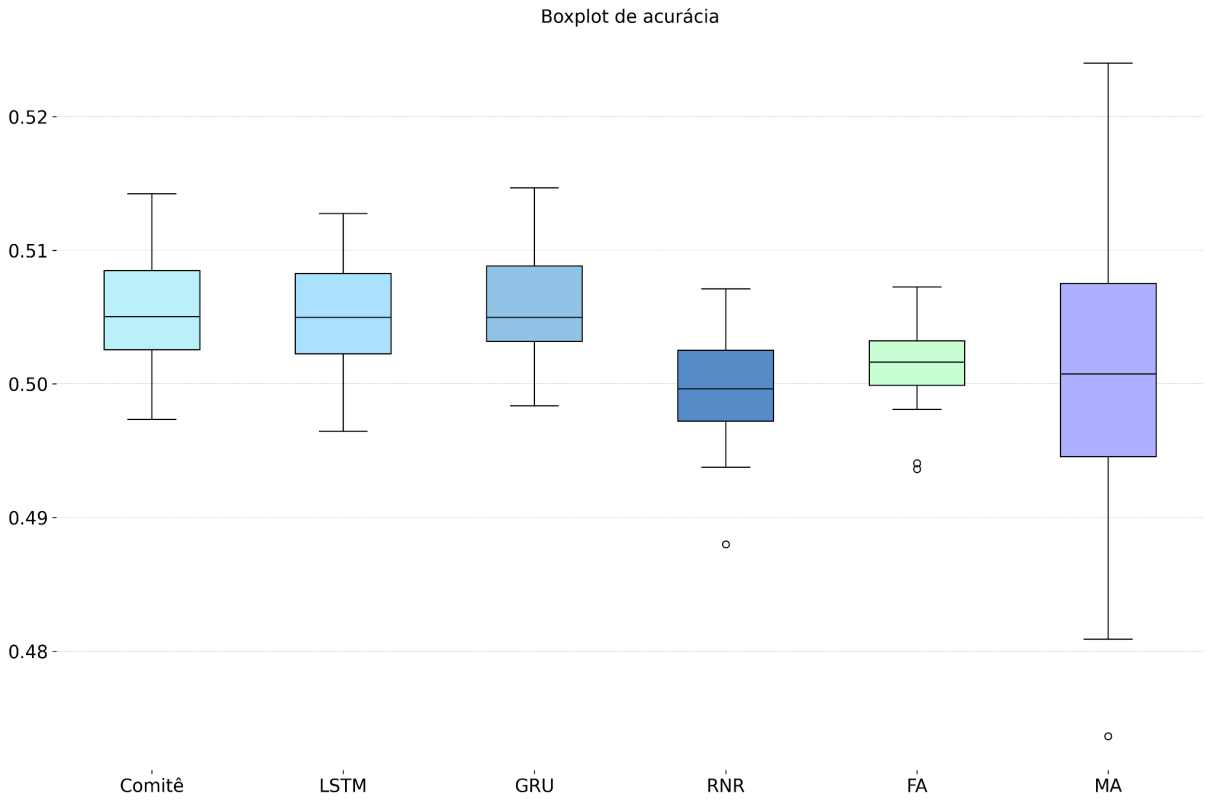
	Comitê	LSTM	GRU	RNR	FA	MA
Comitê	-					
LSTM	**1,000**	-				
GRU	**1,000**	1,000	-			
RNR	**0,103**	0,019*	0,025*	-		
FA	**0,000***	0,000*	0,000*	0,000*	-	
MA	**0,476**	0,114	0,145	1,000	0,000*	-

Fonte: elaborado pelo autor

4.3. Acurácias

Inicialmente, são analisadas as acurácias médias de cada carteira formada utilizando todas as ações por cada modelo, seguindo a fórmula 8. As acurácias encontram-se distribuídas perto de 50% para todos os modelos, inclusive para o MA, um fator que aponta que as técnicas possivelmente não conseguiram aprender padrões nos dados para superar um modelo de adivinhação. A imagem 8 mostra a distribuição das acurácias médias nos quatro modelos treinados, no comitê, e também na abordagem aleatória por meio de um boxplot, e a tabela 5 descreve as distribuições em média, máxima e desvio padrão.

Imagem 10 - Distribuições das acurácias dos modelos



Fonte: elaborado pelo autor

Tabela 5 - Sumário de Acurácias

Modelo	Média	Máxima	Desvio Padrão
Comitê	0,5053	0,5142	0,0041
LSTM	0,5053	0,5127	0,0041
GRU	0,5056	0,5147	0,0040
RNR	0,4998	0,5071	0,0042
FA	0,5016	0,5073	0,0031
MA	**0,5006**	**0,5240**	**0,0117**

Fonte: elaborado pelo autor

Através do boxplot, supõe-se que as acurácias das técnicas, em média, não superam as acurácias do benchmark aleatório. Mais um teste de Kruskal-Wallis com significância de 95% aponta que não existe diferença significativa entre o MA e os outros modelos, e também entre o comitê, a LSTM e a GRU, demonstrado na tabela 6. A proximidade do comitê aos seus constituintes está de acordo com Dietterich

(2000), em que o autor afirma que uma das condições para o sucesso de comitês é que os seus constituintes individuais sejam acurados, o que não é o caso dos modelos em questão.

Tabela 6 - Valores-p do teste Kruskal-Wallis com correção de Bonferroni para diferença entre as acurácias das técnicas de AM, Comitê e MA

	Comitê	LSTM	GRU	RNR	FA	MA
Comitê	-					
LSTM	**1,000**	-				
GRU	**1,000**	1,000	-			
RNR	0,000*	0,000*	0,000*	-		
FA	0,028*	0,033*	0,016*	1,000	-	
MA	**0,083**	**0,094**	**0,050**	**1,000**	**1,000**	-

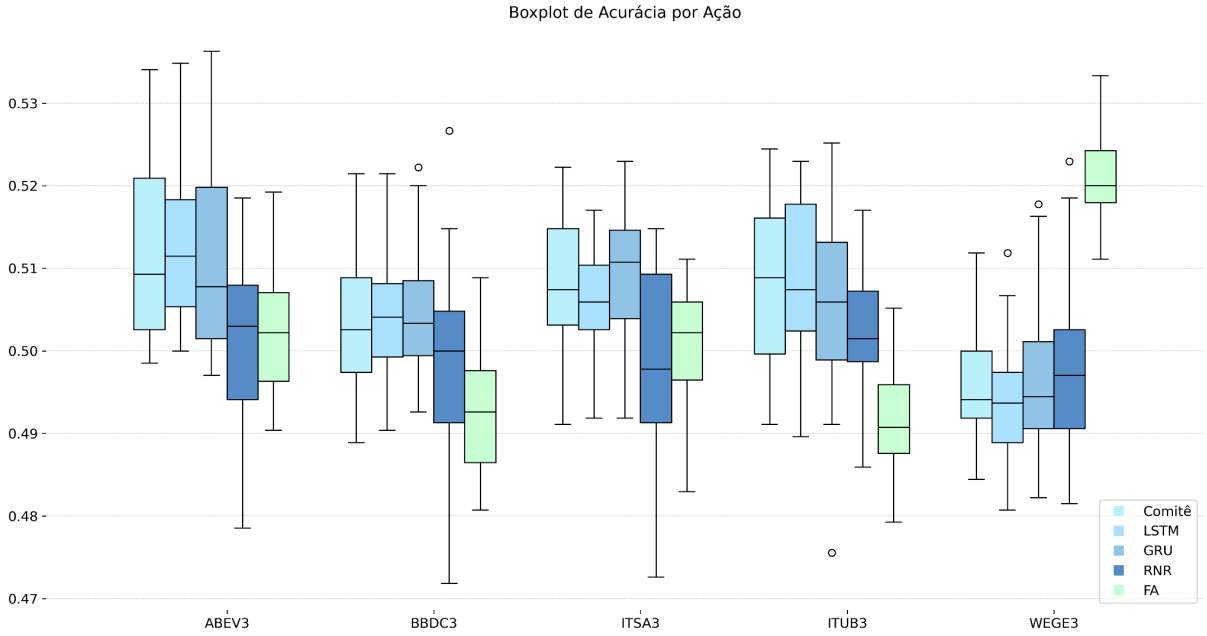
Fonte: elaborado pelo autor

As distribuições de acurácias observadas estão em concordância com o que é visto na literatura similar como em Krauss, Do e Huck (2017), Fjellstrom (2022), Caride, Bariviera e Lanzarini (2018), Nelson, Pereira e Oliveira (2017), Fischer e Krauss (2018). A proximidade das acurácias a 50% e a ausência de diferença significativa entre o MA e as técnicas também corrobora o argumento de Malkiel (2003), de que não há padrões que possam ser explorados no mercado, portanto, não é possível ajustar um modelo para prever os próximos passos com acurácia satisfatória.

4.3.1 Acurácias Por Ação

Nesta seção, são avaliadas as acurácias de cada modelo treinado para cada ação como descrito na fórmula 7, para observar se algum modelo se destaca na série de alguma ação específica. Na imagem 9 observa-se as distribuições das acurácias de cada técnica por ação. Nenhuma distribuição se destaca da casa dos 50%, mas há alguns casos interessantes de se observar, como as acurácias do comitê, LSTM e GRU para ABEV3, que tem pontos distribuídos acima dos 53%, como também da FA para WEGE3, a única instância em que a máxima da FA está localizada acima das máximas dos outros modelos.

Imagem 11 - Acurácias de cada modelo por ação

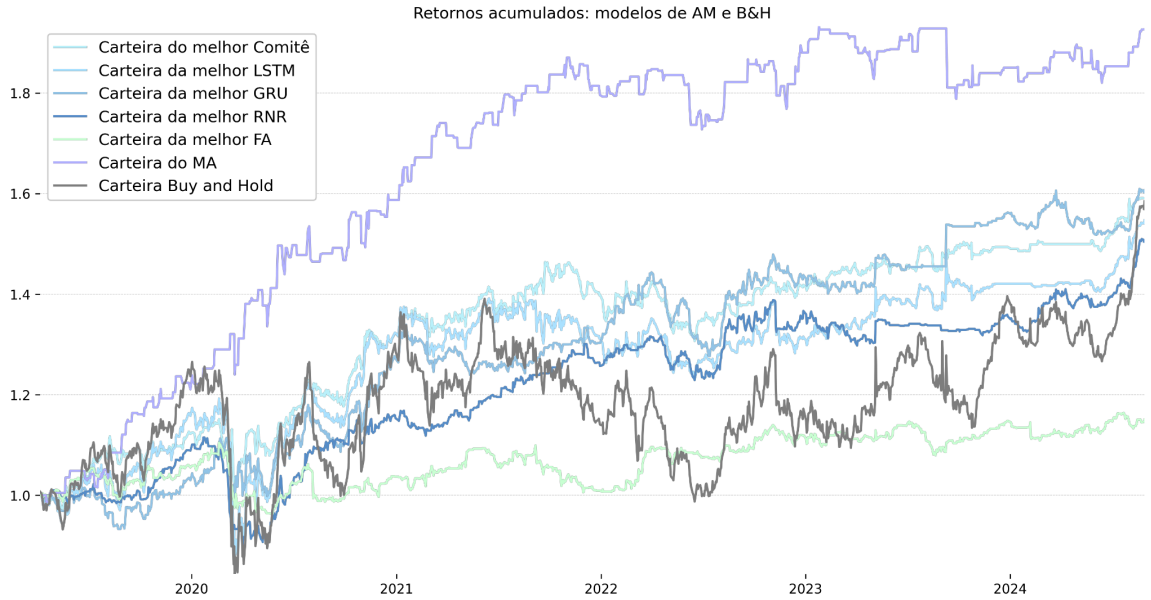


Fonte: elaborado pelo autor

4.4. Retornos

Por último, são avaliados os retornos acumulados dos portfólios gerados pelos experimentos com cada modelo. A figura 10 demonstra o melhor experimento de cada um dos modelos de AM, o MA e a estratégia B&H. É interessante observar que no melhor experimento de cada modelo, na maior parte do tempo, as redes neurais estão com retornos superiores à estratégia B&H, sendo a LSTM e a RNR superadas apenas no final da série, indicando certo potencial para esses modelos superarem o B&H. O melhor experimento da FA fica consistentemente abaixo de todos os modelos e da estratégia B&H, e o melhor desempenho do MA supera todos os melhores desempenhos das demais abordagens. A análise deve se estender para avaliar as distribuições dos retornos acumulados através de todos os experimentos para verificar a performance média, e não apenas os melhores resultados de cada.

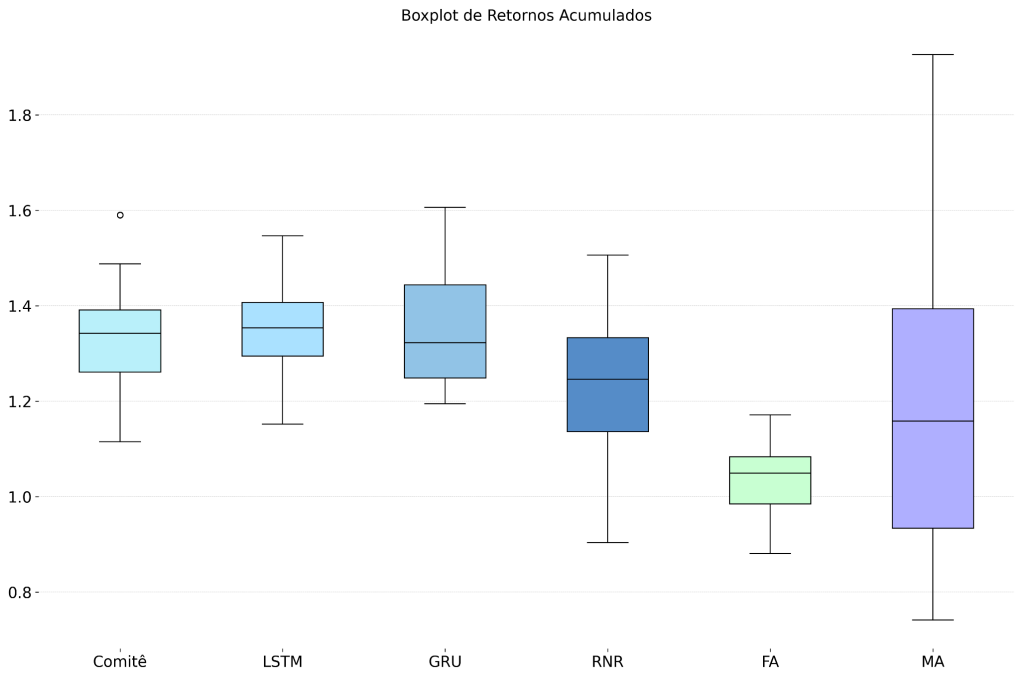
Imagem 12 - Retornos acumulados dos melhores experimentos de cada modelo



Fonte: elaborado pelo autor

A seguir são avaliadas as distribuições de retornos acumulados das carteiras geradas por cada um dos 30 experimentos de cada modelo. Através da tabela 7 e do gráfico na imagem 11, que mostra as distribuições dos retornos acumulados nos experimentos, observa-se que pode não haver diferença significativa entre a distribuição do MA e das técnicas de AM.

Imagem 13 - Distribuição dos retornos acumulados das carteiras hipotéticas geradas



Fonte: elaborado pelo autor

Tabela 7 - Sumário dos retornos acumulados

Modelo	Média	Máxima	Desvio Padrão
Comitê	1,3380	1,5905	0,1081
LSTM	1,3471	1,5467	0,0910
GRU	1,3543	1,6064	0,1126
RNR	1,2326	1,5061	0,1418
FA	1,0352	1,1713	0,0746
MA	1,1825	1,9266	0,2961
Benchmark (B&H)	**1,5846**	**-**	**-**

Fonte: elaborado pelo autor

Se a média do MA não for significativamente diferente das médias das técnicas, é um indício de que os modelos treinados não conseguem superar uma estratégia aleatória. Para verificar essa observação, foram conduzidos testes de Wilcoxon par a par entre a as amostras referentes ao MA e cada um dos modelos de AM, para verificar a hipótese nula (H0) de que não existe diferença entre as distribuições subjacentes, contra a hipótese alternativa (HA) de que a distribuição de um modelo m está localizada acima da distribuição do MA.

Tabela 8 - Resultados dos testes de Wilcoxon entre modelos e MA

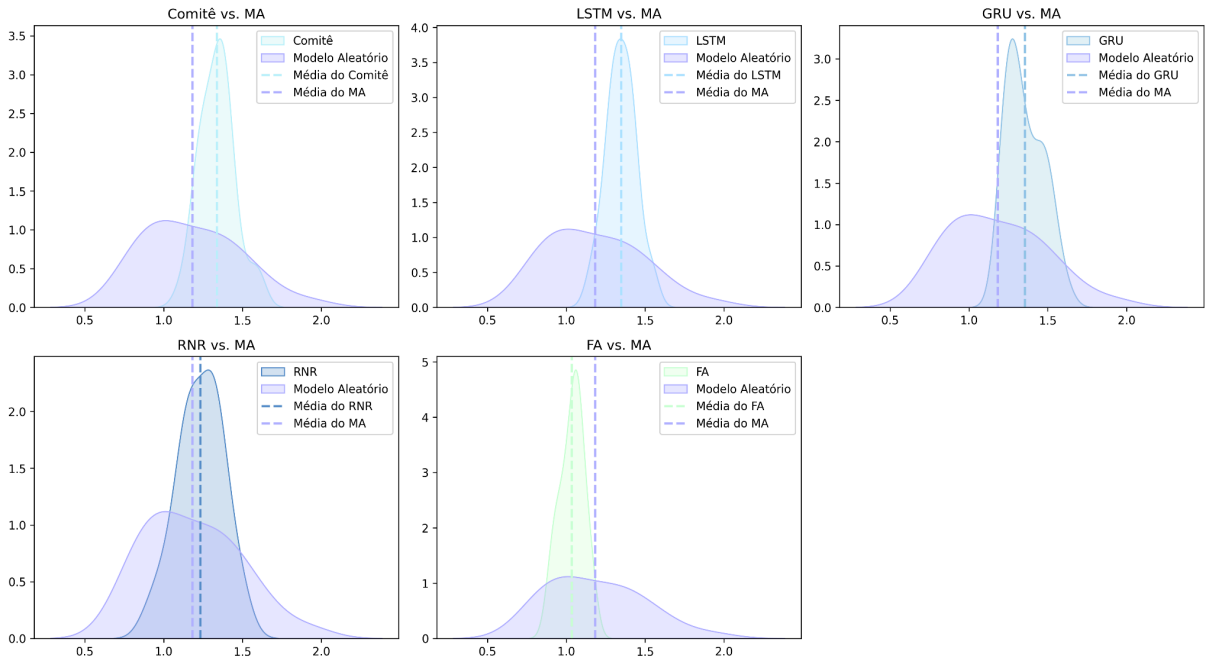
MA vs.	Estatística	p-valor	Decisão
Comitê	355,0	0,0053	Rejeita H0
LSTM	351,0	0,0068	Rejeita H0
GRU	356,0	0,0050	Rejeita H0
RNR	281,0	0,1642	Não Rejeita H0
FA	139,0	0,9738	Não Rejeita H0

Fonte: elaborado pelo autor

Foi observado que o teste de Wilcoxon rejeita $H0$ contra a LSTM, GRU e Comitê, mas não contra a RNR simples e a FA. Os resultados do teste indicam que os 3 modelos que rejeitam a nula acarretam em uma distribuição de retornos superiores ao MA, mesmo que as acurácias médias entre as técnicas e o MA não tenham diferença significativa. A imagem 12 demonstra a função de densidade das

distribuições, em que fica mais claro a localização das distribuições das amostras dos modelos referente à amostra do MA.

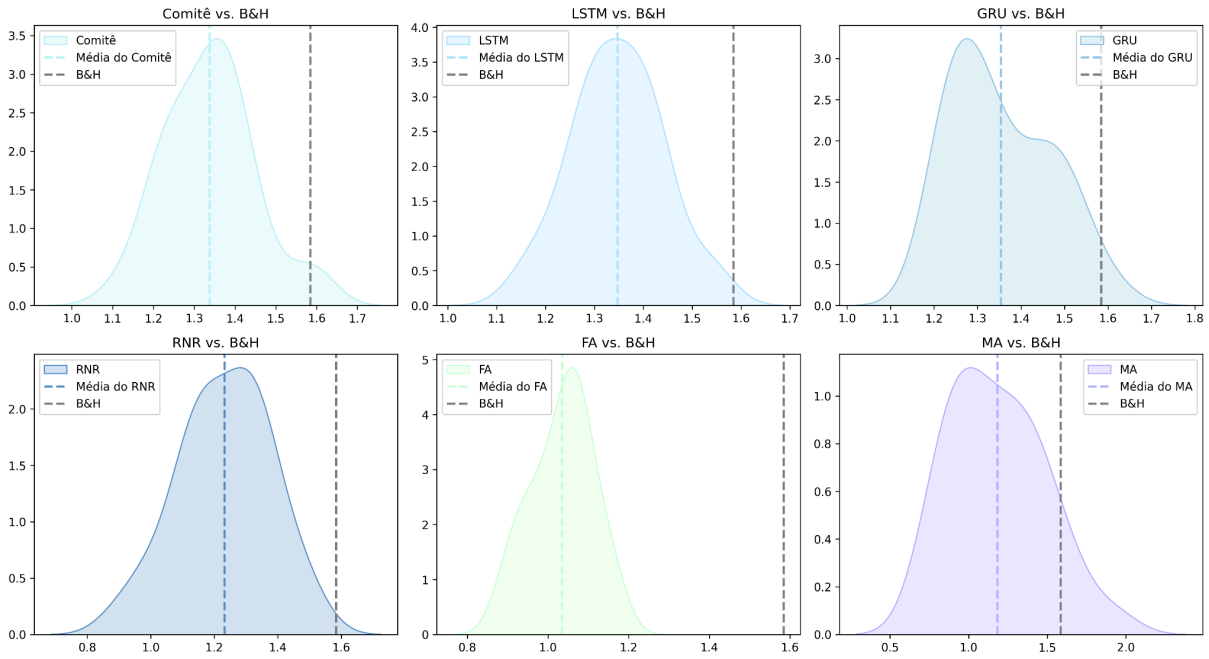
Imagem 14 - Funções Densidade dos pares de amostras



Fonte: elaborado pelo autor

Além de comparar os retornos de cada modelo com a estratégia aleatória, também é interessante comparar cada modelo com a estratégia B&H. Sabendo que, através da tabela 7, o retorno da estratégia B&H está acima da média dos modelos, procura-se verificar que esse é de fato o caso com significância estatística. Para verificar essa suposição, foi conduzido novamente o teste de Wilcoxon par a par entre o valor do retorno acumulado da estratégia B&H e a distribuição de retornos referentes a cada modelo. A imagem 13 mostra a localização do valor do B&H referente à média de cada distribuição de retornos de cada modelo, incluindo o aleatório.

Imagem 15 - Funções Densidade das amostras dos modelos contra o valor do B&H



Fonte: elaborado pelo autor

Os testes apontam com significância de 95%, para todas as técnicas e para o MA, a observação de que o retorno da estratégia B&H está de fato localizado acima do retorno de qualquer modelo empregado neste trabalho.

Tabela 9 - Resultados dos testes de hipótese para diferença entre modelos e B&H

B&H vs.	Estatística	p-valor	Decisão
Comitê	3,0	0,0	Rejeita H0
LSTM	0,0	0,0	Rejeita H0
GRU	1,0	0,0	Rejeita H0
RNR	0,0	0,0	Rejeita H0
FA	0,0	0,0	Rejeita H0
MA	17,0	0,0	Rejeita H0

Fonte: elaborado pelo autor

5. Conclusão

A partir dos experimentos conduzidos, foi observado que nenhum modelo, em média, supera uma estratégia B&H de investimento a longo prazo. Além disso, as acurácias localizadas ao redor de 50% para todos os modelos em todas as ações apontam que os modelos não conseguiram aprender padrões com os dados, e suas

acurácias não são muito melhores do que o lançar de uma moeda. O fato de as acurácias dos modelos base serem insatisfatórias contribuíram para que um comitê de alguns dos modelos treinados siga rendendo resultados insatisfatórios. A observação dessas métricas leva à suposição de que os experimentos que conseguiram retornos acima de 1, ou seja, multiplicaram pelo menos uma pequena fração do dinheiro investido, aconteceram puramente pelo acaso. Contudo, no melhor experimento foi observado que alguns modelos conseguem superar a estratégia B&H. Se aplicada uma análise de IA Explicável, pode-se entender melhor como os modelos tomaram suas decisões nos melhores experimentos, porém restrições de tempo impediram a realização desse tipo de análise.

Ressalta-se o fato de duas das quatro técnicas originais superarem o MA em retornos acumulados, nomeadamente a LSTM e a GRU, de modo que o Comitê também supera o MA, mesmo que a RNR não o faça. Apesar de algumas técnicas serem melhores que uma abordagem aleatória, o que é um bom sinal para as abordagens baseadas em AM, não se pode afirmar que os modelos selecionados, com os hiperparâmetros utilizados, os dados coletados e o intervalo temporal escolhido são viáveis para potencializar decisões de investimento de longo prazo que superem uma estratégia de investimento passiva de comprar as ações e mantê-las na carteira. Essa observação corrobora com o argumento de Malkiel (2003) de que não existem padrões que possam ser explorados no histórico de movimentos de ações, e também vai de acordo com a constatação de Nelson, Pereira e Oliveira (2017) sobre os movimentos no mercado financeiro terem uma proporção de sinal para ruído insignificante.

5.1 Ameaças à Validade

Apesar do rigor na condução dos experimentos e da busca por reprodutibilidade, a presente pesquisa possui limitações importantes que devem ser consideradas na interpretação dos resultados.

Uma das principais limitações está relacionada ao volume e diversidade dos dados utilizados. A seleção de apenas cinco ações do índice Bovespa pode não representar a complexidade e a heterogeneidade do mercado brasileiro e omitir possíveis padrões dos algoritmos de aprendizado, tal como a exclusão de ações de empresas estatais. Além disso, o uso exclusivo do retorno diário como variável

preditora desconsidera outras fontes relevantes de informação, como indicadores técnicos, fatores macroeconômicos ou sentimento de mercado, que poderiam enriquecer o poder preditivo dos modelos.

Outro ponto diz respeito à configuração dos modelos e hiperparâmetros adotados. Todos os modelos foram treinados com configurações básicas, arbitrárias e homogêneas, sem passar por processos de otimização de hiperparâmetros. Isso implica que os resultados obtidos não necessariamente refletem o potencial máximo das técnicas analisadas. Ademais, foram utilizadas apenas arquiteturas básicas de redes recorrentes, com uma única camada oculta e número reduzido de unidades, o que pode limitar a capacidade de modelagem de padrões complexos.

O tamanho das janelas de treino e teste e o número de lags utilizados para classificar cada ponto também foram definidos de forma arbitrária e mantidos fixos ao longo dos experimentos. Janelas maiores ou com configurações diferentes poderiam influenciar a performance dos modelos, especialmente em um domínio tão sensível a sazonalidades e regimes de mercado altamente voláteis como o financeiro. A escolha de 5 lags para cada classificação também poderia ser alterada para que cada decisão fosse tomada com base em mais ou menos dias prévios. Além disso, a resolução temporal diária adotada pode ocultar padrões intradiários potencialmente exploráveis por modelos de AM, possivelmente observável em dados de alta frequência em intervalos de horas ou minutos.

Adicionalmente, os resultados de performance das carteiras simuladas não consideram custos de transação, impostos ou dividendos, para evidenciar a performance dos modelos comparados. Essa escolha invariavelmente afasta a análise de um cenário realista de aplicação. Como o modelo simula operações diárias, esses custos poderiam impactar significativamente o retorno líquido das estratégias.

5.2 Trabalhos Futuros

Pesquisas futuras podem incluir mas não estão limitadas a:

- Estender os experimentos com um volume maior de ações, janelas temporais, épocas de treinamento e profundidade de RNRs.
- Explorar metodologias de seleção e geração de características para previsão específicas do domínio como indicadores micro e macro-econômicos.

- Explorar metodologias de seleção e geração de características baseadas em dados não estruturados como linguagem natural, utilizando *Large Language Models* (LLMs) para incorporar sentimento, opiniões e notícias na previsão.
- Utilização de análises de IA Explicável para compreender como modelos caixa-preta, como as RNA, definem a contribuição de cada característica dos dados.
- Explorar como diferentes funções objetivo e algoritmos de otimização afetam a convergência de algoritmos de aprendizado enfatizando métricas específicas de avaliação no contexto da modelagem de movimentos de mercado ao invés de métricas clássicas de avaliação de classificadores.

REFERÊNCIAS

BANCO CENTRAL DO BRASIL. Histórico das Taxas de Juros. Brasília: Banco Central do Brasil, [s.d.]. Disponível em: https://www.bcb.gov.br/controleinflacao/historicotaxasjuros. Acesso em: 03 mar. 2025.

BASHIR, Daniel et al. An information-theoretic perspective on overfitting and underfitting. [S.l.], 2020. Disponível em: https://doi.org/10.48550/arXiv.2010.06076. Acesso em: 31 mar. 2025.

BREIMAN, Leo. Random Forests. Berkeley: University of California, 2001. Disponível em: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf. Acesso em: 03 nov. 2024.

CARIDE, Martin Iglesias; BARIVIERA, Aurelio F.; LANZARINI, Laura. Stock returns forecast: an examination by means of Artificial Neural Networks. [S.l.], 2018. Disponível em: http://arxiv.org/abs/1801.07960. Acesso em: 03 set. 2024.

CHAVHAN, Suresh et al. Deep Learning Approaches for Stock Price Prediction: A Comparative Study of LSTM, RNN, and GRU Models. In: *INTERNATIONAL CONFERENCE ON SMART AND SUSTAINABLE TECHNOLOGIES (SPLITECH)*, 9., 2024, Bol and Split, Croatia. Anais [...]. IEEE, 2024. p. 01-06. DOI: 10.23919/SpliTech61897.2024.10612666.

CHEN, Weisi et al. Deep learning for financial time series prediction: a state-of-the-art review of standalone and hybrid models. *Computer Modeling in Engineering & Sciences*, v. 139, n. 1, p. 187–224, 2024. Disponível em: https://www.techscience.com/CMES/v139n1/55114. DOI: https://doi.org/10.32604/cmes.2023.031388. Acesso em: 02 abr. 2025.

CHO, Kyunghyun et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:1406.1078 [cs.CL], 2014. Disponível em: https://arxiv.org/abs/1406.1078. DOI: 10.48550/arXiv.1406.1078. Acesso em: 31 mar. 2025.

DIETTERICH, Thomas G. Ensemble methods in machine learning. In: GOOS, Gerhard; HARTMANAIS, Juris; VAN LEEUWEN, Jan (Ed.). *Multiple classifier systems*. Berlin: Springer, 2000. v. 1857, p. 1–15. DOI: 10.1007/3-540-45014-9_1.

ENKE, David; THAWORNWONG, Suraphan. The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, v. 29, n. 4, p. 927-940, nov. 2005. DOI: 10.1016/j.eswa.2005.06.024. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S0957417405001156. Acesso em: 08 nov. 2024.

FAMA, Eugene F. Efficient capital markets: a review of theory and empirical work. *The Journal of Finance*, v. 25, n. 2, p. 383-417, maio 1970. Disponível em: https://www.jstor.org/stable/2325486. Acesso em: 21 set. 2024.

FISCHER, Thomas; KRAUSS, Christopher. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, v. 270, n. 2, p. 654-669, out. 2018. DOI: 10.1016/j.ejor.2017.11.054.

FJELLSTROM, Carmina. Long short-term memory neural network for financial time series. [S.l.], 2022. Disponível em: http://arxiv.org/abs/2201.08218. Acesso em: 20 nov. 2024.

GANAÏE, M. A. et al. Ensemble deep learning: A review. arXiv preprint arXiv:2104.02395v3, 2022. Disponível em: https://arxiv.org/abs/2104.02395. Acesso em: 12 abr. 2025.

GAO, Penglei; ZHANG, Rui; YANG, Xi. The application of stock index price prediction with neural network. *Mathematical and Computational Applications*, v. 25, n. 3, p. 53, 18 ago. 2020. DOI: 10.3390/mca25030053. Disponível em: https://www.mdpi.com/2297-8747/25/3/53. Acesso em: 01 mar. 2025.

GÉRON, Aurélien. *Mãos à obra: aprendizado de máquina com Scikit-Learn, Keras e TensorFlow*. 3. ed. Rio de Janeiro: Alta Books, 2021.

GITHUB. lhckb/tcc. GitHub, 2025. Disponível em: https://github.com/lhckb/tcc. Acesso em: 24 jun. 2025.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. Cambridge: MIT Press, 2016. Disponível em: http://www.deeplearningbook.org. Acesso em: 01 nov. 2024.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural Computation*, v. 9, n. 8, p. 1735-1780, 1997.

HYNDMAN, Rob J.; ATHANASOPOULOS, George. *Forecasting: Principles and Practice*. 2. ed. Melbourne: OTexts, 2018. Disponível em: https://OTexts.com/fpp2. Acesso em: 03 nov. 2024.

IBM. What is Random Forest? *IBM — Think*. [s.d.]. Disponível em: https://www.ibm.com/think/topics/random-forest. Acesso em: 24 jun. 2025.

JULIÃO, Fabricio. Ações da Petrobras caem mais de 6% após falas de Lula sobre corte em privatizações. *CNN Brasil*, São Paulo, 02 jan. 2023. Disponível em: https://www.cnnbrasil.com.br/economia/investimentos/acoes-da-petrobras-caem-apos-falas-de-lula-sobre-corte-em-privatizacoes/. Acesso em: 10 out. 2024.

KRAUSS, Christopher; DO, Xuan Anh; HUCK, Nicolas. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, v. 259, n. 2, p. 689-702, jun. 2017. DOI: 10.1016/j.ejor.2016.10.031. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S0377221716310867. Acesso em: 04 set. 2024.

PRECHELT, Lutz. Early stopping—But when? In: ORR, Genevieve B.; MÜLLER, Klaus-Robert (Eds.). *Neural Networks: Tricks of the Trade*. 2. ed. Berlin: Springer, 2012. p. 53–67. (Lecture Notes in Computer Science, v. 7700). Disponível em: https://doi.org/10.1007/978-3-642-35289-8_5. Acesso em: 2 abr. 2025.

MACHADO, Juliana; GIL, Pedro. Os impactos desastrosos da interferência do governo na Petrobras e na Vale. *Veja*, 15 mar. 2024. Disponível em: https://veja.abril.com.br/economia/os-impactos-desastrosos-da-interferencia-do-governo-na-petrobras-e-na-vale. Acesso em: 10 out. 2024.

MALKIEL, Burton G. The Efficient Market Hypothesis and Its Critics. *Journal of Economic Perspectives*, v. 17, n. 1, p. 59–82, Winter 2003. Disponível em: https://www.aeaweb.org/articles?id=10.1257/089533003321164958. Acesso em: 01 mar. 2025.

MITCHELL, Tom M. *Machine learning*. 1. ed. New York: McGraw-Hill, 1997. 414 p. ISBN 0-07-0428077.

MORETTIN, Pedro A.; BUSSAB, Wilton O. *Estatística básica*. 10. ed. São Paulo: Saraiva Uni, 2023. 624 p.

MORETTIN, Pedro A.; SINGER, Julio da Motta. *Estatística e ciência de dados*. Rio de Janeiro: LTC, 2022. 464 p.

NELSON, David M. Q.; PEREIRA, Adriano C. M.; OLIVEIRA, Renato A. Stock market's price movement prediction with LSTM neural networks. In: *INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN)*, 2017, Anchorage, AK. Anais [...]. IEEE, 2017. p. 1419-1426. DOI: 10.1109/IJCNN.2017.7966019. Disponível em: http://ieeexplore.ieee.org/document/7966019. Acesso em: 04 set. 2024.

NIELSEN, Aileen. *Análise prática de séries temporais: predição com estatística e aprendizado de máquina*. 1. ed. São Paulo: Alta Books, 2021. 480 p. ISBN 978-8550815626.

NIELSEN, Michael. *Neural Networks and Deep Learning*. [S.l.]: Determination Press, 2015. Disponível em: http://neuralnetworksanddeeplearning.com/index.html. Acesso em: 15 out. 2024.

NUMPY DEVELOPERS. NumPy Release 2.1.2 Notes. [S.l.]: [s.n.], [s.d.]. Disponível em: https://numpy.org/devdocs/release/2.1.2-notes.html. Acesso em: 12 abr. 2025.

PANDAS DEVELOPMENT TEAM. What's new in 2.2.3. [S.l.], 20 set. 2024. Disponível em: https://pandas.pydata.org/docs/whatsnew/v2.2.3.html. Acesso em: 12 abr. 2025.

PYTHON SOFTWARE FOUNDATION. Python 3.12. 2023. Disponível em: https://www.python.org/downloads/release/python-3120/. Acesso em: 12 abr. 2025.

RAVANELLI, Mirco et al. Light Gated Recurrent Units for Speech Recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence*, v. 2, n. 2, p. 92-102, abr. 2018. DOI: 10.1109/TETCI.2017.2762739. Disponível em: http://arxiv.org/abs/1803.10225. Acesso em: 30 mar. 2025.

SCIKIT-LEARN DEVELOPERS. scikit-learn: Machine Learning in Python – Release 1.5.2. [S.l.], 2024. Disponível em: https://scikit-learn.org/stable/whats_new/v1.5.html#version-1-5-2. Acesso em: 12 abr. 2025.

SHARPE, William F. The Sharpe Ratio. *Journal of Portfolio Management*, v. 21, n. 1, p. 49-58, 1994.

TANG, Yajiao et al. A survey on machine learning models for financial time series forecasting. *Neurocomputing*, v. 512, p. 363-380, nov. 2022. DOI: 10.1016/j.neucom.2022.09.003. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S0925231222009905. Acesso em: 02 set. 2024.

TENSORFLOW TEAM. What's New in TensorFlow 2.16. [S.l.], 2024. Disponível em: https://blog.tensorflow.org/2024/03/whats-new-in-tensorflow-216.html. Acesso em: 12 abr. 2025.

VILLANUEVA, Wilfredo Jaime P. Comitê de máquinas em predição de séries temporais. 2006. 133 f. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2006.

YAZBEK, Priscila. Petrobras dá adeus aos R$ 30? Falas de Bolsonaro sobre política de preços reforçam cautela com as ações. *Infomoney*, 09 abr. 2021. Disponível em: https://www.infomoney.com.br/mercados/petrobras-da-adeus-aos-r-30-falas-de-bolsonaro-sobre-politica-de-precos-reforcam-cautela-com-as-acoes/. Acesso em: 10 out. 2024.