Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 31, Num. 2 (2024) 147-160

RESEARCH ARTICLE

Prediction of Stock Prices Using Ensemble Models

Predição do Preço de Ações Utilizando Combinação de Modelos

Cláudio Estevam Leite da Silva[1]*, Ricardo Menezes Salgado[1]

Abstract: The financial market encompasses a set of institutions, products, and services aimed at meeting the financial needs of individuals, companies, and governments. Its primary objective is to direct financial resources from investors to projects requiring funding. This is achieved through the issuance and trading of securities such as stocks, debt securities, among others. In this paper, the goal was to develop a machine learning application specifically for the Brazilian financial market, focusing on predicting the market value of eight companies that are representative of the financial sector on the stock exchange. The prediction is based on the closing price history and uses data from the last three years, with the inputs corresponding to the last 60 days immediately preceding the forecast date. For this task, three machine learning models were selected: Long Short-Term Memory (LSTM), Multilayer Perceptron (MLP), and Convolutional Neural Network (CNN). Each of these was fine-tuned using five different optimizers, resulting in a total of 15 models. Subsequently, all 15 models were combined into an Ensemble. After applying data transformations, the models achieved a satisfactory level of error for the analysis. Among the transformations used, the logarithmic transformation stood out as the one that resulted in the most well-adjusted models compared to the others. In second place, the Yeo-Johnson transformation showed slightly higher error but performed better on series with high variation. Additionally, the convolutional models and Ensemble were the most effective.
Keywords: Machine Learning — Financial Market — Price Prediction — Ensemble Modeling

Resumo: O mercado financeiro engloba um conjunto de instituições, produtos e serviços com o propósito de atender as necessidades financeiras das pessoas, empresas e governos. Seu objetivo principal é direcionar recursos financeiros de investidores para projetos que demandam financiamento. Isso é feito através da emissão e negociação de títulos, como ações, títulos de dívida, entre outros. Neste artigo, o objetivo foi desenvolver uma aplicação de aprendizado de máquina específica para o mercado financeiro brasileiro, com foco em prever o valor de mercado de oito empresas que são representativas do setor financeiro na bolsa de valores. A previsão é baseada no histórico de fechamento dos preços e utiliza dados dos últimos três anos, com as entradas correspondendo aos últimos 60 dias imediatamente anteriores à data prevista. Para esta tarefa, foram selecionados três modelos de aprendizado de máquina: *Long Short-Term Memory* (LSTM), *Multilayer Perceptron* (MLP) e Rede Neural Convolucional (CNN). Cada um deles foi ajustado por cinco otimizadores distintos, totalizando 15 modelos. Em seguida, todos os 15 modelos foram combinados em um *Ensemble*. Após a aplicação de transformações nos dados, os modelos conseguiram atingir um nível de erro satisfatório para a análise. Dentre as transformações utilizadas, a transformação logarítmica se destacou como a que resultou em modelos mais ajustados em comparação às outras. Em segundo lugar, a transformação Yeo-Johnson apresentou um erro ligeiramente maior, mas obteve um melhor desempenho em séries com alta variação. Além disso, os modelos convolucionais e *Ensemble* foram os que apresentaram os menores erros tanto nos treinos quanto nos testes.
Palavras-Chave: Aprendizado de Máquina — Mercado Financeiro — Previsão do Preço — Agrupamento de Modelos

[1] *Departamento de Ciência da Computação, Programa de Pós-Graduação em Estatística Aplicada e Biometria (PPGEAB), Universidade Federal de Alfenas, Alfenas, Minas Gerais, Brasil*
*__Corresponding author__: claudio.estevam@sou.unifal-mg.edu.br
DOI: https://doi.org/10.22456/2175-2745.136072 • **Received:** 09/10/2023 • **Accepted:** 20/07/2024

1. Introdução

O termo "Big Data" tem ganhado destaque nos últimos anos, visto que houve um grande avanço tecnológico dos hardwares, que estão se tornando cada vez mais capazes de lidar com o imenso volume de dados que é gerado a cada segundo. Houve também um aumento significativo na captação de dados digitais gerando uma grande quantidade de dados [1].

Essa crescente combinação de recursos, ferramentas e aplicativos tem profundas implicações no campo da modelagem financeira, voltada para previsão de tendências de mercado, apresentando uma oportunidade e um desafio para pesquisadores. De fato, mais dados foram registrados nos últimos dois anos do que em todos os anos anteriores desde o início da história humana. *Big data* está sendo usado para aprimorar diferentes campos como medicina, engenharias, economia, entre outas áreas [2].

Atualmente, existe uma grande quantidade de dados que são gerados a todo segundo vindos de diferentes fontes. Esse aglomerado de informações é maior que a capacidade humana de analisar e tirar conclusões a partir deles. Para gerenciar esses conjuntos de dados novos e potencialmente relevantes, foram desenvolvidos novos métodos de inteligência artificial (IA), mais precisamente ciência de dados, e novos aplicativos na forma de análise preditiva. Como exemplo, bibliotecas construídas em *Python* e *R* com o foco no tratamento e ajuste de modelos visando a extração de informações relevantes de um conjunto de dados brutos.

O mercado financeiro é um dos setores que mais gera dados, direta e indiretamente. Ele é um conjunto de instituições, produtos e serviços que visam atender às necessidades financeiras de pessoas, empresas e governos. É composto por vários segmentos, como o mercado de ações, títulos, câmbio e commodities. Seu objetivo é canalizar recursos financeiros de investidores para tomadores de crédito. Isso é feito por meio da emissão e negociação de títulos, como ações, títulos de dívida e outros [3].

No mercado acionário o valor negociado de uma empresa é determinado não somente pelo seu valor físico, mas também pelo seu potencial de gerar lucros e crescimento financeiro. Além disso, há inúmeros fatores que fazem esse valor oscilar como as variáveis macroeconômicas, crises setoriais e até efeitos da natureza. Isso tudo somado com as especulações realizadas pelos investidores tornam a evolução do preço algo totalmente incerta. Porém, uma ferramenta capaz de realizar a previsão do preço se torna algo de grande valor agregado, ainda que não seja exata, mas se aproxime da realidade [4].

No mercado financeiro, a inteligência artificial (IA) tem sido cada vez mais utilizada para melhorar a tomada de decisões de investimento. Os modelos de IA podem analisar grandes volumes de dados de mercado e identificar padrões e tendências que podem ser úteis para prever o desempenho futuro de ativos. Além disso, os modelos de IA podem ser usados para automatizar tarefas e processos financeiros, o que pode aumentar a eficiência e reduzir os custos [4].

Neste artigo o foco será desenvolver uma aplicação de aprendizado de máquina dentro do mercado financeiro brasileiro, com o objetivo de realizar a previsão do valor de mercado atingido por 8 empresas, que representam as instituições do setor financeiro da bolsa. A previsão será realizada com base no histórico do preço de fechamento utilizando para cada previsão a entrada corresponde aos últimos 60 dias imediatamente anteriores aos dias previstos.

Foram escolhidas 3 arquiteturas de redes neurais a primeira é a *Multilayer Perceptron*, que [5] mostrou ser aplicável em dados que variam ao longo do tempo. O segundo modelo é o LSTM que é especificamente desenhado para armazenar valores passados e dar importância a sequência com que os dados são apresentados, esse modelo segundo [6] se mostrou eficiente para essa aplicação. Por último será ajustado um modelo convolucional que foca em extrair e manipular trechos da série para prever o seu comportamento futuro, como realizado por [7].

Por fim todos os modelos gerados serão reunidos em um *Ensemble* com o objetivo de alcançar uma eficiência maior que os modelos individuais. Sendo a saída de cada modelo um vetor de tamanho 20, a média das saídas será a resposta do agrupamento. Ou seja, todos iram dar uma resposta que esperam ser a correta para o próximo dia subsequente, a média destas respostas será a previsão do *Ensemble* para este dia, a mesma lógica se repete para os demais dias previstos.

Para o ajuste dos modelos serão usados os dados temporais, tendo como referência o preço de fechamento diário de 3 anos. Tais dados são públicos e podem ser encontrados com facilidade na própria bolsa de valores [8].

Este artigo está dividido em cinco partes principais, iniciando com esta introdução, seguida pela seção 2, com o arcabouço teórico utilizado e as justificativas de escolha dos métodos. Posteriormente, a seção 3 aborda os métodos e traz a construção do modelo e quais fatores foram considerados. Em seguida, a seção 4 apresenta os resultados encontrados ao aplicar o algoritmo em uma situação real. Por fim, são apresentadas as conclusões com os ajustes necessários e a constatação sobre a validade dos modelos.

2. Referencial

2.1 Mercado financeiro

O mercado financeiro é composto pela bolsa de valores, títulos, câmbio, commodities e outros ativos financeiros, onde os compradores e vendedores se reúnem para negociar esses ativos. Ele é usado como uma forma de investimento e financiamento para indivíduos, empresas e governos. O mercado de ações é um dos principais mercados, onde as ações de empresas são negociadas e os investidores compram e vendem ações com o objetivo de obter lucro com a valorização das suas participações. Outros mercados incluem o de títulos, onde os investidores compram e vendem títulos de dívida emitidos por governos e empresas, e o de câmbio, onde as moedas são negociadas [3]. O mercado financeiro é altamente volátil, imprevisível e sujeito a influências internas e externas, como

mudanças econômicas, políticas governamentais e eventos mundiais [9].

A B3 (Brasil, Bolsa, Balcão) é a bolsa de valores oficial do Brasil e é responsável por negociar ações, títulos e outros ativos financeiros, além de fornecer informações e serviços relacionados aos mercados financeiros. A B3 foi fundada em 2017, a partir da fusão da Bolsa de Valores de São Paulo (Bovespa) e da BM&F Bovespa. É uma das principais bolsas de valores do mundo, sendo responsável pela negociação de ações de muitas das principais empresas brasileiras, incluindo empresas de setores como petróleo, mineração, bancos, entre outros [8].

A teoria do mercado eficiente definida em [10] propõe que um ativo possui um preço que reflete o seu valor real, ou seja, o único fator que influencia a variação do preço seria a valorização ou desvalorização do quanto aquele bem possui de valor, sem influência de especulações ou manipulações de mercado. Para essa teoria preço e valor são considerados iguais, na maioria das vezes, e as que não são iguais são raras e impossíveis de serem exploradas.

Essa teoria foi revista pelo próprio autor vinte anos mais tarde em [11] onde foi reconhecida a influência de outros fatores no preço real de um ativo negociado na bolsa, sendo assim os preços variam de forma aleatória sem uma variável que possar ser bem definida, capaz de especular qual a próxima direção dos gráficos com precisão todas às vezes. Contudo há métodos que podem ser explorados com o objetivo de levar mais informações sobre o futuro de ativos.

Para previsão do mercado financeiro exitem duas técnicas clássicas que se destacam, a analise técnica e fundamentalista. A primeira, tem como foco encontrar padrões gráficos de suporte e resistências, que são obtidos com valores anteriores do preço [12]. Ou seja, ela foca em inferir sobre o futuro comportamento do preço, baseando-se em como ele se comportou no passado, que pode ser recente ou longo dependendo do estudo e dos índices considerados.

Já a análise fundamentalista obtém a situação da empresa e tira disso seus indicadores para mensurar o quão bom é um investimento dado o preço negociado na bolsa [13]. Seu direcionamento é em mensurar principalmente três campos a lucratividade, estrutura de capital e eficiência operacional. [14]. E com essas informações definir um preço justo por ativo que quando comparado com o preço atual deve ser realizado, ou não, a compra.

Embora a análise fundamental seja um método comprovado que tem sido usado com sucesso há décadas, a inteligência artificial tem vantagens distintas na previsão dos mercados financeiros. Isso se deve à sua capacidade de processar grandes quantidades de dados financeiros, econômicos e de negócios com rapidez e precisão, o que ajuda a prever tendências de mercado com mais eficiência. Vale ressaltar que, ela é capaz de analisar dados não financeiros, como sentimento de mídia social, dados meteorológicos e de tráfego, etc., que também podem ser correlacionados com preços previstos e podem ser adicionados à análise.

Além disso, a IA é capaz de descobrir padrões não lineares complexos em dados financeiros que os humanos não conseguem detectar, levando a previsões mais precisas e melhores oportunidades de investimento. Ela pode aprender com os dados e atualizar suas previsões à medida que o mercado muda, enquanto os modelos de análise fundamentalista são baseados em suposições e fatores estáticos. Mas é preciso ressaltar que a inteligência artificial não é infalível, sendo impossível prever com absoluta certeza o que vai acontecer no mercado financeiro, pois existem muitas incertezas e volatilidade neste campo.

Com base nos benefícios da IA para a previsão de mercados instáveis, este artigo propõe uma solução baseada em aprendizado de máquina para melhorar a precisão e eficiência das previsões. A solução se concentra em analisar séries temporais e agrupamento de modelos, que são técnicas que podem identificar padrões e tendências complexas nos dados financeiros.

Ao analisar séries temporais, é possível identificar padrões e tendências que podem ser usados para prever o comportamento futuro do mercado. O agrupamento de modelos, por sua vez, permite combinar diferentes modelos de previsão para obter uma previsão mais precisa.

Esta solução é diferente dos métodos tradicionais de previsão de mercado, que geralmente se concentram em fatores econômicos e financeiros estáticos. Isso pode oferecer uma vantagem competitiva para os investidores, fornecendo previsões mais precisas e eficientes.

2.2 Aprendizado de máquina em finanças

Para contextualizar o aprendizado de máquina dentro da área de conhecimento, o autor [15] estabelece uma distinção fundamental entre duas subáreas da inteligência artificial: *Machine learning* (aprendizado de máquina) e *Deep Learning* (aprendizado profundo). No *Machine learning*, o foco está na criação de modelos matemáticos que podem aprender a partir de dados, utilizando uma variedade de técnicas, como regressão, árvores de decisão e métodos estatísticos. Esses modelos são capazes de fazer previsões e classificações com base nos padrões extraídos dos dados. Por outro lado, o *Deep Learning* concentra-se na utilização de redes neurais artificiais profundas que emulam a estrutura do funcionamento da inteligência humana. Essas redes são particularmente eficazes na extração de representações complexas dos dados, sendo amplamente empregadas em tarefas que envolvem grande volume de dados, como reconhecimento de padrões em imagens, processamento de linguagem natural e análise de áudio. Ambas as disciplinas desempenham papéis cruciais na ciência da computação e têm aplicações variadas em setores que vão desde a medicina até a indústria automobilística [16].

O artigo [17] é uma revisão sistemática da literatura que analisa o uso de técnicas de *deep learning* para a previsão de séries temporais financeiras. O artigo apresenta uma revisão detalhada dos artigos publicados entre 2005 e 2019, abrangendo diferentes aplicações, como previsão de preços

de ações e taxas de juros, bem como diferentes tipos de dados, como dados de mercado e notícias. Ele também discute os desafios e limitações enfrentados ao usar o aprendizado profundo para previsão de séries temporais financeiras, como lidar com dados não estacionários e altamente voláteis. A conclusão geral do artigo é que estas técnicas têm mostrado desempenho promissor na previsão de séries temporais financeiras, mas ainda há espaço para melhorias e desenvolvimento de novos algoritmos.

No mercado de financeiro existem dois principais fatores que devem ser entendidos pelo modelo para que seja eficaz, a volatilidade e a incerteza, em [5] os autores identificam essas condições e propõem uma abordagem para prever o preço de fechamento de ações usando algoritmos de aprendizado supervisionado, como redes neurais e árvores de decisão. Os autores mostram como essas técnicas foram aplicadas a dados históricos de ações de diferentes empresas e comparam os resultados obtidos. Eles também discutem como as técnicas de aprendizado de máquina podem ser usadas para lidar com problemas como a falta de dados e a alta dimensionalidade.

O artigo citado usou como entrada para os modelos indicadores técnicos como a média móvel e o desvio padrão para diferentes intervalos de tempo, por fim concluiu que as técnicas de aprendizado de máquina são eficazes para prever o preço de fechamento de ações e podem ser usadas como ferramentas valiosas para investidores. Semelhante a conclusão que chegou [18], e completa colocando que aprendizado de máquina é um campo que pode gerar bons frutos na bolsa de valores, mas ainda há espaço para melhorias e desenvolvimento futuro.

Outros autores também aplicaram aprendizado de máquina em dados de ações como em [19] que foi utilizada janelas de tempo do mercado de criptomoedas, utilizou-se o preço de fechamento do *Ethereum*, como entrada para dois tipos de modelo, regressão logística e SVM, para prever a direção do mercado, neste caso foi utilizado a acurácia como medida de desempenho e concluiu que para esses dados o modelo SVM foi superior em 13,54% das vezes. Tal modelo também foi ranqueado em primeiro na pesquisa feito por [20] que traz mais modelos a fim de comparar seus desempenhos, aqui a rede neural recorrente com arquitetura LSTM ficou em segundo lugar, a frente da rede neural com camadas totalmente conectadas. Neste último os modelos citados foram superiores ao modelo *Fuzzy*, GARCH, ARIMA entre outros

Nos estudos apresentados anteriormente foram usados apenas as séries temporais dos ativos, contudo os dados também podem ser obtidos em fontes externas ao mercado, desde que se referindo a ele, como exemplo, a análise de sentimentos que busca com base nas reações dos investidores inferir sobre um possível direcionamento do mercado. Essa proposta é discutida em [21] e experimentada por [22] no qual os autores discutem uma abordagem para prever o mercado de ações usando dados de preços de ações e sentimentos de notícias. Eles coletaram dados de preços de ações e sentimentos das notícias de fontes on-line e usaram técnicas de aprendizado de máquina para prever o desempenho futuro do mercado de ações. Além disso, testaram sua abordagem usando dados do mercado de ações de Hong Kong e mostraram que sua abordagem é capaz de prever o desempenho do mercado com precisão. Por fim, concluem que a incorporação de sentimentos de notícias pode ser uma ferramenta valiosa para melhorar as previsões do mercado de ações.

2.3 Redes neurais artificiais

Redes neurais artificiais (RNA) são modelos computacionais inspirados na estrutura e funcionamento do sistema nervoso biológico. Elas são compostas por várias camadas de neurônios interconectados, que são responsáveis por processar dados e tomar decisões. Esses neurônios são conectados por sinapses, que são encarregados de transmitir informações entre os neurônios [23].

Os neurônios de uma RNA são divididos em camadas, sendo que cada camada é responsável por uma tarefa específica. A camada de entrada é responsável por processar os dados de entrada, a camada oculta por realizar operações complexas e a camada de saída produzir a resposta final.

Todas as redes neurais são treinadas com dados históricos, o que permite que elas aprendam como processar e tomar decisões com base no passado. O treinamento geralmente é realizado usando algoritmos de aprendizado de máquina, como o gradiente descendente, que ajusta os pesos das conexões entre os neurônios de acordo com a precisão da saída da rede [24].

Além disso, as redes neurais artificiais são capazes de aprender de forma não monitorada, onde elas são expostas a dados sem rótulos e aprendem a identificar padrões e características importantes. Isso é conhecido como aprendizado não supervisionado.

Outra característica importante das redes neurais artificiais, segundo [25], é a sua capacidade de generalizar o aprendizado para novos dados. Isso significa que, após o treinamento, a rede é capaz de fazer previsões precisas sobre novos dados, mesmo que essas entradas nunca tenham sido vistos antes.

O livro [16] apresenta as principais características das três principais arquiteturas de redes neurais profundas: redes neurais *feedforward*, redes neurais recorrentes e redes neurais convolucionais.

Redes neurais *feedforward* são as arquiteturas mais simples e amplamente utilizadas, compostas por camadas de neurônios interconectadas que processam a entrada de forma linear. Elas são utilizadas para tarefas de classificação, processamento de linguagem natural e análise de imagens.

O *Multilayer Perceptron* (MLP) é uma arquitetura de rede neural do tipo *feedforward* que possui aprendizado supervisionado e resolve problemas de classificação e regressão. É uma extensão do modelo *Perceptron*, proposto pelo psicólogo Frank Rosenblatt em 1958. O MLP é como uma rede multicamadas formadas por *Perceptrons* com uma ou mais camadas ocultas de neurônios, além de uma de entrada e uma de saída [26].

No MLP, a entrada recebe um conjunto de dados, que são propagados pelas camadas ocultas até a saída, onde é produzida uma estimativa que representa a resposta esperada. Cada neurônio da camada oculta recebe um conjunto de entradas, que são multiplicadas pelos pesos e adicionadas aos valores de viés. Esse valor é então passado por uma função de ativação, como a função sigmóide, ReLU, Leaky ReLU, entre outras. o resultado dessa função decide se o neurônio deve ser ativado ou não. O processo de treinamento de um MLP envolve o ajuste dos pesos e vieses dos neurônios para que o modelo possa fazer boas previsões. Isso é feito por meio de um processo chamado backpropagation, no qual os erros na saída do modelo são propagados para as demais camadas e os pesos e vieses são atualizados para reduzir esses erros [24].

Redes neurais recorrentes são arquiteturas que possuem ciclos e conexões de *feedback*, permitindo que elas lidem com séries temporais e processamento de linguagem natural. Elas são compostas por células de memória, que são responsáveis por armazenar estados e informações passadas.

Redes neurais convolucionais são arquiteturas de redes neurais que foram projetadas para lidar com imagens e sinais espaciais, elas possuem camadas de neurônios que são conectadas de forma a compartilhar pesos. Isso permite que a rede aprenda características e padrões comuns em imagens e sinais, tornando-as muito eficazes para tarefas de classificação e detecção de objetos.

Long Short-Term Memory (LSTM) é uma rede neural recorrente que é capaz de aprender e lembrar informações de longo prazo. Suas células de memória internas permitem que a rede retenha informações passadas por um longo período de tempo. Isso é importante para tarefas que requerem a compreensão de sequências longas de dados, como tradução automática, reconhecimento de fala e processamento de linguagem natural [27].

A estrutura interna da LSTM inclui três tipos de portas: portas de esquecimento, portas de entrada e portas de saída. Essas portas são controladas por sinais de *gating*, que são calculados a partir da entrada e estado anterior da célula.

A LSTM é capaz de lidar com dependências de longo prazo, o que é difícil para as redes neurais recorrentes convencionais, pois elas não possuem uma forma de controlar o que é armazenado na memória e o que é descartado. Isso faz com que a LSTM seja uma boa escolha para tarefas como previsão de séries temporais, processamento de linguagem natural e tradução automática [22].

O algoritmo de *Ensemble* é uma técnica de aprendizado de máquina que combina diferentes modelos em um, que geralmente é mais preciso. O objetivo do *Ensemble* é aumentar a precisão e a robustez do modelo final, reduzindo a variância e o viés dos modelos individuais. O algoritmo é amplamente utilizado em várias aplicações incluindo classificação, regressão e clusterização.

O algoritmo de gradiente descendente, segundo [25], é um método de otimização utilizado para encontrar os parâmetros ótimos de uma função de custo. Ele funciona atualizando os parâmetros de forma iterativa na direção oposta ao gradiente da função de custo em relação aos parâmetros.

Em cada iteração, os parâmetros são atualizados pela seguinte equação:

$$\theta = \theta - \alpha * \nabla\theta J(\theta) \tag{1}$$

Onde θ é o vetor de parâmetros, J(θ) é a função de custo, α é a taxa de aprendizado (um parâmetro de controle que determina o tamanho do passo na direção oposta ao gradiente) e $\nabla\theta J(\theta)$ é o gradiente da função de custo em relação aos parâmetros.

Existem variações do gradiente descendente, como o gradiente descendente estocástico (SGD) e o gradiente descendente com momento. O SGD atualiza os parâmetros com base em amostras individuais de dados, enquanto o gradiente descendente com momento adiciona uma ”inércia”para evitar oscilações em torno do mínimo global.

A eficácia do algoritmo de gradiente descendente depende de vários fatores, incluindo a escolha da função de custo, a inicialização dos parâmetros e a escolha da taxa de aprendizado. Alguns artigos científicos sugerem que técnicas como o uso de um decaimento adaptativo da taxa de aprendizado e o uso de métodos de regularização podem melhorar o desempenho do algoritmo [25].

O cálculo dos gradientes dos pesos de uma rede neural pode ser feito utilizando o algoritmo de *backpropagation*, que é utilizado para que esses pesos possam ser atualizados de forma a minimizar a função de custo. O algoritmo funciona propagando o erro calculado na saída da rede de volta para as camadas intermediárias e de entrada, através da aplicação da regra da cadeia. Isso permite calcular os gradientes dos pesos de forma eficiente e precisa [25].

O artigo [28] realiza uma análise sobre as técnicas de otimização usadas em *deep learning*, incluindo o *backpropagation* e suas variações. O artigo descreve o *backpropagation* como um algoritmo de otimização baseado em gradiente que é amplamente utilizado para treinar redes neurais *feedforward*. Ele explica que o aprendizado é feito pelo cálculo do gradiente da função de perda em relação aos pesos da rede neural e atualizando os pesos de acordo com esse gradiente.

O algoritmo funciona realizando dois passos: primeiro, calcula a saída da rede para uma dada entrada e, em seguida, calcula o erro entre a saída desejada e a saída obtida. Em seguida, o erro é propagado de volta através da rede, com o objetivo de calcular o gradiente da função de perda em relação aos pesos. Esse processo é repetido para cada amostra de treinamento, atualizando os pesos a cada iteração.

O texto ainda conclui que este algoritmo de aprendizado tem como desvantagens a necessidade de grande quantidade de dados para treinamento, a dificuldade em lidar com dados ruidosos e a vulnerabilidade a mínimos locais. Ainda assim, é considerado um algoritmo eficiente e de ampla utilização.

Exite uma variação do *backpropagation* desenvolvido para trabalhar especialmente com redes do tipo recorrente. Cha-

mado de *Backpropagation through time* (BPTT). Segundo o autor [29] a diferença entre redes neurais *feedforward* e redes neurais recorrentes é que, enquanto as redes *feedforward* não possuem ciclos, as redes recorrentes possuem ciclos e conexões de *feedback*, o que torna o cálculo do gradiente para o treinamento mais complexo.

O artigo explica que a BPTT é uma extensão do *backpropagation* convencional para lidar com redes neurais recorrentes. Ele consiste em calcular o gradiente da função de perda em relação aos pesos e vieses da rede neural utilizando a regra da cadeia, mas aplicado a cada instante temporal. Isso significa que é necessário armazenar e calcular o gradiente para cada estado da rede no tempo.

O artigo também discute as dificuldades da BPTT, como a necessidade de armazenar muitos estados para cálculos, o que pode tornar o treinamento muito custoso em termos de memória, e a dificuldade de lidar com gradientes explosivos ou gradientes que vão a zero. Ele também apresenta algumas técnicas para lidar com esses problemas, como o uso de truncamento temporal para limitar a quantidade de estados armazenados e a utilização de regularização para evitar gradientes explosivos.

No ajuste do modelo há três possibilidades em relação ao seu desempenho no conjunto de teste em relação ao conjunto usado no treinamento, sendo que o modelo pode sobre ajustar (*Overfitting*), subajustar (*Underfitting*), ou ter o ajuste ideal.

Overfitting ocorre quando um modelo é "ajustado demais" aos dados de treinamento, ou seja, quando ele se adapta muito bem aos dados de treino, mas não é capaz de generalizar bem para dados novos. Isso acontece quando o modelo aprende os ruídos e variações específicas, em vez de aprender os padrões gerais subjacentes. Isso pode levar a desempenho fraco quando o modelo é usado com dados inéditos [30].

Essa condição pode ser evitada, segundo [31], com uso de modelos menos complexos, técnicas de regularização e validação cruzada. A regularização é uma técnica que limita a capacidade do modelo de se ajustar aos ruídos nos dados de treinamento, enquanto a validação cruzada é uma técnica utilizada para avaliar o desempenho do modelo com dados que não foram utilizados no treino. Isso ajuda a identificar o *overfitting* e ajustar o modelo de forma apropriada. Além disso, o número de características utilizadas no modelo também pode ser um problema, para isso é sugerindo a seleção de características para tornar o modelo mais parcimonioso e menos complexo.

Por outro lado, *underfitting* ocorre quando o modelo não é "ajustado o suficiente" aos dados de treinamento, ou seja, quando ele não se adapta bem. Isso acontece quando o modelo é muito simples ou não tem capacidade suficiente para aprender os padrões subjacentes dos dados. Isso pode levar a baixa precisão e alto erro geral.

O *underfitting* pode ser evitado, com uso de modelos mais complexos que possam capturar melhor as relações existentes nos dados de treinamento. Há também a possibilidade de utilizar técnicas para aumentar a quantidade de dados de treinamento, como o uso de dados gerados artificialmente. Além disso, a validação cruzada é uma técnica útil para avaliar o desempenho de um modelo e identificar o subajuste [31].

O conceito de ajuste ideal encarna um ponto de equilíbrio essencial, no qual o modelo apresenta uma complexidade adequada para apreender com precisão os padrões subjacentes nos dados de treinamento, sem, contudo, adotar uma complexidade excessiva que o torne suscetível ao ruído inerente aos dados. Esse estado ideal é almejado devido ao seu potencial de gerar um modelo que não apenas se adapta de forma eficaz aos dados de treinamento, mas também é capaz de generalizar com acurácia para novos conjuntos de dados, produzindo resultados confiáveis e pragmaticamente úteis [32].

A dimensão da amostra, outro ponto importante a ser observado no ajuste, se refere ao número de dados utilizados para treinar um modelo. Quanto maior a dimensão da amostra, mais informação ele tem para aprender, o que pode resultar em resultados mais precisos. Por outro lado, se a dimensão é muito pequena, pode haver insuficiência de informações para realizar um treino satisfatório.

Os autores [33] sugerem que a escolha da dimensão da amostra é importante na previsão do preço do Bitcoin. Eles analisam como diferentes tamanhos afetam a precisão do modelo e concluem que quantidade maior de dados geralmente resulta em previsões mais próximas da realidade. No entanto, eles também destacam que é importante não usar uma dimensão de amostra muito grande, pois isso pode levar a sobreajuste resultando em um modelo menos preciso.

Neste trabalho, o modelo *Multilayer Perceptron* foi proposto por ser capaz de aprender padrões complexos em dados de entrada e fazer previsões para novas entradas [34]. Além disso, a arquitetura do MLP permite que ele se adapte a diferentes tipos de relações entre as variáveis, o que é útil para séries temporais, já que as relações entre as variáveis podem mudar ao longo do tempo. Com isso ele pode ser treinado com dados históricos de séries temporais e fazer previsões futuras para ajudar a tomar decisões baseadas em tendências e padrões previstos [26].

Por outro lado, o modelo de rede neural recorrente com arquitetura LSTM foi escolhido pela capacidade de lidar com séries temporais. A rede possui uma unidade de memória interna que permite manter o contexto histórico das informações passadas ao longo do tempo [27]. Isso é importante no mercado financeiro, onde tendências e padrões podem ser altamente dependentes do histórico de preços. Além disso, ela consegue lidar com ruídos e incertezas nos dados, o que é comum em mercados financeiros voláteis [35].

Em seguida, foi aplicado a Rede Neural Convolucional de uma dimensão por ser projetada para lidar com séries temporais. Esse tipo de rede é comumente utilizada para trabalhar com dados de imagem, onde a dimensão espacial é importante, mas também podem ser adaptadas para lidar com séries temporais. Ela possui filtros que "deslizam" ao longo da série, permitindo que aprenda padrões e tendências ao longo do tempo [36].

Na implementação de redes neurais, o papel dos algoritmos de otimização é de suma importância para o ajuste dos parâmetros do modelo de forma eficaz. O objetivo é minimizar uma função de custo, frequentemente definida como a diferença entre as previsões do modelo e os valores verdadeiros. Diversos algoritmos de otimização têm sido propostos e empregados em aplicações práticas, cada um com suas próprias vantagens e desvantagens. O otimizador de Descida de Gradiente Estocástico (SGD) é talvez o mais básico e amplamente usado, mas pode ser lento e ineficiente em alguns cenários [37]. Algoritmos mais sofisticados como o Adam [38], o RMSprop [39], o Adamax [38] e o Adagrad [40] oferecem variações que podem acelerar a convergência e melhorar o desempenho do modelo. A escolha do otimizador adequado pode ser crucial e depende da natureza específica da aplicação e dos dados em questão.

Por fim, todos os modelos construídos foram reunidos em um modelo agrupado, essa técnica de aprendizado de máquina combina vários modelos para obter melhores resultados que o uso de apenas um modelo individual. A ideia é que a combinação de vários modelos independentes possa compensar as fraquezas de cada um e levar a uma melhor performance geral. Existem diferentes formas de combinar modelos, como votação, média, pesos, entre outras. Por se tratar de um vetor de preços correspondente ao intervalo de tempo futuro será utilizada a média da previsão realizada em cada dia pelos modelos.

3. Metodologia e Análise dos Dados

Neste artigo, foram selecionadas todas as empresas do setor Financeiro pertencentes à carteira de ações do índice Bovespa. As companhias selecionadas são: Banco Pan (BPAN4), Banco Bradesco (BBDC3, BBDC4), Banco do Brasil (BBAS3), Banco BTG Pactual (BPAC11), Itausa (ITSA4), Itaú Unibanco (ITUB4) e Banco Santander (SANB11), totalizando 8 papéis. A lista das empresas listadas pode sem encontrada em [41].

Este setor foi escolhido por ter uma boa participação no índice e as empresas foram selecionadas por serem consolidadas no mercado e estarem listadas na bolsa a anos, com isso há mais material para análise e entendimento do comportamento do preço. Além disso, por serem empresas do mesmo setor as variáveis que impactam uma geralmente afetam as outras, o que facilita na explicação de mudanças bruscas na tendência.

A escolha dos modelos de aprendizado de máquina utilizados neste estudo foi baseada em critérios específicos que visam capturar diferentes características das séries temporais e maximizar a precisão das previsões. O Multilayer Perceptron foi escolhido devido à sua capacidade de aprender padrões complexos em dados de entrada e fazer previsões para novas entradas. Este modelo é aplicável a dados que variam ao longo do tempo e pode se adaptar a diferentes tipos de relações entre as variáveis, sendo útil para séries temporais onde as relações podem mudar ao longo do tempo.

O Long Short-Term Memory foi selecionado por sua habilidade específica de lidar com dependências de longo prazo em séries temporais. A arquitetura LSTM é desenhada para armazenar valores passados e dar importância à sequência com que os dados são apresentados, tornando-se eficiente para aplicações que requerem a retenção de informações históricas. Esse modelo se mostrou eficaz para prever tendências em séries temporais financeiras, onde os padrões históricos têm um impacto significativo no comportamento futuro dos preços.

A Convolutional Neural Network foi escolhida por sua capacidade de extrair e manipular padrões locais nas séries temporais. Embora geralmente utilizadas em processamento de imagens, as CNNs podem ser adaptadas para séries temporais para capturar variações e tendências locais. Este modelo se destacou especialmente em séries com alta variação, proporcionando uma abordagem complementar às capacidades das MLPs e LSTMs. Esses modelos foram ajustados com cinco otimizadores distintos para explorar uma ampla gama de métodos de ajuste, resultando em um total de 15 modelos distintos. Posteriormente, todos os modelos foram combinados em um Ensemble para gerar previsões mais precisas e robustas do que os modelos individuais poderiam oferecer.

Os dados da variação de preço das ações são públicos e foram coletados através da biblioteca *yfinance* na linguagem de programação *Python*. Definiu-se o tempo de 3 anos iniciados em 30/01/2020 e finalizado em 30/01/2023 no qual o preço de fechamento diário foi filtrado para ser a série temporal utilizada na análise. Os papéis selecionados, assim como a análise descritiva dos dados iniciais, podem ser encontrados na Tabela 1.

Com o objetivo de avaliar a volatilidade intrínseca das séries temporais, foi empregado o coeficiente de variação (CV) como métrica principal. O CV é definido como o desvio-padrão da série temporal dividido pela sua média aritmética, multiplicado por 100. A relevância desta métrica deriva de sua capacidade de fornecer uma medida padronizada da dispersão relativa de uma distribuição, tornando-a imune às influências das unidades de medida empregadas [42].

A escolha do coeficiente de variação como métrica focal para este estudo é justificada por sua eficácia em possibilitar uma comparação direta e robusta da volatilidade entre séries temporais com diferentes escalas e médias de preços [43]. É notável que esta métrica tem sido aplicada extensivamente na literatura financeira para avaliar o risco relativo associado a diferentes classes de ativos.

A análise preliminar, Figura 1, revela que a empresa BPAN4 exibe um coeficiente de variação substancialmente mais elevado em comparação à empresa ITSA4 durante o período considerado. À primeira vista, um coeficiente de variação elevado poderia ser considerado um indicativo de maior volatilidade e, portanto, maior risco financeiro associado ao investimento [44].

Entretanto, em cenários de mercados ascendentes, a volatilidade elevada pode se traduzir em oportunidades significativas para investidores com perfis de risco e retorno mais

Tabela 1. Análise descritiva do fechamento diário do período considerado.

	BPAN4	BBDC3	BBDC4	BBAS3	BPAC11	ITSA4	ITUB4	SANB11
Média	10,67	16,7	19,19	33,79	22,65	8,9	26,5	33,72
Mediana	9,54	16,24	19,11	33,26	23,08	8,83	26,28	32,1
DP	4,84	2,29	2,59	4,48	5,07	0,83	3,03	5,53
CV	0,45	0,14	0,13	0,13	0,22	0,09	0,11	0,16
Mínimo	4,16	11,78	13,28	22,13	6,53	6,52	20,52	22,96
Máximo	25,58	24,61	25,85	51,62	32,27	11,58	35,07	46,19
Amplitude	21,42	12,82	12,58	29,49	25,74	5,06	14,55	23,23

agressivos. A interpretação do coeficiente de variação deve, portanto, ser complementada por uma análise rigorosa de variáveis externas, incluindo o contexto macroeconômico, políticas de governança corporativa, e indicadores setoriais que possam exercer influência sobre a volatilidade dos preços das ações em questão.

Conforme observado na Figura 1, é relevante destacar que a empresa ITSA4, que atua como uma holding, apresentou um coeficiente de variação menor em comparação com outras empresas que não possuem essa estrutura organizacional. Uma razão plausível para este fenômeno é a diversificação intrínseca associada às holdings. Essas empresas geralmente detêm participações em múltiplas subsidiárias que operam em diversos setores ou geografias. Esta diversificação pode atuar como um mecanismo de hedge, amortecendo o impacto de flutuações setoriais ou econômicas adversas em qualquer entidade individual do grupo. Como resultado, a volatilidade agregada dos retornos tende a ser menor, contribuindo para um CV mais baixo, que é interpretado como um indicativo de menor risco financeiro, uma característica que pode ser particularmente atraente para investidores com aversão a perda [45].

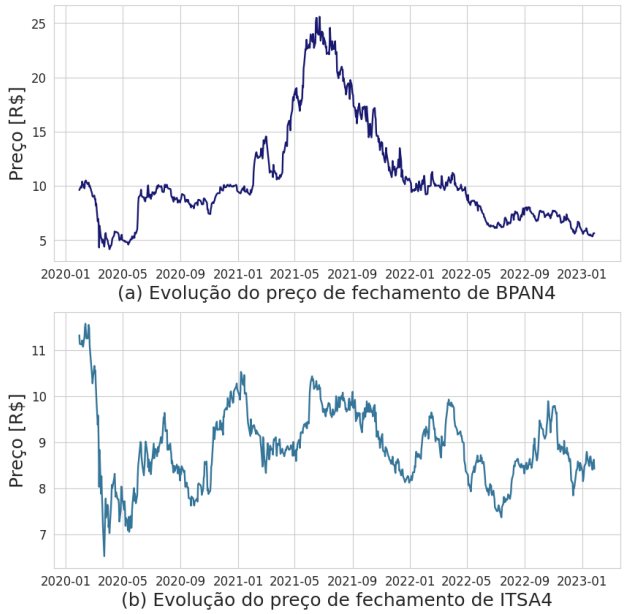

Figura 1. Evolução do preço de fechamento.

Para construir um banco de dados de forma que possua entradas e saídas optou-se pelo método de janela deslizante, que consiste em dividir o conjunto de dados em várias janelas de tamanho fixo e deslizar a janela ao longo dos dados para ter uma série de intervalos em sequência. Sendo 60 dias o tamanho do intervalo usado para os dados de entrada e 20 dias o tamanho de saída. Assim os modelos serão aplicados de forma que a saída seja um vetor que corresponde aos próximos vinte dias a contar a partir do último dia usado como entrada.

A segmentação dos dados em conjuntos específicos para treinamento, teste e validação é um passo crucial na modelagem de aprendizado de máquina. Esta divisão não apenas auxilia na calibração eficaz dos modelos, mas também fornece um meio para avaliar seu desempenho de forma objetiva [23]. O primeiro passo foi isolar 20%, corresponde aos últimos 149 dias considerados para ser o teste. Do restante tomou-se os 40 dias mais recentes deste conjunto como o banco de validação, os dados restantes serão usados no treinamento e correspondem a 596 valores. Esse processo foi igual para a serie temporal de todas as empresas.

Um ponto importante a se relatar é sobre a forma de separar os dados por se tratar de uma série temporal em que a mudança do valor com o passar dos dias é uma relação importante a se levar ao modelo, não seria ideal embaralhar os dados e sortear uma amostra para ser usada como treino, teste e validação. Pois esse efeito de aleatoriedade não iria expor a importância evolutiva dos preços. Por isso optou-se por segregar o conjunto de forma a separar uma porcentagem dos dados mais recentes.

Tendo os conjuntos segregados, foram aplicadas 3 tipos de transformações com o objetivo de obter um melhor desempenho no ajuste e ser possível comparar qual transformação obteve os melhores resultados. A primeira foi a padronização, que transforma a série de forma que ela passe a ter média zero e desvio padrão unitário, retirando assim a tendência e sazonalidade. pode ser visualizada na Equação 2, é obtida realizando o cálculo da diferença entre cada valor da série x e sua média μ, dividido pelo desvio padrão σ. Essa transformação pode ajudar a comparar séries temporais que possuem escalas diferentes e pode melhorar a precisão da modelagem, principalmente quando diferentes variáveis ou séries estão sendo usadas em conjunto para previsão. A padronização também pode ajudar a melhorar a convergência em alguns modelos de aprendizado de máquina, especialmente aqueles

que usam funções de ativação como a sigmóide ou a tangente hiperbólica, que são mais sensíveis à escala dos dados. No entanto, é importante lembrar que a padronização não melhora a qualidade dos dados, ou seja, ela não resolve problemas como outliers, tendências, sazonalidades ou outros padrões não desejados presentes na série.

$$y = \frac{x - \mu}{\sigma} \tag{2}$$

A segunda, foi a transformação logarítmica, Equação 3, com sua aplicação busca-se estabilizar a variância da série e reduzir os efeitos de outliers extremos, o que pode melhorar a capacidade de modelagem e aprendizagem do modelo. Também pode ajudar a tornar a relação entre os valores da série mais linear e, portanto, mais fácil de modelar. Em algumas situações, a transformação logarítmica pode ajudar a remover tendências e sazonalidades não desejadas da série.

$$y = \ln(x) \tag{3}$$

A última foi a transformação Yeo-Johnson, que pode ser aplicada em séries temporais para tornar a distribuição dos dados mais simétrica e aproximadamente normal. Ela é uma extensão da transformação de Box-Cox, que não funciona para dados negativos ou zero. A transformação Yeo-Johnson pode ser observada na Equação 4, onde x é o valor original da série e λ é um parâmetro que controla a transformação.

$$\Psi(\lambda, y) = \begin{cases} \frac{(y+1)^{\lambda}-1}{\lambda}, & \text{se } \lambda \neq 0, y \geq 0 \\ \log(y+1), & \text{se } \lambda = 0, y \geq 0 \\ \frac{-[(-y+1)^{(2-\lambda)}-1]}{2-\lambda}, & \text{se } \lambda \neq 2, y < 0 \\ -\log(-y+1), & \text{se } \lambda = 2, y < 0 \end{cases} \tag{4}$$

Um exemplo de como as modificações afetam a série pode ser observado na Figura 2, na qual são apresentados os gráficos de box-plot que descrevem o comportamento da série e suas variações. Na figura (a), temos a série original sem qualquer modificação, na qual se observa uma forte assimetria e a presença de diversos valores atípicos (*outliers*). Essas características são praticamente idênticas à série padronizada, mostrada na figura (b), diferindo apenas no intervalo de variação dos dados. No entanto, à medida que são realizadas as modificações, esses atributos são atenuados. Por exemplo, a série logaritmizada, representada na figura (c), tende a exibir uma simetria maior e possui menos valores atípicos. Esse efeito se atenua quando aplicamos a transformação de Yeo-Johnson, como mostrado na figura (d), na qual os valores atípicos praticamente desaparecem, resultando em uma distribuição mais simétrica dos dados. Esse padrão se repetiu nas séries de outras empresas.

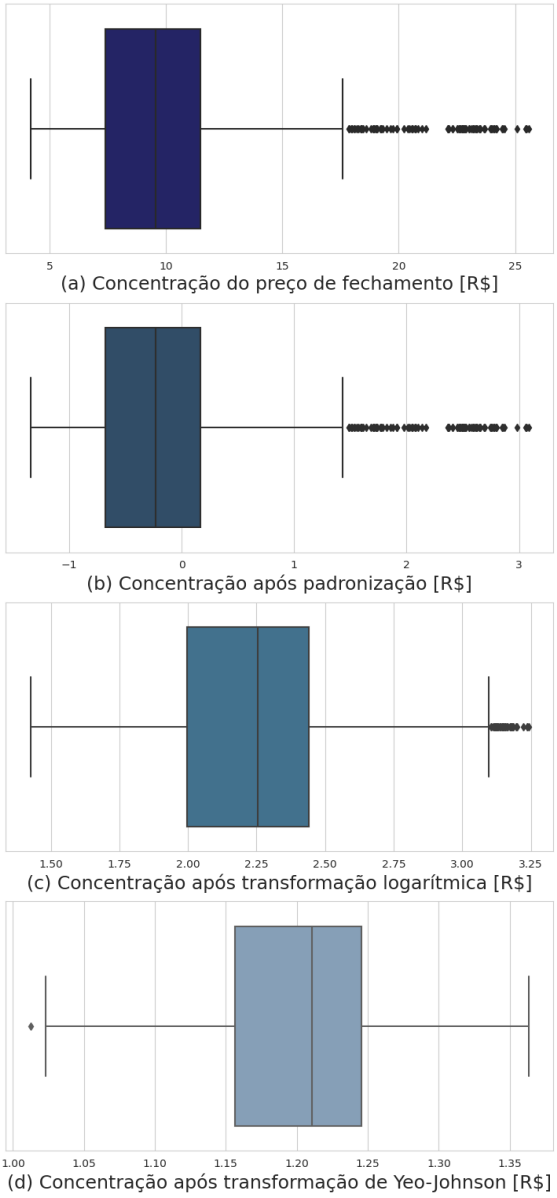

Figura 2. Concentração do preço fechamento do Banco Pan.

4. Resultados e Discussões

Três diferentes estruturas de redes neurais foram ajustadas usando uma variedade de cinco otimizadores: Adam, SGD, RMSprop, Adamax e Adagrad. Estes otimizadores foram escolhidos para oferecer uma gama diversificada de métodos de ajuste. Além disso, cada arquitetura foi exposta a três séries temporais que passaram por diferentes transformações. Esta abordagem foi adotada para tornar os modelos independentes, permitindo que cada um enxergasse os dados de uma forma diferente. Ao final do processo, cada transformação resultou em um conjunto de 15 modelos distintos. Estes foram então combinados em um único "modelo coletivo"com a intenção de gerar previsões mais acuradas do que os modelos individuais poderiam oferecer.

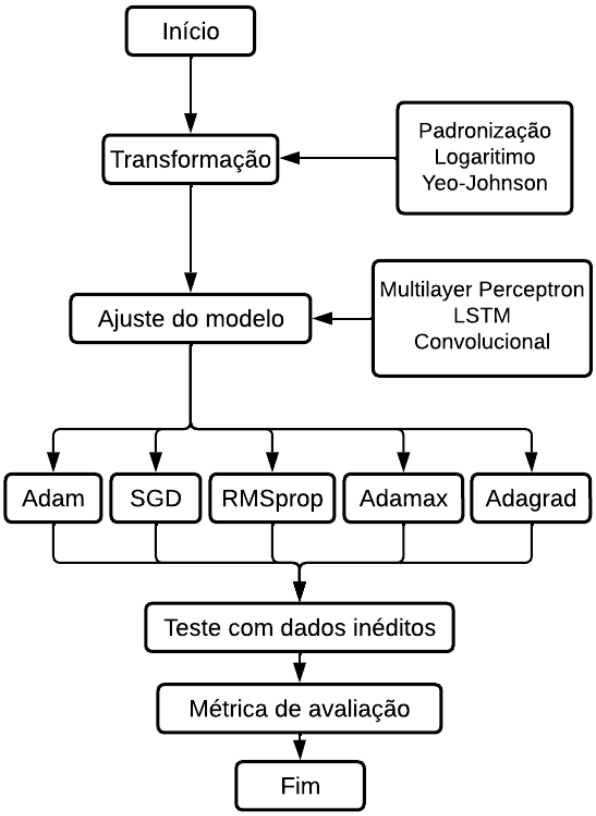

Figura 3. Etapas realizadas para ajuste dos modelos.

O fluxograma, ilustrado na Figura 3, delineia a metodologia empregada para a implementação e avaliação dos modelos de aprendizado de máquina construídos. O pipeline inicia-se com a etapa de "Transformação dos Dados", onde uma normalização é aplicada para garantir que os dados se aproximem de uma distribuição normal. Esta fase é seguida pela etapa de "Ajuste do Modelo", que utiliza uma arquitetura de rede neural específica e usa para cada variação um otimizador. Esses algoritmos de otimização convergem para a etapa de "Teste com Dados Inéditos", que subsequentemente alimenta a fase de "Métrica de Avaliação" para a quantificação objetiva do desempenho do modelo.

Os modelos foram treinados com o critério de parada sendo 500 épocas ou o aprendizado não estar melhorando por 30 gerações consecutivas, esses parâmetros foram definidos com o objetivo de evitar o sobre ajuste e o gasto desnecessário de processamento com o treinamento sendo que está em uma tendência de estagnar, ou seja, chegando a um mínimo.

O método de avaliação e comparação dos modelos escolhidos foi o Erro Percentual Médio Absoluto (MAPE) que é uma medida de precisão usada para avaliar a acurácia de previsões ou projeções em relação aos valores reais. O cálculo do MAPE envolve a comparação entre as previsões e os valores reais correspondentes para cada ponto de dados, no qual o erro percentual absoluto é calculado como a diferença absoluta entre o valor previsto ($\hat{y}$) e o valor real (y), dividido pelo valor real. Em seguida, os erros percentuais absolutos para todos os pontos de dados são somados e divididos pelo número de amostras existentes (n), a fórmula pode ser observada na Equação 5 [46].

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y-\hat{y}}{y}\right| \tag{5}$$

Ajustando os modelos para as séries que foram padronizadas gerou-se os resultados, mensurado pelo MAPE, que podem ser observados nas Tabelas 2 e 3, que apresentam os erros encontrado para o conjunto de treino e teste respectivamente. Analisando-a, é possível concluir que o melhor modelo unitário foi o convolucional com um erro médio de 0,89, este modelo foi superior inclusive ao modelo *Ensemble* que neste caso apresentou o maior erro. Isso pode ser reflexo de um mau ajuste dos modelos, ou seja, os algoritmos não estão sendo eficientes em se adequar à tarefa.

No geral, os modelos construídos a partir de dados padronizados não performaram tão bem, pois apesar do *Ensemble* ter obtido um melhor desempenho conjunto de teste, com erro médio reduzido para 1,5, se for observado as séries individuais é possível perceber que o erro tem uma grande variação, isso mostra que o algoritmo se ajustou melhor a algumas séries como BPAN4 com erro de 0,44 em detrimento de outras como ITUB4 que teve o erro de 4,47. Uma possível causa disso é a distribuição dos dados e presença de outlier, que como apresentado não foram removidos com a padronização, além de possuir distribuição assimétrica, de forma a dificultar o aprendizado da rede neural.

Tabela 2. Comparativo do MAPE no conjunto de treino após a padronização.

Códigos	LSTM	MLP	Conv.	Ensemble
BPAN4	1,53	0,99	0,56	0,48
BBDC3	1,42	1,23	0,70	1,87
BBDC4	1,05	1,00	0,64	6,09
BBAS3	2,75	1,88	1,49	1,55
BPAC11	1,88	1,85	0,82	1,92
ITSA4	2,02	1,54	1,09	1,77
ITUB4	1,70	1,80	1,06	2,14
SANB11	1,19	1,10	0,80	0,50
Média	**1,70**	**1,42**	**0,89**	**2,04**

Contudo, os erros obtidos com a padronização foram exponencialmente maiores que os encontrados nas demais transformações, como pode ser observado nas Tabelas 4 e 5. Após aplicação da função logarítmica nos dados, os modelos tiveram um melhor desempenho, como no caso do *Ensemble* no conjunto de testes, que passou de 1,5, padronizado, para 0,051, logaritmizado, o que representa uma redução de 96.60%. Esse significativo ganho de eficiência repetiu-se nos demais modelos individuais, tanto no treinamento quanto nos testes.

Tabela 3. Comparativo do MAPE no conjunto de teste após a padronização.

Códigos	LSTM	MLP	Conv.	Ensemble
BPAN4	0,70	0,33	0,35	0,44
BBDC3	0,97	0,60	1,13	0,82
BBDC4	1,82	1,48	2,61	1,28
BBAS3	1,91	1,56	2,50	1,68
BPAC11	1,99	1,98	1,92	1,22
ITSA4	1,95	1,70	1,92	1,61
ITUB4	3,94	4,24	6,65	4,47
SANB11	0,65	0,43	0,47	0,50
Média	**1,74**	**1,54**	**2,19**	**1,50**

Essa melhora pode ser justificada pelo melhor tratamento aplicado nos dados de entrada, pois como apresentados na Figura 2 os dados estão mais simétricos e próximos de uma distribuição normal, o que facilita o aprendizado dos algoritmos, além de reduzir consideravelmente a quantidade de outliers.

Tabela 4. Comparativo do MAPE no conjunto de treino após a transformação logarítmica.

Códigos	LSTM	MLP	Conv.	Ensemble
BPAN4	0,108	0,109	0,080	0,141
BBDC3	0,099	0,033	0,039	0,045
BBDC4	0,077	0,032	0,034	0,053
BBAS3	0,087	0,024	0,029	0,062
BPAC11	0,077	0,056	0,038	0,102
ITSA4	0,062	0,034	0,035	0,031
ITUB4	0,092	0,030	0,043	0,051
SANB11	0,076	0,038	0,041	0,014
Média	**0,085**	**0,045**	**0,042**	**0,062**

Tabela 5. Comparativo do MAPE no conjunto de teste após a transformação logarítmica.

Códigos	LSTM	MLP	Conv.	Ensemble
BPAN4	0,091	0,245	0,197	0,157
BBDC3	0,067	0,062	0,052	0,041
BBDC4	0,053	0,071	0,051	0,049
BBAS3	0,079	0,023	0,050	0,048
BPAC11	0,060	0,026	0,031	0,029
ITSA4	0,060	0,032	0,033	0,021
ITUB4	0,088	0,022	0,039	0,043
SANB11	0,054	0,060	0,041	0,017
Média	**0,069**	**0,068**	**0,062**	**0,051**

Por fim, ajustou-se novamente os modelos, agora para os dados que passaram pela transformação Yeo-Johnson, os erros podem ser encontrados nas Tabelas 6 e 7. Os resultados se aproximaram da transformação logarítmica, sendo que para algumas empresas ela foi melhor, como no caso de BPAN4, que também é a série que possui mais variação dos dados inicias. Com isso, foi possível perceber que algumas transformações se saem melhor em dadas situações, visto que no geral os modelos com log foram melhor, contudo não foi uma regra, dado que para séries com alto CV, Yeo-Johnson obteve menores erros.

Tabela 6. Comparativo do MAPE no conjunto de treino após a transformação de Yeo-Johnson.

Códigos	LSTM	MLP	Conv.	Ensemble
BPAN4	0,082	0,064	0,062	0,088
BBDC3	0,044	0,018	0,032	0,024
BBDC4	0,080	0,036	0,036	0,054
BBAS3	0,132	0,020	0,034	0,026
BPAC11	0,358	0,220	0,118	0,331
ITSA4	0,072	0,046	0,032	0,024
ITUB4	0,056	0,024	0,032	0,008
SANB11	0,102	0,100	0,068	0,070
Média	**0,116**	**0,066**	**0,050**	**0,078**

Tabela 7. Comparativo do MAPE no conjunto de teste após a transformação de Yeo-Johnson.

Códigos	LSTM	MLP	Conv.	Ensemble
BPAN4	0,188	0,054	0,128	0,115
BBDC3	0,046	0,020	0,030	0,024
BBDC4	0,104	0,062	0,050	0,050
BBAS3	0,132	0,020	0,034	0,026
BPAC11	0,142	0,223	0,160	0,137
ITSA4	0,072	0,046	0,028	0,029
ITUB4	0,048	0,022	0,028	0,008
SANB11	0,148	0,090	0,120	0,109
Média	**0,110**	**0,067**	**0,070**	**0,062**

Ao analisar os modelos gerados a partir das duas últimas transformações de dados, que demonstraram os melhores desempenhos, observou-se diferenças significativas no comportamento das séries temporais das empresas BPAN4 e ITSA4. Para BPAN4, a série apresentou o pior desempenho na maioria dos modelos. Este resultado pode ser atribuído à alta dispersão inicial dos dados, conforme indicado pelo coeficiente de variação. A Figura 2 ilustra que a transformação dos dados conseguiu minimizar essa condição, mas não o suficiente para tornar o desempenho de BPAN4 comparável ao das outras empresas.

Por outro lado, ITSA4, que inicialmente apresentava o menor CV, registrou o menor erro na maioria dos modelos. Isso ressalta a importância da qualidade dos dados iniciais e sugere que um CV menor pode facilitar o aprendizado da rede neural. Este fenômeno pode estar relacionado com a forma como as redes neurais capturam padrões nos dados, séries temporais com menor dispersão podem ser mais ”fáceis”de modelar.

A importância da transformação dos dados é ainda mais evidenciada quando considerado a minimização da dispersão. A transformação não apenas ajusta a escala dos dados, mas também corrigi anomalias como assimetria e outliers, que

são preservadas em métodos como a padronização. Isso é crucial para o aprendizado eficaz das redes neurais, pois melhora a qualidade dos dados de entrada, tornando-os mais representativos e menos sujeitos a ruído.

Além disso, nas tabelas apresentadas, foi possível observar que diferentes modelos tiveram desempenhos variados para diferentes empresas. Por exemplo, o modelo convolucional teve um dos melhores desempenhos para a ação SANB11, mas foi o pior para ITUB4. Isso indica que não há uma "configuração ideal" para os modelos e que a eficácia pode variar dependendo da série temporal em questão.

Na Tabela 8, é apresentado a média do Erro Percentual Absoluto Médio (MAPE) para cada arquitetura, considerando as diferentes transformações aplicadas no conjunto de teste. Notavelmente, as transformações logarítmica e Yeo-Johnson superaram a padronização. A transformação logarítmica, em particular, obteve uma média de MAPE de 0,051 no *Ensemble*, que foi 17,74% menor que a de Yeo-Johnson e 96,62% menor que os modelos padronizados. Este resultado sugere que a transformação logarítmica pode ser mais eficaz em lidar com as complexidades inerentes aos dados financeiros, devido à sua capacidade de minimizar o impacto de outliers e assimetrias.

Tabela 8. Comparativo do MAPE médio das arquiteturas baseada na transformação.

	Padronização	Log	Yeo-Johnson
LSTM	1,74	0,069	0,110
MLP	1,54	0,068	0,067
Conv.	2,19	0,062	0,070
Ensemble	1,50	0,051	0,062

Também é digno de nota que, embora alguns modelos tenham tido bom desempenho durante o treinamento, eles não se saíram tão bem quando expostos a dados inéditos. Por exemplo, os modelos convolucionais para as empresas ITUB4 e BBDC4 tiveram um aumento de erro de 527,3% e 301,8%, respectivamente, indicando um possível sobreajuste. Em contrapartida, modelos como o MLP mostraram melhorias significativas de desempenho nos testes, como nos modelos para as empresas BPAN4 e SANB11, onde as previsões melhoraram em 66,6% e 60,9%, respectivamente.

Essas observações são comprovadas pelos modelos individuais, que também mostraram um desempenho superior com a transformação logarítmica. A Figura 4 ilustra a proporção real de melhoria, medida pela redução do MAPE, após a aplicação das transformações.

Os resultados indicam que a escolha da transformação de dados e do modelo pode ter um impacto significativo no desempenho da previsão. O *Ensemble*, que combina diferentes modelos, apresentou o melhor desempenho médio geral, conforme mostrado na Tabela 8. Isso sugere que a combinação de diferentes abordagens pode ser uma estratégia eficaz para melhorar a precisão das previsões em ambientes incertos, como o mercado financeiro.

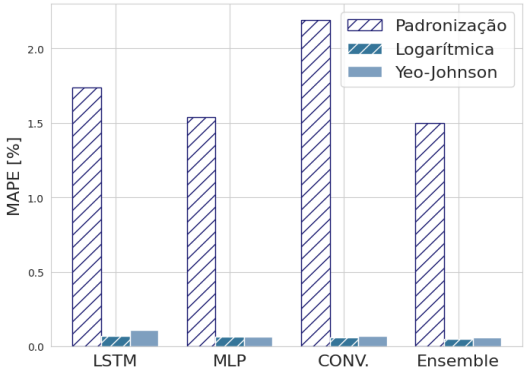

Figura 4. Comparativo da média do MAPE para cada transformação.

A análise dos resultados indica que a transformação logarítmica e a combinação de modelos (Ensemble) apresentaram o melhor desempenho na previsão dos preços das ações. A transformação logarítmica se destacou por ajudar a estabilizar a variância da série temporal, reduzindo o impacto de grandes variações e outliers. Isso facilita a modelagem e melhora a precisão das previsões. Além disso, essa transformação pode tornar as relações entre as variáveis mais lineares, o que é benéfico para muitos algoritmos de aprendizado de máquina que assumem linearidade nas relações subjacentes. Ao aplicar a transformação logarítmica, os valores extremos são suavizados, o que reduz a influência de outliers nos modelos.

A combinação de modelos também demonstrou um desempenho superior devido à sua capacidade de reduzir a variância do erro de previsão. A combinação de múltiplos modelos permite capturar diferentes aspectos dos dados, suavizando erros específicos de cada modelo individual e tornando as previsões mais robustas a variações e ruídos nos dados. Utilizar diferentes modelos de aprendizado de máquina, como LSTM, MLP e CNN, e combiná-los em um Ensemble permite explorar diferentes abordagens para capturar padrões temporais e estruturais nos dados.

Apesar dos resultados promissores, o estudo possui algumas limitações. Primeiramente, o estudo se concentra em apenas oito empresas do setor financeiro brasileiro, o que pode limitar a generalização dos resultados. Ampliar o conjunto de empresas analisadas pode fornecer uma visão mais abrangente e robusta. Além disso, a análise não considera variáveis externas, como notícias, eventos econômicos e políticos, que podem influenciar os preços das ações. Incorporar essas variáveis pode melhorar a precisão das previsões. Outro ponto a considerar é o período de tempo utilizado para validação dos modelos, que foi de três anos. Testar os modelos em períodos de tempo mais longos e com diferentes condições de mercado pode ajudar a avaliar melhor a robustez e a generalização das previsões.

Os resultados deste estudo têm várias implicações práticas. Para os investidores, a utilização de modelos de aprendizado de máquina e a combinação de modelos podem fornecer previsões mais precisas e robustas, auxiliando na tomada de de-

cisões informadas e minimizando riscos. Para gestores de fundos, os resultados sugerem que a combinação de múltiplos modelos pode ser uma estratégia eficaz para a gestão de portfólios, permitindo uma melhor diversificação e controle de risco. Finalmente, para os desenvolvedores de modelos de previsão de preços de ações, os resultados indicam que considerar a transformação logarítmica e o uso de Ensemble pode ser uma abordagem eficaz para melhorar a precisão das previsões.

5. Conclusão

A variação no preço de uma empresa é altamente volátil e imprevisível, principalmente se esta estiver sendo negociada na bolsa de valores, onde diversos fatores especulativos estão dentro das causas do porquê o preço que ela vale cai ou sobre em determinados períodos.

Esta pesquisa teve como objeto realizar previsões do comportamento dos preços das ações presentes no índice Bovespa pertencentes ao setor financeiro. Utilizando métodos de aprendizado de máquina com o objetivo de minimizar a diferença entre as predições e os valores reais.

A grande volatilidade e incerteza que envolve o preço tornam uma previsão exata impossível, contudo os modelos aqui apresentados fazem a série se tornar mais compreensível e podendo ser usado, juntamente com uma análise macro do mercado, como um aliado nas tomadas de decisões.

Os modelos atingiram um nível de erro satisfatório para essa análise, com ênfase para os ajustados após aplicação das transformações. Destacando-se a transformação logarítmica, que resultou em modelos mais ajustados que as demais, em segundo a Yeo-Johnson que teve um erro um pouco maior, porém desempenhou melhor em séries com alta variação. Assim como os modelos convolucionais e *Ensemble* que no geral foram os que atingiram os menores erros tanto nos treinos quanto nos testes.

Em trabalhos futuro existem duas frentes principais que podem ser exploradas, a primeira voltada para o aumento de modelos diversificados ao *Ensemble* criado, essa abordagem poderia trazer mais eficiência ao modelo agrupado uma vez que os novos métodos iriam trazer pontos de vista inéditos até então, o que poderia reduzir o erro.

A segunda possibilidade é o aumento de empresas consideradas na análise e o seu agrupamento em empresas semelhantes. Aqui, o foco seria construir diferentes modelos focados, não em prever uma única série temporal de uma empresa específica, mas sim apresentar ao modelo um grupo de empresas semelhantes isso tornaria o modelo mais robusto para inferir sobre o futuro daquele grupo de ações.

Agradecimentos

A FAPEMIG (Fundação de Amparo à Pesquisa do Estado Minas Gerais) pelo apoio financeiro.

Contribuições dos autores

Cláudio Estevam Leite da Silva: Concepção e elaboração da pesquisa, análise e interpretação dos resultados, redação do manuscrito. Ricardo Menezes Salgado: revisão-escrita e edição, supervisão.

Referências

[1] BRADSHAW, L. *Big Data and What it Means*. Disponível em: ⟨https://www.uschamberfoundation.org/sites/default/files/article/foundation/BigData.pdf⟩.

[2] CUKIER, K. *Big data: A revolution that will transform how we live, work, and think*. [S.l.]: Houghton Mifflin Harcourt, 2013.

[3] GALVãO, A. *Mercado financeiro: Uma Abordagem Prática dos Principais Produtos e Serviços*. [S.l.]: Elsevier, 2006.

[4] FERREIRA, F. G.; GANDOMI, A. H.; CARDOSO, R. T. Artificial intelligence applied to stock market trading: a review. *IEEE Access*, IEEE, v. 9, p. 30898–30917, 2021.

[5] VIJH, M. et al. Stock closing price prediction using machine learning techniques. *Procedia computer science*, Elsevier, v. 167, p. 599–606, 2020.

[6] YU, P.; YAN, X. Stock price prediction based on deep neural networks. *Neural Computing and Applications*, Springer, v. 32, p. 1609–1628, 2020.

[7] CHANDAR, S. K. Convolutional neural network for stock trading using technical indicators. *Automated Software Engineering*, Springer, v. 29, p. 1–14, 2022.

[8] B3. *Institucional*. 2023. https://www.b3.com.br/pt_br/b3/institucional/.

[9] MISHKIN, F. S.; EAKINS, S. G. *Financial markets*. [S.l.]: Pearson Italia, 2019.

[10] 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.

[11] FAMA, E. F. Efficient capital markets: Ii. *The journal of finance*, Wiley Online Library, v. 46, n. 5, p. 1575–1617, 1991.

[12] CHEN, Y.-S.; CHENG, C.-H.; TSAI, W.-L. Modeling fitting-function-based fuzzy time series patterns for evolving stock index forecasting. *Applied intelligence*, Springer, v. 41, n. 2, p. 327–347, 2014.

[13] CAVALCANTE, R. C. et al. Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, Elsevier, v. 55, p. 194–211, 2016.

[14] BASTOS, E.; BORTOLON, P.; MAIA, V. Fundamentalist signals in volatility scenarios: Evidence in the brazilian stock market. *BBR. Brazilian Business Review*, SciELO Brasil, v. 17, p. 621–639, 2021.

[15] JANIESCH, C.; ZSCHECH, P.; HEINRICH, K. Machine learning and deep learning. *Electronic Markets*, Springer, v. 31, n. 3, p. 685–695, 2021.

[16] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning Book, 2018*. 2020.

[17] SEZER, O. B.; GUDELEK, M. U.; OZBAYOGLU, A. M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, Elsevier, v. 90, p. 106181, 2020.

[18] TIWARI, S.; RAMAMPIARO, H.; LANGSETH, H. Machine learning in financial market surveillance: A survey. *IEEE Access*, IEEE, 2021.

[19] POONGODI, M. et al. Prediction of the price of ethereum blockchain cryptocurrency in an industrial finance system. *Computers & Electrical Engineering*, Elsevier, v. 81, p. 106527, 2020.

[20] RYLL, L.; SEIDENS, S. Evaluating the performance of machine learning algorithms in financial market forecasting: A comprehensive survey. *arXiv preprint arXiv:1906.07786*, 2019.

[21] ARRATIA, A. et al. Sentiment analysis of financial news: Mechanics and statistics. In: *Data Science for Economics and Finance*. [S.l.]: Springer, Cham, 2021. p. 195–216.

[22] LI, X.; WU, P.; WANG, W. Incorporating stock prices and news sentiments for stock market prediction: A case of hong kong. *Information Processing & Management*, Elsevier, v. 57, n. 5, p. 102212, 2020.

[23] GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. [S.l.: s.n.], 2021. v. 2ª Edição.

[24] HAYKIN, S. *Redes neurais: princípios e prática*. [S.l.]: Bookman Editora, 2001.

[25] RUDER, S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[26] SILVA, I. da; FLAUZINO, R.; SPATTI, D. Redes neurais artificiais para engenharia. *Artliber.–São Paulo. de*, 2011.

[27] HOUDT, G. V.; MOSQUERA, C.; NÁPOLES, G. A review on the long short-term memory model. *Artificial Intelligence Review*, Springer, v. 53, n. 8, p. 5929–5955, 2020.

[28] WEICHERT, D. et al. A review of machine learning for the optimization of production processes. *The International Journal of Advanced Manufacturing Technology*, Springer, v. 104, n. 5, p. 1889–1902, 2019.

[29] LILLICRAP, T. P.; SANTORO, A. Backpropagation through time and the brain. *Current opinion in neurobiology*, Elsevier, v. 55, p. 82–89, 2019.

[30] LI, Z. et al. Overfitting or underfitting? understand robustness drop in adversarial training. *arXiv preprint arXiv:2010.08034*, 2020.

[31] MITCHELL, T. M. *Machine learning*. [S.l.]: McGraw-hill New York, 1997. v. 1.

[32] MURPHY, K. P. *Machine learning: a probabilistic perspective*. [S.l.]: MIT press, 2012.

[33] CHEN, Z.; LI, C.; SUN, W. Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*, Elsevier, v. 365, p. 112395, 2020.

[34] RAMCHOUN, H. et al. Multilayer perceptron: Architecture optimization and training. International Journal of Interactive Multimedia and Artificial Intelligence ..., 2016.

[35] 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.

[36] KIRANYAZ, S. et al. 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, Elsevier, v. 151, p. 107398, 2021.

[37] RUDER, S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[38] KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[39] TIELEMAN, T.; HINTON, G. Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *Technical report*, 2017.

[40] DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, v. 12, n. 7, 2011.

[41] B3. *Índice Bovespa (Ibovespa B3)*. 2023. https://www.b3.com.br/pt_br/market-data-e-indices/indices/indices-amplos/indice-ibovespa-ibovespa-composicao-da-carteira.htm.

[42] MORETTIN, P. A.; BUSSAB, W. O. *Estatística básica*. [S.l.]: Saraiva Educação SA, 2017.

[43] EVERITT, B. S.; HOWELL, D. C. *Encyclopedia of Statistics in Behavioral Science*. [S.l.]: John Wiley & Sons, Ltd, 2021. v. 1.

[44] CORNELL, B. Esg preferences, risk and return. *European Financial Management*, Wiley Online Library, v. 27, n. 1, p. 12–19, 2021.

[45] BOYD, J. H.; GRAHAM, S. L. et al. Risk, regulation, and bank holding company expansion into nonbanking. *Quarterly Review*, Federal Reserve Bank of Minneapolis, v. 10, n. Spr, p. 2–17, 1986.

[46] HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: principles and practice*. [S.l.]: OTexts, 2018.