



UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

**Avaliando a influência de um fator de correção de preços
de ações na capacidade preditiva de modelos baseados em
redes neurais recorrentes**

Jair Orlando González Carmona

Dissertação de Mestrado apresentada ao Programa
de Pós-Graduação em Computação da Universi-
dade Federal do Rio Grande - FURG, como requi-
sito parcial para a obtenção do grau de Mestre em
Engenharia de Computação

Orientador: Prof. Dr. Eduardo Nunes Borges
Co-orientador: Prof. Dr. Rafael Alceste Berri

Rio Grande, 2024



Dissertação de Mestrado

Avaliando a influência de um fator de correção de preços de ações na capacidade preditiva de modelos baseados em redes neurais recorrentes

Jair Orlando González Carmona

Banca examinadora:

Prof. Dr. Bruno Lopes Dalmazo – FURG

Prof. Dr. Samuel da Silva Gomes – IFRS

Ficha Catalográfica

C287a Carmona, Jair Orlando Gonzalez.

Avaliando a influência de um fator de correção de preços de ações na capacidade preditiva de modelos baseados em redes neurais recorrentes / Jair Orlando Gonzalez Carmona. – 2024.
52 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Computação, Rio Grande/RS, 2024.

Orientador: Dr. Eduardo Nunes Borges.

Coorientador: Dr. Rafael Alceste Berri.

1. *Deep Learning* 2. RNN 3. BovDB 4. Bolsa de Valores Brasileira (B3) 5. MSE I. Borges, Eduardo Nunes II. Berri, Rafael Alceste III. Título.

CDU 004:336.76

Catálogo na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344

RESUMO

CARMONA, Jair Orlando González. **Avaliando a influência de um fator de correção de preços de ações na capacidade preditiva de modelos baseados em redes neurais recorrentes**. 2024. 52 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

Compreender o comportamento financeiro, particularmente no mercado de ações, tem atraído significativo interesse nos últimos anos devido aos avanços na inteligência artificial e seu impacto na economia global. O campo da previsão do mercado de ações, que explora a interação entre finanças e ciência da computação para criar modelos preditivos, visa prever o comportamento de diversos títulos no mercado financeiro. Uma das técnicas mais conhecidas e amplamente utilizadas é o *Deep Learning*, que emprega diferentes estruturas de redes neurais profundas para aprender modelos não lineares. Neste estudo, utilizamos dados abertos de algumas das maiores empresas do Brasil – Petrobras (PETR4), Itaúsa (ITSA4) e Vale (VALE3) – provenientes do BovDB, um conjunto de dados históricos contendo os preços das ações de todas as empresas listadas na bolsa de valores brasileira (B3) de 2000 a 2020. Como parte complementar do pré-processamento, foi aplicado um *fator de correção de preços* para neutralizar os efeitos de eventos de mercado no comportamento das ações, permitindo que o modelo de Rede Neural Recorrente (RNN) processasse essas informações de maneira mais eficaz. Os resultados mostraram que o uso deste fator de correção melhora significativamente as previsões, reduzindo comportamentos abruptos nos preços das ações e diminuindo o erro de previsão do modelo. Os resultados destacam o potencial do uso de um fator de correção de eventos em dados de ações processados por um modelo RNN, facilitando seu treinamento e fornecendo previsões mais confiáveis.

Palavras-chave: Deep Learning, RNN, BovDB, Bolsa de Valores Brasileira (B3), MSE.

ABSTRACT

CARMONA, Jair Orlando González. **Desempenho de um fator de correção na previsão de preços das ações PETR4, ITSA e VALE3 usando redes neurais recorrentes.** 2024. 52 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

Understanding financial behavior, particularly in the stock market, has garnered significant interest in recent years due to advancements in artificial intelligence and its impact on the global economy. The field of stock market prediction, which explores the interaction between finance and computer science to create predictive models, aims to forecast the behavior of various securities in the financial market. One of the most well-known and widely used techniques is Deep Learning, which employs different structures of deep neural networks to learn non-linear models. In this study, we used open data from some of the largest companies in Brazil – Petrobras (PETR4), Itaúsa (ITSA4), and Vale (VALE3) – sourced from BovDB, a historical dataset containing the stock prices of all companies listed on the Brazilian stock exchange (B3) from 2000 to 2020. As a complementary part of the preprocessing, a price correction factor was applied to neutralize the effects of market events on stock behavior, allowing the Recurrent Neural Network (RNN) model to process this information more effectively. The results showed that using this correction factor significantly improves predictions, reducing abrupt price behaviors and decreasing the model's prediction error. The results highlight the potential of using an event correction factor in stock data processed by an RNN model, facilitating its training and providing more reliable forecasts.

Keywords: Stock Market Prediction, factor, Deep Learning, RNN, BovDB, MSE, Brazilian Stock Exchange (B3).

LISTA DE FIGURAS

1	Taxonomia de técnicas de previsão de ações. Adaptado de [35]. . . .	10
2	Utilização de diferentes tipos de dados brutos (Adaptado de [21]). . .	14
3	Variáveis na negociação de ações	14
4	Exemplos de eventos em preço de fechamento de PETR4	15
5	Média e Desvio Padrão da VALE3 entre 1995 e 2020, com e sem fator [8].	17
6	Utilização de diferentes tipos de dados brutos (Adaptado de [21]). . .	18
7	Esquema geral de uma rede Neural recorrente (RNN)	18
8	<i>Forward Propagation</i> em RNN	19
9	Unidade RNN	19
10	Exemplo dos efeitos de learning rate sobre o aprendizado.	20
11	Influência do valor do predecessor na previsão em um <i>windowed da-</i> <i>taset</i> [25].	21
12	Exemplo de agrupamento dos dados utilizando a técnica <i>windowed</i> <i>dataset</i>	21
13	Exemplo de particionamento de um conjunto de dados na sessão de treinamento e validação. [25]	22
14	Volatilidade anual para cada ação.	29
15	Preço de fechamento de PETR4 sem (esq.) e com fator de correção (dir.).	31
16	Preço de fechamento de ITSA4 sem (esq.) e com fator de correção (dir.).	31
17	Preço de fechamento de VALE3 sem (esq.) e com fator de correção (dir.).	31
18	Loss versus Learning rate PETR4.	32
19	Loss versus Learning rate ITSA4.	32
20	Loss versus Learning rate VALE3.	32
21	Função de perda vs época para dados de treinamento PETR4 (azul) e validação (vermelho).	34
22	Dados de validação (linha azul) e previsão do modelo (linha verme- lha) para PETR4.	34
23	Função de perda dos dados de treinamento (azul) e validação (verme- lho) da ITSA4.	35
24	Dados de validação (linha azul) e previsão do modelo (linha verme- lha) para ITSA4.	35

25	Função de perda dos dados de treinamento (azul) e validação (vermelho) da VALE3.	36
26	Dados de validação (linha azul) e previsão do modelo (linha vermelha) para VALE3.	37
27	Preço de fechamento de ITSA4 sem (esq.) e com fator de correção (dir.) usando cinco folds para treinamento e cinco para validação. . .	37
28	Preço de fechamento de VALE3 sem (esq.) e com fator de correção (dir.) usando cinco folds para treinamento e cinco para validação. . .	37
29	Função de perda dos folds de treinamento (azul) e validação (vermelho) da ITSA4	38
30	Função de perda dos folds de treinamento (azul) e validação (vermelho) da VALE3.	38
31	Dados de validação (azul) e previsão do modelo (vermelho) para ITSA4, com folds.	38
32	Dados de validação (azul) e previsão do modelo (vermelho) para VALE3, com folds.	38
33	Função de perda para folds de treinamento (azul) e validação (vermelho) de ITSA4.	39
34	Previsão (vermelho) do modelo com os dados de validação (azul) de ITSA4, sem (esquerda) e com fator de correção (direita), com folds invertidos.	40
35	Função de perda para folds de treinamento (azul) e validação (vermelho) de VALE3.	41
36	Previsão (linha vermelha) do modelo com os dados de validação (linha azul) do VALE3. Sem (Esq) e com (Dir) factor, com folds invertidos.	41
37	Função de perda para dados de treinamento (azul) e validação (vermelho) de ITSA4, com normalização.	42
38	Previsão (vermelho) do modelo com os dados de validação (azul) de ITSA4, com normalização.	42
39	Função de perda para dados de treinamento (azul) e validação (vermelho) de VALE3, com normalização.	43
40	Previsão (vermelho) do modelo com os dados de validação (azul) de VALE3, com normalização.	43
41	Função de perda para dados de treinamento (azul) e validação (vermelho) de ITSA4, com normalização e dropout.	45
42	Previsão (vermelho) do modelo com os dados de validação (azul) de ITSA4, com normalização e dropout.	45
43	Função de perda para dados de treinamento (azul) e validação (vermelho) de VALE3, com normalização e dropout.	46
44	Previsão (vermelho) do modelo com os dados de validação (azul) de VALE3, com normalização e dropout.	46

LISTA DE TABELAS

1	Informação estatística para cada ação.	30
2	Eventos em cada ação analisada	30
3	Melhores hiperparâmetros para cada ação	33
4	melhores hiperparâmetros para ações normalizadas	42
5	Melhores hiperparâmetros para ações com normalização e reguladores	44
6	análise comparativa dos resultados obtidos	47

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
ARIMA	Modelo de Média Móvel Integrada Autoregressiva
B3	Bolsa de valores brasileira
DL	Deep Learning
ESN	Echo State Networks
GPU	Unidade de processamento gráfico
RNN	Reccurent Neural Network
Regul	Valor de Regularização
SVM	Support Vector Machines
SMP	Stock Market Prediction
SemF	Sem factor
ML	Machine Learning
LSTM	Long Short-Term Memory
PIP	Perceptually Important Points
PETR4	Ação de Petrobras
ITSA4	Ação de Itaúsa
VALE3	Ação de Vale

SUMÁRIO

1	Introdução	10
1.1	Objetivo	11
1.2	Objetivos Específicos	12
1.3	Organização do trabalho	12
2	Fundamentação teórica	13
2.1	Previsão de ações no mercado (Stock market prediction)	13
2.2	Eventos nas séries financeiras	15
2.3	Fator de correção de preços	16
2.4	Volatilidade	17
2.5	Rede Neural Recorrente (RNN)	17
2.6	Taxa de aprendizagem (Learning rate)	19
2.7	Função de perda (Loss Function)	20
2.8	Adaptabilidade dos dados	20
2.9	Particionamento	21
3	Trabalhos relacionados	23
4	Metodologia	25
4.1	Dataset	25
4.2	Pré-Processamento	26
4.3	Seleção dos Hiperparâmetros	26
4.3.1	Taxa de Aprendizado	27
4.3.2	Validação do modelo	27
5	Resultados obtidos	29
5.1	Análise comportamental das ações	29
5.1.1	Hiperparâmetros	31
5.1.2	Treinamento e predição	33
5.1.3	Normalização de uma série temporal financeira	41
6	Conclusão	48
	Referências	49

1 INTRODUÇÃO

As ações no mercado de investimento são um instrumento de capitalização onde sua rentabilidade é variável já que dependem do resultado das negociações da empresa emite, as quais se refletem no preço a que a ação pode ser comprada ou vendida [25].

Os dados das ações são analisados principalmente em dois objetivos de negociação, de curto e de longo prazo, apresentando diferentes padrões de lucro. Todos os padrões tendem a variar constantemente, pois são afetados por fatores externos como política, economia e notícias de última hora, gerando alta incerteza no mundo das finanças [41].

Um ponto crucial que contribui para a imprevisibilidade do mercado de ações é sua natureza em constante mudança. No entanto, apesar de sua complexidade, vários algoritmos têm sido explorados, demonstrando sua eficácia na previsão dos preços do mercado [19] [30]. Um dos campos que reúne as áreas de finanças e computação é conhecido como *Stock Market Prediction* (SMP) [21]. Na Figura 1 é mostrado uma hierarquia das técnicas mais utilizadas em SMP.

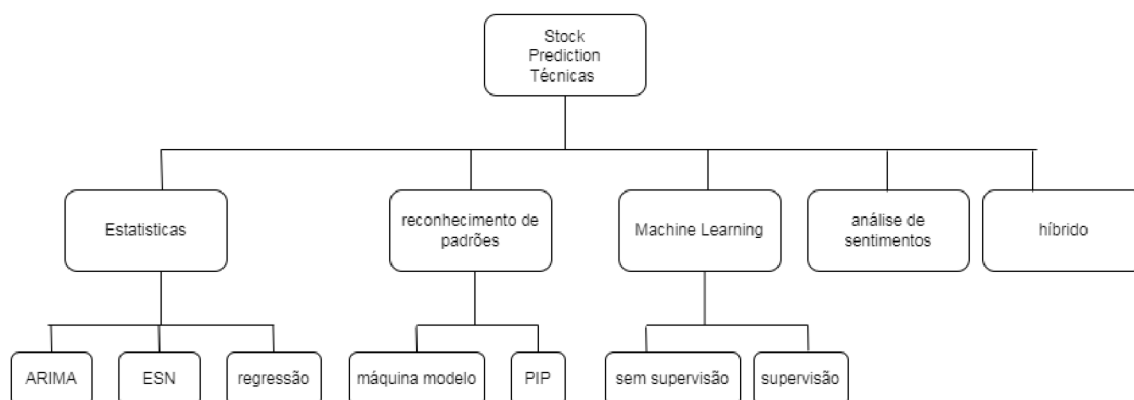


Figura 1: Taxonomia de técnicas de previsão de ações. Adaptado de [35].

Antes do surgimento do aprendizado de máquina (*machine learning*) e o aprendizado profundo (*deep learning*), os modelos estatísticos foram os primeiros a serem desenvolvidos para prever valores no mercado, ao qual assumiam linearidade, sazonalidade e normalidade em seu comportamento, tornando-se ineficazes com dados de tipo temporários [16]. Os mais conhecidos são: o Modelo de Média Móvel Integrada Autore-

gressiva (ARIMA) [20] e o Modelo de Heteroscedasticidade Condicional Autoregressiva Generalizada (GARCH) [6].

Posteriormente, surgiram modelos mais desenvolvidos, como o *machine learning*, que vem sendo amplamente estudado por seu potencial na previsão de preços no mercado [39]. Alguns exemplos são a regressão logística e Support Vector Machines (SVM), onde o SVM foi um dos melhores algoritmos devido à sua alta *accuracy* no mercado [1]. Com o surgimento do Big Data na web [13], com conjuntos de dados muito grandes ou complexos para serem tratados pelo software tradicional de processamento de dados (Notícias, Tweets, áudios, vídeos) [11], novos métodos foram evoluindo para poder processar grandes quantidades dessas informações, sendo um deles e o mais conhecido o *Deep Learning* (DL) [21]. DL é composto por diferentes estruturas de redes neurais profundas que permitem o aprendizado não linear dos dados [17], além de utilizar a capacidade de processamento paralelo das unidades de processamento gráfico (GPU).

Dentre as diferentes aplicações do Deep Learning, uma das mais importantes é a previsão de séries temporais [7], pois tem a capacidade de reconhecer padrões e identificar irregularidades nos dados, despertando considerável interesse no mercado financeiro. Diferentes técnicas e métodos têm sido propostos na literatura na área de previsão de mercado financeiro [8], desenvolvendo uma variedade de pesquisas em previsão de ações [9], sendo o modelo de Rede Neural Recorrente (RNN) [41] um dos mais relevantes na área. Nesses trabalhos, a normalização dos dados varia de acordo com seu propósito, onde Nayak [27] demonstra como a normalização dos dados financeiros afeta significativamente a precisão das previsões. No entanto, nenhum desses estudos aborda especificamente o efeito dos eventos de mercado na previsão do modelo, já que esses comportamentos ainda prevalecem após a normalização dos dados. Trabalhos como [18] [27] [33], que utilizam redes recorrentes (RNN, LSTM), embora normalizem seus dados, ainda apresentam seções com comportamentos complexos devido aos eventos de mercado, como aumentos ou quedas acentuadas nos preços, fazendo com que o modelo falhe em prever essas zonas.

1.1 Objetivo

Levando em consideração o exposto até esse ponto, o objetivo deste trabalho é analisar e quantificar o impacto de um fator de correção aplicado aos preços de ações na performance preditiva do modelo de redes neurais recorrentes (RNN), visando determinar se a inclusão desse fator melhora a precisão das previsões de preços das ações. O estudo enfatiza a avaliação do desempenho do modelo sob diferentes condições, especialmente em termos de sua sensibilidade à inclusão do fator de correção de eventos, projetado para considerar as flutuações e anomalias induzidas pelo mercado. São utilizados para esta análise informações extraídas do BovDB [8], um conjunto de dados construído pelo Grupo de Pesquisa em Gerenciamento de Informações da FURG, a partir da Bolsa de Valores Bra-

sileira (B3), desde o ano 2000 até 2020. Além disso, o estudo visa aprofundar nossa compreensão das técnicas de modelagem preditiva ao explorar o uso de regularizadores e estratégias de normalização no contexto das RNN. Ao fazer isso, pretende esclarecer os potenciais benefícios e limitações dessas técnicas na melhoria da precisão e robustez dos modelos preditivos, particularmente no contexto de mercados financeiros voláteis e dinâmicos.

1.2 Objetivos Específicos

Para que o objetivo geral deste trabalho seja alcançado, alguns objetivos específicos são planteados:

1. Analisar e validar a aplicação do fator de correção de preço (*factor*) nas ações PETR4, ITSA4 e VALE3;
2. Ajustar hiperparâmetros do modelo RNN para melhor adequar aos casos de estudo;
3. Analisar os resultados obtidos em termos de métricas de desempenho adequadas.

1.3 Organização do trabalho

Este documento está organizado da seguinte forma. Inicialmente no Capítulo 2 são apresentados alguns fundamentos teóricos. Posteriormente, são elencadas e discutidas abordagens de trabalhos relacionados no Capítulo 3. A metodologia do trabalho é apresentada no Capítulo 4. No Capítulo 5 são detalhados os resultados obtidos e, por fim, no Capítulo 6, são apresentados a conclusão e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos necessários para o completo entendimento deste trabalho. Para isso, começamos com definições relacionadas com finanças e deep learning, seguido da explicação e definição da variável *factor* que implementa o fator de correção de preços. A seguir o agrupamento dos dados em suas respectivas *features* e *labels* necessários para serem inseridos e processados no modelo são apresentados. Por fim, explicamos o modelo RNN e a relevância de seus hiperparâmetros.

2.1 Previsão de ações no mercado (Stock market prediction)

A previsão do comportamento das ações, também chamada de *Stock Market Prediction*, tem como objetivo determinar o valor futuro das ações de uma empresa ou outro instrumento financeiros negociado em uma bolsa [43]. Portanto tem sido um tópico de pesquisa altamente relevante nos últimos anos, principalmente aquelas que são desenvolvidas através da aplicação de modelos de Deep learning [21], sendo as investigações nas áreas de taxas de câmbio em bancos e crises financeiras de bastante interesse também [28].

A seguir, na Figura 2, apresentamos em um gráfico a relação do número de artigos publicados nos últimos anos com os tipos de dados utilizados na área. É possível observar que um dos tipos de dados mais utilizados na área de stock market prediction tem sido os numéricos focados no setor de mercado mantendo-se constante ao longo dos anos devido o fato de estarem disponíveis em grandes quantidades na rede, algo de grande importância quando se trabalha com modelos de deep learning, pois eles exigem muitas informações de entrada para funcionar corretamente.

Os dados de mercado geralmente contêm todas as atividades de negociação que ocorrem no mercado de ações, que são:

High price e Low price são as variáveis que especificam o valor máximo e mínimo da ação em um determinado período de tempo;

Opening price e Closing price são os preços para os quais a ação começou e terminou sendo negociado no mesmo período;

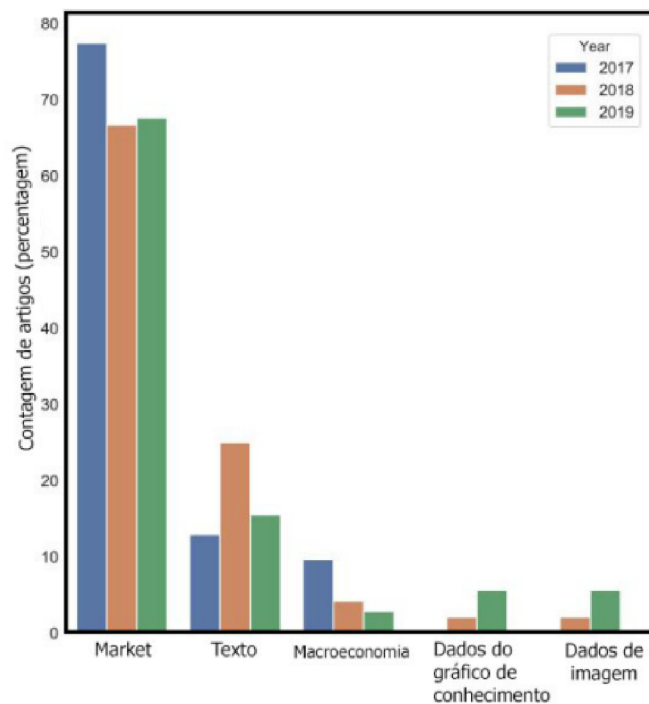


Figura 2: Utilização de diferentes tipos de dados brutos (Adaptado de [21]).

Trading volume é o valor total da atividade de negociação.

Para melhor entendimento, considere a Figura 3, ao qual representa um *candle* contendo as principais variáveis relacionadas com a ação.

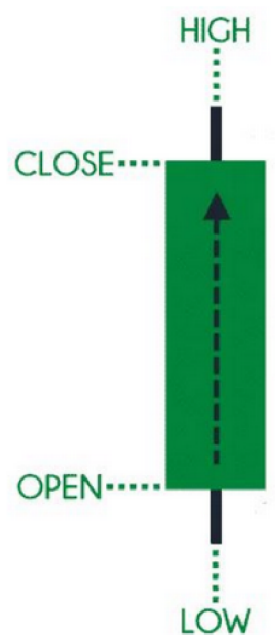


Figura 3: Variáveis na negociação de ações

Onde Open é o preço de abertura, close o preço de fechamento, low e high o menor e maior valor entre a abertura e o fechamento. Por ultimo, a seção verde e a seta para cima

indicam que o valor de fechamento é maior que o de abertura, ou seja, o ativo valorizou.

2.2 Eventos nas séries financeiras

Os acontecimentos de mercado no contexto financeiro referem-se a momentos e situações em que existe um impacto relevante nos preços, na volatilidade ou nas condições gerais do mercado. Estes eventos podem ser de natureza política, social, econômica ou empresarial, com base em decisões dos participantes no mercado, tais como investidores ou comerciantes [32]. Exemplos comuns de eventos de mercado podem ser observados como alterações rápidas e significativas no valor dos ativos financeiros, criando oportunidades ou desafios para os participantes no mercado. A Figura 4 apresenta um exemplo em que o valor de fechamento de PETR4 é alterado por dois eventos conhecidos como *ex-bônus*. Esses eventos mostram uma queda abrupta no preço, que na verdade está relacionada à distribuição de ações bonificadas pela empresa aos seus acionistas atuais.

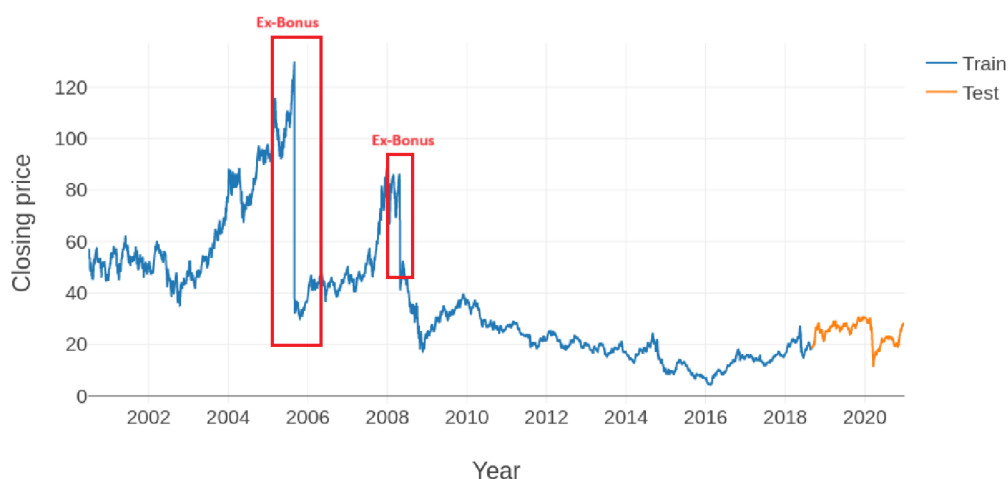


Figura 4: Exemplos de eventos em preço de fechamento de PETR4

Alguns eventos de mercado são explicados abaixo.

- **Ex-bônus:** Um bônus em ações é uma emissão gratuita de ações adicionais aos acionistas existentes. O preço das ações geralmente ajusta-se para refletir a emissão adicional de ações.
- **Ex-rights:** Em uma emissão de direitos, os acionistas têm a oportunidade de comprar ações adicionais, geralmente a um preço com desconto. O preço das ações geralmente cai para refletir o valor dos direitos que os novos compradores não receberão.
- **Ex-split:** Um desdobramento de ações aumenta o número de ações em circulação, dividindo cada ação em várias partes. O preço das ações ajusta-se para baixo pro-

porcionalmente ao aumento do número de ações. O valor total do investimento dos acionistas não muda, mas o número de ações que possuem aumenta.

2.3 Fator de correção de preços

Uma inovação proposta por Cardoso [8] consiste na introdução de uma variável cumulativa denominada *factor*. Essa abordagem destaca-se por incorporar eventos passados para realizar ajustes nos preços das ações atuais. Para uma compreensão mais aprofundada, as equações 1 e 2 apresentam os cálculos específicos associados a essa variável.

$$factor\ isolated(d) = \frac{close(d)}{open(d+1)} \quad (1)$$

$$factor\ cumulative(d+1) = \frac{factor\ cumulative(d)}{factor\ isolated(d)} \quad (2)$$

A variável *factor* desempenha um papel crucial ao considerar o histórico de eventos, fornecendo uma perspectiva temporal na modelagem dos preços das ações, onde d é definido como o dia anterior ao evento e o impacto dos eventos nas ações é calculado dividindo o preço de fechamento do dia anterior, $close(d)$, pelo preço de abertura do dia atual, $open(d+1)$, no qual o evento ocorreu.

Então, o impacto cumulativo e regressivo (de 2020 a 2000, ação por ação) de todos os eventos é calculado multiplicando todos os fatores de uma mesma ação até chegar a uma determinada data, alterando assim o fator com base em todos os fatores anteriores.

Para aplicar o fator de correção de preços ao conjunto de dados, deve-se dividir o atributo a ser utilizado (*closingprice*) pelo fator daquele dia (*factor isolated*), e multiplicar a quantidade do *stock* por *factor cumulative*. Para mostrar a influência dos dados com o fator de correção é apresentada a Figura 5, onde a média e o desvio padrão (linha preta no topo de cada média anual) são comparados entre a ação VALE3 com (barras laranja) e sem (barras azuis) *factor*. Pode-se ver como os dados sem o fator de correção (preços sem qualquer correlação) têm um grande desvio devido a eventos que naturalmente ajustarão os preços das ações em seu respectivo ano. Esse ajuste significa que os preços antigos não são comparados diretamente ao longo do tempo com os atuais. Por outro lado, os dados ajustados com *factor* apresentam um desvio menor e uma média comparável (em quantidade) ao longo de toda a série temporal.

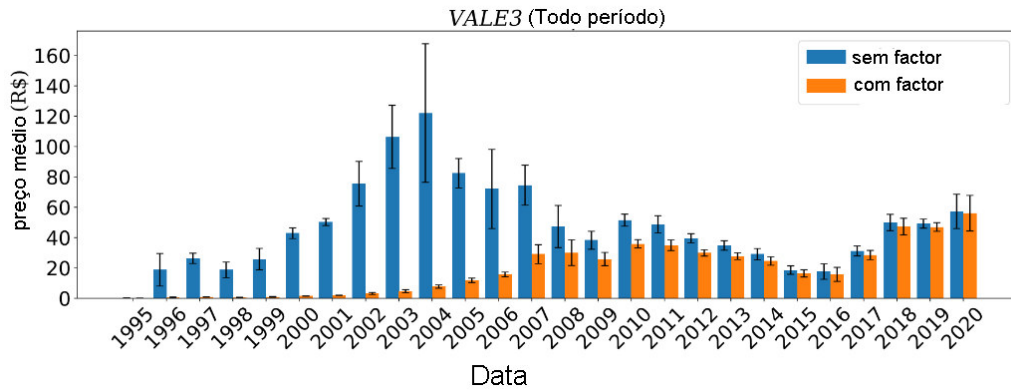


Figura 5: Média e Desvio Padrão da VALE3 entre 1995 e 2020, com e sem fator [8].

2.4 Volatilidade

Uma ação é considerada volátil se seu preço apresenta grandes variações percentuais de um dia para o outro [5]. Embora a alta volatilidade aumente a chance de grandes ganhos, também aumenta a chance de incorrer em grandes perdas, portanto, o risco deve ser cuidadosamente gerenciado ao investir em ações de alta volatilidade.

Uma forma de medir a volatilidade é tomando a faixa do preço de cada dia, calculando a diferença entre o valor mais alto e o mais baixo, para capturar os movimentos intradiários. Para normalizar o resultado, ele é dividido pelo preço de abertura, com o objetivo de ter uma medida comparável entre todos os dias.

$$volatilidade = \frac{High_t - Low_t}{Open_t} \quad (3)$$

2.5 Rede Neural Recorrente (RNN)

Um modelo RNN é uma rede neural artificial [21] que vem sendo amplamente utilizada nos últimos anos na área de pesquisa de stock market prediction, como pode ser visto na Figura 6, devido à sua afinidade em prever padrões temporais como são as flutuações dos preços no mercado.

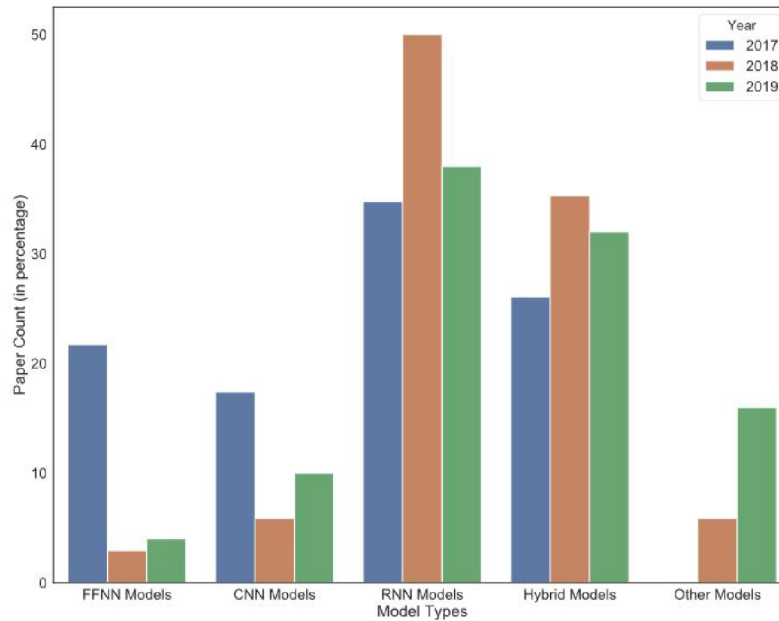


Figura 6: Utilização de diferentes tipos de dados brutos (Adaptado de [21]).

Ao contrário dos modelos anteriores, como a rede neural Feedforward (FNN) [7], os modelos RNN usam sua memória interna para processar dados de entrada e assim prever dados futuros. A arquitetura do modelo RNN geralmente consiste estruturalmente em diferentes números de camadas e unidades em cada uma delas, onde uma das principais características está em seus nós, que são conectados de forma cíclica ao longo de uma sequência, o que o ajuda a apresentar um comportamento dinâmico temporário, como é mostrado nas Figuras 7 e 8.

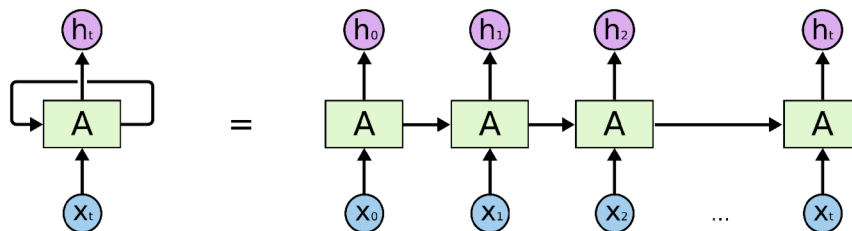


Figura 7: Esquema geral de uma rede Neural recorrente (RNN)

Em geral, as RNNs processam uma sequência de entrada de um elemento por vez em um tempo “t”, mantendo em suas unidades ocultas um *state vector* que contém implicitamente informações sobre o histórico de todos os elementos passados da sequência [22]. A Figura 9 mostra uma unidade de célula de um modelo RNN, que é composta por uma função de ativação, neste caso a função Tanh (Equação 4), devido ao seu bom comportamento com dados temporais, x_t como vetor de entrada (e.g., stock prices), o state vector $a(t-1)$ gerado na célula anterior, e suas saídas a_t para a próxima célula e a resposta Y_t que previamente passa por uma função de ativação para ser classificada de acordo com sua classe.

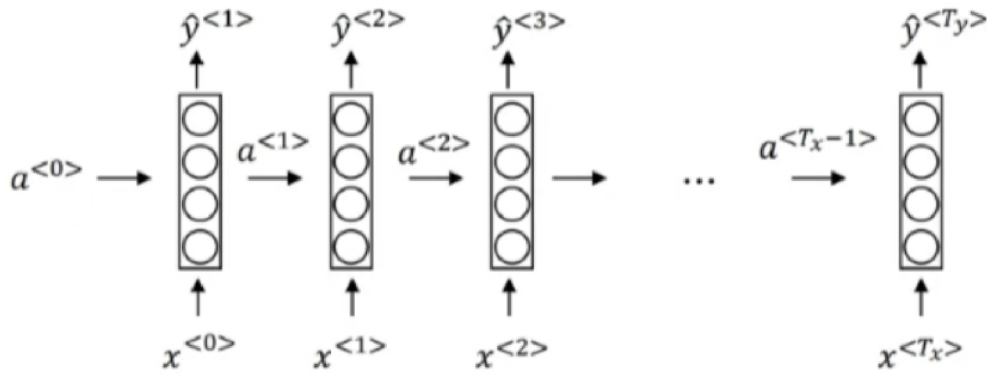


Figura 8: *Forward Propagation* em RNN

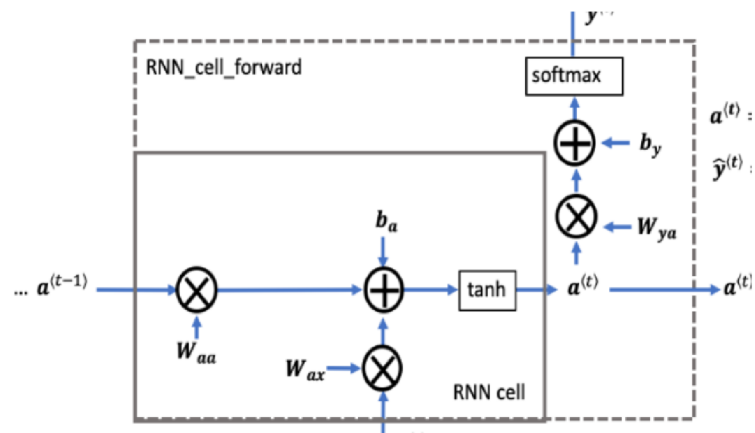


Figura 9: Unidade RNN

Formalmente, um modelo RNN pode ser apresentado da seguinte forma, onde W e B são os pesos e bias que serão treinados em cada iteração junto com a função de ativação $\tanh$ ou softmax .

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

$$a^{<t>} = \text{Tanh}(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad (5)$$

$$\hat{y}^{<t>} = \text{Softmax}(W_{ya}a^{<t>} + b_y) \quad (6)$$

2.6 Taxa de aprendizagem (Learning rate)

O learning rate é um parâmetro de ajuste em um algoritmo de otimização que determina o tamanho do passo em cada iteração enquanto se move para um mínimo de uma função de perda [26]. Ou seja, pode ser entendido como a velocidade com que um modelo “aprende” e atualiza seus parâmetros. Geralmente a direção é determinada pelo gradiente

da função de perda e o Lr especifica quão grande será o passo nessa direção. Como pode ser visto na Figura 10¹, um passo muito grande pode causar *overshoot* em seu valor ótimo e nunca alcançá-lo, da mesma forma, se o passo for muito pequeno, precisará de muitas iterações para convergir para o melhor valor, fazendo com que leve muito tempo.

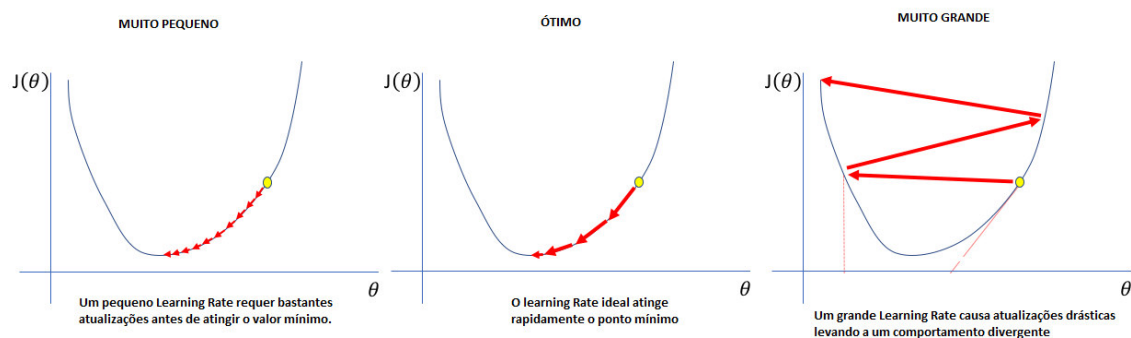


Figura 10: Exemplo dos efeitos de learning rate sobre o aprendizado.

2.7 Função de perda (Loss Function)

Uma função de perda ou custo em geral pode ser vista como um método que permite avaliar o quão bem um algoritmo está modelando o conjunto de dados, sendo esta uma função matemática. Na regressão, um dos métodos mais conhecidos para avaliar essa função é o erro quadrático médio (MSE), equação 7, tomando a diferença entre o valor atual e a previsão do modelo, para depois elevar ao quadrado e tirar a média de todo o conjunto de dados.

$$MSE = \frac{1}{D} \sum_{i=1}^D (x_i - y_i)^2 \quad (7)$$

2.8 Adaptabilidade dos dados

Ao trabalhar com redes neurais e dados de séries temporais, estes devem ser estruturados para serem implementados em um modelo RNN. *Windowed dataset* [25] uma técnica para melhor representar esses casos. A seguir, é apresentado um exemplo desta técnica.

Imagine que se deseja prever o valor de uma ação em um momento “t”, para isso deve-se partir dos dados predecessores naquele momento. Por exemplo, tendo como base a Figura 10, onde é mostrado o comportamento do valor de uma ação genérica no decorrer do tempo, caso seja preciso prever o valor da série temporal no tempo 1200, é preciso considerar os 30 valores predecessores como uma função para prever o último. Portanto, os valores do tempo 1170 a 1199 determinarão o valor no tempo 1200.

¹ Adaptado de: <https://syr.us/79e>

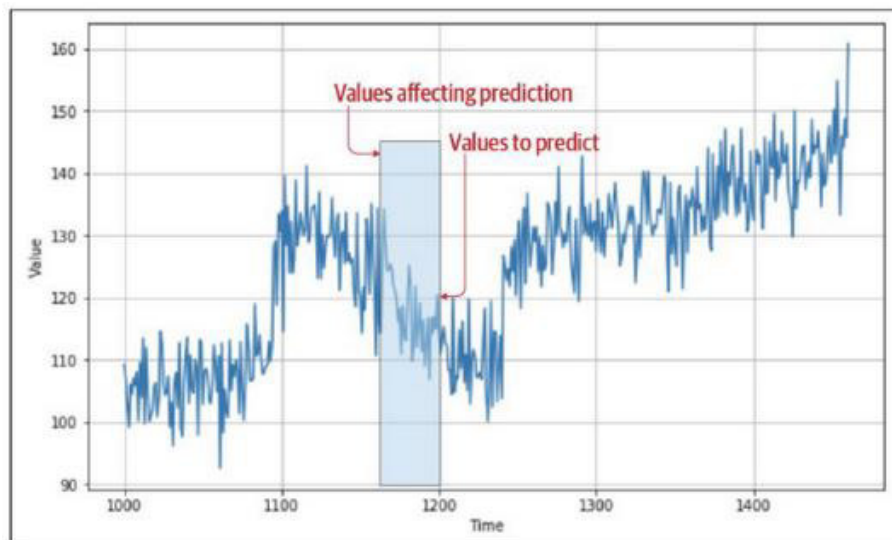


Figura 11: Influência do valor do predecessor na previsão em um *windowed dataset* [25].

Essa metodologia é realizada com cada um dos dados, chegando ao final de um conjunto de dados agrupados em suas respectivas *features* e *labels* (como é mostrado no exemplo da Figura 13), prontos para serem treinados e avaliados pelo modelo.

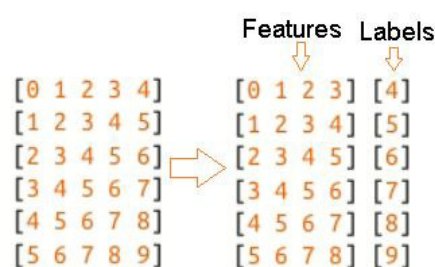


Figura 12: Exemplo de agrupamento dos dados utilizando a técnica *windowed dataset*.

A Figura 12 representa um exemplo de agrupamento de dados de um conjunto formado por um intervalo de 0 a 10 unidades e agrupado em 5 unidades, cada uma com salto de 1 unidade. Portanto, pegando o primeiro agrupamento, sua característica seriam os valores 0,1,2,3 e o rótulo seria o número 4. O mesmo raciocínio é aplicado para todos os casos.

2.9 Particionamento

Existem duas partições comumente aplicadas a um conjunto de dados a ser usado em um modelo de previsão. Estes são conhecidos como períodos de treinamento e validação. Na primeira, os hiperparâmetros são ajustados para ter uma boa previsão e na segunda, é validado se o modelo foi treinado corretamente, simulando como o modelo se comportaria com novos dados.

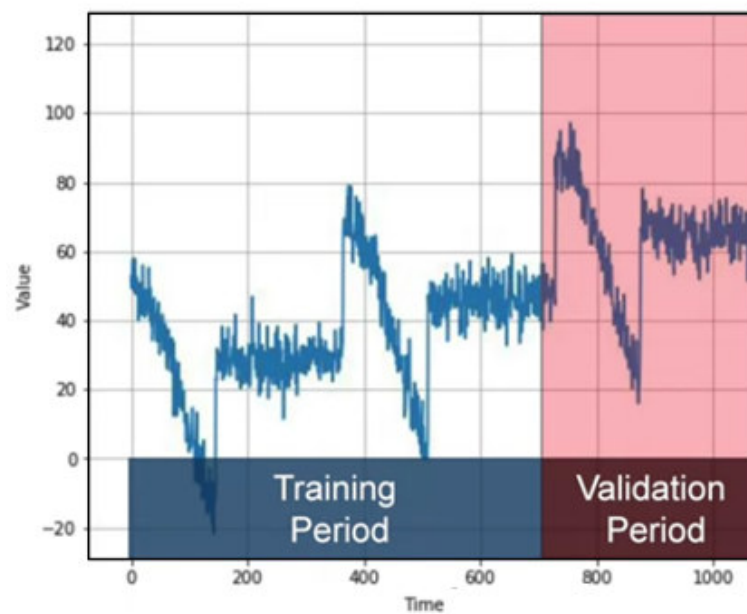


Figura 13: Exemplo de particionamento de um conjunto de dados na sessão de treinamento e validação. [25]

3 TRABALHOS RELACIONADOS

Nos últimos anos, com os avanços na inteligência artificial e o uso de conceitos como machine learning e deep learning na literatura, diferentes setores, tanto acadêmicos quanto industriais, têm buscado aplicar esse conhecimento em suas áreas, sendo o mercado financeiro um deles, onde a capacidade de prever valores futuros do mercado tem atraído considerável atenção.

No trabalho de Ballings [3], diferentes modelos de machine learning foram aplicados para prever valores futuros do mercado. Para isso, 5767 empresas europeias foram usadas como variáveis de entrada, com o modelo de Support Vector Machine (SVM) apresentando o melhor desempenho em suas previsões, seguido pelos modelos de Random Forest, Kernel Factory, AdaBoost e Regressão Logística. Esses trabalhos focados no uso de modelos de machine learning foram estudados com diferentes tipos de aplicabilidade, utilizando várias variáveis de entrada, hiperparâmetros, treinamento, etc. Para demonstrar sua adaptabilidade em diferentes cenários, mostrando que cada problema deve ser abordado especificamente. [40]

Com um número significativo de pesquisas cobrindo estudos realizados com machine learning no campo da previsão de séries temporais financeiras, deep learning tem sido subsequentemente usado nesse mesmo campo, apresentando resultados que superam seus equivalentes tradicionais de ML [34]. No trabalho de Ersan [14] foi discutido o uso dos modelos K-Nearest Neighbor, um algoritmo de aprendizado supervisionado usado para problemas de classificação, regressão, e Redes Neurais Artificiais na área financeira, obtendo melhores resultados em comparação aos fornecidos pelo modelo SVM. Em trabalhos focados em ANN, Li e Ma [23] conduziram uma revisão detalhada das implementações desses modelos em aplicações financeiras, destacando como as redes neurais oferecem resultados promissores devido à sua capacidade de encontrar relações não lineares nos dados, utilizando sistemas dinâmicos não lineares.

Ao lidar com séries temporais como um conjunto de dados, uma abordagem predominante é analisá-las como um modelo de regressão [34]. Portanto, a literatura faz uso de modelos recorrentes (RNN) para essa análise, sendo um dos mais reconhecidos, pois permite modelar os efeitos de defasagem nas séries temporais financeiras. Na literatura,

o trabalho de Sezer [34] menciona o modelo de Recurrent Neural Network (RNN) como um dos modelos iniciais de deep learning, apresentando seu excelente desempenho na previsão de valores financeiros. Isso também foi observado no trabalho de Chandra e Chand [9], onde o modelo RNN teve melhor desempenho em comparação a outros modelos de rede neural. Da mesma forma, no trabalho de Pawar [31], um modelo RNN foi usado para prever valores futuros das ações da empresa Apple, apresentando excelentes resultados e destacando o efeito do hiperparâmetro do intervalo de tempo nos resultados, ou no trabalho de Varadharajan [38] fazendo uso de Modelos LSTM-RNN para tarefas de previsão do mercado de ações, demonstrando sua capacidade de capturar tendências e flutuações em dados de séries temporais financeiras com precisão.

No trabalho de Zhang [42], uma rede recorrente mais avançada, conhecida como Long Short-Term Memory (LSTM), foi usada incorporando o conceito de “gates” em seus cálculos, mostrando resultados notáveis na captura de tendências e sazonalidades ao lidar com grandes conjuntos de dados. De forma semelhante, Sirignano e Cont [37] aplicaram 3 camadas de LSTM em seu modelo utilizando 500 ações de mercado de alta frequência, alcançando excelentes resultados na generalização de suas previsões. No entanto, o custo computacional usado para treinar essa rede LSTM com dados de negociação de alta frequência também foi extremamente alto.

No mercado brasileiro, diferentes pesquisas têm sido conduzidas no campo da inteligência artificial, apresentando resultados muito positivos. Por exemplo, Perreiras [15] aplicou redes neurais para identificar e explorar sistematicamente oportunidades de negociação no mercado brasileiro. Outro estudo significativo, de Oliveira [12], comparou a resposta de vários modelos de recomendação e seus hiperparâmetros, como Arima-Garch e ANN, na previsão de preços em diferentes setores econômicos.

Por outro lado, no contexto financeiro, Andrews e Glascock [24] apresentam a influência comportamental que eventos de mercado têm sobre ações líquidas, particularmente aquelas com alto volume, e como elas subsequentemente tendem a se estabilizar. No trabalho de Bidarkota [2], para evitar depender das previsões de lucros dos analistas, foram avaliadas as implicações empíricas de um modelo de avaliação de preços de ações desenvolvido por Ohlson (1995). Cada variável contábil foi ajustada considerando eventos periódicos de ações, demonstrando desempenho superior na avaliação de ações e previsão de retornos anormais para o próximo período.

Em conclusão, a literatura especializada destaca a eficácia do modelo RNN na previsão de séries temporais, especialmente em relação aos conjuntos de dados do mercado financeiro. As evidências fornecidas por esses estudos apoiam a escolha do RNN devido aos seus resultados superiores em comparação a outros modelos de redes neurais. Portanto, optaremos por usar o modelo RNN na análise do comportamento das ações brasileiras, visando examinar sua influência ao incorporar um fator de correção de eventos nos valores das ações.

4 METODOLOGIA

Neste capítulo será apresentada a metodologia proposta neste trabalho, aplicando os conceitos mencionados anteriormente. Para isso, esta seção está dividida em três partes. Primeiramente são levados em conta os tipos de dados utilizados e as ferramentas necessárias para os análise, depois é apresentado o respectivo agrupamento dos dados a serem inseridos no modelo junto com o método de seleção dos hiperparâmetros, e por fim o esquema e as métricas de validação do modelo.

4.1 Dataset

Como fonte de dados, são utilizadas as ações das empresas Petrobrás (PETR4), Itaúsa (ITSA4) e Vale (VALE3), reconhecidas por serem uma das mais importantes no mercado de ações brasileiro no segundo semestre de 2024, onde VALE3 tem o 13% de participação, PETR4 9% e ITSA4 3% do mercado^[1]. Dado que a extração de dados é uma tarefa complexa, os dados foram obtidos do banco de dados BovDB [8], que contém todas as informações já estruturadas e processadas previamente para cada ação. O conjunto de dados considerado foi desenvolvido pelo Grupo de Pesquisa em Gerenciamento de Informações (Ginfo) da Universidade Federal do Rio Grande (FURG), permitindo aos seus usuários criarem aplicações como análise e previsão de preços de ações. O conjunto de dados inclui as seguintes variáveis para cada ação:

- *trading date*;
- *opening price*;
- *maximum price of the day*;
- *minimum price of the day*;
- *closing price*;
- *volume*;

^[1]https://www.b3.com.br/pt_br/market-data-e-indices/indices/indices-amplos/indice-ibovespa-ibovespa-composicao-da-carteira.htm

- *correction factor*.

A implementação dos experimentos foi realizada utilizando a linguagem de programação Python². Além da linguagem em si, foram empregadas várias bibliotecas, incluindo Tensorflow³, Pandas⁴, scikit-learn⁵ e Numpy⁶.

4.2 Pré-Processamento

Para inserir os dados no modelo, é necessário agrupá-los previamente em *features* e categorizá-los com seus respectivos *labels*. Esse agrupamento foi realizado com um tamanho de *window size* de um mês (30 dias), deixando o último dado como o *label* do agrupamento, conforme indicado por [25].

Recapitulando, o objetivo deste trabalho é avaliar o impacto do fator de correção de eventos nas ações PETR4, ITSA4 e VALE3 durante o treinamento e validação de uma rede neural recorrente. Para realizar essa análise, as ações foram divididas em duas seções:

1. utilizada para o treinamento do modelo;
2. reservada para validar os resultados.

Especificamente, a ação PETR4 possui um total de 5078 preços diários (dias de negociação), dos quais 85% são destinados ao treinamento e os 15% restantes à validação. No caso da ITSA4, dos 6425 registros disponíveis, 80% dos dados são utilizados no treinamento, reservando os 20% restantes para a validação. Da mesma forma, a VALE3 possui 6011 dias de negociação, sendo que 80% são empregados no treinamento e os 20% restantes na validação. Isso assegura que o modelo seja treinado com uma quantidade suficiente de dados e que também tenha um conjunto de dados separado para validar o desempenho do modelo.

4.3 Seleção dos Hiperparâmetros

A seleção dos hiperparâmetros é uma etapa crítica no processo de construção de um modelo de rede neural, já que são as configurações do modelo que não podem ser aprendidas ao longo do treinamento e devem ser ajustadas manualmente para assim alcançar um desempenho ótimo no resultado, sendo este um processo desafiador e demorado.

Os hiperparâmetros considerados no modelo de rede neural foram o tamanho do lote (*batch size*), o tamanho da janela (*window size*), o número de camadas ocultas, o *dropout*, o número de neurônios em cada camada, a taxa de aprendizado e a função de ativação [10].

²www.python.org

³www.tensorflow.org

⁴www.pandas.org

⁵www.scikit-learn.org

⁶www.numpy.org

Neste trabalho, é utilizado o Keras Tuner [29] com o método de busca de hiperparâmetros, visto que é um dos mais populares para ajustar hiperparâmetros em modelos de aprendizado de máquina e deep learning. No contexto do Keras, a pesquisa aleatória pode ser usada para explorar diferentes valores de hiperparâmetros e encontrar as configurações ideais para um modelo de rede neural.

A ideia básica por trás do modelo é gerar configurações aleatórias de hiperparâmetros usando uma distribuição especificada e avaliar seu desempenho em um conjunto de dados. Para executar a pesquisa aleatória, primeiro definimos um espaço de busca especificando a faixa de valores para cada hiperparâmetro, e a função de perda de treinamento foi usada como a função objetivo, já que essa variável representa a diferença entre a saída prevista da rede neural e a saída real para os dados de treinamento.

O tamanho do lote é o número de amostras processadas em uma iteração durante o treinamento, enquanto o tamanho da janela se refere ao número de etapas de tempo consideradas na sequência de entrada. O número de camadas ocultas e neurônios determina a profundidade e a largura da arquitetura da rede neural, respectivamente. Por fim, a função de ativação determina a não linearidade do modelo.

4.3.1 Taxa de Aprendizado

O hiperparâmetro Taxa de Aprendizado ou *Learning Rate* é uma variável crucial para ajustar um modelo de rede neural, pois controla a velocidade com que o modelo atualiza seus pesos durante o processo de treinamento. Neste trabalho, busca-se averiguar o valor da função de perda do modelo obtida usando diferentes taxas de aprendizado para selecionar a taxa ideal que fornece o valor mais baixo de perda para o modelo. Testando sistematicamente uma variedade de taxas de aprendizado, consegue-se identificar a taxa que minimiza a função de perda, melhorando a precisão e eficácia do modelo. Essa abordagem assegura que a rede neural seja treinada ao máximo de seu potencial e fornece uma base mais confiável para fazer previsões.

4.3.2 Validação do modelo

Para avaliar o desempenho do modelo proposto, foi utilizado o erro quadrado médio (MSE) devido ao seu bom desempenho em tarefas de regressão [4]. Também foi utilizado o otimizador gradiente estocástico (SGD) para atualizar os pesos da rede durante o treinamento devido ao seu baixo custo computacional por iteração, é especialmente sendo adequado para modelos RNN. Os resultados são apresentados em termos da função de perda de cada modelo para cada época de treinamento, fazendo uma comparação entre o resultado obtido para os dados de treinamento e os dados de validação com o objetivo de garantir que o modelo não sofra *overfitting* e seja capaz de generalizar dados novos.

Como a tendência natural das ações de uma empresa é crescer ao longo do tempo (ao menos é esse o comportamento esperado), a possível discrepância entre os valores

nos intervalos de treinamento e validação levou à proposição de duas alternativas em seu pré-processamento. A primeira consistiu na criação de *fold*s, nos quais os dados foram distribuídos e divididos em 10 partes. Desses conjuntos, cinco foram utilizados para treinar o modelo, enquanto os outros cinco foram reservados para a validação. Isso com o objetivo de obter uma distribuição mais equilibrada em termos do comportamento de cada ação.

Por outro lado, um dos pré-processamentos mais conhecidos e amplamente utilizados no tratamento de dados é a normalização. Quando aplicada a sistemas de dados multivariados com escalas de valores diferentes para cada atributo, a normalização tem a finalidade de distribuir e dimensionar todos os dados em uma faixa comum, resultando em uma melhoria no tempo de computação durante o treinamento de uma rede neural.

Como segunda alternativa, neste trabalho, ao lidar com uma série temporal univariada (preço de fechamento), optou-se por normalizar os dados de entrada usando a técnica de min-max, que reescala os valores de entrada para uma escala de 0 a 1. A normalização tipo min-max foi escolhida devido à sua sensibilidade a valores atípicos e à sua ampla aplicação no processamento de séries temporais.

5 RESULTADOS OBTIDOS

Nesse capítulo, são apresentados os resultados obtidos a partir dos experimentos realizados com as ações PETR4, ITSA4 e VALE3. É crucial compreender a dinâmica e os padrões de mercado destas ações para a tomada de decisões em termos de investimento. Por esse motivo, inicialmente é apresentada uma visão geral do comportamento de cada ação, mostrando uma comparação entre suas volatilidades e os padrões presentes em cada ano.

5.1 Análise comportamental das ações

Para identificar as variações de preço e as semelhanças de comportamento de cada ação, foi plotada as volatilidades médias de cada ação para cada ano juntamente com seu desvio padrão (linha preta), conforme mostrado na Figura 14, onde as três ações apresentam valores de comportamento semelhantes em cada ano.

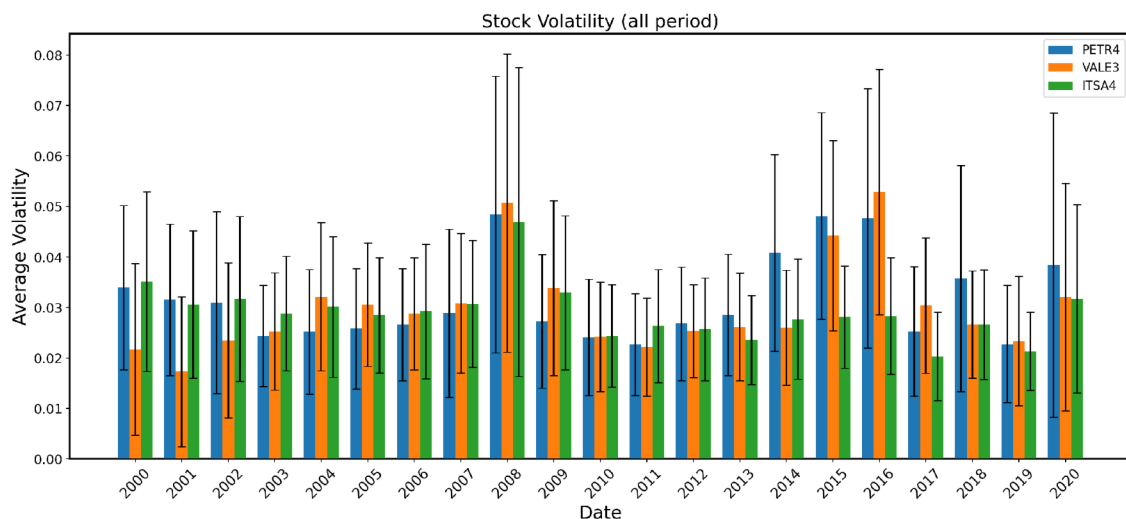


Figura 14: Volatilidade anual para cada ação.

Além disso, com as informações apresentadas na Tabela 1, pode-se observar quantitativamente uma semelhança entre suas médias, medianas e desvio padrão, juntamente com a contagem total de movimentos classificados como alta volatilidade e também os outros

que apresentaram baixa volatilidade para cada uma das ações.

Tabela 1: Informação estatística para cada ação.

Nome	Média	Desvio Padrão	Mediana	Volatilidade	
				Classificação de volatilidade # Alta	# Baixa
PETR4	0.0336	0.0199	0.0277	3233	3195
ITSA4	0.0290	0.0177	0.0254	3213	3212
VALE3	0.0285	0.0199	0.0246	3146	3145

No total, as ações na Tabela 1 apresentam um comportamento global semelhante, especialmente com um número equivalente de movimentos no mercado de ações, classificados como alta e baixa volatilidades para cada uma das ações com base no valor médio de cada ação, indicando a complexidade de seu comportamento. Isso sugere que as ações exibem níveis semelhantes de imprevisibilidade, independentemente da magnitude das mudanças de preço. Essa complexidade pode ser atribuída a uma variedade de fatores, incluindo condições de mercado, desempenho da empresa e eventos externos.

As Figuras 15, 16 e 17 ilustram o comportamento da variável “preço de fechamento” para as ações PETR4, ITSA4 e VALE3, respectivamente, ao longo dos anos. A linha azul representa a seção de treinamento, enquanto a linha vermelha a seção de validação usada no modelo de RNN. As figuras também apresentam as correções realizadas pelo fator de correção para atenuar o impacto dos eventos nos preços das ações. Na Tabela 2, foi compilado uma lista de todos os eventos, destacando sua frequência de ocorrência nas ações, acompanhada por um exemplo ilustrativo que inclui a data em que um desses eventos ocorreu.

Tabela 2: Eventos em cada ação analisada

PETR4				ITSA4				VALE3			
Name	#	ID	Example	Name	#	ID	Example	Name	#	ID	Example
Ex-Divided	20	ED	03-2003	Ex-Divided	85	ED	02-2006	Ex-Divided	22	ED	10-2006
Ex-Bonus	2	EB	08-2005	Ex-Subs	16	ES	11-2008	Ex-Bonus	5	EB	05-2006
Ex-Interest	43	EJ	09-2004	Full Rights	3	INT	04-2007	Ex-Interest	45	EJ	08-2018
Ex-Stock S	3	ER	12-2018	Ex-Bonus	10	EB	05-2009				
Ex-Stock I	1	EG	06-2000	Ex-Interest	83	EJ	06-2013				
Ex-Rights	3	EX	03-2001	Ex-Rights	4	EX	08-2000				

Para as ações PETR4 e VALE3, observa-se nas figuras 15 e 17 uma volatilidade significativa, principalmente com picos bastante pronunciados em anos anteriores a 2010 e posteriormente estabilizando-se. Esses picos são decorrentes de eventos de mercado ocorridos nessas datas específicas, conforme observado nos modelos aplicados com fator para ambos os casos, onde estes não estão mais presentes no comportamento da ação já corrigida. No caso da ITSA4, a ação apresenta crescimento constante até 2010 seguido de um período com alguma volatilidade, mas estável.

Como os eventos *Ex-Interest*, *Ex-Divided* e *Ex-Bonus* são os mais presentes na Tabela 2 para as três ações, infere-se que nos respectivos anos houve uma forte política

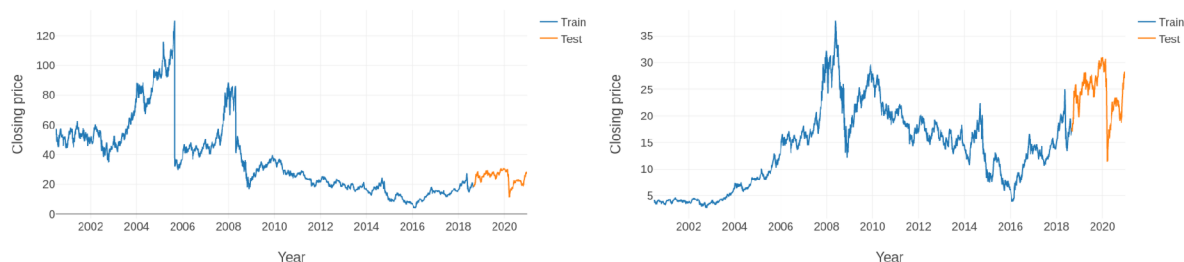


Figura 15: Preço de fechamento de PETR4 sem (esq.) e com fator de correção (dir.).

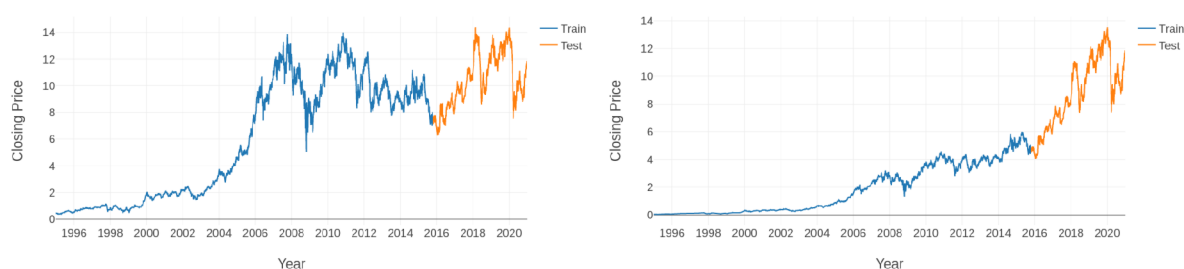


Figura 16: Preço de fechamento de ITSA4 sem (esq.) e com fator de correção (dir.).

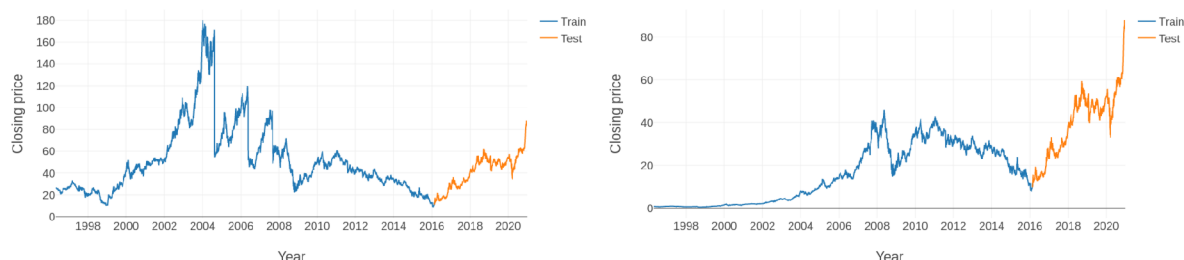


Figura 17: Preço de fechamento de VALE3 sem (esq.) e com fator de correção (dir.).

de distribuição de lucros e ajustes financeiros condições para essas empresas, indicando uma política consistente de distribuição de dividendos e juros, proporcionando oportunidades para os atuais acionistas aumentarem sua participação na empresa em condições preferenciais.

Em resumo, os dados apresentados apontam para um histórico de alterações significativas nos preços das ações, com múltiplos eventos corporativos que podem ter influenciado a percepção e o comportamento dos investidores. Esta análise destaca a importância de considerar eventos corporativos ao avaliar o desempenho das ações ao longo do tempo.

5.1.1 Hiperparâmetros

Com o objetivo de determinar a melhor taxa de aprendizado (L_r) para cada modelo, as Figuras [18](#), [19](#) e [20](#) apresentam a função de perda obtida para diferentes valores de taxa de

aprendizado. Visualmente pode se observar que para Petrobras e Vale (Figuras 18 e 20), o modelo com fator de correção consegue apresentar uma convergência mais pronunciada para o resultado desejado, reduzindo o ruído do treinamento.

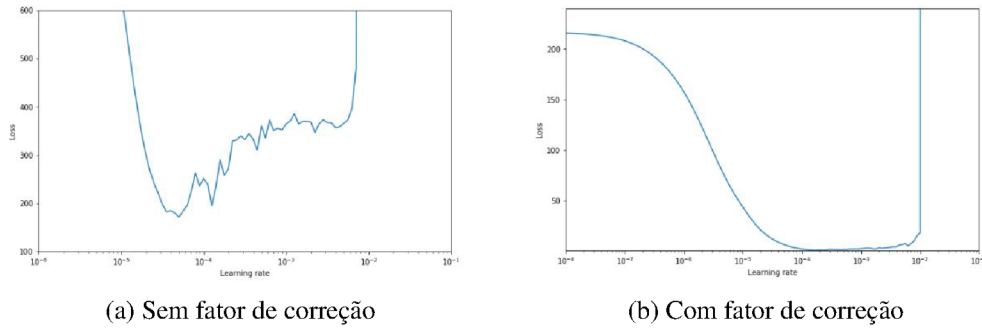


Figura 18: Loss versus Learning rate PETR4.

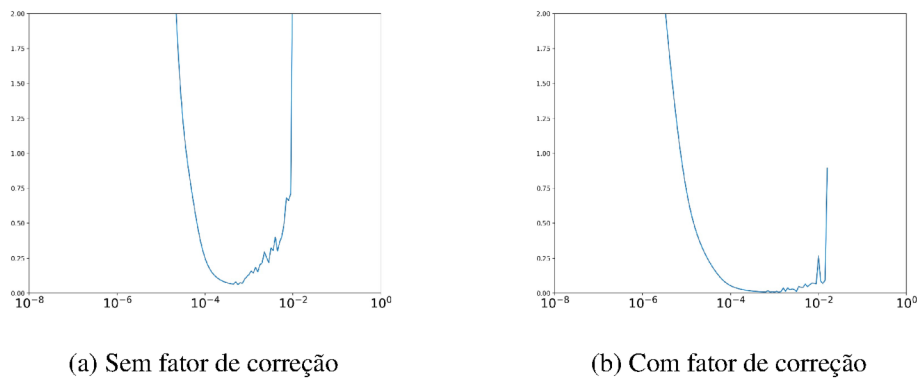


Figura 19: Loss versus Learning rate ITSA4.

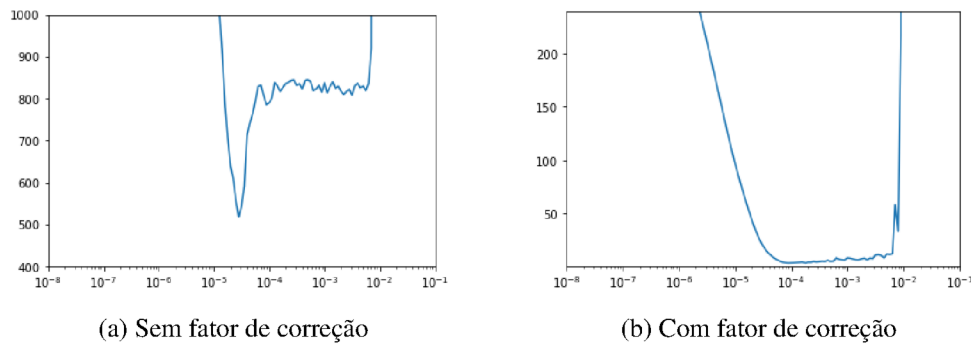


Figura 20: Loss versus Learning rate VALE3.

Em relação aos demais hiperparâmetros mencionados na seção 4.3, por meio de uma pesquisa aleatória, foram obtidos os resultados apresentados na Tabela 3. Nela, são exibidas as três melhores configurações que resultaram no menor valor de perda durante o

treinamento para cada ação, respectivamente. As colunas Units_1 e Units_2 representam a quantidade de neurônios em cada camada. A primeira entrada para cada ação na Tabela 3 corresponde à melhor configuração do respectivo modelo, sendo apresentado o resultado em negrito.

Essas configurações também foram usadas nos modelos treinados com os dados ajustados com o fator de correção correspondente da ação, permitindo uma comparação de desempenho entre as duas abordagens.

Tabela 3: Melhores hiperparâmetros para cada ação

	Configuration	Units_1	Activation	Units_2	Activation	Batch_size	Score_Loss
PETR4	1	32	relu	128	tanh	64	3,215
	2	128	relu	128	relu	64	3,226
	3	32	relu	128	relu	32	3,233
ITSA4	1	128	tanh	128	tanh	64	0,046
	2	50	tanh	32	tanh	64	0,055
	3	50	tanh	50	relu	64	0,057
VALE3	1	128	tanh	32	tanh	64	91,58
	2	64	tanh	128	relu	32	94,693
	3	64	relu	128	relu	32	97,005

5.1.2 Treinamento e predição

A Figura 21 mostra o resultado do treinamento para a ação PETR4 aplicada com e sem o fator de correção. Para o modelo sem o fator, a função de perda diminui constantemente e depois se estabiliza tanto para os dados de treinamento quanto para os de validação. Esse comportamento indica que o modelo é capaz de gerar um aprendizado. No caso do modelo com fator de correção, o comportamento é semelhante, mas com uma menor escala de valores de perdas, mostrando-se mais eficaz, possivelmente por trabalhar com dados mais homogêneos já que estes não contam com a influência de eventos de mercado.

Isso se reflete na Figura 22 onde os dados de validação (linha azul) são comparados com o resultado obtido pelo modelo (linha vermelha). Pode-se observar que o modelo aplicado com o fator apresenta melhor similaridade com os dados em comparação com o modelo sem o fator, onde o erro para ambos casos foi calculado usando a função de erro médio absoluto, resultando em 0,558 para o modelo sem fator e 0,530 para o modelo com fator. Esses resultados demonstram uma melhoria do modelo quando se aplica o método de correção de eventos do mercado aos dados, podendo ser mais confiável no momento da tomada de decisões baseada em previsões.

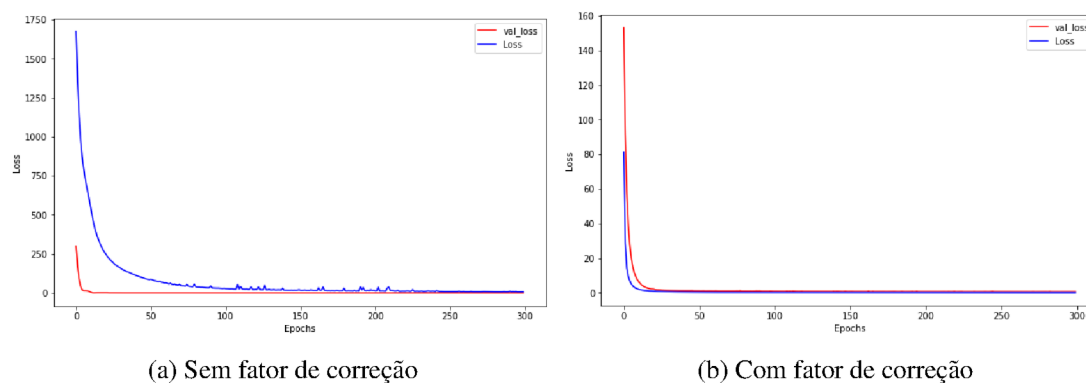


Figura 21: Função de perda vs época para dados de treinamento PETR4 (azul) e validação (vermelho).

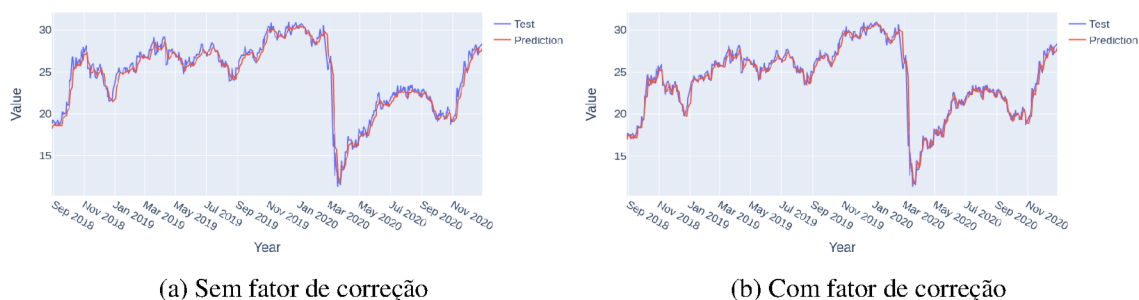


Figura 22: Dados de validação (linha azul) e previsão do modelo (linha vermelha) para PETR4.

A função de perda obtida para a ação ITSA4 é apresentada na Figura [23](#). Pode-se observar que tanto os modelos com e sem o fator de correção de eventos de mercado mostram uma tendência decrescente ao longo de cada época, o que indica que o modelo está melhorando a cada iteração. No entanto, no modelo aplicado com o fator, é perceptível que a função de perda dos dados de validação é significativamente mais alta em comparação com a função de perda dos dados de treinamento, indicando que não está se ajustando bem a essas informações. Isso ocorre quando o modelo está aprendendo os dados de treinamento muito bem, tornando-se muito específico para eles e fazendo impossível generalizar corretamente todos os dados de validação.

Na Figura [24](#), no modelo aplicado sem o fator consegue se ajustar com os dados de validação, mas apresenta algumas falhas em certas partes como pode ser visto em junho do 2018 o que pode indicar uma capacidade limitada do modelo para capturar todos os padrões nos dados. Por outro lado, o modelo aplicado com o fator inicialmente apresenta uma previsão semelhante ao dados de validação, mas após julho de 2016, começa a apresentar falhas.

Para validar a influência do corretor de eventos, o erro obtido até essa data foi cal-

culado em ambos os casos. O modelo sem o fator obteve um valor de erro de 0,204, enquanto o modelo aplicado com o fator teve um erro de 0,112, demonstrando sua influência positiva. O problema que ocorre após essa data, no modelo corrigido, está na diferença na variação do preço de fechamento entre treinamento e validação, onde a primeira varia de valores de 0,05 a 4,97 e a segunda de 4,02 a 13,51 como pode-se observar na Figura 16 (à direita). Esse mesmo comportamento é observado no trabalho de Selvin [33], onde tanto os modelos RNN quanto LSTM falham em capturar as tendências e dinâmicas nos períodos finais da empresa INFOSYS. Onde eles atribuem esse resultado como uma mudança no padrão de comportamento das ações para essas janelas de tempo em comparação com janelas anteriores.

Como estamos trabalhando com modelos de Rede Neural Recorrente (RNN), eles são capazes de extrapolar suas previsões até certo ponto, mas sua capacidade depende de vários fatores, como por exemplo a escala dos dados, variável que será abordada posteriormente. Portanto, como a maioria dos valores de validação está fora da faixa de treinamento, torna-se muito difícil fazer uma previsão precisa de toda a série financeira desta ação.

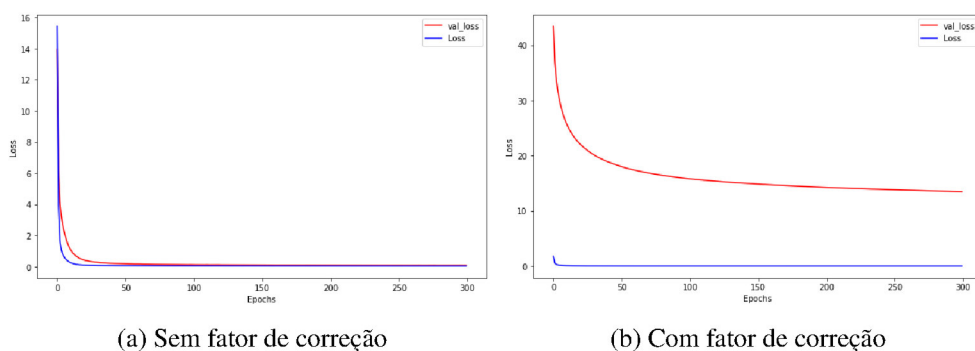


Figura 23: Função de perda dos dados de treinamento (azul) e validação (vermelho) da ITSA4.

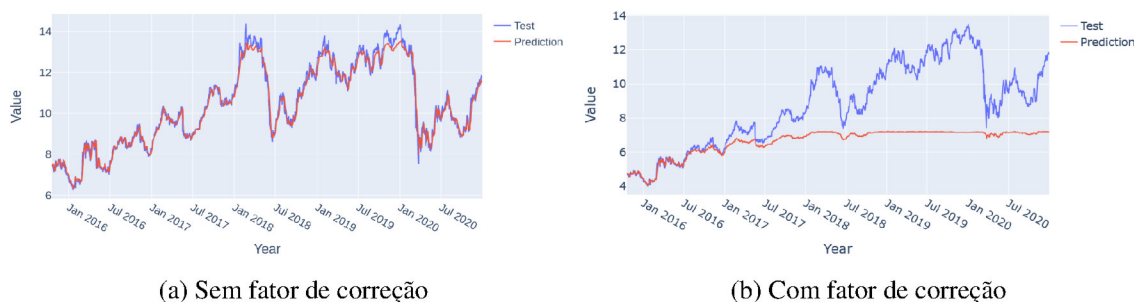
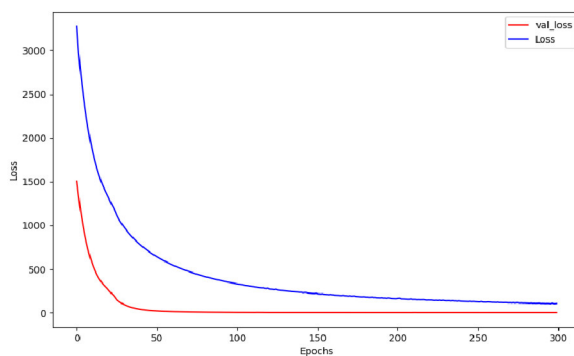


Figura 24: Dados de validação (linha azul) e previsão do modelo (linha vermelha) para ITSA4.

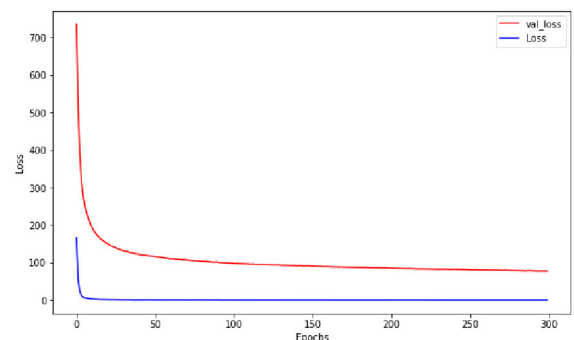
Analisando agora a ação VALE3, apresentado na Figura 25, o modelo sem o fator

mostra uma diminuição progressiva em sua função de perda, indicando o treinamento constante do modelo. Além disso, a função de perda de validação é menor do que a função de perda de treinamento, o que ocorre quando o modelo acaba se ajustando mais aos dados de validação do que aos dados de treinamento. Como mostrado na Figura 17, os dados de treinamento apresentam muito mais variação em seu comportamento do que os dados de validação, tornando muito mais difícil gerar um padrão a partir deles. Por esse motivo, a função de perda dos dados de validação, que é composta por dados muito menos ruidosos do que os dados de treinamento, mostra uma maior diminuição na previsão de cada época em comparação com o valor de perda dos dados de treinamento. Portanto, um valor mais baixo em termos de função de perda é observado para os dados de validação do que para os dados de treinamento.

Por outro lado, o comportamento do modelo aplicado com o fator é semelhante ao que aconteceu com a ação ITSA4, onde os valores de perda de validação são muito mais altos em comparação com a perda obtida com os dados de treinamento. Na Figura 26, os valores reais do modelo são comparados aos valores previstos. Os resultados para o modelo sem a correção são bastante semelhantes aos dados de validação (linha azul), indicando um bom desempenho. No entanto, o modelo com o fator apresenta uma limitação em sua previsão devido à diferença na faixa de valores entre os conjuntos de treinamento e teste. Para calcular o erro, todos os dados até a data dessa limitação foram considerados, resultando em um erro de 0,629 para o modelo sem o fator e 0,584 para o modelo com o fator, demonstrando sua eficácia.



(a) Sem fator de correção



(b) Com fator de correção

Figura 25: Função de perda dos dados de treinamento (azul) e validação (vermelho) da VALE3.

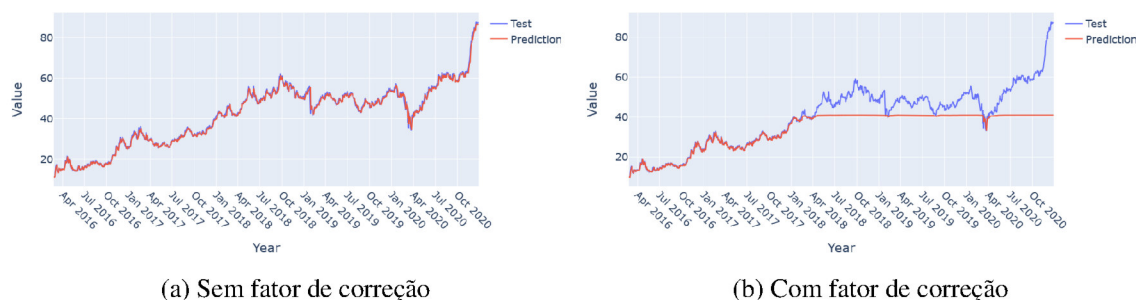


Figura 26: Dados de validação (linha azul) e previsão do modelo (linha vermelha) para VALE3.

Para mitigar/reduzir o impacto negativo causado pela discrepância nos intervalos de valores entre as seções de treinamento e validação ao prever valores fora da faixa de treinamento em um modelo RNN, especificamente nas ações ITSA4 e VALE3 aplicadas com o fator de correção, foram sugeridas neste trabalho duas alternativas, que serão apresentadas a seguir.

5.1.2.1 Folds

Ao realizar uma divisão em 10 folds e empregando os folds vermelhos para o treinamento, as distribuições são apresentados nas figuras 27 e 28.

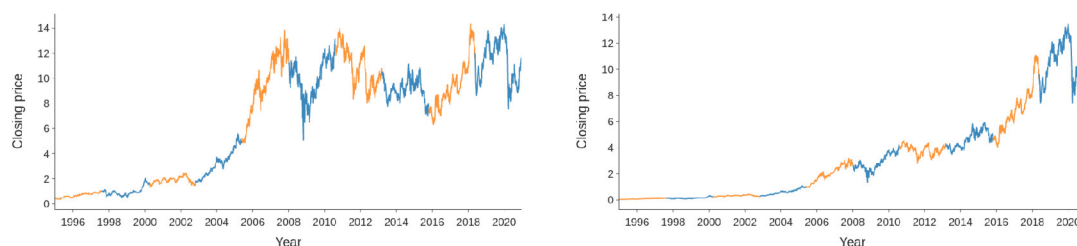


Figura 27: Preço de fechamento de ITSA4 sem (esq.) e com fator de correção (dir.) usando cinco folds para treinamento e cinco para validação.

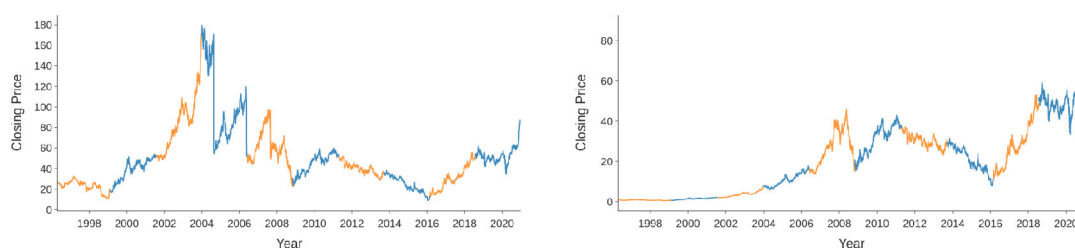


Figura 28: Preço de fechamento de VALE3 sem (esq.) e com fator de correção (dir.) usando cinco folds para treinamento e cinco para validação.

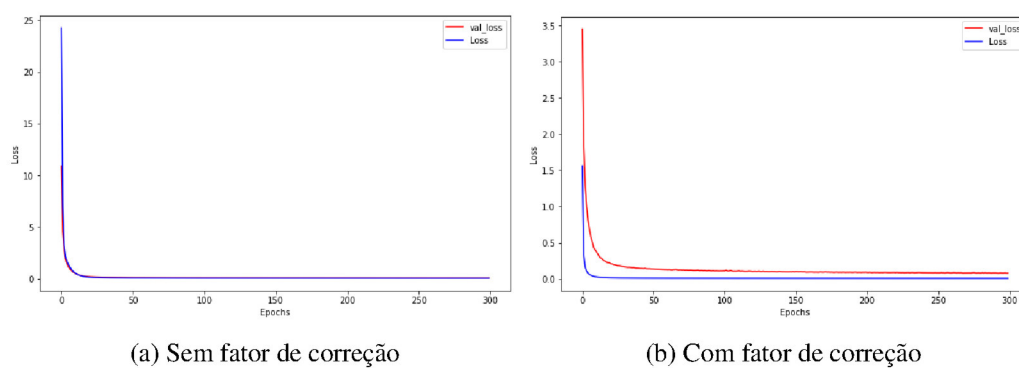


Figura 29: Função de perda dos folds de treinamento (azul) e validação (vermelho) da ITSA4

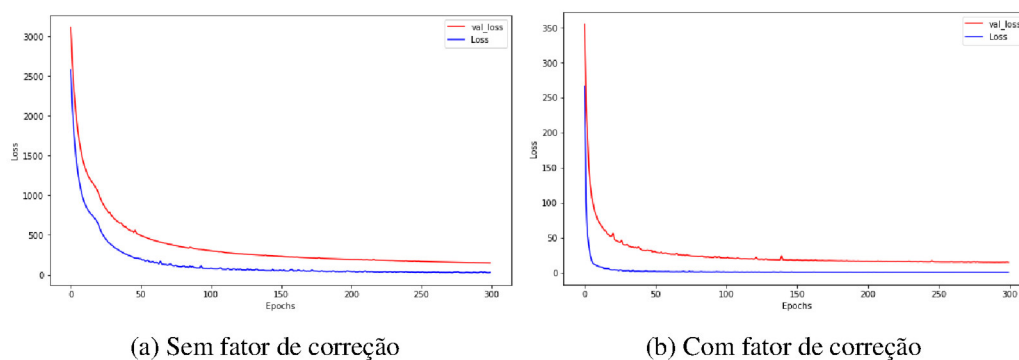


Figura 30: Função de perda dos folds de treinamento (azul) e validação (vermelho) da VALE3.

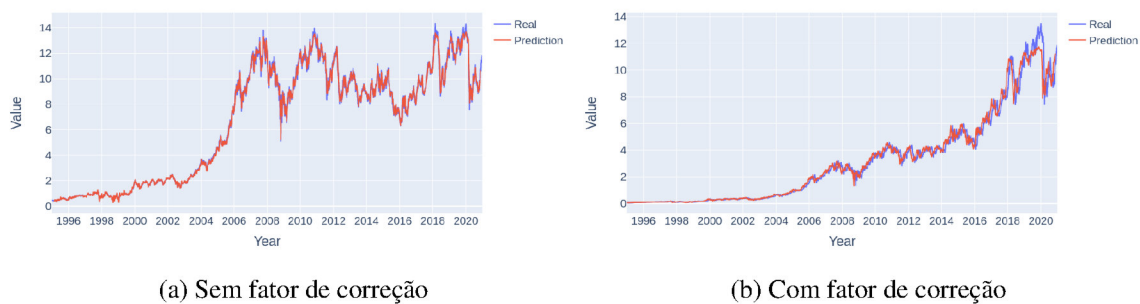


Figura 31: Dados de validação (azul) e previsão do modelo (vermelho) para ITSA4, com folds.

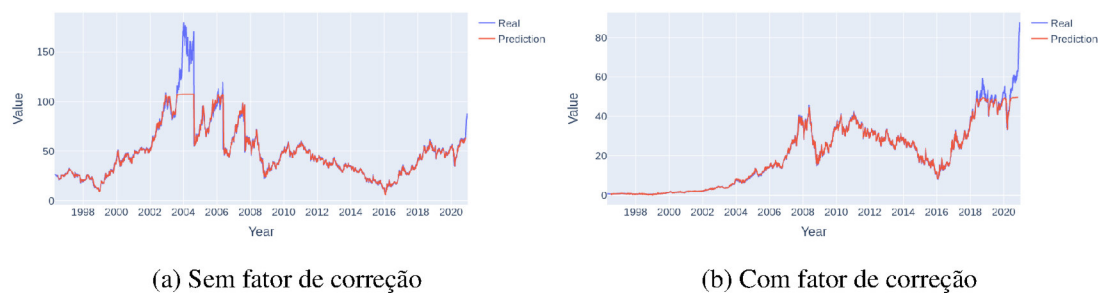


Figura 32: Dados de validação (azul) e previsão do modelo (vermelho) para VALE3, com folds.

Na Figura 29 é possível observar como, em ambos, o treinamento da ação ITSA4 ocorre ao longo de cada época reduzindo o erro. Da mesma forma, na Figura 30 é apresentado o resultado do treinamento em cada época para a ação VALE3, com e sem o fator de correção. A Figura 31 apresenta a previsão do modelo junto com os dados de validação da ação ITSA4, alcançando uma previsão considerável. Em termos quantitativos, o modelo aplicado com o fator de correção obteve um valor de erro de 0,164, em comparação com o modelo sem fator, que obteve um valor de 0,197. Por outro lado, os resultados obtidos para a ação VALE3 também demonstram um bom desempenho em sua previsão, embora ainda haja limitações devido aos valores entre as faixas de treinamento e validação. Em ambos os casos, há uma seção dos dados em que o modelo não pôde gerar um resultado satisfatório. O erro para o modelo com o fator foi de 1,310, enquanto o modelo sem o fator teve um erro de 1,502.

Para abranger todos os casos, as duas partições acima foram invertidas, resultando na seção seguinte.

5.1.2.2 Folds invertidos

Ao inverter os conjuntos de dados entre as partições de treinamento e validação, obtiveram-se os seguintes resultados para a ação ITSA4.

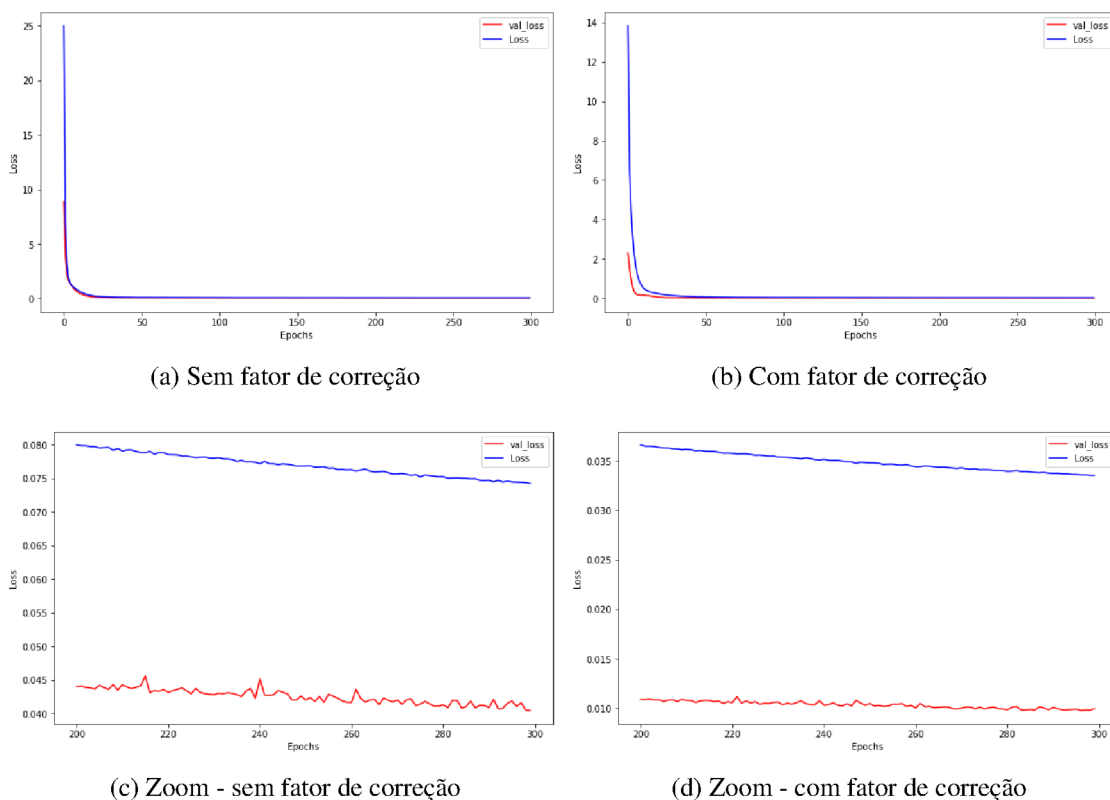


Figura 33: Função de perda para folds de treinamento (azul) e validação (vermelho) de ITSA4.



Figura 34: Previsão (vermelho) do modelo com os dados de validação (azul) de ITSA4, sem (esquerda) e com fator de correção (direita), com folds invertidos.

A Figura 33 demonstra como, em cada época, o treinamento tem uma maior afinidade com os dados de validação, com um valor de perda sendo menor em comparação com o obtido nos dados de treinamento para cada iteração. O resultado é mostrado na Figura 34, onde ambos modelos apresentam uma previsão bastante considerável, com um erro de 0,142 para o modelo sem fator e de 0,121 para o modelo com fator. Demonstrando o efeito positivo que os folds têm na geração de uma previsão, além de apresentar a influência positiva do fator de correção no resultado.

No caso da ação VALE3, o modelo sem o fator de correção apresenta um treinamento bastante instável, como pode ser observado na Figura 35, onde a função de perda apresenta muito ruído durante o treinamento. Por outro lado, o treinamento com o modelo aplicado com o fator obteve um desempenho consistente em cada época.

Na Figura 36 observa-se uma diferença bastante forte entre a previsão dos dados aplicados sem e com o fator de correção, onde o treinamento realizado com o fator apresenta um comportamento muito mais semelhante aos dados de validação para a ação VALE3, enquanto o modelo sem fator apresenta diferentes variações em seu comportamento. O erro obtido para o modelo sem fator foi 6.024 enquanto para o modelo com fator 1.215, demonstrando como ao neutralizar o efeito causado pelos eventos nas ações ao fazer uma previsão, influenciando positivamente o resultado.

De forma geral, os resultados apresentados mostram como o fator de correção de eventos permite obter uma melhor previsão nos modelos devido à diminuição do ruído no seu comportamento, tendo um erro menor comparado aos modelos sem este fator de correção.

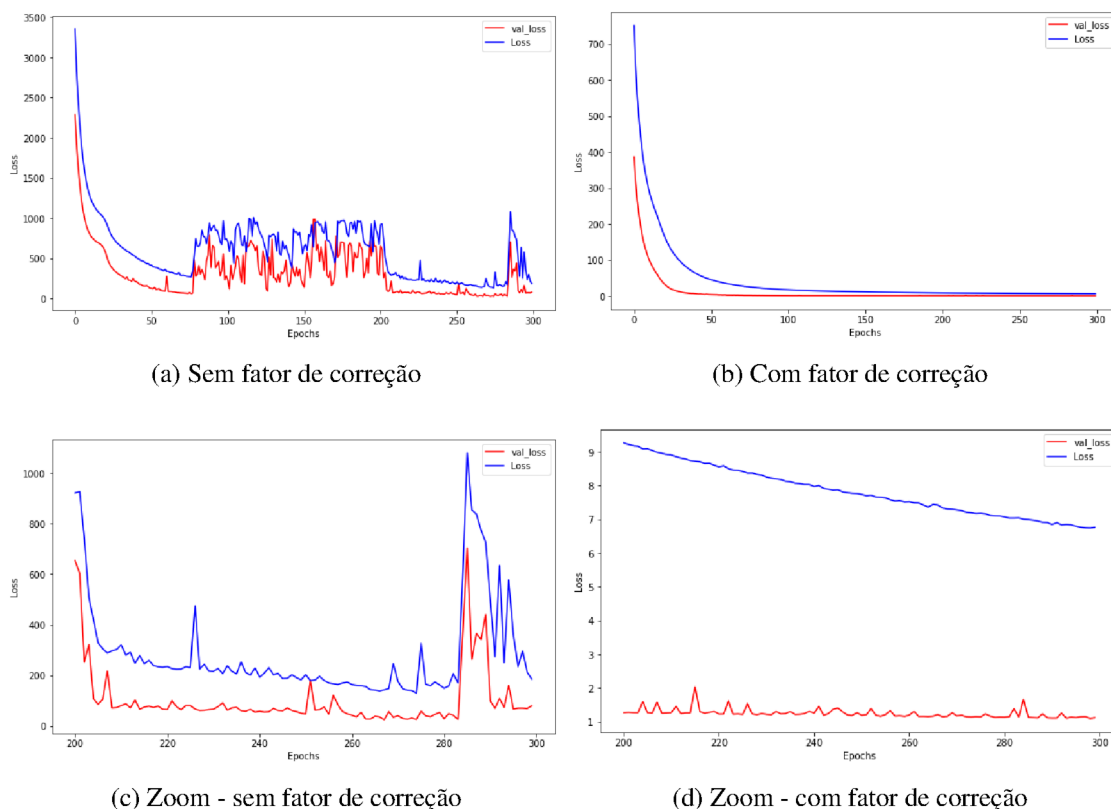


Figura 35: Função de perda para folds de treinamento (azul) e validação (vermelho) de VALE3.



Figura 36: Previsão (linha vermelha) do modelo com os dados de validação (linha azul) do VALE3. Sem (Esq) e com (Dir) factor, com folds invertidos.

5.1.3 Normalização de uma série temporal financeira

Tendo os dados normalizados conforme foi detalhado na metodologia, com o objetivo de avaliar a influência desse pré-processamento na arquitetura do modelo, os hiperparâmetros foram otimizados novamente, resultando na Tabela 4. Nela, é possível observar que, no caso da ação ITSA4, a configuração com o melhor resultado não apresentou alterações no número de neurônios, alterando apenas a função de ativação da segunda camada. Na ação VALE3, houve alterações em ambas funções de ativação e uma diminuição relevante no número de neurônios da primeira camada, passando de 64 para 32 unidades.

Tabela 4: melhores hiperparâmetros para ações normalizadas

	Configuration	Units_1	Activation	Units_2	Activation	Batch_size	Score_Loss
ITSA4	1	128	relu	128	relu	64	0,00800
	2	50	tanh	32	relu	64	0,01100
	3	32	relu	128	tanh	32	0,01500
VALE3	1	32	relu	32	relu	64	0,00028
	2	32	relu	128	tanh	32	0,00032
	3	32	tanh	128	relu	32	0,00037

Uma vez que a nova arquitetura está definida, o modelo para a ação ITSA4 apresentou os seguintes resultados.

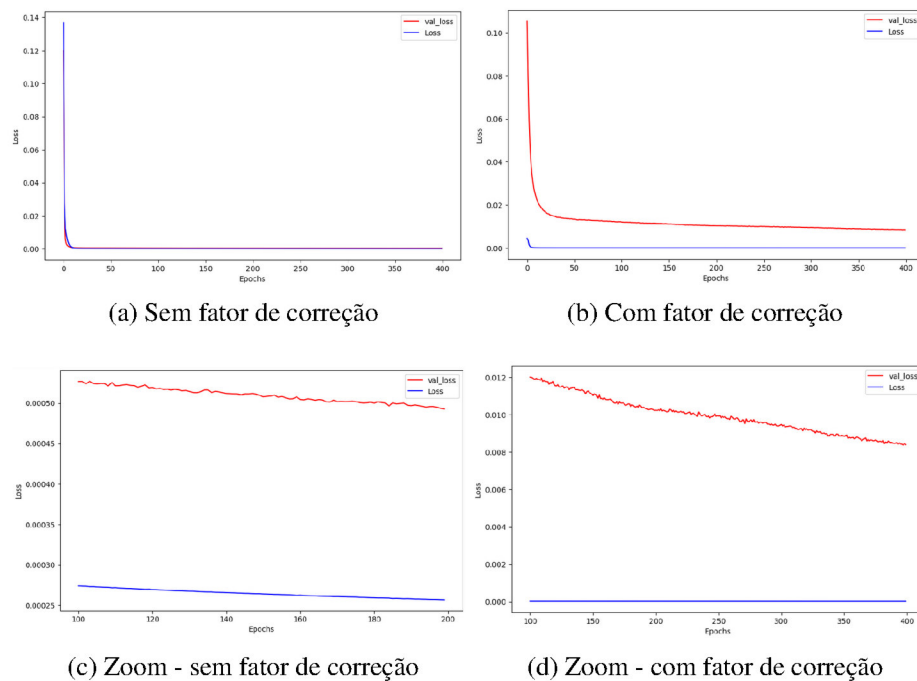


Figura 37: Função de perda para dados de treinamento (azul) e validação (vermelho) de ITSA4, com normalização.

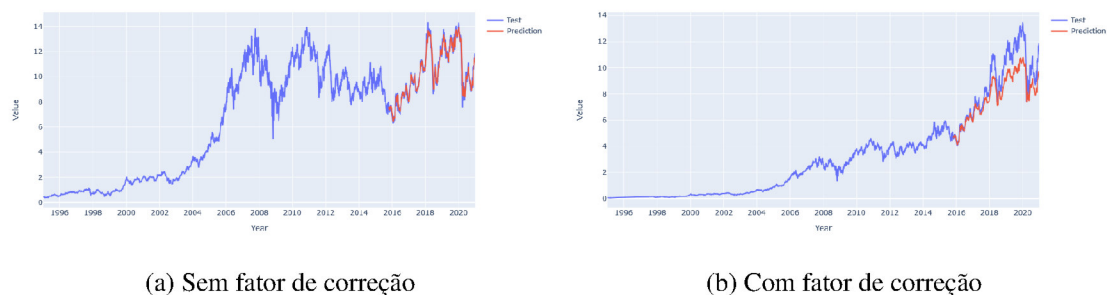


Figura 38: Previsão (vermelho) do modelo com os dados de validação (azul) de ITSA4, com normalização.

As Figuras 37 e 38 mostram que para o modelo aplicado sem fator, ele adota muito bem os dados de novo dando uma boa previsão, mas o resultado mais relevante é o efeito que o modelo com fator está tendo com a normalização, onde embora a previsão não seja exata, o modelo consegue prever valores superiores aos utilizados no momento do treinamento, além de ajustar seu comportamento e tendência. Isso ocorre principalmente porque os modelos RNN tendem a ser bastante sensíveis ao escalonamento de suas entradas, principalmente quando trabalham com funções de ativação como tanh e relu, já que podem ser propensos a problemas de gradientes que desaparecem ou exploram devido aos dados de entrada. [36]. O modelo com fator apresentou um erro de 0,962, enquanto o modelo sem fator registrou um erro de 0,201, representando uma melhoria significativa para este último em comparação com o resultado inicial apresentado na Figura 24. Os resultados normalizados para VALE3 são apresentados a seguir.

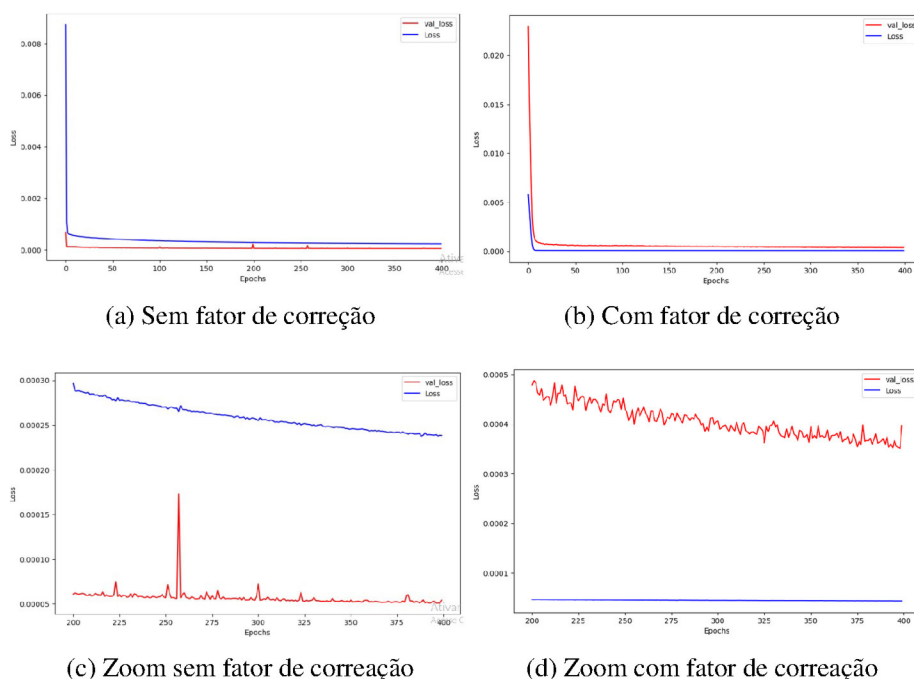


Figura 39: Função de perda para dados de treinamento (azul) e validação (vermelho) de VALE3, com normalização.

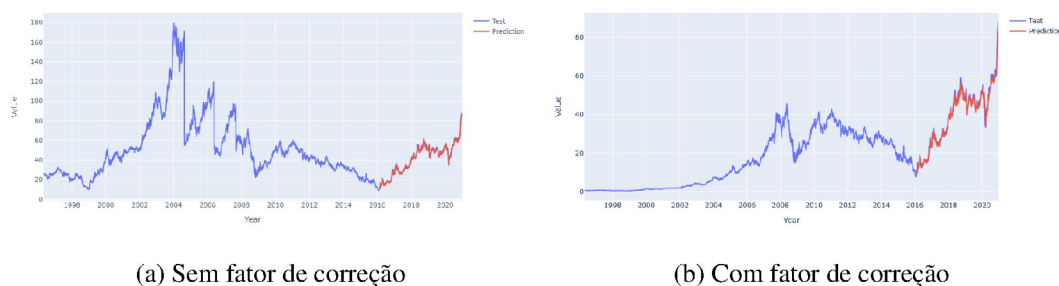


Figura 40: Previsão (vermelho) do modelo com os dados de validação (azul) de VALE3, com normalização.

As Figuras 39 e 40 mostram como o modelo aplicado sem fator apresenta um bom treinamento, com um erro entre sua previsão e os dados de validação de 0,887. Já o modelo aplicado com fator de correção de preços consegue gerar uma boa previsão para valores fora da faixa de treinamento, demonstrando novamente o efeito positivo da normalização em modelos univariáveis e com faixas de validação maiores que os dados de treinamento. Neste caso, o erro do modelo é de 1,202, sendo este maior devido a não atingir os valores mais elevados de validação.

Sendo a normalização o pré-processamento que permite atingir a diferença dos valores entre as faixas de treinamento e validação em um modelo RNN como se mostrou nos resultados anteriores, com o objetivo de melhorar o desempenho do modelo, foi integrado o uso de regularizações. Inicialmente, esta abordagem não apresentou melhoria significativa ao modelo não normalizado (Tabela 3). Agora tomando como referência a arquitetura estabelecida na Tabela 4, os reguladores *dropout* e *recurrent dropout* foram adicionados ao otimizador keras, resultando na Tabela 5, onde *recurrent dropout* é o que apresenta melhor resultado.

Os resultados são apresentados a seguir.

Tabela 5: Melhores hiperparâmetros para ações com normalização e reguladores

	Configuration	Units_1	Activation	regul	Units_2	Activation	regul	Batch_size	Score_Loss
ITSA4	1	128	relu	0.2	128	relu	0	64	0,00800
	2	50	tanh	0.1	32	relu	0.15	64	0,01100
	3	32	relu	0.2	128	tanh	0.25	32	0,01500
VALE3	1	32	relu	0.2	32	relu	0.3	64	0,00028
	2	32	relu	0	128	tanh	0.2	32	0,00032
	3	32	tanh	0.2	128	relu	0.15	32	0,00037

As Figuras 41 e 42 mostram o resultado da aplicação de uma normalização e posteriormente regularização sobre um conjunto de dados univariados em um modelo RNN, onde ambos os casos apresentam bom comportamento em sua previsão, principalmente o modelo com fator de correção de preços. Neste caso, ao conseguir atingir valores maiores que os usados no seu treinamento, permite obter um erro de 0,151, que é menor comparado ao erro do modelo sem fator de 0,173.

No caso do VALE3, o resultado obtido pela normalização dos dados, além da utilização dos reguladores estabelecidos na tabela 5, são apresentados a seguir.

Como resultado, temos as Figuras 43 e 44 onde ambos os modelos apresentam bom treinamento e boa previsão. No caso do modelo aplicado com o fator conseguiu prever os maiores valores do modelo, dando um erro de 0,811, que é menor que o erro de 0,855 do modelo sem o fator, demonstrando mais uma vez que o o modelo aplicado com a correção de eventos influencia positivamente a predição de um modelo RNN.

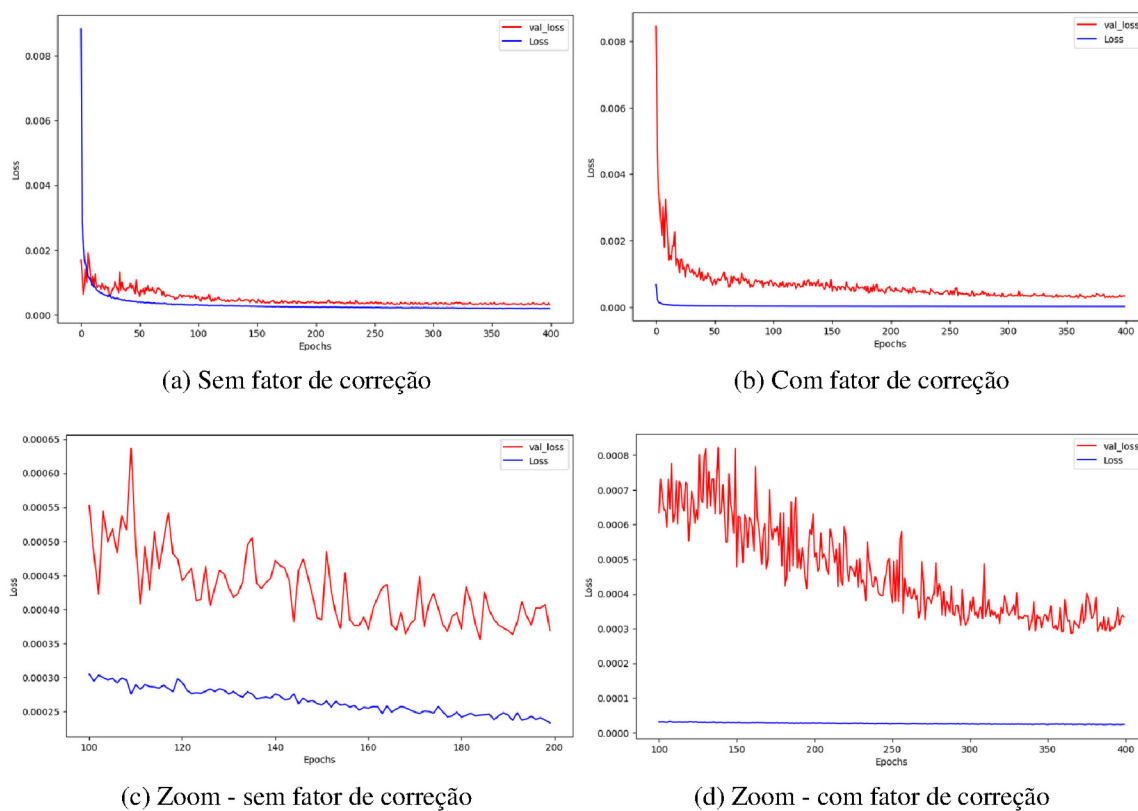


Figura 41: Função de perda para dados de treinamento (azul) e validação (vermelho) de ITSA4, com normalização e dropout.



Figura 42: Previsão (vermelho) do modelo com os dados de validação (azul) de ITSA4, com normalização e dropout.

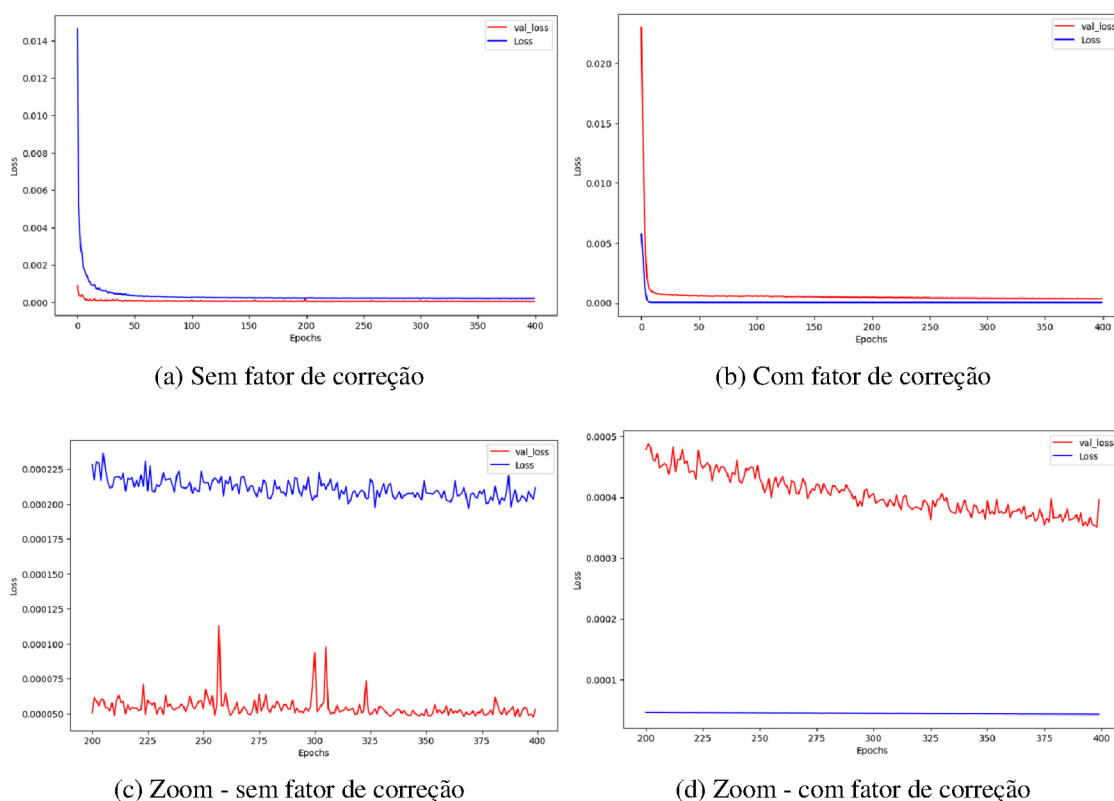


Figura 43: Função de perda para dados de treinamento (azul) e validação (vermelho) de VALE3, com normalização e dropout.

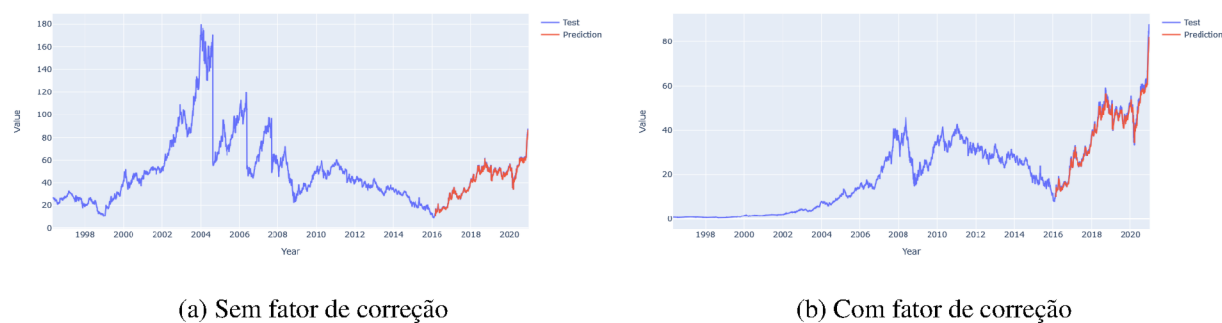


Figura 44: Previsão (vermelho) do modelo com os dados de validação (azul) de VALE3, com normalização e dropout.

Na Tabela [6](#), são apresentados todos os resultados obtidos neste trabalho utilizando os dados corrigidos com fator e sem fator (semF). Além disso, há uma terceira coluna que mostra o percentual de melhoria alcançada pelo modelo ao aplicar o fator de correção em cada caso de análise. Esta tabela permite uma comparação clara e direta entre o desempenho do modelo com e sem a aplicação do fator de correção, evidenciando a eficácia e a importância desse ajuste nos resultados obtidos.

Tabela 6: análise comparativa dos resultados obtidos

Nome	Resultado Factor vs Sem Factor											
	Holdout			Folds			Folds Inv			Norm-Regul		
	factor	semF	factor/SemF	factor	semF	factor/SemF	factor	semF	factor/SemF	factor	semF	factor/SemF
ITSA4	0.112	0.204	45%	0.164	0.196	16%	0.121	0.142	15%	0.151	0.173	13%
VALE3	0.584	0.629	7%	1.310	1.502	13%	1.215	6.024	20%	0.811	0.855	5%

6 CONCLUSÃO

Neste trabalho, realizou-se uma revisão exploratória sobre a capacidade preditiva de um modelo de rede neural recorrente (RNNs) aplicado às ações do mercado financeiro do Brasil, mais especificamente para PETR4, ITSA4 e VALE3. Onde foi realizado um comparativo entre os resultados obtidos usando os dados base e ajustando eles por meio de um fator de correção, que permite uma avaliação de dados desconsiderando eventos de natureza diferente da precificação de oferta e demanda de mercado

A principal contribuição deste trabalho está no efeito positivo gerado na previsão de um modelo RNN ao realizar uma correção nos valores das ações por meio de eventos ocorridos durante o período considerado. Além disso, são apresentadas duas alternativas de pré-processamento ao executar o modelo para avaliar um intervalo de valores diferente daquele utilizado no treinamento. Ao contrário de portais de notícias como InfoMoney ou Yahoo Finance, que fornecem dados já pré-processados com base em eventos não regulares de negociação, mas sem transparência sobre como isso foi implementado.

Os resultados mais relevantes indicam que a aplicação do fator de correção de eventos resultou em melhorias nas previsões, especialmente nos dados de validação. Além disso, ao lidar com previsões fora do intervalo de valores de treinamento, como foi o caso de ITSA4 e VALE3, onde houve discrepâncias significativas entre os conjuntos de treinamento e teste, foram apresentadas duas alternativas. A primeira, que consiste na divisão dos dados para obter uma distribuição mais equitativa entre treinamento e validação, mostrou-se uma estratégia eficaz para melhorar a generalização do modelo. Por outro lado, observou-se que a normalização e a regularização Dropout recorrente aplicadas ao modelo possibilitaram prever com sucesso os valores de validação, alcançando resultados satisfatórios.

Como trabalhos futuros, prevê-se a aplicação dessa metodologia a um maior número de ações para abordar diversos eventos de mercado, obtendo uma perspectiva mais ampla sobre a robustez e aplicabilidade geral do método proposto. Além disso, considera-se a pesquisa e aplicação de técnicas de RNN mais avançadas que possam complementar ou aprimorar ainda mais a previsão das ações no ambiente financeiro.

REFERÊNCIAS

- [1] Alpaydin, E.: Introduction to machine learning. MIT press (2020)
- [2] Awwal, F.M., Bidarkota, P.V.: A state space framework for the residual income valuation model of stock prices. *SN Business & Economics* **1**(4), 61 (2021)
- [3] Ballings, M., Van den Poel, D., Hespeels, N., Gryp, R.: Evaluating multiple classifiers for stock price direction prediction. *Expert systems with Applications* **42**(20), 7046–7056 (2015)
- [4] Bhanja, S., Das, A.: Impact of data normalization on deep neural network for time series forecasting. *arXiv preprint arXiv:1812.05519* (2018)
- [5] Bhowmik, R., Wang, S.: Stock market volatility and return analysis: A systematic literature review. *Entropy* **22**(5), 522 (2020)
- [6] Bollerslev, T.: Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics* **31**(3), 307–327 (1986)
- [7] Brownlee, J.: Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python. *Machine Learning Mastery* (2018)
- [8] Cardoso, F.C., Malska, J.A.V., Ramiro, P.J., Lucca, G., Borges, E.N., de Mattos, V.L.D., Berri, R.A.: Bovdb: a data set of stock prices of all companies in b3 from 1995 to 2020. *Journal of Information and Data Management* **13**(1) (2022)
- [9] Chandra, R., Chand, S.: Evaluation of co-evolutionary neural network architectures for time series prediction with mobile application in finance. *Applied Soft Computing* **49**, 462–473 (2016)
- [10] Chollet, F.: Deep learning with Python. Simon and Schuster (2021)
- [11] Cox, M., Ellsworth, D.: Application-controlled demand paging for out-of-core visualization. In: *Proceedings. Visualization'97* (Cat. No. 97CB36155). pp. 235–244. IEEE (1997)

- [12] De Oliveira, F.A., Nobre, C.N., Zárate, L.E.: Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index—case study of petr4, petrobras, brazil. *Expert systems with applications* **40**(18), 7596–7606 (2013)
- [13] Dietrich, D., Heller, B., Yang, B., et al.: *Data science & big data analytics: discovering, analyzing, visualizing and presenting data*. Wiley (2015)
- [14] Ersan, D., Nishioka, C., Scherp, A.: Comparison of machine learning methods for financial time series forecasting at the examples of over 10 years of daily and hourly data of dax 30 and s&p 500. *Journal of Computational Social Science* **3**(1), 103–133 (2020)
- [15] de Freitas Parreiras, L.P.R., da Silva, R.A., Sinhorino, S., Ouki, P.Y., Beraha, A., da Silveira, A.K.G., de Moraes, D., Granja, S., Alves, A.F., Kamakura, A., et al.: *Arbitragem estatística e inteligência artificial*. Ph.D. thesis, Tese m. sc, Universidade de São Paulo, São Paulo (2007)
- [16] Fu, T.c., Chung, F.I., Luk, R., Ng, C.m.: Preventing meaningless stock time series pattern discovery by changing perceptually important point detection. In: *International Conference on Fuzzy Systems and Knowledge Discovery*. pp. 1171–1174. Springer (2005)
- [17] Goodfellow, I., Bengio, Y., Courville, A.: *Deep learning*. MIT press (2016)
- [18] Hiransha, M., Gopalakrishnan, E.A., Menon, V.K., Soman, K.: Nse stock market prediction using deep-learning models. *Procedia computer science* **132**, 1351–1362 (2018)
- [19] Huang, C.J., Yang, D.X., Chuang, Y.T.: Application of wrapper approach and composite classifier to the stock trend prediction. *Expert Systems with Applications* **34**(4), 2870–2878 (2008)
- [20] Hyndman, R.J., Athanasopoulos, G.: *Forecasting: principles and practice*. OTexts (2018)
- [21] Jiang, W.: Applications of deep learning in stock market prediction: recent progress. *Expert Systems with Applications* **184**, 115537 (2021)
- [22] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *nature* **521**(7553), 436–444 (2015)
- [23] Li, Y., Ma, W.: Applications of artificial neural networks in financial economics: a survey. In: *2010 International symposium on computational intelligence and design*. vol. 1, pp. 211–214. IEEE (2010)

- [24] Lu-Andrews, R., Glascock, J.L.: Liquidity, price behavior, and market-related events. *Eastern Economic Journal* **43**, 318–351 (2017)
- [25] Moroney, L.: *AI and Machine Learning for coders*. O'Reilly Media (2020)
- [26] Murphy, K.P.: *Machine learning: a probabilistic perspective*. MIT press (2012)
- [27] Nayak, S., Misra, B.B., Behera, H.S.: Impact of data normalization on stock index forecasting. *International Journal of Computer Information Systems and Industrial Management Applications* **6**, 13–13 (2014)
- [28] Nikfarjam, A., Emadzadeh, E., Muthaiyah, S.: Text mining approaches for stock market prediction. In: 2010 The 2nd international conference on computer and automation engineering (ICCAE). vol. 4, pp. 256–260. IEEE (2010)
- [29] O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al.: Keras-tuner. <https://github.com/keras-team/keras-tuner> (2019)
- [30] Ou, P., Wang, H., et al.: Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science* **3**(12), 28–42 (2009)
- [31] Pawar, K., Jalem, R.S., Tiwari, V.: Stock market price prediction using lstm rnn. In: *Emerging Trends in Expert Applications and Security: Proceedings of ICETEAS 2018*. pp. 493–503. Springer (2019)
- [32] Quaye, I., Mu, Y., Abudu, B., Agyare, R., et al.: Review of stock markets' reaction to new events: Evidence from brexit. *Journal of financial risk management* **5**(04), 281 (2016)
- [33] Selvin, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V.K., Soman, K.: Stock price prediction using lstm, rnn and cnn-sliding window model. In: 2017 international conference on advances in computing, communications and informatics (icacci). pp. 1643–1647. IEEE (2017)
- [34] 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* **90**, 106181 (2020)
- [35] Shah, D., Isah, H., Zulkernine, F.: Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies* **7**(2), 26 (2019)
- [36] Shan, F., He, X., Armaghani, D.J., Sheng, D.: Effects of data smoothing and recurrent neural network (rnn) algorithms for real-time forecasting of tunnel boring machine (tbm) performance. *Journal of Rock Mechanics and Geotechnical Engineering* **16**(5), 1538–1551 (2024)

- [37] Sirignano, J., Cont, R.: Universal features of price formation in financial markets: perspectives from deep learning. In: *Machine learning and AI in finance*, pp. 5–15. Routledge (2021)
- [38] Varadharajan, V., Smith, N., Kalla, D., Samaah, F., Polimetla, K., Kumar, G.R.: Stock closing price and trend prediction with lstm-rnn. *Journal of Artificial Intelligence and Big Data* **4**, 877 (2024)
- [39] Wu, H., Zhang, W., Shen, W., Wang, J.: Hybrid deep sequential modeling for social text-driven stock prediction. In: *Proceedings of the 27th ACM international conference on information and knowledge management*. pp. 1627–1630 (2018)
- [40] Yu, L., Wang, S., Lai, K.K.: *Foreign-exchange-rate forecasting with artificial neural networks*, vol. 107. Springer Science & Business Media (2007)
- [41] Zhang, L., Aggarwal, C., Qi, G.J.: Stock price prediction via discovering multi-frequency trading patterns. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. pp. 2141–2149 (2017)
- [42] Zhang, R.: Lstm-based stock prediction modeling and analysis. In: *2022 7th International Conference on Financial Innovation and Economic Development (ICFIED 2022)*. pp. 2537–2542. Atlantis Press (2022)
- [43] Zhang, Y., Wu, L.: Stock market prediction of s&p 500 via combination of improved bco approach and bp neural network. *Expert systems with applications* **36**(5), 8849–8854 (2009)