



**PREDIÇÃO VIA LSTM DE SÉRIES
TEMPORAIS MULTIVARIÁVEIS COM
APLICAÇÃO EM PORTFÓLIO DE
AÇÕES**

FÁBIO OLIVEIRA GUIMARÃES

**TRABALHO DE CONCLUSÃO DE CURSO EM ENGENHARIA DE
COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO**

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO**

**PREDIÇÃO VIA LSTM DE SÉRIES
TEMPORAIS MULTIVARIÁVEIS COM
APLICAÇÃO EM PORTFÓLIO DE
AÇÕES**

FÁBIO OLIVEIRA GUIMARÃES

Orientador: PROF. DR. JOSÉ ALFREDO RUIZ VARGAS

**TRABALHO DE CONCLUSÃO DE CURSO EM ENGENHARIA DE
COMPUTAÇÃO**

**PUBLICAÇÃO -
BRASÍLIA-DF, 18 DE FEVEREIRO DE 2023.**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO**

**PREDIÇÃO VIA LSTM DE SÉRIES
TEMPORAIS MULTIVARIÁVEIS COM
APLICAÇÃO EM PORTFÓLIO DE
AÇÕES**

FÁBIO OLIVEIRA GUIMARÃES

TRABALHO DE CONCLUSÃO DE CURSO ACADÊMICO SUBMETIDO AO DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO DE COMPUTAÇÃO.

APROVADA POR:

Prof. Dr. José Alfredo Ruiz Vargas
Orientador - ENE/UnB

Prof. Dr. Max Eduardo Vizcarra Melgar
Membro - UFC

Prof. Dr. Kevin Herman Muraro Gularte
Membro - UnB

BRASÍLIA, 18 DE FEVEREIRO DE 2023.

FICHA CATALOGRÁFICA

FÁBIO OLIVEIRA GUIMARÃES

Predição Via LSTM de Séries Temporais Multivariáveis Com Aplicação em Portfólio de Ações

2023xv, 25p., 201x297 mm

(ENE/FT-UnB/FT/UnB, Engenheiro de Computação, Engenharia de Computação, 2023)

Trabalho de Conclusão de Curso - Universidade de Brasília

Faculdade de Tecnologia - Departamento de Engenharia de Computação

REFERÊNCIA BIBLIOGRÁFICA

GUIMARÃES, F. O. Predição Via LSTM de Séries Temporais Multivariáveis Com Aplicação em Portfólio de Ações. Trabalho de Conclusão de Curso em Engenharia de Computação, Publicação, Departamento de Engenharia de Computação, Universidade de Brasília, Brasília, DF, 2023. 25p.

CESSÃO DE DIREITOS

AUTOR: Fábio Oliveira Guimarães

TÍTULO: Predição Via LSTM de Séries Temporais Multivariáveis Com Aplicação em Portfólio de Ações.

GRAU: Engenheiro de Computação ANO: 2023

É concedida à Universidade de Brasília permissão para reproduzir cópias deste trabalho de conclusão de curso e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor se reserva a outros direitos de publicação e nenhuma parte deste trabalho de conclusão de curso pode ser reproduzida sem a autorização por escrito do autor.

Fábio Oliveira Guimarães

UnB - Asa Norte, Brasília - DF, 70910-900

Agradecimentos

Agradeço à minha mãe Eliana, ao meu pai Arquimedes, aos meus irmãos Felype e Fabrício e em especial a companheira de jornada Adryelle que sempre estiveram ao meu lado.

Agradeço ao meu orientador, professor Dr. José Alfredo Ruiz Vargas, pelos seus ensinamentos que vão além do conteúdo acadêmico.

Muito Obrigado!

Resumo

Neste trabalho propõe-se a predição das cotações de ações do mercado financeiro brasileiro via séries temporais a partir do modelo LSTM, avaliando a sua eficácia e desempenho de processamento na utilização do paradigma *offline*. O Contexto do trabalho é a bolsa de valores brasileira, mais precisamente os ativos que compõe o índice Ibovespa, principal indicador da Bolsa de Valores (BOVESPA). A relevância do trabalho está na pouca literatura existente referente ao mercado acionário brasileiro e por conta do aumento de investidores brasileiros no mercado de renda variável. Por isso, a construção de técnica de negociação de ativos (ações) atrelada a um modelo de *Deep Learning* ajudará a dar uma maior segurança e, porque não, um maior retorno aos investidores com pouco ou mesmo baixa experiência no investimento de ativos de alto risco.

Palavras-chave - Aprendizado profundo, Séries Temporais, Bolsa de Valores, Predição, Estratégia de Compra e Venda.

Abstract

This work proposes the prediction of stock prices in the Brazilian financial market via time series from the LSTM model, evaluating its efficiency and processing performance when using the offline paradigm. The context of the work is the Brazilian stock exchange, more precisely the assets that make up the Ibovespa index, the main indicator of the Stock Exchange (BOVESPA). The relevance of the work is in the little existing literature referring to the Brazilian stock market and due to the increase of Brazilian investors in the variable income market. Therefore, the construction of an asset (stock) trading technique linked to a Deep Learning model will help to provide greater security and, why not, a greater return to investors with little or even low experience in investing in high-risk assets.

Keywords - Deep Learning, Time Series, Stock Exchange, Prediction, Buying and Selling Strategy.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	REVISÃO DA LITERATURA.....	1
1.2	ESTADO DA ARTE.....	2
1.3	OBJETIVOS.....	2
1.4	ORGANIZAÇÃO DO TRABALHO.....	3
2	DEFINIÇÕES E CONCEITOS PRELIMINARES.....	4
2.1	MÉTRICAS DE AVALIAÇÃO	4
2.2	MODELO LSTM	5
3	DESENVOLVIMENTO DOS MODELOS ANALÍTICOS.....	8
3.1	INTRODUÇÃO	8
3.2	AQUISIÇÃO DOS DADOS	8
3.3	PREPARAÇÃO DE DADOS.....	9
3.4	TREINAMENTO DO MODELO LSTM	9
3.5	CONSTRUÇÃO DA ESTRATÉGIA DE COMPRA E VENDA DE ATIVOS	10
3.6	ANÁLISE DE PERFORMANCE	11
4	RESULTADOS E DISCUSSÕES	13
5	CONCLUSÕES.....	17
A	CÓDIGOS DE IMPLEMENTAÇÃO EM PYTHON.....	19

LISTA DE FIGURAS

1.1	Linha do Tempo de aplicação das técnicas de <i>Machine Learning</i> aplicados a finanças. Fonte: (RUNDO et al., 2019)	2
2.1	Módulo de Repetição da rede LSTM contendo as quatro camadas (HOCHREITER; SCHMIDHUBER, 1997)	5
2.2	Equações de uma célula da Arquitetura LSTM (KEXUGIT, 2022)	6
3.1	Parâmetros de treinamento da LSTM.....	10
3.2	Camadas do modelo LSTM.....	10
3.3	Fluxo do Sistema de Compra de Ativos.....	11
3.4	Registro de log do funcionamento do sistema de Compra e Venda de ativos	12
3.5	Painel resumo das transações realizadas para o ativo ITUB4.SA.....	12
4.1	Valores Reais, Predições em Treino e Predições em Teste do papel VALE3.....	13
4.2	Valores Reais, Predições em Treino e Predições em Teste do papel PETR4.....	14
4.3	Valores Reais, Predições em Treino e Predições em Teste do papel ITUB4	14
4.4	Valores Reais, Predições em Treino e Predições em Teste do papel BBDC4.....	15
4.5	Valores Reais, Predições em Treino e Predições em Teste do papel PETR3.....	15

LISTA DE SÍMBOLOS

B3 - Brasil, Bolsa, Balcão

CDB - Certificado de Depósito Bancário

Ibovespa - Índice Bovespa

LSTM - Memória de Curto Prazo Longa

RF - Retorno de Referência

RMSE - Raiz Quadrada do Erro Médio

RMSP - Raiz Quadrada da Propagação Médio

Vol - Volatilidade

Capítulo 1

Introdução

1.1 Revisão da Literatura

Aprendizado de Máquina ou *Machine Learning* (ML) é um dos campos de estudo da grande área do conhecimento denominada Inteligência Artificial (IA). De acordo com (RUSSELL; NORVIG, 2020) IA é o desenvolvimento de algoritmos ou paradigmas utilizando máquinas para soluções de tarefas cognitivas.

A IA possui diversas áreas de estudos como: aprendizado de máquina, processamento de linguagem natural (PLN), sistemas especialistas, robótica, visão computacional, reconhecimento de fala e planejamento.

O foco desse trabalho será apresentar o que é aprendizagem de máquina e como esse ramo do conhecimento pode ser utilizado em finanças.

Dentre as variadas definições de ML pode-se citar que Aprendizado de Máquina é a utilização de algoritmos que possibilitem extrair padrões ou conhecimentos de um grande volume de dados. A partir dessa definição recaímos em outro problema que é a definição de aprendizado, como ensinar uma máquina a aprender.

O processo de aprendizagem ML é dividido em 3 abordagens: Supervisionado, Não-Supervisionada e Aprendizagem por Reforço como apresentado por (MACHADO; DE ASSIS; PEREIRA, 2020) . Na aprendizagem Supervisionada o algoritmo usa informações prévias fornecidas pelo cientista para ensinar.

Na abordagem Não-Supervisionada o algoritmo utiliza técnicas matemáticas, como a teoria da informação, para identificar dentre as diversas variáveis quais são as mais relevantes e como elas estão relacionadas. Já na abordagem por Reforço o algoritmo bonifica ou penaliza a resposta dependendo do acerto ou erro em relação a resposta, levando o algoritmo a aprender com o seu erro ou acerto.

Todas as três abordagens podem ser utilizadas no mercado financeiro para operar compras e vendas de diversos tipos de ativos como ações, *commodities*, ETFs, índices de bolsas e etc.

Como o intuito é utilizar máquinas através de algoritmos para imitar a transações que seriam realizadas por um humano, mas especificamente um profissional especializado na área.

A utilização de técnicas de *Machine Learning* nas operações de compras e vendas de ativos financeiros não elimina a possibilidade de prejuízos, mas evita que o investidor tenha um retorno inferior aos índices de mercado, como o índice Ibovespa e a Inflação do Período.

1.2 Estado da Arte

Na área de *Machine Learning* aplicado à finanças (RUNDO et al., 2019) apresenta várias técnicas quantitativas nos três tipos de abordagem de ML. Dentre as técnicas citadas no artigo tem-se a ARIMA (séries temporais), Máquina de Vetor de Suporte (SVM), K Vizinhos mais próximos (KNN), entre outras.

O autor cita que a sua estratégia para desenvolvimento de aplicações financeiras segue uma linha do tempo, conforme figura 1.1. Os primeiros algoritmos testados são os Autorregressivos (Séries Temporais), depois passam para algoritmos de Machine Learning (SVM, KNN), posteriormente para algoritmos de Deep Learning, depois para a fusão de técnicas, denominadas de Modelos híbridos e por fim os modelos de aprendizado por reforço.

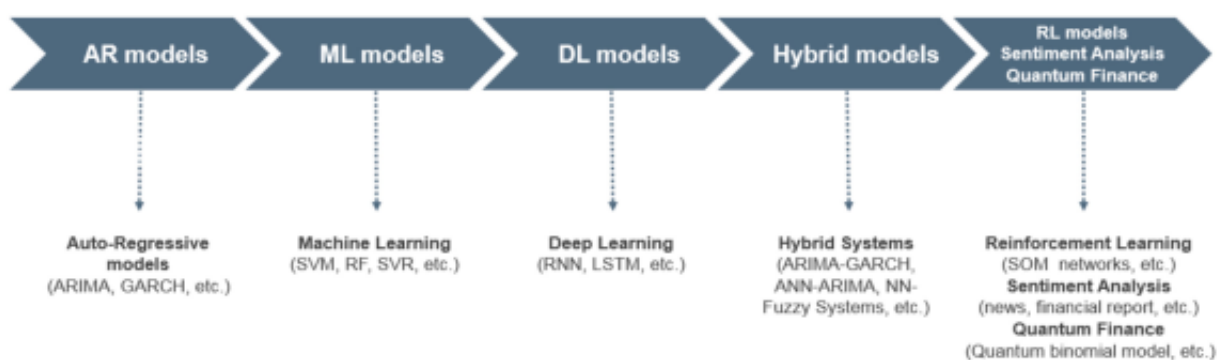


Figura 1.1: Linha do Tempo de aplicação das técnicas de *Machine Learning* aplicados a finanças. Fonte: (RUNDO et al., 2019)

Entre o levantamento de estudos sobre o tema (RUNDO et al., 2019) levantou a tabela 1.1 com a presença de diversos artigos com a temática de *Machine Learning*.

1.3 Objetivos

Este trabalho tem como objetivo propor uma suíte de modelos preditivos supervisionados para ajudar os investidores a realizarem compras e vendas de ações no mercado financeiro

Article	Techniques	Main Goal	Dataset
Patel et al. (2014) [14]	Fusion approach involving ANN, RF and SVR models	Stock prices prediction	Indian Stock market indices
Chen et al. (2015) [15]	LSTM model	Stock returns prediction	China Stock Market data
Hafezi et al. (2015) [16]	bat-neural network multi-agent system (BNNMAS)	Stock price prediction	DAX stock price
Fischer et al. (2017) [17]	LSTM model	Large-scale financial market prediction	S&P 500 (1992–2015)
Yümlü et al. (2005) [21]	Global, recurrent and smoothed-piecewise neural models	Financial time series prediction	Istanbul Stock Exchange (1990–2002)
Selvin et al. (2017) [22]	LSTM-RNN-CNN sliding window-ARIMA	Short term future prediction	NSE listed companies' data (2014–2015)
Samarawickrama et al. (2017) [23]	RNN model	Daily stock prices prediction	Sri Lankan Stock Exchange
Siarni-Namini et al. (2018) [24]	LSTM vs. ARIMA	Comparison	N225, IXIC, HIS, GSPC, DJ
Rundo et al. (2019) [25]	LSTM model	Forecasting	ENEL.MI ISP.MI UCG.MI CVAL.MI MPS.MI
Rundo et al. (2019) [26]	Grid algorithmic trading system (Non-linear Regression Network)	Forecasting	FX market (EUR/USD)

Tabela 1.1: Levantamento de artigos baseados em ML levantados por (RUNDO et al., 2019)

brasileiro, de tal forma que os usuários sem muito conhecimento de ações e modelos preditivos consiga investir.

Sendo assim, o investidor não precisará de suporte de profissionais especializados em investimentos para conquistar a sua liberdade financeira. Para isso, será utilizado o *Visual Studio Code* com a linguagem Python para extrair os dados e treinar os modelos de forma *offline*.

1.4 Organização do trabalho

O trabalho está organizado da seguinte forma:

- Capítulo 1: Encontra-se uma breve revisão da literatura, bem como o estado da arte.
- Capítulo 2: São apresentadas as definições e conceitos necessários para a compreensão deste trabalho.
- Capítulo 3: Desenvolvimento dos modelos preditivos de *Machine Learning* para séries temporais.
- Capítulo 4: São discutidos os resultados encontrados pelo sistema desenvolvido de compra e venda de ações.
- Capítulo 5: Apresentam-se as principais conclusões deste Trabalho de Conclusão de Curso.

Capítulo 2

Definições e conceitos preliminares

Neste capítulo serão apresentados métricas de avaliação de séries temporais (RMSE) e indicadores de investimento como Volatilidade e SHARPE. Também, será apresentado o funcionamento da Rede Neural LSTM.

2.1 Métricas de Avaliação

Para avaliar se o algoritmo conseguirá acompanhar os valores das cotações será utilizado uma métrica bem conhecida em sistemas de regressão que é o RMSE – Root Mean Squared Error que calcula a raiz quadrática média dos erros. Onde, n é a quantidade de amostras, y_j o valor da variável avaliada e \hat{y}_j o valor médio da variável, conforme equação 2.1.

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (2.1)$$

A volatilidade é uma medida de importante quando trata-se de mercado financeiro, e faz parte do rol de indicadores de risco dos ativos financeiros. A volatilidade mede a taxa de variação do ativo da data atual em comparação com a data do dia anterior, sendo P_i o preço ajustado do ativo hoje e P_{i-1} o valor do ativo ajustado de ontem e n é a quantidade de amostras. A equação 2.2 mostra o calculo de r e a equação 2.3 apresenta a volatilidade.

$$r = \ln \frac{P_i}{(P_{i-1})} \quad (2.2)$$

$$Vol = \sqrt{\frac{\sum |r - \bar{r}|^2}{n}} \quad (2.3)$$

O índice de Sharpe é a medida que relaciona o risco com o retorno, ou seja, quanto maior o índice melhor é o retorno do ativo comparado com o RF, retorno financeiro, calculado em

percentual subtraído do índice Ibovespa, tudo dividido pela volatilidade, conforme equação 2.4.

$$\text{Sharpe} = \frac{RF - \text{Ibovespa}}{(\text{Vol})} \quad (2.4)$$

2.2 Modelo LSTM

Após escolher as métricas de sucesso do modelo, falta escolher qual o algoritmo para o artigo, conforme comentado o LSTM foi o escolhido, ele se enquadra nos algoritmo de Redes Neurais recorrentes e a sua tradução seria Redes de Memória de Curto Prazo Longo, tal algoritmo foi proposto por (HOCHREITER; SCHMIDHUBER, 1997).

O LSTM tem o objetivo de evitar o problema de dependências de longo prazo das séries temporais, sua estrutura possui uma cadeia onde cada estrutura possui quatro camadas de repetição, interagindo de forma particular, exemplificado na figura 2.1 (JUNIOR, 2019).

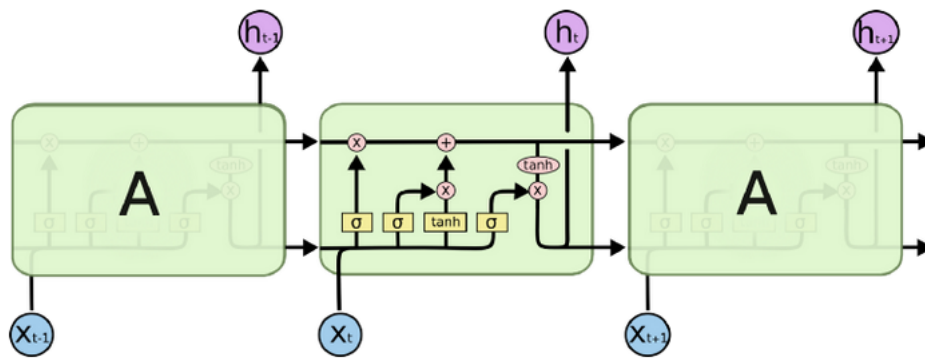


Figura 2.1: Módulo de Repetição da rede LSTM contendo as quatro camadas (HOCHREITER; SCHMIDHUBER, 1997)

Para melhorar o entendimento da arquitetura LSTM temos a figura 2.2, onde a saída explícita é $h(t)$ e onde $h(t-1)$ e $c(t-1)$ são as entradas. O índice $t-1$ significa a etapa de tempo anterior. O c representa o estado da célula.

A equação 2.5 apresenta a porção do LSTM de longo prazo, onde os valores de W_f representam os pesos da rede e n a quantidade de entradas.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2.5)$$

As equações 2.6 e 2.7 referem-se a porção de curto prazo da arquitetura LSTM, onde os valores de W_i e W_o representam os pesos da rede e n a quantidade de entradas.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2.6)$$

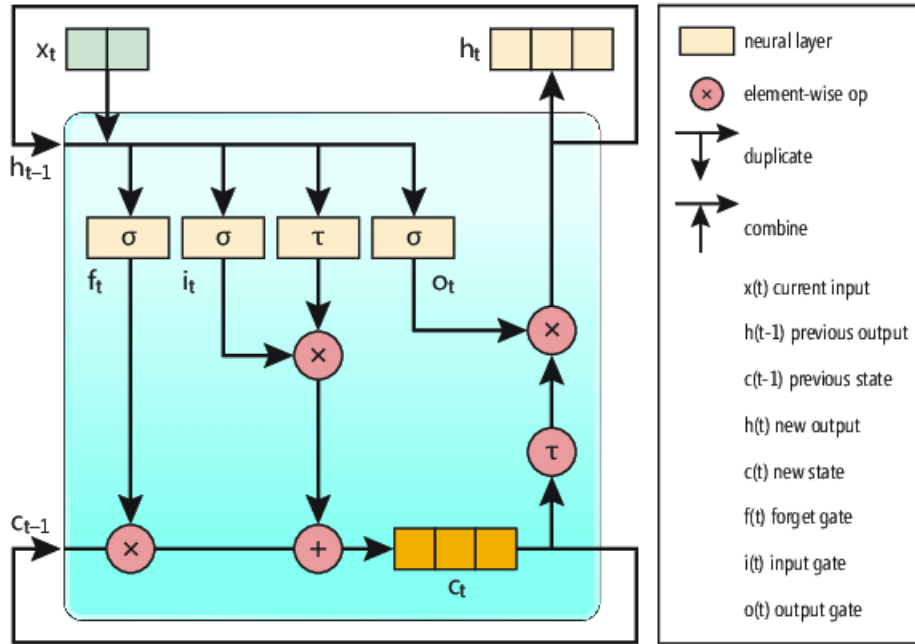


Figura 2.2: Equações de uma célula da Arquitetura LSTM (KEXUGIT, 2022)

$$o_t = \sigma (W_o x_t + U_o h_{t-1} + b_o) \quad (2.7)$$

A equação 2.8 representa a atenuação de curto prazo na rede de longo prazo da rede LSTM.

$$c_t = f_t \circ c_{t-1} + i_t \circ \tau (W_c x_t + U_c h_{t-1} + b_c) \quad (2.8)$$

E a equação 2.9 é a combinação dos pesos totais.

$$h_t = o_t \circ \tau (c_t) \quad (2.9)$$

Variáveis da arquitetura LSTM:

n : Tamanho de entrada;

m : Tamanho do estado da célula de saída;

x_t : Vetor de entrada;

f_t : Fator de esquecimento da porta;

i_t : Vetor da camada de entrada;

O_t : Vetor da camada de saída;

h_t : Vetor de saída;

c_t : Vetor de estado celular;

$W_f, W_i, W_\alpha W_c$: Matrizes de peso da entrada;

U_f, U_i, U_o, U_c : Matrizes de peso da porta de saída;

b_f, b_i, b_o, b_c : vetor de bias;

σ : Função de ativação sigmóide logística;

τ : Função de ativação tangente hiperbólica;

Capítulo 3

Desenvolvimento dos Modelos Analíticos

3.1 Introdução

Esta seção apresenta os passos realizados para construção do sistema de recomendação de compra e venda de ativos financeiros da Bovespa, todas as etapas realizadas foram desenvolvidas na linguagem de programação Python via IDE Visual Studio Code. Os passos realizados, em ordem, foram: aquisição do dados, preparação dos dados, treinamento do modelo LSTM, construção da estratégia de compra e venda de ativos e análise da performance.

3.2 Aquisição dos Dados

Na aquisição dos dados utilizou-se a biblioteca pandas-datareader, entre as opções de fonte de dados para download das cotações foi selecionado a origem Yahoo Finanças. Os dados vieram no formato dataframe contendo as informações: data do pregão, cotação mais alta do dia, cotação mais baixa do dia, cotação de abertura, cotação de fechamento, volume de ativos negociados no dia e valor de fechamento ajustado.

Para o artigo foram selecionados as 5 ações com os maiores percentuais de participação no índice Bovespa, conforme tabela 3.1, além do próprio índice Bovespa que será utilizado como referência de desempenho. Foram utilizadas as cotações de fechamento corrigidas de 01/01/2017 até 30/06/2022.

Código	Ação	Tipo	Part. (%)
VALE3	VALE	ON NM	15,379
PETR4	PETROBRAS	PN N2	6,716
ITUB4	ITAUUNIBANCO	PN EJ N1	6,526
BBDC4	BRADESCO	PN EJ N1	4,994
PETR3	PETROBRAS	ON N2	4,418

Tabela 3.1: As 5 ações com maior percentual de participação índice Bovespa (Bovespa, 2022)

3.3 Preparação de Dados

Na preparação dos dados selecionou-se apenas a coluna de valor de fechamento ajustado e a data de referência. Para centralizar todos os dados, o dataframe com as cotações foram unidos de forma que o nome da coluna com o valor da cotação ajustado foi nomeado com o nome do ativo, na tabela 3.2 têm-se os cinco primeiros registros como amostra.

	VALE3	PETR4	ITUB4	BBDC4	PETR3
Date					
2017-01-02	16.782001	7.383342	17.800673	11.674997	8.719831
2017-01-03	17.525337	7.806397	18.554541	12.169475	9.273719
2017-01-04	17.210592	7.806397	18.634171	12.092397	9.162943
2017-01-05	17.866873	7.932308	18.888994	12.226262	9.368674
2017-01-06	17.391403	7.886983	18.952700	12.303338	9.226244

Tabela 3.2: Cotações de fechamento ajustado dos 5 principais ativos do índice Bovespa

Com os dados escolhidos o próximo passo é normalizar os dados com a função `MinMaxScaler` que normaliza as cotações de fechamento entre 0 e 1, tal procedimento é indispensável para o modelo utilizado o LSTM. Outro passo importante é o reshape do dataset, onde transforma-se o dado, já normalizado, que está em coluna para um array em linha para a termos uma entrada específica para o LSTM.

Uma boa prática em toda criação de modelos de Machine Learning ou Deep Learning é a quebra da base em treinamento e teste, a fim de garantir que o modelo não está com overfitting, no caso desse artigo a quebra utilizada foi 2/3 para treinamento e 1/3 para teste, sendo que os dados foram segmentados em ordem cronológica, ou seja, dos dados capturados no item 2.1 os primeiros 2/3 foram utilizados apenas para o treinamento.

3.4 Treinamento do Modelo LSTM

O modelo escolhido para previsão da Série Temporal foi o LSTM devido ao seu baixo RMSE em comparação com o ARIMA e o RNN, conforme mencionado em (RUNDO et al., 2019). Para o artigo, customizou-se os parâmetros de treinamento com o otimizador Adam que combina dois métodos de descida do gradiente: Momentum e o RMSP.

A função de perda utilizada foi a *Mean Squared Error* e o *Look Back* de 1, ou seja, utilizou-se o janelamento de um 1, sendo assim, a partir do fechamento do dia anterior se prediz o valor do fechamento do papel de hoje, evidenciado na figura 3.1.

Com uma camada LSTM e uma camada densa no modelo, totalizando 101 parâmetros, conforme figura 3.2.

```

# create and fit the LSTM network
model = Sequential()
model.add(LSTM(4, input_shape=(1, look_back)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(trainX, trainY, epochs=100, batch_size=1, verbose=2)

```

Figura 3.1: Parâmetros de treinamento da LSTM

```

Model: "sequential_16"
-----
Layer (type)                Output Shape         Param #
-----
lstm_16 (LSTM)              (None, 4)           96
dense_16 (Dense)            (None, 1)           5
-----
Total params: 101
Trainable params: 101
Non-trainable params: 0

```

Figura 3.2: Camadas do modelo LSTM

3.5 Construção da Estratégia de Compra e Venda de Ativos

A estratégia desenvolvida neste artigo tem com premissa a identificação de viés, tanto de alta quanto de baixa, funcionando com duas abordagens. Para tanto, o sistema de compra e venda de ativos começa com um valor específico de dinheiro em caixa para inicialização das comercializações, nas simulações foi escolhido o valor de R\$ 1.000,00 por ativo.

A primeira coisa que o sistema faz é identificar o valor de fechamento do ativo no dia 0 e no primeiro dia, comprar metade do caixa em ativos, deixando um valor aproximado de R\$ 500,00 para futuras compras, o valor é aproximado pois depende do valor do ativo de cada empresa.

Em seguida, quando o algoritmo identificar uma previsão de alta para o dia, maior que 0,3%, em relação ao fechamento do ativo no dia anterior o sistema irá comprar o ativo na abertura do pregão para a posteriormente vender pelo preço previsto do ativo até o fechamento do dia, caso a ação não seja comercializada devido ao comportamento de baixa do ativo para o dia, o sistema identificará que há um papel que não teve a sua venda realizada, informando que há uma ordem de venda em aberto, conforme apresentado na figura 3.4.

Agora, caso o algoritmo identifique uma previsão de baixa para o dia, baixa maior que 0,3%, em relação ao fechamento do ativo no dia anterior o sistema irá vender um dos seus ativos em estoque, ou os comprados no dia 0 ou os que tiveram ordem de venda em aberto, e posteriormente irá recomprar o ativo ao valor previsto pelo algoritmo até o final do dia, caso isso não ocorra devido uma alta do ativo, o sistema identificará que temos um ordem de compra em aberto para futuro acerto.

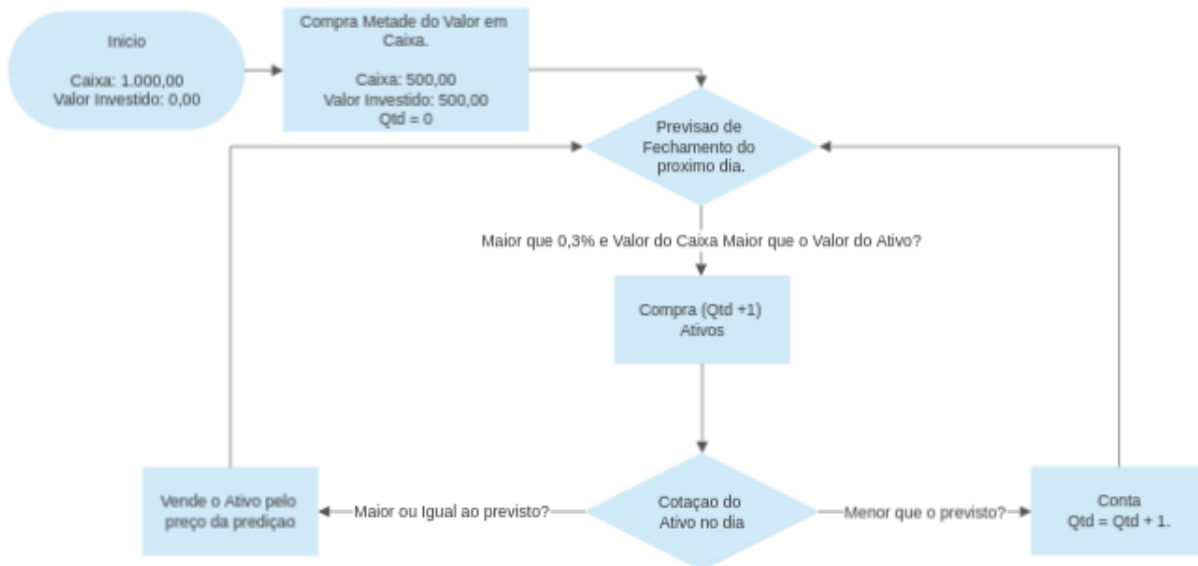


Figura 3.3: Fluxo do Sistema de Compra de Ativos

O valor de referência de 0,3% utilizado nas operações de compra e venda dos ativos foi parametrizada avaliando o percentual médio de variação do fechamento dos ativos no período avaliado, que foi de 0,12%. Sendo assim, para que o sistema atue-se na compra/venda a previsão de fechamento deverá ser superior a mais que o dobro da variação média de fechamento do ativo.

Com isso, o algoritmo tentará ganhar dinheiro tanto na alta do ativo quanto na baixa, caso a sua previsão não se concretize o sistema identifica que há ordens pendentes e acumula para quando o cenário estiver favorável. Idealmente, se houver sequências de alta e de baixa o sistema conseguirá se manter em harmonia, sempre preservando o valor de metade do dinheiro em caixa para próximas compras e metade do valor investido para próximas vendas.

Como não pode-se contar com a sorte o sistema tem limitadores programados, onde não é possível vender ativos que não se tenha e não poderá realizar compra de ativos se o valor ultrapassar o dinheiro em caixa, para não entrar em estagnação, caso não seja possível comprar ou vender ativos por algum tipo de limitação o sistema tentará ou comprar menos papéis até o limite do dinheiro em caixa ou vender os ativos até o limite que se possua.

Para facilitar o acompanhamento das transações, foi criado um sistema de log que informa qual tipo de transação foi realizada e se houve algum atingimento de limite, conforme apresentado na figura 3.4.

3.6 Análise de Performance

Ao final de cada dia o sistema calcula os parâmetros básicos de quanto se lucrou nas transações de Day Trade, quantos dias em operação o sistema está executando, o valor de dinheiro em

```

Compras Pendentes: 1
Tentou comprar, mas ninguém vendeu pelo preço solicitado
Dia com Projeção de Subida: 1
Projeção para Hoje: 23.0413
Cotação no Fechamento do dia Anterior: 22.895586013793945
Cotação no Fechamento de Hoje: 23.45171546936035
Ganho Real: 0.2914276123046875
Percentual: 0.6364275020720382
Caixa: 496.2971076965332
Investido: 503.7028923034668
Quantidade de Ativos: 22
Compras Pendentes: 1
Tentou comprar, mas ninguém vendeu pelo preço solicitado
Compras Pendentes: 2
Tentou comprar, mas ninguém vendeu pelo preço solicitado
Compras Pendentes: 3
Tentou comprar, mas ninguém vendeu pelo preço solicitado
Compras Pendentes: 4
Tentou comprar, mas ninguém vendeu pelo preço solicitado
Compras Pendentes: 5

```

Figura 3.4: Registro de log do funcionamento do sistema de Compra e Venda de ativos

caixa, o somatório de quanto foi pago em todas as compras já realizadas dos ativos em estoque, a quantidade de ativos em estoque, o valor atual do estoque considerando o preço atual, o valor total somando caixa mais ativos, o ganho/perda desde o início da operação, o rendimento total, o comparativo do índice Bovespa (B3, 2022) para o período e a valorização da ação desde o início da operação (modelo buy and hold). Esses dados, mostrado na figura 3.5, apresenta um direcionador para o investidor.

```

+++++
Ganho nas Compras e Vendas de Ativos: 88.52847862243652
Rendimento no Day Trade: 386.6617721385275
Dias de Operação: 449
Dinheiro em Caixa: 496.2971076965332
Valor Investido: 503.7028923034668
Quantidade de Ativos: 22
Valor dos Ativos: 500.33874130249023
Valor Total (Caixa + Ativos): 996.6358489990234
Ganho Total (Caixa + Ativos): -3.3641510009765625
Rendimento Total (Caixa + Ativos): -0.33641510009765624
Rendimento Ibov -2.666877382904328
Rendimento da Ação -0.6678839951845574

```

Figura 3.5: Painel resumo das transações realizadas para o ativo ITUB4.SA

Capítulo 4

Resultados e Discussões

Agora, serão apresentados os resultados do sistema de compra e vendas de ativos das 5 ações selecionadas (VALE3, PETR4, ITUB4, BBDC4 e PETR3). Após o treinamento dos modelos LSTM observa-se que para o papel VALE3, figura 4.1, é possível notar que nos primeiros 3 meses a predição é bem próxima do valor real, mas após os meses iniciais a predição não consegue acompanhar a curva, apenas as tendências de altas e baixas.



Figura 4.1: Valores Reais, Predições em Treino e Predições em Teste do papel VALE3

Nos modelos treinado para a PETR4, figura 4.2, e PETR3 na figura 4.5 temos um desempenho superior ao observado no ativo VALE3, onde os modelos acompanha as curvas reais por quase um ano. Nos modelo treinados do ativo ITUB4, figura 4.3 e no ativo BBDC4 na figura 4.4 observa-se um desempenho melhor que os 3 anteriores, acompanhando muito bem a variação dos ativos por quase 2 anos.

Em todas os 5 ativos a performance do indicador Sharpe foi positiva apresentando um desempenho do sistema de compra e venda de ativos superior ao índice Bovespa.

Deve-se levar em consideração os custos associados as compras e vendas de ações, como

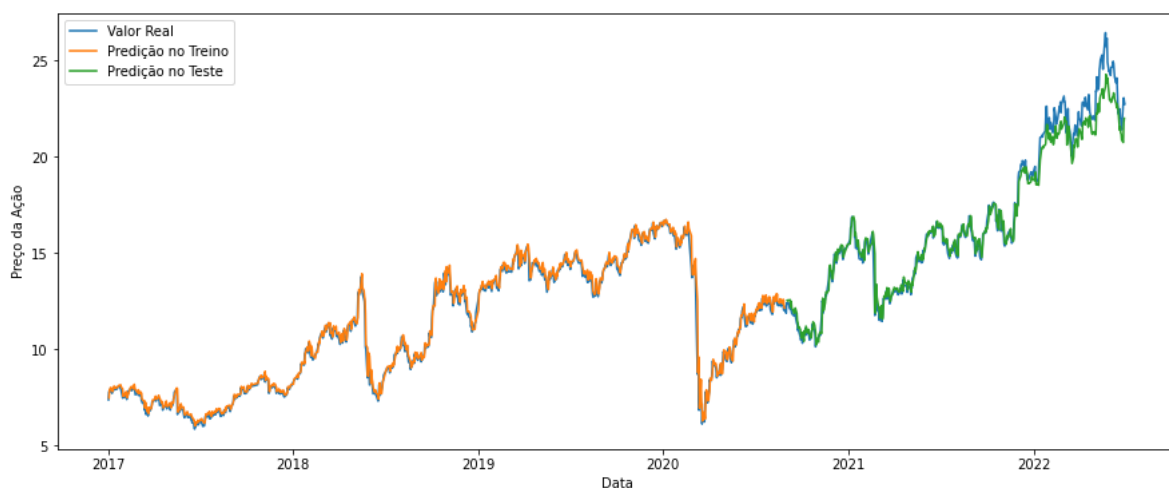


Figura 4.2: Valores Reais, Predições em Treino e Predições em Teste do papel PETR4

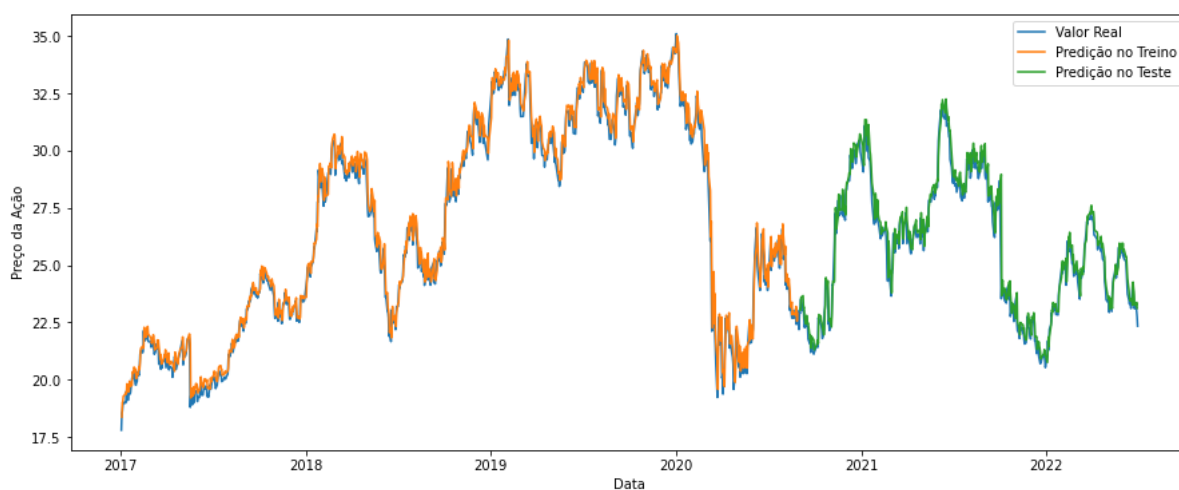


Figura 4.3: Valores Reais, Predições em Treino e Predições em Teste do papel ITUB4

a taxa de corretagem e o Imposto de Renda. Primeiramente, diversas corretoras deixaram de cobrar a taxa de corretagem, entre elas pode-se citar: Rico, Toro, Órama, C6 Bank, Clear, CM Capital, Genial, Inter, NuInvest, Empiricus e Warren, ou seja, a taxa de corretagem deixou de ser um custo nas transações.

Com relação a cobrança de Imposto de Renda sobre as transações de *Day Trade* a Receita Federal cobra uma taxa fixa de 20% sobre o lucro da transação, sendo assim, deve-se considerar este custo no final das transações.

Analisando os dados da tabela 4.1 é possível verificar que apenas no ativo ITUB4 no sistema de compras e vendas teve prejuízo (0,34%) e os outros 4 ativos obteve-se lucro, com destaque para o ativo PETR3 com 64,7% de valorização. Analisado o comparativo do índice Bovespa, apenas o ITUB4 teve desempenho inferior e comparando com um índice de renda fixa, o CDB, novamente só o ITUB4 teve rendimento inferior.

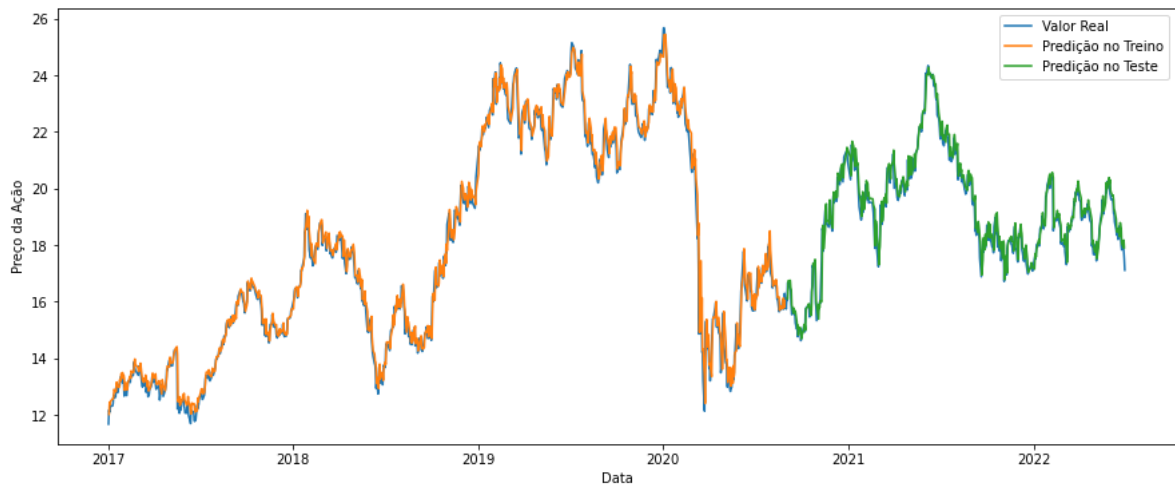


Figura 4.4: Valores Reais, Predições em Treino e Predições em Teste do papel BBDC4

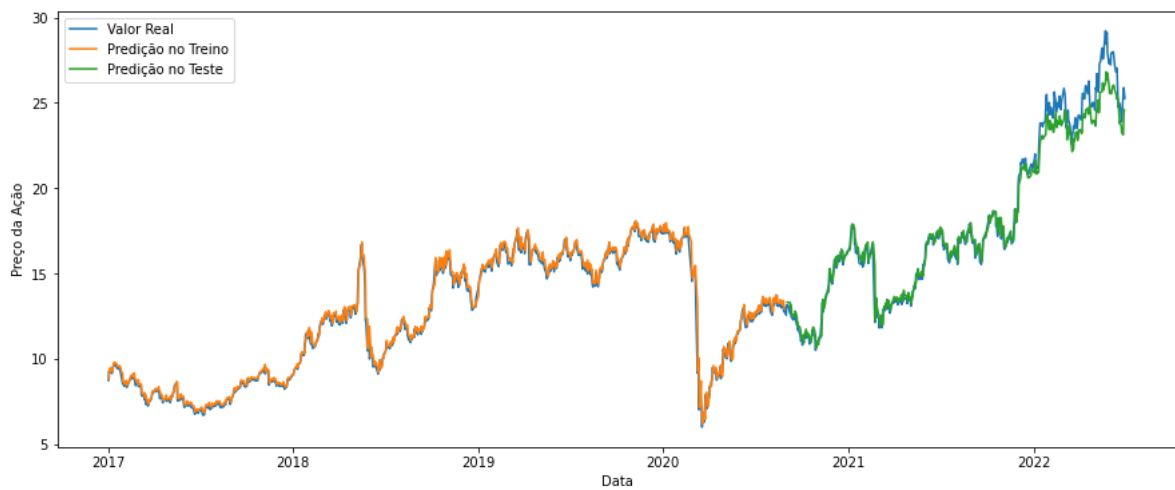


Figura 4.5: Valores Reais, Predições em Treino e Predições em Teste do papel PETR3

	VALE3	PETR4	ITUB4	BBDC4	PETR3
RMSE % (Treino)	7,45	0,68	0,58	0,40	0,79
Sharpe %	21,78	22,74	1,11	3,62	25,58
Ganho nas transações R\$	91,56	137,47	87,34	38,84	167,77
Dias de Operação	449	449	449	449	449
Valor Investido	1000	1000	1000	1000	1000
Dinheiro em Caixa	635,69	655,57	496,30	504,09	677,27
Quantidade de Ativos	11,00	40,00	22,00	31,00	38,00
Valor dos Ativos	853,98	914,61	500,34	542,86	970,68
Valor Total (Caixa + Ativos)	1459,67	1570,18	996,63	1046,96	1647,95
Ganho Total	459,67	570,18	-3,37	46,96	647,95
Ganho Total Líquido	367,74	456,14	-2,70	37,57	518,36
Ganho CDB no período	231,10	231,10	231,10	231,10	231,10
Rendimento Bovespa %	-2,66	-2,66	-2,66	-2,66	-2,66
Valorização do Ativo %	63,52	84,72	-0,67	9,14	94,86

Tabela 4.1: Indicadores de performance dos 5 ativos

Capítulo 5

Conclusões

Neste trabalho utilizou-se um Modelo LSTM para previsão do valor de fechamento da ação do dia posterior ao capturado pelo algoritmo, em seguida o valor predito foi utilizado no sistema de compra e venda para indicação da operação a ser realizada.

Assim, pode-se concluir que o sistema automatizado de compra e venda de ativos performou melhor que o índice Bovespa do período, o que pode ser visualizado pelo indicador Sharpe. O modelo LSTM conseguiu manter um bom acompanhamento da curva real após o treinamento, entretanto percebe-se um descolamento da curva em alguns ativos após um certo tempo de uso, o que demanda um retreino em um período menor que 6 meses.

Outro achado interessante é que mesmo em um período conturbado economicamente (pandemia da COVID-19, retomada da atividade econômica pós-pandemia e guerra na Ucrânia) houve um rendimento positivo na somatório dos 5 ativos.

Um sistema que integrasse as três abordagens (curto prazo – *Day Trade*, médio prazo – Valorização do Ativo e longo prazo – Fundamentalista) traz uma diversidade maior de transações e uma possibilidade de diminuição do risco e volatilidade do ativo com um ganho a mais de dividendos e juros sobre o capital próprio.

Pensando em trabalhos futuros o sistema poderia ser evoluído para escolher o papel que mais tivesse variação de alta ou baixa para realizar a transação, de forma a ter uma carteira de ativos mais ampla melhorando os ganhos totais. Além disso, a construção de mais dois módulos de ativos seria interessante, um módulo que escolhesse os melhores ativos a longo prazo, considerando indicadores fundamentalista, a fim de utilização da técnica *buy and hold* para longo prazo e um módulo de comercialização de ativos usando a técnica de ativo valorizado e ativo barato a fim de usar uma abordagem de médio prazo.

Referências

B3. *Índice IBOVESPA - Composição da Carteira: B3*. [S.l.]: B3, jun. 2022. Disponível em: <https://www.b3.com.br/pt_br/market-data-e-indices/indices/indices-amplos/indice-ibovespa-ibovespa-composicao-da-carteira.htm>.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. eng. *Neural computation*, MIT Press, One Rogers Street, Cambridge, MA 02142-1209, USA, v. 9, n. 8, p. 1735–1780, 1997. ISSN 0899-7667.

JUNIOR, J. R. F. *Redes Neurais recorrentes-LSTM*. [S.l.]: Medium, jun. 2019. Disponível em: <<https://medium.com/@web2ajax/redes-neurais-recorrentes-lstm-b90b720dc3f6>>.

KEXUGIT. *Execução de Teste - Reconhecimento de Células LSTM USANDO C*. [S.l.: s.n.], jun. 2022. Disponível em: <<https://learn.microsoft.com/pt-br/archive/msdn-magazine/2018/april/test-run-understanding-lstm-cells-using-csharp>>.

MACHADO, E. J.; DE ASSIS, C. A. S.; PEREIRA, A. C. M. Modelagem, Implementação e Avaliação de Estratégias de Negociação Baseadas em Algoritmos de Aprendizado de Máquina para o Mercado Financeiro. eng. *Revista Brasileira de Computação Aplicada.*, Universidade de Passo Fundo (UPF), v. 12, n. 1, p. 16–31, 2020. ISSN 2176-6649.

RUNDO, F. et al. Machine Learning for Quantitative Finance Applications: A Survey. eng. *Applied sciences*, MDPI AG, Basel, v. 9, n. 24, p. 5574, 2019. ISSN 2076-3417.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach (4th Edition)*. [S.l.]: Pearson, 2020. ISBN 9780134610993. Disponível em: <<http://aima.cs.berkeley.edu/>>.

Apêndice A

Códigos de Implementação em Python

Nesta seção encontram-se os códigos em Python utilizados neste trabalho.

A.0.1 Códigos de Preparação, Modelagem e Predição dos Dados

Listagem A.1: Código em Python

```
1 # Importando as Bibliotecas
2 import pandas as pd
3 from pandas_datareader import data as web
4 import matplotlib.pyplot as plt
5
6 # %%
7 data_inicial = "01/01/2017"
8 data_final = "06/30/2022"
9
10 # Carrega o excel com as cota es
11 arquivo = pd.ExcelFile("Carteira.xlsx")
12
13 # 'extrai' aba Maiores
14 empresas_df = pd.read_excel(arquivo, sheet_name='Maiores')
15 empresas_df = empresas_df[['Codigo']]
16 #display(empresas_df)
17
18 #empresas_df = pd.DataFrame({"Codigo": ['VALE3', 'PETR4']})
19
20 cotacoes = pd.DataFrame()
21 ibov_cotacoes = pd.DataFrame()
22 print("Baixando os Pap is:")
23 for empresa in empresas_df['Codigo']:
24     print(f"{empresa}")
25
26 # Baixando os pap is
27 df = web.DataReader(f'{empresa}.SA', data_source='yahoo', start=
    data_inicial, end=data_final)[['Adj Close']]
28
```

```

29     # Ajustando o dataframe
30     df = df.reset_index()
31     cotacoes["Date"] = df['Date']
32     cotacoes[f"{empresa}"] = df['Adj Close']
33
34     #display(df)
35     #df["Adj Close"].plot(figsize=(15, 10))
36     #plt.show()
37
38     print("^BVSP")
39     ibov = web.DataReader('^BVSP', data_source='yahoo', start=data_inicial, end
40         =data_final)[['Adj Close']]
41     ibov = ibov.reset_index()
42     ibov_cotacoes["Date"] = ibov['Date']
43     ibov_cotacoes["BVSP"] = ibov['Adj Close']
44     ibov_cotacoes.set_index('Date', inplace = True)
45
46     print("Download Completo!")
47
48     # corrigindo o index o dataframe
49     cotacoes.set_index('Date', inplace = True)
50
51     # %%
52     # Rodando LSTM
53
54     import numpy as np
55     import matplotlib.pyplot as plt
56     import pandas as pd
57     import tensorflow as tf
58     from tensorflow.keras.models import Sequential
59     from tensorflow.keras.layers import Dense
60     from tensorflow.keras.layers import LSTM
61     from sklearn.preprocessing import MinMaxScaler
62     from sklearn.metrics import mean_squared_error
63
64     # %%
65     # fix random seed for reproducibility
66     tf.random.set_seed(7)
67
68     dataset = cotacoes['PETR3'].values
69     dataset
70
71     # normalize the dataset
72     scaler = MinMaxScaler(feature_range=(0, 1))
73     dataset = dataset.reshape(-1, 1)
74     dataset = scaler.fit_transform(dataset)
75
76     # %%
77     # split into train and test sets
78     train_size = int(len(dataset) * 0.67)

```

```

78 test_size = len(dataset) - train_size
79 train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]
80 print(len(train), len(test))
81
82 # %%
83 # fix random seed for reproducibility
84 tf.random.set_seed(7)
85
86 dataset_ibov = ibov_cotacoes['BVSP'].values
87 dataset_ibov
88
89 # split into train and test sets
90 train_size_ibov = int(len(dataset_ibov) * 0.67)
91 test_size_ibov = len(dataset_ibov) - train_size_ibov
92 train_ibov, test_ibov = dataset_ibov[0:train_size_ibov], dataset_ibov[
    train_size_ibov:len(dataset_ibov)]
93 print(len(train_ibov), len(test_ibov))
94
95 # %%
96 # convert an array of values into a dataset matrix
97 def create_dataset(dataset, look_back=1):
98     dataX, dataY = [], []
99     for i in range(len(dataset)-look_back-1):
100         a = dataset[i:(i+look_back), 0]
101         dataX.append(a)
102         dataY.append(dataset[i + look_back, 0])
103     return np.array(dataX), np.array(dataY)
104
105 # %%
106 # reshape into X=t and Y=t+1
107 look_back = 1
108 trainX, trainY = create_dataset(train, look_back)
109 testX, testY = create_dataset(test, look_back)
110
111 # %%
112 # reshape input to be [samples, time steps, features]
113 trainX = np.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
114 testX = np.reshape(testX, (testX.shape[0], 1, testX.shape[1]))
115
116 # %%
117 # create and fit the LSTM network
118 model = Sequential()
119 model.add(LSTM(4, input_shape=(1, look_back)))
120 model.add(Dense(1))
121 model.compile(loss='mean_squared_error', optimizer='adam')
122 model.fit(trainX, trainY, epochs=100, batch_size=1, verbose=2)
123
124 # %%
125 # predi o futura
126 import numpy

```

```

127 validPredict = []
128 predito = model.predict(testX[0:1])
129 validPredict = predito
130
131 temp = testX[0:1][0][0][1:look_back]
132 temp = numpy.append([[[]]], [[temp]], axis=2)
133 temp = numpy.append(temp, [predito], axis=2)
134
135 for i in range(0, len(test)-look_back-2, 1):
136     predito = model.predict(temp)
137     temp = numpy.append([[temp[0:1][0][0][1:look_back]]], [predito], axis=2)
138     validPredict = numpy.append(validPredict, predito, axis = 0)
139
140 validPredict = scaler.inverse_transform(validPredict)
141
142 # %%
143 # make predictions
144 trainPredict = model.predict(trainX)
145 testPredict = model.predict(testX)
146 # invert predictions
147 trainPredict = scaler.inverse_transform(trainPredict)
148 trainY = scaler.inverse_transform([trainY])
149 testPredict = scaler.inverse_transform(testPredict)
150 testY = scaler.inverse_transform([testY])
151 # calculate root mean squared error
152 trainScore = np.sqrt(mean_squared_error(trainY[0], trainPredict[:,0]))
153 print('Train Score: %.2f RMSE' % (trainScore))
154 testScore = np.sqrt(mean_squared_error(testY[0], testPredict[:,0]))
155 print('Test Score: %.2f RMSE' % (testScore))
156
157 # %%
158 dif = []
159 ganho = 0
160 positivo = 0
161 negativo = 0
162 dias = 0
163 caixa = 1000
164 investido = 0
165 quantidade = 0
166 venda_pendente = 0
167 compra_pendente = 0
168 limite = 0.005
169
170 Total_Dias = len(testPredict)-1
171
172 #for i in range(0, len(testPredict)-1, 1):
173 for i in range(0, Total_Dias, 1):
174     if i == 0:
175         quantidade = round((caixa/2) / testY[0, i])
176         investido = investido + quantidade*testY[0, i]

```



```

177         caixa = caixa - quantidade*testY[0,i]
178
179     dias = dias + 1
180     dif.append(( testPredict[i,0] - testY[0,i-1])/testY[0,i-1] )
181     if dif[i] < -limite:
182         if quantidade <= 0:
183             print("Tentou Vender a a      o e n o tinha no Portfolio")
184         else:
185
186             if testPredict[i,0] > testY[0,i]:
187
188                 # Ajusta as vendas pendentes conforme as quantidades de
189                 # a     s
190                 while quantidade < (1 + venda_pendente):
191                     venda_pendente = venda_pendente - 1
192
193                 caixa = caixa + testY[0,i-1]*(1 + venda_pendente)
194                 investido = investido - testY[0,i-1]*(1 + venda_pendente)
195                 quantidade = quantidade - (1 + venda_pendente)
196
197                 ganho = ganho + (testY[0,i-1] - testPredict[i,0])*(1 +
198                 venda_pendente)
199                 caixa = caixa - testY[0,i]*(1 + venda_pendente)
200                 investido = investido + testY[0,i-1]*(1 + venda_pendente)
201                 quantidade = quantidade + (1 + venda_pendente)
202                 venda_pendente = 0
203
204                 print('Dia com Proje      o de Queda: ',i)
205                 print('Proje      o para Hoje: ',testPredict[i,0])
206                 print('Cota      o no Fechamento do dia Anterior: ',testY[0,i
207                 -1])
208                 print('Cota      o no Fechamento de Hoje: ',testY[0,i])
209                 print('Ganho Real: ',testY[0,i-1] - testY[0,i])
210                 print('Percentual: ',-dif[i]*100)
211                 print('Caixa: ',caixa)
212                 print('Investido: ',investido)
213                 print('Quantidade de Ativos: ',quantidade)
214
215             else:
216                 venda_pendente = venda_pendente + 1
217                 print("Vendas Pendentes: ", venda_pendente)
218                 print("Colocou      venda, mas ningu m comprou")
219
220     if dif[i] > limite:
221         if caixa <= 0:
222             print("Tentou Comprar a a      o e n o tinha dinheiro em Caixa")
223         else:
224
225             if testPredict[i,0] < testY[0,i]:

```

```

224         # Ajusta as compras pendentes conforme o caixa
225         while caixa < testY[0,i-1]*(1 + compra_pendente):
226             compra_pendente = compra_pendente - 1
227
228             caixa = caixa - testY[0,i-1]*(1 + compra_pendente)
229             investido = investido + testY[0,i-1]*(1 + compra_pendente)
230             quantidade = quantidade + (1 + compra_pendente)
231
232             ganho = ganho + (testPredict[i,0]-testY[0,i-1])*(1 +
                compra_pendente)
233             caixa = caixa + testY[0,i-1]*(1 + compra_pendente)
234             investido = investido - testY[0,i-1]*(1 + compra_pendente)
235             quantidade = quantidade - (1 + compra_pendente)
236             compra_pendente = 0
237
238             print('Dia com Projeção de Subida: ',i)
239             print('Projeção para Hoje: ',testPredict[i,0])
240             print('Cotação no Fechamento do dia Anterior: ',testY[0,i
                -1])
241             print('Cotação no Fechamento de Hoje: ',testY[0,i])
242             print('Ganho Real: ',ganho)
243             print('Percentual: ',dif[i]*100)
244             print('Caixa: ',caixa)
245             print('Investido: ',investido)
246             print('Quantidade de Ativos: ',quantidade)
247
248         else:
249             compra_pendente = compra_pendente + 1
250             print("Compras Pendentes: ", compra_pendente)
251             print("Tentou comprar, mas ninguém vendeu pelo preço
                solicitado")
252
253     print('+++++')
254     print('Ganho nas Compras e Vendas de Ativos: ', ganho)
255     print('Rendimento no Day Trade: ', ganho/testY[0,0]*100)
256     print('Dias de Operação: ', dias)
257     print('Dinheiro em Caixa: ', caixa)
258     print('Valor Investido: ', investido)
259     print('Quantidade de Ativos: ',quantidade)
260     print('Valor dos Ativos: ',quantidade*testY[0,Total_Dias])
261     print('Valor Total (Caixa + Ativos): ',caixa + quantidade*testY[0,
        Total_Dias])
262     print('Ganho Total (Caixa + Ativos): ', (caixa + quantidade*testY[0,
        Total_Dias]) -1000)
263     print('Rendimento Total (Caixa + Ativos): ', 100*((caixa + quantidade*testY
        [0,Total_Dias]) -1000)/1000)
264     print('Rendimento Ibov ', 100*(test_ibov[Total_Dias-1] - test_ibov[0])/
        test_ibov[0])
265     print('Rendimento da Ação ', 100*(testY[0,Total_Dias] - testY[0,0])/testY
        [0,0])

```

```

266
267
268 # %%
269 # shift train predictions for plotting
270 trainPredictPlot = np.empty_like(dataset)
271 trainPredictPlot[:, :] = np.nan
272 trainPredictPlot[look_back:len(trainPredict)+look_back, :] = trainPredict
273
274 # shift test predictions for plotting
275 testPredictPlot = np.empty_like(dataset)
276 testPredictPlot[:, :] = np.nan
277 testPredictPlot[len(trainPredict)+(look_back*2)+1:len(dataset)-1, :] =
    testPredict
278
279 # plot baseline and predictions
280 plt.figure(figsize=(15, 6))
281 plt.plot(cotacoes.index, scaler.inverse_transform(dataset))
282 plt.plot(cotacoes.index, trainPredictPlot)
283 plt.plot(cotacoes.index, testPredictPlot)
284 plt.xlabel('Data')
285 plt.ylabel('Pre o da A o')
286 plt.legend(['Valor Real', 'Predi o no Treino', 'Predi o no Teste'],
    loc=0)
287 #plt.plot(validPredictPlot)
288 plt.show()
289
290 # %%
291 plt.figure(figsize=(15, 6))
292 plt.plot(dif[1:len(dif)])
293 plt.plot([limite]*len(dif))
294 plt.plot([-limite]*len(dif))
295
296 # %%
297 import ta.others
298 import ta.volatility
299
300 ativo = pd.DataFrame(np.transpose(testY))
301 ativo.columns = ['Valor']
302 ativo['LogRet'] = ta.others.daily_log_return(close = ativo['Valor'], fillna
    =True)
303 vol = np.sqrt(np.var(ativo['LogRet']))
304
305 RF = 100*((caixa + quantidade*testY[0, Total_Dias]) - 1000)/1000
306 Ibov = 100*(test_ibov[Total_Dias - 1] - test_ibov[0])/test_ibov[0]
307 print("Sharpe: ", (RF - Ibov)/vol)

```