

CENTRO UNIVERSITÁRIO DA FEI

VICTOR BIAZON

**REDES DE UNIDADES RECORRENTES CHAVEADAS E TRANSFORMADAS  
DISCRETAS DE ONDALETAS APLICADAS À PREVISÃO E OPERAÇÃO NO  
MERCADO FINANCEIRO**

São Bernardo do Campo

2021

VICTOR BIAZON

**REDES DE UNIDADES RECORRENTES CHAVEADAS E TRANSFORMADAS  
DISCRETAS DE ONDALETAS APLICADAS À PREVISÃO E OPERAÇÃO NO  
MERCADO FINANCEIRO**

Dissertação de Mestrado, apresentada ao Centro Universitário da FEI para obtenção do título de Mestre em Engenharia Elétrica. Orientado pelo Prof. Dr. Reinaldo Augusto da Costa Bianchi.

São Bernardo do Campo

2021

Biazon, Victor.

REDES DE UNIDADES RECORRENTES CHAVEADAS E  
TRANSFORMADAS DISCRETAS DE ONDALETAS APLICADAS À  
PREVISÃO E OPERAÇÃO NO MERCADO FINANCEIRO / Victor  
Biazon. São Bernardo do Campo, 2020.

109 f. : il.

Dissertação - Centro Universitário FEI.

Orientador: Prof. Dr. Reinaldo Augusto da Costa Bianchi.

1. Redes Neurais Recorrentes. 2. Mercado de Ações. 3.  
Ondaletas. I. Augusto da Costa Bianchi, Reinaldo, orient. II.  
Título.

**Aluno:** Victor Biazon

**Matrícula:** 119115-4

**Título do Trabalho:** Rede de unidades recorrentes chaveadas e transformadas discretas de ondaletas aplicadas à previsão e operação no mercado financeiro.

**Área de Concentração:** Inteligência Artificial Aplicada à Automação e Robótica

**Orientador:** Prof. Dr. Reinaldo Augusto da Costa Bianchi

**Data da realização da defesa:** 08/12/2020

**ORIGINAL ASSINADA**

**Avaliação da Banca Examinadora:**

A banca foi realizada no dia 08 de dezembro às 14:00 horas, e se iniciou com a apresentação do aluno, que foi muito boa, e seguiu para a arguição, onde o aluno respondeu a todas as questões de forma adequada demonstrando conhecimento pleno do tema.

A aprovação foi por unanimidade.

São Bernardo do Campo,     /     /     .

**MEMBROS DA BANCA EXAMINADORA**

Prof. Dr. Reinaldo Augusto da Costa Bianchi

Ass.: \_\_\_\_\_

Prof. Dr. Carlos Eduardo Thomaz

Ass.: \_\_\_\_\_

Prof. Dr. Adriano Mussa

Ass.: \_\_\_\_\_

A Banca Julgadora acima-assinada atribuiu ao aluno o seguinte resultado:

APROVADO

REPROVADO

**VERSÃO FINAL DA DISSERTAÇÃO**

**APROVO A VERSÃO FINAL DA DISSERTAÇÃO EM QUE  
FORAM INCLUÍDAS AS RECOMENDAÇÕES DA BANCA  
EXAMINADORA**

Aprovação do Coordenador do Programa de Pós-graduação

\_\_\_\_\_  
Prof. Dr. Carlos Eduardo Thomaz

A Deus, meus pais, amigos e colegas de pesquisa.

## **AGRADECIMENTOS**

Agradeço primeiramente a meus pais que desde pequeno sempre incentivaram o estudo acima de tudo. Meus amigos e colegas de pesquisa pelas opiniões e dicas sobre direções de pesquisa. Aos professores avaliadores pelas importantes remarques e direcionamento claro e conciso. Ao meu professor orientador, Reinaldo Bianchi, por acreditar em minha pesquisa e conceder todo suporte técnico necessário para sua confecção. E finalmente aos meus amigos que foram deixados levemente de lado durante o árduo processo de pesquisa, mas sempre me apoiaram.

“Os analistas não se preocupam em estar errados no geral, mas odeiam estar errados em particular.”

Benjamin Graham

“Se você quer ter resultados melhores do que a maioria das pessoas, faça as coisas de forma diferente da maioria.”

John Templeton

“Eu seria mendigo nas ruas com uma caneca de lata se os mercados fossem eficientes.”

Warren Buffet

## RESUMO

Operar no mercado de ações sempre trás consigo o desafio de escolher a melhor decisão a ser tomada a cada passo no tempo. O problema é intensificado pela teoria de que não é possível prever uma série temporal do mercado financeiro uma vez que toda informação relacionada ao preço da ação já está contida em si, teoria conhecida como Hipótese dos Mercados Eficientes - *Efficient Market Hypothesis* (EMH). Embora o mercado em geral não tenha tendências distinguíveis, portanto sendo consistente com a EMH, existem várias janelas de tempo onde há alguma previsibilidade de certa extensão nos dados caso consideremos o uso de indicadores técnicos. Neste trabalho é proposto um novo modelo que busca se beneficiar de tais períodos operando para escolher suas decisões e aguardando o melhor momento para executá-las. Este modelo, chamado Rede de Transformada Discreta de Ondas e Unidade Recorrente Chaveada - *Discrete Wavelet Transform Gated Recurrent Unit Network* (DWT-GRU), é dividido em três módulos, sendo eles, o pré processamento dos dados pela transformada de wavelets, o treinamento e a predição do preço de fechamento da ação para dois dias no futuro e a tomada da decisão baseada na avaliação do gradiente do preço de fechamento. O modelo proposto foi comparado com outras arquiteturas de Rede Neural Recorrente - *Recurrent Neural Network* (RNN), com e sem o uso de pré processamento de ondas, e a estratégia de "comprar e segurar". Os resultados mostraram que o modelo proposto superou todos as métricas estatísticas, de acurácia, precisão, revocação e F1, e retorno financeiro de todos os modelos de comparação estabelecidos nas ações analisadas do mercado financeiro brasileiro. As ações analisadas como base para o estudo foram as *blue-chips* do índice do IBOVESPA sendo elas a PETR4, VALE3, ITUB4, ABEV3, e a Fundo de Índice Negociado em Bolsa - *Exchange-Traded Fund* (ETF) que espelha o próprio índice, BOVA11. Para dados de treinamento foram utilizados dados desde 2001 para as ações e desde 2008 para a Fundo de Índice Negociado em Bolsa - *Exchange-Traded Fund* (ETF) BOVA11. Por fim é apresentado o resultado financeiro da aplicação do algoritmo em operações em tempo real de *swing-trade* comprovando sua eficácia e vencendo a estratégia de "comprar e segurar".

Palavras-chave: Redes. Neurais. Recorrentes. Mercado. Ações. Ondas.

## ABSTRACT

Trading in the stock market always comes with the challenge of choosing the best decision to take on each time step. The problem is intensified by the theory that it is not possible to predict stock market time series as all information related to the stock price is already contained in it, which theory is known as Efficient Market Hypothesis. Although the market, in general, has no distinguishable tendencies, thus being consistent to the EMH, there are several time windows where there is some predictability, to some extent in the data, if we consider the use of technical indicators. In this work, a novel model is proposed to seek benefit from said periods operating to choose its actions and waiting for the best moment to execute them. This model, called Discrete Wavelet Transform Gated Recurrent Unit Network (DWT-GRU), is divided in three modules, them being, the preprocessing of the data by the wavelet transform, the training and prediction of the closing price two days in the future and the decision making based on the evaluation of the gradient of the closing price. The proposed model was compared to other RNN architectures, with and without the use of wavelet preprocessing, and the "buy-and-hold" strategy. The results shown that the proposed model surpassed all the statistical metrics of accuracy, precision, recall, F1 and financial return of all the established comparison models in the analysed stocks of the Brazilian Stock Market. The analysed stocks as the base for the study were the blue-chips of the IBOVESPA index, them being, PETR4, VALE3, ITUB4, ABEV3, and the ETF that mirrors the index itself, BOVA11. As training data the analysed period was since 2001 for the stocks and 2008 for the Fundo de Índice Negociado em Bolsa - *Exchange-Traded Fund* (ETF) BOVA11. At last, it is presented the financial results of the application of the algorithm in real time swing-trade operations validating its efficiency and winning over the buy-and-hold strategy.

Keywords: Networks. Neural. Recurrent. Market. Stock. Wavelets.

## LISTA DE ILUSTRAÇÕES

Figure 1	– Diagrama de Blocos de uma Célula LSTM. . . . .	25
Figure 2	– Comparação LSTM e GRU . . . . .	28
Figure 3	– Decomposição DWT de um sinal em dois níveis. . . . .	30
Figure 4	– Transformada Empírica de <i>Wavelets</i> da ação PETR4 em 5 componentes. . . . .	31
Figure 5	– Exemplo da Filtragem de Ruído DWT aplicada a ação PETR4 no ano 2019. . . . .	31
Figure 6	– Exemplo de escala e translação no tempo - Wavelet Laplaciano da Gaussiana. . . . .	32
Figure 7	– Funções de <i>wavelets</i> e escala para as funções Daubechies de ordem 1 a 6 . . . . .	35
Figure 8	– Funções de <i>wavelets</i> e escala para as funções Coiflets de ordem 1 a 5. . . . .	36
Figure 9	– Funções de <i>wavelets</i> e escala para as funções Symlets de ordem 1 a 6. . . . .	37
Figure 10	– Funções de <i>wavelets</i> e escala para as funções Bi-ortogonais de ordem 1 a 3. . . . .	38
Figure 11	– Exemplo de métodos do ATR . . . . .	46
Figure 12	– Retorno sobre investimento do trabalho de Nelson, Pereira e Oliveira (2017) no ano de 2017 . . . . .	56
Figure 13	– Decomposição de <i>Wavelets</i> empíricas do trabalho de Altan, Karasu e Bekiros (2019) para o BTC-USD em seu preço histórico até 2019. . . . .	58
Figure 14	– Resultados da proposta de Sezer, Ozbayoglu e Dogdu (2017) na Dow 30 no ano de 2017 . . . . .	60
Figure 15	– Retorno sobre investimento do trabalho de Pimenta et al. (2018) no ano de 2013 a 2015 . . . . .	61
Figure 16	– Diagrama de Blocos do Sistema . . . . .	64
Figure 17	– Divisão dos dados de treinamento. . . . .	67
Figure 18	– Modelo de Rede Neural Proposta . . . . .	68
Figure 19	– Primeiro e segundo gradientes do sinal filtrado da ação PETR4 no ano 2019. . . . .	69
Figure 20	– Ações do algoritmo e lucro na ação PETR4 - Modelo Puro LSTM . . . . .	71
Figure 21	– Ações do algoritmo e lucro na ação PETR4 - Modelo Puro GRU . . . . .	72
Figure 22	– Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-LSTM . . . . .	72
Figure 23	– Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-GRU . . . . .	73
Figure 24	– Lucro de Retorno sobre Investimento para Cada Ação . . . . .	75
Figure 25	– Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-GRU - Janela Deslizante em 2015 . . . . .	79

Figure 26 – Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-GRU - Janela Deslizante em 2016 . . . . .	79
Figure 27 – Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-GRU - Janela Deslizante em 2017 . . . . .	80
Figure 28 – Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-GRU - Janela Deslizante em 2018 . . . . .	80
Figure 29 – Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-GRU - Janela Deslizante em 2019 . . . . .	81
Figure 30 – Ações do algoritmo e lucro na <i>commodity</i> do petróleo CL=F - Modelo DWT-GRU - Normalização Local em 2019 . . . . .	83
Figure 31 – Ações do algoritmo e lucro na <i>commodity</i> do ouro GC=F - Modelo DWT-GRU - Normalização Local em 2019 . . . . .	84
Figure 32 – Ações do algoritmo e lucro na <i>commodity</i> da prata SI=F - Modelo DWT-GRU - Normalização Local em 2019 . . . . .	85
Figure 33 – Ações do algoritmo e lucro na PETR4 - Modelo Floresta Aleatória sem DWT em 2019 . . . . .	86
Figure 34 – Ações do algoritmo e lucro na PETR4 - Modelo Floresta Aleatória com DWT em 2019 . . . . .	87
Figure 35 – Ações do algoritmo e lucro na PETR4 - Modelo Regressão Linear sem DWT em 2019 . . . . .	87
Figure 36 – Ações do algoritmo e lucro na PETR4 - Modelo Regressão Linear com DWT em 2019 . . . . .	88
Figure 37 – Ações do algoritmo e lucro na PETR4 - Modelo DWT-GRU no período de 22/09/2020 à 13/11/2020 . . . . .	90
Figure 38 – Relatório do MetaTrader5 para as operações - Modelo DWT-GRU no período de 22/09/2020 à 13/11/2020 . . . . .	91
Figure 39 – Reta tangente PQ de Q tendendo a P. . . . .	102
Figure 40 – Exemplo de reta e dados de uma regressão linear de uma dimensão. . . . .	105
Figure 41 – Espaço de decisão de uma árvore de decisão em duas dimensões. . . . .	108
Figure 42 – Árvore de decisão resultante da separação em folhas do espaço de estado de duas dimensões . . . . .	109

## LISTA DE TABELAS

Table 1	– Métricas do Modelo DWT-GRU . . . . .	75
Table 2	– Métricas do Modelo LSTM . . . . .	75
Table 3	– Métricas do Modelo GRU . . . . .	76
Table 4	– Métricas do Modelo DWT-LSTM . . . . .	76
Table 5	– Métricas do Modelo DWT-GRU na ação PETR4 - Janela Deslizante . . . .	77
Table 6	– Métricas do Modelo DWT-GRU na ação VALE3 - Janela Deslizante . . . .	77
Table 7	– Métricas do Modelo DWT-GRU na ação ITUB4 - Janela Deslizante . . . .	77
Table 8	– Métricas do Modelo DWT-GRU na ação ABEV3 - Janela Deslizante . . . .	77
Table 9	– Métricas do Modelo DWT-GRU na ação BOVA11 - Janela Deslizante . . . .	78
Table 10	– Comparação das métricas da janela fixa x deslizante para PETR4 . . . . .	78
Table 11	– Comparação das métricas da normalização global x normalização local para PETR4 . . . . .	81
Table 12	– Comparação das métricas da normalização global x normalização local para VALE3 . . . . .	82
Table 13	– Comparação das métricas da normalização global x normalização local para ITUB4 . . . . .	82
Table 14	– Comparação das métricas da normalização global x normalização local para ABEV3 . . . . .	82
Table 15	– Comparação das métricas da normalização global x normalização local para BOVA11 . . . . .	82
Table 16	– Métricas do Modelo PETR4 com Normalização Local - <i>Commodities</i> . . . .	84
Table 17	– Comparação das métricas da floresta aleatória com e sem DWT . . . . .	88
Table 18	– Comparação das métricas da regressão linear com e sem DWT . . . . .	88
Table 19	– Métricas estatísticas do Modelo DWT-GRU no período de 22/09/2020 à 13/11/2020 . . . . .	90
Table 20	– Taxas de emolumentos B3 para negociação e liquidação . . . . .	92

## LIST OF ALGORITHMS

Algoritmo 1 – Algoritmo de Tomada de Decisão . . . . .	69
Algoritmo 2 – Algoritmo de Avaliação de Retorno Financeiro . . . . .	74

## LISTA DE ABREVIATURAS

ABC	Colônia Artificial de Abelhas - <i>Artificial Bee Colony</i>
ADI	Índice de Distribuição de Acúmulo - <i>Accumulation Distribution Index</i>
ADX	Índice Médio Direcional - <i>Average Directional Index</i>
AI	Inteligência Artificial - <i>Artificial Intelligence</i>
ANN	Rede Neural Artificial - <i>Artificial Neural Network</i>
AO	Oscilador Incrível - <i>Awesome Oscillator</i>
API	Interface de Programação de Aplicações - <i>Application Programming Interface</i>
ATR	Variação Verdadeira Média - <i>Average True Range</i>
BB	Bandas de Bollinger - <i>Bollinger Bands</i>
BGRU	Unidade Recorrente Chaveada Bi-direcional - <i>Bidirectional Gated Recurrent Unit</i>
BNH	Comprar e Segurar - <i>Buy and Hold</i>
BPTT	Retro Propagação Através do Tempo - <i>Back-Propagation Through Time</i>
CCI	Índice de Canal de Mercadoria - <i>Commodity Channel Index</i>
CEC	Carrossel de Erro Constante - <i>Constant Error Carousel</i>
CMF	Fluxo Monetário de Chaikin - <i>Chaikin Money Flow</i>
CNN	Rede Neural Convolucional - <i>Convolutional Neural Network</i>
CS	Busca de Cuco - <i>Cuckoo Search</i>
CWT	Transformada Contínua de Ondaletas - <i>Continuous Wavelet Transforms</i>
DC	Canais de Donchian - <i>Donchian Channels</i>
DMI	Indicador Direcional de Movimento - <i>Directional Movement Indicator</i>
DNN	Rede Neural Profunda - <i>Deep Neural Network</i>
DPO	Oscilador de Preço Distendente - <i>Detrended Price Oscillator</i>
DRNN	Rede Neural Recorrente Profunda - <i>Deep Recurrent Neural Network</i>
DWT	Transformada Discreta de Ondaletas - <i>Discrete Wavelet Transforms</i>
DWT-GRU	Rede de Transformada Discreta de Ondaletas e Unidade Recorrente Chaveada - <i>Discrete Wavelet Transform Gated Recurrent Unit Network</i>
DWT-LSTM	Rede de Transformada Discreta de Ondaletas e Memória Longa de Curto Prazo - <i>Discrete Wavelet Transform Long Short-Term Memory Network</i>
EMA	Média Móvel Exponencial - <i>Exponential Moving Average</i>
EMH	Hipótese dos Mercados Eficientes - <i>Efficient Market Hypothesis</i>

EoM	Facilidade de Movimento - <i>Ease of Movement</i>
ETF	Fundo de Índice Negociado em Bolsa - <i>Exchange-Traded Fund</i>
EWT	Transformada Empírica de Ondas - <i>Empirical Wavelet Transform</i>
FFNN	Rede Neural Sem Realimentação - <i>Feed-forward Neural Network</i>
FI	Índice de Força - <i>Force Index</i>
GA	Algoritmo Genético - <i>Genetic Algorithm</i>
GRU	Unidade Recorrente Chaveada - <i>Gated Recurrent Unit</i>
IDWT	Transformada Inversa Discreta de Ondas - <i>Inverse Discrete Wavelet Transform</i>
KAMA	Média Móvel Adaptiva de Kaufman - <i>Kaufman's Adaptive Moving Average</i>
KC	Canais de Keltner - <i>Keltner Channels</i>
KST	Sabe com Certeza - <i>Know Sure Thing</i>
LSTM	Memória Longa de Curto Prazo - <i>Long Short-Term Memory</i>
MACD	Média Móvel de Convergente e Divergente - <i>Moving Average Convergence Divergence</i>
MFI	Índice de Fluxo Monetário - <i>Money Flow Index</i>
MI	Índice de Massa - <i>Mass Index</i>
MLP	Perceptron de Multi Camada - <i>Multi-Layer Perceptron</i>
NVI	Índice de Volume Negativo - <i>Negative Volume Index</i>
OBV	Volume Balanceado - <i>On-Balance Volume</i>
OHLC	Abertura, Máximo, Mínimo e Fechamento - <i>Open, High, Low and Close</i>
PCA	Análise de Componente Principal - <i>Principal Component Analysis</i>
RBFNN	Rede Neural de Função Base Radial - <i>Radial Basis Function Neural Network</i>
RNN	Rede Neural Recorrente - <i>Recurrent Neural Network</i>
ROC	Taxa de Mudança - <i>Rate of Change</i>
ROI	Retorno Sobre Investimento - <i>Return Over Investment</i>
RSI	Índice de Força Relativa - <i>Relative Strength Index</i>
SMA	Média Móvel Simples - <i>Simple Moving Average</i>
SO	Oscilador Estocástico - <i>Stochastic Oscillator</i>
SRNN	Rede Neural Recorrente Simples - <i>Simple Recurrent Neural Network</i>
SVM	Máquina de Vetor de Suporte - <i>Support Vector Machine</i>
SVR	Regressão de Vetor de Suporte - <i>Support Vector Regression</i>
TP	Preço Típico - <i>Typical Price</i>

TR	Varição Verdadeira - <i>True Range</i>
TRIX	Média Exponencial Tripla - <i>Triple Exponential Average</i>
TSI	Índice de Força Verdadeira - <i>True Strength Index</i>
UO	Oscilador Definitivo - <i>Ultimate Oscillator</i>
VPT	Tendência Volume-Preço - <i>Volume-Price Trend</i>
W %R	Indicador Williams %R - <i>Williams %R Indicator</i>



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	20
1.1	Motivação	21
1.2	Objetivo	21
1.3	Organização do Texto	22
<b>2</b>	<b>TEORIA</b>	23
2.1	Redes Neurais Recorrentes	23
<b>2.1.1</b>	<b>Memória Longa de Curto Prazo (<i>Long Short-Term Memory</i>)</b>	24
<b>2.1.2</b>	<b>Unidades Recorrentes Chaveadas (<i>Gated Recurrent Unit</i>)</b>	27
2.2	Ondaletas ( <i>Wavelets</i> )	29
<b>2.2.1</b>	<b>Filtragem de Ruído de Ondaletas (<i>Wavelet Denoising</i>)</b>	30
<b>2.2.2</b>	<b>Transformadas de Ondaletas (<i>Wavelets Transform</i>)</b>	32
<b>2.2.3</b>	<b>Transformada Contínua de Ondaletas (<i>Continuous Wavelet Transform</i>)</b>	33
<b>2.2.4</b>	<b>Transformada Discreta de Ondaletas (<i>Discrete Wavelet Transform</i>)</b>	33
2.3	Tipos de Ondaletas	34
<b>2.3.1</b>	<b>Ondaletas Daubechies</b>	34
<b>2.3.2</b>	<b>Ondaletas Coiflets</b>	34
<b>2.3.3</b>	<b>Ondaletas Symlets</b>	35
<b>2.3.4</b>	<b>Ondaletas Bi-ortogonais</b>	35
2.4	Análise Técnica no Mercado de Ações	36
<b>2.4.1</b>	<b>Indicadores de Inércia</b>	39
<b>2.4.2</b>	<b>Indicadores de Volume</b>	42
<b>2.4.3</b>	<b>Indicadores de Volatilidade</b>	45
<b>2.4.4</b>	<b>Indicadores de Tendência</b>	47
<b>2.4.5</b>	<b>Análise de Retorno Sobre Investimento</b>	52
2.5	Métricas Estatísticas de classificação	52
<b>2.5.1</b>	<b>Acurácia</b>	53
<b>2.5.2</b>	<b>Precisão</b>	53
<b>2.5.3</b>	<b>Revocação</b>	53
<b>2.5.4</b>	<b>F1</b>	54

<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	55
3.1	Redes Neurais Recorrentes	55
3.2	Ondaletas	57
3.3	Análise Técnica no Mercado de Ações	59
<b>4</b>	<b>METODOLOGIA</b>	63
4.1	Arquitetura do Sistema	63
4.2	Aquisição de Dados	65
4.3	Indicadores Técnicos	65
4.4	Pré-processamento dos Dados	65
<b>4.4.1</b>	<b>Janela de treinamento</b>	66
<b>4.4.2</b>	<b>Normalização dos dados</b>	66
4.5	Modelo Proposto	67
4.6	Tomada de Decisão	68
<b>5</b>	<b>EXPERIMENTOS E RESULTADOS</b>	70
5.1	Avaliação do Modelo	71
5.2	Desenvolvimento de Outros Modelos	72
5.3	Análise	73
<b>5.3.1</b>	<b>Resultado Financeiro</b>	73
<b>5.3.2</b>	<b>Métricas Estatísticas</b>	74
5.4	Treinamento com Janela Deslizante	76
5.5	Normalização Local	81
<b>5.5.1</b>	<b>Aplicação de Modelo treinado à <i>commodities</i></b>	83
5.6	Modelos clássicos de aprendizado de máquina	85
5.7	Testes em Tempo Real	89
<b>5.7.1</b>	<b>Resultados e Discussão</b>	89
<b>5.7.2</b>	<b>Emolumentos</b>	91
<b>5.7.3</b>	<b>Resumo da Seção</b>	92
<b>6</b>	<b>CONCLUSÃO</b>	94
6.1	Trabalhos Futuros	95
	<b>REFERÊNCIAS</b>	97
6.2	Apêndice I - Cálculo	102
<b>6.2.1</b>	<b>Tangentes</b>	102
<b>6.2.2</b>	<b>Derivada</b>	103

6.3	Apêndice - Algoritmos de Aprendizado de Máquina . . . . .	104
<b>6.3.1</b>	<b>Regressão Linear</b> . . . . .	104
<b>6.3.2</b>	<b>Regressão Polinomial</b> . . . . .	106
<b>6.3.3</b>	<b>Aprendizado por Conjunto (<i>Ensemble</i>)</b> . . . . .	107
<b>6.3.4</b>	<b>Floresta Aleatória (<i>Random Forest</i>)</b> . . . . .	107

## 1 INTRODUÇÃO

A teoria de Dow é o alicerce da análise técnica moderna e afirma que todos os aspectos que possam influenciar o preço de uma determinada ação já são levados em consideração naturalmente pelo mercado em si. Embora notícias e desastres climáticos não possam ser previstos, o mercado rapidamente se adapta assimilando os efeitos de tais eventos no preço atual (BROWN; GOETZMANN; KUMAR, 1998).

Como disse Rhea (1932) em seu livro sobre a teoria de Dow: “A média desconta tudo: As flutuações dos preços diários de fechamento da ‘Dow-Jones Ferrovia e Industria’ é um índice médio composto por todas as esperanças, desapontamentos, e conhecimento de todos que sabem qualquer coisa de assuntos financeiros, e por tal razão os efeitos dos eventos futuros (excluindo-se atos de Deus) são sempre propriamente antecipados em seu movimento. A média rapidamente avalia tais calamidades como incêndios e terremotos.”

A Hipótese dos Mercados Eficientes - *Efficient Market Hypothesis* (EMH) (MALKIEL; FAMA, 1970) diz que as séries temporais dos preços do mercado de ações são quase sempre imprevisíveis, pois toda parte de informação relevante que pode influenciar o preço já foi tomada em consideração incluindo valores e volumes passados. Isso implica que o preço responde imediatamente a nova informações e não está atrelado a nenhuma tendência ou padrão. A EMH afirma que qualquer tipo de predição está fadada a não ter melhor resultado que operações aleatórias, e o preço da ação será sempre o justo, portanto imprevisível (NOBRE; NEVES, 2019).

Como dito por Malkiel e Fama (1970) em sua teoria, a eficiência do mercado pode ser classificada como fraca, semi-forte ou forte. Desta forma quanto mais forte a eficiência do mercado, menos podemos prever a direção dos preços mesmo que tenhamos novas informações sobre eles.

Este trabalho propõe que mesmo que o mercado em geral oscile em uma forma pseudo-aleatória, existem certos períodos de previsibilidade em certa extensão, onde a eficiência do mercado seja mais leve, que podem ser explorados para se adquirir mais lucro do que a estratégia de Comprar e Segurar - *Buy and Hold* (BNH) em mercados crescentes (*Bull Markets*) e tendo resultados positivos em mercados em queda (*Bear Markets*).

## 1.1 Motivação

A pesquisa na área de automatização de decisões no mercado de ações com redes neurais é um tema já consolidado e existem diversas aplicações onde outros autores buscaram de diversas formas obter êxito em prever preços ou tendências de mercado, como será discutido na seção 3 de trabalhos relacionados. Como é mostrado pela pesquisa de Bigiotti e Navarra (2018), no ano de 2018 cerca de 80% das operações financeiras de câmbio já eram feitas por robôs de investimento ou automação em algum grau.

Na tentativa de se entender o mercado de ações e *commodities*, diversos matemáticos, estatísticos e investidores profissionais desenvolveram metodologias as quais hoje são conhecidos como indicadores técnicos financeiros, para analisar as variações dos preços em função do tempo e quantificar sua variação com o objetivo de identificar tendências ou sinais de reversão nas mesmas.

Devido a queda nas taxas básicas de juros, os números de investidores de pessoa física tem crescido muito nos últimos tempos. Tais investidores entram no mercado com pouca ou nenhuma instrução de como operar com segurança e acabam tomando riscos elevados com a promessa de retornos maiores ainda. Devido ao grande interesse do público, também cresceu a procura por métodos mais seguros de se obter retorno na bolsa de valores <sup>1</sup>

## 1.2 Objetivo

Esta pesquisa tem como objetivo o uso da Transformada Discreta de Ondas - *Discrete Wavelet Transforms* (DWT) aliada à capacidade de aprendizado da Rede Neural Recorrente - *Recurrent Neural Network* (RNN), mais especificamente a relativamente recém desenvolvida Unidade Recorrente Chaveada - *Gated Recurrent Unit* (GRU), para previsão e operações de compra, venda ou aguardo no mercado financeiro brasileiro.

Este trabalho terá como foco principal a aplicação de seus esforços à bolsa de valores brasileira para estudos iniciais e validação dos métodos. Serão utilizadas, principalmente, as ações de maior expressão na bolsa como é o caso das chamadas *blue chips*. Após estruturada e testada a arquitetura proposta será aplicada outros tipos de ativos, como por exemplo *commodities*, para análise de sua eficiência em outras economias e ativos independentes.

---

<sup>1</sup><https://valorinveste.globo.com/mercados/renda-variavel/noticia/2020/09/02/numero-de-pessoas-fisicas-na-bolsa-cresce-ainda-mais-em-agosto-e-encosta-em-3-milhoes.ghtml>

Para avaliação do resultado da pesquisa foram formuladas comparações com outros modelos de redes neurais recorrentes, com e sem pré processamento de DWT, analisando o retorno financeiro e as métricas estatísticas de cada um.

Para o treinamento serão utilizados os dados históricos das ações em questão de 2001 a 2015, sendo o treinamento validado de 2017 a 2018 e testado no ano de 2019.

### 1.3 Organização do Texto

A organização do texto se dá da seguinte forma: O capítulo 2 apresenta a teoria base do modelo proposto; Capítulo 3 revisa trabalho correlatos; Capítulo 4 propõe a metodologia de desta pesquisa, explicitando o modelo proposto; Capítulo 5 demonstra os resultados e as comparações do modelo e o capítulo 6 apresenta a conclusão, contribuição de pesquisa e trabalho futuros

## 2 TEORIA

Nesta seção serão apresentadas as teorias, conceitos e equações utilizadas na concepção da proposta de pesquisa apresentada nas próximas seções deste trabalho. A divisão se dá em subseções agrupadas por tema, sendo elas: Redes Neurais Recorrentes, Ondas e Análise Técnica no Mercado Financeiro, Métricas estatísticas, Cálculo e Aprendizado de Máquina.

### 2.1 Redes Neurais Recorrentes

Uma Rede Neural Recorrente - *Recurrent Neural Network* (RNN) (WEIGEND; RUMELHART; HUBERMAN, 1991) compõe uma classe diferente de Rede Neural Artificial - *Artificial Neural Network* (ANN) (HAYKIN et al., 2009) construída especificamente para analisar e aprender a partir de dados sequenciais com, ou sem, dependências multi-variável. Geralmente RNNs podem processar dados de comprimento variável o que não ocorre no método de processamento sequencial de dados de uma Rede Neural Sem Realimentação - *Feed-forward Neural Network* (FFNN) (TANG; FISHWICK, 1993) que deve receber valores presentes e passados com comprimento fixo.

Diversas variações de RNNs foram desenvolvidas ao longo dos anos, como a Memória Longa de Curto Prazo - *Long Short-Term Memory* (LSTM) (HOCHREITER; SCHMIDHUBER, 1997), Unidade Recorrente Chaveada - *Gated Recurrent Unit* (GRU) (CHUNG et al., 2014) e Rede Neural Recorrente Profunda - *Deep Recurrent Neural Network* (DRNN) (PASCANU et al., 2013). Todos os modelos compartilham a necessidade de lembrar entradas e saídas passadas para prever o próximo estado das redes e adaptar a si mesmos (GOODFELLOW; BENGIO; COURVILLE, 2016). Como afirmado por Gopalswamy, Tighe e Rashidi (2017) existem diversas aplicações onde o uso de RNNs pode ser o fator determinante em prever corretamente o resultado dos dados analisados, como por exemplo: tradução de máquina, reconhecimento de fala, entendimento de linguagem, classificação de imagens e previsão de séries temporais.

As redes do tipo LSTM e GRU foram especificamente desenvolvidas para superar a dificuldade de treinamento da explosão ou desaparecimento de gradiente. Estes modelos propagam os erros através de estados internos em unidade especiais controladas por "chaves" ativadas em cada célula pelo algoritmo de aprendizado (HOCHREITER; SCHMIDHUBER, 1997) (CHO et al., 2014).

Já no caso da implementação de uma Rede Neural Recorrente Profunda - *Deep Recurrent Neural Network* (DRNN) existem três possibilidades para se aumentar a profundidade das redes como afirmado por Pascanu et al. (2013):

- a) Os estados recorrentes ocultos podem ser segmentados em grupos organizados hierarquicamente.
- b) Computação profunda (por exemplo uma FFNN) pode ser introduzida entre as camadas: entrada-camada oculta, camada oculta - camada oculta e camada oculta - saída. Isto inclui entre tais camadas a capacidade de representação de dados das FFNN de selecionar as características importantes de cada conjunto de dados em sua entrada, facilitando o aprendizado.
- c) O efeito de aumento do caminho pode ser mitigado ao se introduzir conexões esparsas.

### 2.1.1 Memória Longa de Curto Prazo (*Long Short-Term Memory*)

Esta subseção é baseada no trabalho de Hochreiter e Schmidhuber (1997) o qual propôs as redes LSTM.

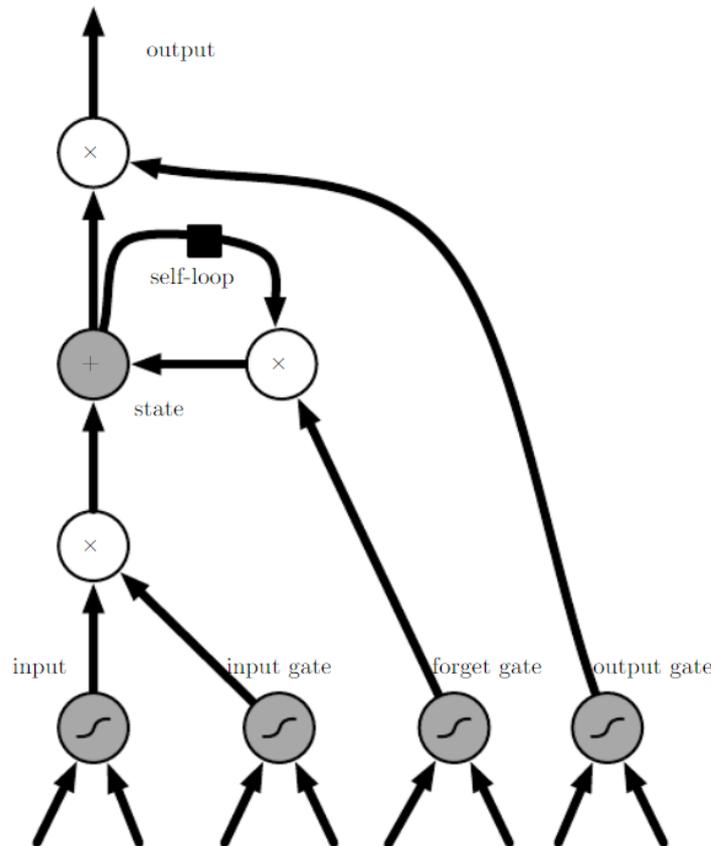
A LSTM, como mencionado anteriormente, foi criada para vencer a explosão ou desaparecimento do gradiente inerente de aplicações em aprendizado profundo. LSTMs podem aprender padrões de 1000 passos discretos no tempo ao propagar o fluxo de erro através do Carrossel de Erro Constante - *Constant Error Carousel* (CEC) dentro de suas unidades especiais, que são formados internamente para propagar o erro às unidades e realizar o treinamento mitigando o desaparecimento do gradiente. Para realizar tal treinamento são utilizadas chaves que "abrem ou fecham" as entradas e saídas de informação das unidades internas da LSTM. As chaves das unidades aprendem a abrir ou fechar conforme necessário para garantir acesso ao fluxo de erro. A LSTM tem uma complexidade temporal computacional para cada passo no tempo e peso de  $O(1)$  e resolve diversos problemas de Inteligência Artificial - *Artificial Intelligence* (AI) nunca antes abordados com êxito por RNNs convencionais.

Para resolver definitivamente as dificuldades apresentadas Hochreiter e Schmidhuber (1997) elaborou uma nova arquitetura composta de **células de memória e unidades "chave"**. A arquitetura LSTM estende o CEC com recursos adicionais projetados para melhorar o aprendizado. Uma chave multiplicativa de entrada para proteger os conteúdos na memória guardados em  $j$  das perturbações de entradas irrelevantes e uma chave multiplicativa de saída para proteger

outras unidades das perturbações de memórias irrelevantes atualmente guardadas em  $j$ , o que resulta na célula de unidade de memória. Cada célula de memória tem uma conexão interna linear em si mesma.

A seguinte Figura 1 ilustra o diagrama de blocos completo de uma unidade LSTM.

Figura 1 – Diagrama de Blocos de uma Célula LSTM.



Fonte: Hochreiter e Schmidhuber (1997).

O problema principal da Rede Neural Recorrente Simples - *Simple Recurrent Neural Network* (SRNN) é que o algoritmo Retro Propagação Através do Tempo - *Back-Propagation Through Time* (BPTT) usado para treiná-las tende a ter seus pesos explodindo ou desaparecendo, o que causa uma grande oscilação nos pesos ou um tempo impraticável de treinamento, respectivamente. Este problema é mitigado na rede LSTM ao se utilizar suas unidades especiais internas de memória e chaveamento para controlar o fluxo de propagação de erro criando o chamado Carrossel de Erro Constante - *Constant Error Carousel* (CEC) que possibilita o aprendizado de longas sequências sem ocorrer o desvanecimento ou a explosão do gradiente.

As equações para cada chave, entrada e saída são descritos abaixo:

**Chave de Esquecimento (*Forget Gate*)** ( $f_i^t$ ) controla o peso do conexão própria interna ajustando o peso de 0 a 1 por uma sigmoide ( $\sigma$ ):

$$f_i^{(t)} = \sigma(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}) \quad (1)$$

Onde:  $x^{(t)}$  = Vetor Atual de Entrada

$h^{(t)}$  = Vetor Atual da Camada Oculta

$b$  = Viés (Bias)

$U$  = Pesos de Entrada

$W$  = Pesos Recorrentes

**Chave de Entrada Externa (*External Input Gate*)** ( $g_i^t$ ) é similar à Equação 1 também com ativação sigmoide de 0 a 1:

$$g_i^{(t)} = \sigma(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}) \quad (2)$$

**Unidade de Estado (*State Unit*)** ( $s_i^t$ ) que é unidade mais importante da célula LSTM, tendo uma conexão interna própria linear controlada pela chave de esquecimento:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}) \quad (3)$$

**Saída (*Output*)** ( $h_i^{(t)}$ ):

$$h_i^{(t)} = \tanh(s_i^{(t)}) q_i^{(t)} \quad (4)$$

**Chave de Saída (*Output Gate*)** ( $q_i^{(t)}$ ) que pode desligar a saída da célula via:

$$q_i^{(t)} = \sigma(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}) \quad (5)$$

É importante ressaltar que o número de pesos das conexões de treinamento de uma LSTM é muito maior do que o de uma ANN comum e segue a seguinte Equação 6:

$$N_{par} = 4 * (nm + n^2 + n) \quad (6)$$

Onde:  $n$  = Número de Neurônios da Camada de Saída.

$m$  = Número de Neurônios da Camada de Entrada.

Apesar de já ter sido desenvolvida há mais de duas décadas a LSTM está sendo muito utilizada nos dias de hoje devido à recém evolução da capacidade computacional que era um empecilho até alguns anos atrás para estes métodos de aprendizagem profunda. Recentemente a LSTM sofreu algumas alterações originando uma nova RNN chamada Unidade Recorrente Chaveada - *Gated Recurrent Unit* (GRU) que é mais simples em sua arquitetura, ocasionando em uma diminuição significativa no tempo de treinamento sem perder a qualidade do aprendizado.

### 2.1.2 Unidades Recorrentes Chaveadas (*Gated Recurrent Unit*)

As redes de Unidade Recorrente Chaveada - *Gated Recurrent Unit* (GRU) são mais simples que as redes de LSTM descritas anteriormente. Ambas tem chaves de controle do fluxo afluente e efluente de informação e propagação de erro, mas a GRU não tem uma célula de memória separada como a da LSTM. A GRU foi proposta por Cho et al. (2014) para fazer cada unidade recorrente capturar adaptativamente as dependências de diferentes escalas no tempo. Como dito por Goodfellow, Bengio e Courville (2016) a principal diferença é que a GRU tem uma única unidade de chaveamento responsável por decidir se esquece ou atualiza sua unidade de estado.

A primeira aplicação da rede proposta por (CHO et al., 2014) foi um codificador-decodificador utilizado em tradução de máquina. O modelo inicial se mostrou melhor do que os existentes, até o momento de sua publicação, para identificação de padrões linguísticos existentes nas frases analisadas. Nesta publicação, o autor especula que apesar de ter criado a nova rede GRU para tradução esta poderia ser implementada em outras aplicações.

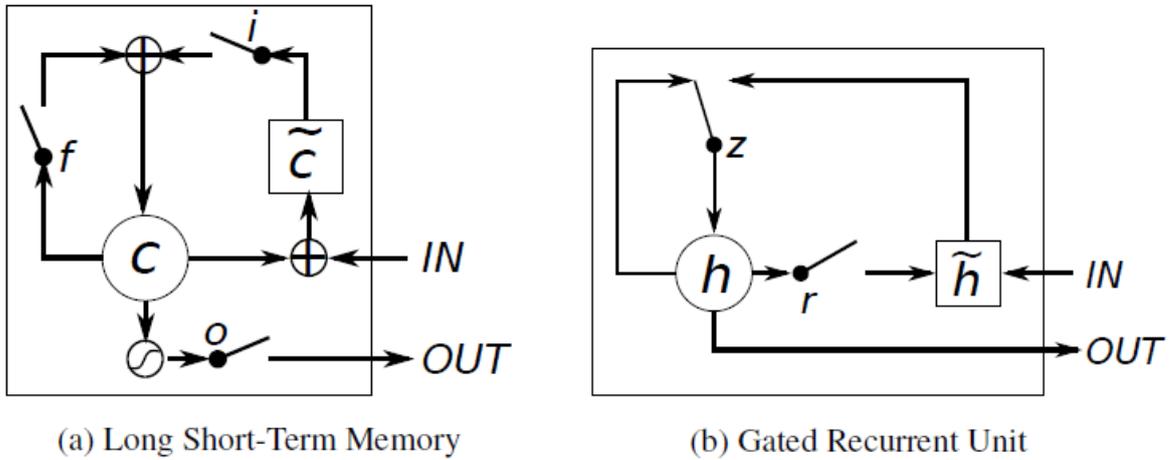
A seguinte Figura 2 ilustra as diferenças entre as arquiteturas das células LSTM e GRU e foi apresentada por Chung et al. (2014).

Então são apresentadas as equações de controle conforme segue:

**Unidade de Estado (*State Unit*) ( $h_i^{(t)}$ ):**

$$h_i^{(t)} = u_i^{(t-1)} h_i^{(t-1)} + (1 - u_i^{(t-1)}) \sigma(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} r_j^{(t-1)} h_j^{(t-1)}) \quad (7)$$

Figura 2 – Comparação LSTM e GRU



(a)  $i$ ,  $f$  e  $o$  são as chaves de entrada, esquecimento e saída, respectivamente.  $c$  e  $\tilde{c}$  denotam a célula de memória e o novo conteúdo da célula de memória. (b)  $r$  e  $z$  são as chaves de esquecimento e atualização, e  $h$  e  $\tilde{h}$  são a ativação anterior e a ativação candidata.

Fonte: Chung et al. (2014).

Onde:  $u$  = Chave de Atualização (*Update Gate*)

$r$  = Chave de Redefinição (*Reset Gate*)

$b$  = Viés (Bias)

$U$  = Pesos de Entrada

$W$  = Pesos Recorrentes

**Chave de Atualização (*Update Gate* -  $u_i^{(t)}$ ):**

$$u_i^{(t)} = \sigma(b_i^u + \sum_j U_{i,j}^u x_j^{(t)} + \sum_j W_{i,j}^u h_j^{(t-1)}) \quad (8)$$

**Chave de Redefinição (*Reset Gate* -  $r_i^{(t)}$ ):**

$$r_i^{(t)} = \sigma(b_i^r + \sum_j U_{i,j}^r x_j^{(t)} + \sum_j W_{i,j}^r h_j^{(t-1)}) \quad (9)$$

Assim como afirmado anteriormente, as redes GRU tem um número menor de parâmetros de peso treináveis, portanto sendo menos exigente computacionalmente e temporalmente. A seguinte Equação 10 descreve o seu cálculo:

$$N_{par} = 3 * (nm + n^2 + n) \quad (10)$$

Onde:  $n$  = Número de Neurônios da Camada de Saída.

$m$  = Número de Neurônios da Camada de Entrada.

Pelos motivos citados nesta seção, como capacidade de aprendizado de longos padrões no tempo, e a diminuição da complexidade de treinamento da GRU em relação às redes LSTM que a mesma foi escolhida para compor o modelo proposto.

## 2.2 Ondaletas (*Wavelets*)

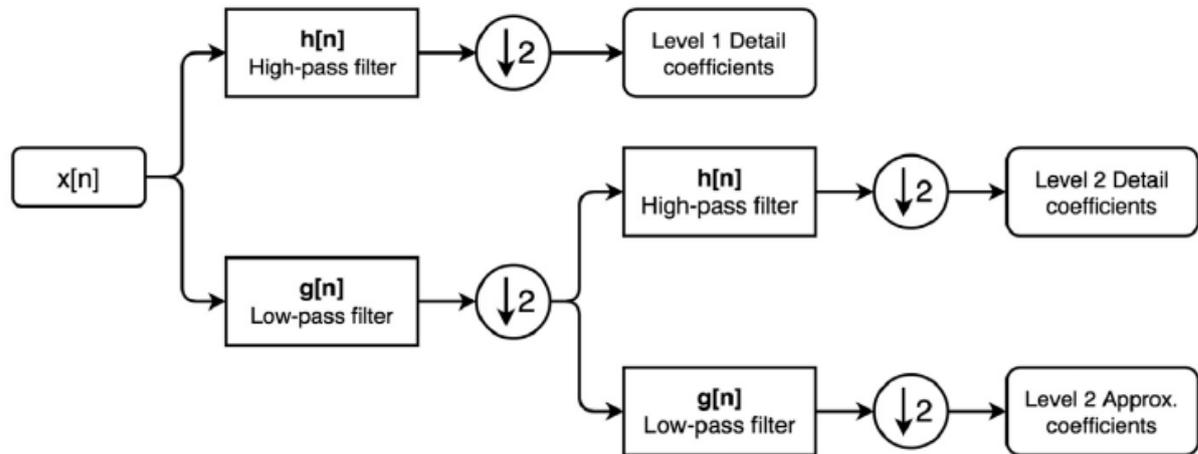
*Wavelets* são pequenas ondas de período e forma flexíveis que podem ser esticadas, ter sua fase invertida e serem deslocadas no tempo para se adequar ao sinal analisado (NOBRE; NEVES, 2019). A seleção do formato da *wavelet* a ser utilizada depende do sinal a ser estudado como dito por Galli, Heydt e Ribeiro (1996).

*Wavelets* podem ser traçadas como um paralelo a tons musicais e sua assimilação e superposição com o correto posicionamento no tempo, com coeficientes para descrever sua compressão ou dilatação e intensidade, podem reconstruir o sinal transformado como se fosse uma musica decomposta em tons. Geralmente, transformadas de *wavelets* são classificadas em duas categorias principais, sendo elas: Transformada Discreta de Ondaletas - *Discrete Wavelet Transforms* (DWT) e Transformada Contínua de Ondaletas - *Continuous Wavelet Transforms* (CWT).

A CWT é baseada em continuamente escalar e deslocar a ondaleta-mãe (*mother wavelet*), que é a *wavelet* original escolhida para a transformada, em diferentes *wavelets* chamadas ondaletas-filhas (*daughter wavelets*). O processo contínuo cria um grande número de dados e coeficientes redundantes, não sendo ideal para sinais com rápidas mudanças ou análise em tempo real. No entanto DWT comprime a informação em coeficientes discretos obtidos através de um processamento do sinal em um algoritmo semelhante a estrutura de um filtro passa-banda como ilustrado na Figura 3, imagem esta que foi apresentada por Nobre e Neves (2019). O sinal representado pelo circulo com uma seta para baixo e o número 2, representa que para cada duas amostras se mantém apenas uma.

*Wavelets* foram concebidas para superar as dificuldades inerentes em se aplicar transformadas de Fourier a sinais não periódicos. Embora a transformada de Fourier seja capaz de expressar as componentes de frequência do sinal analisado em uma série de senos e cossenos, lhe falta a capacidade de determinar mudanças no espectro da frequência de tempos em tempos.

Figura 3 – Decomposição DWT de um sinal em dois níveis.



Fonte: Nobre e Neves (2019).

Outra limitação da transformada de Fourier é que o sinal é considerado periódico na janela de tempo analisada e será repetido exatamente da mesma forma depois de tal período, o que é uma limitação quando o "sinal" a ser analisado não tem nenhuma periodicidade clara (VALENS, 1999).

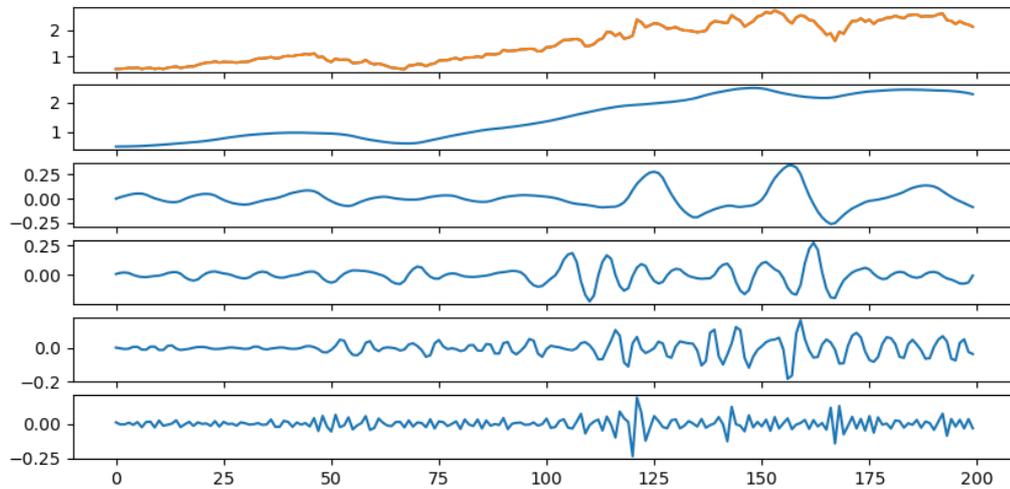
Por outro lado, transformadas de *Wavelets* consideram tempo e frequência em sua análise resultando em um entendimento melhor de onde e quando no sinal cada componente de *wavelets* manifesta seus efeitos. Estas características específicas da transformada de *Wavelets* tornam possível analisar sinais evoluindo no tempo simplesmente formatando as *wavelets* e extraindo novos coeficientes e posições, adicionando-os à análise existente.

A Transformada Empírica de Ondaletas - *Empirical Wavelet Transform* (EWT), é uma outra forma de analisar sinais não periódicos e consiste principalmente em decompor o sinal em diversos passos utilizando a DWT, e então reconstruir sinais a partir de um nível de detalhe a cada vez. Este processo resulta em diversos sinais, que se somados resultam no sinal original reconstruído (ALTAN; KARASU; BEKIROS, 2019). O resultado deste processo é exemplificado na Figura 4. O sinal original em laranja e as *wavelets* empíricas resultantes em azul, das frequências mais baixas até as mais altas.

### 2.2.1 Filtragem de Ruído de Ondaletas (*Wavelet Denoising*)

O algoritmo desenvolvido por Mallat (1999) decompõe o sinal em diversos níveis de filtros passa-alta e passa-baixa, cada nível sendo subsequentemente decomposto para se retirar

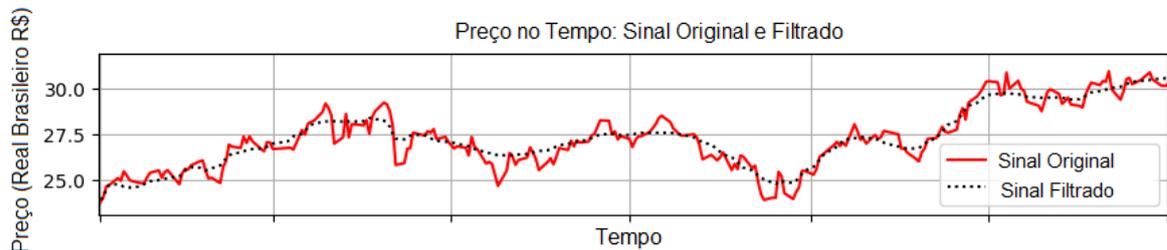
Figura 4 – Transformada Empírica de *Wavelets* da ação PETR4 em 5 componentes.



Fonte: Autoria Própria.

mais detalhes de cada um deles. A transformada de *wavelets* é uma ferramenta poderosa para analisar sinais mesmo em diversas dimensões de dados, e pode também ser utilizada para retirar ruído de sinais como exemplificado por Sardy, Tseng e Bruce (2001). O processo consiste em não considerar os coeficientes das frequências mais altas quando realizada a Transformada Inversa Discreta de Ondaletas - *Inverse Discrete Wavelet Transform* (IDWT) para reconstruir o sinal. A Figura 5 exemplifica os efeitos desta filtragem de ruído em se aplicada a DWT e retirar as frequências mais altas de uma dada ação.

Figura 5 – Exemplo da Filtragem de Ruído DWT aplicada a ação PETR4 no ano 2019.



Fonte: Autoria Própria.

### 2.2.2 Transformadas de Ondaletas (*Wavelets Transform*)

As transformadas de *wavelets* são baseadas em escalar e transladar no tempo as wavelets-mãe (*mother wavelets*) em ondaletas-filhas (*daughter wavelets*) (DAUBECHIES, 1992). Dito isto, a equação que dita o comportamento é descrita na Equação 11 mostrada abaixo:

$$(T^{wav} f)(a,b) = |a|^{-1/2} \int f(t) \psi\left(\frac{t-b}{a}\right) dt \quad (11)$$

Onde:  $a$  = Fator de Escala

$b$  = Fator de Translação no Tempo

$\psi$  = Ondaleta Mãe

Esta Equação 11 descreve o conceito generalizado de se adquirir as ondaletas-filhas baseado em escalar e transladar a ondaleta-mãe.

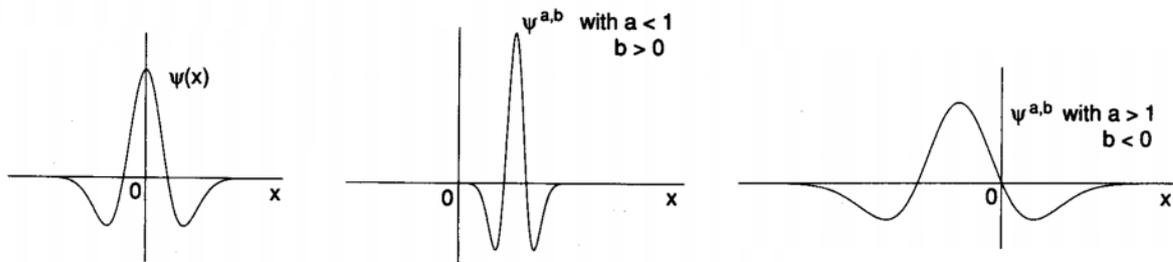
É assumido que a função da ondaleta mãe satisfaz a Equação 12 abaixo:

$$\int \psi(t) dt = 0 \quad (12)$$

Um exemplo visual da escala e translação no tempo das *wavelets* é apresentado abaixo utilizando a segunda derivada da Função Gaussiana. Abaixo podemos ver a Equação 13 que a descreve e os exemplos de escala na Figura 6 aplicando a Equação 11 na Equação 13:

$$\psi(t) = (1 - t^2) e^{-t^2/2} \quad (13)$$

Figura 6 – Exemplo de escala e translação no tempo - Wavelet Laplaciano da Gaussiana.



Fonte: Daubechies (1992).

### 2.2.3 Transformada Contínua de Ondaletas (*Continuous Wavelet Transform*)

Como dito por Daubechies (1992), a CWT é baseada em variar continuamente os parâmetros de dilatação e translação, respectivamente  $a$  e  $b$ , sobre o domínio  $\mathbb{R}$  (com a condição que  $a \neq 0$ ). A equação de *wavelet*  $\psi_{a,b}$  é dada pela Equação 11. A função pode ser reconstruída por meios da equação:

$$f = C_\psi^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{da db}{a^2} \langle f, \psi^{a,b} \rangle \psi^{a,b} \quad (14)$$

Onde:

$$\psi^{a,b}(x) = |a|^{-1/2} \psi\left(\frac{x-b}{a}\right) \quad (15)$$

e  $\langle , \rangle$  denota o produto interno  $L^2$ . A constante  $C_\psi$  depende apenas de  $\psi$  e é dada pela equação:

$$C_\psi = 2\pi \int_{-\infty}^{\infty} d\xi |\psi(\xi)|^2 |\xi|^{-1} \quad (16)$$

A Equação 14 pode ser interpretada como uma forma de reconstruir a função uma vez que sua transformada de *wavelet*  $T^{wav}$  é conhecida ou como uma forma de escrever  $f$  como uma superposição de *wavelets*  $\psi^{a,b}$ , cujos coeficientes são dados pela transformada de *wavelets* de  $f$ .

### 2.2.4 Transformada Discreta de Ondaletas (*Discrete Wavelet Transform*)

A DWT, de acordo com Daubechies (1992), toma uma abordagem similar a da CWT, mas considera apenas variações discretas dos parâmetros  $a$  e  $b$ . Para  $a$  toma-se números positivos e negativos elevados a um parâmetro de escala fixo  $a > 1$ , que é  $a = a_0^m$ , e o parâmetro  $m$  determina a largura das *wavelets*.

O parâmetro de translação  $b$  também depende do parâmetro  $m$ . *Wavelets* estreitas (alta frequência) são transladas por pequenos passos para cobrir todo a faixa no tempo, enquanto *wavelets* largas (baixa frequência) são transladas por passos maiores. Portanto para discretizar  $b$  é escolhido  $b = nb_0 a_0^m$ , onde  $b_0 > 0$  é fixo, e  $n \in \mathbb{Z}$ . Então podemos descrever as *wavelets* classificadas discretas como:

$$\psi_{m,n}(x) = a_0^{-m/2} \psi(a_0^m x - nb_0) \quad (17)$$

Então assume-se um  $\psi$  razoável e  $a_0$  e  $b_0$  apropriados a reconstrução da função pode ser feita por:

$$f = \sum_{m,n} \langle f, \psi_{m,n} \rangle \widetilde{\psi}_{m,n} \quad (18)$$

Que pode ser traduzido como sendo a função de reconstrução a soma de todos os coeficientes de *wavelet*  $\langle f, \psi_{m,n} \rangle$ , para cada  $m$  e  $n$ , sobre a inversa da *wavelet*  $\widetilde{\psi}_{m,n}$ .

### 2.3 Tipos de Ondaletas

A seguir serão mostrados alguns tipos de *wavelets* mais comuns na utilização com as DWTs. Sua aplicação conforme dito anteriormente depende do sinal a ser analisado, sendo cada formato mais adequado para uma análise do que a outra, não havendo um que se destaque em todas as aplicações.

#### 2.3.1 Ondaletas Daubechies

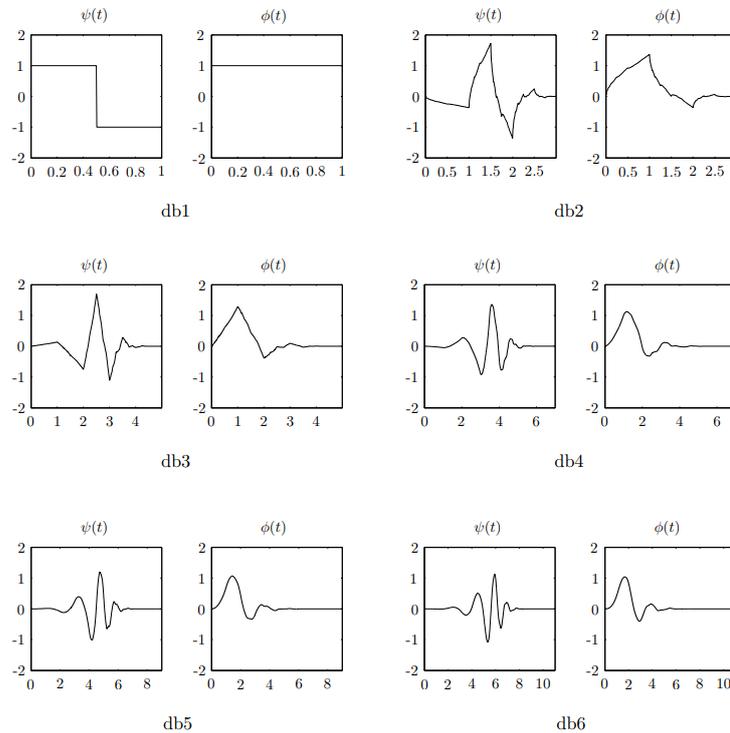
A família de ondaletas Daubechies é chamada assim a partir do nome de seu criador Ingrid Daubechies que inventou as ondaletas ortonormais compactas suportadas tornando a análise de *wavelets* possível em tempo factível. A *wavelet* de primeira ordem de Daubechies é conhecida como *Wavelet* de Haar (db1), e relembra uma função degrau (MERRY, 2005). A Figura 7 ilustra alguma de suas funções.

Ordens maiores das funções de Daubechies não são tão facilmente descritas analiticamente, pois a ordem denota o número de pontos de esmaecimento, ou seja, o números de zeros da função da ondaleta. Isto está levemente atrelado ao número de oscilações da ondaleta. Quanto maior o grau da função da *wavelet* melhor é sua localização na decomposição da frequência.

#### 2.3.2 Ondaletas Coiflets

As Coiflets também foram criadas por Ingrid Daubechies por solicitação de Ronald Coifman. *Wavelets* de Coiflets são ondaletas ortogonais compactas suportadas com o maior número de pontos de esmaecimento para ambas a função *wavelet* e de escala para uma determinada

Figura 7 – Funções de *wavelets* e escala para as funções Daubechies de ordem 1 a 6



Fonte: Merry (2005).

largura de suporte. As Coiflets são mais simétricas e tem mais pontos de esmaecimento que as *wavelets* de Daubechies (MERRY, 2005). A Figura 8 ilustra alguma de suas funções.

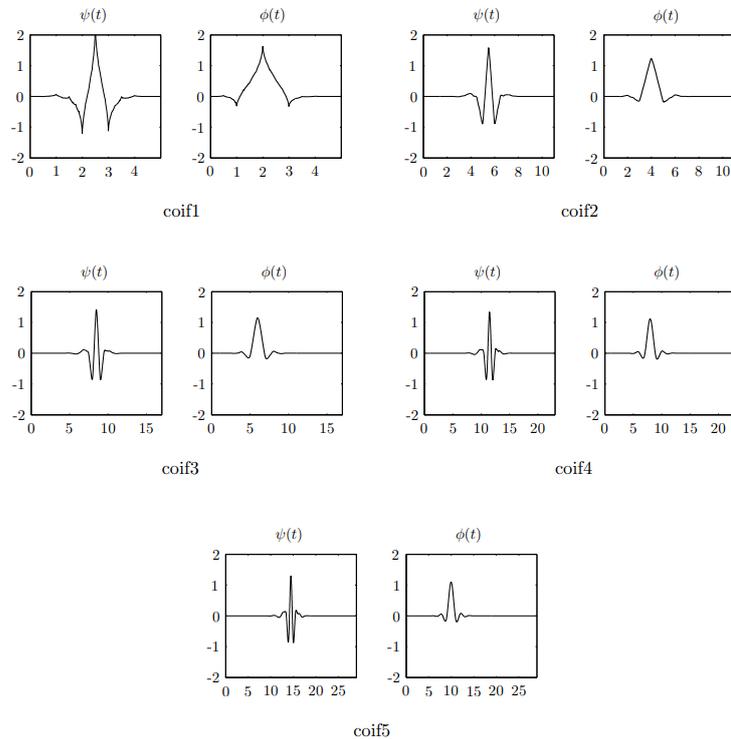
### 2.3.3 Ondaletas Symlets

As Symlets, assim como a anterior, são ortogonais e compactamente suportadas. Também foram propostas por Ingrid Daubechies modificando a família Daubechies. Suas características são praticamente idênticas as das Daubechies, mas são praticamente totalmente simétricas (MERRY, 2005). A Figura 9 ilustra alguma de suas funções.

### 2.3.4 Ondaletas Bi-ortogonais

A família de *wavelets* bi-ortogonais contém ondaletas compactamente suportadas bi-ortogonais. Com estas a exata simetria e reconstrução são possíveis utilizando filtros de resposta a impulsos finitos, o que é impossível para os filtros ortogonais (exceto para *wavelet* de Haar). A simetria significa que os filtros tem fase linear.

Figura 8 – Funções de *wavelets* e escala para as funções Coiflets de ordem 1 a 5.



Fonte: Merry (2005).

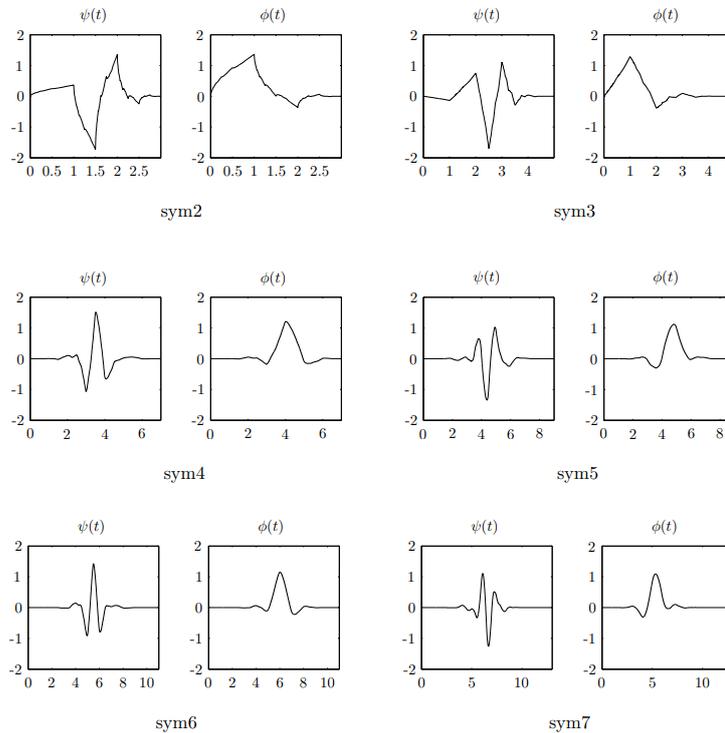
A família bi-ortogonal usa diferentes funções de análise e síntese do sinal para as funções *wavelets* e de escala. A IDWT utiliza a família bi-ortogonal de síntese para a análise, e vice-versa. A Figura 10 ilustra alguma de suas funções.

## 2.4 Análise Técnica no Mercado de Ações

A análise técnica na teoria do investimento em ações é uma análise metodológica para prever a direção dos preços através de pesquisa nos dados passados da ação (ZHANG et al., 2019) e evoluiu do desejo de se entender o comportamento do mercado ao desenvolvimento de diversas fórmulas matemáticas e algoritmos para descrever e extrair informação das ações analisadas.

Indicadores Técnicos são os ditos modelos matemáticos para se interpretar a direção futura e tendências das ações utilizando o histórico passado. Estes indicadores são principalmente utilizados por investidores de curto-prazo na tarefa de se identificar estratégias de operação para se tomar decisões no melhor momento possível. Existem dezenas destes indicadores disponíveis na literatura e os mesmos são geralmente utilizados em conjunto para formar uma estratégia

Figura 9 – Funções de *wavelets* e escala para as funções Symlets de ordem 1 a 6.



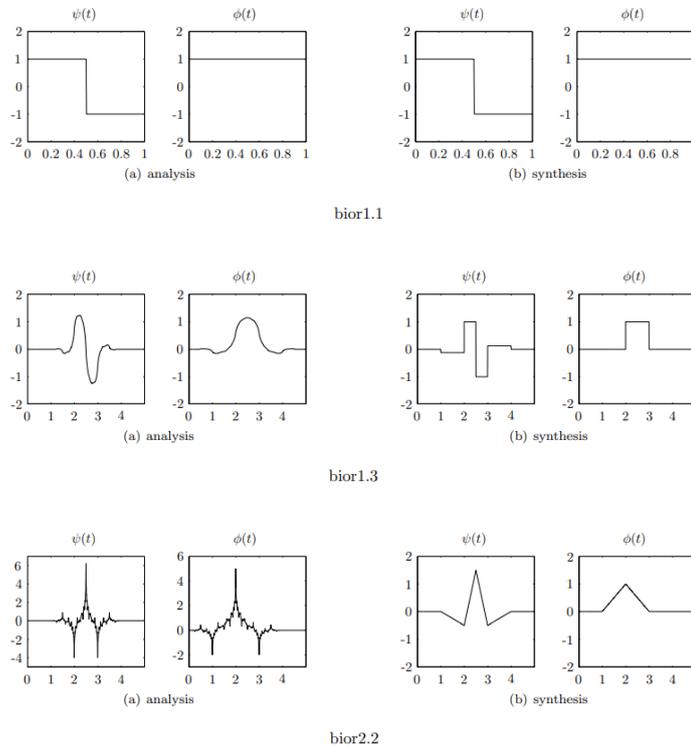
Fonte: Merry (2005).

de operação. Isto deixa claro que a escolha dos indicadores a serem associados é extremamente importante para o resultado da estratégia (ULLAH et al., 2019).

Em seu artigo Murphy (1999)(citado em (PIMENTA et al., 2018)) afirma que existem diversos tipos de indicadores, sendo:

- a) Seguidores de Tendência: os indicadores desta categoria identificam o movimento principal dos preços do ativo num determinado período. São exemplos desta categoria: Média Móvel Simples, Média Móvel Exponencial, Canais de Donchian e o Ativador Hilo.
- b) Osciladores: os indicadores desta categoria monitoram as variações do preço de um ativo em uma certa janela de avaliação para identificar possíveis pontos de reversão. Exemplos: Oscilador de Chaikin, Oscilador de Volatilidade de Chaikin e Williams %R.
- c) Sistemas de Bandas: São constituídos de três curvas desenhadas ao redor dos preços. Estas curvas são calculadas a partir de uma distância particular de uma média móvel. A banda intermediária é geralmente uma média móvel simples, e o intervalo entre as bandas é determinado pela volatilidade do preço. Quando não se

Figura 10 – Funções de *wavelets* e escala para as funções Bi-ortogonais de ordem 1 a 3.



Fonte: Merry (2005).

há uma tendência clara, a convenção é de que se deve vender caso o preço esteja acima da banda superior, ou comprar caso o preço esteja abaixo da banda inferior. Exemplos: Canal de Keltner, Bandas de Bollinger e Oscilador de Bollinger.

- d) Identificadores de Divergências: Estes indicadores são baseados no princípio que toda tendência sofre correções. A divergência ocorre quando comparando o comportamento do indicador em relação ao movimento do preço do ativo. Exemplos: Índice de Distribuição de Acúmulo, Volume Balanceado, Índice de Força Relativa.

Abaixo serão descritas equações e algoritmos de cada indicador técnico disponível sendo estes separados por seus tipos: Inércia, Volatilidade, Volume, Tendência e outros.

Alguns indicadores abaixo utilizam a Média Móvel Simples - *Simple Moving Average* (SMA) (BULKOWSKI, 2011) que é calculada tomando a média de um certo período no passado, seu cálculo é generalizado na seguinte equação:

$$SMA_n = \frac{A_1 + A_2 + A_3 + \dots + A_n}{n} \quad (19)$$

Onde:  $A_n$  = A preço do ativo em um determinado instante  $n$

$n$  = Número total de períodos

Como a SMA, existem indicadores que utilizam a Média Móvel Exponencial - *Exponential Moving Average*, calculada conforme abaixo (LAWRANCE; LEWIS, 1977):

$$EMA = Preço de Fechamento * Multiplicador + EMA_{t-1} * (1 - Multiplicador) \quad (20)$$

O multiplicador nesta equação dita a sensibilidade às mudanças no preço, geralmente é utilizado o valor 2.

#### 2.4.1 Indicadores de Inércia

- a) **Oscilador Incrível - *Awesome Oscillator* (AO):** AO é baseado na subtração de uma SMA de 5 dias de uma SMA de 34 dias. As SMAs são calculadas nos preços médios entre os máximos e mínimos de cada dia. AO é geralmente aplicado para identificar inversões em tendências (WILLIAMS, B. M., 1998).

$$PreçoMédio = \frac{Máximo + Mínimo}{2} \quad (21)$$

$$AO = SMA_5(PreçoMédio) - SMA_{34}(PreçoMédio) \quad (22)$$

- b) **Média Móvel Adaptiva de Kaufman - *Kaufman's Adaptive Moving Average* (KAMA):** KAMA é uma média móvel projetada para levar em consideração oscilações no mercado e sua volatilidade. KAMA segue de perto os preços quando a volatilidade está baixa e ajusta sua posição quando os preços tem movimentos mais bruscos voltando a seguir o preço de uma maior distância. Este pode ser comparado com uma média móvel robusta com redução de ruído (KAUFFMAN; MACREADY, 1995). O indicador KAMA é calculado em três passos:

Taxa de eficiência - *Efficiency Ratio* (ER)

$$ER = \frac{|PreçoFinal - PreçoFinal_n|}{\sum_0^{x=-n} |PreçoFinal - PreçoFinal_n|} \quad (23)$$

Constante de suavização - *Smoothing Constant* (SC)

$$SC = [ER * (EMA_{Rápida} - EMA_{Lenta}) + EMA_{Lenta}]^2 \quad (24)$$

Onde:  $EMA_{Rápida}$  = Média Móvel Exponencial de 2 dias;  
 $EMA_{Lenta}$  = Média Móvel Exponencial de 30 dias.

$$KAMA = KAMA_{t-1} + SC * (PreçoFinal - KAMA_{t-1}) \quad (25)$$

- c) **Taxa de Mudança - *Rate of Change* (ROC):** O indicador ROC é um oscilador de inércia que mede o percentual de mudança entre o preço de  $n$  períodos. O cálculo do ROC compara o preço da ação  $n$  períodos no passado com o preço atual. O resultado forma uma oscilação abaixo e acima da linha zero (PRING, 2014).

$$ROC = \frac{PreçoFinal - PreçoFinal_n}{PreçoFinal_n} * 100 \quad (26)$$

Onde:  $PreçoFinal$  = Preço de Fechamento do dia

$PreçoFinal_n$  = Preço de fechamento  $n$  períodos atrás

- d) **Índice de Força Relativa - *Relative Strength Index* (RSI):** O RSI é um indicador de inércia que mede quanto o preço mudou na janela de tempo recente, e tem a capacidade de indicar sobre-compra ou sobre-venda de uma ação comparando se o indicador está acima de 70 ou abaixo de 30, respectivamente (WILDER, 1978). Este é calculado seguindo os dois passo abaixo:

$$RSI_{PassoUm} = 100 - \frac{100}{1 + \frac{GanhoMédio}{PerdaMédia}} \quad (27)$$

O ganho e a perda média são calculados examinando-se o passado, geralmente 14 dias, se o preço fechou mais alto ou baixo que o imediatamente anterior, e extraíndo a média de lucro ou prejuízo destes dois grupos. Isto assegura que se na última janela de tempo analisada os preços apenas subiram o indicador chega próximo a 100, e se apenas caíram se aproxima de 0.

Uma vez que existam 14 dias de dados, o passo dois pode ser calculado como:

$$RSI_{Passo2} = 100 - \frac{100}{1 + \frac{GanhoMédioAnterior - Ganhoatual}{PerdaMédiaAnterior - PerdaAtual}} \quad (28)$$

- e) **Oscilador Estocástico - *Stochastic Oscillator* (SO):** O SO é composto de dois sinais representando um oscilador que segue a velocidade e a inércia do preço

(LANE, s.d.). É geralmente utilizado para identificar reversões e pode ser calculado como:

$$\%K = \frac{\text{PreçoFinalAtual} - \text{MínimodoPeríodo}}{\text{MáximodoPeríodo} - \text{MínimodoPeríodo}} * 100 \quad (29)$$

$$\%D = SMA_3(\%K) \quad (30)$$

Onde: *MínimodoPeríodo* = Preço mínimo no período analisado;  
*MáximodoPeríodo* = Preço máximo do período analisado;  
 Período = Geralmente 14 dias.

- f) **Indicador Williams %R - Williams %R Indicator (W %R):** Indicador Williams %R - *Williams %R Indicator*, ou apenas W %R, é um oscilador de inercia muito semelhante ao SO, sendo a inversa do supracitado indicador, comparando o preço de fechamento com o preço máximo do período, e corrigindo o sinal multiplicando o resultado por -100. Este oscila de 0 a -100, sendo valores maiores que -20 considerados sobre-compra e abaixo de -80 sobre-venda (WILLIAMS, L. R., 1979).

$$W\%R = \frac{\text{MáximodoPeríodo} - \text{PreçoFinalAtual}}{\text{MáximodoPeríodo} - \text{MínimodoPeríodo}} * -100 \quad (31)$$

Onde: *MínimodoPeríodo* = Preço mínimo no período analisado;  
*MáximodoPeríodo* = Preço máximo do período analisado;  
 Período = Geralmente 14 dias.

- g) **Oscilador Definitivo - Ultimate Oscillator (UO):** O UO foi desenvolvido para superar as dificuldades inerentes de um oscilador ao levar-se em consideração dados de mais de uma janela no tempo, e incorpora então três janelas de tempo em seu sinal. Este identifica sinais de compra ou venda baseado na divergência de tendência (WILLIAMS, L., 1985). O mesmo é calculado conforme segue abaixo:

$$UO = \frac{4 * SMA_7 + 2 * SMA_{14} + SMA_{28}}{7} * 100 \quad (32)$$

- h) **Índice de Força Verdadeira - *True Strength Index* (TSI):** O TSI é um oscilador de inércia baseado em uma dupla suavização das mudanças nos preços. Assim como outros indicadores mencionados anteriormente, este indica compra ou venda baseado em divergências de tendência (BLAU, s.d.).

$$PC = PreçoFinal - PreçoFinal_{t-1} \quad (33)$$

Primeira Suavização - *First Smoothing* (FS)

$$FS = EMA_{25}(PC) \quad (34)$$

Segunda Suavização - *Second Smoothing* (SS)

$$SS = EMA_{13}(FS) \quad (35)$$

#### 2.4.2 Indicadores de Volume

- a) **Fluxo Monetário de Chaikin - *Chaikin Money Flow* (CMF):** O CMF é a base do indicador Índice de Distribuição de Acúmulo - *Accumulation Distribution Index* (ADI), e considera um período de tempo de geralmente 20 dias. Este é utilizado para identificar mudanças no fluxo monetário de um ativo, e oscila acima e abaixo da linha zero, indicando pressão de compra ou venda (NARCOUZI, 2000). O mesmo é calculado como segue:

Multiplicador de Fluxo Monetário - *Money Flow Multiplier* (MFM)

$$MFM = \frac{(PreçoFinal - Mínimo) - (Máximo - PreçoFinal)}{(Máximo - Mínimo)} \quad (36)$$

Volume de Fluxo Monetário - *Money Flow Volume* (MFV)

$$MFV = MFM \times Volume \quad (37)$$

$$CMF = \frac{\sum_0^{x=-20dias} MFV}{\sum_0^{x=-20dias} Volume} \quad (38)$$

- b) **Índice de Distribuição de Acúmulo - *Accumulation Distribution Index* (ADI):** ADI é um indicador projetado para medir o acúmulo de dinheiro entrando ou saindo de um ativo. Este é utilizado principalmente para confirmar reversões indicadas por aliá-lo a outros indicadores já que ele avalia se o ativo está sob uma

pressão forte de compra ou venda (NARCOUZI, 2000). O mesmo é calculado usando a Equação 36 e 37 do CMF com um terceiro passo descrito abaixo:

$$ADI = ADI_{t-1} + MFV \quad (39)$$

- c) **Índice de Fluxo Monetário - Money Flow Index (MFI):** O MFI pode ser descrito como um RSI ponderado pelo volume. O fluxo monetário fica acima de zero se o Preço Típico - *Typical Price* (TP) cresce e abaixo se o TP cai. É geralmente aplicado para identificar reversões e extremos sendo utilizado para identificar sinais de compra e venda (QUONG; SOUDACK, 1989). Os passos seguintes são aplicados para calculá-lo:

Preço Típico - *Typical Price* (TP)

$$TP = (Máximo + Mínimo + PreçoFinal)/3 \quad (40)$$

Fluxo Monetário Cru - *Raw Money Flow* (RMF)

$$RMF = TP * Volume \quad (41)$$

Taxa de Fluxo Monetário - *Money Flow Ratio* (MFR)

$$MFR = \frac{\text{Fluxo Monetário Positivo de 14 dias}}{\text{Fluxo Monetário Negativo de 14 dias}} \quad (42)$$

$$MFI = 100 - 100/(1 + MFR) \quad (43)$$

- d) **Volume Balanceado - On-Balance Volume (OBV):** O indicador OBV calcula o volume total assumindo para cada dia um valor positivo ou negativo dependendo se o último preço foi mais alto ou baixo que o anterior. É construído para indicar divergências em tendências e sinais de compra e venda (GRANVILLE, 1976). É calculado como segue:

$$OBV = OBV_{t-1} + \begin{cases} Volume, & \text{if } PreçoFinal > PreçoFinal_{t-1} \\ 0, & \text{if } PreçoFinal = PreçoFinal_{t-1} \\ -Volume, & \text{if } PreçoFinal < PreçoFinal_{t-1} \end{cases} \quad (44)$$

- e) **Índice de Força - Force Index (FI):** O FI indica quão forte a tendência atual é avaliando sua extensão direção e volume, criando um oscilador. Geralmente é utilizada para identificar pontos de mudança em tendências (ELDER, 1993). Seu cálculo é descrito abaixo:

$$FI_1 = PreçoFinal - PreçoFinal_{t-1} * Volume \quad (45)$$

$$FI_{13} = EMA_{13}(FI_1) \quad (46)$$

- f) **Facilidade de Movimento - Ease of Movement (EoM):** O EoM destaca o relacionamento entre preço e volume, sendo utilizado para indicar a força de uma tendência (ARMS JR, 1990). Este é construído seguindo as seguintes equações:

Distância Movida - *Distance Moved (DM)*

$$DM = \frac{Máximo + Mínimo}{2} - \frac{Máximo_{t-1} + Mínimo_{t-1}}{2} \quad (47)$$

Relação Caixa - *Box Ratio (BR)*

$$BR = \frac{\frac{Volume}{Escala}}{Máximo - Mínimo} \quad (48)$$

A escala pode variar de 1000 a 10000000 dependendo do volume no período.

$$EoM = \frac{DM}{BR} \quad (49)$$

- g) **Tendência Volume-Preço - Volume-Price Trend (VPT):** O indicador VPT correlaciona o preço e volume para expressar indicações de tendência baseado em valores previamente calculados. É utilizado para detectar divergências em tendências no mercado de ações (DORMEIER, 2011). A equação se dá como segue:

$$VPT = VPT_{t-1} + Volume * \frac{PreçoFinal - PreçoFinal_{t-1}}{PreçoFinal_{t-1}} \quad (50)$$

- h) **Índice de Volume Negativo - Negative Volume Index (NVI):** O NVI é projetado para indicar mudanças cumulativas no volume para decidir quando o "dinheiro especialista" está ativo, que é o dinheiro investido por pessoas com conhecimento sobre o ativo ou o mercado (DI LORENZO, 2013). A linha de sinal é formada pelos passos abaixo:

- O indicador inicia em 1000;
- Se o preço cair, adiciona-se o percentual de mudança ao NVI cumulativo;
- Aplica-se uma EMA de 255 dias no indicador NVI.

### 2.4.3 Indicadores de Volatilidade

- a) **Bandas de Bollinger - *Bollinger Bands* (BB):** As Bandas de Bollinger - *Bollinger Bands* são utilizadas para indicar compra e venda ao se avaliar se o preço atual toca ou não as bandas inferior e superior, respectivamente. Mais volatilidade no preço da ação resulta em bandas mais largas, e menos volatilidade bandas mais estreitas. O centro serve como guia para as bandas superior e inferior (BOLLINGER, 1992). As bandas são calculadas conforme as equações abaixo:

Linha Central das Bandas de Bollinger - *Bollinger Bands Middle Line* (BBML)

$$BBML = SMA_{20}(PreçoFinal) \quad (51)$$

Banda Superior das Bandas de Bollinger - *Bollinger Bands Upper Band* (BBUB)

$$BBUB = SMA_{20}(PreçoFinal) + 2 * \sigma_{20}(PreçoFinal) \quad (52)$$

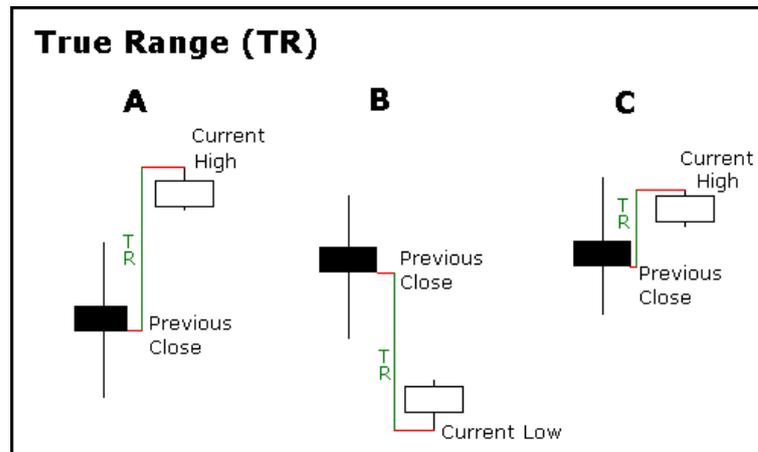
Banda Inferior das Bandas de Bollinger - *Bollinger Bands Lower Band* (BBLB)

$$BBLB = SMA_{20}(PreçoFinal) - 2 * \sigma_{20}(PreçoFinal) \quad (53)$$

Onde:  $\sigma_{20}$  = Desvio Padrão de 20 dias

- b) **Varição Verdadeira Média - *Average True Range* (ATR):** O indicador ATR indica volatilidade na ação avaliada analisando a volatilidade que os outros indicadores falham em levar em consideração em seus resultados. Se o indicador é apenas baseado na faixa máximo-mínimo o mesmo falha em capturar a volatilidade no espaço entre eles. Este indicador é adequado para indicar volatilidade absoluta e não a tendência da direção (WILDER, 1978). O processo para construir o indicador é descrito abaixo:

Figura 11 – Exemplo de métodos do ATR



Fonte: Wilder (1978).

Primeiro deve-se calcular o Variação Verdadeira - *True Range* (TR) em uma das três possibilidades ilustradas na Figura 11:

- Método A

$$TR = Máximo - Mínimo \quad (54)$$

- Método B

$$TR = |Máximo - PreçoFinal_{t-1}| \quad (55)$$

- Método C

$$TR = |Mínimo - PreçoFinal_{t-1}| \quad (56)$$

Então o indicador é calculado pela equação:

$$ATR = \frac{ATR_{t-1} * 13 + TR}{14} \quad (57)$$

- c) **Canais de Keltner - *Keltner Channels* (KC):** As Canais de Keltner - *Keltner Channels* se comportam em uma maneira muito similar às Bandas de Bollinger mas usam EMAs e o ATR ao invés de SMAs, Equação 57, é usado para calcular os canais superior e inferior (KELTNER, 1960). As equações para cálculo de cada elemento seguem abaixo:

Linha Central de Keltner - *Keltner Middle Line* (KML)

$$KML = EMA_{20}(PreçoFinal) \quad (58)$$

Banda Superior de Keltner - *Keltner Upper Band* (KUB)

$$KUB = EMA_{20}(PreçoFinal) + 2 * ATR(PreçoFinal 10 dias) \quad (59)$$

Banda Inferior de Keltner - *Keltner Lower Band (KLB)*

$$KLB = EMA_{20}(PreçoFinal) - 2 * ATR(PreçoFinal 10 dias) \quad (60)$$

- d) **Canais de Donchian - *Donchian Channels (DC)***: Os Canais de Donchian - *Donchian Channels (DC)* são três linhas geradas por cálculos de médias móveis que compõe um indicador formado por bandas superior, inferior e linha central. O objetivo do indicador é identificar tendências (DONCHIAN, 1957). Os canais são determinados seguindo as seguintes equações:

Canal Superior de Donchian - *Donchian Upper Channel (DUC)*

$$DUC = \text{Máximo Preço no Período avaliado} \quad (61)$$

Canal Inferior de Donchian - *Donchian Lower Channel (DLC)*

$$DLC = \text{Mínimo Preço no Período Avaliado} \quad (62)$$

Linha Central de Donchian - *Donchian Middle Line (DML)*

$$DML = \frac{DUC + DLC}{2} \quad (63)$$

#### 2.4.4 Indicadores de Tendência

- a) **Média Móvel de Convergente e Divergente - *Moving Average Convergence Divergence (MACD)***: O indicador MACD é um seguidor de tendência baseado em duas linhas. Uma baseada em uma EMA de 9 dias e outra baseada na subtração de uma EMA de 12 dias de uma EMA de 26 dias. MACD é não lastreado, portanto sendo possível identificar sinais de sobre-compra ou sobre-venda. Os valores flutuam acima e abaixo da linha zero e mudanças na direção podem ser interpretadas como mudanças na tendência atual (APPEL, 1985). As equações das linhas seguem abaixo:

Linha MACD:

$$MACD = EMA_{12}(PreçoFinal) - EMA_{26}(PreçoFinal) \quad (64)$$

Linha de Sinal - *Signal*:

$$Signal = EMA_9(PreçoFinal) \quad (65)$$

- b) **Índice Médio Direcional - Average Directional Index (ADX):** O ADX é composto por duas linhas de sinal: Indicador Direcional Negativo - *Minus Directional Indicator* (-DI) e Indicador Direcional Positivo - *Plus Directional Indicator* (+DI). Ambos são derivados de médias suavizadas para extrair a direção da tendência. Estes dois indicadores são frequentemente chamados de Indicador Direcional de Movimento - *Directional Movement Indicator* (DMI) quando combinados. O uso destes três indicadores em conjunto dá corpo ao ADX (WILDER, 1978). Os passos de construção deste indicador são descritos abaixo:

$$+DM = \begin{cases} \text{Máximo Atual} - \text{Máximo Anterior}, & \text{se Máximo atual} - \text{Máximo Anterior} > 0 \\ 0, & \text{se Máximo Atual} - \text{Máximo Anterior} \leq 0 \end{cases} \quad (66)$$

$$-DM = \begin{cases} \text{Mínimo Anterior} - \text{Mínimo Atual}, & \text{se Mínimo Anterior} - \text{Mínimo Atual} > 0 \\ 0, & \text{se Mínimo Anterior} - \text{Mínimo Atual} \leq 0 \end{cases} \quad (67)$$

Operação de Suavização - *Smoothing Operation*

$$+ / - DM_{Suavizado} = \sum_{t=1}^{14} DM - \left( \frac{\sum_{t=1}^{14} DM}{14} + DM_{Atual} \right) \quad (68)$$

$$+ DI = \frac{+DM_{Suavizado}}{ATR} * 100 \quad (69)$$

$$- DI = \frac{-DM_{Suavizado}}{ATR} * 100 \quad (70)$$

Então:

$$ADX = \frac{ADX_{t-1} * 13 + ADX}{14} \quad (71)$$

- c) **Vortex:** O indicador Vortex pode ser utilizado para sinalizar mudanças no movimento da tendência quando as duas linhas que o compõe criam uma zona de instabilidade perto da linha zero (BOTES; SIEPMAN, 2010). As seguintes equações são usadas para criar as linhas:

$$+ VM = |\text{Máximo Atual} - \text{Mínimo Anterior}| \quad (72)$$

$$- VM = |\text{Mínimo Atual} - \text{Máximo Anterior}| \quad (73)$$

$$+ VM14 = \sum_{t=1}^{14} +VM \quad (74)$$

$$- VM14 = \sum_{t=1}^{14} -VM \quad (75)$$

$$TR14 = \sum_{t=1}^{14} TR \quad (76)$$

Então normalizamos as linhas através de:

$$+ VI14 = \frac{+VM14}{TR14} \quad (77)$$

$$- VI14 = \frac{-VM14}{TR14} \quad (78)$$

- d) **Média Exponencial Tripla - *Triple Exponential Average* (TRIX):** TRIX é um indicador de tendência que mostra o percentual da taxa de mudança de uma média móvel triplamente suavizada. Os sinais gerados por este são bem similares aos do MACD, e servem como indicações de tendências de crescimento ou queda (HUTSON, 1983). A equação segue abaixo:

– 1.

$$\text{PrimeiroEstágio} = EMA_{15}(\text{PreçoFinal}) \quad (79)$$

– 2.

$$\text{SegundoEstágio} = EMA_{15}(\text{PrimeiroEstágio}) \quad (80)$$

– 3.

$$\text{TerceiroEstágio} = EMA_{15}(\text{SegundoEstágio}) \quad (81)$$

– 4.

$$TRIX = \% \text{ de mudança em 1 período do Terceiro Estágio} \quad (82)$$

- e) **Índice de Massa - *Mass Index* (MI):** O MI usa a faixa de máximo-mínimo para identificar reversões nas tendências baseado na expansão das faixas. É baseado

em EMAs e a soma de uma EMA composta para indicar mudanças na direção de uma tendência (DORSEY, 1995). A seguinte equação dita seu comportamento:

$$MI = \sum_{t=1}^{25} \frac{EMA_9(PreçoFinal)}{EMA_9(EMA_9(PreçoFinal))} \quad (83)$$

- f) **Índice de Canal de Mercadoria - Commodity Channel Index (CCI):** CCI é um indicador versátil que pode ser usado para identificar uma nova tendência ou avisar sobre condições extremas, e mede a correlação entre o preço atual e a média dos preços (LAMBERT, 1983). É calculado como segue:

$$CCI = \left( \frac{TP - SMA_{20}(TP)}{0.015 * DesvioMédio} \right) \quad (84)$$

A equação do CCI usa a equação do Preço Típico - *Typical Price* (TP) previamente apresentada como Equação 40.

- g) **Oscilador de Preço Distendente - Detrended Price Oscillator (DPO):** DPO extrai um sinal de tendência do preço e torna mais fácil para se identificar ciclos. DPO é baseada em uma média móvel deslocada, portanto não levando em consideração o valor atual. No entanto, o alinhamento com os dados mais recentes não é relevante, pois o DPO não é um indicador de inércia. Pelo contrário, DPO é utilizado para identificar ciclos de máximos/mínimos e estimar a extensão do ciclo (EHLERS, 2000). A equação do indicador é:

$$DPO = Preço_{t-(n/2+1)} - SMA_n(PreçoFinal) \quad (85)$$

Onde:  $n$  = Número de dias no período de avaliação.

- h) **Sabe com Certeza - Know Sure Thing (KST):** O oscilador KST é um complexo indicador de tendência e indica a direção e velocidade da mesma. As equações a seguir são utilizadas em sua composição:

A quatro componentes que são necessárias para calcular o KST são baseadas no indicador ROC com Equação 26.

$KST =$

$$\begin{aligned} & SMA_{10}(ROC(PreçoFinal_{10})) * W1 + \\ & SMA_{10}(ROC(PreçoFinal_{15})) * W2 + \\ & SMA_{10}(ROC(PreçoFinal_{20})) * W3 + \\ & SMA_{15}(ROC(PreçoFinal_{30})) * W4 \end{aligned}$$

Onde:  $W1 = 1$

$W2 = 2$

$W3 = 3$

$W4 = 4$

- i) **Ichimoku Cloud** O indicador Ichimoku Cloud, também conhecido como Ichimoku Kinko Hyo, define suporte e resistência, identifica direção de tendência, mede inércia e provê sinais de operação (PATEL, 2010). As 5 linhas são compostas pelas equações a seguir:

Tenkan-sen (Linha de conversão - *Conversion Line*)

$$Tenkan - sen = \frac{SMA_9(Máximo + SMA_9(Mínimo))}{2} \quad (86)$$

Kijun-sen (Linha Base - *Base Line*)

$$Kijun - sen = \frac{SMA_{26}(Máximo + SMA_{26}(Mínimo))}{2} \quad (87)$$

Senkou Span A (Período Líder A - *Leading Span A*)

$$SenkouA = \frac{Tenkan - sen + Kijun - sen}{2} \quad (88)$$

Senkou Span B (Período Líder B - *Leading Span B*)

$$SenkouB = \frac{SMA_{52}(Máximo + SMA_{52}(Mínimo))}{2} \quad (89)$$

Chikou Span (Período Atrasado - *Lagging Span*)

$$Chikou = \text{Preço Final 26 dias atrasado} \quad (90)$$

- j) **Aroon:** O Indicador Aroon identifica quando tendências tem possibilidade de inverter sua direção. O mesmo o faz gerando 3 linhas, a primeira contando de 25 a 0 toda vez que ocorre um novo máximo no período da janela dos últimos 25 dias, a segunda faz o mesmo mas contando a partir do mínimo no período. A terceira linha é uma subtração da segunda pela primeira (GOLD, 2015). As seguintes equações constroem as primeiras duas linhas:

$$AroonUP = \frac{25 - \text{Período desde o Último Máximo}}{25} * 100 \quad (91)$$

$$AroonDN = \frac{25 - \text{Período desde o Último Mínimo}}{25} * 100 \quad (92)$$

#### 2.4.5 Análise de Retorno Sobre Investimento

Como base para análise de investimentos financeiros está o cálculo da parcela de Retorno Sobre Investimento - *Return Over Investment* (ROI), que evidência quanto houve de lucro sobre certo investimento em determinado tempo. Para realização de seu cálculo, a mesma segue a seguinte equação:

$$ROI = \frac{(B_F + I_V) - B_0}{B_0} * 100\% \quad (93)$$

Onde:  $B_0$  = Caixa Inicial

$B_F$  = Caixa Final

$I_V$  = Valor Investido

#### 2.5 Métricas Estatísticas de classificação

As métricas estatísticas são fórmulas matemáticas que visam a comparação da grandeza do número de classificações corretas ou incorretas realizadas em relação aos valores reais. Estas métricas servem para inferência da qualidade de um classificador ou da decisão de um algoritmo. As mais comuns a serem utilizadas para tal são: Acurácia, Precisão, Revocação e o F1, que é uma média harmônica da precisão e da revocação. Os valores podem ser encontrados comumente na literatura em frações ou porcentagens (POWERS, 2020).

### 2.5.1 Acurácia

A acurácia leva em conta o número de predições corretas, dividida pelo total de predições. Esta métrica representa quão "correto" o classificador ou decisor é. A seguinte Equação 94 demonstra o cálculo:

$$Acurácia = \frac{NúmeroCorretos}{NúmeroTotal} \times 100\% \quad (94)$$

### 2.5.2 Precisão

A precisão expressa a qualidade do classificador/decisor no quesito de quantas amostras foram corretamente classificadas em relação aos "falsos positivos", ou seja, quantas amostras foram erroneamente classificadas em uma determinada classe. Esta é expressa pela Equação 95 abaixo:

$$Precisão = \frac{PC}{PC + FP} \times 100\% \quad (95)$$

Onde:  $PC$  = Positivos Corretos

$FP$  = Falsos Positivos

### 2.5.3 Revocação

A revocação expressa a qualidade do classificador/decisor no quesito de quantas amostras foram corretamente classificadas em relação aos "falsos negativos", ou seja, quantas amostras deixaram de ser corretamente classificadas em uma determinada classe. Esta é expressa pela Equação 96 abaixo:

$$Revocação = \frac{PC}{PC + FN} \times 100\% \quad (96)$$

Onde:  $PC$  = Positivos Corretos

$FN$  = Falsos Negativos

### 2.5.4 F1

A métrica F1, é uma média harmônica entre as métricas Precisão e Revocação, e exprime a relação entre as duas métricas, sendo sempre tendenciosa a métrica de menor valor entre as duas. A Equação 97 abaixo demonstra seu cálculo:

$$F1 = 2 * \frac{Precisão * Revocação}{Precisão + Revocação} x 100\% \quad (97)$$

### 3 TRABALHOS RELACIONADOS

A seção seguinte apresenta artigos, nos quais foram baseados a metodologia e experimentos, agrupados pelo tema principal, sendo estes: Redes Neurais Recorrentes, Transformada de *Wavelets* e Análise Técnica Financeira.

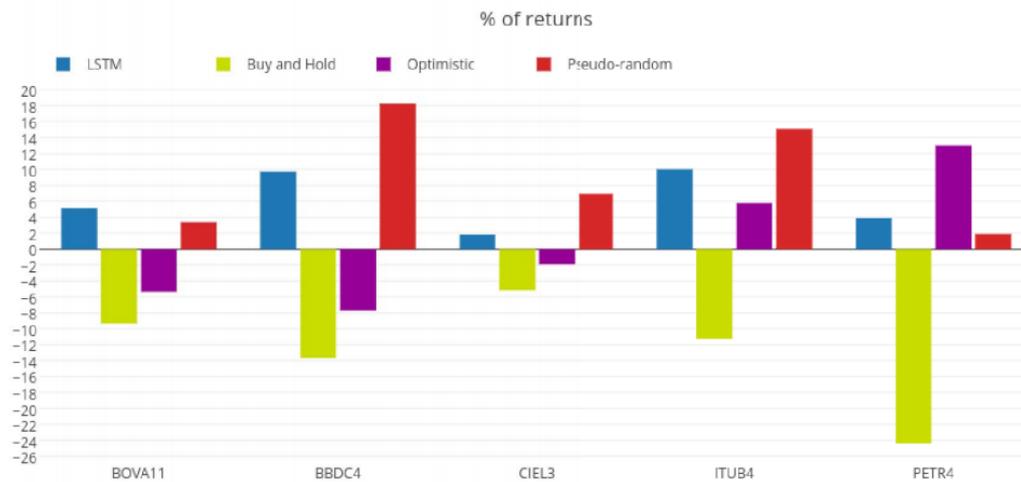
#### 3.1 Redes Neurais Recorrentes

Tendo em vista que RNNs são capazes de aprender padrões de longa e curta duração nos dados, diversos artigos nos últimos anos tentaram prever tendências ou preços no mercado de ações aplicando-as. Em seu estudo Alonso-Monsalve et al. (2020) criou um algoritmo híbrido chamado CLSTM que combina uma Rede Neural Convolucional - *Convolutional Neural Network* para pré processar os dados de ações consistindo no preço e diversos indicadores técnicos, em seguida alimentando os resultados à uma rede LSTM para prever o preço do próximo instante no tempo. Os autores compararam os resultados com uma rede LSTM pura, uma Rede Neural de Função Base Radial - *Radial Basis Function Neural Network* (RBFNN) e uma CNN, concluindo que o algoritmo proposto teve melhores resultados que os modelos mencionados.

Em seu trabalho Zhang et al. (2019) aplicou uma LSTM como o gerador das previsões do preço de fechamento de ações. O gerador é construído de forma a minerar as distribuições de dados de uma determinada ação e gerar dados nas mesmas distribuições, onde uma rede Perceptron de Multi Camada - *Multi-Layer Perceptron* (MLP) discriminadora é empregada para diferenciar os dados reais dos dados gerados. Os dados escolhidos para o estudo foram os valores diários do índice S&P 500 e diversas ações em uma larga faixa de dias de operação e tentou prever o preço diário de fechamento. O modelo proposto superou os modelos de comparação, que foram uma LSTM pura, um MLP e o modelo Regressão de Vetor de Suporte - *Support Vector Regression* (SVR), acumulando menos perdas.

Para executar sua pesquisa Nelson, Pereira e Oliveira (2017) treinou um modelo LSTM para prever o movimento de preços da BOVESPA (Bolsa de Valores de São Paulo) e acumulou 55.9% de acurácia em prever se o preço subiria ou cairia no futuro próximo. Sua metodologia utilizou não só os preços históricos das ações mas também diversos indicadores técnicos de inércia.

Figura 12 – Retorno sobre investimento do trabalho de Nelson, Pereira e Oliveira (2017) no ano de 2017



Fonte: Nelson, Pereira e Oliveira (2017).

Em seu artigo Huynh, Dang e Duong (2017) desenvolveu um novo modelo chamado Unidade Recorrente Chaveada Bi-direcional - *Bidirectional Gated Recurrent Unit (BGRU)*. Este modelo faz uso de notícias financeiras online e dados históricos de preços das ações para prever os movimentos das ações no futuro. Resultados experimentais mostram que seu modelo alcançou acurácia de aproximadamente 60% no índice S&P 500, onde para predição em ações individuais o valor é acima de 65%. Seus resultados foram comparados com redes GRU e LSTM, superando ambas.

Em outro trabalho, Hiransha et al. (2018) faz a comparação entre MLP, RNN, CNN e LSTM para prever o índice do *National Stock Exchange (NSE)* e o *New York Stock Exchange (NYSE)*. Em sua pesquisa a rede CNN deu melhores resultados que os outros algoritmos porque podia reagir mais rapidamente a mudanças abruptas no índice como esta considera apenas a janela de tempo na predição e não todo o histórico, segundo suas conclusões.

Samarawickrama e Fernando (2017) fez uma comparação entre os algoritmos que tomam utilizam o método das RNNs. Os algoritmos escolhidos foram MLP, GRU, LSTM e Rede Neural Recorrente Simples - *Simple Recurrent Neural Network (SRNN)*. Eles concluíram que LSTM e SRNN tiveram erros menores em geral exceto casos específicos, onde o MLP teve o menor erro. Neste trabalho em particular GRU teve o pior resultado em acurácia quando comparado com os outros modelos, contrastando com outras pesquisas recentes na área.

Shen et al. (2018) aplicou redes GRU aliadas com Máquina de Vetor de Suporte - *Support Vector Machine* (SVM) criando um novo modelo chamado GRU-SVM. A pesquisa comparou uma rede simples de GRU, o modelo GRU-SVM, uma Rede Neural Profunda - *Deep Neural Network* (DNN) e o SVM puro. Todos os modelos foram treinados para prever se a tendência da ação era subir ou cair. Os modelos que aplicaram redes GRU tiveram em todos os casos acurácia mais alta que os comparados, ao prever a tendência nos preços e ultrapassou o o valor de 50% o que valida a sua utilização em uma estratégia dólar neutra (*dollar neutral strategy*).

Estas pesquisas estabelecem o precedente que RNNs, em particular LSTM e GRU, estão sendo aplicadas com sucesso em prever a direção de tendências, mas que os melhores resultados vem de associações de RNNs com algum outro método de pré processamento dos dados.

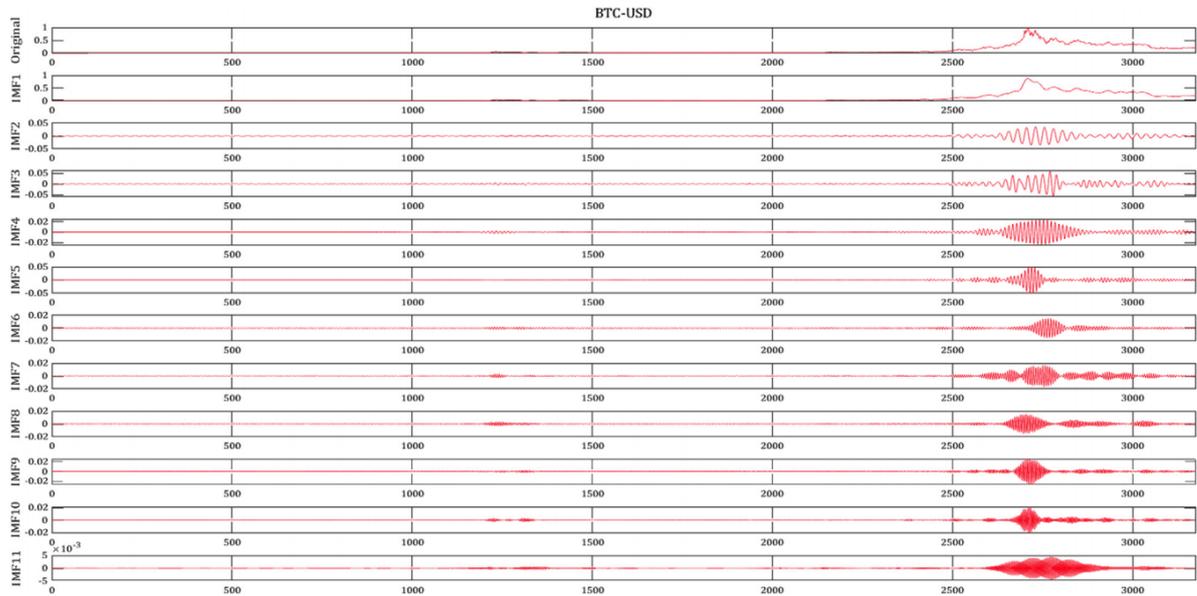
### 3.2 Ondaletas

Em seu artigo Nobre e Neves (2019) utilizou DWT combinado com XGBoost (abreviação de *Extreme Gradient Boosting*, que utiliza um grande número de preditores fracos para fazer um forte) e Análise de Componente Principal - *Principal Component Analysis* (PCA), para analisar e prever tendências no mercado de ações para comprar ou vender eficientemente, baseado no preço das ações e em diversos indicadores técnicos. Neste trabalho os autores afirmam que o uso da DWT melhorou a precisão na predição melhorando as métricas estatísticas de predição. Outro pesquisador utilizou *wavelets* combinando-as com RNNs e Colônia Artificial de Abelhas - *Artificial Bee Colony* (ABC) para prever o preço de ações e maximizar lucros. A DWT foi utilizada para filtragem de ruído em todas as variáveis e então alimentadas no modelo (HSIEH; HSIAO; YEH, 2011), reduzindo as perdas de predição consideravelmente.

No campo de predição do mercado de cripto moedas, Altan, Karasu e Bekiros (2019) aplicou a Transformada Empírica de Ondaletas - *Empirical Wavelet Transform* (EWT) inicialmente proposta por Gilles (2013) que consiste em extrair diferentes modos intrínsecos de uma série temporal através da construção de *wavelets* adaptativas, como pode ser visto na Figura 13. O algoritmo aliou Transformada Empírica de Ondaletas - *Empirical Wavelet Transform* (EWT) com LSTM para prever o valor de cripto-moedas, e Busca de Cuco - *Cuckoo Search* (CS) para otimizar o treinamento da LSTM. Este trabalho específico mostra que a decomposição de

um sinal em *wavelets* empíricas resultou em uma expressiva redução nas perdas de predição do algoritmo.

Figura 13 – Decomposição de *Wavelets* empíricas do trabalho de Altan, Karasu e Bekiros (2019) para o BTC-USD em seu preço histórico até 2019.



Fonte: Altan, Karasu e Bekiros (2019).

Em seu trabalho Stocchi e Marchesi (2018) uma rápida DWT é apresentada. Esta consiste em utilizar *wavelets* Daubechies compactas para transformar o sinal em diversos coeficientes e então realizar a IDWT em um grupo integrado de máquinas estimadoras para prever o preço de cripto-moedas.

No campo de consumo de energia, DWT foi empregada para decompor e reduzir a complexidade de um sinal de entrada, então aplicado em uma LSTM Multi camada bi-direcional para capturar recursos das subcomponentes e então prever a demanda de consumo de combustível baseado no passado e no consumo atual. Neste artigo Algoritmo Genético - *Genetic Algorithm* (GA) é utilizado para otimização das camadas do preditor (SU et al., 2019). Os autores argumentam que a DWT melhorou a capacidade do modelo de aprendizado de aprender os padrões de consumo de combustível de forma a aumentar a qualidade da predição.

Os trabalhos supracitados mostram que a transformada de *wavelets* é um método eficaz para filtrar ruído e decompor dados, e sua aplicação em dados financeiros melhorou drasticamente a acurácia e reduziu perdas nos modelos treinados.

### 3.3 Análise Técnica no Mercado de Ações

Em seu artigo Nobre e Neves (2019) gerou 26 indicadores técnicos com a biblioteca do Python TA-lib, e adicionou-os ao histórico de preços (Abertura, Máximo, Mínimo e Fechamento - *Open, High, Low and Close* (OHLC) e volume) e então normalizou os dados e alimentou uma PCA para extrair os mais relevantes. Este processo segundo o autor melhorou drasticamente os resultados do algoritmo.

Alonso-Monsalve et al. (2020) aplicou um total de 18 indicadores, todos indicadores de inércia. Eles afirmam que, mesmo que os resultados tenham sido satisfatórios para validar o estudo, se mais indicadores houvessem sido empregados os resultados teriam sido melhores. Justificam então, que em seu estudo empregaram apenas indicadores de inércia pois seu modelo Convolutional LSTM tem um alto custo computacional e o mesmo aumentaria muito de outra forma.

No trabalho de Nelson, Pereira e Oliveira (2017) sua LSTM tem a camada de entrada com uma dimensionalidade de 180 variáveis, que consistem em uma gama de indicadores técnicos (todos gerados pela Python TA-Lib) mais o preço histórico (Abertura, Máximo, Mínimo e Fechamento - *Open, High, Low and Close* (OHLC) e volume). A saída do modelo foi uma regressão utilizando a função tanh conectada a camada de saída da rede em vinte conexões. Segundo os mesmos, o modelo proposto, superou os resultados dos modelos de comparação tanto em retorno financeiro quanto reduzindo o valor máximo de perda de capital em uma única operação.

Hsieh, Hsiao e Yeh (2011) executou o treinamento de sua rede utilizando 14 indicadores técnicos (de inércia, bandas e médias) e os dados históricos de preço pré processados por *wavelets* de Haar e então alimentou-os a uma rede LSTM para aprender os padrões. Seu algoritmo teve os pesos otimizados pelo algoritmo Colônia Artificial de Abelhas - *Artificial Bee Colony* (ABC). Seus resultados foram estatisticamente satisfatórios e o mesmo alega que os indicadores técnicos, sendo escolhidos de forma correta, contribuíram para o sucesso da implementação da proposta.

Embora alguns autores afirmem que quanto mais indicadores melhor o resultado da acurácia, em seu trabalho Sezer, Ozbayoglu e Dogdu (2017) empregou apenas três indicadores consistindo no Índice de Força Relativa - *Relative Strength Index* (RSI), Média Móvel de Convergente e Divergente - *Moving Average Convergence Divergence* (MACD) e Indicador Williams %R - *Williams %R Indicator* (W %R). Os autores argumentam que estes foram escolhidos

pra manter a simplicidade e devido a sua grande aceitação. O modelo proposto foi testado nas ações do Dow 30 tendo resultados próximos do BNH superando o mesmo em grande parte dos papéis avaliados.

Neste trabalho podemos ver a figura 14, onde são expostos seus resultados para as ações do Dow 30, as 30 ações mais importantes da Dow Jones. As siglas significam em ordem: OUR - O resultados financeiros da proposta, BaH - Os resultados financeiros da estratégia de "comprar e segurar" (*Buy and Hold*), OURr - Rendimento Anualizado da proposta, BaHr - Rendimento anualizado do "comprar e segurar", AnT - Número anualizado de operações, PoS - Porcentagem de sucesso (operações com lucro), ApT - Lucro médio por transação, L - Duração média entre transações, MpT - Lucro máximo por transação, MIT - Prejuízo máximo por transação e MxC - Capital Máximo.

Figura 14 – Resultados da proposta de Sezer, Ozbayoglu e Dogdu (2017) na Dow 30 no ano de 2017

Share	OUR	BaH	OURr	BaHr	AnT	Pos	ApT	L	MpT	MIT	MxC
MMM	\$15234.16	\$29324.88	6.33%	16.99%	12.0	67.07%	0.63%	5	12.54%	-8.20%	\$15505.23
AXP	\$14727.15	\$15157.78	5.80%	6.25%	15.6	57.01%	0.68%	13	43.09%	-13.75%	\$21180.43
AAPL	\$14742.93	\$104256.20	5.83%	40.79%	5.8	60.00%	1.26%	20	23.28%	-7.33%	\$17804.25
BA	\$17010.05	\$22809.31	8.05%	12.78%	20.3	66.91%	0.50%	5	6.17%	-8.90%	\$18302.59
CAT	\$10252.42	\$21030.51	0.36%	11.44%	31.0	62.91%	0.12%	4	11.54%	-8.31%	\$12895.92
CVX	\$17907.21	\$22968.13	8.87%	12.89%	20.6	67.38%	0.48%	3	12.53%	-9.98%	\$18349.65
CSCO	\$21182.93	\$13126.52	11.57%	4.05%	22.0	66.89%	0.64%	7	8.96%	-9.39%	\$21594.89
KO	\$17258.98	\$23354.41	8.29%	13.18%	8.6	76.27%	1.03%	9	4.53%	-12.17%	\$17258.98
DIS	\$28859.03	\$34368.91	16.71%	19.72%	22.3	70.59%	0.82%	8	11.85%	-5.41%	\$30457.32
DD	\$17750.91	\$22197.39	8.74%	12.34%	18.0	66.67%	0.60%	5	19.97%	-6.65%	\$19297.54
XOM	\$18385.49	\$15946.05	9.30%	7.05%	23.7	66.67%	0.47%	5	20.27%	-5.78%	\$18868.03
GE	\$12663.52	\$12399.64	3.50%	3.19%	21.9	65.33%	0.31%	6	15.30%	-14.68%	\$13237.47
GS	\$14230.22	\$12238.97	5.28%	2.99%	23.2	64.78%	0.39%	5	24.93%	-15.50%	\$14230.22
HD	\$15088.71	\$43768.70	6.19%	24.04%	24.7	68.64%	0.36%	6	2.89%	-7.65%	\$18299.88
IBM	\$17151.82	\$21143.52	8.19%	11.55%	17.2	70.34%	0.54%	5	5.66%	-7.86%	\$19265.33
INTC	\$27965.75	\$23656.29	16.21%	13.40%	22.2	68.42%	0.80%	6	7.05%	-7.18%	\$31877.34
JNJ	\$19043.10	\$23687.77	9.86%	13.42%	17.8	73.77%	0.58%	6	8.85%	-6.99%	\$19279.80
JPM	\$49181.78	\$22092.57	26.17%	12.26%	24.5	67.26%	1.21%	5	27.14%	-8.62%	\$49181.78
MCD	\$17519.35	\$38489.77	8.53%	21.75%	15.5	70.75%	0.56%	2	3.79%	-4.01%	\$18445.34
MRK	\$29081.32	\$18865.70	16.86%	9.71%	22.5	69.48%	0.79%	6	8.86%	-6.71%	\$29389.65
MSFT	\$37923.78	\$25820.00	21.48%	14.85%	22.6	69.03%	0.97%	6	6.28%	-5.77%	\$37923.78
NKE	\$22940.48	\$48496.06	12.89%	25.93%	16.9	67.24%	0.93%	13	28.39%	-8.50%	\$28257.09
PFE	\$11094.86	\$18953.47	1.53%	9.78%	22.9	64.33%	0.16%	6	7.07%	-8.52%	\$11653.85
PG	\$20278.23	\$17434.55	10.88%	8.46%	18.8	68.99%	0.62%	9	10.23%	-5.48%	\$20278.23
TRV	\$64371.78	\$31098.53	31.23%	18.01%	24.1	75.15%	1.26%	7	33.97%	-6.54%	\$64371.78
UTX	\$18540.16	\$20932.55	9.43%	11.38%	24.1	68.48%	0.47%	5	21.00%	-10.23%	\$19360.46
UNH	\$9343.90	\$34464.65	-0.99%	19.80%	15.2	57.69%	0.25%	9	10.67%	-17.73%	\$12030.98
VZ	\$12147.37	\$24315.17	2.88%	13.83%	16.2	61.26%	0.26%	6	22.05%	-5.97%	\$13267.72
WMT	\$32230.01	\$18389.92	18.63%	9.30%	18.5	73.23%	0.98%	7	11.07%	-8.07%	\$32230.01

Fonte: Sezer, Ozbayoglu e Dogdu (2017).

Pimenta et al. (2018) empregou 11 indicadores técnicos subdivididos em osciladores, seguidores de tendência, sistemas de bandas e identificadores de divergências, treinando um algoritmo genético para aprender a direção da bolsa de valores brasileira (BOVESPA), tendo bons resultados na acurácia da predição da direção da tendência, e superando expressivamente a estratégia BNH em termos de retorno sobre investimento. Na figura 15 podemos ver o retorno

sobre o investimento no período de maio/2013 a fevereiro/2015. Ullah et al. (2019) indo na mesma direção utilizou Média Móvel Simples - *Simple Moving Average* (SMA), Média Móvel de Convergente e Divergente - *Moving Average Convergence Divergence* (MACD), Oscilador Estocástico - *Stochastic Oscillator* (SO), Média Exponencial Tripla - *Triple Exponential Average* (TRIX) e Índice Médio Direcional - *Average Directional Index* (ADX), e alimentou o valores a uma DNN para prever o preço de ações na Bolsa de ações Paquistanesa (*Pakistan Stock Exchange* - PSE). Os autores, no entanto, contradizem outros trabalhos na mesma área. Argumentam que embora relevantes para operadores de ações no dia a dia, para o modelo proposto não foi notada melhora com o uso dos indicadores técnicos.

Figura 15 – Retorno sobre investimento do trabalho de Pimenta et al. (2018) no ano de 2013 a 2015

Share	Proposed	B&H (%)	TA-10 (%)	TA-36 (%)
BBAS3	19.83% ± 2.72%	-18.33	5.73	11.13
BOVA11	3.16% ± 0.15%	5.69	9.87	-1.03
CMIG4	23.33% ± 4.61%	-22.42	8.76	3.47
EMBR3	14.36% ± 1.44%	-23.62	12.18	6.43
GGBR4	-2.11% ± 1.99%	-3.28	4.51	3.92
VALE5	50.58% ± 6.18%	-8.78	2.03	-3.14
Portfolio	18.19%	-11.79	7.18	3.46

Fonte: Pimenta et al. (2018).

Estes artigos mostram que a escolha dos indicadores técnicos financeiros podem ter grande influência no resultado do estudo. Como mencionado por alguns autores, mais indicadores técnicos geralmente resulta em melhor acurácia nos modelos preditivos. Pode-se extrair de tantos trabalho aplicando estes modelos matemáticos que indicadores técnicos financeiros podem servir como uma confiável fonte de informação para os modelos preditivos aprenderem tendências financeiras complexas e extrair mais detalhes sobre o comportamento intrínseco de um determinado ativo.

Como pode ser notado pela grande variedade de aplicações de RNNs com diversas formas de pré-processamento e treinamento, o uso das mesmas está consolidado e validado para análise e predição em séries temporais. No entanto os trabalho revisados utilizam apenas uma ou duas das técnicas discutidas neste trabalho ao mesmo tempo, não aliando toda a capacidade de filtragem e decomposição das *wavelets* ao poder de análise dos padrões no tempo das RNNs. Portanto além do pré-processamento de ruído como proposto na aplicação de Nobre e Neves

(2019), pode-se utilizar da decomposição em *wavelets* empíricas como fez Altan, Karasu e Bekiros (2019) e então estes dados combinados com indicadores técnicos como dados auxiliares no treinamento das redes para ter uma capacidade maior de aprendizado no treinamento.

## 4 METODOLOGIA

Nesta pesquisa é proposto um novo modelo de treinamento e predição de valores e operação na bolsa de valores a partir dos preços históricos, pré-processamento dos dados via DWT e utilização de indicadores técnicos financeiros para criação do modelo Rede de Transformada Discreta de Ondas e Unidade Recorrente Chaveada - *Discrete Wavelet Transform Gated Recurrent Unit Network* (DWT-GRU). Nesta seção, será discutida a metodologia para criação do sistema de tomada de decisão baseado na aquisição e pré-processamento dos dados, sua utilização no treinamento do modelo proposto e predição da série temporal, avaliação das decisões a serem tomadas com base na predição e posterior avaliação do resultado financeiro.

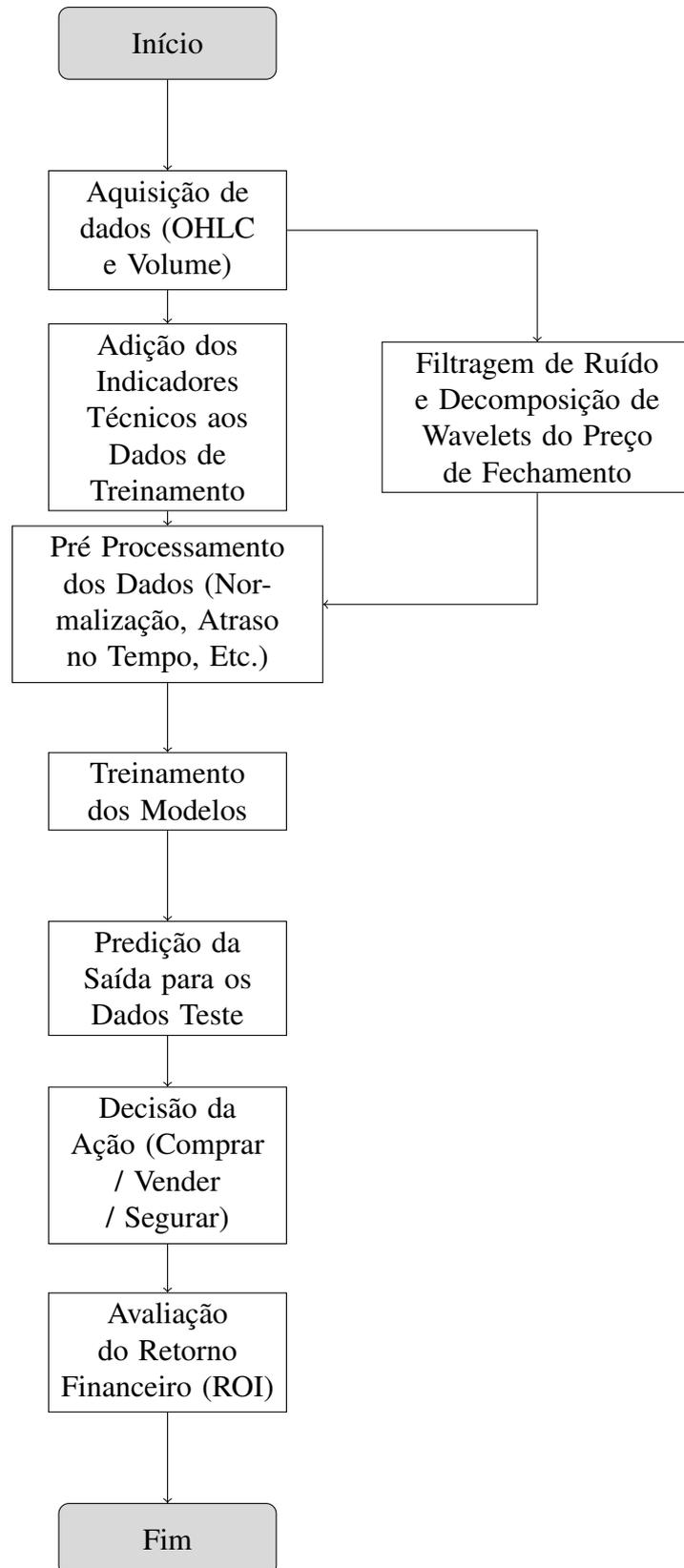
### 4.1 Arquitetura do Sistema

O modelo proposto consiste em combinar a capacidade de filtragem de ruído e decomposição de sinais da DWT e aplicar os dados pré processados para serem treinados por uma RNN baseada principalmente em uma camada GRU. Este novo modelo chamado Rede DWT-GRU tem como objetivo superar os resultados de simples implementações de apenas redes LSTM ou GRU que não processem previamente os dados com uma DWT, bem como, a estratégia Comprar e Segurar - *Buy and Hold* (BNH) que é frequentemente utilizada para comparação em algoritmos de predição de preços no mercado de ações.

No sistema proposto, a DWT posa como um método para melhorar a eficiência do aprendizado das RNNs, pois com a filtragem do ruído pela DWT o modelo não é influenciado por perturbações de alta frequência, permitindo o aprendizado mais claro das tendências nos dados.

A arquitetura de sistema proposta é composta de diversos processos consistindo principalmente na aquisição de dados e pré processamento dos mesmos, treinamento da rede neural, cálculo dos resultados e avaliação do modelo. A Figura 16 ilustra as ações executadas pelo sistema e os detalhes de cada uma delas será explicado nas próximas subseções.

Figura 16 – Diagrama de Blocos do Sistema



## 4.2 Aquisição de Dados

Os dados utilizados neste estudo foram adquiridos do banco de dados do *Yahoo Finance*<sup>1</sup> para cada ação individualmente em informações de preço diário sem tratamento prévio (OHLC e Volume).

Os dados escolhidos para o estudo foram baseado na Bolsa de Valores do Estado de São Paulo (BOVESPA) e são principalmente focados no Fundo de Índice Negociado em Bolsa - *Exchange-Traded Fund* (ETF) BOVA11 e as ações ITUB4, ABEV3, PETR4 and VALE3, que são bem conhecidas por seu grande valor de mercado e contam com um grande histórico de crescimento consistente e confiável no longo prazo. Os dados para a BOVA11 datam desde Dezembro de 2008 em um total de aproximadamente 2800 dias de operação em dados históricos. Já para as outras quatro ações supracitadas todas datam desde o ano 2000 com um histórico de dias de operação de quase 5000 dias.

## 4.3 Indicadores Técnicos

Nesta pesquisa serão utilizados todos os indicadores técnicos apresentados anteriormente descritos na seção 2. Serão utilizados indicadores das quatro classes apresentadas para extrair o máximo de informação possível dos dados históricos de preços dos ativos analisados.

## 4.4 Pré-processamento dos Dados

O pré processamento dos dados é subdividido em diversos passos descritos abaixo:

- a) Adição dos indicadores técnicos à tabela de dados da ação, totalizando neste passo 74 variáveis;
- b) Aplicar a DWT no sinal de preço de fechamento e filtrá-lo reconstruindo-o sem as 2% frequências mais altas, em seguida então aplicar a primeira e a segunda derivada ao sinal reconstruído para extrair os gradientes, e finalmente adicionar os resultados à tabela de dados, acumulando 77 variáveis;
- c) Decompor o sinal do preço de fechamento em 6 *wavelets* empíricas utilizando a EWT e adicionar os resultados à tabela de dados, resultando em um total de 83 variáveis;

---

<sup>1</sup><https://finance.yahoo.com/> [Accessed: 26/07/2020]

- d) Normalizar os dados.;
- e) Aplicar a cada variável da tabela de dados (um total de 83 colunas) o atraso no tempo que primeiramente copia as colunas um determinado número de vezes, em seguida os desloca passo a passo para fazer com que cada observação seja composta de um número  $n$  resultando em  $Entrada = x(t - n)...x(t - 1), x(t)$ . Neste caso o número de dias escolhido para o estudo foram 5 dias;
- f) A saída desejada para treinar os modelos é selecionada escolhendo-se o sinal de preço de fechamento 2 dias após o atraso no tempo, o que resulta no alvo do treinamento ser a predição de 2 dias no futuro baseado nos dados disponíveis;
- g) O último passo é a divisão dos dados em três partes, seguindo a distribuição de 80/20% entre dados de treinamento e validação, então os dados de validação são novamente divididos seguindo a proporção de 75/25% entre dados de validação e teste, respectivamente.

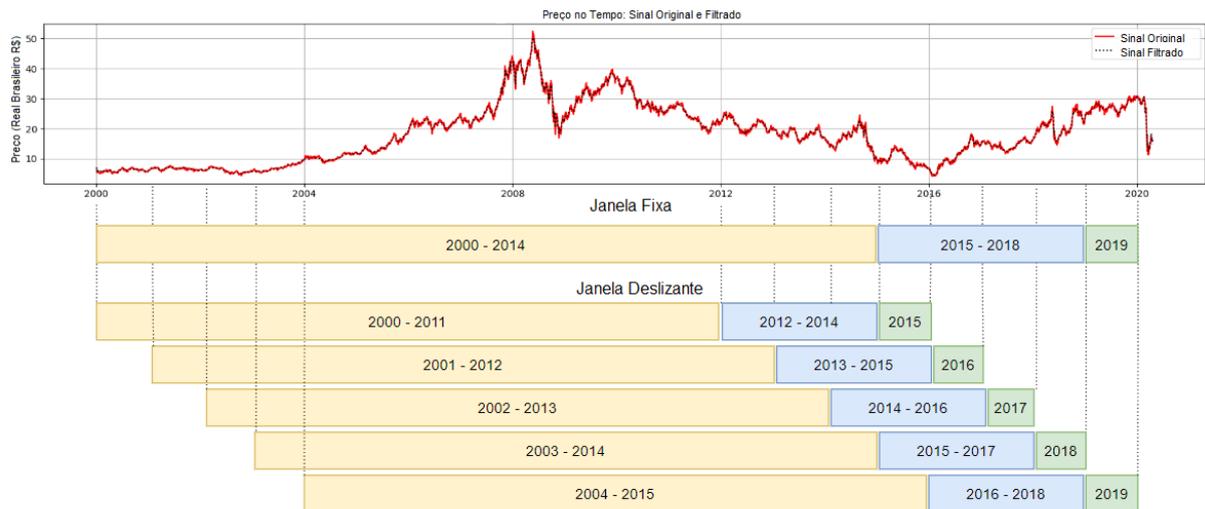
#### 4.4.1 Janela de treinamento

É importante ressaltar que os resultados iniciais da próxima seção foram baseados no treinamento com a divisão de *datasets* de treino descrita acima. Também foi implementado a divisão de forma janela deslizante (*rolling window*) para verificação e validação cruzada do algoritmo. Abaixo está exemplificado na Figura 17 o sistema proposto de treinamento, validação e teste, sendo a primeira barra a organização de janela fixa (*fixed window*) que foi testada para os resultados iniciais, e as barras seguintes a *rolling window* proposta para as próximas análises.

#### 4.4.2 Normalização dos dados

A normalização dos dados feita calculando a média e o desvio padrão da distribuição de dados e em seguida retirando a média e dividindo cada amostra pelo desvio padrão. Os testes iniciais foram em **normalização global** para cada variável em sua completude para cada uma, ou seja, a normalização abrangeu toda a amplitude dos dados. Em uma segunda análise foi feita a **normalização local** dos dados, ou seja, feita para cada amostra de dados que contém cinco dias de entrada mais o dia de previsão do futuro. Esta amostra é a que seria alimentada ao algoritmo no treinamento e posteriormente no teste sendo calculada sua média e desvio padrão para cada amostra. Isto teve como finalidade a verificação da possibilidade da criação

Figura 17 – Divisão dos dados de treinamento.



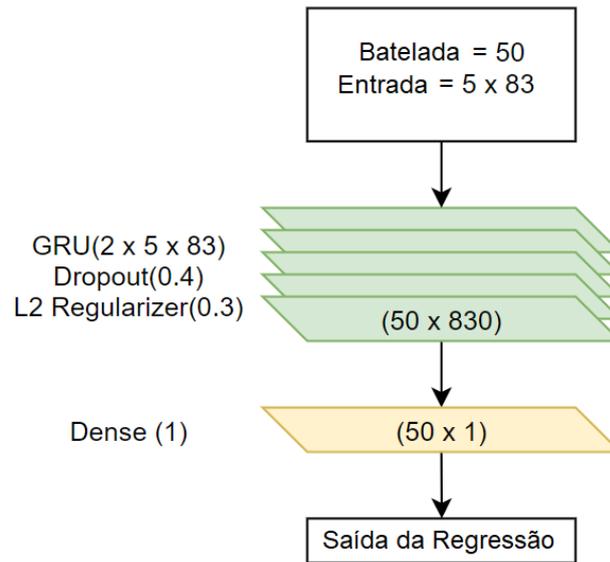
Fonte: Autoria Própria.

de um modelo que, ao ser treinado em uma determinada amplitude de dados que compõe o preço histórico de uma ação da bolsa de valores, pudesse ser utilizado em outras ações sem a dependência da média e desvio padrão da distribuição dos preços dos dados de treinamento originais do modelo. Assim será possível a utilização de modelos "genéricos" e não apenas especializados para cada ação.

#### 4.5 Modelo Proposto

A rede GRU foi escolhida baseado em sua capacidade par com as redes LSTM, mas sendo menos exigente em termos de esforço computacional, portanto sendo treinada mais rapidamente e convergindo mais facilmente em resultados satisfatórios. A Figura 18 ilustra as camadas utilizadas na construção da rede neural responsável pela predição no sistema. A primeira camada consiste em uma camada GRU com sua dimensão baseada na multiplicação o formato de entrada dos dados por 2 e contando com um "dropout" de 0.4 com o valor de regularizador L2 de 0.3 para prevenir contra o sobre-aprendizado dos dados e uma única camada totalmente conectada com um único neurônio para predizer o preço da ação como a saída da rede.

Figura 18 – Modelo de Rede Neural Proposta



Fonte: Autoria Própria

#### 4.6 Tomada de Decisão

Após o treinamento e validação, o próximo passo consiste em utilizar o modelo treinado para prever a saída dos dados de teste. O sinal predito é então tratado em uma função específica para decisão entre as ações possíveis para indicar "compra", "venda" ou "segurar".

A decisão é baseada em detectar cruzamentos com a linha zero na primeira derivada do sinal predito e avaliar se após o cruzamento esta se encontra acima ou abaixo de tal linha. Por exemplo, se o valor cruza a linha zero e o próximo valor é positivo, existe uma tendência no preço dele subir, portanto a ação a ser tomada é "Compra". Se a derivada do sinal de saída predito cruza o zero e próximo valor é negativo, então a ação a ser tomada é "Venda". Todos os outros casos que não caem nestas duas condições por eliminação a ação tomada é "Segurar".

O algoritmo 1 abaixo ilustra o funcionamento descrito no parágrafo acima.

A Figura 19 abaixo ilustra a os gradientes do sinal filtrado do ano de 2019 para melhor entendimento do algoritmo. Pode-se notar que os picos e os vales do sinal filtrado coincidem com os zeros do primeiro gradiente.

Desta forma podemos identificar os momentos de inversão de tendências, ou pontos específicos de compra ou venda de um ativo.

---

**Algoritmo 1 – Algoritmo de Tomada de Decisão**


---

**Entrada:** Valor\_Predito, Tamanho\_Valor\_Predito

**Saída:** Decisão

*Inicio*  $\leftarrow$  0;

*Fim*  $\leftarrow$  *Inicio* + *Tamanho\_Valor\_Predito*;

*Sinal\_Velocidade*  $\leftarrow$  *Gradiente*(*Valor\_Predito*);

*Decisão*  $\leftarrow$  *Lista*[ ];

**para** *i* em *Sinal\_Velocidade*[*Inicio*:*Fim* - 1] **faça**

**se** *Sinal\_Velocidade*[*i*] < 0 **E** *Sinal\_Velocidade*[*i* + 1] >= 0 **então**

    | *Decisão*[*i* + 1]  $\leftarrow$  "Compra";

**fim**

**senão se** *Sinal\_Velocidade*[*i*] > 0 **E** *Sinal\_Velocidade*[*i* + 1] <= 0 **então**

    | *Decisão*[*i* + 1]  $\leftarrow$  "Venda";

**fim**

**senão**

    | *Decisão*[*i* + 1]  $\leftarrow$  "Aguardo";

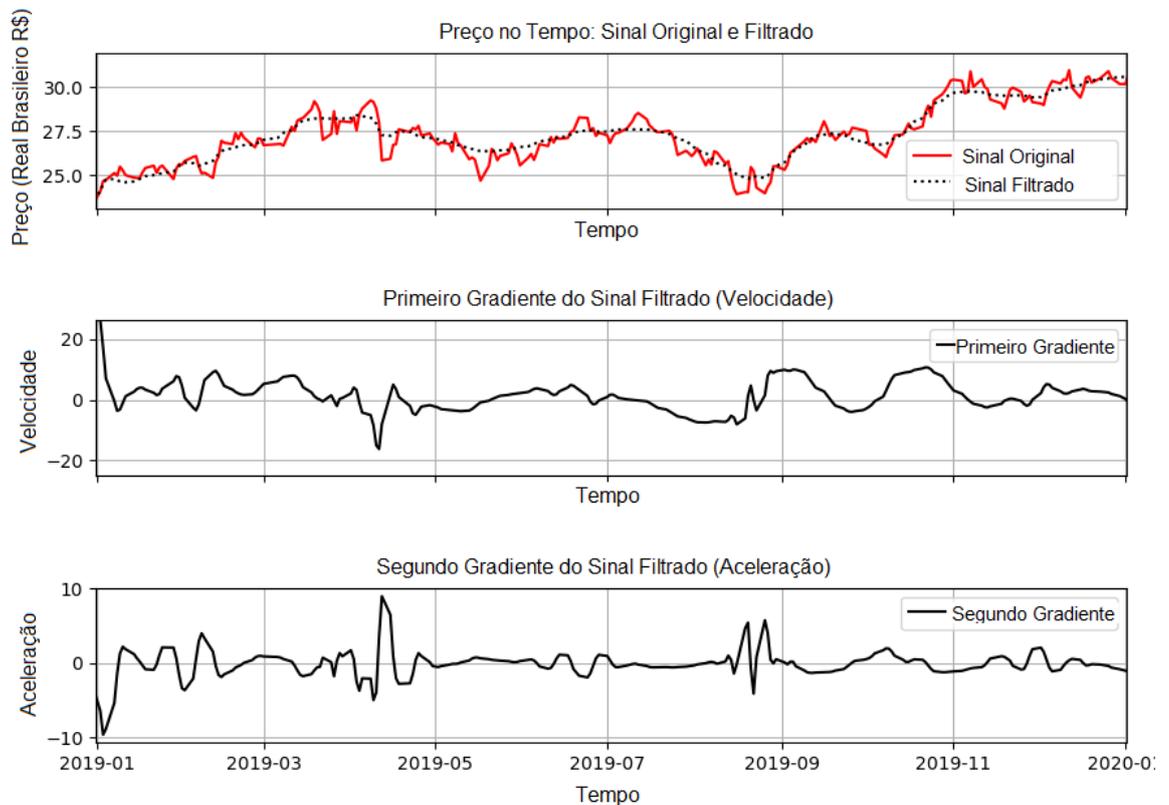
**fim**

**fim**

**retorna** *Decisão*

---

Figura 19 – Primeiro e segundo gradientes do sinal filtrado da ação PETR4 no ano 2019.



Fonte: Autoria Própria.

## 5 EXPERIMENTOS E RESULTADOS

Nesta seção serão apresentados experimentos e resultados de testes realizados para verificar a eficácia e validade do uso do algoritmo proposto, bem como a comparação do mesmo com outros modelos de RNNs e aprendizado clássico de máquina. Serão discutidos as métricas estatísticas e retorno financeiro de cada modelo e avaliadas suas implicações para o resultado da pesquisa. Serão apresentados inicialmente testes com as ações da BOVESPA, sendo elas: PETR4, VALE3, ITUB4, ABEV3 e também o ETF BOVA11. Também será apresentada a comparação do resultado para o uso de normalizações global ou local, e a viabilidade do uso de um modelo treinado em um ativo ser utilizado em outra distribuição de dados sem relação direta com este. Por fim será apresentado o resultado de teste em tempo real do algoritmo para confirmar a viabilidade prática de utilização do mesmo em aplicações reais.

Os experimentos foram conduzidos inicialmente testando os modelos treinados nos anos 2001 a 2016 nos dados históricos do ano de 2019. O banco de dados do Yahoo Finance exporta um arquivo .csv (valores separados por vírgula - *comma separated values*) e os dados são então importados no sistema por meio da biblioteca Pandas (TEAM, 2020) em uma tabela de dados.

O modelo DWT-GRU foi comparado com diversos modelos base incluindo o Rede de Transformada Discreta de Ondas e Memória Longa de Curto Prazo - *Discrete Wavelet Transform Long Short-Term Memory Network* (DWT-LSTM), uma rede pura GRU e uma rede LSTM. O ROI de cada modelo, bem como a estratégia Comprar e Segurar - *Buy and Hold* (BNH), também serão discutidos. O modelo é treinado para prever dois dias no futuro já que isto aumenta a capacidade do mesmo de tomar ações no melhor momento possível sem perder acurácia.

Nesta pesquisa foram utilizados todos os indicadores técnicos disponíveis na biblioteca de análise técnica financeira do Python chamada "Python Technical Analysis Library"(Python TA-Lib)<sup>1</sup> e deixar a cargo do modelo decidir quais são relevantes ou não através do treinamento.

Para a normalização dos dados foi utilizada a função `StandardScaler` da biblioteca do Python chamada `SkLearn` (PEDREGOSA et al., 2011), que calcula a média e o desvio padrão dos dados, e em seguida retira a média e divide pelo desvio padrão cada uma das amostras.

O treinamento e a construção do modelo foi feita utilizando a estrutura do Keras (CHOLLET et al., 2015) e a base de funções do TensorFlow (MARTÍN ABADI et al., 2015), ambos baseados em Python.

---

<sup>1</sup><https://technical-analysis-library-in-python.readthedocs.io/en/latest/ta.html> [Acessado: 26/07/2020]

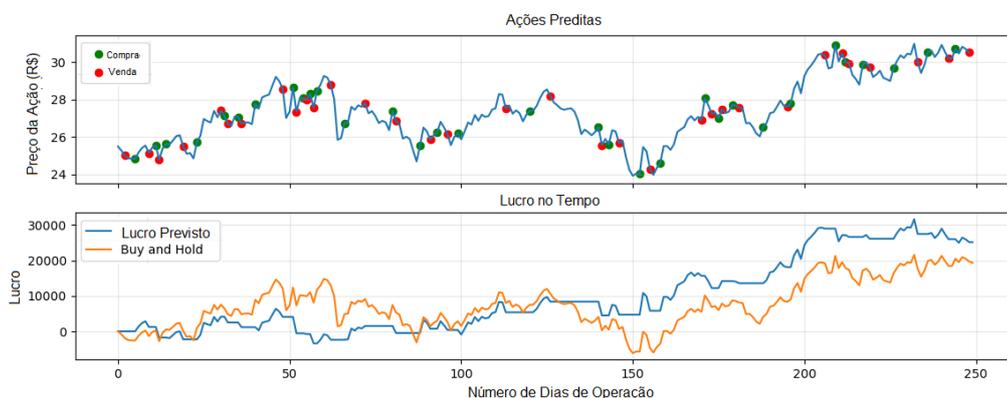
## 5.1 Avaliação do Modelo

A avaliação do modelo se dá realizando a simulação de compra e venda de acordo com as decisões baseadas no período de um ano de dados para testar o comportamento do modelo com dados não utilizados em seu treinamento e verificar qual o lucro obtido em tais operações a partir do uso de uma quantia limitada de dinheiro fictício para tais execuções. Foi comparada também o comportamento previsto da rede com o considerado "ideal" a partir dos dados já conhecidos, ou seja, verificar se a ação que deveria ter sido tomada naquele espaço de tempo realmente foi tomada, extraíndo-se assim as métricas estatísticas para avaliação do modelo (Acurácia, Precisão, Revocação e F1, que é uma média harmônica entre a Precisão e a Revocação.)

Para avaliação do resultado financeiro serão desconsideradas taxas de corretagem de operações de *Swing Trade*, que se trata de executar apenas uma única decisão (compra ou venda) por dia, que é o caso do algoritmo proposto. Esta abordagem se dá por existirem atualmente corretoras que, para estes tipos de operação, oferecem planos sem corretagem viabilizando a comparação mais simples do modelo proposto com a estratégia BNH.

As seguintes figuras: Figura 20, Figura 21 e Figura 22, mostram o comportamento do algoritmo nos dados históricos de teste do ano 2019 da ação PETR4 para os três modelos base, todos comparados com a estratégia BNH. Foram sinalizadas apenas as ações de "Compra" e "Venda", sendo todas as outras ações não sinalizadas sempre "Segurar".

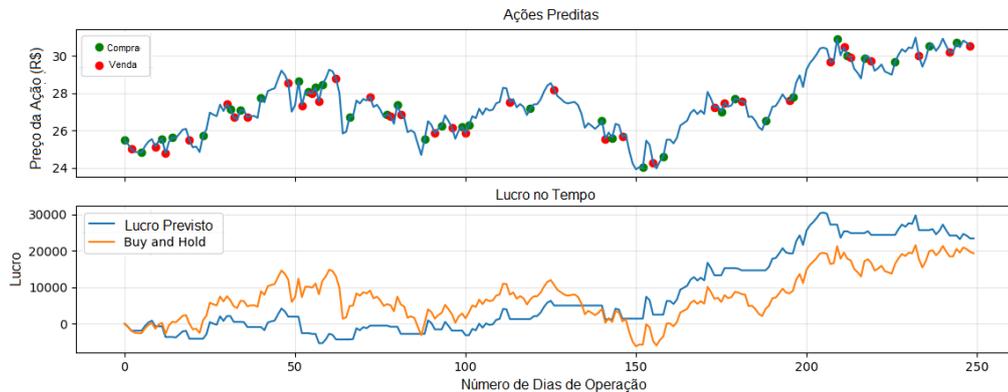
Figura 20 – Ações do algoritmo e lucro na ação PETR4 - Modelo Puro LSTM



Fonte: Autoria Própria.

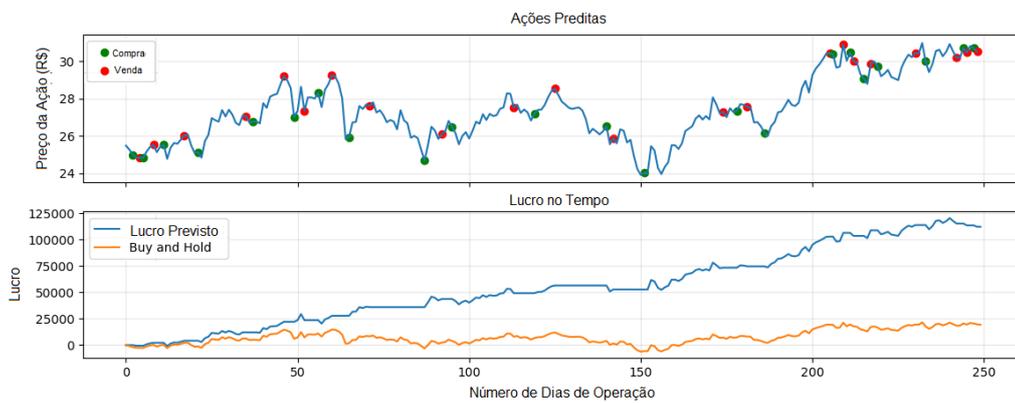
A Figura 23 ilustra o comportamento do algoritmo nos dados históricos do ano de 2019 da ação PETR4, e mostra um lucro maior utilizando o algoritmo DWT-GRU do que a estratégia BNH, também um ROI maior que as outras bases de comparação. Mesmo que a Figura 22

Figura 21 – Ações do algoritmo e lucro na ação PETR4 - Modelo Puro GRU



Fonte: Autoria Própria.

Figura 22 – Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-LSTM



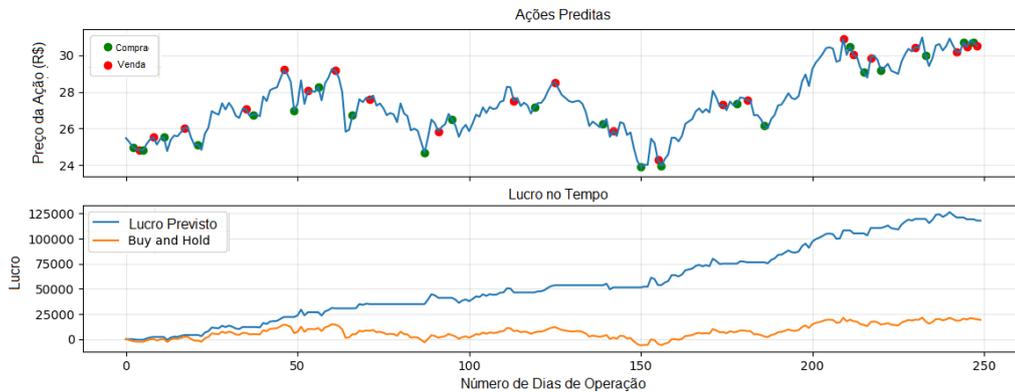
Fonte: Autoria Própria.

pareça similar a Fig.23, o modelo proposto adquiriu um lucro levemente maior do que a a estratégia do modelo DWT-LSTM.

## 5.2 Desenvolvimento de Outros Modelos

Para fins de comparação de capacidade de aprendizado e generalização também serão testados modelos de aprendizado de máquina mais simples dos que as complexas redes neurais, sendo comparados seus resultados com os modelos de RNNs discutidos anteriormente. A validade da comparação se dá para verificar se o pré processamento de dados realizado pela DWT influencia o suficiente no aprendizado para modelos mais simples terem resultados satisfatórios. Para fins de estudo será mantido a mesma entrada de dados tanto os modelos mais simples e as

Figura 23 – Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-GRU



Fonte: Autoria Própria.

RNNs. Os modelos escolhidos para comparação são a regressão linear e o algoritmo de floresta aleatória.

### 5.3 Análise

A avaliação do Retorno Sobre Investimento - *Return Over Investment* (ROI) é baseada em iniciar o algoritmo com uma quantia limitada de "dinheiro" disponível para que o mesmo o utilize em suas ações. Toda vez que o algoritmo decide por uma ação de "Compra" o mesmo avalia quantas cotas consegue comprar com o "caixa" disponível e simula a transação subtraindo do caixa o valor de quantas cotas, múltiplas de 100, ele "comprou" transferindo o valor para uma outra variável chamada "valor investido". Então quando o mesmo decide por "venda", o algoritmo checa quantas cotas da ação tem em custódia e o preço atual da ação para determinar quanto as mesmas valem no momento e zera a variável "valor investido" transferindo este valor de volta para o "caixa". Quando o período de avaliação determinado termina, o algoritmo calcula qual foi o valor de lucro que a estratégia obteve utilizando a equação 93 anteriormente descrita na seção 2 de Teoria. O Algoritmo 2 abaixo ilustra a explicação supramencionada.

#### 5.3.1 Resultado Financeiro

O novo modelo DWT-GRU teve um bom resultado comparado aos modelos base e a estratégia de Comprar e Segurar - *Buy and Hold* como ilustrado na Figura 24. Como podemos observar o lucro total do modelo proposto foi maior em todas as ações avaliadas superando

---

**Algoritmo 2 – Algoritmo de Avaliação de Retorno Financeiro**


---

**Entrada:** Preço\_Fechamento, Vetor\_Decisão, Tamanho\_Período, Balanço\_Inicial

**Saída:** Retorno\_Sobre\_Investimento

$Inicio \leftarrow 0;$

$Fim \leftarrow Inicio + Tamanho\_Período;$

$Balanço \leftarrow Balanço\_Inicia;$

$Valor\_Investido \leftarrow 0;$

$Número\_de\_Cotas \leftarrow 0;$

**para**  $i$  em Vetor\_Decisão[Inicio:Fim] **faça**

$Cotas\_Possíveis \leftarrow Balanço/Preço\_Fechamento[i];$

$Cotas\_Possíveis \leftarrow Cotas\_Possíveis - Cotas\_Possíveis\%100;$

**se** Vetor\_Decisão[ $i$ ] == "Compra" E Balanço  $\geq 100 * Preço\_Fechamento[i]$

**então**

$Balanço \leftarrow Balanço - Preço\_Fechamento[i] * Cotas\_Possíveis;$

$Número\_de\_Cotas \leftarrow Cotas\_Possíveis;$

**fim**

**senão se** Vetor\_Decisão[ $i$ ] == "Venda" E Número\_de\_Cotas  $> 0$  **então**

$Balanço \leftarrow Balanço + Preço\_Fechamento[i] * Número\_de\_Cotas;$

$Número\_de\_Cotas \leftarrow 0;$

**fim**

**fim**

$Valor\_Investido \leftarrow Preço\_Fechamento[Fim] * Número\_de\_Cotas;$

$Retorno\_Sobre\_Investimento \leftarrow$

$((Balanço + Valor\_Investido) - Balanço\_Inicial)/Balanço\_Inicial * 100;$

**retorna** Retorno\_Sobre\_Investimento

---

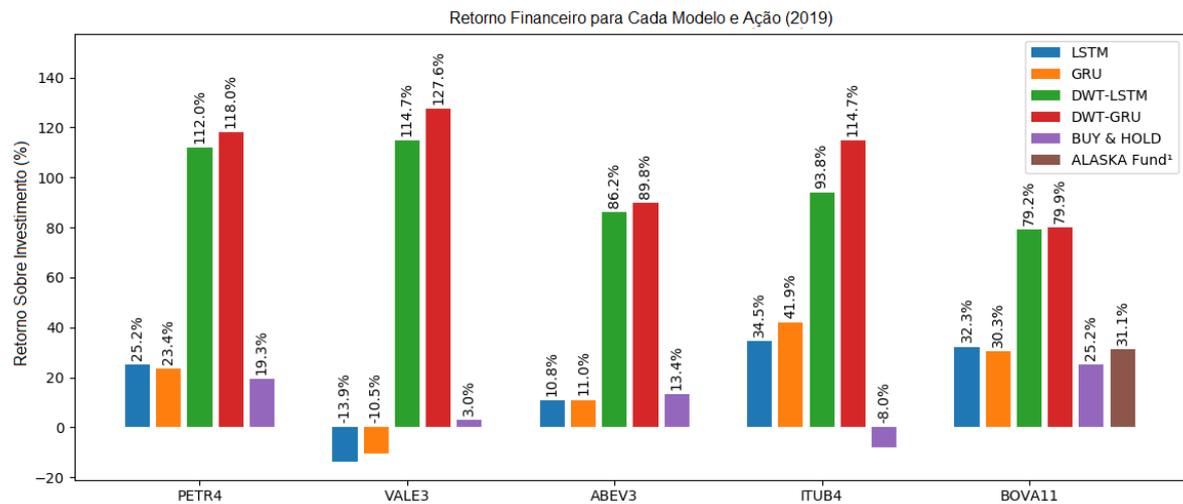
todos os modelos de comparação. Especialmente para a ETF BOVA11 foram comparados os resultados dos modelos à um fundo de investimento independente chamado Alaska Black. O modelo proposta também superou o ROI deste fundo <sup>2</sup> no período analisado.

### 5.3.2 Métricas Estatísticas

A Tabela 1 seguinte agrupa as métricas do modelo proposto para cada ação da bolsa de valores analisada, cruzando as ações esperadas em cada intervalo no tempo com a executada pelo algoritmo. Como podemos notar a acurácia foi satisfatória para um modelo de predição utilizando tantas variáveis de entrada e este tipo de complexidade. Como todos as métricas estatísticas se mantiveram acima de 70% pode-se afirmar que o algoritmo tem consistência na classificação das decisões a serem tomadas nos momentos esperados.

<sup>2</sup>[https://www.alaska-asset.com.br/pdf/Fundos/BLACK\\_INSTITUCIONAL.pdf](https://www.alaska-asset.com.br/pdf/Fundos/BLACK_INSTITUCIONAL.pdf) [Acessado: 21/04/2020]

Figura 24 – Lucro de Retorno sobre Investimento para Cada Ação



Fonte: Autoria Própria.

Tabela 1 – Métricas do Modelo DWT-GRU

Ação	Acurácia	Precisão	Revocação	F1
PETR4	0.852	0.875	0.852	0.861
VALE3	0.788	0.833	0.788	0.806
ABEV3	0.740	0.787	0.740	0.759
ITUB4	0.756	0.801	0.756	0.773
BOVA11	0.700	0.790	0.700	0.736

Para comparação, é válido apresentar também as tabelas de métricas dos modelos utilizados na comparação, com elas sendo as tabelas 2, 3 e 4 representando as métricas dos modelos LSTM puro, GRU puro e DWT-LSTM, respectivamente.

Tabela 2 – Métricas do Modelo LSTM

Ação	Acurácia	Precisão	Revocação	F1
PETR4	0.460	0.449	0.460	0.450
VALE3	0.396	0.415	0.396	0.405
ABEV3	0.424	0.424	0.424	0.424
ITUB4	0.484	0.491	0.484	0.488
BOVA11	0.448	0.493	0.448	0.469

Também nota-se a proximidade dos valores de acurácia do modelo DWT-LSTM o que reforça a melhora notável das métricas estatísticas dos modelos beneficiados pelo pré processamento de *wavelets*. Enquanto para os modelos puros as métricas estatísticas se mostraram inferiores a 50%, portanto não sendo melhores do que operação randômica.

Tabela 3 – Métricas do Modelo GRU

Ação	Acurácia	Precisão	Revocação	F1
PETR4	0.448	0.446	0.448	0.447
VALE3	0.400	0.440	0.400	0.418
ABEV3	0.400	0.409	0.400	0.404
ITUB4	0.476	0.487	0.476	0.482
BOVA11	0.480	0.463	0.480	0.471

Tabela 4 – Métricas do Modelo DWT-LSTM

Ação	Acurácia	Precisão	Revocação	F1
PETR4	0.824	0.858	0.824	0.837
VALE3	0.784	0.818	0.784	0.799
ABEV3	0.784	0.817	0.784	0.797
ITUB4	0.772	0.800	0.772	0.783
BOVA11	0.716	0.769	0.716	0.739

#### 5.4 Treinamento com Janela Deslizante

Para validar o algoritmo de forma a se avaliar se o mesmo se comporta satisfatoriamente realizando o seu treinamento, validação e testes em diversas janelas no tempo conforme ilustrado na seção 4 de metodologia sobre a janela de treinamento, foram executados testes nas ações já testadas com a janela fixa anteriormente.

Este teste tem como finalidade não avaliar os resultados financeiros de cada um dos anos de teste avaliados, mas validar a consistência das métricas estatísticas em diversas condições de treinamentos nos dados históricos. Desta forma pode-se avaliar se o algoritmo se comporta bem em todas as condições do mercado, sendo elas: alta, baixa e mercado lateralizado.

Os treinamentos foram realizados com janelas de treinamento de um total de 720 dias de treinamento, 240 de validação e 240 dias teste. Cada 240 dias representam um ano de dados históricos de preços das ações. Foram treinados 5 modelos para cada ação, iniciando-se no período de 2011 à 2013 para treinamento, 2014 para validação e 2015 para teste, incrementando-se um ano por vez para cada janela deslizante e terminando na janela de 2015 à 2017 para treinamento, 2018 para validação e 2019 para teste.

Abaixo temos os resultados nas tabelas 5, 6, 7, 8 e 9, para cada uma das ações avaliadas previamente com a janela fixa, agora podemos extrair a média e desvio padrão de cada uma das métricas estatísticas do modelo proposta para cada ação.

Tabela 5 – Métricas do Modelo DWT-GRU na ação PETR4 - Janela Deslizante

Métrica	Média	Desvio Padrão
Acurácia	0.746	$\pm 0.040$
Precisão	0.769	$\pm 0.023$
Revocação	0.746	$\pm 0.040$
F1	0.755	$\pm 0.032$

Tabela 6 – Métricas do Modelo DWT-GRU na ação VALE3 - Janela Deslizante

Métrica	Média	Desvio Padrão
Acurácia	0.731	$\pm 0.037$
Precisão	0.766	$\pm 0.038$
Revocação	0.731	$\pm 0.037$
F1	0.747	$\pm 0.037$

Tabela 7 – Métricas do Modelo DWT-GRU na ação ITUB4 - Janela Deslizante

Métrica	Média	Desvio Padrão
Acurácia	0.690	$\pm 0.040$
Precisão	0.730	$\pm 0.036$
Revocação	0.690	$\pm 0.040$
F1	0.708	$\pm 0.036$

Nota-se com este experimento que o algoritmo teve um desvio padrão de menos de 5% para todas as ações avaliadas. Todas as métricas superam 70%, validando estatisticamente o uso do algoritmo. Tendo em vista que a decisão foi comparada com o "ideal" conhecendo-se os reais pontos de inflexão da tendência determinados pelo sinal de fechamento original filtrado por *wavelets*, e estas métricas consideram apenas ações tomadas exatamente no dia correto, os

Tabela 8 – Métricas do Modelo DWT-GRU na ação ABEV3 - Janela Deslizante

Métrica	Média	Desvio Padrão
Acurácia	0.786	$\pm 0.025$
Precisão	0.809	$\pm 0.034$
Revocação	0.786	$\pm 0.025$
F1	0.796	$\pm 0.028$

Tabela 9 – Métricas do Modelo DWT-GRU na ação BOVA11 - Janela Deslizante

Métrica	Média	Desvio Padrão
Acurácia	0.770	$\pm 0.042$
Precisão	0.797	$\pm 0.030$
Revocação	0.770	$\pm 0.042$
F1	0.782	$\pm 0.036$

valores seriam mais elevados se considerássemos que ações tomadas um dia antes, ou em um dia depois, pudessem ser dadas como corretas.

Na Tabela 10 a seguir são comparados os valores do algoritmo treinado em janela fixa com os dados da janela deslizante para a ação da PETR4.

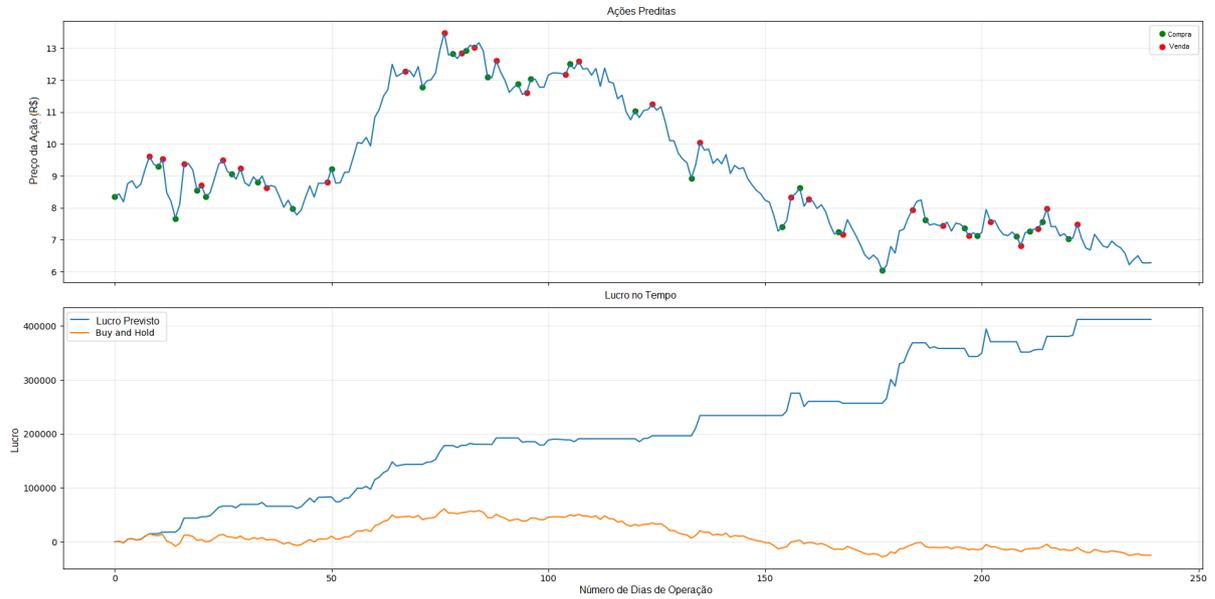
Tabela 10 – Comparação das métricas da janela fixa x deslizante para PETR4

Métricas	Janela Fixa	Janela Deslizante
Acurácia	<b>0.852</b>	0.746 $\pm 0.040$
Precisão	<b>0.875</b>	0.769 $\pm 0.023$
Revocação	<b>0.852</b>	0.746 $\pm 0.040$
F1	<b>0.861</b>	0.755 $\pm 0.032$

Os valores das métricas da janela fixa são notavelmente mais elevados do que os da janela deslizante, isso deve ao fato de a janela fixa ter sido treinada de 2001 à 2016, validado em 2017 e 2018, e testado em 2019, tendo portanto uma quantidade muito maior de dados para que se fosse feito o treinamento. Também é possível que haja influencia de atenuantes, pois como se trata de uma média entre 5 anos de teste avaliados, o ano de 2019 onde foi realizado o teste da janela fixa pode ter sido especialmente bom para a predição, enquanto os demais anos tiveram resultados inferiores diminuindo a média. Ainda assim a validade do algoritmo se mantém devido a sua média relativamente estável de acurácia e precisão em todos os ativos estudados.

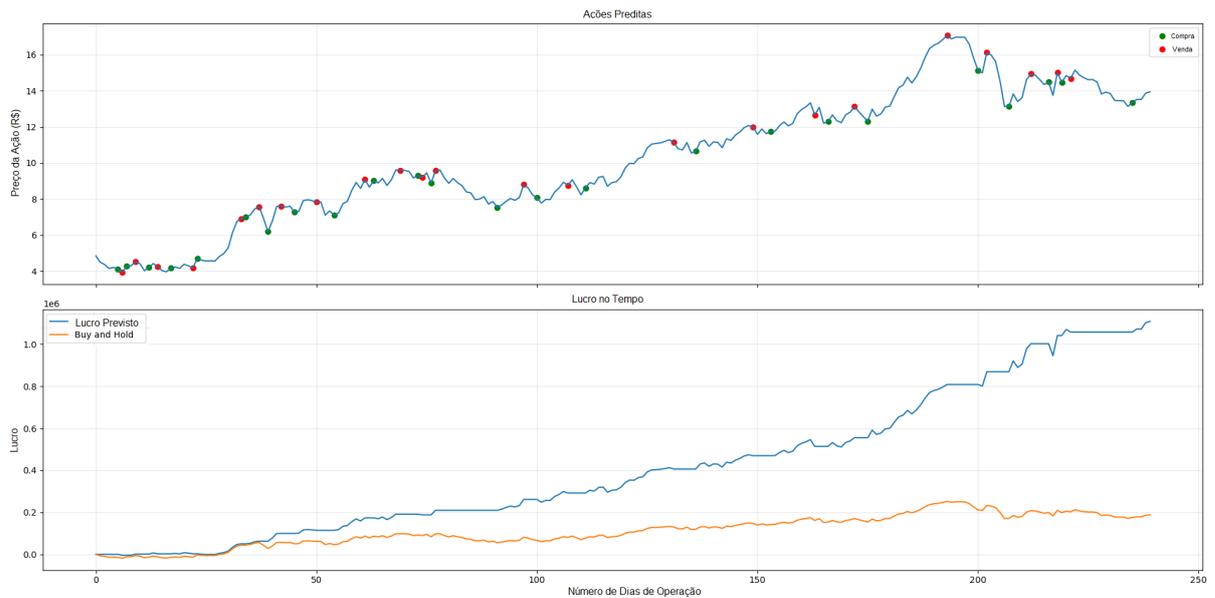
Podemos ver nas Figuras 25, 26, 27, 28 e 29, as decisões e a evolução do ROI no tempo de cada um dos anos avaliados.

Figura 25 – Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-GRU - Janela Deslizante em 2015



Fonte: Autoria Própria

Figura 26 – Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-GRU - Janela Deslizante em 2016



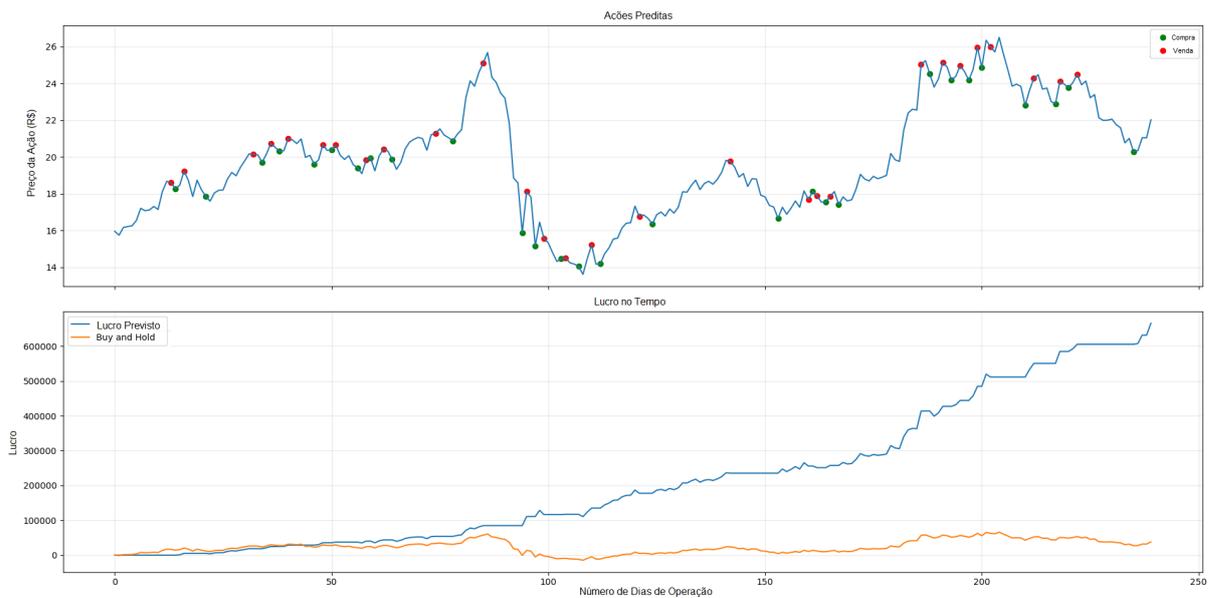
Fonte: Autoria Própria

Figura 27 – Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-GRU - Janela Deslizante em 2017



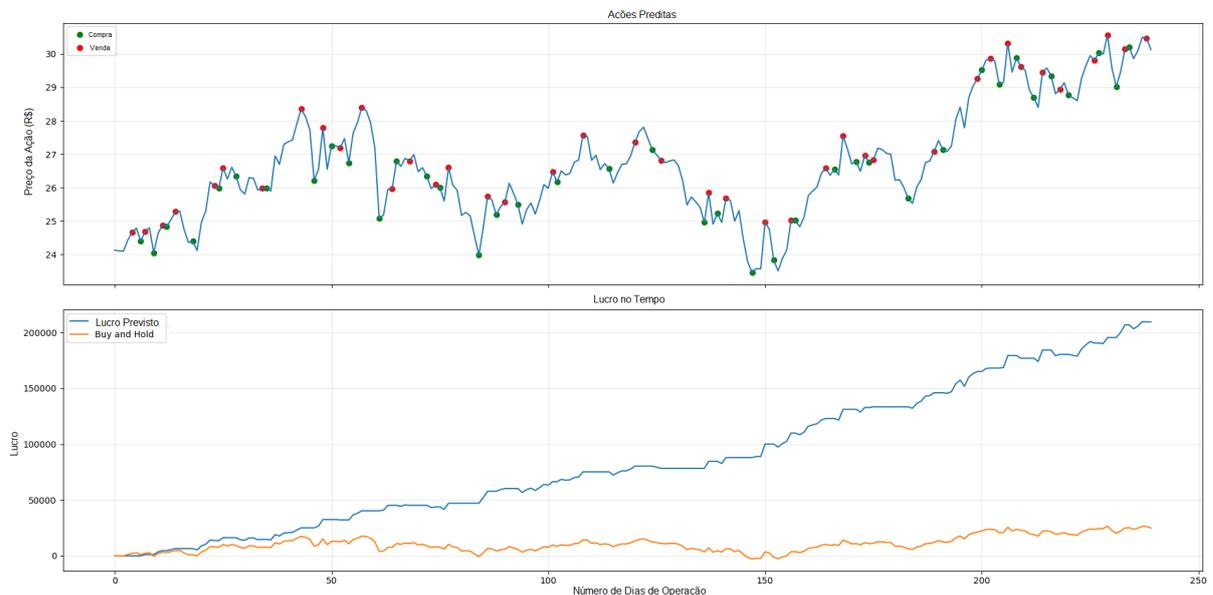
Fonte: Autoria Própria

Figura 28 – Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-GRU - Janela Deslizante em 2018



Fonte: Autoria Própria

Figura 29 – Ações do algoritmo e lucro na ação PETR4 - Modelo DWT-GRU - Janela Deslizante em 2019



Fonte: Autoria Própria

## 5.5 Normalização Local

Para verificar a diferença prática entre a aplicação da normalização de dados na tabela inteira de dados, ou seja nos dados em sua total extensão, e a aplicação da normalização local. Foi alterado no passo de pré-processamento de dados a normalização dos dados para normalizar apenas a amostra que será alimentada para a rede neural.

Abaixo se encontram as Tabelas 11, 12, 13, 14 e 15, que comparam as métricas estatísticas extraídas com treinamento em formato janela deslizante para também avaliarmos a média e desvio padrão.

Tabela 11 – Comparação das métricas da normalização global x normalização local para PETR4

Métricas	Norm. Global	Norm. Local
Acurácia	<b>0.746 ±0.040</b>	0.684 ±0.038
Precisão	<b>0.769 ±0.023</b>	0.687 ±0.037
Revocação	<b>0.746 ±0.040</b>	0.684 ±0.038
F1	<b>0.755 ±0.032</b>	0.685 ±0.038

A comparação entre as normalizações global e local mostram que não houve melhora ou piora significativa nas métricas em nenhuma das ações avaliadas. Em quase todos os casos a

Tabela 12 – Comparação das métricas da normalização global x normalização local para VALE3

Métricas	Norm. Global	Norm. Local
Acurácia	<b>0.731 ±0.037</b>	0.721 ±0.042
Precisão	<b>0.766 ±0.038</b>	0.724 ±0.039
Revocação	<b>0.731 ±0.037</b>	0.721 ±0.042
F1	<b>0.747 ±0.037</b>	0.722 ±0.041

Tabela 13 – Comparação das métricas da normalização global x normalização local para ITUB4

Métricas	Norm. Global	Norm. Local
Acurácia	0.690 ±0.040	<b>0.705 ±0.058</b>
Precisão	<b>0.730 ±0.036</b>	0.707 ±0.054
Revocação	0.690 ±0.040	<b>0.705 ±0.058</b>
F1	<b>0.708 ±0.038</b>	0.706 ±0.056

Tabela 14 – Comparação das métricas da normalização global x normalização local para ABEV3

Métricas	Norm. Global	Norm. Local
Acurácia	<b>0.786 ±0.025</b>	0.740 ±0.030
Precisão	<b>0.809 ±0.034</b>	0.744 ±0.035
Revocação	<b>0.786 ±0.025</b>	0.740 ±0.030
F1	<b>0.796 ±0.028</b>	0.741 ±0.032

normalização global se mostrou mais eficiente que a local, sendo que apenas na ação do ITUB4 duas das métricas (acurácia e revocação) foram maiores na normalização local que na global. Isto no entanto não invalida o uso da normalização local, já que embora a normalização local seja levemente inferior em capacidade de classificação das ações corretas a serem tomadas, seu uso possibilita a generalização do modelo para outras ações. Isto se dá devido aos modelos

Tabela 15 – Comparação das métricas da normalização global x normalização local para BOVA11

Métricas	Norm. Global	Norm. Local
Acurácia	<b>0.770 ±0.042</b>	0.698 ±0.019
Precisão	<b>0.797 ±0.030</b>	0.703 ±0.021
Revocação	<b>0.770 ±0.042</b>	0.698 ±0.019
F1	<b>0.782 ±0.036</b>	0.700 ±0.019

treinados em normalização global serem especializados para aquela distribuição de dados, e no caso da normalização local a variação depende apenas da janela amostral.

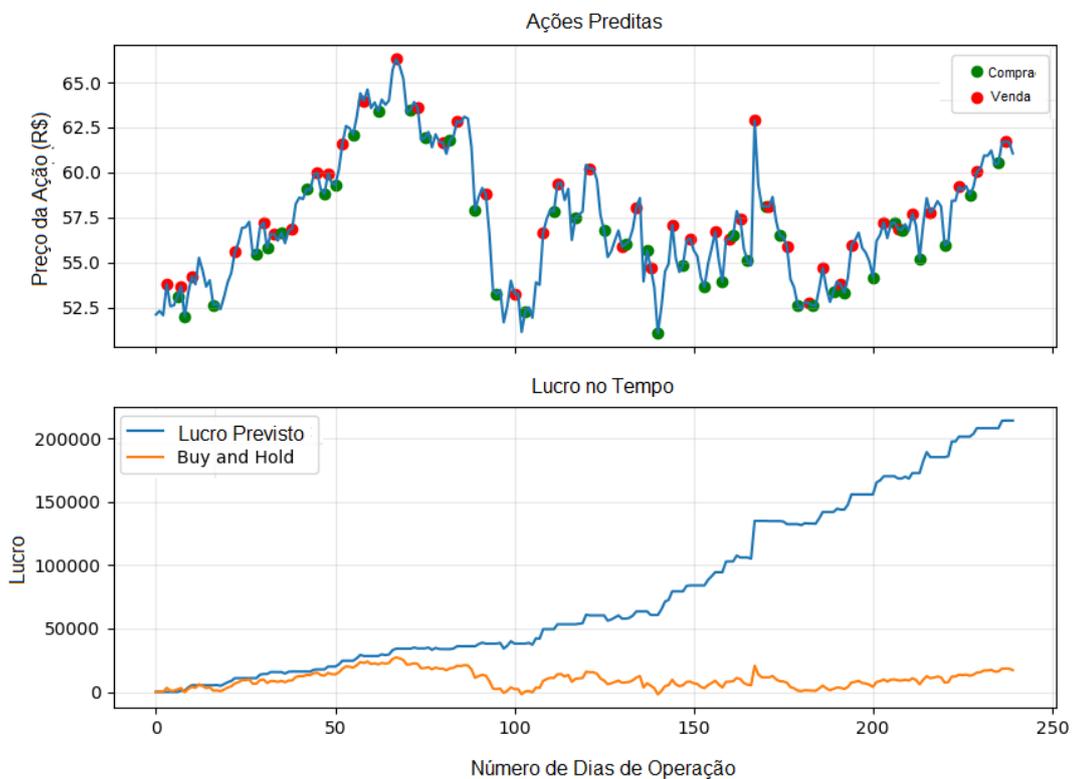
### 5.5.1 Aplicação de Modelo treinado à *commodities*

Abaixo podemos verificar o uso do modelo treinado na PETR4 sendo aplicado em três *commodities*, sendo elas o petróleo (CL=F) que é diretamente relacionado, e o ouro (GC=F) e prata (SI=F) que não tem relação com os papéis da empresa.

O modelo utilizado abaixo foi treinado nos dados históricos da PETR4 com janela fixa (treinamento 2010 - 2016, validação 2017 - 2018 e teste 2019), normalização local e testado no ano de 2019 dos dados históricos das três *commodities*.

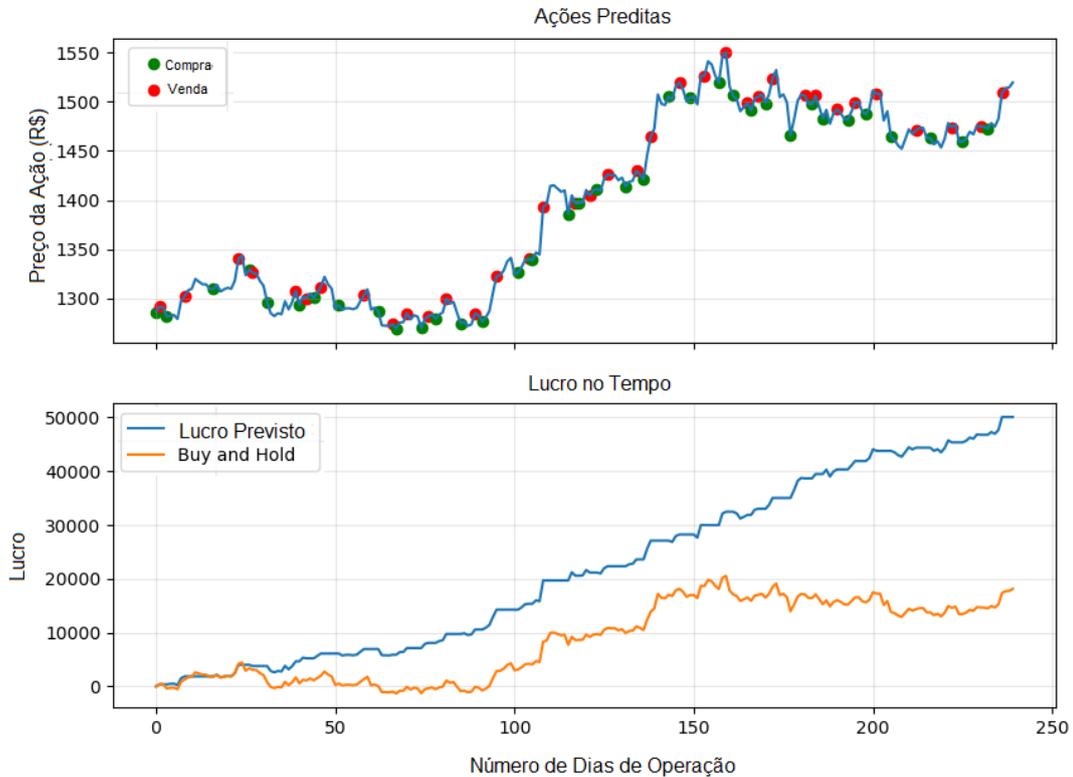
As figuras 30, 31 e 32, mostram a evolução das decisões do algoritmo treinado com normalização local nos dados históricos de preços da PETR4.

Figura 30 – Ações do algoritmo e lucro na *commodity* do petróleo CL=F - Modelo DWT-GRU - Normalização Local em 2019



Fonte: Autoria Própria

Figura 31 – Ações do algoritmo e lucro na *commodity* do ouro GC=F - Modelo DWT-GRU - Normalização Local em 2019



Fonte: Autoria Própria

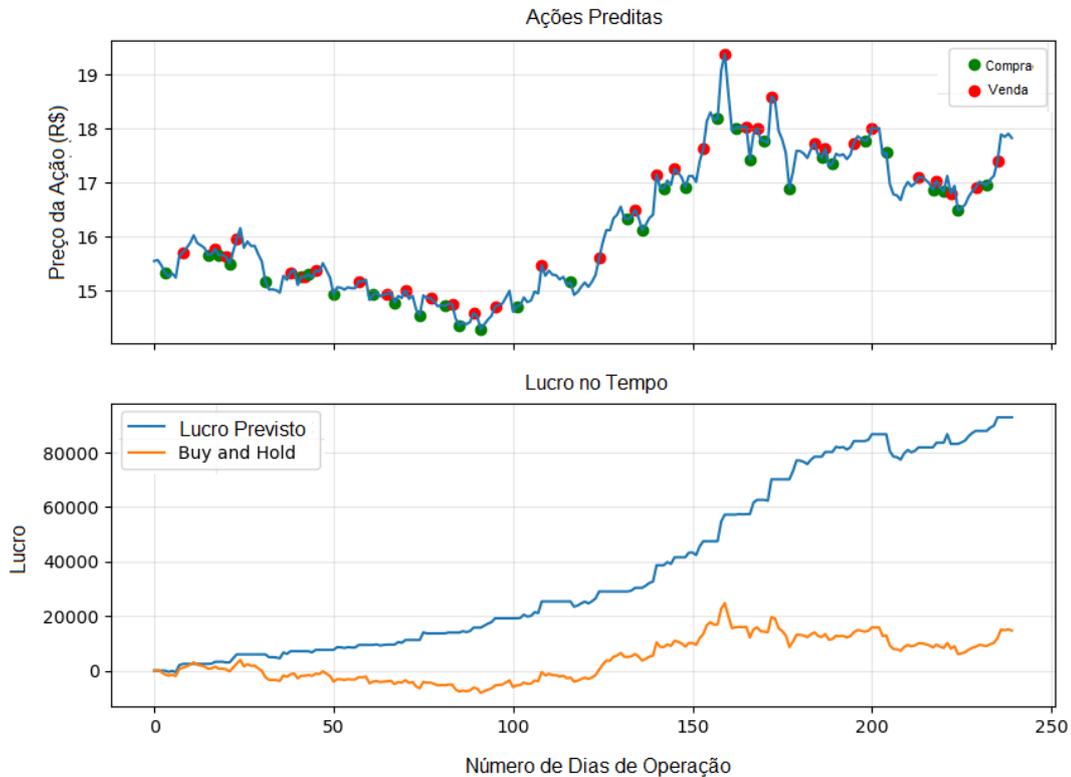
Como pode ser visto o ROI e as decisões tomadas pelo algoritmo foram pouco influenciados pela origem dos dados históricos, validando o uso do modelo treinado em normalização local a ser aplicado em outros ativos. Sendo o lucro de 219.47% para CL=F, 50.08% para GC=F, e 93.49% para SI=F, ganhando do BNH em todas as três aplicações.

Abaixo temos a Tabela 16 com as métricas estatísticas do modelo nas três *commodities* supracitadas, constatando sua eficácia na tomada correta de decisão mesmo não tendo sido treinado para os dados das mesmas.

Tabela 16 – Métricas do Modelo PETR4 com Normalização Local - *Commodities*

<i>commodity</i>	Acurácia	Precisão	Revocação	F1
CL=F	0.496	0.491	0.496	0.493
GC=F	0.471	0.441	0.471	0.455
SI=F	0.504	0.463	0.504	0.481

Figura 32 – Ações do algoritmo e lucro na *commodity* da prata SI=F - Modelo DWT-GRU - Normalização Local em 2019



Fonte: Autoria Própria

Com tais dados podemos verificar que embora eficaz o uso de um modelo treinado em dados históricos de outro ativo para fins de retorno financeiro, as decisões não são tomadas em sua grande maioria nos melhores momentos, podendo diminuir o lucro máximo do algoritmo. Desta forma pode-se contornar este desvio treinando o modelo com os dados históricos do próprio ativo, ou adicionar os dados de tal ativo no treinamento do modelo genérico, tornando-se um modelo não especializado por não ter sido treinado com dados de um único ativo. Esta segunda abordagem aumenta a eficácia média do modelos em diversos ativos, mas a torna inferior quando comparado com modelos especializados.

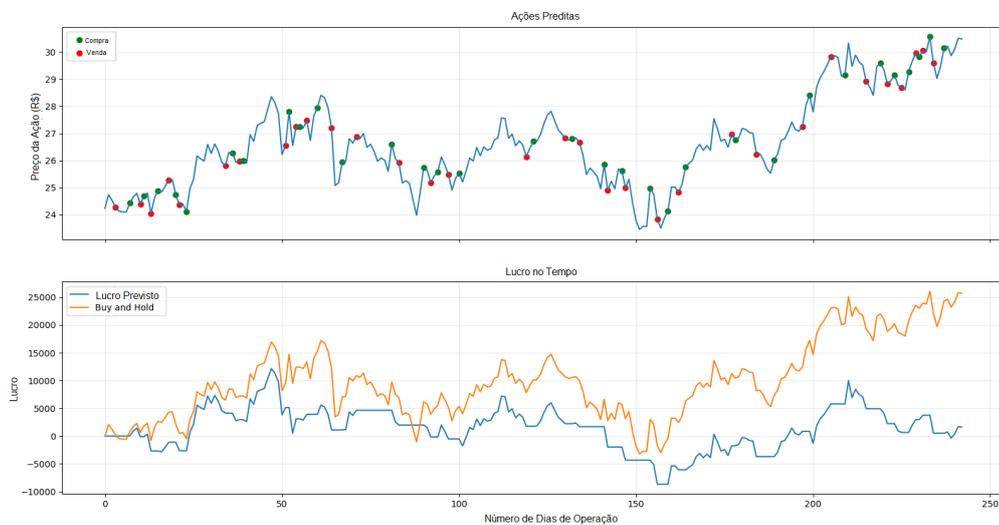
## 5.6 Modelos clássicos de aprendizado de máquina

Para verificar o impacto do uso do pré-processamento de dados com a DWT foi formulado um teste utilizando os dados históricos aliados os mesmos indicadores técnicos que foram utilizados para treinamento dos modelos de RNNs. Foram utilizados dois modelos já expli-

cados previamente, sendo eles a Regressão Linear e a Floresta Aleatória. Para cada modelo foram treinados com dados de 2010 a 2018, e testados no ano de 2019 para comparação com os resultados anteriores do modelo proposto. A normalização utilizada foi a global e a janela de treinamento fixa.

As Figuras 33, 34, 35 e 36, mostram as decisões e o ROI acumulado de cada um dos modelos, com e sem o pré processamento de *wavelets*.

Figura 33 – Ações do algoritmo e lucro na PETR4 - Modelo Floresta Aleatória sem DWT em 2019



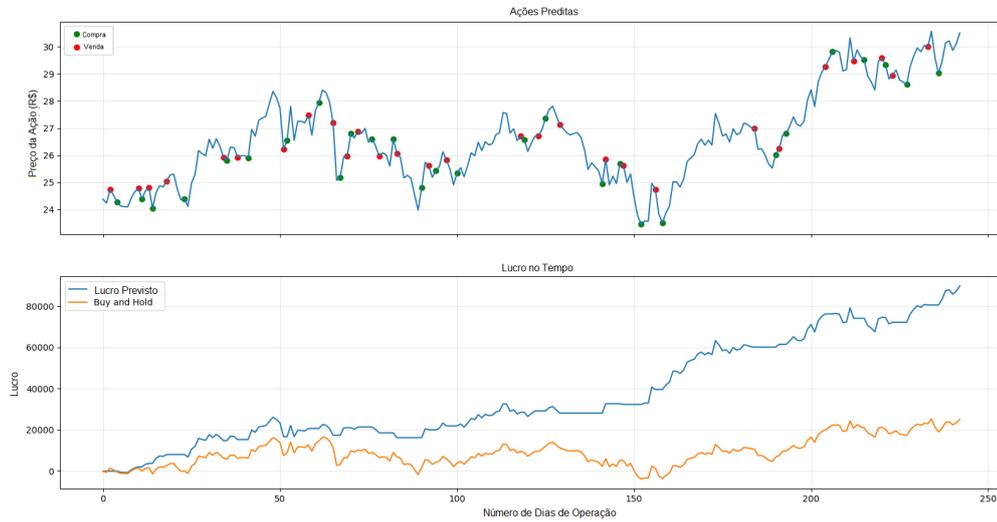
Fonte: Autoria Própria

É notável que os modelos foram beneficiados pela aplicação da DWT no pré processamento e como alvo de treinamento e predição para identificação dos pontos de mudança de tendencia, já que sem este as ações tomadas pelos algoritmos se assemelham a tomadas de decisão aleatórias.

As métricas estatísticas também melhoraram com o uso da DWT, e embora isto não seja determinante para o resultado financeiro como um todo, mostra que o uso das *wavelets* influencia positivamente na melhora do resultado do algoritmo. Abaixo seguem as tabelas 17 e 18 e mostram a comparação das métricas entre os modelos com e sem o uso da DWT.

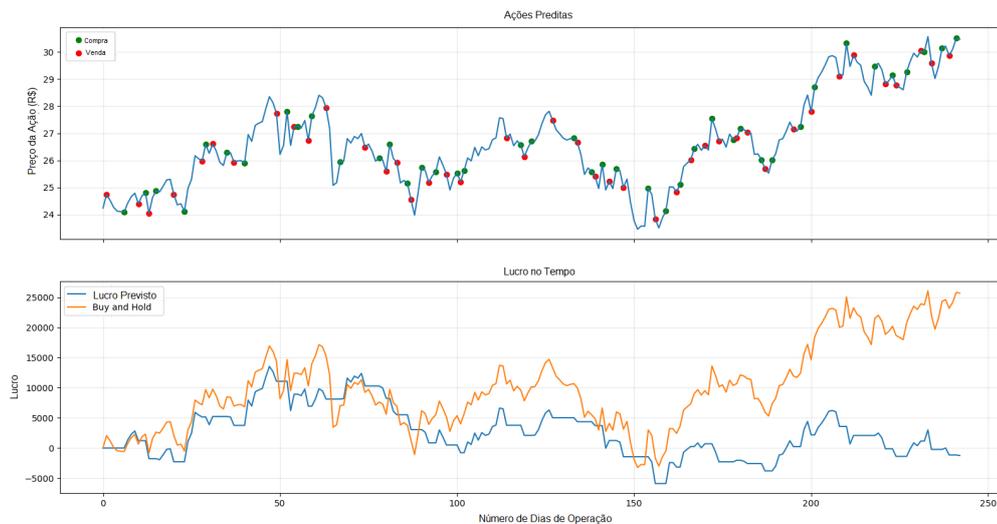
É possível perceber pelos valores das métricas que especialmente para a regressão linear as métricas foram maiores que inclusive as da floresta aleatória com DWT. No entanto o resultado financeiro foi menor. Isto caracteriza que as métricas estatísticas dão uma boa ideia

Figura 34 – Ações do algoritmo e lucro na PETR4 - Modelo Floresta Aleatória com DWT em 2019



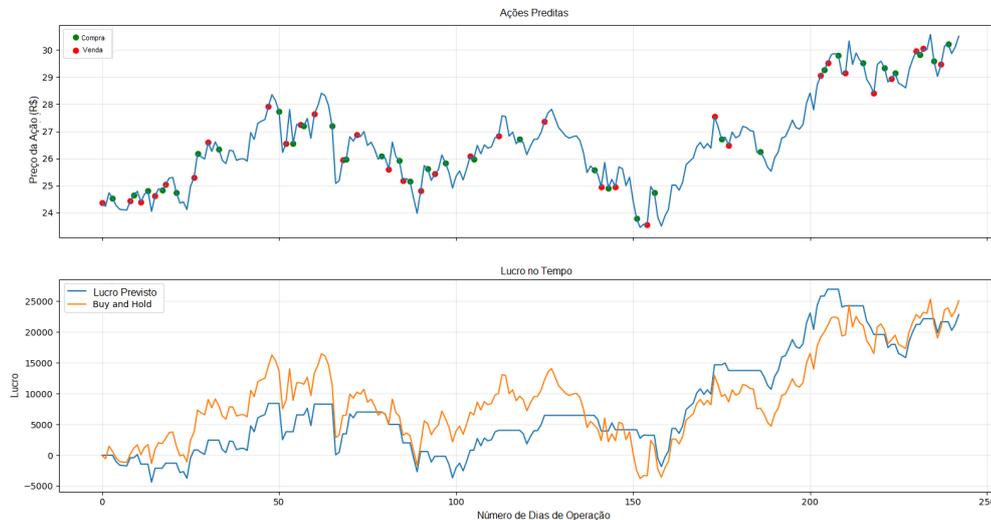
Fonte: Autoria Própria

Figura 35 – Ações do algoritmo e lucro na PETR4 - Modelo Regressão Linear sem DWT em 2019



Fonte: Autoria Própria

Figura 36 – Ações do algoritmo e lucro na PETR4 - Modelo Regressão Linear com DWT em 2019



Fonte: Autoria Própria

Tabela 17 – Comparação das métricas da floresta aleatória com e sem DWT

Métricas	Sem DWT	Com DWT
Acurácia	0.572	0.629
Precisão	0.548	0.646
Revocação	0.572	0.629
F1	0.559	0.637

Tabela 18 – Comparação das métricas da regressão linear com e sem DWT

Métricas	Sem DWT	Com DWT
Acurácia	0.518	0.637
Precisão	0.540	0.683
Revocação	0.518	0.637
F1	0.528	0.658

da qualidade do algoritmo, mas não necessariamente representam um retorno financeiro mais alto.

Também podemos notar que nos modelos onde não foi utilizada a DWT houveram muito mais ações tomadas. Este fato contribui para que caso sejam tomadas em momentos errados, como foram em grande parte do tempo, esta frequência maior de operação pode piorar muito o resultado financeiro do modelo.

## 5.7 Testes em Tempo Real

Após a validação do treinamento dos modelos e sua avaliação com dados históricos, o próximo passo foi dado em se avaliar os resultados da pesquisa aplicando o conhecimento adquirido pelas redes em prever e executar suas decisões com dados em tempo real. Para isto é utilizado o software de negociação MetaTrader 5<sup>3</sup> através de sua Interface de Programação de Aplicações - *Application Programming Interface* (API)<sup>4</sup> homônima para Python.

Estes testes terão como finalidade a validação do sistema proposto em uma aplicação real. Verificando sua eficácia em ambiente independente com interação baseada apenas na comunicação do preço atual do ativo e das operações realizada a partir da tomada de decisão feita pelo sistema. Foi feita a coleta de dados dos testes em tempo real e uma comparação com as decisões que o algoritmo tomaria caso tivesse as informações do período avaliado em forma de dados históricos.

### 5.7.1 Resultados e Discussão

Os testes em tempo real foram executados utilizando a API de conexão do Python com o MetaTrader 5 no período de 22/09/2020 até 13/11/2020 para verificação da capacidade de decisão do algoritmo com dados em tempo real fornecidos pela API.

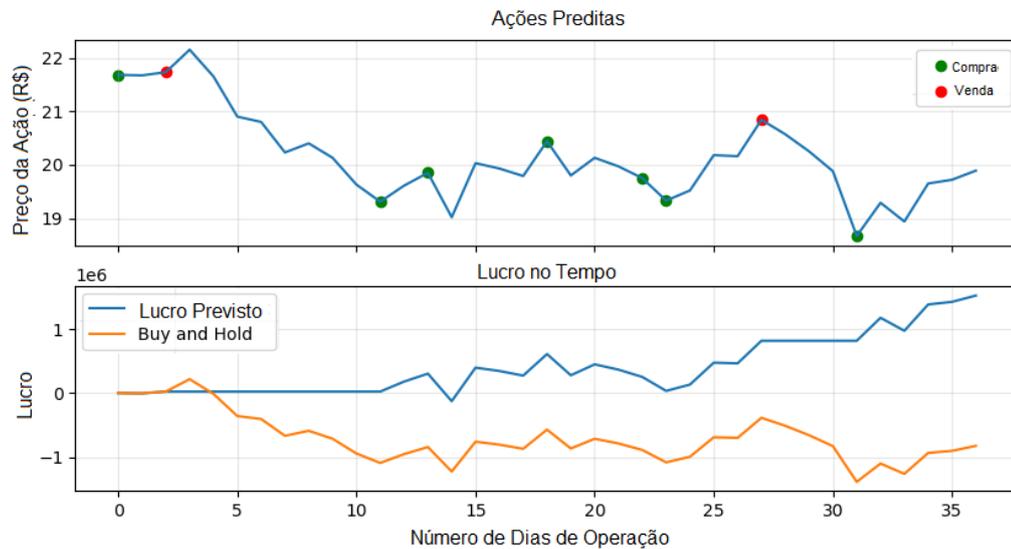
Na Figura 37 são ilustradas as decisões tomadas pelo algoritmo em tempo real executando as decisões todos os dias úteis entre as 16:00h e as 17:00h. Os recursos disponíveis para teste foram os R\$10.000.000,00 na conta de demonstração disponibilizada pela XP Inc.. Todas as decisões foram executadas comprando o número de cotas possíveis, múltiplas de 100, com todo o capital disponível. Sendo executada apenas o primeiro sinal de compra ou de venda, não sendo feitas compras ou vendas a descoberto.

Como é possível notar o modelo se comportou melhor do que o BNH no período, que embora seja relativamente curto, já conseguiu um resultado financeiro expressivamente melhor que a estratégia base. Nota-se que houveram alguns sinais de "compra" seguidos, provavelmente devido a indicações pontuais nos dados que causaram no algoritmo variações nas predições do sinal filtrado e mudando então seus pontos de inflexão. Como o algoritmo sempre executa as ações com todo o capital disponível, o mesmo não foi afetado pelas consecutivas

<sup>3</sup><https://www.metatrader5.com/pt> [Accessed: 26/07/2020]

<sup>4</sup>[mql5.com/pt/docs/integration/python\\_metatrader5](https://mql5.com/pt/docs/integration/python_metatrader5) [Accessed: 26/07/2020]

Figura 37 – Ações do algoritmo e lucro na PETR4 - Modelo DWT-GRU no período de 22/09/2020 à 13/11/2020



Fonte: Autoria Própria

ordens de compra, pois executou apenas a primeira. O algoritmo acumulou um ROI de aproximadamente 15.2% no período, contra -8.2% da estratégia BNH, totalizando uma diferença de 23.4%.

Para verificação da validade estatística da execução do modelo em tempo real, foram comparadas as ações no período com as decisões que seriam tomadas conhecendo-se os dados de preço. Na Tabela 19 abaixo seguem as métricas estatísticas do modelo em comparação com as decisões "ideais" a serem tomadas.

Tabela 19 – Métricas estatísticas do Modelo DWT-GRU no período de 22/09/2020 à 13/11/2020

Modelo	Acurácia	Precisão	Revocação	F1
DWT-GRU	0.6875	0.730	0.6875	0.674

Embora as métricas estejam em valores considerados satisfatórios, as mesmas não podem ser consideradas expressivas para esta análise, uma vez que como o modelo sendo testado em tempo real, este pode tomar ações de compra ou venda repetidas e o resultado das métricas é fortemente afetado por tal fato sem que necessariamente seja prejudicado o resultado financeiro do modelo.

No entanto o período de avaliação do modelo em tempo real é considerado curto prazo. Isto nos leva a avaliar a necessidade de adicionar a esta pesquisa dados de testes em um período maior para validar no médio e longo prazo a efetividade do modelo.

Para fins de validação do teste o relatório extraído do MetaTrader segue na Figura 38 abaixo. Como pode-se notar o saldo final do teste é de R\$ 11.523.876,00 com a atual posição comprada como mostrado na Figura 37. Como dito anteriormente totalizando o ROI de 15.2%.

Figura 38 – Relatório do MetaTrader5 para as operações - Modelo DWT-GRU no período de 22/09/2020 à 13/11/2020

<b>Relatório da Conta de Negociação</b>											
Nome: <b>VICTOR BIAZON</b>											
Conta: <b>93128556 (BRL, 1:1, XPMT5-Demo, demo, Netting)</b>											
Corretora: <b>XP Investimentos CCTVM S/A</b>											
Data: <b>2020.11.13 20:12</b>											
<b>Posições</b>											
Ativo	Position	Horário	Tipo	Volume	Preço	S / L	T / P	Preço de Mercado	Swap	Lucro	Comentário
PETR4	513767833	2020.11.06 17:01:50	buy	509.2K	18.67			22.63	0.00	706.876	Buy Order
									<b>0.00</b>	<b>706.876</b>	
		Saldo:		<b>11 523 876.00</b>				Margem Livre:		<b>8 819 615.70</b>	
		Facilidade do Crédito:		<b>0.00</b>				Margem:		<b>464 141.30</b>	
		Flutuante P/L:		<b>0.00</b>				Nível de Margem:		<b>2 000.20%</b>	
		Capital Líquido:		<b>11 523 876.00</b>							

Fonte: Autoria Própria

## 5.7.2 Emolumentos

A B3 (Bolsa Brasil Balcão) gerencia as operações de papéis no território brasileiro e surgiu oficialmente em 2017, logo após a fusão entre a BM&F Bovespa (Bolsa de Valores, Mercadorias e Futuros de São Paulo) e a Cetip (Central de Custódia e de Liquidação Financeira de Títulos)<sup>5</sup>.

Para se realizar operações no bolsa brasileira são cobradas taxas de operação chamadas Emolumentos, que para toda operação de negociação ou liquidação subtrai uma porcentagem do montante total negociado como taxa de operação. Na tabela 20 abaixo seguem as taxas de operação para pessoas físicas e fundos de investimento<sup>6</sup>.

<sup>5</sup><https://www.sunoresearch.com.br/artigos/b3/>

<sup>6</sup>[http://www.b3.com.br/pt\\_br/produtos-e-servicos/tarifas/listados-a-vista-e-derivativos/renda-variavel/tarifas-de-acoes-e-fundos-de-investimento/a-vista/](http://www.b3.com.br/pt_br/produtos-e-servicos/tarifas/listados-a-vista-e-derivativos/renda-variavel/tarifas-de-acoes-e-fundos-de-investimento/a-vista/)

Tabela 20 – Taxas de emolumentos B3 para negociação e liquidação

Investidor	Negociação	Liquidação	Total
Pessoas físicas e demais investidores	0,003247%	0,0275%	0,030747%
Fundos e clubes de investimento locais	0,003247%	0,0200%	0,023247%

Para Avaliar o impacto dos emolumentos no resultado financeiro, uma vez que a estratégia BNH seria injustamente comparada com os modelos caso não fossem consideradas taxas, foi calculada o desconto médio por execução para verificar se podemos ou não desconsiderar as taxas da B3 assim como foi desconsiderada a corretagem.

Foi constatado que os modelos executam de 50 - 100 operações por ano, sendo a média 75 operações e divisão de aproximadamente 50% para cada decisão possível (compra ou venda). Baseado neste fato, foram consideradas em média 37 negociações de compra e 37 liquidações (venda). Consideramos para este estudo as taxas para investidores de pessoa física e a taxa total por operação de 0.030747%. O lucro médio entre os modelos que utilizaram a DWT foi de 101.61%. Isto representa um acúmulo no balanço final médio de R\$201.610,00 e sendo descontado o valor inicial de investimento de R\$100.000,00, o capital médio durante todo o período de operação foi de R\$150.810,00. Com estes valores em mãos podemos calcular o desconto médio por operação que resulta de  $DescMédio = R\$150.810,00 * 0,00030747 = R\$46,36$ . E o valor médio total descontado sendo  $Desc.Emolumentos = 75 * R\$46,36 = R\$3477,00$ . Este valor representa apenas 1,72% de desconto no montante total, e como a diferença entre o BNH e o algoritmo proposto sempre foi de mais de 40% não é necessário a consideração desta taxa para comparação justa.

### 5.7.3 Resumo da Seção

Como pudemos verificar nesta seção, o modelo proposto teve os melhores resultados de métricas estatísticas e de retorno financeiro entre todas as comparações feitas. Também foi verificada a validade do treinamento em diversas janelas de tempo (janela deslizante) que, embora tenham tido um resultado inferior ao de janela fixa, mostra que os desvios padrões do treinamento foram baixos constatando baixa variação na qualidade do treinamento do modelo proposto independente de quais dados foram utilizados para o treinamento.

Outra aplicação verificada foi a diferença dos resultados para treinamentos em normalização local e global, e constatado que embora a normalização global tenha melhores resultados

estatísticos e de retorno financeiro, a normalização local possibilita o uso do algoritmo em outros ativos os quais os dados não foram inseridos durante o treinamento, e isto foi validado testando o modelo treinado na PETR4 nas *commodities* do petróleo, ouro e prata.

Também verificamos que embora os modelos clássicos de aprendizado de máquina, regressão linear e floresta aleatória, não tenham tido resultado financeiro tão expressivo quanto o modelo proposto, a qualidade do seu aprendizado foi fortemente influenciada pelo pré processamento de dados feito com a DWT reforçando sua validade como contribuição principal desta pesquisa.

Por fim foi apresentado o resultado financeiro da aplicação do modelo em tempo real que mostra que é possível utilizá-lo como ferramenta de operação em aplicações do mundo real.

## 6 CONCLUSÃO

Neste trabalho foi proposto uma comparação entre modelos RNN aliados, ou não, com Transformada Discreta de Ondas - *Discrete Wavelet Transforms* (DWT) para processar os dados de uma ação do mercado financeiro para ajudar a entender melhor as tendências em cada uma das ações analisadas. O sistema desenvolvido foi comparado com redes GRU e LSTM puras e com o modelo DWT-LSTM, com a estratégia Comprar e Segurar - *Buy and Hold* (BNH) e particularmente no caso do Fundo de Índice Negociado em Bolsa - *Exchange-Traded Fund* (ETF) BOVA11 com um fundo de investimentos independente chamado Alaska Black.

Pôde-se observar que, em geral, o modelo proposto teve um melhor Retorno Sobre Investimento - *Return Over Investment* (ROI) que os algoritmos comparados e a estratégia Comprar e Segurar - *Buy and Hold* (BNH). A implementação do modelo DWT-GRU mostra resultados promissores sobre a capacidade atual de prever e tomar ações no mercado de ações. As métricas estatísticas do modelo também foram satisfatórias, o que valida a aplicação do modelo em outros experimentos com ações e a continuidade desta pesquisa.

Foi apresentada também a comparação da comparação entre o treinamento de janela fixa e de janela deslizante, onde foi constatado que o de janela fixa teve métricas e resultado financeiro mais elevados devido a sua maior disponibilidade de padrões para aprendizado durante o treinamento. No entanto o treinamento com janela deslizante mostrou que, apesar de ter resultados inferiores na comparação, independente dos dados escolhidos para o treinamento o algoritmo proposta consegue generalizar os padrões de comportamento dos preços o suficiente para acumular resultado financeiro satisfatório em todas as condições testadas.

Também foi proposta uma forma alternativa de normalização dos dados, sendo esta a normalização local das amostras, normalizando a cada 5 dias de dados mais a saída esperada, para que fosse possível avaliar se a distribuição simples dos dados das amostras seria o suficiente para o treinamento. Com este teste foi verificado que, embora o resultado estatístico seja inferior ao da normalização global, este método resulta num modelo treinado de forma "genérica" não sendo necessariamente especializado à distribuição de dados de um único ativo financeiro. Isto possibilita o uso deste modelo treinado em outros ativos não sendo necessário seu treinamento novamente na nova distribuição de dados.

Para testar esta generalidade do modelo treinado em normalização local foi realizado o teste do mesmo em três *commodities* (ouro, prata e petróleo) e constatado que o modelo,

apesar de não ser especializado para tais distribuições de dados, consegue realizar as operações satisfatoriamente nos três ativos mesmo sem ter sido treinado para tal.

Foram realizados também testes da capacidade de classificação correta das decisões a serem tomadas por algoritmos clássicos de aprendizado de máquina, verificando assim a influência do pré processamento proposto na pesquisa na qualidade do aprendizado dos algoritmos. Com este teste foi constatado que, embora os algoritmos mais simples não tenham sido tão lucrativos ou tenham tido métricas estatísticas tão altas quanto o modelo proposto, os mesmos foram beneficiados pelo pré processamento da DWT. Isto nos reafirma que a principal contribuição desta pesquisa, além do modelo proposto, é a abordagem no tratamento dos dados para alimentação ao treinamento dos algoritmos.

O modelo foi testado em tempo real junto ao *software* MetaTrader5 para averiguação da capacidade de operação em tempo real do mesmo. Foi constatado que o algoritmo conseguiu operar satisfatoriamente durante o período de teste acumulando significativamente mais lucros do que o Comprar e Segurar - *Buy and Hold* e validando a aplicação do mesmo em ambiente não controlado.

É importante ressaltar que devido à análise ter sido feita em dados do mercado brasileiro de valores, os resultados desta pesquisa se aplicam com certeza apenas a este. Para análise em outros mercados e ativos, a pesquisa deve ser novamente validada.

Finalmente, uma contribuição deste trabalho de pesquisa foi a publicação de um artigo científico com a compilação dos resultados iniciais do mesmo no Encontro Nacional de Inteligência Artificial e Computacional (ENIAC) do ano de 2020. O artigo foi publicado como *Gated Recurrent Unit Networks and Discrete Wavelet Transforms Applied to Forecasting and Trading in the Stock Market* e relata a metodologia e experimentos na BOVESPA que deram início a este trabalho. Também é importante ressaltar que este ficou em primeiro lugar como melhor artigo do congresso em questão, validando a relevância desta pesquisa junto a comunidade científica brasileira.

## 6.1 Trabalhos Futuros

Para trabalhos futuros existe a necessidade de testar o algoritmo proposto em aplicações do mundo real de médio e longo prazo, para validar sua real capacidade de operar em dados em tempo real com mais dados e outras ações.

Também é válida a implementação e treinamento de um modelo genérico em uma grande distribuição de dados provenientes de diversas ações, commodities e cripto-moedas para que o modelo consiga absorver o maior número de padrões possível e assim verificar a sua capacidade de operação em uma ação a qual ele não tenha sido treinado. Isso poderá estabelecer um modelo que ao aprender com dados de diversas fontes se adapta a qualquer ação, sem que precise do histórico de preços da mesma para treinamento, assim utilizando apenas os dados recentes e os indicadores técnicos.

Para otimização do treinamento e da predição, é viável a aplicação de algoritmos evolucionários e genéticos com a finalidade de melhora da retenção de informação útil. Outros artigos e trabalhos mostraram que o uso destes algoritmos beneficia o modelo tanto em resultado financeiro quanto nas métricas estatísticas, validando a busca de sua implementação futura.

Para mitigar a variância de resultados do treinamento dos modelos, devido a sua aleatoriedade de inicialização dos pesos das redes, é possível a implementação futura de modelos de *ensemble* nos mesmos. Assim é possível combinar a predição de diversos modelos treinados e extraíndo-se a média dos mesmos, assim como o algoritmo de floresta aleatória faz, extrair uma predição mais precisa. Isto retiraria a dependência do modelo de uma "boa inicialização" que ocorre ao acaso.

Para interferência direta na decisão do modelo, será de grande valia a criação de parâmetros que um supervisor consiga alterar e modificar o comportamento do sistema e a forma como esse opera no mercado. Desta forma podendo atuar de forma diferente em períodos de crise como o deste ano corrente de 2020.

Como o algoritmo pode ser treinado em normalização local, é possível o teste do mesmo para outros períodos de operação tanto mais longos (semanas, meses) como mais curtos (horas, minutos). Isto pode ser verificado utilizando os dados do próprio MetaTrader5 no futuro tanto para aumentar a quantidade de dados de treinamento, quanto para operação do algoritmo em tempo real.

## REFERÊNCIAS

- ALONSO-MONSALVE, Saúl et al. Convolution on Neural Networks for High-Frequency Trend Prediction of Cryptocurrency Exchange Rates Using Technical Indicators. **Expert Systems with Applications**, Elsevier, p. 113250, 2020.
- ALPAYDIN, Ethem. **Introduction to machine learning**. [S.l.]: MIT press, 2020.
- ALTAN, Aytaç; KARASU, Seçkin; BEKİROS, Stelios. Digital currency forecasting with chaotic meta-heuristic bio-inspired signal processing techniques. **Chaos, Solitons & Fractals**, Elsevier, v. 126, p. 325–336, 2019.
- APPEL, Gerald. **The moving average convergence-divergence trading method: advanced version**. [S.l.]: Scientific Investment Systems, 1985.
- ARMS JR, Richard W. Ease Of Movement. **Technical Analysis of STOCKS**, 1990.
- BIGIOTTI, Alessandro; NAVARRA, Alfredo. Optimizing Automated Trading Systems. In: SPRINGER. THE 2018 International Conference on Digital Science. [S.l.: s.n.], 2018. P. 254–261.
- BLAU, William. Technical Analysis of Stocks and Commodities: True Strength Index. v. 9, p. 438–446.
- BOLLINGER, John. Using bollinger bands. **Stocks & Commodities**, v. 10, n. 2, p. 47–51, 1992.
- BOTES, Etienne; SIEPMAN, Douglas. The vortex indicator. **Technical Analysis of Stocks & Commodities**, v. 28, n. 1, p. 20–30, 2010.
- BROWN, Stephen J; GOETZMANN, William N; KUMAR, Alok. The Dow theory: William Peter Hamilton’s track record reconsidered. **The Journal of finance**, Wiley Online Library, v. 53, n. 4, p. 1311–1333, 1998.
- BULKOWSKI, Thomas N. **Encyclopedia of chart patterns**. [S.l.]: John Wiley & Sons, 2011. v. 225.
- BURKOV, Andriy. **The hundred-page machine learning book**. [S.l.]: Andriy Burkov Quebec City, Can., 2019. v. 1.
- CHO, Kyunghyun et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014.
- CHOLLET, François et al. **Keras**. [S.l.: s.n.], 2015. <https://keras.io>.
- CHUNG, Junyoung et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. **arXiv preprint arXiv:1412.3555**, 2014.

DAUBECHIES, Ingrid. **Ten lectures on wavelets**. [S.l.]: Siam, 1992. v. 61.

DI LORENZO, Renato. Volumes. In: **BASIC Technical Analysis of Financial Markets**. [S.l.]: Springer, 2013. P. 169–188.

DONCHIAN, Richard D. Trend-Following Methods in Commodity Price Analysis. **Commodity Year Book**, p. 35–47, 1957.

DORMEIER, Buff. **Investing with Volume Analysis: Identify, Follow, and Profit from Trends**. [S.l.]: FT Press, 2011.

DORSEY, Donald. Refining the Relative Volatility Index. **Stocks & Commodities**, v. 13, p. 388–391, 1995.

EHLERS, John F. Adaptive Trends And Oscillators. **TECHNICAL ANALYSIS OF STOCKS AND COMMODITIES-MAGAZINE EDITION-**, TECHNICAL ANALYSIS, INC., v. 18, n. 5, p. 18–20, 2000.

ELDER, Alexander. **Trading for a living: psychology, trading tactics, money management**. [S.l.]: John Wiley & Sons, 1993. v. 31.

GALLI, AW; HEYDT, GT; RIBEIRO, PF. Exploring the power of wavelet analysis. **IEEE Computer Applications in Power**, IEEE, v. 9, n. 4, p. 37–41, 1996.

GILLES, Jerome. Empirical wavelet transform. **IEEE transactions on signal processing**, IEEE, v. 61, n. 16, p. 3999–4010, 2013.

GOLD, Steven. The Viability of Six Popular Technical Analysis Trading Rules in Determining Effective Buy and Sell Signals:: MACD, AROON, RSI, SO, OBV, and ADL. **Journal of Applied Financial Research**, Academy of Business Research, v. 2, p. 8, 2015.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. [S.l.]: MIT press, 2016.

GOPALSWAMY, Shruthi; TIGHE, Patrick J; RASHIDI, Parisa. Deep recurrent neural networks for predicting intraoperative and postoperative outcomes and trends. In: IEEE. 2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI). [S.l.: s.n.], 2017. P. 361–364.

GRANVILLE, Joseph Ensign. **Granville's New Strategy of Daily Stock Market Timing for Maximum Profit**. [S.l.]: Prentice-Hall, 1976.

HAYKIN, Simon S et al. **Neural networks and learning machines/Simon Haykin**. [S.l.]: New York: Prentice Hall, 2009.

HIRANSHA, M et al. NSE stock market prediction using deep-learning models. **Procedia computer science**, Elsevier, v. 132, p. 1351–1362, 2018.

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

HSIEH, Tsung-Jung; HSIAO, Hsiao-Fen; YEH, Wei-Chang. Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. **Applied soft computing**, Elsevier, v. 11, n. 2, p. 2510–2525, 2011.

HUTSON, Jack K. TRIX-triple exponential smoothing oscillator. **Technical Analysis of Stocks and Commodities**, p. 105–108, 1983.

HUYNH, Huy D; DANG, L Minh; DUONG, Duc. A new model for stock price movements prediction using deep neural network. In: PROCEEDINGS of the Eighth International Symposium on Information and Communication Technology. [S.l.: s.n.], 2017. P. 57–62.

KAUFFMAN, Stuart; MACREADY, William. Technological evolution and adaptive organizations: Ideas from biology may find applications in economics. **Complexity**, Wiley Online Library, v. 1, n. 2, p. 26–43, 1995.

KELTNER, Chester W. **How to make money in commodities**. [S.l.]: Keltner Statistical Service, 1960.

LAMBERT, Donald R. Commodity channel index: Tool for trading cyclic trends. **Technical Analysis of Stocks & Commodities**, v. 1, p. 47, 1983.

LANE, George C. Lane's stochastics.

LAWRANCE, AJ; LEWIS, PAW. An exponential moving-average sequence and point process (EMA1). **Journal of Applied Probability**, Cambridge University Press, v. 14, n. 1, p. 98–113, 1977.

MALKIEL, Burton G; FAMA, Eugene F. Efficient capital markets: A review of theory and empirical work. **The journal of Finance**, Wiley Online Library, v. 25, n. 2, p. 383–417, 1970.

MALLAT, Stéphane. **A wavelet tour of signal processing**. [S.l.]: Elsevier, 1999.

MARTÍN ABADI et al. **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. [S.l.: s.n.], 2015. Software available from tensorflow.org. Disponível em: <<http://tensorflow.org/>>.

MERRY, RJE. Wavelet theory and applications: a literature study. **DCT rapporten**, Technische Universiteit Eindhoven, v. 2005, 2005.

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

NARCOUZI, Christopher. Chaikin's Money Flow. **TECHNICAL ANALYSIS OF STOCKS AND COMMODITIES-MAGAZINE EDITION-**, TECHNICAL ANALYSIS, INC., v. 18, n. 8, p. 42–47, 2000.

NELSON, David MQ; PEREIRA, Adriano CM; OLIVEIRA, Renato A de. Stock market's price movement prediction with LSTM neural networks. In: IEEE. 2017 International joint conference on neural networks (IJCNN). [S.l.: s.n.], 2017. P. 1419–1426.

NOBRE, João; NEVES, Rui Ferreira. Combining principal component analysis, discrete wavelet transform and XGBoost to trade in the financial markets. **Expert Systems with Applications**, Elsevier, v. 125, p. 181–194, 2019.

PASCANU, Razvan et al. How to construct deep recurrent neural networks. **arXiv preprint arXiv:1312.6026**, 2013.

PATEL, Manesh. **Trading with Ichimoku clouds: the essential guide to Ichimoku Kinko Hyo technical analysis**. [S.l.]: John Wiley & Sons, 2010. v. 473.

PEDREGOSA, Fabian et al. Scikit-learn: Machine learning in Python. **the Journal of machine Learning research**, JMLR. org, v. 12, p. 2825–2830, 2011.

PIMENTA, Alexandre et al. An automated investing method for stock market based on multiobjective genetic programming. **Computational Economics**, Springer, v. 52, n. 1, p. 125–144, 2018.

POWERS, David MW. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. **arXiv preprint arXiv:2010.16061**, 2020.

PRING, Martin J. **Study guide for technical analysis explained**. [S.l.]: McGraw-Hill Education, 2014.

QUONG, Gene Soudack; SOUDACK, Avrum. A.,. Volume-weighted rsi: Money flow. **Technical Analysis of Stocks & Commodities ()**, 1989.

RHEA, Robert. **Dow Theory Comment**. [S.l.: s.n.], 1932.

SAFAVIAN, S Rasoul; LANDGREBE, David. A survey of decision tree classifier methodology. **IEEE transactions on systems, man, and cybernetics**, IEEE, v. 21, n. 3, p. 660–674, 1991.

SAMARAWICKRAMA, AJP; FERNANDO, TGI. A recurrent neural network approach in predicting daily stock prices an application to the Sri Lankan stock market. In: IEEE. 2017 IEEE International Conference on Industrial and Information Systems (ICIIS). [S.l.: s.n.], 2017. P. 1–6.

SARDY, Sylvain; TSENG, Paul; BRUCE, Andrew. Robust wavelet denoising. **IEEE Transactions on Signal Processing**, IEEE, v. 49, n. 6, p. 1146–1152, 2001.

SEZER, Omer Berat; OZBAYOGLU, A Murat; DOGDU, Erdogan. An artificial neural network-based stock trading system using technical analysis and big data framework. In: PROCEEDINGS of the southeast conference. [S.l.: s.n.], 2017. P. 223–226.

SHEN, Guizhu et al. Deep learning with gated recurrent unit networks for financial sequence predictions. **Procedia computer science**, Elsevier, v. 131, p. 895–903, 2018.

STEWART, James; ROMO, Jorge Humberto. **cálculo**. [S.l.]: Pioneira Thomson Learning, 2006.

STOCCHI, Marco; MARCHESI, Michele. Fast wavelet transform assisted predictors of streaming time series. **Digital Signal Processing**, Elsevier, v. 77, p. 5–12, 2018.

SU, Huai et al. A hybrid hourly natural gas demand forecasting method based on the integration of wavelet transform and enhanced Deep-RNN model. **Energy**, Elsevier, v. 178, p. 585–597, 2019.

TANG, Zaiyong; FISHWICK, Paul A. Feedforward neural nets as models for time series forecasting. **ORSA journal on computing**, Informs, v. 5, n. 4, p. 374–385, 1993.

TEAM, The pandas development. **pandas-dev/pandas: Pandas**. [S.l.]: Zenodo, fev. 2020. DOI: 10.5281/zenodo.3509134. Disponível em: <<https://doi.org/10.5281/zenodo.3509134>>.

ULLAH, Shakir et al. Stock Price Forecast Using Recurrent Neural Network. **Urdu News Headline, Text Classification by Using Different Machine Learning Algorithms**, p. 47, 2019.

VALENS, Clemens. A really friendly guide to wavelets. ed. **Clemens Valens**, 1999.

WEIGEND, Andreas S; RUMELHART, David E; HUBERMAN, Bernardo A. Back-propagation, weight-elimination and time series prediction. In: **CONNECTIONIST models**. [S.l.]: Elsevier, 1991. P. 105–116.

WILDER, J Welles. **New concepts in technical trading systems**. [S.l.]: Trend Research, 1978.

WILLIAMS, Bill M. **New trading dimensions: how to profit from chaos in stocks, bonds, and commodities**. [S.l.]: John Wiley & Sons, 1998. v. 72.

WILLIAMS, Larry. The ultimate oscillator. **Technical Analysis of Stocks and Commodities**, v. 3, n. 4, p. 140–141, 1985.

WILLIAMS, Larry R. **How I Made One Million Dollars... Last Year... Trading Commodities**. [S.l.]: Windsor Books, 1979.

ZHANG, Kang et al. Stock market prediction based on generative adversarial network. **Procedia computer science**, Elsevier, v. 147, p. 400–406, 2019.

## 6.2 Apêndice I - Cálculo

Na seção seguinte serão apresentados conceitos base de cálculo para se compreender o gradiente utilizado mais a frente na decisão das ações do algoritmo. Os mesmos foram baseados no livro de (STEWART; ROMO, 2006).

### 6.2.1 Tangentes

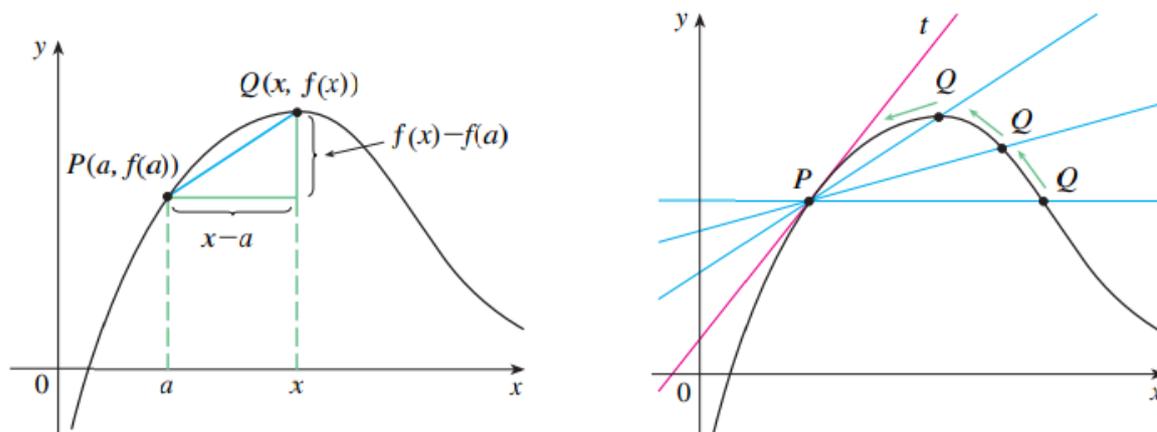
A reta tangente de uma curva pode ser interpretada como a taxa de variação de uma função num ponto específico arbitrário. Esta reta representa a reta de inclinação equivalente a curva no ponto escolhido para análise.

Para uma curva  $C$  que tem equação  $y = f(x)$  a reta tangente a  $C$  em um ponto  $P(a, f(a))$  pode ser encontrada ao aproximar-se um ponto  $Q(x, f(x))$ , onde  $x \neq a$  e então calculamos a reta secante  $PQ$  como:

$$m_{pq} = \frac{f(x) - f(a)}{x - a} \quad (98)$$

Desta forma fazemos  $Q$  se aproximar de  $P$ , obrigando  $x$  tender a  $a$ . Então se  $m_{pq}$  tender a um número  $m$ , podemos definir a tangente  $t$  como a reta que passa por  $P$  e tem inclinação  $m$ . Sendo então  $t$  a posição limite da reta secante  $PQ$  quando  $Q$  tende a  $P$  conforme a figura 39.

Figura 39 – Reta tangente PQ de Q tendendo a P.



Fonte: Stewart e Romo (2006).

Então a definição da reta tangente à curva  $y = f(x)$  em um ponto  $P(a, f(a))$  é a reta passando por  $P$  com inclinação seguindo a equação 99:

$$m = \lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a} \quad (99)$$

Uma outra equação, mais fácil de ser utilizada para expressar a reta tangente utiliza  $h = x - a$  e substitui  $x$  por  $x = a + h$ , resultando na inclinação da reta secante  $PQ$  sendo a equação 100:

$$m = \lim_{h \rightarrow 0} \frac{f(a + h) - f(a)}{h} \quad (100)$$

### 6.2.2 Derivada

A derivada de uma função pode ser definida como a taxa pontual de variação do  $y = f(x)$  num determinado ponto  $P$ .

A definição da derivada de uma função  $f$  em um número  $a$  é dada por  $f'(a)$  na equação 101 se o limite existir:

$$m = \lim_{h \rightarrow 0} \frac{f(a + h) - f(a)}{h} \quad (101)$$

Portanto sendo a derivada por definição a taxa de variação de  $y$  em função de  $x$ , se variarmos  $x$  de  $x_1$  a  $x_2$ , então a variação em  $x$ , que pode ser chamada de incremento de  $x$  será:

$$\Delta x = x_2 - x_1 \quad (102)$$

Desta forma a variação correspondente em  $y$  será de:

$$\Delta y = f(x_2) - f(x_1) \quad (103)$$

E o quociente das diferenças, denominado taxa média de variação de  $y$  em relação a  $x$  no intervalo  $[x_1, x_2]$  será:

$$\frac{\Delta y}{\Delta x} = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \quad (104)$$

Por fim definimos a taxa instantânea de variação de  $y$  em relação a  $x$  como:

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \lim_{x_2 \rightarrow x_1} \frac{f(x_2) - f(x_1)}{x_2 - x_1} \quad (105)$$

É importante citar que quando a função atinge seu mínimo ou máximo local sua derivada tende a zero. Este conceito será importante mais a frente para entendimento da decisão do algoritmo em operar compra ou venda.

### 6.3 Apêndice - Algoritmos de Aprendizado de Máquina

A seguinte seção se baseia no trabalho de (BURKOV, 2019) e no livro de (ALPAYDIN, 2020). Serão apresentados os conceitos básicos para entendimentos dos algoritmos base de aprendizado de máquina utilizados mais a frente neste trabalho para estudo de capacidade de aprendizado e comparação com o modelo proposto.

#### 6.3.1 Regressão Linear

A regressão linear se trata de um modelo de combinação de características lineares com o intuito de descrever o comportamento dos dados de entrada.

Para contextualizar, suponhamos que haja uma coleção de exemplos classificados  $(X_i, y_i)_{i=1}^N$ , onde  $N$  é o tamanho desta coleção,  $X_i$  é o vetor de  $D$  dimensões, como por exemplo  $i = 1, \dots, N$ ,  $y_i$  é o valor real alvo e todo valor de  $x_i$  também é um valor real.

Se desejamos construir um modelo  $f_{w,b}(X)$  como combinação linear dos dados em  $X$ , a seguinte equação 106 se aplica:

$$f_{w,b}(X) = wx + b \quad (106)$$

Onde:  $w$  = Vetor de parâmetros de  $D$  Dimensões

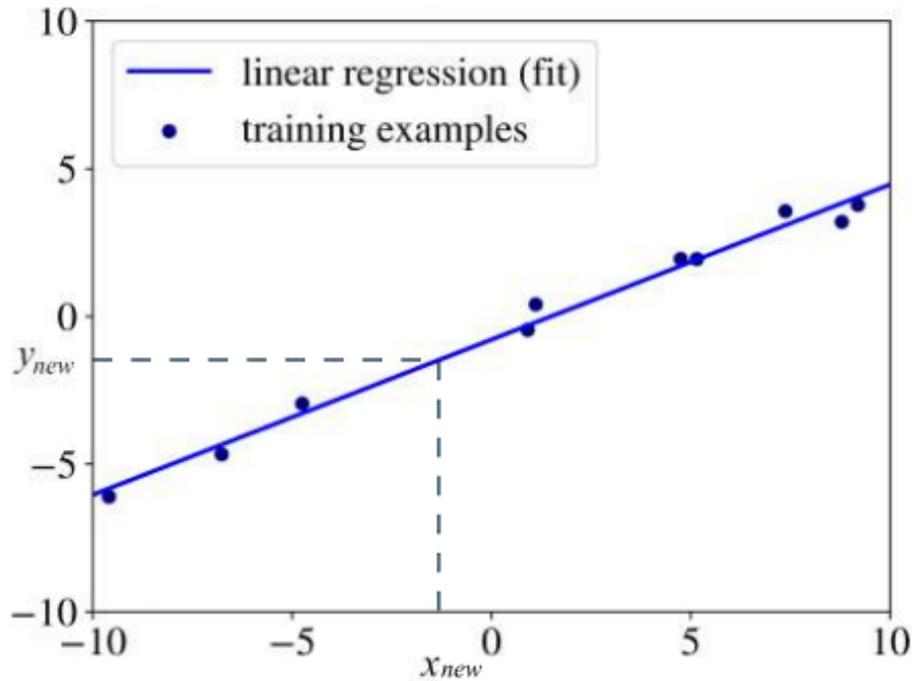
$b$  = Número Real

Para tal devemos encontrar os valores ótimos de  $w$  e  $b$ , estes que resultam na melhor previsão possível com o modelo de regressão linear.

Para exemplificar um caso de aplicação da regressão linear e seus pontos utilizados no treinamento é apresentada a figura 40. Esta mostra a regressão linear para dados de uma dimensão. Para casos onde a regressão linear é feita para duas dimensões ( $D = 2$  o modelo de regressão é um plano e não uma reta. E no caso de  $D > 2$  a regressão resulta num hiperplano).

Para se encontrar os valores ótimos de  $w^*$  e  $b^*$  nós tentamos minimizar a seguinte expressão:

Figura 40 – Exemplo de reta e dados de uma regressão linear de uma dimensão.



Fonte: Burkov (2019).

$$\frac{1}{N} \sum_{i=1 \dots N} f_b(X)(f_{w,b}(X_i) - y_i)^2 \quad (107)$$

Esta é a função objetivo, e a expressão  $(f_{w,b}(X_i) - y_i)^2$  é chamada de função erro, sendo esta a medida da penalidade pela classificação errônea do exemplo  $i$ . Esta função particular é conhecida como Perda de Erro Quadrático (*Squared Error Loss*). Todos os modelos de aprendizado tem uma função de perda e o modelo busca minimizar o objetivo conhecido como função custo. Em uma regressão linear, a função custo é dada pela perda média, também chamada de risco empírico. A perda média, ou risco empírico, de um modelo é dada por todas as penalidades obtidas aplicando o modelo aos dados de treinamento.

Agora utilizando a notação do livro de (ALPAYDIN, 2020), chamaremos os coeficientes  $w$  de vetor  $W$  que aproxima para diversas dimensões a regressão linear ao menor erro.

Para um modelo de regressão linear com uma dimensão então temos a equação 108:

$$g(x^t | w_1, w_0) = w_1 x^t + w_0 \quad (108)$$

Então utilizando a derivada da equação de erro  $(f_{w,b}(X_i) - y_i)^2$ , chegamos a duas equações com dois fatores desconhecidos  $w_1$  e  $w_0$ :

$$\sum_t r^t = Nw_0 + w_1 \sum_t x^t \quad (109)$$

$$\sum_t r^t x^t = w_0 \sum_t x^t + w_1 \sum_t (x^t)^2 \quad (110)$$

Estes podem ser escritos em forma de matrizes da seguinte forma:

$$A = \begin{bmatrix} N & \sum_t x^t \\ \sum_t x^t & \sum_t (x^t)^2 \end{bmatrix}, w = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, y = \begin{bmatrix} \sum_t r^t \\ \sum_t r^t x^t \end{bmatrix} \quad (111)$$

que pode ser resolvida pela equação 112:

$$w = A^{-1}y \quad (112)$$

### 6.3.2 Regressão Polinomial

A regressão polinomial semelhante a linear aproxima uma linha, plano ou hiperplano aos valores de treinamento através da minimização da função de custo. Como generalização de seu modelo em  $x$  de ordem  $k$  temos:

$$g(x^t | w_k, \dots, w_1, w_0) = w_k (x^t)^k + \dots + w_1 x^t + w_0 \quad (113)$$

O modelo ainda é linear em relação aos seus parâmetros e tomando as derivadas, chegamos a  $k + 1$  equações com  $k + 1$  parâmetros desconhecido que podem ser escritos de forma generalizada como;

$$A = \begin{bmatrix} N & \sum_t x^t & \dots & \sum_t (x^t)^k \\ \sum_t x^t & \sum_t (x^t)^2 & \dots & \sum_t (x^t)^{k+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_t (x^t)^k & \sum_t (x^t)^{k+1} & \dots & \sum_t (x^t)^{2k} \end{bmatrix}, w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_k \end{bmatrix}, y = \begin{bmatrix} \sum_t r^t \\ \sum_t r^t x^t \\ \vdots \\ \sum_t r^t (x^t)^k \end{bmatrix} \quad (114)$$

Podemos então escrever  $A = D^T D$  e  $y = D^T r$  onde:

$$D = \begin{bmatrix} 1 & x^1 & \dots & \sum_t (x^1)^k \\ 1 & x^2 & \dots & \sum_t (x^2)^k \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x^N & \dots & \sum_t (x^N)^k \end{bmatrix}, r = \begin{bmatrix} r^1 \\ r^2 \\ \vdots \\ r^k \end{bmatrix} \quad (115)$$

E podemos solucionar para os parâmetros  $w$  pela equação 116:

$$w = (D^T D)^{-1} D^T r \quad (116)$$

Na subseção seguinte veremos a teoria de modelos de aprendizado por conjunto, que utilizam modelos mais simples para construir um mais complexo como combinação deles.

### 6.3.3 Aprendizado por Conjunto (*Ensemble*)

O aprendizado por conjunto é um paradigma de aprendizado que ao invés de tentar desenvolver um modelo de predição com alta precisão e acurácia, foca em desenvolver um grande número de preditores "ruins" para que a combinação ponderada destes preditores resulte em um meta-modelo com alta acurácia.

Os preditores fracos, como são chamados os submodelos destes conjuntos, são algoritmos de aprendizado que não conseguem aprender distribuições complexas e tem seu treinamento tipicamente rápido comparado com modelos independentes mais complexos. O preditor fraco mais comumente utilizado é árvores de decisão rasas, que geralmente param seu aprendizado apenas algumas interações após seu início. Estas não são particularmente precisas, porém a ideia do aprendizado de conjunto reside na suposição de que se os preditores fracos não forem idênticos e tiverem uma predição pelo menos um pouco melhor que resultados aleatórios, podemos conseguir um modelo de grande precisão combinando tais preditores.

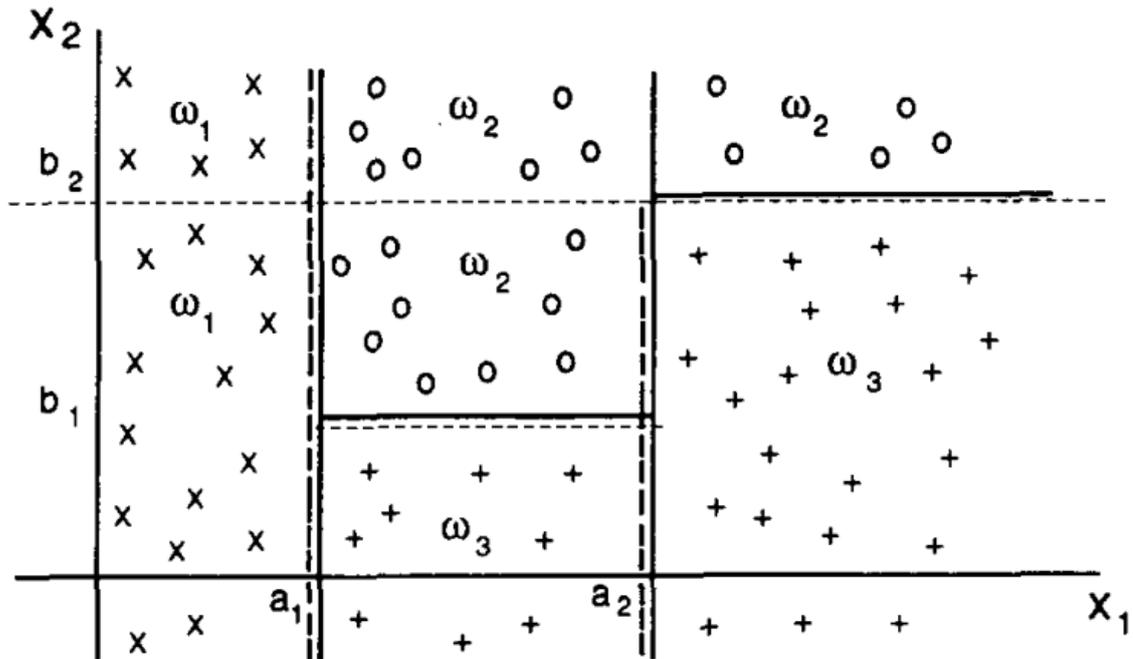
Portanto o resultado do modelo consiste em combinar e extrair a média do resultado de tais preditores para uma entrada  $x$  e assim avaliar sua saída  $y$  como uma ponderação de todas as predições.

### 6.3.4 Floresta Aleatória (*Random Forest*)

A floresta aleatória se baseia, como dito anteriormente, em utilizar um número grande de árvores de decisão/regressão fracos para combinar seus resultados de predição e então extrair sua classificação/regressão. A *Random Forest* especificamente é baseada no *ensemble* do tipo encasacamento (*bagging*) que consiste em criar várias cópias dos dados de treinamento (cada cópia sendo levemente diferente uma da outra para reforçar a diversidade necessária nos preditores fracos) e treinando cada preditor fraco para aprender com uma e finalmente combinando-os.

As árvores utilizadas são ilustradas na figura 41 e 42 apresentadas originalmente por (SAFAVIAN; LANDGREBE, 1991). O espaço exemplifica a divisão de um espaço bidimensional em "folhas" que representam as fronteiras de decisão entre os grupos a serem classificados e sua árvore resultante é apresentada na figura 42.

Figura 41 – Espaço de decisão de uma árvore de decisão em duas dimensões.



Fonte: Safavian e Landgrebe (1991).

O algoritmo de *bagging* clássico funciona criando um  $B$  número de amostras dos dados de treinamento  $S_b$  (para cada  $b = 1, \dots, B$ ) e um modelo de árvore de decisão  $f_b$  utilizando cada amostra  $S_b$  como dado de treinamento. Para se selecionar as amostras  $S_b$ , se inicia com um conjunto vazio e se seleciona cópias parciais dos dados de treinamento aleatoriamente agrupando-as.

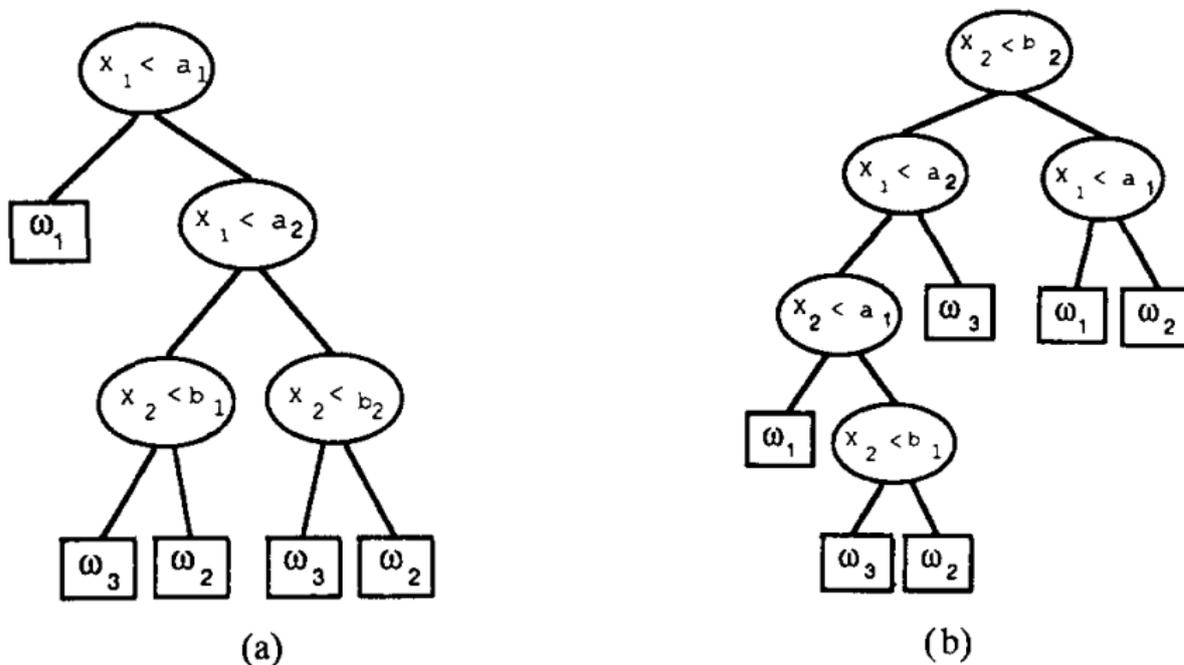
Depois de se treinar  $B$  árvores de decisão, a predição é resultado da média de  $B$  previsões seguindo a equação 117:

$$y \leftarrow \hat{f}(X) = \frac{1}{B} \sum_{b=1}^B f_b(X) \quad (117)$$

O resultado da predição é a média dos valores preditos por cada preditor fraco, e no caso da classificação por voto majoritário.

O algoritmo de floresta aleatória é diferente do encasacamento clássico apenas no uso de árvore modificadas para inspecionar em cada nova subdivisão um subconjuntos aleatório de

Figura 42 – Árvore de decisão resultante da separação em folhas do espaço de estado de duas dimensões .



Sendo (a) a decisão para as fronteiras de linha contínua e (b) a decisão para as fronteiras pontilhadas.

Fonte: Safavian e Landgrebe (1991).

características. A razão para isto é para se evitar correlação entre as árvores, o que influenciaria negativamente no resultado da predição. O motivo do da alta acurácia do meta-modelo é que bons modelos fracos provavelmente concordarão em uma mesma predição e modelos ruins discordarão em predições diferentes. A correlação pode causar maus modelos a concordarem o que levaria a influenciar o voto majoritário de forma equivocada.

O maior motivo para a grande popularidade do algoritmo de floresta aleatória é que devido a sua capacidade de decisão conjunta a variância do conjunto final diminui, o que significa uma possibilidade muito menor de sobre-ajuste (*overfitting*) do modelo.