

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

NEWTON PAULO LINCEN

**Proposta e Avaliação de Features para
Negociação Algorítmica em Mercado
Financeiro**

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciência da
Computação

Orientador: Profa. Dra. Renata Galante

Porto Alegre
2023

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Linchen, Newton Paulo

Proposta e Avaliação de Features para Negociação Algorítmica em Mercado Financeiro / Newton Paulo Linchen. – Porto Alegre: PPGC da UFRGS, 2023.

94 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2023. Orientador: Renata Galante.

1. Aprendizado de máquina. 2. Algotrading. 3. Mercado de Ações e Futuros. 4. Quadratic Discriminant Analysis. 5. Engenharia de Features. 6. Classificação. 7. Previsão. 8. Séries Temporais. I. Galante, Renata. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões

Vice-Reitora: Prof^a. Patricia Pranke

Pró-Reitor de Pós-Graduação: Prof. Celso Giannetti Loureiro Chaves

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do PPGC: Prof. Claudio Rosito Jung

Bibliotecária-chefe do Instituto de Informática: Alexsander Borges Ribeiro

AGRADECIMENTOS

Agradeço a todos os que contribuíram para que esse trabalho fosse possível. Ao meu filho, Bernardo Bonamigo Linchen, razão de todos os meus esforços. A Lavoisier Paulo Linchen, in memoriam. A Maria Luiza Linchen, minha mãe. À Professora Doutora Renata Galante, por sua orientação e estímulo. Ao Professor Doutor Adriano Pereira, por sua orientação, estímulo e por compartilhar do interesse pelo mercado financeiro e algo trading. Aos meus amigos Eduardo Pouzada da Rosa e Cristiano Benjamin, por seu apoio. Ao meu Maguid Charles Kiefer, pelo Avodat Hashem e por ter me auxiliado a receber a Shefa.

RESUMO

Séries temporais de ativos financeiros de negociação são conhecidas por ter propriedades estocásticas que tornam a previsão uma tentativa quase fútil. Na Economia, a teoria dominante da Hipótese do Mercado Eficiente propõe que qualquer tentativa de prever os preços futuros de um ativo negociável é em vão e não deve ser perseguida. Nas últimas décadas, o surgimento de algoritmos de aprendizado de máquina deu à comunidade de investimentos ferramentas interessantes para avançar na pesquisa de previsão. No entanto, entendemos que algoritmos não são suficientes para fazer previsões bem-sucedidas: para construir modelos melhores, o pesquisador deve empregar o desenvolvimento de recursos, especialmente com o conhecimento e experiência de um praticante e especialista no campo. Neste trabalho, avaliamos o desempenho de um algoritmo de classificação (QDA - Análise Discriminante Quadrática) com a adição de recursos, comparando os resultados com um benchmark (compra e retenção) e um experimento de linha de base. A investigação foi realizada usando o Contrato Futuro do Índice Bovespa (Ibovespa Futuro), fazendo previsões de curto prazo em um ambiente simulado.

Palavras-chave: Aprendizado de máquina. Algotrading. Mercado de Ações e Futuros. Quadratic Discriminant Analysis. Engenharia de Features. Classificação. Previsão. Séries Temporais.

ABSTRACT

Time series of financial trading assets are known to have stochastic properties which turn prediction into an almost futile endeavor. In Economics, the mainstream theory of the Efficient Market Hypothesis proposes that any attempt to predict the future prices of a tradable asset is in vain, and should not be pursued. In the last decades, the advent of machine learning algorithms gave the investing community interesting tools for advancing the prediction research. However, we understand that algorithms are not enough to make successful predictions: in order to build better models the researcher should employ feature development, especially with the knowledge and experience of a practitioner and specialist in the field. In this work, we evaluate the performance of a classification algorithm (QDA - Quadratic Discriminant Analysis) with the addition of features, comparing the results with a benchmark (buy-and-hold) and a baseline experiment. The investigation was carried out using the Bovespa Index Futures Contract (Ibovespa Futuro), by making short-term predictions in a simulated environment.

Keywords: Feature Engineering, Machine Learning, Algotrading, Classification, Prediction, Stock Markets, Time Series, Quadratic Discriminant Analysis.

LISTA DE ABREVIATURAS E SIGLAS

QDA	- Quadratic Discriminant Analysis
EMH	- Efficient Market Hypothesis
HME	- Hipótese de Mercado Eficiente
OHLC	- Open, High, Low, Close
MACD	- Moving Average Convergence Divergence
MME	- Média Móvel Exponencial
IFR	- Índice de Força Relativa
RSI	- Relative Strength Index
W%R	- William's Percent Range
WILLR	- William's Percent Range
ROC	- Rate Of Change
OBV	- On Balance Volume
CCI	- Commodity Channel Index
PT	- Preço Típico
MM	- Média Móvel
DI	- Directional Index
TR	- True Range
ATR	- Average True Range
DM	- Directional Movement
MMA	- Média Móvel
TRIX	- Triple Exponential Average
EMA	- Exponential Moving Average
A/D	- Advance/Decline
LDA	- Linha de Avanço e Declínio

EBIDTA- Earnings Before Interest, Taxes, Depreciation and Amortization

SALIB - Structural Analysis Library

CTO - Close-to-open

IBS - Internal Bar Strength

FHIGH - From High

FH - From High

FLOW - From Low

FL - From Low

ADX - Average Directional Index

MFI - Market Facilitation Index

STOCH - Estocástico

LISTA DE FIGURAS

Figura 3.1	Representatividade de Features nos Trabalhos por Ano.....	39
Figura 4.1	Etapas da Metodologia	42
Figura 4.2	46
Figura 4.3	48
Figura 5.1	51
Figura 5.2	52
Figura 5.3	52
Figura 5.4	53
Figura 5.5	Gráfico representativo do ativo utilizado no <i>dataset</i>	55
Figura 5.6	67
Figura 5.7	Gráfico representativo do <i>Benchmark</i> Dados de preço do início do período do <i>Benchmark</i>	68
Figura 5.8	Dados de preço do fim do período do <i>Benchmark</i>	68
Figura 5.9	70
Figura 5.10	74
Figura 5.11	74
Figura 5.12	78
Figura 5.13	79

LISTA DE TABELAS

Tabela 3.1	Ocorrência de features na literatura.....	34
Tabela 3.2	Número de ocorrências de Indicadores Técnicos na literatura.....	35
Tabela 3.3	Número de ocorrências de features "Dados de Mercado"na literatura.....	36
Tabela 3.4	Número de ocorrências das features "Transformações Estatísticas"na literatura	36
Tabela 3.5	Distribuição de tipos de features por tipos de algoritmo na literatura.....	38
Tabela 5.1	Retorno do ativo por janelas de tempo.	53
Tabela 5.2	Instanciações das <i>Features SALIB</i>	64
Tabela 5.3	Resultados das <i>Features SALIB</i> (por número de <i>features</i> nas investigações).72	
Tabela 5.4	<i>Feature Importance SALIB</i>	75
Tabela 5.5	Melhores <i>Features SALIB</i> . Critério 1.	76
Tabela 5.6	Melhores <i>Features SALIB</i> . Critério 2.	76
Tabela 5.7	Resultados das <i>Features TALIB</i> (por número de <i>features</i> nas investigações).77	
Tabela 5.8	<i>Feature Importance TALIB</i>	79
Tabela 5.9	Melhores <i>Features TALIB</i> . Critério 1.	80
Tabela 5.10	Melhores <i>Features TALIB</i> . Critério 2.	81
Tabela 5.11	Melhores <i>Features SALIB + TALIB</i> . Critério 1.	82
Tabela 5.12	Melhores <i>Features SALIB + TALIB</i> . Critério 2.	83
Tabela 5.13	<i>Features SALIB</i> mais relevantes.	84
Tabela 5.14	<i>Features TALIB</i> mais relevantes.	85
Tabela 5.15	Melhores resultados.....	86

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Objetivo.....	14
1.2 Organização do trabalho	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 Definição de <i>Feature</i>	16
2.2 Engenharia de <i>Features</i>	17
2.3 Desenvolvimento ou descoberta de <i>features</i>	18
2.4 Desenvolvimento de <i>features</i> por especialista na área	18
2.5 <i>Features</i> encontradas na revisão da literatura	19
2.6 <i>Features</i> utilizadas nos experimentos realizados.....	29
2.6.1 <i>Features</i> SALIB	29
2.6.2 <i>Features</i> TALIB	30
2.7 Considerações finais.....	31
3 TRABALHOS RELACIONADOS	32
3.1 Discussão dos Trabalhos Relacionados	32
3.2 Análise Comparativa	34
3.3 Considerações do Capítulo.....	38
4 METODOLOGIA	42
4.1 Visão Geral	42
4.2 Descrição da Metodologia	43
4.2.1 Definição do <i>dataset</i>	43
4.2.2 Definição do algoritmo	44
4.2.3 Definição das <i>features</i>	45
4.3 Realização de experimentos	45
4.3.1 Avaliação dos resultados	48
5 EXPERIMENTOS	50
5.1 Definição do <i>dataset</i>	50
5.2 Definição do algoritmo.....	55
5.3 Definição das <i>features</i>	56
5.4 Experimentos	59
5.4.1 Descrição do Primeiro Experimento - <i>Benchmark</i> e <i>Baseline</i>	59
5.4.1.1 Descrição do <i>Benchmark</i>	60
5.4.1.2 Descrição do <i>Baseline</i>	61
5.4.2 Descrição do Segundo Experimento - <i>Features</i> SALIB	61
5.4.3 Descrição do Terceiro Experimento - <i>Features</i> TALIB	65
5.4.4 Descrição do Quarto Experimento - <i>Features</i> SALIB e <i>Features</i> TALIB	66
5.5 Resultados dos Experimentos e sua Avaliação	66
5.5.1 Resultados do Primeiro experimento - <i>Benchmark</i> e <i>Baseline</i>	66
5.5.1.1 Resultados do <i>Benchmark</i>	66
5.5.1.2 Resultados do <i>Baseline</i>	68
5.5.2 Resultados do Segundo Experimento - <i>Features</i> SALIB	70
5.5.3 Resultados do Terceiro Experimento - <i>Features</i> TALIB	77
5.5.4 Resultados do Quarto Experimento - <i>Features</i> SALIB e <i>Features</i> TALIB.....	81
5.5.5 Análise das <i>Features</i> mais relevantes.....	83
5.5.6 Melhores resultados	85
5.6 Considerações finais.....	86
6 CONCLUSÃO	88
REFERÊNCIAS	92

1 INTRODUÇÃO

Análise e previsão de séries temporais financeiras é tradicionalmente um objeto de estudo da Economia, mais notadamente da disciplina de Econometria. Sejam dados de preços de ações no mercado de capitais, dados de consumo, de inflação, de variáveis macroeconômicas diversas, o estudo destas séries desenvolveu-se principalmente com base nos modelos econométricos, cuja função principal é determinar a presença de relações de persistência estatística entre dados da série, (ou séries, no caso de análise multivariada). Para alcançar esse objetivo, os dados devem se conformar às premissas dos modelos econométricos, tendo sua tendência removida e sua memória alterada, reduzida.

Esse tratamento dos dados, para se ajustarem às restritas condições dos modelos, remove uma quantidade significativa de informações das séries temporais, informações essas que poderiam resultar em uma maior acurácia nas previsões.

A partir do momento que os Cientistas de Dados (oriundos principalmente da disciplina da Ciência da Computação) começaram a abordar o problema da análise e previsão de séries temporais financeiras, trouxeram, como era de se esperar, sua própria abordagem e metodologia de solução de problemas para essa área de estudo. Não tendo necessidade, a priori de propor qualquer redução na série temporal estudada, utilizaram sua “caixa de ferramentas” da mesma forma que tratavam qualquer outro problema computacional: experimentando, desconstruindo o problema em partes menores e utilizando algoritmos computacionais cuja tarefa, ao invés de encontrar relações estatísticas persistentes entre os dados, era apenas otimizar a qualidade da previsão gerada.

Em outras palavras, aproximaram-se do problema por um outro ângulo, com uma abordagem própria e um *modus operandi* totalmente diverso dos economistas.

Ainda hoje, a reação dos economistas não é particularmente favorável: como disciplina, a Economia propõe postulados teóricos apriorísticos, que, aceitos como premissas, guiam toda a sua atividade de pesquisa. Contestar tais premissas por meio de experimentos com os dados é visto como uma perda de tempo, senão uma flagrante ofensa ao que já foi validado como conhecimento econômico.

Uma dessas premissas está contida na Hipótese de Mercado Eficiente, proposta por French e Fama na década de 60. (Ao se referir à possibilidade de previsão do movimento dos preços de ações (que são séries temporais financeiras), a HME propõe que a tarefa é infrutífera e fadada ao fracasso, uma vez que os preços seguem um passeio aleatório (*random walk*) determinado por processos estocásticos (movimento browniano), o

que torna impossível a tarefa de previsão. (Para um histórico da Hipótese de Mercado Eficiente, vide (SEWELL, 2011)).

Quando do surgimento da HME, já haviam duas escolas de pensamento e análise no mercado financeiro: a escola conhecida como Análise Fundamentalista e a escola conhecida como Análise Técnica. Realmente, em alguns aspectos elas antecederam o surgimento da HME, pois temos as suas obras germinais escritas entre as décadas de 30 e 50 (enquanto que a HME foi formalmente proposta na década de 60).

A Análise Fundamentalista propõe que o movimento dos preços das ações na Bolsa de Valores tende a corresponder, no médio e longo prazos, ao valor de mercado atribuído às empresas. Ou seja, empresas fortes, geradoras de lucro consistente, tenderiam a ser precificadas favoravelmente pelo mercado. No caso em que isso não ocorresse, e uma empresa de valor estivesse sendo negociada abaixo de seu real valor, isso significaria uma excelente oportunidade de compra, pois em algum momento o mercado corrigiria essa ineficiência no preço das ações, valorizando as ações dessa empresa com um movimento de alta. Em 1934, Benjamin Graham e David Dodd publicaram o que seria considerado um marco na Análise Fundamentalista, sua obra germinal, “Security Analysis” (GRAHAM; DODD; DODD, 1934). Em sequência, outras obras sedimentaram esta linha de análise: “The Theory Of Investment Value”, em 1938, por John Burr, (WILLIAMS, 1938), e “The Intelligent Investor”, também de Benjamin Graham, em 1949 (GRAHAM, 1949).

Em paralelo, a chama escola da Análise Técnica tinha também suas principais obras vindo a público, também ao redor das décadas de 30 a 50 (anteriormente à HME, portanto). “The Dow Theory” foi publicado em 1932 por Robert Rhea (RHEA, 1932), enquanto que a obra magna da Análise Técnica, “Technical Analysis Of Stock Trends”, foi publicada por John Magee em 1948 (EDWARDS; MAGEE, 1948). A escola de Análise Técnica propõe que existem padrões identificáveis na série temporal de preços, especialmente quando plotada em gráficos de preço x tempo, motivo pelo qual os praticantes dessa forma de análise ficaram conhecidos como “grafistas”. Sua análise é eminentemente “visual”, e pressupõe que existam “tendências” no movimento dos preços das ações e ativos financeiros.

Os Cientistas de Dados, por outro lado, não cultivam a priori nenhum preconceito relativo à natureza das séries temporais financeiras. Atentos ao fato de que nem toda informação útil (no sentido de relevante à tarefa preditiva) é visível, claramente identificável (em outras palavras, “linear”), abordam o problema de previsão da mesma forma que to-

dos os problemas na Ciência da Computação: com criatividade, recursos computacionais (o estado da arte dos diversos algoritmos desenvolvidos pela área) e, muitas vezes, uma sequência de passos que une abordagens diferentes para resolver um problema único. Em outras palavras, usam modelagem algorítmica.

Ao desenvolver modelos, os cientistas de dados utilizam diversos algoritmos disponíveis na Ciência da Computação, muitas vezes criando protocolos próprios de pesquisa. A Ciência de Dados teve, desde sua origem, (e acreditamos que ainda mantém esse ethos) um caráter de experimentação. Desse modo, experimentar diferentes procedimentos computacionais em uma série de dados, simplesmente para observar os resultados obtidos, ou seja, sem se preocupar com um conjunto de premissas determinadas a priori, para o Cientista de Dados não é uma “heresia”, ou perda de tempo. Ao contrário, é a essência mesma do espírito investigativo que leva a descobertas de caráter interdisciplinar e inovador.

Em resumo, para o Cientista de Dados as séries temporais financeiras não possuem características definidas de antemão, muito ao contrário: são um conjunto de dados cujo valor será extraído por investigação profunda de caráter experimental.

Essa não era a visão da Economia. Afinal, porque buscar prever preços de ações e outras séries temporais financeiras, se tal exercício é fútil? Afinal, como instituiu a HME, os homens são seres racionais com completo acesso à informação e cujo objetivo é maximizar sua função de valor.

Rory Sutherland, vice-presidente da Ogilvy U.K. (uma das maiores agências de publicidade do mundo), declara que este “homo economicus” proposto pela Economia, este ser racional, objetivo, com completa confiança e acesso à informação é encontrado na fronteira entre “quase nunca e lugar nenhum”. Ou seja, a atividade humana (e a atividade econômica é a epítome da atividade humana) é permeada por irracionalidade, decisões emocionais, impulsos e muito pouca objetividade.

Desse modo, o trabalho do Cientista de Dados ao tentar prever séries temporais financeiras é colher do chão o que foi apressadamente descartado pelo economista como sendo “inútil”. E tratar cada bit de dados como igualmente importante.

O mercado de capitais é um ambiente complexo e dinâmico, onde os investidores buscam maximizar seus lucros e minimizar seus riscos. A negociação de ativos financeiros exige uma análise cuidadosa dos dados e informações disponíveis, a fim de tomar decisões informadas e estratégicas. Com o avanço da tecnologia, o uso de técnicas de *machine learning* tem se mostrado cada vez mais eficaz na análise e previsão do com-

portamento do mercado de capitais. No entanto, para que esses modelos sejam realmente úteis, é preciso considerar a seleção adequada de *features* para o treinamento.

1.1 Objetivo

Neste trabalho, apresentamos a importância do uso de *features* no treinamento de modelos de machine *learning* para negociação no mercado de capitais. Demonstramos como a escolha correta das *features* pode influenciar significativamente a precisão e a eficácia do modelo, permitindo que o investidor obtenha melhores resultados. As *features* são variáveis que fornecem informações relevantes para o modelo, permitindo que ele faça previsões mais precisas e consistentes. Ao utilizar *features* adequadas, é possível capturar padrões e tendências que não seriam visíveis em outras circunstâncias. Com isso, é possível tomar decisões mais informadas e estratégicas, reduzindo os riscos e aumentando os lucros.

Estudaremos neste trabalho como o uso de *features* pode melhorar significativamente os resultados de previsões dos modelos algorítmicos da Ciência de Dados para o mercado financeiro. Este, portanto, é um trabalho dedicado ao estudo do impacto das *features* na qualidade dos resultados obtidos por algoritmos de machine *learning* no contexto de séries temporais financeiras. Partimos do pressuposto que a presença de um especialista da área, capaz de elaborar e propor *features*, enriquece o trabalho de pesquisa e permite alcançar resultados superiores aos que seriam obtidos apenas com o uso dos algoritmos.

O trabalho deverá ser realizado de modo obtermos resultados via Experimentos (vide Capítulo 5). Os Experimentos serão propostos de modo a tentar responder as seguintes questões de investigação:

“Algoritmos retornam resultados superiores quando treinados com *features* criadas por um especialista?”, “À medida que *features* de um mesmo tipo são inseridas, qual o impacto nos resultados?”, “O que ocorre com os resultados quando o número de *features* aumenta?”, “O que ocorre com os resultados quando misturamos dois tipos de *features* de origens diferentes?”

1.2 Organização do trabalho

Neste trabalho, a presente introdução (capítulo 1), inicia a discussão sobre a validade do uso das *features* criadas por um especialista na área, no contexto de pesquisa utilizando algoritmos de *machine learning* em modelagem de séries temporais financeiras (mercado de capitais). Em seguida, durante o Capítulo 2, o trabalho descreve o uso de *features* na literatura e aborda a construção de diversos tipos de *features*, desde as utilizadas amplamente em bibliotecas python, como *talib.py*, como as criadas pelos próprios autores dos trabalhos. Descreve, também, as *features* utilizadas nessa pesquisa, segundo a metodologia proposta no Capítulo 4 - Metodologia e instanciada no Capítulo 5 - Experimentos.

A análise da literatura pertinente fica no Capítulo 3, onde são analisados diversos *papers*, dissertações, artigos e trabalhos relacionados à pesquisa de *algotrading* utilizando tanto algoritmos de *machine learning* como de *deep learning*. A ênfase da análise foi na extração das *features* utilizadas nos trabalhos, a classificação das mesmas em diferentes grupos e a correspondência entre complexidade dos algoritmos utilizados e o esforço no desenvolvimento das *features*. O próximo passo, substanciado no Capítulo 4, é a proposição da metodologia que será utilizada na pesquisa. Esta metodologia habilita qualquer pesquisador do assunto a reproduzir os experimentos descritos no Capítulo 5 - Experimentos, além de fornecer o passo-a-passo para customização e ampliação da pesquisa, de acordo com sua preferência.

Os experimentos, propriamente ditos, são abordados no Capítulo 5, que é destinado a descrever os diversos experimentos realizados para avaliar o impacto das *features* no resultado da modelagem de algoritmos de *machine learning* para *algotrading*. Traz a sequência de experimentos e analisa os resultados obtidos. Finalmente, apresentamos uma conclusão para o trabalho no Capítulo 6, onde oferecemos um resumo do que foi encontrado na pesquisa, e as conclusões teóricas e práticas que derivaram dos experimentos, encerrando este trabalho e trazendo novas linhas de pesquisa para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo apresentamos a fundamentação teórica que sustenta o trabalho, com a conceituação e descrição das *features*, e ainda o espaço que elas ocupam no esforço de pesquisa com modelagem algorítmica. Na Seção 2.1 vemos a definição geral do termo *feature*, e sua conceituação, na Seção 2.2 observamos o conceito de *feature engineering*, e como ele extrai *features* de modo automático (algorítmico) do *dataset*. Em seguida, na Seção 2.3 entendemos a *feature discovery* ou a descoberta de *features* e seu papel na modelagem algorítmica e na Seção 2.4 analisamos o papel de um especialista no domínio para o desenvolvimento de *features*. Na Seção 2.5 listamos as *features* encontradas na revisão da literatura (Capítulo 3). Na Seção 2.6 listamos as *features* utilizadas nos experimentos deste trabalho (Capítulo 5). Por fim, na Seção 2.7 tecemos as considerações finais do capítulo.

2.1 Definição de *Feature*

Em aprendizado de máquina, uma *feature*, também conhecida como característica, é uma medida quantitativa que descreve uma determinada propriedade ou aspecto de um objeto ou evento. As *features* são usadas como entradas para um modelo de *machine learning*, onde o modelo tenta aprender a relação entre as *features* e uma variável de interesse, também conhecida como variável dependente.

As *features* são extremamente importantes no contexto da modelagem de algoritmos de *machine learning* porque determinam em grande parte a qualidade e a precisão do modelo. Se as *features* selecionadas não forem relevantes para a variável dependente, o modelo pode ter dificuldade em aprender uma relação útil. Por outro lado, se as *features* forem altamente relevantes, o modelo pode ser capaz de fazer previsões altamente precisas.

A seleção de *features* é uma das etapas mais críticas na construção de modelos de *machine learning* e pode envolver uma série de técnicas, incluindo a análise exploratória de dados, a seleção manual de *features* com base no conhecimento do domínio, ou a aplicação de algoritmos de seleção de *features* automatizados.

A engenharia e descoberta de *features* são elementos fundamentais no treinamento de modelos de aprendizado de máquina financeiro. Marcos López de Prado, um dos principais especialistas em finanças quantitativas e investimentos algorítmicos, destaca a im-

portância desses elementos em seu livro (PRADO, 2018a). O papel da engenharia de *features* e da descoberta de *features* no treinamento de modelos financeiros foi destacado por Lopez de Prado em vários de seus trabalhos. Em (PRADO, 2018b) Lopez de Prado argumenta que a engenharia de *features* é um dos elementos mais importantes do processo de modelagem financeira. Ele afirma que “um modelo pode ser tão bom quanto suas *features*”, e que a seleção e criação cuidadosa de *features* é fundamental para garantir a precisão e robustez do modelo. Prado destaca que a engenharia e descoberta de *features* são especialmente importantes no mercado financeiro, devido à complexidade dos dados financeiros. Os dados financeiros podem ser altamente voláteis e altamente não-lineares, o que pode tornar difícil encontrar padrões significativos. Além disso, o mercado financeiro pode ser altamente sensível a fatores externos, como mudanças nas políticas governamentais ou eventos geopolíticos, o que pode tornar difícil prever com precisão o comportamento futuro do mercado.

2.2 Engenharia de *Features*

A engenharia de *features* (ou *feature engineering*, em Inglês) é um processo importante no campo da ciência de dados que envolve a criação e seleção de variáveis ou *features* relevantes para um modelo de aprendizado de máquina. O objetivo é maximizar o desempenho do modelo, melhorando sua capacidade de fazer previsões precisas.

O processo de engenharia de *features* pode envolver várias etapas, como a identificação de variáveis relevantes para a tarefa de modelagem, a limpeza e transformação dos dados, a criação de novas variáveis a partir das existentes, a normalização e padronização dos dados, e a seleção de um conjunto final de *features* para usar no modelo.

A engenharia de *features* é uma tarefa desafiadora que requer conhecimento de domínio e habilidades em matemática, estatística e programação. No entanto, um bom trabalho de engenharia de *features* pode fazer a diferença entre um modelo de aprendizado de máquina com baixa precisão e um modelo com alta precisão e generalização.

Prado também enfatiza a importância da normalização dos dados na engenharia de *features*. A normalização envolve a transformação dos dados em uma escala comum, a fim de permitir a comparação entre diferentes variáveis. Isso é especialmente importante quando se trabalha com conjuntos de dados com diferentes unidades de medida ou escalas diferentes. Em resumo, a engenharia de *features* é uma parte crucial do processo de modelagem de aprendizado de máquina que envolve a seleção e criação de variáveis

relevantes para melhorar o desempenho do modelo.

2.3 Desenvolvimento ou descoberta de *features*

A descoberta de *features* (ou *feature discovery*, em Inglês) é o processo de identificar novas variáveis ou *features* que podem ser relevantes para um modelo de aprendizado de máquina. Ao contrário da engenharia de *features*, que envolve a criação de novas variáveis a partir de dados existentes, a descoberta de *features* é um processo exploratório que busca encontrar variáveis ou *features* ocultas nos dados que podem ser úteis para a tarefa de modelagem.

A descoberta de *features* pode envolver a aplicação de técnicas estatísticas e matemáticas avançadas, como análise de componentes principais, análise de cluster, análise discriminante, análise de variância, entre outras. O objetivo é identificar padrões e estruturas ocultas nos dados que podem ser relevantes para o modelo.

Esse processo pode ser desafiador, pois pode ser necessário explorar grandes quantidades de dados e fazer escolhas cuidadosas sobre quais variáveis são relevantes e quais não são. No entanto, uma boa descoberta de *features* pode levar a novos *insights* sobre os dados e ajudar a criar modelos mais precisos e eficazes.

Outra contribuição importante de Prado é a ideia de que a descoberta de *features* é fundamentalmente uma tarefa de redução de dimensionalidade. A redução de dimensionalidade envolve a identificação de um subconjunto dos *features* originais que contêm a maior parte da informação relevante. Isso é importante porque os modelos de aprendizado de máquina podem ser facilmente sobrecarregados por um grande número de *features*, o que pode levar a uma redução na precisão.

Em resumo, a descoberta de *features* é um processo exploratório que busca identificar novas variáveis ou *features* relevantes para um modelo de aprendizado de máquina. Ele pode envolver o uso de técnicas avançadas de estatística e matemática e pode levar a novos *insights* sobre os dados.

2.4 Desenvolvimento de *features* por especialista na área

O papel do especialista de campo na descoberta e desenvolvimento de *features* (*features*) é fundamental para garantir a relevância das *features* usados em um modelo

de aprendizado de máquina. O especialista de campo é um profissional que possui conhecimento especializado em um determinado domínio, seja ele científico, técnico ou de negócios.

O especialista de campo tem a tarefa de identificar quais *features* são relevantes para o modelo de aprendizado de máquina em questão, com base em seu conhecimento e experiência no domínio. Ele pode ajudar a identificar padrões nos dados que podem ser úteis para a modelagem, além de sugerir novas *features* que podem ser relevantes.

Além disso, o especialista de campo pode ajudar a interpretar os resultados do modelo, identificando quais *features* são mais importantes e por que são relevantes para a tarefa em questão. Ele também pode ajudar a definir métricas de avaliação relevantes para o domínio, garantindo que o modelo seja avaliado de acordo com as necessidades específicas do negócio ou área de pesquisa.

Em resumo, o papel do especialista de campo na descoberta e desenvolvimento de *features* é essencial para garantir a relevância e a precisão do modelo de aprendizado de máquina. O especialista de campo pode identificar *features* relevantes com base em seu conhecimento e experiência no domínio, além de ajudar a interpretar os resultados do modelo e definir métricas de avaliação relevantes.

2.5 Features encontradas na revisão da literatura

No contexto dos artigos selecionados para esta revisão da literatura, de modo a avaliar os trabalhos relacionados, classificamos o uso de *features* de modo bastante amplo, considerando como *feature* qualquer transformação na base de dados original. Consideramos como "base de dados original" a série temporal de preços dos ativos estudados. Portanto, qualquer distinção (resultante de transformação) entre a base de dados final e a original, implicaria, segundo nossa definição ampla, no uso de *features*, ainda que o desenvolvimento de *features* não tenha sido o objeto da pesquisa. Partindo dessa definição ampla de *feature* como uma transformação dos dados brutos, em seguida dividimos as *features* em categorias, de acordo com o tipo de transformação realizada.

Os dados de uma série temporal financeira são, geralmente, dados de preço (cotações, ou preço em que o ativo foi negociado) e volume (quantidade de ativos negociados, ou quantidade de dinheiro utilizado nas negociações). Consideramos *features*, portanto, quaisquer transformações que utilizem dados de preço e/ou de volume. As categorias propostas no estudo das *features* apresentadas nos trabalhos estudados são:

a) **Dados de Mercado**

Nesta categoria estão as transformações diretas de preço e volume, sendo transformações padrão (de uso comum no mercado de capitais), ou criadas pelo próprio pesquisador:

i) OHLC:

1. O = “*Open*”, ou preço de abertura do período (granularidade) utilizado na série temporal;
2. H = “*High*”, ou preço máximo do período;
3. L = “*Low*”, ou preço mínimo do período;
4. C = “*Close*”, ou preço de fechamento no período.

ii) Retorno:

1. Retorno, ou diferença entre o preço de fechamento do período e o preço de abertura do período, dado pela fórmula: $close.n - open.n$, onde "n" indica o período de avaliação, que pode ser um dia, uma semana, ou algumas horas ou minutos.

iii) Volume:

1. Volume Financeiro: A quantidade, em moeda corrente, utilizada (dispendida) para realizar as negociações (exemplo: 1 milhão de reais, 5 milhões de dólares americanos, etc).
2. Volume Quantidade: A quantidade, em número de ativos, objeto da negociação (exemplo: 100 ações, 1.000 ações, etc).

iv) Séries Exógenas:

1. Trata-se do uso dos mesmos elementos acima (preço e volume), porém de séries exógenas, ou de outros ativos. (Por exemplo: usar série histórica de preços do petróleo para analisar a série histórica de preços das ações da companhia Petrobrás).

v) *Market Internals*:

1. Transformações de preço, volume e outras informações, criando uma série de dados baseada em relações desses elementos. (Por exemplo: número cumulativo das negociações que ocorreram na ordem de compra versus as negociações que ocorreram na ordem de venda do livro de ofer-

tas).

b) Indicadores Técnicos

São fórmulas matemáticas que geram indicadores plotáveis em tela. Em geral, os participantes do mercado utilizam softwares de análise gráfica para acompanhar as cotações dos ativos negociados. Esses softwares permitem o acompanhamento do histórico dos preços em gráficos com dois eixos: o eixo vertical contém a escala de preço, enquanto que o eixo horizontal apresenta a escala de tempo. Desse modo, é possível visualizar o desenvolvimento da série de preços ao longo do tempo. É permitida também a visualização dos preços em diferentes granularidades de tempo. A seguir, a lista dos Indicadores Técnicos encontrados nos trabalhos estudados:

i) Média Móvel - O indicador média móvel é utilizado para identificar a tendência do preço. Pode ser instanciado utilizando-se uma única referência de tempo, (por exemplo: média móvel de 20 períodos), ou mais de uma referência de tempo, (por exemplo: média de 5, média de 10 e média de 20 períodos).

$$\text{Fórmula: Média Móvel Simples} = \frac{(V_1 + V_2 + \dots + V_n)}{N}$$

ii) MACD - A sigla MACD significa “*Moving Average Convergence Divergence*”. Esse indicador busca identificar a relação de proximidade entre diversas médias móveis. Se elas estão se aproximando ou se distanciando uma das outras, o indicador demonstra essa relação e essa informação fica disponível para análise.

Fórmula: O cálculo é feito a partir de três médias móveis:

a) MME1 – A média móvel exponencial rápida;

b) MME2 – A média móvel exponencial lenta;

c) MACD – O resultado da subtração da média rápida pela média lenta.

d) A MACD, então, é filtrada com uma média móvel exponencial para gerar o “sinal”, e assim ser usado como gatilho de compras e vendas ao cruzar com a linha MACD. Sua fórmula é: $\text{MACD}(x, y, z) = \text{MME1}(x) - \text{MME2}(y)$

iii) IFR (RSI) - Índice de Força Relativa (ou *Relative Strength Index*) é um indicador que identifica a posição relativa do preço atual em relação ao preço máximo e mínimo registrado em uma janela ou período de tempo. Com base nessa posição relativa, estima-se a “força” (no sentido de movimento ascendente) ou “fraqueza” (movimento descendente) do preço.

Fórmula de cálculo do IFR

$$IFR = 100 - \frac{100}{(1 + FR)}$$

FR = Média Altas / Média Baixas *

(*) Para o cálculo da média das altas, usa-se apenas os períodos de alta ou o valor zero nos períodos de baixa. O oposto vale para a média das baixas.

- iv) Estocástico - Esse indicador, apesar do nome, não tem nada a ver com Cálculo Estocástico. Classificado como “oscilador”, ele estima a posição relativa do preço em relação à máxima e mínima de um período de tempo informado.

O indicador Estocástico pode ser calculado através da fórmula:

$$\%K = \frac{100 * (\text{Preço} - LN)}{(HN - LN)}$$

$$\%D = MMA(\%K, X)$$

Onde %K é a linha de sinal do indicador; Preço é o preço do ativo no momento, LN é a mínima dos últimos N dias, HN é a máxima dos últimos N dias, %D é a linha mais lenta do indicador e MMA(%K, X) é a média móvel aritmética de X períodos de %K.

Já, para calcular o Estocástico Lento, usa-se novamente uma média móvel aritmética, mas agora na linha %D e cria o %D lento. Então seu oscilador usa a linha %D como sinal e o %Dlento como linha lenta.

$$\%Dlento = MMA(\%D, X)$$

- v) W%R - Criado pelo investidor Larry Williams, o indicador W%R (Williams Percent Range) é também classificado como indicador “oscilador”, estimando a posição relativa do preço atual em relação à diferença entre a máxima e mínima de um período informado. (“Range” é o nome dado a essa diferença entre máxima e mínima).

Fórmula:

$$\%R = \frac{-(MÁXIMA (MÁXIMA(i - n)) - FECHAMENTO(i))}{(MÁXIMA (MÁXIMA (i - n)) - MIN (MÍNIMA (i - n)))} * 100$$

Onde:

FECHAMENTO (i) – preço de fechamento de hoje;

MAX (MÁXIMA (i - n)) – o máximo mais alto para períodos n anteriores;

MIN (MÍNIMA (i - n)) – o mínimo mais baixo para períodos n anteriores.

- vi) ROC - “*Rate Of Change*” ou “taxa de variação”, é considerado também um indicador “oscilador”, que mede a variação percentual no preço de uma ação. Ou seja, ele compara o preço atual do ativo com o preço de n períodos anteriores. A formação da linha única que compõe o indicador é baseada no seguinte cálculo:

$$\text{ROC} = \frac{(\text{fechamento atual} - \text{fechamento n períodos anteriores})}{(\text{fechamento n períodos anteriores})} * 100$$

- vii) *Aaron Oscillator* - O indicador é composto por duas linhas, chamadas de *Aroon Up* e *Aroon Down*. A primeira procura medir a força da tendência de alta, enquanto que a segunda mede a força da tendência de baixa:

$$\text{Aaron Up} = \frac{\text{Período} - \text{Período max}}{\text{Período}} * 100$$

$$\text{Aaron Down} = \frac{\text{Período} - \text{Período min}}{\text{Período}} * 100$$

Na fórmula, “Período” é um parâmetro que estabelece o tamanho da janela de cálculo, “PeríodoMax” representa o número de barras desde a última máxima, enquanto que “PeríodoMin” o número de barras desde a última mínima.

- viii) OBV - ‘*On Balance Volume*’ é um dos raros indicadores que utiliza não somente as informações de preço, mas também de volume negociado.

Para medir a chamada “pressão compradora” e a “pressão vendedora” o OBV soma o volume negociado nos períodos de alta e subtrai o volume negociado nos períodos de baixa.

O volume negociado em determinado período é somado quando o fechamento é superior à abertura ou subtraído quando o fechamento é inferior à abertura:

Em caso de alta no período (preço fechamento > preço abertura):

$$\text{OBV atual} = \text{OBV anterior} + \text{Volume do Último Período}$$

Em caso de baixa no período (preço fechamento < preço abertura):

$$\text{OBV atual} = \text{OBV anterior} - \text{Volume do Último Período}$$

Se o preço ficou inalterado (preço fechamento = preço abertura)

OBV atual = OBV anterior

- ix) *Bandas de Bollinger* - Criado pelo analista John Bollinger, esse indicador mede a volatilidade da série de preços, baseada no movimento medido em desvios-padrão,

As bandas *Bollinger* são constituídas por um conjunto de três curvas desenhadas em relação aos preços. Elas são traçadas a partir de uma determinada distância de uma média móvel. A banda intermediária é uma medida de tendência intermediária, geralmente uma média móvel simples, que serve como base para as bandas inferiores e superiores.

O intervalo entre as bandas superiores, intermediárias e inferiores é determinado pela volatilidade, geralmente o desvio padrão da mesma data que foi utilizada pela média. Assim, quanto maior a volatilidade de um ativo maior seu desvio padrão. As Bandas de *Bollinger* consistem de:

- Bandas *Bollinger* Intermediárias sendo uma N -período média móvel simples;
 - Bandas *Bollinger* Superiores sendo Bandas *Bollinger* Intermediárias + ($K \times N$ -período desvio padrão);
 - Bandas *Bollinger* Inferiores = Bandas *Bollinger* Intermediárias - ($K \times N$ -período desvio padrão). Valores típicos para K e N são 2 e 20, respectivamente.
- x) CCI - *Commodity Channel Index* - Esse indicador mede o nível atual de preços em relação a um nível médio, em uma janela de tempo determinada pelo usuário. Quanto mais distante o preço estiver desse ponto médio, maior será o valor do indicador. Esse indicador apresenta leituras positivas se o preço estiver acima de média, e negativas se estiver abaixo. Abaixo segue a fórmula do CCI:

$$CCI = \frac{1}{0,015} \times \frac{PT - MM(PT)}{\sigma(PT)}$$

Na equação, temos:

PT = o preço típico, calculado pela soma entre a máxima, a mínima e o fechamento divididos por 3.

MM = a média móvel do preço típico.

$\sigma(PT)$ = o desvio médio absoluto do preço típico.

0,015 = uma constante usada no denominador para facilitar a visualização

dos resultados, contribuindo para que a maior parte dos valores (entre 70% e 80%) fiquem dentro do intervalo entre -100 e 100 .

- xi) *DI - Directional Index* - Foi criado para representar movimento direcional do preço. Pode ser definido como positivo (+DI) ou negativo (-DI). Ambos, +DI e -DI compõem um indicador ensemble chamado ADX, ou *Average Directional Index*. Para estimar o ADX, primeiramente é preciso calcular o +DM, -DM e *True Range* (TR) para cada período. Em geral, se usa 14 períodos por padrão.

$$+DM = H_0 - H_1$$

$$-DM = L_1 - L_0$$

$$TR = \max\{H_0 - L_0, H_0 - C_1, L_0 - C_1\}$$

Onde +DM é o positive directional movement (movimento direcional positivo), H_0 é a máxima atual, H_1 é a máxima do período anterior, -DM é o *negative directional movement* (movimento direcional negativo), L_0 é a mínima atual, L_1 é a mínima do período anterior, TR é o *True Range* (intervalo real), que é o maior intervalo entre a máxima atual e a mínima atual, a máxima atual e o fechamento do período anterior ou a mínima atual e o fechamento anterior.

Em seguida, é preciso pegar as médias móveis de 14 períodos de cada um desses indicadores:

$$ATR = MMA(TR, 14)$$

$$+DI = \frac{MMA(+DM, 14)}{ATR \times 100}$$

$$-DI = \frac{MMA(-DM, 14)}{ATR \times 100}$$

ATR é o *Average True Range* (Intervalo Real Médio), +DI é o *positive Directional Index*, e o -DI é o *negative Directional Index*.

Por fim, é calculado o DX a partir do +DI e -DI e o ADX é a média móvel

de 14 períodos do DX, conforme pode ser visto nas fórmulas abaixo:

$$DX = \frac{|+DI - -DI|}{|+DI + -DI|} \times 100$$

$$ADX = MMA(DX, 14)$$

- xii) *ATR - Average True Range* é um indicador que busca identificar a média “real” de variação do preço. Esse conceito incorpora a ideia de que as diferenças entre o preço de abertura do dia corrente e o fechamento do dia anterior deveriam ser acrescentados ao cômputo da variação do dia atual. A variação “real” do preço (chamada *True Range*), é dada pela fórmula:

$$TR = \max\{high, close[prev]\} - \min\{low, close[prev]\}$$

ATR então seria a média das TR (*True Ranges*) do período selecionado para análise, sendo que o valor padrão é de 14 períodos, mas esse valor pode ser livremente modificado pelo usuário do indicador.

- xiii) *TRIX - Triple Exponential Average* é um indicador que se origina de três médias móveis. A primeira média móvel é calculada a partir do preço. A segunda média móvel é calculada a partir da primeira média móvel. E a terceira média móvel é calculada a partir da segunda média móvel. A direção da linha TRIX indica a direção do movimento de preços.

Pode ser calculada seguindo os seguintes passos:

1. $EMA1 = EMA(\text{Fechamento})$ (calculamos e média móvel exponencial do preço de fechamento)
2. $EMA2 = EMA(EMA1)$ (calculamos e média móvel exponencial do resultado do item anterior)
3. $EMA3 = EMA(EMA2)$ (calculamos e média móvel exponencial do resultado do item anterior)
4. $TRIX = (EMA3[\text{hoje}] - EMA3[\text{ontem}]) / EMA3[\text{ontem}]$ (o indicador TRIX é a mudança percentual do resultado do item anterior de ontem para hoje)

- xiv) *A/D - Advance/Dcline* é um indicador utilizado para medir a quantidade de ações (de um índice ou carteira) que participaram do movimento de alta ou de

baixa do preço. Também é chamada LDA (Linha de Avanço e Declínio).

O cálculo da LAD é extremamente simples. Segue abaixo um passo a passo:

1. Escolher os ativos que participarão da linha de avanço e declínio;
2. Fazer a contagem de quantos fecharam em alta e, igualmente, uma contagem dos ativos que fecharam com viés baixista;
3. Subtrair os ativos que fecharam com viés de baixa dos ativos que fecharam em alta;
4. Plotar este valor em um gráfico.
5. Os próximos valores, na sequência, são calculados simplesmente somando o cálculo do passo a passo acima ao valor da LAD encontrado anteriormente. A fórmula é a seguinte:

$$LAD = LAD \text{ anterior} + \text{Quantidade de altas} - \text{Quantidade de baixas}$$

- xv) Volatilidade - A volatilidade do preço é dada pela média da variação percentual do ativo nos n períodos (em geral, dias). Na fórmula, C é o preço de fechamento (“Close”) no período t , onde t é o dia atual.

$$\frac{\sum_{t-n+1}^t \frac{C_i - C_{i-1}}{C_{i-1}}}{n}$$

- xvi) *Momentum* - O *Momentum* de mercado é medido por meio da diferença no preço em um determinado período de tempo. Para criar uma linha de *Momentum* de 10 dias, subtraímos o preço de fechamento de 10 dias atrás, do último preço de fechamento. Este valor, seja positivo ou negativo, é então marcado em um gráfico ao redor da linha de zero. A fórmula para o *Momentum* é:

$$M = V - V_x$$

Onde V é o último preço de fechamento, e V_x é o preço de fechamento de x dias atrás.

- xv) Disparidade - A disparidade é um indicador da análise técnica que mede a posição relativa do último preço de fechamento para uma média móvel específica, o resultado é representado em porcentagem. Esse indicador tem a

seguinte fórmula:

$$\text{Disparidade} = \frac{\text{preço de fechamento}}{\text{média móvel de n preços de fechamento}}$$

- c) **Indicadores Fundamentalistas:** Os "múltiplos"fundamentalistas derivam de dados econômicos da própria empresa em análise, como, por exemplo, EBIDTA, Fluxo de Caixa Descontado, entre outros. Eles indicam medidas econômicas e contábeis, considerando a estrutura de fluxo de caixa, endividamento, perspectivas de faturamento, entre outros elementos que servem para compor o *valuation* (cálculo do valor) de uma empresa. Para uma referência no assunto, ver (DAMODARAN, 2006)
- d) **Features proprietárias (data mining):** Nesta categoria encontram-se *features* criadas pelos próprios pesquisadores, ou derivadas de técnicas de *data mining* como *webscrapping*, entre outras. Essas *features* são caracterizadas por não fazerem parte de uma biblioteca disponível no formato open source, como é o caso da biblioteca talib.py. Encontram-se aqui quaisquer *features* desenvolvidas pelos próprios autores dos trabalhos, em seus esforços de pesquisa, como, por exemplo, a conversão de notícias sobre as empresas negociadas em bolsa em *features* aproveitáveis para a modelagem.
- e) **Transformações Estatísticas:** Aqui temos *features* que derivam de *feature engineering*, ou seja, extraídas por procedimentos estatísticos (e não necessariamente derivado de experiência de um especialista na área). Essas *features* foram obtidas por tratamento estatístico dos dados, (como por exemplo, encontrar o desvio padrão do conjunto de dados, etc).

i) *Scaling*

$$y_i = 2 \frac{x_i - \min_j x_j}{\max_j x_j - \min_j x_j} - 1$$

ii) Desvio Padrão

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^n (x_i - \mu)^2}$$

iii) Log-retorno (*Daily Log Return*)

$$y_i = \log x_i - \log x_{i-1}$$

2.6 Features utilizadas nos experimentos realizados

Nas subseções 2.6.1 e 2.6.2 estão listadas as *features* que foram utilizadas pelo autor deste trabalho nas investigações realizadas no Capítulo 5 - Experimentos. As *features* que já foram descritas anteriormente neste capítulo são apenas mencionadas (como é o caso da maioria das *features* TALIB), enquanto que as demais recebem uma descrição e sua estrutura (fórmula).

2.6.1 Features SALIB

SALIB - Essas *features* são derivadas do preço do ativo, e estão instanciadas numa biblioteca *python* intitulada *myfeatures.py*. São *features* desenvolvidas pelo autor, e constituem uma proposta de análise estrutural, portanto, no código em que são instanciadas nos experimentos, são chamadas de *features* SALIB (*Structural Analysis Library*). As *features* são descritas abaixo.

1. *Hour* - Indica a hora de cada *datapoint*, uma vez que a série temporal transcorrem em periodicidade de 15 minutos ("intradia").
2. *Gap* - A diferença percentual entre o preço de abertura do dia atual e o preço de fechamento do dia anterior. Fórmula: $\frac{\text{Open} - \text{Close.1}}{\text{Close.1}}$.
3. CTO - "*Close-to-open*" - A variação percentual entre o o fechamento (*close*) e a abertura (*open*) do candle diário. (O dia em que é calculado depende da instanci-
ação: CTO.0 significa a variação percentual do dia atual, no momento do cálculo, CTO.1 refere-se à variação percentual do dia anterior, CTO.2 refere-se ao dois dias atrás, e assim por diante). Fórmula: $\frac{\text{Close} - \text{Open}}{\text{Open}}$.
4. *Quadrant* - Ou "quadrante do mês- O mês do calendário é dividido em três "terços", e o preço atual é identificado como pertencendo a um desses "terços" do mês: início, meio ou fim. Além disso, também posiciona o preço atual no contexto "positivo" ou "negativo", em relação ao preço de abertura do mês.
5. IBS - "*Internal Bar Strength*" - Mede a "força" de compra ou venda no dia de acordo com a posição do preço de fechamento em relação à máxima e a mínima do período. Fórmula:

$$IBS = \frac{(\text{Close} - \text{Low})}{(\text{High} - \text{Low})}$$

6. *FHIGH* - “*From High*” - Calcula a distância percentual entre o preço atual e a “maior máxima” de preço na janela de avaliação. Fórmula:

$$FHIGH = \frac{(H_n - C_0)}{H_n}$$

7. *FLOW* - “*From Low*” - Calcula a distância percentual entre o preço atual e a “menor mínima” de preço na janela de avaliação. Fórmula:

$$FLOW = \frac{(C_0 - L_n)}{L_n}$$

8. *DAILYPOSITION* - Indica a posição relativa do dia em relação ao dia anterior. Considera-se que o dia atual está “Acima” do dia anterior se sua máxima estiver acima da máxima anterior e sua mínima estiver acima da mínima anterior: $ACIMA = H > H_1$ e $L > L_1$. Considera-se que o dia atual está “Dentro” do dia anterior se sua máxima estiver abaixo da máxima anterior e sua mínima estiver acima da mínima anterior: $DENTRO = H < H_1$ e $L > L_1$. Considera-se que o dia atual está “Abaixo” do dia anterior se sua máxima estiver abaixo da máxima anterior e sua mínima estiver abaixo da mínima anterior: $ABAIXO = H < H_1$ e $L < L_1$. Considera-se que o dia atual está “Fora” do dia anterior se sua máxima estiver acima da máxima anterior e sua mínima estiver abaixo da mínima anterior: $H > H_1$ e $L < L_1$. Esse indicador pode ser instanciado com dados de outros dias, seguindo a série temporal: 0, 1, 2, 3, etc.
9. *WEEKLYPOSITION* - Indica a mesma relação de posição relativa apresentada pelo *DAILY POSITION*, porém utiliza dados de periodicidade semanal, desse modo, teremos uma semana “Acima”, “Dentro”, “Abaixo” ou “Fora” da semana prévia.

2.6.2 Features TALIB

TALIB - Aqui são listadas as features Indicadores Técnicos que serão utilizadas nos experimentos (Capítulo 5 - Experimentos). Elas são instanciadas a partir da biblioteca *talib.py*, que é de uso e acesso comum (*opensource*). Serão utilizados os seguintes indicadores técnicos:

1. IFR (RSI)
2. Estocástico

3. MACD
4. ADX
5. CCI
6. MFI - O indicador “*Market Facilitation Index*” mede a força ou fraqueza por trás das movimentações do preço. Sua fórmula é:

$$MFI = \frac{(HIGH - LOW)}{VOLUME}.$$

OBSERVAÇÃO: O MFI foi descrito neste ponto pois é o único indicador técnico que não figurou nos trabalhos relacionados na revisão da literatura. Os demais indicadores técnicos já foram descritos acima.

7. Williams % R (WILLR)
8. TRIX

2.7 Considerações finais

Neste capítulo abordamos a fundamentação teórica do elemento-chave deste trabalho: as *features*. Observamos a importância atribuída a elas no contexto de modelagem algorítmica, seu papel em enriquecer o modelo treinado com relações lineares e não-lineares, a forma como *features* são extraídas do dataset pela engenharia de *features*, o modo como pode ser descobertas ou desenvolvidas e o papel do especialista na área de atuação para atribuir critérios de relevância para as *features*, economizando tempo e recursos de processamento. Em seguida, vimos as *features* encontradas na literatura e as descrevemos, finalizando com as *features* que figuram nos experimentos deste trabalho, instanciados no Capítulo 5 - Experimentos.

3 TRABALHOS RELACIONADOS

Neste capítulo, faremos uma revisão da literatura para avaliar o uso de *features* (atributos) na modelagem de algoritmos de *machine learning* para a criação de modelos preditivos para negociação no mercado de capitais (*algotrading*). Para observarmos o uso de *features* neste contexto, pesquisamos a literatura e organizamos este estudo da seguinte forma: na seção 3.1. fizemos uma discussão dos resultados obtidos nos trabalhos relacionados, na forma de uma apresentação dos trabalhos com ênfase na descrição das *features* encontradas; na seção 3.2. trouxemos uma análise comparativa, classificando os trabalhos por tipo de *features* utilizadas e outras métricas; por fim, na seção 3.3. concluímos com uma apresentação dos resultados encontrados e formamos uma definição quanto ao uso de *features* na literatura corrente.

3.1 Discussão dos Trabalhos Relacionados

Nesta seção, faremos uma discussão sintética sobre cada trabalho. Nosso objetivo não é avaliar a natureza dos experimentos, nem a estrutura dos algoritmos utilizados nem da abordagem utilizada na metodologia de pesquisa, antes, nossa avaliação se trata das características das *features* utilizadas.

Os trabalhos selecionados abrangem o ano 2001 a 2020, e foram realizados por autores dos seguintes países: Brasil, Estados Unidos, China, Cingapura, Canadá, Grécia, Holanda, Índia, Turquia, Marrocos, Tailândia, Filipinas e Taiwan, donde se conclui que o interesse pela negociação por algoritmos ultrapassa fronteiras e encontra entusiastas em diversos continentes.

Apesar da diversidade na nacionalidade dos autores, observamos algumas regularidades: 60% dos trabalhos utilizaram *features* do tipo “Indicadores Técnicos”, 73% dos trabalhos utilizaram *features* do tipo “Dados de Mercado”. Porém, essa desproporção não se sustenta quando calculamos o número de ocorrências em cada tipo: houveram 67 ocorrências de *features* “Indicadores Técnicos”, contra 39 *features* do tipo “Dados de Mercado”. Ou seja, embora mais trabalhos apresentem “Dados de Mercado”, observamos que se limitam a poucas *features*. Já os trabalhos com “Indicadores Técnicos” fazem uso de um maior número de *features*.

Em relação à quantidade de *features* específicas em cada categoria, encontramos apenas 5 *features* do tipo “Dados de Mercado”, contra 17 *features* do tipo “Indicadores

Técnicos”. Em geral, os trabalhos combinam diferentes tipos de *features*. Observamos que muito raramente os autores escolheram uma categoria única de *feature*. Dos trabalhos analisados, apenas 5 utilizaram exclusivamente “Indicadores Técnicos”, enquanto que apenas 9 utilizaram exclusivamente “Dados de Mercado”.

Para dar uma perspectiva mais abrangente, separamos os trabalhos entre aqueles que usaram algoritmos considerados mais simples (*machine learning* com aprendizado supervisionado e não-supervisionado) e os que utilizaram algoritmos considerados mais complexos (aprendizado por reforço e aprendizado profundo). Usando exclusivamente *features* do tipo “Indicadores Técnicos” e *machine learning* (aprendizado supervisionado e não-supervisionado), temos (LOUWERSE; ROTHKRANTZ, 2014), (UPADHYAY et al., 2016), (SEZER; OZBAYOGLU; DOGDU, 2017), (ZHANG et al., 2018), (WANG, 2020). Ou seja, todos os trabalhos que apresentaram unicamente “Indicadores Técnicos” como *features*, optaram por não utilizar abordagens mais complexas, como aprendizado por reforço ou aprendizado profundo.

Em seguida, os trabalhos que utilizaram exclusivamente “Dados de Mercado” como *features*. (WENG, 2009), (ROCA; M’oL, 2015), (CASTRO et al., 2023) e (JOOSERY; DEEPA, 2019), utilizaram algoritmos considerados mais simples, ou seja, (*machine learning* com aprendizado supervisionado e não supervisionado).

Por outro lado, Moody & (MOODY; SAFFELL, 2001), (WANG et al., 2017), (LI; RAO; SHI, 2018), (CHALVATZIS; HRISTU-VARSAKELIS, 2019) e (YOU et al., 2019), que também usaram exclusivamente “Dados de Mercado” como *features*, optaram por utilizar algoritmos considerados mais complexos (aprendizado por reforço e aprendizado profundo).

Um número maior de trabalhos trouxe uma abordagem híbrida, utilizando mais de um tipo de *feature* dentre as observadas (“Indicadores Técnicos”, “Dados de Mercado”, “Transformações Estatísticas”, “Indicadores Fundamentalistas” e “*features* Proprietárias”). Também separamos esses trabalhos entre aqueles que usaram algoritmos mais simples (*machine learning*) e mais complexos (*reinforcement learning* e *deep learning*).

Com a abordagem híbrida no uso de *features*, e utilizando *machine learning* (aprendizado supervisionado ou não-supervisionado), temos (CAO; TAY, 2001), (TAY; CAO, 2002), (CAO; TAY, 2003), (MARTINEZ et al., 2009), (CARAMICO, 2010), (MADGE; BHATT, 2015), (CALAINHO, 2015) e (LABIAD; BERRADO; BENABBOU, 2018).

Com a abordagem híbrida no uso de *features*, porém utilizando algoritmos con-

siderados mais complexos (aprendizado por reforço e aprendizado profundo), temos (CHEN; CHEN; FUSHIMI, 2017), (TORRALBA, 2019), (CHO et al., 2019), (LI; ZHENG; ZHENG, 2019) e (PATIL et al., 2020). Dos trabalhos que consideramos *sui generis*, por utilizarem *features* de outra natureza, temos (WENG, 2009) e (LEUNG; MACKINNON; WANG, 2014), com o uso de *features* derivadas da chamada “Análise Fundamentalista” no mercado de capitais, temos (NAMETALA et al., 2016), autores que inovaram com o uso de modelos econométricos como *features*, e, também, (ONCHAROEN; VATEEKUL, 2018), com o uso de *features* proprietárias derivadas de notícias extraídas dos sites Reuters (notícias financeiras) e Reddit (notícias).

3.2 Análise Comparativa

A seguir veremos uma análise comparativa dos trabalhos. Essa análise não trata dos resultados financeiros (simulados ou reais) que os trabalhos produziram. O objetivo foi identificar o modo como as *features* são utilizadas nos trabalhos cujo objeto de pesquisa era *algotrading*. A comparação dos trabalhos seguiu os seguintes critérios: a presença de *features*, o tipo de *features* encontradas, a complexidade dos algoritmos utilizados nos trabalhos, a subdivisão dos tipos de *features* em seus casos específicos (por exemplo, como tipo de *feature* podemos ter “Indicadores Técnicos”, e como caso específico podemos ter “Médias Móveis”), e uma comparação temporal da evolução do uso das *features* ao longo das duas décadas (2000 a 2020) em que os trabalhos foram produzidos.

As *features* encontradas se dividem em três tipos: “Dados de Mercado”, “Indicadores Técnicos” e “Transformações Estatísticas”. Conforme mostra a Tabela 3.1, essa foi a proporção encontrada: *features* do tipo “Dados de Mercado” foram encontradas 39 ocorrências, *features* do tipo “Indicadores Técnicos” foram encontradas 67 ocorrências, e as *features* do tipo “Transformações Estatísticas” foram encontradas 2 ocorrências.

Tabela 3.1: Ocorrência de features na literatura

Tipo de Feature	Ocorrência nos Trabalhos
INDICADORES TÉCNICOS	67
DADOS DE MERCADO	39
TRANSFORMAÇÕES ESTATÍSTICAS	2

No conjunto das *features* tipo “Indicadores Técnicos”, temos a seguinte proporção: MA (médias móveis) com 14 casos ou 21% das ocorrências, MACD com 9 casos ou 13% das ocorrências, IFR com 9 casos ou 13% das ocorrências, W%R com 5 casos ou 7%

das ocorrências, ROC (Rate Of Change) com 5 casos ou 7% das ocorrências, Bandas de Bollinger com 4 casos ou 6% das ocorrências, CCI (Commodity Channel Index) com 4 ou 6% das ocorrências, STOCH (Estocástico) com 3 ou 4% das ocorrências. Todos os indicadores a seguir foram agrupados sob o rótulo “Outros”, que contém 21% das ocorrências: ATR com 3 casos, MOMENTUM com 3 casos, OBV com 2 casos, e os próximos indicadores tiveram apenas 1 caso: AARON OSCILATOR, DI, TRIX, A/D, VOLATILITY e DISPARITY. A Tabela 3.2 mostra o número de ocorrências de cada Indicator Técnico:

Tabela 3.2: Número de ocorrências de Indicadores Técnicos na literatura

Indicador Técnico	Ocorrência nos Trabalhos
MA	14
MACD	9
IFR	9
W%R	5
ROC	5
BB	4
CCI	4
STOCH.	3
ATR (EM: OUTROS)	3
MOMENTUM (EM: OUTROS)	3
OBV (EM: OUTROS)	2
AARON OSCIL. (EM: OUTROS)	1
DI (EM: OUTROS)	1
TRIX (EM: OUTROS)	1
A/D (EM: OUTROS)	1
VOLATILITY (EM: OUTROS)	1
DISPARITY (EM: OUTROS)	1

No conjunto das *features* tipo “Dados de Mercado”, temos a seguinte proporção: OHLC com 18 casos ou 46% das ocorrências, RETORNO com 9 casos ou 23% das ocorrências, VOLUME com 6 casos ou 15% das ocorrências, SÉRIES EXÓGENAS com 5 casos ou 13% das ocorrências e MARKET INTERNALS com 1 caso ou 3% das ocorrências. A Tabela 3.3 nos mostra o número de ocorrências:

Tabela 3.3: Número de ocorrências de features "Dados de Mercado" na literatura

Dado de Mercado	Ocorrência nos Trabalhos
OHLC	18
RETORNO	9
VOLUME	6
SÉRIES EXÓGENAS	5
MARKET INTERNALS	1

Finalmente, no conjunto das *features* tipo “Transformações Estatísticas”, temos a seguinte proporção: LOG-RETORNO apresenta 1 caso, DESVIO-PADRÃO também apresenta 1 caso. (Desse modo, não há uma representação gráfica adequada para as *features* do tipo “Transformações Estatísticas”, por falta de dados.) A Tabela 3.4 referencia o número de ocorrências:

Tabela 3.4: Número de ocorrências das features "Transformações Estatísticas" na literatura

Transformação Estatística	Ocorrência nos Trabalhos
LOG-RETORNO	1
DESVIO PADRÃO	1

Relativo às *features* do tipo “Dados de Mercado”, vemos que os trabalhos (MOODY; SAFFELL, 2001), (JOSEPH; NOBAKHT; LONI, 2012), (WENG, 2009), (TAY; CAO, 2002), (MARTINEZ et al., 2009), (SEZER; OZBAYOGLU; DOGDU, 2017), (LI; RAO; SHI, 2018), (YOU et al., 2019), (TORRALBA, 2019), (CHO et al., 2019), (SONG; LEE, 2019), (JOOSERY; DEEPA, 2019), (LOUWERSE; ROTHKRANTZ, 2014), (PATIL et al., 2020) e (UPADHYAY et al., 2016) apresentaram dados de mercado do tipo OHLC, ou seja, preço de abertura, máxima, mínima ou fechamento. Apenas os trabalhos (LI; RAO; SHI, 2018), (CASTRO et al., 2023), (ROCA; M’oL, 2015), (CARAMICO, 2010), (CALAINHO, 2015), (TORRALBA, 2019), (CHO et al., 2019) e (WANG, 2020) utilizaram dados de mercado do tipo “Retorno”. A *feature* “Volume”, indicando o número de ativos negociados no dia, foi utilizada somente nos trabalhos (WENG, 2009), (LI; RAO; SHI, 2018), (TORRALBA, 2019), (CHO et al., 2019) e (LOUWERSE; ROTHKRANTZ, 2014). Somente os trabalhos (JOSEPH; NOBAKHT; LONI, 2012), (WENG, 2009), (TAY; CAO, 2002) e (LI; RAO; SHI, 2018) utilizaram outras séries temporais (exógenas) para auxiliar na previsão da série objeto do estudo.

No aspecto econômico, como são dados de periodicidade maior (semanas, meses,

anos), apenas os trabalhos (WENG, 2009), (LI; RAO; SHI, 2018), (CAO; TAY, 2001) e (SONG; LEE, 2019) utilizaram dados de Análise Fundamentalista (múltiplos). Além disso, apesar de contraintuitivo, por tratarmos de pesquisa sobre mercado financeiro, apenas dois trabalhos ((MARTINEZ et al., 2009) e (SONG; LEE, 2019)) utilizaram dados econômicos!

Os “Indicadores Técnicos” foram o tipo de *feature* mais utilizado, figurando nos trabalhos: (CASTRO et al., 2023), (CARAMICO, 2010), (MADGE; BHATT, 2015), (CALAINHO, 2015), (CAO; TAY, 2003), (CHALVATZIS; HRISTU-VARSAKELIS, 2019), (LEUNG; MACKINNON; WANG, 2014), (ZHANG et al., 2018), (LABIAD; BERRADO; BENABBOU, 2018), (SONG; LEE, 2019), (WANG, 2020), (CHEN; CHEN; FUSHIMI, 2017), (LOUWERSE; ROTHKRANTZ, 2014) e (PATIL et al., 2020). Seu amplo uso será explicado abaixo, em “Considerações do Capítulo”.

Quanto aos demais tipos de *features*, “Transformações Estatísticas” foram utilizadas apenas nos trabalhos (YOU et al., 2019) e (LABIAD; BERRADO; BENABBOU, 2018), enquanto que “Notícias” foram usadas como *feature* apenas no trabalho (ONCHAROEN; VATEEKUL, 2018). A seguir, uma análise entre complexidade dos algoritmos vis-a-vis o tipo e número de *features* utilizadas. Para tal, separamos os trabalhos em algoritmos de “Machine Learning” e de “Deep Learning”.

Dezenove trabalhos utilizaram algoritmos de *machine learning* (Redes Neurais, *Support Vector Machines*, etc): (MOODY; SAFFELL, 2001), (JOSEPH; NOBAKHT; LONI, 2012), (WENG, 2009), (TAY; CAO, 2002), (LI; RAO; SHI, 2018), (CASTRO et al., 2023), (ROCA; M’oL, 2015), (CARAMICO, 2010), (MADGE; BHATT, 2015), (CALAINHO, 2015), (CAO; TAY, 2001), (CAO; TAY, 2003), (CHALVATZIS; HRISTU-VARSAKELIS, 2019), (LEUNG; MACKINNON; WANG, 2014), (ZHANG et al., 2018), (LABIAD; BERRADO; BENABBOU, 2018), (JOOSERY; DEEPA, 2019), (CHEN; CHEN; FUSHIMI, 2017) e (BABA; INOUE; YANJUN, 2002). Onze trabalhos utilizaram algoritmos de Deep Learning: (MARTINEZ et al., 2009), (SEZER; OZBAYOGLU; DOGDU, 2017), (YOU et al., 2019), (ONCHAROEN; VATEEKUL, 2018), (TORRALBA, 2019), (CHO et al., 2019), (SONG; LEE, 2019), (WANG, 2020), (LOUWERSE; ROTHKRANTZ, 2014), (PATIL et al., 2020) e (UPADHYAY et al., 2016).

No conjunto dos trabalhos utilizando *machine learning*, foram utilizadas 17 *features* “Dados de Mercado” (25,37%), 3 *features* de “Economia” (4,48%), e um total de 46 *features* “Indicadores Técnicos” (68,66%), 1 *feature* “Transformações Estatísticas” (1,49%) e 0 (zero) *features* “Notícias”.

No conjunto dos trabalhos utilizando Deep Learning, foram utilizadas 15 *features* “Dados de Mercado” (36,59%), 3 *features* de “Economia” (7,32%), e um total de 21 *features* “Indicadores Técnicos” (51,22%), 1 *feature* “Transformações Estatísticas” (2,44%) e 1 *feature* “Notícias” (2,44%). Observamos na Tabela 3.5 a distribuição de ocorrência de *features* por tipo de algoritmo:

Tabela 3.5: Distribuição de tipos de *features* por tipos de algoritmo na literatura

	Dados de Mercado	Economia	Indicadores Técnicos	Transformações Estatísticas	Notícias
Machine Learning	25,37%	4,48%	68,66%	1,49%	0%
Deep Learning	36,59%	7,32%	51,22%	2,44%	2,44%

Observamos que a complexidade do algoritmo não influenciou na representatividade do tipo de *feature*: as *features* “Indicadores Técnicos” representam a maioria, seguidas de “Dados de Mercado”. As demais *features* (“Economia”, “Transformações Estatísticas” e “Notícias” tem baixa representatividade.

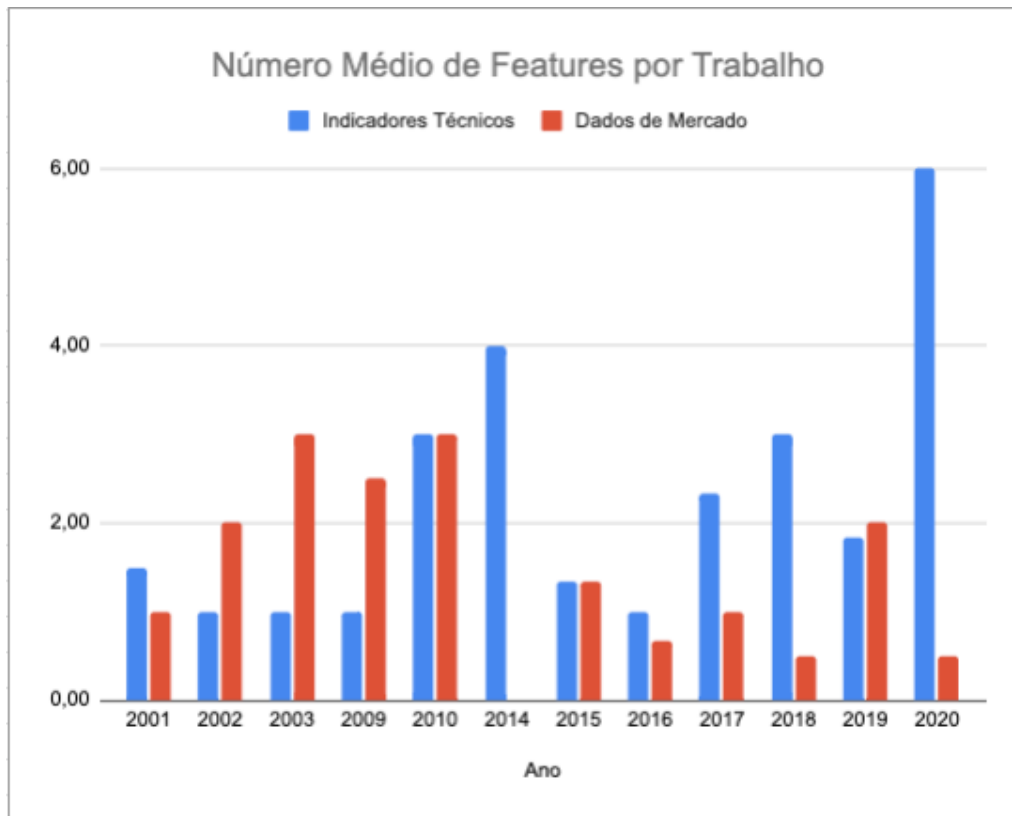
Quanto uma possível evolução do uso de *features* ao longo do tempo, (os trabalhos analisados abrangem 19 anos, desde 2001 a 2020), plotamos o uso das *features* mais representativas, “Indicadores Técnicos” (AT, de Análise Técnica) e “Dados de Mercado” (OHLC), e analisamos o uso das mesmas ano a ano. Para normalizar os dados, dividimos o número de *features* pelo número de trabalhos de cada ano, conforme a Figura 3.1. Observamos que o número médio de *features* “Dados de Mercado” (OHLC) caiu a partir de 2010, enquanto que a partir desse período, aumentou o uso de *features* “Indicadores Técnicos”.

3.3 Considerações do Capítulo

De modo geral, observamos que as *features* não ocupam um ponto de relevância nos esforços de pesquisa. A maior parte dos trabalhos que encontramos foca nos aspectos técnicos do algoritmos utilizados, e, quando é feito algum trabalho de customização, geralmente são alterações propostas no algoritmo em si, ou no modo de utilizá-lo.

Essa declaração de que as *features* não ocupam papel central na pesquisa se deve ao fato da enorme representatividade das *features* do tipo “Indicadores Técnicos”. Por que isso ocorre?

Figura 3.1: Representatividade de Features nos Trabalhos por Ano



Em primeiro lugar, acreditamos que a ampla utilização dos Indicadores Técnicos pelos participantes do mercado responda por sua representatividade ao escolherem-se *features* de mercado de capitais. Justamente por serem acessíveis e amplamente divulgadas na mídia especializada (revistas, cursos e sites), tornam-se “lugar comum” quando se pensa em *features* dessa natureza.

Ainda, a presença da biblioteca talib na linguagem Python (a linguagem mais amplamente utilizada quando se trata de *machine learning* e Deep Learning), torna fácil a instanciação e uso dessas *features* no contexto da modelagem. Instanciar uma *feature* presente em biblioteca é relativamente mais fácil do que criar uma *feature* proprietária, que exigiria tempo e análise da série temporal com o intuito de extrair *features* novas.

Todas as outras *features* (Dados de Mercado, Transformações Estatísticas, Notícias e *features* Econômicas) exigem um trabalho extra de pesquisa (pela fonte da *feature*), ou de transformação do dataset original (série temporal do ativo estudado) em *features* que derivam do preço, o que constituiria um trabalho de *feature engineering*. Claramente, o escopo dos trabalhos era comprovar a eficácia de um algoritmo específico ou arquitetura de algoritmos na tarefa de previsão, de modo que acrescentar complexidade maior ao uso

de *features* ocuparia recursos de tempo, esforço e pesquisa que poderiam ser melhores utilizados no escopo dos próprios trabalhos.

Embora os trabalhos apresentem diferenças em complexidade dos algoritmos utilizados, observamos que a maior complexidade nos algoritmos não levou a um esforço de utilizar ou produzir *features* de maior complexidade. Ao contrário, as *features* foram tratadas como elementos que “já existem”, (particularmente em bibliotecas como *talib*), ao invés de elementos “a serem desenvolvidos”.

Nosso trabalho avança a literatura do tema ao sugerir que as *features* ocupem um papel central como elemento de pesquisa. Por menor ou maior que seja a complexidade de um algoritmo, salvo se tenha sido desenvolvido de forma proprietária (isto é, não open source), ele geralmente está disponível à comunidade de desenvolvedores e pesquisadores, por meio de bibliotecas na linguagem Python (ou em outra linguagem). Existe mesmo um esforço em tornar um algoritmo público, para angariar usuários e, possivelmente, qualificar o algoritmo como mainstream entre os recursos de pesquisa.

Desse modo, os algoritmos são distribuídos livremente, e qualquer pesquisador, cientista ou diletante, pode ter acesso via biblioteca da linguagem de programação em que o algoritmo foi desenvolvido. Com isso se observa que o uso dos algoritmos difere apenas em relação ao objeto da pesquisa (dataset sobre o qual o algoritmo será utilizado), com um grau maior ou menor de modificações proprietárias por parte dos pesquisadores. A maioria, porém, utiliza os algoritmos no estado em que se encontram disponíveis, sem propor alterações em sua estrutura.

Ao propor as *features* como elemento central da pesquisa em *machine learning* aplicado ao *algotrading*, trazemos à tona duas proposições: a primeira é relativa à noção de que é muito mais fácil propor uma inovação no campo das *features* do que no desenvolvimento de novos algoritmos, e, mesmo assim, não observamos um esforço real neste sentido. É como se houvesse uma área de maior facilidade com potencial de maior retorno para os objetivos de pesquisa, porém essa oportunidade fosse totalmente negligenciada. A segunda proposição é relativa à necessidade da presença de um especialista da área, pois o esforço de desenvolver *features* novas depende de um conhecimento profundo do objeto de pesquisa.

Segundo o conceito “*garbage in, garbage out*”, acreditamos que as *features* são um fator relevante para diminuir ou limitar a quantidade de “*garbage*” dos *datasets* que alimentam um algoritmo. Os algoritmos, sozinhos, não são capazes de lidar com a quantidade de ruído de uma série temporal financeira, área em que a presença da aleatoriedade

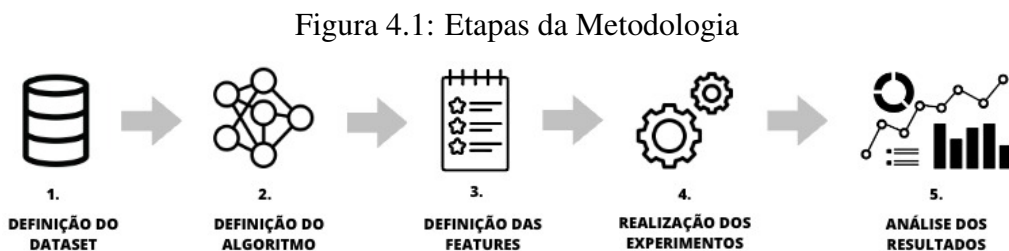
imperera. Sobra para as *features*, portanto, o papel redentor de qualificação dos *datasets*, tornando-os mais adequados à pesquisa.

4 METODOLOGIA

Este capítulo apresenta a metodologia que será utilizada na pesquisa deste trabalho, cujo objetivo é observar o impacto do uso de *features* no retorno apresentado por um algoritmo de *machine learning* no contexto de *algotrading* (negociação automática de ativos financeiros). As principais características dessa metodologia são, em primeiro lugar, sua fácil aplicabilidade (reprodutividade), sua proposta sequencial linear e a possibilidade de variação de seus elementos (*dataset*, algoritmo, *features*, experimentos) de acordo com os objetivos do pesquisador. O capítulo está dividido em: 4.1 -Visão Geral (onde abordamos o contexto em que se insere a pesquisa), 4.2 - Proposta Metodológica (com a apresentação da metodologia utilizada) e 4.3- Descrição da Metodologia (com informações sobre cada aspecto da mesma).

4.1 Visão Geral

Para nossa investigação sobre o papel (que consideramos importante) das *features* na tarefa de desenvolvimento de algoritmos para o mercado financeiro, precisamos delinear uma abordagem de pesquisa que seja, ao mesmo tempo, eficaz e facilmente replicável, uma vez que a revisão pelos pares é uma das formas de garantir a legitimidade científica. A metodologia deve estar alinhada com o objetivo desta pesquisa, ou seja: “Avaliar a importância do desenvolvimento e uso de *features* em treinamento de modelos para o mercado de capitais.” A Metodologia segue conforme a Figura 4.1:



Fonte: elaborada pelo autor.

4.2 Descrição da Metodologia

4.2.1 Definição do *dataset*

Na nossa pesquisa, o *dataset* corresponde à série temporal de um ativo financeiro negociável. Além de um algoritmo único para a pesquisa, escolhemos também um único ativo para a mesma. Desse modo podemos avaliar claramente os diferentes resultados encontrados como sendo derivados unicamente das diferentes combinações de *features*, e não devido às características intrínsecas de ativos financeiros diversos (caso tivéssemos optado por mais de um ativo).

A metodologia pode ser aplicada a diferentes ativos, de diversos mercados, bastando que tenham uma série temporal negociável (ativos como índices, que são apenas *benchmarks* não negociados, não se aplicam). O ativo precisa ter:

- Série histórica composta das seguintes informações a cada *datapoint*:
 - Preço de abertura (o “O”, de OHLC, correspondente a “*Open*”).
 - Preço de máxima no período (o “H”, de OHLC, correspondente a “*High*”).
 - Preço de mínima no período (o “L”, de OHLC, correspondente a “*Low*”).
 - Preço de fechamento (o “C”, de OHLC, correspondente a “*Close*”).
 - Alternativamente, pode ser incluída a informação de Volume ou de Número de Contratos (ou Ações, se for o caso) negociados em cada *datapoint*.

É importante que a série histórica seja o mais extensa possível, para permitir a separação em, no mínimo, dois conjuntos de dados: um correspondendo ao período de treino (modelagem) do algoritmo, e outro correspondendo ao período de teste, para comparação de resultados. Porém, para evidenciar o impacto exclusivo das *features*, preferimos, no âmbito deste trabalho, utilizar um único ativo. Em pesquisa futura, poderemos avançar incluindo diferentes ativos (*datasets*) e ampliar o conhecimento obtido pelos experimentos.

Para que a pesquisa seja representativa, precisaremos escolher um ativo que seja o menos correlacionado possível a uma área específica da economia, ou seja, que não possua viés intrínseco. Por exemplo, se escolhêssemos as ações da Petrobrás como objeto de pesquisa, teríamos possivelmente um viés atrelado ou correlacionado com a variação dos preços do petróleo. Nosso objetivo nesse aspecto seria evitar vínculos diretos com algum setor específico.

Também é um objetivo de nossa pesquisa que o ativo escolhido seja amplamente conhecido no mercado financeiro, e frequentemente negociado (com bom volume diário negociações), ou seja, preferimos não escolher ativos “obscuros”, só conhecidos de investidores muito sofisticados. Para a nossa pesquisa, quanto mais “popular” o ativo, melhor, pois há maiores chances dos resultados encontrados serem mérito das *features* desenvolvidas, e não de vieses intrínsecos do ativo. (Ativos “populares” possuem maior número de negociações, ou seja, volume negociado maior, de modo que são menos suscetíveis de manipulação por parte de participantes do mercado.)

4.2.2 Definição do algoritmo

Nesta etapa escolhemos um algoritmo de machine learning para conduzir os experimentos. O propósito da pesquisa é avaliar o uso de *features* na modelagem de algoritmos para o mercado financeiro. Desse modo, a ênfase está na construção e escolha das *features*, e não, necessariamente, na escolha de um algoritmo. Porém, como parte integrante da pesquisa, o algoritmo precisa ser escolhido de forma adequada. Vamos ver isso logo abaixo.

O primeiro passo é identificar o tipo de algoritmo a ser escolhido. Em negociação no mercado financeiro, existem três ações básicas que o investidor pode tomar: comprar o ativo, para buscar lucros num movimento de alta dos preços, vender a descoberto, para buscar lucros num movimento de baixa dos preços, ou, alternativamente, não negociar, caso identifique um cenário neutro ou sem direção definida.

Dessa forma, as previsões do algoritmo serão, necessariamente, dessas três classes: “compra”, “venda” ou “não negociar”. Por esse motivo, escolheremos um algoritmo do tipo Aprendizado Supervisionado, uma vez que é necessário treinar o modelo para aprender a identificar esses três tipos de cenários, que podem ser descritos como classes (*labels*).

O segundo aspecto é que damos preferência para algoritmos de Classificação, pelo seguinte motivo: não estamos tentando prever um preço específico para o próximo *datapoint* (o que nos levaria a usar algoritmos que utilizam Regressão), mas, sim, estamos tentando prever o movimento “genérico” do próximo período (composto de n *datapoints*), onde o que importa é a descrição do movimento, e não a variação exata do preço. Desse modo, algoritmos de Classificação atendem a nossa necessidade.

O terceiro e último aspecto envolvido na escolha do algoritmo é a necessidade

do algoritmo ter características tais que, em si mesmo ele sirva como base estável para diferentes testes. Ou seja, o elemento “algoritmo” deveria ser imutável, não passível de alteração em seus parâmetros de um teste para o outro.

Isso porque a investigação está centrada na variação do uso de *features*, e precisamos medir o seu impacto nos resultados de forma precisa. Para tanto, precisamos de um algoritmo que se mantenha constante através dos diferentes testes feitos ao longo da nossa investigação, testes em que alteramos a quantidade e tipo de *features* e comparamos os resultados.

Desse modo, a escolha recai em um algoritmo que prescindia de hiperparâmetros, ou, alternativamente, que tenha hiperparâmetros mas que esses não variem de um experimento para o outro. Ou seja, caso tenha hiperparâmetros, eles devem se manter fixos durante todos os experimentos, de modo a não influenciar nos resultados.

4.2.3 Definição das *features*

Nessa etapa, iniciamos a escolha das *features* que serão utilizadas nos experimentos da pesquisa. No capítulo anterior (“Trabalhos Relacionados”) foram definidos diversos tipos de *features* possíveis de serem utilizadas em modelagem para *algotrading* no mercado financeiro.

No próximo capítulo (“Experimentos”) detalharemos as *features* utilizadas, que correspondem a duas categorias: *features* SALIB (desenvolvidas pelo autor e derivadas do preço do ativo) e *features* da biblioteca TALIB (compostas de indicadores de Análise Técnica). Será fornecida a lista completa das *features* instanciadas nos experimentos, bem como suas combinações em diferentes treinamentos do algoritmo.

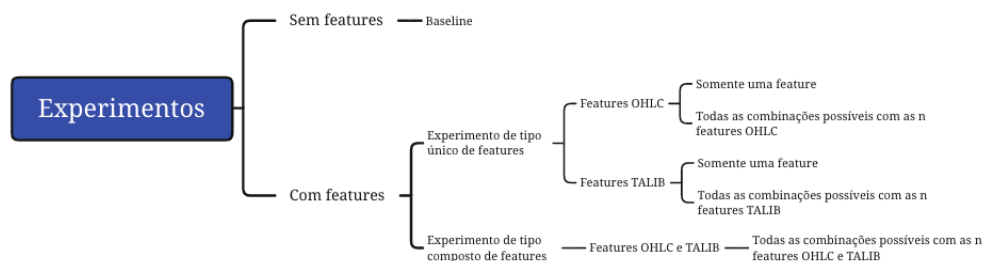
4.3 Realização de experimentos

O objetivo dos experimentos é investigar a premissa deste trabalho, de que o uso de *features* desenvolvidas por um especialista na área permite obter retornos melhores na modelagem de algoritmos de machine learning para *algotrading* no mercado de capitais do que a alternativa (que constitui o *baseline*), que seria utilizar o algoritmo de machine learning sem quaisquer *features*. Para realizar essa investigação propomos experimentos que partem do *baseline* (algoritmo utilizado sem *features*) e, gradativamente, incluem

features desenvolvidas pelo autor e *features* derivadas de Indicadores Técnicos, de amplo uso no mercado financeiro.

Além da utilização crescente de *features* ao longo dos experimentos, são feitas também combinações entre diferentes *features* de um mesmo tipo, (explorando todas as combinações possíveis das *features* disponíveis), e, também, combinações de *features* de tipos diversos. A imagem abaixo (4.2) descreve o panorama dos testes a serem realizados (configuração experimental) e o uso crescente de *features* à medida que os experimentos avançam. Essa imagem serve de base para a estrutura dos testes e será detalhada no próximo capítulo.

Figura 4.2



Fonte: elaborada pelo autor.

Experimentos - Sem features - Baseline - Conforme podemos ver na Figura acima (4.2), inicialmente é realizado um experimento sem o uso de *features*, em que o algoritmo é treinado apenas com os dados de preço e de retorno percentual na janela de tempo utilizada. Para a janela de tempo de previsão (em minutos) foi considerada uma negociação de compra (compra seguida de venda), de modo a buscar resultado com uma possível alta do mercado. Significa comprar a cada 15 minutos (periodicidade do dataset) e vender no final do período de tempo utilizado na previsão. Essa abordagem corresponde ao que no mercado financeiro se chama de “*long only*”, ou seja, tentar lucrar com a alta dos preços, estando exposto a risco caso haja uma queda dos mesmos. Desse modo, o *baseline* ou ponto de partida sendo um treino sem *features* nos permite observar adequadamente a diferença de retorno nos experimentos seguintes, onde gradativamente serão inseridas diferentes *features* no treinamento, conforme o objetivo dessa pesquisa.

Experimentos - Com Features - Experimento de tipo único de features - Em seguida, como vemos na Figura 4.2, iniciam-se os experimentos com *features*, partindo de experimentos com um tipo único de *features* (OHLC ou TALIB), onde as *features* OHLC foram desenvolvidas pelo autor, e as *features* TALIB correspondem a Indicadores Técni-

cos de amplo uso no mercado. Nessa primeira etapa experimental, são utilizadas apenas *features* do tipo OHLC ou TALIB, mas não os dois tipos de *features* em conjunto. Mesmo não havendo mistura de *features* de tipos diferentes, nessa etapa são feitos experimentos com *features* individuais, e também experimentos com as combinações possíveis das *features* de mesmo tipo.

Experimentos - Com Features - Experimento de tipo composto de features - Por fim, ainda de acordo com a Figura 4.2, temos os experimentos de tipo composto de *features*, ou seja, experimentos em que são usados ao mesmo tempo *features* OHLC e TALIB, com todas as combinações possíveis. Todos os experimentos são realizados utilizando-se um *split* do *dataset*, a saber: período de treinamento (correspondendo a 80% do *dataset*) e período de teste (correspondendo a 20% do *dataset*). Como o objetivo do trabalho não é o desenvolvimento completo de uma estratégia operacional (o que exigiria também um período pós-teste, ou de validação), não será utilizada a etapa de validação dos modelos. O objetivo está constricto à avaliação dos modelos treinados num período *out-of-sample* (teste), enquanto que o período *in sample* corresponde ao treinamento com as *features*.

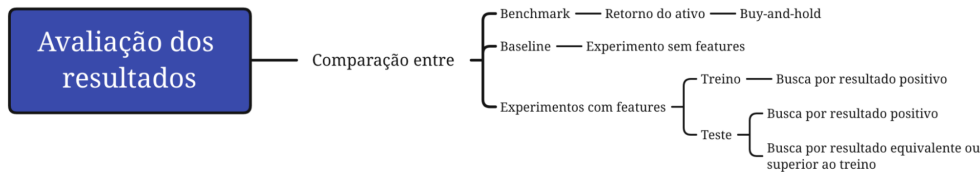
A escolha de 80% do *dataset* para treino do algoritmo assenta-se numa prática empírica relevante no mercado financeiro, segundo a qual um treinamento (ou teste histórico) é considerado mais relevante quanto maior for a janela de tempo utilizada no *dataset*. A ideia por trás dessa prática é de que, com um número maior de observações (*datapoints*), o *dataset* representa o ativo financeiro em uma maior quantidade de cenários de negociação em Bolsa, ou seja: cenários de alta (otimistas), de baixa (pessimistas), laterais (incertos), períodos de pequena ou de grande volatilidade, crises, *crashes*, etc. Desse modo, quanto maior for o período de tempo do *dataset*, ele é considerado mais robusto por fornecer o comportamento do ativo nesses diversos cenários.

A escolha de 20% do *dataset* para teste é justificada pela necessidade de termos um resultado *out-of-sample*, ou seja, não contaminado pelo aprendizado, no qual o modelo treinado pode então realizar previsões com novos *datapoints*. Como veremos abaixo, na Avaliação dos Resultados (4.3.6), é justamente na correspondência entre os resultados de treino e de teste que avaliamos o impacto positivo ou negativo do uso das diferentes *features*.

4.3.1 Avaliação dos resultados

A Figura 4.3 descreve o procedimento utilizado para avaliar os resultados dos experimentos. São feitas comparações entre os retornos obtidos pelo *benchmark*, *baseline* e experimentos com *features*:

Figura 4.3



Fonte: elaborada pelo autor.

Benchmark - Segundo a 4.3, acima, a primeira referência de retorno obtida na pesquisa é o retorno *benchmark* do ativo, considerando-se sua valorização percentual ao longo do tempo, no período avaliado (janela de tempo do dataset). Essa é uma forma comum de avaliação de ativos no mercado financeiro, conhecida como *buy-and-hold* (comprar e segurar), ou gestão passiva, onde não é feito nenhum esforço técnico para incrementar o resultado da valorização do ativo. (O contrário seria gestão ativa, onde há uma atuação direta do investidor em termos de negociação ativa - baseada em compras e vendas). A abordagem *buy-and-hold*, composta por uma negociação de compra no início do período e uma venda apenas no último ponto do período. Ou seja, se o período de teste corresponde a x meses (ou x dias), *buy-and-hold* significaria comprar no primeiro *datapoint* desta série e vender apenas no último. Corresponde à abordagem padrão dos investidores no mercado e é o nosso *benchmark* nesta pesquisa.

Baseline - Em seguida, obtém-se o retorno percentual cumulativo da abordagem “*long only*”, configurada como *baseline*. É a partir desta que observaremos a diferença de resultados nos experimentos seguintes, os quais incluirão *features*.

Experimentos com features: em seguida, temos os experimentos propriamente ditos, utilizando *features* diversas no treinamento do algoritmo. A avaliação consistirá na comparação das Métricas entre os resultados obtidos no período de treino (80% do *dataset*) e no período de teste (20% do *dataset*).

A comparação entre os resultados do *benchmark*, *baseline* e experimentos será feita com o uso de Métricas de Avaliação. Essas métricas serão explicadas no capítulo

seguinte (Capítulo 5 - Experimentos).

5 EXPERIMENTOS

Este capítulo apresenta os experimentos que foram realizados na pesquisa das questões de investigação deste trabalho. Desejamos investigar o impacto gerado nos resultados de um modelo de *machine learning* quando utilizado para a previsão de séries temporais financeiras, à medida que inserimos *features* de diversos tipos e com diferentes combinações entre si.

As questões que pretendemos investigar se referem à variação dos resultados em relação ao tipo de *features* utilizadas (OHLC ou TALIB), à progressão do número de *features* e do número de *features* utilizadas nas diferentes combinações. Os experimentos derivam diretamente das questões de investigação propostas, desse modo, são uma instanciamento da metodologia apresentada no Capítulo 4 - Metodologia.

Neste capítulo descrevemos experimentos e seus resultados. Na Seção 5.1 temos a definição da base de dados (*dataset*) ou ativo financeiro negociável que é utilizado como base para os experimentos, na Seção 5.2. definimos o algoritmo utilizado nos experimentos e suas características, na Seção 5.3. são escolhidas as *features* para alimentar o algoritmo em seu treinamento, na Seção 5.4. descrevemos os experimentos que foram realizados, na Seção 5.5. apresentamos os resultados obtidos, e, na Seção 5.6, apresentamos as considerações finais do capítulo.

5.1 Definição do *dataset*

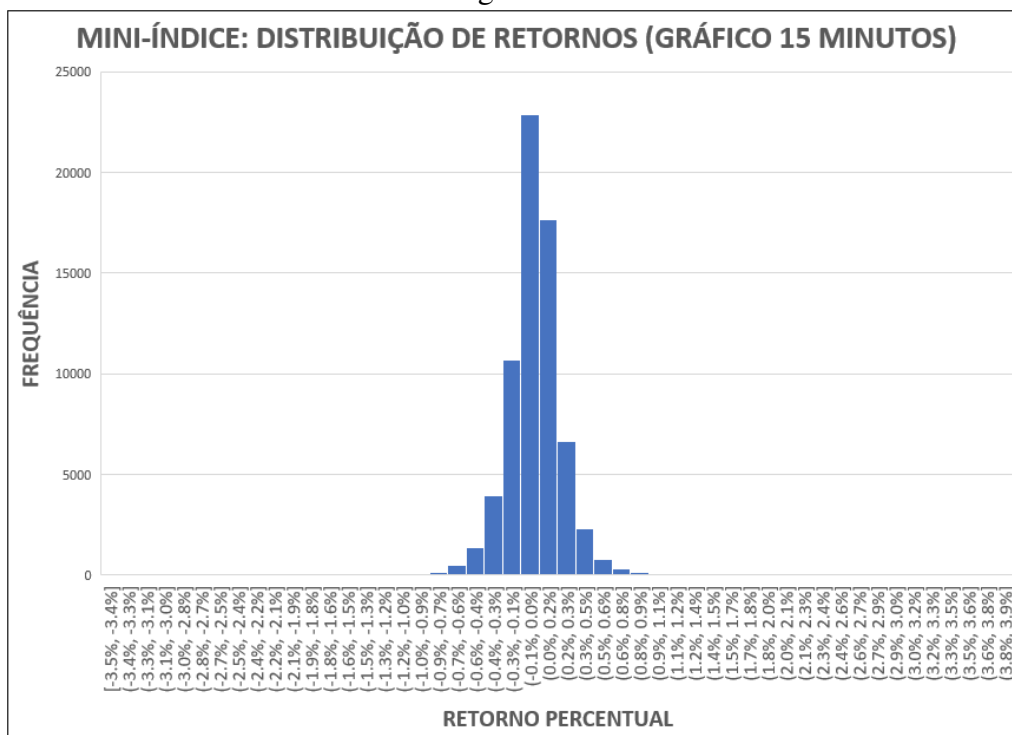
Na pesquisa de algo trading para mercado de capitais, o *dataset* corresponde a um ativo financeiro negociável. Poderíamos modelar ativos não-negociáveis (como o índice Bovespa, ou algum outro índice de referência), mas sua série teria a qualidade de ser sintética, ou seja, não derivada de negociações reais, o que teria uma importante perda de informação. Por esse motivo, damos preferência para ativos negociáveis (ações e derivativos - opções, contratos futuros), pois refletem a real dinâmica de oferta e demanda presente no mercado.

Escolhemos como objeto de nossa pesquisa o Contrato Futuro do Índice Bovespa - também chamado de “Ibovespa Futuro” ou “Índice Futuro”. Mais especificamente, o *dataset* é composto da série histórica de negociações do ativo “Mini-índice futuro”, que consiste na versão fracionada do “Ibovespa Futuro”, e se tornou o contrato futuro de índice com maior volume de negociações devido à sua baixa requisição de margem operacional.

O índice futuro foi escolhido por se tratar de um ativo com um grau de aleatoriedade importante, pois seu movimento é influenciado pelo movimento de cada ação que compõe o Ibovespa. Desse modo, não possui um viés próprio determinístico nem sazonal, nem está sujeito exclusivamente às oscilações de um único setor da economia brasileira. Este ativo possui uma distribuição com média próxima de zero. Será utilizado um dataset composto da série temporal do ativo, com periodicidade de 15 minutos (cada data point corresponde aos valores de abertura, valor máximo, valor mínimo e valor de fechamento deste período de tempo).

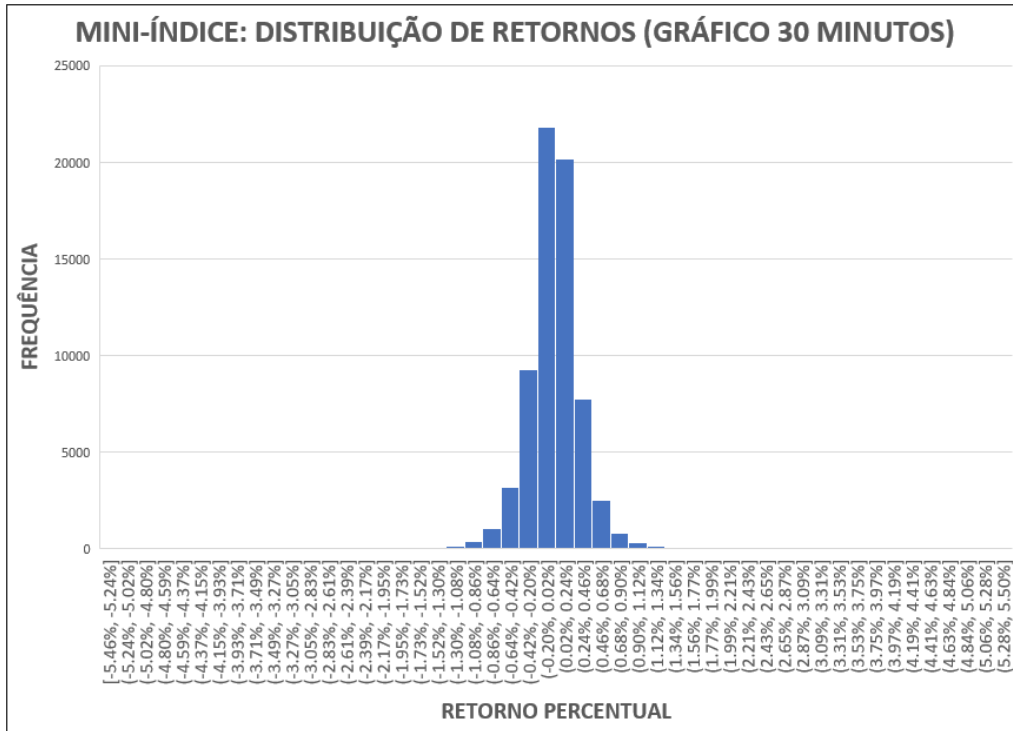
As Figuras 5.1, 5.2, 5.3 e 5.4 a seguir trazem um gráfico histograma com os retornos encontrados na série temporal do ativo. Podemos observar a ausência de skew (inclinação, que corresponderia a uma predominância de um tipo de resultado sobre outro) neste ativo, nos períodos de 15 minutos, 30 minutos, 60 minutos e 240 minutos.

Figura 5.1



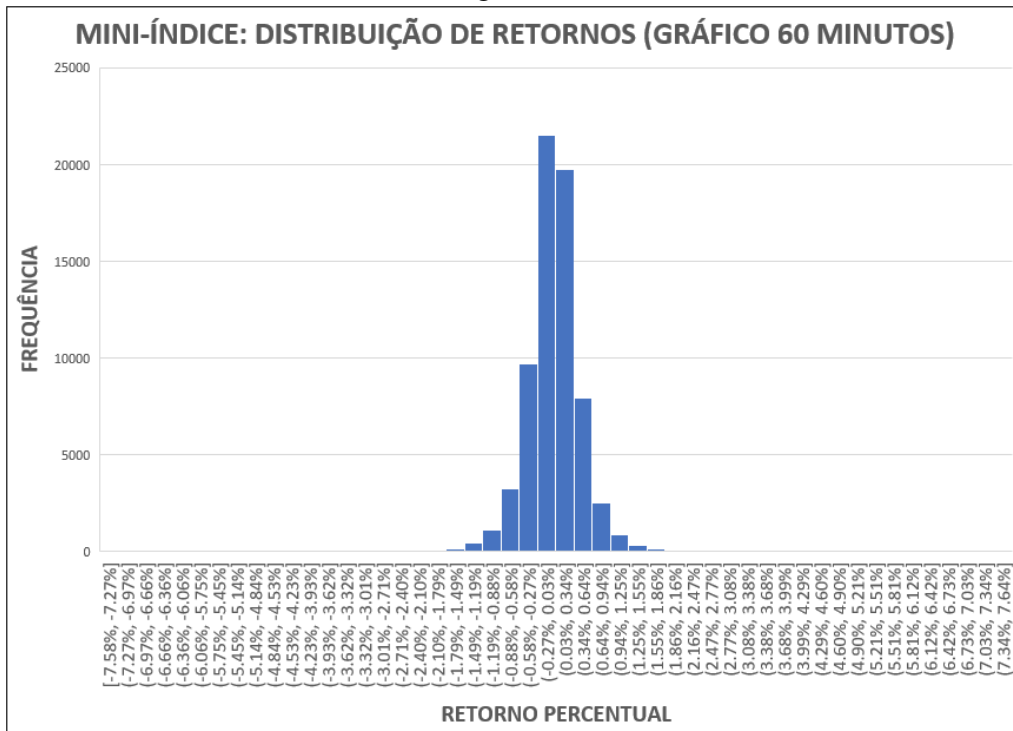
Fonte: elaborada pelo autor.

Figura 5.2



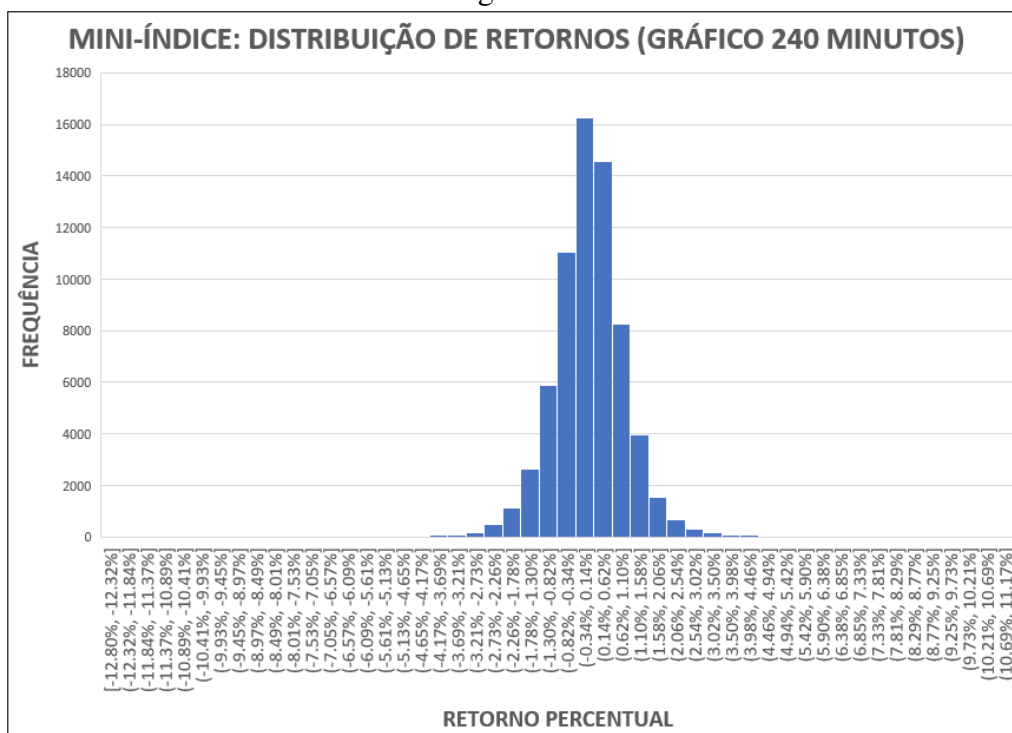
Fonte: elaborada pelo autor.

Figura 5.3



Fonte: elaborada pelo autor.

Figura 5.4



Fonte: elaborada pelo autor.

A distribuição de retornos para cada uma das janelas de tempo analisadas (15 minutos, 30 minutos, 60 minutos e 240 minutos) é expressa na Tabela 5.1. Essa tabela traz a média de retorno observada para cada periodicidade ou tempo gráfico, e também o desvio padrão encontrado em cada tempo gráfico.:

Tabela 5.1: Retorno do ativo por janelas de tempo.

Tempo gráfico	15 minutos	30 minutos	60 minutos	240 minutos
Média de retorno:	0,0005%	0,0002%	0,0001%	0,017%
Desvio padrão:	0,23%	0,33%	0,46%	1,05%

Fonte: Elaborada pelo autor.

Observamos que o retorno médio ficou em zero, com aumento no desvio padrão de acordo com o aumento da janela de tempo, o que seria esperado pela possibilidade do movimento *browniano* (estocástico) se afastar da média quanto maior o tempo transcorrido. Conforme observamos nas distribuições acima, o ativo não apresenta qualquer viés, altista ou baixista (para usar o jargão do mercado financeiro), para os períodos de tempo testados. Isso significa que não há oportunidade de ganhos pela aplicação simples de um

procedimento operacional (de compra, para lucrar com a alta do preço, ou de venda, para lucrar com a baixa). Para que houvesse tal oportunidade, a média deveria ser diferente de zero (e o quanto mais distante de zero, maior a oportunidade ou viés).

O conjunto de dados é composto das negociações realizadas entre 13 de fevereiro de 2014 a 20 de setembro de 2022, com a série histórica apresentada em intervalos de 15 minutos (tempo gráfico 15 minutos), com as informações de OHLC - Abertura, Máxima, Mínima e Fechamento de cada intervalo de 15 minutos. Essa periodicidade foi escolhida pelos seguintes motivos:

- Representa o conjunto de negociações intradiárias, ou seja, que ocorrem durante o pregão da Bolsa (B3);
- Negociação intradiária é amplamente utilizada pelos investidores, especialmente os de curto prazo (*traders*);
- Permite um grande número de *data points*, o que torna a pesquisa estatisticamente significativa;
- Ocorre um importante grau de aleatoriedade nesta periodicidade, ou seja, a média de retornos é próxima de zero, o que implica numa real qualidade dos modelos treinados que conseguirem obter retornos.

Este *dataset* foi obtido da plataforma Metatrader 5, disponibilizada por diversas corretoras de valores no Brasil. Mais especificamente, foi utilizada a base de dados fornecida pela corretora Terra Investimentos, a qual possuía uma série histórica de mini-índice mais extensa do que as demais corretoras. A Figura 5.5 ilustra a série temporal representada pelo *dataset*, ou seja, as negociações do mini-índice futuro agrupadas em informações OHLC (gráfico de *candles*) em intervalos de 15 minutos:

Figura 5.5: Gráfico representativo do ativo utilizado no *dataset*.



Fonte: Plataforma Metatrader 5.

5.2 Definição do algoritmo

A pesquisa com o uso de *features* pode ser feita com quaisquer algoritmos. Porém, nosso objetivo de pesquisa busca isolar completamente a variável “algoritmo”, nos experimentos, e observar exclusivamente o impacto das features nos retornos obtidos.

Por esse motivo, escolhemos um classificador considerado clássico em *machine learning*: o QDA (*Quadratic Discriminant Analysis*). Este algoritmo foi escolhido por prescindir de hiperparâmetros e manter-se estável nos experimentos. Por “manter-se estável” implicamos que ele não altera suas propriedades de um treinamento para o outro. (Poderíamos ter utilizado quaisquer algoritmos, mesmo que tivessem hiperparâmetros, desde que os hiperparâmetros não mudassem de uma investigação treino-teste para outra. Porém, essa abordagem implicaria na escolha (*a priori*) de hiperparâmetros para configurar o uso do algoritmo. E nessa escolha haveria um caráter de arbitrariedade por parte do pesquisador. Outro motivo pelo qual foi escolhido o algoritmo QDA deve-se ao seu uso na prática profissional do autor deste trabalho, com resultados positivos.)

Em nossa pesquisa, utilizamos a implementação do QDA fornecida pela biblioteca Scikit-Learn disponível na linguagem de programação Python, (https://scikit-learn.org/stable/modules/lda_qda.html), embora exortamos os leitores a escolher qualquer implementação e linguagem de sua preferência, e, também, quaisquer algoritmos que correspondam às qualidades observadas no Capítulo 4 - Metodologia.

Seria possível, também, realizar os mesmos testes com quaisquer outros algoritmos, mesmo sem corresponderem aos parâmetros observados no Capítulo 4 - Metodologia. Porém, alertamos para o fato de que algoritmos que permitem variação de hiperparâmetros entre um treinamento/teste e outro, conduzirão a alterações nos resultados que não se originam pelo uso de novas *features*, e, desse modo, o escopo da pesquisa ficaria comprometido.

5.3 Definição das *features*

As *features* ocupam o espaço central nesta pesquisa, desse modo, foram instanciados dois conjuntos diferentes de *features* nos experimentos: as *features* que são Indicadores Técnicos (termo que engloba um grande número de transformações baseadas em preço e volume do ativo negociado, e que são de amplo conhecimento dos praticantes do mercado), instanciadas a partir da biblioteca talib.py e neste trabalho chamadas de *Features TALIB* e, as *features* propostas pelo próprio autor, baseadas nos dados OHLC (preço de abertura do período, preço máximo do período, preço mínimo do período e preço de fechamento do período), que foram instanciadas na biblioteca salib.py e que neste trabalho serão chamadas de *Features SALIB*.

Ambos os conjuntos de *features* foram detalhados no Capítulo 2 (“Fundamentação Teórica”), o qual retrata as *features* e suas composições e fórmulas. Neste momento, vamos fazer referência às *features* e o modo como foram instanciadas nos experimentos. Os experimentos utilizaram inicialmente as *features SALIB*, depois as *features TALIB*, e, por fim, foram realizados experimentos compostos com ambos os tipos de *features*.

Features SALIB - Essas são as *features SALIB* instanciadas nos experimentos:

1. Hour - Foi utilizada para dar a referência de cada *datapoint*, visto que a série temporal era composta por dados de negociação agrupados em dados OHLC (Abertura, Máxima, Mínima e Fechamento) de 15 minutos. Ou seja, cada *datapoint* era composto dessas quatro informações de preço do intervalo de 15 minutos observado. É

a única *feature* que estará presente em todos os experimentos, pois constitui parte da estrutura das investigações.

2. Gap - Foi utilizado para dar a dimensão da diferença entre o último preço negociado no dia anterior (preço de fechamento) e o primeiro preço negociado no dia atual (preço de abertura).
3. CTO - A *feature* CTO (“Close-to-open”) retrata a variação percentual entre os preços de abertura e fechamento do período anterior (em dias). Foram utilizados “cto1” e “cto2”, ou seja, a variação percentual do dia anterior (cto1) e do dia prévio a ele (cto2), visto que numa série temporal o dia “atual” é chamado de 0 (zero), o dia de ontem é 1 (um) e o dia anterior a ontem (“anteontem”) é chamado de 2 (dois). A série temporal é contada da direita para a esquerda, ou, do presente para o passado.
4. Quadrant - Esta *feature* foi inserida nos experimentos para dar uma qualidade espaço-temporal aos datapoints individuais. Desse modo, é possível inserir o *datapoint* (preços OHLC) atual no contexto da movimentação do preço do ativo no espaço de um mês.
5. IBS - “*Internal Bar Strength*” - Foi utilizada para oferecer um “resumo” da movimentação do dia anterior, especialmente, relacionando a relação entre o preço de fechamento e a máxima e mínima do dia. O fechamento foi na parte superior da barra (candle) do dia anterior? Foi no meio? Foi na parte inferior? Essa informação nos permite inferir se no dia anterior a maior força no ativo foi “compradora” ou “vendedora”. Nos experimentos, utilizados “ibs1” e “ibs2”, indicando essa relação para o dia anterior (ontem) e dia prévio a ele (anteontem).
6. FHIGH - “From High” - Essa *feature* compara o preço de abertura do *datapoint* atual com o preço máximo do período configurado. Por exemplo, “fhigh4” computa todos os preços de “máxima” (o “H” de “OHLC”, sendo “H” = “High”, ou “máxima”) do período compreendido entre o *datapoint* atual e quatro dias atrás. Nos experimentos utilizamos as seguintes instanciações dessa *feature*: “fhigh0” (utiliza a máxima do dia atual), “fhigh4” (utiliza a máxima de uma janela de 4 dias), “fhigh10” (utiliza a máxima da janela de 10 dias) e “fhigh20” (utiliza a máxima da janela de 20 dias).
7. FLOW - “From Low” - Essa *feature* é correlata à anterior (“FHIGH”), mas compara o preço de abertura do *datapoint* atual com o preço mínimo do período configurado. Por exemplo, “flow4” computa todos os preços de “mínima” (o “L” de “OHLC”, sendo “L” = “Low”, ou “mínima”) do período compreendido entre o *datapoint* atual

e quatro dias atrás. Nos experimentos utilizamos as seguintes instâncias dessa *feature*: “flow0” (utiliza a mínima do dia atual), “flow4” (utiliza a mínima de uma janela de 4 dias), “flow10” (utiliza a mínima da janela de 10 dias) e “flow20” (utiliza a mínima da janela de 20 dias).

8. *DAILY POSITION* - Essa *feature* busca identificar a posição relativa do dia atual (sendo que o dia é composto de todas as negociações desde o momento da abertura do pregão até o horário específico do *datapoint*), em relação ao movimento do dia anterior (lembrando que ele poderá estar “acima”, “abaixo”, “dentro” ou “fora” dos limites do dia anterior). Nos experimentos, essa *feature* foi instanciada da seguinte forma: “daily_position0” (que trata da posição relativa do dia atual em comparação com o dia anterior, ou seja, “ontem”), “daily_position1” (que compara “ontem” com “anteontem”), e, “daily_position2” (que compara “anteontem” com o dia anterior).
9. *WEEKLY POSITION* - A mesma fórmula utilizada na *feature DAILY POSITION*, foi aplicada no contexto da movimentação semanal do ativo, gerando a *feature WEEKLY POSITION*, que, nos experimentos, foi instanciada da seguinte forma: “weekly_position0” (que trata da posição relativa da semana atual em comparação com a semana anterior), “weekly_position1” (que compara a semana anterior, com a semana prévia), e, “weekly_position2” (que duas semanas atrás, com a semana anterior a ela).

Features TALIB - As *features* da biblioteca talib.py foram instanciadas com os valores *default* encontrados na biblioteca. Embora fosse possível alterar os parâmetros de cada *feature* (Indicador Técnico), os experimentos foram realizados sem alteração desses parâmetros, porque são os parâmetros mais comumente encontrados. Essas são as *features TALIB* instanciadas nos experimentos:

1. IFR (RSI) - O indicador IFR (Índice de Força Relativa) é do tipo “oscilador”.
2. Estocástico - O indicador Estocástico também é da categoria dos “osciladores”.
3. MACD - É um indicador que busca identificar a tendência do mercado, pela proximidade de duas médias móveis.
4. ADX - (*Average Directional Index*) - É utilizado para indicar movimento direcional positivo e negativo. Em outras palavras, para identificar tendências.
5. CCI - (*Commodity Chanel Index*) - O indicador mede o nível atual de preços em relação a um nível médio, em uma determinada janela de tempo. Dessa forma, quanto mais distante da média, maior o seu valor.

6. MFI - (*Market Facilitation Index*) - O MFI é uma medida da facilidade ou da eficiência com a qual o preço de um ativo se move para cada unidade de volume. A direção do movimento dos preços não importa, apenas sua intensidade.
7. Williams % R (WILLR) - Este indicador é da categoria dos Osciladores, comparando o preço atual à *range* (diferença entre os preços máximo e mínimo do período).
8. TRIX - (*Triple Exponential Average*) é um indicador ao mesmo tempo da categoria dos “osciladores” e “*momentum*”.

5.4 Experimentos

O objetivo dos experimentos é identificar o aumento da capacidade de previsão de um modelo de *machine learning* treinado em uma base de dados que corresponde a uma série histórica de um ativo negociável em Bolsa, à medida que *features* diversas fossem inseridas no treinamento. As questões de investigação (tratadas no Capítulo 1) buscam identificar a contribuição das *features*, tanto em número, como em qualidade, além do impacto da combinação de *features* de origens diversas.

Sendo que os experimentos buscam resolver estas questões, sequência adequada dos experimentos nos pareceu ser da menor para a maior complexidade: do menor para o maior número de features, de combinações simples a combinações mais complexas, e assim por diante. Justamente como um espectro de testes diversos nos quais o impacto nos resultados pudesse variar de acordo com a complexidade do uso das *features*.

5.4.1 Descrição do Primeiro Experimento - *Benchmark e Baseline*

O primeiro experimento traz as formas clássicas de comparação para definir se uma estratégia (ou investigação) obteve resultados relevantes ou não. A forma de avaliar isso é comparando os resultados da estratégia (investigação, composta de treino e teste), com dois resultados-base: o *Benchmark* e o *Baseline*.

O *Benchmark* traz o retorno que teria sido obtido simplesmente comprando o ativo financeiro e mantendo essa posição comprada até o final do período avaliado (estratégia conhecida como “comprar-e-segurar”, ou, “*buy-and-hold*”). Ou seja, essa abordagem não impõe ao ativo negociado nenhuma regra de negociação. É, portanto, a medida básica

com a qual a inserção de regras de negociação (ou seja, os parâmetros de uma estratégia) são avaliados: inserir regras de negociação melhoram ou pioram o resultado original, sem nenhuma regra?

O segundo passo é estabelecer um *Baseline*. Qual a diferença entre *Benchmark* e *Baseline*? Enquanto o *Benchmark* demonstra o resultado comprando no início do período e vendendo no final, o *Baseline* demonstra o resultado comprando em todos os *datapoints* (momentos históricos do preço do ativo) e vendendo após o período de tempo estipulado no experimento.

Vamos dar um exemplo: o experimento foi configurado para negociar (ou não), com base no treinamento do algoritmo, a cada intervalo de 15 minutos, com tempo de negociação (*holding period*) de 2 *candles* ou 30 minutos. Então, a cada 15 minutos o modelo treinado escolhe se irá comprar, (com expectativa de ganho com a alta do preço), vender (com expectativa de ganho com a queda do preço) ou não realizar nenhuma negociação, pois não havia sinal definido pelo treinamento (“ficar de fora” do mercado). Essa é a configuração dos experimentos (investigações). Porém, o *Baseline* é configurado para “forçadamente” comprar (com expectativa de ganho com a alta do preço) a cada 15 minutos, sem exceção.

Ao compararmos o resultado de uma estratégia (ou investigação, treino-teste) com o *Benchmark*, identificamos se a estratégia teve resultado superior a um investimento passivo no ativo negociado. Ao compararmos o resultado com o *Baseline*, identificamos se a estratégia teve resultado superior a uma compra “forçada” a cada oportunidade de negociação. A teoria por trás dessa configuração de *Baseline* é a seguinte: observar se o *dataset* continha algum viés definido durante os períodos de negociação dos modelos treinados. Ou seja, se houvesse um claro viés “altista” (predominância de movimentos de alta do preço), ele seria identificado ou “capturado” pela estratégia *Baseline*, que “comprava” a cada *datapoint*. Por outro lado, se houvesse um claro viés “baixista” (predominância de movimentos de baixa do preço), a estratégia *Baseline* teria uma curva de capital sistematicamente negativa. Contrapor o resultado das estratégias dos experimentos ao *Baseline*, portanto, permitiria identificar se os experimentos produziram resultados superiores aos vieses possivelmente encontrados no *dataset*.

5.4.1.1 Descrição do Benchmark

O *Benchmark*, como resulta de uma suposta negociação de compra no início do período e o encerramento da operação (com uma venda) no final do período, pode ser

representado pela própria trajetória de preços do ativo. Ou seja, a curva de capital do *Benchmark* é exatamente igual à curva de capital do próprio ativo negociado.

5.4.1.2 Descrição do *Baseline*

Para esta pesquisa, utilizamos o dataset da série temporal do mini-índice futuro, na periodicidade de 15 minutos, onde cada datapoint corresponde aos valores de Abertura, Máxima, Mínima e Fechamento (OHLC) da janela de tempo de quinze minutos, transcorridos. O *baseline* consiste em operações *buy only* (apenas negociações de compra), realizadas a cada *datapoint*.

Foi escolhida apenas a *feature* “*hour*”, da biblioteca *SALIB*, pois a função dessa *feature* específica é apenas referenciar o *timestamp* (hora) do *datapoint*. Desse modo, ela tem impacto irrisório na qualidade da previsão.

O horizonte de previsão escolhido foi de 30 minutos, pois o intuito era analisar a viabilidade de criarmos um modelo de previsão para *daytrading* (modalidade de negociação na qual o investidor ou *trader* realiza múltiplas negociações no intervalo de um dia, ou pregão. Isso significa que uma negociação de “compra” era realizada a cada *datapoint* (intervalo de 15 minutos) e essa negociação era encerrada com uma “venda” no final de 30 minutos.

5.4.2 Descrição do Segundo Experimento - *Features SALIB*

O segundo experimento tratou das *features SALIB* - acrônimo de *Structural Analysis Library* - que foram criadas pelo autor para refletir relações entre os preços da série temporal do ativo, pela perspectiva de um observador (e praticante) do mercado de capitais.

Em outras palavras, essas *features* retratam o modo como um investidor profissional de curto prazo (*trader*) costuma olhar para o gráfico de preços, observando relações entre diferentes pontos da série temporal, para inferir a direção futura dos preços. As *features* utilizadas nesse experimento foram descritas no Capítulo 2 - Fundamentação Teórica, e acima, em 5.3. Definição das *features*.

O experimento consistiu em investigar o impacto das *features*, sendo que definimos como condições do experimento:

1. Definição da(s) *feature(s)* a serem investigadas.

2. Realização de um treinamento com as *features* selecionadas utilizando o algoritmo QDA em 80% do *dataset*.
3. Utilizar o modelo treinado em um teste com os restantes 20% do *dataset*.
4. O experimento simula negociações com avaliação a cada *datapoint* (intervalos de 15 minutos do preço, ou gráfico temporal de 15 minutos), com duração de 30 minutos (dois “*candles*” ou barras de informação OHLC). O algoritmo poderá escolher entre “compra” (para aproveitar um possível movimento de alta), “venda” (para aproveitar um possível movimento de baixa), ou “neutro” (abster-se de negociar naquele momento).

Como não estamos buscando criar uma estratégia completa, mas somente avaliar o impacto das *features*, omitimos as demais etapas de criação de estratégia, como: validação cruzada, inserção de custos operacionais, período de validação (pós-teste), etc. O objetivo foi observar isoladamente o efeito do uso crescente de *features* nos treinamentos do algoritmo. A separação do *dataset* em 80% treino e 20% teste ocorreu devido à prática utilizada no mercado financeiro de utilizar o máximo possível de dados para teste histórico (*backtest*) e um período razoável para o teste *forward*, ou *out-of-sample* (o suficiente para permitir obter um número de ocorrências mínimo para avaliação estatística).

Como o objetivo deste trabalho era investigar o uso de diferentes *features*, e não derivar uma estratégia finalizada, disponível para uso prático imediato, diversas etapas de desenvolvimento foram omitidas, devido às limitações do escopo do trabalho. De fato, para desenvolver uma estratégia completa, muitas outras etapas seriam necessárias, como, talvez, o uso de validação cruzada, a inserção de custos operacionais e imposto de renda, a determinação de alvos de lucro e limites de risco (como ordens *stop-loss*), etc. Um experimento completo de desenvolvimento de estratégia operacional poderá ficar para trabalhos futuros. O atual experimento foi estruturado da seguinte maneira:

1. Investigar (treino/teste) cada *feature SALIB* individualmente.
2. Investigar o uso crescente de *features*, com as diversas combinações possíveis das *features SALIB*. (Desde que respeitando a hierarquia crescente da instanciação das *features* que permitiam mais de uma instanciação, por exemplo: se a *feature X* foi instanciada com o *datapoint* “0” (ou seja, relativo ao próprio dia atual, ou ao *candle* atual), e depois ela foi instanciada com o *datapoint* “1”, (relativo ao anterior a “0”), a *feature X.1* não aparecerá sozinha em uma investigação (treino-teste), ou seja, ela deve estar acompanhada da *feature X.0*. Desse modo, *features* instanciadas

com valores maiores de *datapoints* ficaram sempre acompanhadas nos testes com as mesmas *features* instanciadas com os valores menores (precedentes) de *datapoints*. Outra observação: as *features FH - From High* sempre foram acompanhadas das suas contrapartes, as *features FL - From Low*, pois são consideradas complementares. Isso diminuiu o espectro das investigações, pois não foram realizadas todas as combinações “possíveis”, mas apenas as combinações que fizeram sentido com as restrições mencionadas acima.).

Desse modo, apesar de termos 9 (nove) *features* SALIB disponíveis, na prática elas são instanciadas num total de 40 derivações das *features*. A *feature “Hour”* é descartada da contagem, (pois, sendo estrutural, ela figura em todos os experimentos.) As demais *features* são instanciadas em seguindo as orientações do parágrafo 2, acima.

Apesar das 40 instanciações possíveis das 9 *features*, resultam apenas 16 conjuntos de *features* instanciadas. Isso ocorre por alguns motivos: *features* que somente são instanciadas em paralelo (nunca sozinhas) como *From High* e *From Low*, são colocadas no mesmo conjunto, *features* instanciadas com valores maiores nunca figuram sozinhas, mas sempre juntamente com os valores menores, em ordem crescente. A Tabela 5.2 ilustra a relação entre as *features*, suas instanciações e os conjuntos formados:

Tabela 5.2: Instanciações das *Features SALIB*.

FEATURE	CONJUNTO DE INSTANCIÇÃO
HOUR	HOUR
GAP	GAP
CTO	CTO.1
	CTO.1, CTO.2
QUADRANT	QUADRANT
IBS	IBS.1
	IBS.1, IBS.2
FHIGH / FLOW	FHIGH.0 / FLOW.0
	FHIGH.0 / FLOW.0, FHIGH.4 / FLOW.4
	FHIGH.0 / FLOW.0, FHIGH.4 / FLOW.4, FHIGH.10 / FLOW.10
	FHIGH.0 / FLOW.0, FHIGH.4 / FLOW.4, FHIGH.10 / FLOW.10, FHIGH.20 / FLOW.20
DAILY POSITION	DAILY_POSITION.0
	DAILY_POSITION.0, DAILY_POSITION.1
	DAILY_POSITION.0, DAILY_POSITION.1, DAILY_POSITION.2
WEEKLY POSITION	WEEKLY_POSITION.0
	WEEKLY_POSITION.0, WEEKLY_POSITION.1
	WEEKLY_POSITION.0, WEEKLY_POSITION.1, WEEKLY_POSITION.2

Fonte: Elaborada pelo autor.

Neste ponto, cabe uma explicação sobre a motivação das restrições impostas às possibilidades de combinação das *features SALIB*. Essas *features* foram criadas pelo autor com base em anos de atividade como Analista de investimentos no mercado financeiro, e, especialmente, com base na sua experiência de negociação do ativo financeiro objeto deste estudo, como também de outros ativos.

As restrições impostas às possibilidades de combinações tem a ver com a forma como um negociador no mercado financeiro observa o gráfico da série temporal do ativo. Por exemplo, toda negociação de *day trading* é realizada no “momento presente”, durante o pregão, ou, no que chamamos de dia “0” (zero) na série temporal. Desse modo, o dia anterior (“ontem”), seria o dia “1”, e o anterior a ele seria o dia “2”, e assim por diante, dado que a série temporal é numerada “da direita para a esquerda”, a partir do ponto mais recente (ou “hoje”).

Dessa forma, seguindo uma prática comum no mercado financeiro, o autor não via a relevância de uma *feature* ser instanciada 20 dias atrás (como o exemplo da *From High*), sem que fosse instanciada também no dia “0”, ou atual. Na prática de negociação,

o autor desconhece investidores que negociem no momento presente olhando apenas as informações de 20 dias atrás. . . Essas informações de 20 dias atrás podem estar presentes para gerar um contexto de análise, mas não são suficientes para motivar, sozinhas, uma negociação no ativo. Assim, o autor optou por somente instanciar *features* com valores maiores (portanto mais distantes do ponto da negociação, na série temporal) com as suas contrapartes de valores menores. (Por exemplo, *feature.20* tem de estar acompanhada dos outros valores instanciados, como *.10*, *.4* e *.0*).

Outra restrição foi a decisão de somente instanciar as *features From High* e *From Low* em conjunto. Isso ocorreu porque, na prática de negociação financeira, um investidor (“trader”), caso observe a relação do preço atual com as máximas de dias anteriores, também analisa a relação do preço atual com as mínimas de dias anteriores. Por esse motivo, essas duas *features* foram instanciadas conjuntamente.

Este experimento resultou em 2.879 investigações (treino/teste), com combinações diferentes das *features SALIB*. As combinações utilizadas estão no Anexo A, ([Link para Anexo A](#)), ítem 1 (SEGUNDO EXPERIMENTO - *FEATURES SALIB*), para consulta.

5.4.3 Descrição do Terceiro Experimento - *Features TALIB*

O terceiro experimento tratou das *features TALIB* - que foram compiladas pela comunidade *open source* na biblioteca *python talib.py*, e cujo significado é “*Technical Analysis Library*”, ou biblioteca de indicadores de Análise Técnica, uma forma de análise praticada amplamente no mercado de capitais. O experimento foi estruturado da seguinte maneira:

1. Investigar (treino/teste) cada *feature TALIB* individualmente,
2. Investigar o uso crescente de *features*, com as diversas combinações possíveis das *features TALIB*.

Este experimento resultou em 255 investigações (treino/teste), com combinações diferentes das *features TALIB*. As combinações utilizadas estão no Anexo A, ([Link para Anexo A](#)) ítem 2 (TERCEIRO EXPERIMENTO - *FEATURES TALIB*), para consulta.

5.4.4 Descrição do Quarto Experimento - *Features SALIB e Features TALIB*

Neste experimento foram utilizados ambos os conjuntos de *features* trabalhados nos experimentos anteriores: as *features SALIB* e as *TALIB*, o propósito dessa composição foi observar se a sinergia entre os dois métodos de criação de *features* especializadas promoveria resultados superiores. Este experimento foi estruturado da seguinte maneira:

- Escolhidas as 16 melhores combinações de *features SALIB*, tendo como critério a relação entre o ratio (fator de lucro) do período de teste versus o ratio do período de treinamento).
- Com cada uma dessas combinações de *features SALIB*, foram criadas combinações com as diferentes *features TALIB*.

Este experimento resultou em 4.080 investigações (treino/teste), com combinações diferentes das *features SALIB* e *TALIB*. As combinações utilizadas estão no Anexo A, ([Link para Anexo A](#)) ítem 3, para consulta.

5.5 Resultados dos Experimentos e sua Avaliação

Nesta etapa, analisamos os resultados obtidos em cada um dos experimentos acima, começando pelo *Baseline*, seguido do experimento que utilizou somente *features SALIB*, depois o experimento com as *features TALIB*, e, por fim, o experimento com o uso dos dois tipos de *features SALIB + TALIB*).

A avaliação dos resultados é feita com a partição final de 20% do *dataset*, que corresponde ao período utilizado para Teste, nos experimentos. (O período utilizado para treino não é passível de comparação e avaliação, pois o algoritmo ainda estava sendo treinado, em vias de se tornar um modelo finalizado).

5.5.1 Resultados do Primeiro experimento - *Benchmark e Baseline*

5.5.1.1 Resultados do Benchmark

Os resultados do *Bechnmark* são os resultados da própria série temporal do ativo negociado. Neste caso, a trajetória dos preços do mini-índice futuro. Para obtermos o resultado do *Benchmark*, estipulamos uma operação de “compra” no dia 18 de janeiro de

2021 e o encerramento desta posição comprada com uma negociação de “venda”, no dia 20 de setembro de 2022.

A Figura 5.6 ilustra a série temporal do mini-índice neste período, como um gráfico de Tempo \times Preço. Neste gráfico, o período de avaliação (*Benchmark*) é o movimento de preço realizado no lapso temporal entre os pontos de compra e de venda, identificados em vermelho.

Figura 5.6



Fonte: Plataforma Metatrader 5.

O resultado obtido pelo *Benchmark (buy-and-hold)* foi a diferença percentual entre o preço do final do *dataset* e o preço do início, ou seja, o preço de fechamento do dia 22/09/2022 menos o preço de abertura do dia 18/01/2021. A Figura 5.7 contém dados OHLC do *datapoint* em que foi realizada a negociação simulada de “compra”, e a Figura 5.8 contém dados OHLC do *datapoint* em que foi realizada a negociação simulada de “venda”.

Figura 5.7: Gráfico representativo do *Benchmark* Dados de preço do início do período do *Benchmark*.

Janela de dados	
WIN\$D,Daily	
Date	2021.01.18
Open	140400
High	142450
Low	139630
Close	140935

Fonte: Plataforma Metatrader 5.

Figura 5.8: Dados de preço do fim do período do *Benchmark*.

Janela de dados	
WIN\$D,Daily	
Date	2022.09.20
Open	118905
High	119675
Low	118170
Close	119415

Fonte: Plataforma Metatrader 5.

Desse modo, o resultado do *Benchmark* foi -14,95% (negativos quatorze vírgula noventa e cinco por cento), derivado da fórmula de retorno percentual:

$$\frac{\text{preço final} - \text{preço inicial}}{\text{preço inicial}}$$

5.5.1.2 Resultados do *Baseline*

A avaliação do experimento *Baseline* é bastante direta, pois consiste apenas dos resultados de um único teste. Este teste, conforme descrito acima realizou negociações a cada intervalo de 15 minutos, mantendo a posição durante dois *candles* (30 minutos). Todas as negociações foram de “compra” (expectativa de lucrar com a alta). O resultado

foi obtido com os 20% finais do *dataset*, para ser comparável com os resultados de teste (e não de treino) dos experimentos seguintes. Os resultados foram:

Buy Only (Baseline) resultado em percentual (%):

Retorno (*Total return*): -19.6518%

Negociações (*Entries*): 15475

Retorno médio percentual (*Return per entry*): -0.0013%

Acurácia (*Success*): 50.33%

Fator de lucro (*Ratio*): 0.99

Número de pregões (*Total of days*): 416

Número de meses (*Total of months*): 19.8095

Retorno mensal percentual médio (*Monthly return*): -0.992%

Prejuízo cumulativo percentual máximo (*Max drawdown*): -74.8232%

Prejuízo cumulativo percentual médio (*Mean drawdown*): -32.374%

Sinais de compra (*Buy signals*): 15475

Sinais de venda (*Sell signals*): 0

Buy Only (Baseline) resultado em pontos de mini-índice:

Retorno (*Total return*): -28250.0

Negociações (*Entries*): 15475

Retorno médio em pontos (*Return per entry*): -1.8255

Acurácia (*Success*): 50.33 %

Fator de lucro (*Ratio*): 0.98

Número de pregões (*Total of days*): 416

Número de meses (*Total of months*): 19.8095

Retorno mensal médio em pontos (*Monthly return*): -1426.0817

Prejuízo cumulativo máximo em pontos (*Max drawdown*): -88285.0

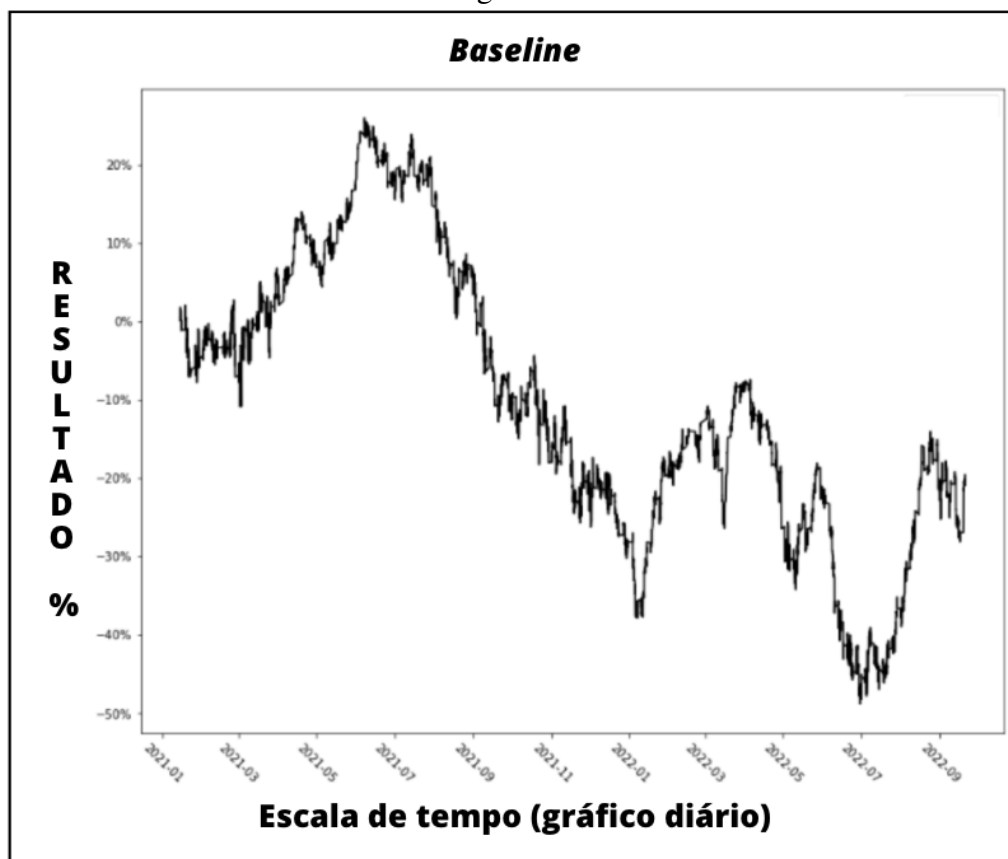
Prejuízo cumulativo médio em pontos (*Mean drawdown*): -39778.9764

Sinais de compra (*Buy signals*): 15475

Sinais de venda (*Sell signals*): 0

Como podemos observar, o *baseline* resultou negativo, o que não é de surpreender, dada a natureza aleatória do ativo negociado. Esse comportamento estocástico, aleatório foi demonstrado no início deste Capítulo 5, quando analisamos o *dataset* (5.1 “Definição do dataset”). A Figura 5.9 mostra a curva de capital do *Baseline* no período de teste:

Figura 5.9



Fonte: Elaborada pelo autor.

5.5.2 Resultados do Segundo Experimento - *Features SALIB*

A Tabela 5.3 mostra os resultados por números de *features* nas investigações. Lembrando que o termo “investigações” foi designado para identificar uma “passagem” completa do algoritmo *QDA* pela base de dados, utilizando um conjunto específico de *features*, compreendendo tanto o treinamento do algoritmo, como o teste do modelo treinado. Os próximos parágrafos são utilizados para descrever as métricas dessa tabela.

A primeira coluna indica o número de *features* utilizados nas investigações. Nesse caso, uma *feature* que tenha sido instanciada mais de uma vez foi tratada como diversas *features* diferentes. (Por exemplo, a *feature IBS*, que foi instanciada como *ibs.1* e *ibs.2*, contará como duas *features*, e não apenas uma). Todos os resultados da mesma linha derivam de investigações com o número específico de *features* indicado nesta coluna.

A segunda coluna mostra o número de combinações feitas com as *features* disponíveis na primeira coluna. Por exemplo, temos oito *features SALIB*, porém apenas seis

foram utilizadas individualmente (as duas restantes, “*fhigh*” e “*flow*” são utilizadas apenas em conjunto, e não individualmente), por esse motivo, temos seis combinações para “uma *feature*” utilizada. Ao aumentarmos o número de *features* utilizadas nas investigações, o número de combinações aumenta. Até diminuírem a partir de um ponto, resultando em apenas uma combinação com todas as vinte *features* disponíveis.

A terceira coluna mostra uma métrica criada pelo autor, intitulada “Grau de Consistência Teste/Treino”. Essa métrica foi desenhada para identificar a capacidade de generalização do algoritmo treinado com o conjunto específico de *features*. Essa métrica é calculada do seguinte modo: $ratio\ teste / ratio\ treino$. Das diversas métricas apresentadas na planilha de resultados (link), a métrica “*ratio*” é calculada dividindo o lucro médio (retorno positivo médio) pelo prejuízo médio (retorno negativo médio). Essa métrica (também conhecida no mercado financeiro como Fator de Lucro, ou *Profit Factor*), indica a expectativa de retorno da estratégia: se o número for igual a “1”, ela tem expectativa de retorno igual a zero (pois para cada real investido, retornou um real ganho), se o número for negativo, a expectativa é de perda ou retorno negativo (pois para cada real investido, o resultado foi inferior a um real, e números acima de “1” indicam expectativa de retorno positivo (pois cada real investido teve como retorno um valor maior do que um real). O *ratio* (fator de lucro) “normaliza” a informação de retorno, de modo a termos um índice que não depende do número de ocorrências (n). O “Grau de Consistência Teste/Treino” é, portanto, o resultado da divisão entre o fator de lucro (*ratio*) encontrado no teste pelo encontrado no treino. Se o número for igual a “1”, temos que o *ratio* (fator de lucro) foi igual no treino e no teste. Se o número for menor que “1”, significa que o teste teve uma performance inferior ao treino, e, portanto, o algoritmo treinado com o conjunto de *features* específico não generalizou bem. Se o número for superior a “1”, indica que o teste teve um fator de lucro superior ao do treino, o que indica uma boa generalização do algoritmo.

A quarta coluna mostra o número médio de trades (ou negociações) realizados no treino do algoritmo, lembrando que os treinos foram realizados com 80% do *dataset*.

A quinta coluna mostra o número médio de trades (ou negociações) realizados no teste (utilizando o modelo treinado), sendo que os testes foram realizados utilizando os 20% restantes do *dataset*.

A sexta coluna indica o Retorno percentual médio para os resultados dos treinos do algoritmo. O número de treinos específicos realizados corresponde ao número de combinações de *features* possíveis, indicados na segunda coluna. (Por exemplo, para 1

(uma) *feature*, foram possíveis 6 (seis) combinações, portanto, houveram 6 (seis) treinos do algoritmo (e também 6 (seis) testes). Dessa forma, o Retorno médio percentual indicado nesta sexta coluna corresponde à média de retorno percentual dos “n” treinamentos possíveis, indicados pelo número de combinações da segunda coluna.

A sétima coluna representa o Retorno percentual médio para os resultados dos testes do algoritmo, ou seja, para os resultados dos modelos já treinados (no período de treino) e agora testados no remanescente 20% do *dataset*.

Tabela 5.3: Resultados das *Features SALIB* (por número de *features* nas investigações).

Resultado das <i>Features SALIB</i>						
Nº Features SALIB	Nº Combinações	Grau de Consistência Teste/Treino	Nº médio de trades (treino)	Nº médio de trades (teste)	Retorno médio % (treino)	Retorno médio % (teste)
1	6	0,94	23780	5661	173,1	-5,32
2	20	0,94	22441	5282	187	-1,76
3	48	0,92	21680	5038	207,5	-10,27
4	91	0,91	20974	4788	231,5	-10,18
5	144	0,91	20274	4522	254,5	-5,28
6	197	0,91	19485	4217	278,5	-0,23
7	240	0,9	18568	3854	298,8	2,57
8	268	0,9	17553	3444	317,1	5,14
9	282	0,9	16517	3023	336,9	9,94
10	286	0,89	15589	2643	362,1	12,26
11	282	0,88	14876	2353	390,9	11,3
12	268	0,86	14404	2168	416,6	10,11
13	240	0,85	14130	2070	439,2	9,88
14	197	0,85	13982	2026	460,1	10,86
15	144	0,85	13899	2011	487	13,18
16	91	0,85	13847	2008	494,2	16,91
17	48	0,86	13810	2008	506,6	19,93
18	20	0,86	13783	2005	516,1	22,96
19	6	0,88	13775	1992	532,9	28,65
20	1	0,9	13727	1952	527,4	35,62

Fonte: Elaborada pelo autor.

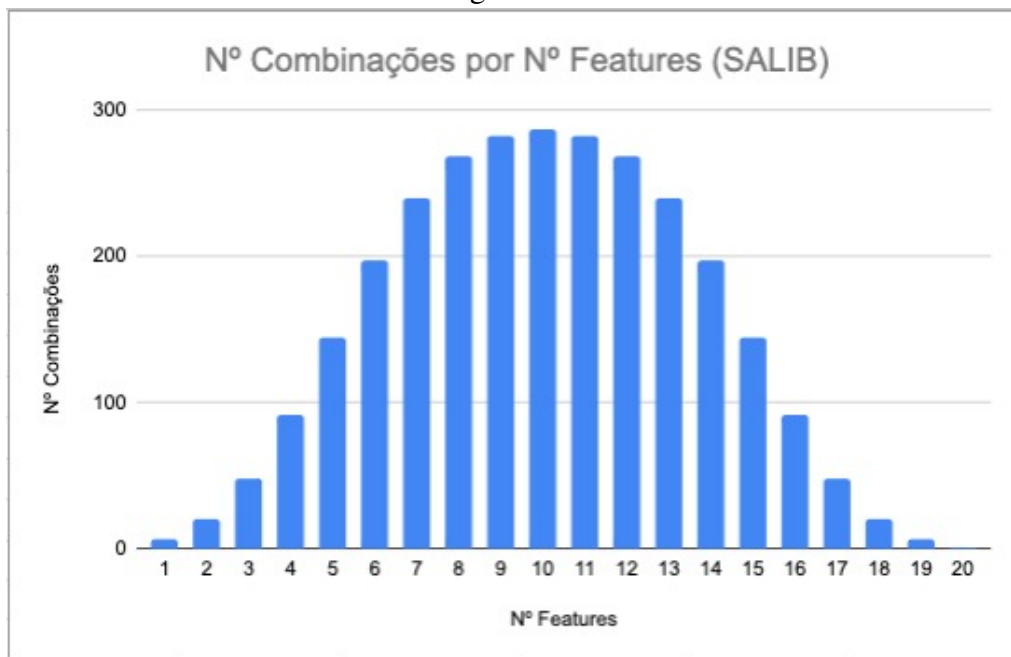
Dessa Tabela 5.3 inferimos o seguinte:

1. O número de combinações possíveis com “n” *features* aumenta à medida que aumentam as *features* disponíveis, mas chegam num ponto máximo ao redor de dez *features*, e a partir daí caem até apenas uma combinação, com a totalidade das 20 *features* disponíveis. (Essa informação foi isolada na Figura 5.10, para demonstrar ascensão e queda do número de combinações).
2. Em média, a métrica “Grau de Consistência Teste/Treino” apresentou valores decrescentes à medida que aumentaram o número de *features* utilizadas nas investi-

gações. Isso pode indicar que a presença de mais *features* pode levar à ocorrência de *overfitting*, dado que essa métrica busca identificar o poder de generalização do modelo treinado. (Porém, como se tratam de valores médios, não se descarta a possibilidade de bons resultados individuais).

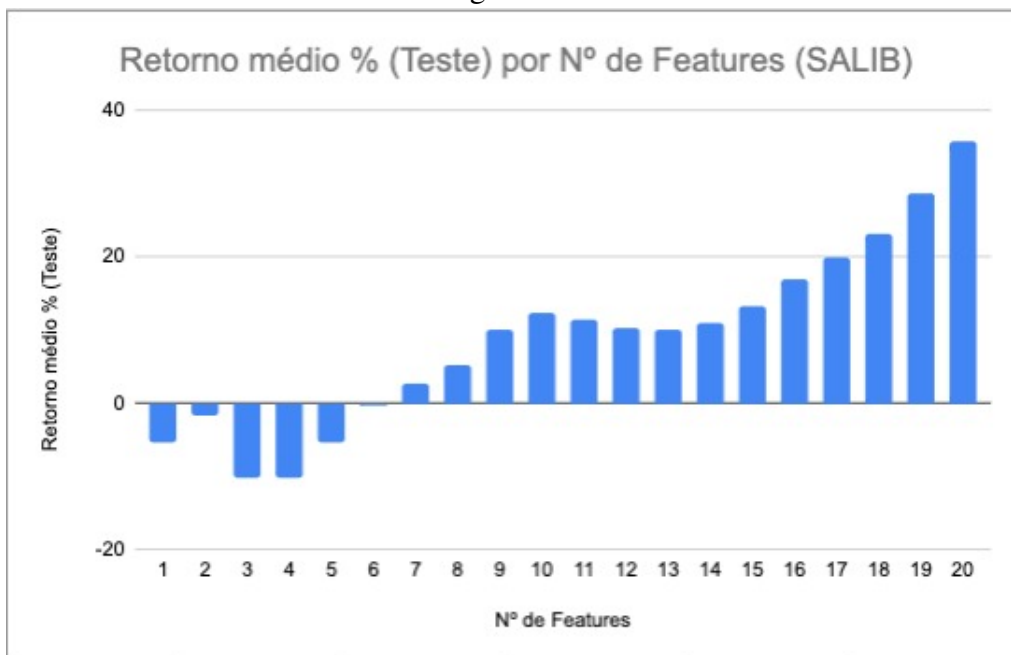
3. O número médio de “trades” (negociações simuladas) diminuiu à medida em que se aumentou o número de *features* presentes nas investigações. Isso ocorre porque o algoritmo foi treinado para identificar as melhores oportunidades de compra ou de venda (operação de venda a descoberto) no *dataset*, ou seja, valores acima de 66% de acurácia para identificar oportunidades de compra, e abaixo de 33% de acurácia para identificar oportunidades de venda. Desse modo, com o aumento do número de *features*, progressivamente tornou-se mais difícil encontrar ocasiões no *dataset* em que as *features* estivessem com os parâmetros aprendidos como adequados no período de treino. Essa diminuição do número de trades à medida que aumentaram o número de *features* ocorreu tanto nos treinos quanto nos testes.
4. O Retorno percentual médio dos treinos aumentou à medida em que aumentaram o número de *features* utilizadas no experimento.
5. O Retorno percentual médio dos testes iniciou com valores negativos nas investigações entre 1 e 6 *features*, indicando a presença de *underfitting* e incapacidade de generalização do modelo treinado. A partir dos testes com 7 *features*, o Retorno percentual médio ficou positivo e seu valor subiu atingindo seu limite máximo em paralelo com o limite máximo de *features*. (Essa informação foi isolada na Figura 5.11, para melhor visualização do aumento do resultado à medida que se acrescentam novas *features*).

Figura 5.10



Fonte: Elaborada pelo autor.

Figura 5.11



Fonte: Elaborada pelo autor.

Tabela 5.4: *Feature Importance SALIB*.

Relevância das <i>Features</i> SALIB						
Teste das <i>features</i> SALIB mais representativas	Nº Ocorrências	Grau de Consistência Teste/Treino	Nº médio de trades (treino)	Nº médio de trades (teste)	Retorno médio % (treino)	Retorno médio % (teste)
gap	1440	0,9	15253	2624	365,9	13,35
No gap	1439	0,87	17448	3294	365,4	1,56
cto	1920	0,88	16265	2889	389,1	7,18
No cto	959	0,9	16523	3099	318,8	8,02
quadrant	1440	0,88	16354	2960	365,6	7,44
No quadrant	1439	0,88	16347	2958	365,8	7,48
ibs	1920	0,88	16512	2992	378,2	6,25
No ibs	959	0,9	16027	2893	340,6	9,89
fhigh_flow	2304	0,88	15148	2480	382,5	11,55
No figh_flow	575	0,89	21170	4877	298,3	-8,93
daily_position	2160	0,88	16417	2973	376,4	6,8
No daily_position	719	0,9	16150	2917	333,3	9,44
weekly_position	2160	0,88	16449	3008	376	8,42
No weekly_position	719	0,89	16053	2813	334,5	4,58

Fonte: Elaborada pelo autor.

A Tabela 5.4 busca identificar o grau de importância de *features* específicas nos resultados das investigações. Para tanto, foi obtida a média de resultados dos treinos e testes em que a *feature* esteve presente, e comparados com a média de resultados sem a presença da *feature*. A diferença entre os resultados evidencia a relevância da *feature* em comparação com as demais:

1. *Gap* - O Retorno médio percentual nos testes foi 13,35% com a *feature*, e 1,5% sem a *feature*. Desse modo, identificamos que é uma *feature* relevante.
2. *CTO* - O Retorno médio percentual nos testes foi 7,18% com a *feature*, e 8,02% sem a *feature*. Desse modo, identificamos que é uma *feature* que pode ter contribuído no quadro geral dos resultados, mas sua ausência não diminuiu os resultados.
3. *Quadrant* - O Retorno médio percentual nos testes foi 7,44% com a *feature*, e 7,48% sem a *feature*. Desse modo, identificamos que é uma *feature* que pode ter contribuído no quadro geral dos resultados, mas sua ausência não diminuiu os resultados.
4. *IBS* - O Retorno médio percentual nos testes foi 6,25% com a *feature*, e 9,89% sem a *feature*. Desse modo, identificamos que é uma *feature* que pode ter contribuído no quadro geral dos resultados, mas sua ausência não diminuiu os resultados.
5. *Fhigh_Flow* - O Retorno médio percentual nos testes foi 11,55% com a *feature*, e -8,93% sem a *feature*. Desse modo, identificamos que é uma *feature* relevante.
6. *Daily_position* - O Retorno médio percentual nos testes foi 6,8% com a *feature*,

e 9,44% sem a *feature*. Desse modo, identificamos que é uma *feature* que pode ter contribuído no quadro geral dos resultados, mas sua ausência não diminuiu os resultados.

7. *Weekly_position* - O Retorno médio percentual nos testes foi 8,42% com a *feature*, e 4,58% sem a *feature*. Desse modo, identificamos que é uma *feature* relevante.

Tabela 5.5: Melhores *Features SALIB*. Critério 1.

Ranking 10 melhores combinações Features SALIB por: Grau de Consistência Teste/Treino					
Features (Combinação)	Grau de Consistência Teste/Treino	Nº de trades (treino)	Nº de trades (teste)	Retorno % (treino)	Retorno % (teste)
quadrant fhigh0 flow0 weekly_position0 weekly_position1	1,07	21132	4917	134,77	72,10
fhigh0 flow0 weekly_position0 weekly_position1	1,07	21132	4917	134,77	72,10
gap cto1 quadrant lbs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,05	16988	3481	261,61	66,19
gap cto1 lbs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,05	16988	3481	261,61	66,19
quadrant fhigh0 flow0	1,04	22861	5050	71,64	37,94
gap quadrant fhigh0 flow0	1,04	14347	2688	50,14	19,22
fhigh0 flow0	1,04	22861	5050	71,64	37,94
gap fhigh0 flow0	1,04	14347	2688	50,14	19,22
gap quadrant fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2	1,03	16794	3695	149,28	42,08
quadrant weekly_position0 weekly_position1 weekly_position2	1,03	23739	6075	170,68	64,25

Fonte: Elaborada pelo autor.

A Tabela 5.5 mostra as dez melhores combinações de *features SALIB*, utilizando como critério o Grau de Consistência Teste/Treino, que foi descrito acima como uma métrica desenvolvida pelo autor para identificar o poder de generalização do modelo treinado com o conjunto específico de *features*. Significa que essas dez combinações de *features* proporcionaram os melhores resultados de testes comparados aos resultados de treinamento do modelo.

Tabela 5.6: Melhores *Features SALIB*. Critério 2.

Ranking 10 melhores combinações de Features SALIB por: Retorno nos Testes					
Features (Combinação)	Grau de Consistência Teste/Treino	Nº de trades (treino)	Nº de trades (teste)	Retorno % (treino)	Retorno % (teste)
quadrant daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2	1,01	23878	6108	350,00	90,00
daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2	1,01	23878	6108	350,00	90,00
quadrant fhigh0 flow0 weekly_position0 weekly_position1	1,07	21132	4917	135,00	72,00
fhigh0 flow0 weekly_position0 weekly_position1	1,07	21132	4917	135,00	72,00
gap cto1 quadrant lbs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,05	16988	3481	262,00	66,00
gap cto1 lbs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,05	16988	3481	262,00	66,00
gap cto1 cto2 quadrant lbs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,03	17046	3237	351,00	66,00
gap cto1 cto2 lbs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,03	17046	3237	351,00	66,00
quadrant weekly_position0 weekly_position1 weekly_position2	1,03	23739	6075	171,00	64,00
weekly_position0 weekly_position1 weekly_position2	1,03	23739	6075	171,00	64,00

Fonte: Elaborada pelo autor.

A Tabela 5.6 mostra as dez melhores combinações de *features SALIB*, utilizando como critério o Retorno Percentual nos Testes, que mostra o resultado obtido no período pós-treino (teste), do modelo treinado com o conjunto específico de *features*. Significa

que essas dez combinações de *features* proporcionaram os melhores resultados entre todas as combinações que foram feitas.

5.5.3 Resultados do Terceiro Experimento - *Features TALIB*

Tabela 5.7: Resultados das *Features TALIB* (por número de *features* nas investigações).

Resultado das <i>Features TALIB</i>						
Nº Features TALIB	Nº Combinações	Grau de Consistência Teste/Treino	Nº médio de trades (treino)	Nº médio de trades (teste)	Retorno médio % (treino)	Retorno médio % (teste)
1	8	0,94	1279	513	36,6	1,43
2	28	10,08	2787	1072	73,2	4,6
3	56	0,94	4447	1682	102,6	8,31
4	70	0,92	5683	2191	134,4	10,53
5	56	0,9	6878	2631	166,6	12,36
6	28	0,9	7709	3142	199,6	17,25
7	8	0,9	8393	3422	226,6	21,93
8	1	0,89	8778	3690	262,8	26,37
2 (Sem outlier)	27	0,84	2886	1111	75,9	4,68

Fonte: Elaborada pelo autor.

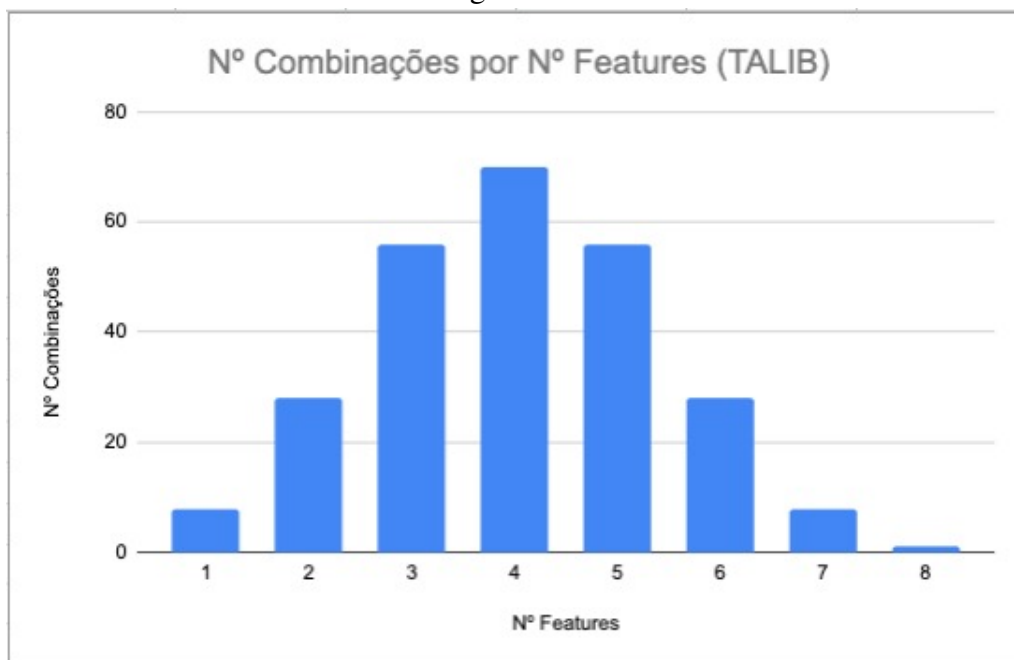
Dessa Tabela 5.7 inferimos o seguinte:

1. O número de combinações possíveis com “n” *features* aumenta à medida que aumentam as *features* disponíveis, mas chegam num ponto máximo ao redor de quatro *features*, e a partir daí caem até apenas uma combinação, com a totalidade das 8 *features* disponíveis. (Essa informação foi isolada na Figura 5.12, para demonstrar ascensão e queda do número de combinações).
2. Em média, a métrica “Grau de Consistência Teste/Treino” apresentou valores decrescentes à medida que aumentaram o número de *features* utilizadas nas investigações. Isso pode indicar que a presença de mais *features* pode levar à ocorrência de *overfitting*, dado que essa métrica busca identificar o poder de generalização do modelo treinado. (Porém, como se tratam de valores médios, não se descarta a possibilidade de bons resultados individuais). Houve a presença de um *outlier* com o uso de duas *features*, com um teste em que houveram poucas ocorrências de negociações (apenas 13) e isoladas numa possível máxima local, de modo que o ratio (fator de lucro) ficou muito maior no teste do que no treino. Isso levou a um valor de 10,08 na métrica “Grau de Consistência Teste/Treino”, totalmente destacado em

relação aos demais valores da mesma métrica. Por esse motivo, refizemos este estudo excluindo o *outlier*, e apresentamos os resultados na última linha, (em cinza). Nesse caso, o valor da métrica “Grau de Consistência Teste/Treino” ficou em 0,84, ou seja, alinhada com os demais valores.

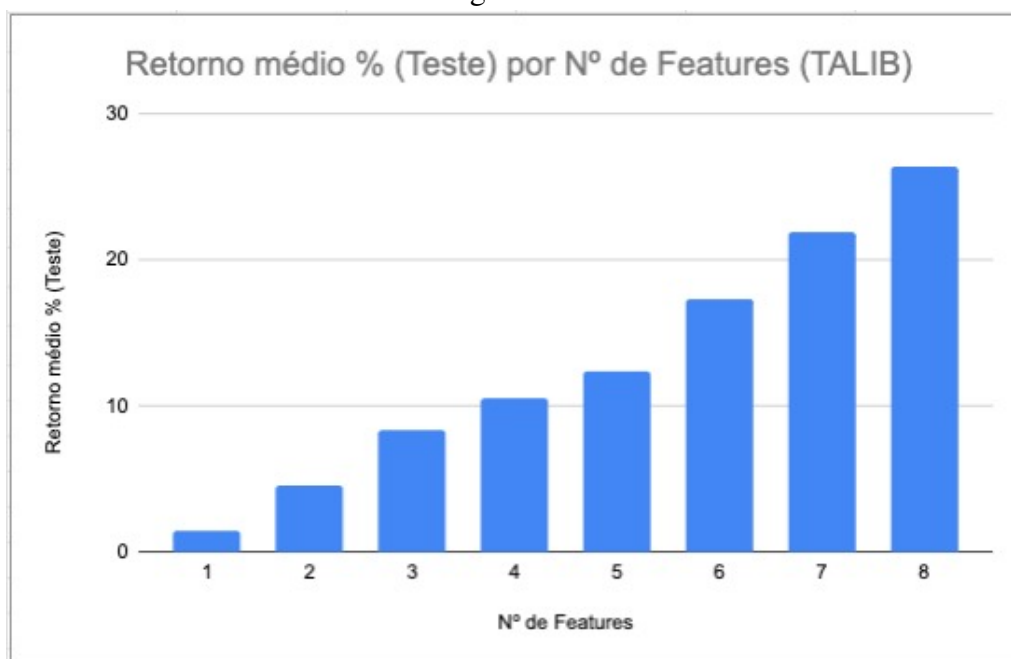
3. Ao contrário do que ocorreu com as *features* *SALIB*, no caso das *features* *TALIB* o número médio de “trades” (negociações simuladas) aumentou à medida em que se aumentou o número de *features* presentes nas investigações.
4. O Retorno percentual médio dos treinos aumentou à medida em que aumentaram o número de *features* utilizadas no experimento.
5. O Retorno percentual médio dos testes aumentou à medida em que aumentaram o número de *features* utilizadas no experimento. (Essa informação foi isolada na Figura 5.13, para melhor visualização do aumento do resultado à media que se acrescentam novas *features*).

Figura 5.12



Fonte: Elaborada pelo autor.

Figura 5.13



Fonte: Elaborada pelo autor.

Tabela 5.8: *Feature Importance TALIB.*

Relevância das <i>Features</i> TALIB						
Teste das features TALIB mais representativas	N° Ocorrências	Grau de Consistência Teste/Treino	N° médio de trades (treino)	N° médio de trades (teste)	Retorno médio % (treino)	Retorno médio % (teste)
IFR (RSI)	128	0,92	6026	2292	152,2	14,59
No IFR (RSI)	127	2,93	5112	2036	120,3	6,9
No IFR (RSI) - Sem outlier	126	0,89	5152	2052	121,2	6,94
Estocástico	128	0,9	5738	2194	140,5	10,34
No Estocástico	127	2,96	5403	2135	132	11,19
No Estocástico - Sem outlier	126	0,92	5445	2152	133,1	11,26
MACD	128	0,89	7122	3521	191	14,25
No MACD	127	2,97	4008	797	81,1	7,25
No MACD - Sem outlier	126	0,93	4039	804	81,7	7,29
ADX	128	2,93	6302	2374	152,6	14,33
ADX - Sem outlier	127	0,91	6351	2393	153,8	14,42
No ADX	127	0,91	4834	1953	119,8	7,17
CCI	128	0,89	5770	2211	142,3	7,94
No CCI	127	2,97	5371	2118	130,2	13,61
No CCI - Sem outlier	126	0,93	5413	2135	131,3	13,7
MFI	128	0,94	5826	2193	133,4	13,49
No MFI	127	2,92	5314	2136	139,2	8,01
No MFI - Sem outlier	126	0,88	5355	2153	140,3	8,06
Williams % R (WILLR)	128	2,93	5725	2193	139,5	9,94
Williams % R (WILLR) - Sem outlier	127	0,91	5769	2211	140,5	10
No Williams % R (WILLR)	127	0,91	5416	2136	133,1	11,6
TRIX	128	0,93	6805	2271	164	13,14
No TRIX	127	2,93	4327	2057	108,3	8,37
No TRIX - Sem outlier	126	0,89	4360	2073	109,2	8,42

Fonte: Elaborada pelo autor.

A Tabela 5.8 busca identificar o grau de importância de *features* específicas nos

resultados das investigações. Para tanto, foi obtida a média de resultados dos treinos e testes em que a *feature* esteve presente, e comparados com a média de resultados sem a presença da *feature*. A diferença entre os resultados evidencia a relevância da *feature* em comparação com as demais:

IFR - O Retorno médio percentual nos testes foi 14,59% com a *feature*, e 6,9% sem a *feature*. Desse modo, identificamos que é uma *feature* relevante.

Estocástico - O Retorno médio percentual nos testes foi 10,34% com a *feature*, e 11,19% sem a *feature*. Desse modo, identificamos que é uma *feature* que pode ter contribuído no quadro geral dos resultados, mas sua ausência não diminuiu os resultados.

MACD - O Retorno médio percentual nos testes foi 14,25% com a *feature*, e 7,25% sem a *feature*. Desse modo, identificamos que é uma *feature* relevante.

ADX - O Retorno médio percentual nos testes foi 14,33% com a *feature*, e 7,17% sem a *feature*. Desse modo, identificamos que é uma *feature* relevante.

CCI - O Retorno médio percentual nos testes foi 7,94% com a *feature*, e 13,61% sem a *feature*. Desse modo, identificamos que é uma *feature* que pode ter contribuído no quadro geral dos resultados, mas sua ausência não diminuiu os resultados.

MFI - O Retorno médio percentual nos testes foi 13,49% com a *feature*, e 8,01% sem a *feature*. Desse modo, identificamos que é uma *feature* relevante.

Williams % R - O Retorno médio percentual nos testes foi 9,94% com a *feature*, e 11,6% sem a *feature*. Desse modo, identificamos que é uma *feature* que pode ter contribuído no quadro geral dos resultados, mas sua ausência não diminuiu os resultados.

TRIX - O Retorno médio percentual nos testes foi 13,14% com a *feature*, e 8,37% sem a *feature*. Desse modo, identificamos que é uma *feature* relevante.

Tabela 5.9: Melhores *Features TALIB*. Critério 1.

Ranking 10 melhores combinações de <i>Features TALIB</i> por: Grau de Consistência Teste/Treino					
<i>Features</i> (Combinação)	Grau de Consistência Teste/Treino	Nº de trades (treino)	Nº de trades (teste)	Retorno % (treino)	Retorno % (teste)
ADX WILLR	259,41	97	13	2,00	2,41
RSI	1,27	474	98	28,16	6,93
RSI CCI MFI	1,24	1922	444	42,00	18,00
ESTOCÁSTICO ADX MFI	1,17	1418	378	7,00	7,00
ESTOCÁSTICO ADX MFI WILLR	1,17	2328	605	16,00	14,00
ADX MFI TRIX	1,17	5511	988	43,00	23,13
RSI ESTOCÁSTICO MFI	1,17	2500	593	55,00	20,00
RSI MFI WILLR	1,15	1982	476	51,00	17,24
RSI ESTOCÁSTICO MFI WILLR	1,14	3031	725	53,00	21,00
RSI ESTOCÁSTICO TRIX	1,11	4047	515	42,00	11,00

Fonte: Elaborada pelo autor.

A Tabela 5.9 mostra as dez melhores combinações de *features TALIB*, utilizando como critério o Grau de Consistência Teste/Treino, que foi descrito acima como uma métrica desenvolvida pelo autor para identificar o poder de generalização do modelo treinado com o conjunto específico de *features*. Significa que essas dez combinações de *features* proporcionaram os melhores resultados de testes comparados aos resultados de treinamento do modelo.

Tabela 5.10: Melhores *Features TALIB*. Critério 2.

Ranking 10 melhores combinações de <i>Features TALIB</i> por: Retorno nos Testes					
<i>Features</i> (Combinação)	Grau de Consistência Teste/Treino	Nº de trades (treino)	Nº de trades (teste)	Retorno % (treino)	Retorno % (teste)
RSI MACD ADX MFI WILLR TRIX	0,95	8187	3603	229	45,97
RSI ESTOCÁSTICO MACD ADX MFI TRIX	0,95	8136	3574	231	44,39
RSI MACD ADX CCI MFI TRIX	0,94	8153	3577	231	41,89
RSI MACD ADX MFI TRIX	0,92	7851	3608	259	40,33
RSI ESTOCÁSTICO ADX MFI WILLR TRIX	0,93	8423	3588	227	39,11
MACD ADX CCI MFI TRIX	0,93	8220	3642	216	34,85
ESTOCÁSTICO MACD ADX MFI WILLR TRIX	0,96	8296	3573	187	34,4
RSI ESTOCÁSTICO MACD MFI TRIX	0,94	7202	3314	193	33,07
RSI MACD ADX WILLR TRIX	0,89	8132	3685	267	31,85
MACD ADX MFI TRIX	0,96	8339	3646	166	31,74

Fonte: Elaborada pelo autor.

A Tabela 5.10 mostra as dez melhores combinações de *features TALIB* tendo como critério exclusivamente o resultado nos testes, sem comparação com qualquer outra métrica.

5.5.4 Resultados do Quarto Experimento - *Features SALIB* e *Features TALIB*

Este experimento foi caracterizado pelo uso conjunto dos dois tipos de *features*: *SALIB* e *TALIB*. Desse modo, não faremos novamente uma avaliação das *features* de modo individual, pois já foram objeto de estudo neste capítulo. A avaliação nesse momento reside nos resultados obtidos com o uso conjunto dessas *features*.

Tabela 5.11: Melhores *Features SALIB + TALIB*. Critério 1.

Ranking 10 melhores combinações de Features SALIB + TALIB por: Grau de Consistência Teste/Treino					
Features (Combinação)	Grau de Consistência Teste/Treino	Nº de trades (treino)	Nº de trades (teste)	Retorno % (treino)	Retorno % (teste)
gap quadrant fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 MFI TRIX	1,08	15187	2953	196,70	64,59
gap fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 MFI TRIX	1,08	15187	2953	196,70	64,59
gap quadrant fhigh0 flow0 MFI TRIX	1,07	13926	2380	196,43	49,47
gap fhigh0 flow0 MFI TRIX	1,07	13926	2380	196,43	49,47
gap quadrant fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 ADX MFI TRIX	1,07	15167	2951	239,64	66,44
gap fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 ADX MFI TRIX	1,07	15167	2951	239,64	66,44
gap quadrant fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 TRIX	1,07	15133	2910	147,68	45,32
gap fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 TRIX	1,07	15133	2910	147,68	45,32
gap quadrant fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 ADX MFI	1,06	16580	3603	184,87	65,92
gap fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 ADX MFI	1,06	16580	3603	184,87	65,92

Fonte: Elaborada pelo autor.

A Tabela 5.11 mostra as dez melhores combinações de *features SALIB + TALIB*, utilizando como critério o Grau de Consistência Teste/Treino, que foi descrito acima como uma métrica desenvolvida pelo autor para identificar o poder de generalização do modelo treinado com o conjunto específico de *features*. Significa que essas dez combinações de *features* proporcionaram os melhores resultados de testes comparados aos resultados de treinamento do modelo.

Observamos a seguinte proporção de *features SALIB* e *TALIB* neste conjunto dos dez melhores resultados:

1. Em primeiro lugar no ranking de resultados, tivemos sete *features SALIB* e duas
2. Em segundo lugar, uma combinação de seis *features SALIB* e duas *features TALIB*.
3. Em terceiro lugar, uma combinação de quatro *features SALIB* e duas *features TALIB*.
4. Em quarto lugar, uma combinação de três *features SALIB* e duas *features TALIB*.
5. Em quinto lugar, uma combinação de sete *features SALIB* e três *features TALIB*.
6. Em sexto lugar, uma combinação de seis *features SALIB* e três *features TALIB*.
7. Em sétimo lugar, uma combinação de sete *features SALIB* e uma *feature TALIB*.
8. Em oitavo lugar, uma combinação de seis *features SALIB* e uma *feature TALIB*.
9. Em nono lugar, uma combinação de sete *features SALIB* com duas *features TALIB*.
10. Em décimo lugar, uma combinação de seis *features SALIB* e duas *features TALIB*.

Tabela 5.12: Melhores *Features SALIB + TALIB*. Critério 2.

Ranking 10 melhores combinações de <i>Features SALIB + TALIB</i> por: Retorno nos Testes					
<i>Features</i> (Combinação)	Grau de Consistência Teste/Treino	Nº de trades (treino)	Nº de trades (teste)	Retorno % (treino)	Retorno % (teste)
quadrant weekly_position0 weekly_position1 weekly_position2 TRIX	1,05	22296	5488	178,82	69,33
weekly_position0 weekly_position1 weekly_position2 TRIX	1,05	22296	5488	178,82	69,33
gap quadrant fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 ADX MFI TRIX	1,07	15167	2951	239,64	66,44
gap fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 ADX MFI TRIX	1,07	15167	2951	239,64	66,44
quadrant weekly_position0 weekly_position1 weekly_position2 ADX MFI WILLR	0,98	24120	6186	326,71	65,98
weekly_position0 weekly_position1 weekly_position2 ADX MFI WILLR	0,98	24120	6186	326,71	65,98
gap quadrant fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 ADX MFI	1,06	16580	3603	184,87	65,92
gap fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 ADX MFI	1,06	16580	3603	184,87	65,92
quadrant weekly_position0 weekly_position1 weekly_position2 ADX	1,04	23870	6160	159,63	64,97
weekly_position0 weekly_position1 weekly_position2 ADX	1,04	23870	6160	159,63	64,97

Fonte: Elaborada pelo autor.

A Tabela 5.12 mostra as dez melhores combinações de *features SALIB + TALIB*, utilizando como critério o retorno médio percentual nos testes (não computando o retorno nos treinos dos modelos). Observamos a seguinte proporção de *features SALIB* e *TALIB* neste conjunto dos dez melhores resultados:

1. Em primeiro lugar no ranking de resultados, tivemos quatro *features SALIB* e uma *feature TALIB*.
2. Em segundo lugar, uma combinação de três *features SALIB* e uma *feature TALIB*.
3. Em terceiro lugar, uma combinação de sete *features SALIB* e três *features TALIB*.
4. Em quarto lugar, uma combinação de seis *features SALIB* e três *features TALIB*.
5. Em quinto lugar, uma combinação de quatro *features SALIB* e três *features TALIB*.
6. Em sexto lugar, uma combinação de três *features SALIB* e três *features TALIB*.
7. Em sétimo lugar, uma combinação de sete *features SALIB* e duas *features TALIB*.
8. Em oitavo lugar, uma combinação de seis *features SALIB* e duas *features TALIB*.
9. Em nono lugar, uma combinação de quatro *features SALIB* com uma *feature TALIB*.
10. Em décimo lugar, uma combinação de três *features SALIB* e uma *feature TALIB*.

5.5.5 Análise das *Features* mais relevantes

Dos experimentos, a partir do *Baseline*, ficou claro que o uso de *features* exerce uma influência importante nos resultados do algoritmo, seja nos treinamentos, seja nos testes subsequentes. Como foram utilizadas várias *features SALIB* e *TALIB*, naturalmente surge a curiosidade sobre a relevância de cada uma delas, e se podemos escolher algumas como sendo as mais relevantes para investigação futura.

Computamos os vinte melhores resultados para cada conjunto de *features* (dez resultados pela métrica Grau de Consistência Teste/Treino, e dez pela métrica Retorno nos Testes) com objetivo de identificar as *features* mais presentes nesses resultados.

Das *features SALIB*, observamos o seguinte *ranking* de presença nos resultados (Tabela 5.13):

Tabela 5.13: *Features SALIB* mais relevantes.

weekly position	34
fhigh	29
flow	29
gap	23
quadrant	20
daily position	8
cto	6
ibs	6

Fonte: Elaborada pelo autor.

Encontramos como *feature SALIB* mais relevante a “*weekly position*”, cujo critério é justamente dar uma noção de contexto “semanal” para a série de preços (apesar da negociação ser intradiária), isso significa estimar a posição relativa desta semana (na série temporal) em relação a semanas anteriores. Isso nos permite uma visão de “cenário”. Na mesma linha de prover um “cenário”, temos em segundo lugar o empate das *features* “*From High*” e “*From Low*” (empate que se justifica pelo fato de terem sido instanciada sempre em pares). Em terceiro lugar, a *feature* “*Gap*” trouxe o primeiro elemento de análise relativo ao próprio dia, pois o “*Gap*” é a diferença percentual entre o preço de abertura do dia atual em relação ao fechamento do dia anterior. As demais *features* tiveram relevância reduzida e constam na tabela para observação.

Das *features TALIB*, observamos o seguinte ranking de presença nos resultados (Tabela 5.14):

Tabela 5.14: *Features TALIB* mais relevantes.

MFI	29
ADX	25
TRIX	25
IFR	12
Williams % R	10
Estocástico	9
MACD	8
CCI	3

Fonte: Elaborada pelo autor.

Encontramos como *feature TALIB* mais relevante o indicador "*MFI - Market Facilitation Index*", o que é curioso, por se tratar de um indicador teoricamente mais completo (e complexo) do que os demais, por computar não somente dados de preço do ativo, mas também de volume negociado. Em segundo lugar, temos um empate entre os indicadores "ADX" e "TRIX". O indicador "ADX", que é um "medidor de tendência", (faz sentido porque "tendência" é um conceito importante tanto na análise de séries temporais como na chamada escola de Análise Técnica no mercado financeiro). O indicador "TRIX" vem na mesma linha de indicadores que fornecem um "cenário" para o preço do ativo. Os indicadores restantes tiveram relevância menor e estão na tabela para consulta.

5.5.6 Melhores resultados

Até agora vimos os resultados dos experimentos em correspondência com os tipos de *features*. Nesse momento, veremos um ranking dos melhores resultados, independentemente do tipo de *feature* utilizado, ou seja, uma análise dos 20 melhores resultados de todas as investigações feitas nos experimentos.

Tabela 5.15: Melhores resultados.

Ranking 20 melhores combinações de <i>Features</i> por: Retorno nos Testes					
<i>Features</i> (Combinação)	Grau de Consistência Teste/Treino	Nº de trades (treino)	Nº de trades (teste)	Retorno % (treino)	Retorno % (teste)
quadrant daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2	1,01	23878	6108	350	90
daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2	1,01	23878	6108	350	90
quadrant fhigh0 flow0 weekly_position0 weekly_position1	1,07	21132	4917	135	72
fhigh0 flow0 weekly_position0 weekly_position1	1,07	21132	4917	135	72
gap cto1 quadrant ibs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,05	16988	3481	262	69,33
gap cto1 ibs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,05	16988	3481	262	69,33
gap cto1 cto2 quadrant ibs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,03	17046	3237	351	66,44
gap cto1 cto2 ibs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,03	17046	3237	351	66,44
quadrant weekly_position0 weekly_position1 weekly_position2	1,03	23739	6075	171	66
weekly_position0 weekly_position1 weekly_position2	1,03	23739	6075	171	66
gap quadrant daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2	0,96	19879	4805	380	65,98
gap daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2	0,96	19879	4805	380	65,98
gap cto1 cto2 quadrant ibs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1	1,03	16775	3070	287	65,92
gap cto1 cto2 ibs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1	1,03	16775	3070	287	65,92
quadrant weekly_position0	1,02	24267	5971	173	66
weekly_position0	1,02	24267	5971	173	66
quadrant ibs1 daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2	0,96	23977	6176	348	64,97
ibs1 daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2	0,96	23977	6176	348	64,97
gap cto1 quadrant ibs1 fhigh0 flow0 daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2	0,99	17046	3490	343	64,59
gap cto1 ibs1 fhigh0 flow0 daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2	0,99	17046	3490	343	64,59

Fonte: Elaborada pelo autor.

A Tabela 5.15 traz os 20 melhores resultados sem critério de tipo de *feature*. Esses resultados utilizaram como critério o Retorno Percentual nos Testes, ou seja, o retorno que os modelos treinados geraram após o período de treino. São, portanto, negociações pós-treino, ou "no futuro" (*out-of-sample*), no *dataset*. Nesse período não ocorre aprendizado por parte dos modelos, somente execução das regras de negociação aprendidas no treinamento. Observamos que as *features* que compõem os 20 melhores resultados são do tipo *SALIB*. Uma análise mais detalhada do conjunto dos experimentos é feita na próxima seção, 5.4, Considerações finais.

5.6 Considerações finais

Com base nos experimentos realizados, observamos que o uso de *features* derivadas da experiência de um especialista no assunto, (neste trabalho, o mercado financeiro), proporcionou um incremento no resultado obtido em negociações simuladas, instanciadas nos experimentos.

O resultado *Benchmark* (*buy-and-hold*, ou investimento passivo) trouxe no período o retorno de -14,95%, o experimento *Baseline* (*buy-only*, a cada *datapoint*) obteve o retorno de -19,65%. Quando foram introduzidas *features* tanto criadas pelo autor como obtidas na biblioteca de Indicadores Técnicos (utilizada amplamente por participantes do mercado financeiro), os resultados passaram do campo negativo para o positivo: 90% de retorno no melhor caso de uso das *features SALIB*, 45,97% de retorno para a melhor combinação de *features TALIB*, e, 69,33% de retorno com a combinação mista de *features*

SALIB e TALIB.

Estes experimentos constituem apenas investigações sobre o uso de *features*, e não estratégias de negociação validadas para uso imediato. Para transformar os experimentos em estratégias específicas, seriam necessários outros procedimentos, como: estipulação de critérios de saída como alvo de lucro (*target*) e limite de risco (*stop-loss*), além da possibilidade de gestão de exposição variando o número de contratos utilizado em cada negociação, e a inserção dos custos operacionais e *slippage* (diferença provocada por flutuações de liquidez ou volume disponível no momento das negociações) para simulação de condições "reais" de mercado.

Apesar dos experimentos não conterem esses elementos que os tornariam estratégias operacionais completas, (especialmente a inserção de custos operacionais), eles cumprem seu propósito em evidenciar a validade do uso de *features* como parte do processo de pesquisa, pois o experimento *Baseline*, (que corresponderia a negociações a cada *datapoint*), também não sofreu o impacto de custos operacionais. Sobre o uso das *features*, sugerimos que investigações subsequentes por parte do leitor possam ser mais proveitosas se utilizarem as *features* com maior presença na amostragem dos melhores resultados, evidenciadas nas Tabelas 5.13 e 5.14. O repositório com o código desenvolvido para uso nos experimentos, bem como com o dataset utilizado estão disponíveis neste link: ([Link para o repositório Github](#)).

6 CONCLUSÃO

Este trabalho teve como objetivo estudar a importância do uso de *features* no treinamento de modelos de machine *learning* para negociação no mercado de capitais. O autor teve esta motivação por não encontrar na literatura uma ênfase a esse aspecto da pesquisa: sua observação foi de que frequentemente os autores focavam na estrutura do algoritmo utilizado, muitas vezes propondo avanços a essa estrutura (inovação no desenho de algoritmos), e as *features* ocupavam um espaço secundário no esforço de pesquisa.

A visão do autor era de que, em algum momento, todos os algoritmos se tornam disponíveis (*open source*) à comunidade global de pesquisadores. Desse modo, como obter resultados superiores, ao aplicar algoritmos aos investimentos? Segundo o autor, nesse ponto entra a experiência do especialista na área, e sua capacidade em desenvolver *features* relevantes ao *dataset* e ao objeto de pesquisa. Este trabalho, portanto, dedicou-se a encontrar a relevância das *features* no desenvolvimento de *algotrading*.

Foi proposto um experimento duplo inicial para a criação de referências que serviriam de base de comparação para os experimentos posteriores que utilizaram *features*: o experimento *Benchmark* (investimento passivo ou *buy-and-hold* no ativo objeto da pesquisa), e o experimento *Baseline*, (que correspondia a negociações simuladas de “compra” a cada *datapoint*). Esses experimentos forneceram a possibilidade de comparação e avaliação dos experimentos seguintes, que usariam diversas combinações de *features*.

Para estudar o impacto das *features* na modelagem algorítmica para o mercado de capitais, foram realizados experimentos com diferentes conjuntos de *features* no treinamento dos modelos. Nos experimentos propriamente ditos, foram utilizadas as *features* propostas pelo autor (*SALIB - Structural Analysis Library*), e também *features* encontradas na biblioteca *talib.py* (*TALIB - Technical Analysis Library*), de forma separada, e também, conjuntamente. Cada “investigação” proposta correspondeu à modelagem e teste do algoritmo com o conjunto de *features* específico, (compreendendo Treino e Teste). Utilizando somente as *features* *SALIB* foram feitas 2.879 investigações, utilizando somente as *features* *TALIB*, foram feitas 255 investigações, e, finalmente, utilizando combinações de ambos os tipos de *feature* (*SALIB* e *TALIB*) foram feitas 4.080 investigações. Desse modo, a pesquisa contou com 7.214 conjuntos diferentes de Treino/Teste do algoritmo. Consideramos os resultados obtidos satisfatórios, demonstrando que o uso de *features* pode melhorar significativamente a performance dos modelos de machine *learning* na previsão de movimentos do ativo financeiro escolhido nesta pesquisa, o mini-índice futuro (mini

contrato futuro do Índice Bovespa).

Ao avaliar os principais resultados obtidos, não podemos ignorar que tanto o *Benchmark* como o *Baseline* obtiveram resultados negativos. Esse fato é muito importante, pois, no caso do *Benchmark*, não houve uma tendência positiva nos preços do ativo (tendência de alta), ou seja, não houve um resultado positivo pelo investimento “passivo” (sem movimentações), e no caso do *Baseline*, a negociação ativa (frequente) com expectativa de ganhos com a alta do preço, também se mostrou negativa, em seus resultados. Esses dois casos são importantes, pois evidenciam que qualquer resultado positivo obtido nos experimentos seguintes foi mérito do uso das *features* em conjunto com o algoritmo selecionado. E quanto maior o resultado, mais o algoritmo e as *features* conseguiu extrair valor do *dataset*, especialmente em relação aos resultados nos períodos de Teste, onde o algoritmo não aprendia mais, e somente utilizava o conhecimento já aprendido no Treino.

Quanto aos resultados das *features SALIB*, observamos que eles ficavam mais positivos quanto mais *features* figurassem nas investigações (cada investigação correspondendo a um único conjunto ou combinação de *features*). O melhor resultado foi obtido pelo conjunto de *features*: (*quadrant daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2*), com 90% de retorno percentual no período de Teste.

As *features TALIB* testadas também obtiveram retornos positivos, e encontramos o mesmo fenômeno do resultado crescer de forma linear ao número de *features* utilizado. O melhor resultado foi encontrado com o seguinte conjunto de *features*: (*RSI MACD ADX MFI WILLR TRIX*), obtendo 45,97% de retorno percentual no período de Teste.

Ao misturar *features SALIB* com as *TALIB*, gerando uma combinação composta, também foi possível obter resultados positivos (e portanto, superiores ao *Benchmark* e *Baseline*), aproveitando uma possível sinergia entre os dois tipos de *features*. O resultado mais expressivo foi obtido com a seguinte combinação de *features*: (*quadrant weekly_position0 weekly_position1 weekly_position2 TRIX*), com 69,33% de retorno no período de Teste do experimento.

Tais resultados, não somente por serem positivos, mas, especialmente, por serem positivos num contexto negativo da série temporal do ativo objeto de estudo, são altamente representativos da capacidade que o algoritmo teve em generalizar seu aprendizado com os conjuntos de *features*. No mercado financeiro, toda vez que uma estratégia operacional “bate o *Benchmark*”, isto é, supera o resultado obtido passivamente no índice de referência (Ibovespa), é sinal de que a estratégia contém mérito em obter resultados positivos acima da média de mercado. Como o ativo objeto deste estudo era o próprio contrato futuro

do Ibovespa, o *Benchmark* escolhido não somente corresponde à performance do ativo em si mesmo, como também representa o *benchmark* do mercado financeiro como um todo. Acreditamos que a este trabalho trouxe duas contribuições importantes para o tema: em primeiro lugar, dar às *features* o lugar de destaque que merecem no repertório de ferramentas disponíveis aos pesquisadores, ressaltando a importância da presença de um especialista da área e a necessidade de encontrar *features* que representem adequadamente os atributos do *dataset*.

Em segundo lugar, a contribuição trazida pelo autor na criação das *features SALIB* (*Structural Analysis Library*), que, salvo melhor juízo, não estavam disponíveis à comunidade de pesquisadores, e que, além de corresponder a elementos trazidos pela experiência de mais de quinze anos no mercado financeiro, permitem múltiplas instanciações e novos desenvolvimentos.

Em trabalhos futuros, o desenvolvimento de *features* poderá ser ampliado, e sua utilização com outros algoritmos (além do algoritmo utilizado neste trabalho - *QDA* - *Quadratic Discriminant Analysis*) é estimulada e bem-vinda. Com efeito, acreditamos que o esforço em desenvolver *features* permitirá que diferentes algoritmos potencialmente alcancem resultados superiores, por trabalharem de forma mais adaptada ao *dataset*.

Também em trabalhos futuros podem ser explorados outros ativos financeiros, além do ativo utilizado neste trabalho (mini-índice futuro). Uma avenida interessante de investigação, também, é a de análise multivariada, onde um *dataset* e suas *features* figure como *feature* para previsão em outro *dataset*. Por exemplo, os preços da cotação do ativo “petróleo”, (negociado na Bolsa de Chicago, e referência para o mundo), e suas *features* atuando como *features* para previsão da série temporal da ação Petrobrás. Inúmeras possibilidades surgem ao combinar séries exógenas.

Trabalhos futuros também deverão contemplar uma investigação utilizando algoritmos de *feature selection*. Como utilizamos diversas *features* neste trabalho, e o número de *features* pode ainda ser incrementado, uma pesquisa utilizando procedimentos de *feature selection* possui mérito e deverá ser desenvolvida.

Em conclusão, observamos que este trabalho nos permitiu validar a premissa de que “*features* importam”, e não somente importam como também permitem a um algoritmo de *machine learning* extrair o máximo de valor disponível em um *dataset*. A capacidade de identificar ou desenvolver *features* relevantes deveria ser um tema de estudo e fazer parte do currículo formal dos pesquisadores que utilizam modelagem algorítmica. Na ausência de *insights* quanto à natureza do *dataset*, o pesquisador deveria consultar um

especialista na área, pois sua experiência permite identificar os aspectos mais relevantes presentes no *dataset*.

Concluimos, também, que o uso de *features* é uma técnica importante e eficaz no treinamento de modelos de *machine learning* para negociação no mercado de capitais, cujo tipo de série temporal contém altos níveis de ruído e de aleatoriedade. A escolha adequada das *features* e dos modelos de aprendizado de máquina pode levar a previsões mais precisas e, conseqüentemente, a melhores resultados financeiros. O uso de *features* no treinamento de modelos de *machine learning* para negociação no mercado de capitais deveria ser considerada uma prática fundamental e ser amplamente adotada pelos investidores. Acreditamos que essa abordagem pode revolucionar o mercado financeiro e trazer novas oportunidades de lucro para aqueles que a utilizam em seus esforços de análise com fins de negociação.

REFERÊNCIAS

BABA, N.; INOUE, N.; YANJUN, Y. Utilization of soft computing techniques for constructing reliable decision support systems for dealing stocks. In: **IEEE. Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)**. [S.l.], 2002. v. 3, p. 2150–2155.

CALAINHO, F. D. Previsão da direção de índices da bovespa por intermédio de máquina de suporte vetorial. 2015.

CAO, L.; TAY, F. E. Financial forecasting using support vector machines. **Neural Computing & Applications**, Springer, v. 10, p. 184–192, 2001.

CAO, L.-J.; TAY, F. E. H. Support vector machine with adaptive parameters in financial time series forecasting. **IEEE Transactions on neural networks**, IEEE, v. 14, n. 6, p. 1506–1518, 2003.

CARAMICO, A. Proposição de um modelo preditivo do ibovespa por meio da utilização de redes neurais artificiais. 01 2010.

CASTRO, C. et al. Aplica ao de um modelo ensemble de redes neurais artificiais para previsão de séries temporais não estacionárias. 03 2023.

CHALVATZIS, C.; HRISTU-VARSAKELIS, D. High-performance stock index trading: making effective use of a deep lstm neural network. **arXiv preprint arXiv:1902.03125**, 2019.

CHEN, G.; CHEN, Y.; FUSHIMI, T. Application of deep learning to algorithmic trading. **Tech. Rep., Tech. Rep.**, Stanford University, 2017.

CHO, C.-H. et al. Toward stock price prediction using deep learning. In: **Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion**. [S.l.: s.n.], 2019. p. 133–135.

DAMODARAN, A. **Damodaran on Valuation: Security Analysis for Investment and Corporate Finance**. [S.l.]: Wiley Finance, 2006.

EDWARDS, R.; MAGEE, J. **Technical Analysis of Stock Trends**. Stock Trend Service, 1948. Available from Internet: <<https://books.google.com.br/books?id=1ywJAQAAIAAJ>>.

GRAHAM, B. **The Intelligent Investor**. Prabhat Prakashan, 1949. Available from Internet: <<https://books.google.com.br/books?id=meDYDQAAQBAJ>>.

GRAHAM, B.; DODD, D.; DODD, D. **Security Analysis: The Classic 1934 Edition**. McGraw-Hill Education, 1934. ISBN 9780070244962. Available from Internet: <<https://books.google.com.br/books?id=wXlInZ1uqK0C>>.

JOOSERY, B.; DEEPA, G. Comparative analysis of time-series forecasting algorithms for stock price prediction. In: **Proceedings of the International Conference on Advanced Information Science and System**. [S.l.: s.n.], 2019. p. 1–6.

JOSEPH, C.-E.; NOBAKHT, B.; LONI, B. Stock market analysis and prediction using hidden markov models. **Journal of Business and Economics Research (JBER)**, Clute Institute, v. 10, n. 7, p. 397–410, 2012.

LABIAD, B.; BERRADO, A.; BENABBOU, L. Short term prediction framework for moroccan stock market using artificial neural networks. In: **Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications**. [S.l.: s.n.], 2018. p. 1–6.

LEUNG, C. K.-S.; MACKINNON, R. K.; WANG, Y. A machine learning approach for stock price prediction. In: **Proceedings of the 18th International Database Engineering & Applications Symposium**. [S.l.: s.n.], 2014. p. 274–277.

LI, J.; RAO, R.; SHI, J. Learning to trade with deep actor critic methods. In: **2018 11th International Symposium on Computational Intelligence and Design (ISCID)**. [S.l.: s.n.], 2018. v. 02, p. 66–71.

LI, Y.; ZHENG, W.; ZHENG, Z. Deep robust reinforcement learning for practical algorithmic trading. **IEEE Access**, IEEE, v. 7, p. 108014–108022, 2019.

LOUWERSE, V.; ROTHKRANTZ, L. Intraday stock forecasting. In: **Proceedings of the 15th International Conference on Computer Systems and Technologies**. [S.l.: s.n.], 2014. p. 202–209.

MADGE, S.; BHATT, S. Predicting stock price direction using support vector machines. **Independent work report spring**, Princeton University Princeton, NJ, USA, v. 45, 2015.

MARTINEZ, L. et al. From an artificial neural network to a stock market day-trading system: A case study on the bm&f bovespa. In: . [S.l.: s.n.], 2009. p. 2006–2013.

MOODY, J.; SAFFELL, M. Learning to trade via direct reinforcement. **IEEE Transactions on Neural Networks**, v. 12, n. 4, p. 875–889, 2001.

NAMETALA, C. A. et al. An automated investment strategy using artificial neural networks and econometric predictors. In: **Proceedings of the XII Brazilian Symposium on Information Systems on Brazilian Symposium on Information Systems: Information Systems in the Cloud Computing Era-Volume 1**. [S.l.: s.n.], 2016. p. 152–159.

ONCHAROEN, P.; VATEEKUL, P. Deep learning using risk-reward function for stock market prediction. In: **Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence**. [S.l.: s.n.], 2018. p. 556–561.

PATIL, P. et al. Stock market prediction using ensemble of graph theory, machine learning and deep learning models. In: **Proceedings of the 3rd International Conference on Software Engineering and Information Management**. [S.l.: s.n.], 2020. p. 85–92.

PRADO, M. Lopez de. **Advances in Financial Machine Learning**. [S.l.]: John Wiley & Sons, 2018.

PRADO, M. Lopez de. A framework for financial machine learning. **The Journal of Financial Data Science**, Wiley Online Library, v. 1, n. 1, p. 1–15, 2018.

RHEA, R. **The Dow Theory: An Explanation of Its Development and an Attempt to Define Its Usefulness as an Aid in Speculation**. Barron's, 1932. ISBN 9788175980334. Available from Internet: <<https://books.google.com.br/books?id=F8DPzAEACAAJ>>.

ROCA, R. L. R. d. S.; M'OL, A. L. R. Prediction of the bovespa index by means of artificial neural networks. 2015.

SEWELL, M. A history of the efficient market hypothesis. 01 2011.

SEZER, O. B.; OZBAYOGLU, A. M.; DOGDU, E. 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.

SONG, Y.; LEE, J. Design of stock price prediction model with various configuration of input features. In: **Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing**. [S.l.: s.n.], 2019. p. 1–5.

TAY, F. E.; CAO, L. Modified support vector machines in financial time series forecasting. **Neurocomputing**, Elsevier, v. 48, n. 1-4, p. 847–861, 2002.

TORRALBA, E. M. Development of a deep learning-lstm trend prediction model of stock prices. In: **Proceedings of the 2019 International Conference on Management Science and Industrial Engineering**. [S.l.: s.n.], 2019. p. 126–133.

UPADHYAY, V. P. et al. Forecasting stock market movements using various kernel functions in support vector machine. In: **Proceedings of the International Conference on Advances in Information Communication Technology & Computing**. [S.l.: s.n.], 2016. p. 1–5.

WANG, S. The prediction of stock index movements based on machine learning. In: **Proceedings of the 2020 12th International Conference on Computer and Automation Engineering**. [S.l.: s.n.], 2020. p. 1–6.

WANG, Y. et al. Deep q-trading. 2017.

WENG, C. K. C. V. J. A svm approach to stock trading. Citeseer, 2009.

WILLIAMS, J. **The Theory of Investment Value**. Harvard University Press, 1938. (Investment value). ISBN 9780678080504. Available from Internet: <<https://books.google.com.br/books?id=cIhCAAAAIAAJ>>.

YOU, D. et al. Stock index prediction method based on dynamic weighted ensemble learning. In: **Proceedings of the 2019 International Conference on Robotics Systems and Vehicle Technology**. [S.l.: s.n.], 2019. p. 41–46.

ZHANG, C. et al. Predicting chinese stock market price trend using machine learning approach. In: **Proceedings of the 2nd International Conference on Computer Science and Application Engineering**. [S.l.: s.n.], 2018. p. 1–5.