

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA  
UNIDADE DE PÓS-GRADUAÇÃO, EXTENSÃO E PESQUISA  
MESTRADO PROFISSIONAL EM GESTÃO E TECNOLOGIA EM  
SISTEMAS PRODUTIVOS

CAMILO ILZO SHIMABUKURO

UMA CONTRIBUIÇÃO PARA IMPLEMENTAÇÃO DO *DEEP LEARNING* APLICADO À  
NEGOCIAÇÃO DE AÇÕES POR ALGORITMOS NA B3

São Paulo  
Março/2020

CAMILO ILZO SHIMABUKURO

UMA CONTRIBUIÇÃO PARA IMPLEMENTAÇÃO DO *DEEP LEARNING* APLICADO À  
NEGOCIAÇÃO DE AÇÕES POR ALGORITMOS NA B3

Dissertação apresentada como exigência parcial para a obtenção do título de Mestre em Gestão e Tecnologia em Sistemas Produtivos do Centro Estadual de Educação Tecnológica Paula Souza, no Programa de Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos, sob a orientação do Prof. Dr. Napoleão Verardi Galegale

São Paulo

Março/2020

FICHA ELABORADA PELA BIBLIOTECA NELSON ALVES VIANA  
FATEC-SP / CPS

S556c Shimabukuro, Camilo Ilzo  
Uma contribuição para implementação do deep learning aplicado  
à negociação de ações por algoritmos na B3 / Camilo Ilzo  
Shimabukuro . – São Paulo: CPS, 2020.  
119 f. : il.

Orientador: Prof. Dr. Napoleão Verardi Galegale  
Dissertação (Mestrado Profissional em Gestão e Tecnologia em  
Sistemas Produtivos) – Centro Estadual de Educação Tecnológica  
Paula Souza, 2020.

1. Sistemas produtivos. 2. Mercado de ações. 3. Algorithmic  
trading. 4. Deep learning. 5. LSTM. I. Galegale, Napoleão Verardi.  
II. Centro Estadual de Educação Tecnológica Paula Souza. III.  
Título.

CRB8-8390

CAMILO ILZO SHIMABUKURO

UMA CONTRIBUIÇÃO PARA IMPLEMENTAÇÃO DO *DEEP LEARNING* APLICADO À  
NEGOCIAÇÃO DE AÇÕES POR ALGORITMOS NA B3

---

Prof. Dr. Napoleão Verardi Galeale

---

Prof. Dr. José Odílio dos Santos

---

Prof. Dr. Marcelo Tsuguio Okano

São Paulo, 22 de maio de 2020

À memória de meus pais que por seus esforços  
possibilitaram minha realização pessoal e  
profissional.

## **AGRADECIMENTOS**

Ao Prof. Dr. Napoleão Verardi Galegale, meu orientador, por sua atenção, capacidade de transmissão de conhecimento, visão sobre a produção acadêmica de qualidade e por indicar os caminhos a seguir na realização deste trabalho.

Ao Prof. Dr. Marcelo Tsuguo Okano, por suas valiosas contribuições ao longo do processo de qualificação e por inculcar-nos, durante as disciplinas, os saberes metodológicos na produção de conhecimento científico.

Ao Prof. Dr. José Odílio dos Santos, por compartilhar seus conhecimentos sobre o mercado financeiro e de capitais, e propor melhorias de grande valor ao conteúdo desta dissertação.

Aos Profs. Drs. Carlos Hideo Arima, Getúlio K. Akabane e Marcelo Duduchi por suas contribuições, tanto em sala de aula como em eventos acadêmicos, partilhando suas experiências e visões sobre pesquisa, docência e extensão.

Aos colegas de mestrado sempre dispostos a oferecer apoio aos demais e dedicados na busca do conhecimento.

Aos funcionários e à administração da Unidade de Pós Graduação do Centro Paula Souza, por sua colaboração e competência.

À minha esposa Vilma e à toda a família pelo apoio e dedicação nesta trajetória.

*The errors which arise from the absence of facts are far more numerous and more durable than those which result from unsound reasoning respecting true data.*

(Charles Babbage, 1832)

## RESUMO

SHIMABUKURO, C. I. **Uma contribuição para implementação do *Deep Learning* aplicado à negociação de ações por algoritmos na B3**. 119 f. Dissertação (Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos). Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2020.

Revisões da literatura no período de 2009 a 2015 apresentavam a área de finanças como campo pouco explorado pelo *Deep Learning*. Considerando-se os resultados já alcançados pela pesquisa e a disponibilização de novas ferramentas computacionais, identificou-se uma oportunidade para a investigação da produção científica nesta área, no período de 2016 a 2019. Tal investigação evidenciou um crescimento recente das publicações científicas sobre *Deep Learning* aplicado ao *Algorithmic Trading* a partir de 2018. O objetivo deste trabalho é pesquisar uma abordagem baseada em *Deep Learning* aplicada à negociação de ações baseada em algoritmos sobre ativos listados na bolsa de valores brasileira, B3, e implementar um artefato na forma de software com esta abordagem. Utiliza-se o método *Design Science Research* para a implementação, execução e avaliação do artefato de software. A abordagem *Long Short-Term Memory* foi escolhida por sua capacidade de extração de características das relações não lineares de séries temporais, de retenção seletiva das informações passadas relevantes, e de manter o aprendizado evitando o *vanishing gradient* e o *exploding gradient*. O artefato de software, representado por modelo lógico e código fonte, foi testado em duas ações negociadas na B3, tendo como dados de entrada os preços de fechamento diários de um período de dez anos, de 2008 a 2018. A análise dos resultados evidenciou aderência entre preços alvo e de predição em momentos de manutenção de tendência, e atraso no reconhecimento do novo padrão em momentos de reversão de preços ou maior volatilidade. As variâncias da acurácia da predição sobre preços padronizados para PETR4 e VALE3 foram 0,000193 e 0,000259 respectivamente. Tais resultados indicam a aplicabilidade do modelo LSTM no mercado de ações brasileiro.

**Palavras-chave:** Mercado de Ações. *Algorithmic Trading*. *Machine Learning*. *Deep Learning*. Sistemas Produtivos.

## ABSTRACT

SHIMABUKURO, C. I. **A contribution for the implementation of Deep Learning applied to stock trading by algorithms on B3**: state of the art of research and implementation of a model. 119f. Dissertação (Mestrado Profissional em Gestão e Desenvolvimento da Educação Profissional). Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2020.

Literature reviews from 2009 to 2015 presented finance as a field little explored by Deep Learning. Considering the results already achieved by the research and the availability of new computational tools, it has been identified an opportunity for the investigation of scientific production in this area, in the period from 2016 to 2019. Such investigation has evidenced a recent growth of scientific publications on Deep Learning applied to Algorithmic Trading from 2018. The objective of this work is to research an approach based on Deep Learning applied to Algorithmic Trading on Brazilian stock exchange and to present an artifact in the form of software with this approach. The Design Science Research method is used to implement, execute, and evaluate the software artifact. The Long Short-Term Memory approach was chosen for its ability to extract characteristics of non-linear time series relationships, for selective retention of relevant past information, and to maintain learning by avoiding vanishing gradient and exploding gradient. The software artifact, represented by the logical model and source code, was tested on two shares traded at B3, using daily closing prices from 2008 to 2018 as input data. Analysis of results showed adherence between target and prediction prices in times of trend formation, and a delay in recognizing the new pattern in times of price reversal or higher volatility. The variances of prediction accuracy over standardized prices for PETR4 and VALE3 were 0.000193 and 0.000259, respectively. Such results indicate the applicability of the LSTM model in the Brazilian stock market.

Keywords: Stock Markets. Algorithmic Trading. Machine Learning. Deep Learning. Productive Systems.

## LISTA DE QUADROS

Quadro 1 – Critérios de busca sobre AT .....	26
Quadro 2 – Critérios de seleção dos documentos sobre riscos e benefícios do AT .....	30
Quadro 3 – Critérios de busca para ML .....	39
Quadro 4 – Critérios de busca para DL .....	49
Quadro 5 – Quatro tipos de dados financeiros .....	64
Quadro 6 – Critérios de busca sobre abordagens de ML aplicadas ao AT.....	64
Quadro 7 – Critérios de busca sobre abordagens de DL aplicadas ao AT .....	71
Quadro 8 – Comparativo das abordagens de DL mais frequentes .....	82

## LISTA DE TABELAS

Tabela 1 – Número de publicações das bases consultadas sobre AT .....	27
Tabela 2 – Crescimento das publicações sobre AT .....	28
Tabela 3 – Dez Países que mais publicaram sobre AT .....	29
Tabela 4 – Artigos selecionados e número de citações sobre riscos e benefícios do AT .....	31
Tabela 5 – Principais autores sobre riscos e benefícios do AT, artigos, afiliação e índice h ...	34
Tabela 6 – Publicações sobre ML .....	39
Tabela 7 – Crescimento das publicações sobre ML nas bases Scopus e Web of Science .....	41
Tabela 8 – Dez Países que mais publicaram sobre ML .....	41
Tabela 9 – Dez publicações com maior número de citações sobre ML .....	42
Tabela 10 – Número de publicações das bases consultadas sobre DL .....	50
Tabela 11 – Crescimento das publicações sobre DL nas bases Scopus e Web of Science .....	51
Tabela 12 – Dez Países que mais publicaram sobre DL .....	52
Tabela 13 – Vinte publicações com maior número de citações sobre DL .....	53
Tabela 14 – Autores de maior impacto e produtividade em DL. ....	54
Tabela 15 – Publicações sobre abordagens de ML aplicadas ao AT .....	65
Tabela 16 – Crescimento das publicações sobre abordagens de ML aplicadas ao AT .....	66
Tabela 17 – Dez Países que mais publicaram sobre abordagens de ML aplicadas ao AT .....	67
Tabela 18 – Dez publicações com maior número de citações sobre ML aplicado ao AT .....	68
Tabela 19 – Publicações sobre abordagens de DL aplicadas ao AT .....	71
Tabela 20 – Crescimento das publicações sobre abordagens de DL aplicadas ao AT .....	73
Tabela 21 – Dez Países que mais publicaram sobre abordagens de DL aplicadas ao AT .....	73
Tabela 22 – Artigos sobre abordagens de DL aplicados ao AT mais citados .....	74
Tabela 23 – Autores sobre abordagens de DL aplicados ao AT mais citados .....	76
Tabela 24 – Critérios de seleção sobre abordagens de DL aplicadas ao AT .....	77
Tabela 25 – Fontes de dados das abordagens de DL aplicadas ao AT .....	78
Tabela 26 – Objetivos de saída das abordagens de DL aplicadas ao AT .....	78
Tabela 27 – Frequência de Abordagens de DL em artigos de AT .....	79
Tabela 28 – Abordagens de DL aplicadas ao AT por objetivo de saída .....	80
Tabela 29 – Avaliação geral de desempenho do DL aplicado ao AT .....	81
Tabela 30 – Avaliação de desempenho das abordagens de DL aplicadas ao AT .....	81
Tabela 31 – Medidas de Dispersão da Acurácia .....	101

## LISTA DE FIGURAS

Figura 1 – Evolução tecnológica das Bolsas de Valores .....	19
Figura 2 – Contexto da intervenção e finalidades da pesquisa.....	20
Figura 3 – Estrutura do Referencial Teórico .....	22
Figura 4 – Principais módulos do sistema de negociação de ações.....	26
Figura 5 – Evolução das publicações sobre AT .....	27
Figura 6 – Evolução das publicações sobre AT .....	28
Figura 7 – Dez países que mais publicaram sobre AT. ....	29
Figura 8 – Publicações por Ano sobre ML (Scopus).....	40
Figura 9 – Publicações por Ano sobre ML (Web of Science).....	40
Figura 10 – Dez países que mais publicaram sobre ML.....	42
Figura 11 – Classificação de Máquinas de Vetores de Suporte .....	44
Figura 12 – A evolução da Inteligência Artificial. ....	46
Figura 13 – Modelo matemático de uma unidade neural. ....	47
Figura 14 – Busca do mínimo de uma função .....	48
Figura 15 – Publicações por Ano sobre DL (Scopus) .....	50
Figura 16 – Publicações por Ano sobre DL (Web of Science) .....	51
Figura 17 – Dez países que mais publicaram sobre DL. ....	52
Figura 18 – Publicações sobre abordagens de ML aplicadas ao AT (Scopus).....	65
Figura 19 – Publicações sobre abordagens de ML aplicadas ao AT (Web of Science) .....	66
Figura 20 – Países que mais publicaram abordagens de ML aplicadas ao AT .....	67
Figura 21 – Publicações sobre abordagens de DL aplicadas ao AT (Scopus) .....	72
Figura 22 – Publicações sobre abordagens de DL aplicadas ao AT (Web of Science).....	72
Figura 23 – Países que mais publicaram abordagens de DL aplicadas ao AT .....	74
Figura 24 – Sequência de módulos da LSTM .....	83
Figura 25 – Estrutura de comporta sigmoide de unidade LSTM .....	84
Figura 26 – Detalhamento da estrutura interna de unidade LSTM .....	85
Figura 27 – Modelo lógico – diagrama de processos .....	91
Figura 28 – PETR4: Alvo x Predição.....	98
Figura 29 – VALE3: Alvo x Predição .....	98
Figura 30 – PETR4: Alvo – Predição.....	100
Figura 31 – VALE3: Alvo – Predição .....	100

## LISTA DE SIGLAS

ADAM	<i>Adaptative Moment Estimation</i>
ADR	<i>American Depositary Receipt</i>
AI	Inteligência Artificial
ANN	Rede Neural Artificial
API	<i>Application Programming Interface</i>
AT	<i>Algorithmic Trading</i>
BP	<i>Backpropagation</i>
BSD	<i>Berkeley Software Distribution</i>
CAFFE	<i>Convolutional Architecture for Fast Feature Embedding</i>
CAPM	<i>Capital Asset Price Model</i>
CNN	<i>Convolutional Neural Network</i>
CUDA	<i>Compute Unified Device Architecture</i>
DBN	<i>Deep Belief Network</i>
DL	<i>Deep Learning</i>
DNN	<i>Deep Neural Network</i>
DSR	<i>Design Science Research</i>
DT	<i>Decision Tree</i>
ECN	Redes de Comunicação Eletrônica
EMH	Hipótese do Mercado Eficiente
ES	<i>Expert Systems</i>
EWMA	Médias Móveis Ponderadas Exponencialmente
FFNN	<i>Feed Forward Neural Network</i>
FIX	<i>Financial Information eXchange</i>
FWCI	<i>Field-Weighted Citation Impact</i>
GA	Algoritmo Genético
GARCH	Heterocedasticidade Condicional Auto-Regressiva Generalizada
GEP	<i>Gene Expression Programming</i>
GPU	<i>Graphics Processing Unit</i>
GRU	<i>Gated Recurrent Unit</i>
GTN	<i>Graph Transformer Network</i>
HFT	<i>High Frequency Trading</i>

HIS	<i>Hybrid Intelligence System</i>
HMM	<i>Hidden Markov Model</i>
IPO	Oferta Pública Inicial de Ações
IS	<i>Implementation Shortfall</i>
KNN	<i>K-Nearest Neighbors</i>
LOB	Livro de Ordens Limitadas
LSTM	<i>Long Short-Term Memory</i>
MCTS	<i>Monte Carlo Tree Search</i>
MD	Dados de Mercado
ML	<i>Machine Learning</i>
MLP	<i>Multi-Layer Perceptron</i>
MSE	<i>Mean Squared Error</i>
NLP	<i>Natural Language Processing</i>
NNR	<i>Neural Network Regression</i>
OLHC	Abertura, Mínimo, Máximo, Fechamento
PCA	<i>Principal Component Analysis</i>
PoV	<i>Percentage of Volume</i>
PSO	<i>Particle Swarm Optimization</i>
RBF	<i>Radial Basis Function</i>
RBM	<i>Restricted Boltzmann Machine</i>
RL	<i>Reinforcement Learning</i>
RNN	Rede Neural Recorrente
ROC	<i>Receiver Operating Characteristics</i>
SGD	<i>Stochastic Gradient Descent</i>
SL	Aprendizado Supervisionado
SOR	<i>Smart Order Routing</i>
SVM	<i>Support Vector Machine</i>
TWAP	Preço Médio Ponderado no Tempo
UL	Aprendizado Não-Supervisionado
VWAP	Preço Médio Ponderado por Volume

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>16</b>
<b>1.1 Questão de pesquisa .....</b>	<b>19</b>
<b>1.2 Objetivo.....</b>	<b>19</b>
<b>1.3 Justificativa da Pesquisa.....</b>	<b>19</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>22</b>
<b>2.1 Mercado de Ações .....</b>	<b>22</b>
<b>2.2 Negociação de Ações por Algoritmos .....</b>	<b>24</b>
2.2.1 <i>Contexto.....</i>	24
2.2.2 <i>Bibliometria.....</i>	26
2.2.3 <i>Análise de Riscos e Benefícios.....</i>	30
<b>2.3 Machine Learning .....</b>	<b>35</b>
2.3.1 <i>Contexto.....</i>	35
2.3.2 <i>Bibliometria.....</i>	38
<b>2.4 Deep Learning .....</b>	<b>45</b>
2.4.1 <i>Contexto.....</i>	45
2.4.2 <i>Bibliometria.....</i>	49
<b>2.5 Machine Learning aplicado à Negociação por Algoritmos.....</b>	<b>57</b>
2.5.1 <i>Contexto.....</i>	57
2.5.2 <i>Bibliometria.....</i>	64
<b>2.6 Deep Learning aplicado à Negociação por Algoritmos.....</b>	<b>70</b>
2.6.1 <i>Contexto.....</i>	70
2.6.2 <i>Bibliometria.....</i>	71
2.6.3 <i>Revisão Descritiva da Literatura.....</i>	76
<b>3 MÉTODO .....</b>	<b>86</b>
<b>4 RESULTADOS E DISCUSSÃO.....</b>	<b>88</b>
<b>4.1 Identificação do Problema e Motivação.....</b>	<b>88</b>
<b>4.2 Definição dos Objetivos .....</b>	<b>88</b>
<b>4.3 Projeto e Desenvolvimento .....</b>	<b>88</b>
<b>4.4 Demonstração .....</b>	<b>91</b>
<b>4.5 Avaliação.....</b>	<b>99</b>
<b>4.6 Comunicação .....</b>	<b>101</b>
<b>5 CONCLUSÃO .....</b>	<b>102</b>
<b>REFERÊNCIAS.....</b>	<b>103</b>
<b>APÊNDICE – Logs de execução.....</b>	<b>114</b>
<b>ANEXO – Certificado de Registro de Programa de Computador - INPI.....</b>	<b>119</b>

## 1 INTRODUÇÃO

As empresas que constituem os sistemas produtivos têm entre suas atribuições a produção de bens e serviços (*outputs*) por meio do uso e transformação de insumos (*inputs*), tais como: materiais, capital, trabalho e informações. Para tanto, exercem as três funções básicas de produção: *Marketing*, Produção e Finanças. Sob uma perspectiva mais ampla e considerando-se o conjunto das empresas que integram os sistemas produtivos, o desempenho das funções de produção pode afetar a competitividade e a economia de uma nação (STEVENSON, 2015; HEIZER; RENDER; MUNSON, 2017).

A função de finanças compreende a captação e gerenciamento de recursos financeiros provenientes de investidores, de empréstimos, e da venda de bens, produtos e serviços (KRAJEWSKI; MALHOTRA; RITZMAN, 2016). Responde também pela alocação de tais recursos por toda a organização, orçando e provisionando despesas, analisando propostas de investimentos e provendo recursos para a produção (STEVENSON, 2015).

Jacobs e Chase (2018) destacam, entretanto, a importância estratégica da criação de valor de forma sustentável para os acionistas de uma empresa, baseada no trinômio: Prosperidade Econômica, Responsabilidade Social e Governança Ambiental. Quanto ao aspecto Econômico, a firma deve compensar os acionistas por meio de retornos sobre o investimento. A dimensão econômica, entretanto, vai além de apenas gerar lucro para a empresa, ela deve fornecer benefícios econômicos duráveis para a sociedade.

Para acessar recursos de longo prazo, as empresas podem abrir seu capital por meio de uma oferta pública inicial de ações (IPO), no chamado mercado primário. O custo de capital, cuja estrutura resulta da participação dos acionistas e da emissão de dívidas, é um insumo estratégico para as empresas, afetando diretamente a sua capacidade de investimento e competitividade (DIAMOND, 1967; BLACK; GILSON, 1998; ROSS; WESTERFIELD; JAFFE, 2011).

Empresas de capital aberto divulgam periodicamente seus balanços e demonstrativos e suas ações são livremente negociadas em bolsas de valores no mercado secundário. A divulgação dos dados corporativos e a transparência na formação dos preços das ações permitem aos investidores avaliarem seu valor de mercado, seja por suas perspectivas de crescimento de lucros, geração de caixa, condições macroeconômicas e tendências de mercado (ROSS; WESTERFIELD; JAFFE, 2011; DAMODARAN, 2012).

A negociação de ações, que anteriormente acontecia nos pregões em viva voz, passou a se realizar de forma eletrônica ao longo da década de 1990, período em que se consolidaram, nos Estados Unidos, as primeiras Redes de Comunicação Eletrônica (ECN), como a NASDAQ (*National Association of Securities Dealers Automated Quotations*), e se criaram os meios de execução eletrônica de ordens nas bolsas de valores e futuros tradicionais como NYSE (*New York Stock Exchange*) e CME (*Chicago Mercantile Exchange*) (JOHNSON, 2010).

A partir de 1998, a comunicação eletrônica entre agentes do mercado acionário foi impulsionada pela adoção de um protocolo de mensagens padronizadas, o FIX Protocol (*Financial Information eXchange Protocol*), estabelecido por um consórcio internacional, o FIX Trading Community™, em que participam bolsas de valores, agências reguladoras, instituições financeiras e investidores institucionais (FIX TRADING COMMUNITY, 2018).

A evolução tecnológica e a crescente conectividade entre os participantes do mercado acionário criaram um ambiente propício para o surgimento de estratégias de negociação automatizadas baseadas em algoritmos, ou *Algorithmic Trading* (AT). As instituições do setor passaram a investir de forma consistente em recursos humanos, como engenheiros de *software* e cientistas de dados, bem como em infraestrutura tecnológica, como conexões de baixa latência baseadas em redes de fibras óticas e hardware dedicado ao processamento de alto desempenho (HENDERSHOTT; JONES; MENKVELD, 2011).

Como consequência, surge a *High Frequency Trading* (HFT), uma nova modalidade de AT que busca aproveitar oportunidades de curtíssimo prazo, em escalas de tempo que podem ir de horas a milissegundos, aproveitando discrepâncias momentâneas de preços durante uma sessão de negociação (JOHNSON, 2010; KISSELL, 2014).

Nuti *et al.* (2011) mencionam também os avanços em telecomunicações e computação como impulsionadores do desenvolvimento do AT. Descrevem o HFT como uma modalidade de negociação que busca capturar anomalias transitórias nos preços dos ativos, analisando padrões estatísticos dentro de um mercado ou entre mercados, ou otimizar a execução de ordens de compra ou venda. Em 2009, o HFT alcançou sessenta por cento do volume financeiro negociado no mercado de ações americano. Em 2011, o AT respondia por cerca de sessenta por cento das ações negociadas nos Estados Unidos e Europa.

Nesse contexto, cabe destacar o simpósio do *Institute for Applied Computational Science* (IACS) da *Harvard University*, ocorrido em janeiro de 2017, que discutiu o impacto da computação na atividade econômica, apresentando o processo de transformação de um grande

banco de investimentos americano, que no ano 2000 empregava por volta de 600 operadores de mercado, e em 2017 foram substituídos por dois operadores e mais 200 engenheiros da computação para manter a nova plataforma baseada em sistemas automatizados de negociação. (HARVARD UNIVERSITY-IACS, 2017; MIT TECHNOLOGY REVIEW, 2017).

As primeiras tentativas de utilização da Inteligência Artificial (AI) em sistemas de AT, em especial por meio de abordagens de *Machine Learning* (ML), surgiram ainda em meados da década de 2000 (JOHNSON, 2010).

No mercado de ações, a ampliação do acesso dos investidores ao AT, a equiparação da velocidade de execução de ordens em grande parte da indústria financeira e os resultados das estratégias quantitativas contribuíram para o aumento do interesse sobre abordagens de ML, ainda que vistas com algum ceticismo face aos resultados obtidos pelas primeiras tentativas em anos anteriores (IEEE SPECTRUM, 2017).

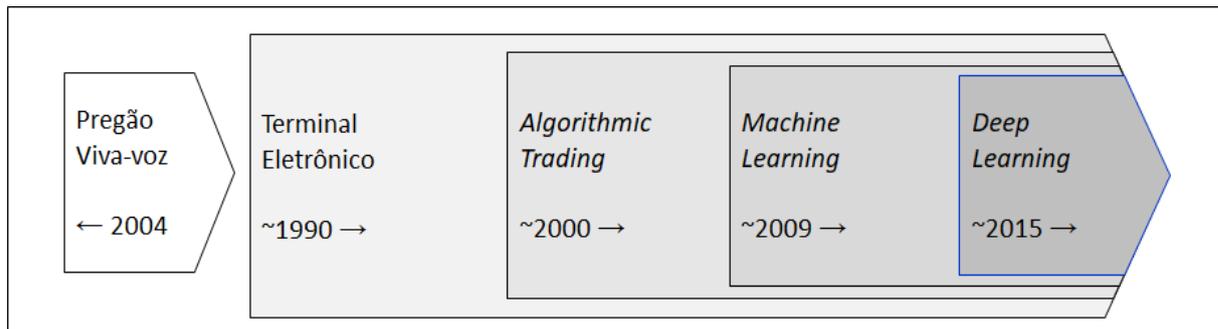
Na segunda metade da década de 2000, novas abordagens de ML baseadas em *Deep Learning* (DL), pesquisadas por Geoffrey Hinton, Yoshua Bengio, Yann LeCun, entre outros, ganharam evidência em áreas como reconhecimento de imagens, processamento natural da linguagem e controle de veículos autônomos, trazendo novas perspectivas para aplicação em outros segmentos. (GOODEFELLOW; BENGIO; COURVILLE, 2016).

Em 2016, o software AlphaGo, baseado em abordagens de DL, ganhou destaque ao derrotar campeões mundiais de um jogo asiático de tabuleiro chamado Go, tido como mais complexo que o xadrez. O AlphaGo utilizou um algoritmo baseado em uma combinação de Redes Neurais Convolucionais (CNN), Monte Carlo *Tree Search* (MCTS) e Aprendizado por Reforço (RL). O software foi desenvolvido pela empresa Deep Mind, hoje pertencente ao grupo Alphabet (Google) (SILVER, 2017).

Cooperaram também para a disseminação dessas abordagens, a utilização de linguagens de programação de código aberto, como Python e R, e a incorporação de bibliotecas de *software* dedicadas ao tratamento de dados e ML, como Numpy, Pandas, Scikit-Learn, Keras, TensorFlow, entre outras, bem como o uso de placas de aceleração gráfica para o processamento algébrico de matrizes de grandes dimensões (CHOLLET, 2018; GÉRON, 2017).

Entretanto, revisões da literatura realizadas no período de 2009 a 2015 apresentavam o DL como campo pouco explorado na área de finanças e menos ainda no mercado de capitais brasileiro (CAVALCANTE *et al.*, 2016). A Figura 1 apresenta o processo de transformação das bolsas de valores, do ponto de vista tecnológico, ao longo das duas últimas décadas.

Figura 1 – Evolução tecnológica das Bolsas de Valores



Fonte: Autor

Considerando-se os impactos e os resultados da pesquisa em DL em diversas áreas do conhecimento, bem como a disponibilização de ferramentas computacionais acima citadas, identifica-se uma oportunidade para a investigação das formulações epistemológicas referentes às abordagens de DL aplicadas ao AT no período de 2016 a 2019.

### 1.1 Questão de pesquisa

Este trabalho busca responder à questão de pesquisa: como aplicar uma abordagem de *Deep Learning* em algoritmos de negociação de ações na bolsa de valores brasileira B3?

### 1.2 Objetivo

O objetivo deste trabalho é pesquisar uma abordagem baseada em *Deep Learning* aplicada à negociação de ações baseada em algoritmos sobre ativos listados na bolsa de valores brasileira, B3, e apresentar um artefato na forma de software que implemente esta abordagem.

### 1.3 Justificativa da Pesquisa

Este trabalho busca cooperar para o desenvolvimento dos setores produtivos pesquisando sistemas de negociação de ações em bolsas de valores baseados em algoritmos, que contribuam para o acesso das empresas ao capital de investimento, que estimulem a liquidez e a transparência das operações, traduzindo-se em redução de incertezas e de custos de transação aos investidores.

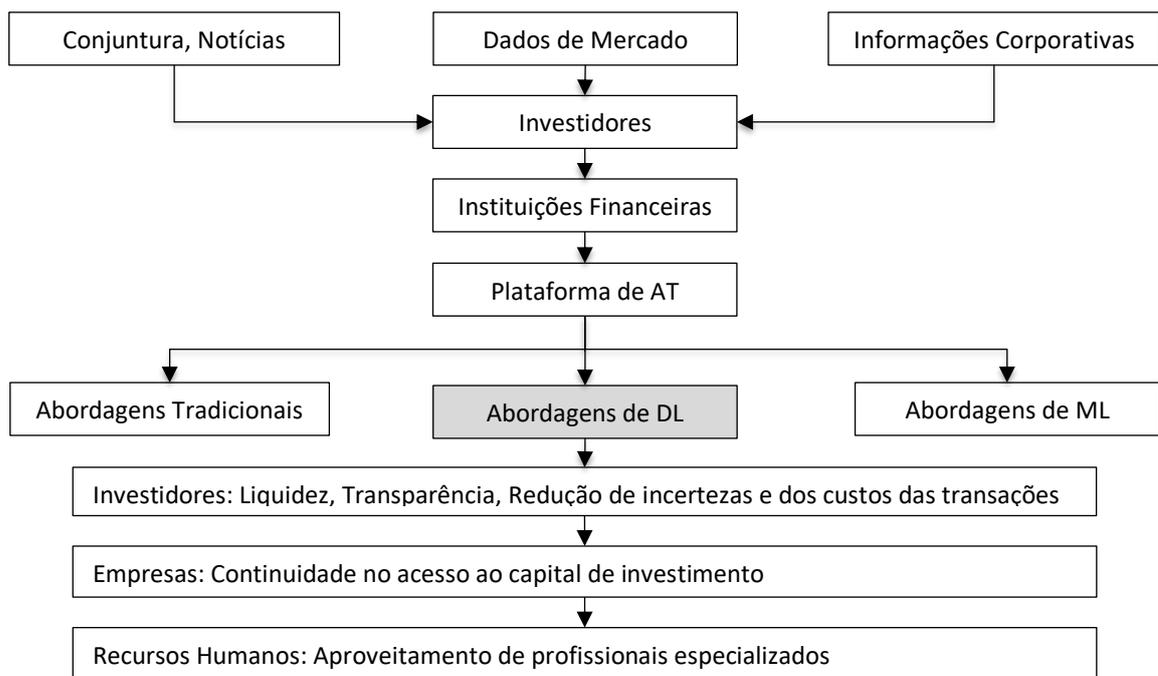
A evolução tecnológica no mercado de capitais implicou em mudanças no perfil dos recursos humanos do setor, outrora baseado em operadores de mercado, hoje operado por engenheiros de *software*, cientistas de dados e profissionais de infraestrutura de TI (HARVARD UNIVERSITY-IACS, 2017; MIT TECHNOLOGY REVIEW, 2017).

As pesquisas em DL, que já consolidaram resultados importantes em diversas áreas do conhecimento e confirmados na literatura científica, destinaram, entretanto, menor atenção ao segmento do mercado financeiro e de capitais, e mais especificamente em aplicações de AT. Busca-se, assim, cooperar para o enriquecimento da massa crítica sobre o tema, investigando esta lacuna do conhecimento da pesquisa recente.

### Contexto da Intervenção

O contexto de negócios em que se insere a pesquisa, o ponto de intervenção (bloco em fundo cinza) e suas finalidades são apresentados na Figura 2.

Figura 2 – Contexto da intervenção e finalidades da pesquisa



Fonte: Autor

## **Estrutura da Dissertação**

No Capítulo 1, INTRODUÇÃO, foram apresentados o contexto da pesquisa, sua orientação em benefício dos sistemas produtivos, as funções de produção, o papel da administração financeira, o impacto da evolução tecnológica sobre instituições financeiras, e o desenvolvimento da negociação de ações por algoritmos. Descreveu-se também o contexto do surgimento da AI, sua evolução para o ML, e o posterior desenvolvimento do DL. Ao final, apresenta-se a questão de pesquisa, o objetivo e a justificativa deste trabalho.

O Capítulo 2, FUNDAMENTAÇÃO TEÓRICA, se inicia apresentando conceitos e características do mercado de ações e do AT. É apresentado o contexto do surgimento da AI, os conceitos de ML, as formas de aprendizado, os modelos de parametrização, e as principais abordagens. Descreve os conceitos de DL, seguidos das abordagens de ML aplicadas ao AT. Os conceitos de AT, ML, DL e aplicações de ML ao AT são acompanhadas das respectivas análises bibliométricas. As abordagens de DL aplicadas ao AT são pesquisadas por meio de análise bibliométrica, seguida de revisão descritiva da literatura identificando os conceitos, características, frequência de ocorrência das abordagens, análise de suas aplicações comparadas às abordagens tradicionais e ao ML. Finalmente apresenta uma análise da abordagem Long Short-Term Memory (LSTM).

O Capítulo 3, MÉTODOS, apresenta o método *Design Science Research* (DSR) utilizado para a pesquisa da abordagem de DL e a implementação do artefato.

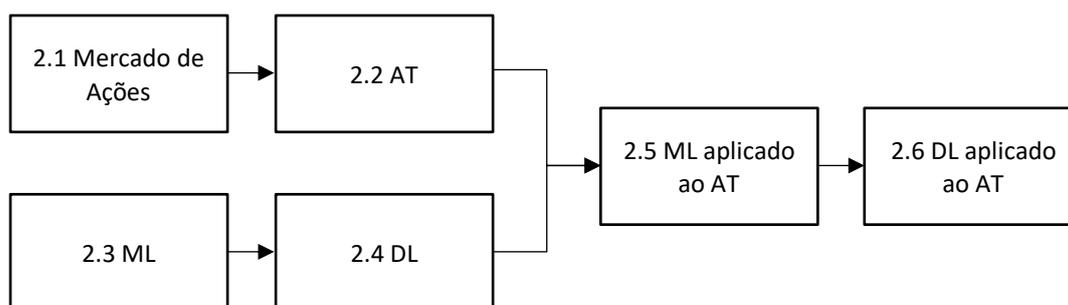
O Capítulo 4, RESULTADOS E DISCUSSÃO, em conformidade com as atividades da pesquisa em DSR, apresenta a configuração, a execução e avaliação de artefato de software baseado no modelo LSTM.

Finalmente, no Capítulo 5, CONCLUSÃO, apresentam-se considerações quanto aos resultados obtidos e contribuições da pesquisa, e são levantadas possibilidades para a continuidade deste trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo se estrutura em: 2.1 Mercado de ações; 2.2 Negociação de Ações por Algoritmos; 2.3 *Machine Learning*; 2.4 *Deep Learning*; 2.5 *Machine Learning* aplicado ao AT; 2.6 *Deep Learning* aplicado ao AT, conforme Figura 3.

Figura 3 – Estrutura do Referencial Teórico



Fonte: Autor

### 2.1 Mercado de Ações

A avaliação do preço de uma ação, um derivativo, ou outro ativo financeiro, dependem de abordagens, interesses e pressupostos de cada investidor, em que, para cada negócio realizado, as partes compradora e vendedora negociam uma quantidade de um ativo a um determinado preço. A abordagem de avaliação de ações conhecida como Análise Fundamentalista parte dos dados sobre o desempenho passado da empresa, como balanços, resultados, fluxo de caixa, múltiplos e avaliações sobre sua administração, para prognosticar ou estimar o resultado futuro e estabelecer um valor teórico para a empresa (SHARPE; ALEXANDER; BAILEY, 1999; DAMODARAN, 2012).

Damodaran (2006) apresenta três abordagens para a avaliação de ações. A avaliação pelo fluxo de caixa descontado, que relaciona o valor de um ativo ao valor presente do fluxo de caixa esperado de um ativo, descontado por uma taxa de juros que reflete o grau de risco ou incerteza da estimativa do fluxo de caixa. A avaliação relativa, que estima o valor do ativo pela análise e comparação de variáveis comuns a outras empresas, como ganhos, fluxos de caixa, valor contábil ou vendas. Esta abordagem, entretanto, assume que o mercado esteja

aproximadamente correto em sua estimativa. A terceira abordagem usa modelos de precificação de opções para mensurar o valor de ativos que compartilham as mesmas características da opção.

A avaliação de ativos desempenha um papel chave em várias áreas das finanças como fusões e aquisições, finanças corporativas e administração de carteiras de investimentos. Deve-se considerar, entretanto, que os ativos caixa-equivalentes são descontados por taxas de juros ajustadas ao risco (custo médio de capital), entretanto deveriam ser descontados por uma taxa mais baixa, por apresentarem um risco menor (DAMODARAN, 2006).

Dentre os indicadores mais utilizados pelos investidores, cabe destacar o P/L, razão do Preço da ação sobre Lucros por ação, que fornece uma noção do nível do preço da ação comparada à sua expectativa de lucros, informando o número de vezes do preço em relação ao lucro esperado. O índice P/L varia substancialmente entre empresas e entre países. Pode ser calculado utilizando os lucros passados, chamado de *trailing*, sobre o lucro atual, *current*, e sobre lucros futuros, *forward* (DAMODARAN, 2006, ROSS *et al.*, 2013).

Por outro lado, a Hipótese de Mercado Eficiente (*Efficient Market Hypothesis*), frequentemente utilizada na precificação de ativos, enuncia que não é possível prever o preço de uma ação com base nos preços passados, e que toda informação conhecida já está refletida no seu preço (FAMA, 1970).

Nesse sentido, a Hipótese do Passeio Aleatório (*Random Walk Hypothesis*) propõe que os preços atuais não dependem dos preços passados, de forma que os padrões não podem ser utilizados simplesmente porque os preços não seguem uma tendência (PEARSON, 1905; MARKOV, 1971).

A Análise Técnica (ou Análise Gráfica), por sua vez, utiliza os padrões de preços exibidos no passado e as séries de tempo para prever o preço de um ativo. Esta abordagem considera que tudo que possa afetar o preço, sejam argumentos fundamentalistas, políticos, psicológicos ou outros quaisquer, já estão refletidos no preço, isto é, que as variações nos preços refletem as mudanças na oferta e demanda do ativo. Desse raciocínio pode-se depreender que os analistas gráficos estão indiretamente avaliando os efeitos das forças econômicas subjacentes, sem se preocuparem com as causas efetivas dos movimentos nos preços. Um aspecto fundamental dessa análise é o pressuposto de que os preços seguem uma tendência, e que ao se identificar um padrão nos estágios iniciais, pode-se conduzir a negociação nessa direção (MURPHY, J., 1999).

Utilizando um enfoque diferenciado, as Finanças Comportamentais estudam a influência de vieses psicológicos sobre as decisões financeiras dos investidores. Os conceitos pesquisados são o comportamento de manada, o efeito de ancoragem, a contabilidade mental e a ilusão da compreensão.

No comportamento de manada, os investidores apresentam uma tendência a pensar e agir de forma similar e, devido a limitações de tempo e cognição, não percebem as informações relevantes (SHILLER, 1995). A ancoragem se refere à tendência de basear fortemente uma decisão em uma informação ou característica parcial, resultando em um ajustamento insuficiente às demais evidências (TVERSKY; KAHNEMAN, 1974). A contabilidade mental é um viés cognitivo que ocorre quando se atribui diferentes valores ou critérios subjetivos de uso para o mesmo montante de dinheiro (THALLER, 1999). O viés da ilusão de compreensão ocorre quando não se consegue deixar de lidar com uma informação limitada ou parcial, como se fosse uma visão completa ou abrangente. Relaciona-se ao efeito Dunning-Krugger, em que um indivíduo, com poucos conhecimentos sobre um assunto, acredita dominar o assunto melhor que indivíduos melhor preparados, levando-o a decisões incorretas (KRUGER; DUNNING, 1999; KAHNEMAN, 2011).

## 2.2 Negociação de Ações por Algoritmos

### 2.2.1 Contexto

Johnson (2010) define o AT como um sistema baseado em regras, responsável pela execução de ordens de compra ou venda de um dado ativo. Utilizando a expressão “Negociação Quantitativa”, Chan (2009) se refere ao AT como uma modalidade de compra e venda de ativos baseada estritamente em algoritmos de computadores, projetada, e por vezes, programada por operadores de mercado. Hilpisch (2018) destaca a disponibilização de grandes massas de dados e do aumento da capacidade computacional a custos decrescentes como promotores do desenvolvimento das plataformas de AT.

Dentre as estratégias de AT mais utilizadas no segmento institucional, destacam-se:

- Negociação de Carteiras de Ações (*Portfolio Trading* ou *Program Trading*): estratégias que operam uma multiplicidade de ativos para o ajuste ou balanceamento de uma carteira, buscando manter conformidade com um índice ou linha de investimentos;

- HFT: estratégias que buscam aproveitar discrepâncias momentâneas de preços; e
- Negociação de Execução (*Execution Trading*): destinado à execução de grandes volumes, minimizando o impacto no mercado e os custos de transação, sendo usualmente operado nas modalidades: Preço Médio Ponderado no Tempo (TWAP), Preço Médio Ponderado por Volume (VWAP), Percentual do Volume (PoV), *Implementation Shortfall* (IS) que administra a diferença entre o preço escolhido e o de execução, e *Smart Order Routing* (SOR) visando à busca por liquidez nos mercados (JOHNSON, 2010).

As arbitragens, estratégias típicas de HFT, se caracterizam pela compra e venda dos mesmos ativos, ou de ativos similares, em mercados distintos por uma diferença vantajosa de preços. Entre as arbitragens, destacam-se as de pares negociados em bolsas distintas, por exemplo: Ação local vs. *American Depositary Receipts* (ADR), de carteira de índice de ações (*index arbitrage*) (REVERRE, 2001), de volatilidade implícita de opções (*implied volatility arbitrage*) (NATENBERG, 1994; TALEB, 1997; WILMOTT, 1998), e de pares de ações no mesmo mercado, também chamada de arbitragem estatística (*statistical arbitrage*), comprado vs. vendido (*long-short*) ou negociação de pares (*pairs trading*) (VIDYAMURTHY, 2004; JOHNSON, 2010; GUO *et al.*, 2017).

Um sistema de AT, de forma geral, é composto de quatro módulos: 1. análise de pré-negociação, 2. geração do sinal de negociação, 3. execução da ordem e 4. análise de pós-negociação (TRELEAVEN; GALAS; LALCHAND, 2013).

1. A análise de pré-negociação refere-se à análise dos dados de mercado dos ativos operados pela estratégia, como livros de oferta, preços e quantidades negociados, situação do ativo e eventos corporativos.

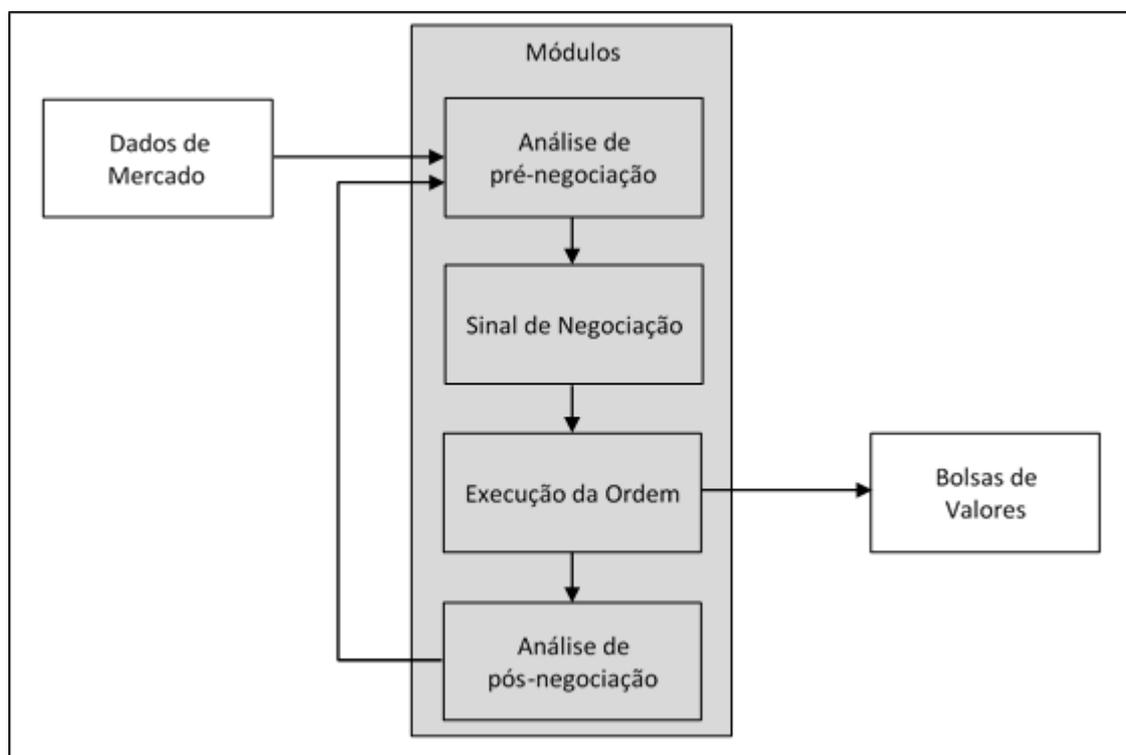
2. A geração do sinal de negociação estabelece os parâmetros das ordens de compra ou venda em função da estratégia adotada, dos dados de mercado, dos custos envolvidos e do retorno esperado.

3. O módulo de execução administra o envio das ordens às bolsas por meio de parâmetros estabelecidos na geração do sinal de negociação, conexões com as bolsas e processos de recuperação das posições em casos de interrupção.

4. A pós-negociação compara os dados da transação aos valores esperados, identifica exposições remanescentes e realimenta o módulo de pré-negociação (NUTI *et al.*, 2011).

Os quatro módulos principais de um sistema de AT são mostrados na Figura 4.

Figura 4 – Principais módulos do sistema de negociação de ações



Fonte: Treleaven, Galas e Lalchand (2013)

### 2.2.2 Bibliometria

Para a mensuração do volume de publicações na literatura científica sobre o AT, as bases Scopus, Web of Science, ACM e IEEE Xplore foram consultadas, conforme critérios apresentados no Quadro 1:

Quadro 1 – Critérios de busca sobre AT

Atributo	Critério
Expressão de Busca	<i>“Algorithmic Trading” OR “High Frequency Trading” OR “Systematic Trading” OR “Quantitative Trading” OR “Automated Trading”</i>
Período	1990 a 2019
Idioma	Inglês
Tipo de Publicação	Artigos de periódicos e conferências
Exclusão de domínios de pesquisa	Artes e Humanidades, Psicologia, Medicina, Agricultura e Biociências, Química e Geociências.

Fonte: autor

A Tabela 1 apresenta o número de documentos retornados pelas bases consultadas:

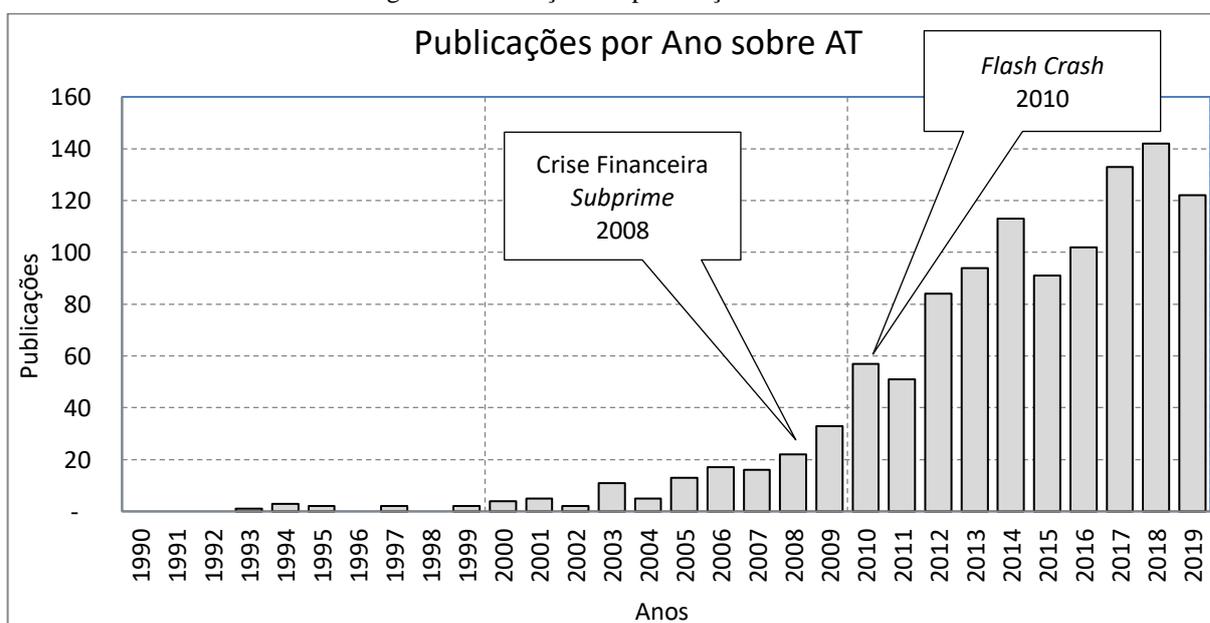
Tabela 1 – Número de publicações das bases consultadas sobre AT

Base	Campo	Publicações
Scopus	Títulos, Resumos e Palavras-chave	1.127
Web of Science	Tópico	875
ACM Digital Library	Títulos, Resumos e Palavras-chave	296
IEEE Xplore	All-Metadata	6.580

Fonte: Scopus (2019), Web of Science (2019), ACM Digital Library (2019), IEEE Xplore (2019)

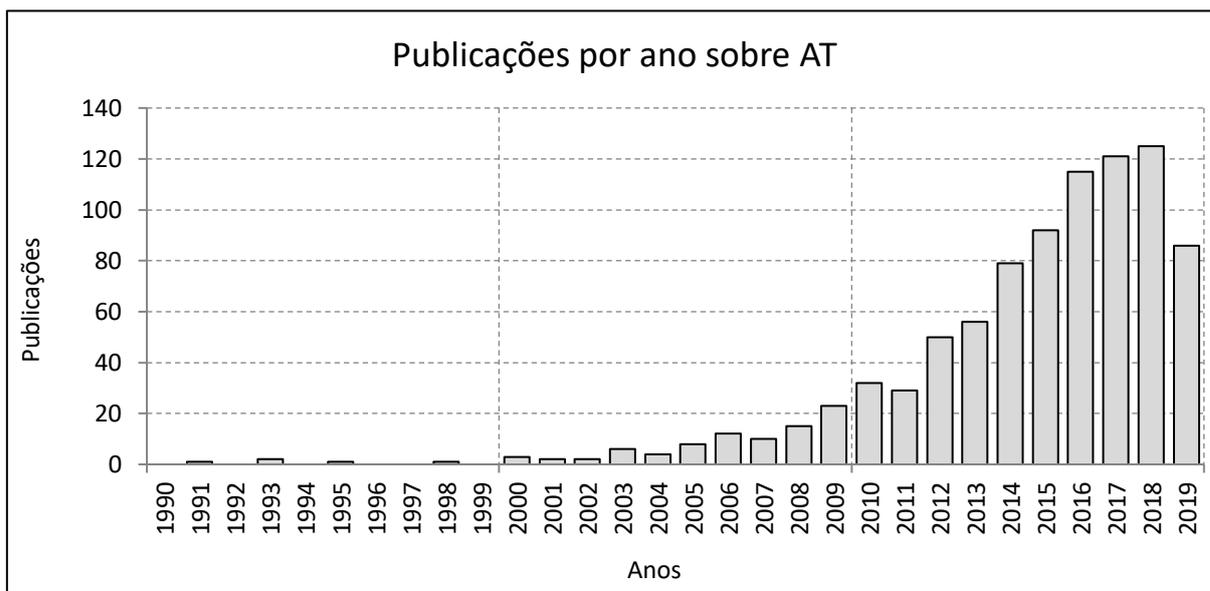
O volume da produção científica sobre AT ao longo dos anos é apresentado no gráfico das bases Scopus e Web of Science, conforme Figura 5 e Figura 6. Observa-se um crescimento no número de publicações, mais notadamente a partir de 2003, ganhando novo impulso a partir de 2009, ano que sucedeu a crise financeira dos títulos *subprime* no mercado americano. Após a ocorrência do *Flash Crash*, a queda abrupta das bolsas americanas ocorrida em maio de 2010, e mais pronunciadamente a partir de 2012, alcança novo patamar, apresentando uma oscilação em 2015 e 2016, e mantendo um crescimento discreto até 2019. O *Flash Crash* será abordado adiante no item de análise de riscos e benefícios do AT.

Figura 5 – Evolução das publicações sobre AT



Fonte: Adaptado de Scopus (2019)

Figura 6 – Evolução das publicações sobre AT



Fonte: Web of Science (2019)

Na Tabela 2 são apresentados os crescimentos percentuais das publicações sobre AT nas bases Scopus e Web of Science, nos períodos de 10 anos, 5 anos e 1 ano.

Tabela 2 – Crescimento das publicações sobre AT

Períodos	10 anos	5 anos	1 ano
Bases	2008 a 2018	2013 a 2018	2017 a 2018
Scopus	545 %	51 %	7 %
Web of Science	733 %	123 %	3 %

Fonte: Scopus (2019) e Web of Science (2019)

Foram identificados também os países com maior número de publicações na base Scopus. A Tabela 3 mostra os dez países que mais publicaram sobre o tema, em que se observa a predominância dos Estados Unidos e Reino Unido, países que concentram os maiores centros financeiros, com bolsas de valores reconhecidamente dentre as mais líquidas e relevantes.

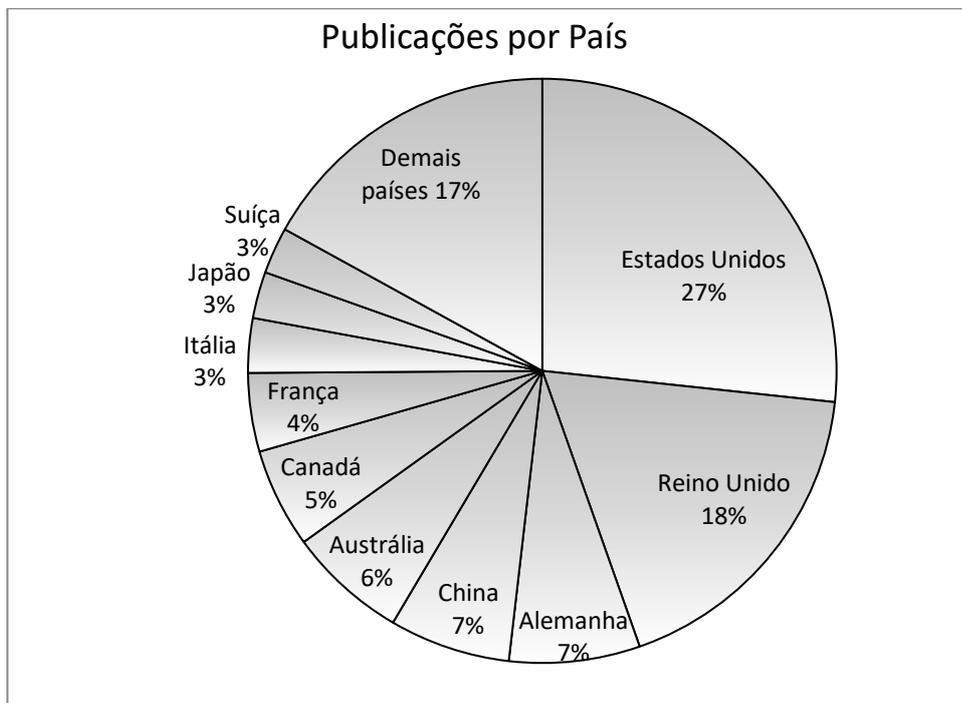
Tabela 3 – Dez Países que mais publicaram sobre AT

Países	Publicações
Estados Unidos	301
Reino Unido	202
Alemanha	81
China	75
Austrália	74
Canadá	62
França	49
Itália	34
Suíça	29
Japão	29
Demais países	191

Fonte: Scopus (2019)

A Figura 7 mostra a participação percentual dos dez países com maior número de publicações, conforme a base Scopus. Os Estados Unidos respondem por mais de um quarto das publicações, enquanto o Reino Unido publica quase vinte por cento da produção científica. Os demais cinco países apresentam participação entre quatro e sete por cento das publicações.

Figura 7 – Dez países que mais publicaram sobre AT.



Fonte: Scopus (2019)

### 2.2.3 Análise de Riscos e Benefícios

Não obstante a quantidade crescente de artigos publicados, o AT tem sido objeto de questionamentos quanto à sua utilidade e riscos inerentes, cabendo, portanto, uma análise sobre seus riscos e benefícios, em especial quanto à liquidez e transparência das negociações.

Para tanto, o resultado da busca de artigos foi listado em ordem decrescente do número de citações. A pré-seleção dos documentos foi realizada mediante leitura dos resumos, conforme os critérios do Quadro 2.

Quadro 2 – Critérios de seleção dos documentos sobre riscos e benefícios do AT

<b>Tipo de Critério</b>	<b>Critério</b>
Inclusão	<p>Conteúdo se concentra nos riscos e benefícios do AT</p> <p>Conteúdo apresenta aspecto relevante e diferenciado sobre riscos e benefícios do AT</p> <p>Contagem de citações &gt; 10 ou FWCI &gt; 10</p>
Exclusão	<p>Conteúdo fora da área de interesse ou marginalmente relacionado à busca</p> <p>Documento não disponível para consulta online</p> <p>Documento duplicado</p>

Fonte: autor

A partir da pré-seleção, os artigos foram lidos para a seleção dos dez mais pertinentes quanto aos benefícios e riscos do AT, e depois listados em ordem decrescente de contagem de citações (CC) e *Field-Weighted Citation Impact* (FWCI), conforme Tabela 4.

Os documentos foram analisados para a apresentação de suas principais conclusões quantos aos riscos e benefícios do AT:

Hendershott, Jones e Menkveld (2011), analisando o efeito da negociação por algoritmos em bolsas de valores, sugerem uma redução dos custos de negociação e um aumento no grau de informação das cotações. Surpreendentemente, com o uso do AT os provedores de liquidez tiveram suas receitas aumentadas, ainda que temporariamente. O AT melhora as ligações entre mercados, de forma que os índices futuros e os ativos subjacentes apresentam menor discrepância de preços. Da mesma forma ocorre uma melhora na liquidez do mercado e na eficiência dos preços das opções sobre ações. Os autores ressaltam, no entanto, que o período de amostra cobriu um período de preços crescentes, restando em aberto a questão de se o suprimento de liquidez seria igualmente benéfico em mercados mais turbulentos ou em

declínio. Outra ressalva se refere à diminuição na profundidade do mercado, isto é, nos níveis de preços das filas de compras e vendas de um livro de ofertas, possivelmente compensada pelo surgimento de centros alternativos de liquidez como LiquidNet e Pipeline, também chamadas de *dark pools*.

Tabela 4 – Artigos selecionados e número de citações sobre riscos e benefícios do AT

<b>Título</b>	<b>Autores</b>	<b>Fonte</b>	<b>Ano</b>	<b>CC</b>	<b>FWCI</b>
<i>Does algorithmic trading improve liquidity?</i>	Hendershott, T.; Jones, C.; Menkveld, A.J.	Journal of Finance	2011	363	25,98
<i>High Frequency Trading and price discovery</i>	Brogaard, J.; Hendershott, T.; Riordan, R.	Review of Financial Studies	2014	168	23,58
<i>Low-latency trading</i>	Hasbrouck, J.; Saar, G.	Journal of Financial Markets	2013	142	17,42
<i>High Frequency Trading and the new market makers</i>	Menkveld, A.J.	Journal of Financial Markets	2013	128	16,19
<i>Algorithmic trading and the market for liquidity</i>	Hendershott, T.; Riordan, R.	Journal of Financial & Quant.Analys.	2013	81	10,26
<i>Rise of the machines: Algorithmic trading in the foreign exchange market</i>	Chaboud, A.P.; Chiquoine, B.; Hjalmarsson, E.; Vega, C.	Journal of Finance	2014	80	11,60
<i>The diversity of High Frequency Traders</i>	Hagströmer, B.; Nordén, L.	Journal of Financial Markets	2013	60	7,99
<i>High Frequency Market Microstructure</i>	O'Hara, M.	Journal of Financial Economics	2015	53	9,49
<i>Moore's law versus Murphy's law: Algorithmic trading and its discontents</i>	Kirilenko, A.A.; Lo, A.W.	Journal of Economic Perspectives	2013	41	4,44
<i>The Flash Crash: High Frequency Trading in an Electronic Market</i>	Kirilenko, A. et al.	Journal of Finance	2017	31	20,61

Fonte: Scopus (2019)

Ainda que possa haver uma associação entre mudanças na latência e a atividade de HFT, não foram encontradas associações mensuráveis entre HFT e os custos de transações (BROGAARD; HENDERSHOTT; RIORDAN, 2014).

Hasbrouck e Saar (2013), ao estudar medidas da atividade de negociação do AT em baixa latência e seus impactos na qualidade do mercado, concluem que há uma melhora na liquidez e na volatilidade de curto prazo. E de particular importância foi evidenciar que mesmo em tempos de queda de preços e ansiedade no mercado, os benefícios sobre a qualidade do mercado se mantêm. Todavia, tais benefícios não se verificam na ocorrência de condições repentinas e severas como o *'Flash Crash'*, quando os grandes participantes se retiraram do mercado. Embora os resultados sugiram que a qualidade do mercado tenha melhorado, acredita-se que a fragilidade episódica dos mercados seja uma questão ainda não resolvida e espera-se que futuras pesquisas possam esclarecê-la.

Menkveld (2013) ressalta, entretanto, que as taxas de negociação (custos) são parte substancial dos lucros e prejuízos das estratégias de HFT. Não surpreende o fato de que as novas bolsas de valores, com seus baixos custos de negociação, tenham atraído uma grande quantidade de operadores de HFT.

A negociação por algoritmos monitora mais ativamente a liquidez do mercado, comparada à atuação humana, respondendo mais rapidamente a mudanças nas condições dos mercados. Em estudos na bolsa de valores da Alemanha (*Deutsche Boerse*), a negociação por algoritmos apresenta a vantagem de consumir liquidez quando os *spreads*, diferenças entre preços de compra e de venda, estão estreitos e fornecer liquidez quando estão largos, reduzindo a volatilidade da liquidez. O aumento da participação do AT no mercado de ações apresenta implicações para órgãos reguladores e para desenvolvedores de plataformas de negociação, no sentido de promover a competição entre os fornecedores de liquidez e a igualdade de condições. (HENDERSHOTT; RIORDAN, 2013).

Em artigo sobre AT no mercado de câmbio, Chaboud *et al.* (2014) encontraram evidências de que o AT reduz oportunidades de arbitragens, atuando sobre ofertas colocadas por operadores não-algorítmicos. Tais resultados são consistentes com a visão de que a negociação por algoritmos melhora a eficiência dos preços, a eficiência das informações e a liquidez do mercado, ainda que com possíveis efeitos adversos nos custos para operadores mais lentos, porém sem causar degradação na qualidade do mercado. A melhoria na eficiência informacional dos preços viria do fato de que o AT refletiria novas informações mais rapidamente. Observaram também que as estratégias de AT são menos diversificadas e mais correlacionadas que as estratégias não algorítmicas. Entretanto não encontraram evidências que tal nível de correlação das estratégias de AT implicaria em aumento de volatilidade.

Utilizando um estudo baseado em mudanças dos intervalos mínimos de preços (*tick*

*size*), verificou-se que o HFT é benéfico para a qualidade do mercado por reduzir a volatilidade de curto prazo (HAGSTROMER; NORDÉN, 2013).

O'Hara (2015), por sua vez, destaca aspectos regulatórios, de transparência e de equidade para lidar com a complexidade crescente dos mercados. Cita a internalização, em que grandes instituições financeiras executam ordens em suas mesas de operações ou nos seus centros de liquidez (*dark pools*) cruzando ordens de diversos clientes, sem passar pelas bolsas de valores, ainda que obedecendo à regulação da melhor oferta de compra e venda (*national best bid and offer*). Tudo isso sugere a necessidade de mais pesquisas orientadas a políticas sobre a microestrutura do mercado.

Quanto aos aspectos regulatórios, Kirilenko e Lo (2013) propõem quatro princípios, referidos como Regulação Financeira 2.0 (*Financial Regulation 2.0*), com a finalidade de atualizar a estrutura regulatória desses mercados:

1. Engenharia de Sistemas: a regulação financeira deve abordar os mercados automatizados, compostos de múltiplos aplicativos de software, equipamentos (*hardware devices*) e recursos humanos, para promover as melhores práticas em projetos de sistemas e gerenciamento de complexidade.
2. Salvaguardas efetivas: o marco regulatório deve estimular a implementação de controles e salvaguardas em vários níveis do sistema, que sejam consistentes com protocolos de comunicação interpretados por máquinas, bem como pela supervisão de humanos.
3. Ampla transparência: controle de versões ao implementar novas regras de negócios, testes e validação de código, com disponibilização para órgãos reguladores e o público.
4. Plataforma neutra: ainda que a regulação estimule a inovação em tecnologia e finanças, deve ser imparcial quanto às especificações de tecnologias como sistemas operacionais, bancos de dados, interfaces de usuário, soluções de *hardware*, e aplicativos de *software* (KIRILENKO; LO, 2013).

Em adição aos dez artigos selecionados, foram identificados documentos mais recentes sobre aspectos regulatórios que ganharam atenção em face de eventos severos e inesperados ocorridos na última década. Quanto a estes aspectos, cabe destacar as preocupações dos órgãos reguladores quanto aos riscos e as vulnerabilidades potenciais dessa modalidade de negociação, tomando-se como exemplo a queda abrupta de 6 de maio de 2010, o *Flash Crash*, quando o índice industrial Dow Jones caiu mais de 900 pontos em cinco minutos (EASLEY; PRADO; O'HARA, 2011).

Entretanto, em estudo sobre o mercado de minicontratos de futuros de S&P 500, antes e depois do *Flash Crash*, quando houve uma grande e momentânea pressão vendedora, as estratégias de HFT não assumiram grandes riscos de posição (*inventories*), mesmo quando deparando-se com grandes desequilíbrios de liquidez (KIRILENKO *et al.*, 2017)

Sob outra perspectiva, Brogaard, Hendershott e Riordan (2017), abordando o banimento da venda a descoberto (*short-sale ban*) durante a crise financeira de 2008, destacam que um componente da atividade do HFT possa ser prejudicial em momentos de grande tensão no mercado, e que a extensão desse dano possa ser diminuída pelo banimento da venda a descoberto, ainda que temporariamente.

Hjalmarsson *et al.* (2017) encontraram forte sustentação para a noção de que estratégias de HFT tendem a se correlacionar. Entretanto, os estudos também sugerem que tais correlações não são desestabilizadoras do mercado, em vez disso, refletem que operadores de HFT estão negociando sob a mesma informação.

Novas evidências surgiram quanto à estabilidade do suprimento de liquidez pelo HFT nos mercados modernos. Ainda que não tenham a função de promover a estabilidade durante momentos de grande oscilação, as estratégias de HFT provaram ser provedoras de liquidez durante movimentos extremos de preços. Os resultados mostram também que o HFT não parece causar episódios de extremo movimento nos preços (BROGAARD *et al.*, 2014).

A Tabela 5 apresenta os principais autores na área de AT citados na busca, o número de artigos selecionados, a afiliação, o índice-h e o número total de publicações, ordenado por número de artigos selecionados e por h-index.

Tabela 5 – Principais autores sobre riscos e benefícios do AT, artigos, afiliação e índice h

<b>Autores</b>	<b>Artigos</b>	<b>Afiliação</b>	<b>h-index</b>
Hendershott, T.	5	University of California, Berkeley, Berkeley, United States	18
Menkveld, A.J.	4	Tinbergen Institute - TI, Research Centre for Economic Policy, Rotterdam, Netherlands	12
Brogaard, J.	4	University of Washington, Seattle, Foster School of Business, Seattle, United States	5
Riordan, R.	3	Queen's University, Kingston, School of Business, Kingston, Canada	7
Kirilenko, A. A.	3	MIT Sloan School of Management, Cambridge, United States	7
O'Hara, M.	2	Cornell University, Samuel Curtis Johnson Graduate	37

<b>Autores</b>	<b>Artigos</b>	<b>Afiliação</b>	<b>h-index</b>
		School of Management, Ithaca, United States	
Treleaven, P.	2	UCL, Department of Computer Science, London, United Kingdom	18
Nordén, L.	2	Stockholm Business School, Stockholm, Sweden	15
Hjalmarsson, E.	2	Goteborgs Universitet, Department of Economics, Gothenburg, Sweden	8
Hagströmer, B.	2	Stockholm Business School, Stockholm, Sweden	5
Lo, A.W.	1	MIT Sloan School of Management, Cambridge, United States	36
Jarrow, R.A.	1	Kamakura Corporation, Honolulu, United States	32
Hasbrouck, J.	1	Leonard N. Stern School of Business, New York, United States	18
Jones, C.M.	1	Columbia Business School, New York, United States	15
Chai, S.	1	Ewha Woman University, Seoul, South Korea	8
Chaboud, A.P.	1	Federal Reserve System, Division of International Finance, Washington, United States	7
Chiquoine, B.	1	Stanford University, Palo Alto, United States	3
Hoffmann, P.	1	European Central Bank, Frankfurt am Main, Germany	2

Fonte: Scopus (2019)

## 2.3 *Machine Learning*

### 2.3.1 *Contexto*

As primeiras iniciativas em AI remontam a meados do século XX com a criação do primeiro neurônio artificial por McCulloch e Pitts em 1943, e a construção do primeiro mecanismo baseado em Redes Neurais Artificiais (ANN) por Marvin Minsky e Dean Edmonds na universidade de Harvard em 1950. Nesse mesmo ano Alan Turing apresentaria os conceitos de ML, o Teste de Turing, os Algoritmos Genéticos (GA), e o RL em seu artigo “*Computing Machinery and Intelligence*” (TURING, 1950; RUSSELL; NORVIG, 2016).

Em 1956, John McCarthy e outros pesquisadores organizaram um workshop no Dartmouth College que contribuiu para o reconhecimento da AI como um campo de pesquisa, tendo como integrantes acadêmicos do MIT, Carnegie Mellon, Stanford e IBM (RUSSELL; NORVIG, 2016).

Vale destacar a contribuição de Frank Rosenblatt e colaboradores ao desenvolver em 1958, em Cornell, o *neurocomputer*, uma implementação de redes neurais baseada em

*perceptrons*, que reconheceu algarismos a partir de uma imagem de 20 por 20 *pixels* por meio de um sensor (ROSENBLATT, 1958; KRIESEL, 2005).

No ano seguinte, Samuel (1959) apresentou uma abordagem utilizando ML sobre o conhecido jogo de tabuleiro *checkers* (jogo de damas), em que após um ‘treinamento’ de oito a dez horas, o programa superou o desempenho de um jogador mediano.

Cerca de duas décadas mais tarde, Bellman (1978) definiria a AI como a automação de atividades associadas ao pensamento humano, como a tomada de decisão, solução de problemas e aprendizado. Kurzweil (1990) viria a descrever a AI como a arte de criar máquinas que realizam funções que requerem inteligência quando realizadas por pessoas.

O ML foi descrito por Mitchell (1997) como um programa de computador que melhora seu desempenho em alguma tarefa por meio da experiência obtida a partir de grandes massas de dados como registros de manutenção de equipamentos, transações financeiras, registros médicos, entre outros; e que se pode definir como:

“Um programa de computador aprende com a experiência E com relação a alguma classe de tarefa T e medida de desempenho P, se seu desempenho nas tarefas em T, na forma medida por P, melhora com a experiência E.”  
(MITCHELL, 1997).

Para Alpaydin (2011) ML é a programação de computadores utilizada para a otimização de um critério de performance usando dados de exemplos ou experiência passada. Kelleher, Mac Namee e D’Arcy (2015) definem ML como um processo automatizado para extrair padrões a partir de dados.

As primeiras implementações convencionais de ML apresentavam desempenho pouco satisfatório devido a limitações na capacidade de processamento de dados. A construção de um sistema de reconhecimento de padrões exigia o conhecimento especializado da área de aplicação para se projetar um extrator de características que transformasse dados brutos em uma representação interna adequada, como um vetor de características a partir do qual se pudesse detectar ou classificar padrões (LECUN; BENGIO; HINTON, 2015).

Quanto à **Interação com os Dados**, as formas mais usuais de aprendizado são:

**Aprendizado Supervisionado (SL):** os dados de entrada contêm propriedades associadas a rótulos (*labels*) ou objetivos (*targets*). (GOODFELLOW; BENGIO; COURVILLE, 2016). Durante a fase de treinamento, o algoritmo analisa exemplos de dados, na forma de pares de entrada e saída, e ‘aprende’ uma função que mapeia a saída a partir da

entrada de dados (MURPHY, K. P., 2012). O aprendizado é a busca de uma função no espaço de soluções possíveis e que apresente bom desempenho sobre novos exemplos além do conjunto inicial. Diz-se que a função é generalizável se corretamente predizer a saída para novos exemplos de entrada. Por vezes essa função não é estritamente uma função da entrada, podendo ser estocástica. Nesse caso o processo de aprendizado retornará uma distribuição condicional de probabilidades. Quando a saída é um conjunto de valores finitos, o processo de aprendizado é chamado de classificação, e quando o conjunto é de apenas dois valores, é chamado de Booleano ou binário. Quando saída é uma variável contínua, o aprendizado é chamado de regressão. (RUSSELL; NORVIG, 2016).

**Aprendizado Não-Supervisionado (UL):** algoritmos de UL analisam conjuntos de dados contendo propriedades (*features*) e aprendem tais propriedades a partir da sua estrutura, como por exemplo, da distribuição de probabilidades que gerou esses dados, ou da divisão dos dados em agrupamentos (*clusters*) que apresentem similaridades (GOODFELLOW; BENGIO; COURVILLE, 2016). No UL não se fornece um retorno a um estímulo (*feedback*) explícito ou uma resposta correta a partir da qual a rede neural possa ser treinada. O UL é mais característico do aprendizado humano e animal, sendo mais amplamente aplicável do que o SL por não requerer um especialista humano para rotular manualmente os dados, o que tornaria o tratamento de dados mais dispendioso, e com relativamente pouca informação adicionada (MURPHY, K. P., 2012). Os filtros de *spam* são exemplos de UL, em que se detectam anomalias a partir de uma massa de dados de e-mails não classificados (SHALEV-SHWARTZ; BEN-DAVID, 2014). O Agrupamento (*clustering*) é um exemplo típico de UL, em que se procura agrupar objetos ou dados similares. A ideia é estimar a distribuição sobre o número de grupos, isto é, avaliar a existência de subpopulações dentro dos dados. Existem dois tipos de entradas, a baseada em similaridade e a baseada em características. Quanto às saídas existem dois tipos: o particionado, em que os objetos são colocados em grupos desconexos, e o hierárquico, em que se criam árvores aninhadas de partições (MURPHY, K. P., 2012).

**Reinforcement Learning (RL):** o processo de aprendizado é um agente de tomada de decisão que executa ações em um ambiente e é recompensado ou penalizado por suas ações na tentativa de resolver um problema. Após executar um conjunto de tentativas e erros, o agente deveria identificar a sequência de ações que maximiza as recompensas (ALPAYDIN, 2011). No RL o algoritmo interage com um ambiente, de forma que haja um ciclo de *feedback* entre o aprendizado e suas experiências (GOODFELLOW; BENGIO; COURVILLE, 2016). O algoritmo é treinado, ou ‘aprende’, a partir de uma série de reforços, seja por recompensa ou

penalidade (RUSSELL; NORVIG, 2016).

### **Modelos Paramétricos e Não-Paramétricos**

Um modelo é chamado de paramétrico quando assume uma distribuição de probabilidades com um número fixo de parâmetros. Quando o número de parâmetros não é fixo e aumenta com o volume de dados de treinamento, é chamado de não-paramétrico. Modelos paramétricos têm a vantagem de serem de uso mais imediato, mas apresentam a desvantagem de requerer suposições mais fortes sobre a natureza da distribuição de dados. Modelos não-paramétricos são mais flexíveis, mas frequentemente de difícil tratamento computacional para grandes conjuntos de dados (MURPHY, 2012).

Dentre os modelos paramétricos, a Regressão Linear é uma das abordagens de SL mais utilizadas, cuja saída é expressa por uma função linear das entradas (MURPHY, 2012).

Os classificadores *Naive Bayes* pertencem à família de classificadores probabilísticos paramétricos. Este modelo assume que os eventos condicionais são independentes. Embora esse pressuposto nem sempre seja verdadeiro, ele simplifica sensivelmente a estimação. Não obstante, os classificadores *Naive Bayes* apresentam desempenho por vezes melhor que alternativas mais sofisticadas (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

O *K-Nearest Neighbors* (KNN) é um exemplo de classificador não-paramétrico, que busca os K pontos no conjunto de dados de treinamento que estão mais próximos da entrada de teste, conta quantos membros de cada classe estão nesse conjunto, e retorna uma fração empírica como estimativa. (MURPHY, K. P., 2012).

As *Decision Trees* (DT), também classificadas como modelos não-paramétricos, situam-se entre as formas mais usuais de ML. Consistem em funções que, a partir de um vetor de atributos, retornam uma decisão, isto é, um valor de saída. Uma DT chega a uma decisão pela realização de uma sequência de testes, em que cada nó da árvore corresponde a um teste do valor de atributo de entrada e a um valor de saída. Assim, a DT pode ser útil para uma grande variedade de problemas. Entretanto, para funções que não podem ser representadas de forma concisa, o conjunto de nós de decisão pode alcançar valores excessivamente altos, tornando impraticável a sua implementação (RUSSELL; NORVIG, 2016).

#### 2.3.2 *Bibliometria*

As bases foram novamente consultadas para prospectar a produção científica sobre ML,

compreendendo artigos científicos publicados por periódicos ou conferências. Os critérios de busca são apresentados no Quadro 3:

Quadro 3 – Critérios de busca para ML

<b>Atributo</b>	<b>Critérios</b>
Expressão de busca	"Machine Learning"
Período	1990 a 2019
Idioma	Inglês
Tipo de Publicação	Artigos publicados em periódicos e conferências

Fonte: Autor

A Tabela 6 apresenta o número de documentos retornados pelas bases consultadas:

Tabela 6 – Publicações sobre ML

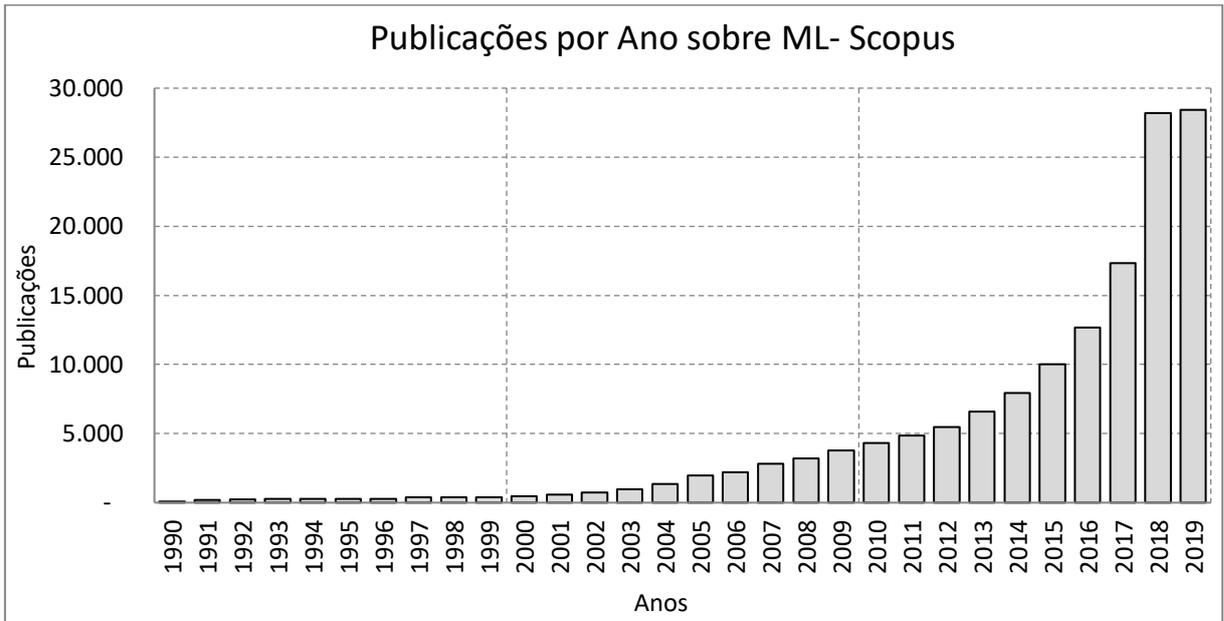
<b>Base</b>	<b>Campos</b>	<b>Publicações</b>
Scopus	Títulos, Resumos e Palavras-chave	146.668
Web of Science	Tópico	103.381
ACM Digital Library	Títulos, Resumos e Palavras-chave	71.676
IEEE Xplore	All-Metadata	66.694

Fonte: Scopus (2019), Web of Science (2019), ACM Digital Library (2019), IEEE Xplore (2019)

Com os arquivos de dados consolidados gerados pelas bases Scopus e Web of Science, foram elaborados gráficos, conforme Figura 7 e Figura 8, que apresentam a evolução da produção científica sobre ML, em número de publicações, ao longo dos anos.

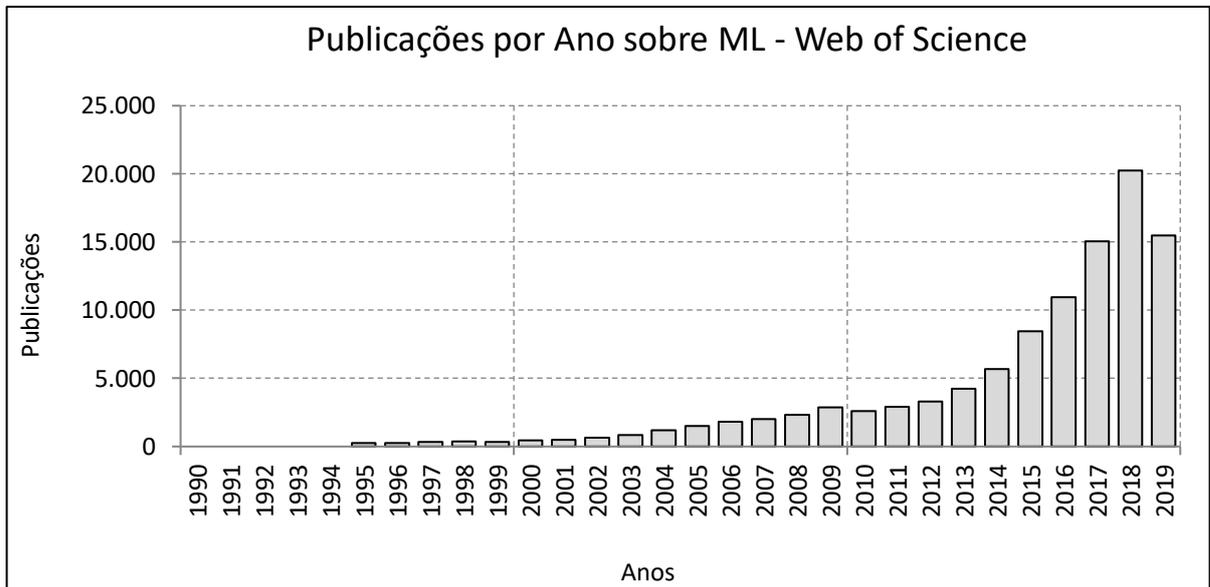
Nestes gráficos se observa um discreto crescimento no número de publicações a partir da década de 2000, tornando-se mais pronunciado na década de 2010, e mais acentuado no ano de 2018. Quanto ao ano de 2019, deve-se considerar que os dados são parciais, acumulados até o mês de setembro. Ainda assim se verifica que o patamar do número de publicações se mantém. Constata-se assim um interesse relevante em volume de publicações e consistente ao longo desta década.

Figura 8 – Publicações por Ano sobre ML (Scopus)



Fonte: Scopus (2019)

Figura 9 – Publicações por Ano sobre ML (Web of Science)



Fonte: Web of Science (2019)

Na Tabela 7 são apresentados os crescimentos percentuais das publicações nas bases Scopus e Web of Science, nos períodos de 10 anos, 5 anos e 1 ano.

Tabela 7 – Crescimento das publicações sobre ML nas bases Scopus e Web of Science

Bases	Períodos	10 anos	5 anos	1 ano
		2008 a 2018	2003 a 2018	2017 a 2018
Scopus		778 %	328 %	63 %
Web of Science		766 %	376 %	35 %

Fonte: Scopus (2019) e Web of Science (2019)

A consulta à base Scopus possibilitou a identificação dos países com maior número de publicações sobre ML. A Tabela 8 mostra os 10 países que mais publicaram. Verifica-se a liderança dos Estados Unidos, com a China em segundo, seguidos da Índia, Reino Unido e Alemanha.

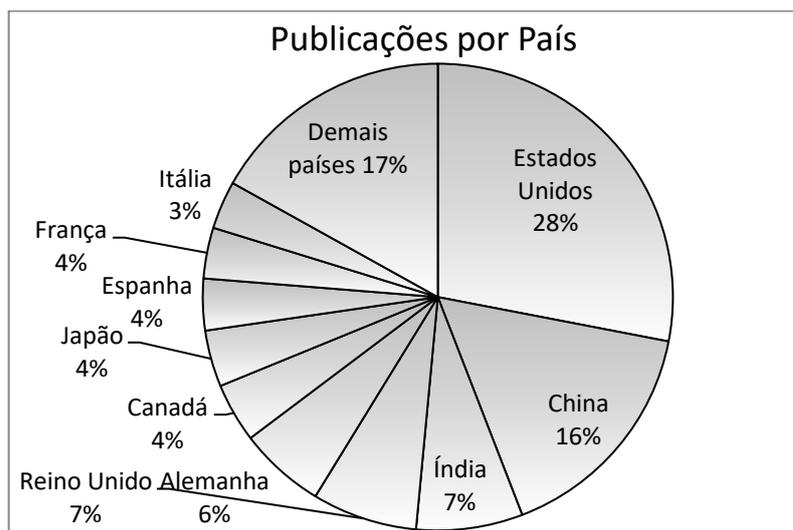
Tabela 8 – Dez Países que mais publicaram sobre ML

Países	Publicações
Estados Unidos	41.137
China	23.628
Índia	10.743
Reino Unido	10.616
Alemanha	8.739
Canadá	6.022
Japão	5.737
Espanha	5.251
França	5.169
Itália	4.873
Demais países	24.753

Fonte: Scopus (2019)

Na Figura 10, visualiza-se a participação percentual dos dez países que mais publicaram, também conforme dados da base Scopus.

Figura 10 – Dez países que mais publicaram sobre ML.



Fonte: Scopus (2019)

A Tabela 9 apresenta dez publicações com conteúdo pertinente ao ML, dentre as mais citadas, conforme base Scopus. Para os documentos selecionados, é fornecido também o FWCI.

Tabela 9 – Dez publicações com maior número de citações sobre ML

Título	Autores	Fonte	Ano	CC	FWCI
Random forests	Breiman L.	Machine Learning	2001	30.407	24,33
LIBSVM: A Library for support vector machines	Chang C. C., Lin C.-J.	ACM Transactions Intelligent Systems	2011	18.069	708,13
Deep learning	Lecun Y., Bengio Y., Hinton G.	Nature	2015	11.756	129,91
Gradient-based learning applied to document recognition	LeCun Y., Bottou L., Bengio Y., Haffner P.	Proceedings of the IEEE	1998	11.657	7,41
Scikit-learn: Machine learning in Python	Pedregosa F., <i>et al.</i>	Jrnl. Machine Learning Research	2011	10.629	88,28
Rapid object detection using a boosted cascade of simple features	Viola P., Jones M.	IEEE Computer Vision and Pattern Recognition	2001	10.266	53,32
An introduction to ROC analysis	Fawcett T.	Pattern Recognition Letters	2005	7.927	73,73
Dropout: A simple way to prevent neural networks from overfitting	Srivastava N., <i>et al.</i>	Journal of Machine Learning Research	2014	7.403	308,43
Maximum entropy modeling of species geographic distributions	Phillips S.J., Anderson R.P., Schapire R.E.	Ecological Modelling	2005	6.775	39,49
Distributed optimization and statistical learning via the alternating direction method of multipliers	Boyd S., <i>et al.</i>	Foundations and Trends in Machine Learning	2010	6.415	28,78

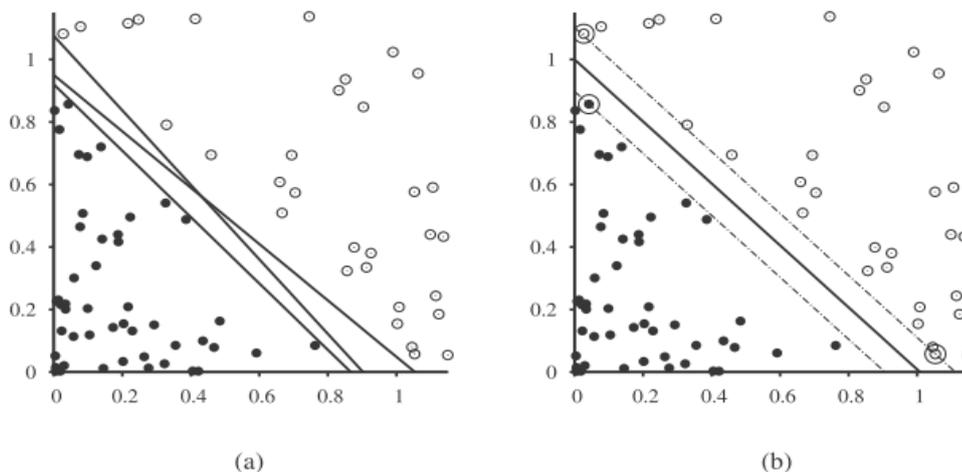
Fonte: Scopus (2019)

Dentre os documentos listados, vale destacar o artigo *Random Forests* de Breiman (2001), com mais de trinta mil citações. As *Random Forests* são uma combinação de preditores ou classificadores estruturados em árvore, em que cada uma depende dos valores de um vetor aleatório independentemente amostrado e com a mesma distribuição por todas as árvores da floresta. Melhorias significativas na precisão da classificação são resultado do crescimento de uma combinação de árvores, que votam para a classe mais popular. Por não apresentarem *overfitting*, as *Random Forests* são ferramentas de predição efetivas. Tornam-se classificadores e regressores precisos com uma apropriada injeção de aleatoriedade. Encontram aplicações em diagnósticos médicos e recuperação de documentos com precisão melhorada comparada a classificadores em árvore. *Overfitting* é uma expressão da estatística usada para designar a situação em que um modelo estatístico se ajusta muito bem a um conjunto de dados, mas não se ajusta ou não apresenta efetiva capacidade de previsão a novos dados (BREIMAN, 2001).

Sobre a *Support Vector Machine* (SVM), o artigo de Chang e Lin (2011), “*LIBSVM: A Library for support vector machines*”, alcançou fator de impacto acima de 700 figurando na lista dos dez mais citados artigos em três das quatro bases consultadas. O *Support Vector Machine* (SVM) é um algoritmo de ML muito utilizado para classificação, regressão, e outras formas de aprendizado, cuja principal ideia é separar duas classes pela escolha de um hiperplano que maximize a margem entre os pontos de treinamento das duas classes (CHANG; LIN, 2011).

O SVM é de especial utilidade quando não se têm conhecimento prévio sobre um domínio, tornando-o uma das abordagens mais utilizadas de SL, em especial por suas propriedades de: construir um separador de margem máximo – uma fronteira de decisão com a maior distância possível aos pontos de exemplos; criar um hiperplano de separação linear; e ter flexibilidade para representar funções complexas, combinando as vantagens dos modelos paramétricos e não-paramétricos. Na Figura 11 apresenta-se a Classificação de SVM em: (a) duas classes de pontos (círculos brancos e pretos) e três separadores lineares candidatos. (b) O máximo separador de margem (linha grossa), está no centro da margem (área entre as linhas tracejadas). Os vetores de suporte (pontos com círculos maiores) são exemplos mais próximos ao separador (HASTIE; TIBSHIRANI; FRIEDMAN, 2009; RUSSELL; NORVIG, 2016).

Figura 11 – Classificação de Máquinas de Vetores de Suporte



Fonte: Russell e Norvig (2016)

Com a finalidade de facilitar o uso e aplicação de SVM, Chang e Lin (2011) desenvolveram a biblioteca de software LIBSVM, de grande utilidade em ML, solucionando problemas de otimização, convergência, classificação multiclasse, estimativas de probabilidade e seleção de parâmetros.

Cabe menção especial ao artigo “*Deep Learning*” de Hinton, Bengio e LeCun (2015), autores precursores da pesquisa sobre DL.

LeCun *et al.* (1998) destacam o reconhecimento de documentos baseados em redes neurais multicamadas com algoritmo de *backpropagation* como o exemplo mais bem sucedido, então, de técnica de aprendizado por gradiente. Algoritmos de aprendizado baseados em gradiente, considerando-se uma arquitetura de rede adequada, podem ser usados para sintetizar uma superfície de decisões complexas que possam classificar padrões de grandes dimensões com pré-processamento mínimo. Vários métodos para reconhecimento de caracteres são comparados à tarefa de reconhecimento de dígitos manuscritos padrão. As redes convolucionais superam as demais técnicas ao lidar com imagens bidimensionais. Sistemas de reconhecimento de documentos reais são compostos de módulos como extração de campo, segmentação, reconhecimento e modelagem da linguagem. A *Graph Transformer Network* (GTN) permite que tais sistemas sejam treinados usando métodos baseados em gradiente que minimizam uma medida geral de desempenho (LECUN *et al.*, 1998).

Pedregosa *et al.* (2011) apresentam uma avaliação sobre a biblioteca Scikit-learn que integra uma ampla variedade de algoritmos de ML no estado da arte voltados ao aprendizado supervisionado e não-supervisionado. O módulo é incorporado à linguagem Python com ênfase

em facilidade de uso, desempenho, documentação e consistência da API, e distribuído sob a licença simplificada BSD, estimulando o uso acadêmico e comercial.

Sobre o reconhecimento de imagens, Viola e Jones (2004) apresentam uma abordagem de ML para detecção de objetos que processam imagens de forma extremamente rápida, alcançando altas taxas de detecção. Nesse estudo distinguem-se três contribuições principais: uma nova representação de imagens chamada *Integral Image*, um algoritmo de aprendizado baseado no *AdaBoost* que seleciona um pequeno número de características visuais críticas de um amplo conjunto e retorna classificadores eficientes; e um método para combinar classificadores complexos em sequência que descarta imagens de fundo enquanto destina o processamento às regiões-objeto.

Fawcett (2005) apresenta as técnicas de gráficos *Receiver Operating Characteristics* (ROC), para visualização, organização e seleção de classificadores baseadas em seu desempenho, de uso crescente em ML e mineração de dados. O ROC vem, há muito, sendo usado em teoria de detecção de sinal para representar o balanço entre a taxas de acertos e de falsos positivos.

Phillips, Anderson e Schapire (2005) abordam o método da entropia máxima (Maxent), uma abordagem de ML de propósito geral com formulação matemática simples e precisa, que dispõe de características que a tornam adequadas a modelagem de distribuição de espécies.

Boyd *et al.* (2010) apresentam um amplo estudo sobre otimização convexa, destacando a necessidade de soluções de problemas com grande número de características e exemplos de treinamento, devido à explosão em tamanho e complexidade dos conjuntos de dados modernos. Propõem o *Alternating Direction Method of Multipliers* para a otimização convexa distribuída, e em particular para problemas de larga escala em estatística, ML e áreas relacionadas.

Os artigos “*Deep Learning*”, de LeCun, Bengio e Hinton (2015) e “*Dropout: A Simple Way to Prevent Neural Networks from Overfitting*”, de Srivastava *et al.* (2014), são abordados na seção 2.5 *Deep Learning*.

## 2.4 *Deep Learning*

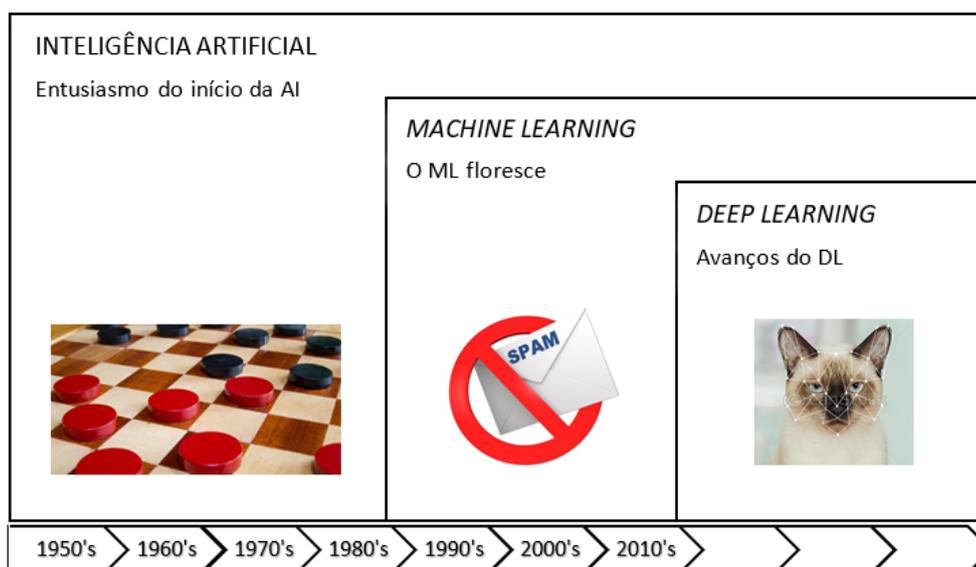
### 2.4.1 *Contexto*

Dentre as abordagens de ML, o DL, a partir da segunda metade da década de 2000,

apresentou melhorias significativas de desempenho em diversas aplicações cognitivas como o *Natural Language Processing* (NLP), os aplicativos de tradução e o reconhecimento de imagens (SCHUSTER *et al.*, 2016).

A Figura 12 mostra a evolução da AI, inicialmente superando humanos em partidas de damas (*checkers*); o desenvolvimento o ML, tendo como exemplo o algoritmo de filtragem de spam em contas de e-mail; e o surgimento da DL, utilizado no reconhecimento de imagens.

Figura 12 – A evolução da Inteligência Artificial.

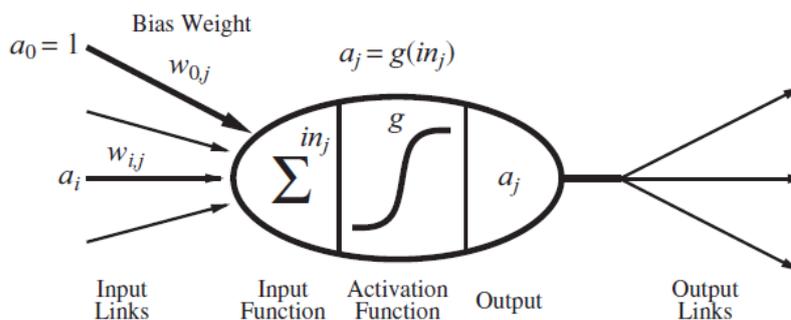


Fonte: adaptado de Nvidia (2019).

### Redes Neurais Artificiais (ANN)

Inspirada no funcionamento das células neurais, a ANN é um conjunto de unidades neurais artificiais conectadas, em que cada unidade se baseia em uma representação matemática de um neurônio, na qual se combina um conjunto de entradas, coordenadas por uma função de entrada, uma função de ativação e um conjunto de saídas. Em uma ANN, assim que uma combinação linear dessas entradas atinge ou ultrapassa um determinado limite, o componente neural dispara um sinal de saída. A Figura 13 mostra o modelo matemático de uma unidade neural, ou *perceptron*. A unidade de ativação de saída é uma combinação linear de pesos e um conjunto de entradas (RUSSELL; NORVIG, 2016).

Figura 13 – Modelo matemático de uma unidade neural.



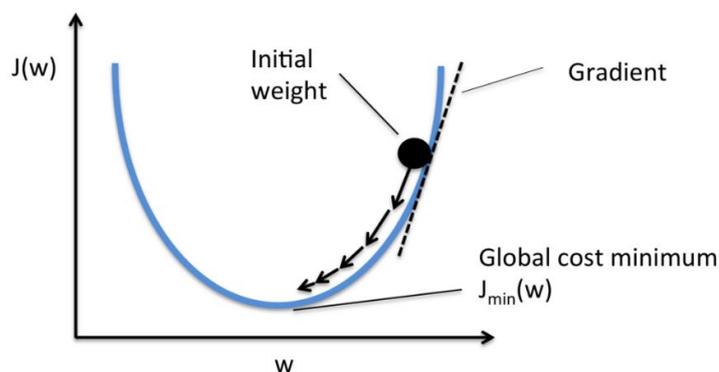
Fonte: Russell e Norvig (2016)

Para o treinamento da rede neural, executa-se uma função de perda que mede o erro entre os resultados de saída e os esperados. O algoritmo então ajusta seus parâmetros internos, os pesos, para reduzir o erro. O ajuste de parâmetros é realizado por vetor gradiente da função de perda, que para cada peso indica quanto o erro aumenta ou diminui em função de pequenos ajustes no peso. O vetor de pesos é ajustado então na direção oposta ao vetor gradiente, o qual indica a direção em que a função de perda atinge o mínimo de forma mais rápida. (LECUN; BENGIO; HINTON, 2015).

**Backpropagation (BP):** este método retro transmite os dados da função de perda na rede neural, partindo da saída para a entrada, com a finalidade de construir o vetor gradiente da função. A função de perda é uma medida de desempenho da rede neural, em geral obtida pela soma dos quadrados das diferenças entre as saídas da rede neural em treinamento e seus valores esperados. O BP executa a regra da cadeia (*chain rule*) de forma recursiva. A regra da cadeia é utilizada para computar as derivadas das funções formadas pela composição de outras funções cujas derivadas são conhecidas (GOODFELLOW; BENGIO; COURVILLE, 2016).

**Stochastic Gradient Descent (SGD):** os procedimentos de busca do mínimo da função de perda para o treinamento de uma rede neural, como o SGD, são os mais usuais. Como procedimento, escolhe-se um ponto inicial e, pela função de inclinação, primeira derivada da função, move-se na direção do mínimo mais próximo. A Figura 14 expressa o conceito de primeira derivada para a busca do mínimo da função de perda. Os pesos são ajustados pela nova posição na função de perda e obtém-se, ao final do processo, a menor diferença possível entre o valor de saída e o valor esperado (GOODFELLOW; BENGIO; COURVILLE, 2016).

Figura 14 – Busca do mínimo de uma função



Fonte: Suryansh (2012)

As primeiras abordagens para o processamento computacional do BP eram de difícil implementação, com massiva operação de tensores e consumo excessivo de memória. Entretanto a partir de abordagens propostas por Rumelhart, Hinton e Williams (1986), foi possível viabilizar o processo de derivação e o processamento do SGD, aplicando-se técnicas como o *Automatic Differentiation*, utilizadas para calcular derivadas de forma algorítmica (GOODFELLOW; BENGIO; COURVILLE, 2016).

**Convolação:** Segundo Goodfellow, Bengio e Courville (2016), na sua forma geral, a convolação é uma operação linear sobre duas funções de variáveis reais, em que o primeiro termo de entrada é referido como *input*, o segundo termo tem a característica de uma ponderação, por exemplo, uma função densidade de probabilidade, referido como *kernel*, e a saída é referida como *feature map*.

Em aplicações de ML, o *input* em geral é uma matriz multidimensional de dados e o *kernel*, uma matriz multidimensional de parâmetros que são adaptados pelo algoritmo de aprendizado. Essas matrizes multidimensionais são referidas com tensores (GOODFELLOW; BENGIO; COURVILLE, 2016).

**Convolutional Neural Networks (CNN):** Conforme LeCun e Bengio (1995), as CNNs combinam três ideias de arquitetura: campos de recepção locais, pesos compartilhados e, em alguns casos, uma subamostragem espacial ou temporal. O plano de entrada recebe imagens de caracteres aproximadamente centralizadas e normalizadas quanto ao tamanho. Cada unidade neuronal de uma camada recebe o sinal de entrada de um conjunto de unidades localizadas nas proximidades da camada anterior. Por meio dos campos de recepção locais, as unidades neuronais podem extrair características visuais elementares como bordas, cantos, ou extremidades, que por sua vez são combinadas pelas camadas superiores.

**Recurrent Neural Network (RNN):** Goodfellow, Bengio e Courville (2016) descrevem a RNN como uma classe de redes neurais para processamento de dados sequenciais, inclusive de comprimentos variáveis. O compartilhamento de parâmetros por diferentes partes de um modelo torna possível estender e aplicar o modelo a diferentes formas, ou comprimentos. Uma RNN pode ser bidirecional, em que uma predição de saída depende de toda uma sequência de entrada, como no caso do reconhecimento da fala e reconhecimento da escrita manual.

**Long Short-Term Memory (LSTM):** é uma modalidade de RNN associada a um algoritmo de aprendizado baseado em gradiente, projetado para superar os erros de retorno gerados na BP convencional, cuja evolução temporal depende exponencialmente dos pesos. Se divergirem podem levar a pesos oscilantes, se tenderem a zero, podem redundar em tempos de processamento proibitivos ou mesmo não funcionar (HOCHREITER; SCHMIDHUBER, 1997). Sua principal contribuição é introduzir recorrência interna (*self-loops*) para produzir caminhos onde os gradientes possam fluir por uma longa duração, em que os pesos não são fixos, mas condicionados ao contexto. É uma abordagem extremamente bem-sucedida em reconhecimento de textos manuscritos, reconhecimento da fala, tradução e geração de legenda de vídeos (GOODFELLOW; BENGIO; COURVILLE, 2016). Sua estrutura hierárquica se combina a muitas camadas ocultas para automaticamente extrair características, por exemplo, de sequências ordenadas de palavras e capturar relações altamente não lineares como significados dependentes do contexto (KRAUS; FEUERRIEGEL, 2017).

#### 2.4.2 Bibliometria

As bases foram consultadas o levantamento de publicações sobre DL. O recorte temporal partiu do ano de 2000, baseado na fundamentação teórica, concentrando os estudos no período de surgimento e maior produção do DL. Os critérios são apresentados no Quadro 4:

Quadro 4 – Critérios de busca para DL

Atributo	Critérios
Expressão de Busca	"Deep Learning"
Período	2000 a 2019
Idioma	Inglês
Tipo de Publicação	Artigos publicados em periódicos e conferências

Fonte: Autor

A Tabela 10 apresenta o número de documentos retornados pelas bases consultadas:

Tabela 10 – Número de publicações das bases consultadas sobre DL

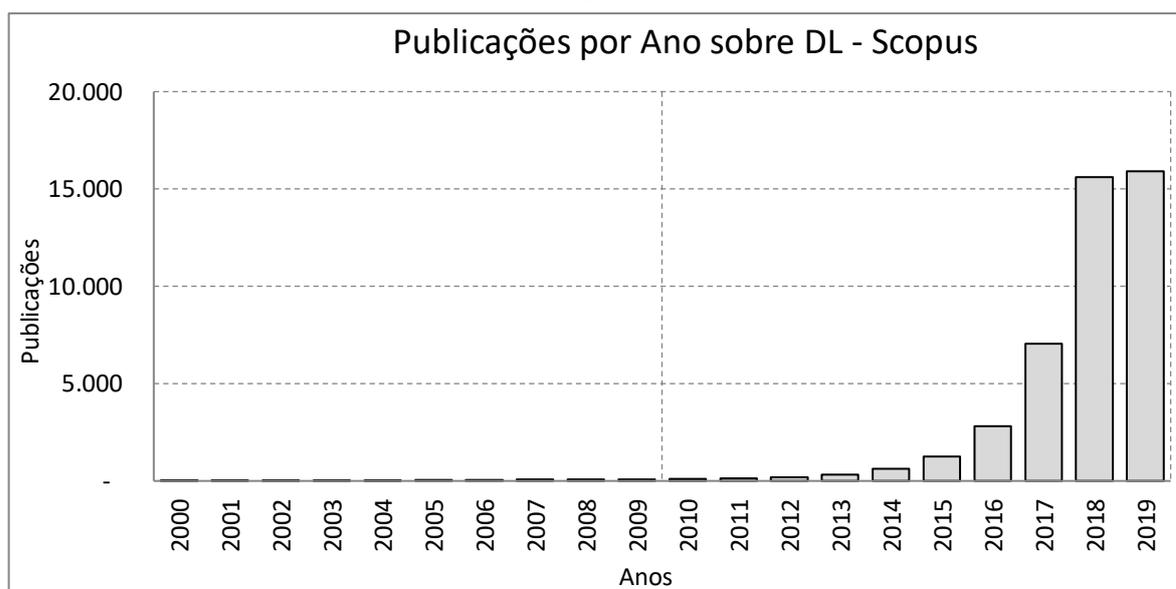
Base	Campo	Publicações
Scopus (Elsevier)	Títulos, Resumos e Palavras-chave	44.480
Web of Science (Clarivate Analytics)	Tópico	38.491
ACM Digital Library	Títulos, Resumos e Palavras-chave	3.445
IEEE Xplore	All-Metadata	18.834

Fonte: Scopus (2019), Web of Science (2019), ACM Digital Library (2019), IEEE Xplore (2019)

A evolução da produção científica sobre DL é apresentada nos gráficos seguintes, na Figura 15, conforme dados da base Scopus e na Figura 16, da base Web of Science.

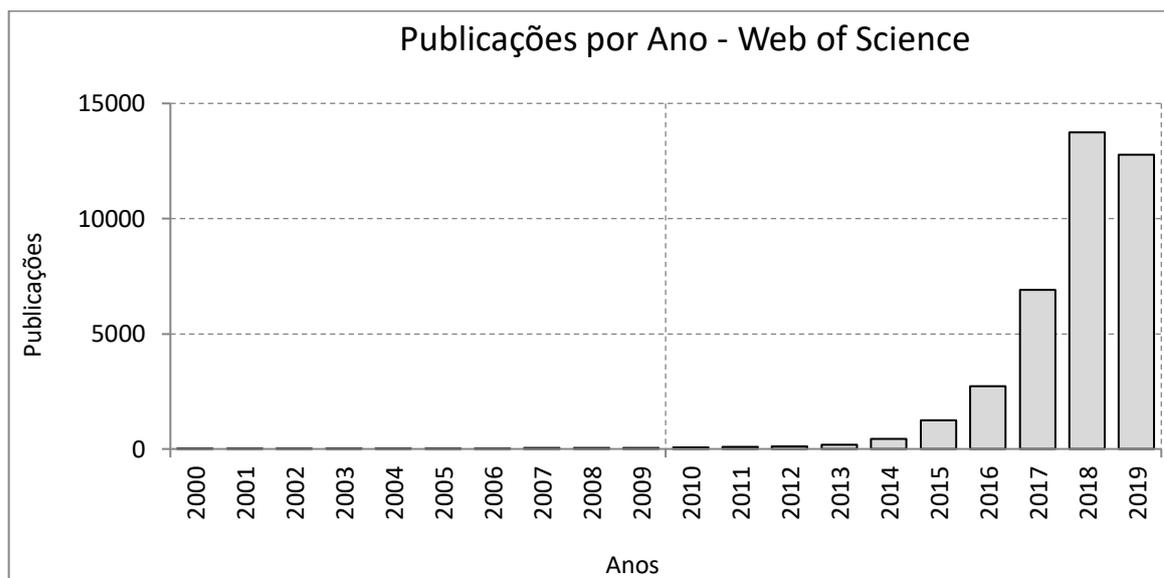
Em ambos se observa também um crescimento discreto no número de publicações, iniciando-se desta vez na década de 2010, tornando-se mais evidente em 2016, e apresentando um crescimento acentuado até o ano de 2018. Novamente, deve-se considerar que no ano de 2019 os dados são parciais, acumulados até o mês de setembro.

Figura 15 – Publicações por Ano sobre DL (Scopus)



Fonte: Scopus (2019)

Figura 16 – Publicações por Ano sobre DL (Web of Science)



Fonte: Web of Science (2019)

Na Tabela 11 são apresentados os crescimentos percentuais das publicações sobre DL nas bases Scopus e Web of Science, nos períodos de 5 anos, 2 anos e 1 ano. Verifica-se que o crescimento relativo da produção sobre DL foi maior que a sobre ML, ainda que em números absolutos seja menor.

Tabela 11 – Crescimento das publicações sobre DL nas bases Scopus e Web of Science

Bases	Períodos	5 anos	2 anos	1 ano
		2013 a 2018	2016 a 2018	2017 a 2018
Scopus		4.904 %	452 %	121 %
Web of Science		7.192 %	404 %	99 %

Fonte: Scopus (2019) e Web of Science (2019)

A consulta à base Scopus possibilitou também a identificação dos países com maior número de publicações. A Tabela 12 mostra os 10 países que mais publicaram. Cabe registrar o aumento da participação da China, passando a figurar como o país com maior número de publicações no período analisado, seguida dos Estados Unidos, Reino Unido, Índia. Destaque também para a Coreia do Sul, que passou a figurar na quinta colocação.

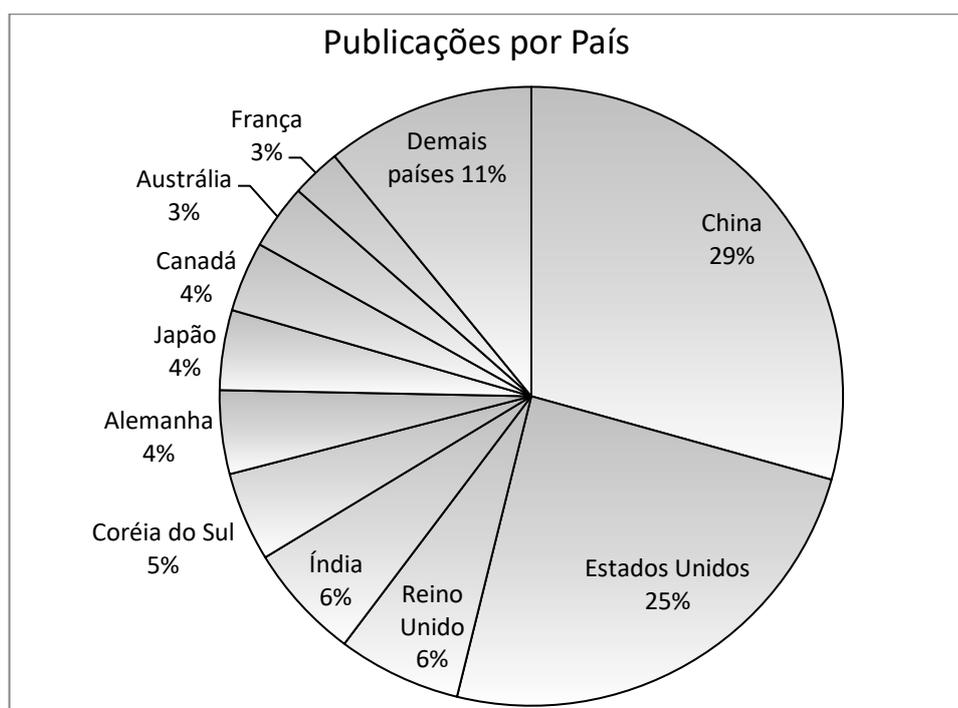
Tabela 12 – Dez Países que mais publicaram sobre DL

Países	Publicações
China	13.053
Estados Unidos	10.893
Reino Unido	2.849
Índia	2.688
Coréia do Sul	2.075
Alemanha	1.943
Japão	1.858
Canadá	1.631
Austrália	1.510
França	1.145
Demais países	4.835

Fonte: Scopus (2019)

A Figura 17 ilustra a participação percentual dos dez países que mais publicaram sobre DL, conforme a base Scopus.

Figura 17 – Dez países que mais publicaram sobre DL.



Fonte: Scopus (2019)

A Tabela 13 apresenta as vinte publicações sobre DL mais citadas, obtidas da base Scopus. Para os documentos selecionados, é fornecido também o FWCI.

Tabela 13 – Vinte publicações com maior número de citações sobre DL

<b>Título</b>	<b>Autores</b>	<b>Fonte</b>	<b>Ano</b>	<b>CC</b>	<b>FWCI</b>
<i>Deep learning</i>	Lecun Y., Bengio Y., Hinton G.	Nature	2015	11.756	130,43
<i>Dropout: A simple way to prevent neural networks from overfitting.</i>	Srivastava N., et al.	Journal of Machine Learning Research	2014	7.403	308,48
<i>Caffe: Convolutional architecture for fast feature embedding</i>	Jia Y., et al.	ACM Conference on Multimedia 2014	2014	4.846	440,94
<i>Deep Learning in neural networks: An overview</i>	Schmidhuber J.	Neural Networks	2014	3.646	
<i>Representation learning: A review and new perspectives</i>	Bengio Y., Courville A., Vincent P.	IEEE Pattern Analysis Machine Intelligence	2013	2.970	116,67
<i>Stacked denoising autoencoders: Learning Useful Representations in a Deep Network</i>	Vincent P., et al.	Journal of Machine Learning Research	2010	2.400	12,06
<i>Data Mining: Practical Machine Learning Tools and Techniques</i>	Witten I.H., et al.	Waikato Environment for Knowledge Analysis	2016	1.990	144,35
<i>FaceNet: A unified embedding for face recognition and clustering</i>	Schroff F., Kalenichenko D., Philbin J.	IEEE Computer Vision, Pattern Recognition	2015	1.915	164,20
<i>CNN features off-the-shelf: An astounding baseline for recognition</i>	Razavian A.S., et al.	IEEE Computer Vision, Pattern Recognition	2014	1.679	249,41
<i>3D Convolutional neural networks for human action recognition</i>	Ji S., Xu W., Yang M., Yu K.	IEEE Pattern Analysis Machine Intelligence	2013	1.650	49,17
<i>Random search for hyper-parameter optimization</i>	Bergstra J., Bengio Y.	Journal of Machine Learning Research	2012	1.435	33,69
<i>Image Super-Resolution Using Deep Convolutional Networks</i>	Dong C., Loy C.C., He K., Tang X.	IEEE Pattern Analysis Machine Intelligence	2016	1.415	178,32
<i>MatConvNet: Convolutional neural networks for MATLAB</i>	Vedaldi A., Lenc K.	ACM 2015 Multimedia Conference	2015	1.282	243,01
<i>Learning hierarchical features for scene labeling</i>	Farabet C., et al.	IEEE Pattern Analysis Machine Intelligence	2013	1.268	66,53
<i>DeepWalk: Online learning of social representations</i>	Perozzi B., Al-Rfou R., Skiena S.	ACM Knowledge Discovery Data Mining	2014	1.108	64,35
<i>A survey on deep learning in medical image analysis</i>	Litjens G., et al.	Medical Image Analysis	2017	1.083	
<i>Multimodal deep learning</i>	Ngiam J., et al.	Machine Learning, 28th ICML 2011	2011	1.068	45,40

<b>Título</b>	<b>Autores</b>	<b>Fonte</b>	<b>Ano</b>	<b>CC</b>	<b>FWCI</b>
<i>Large scale distributed deep networks</i>	Dean J., et al.	Advances in Neural Information Processing Systems	2012	1.037	80,86
<i>Fully Convolutional Networks for Semantic Segmentation</i>	Long J., Shelhamer E., Darrell T.	IEEE Pattern Analysis Machine Intelligence	2017	1.026	177,82
<i>Why does unsupervised pre-training help deep learning?</i>	Erhan D., et al.	Journal of Machine Learning Research	2010	980	12,52

Fonte: Scopus (2019)

Para identificar os autores de maior impacto, os títulos e palavras-chave dos 100 artigos mais citados foram lidos e pré-selecionados quanto à pertinência ao tema de busca. Em seguida os resumos dos documentos resultantes foram lidos para a seleção de autores, considerando um recorte longitudinal dos últimos 15 anos, partindo de 2004, visando à pesquisa mais recente. Dentre os autores selecionados, a Tabela 14 mostra aqueles com maior número de citações, índice-h e publicações.

Tabela 14 – Autores de maior impacto e produtividade em DL.

<b>Autor</b>	<b>Instituição</b>	<b>País</b>	<b>Publicações</b>	<b>Índice-h</b>	<b>Citações</b>
Hinton, G.E.	U. Toronto	Canada	218	85	111.734
Bengio, Y.	MIT	EUA	355	82	79.839
Ng, A.Y.	Stanford U.	EUA	189	80	51.361
LeCun, Y.	Facebook	EUA	186	67	44.080
Darrell, T.J.	U. California Bk	EUA	316	74	41.012
Schmidhuber, J.U.	U. Svizzera	Suíça	300	60	32.786
Malik, J.	U. California Bk	EUA	265	94	68.579
Zhang, L.	HK Polytechnic	Hong Kong	365	86	30.079
Tang, X.	Chinese U. HK	Hong Kong	389	83	32.513
Wang, X.	Chinese U. HK	Hong Kong	227	76	20.181
Schiele, B.	Max Plank I.I.	Alemanha	279	67	20.182
Torr, P.H.S.	U. Oxford	Reino Unido	226	65	17.568
Yuille, A.L.	Johns Hopkins	EUA	342	66	17.155
Deng, L.	Artificial Intelligence	EUA	319	61	18.821
Fei-Fei, L.	Stanford	EUA	164	60	25.246
Salakhutdinov, R.R.	Carnegie Mellon	EUA	115	47	25.981
Domingos, P.	U. Washington	EUA	141	52	16.601
Courville, A.C.	U. Montreal	Canada	92	34	15.144
Witten, I. H.	U. Waikato	N. Zelândia	269	48	13.479
Girshick, R.B.	Facebook	EUA	63	41	37.107
Russell, S.J.	U. California Bk	EUA	112	31	4.660

Fonte: Scopus (2019)

A análise dos artigos selecionados revela que o desempenho de métodos de ML é altamente dependente da escolha da representação de dados ou características. A engenharia de características, embora seja uma técnica importante, é trabalhosa e explicita a fraqueza de certos algoritmos de aprendizado por sua inabilidade de extrair e organizar informações a partir dos dados. O *Representation Learning* simplifica a extração de informações, transformando características em representações intermediárias para então mapeá-las para predições finais (NGIAM *et al.*, 2011; BENGIO; COURVILLE; VINCENT, 2013; WITTEN *et al.*, 2016) .

Para LeCun, Bengio e Hinton (2015), o DL compreende métodos de *Representation Learning* combinados em redes neurais de múltiplos níveis, obtidos pela composição de módulos simples, porém não-lineares que transformam a representação de um determinado nível para outro mais alto, mais abstrato. O *Representation Learning* permite que um programa seja alimentado com dados brutos para identificar as representações necessárias para a detecção ou classificação. Com a combinação de sucessivas transformações pode-se aprender funções complexas. Em tarefas de classificação, os níveis mais altos de representação amplificam os aspectos de entrada importantes para a discriminação e suprimem variações irrelevantes.

Como exemplo, considere-se uma imagem na forma de uma matriz de valores de *pixels*. Na primeira camada de representação, as características aprendidas identificam a presença ou ausência de bordas ou limites em orientações e localizações específicas da imagem. A segunda camada detecta o motivo ou tema pelo reconhecimento de arranjos particulares de bordas, independentemente de pequenas variações nas suas posições. A terceira camada pode combinar os temas em combinações maiores que correspondam a partes de objetos familiares. Camadas subsequentes poderiam detectar objetos como combinações dessas partes. O aspecto chave do DL é que essas camadas de representação não são criadas por humanos, são aprendidas a partir dos dados fazendo uso de um procedimento de cognitivo (LECUN; BENGIO; HINTON, 2015).

Bergstra e Bengio (2012) abordam estratégias para a otimização de hiper parâmetros, em que tentativas de escolha aleatórias se mostram mais eficientes que buscas em rede (*grid search*). Os hiper parâmetros são valores escolhidos que controlam o processo de aprendizado, diferentemente dos pesos, que são aprendidos durante o treinamento da rede neural.

Embora as *Deep Neural Networks* (DNN) com grande número de parâmetros sejam sistemas poderosos de ML, o *overfitting* permanece um problema crítico. Grandes redes neurais apresentam processamento lento, dificultando lidar com o *overfitting* pela combinação das

predições de variadas e amplas redes neurais em tempo de teste. O *Dropout* é uma técnica para resolver esse problema, cuja ideia básica é desconectar aleatoriamente algumas unidades neurais da rede durante o treinamento, prevenindo o excesso de co-adaptação entre elas (SRIVASTAVA *et al.*, 2014).

As CNNs são compostas de múltiplos estágios, cada qual contendo um módulo de banco de filtro, uma função de não linearidade e um módulo de gerenciamento espacial. As CNNs, treinadas de ponta a ponta, podem automaticamente aprender representações de características hierárquicas. Na área médica, tornaram-se o método de escolha para análise de imagens (FABERET *et al.*, 2013; LONG; SHELHAMER; DARRELL, 2015; LITJENS *et al.*, 2017).

O surgimento de novas *Graphics Processing Units* (GPU) a custos decrescentes, originalmente destinadas para jogos computadorizados (*video games*), possibilitou a implementação eficiente de operações de convolução, tornando o treinamento muito mais rápido (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Tal evolução tecnológica de hardware, somada à maior profundidade das camadas de representação, ao desenvolvimento das CNNs, às técnicas de treinamento distribuído em larga escala, aos algoritmos de aprendizado baseados em gradiente, ao grande repositório de imagens rotuladas destinado a pesquisas, e aos projetos ImageNet e FaceNet, contribuíram para a redução da taxa de erro das *Deep CNN*, que passaram a apresentar desempenho muito próximo ao humano para imagens estáticas e reconhecimento facial (DEAN *et al.*, 2012; JI *et al.*, 2013; RAZAVIAN *et al.*, 2014; SCHMIDHUBER, 2014; SCHROFF; KALENICHENKO; PHILBIN, 2015; DONG *et al.*, 2016).

Jia et al. (2014) apresentam o *framework* Caffe (*Convolutional Architecture for Fast Feature Embedding*) para implementação de algoritmos de DL no estado da arte. Originalmente desenvolvido pela Universidade de Berkeley, é disponibilizado sob licença BSD e integra-se à linguagem Python e ao software MATLAB para utilização em CNNs e outros modelos de aprendizado. O Caffe pode ser executado sobre plataforma de processamento paralelo como CUDA (*Compute Unified Device Architecture*). O MatConvNet é um framework para arquiteturas de CNN e implementado para o software MATLAB. Suporta processamento sobre GPU para treinamento de modelos complexos como ImageNet (VEDALDI; LENC, 2015)

Erhan et al. (2010) e Vincent et al. (2010) exploram abordagens de DBN baseados variações de *Stacked Autoencoders* com aprendizado não-supervisionado sobre conjuntos de dados de imagem e linguagem.

## 2.5 *Machine Learning* aplicado à Negociação por Algoritmos

### 2.5.1 *Contexto*

Para Gadre-Patwardhan, Katdare e Joshi (2016), a incerteza está entre os maiores desafios enfrentados por pesquisadores e gestores no campo das finanças, ao introduzir um inevitável fator de risco, que não apenas torna complexa a tomada de decisões, como também cria oportunidades de lucro a investidores que analisem os riscos de forma eficiente e efetiva. Advogam, portanto, uma abordagem interdisciplinar para lidar com a complexidade desta área do conhecimento.

Para Guo *et al.* (2017), estratégias quantitativas rentáveis envolvem uma eficiente coleta e análise de informações, bem como o desenvolvimento de estratégias dinâmicas que otimizem os critérios de desempenho prescritos na presença da incerteza.

Segundo Prado (2018), a administração de investimentos está entre as mais multidisciplinares áreas de pesquisa. Uma compreensão abrangente requer o conhecimento de ML, microestrutura do mercado, administração de carteiras, matemática financeira, estatística, econometria, álgebra linear, otimização convexa, matemática discreta, processamento de sinais, teoria da informação, programação orientada a objetos, processamento paralelo e computação de alto desempenho (HPC). Além dessas áreas, apoia-se também no domínio de linguagens de programação como R ou Python, e no conhecimento das bibliotecas de programas orientadas ao ML como: Scikit-learn, Pandas, Numpy, entre outras.

Easley, Prado e O'Hara (2011) argumentam, entretanto, que os investimentos atuais em HPC para a ciência seriam efetivamente suficientes para lidar com o grande volume de informações gerados pelos dados de séries de preços e livros de ofertas, utilizando ferramentas de software aberto (*open-source*), amplificadas por estruturas de dados especializadas e algoritmos de análise. Enxergam este desprendimento da dependência dos bancos de dados monolíticos como característica principal do *big data*. Para tanto, propõem como peças centrais, um formato de arquivos eficiente para organizar os dados, um conjunto de técnicas para computar indicadores prévios e a medida da volatilidade, e uma estratégia para paralelizar o processamento e aproveitar os vários núcleos disponíveis na maioria dos computadores.

Chan (2009) destaca os conceitos de estacionariedade e cointegração para uma melhor compreensão sobre o comportamento dos dados. Uma série de tempo é estacionária quando não se desvia continuamente do seu valor inicial. Em termos técnicos, séries estacionárias são ‘integradas de ordem zero’. Os preços das ações são, tipicamente, não estacionários, exibindo um passeio randômico geométrico, distanciando-se mais e mais do seu valor inicial. Entretanto, o diferencial de pares de ações nas arbitragens estatísticas (*Long-Short*) são séries estacionárias, e as séries de tempo de cada ação são ditas cointegradas, pois uma combinação linear delas é integrada de ordem zero.

Prado (2018) chama a atenção para os cuidados com o *overfitting* e o teste de modelos com base em dados históricos (*back tests*), pois a pesquisa na área das finanças lida com conjuntos de dados pouco extensos e uma baixa razão sinal/ruído.

Ainda que os preços de ações apresentem em média um comportamento próximo ao passeio aleatório, sob determinadas condições e horizontes de tempo, podem apresentar algum grau de reversão à média ou comportamento de tendência. Nessas situações podem-se utilizar abordagens de ML como *Hidden Markov Model* (HMM), filtro de Kalman ou Redes Neurais. Outro comportamento observável em mercados financeiros é o de regimes, evidenciando a importância da capacidade de se prever o momento da mudança de regime, ou do ponto de inflexão, em que também se pode utilizar, ainda que com baixa capacidade preditiva, o HMM, também chamado de Mudança de Regime de Markov, e se baseia no modelo probabilístico de Bayes (CHAN, 2009; LÄNGKVIST, 2014).

Aspecto importante no estudo das séries de preços em finanças é o conceito de ruído presente na sequência temporal de dados, resultante da atuação de uma grande quantidade de participantes do mercado, comprando e vendendo em diferentes momentos e para variados propósitos. Podem resultar também de choques de preços, como eventos inesperados, como notícias ou fatos econômicos, cujos impactos podem persistir por variados períodos no futuro. O ruído apresenta muitas das características das sequências randômicas. Quase metade dos movimentos de preços muda de direção no dia seguinte, cerca de 25% mantém a direção por dois dias seguidos, e 12,5% por três dias, e assim por diante. Por outro lado, a intensidade dos movimentos de preço também apresenta característica randômica, 50% são relativamente pequenos, 25% apresentam o dobro de tamanho, 12,5% são quatro vezes maiores, e pouquíssimos são extremamente grandes (KAUFMAN, 2013)

A computação financeira (*computational finance*), uma divisão da ciência da computação aplicada, pode ser definida como o estudo de dados e algoritmos utilizados em

finanças, campo interdisciplinar que combina métodos numéricos e matemática financeira. Pesquisadores utilizam seus modelos para propor soluções em finanças, em áreas como: previsão de tendências, análise comportamental do investidor, administração de portfólio de investimentos, detecção de fraudes, avaliação de risco, insolvência, previsão de ações, identificação de padrões no movimento de preços de ativos, entre outros. Para tanto, fazem uso de métodos estatísticos paramétricos e não-paramétricos, bem como métodos computacionais (GADRE-PATWARDHAN; KATDARE; JOSHI, 2016).

Os métodos estatísticos paramétricos assumem que os dados sejam coletados de sistemas distribuídos e integrados para deduzir inferências sobre os parâmetros da distribuição. Os dois tipos desses métodos são a análise discriminante e a regressão logística. A análise discriminante utiliza de uma função classificadora que distribui itens de dados em classes, grupos ou categorias. A regressão logística consiste em um método de predição que modela o relacionamento entre variáveis independentes e dependentes. Os métodos estatísticos não-paramétricos não requerem que os dados obedeçam a uma distribuição normal, como DTs e *Nearest Neighbor*. Dentre as técnicas computacionais mais aceitas e aplicadas estão: ANN, Lógica Fuzzy, SVM e GA (GADRE-PATWARDHAN; KATDARE; JOSHI, 2016).

Ainda quanto às técnicas computacionais, Gadre-Patwardhan, Katdare e Joshi (2016) enfatizam as abordagens de ANN, *Expert Systems* (ES) e *Hybrid Intelligence System* (HIS). ANNs seriam mais eficientes no tratamento da incerteza na área de finanças, como reconhecimento de padrões e análise de tendências futuras, mesmo na presença de grande ruído nos dados. Diferentemente das técnicas de estatísticas, como discriminante ou regressões, não depende de pressupostos sobre a distribuição dos dados, o que lhe permite ser aplicada a uma maior diversidade de situações, e acomodar dados novos sem necessidade de reprocessamento, o que a torna particularmente útil na predição em finanças. Destacam também os ESs dentre os sistemas aplicados à solução de problemas críticos em um domínio específico, que utilizam um motor de inferências e uma base de conhecimento para a tomada de decisão. A base de conhecimento é codificada na forma de regras, redes semânticas, predicados e objetos. Tais sistemas são eficientes, permanentes, consistentes, concebendo conclusões a partir de relações e lidando com incertezas, o que explica seu uso em finanças.

Middleton, Theofilatos e Karathanasopoulos (2016), propõem uma alternativa aos métodos lineares que restringem o foco a séries de tempo e aos primeiros modelos de ML que resultavam em baixa acurácia e lucratividade devido às suas arquiteturas rígidas. Tais alternativas são de particular importância em momentos de crise, quando aumentam as

correlações entre diferentes classes de ativos e as séries de tempo. Por outro lado, a maioria dos métodos não-lineares buscam estimadores globais ótimos, que na maior parte do tempo podem nem existir devido à natureza dinâmica das séries de tempo financeiras. Assim propõem o treinamento o modelo de ML usando uma janela móvel de variáveis explanatórias buscando o preditor ótimo a cada dia, por meio de uma combinação de uma versão adaptada do algoritmo de *Particle Swarm Optimization* (PSO) com a *Radial Basis Function* (RBF) sobre redes neurais.

A *Gene Expression Programming* (GEP) pertence à ampla categoria dos algoritmos evolucionários e da programação genética. Baseiam-se no princípio Darwiniano da reprodução e sobrevivência do mais apto, modelando problemas de regressão e classificação usando uma tipologia baseada em árvores. Estas estruturas baseadas em árvores representam modelos de relação entrada-saída que evoluem e para produzir novas soluções até atingir um critério ou nível de desempenho pré-definido. O GEP aplica operações de biologia genética tais como recombinação de cruzamento (*crossover*) e mutação para identificar complexos padrões não-lineares e não-estacionários. Apesar de suas limitações em previsões financeira, comparado modelos de redes neurais, o GEP não corre o risco de se prender na armadilha do ótimo local, atingindo a solução ótima mais rapidamente (KARATHANASOPOULOS; MIDDLETON; THEOFILATOS; GEORGOPOULOS, 2016).

Operações financeiras que requerem a tomada de decisões baseadas em regras pré-definidas, como nos algoritmos de negociação, na precificação de opções e monitoramento de riscos, transformaram os mercados financeiros em redes de troca de informações extremamente rápidas e conectadas, resultando no surgimento do HFT (KEARNS; NEVMYVAKA, 2013).

Kearns e Nevmyvaka (2013) reiteram que a utilização de dados históricos para inferência preditiva em finanças quantitativas tem sido amplamente utilizada, tendo como exemplos o *Capital Asset Pricing Model* (CAPM) e a Hipótese do Mercado Eficiente (EMH). Todavia, a dificuldade do ML para lidar com HFT surge da microestrutura do mercado, da granularidade dos dados, com execuções parciais, cancelamentos de ordens, liquidez oculta, em que não se tem ao menos uma intuição de como a distribuição de liquidez dos livros de ordens limitadas (LOB) se relaciona aos movimentos futuros dos preços. Assim, propõem a seleção de características (*feature selection*) ou engenharia de características (*feature engineering*) como caminhos para aplicação de ML ao AT, em especial o HFT.

- Considerando-se que o desenvolvimento de características que permitam uma previsão confiável da direção de movimento dos preços a partir de certos estados do

LOB, em que “confiável” signifique bom o suficiente para que operações lucrativas superem as não lucrativas, em oposição a estarem corretas a maior parte do tempo; e

- Que o desenvolvimento de algoritmos de aprendizado capture essa previsibilidade (*alpha*) a custos suficientemente baixos,

é possível concluir, a partir de estudos de caso realizados, que o ML fornece um framework poderoso, escalável e fundamentado para análise de dados e previsão, mas não um caminho fácil para a rentabilidade (KERNNS; NEVMYVAKA, 2013).

Os métodos de ML se aplicam também aos novos mecanismos de negociação como o SOR e os centros alternativos de liquidez (*dark pools*), onde os dados são consideravelmente menos abundantes. Neste caso, as ordens são enviadas a várias *dark pools* que competem entre si na oferta de diferentes perfis de liquidez. Estes centros de negócios foram originalmente concebidos para uma maior oferta de liquidez do que para a melhoria de preços, atendendo volumes a um preço intermediário entre oferta e procura, que todavia se fossem executados numa bolsa de valores tradicional, redundaria em custos de transação inaceitavelmente altos devido ao impacto no mercado. Entretanto, as *dark pools* não disponibilizam os dados do LOB, nem um histórico das execuções, com seus preços e quantidades. Portanto o SOR deve buscar aprender a distribuição aproximada de execuções por volume de cada uma. Como os dados de liquidez estão ocultos, sugere-se o uso do estimador Kaplan-Meier, também conhecido como estimador limite produto (KEARNS; NEVMYVAKA, 2013).

Sobre o uso de RL em sistemas de AT, Kearns e Nevmyvaka (2013) lembram que esse método teve origem num campo mais antigo, a teoria de controle. O RL é um ramo do ML destinado explicitamente para aprender políticas baseadas em estados dinâmicos, como os livros de oferta. Os elementos primários de uma aplicação em RL são:

- identificação do espaço de estado, cujos elementos representam as condições variáveis sob as quais se escolhem ações. No caso de AT, a função de variação do volume no tempo, bem como componentes adicionais ou características que capturam o estado do LOB.
- identificação do conjunto de ações disponíveis de cada estado. Em AT significa preparar ordens limitadas para o volume remanescente a preços variáveis, de forma que se tenha uma única ordem a cada momento, e que seja repostada em resposta à variação do volume corrente.
- identificação de um modelo de impacto ou influência, na forma de probabilidades de execução de acordo com os estados, aprendido com dados históricos.
- identificação de uma função de custo ou recompensa que indique o retorno médio ou esperado para uma dada ação sob um determinado estado. No caso do AT, o custo de se

colocar uma ordem limitada num determinado estado seriam as despesas decorrentes de uma execução, ainda que parcial, da ordem.

- algoritmos para aprender a política ótima, mapeando estados em ações que minimizem o custo empírico (custos de transação) sobre os dados de treinamento.
- validação da política aprendida nos dados de testes pela estimativa de desempenho fora da amostra.

As técnicas de RL se diferenciam das abordagens preditivas tradicionais, como regressão, por que aprendem como agir diretamente no espaço de estado, em vez de simplesmente prever valores alvos. Assim os métodos de RL se aplicam à execução otimizada de ordens, utilizando as escolhas para estados, ações, impactos e recompensas acima mencionadas para tratar a microestrutura dos dados de várias ações líquidas (KEARNS; NEVMYVAKA, 2013).

A Engenharia de Características é uma importante técnica preditiva dentre os métodos de regressão não-paramétricos, aplicado a janelas móveis de tempo. No contexto dos dados do LOB, a classificação busca a predição da direção da mudança de preços, enquanto a regressão visa o tamanho ou intensidade da mudança de preços, baseando-se em modelos de regressão linear, logística e não linear, associados a métodos *forward*, *backward* e *stepwise* de seleção de variáveis. A regressão não paramétrica é basicamente a regressão linear aplicada a muitas funções de base (GUO *et al.*, 2017).

Constituindo uma grande classe de modelos preditivos, as DTs operam com amostras de treinamento ponderadas para reduzir a variância da decisão. Esses modelos experimentam uma fase de testes que combinam as árvores por meio de um algoritmo de impulsão (*boosting algorithms*), como o AdaBoost. As *Random Forests*, por sua vez, treinam várias DTs usando subconjuntos de dados e estabelece uma previsão baseada na média (HASTIE; TIBSHIRANI; FRIEDMAN, 2009; GUO *et al.*, 2017).

As ANNs são métodos adaptativos orientados a dados adequados a situações em que há poucos pressupostos sob o modelo em estudo, sendo caracterizados como estimadores universais na forma de funções contínuas (ZHANG; PATUWO; HU, 1998; HORNIK; STINCHCOMBE; WHITE, 1989). Os modelos de *Neural Network Regression* (NNR) começaram a ser aplicados nos mercados financeiros pela possibilidade de tratar as não linearidades entre variáveis independentes e dependentes (DUNIS; JALILOV, 2002). Para Dunis e Williams (2003), não obstante, os modelos de NNR têm sido criticados pela sua

natureza fechada (*black-box*), por requerer um tempo excessivo para o treinamento e apresentar riscos de *overfitting*.

A função de ativação frequentemente utilizada é a sigmoide logística, que introduz um grau de não linearidade ao modelo e evita que a saída apresente comportamento divergente, o que poderia paralisar a rede neural, inibindo o treinamento (ZHANG; PATUWO; HU, 1998; DUNIS; WILLIAMS, 2003).

Entretanto, o processo de construção de modelos NNR não é trivial, envolvendo questões que afetam seu desempenho, devendo ser cuidadosamente consideradas quanto aos métodos estatísticos, a capacidade de generalização e o problema de *overfitting*. Tais questões requerem especial atenção em finanças, em que as séries de tempo apresentam características randômicas, com presença de ruído no sinal. Recomenda-se, portanto, que as NNR comecem com poucos nós nas primeiras camadas, e aumentem a complexidade enquanto se monitora a capacidade de generalização. O número de camadas ocultas e o de nós em cada camada é fator crucial de sucesso para uma a generalização. Por serem modelos de reconhecimento de padrões, a representação de dados também é crítica para um projeto de NNR eficaz. E não menos crítica é a taxa de aprendizado na BP, pois determina o tamanho das mudanças nos pesos. Baixas taxas tornam o aprendizado mais lento, enquanto taxas muito altas fazem com que a função de erro se altere descontroladamente, sem uma melhora contínua. Para aperfeiçoar esse processo utiliza-se um parâmetro de impulso (*momentum*) que verifica como as alterações passadas afetaram as mudanças nos pesos atuais, ao fazer a próxima alteração de peso aproximadamente na mesma direção da anterior (ZHANG; PATUWO; HU, 1998; DUNIS; WILLIAMS, 2003).

Os dados financeiros utilizados por uma plataforma de AT podem ser classificados em: dados fundamentais, dados de mercado (MD), dados analíticos e dados alternativos. Dados Fundamentais são aqueles fornecidos em publicações corporativas como balanços, demonstrativos e demais dados contábeis. São informações de divulgação obrigatória que seguem procedimentos definidos e são fiscalizadas por órgãos reguladores. O MD inclui toda atividade de negociação que ocorre numa bolsa de valores ou outros mercados alternativos de negociação. Os dados analíticos são dados secundários, derivados de outras fontes, resultantes de um processo de análise por parte de um terceiro, que pode ser um banco de investimento ou uma firma de análise de empresas. Os dados alternativos são aqueles produzidos por notícias, redes sociais, registros e processos comerciais e governamentais, como níveis de vendas, estoques, dados econômicos, taxas de juros, entre outros.

O Quadro 5 mostra os quatro tipos de dados financeiros (PRADO, 2018).

Quadro 5 – Quatro tipos de dados financeiros

Dados Fundamentais	Dados de Mercado	Dados Analíticos	Dados Alternativos
Ativos	Preço, Retorno	Recomendações	Notícias:
Passivos	Volume negociado	Risco de Crédito	Redes Sociais, Internet,
Vendas	Proventos	Previsão de Lucros	Governamentais,
Custos / Receitas	Ofertas	Análise de Sentimento	Comércio, Indústria,
Variáveis Macro	Volatilidade	Análise Técnica	Agronegócio
Múltiplos	Dados históricos		

Fonte: Prado (2018)

### 2.5.2 Bibliometria

Com a finalidade de prospectar a produção científica sobre AT utilizando abordagens de ML, as bases foram pesquisadas utilizando-se uma expansão dos termos ligados ao AT e relacionados ao mercado de ações para alcançar artigos com abordagens de ML relativos à predição do comportamento de preços de ações. Os critérios de pesquisa são apresentados no Quadro 6:

Quadro 6 – Critérios de busca sobre abordagens de ML aplicadas ao AT

Atributo	Critério
Expressão de Busca	<i>("Machine Learning") AND ("Algorithmic Trading" OR "High Frequency Trading" OR "Systematic Trading" OR "Quantitative Trading" OR "Automated Trading" OR "Stock Market" OR "Stock Trading" OR "Stock Price")</i>
Período	1990 a 2019
Idioma	Inglês
Tipo de Publicação	Artigos publicados em periódicos e conferências
Exclusão de domínios de pesquisa	Artes e Humanidades, Psicologia, Medicina, Agricultura e Biociências, Química e Geociências.

Fonte: autor

A Tabela 15 apresenta o número de documentos retornados pelas bases consultadas:

Tabela 15 – Publicações sobre abordagens de ML aplicadas ao AT

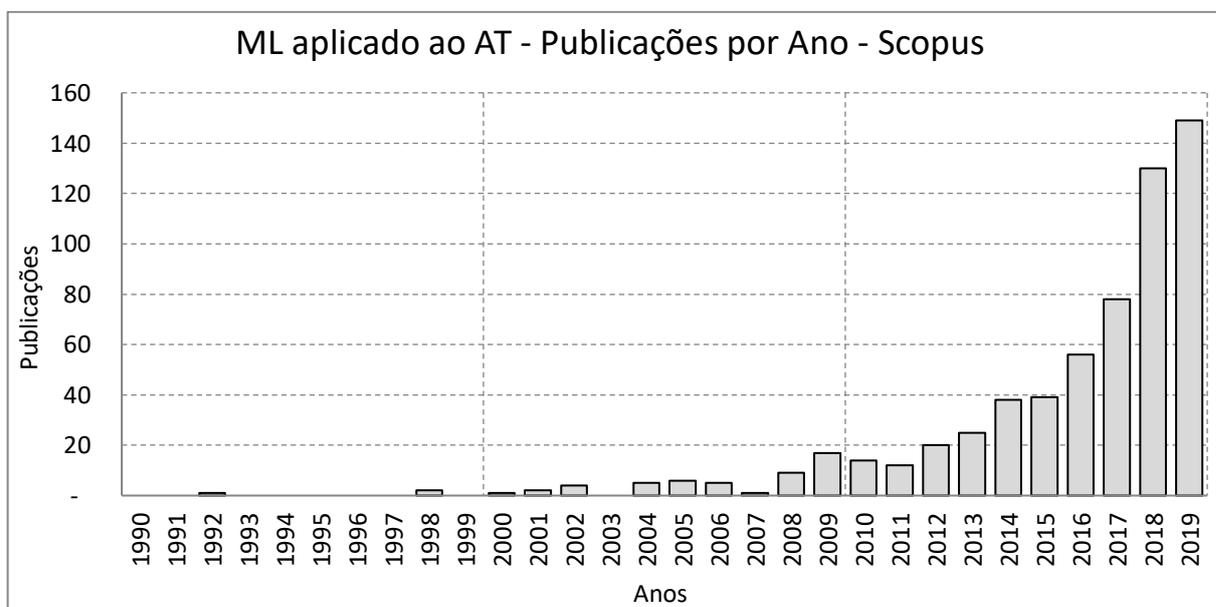
Base	Campo	Publicações
Scopus	Títulos, Resumos e Palavras-chave	614
Web of Science	Tópicos	397
ACM Digital Library	Títulos, Resumos e Palavras-chave	163
IEEE Xplore	All-Metadata	301

Fonte: Scopus (2019), Web of Science (2019), ACM Digital Library (2019), IEEE Xplore (2019)

Diferentemente das buscas anteriores, observa-se um número maior de publicações nas bases ACM DL e IEEE Xplore em relação às bases Scopus e Web of Science.

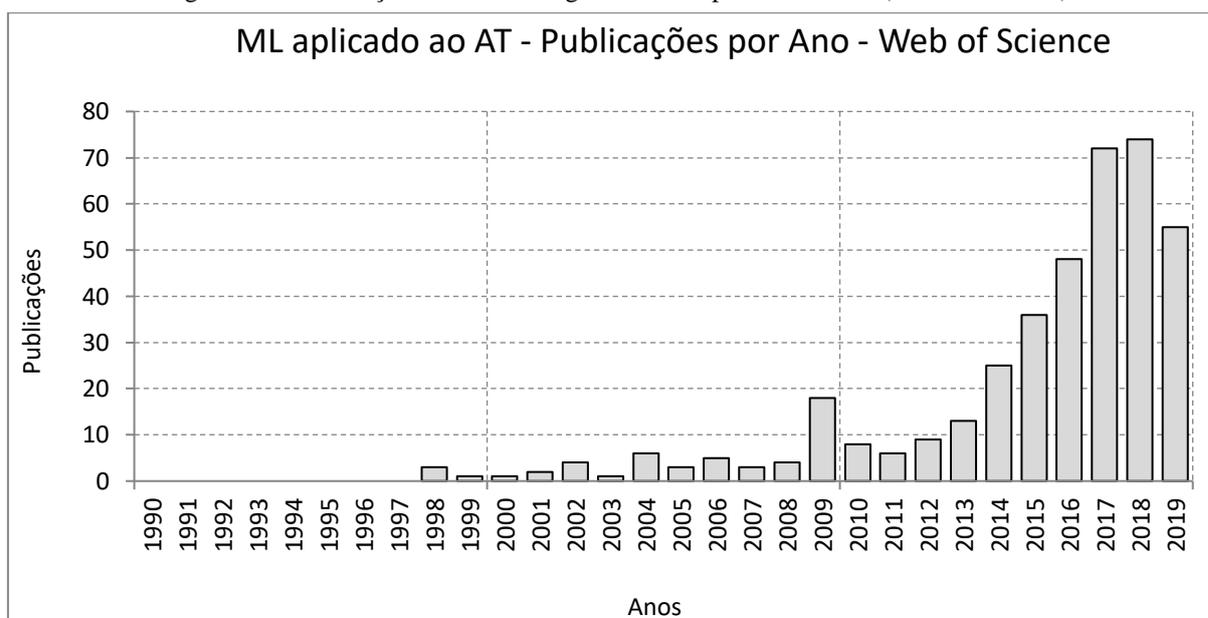
O crescimento de publicações por ano é apresentado na Figura 18, conforme base Scopus e na Figura 19, conforme Web of Science. A partir do ano de 2008 observa-se um aumento no patamar de publicações e a partir de 2015 o crescimento se acentua. O ano de 2019 apresenta parciais até o mês de setembro.

Figura 18 – Publicações sobre abordagens de ML aplicadas ao AT (Scopus)



Fonte: Scopus (2019).

Figura 19 – Publicações sobre abordagens de ML aplicadas ao AT (Web of Science)



Fonte: Web of Science (2019).

A Tabela 16 apresenta os crescimentos percentuais das publicações sobre AT nas bases Scopus e Web of Science, nos períodos de 10 anos, 5 anos e 1 ano.

Tabela 16 – Crescimento das publicações sobre abordagens de ML aplicadas ao AT

Bases	Períodos	10 anos	5 anos	1 ano
		2008 a 2018	2013 a 2018	2017 a 2018
Scopus		1.344 %	420 %	67 %
Web of Science		1.750 %	469 %	3 %

Fonte: Scopus (2019) e Web of Science (2019)

A Tabela 17 mostra os dez países que mais publicaram sobre o tema, conforme a base Scopus. A Índia, China e Estados Unidos prevalecem nesse segmento. Cabe destacar a presença do Brasil em sexto lugar ao lado de Canadá e Taiwan.

Tabela 17 – Dez Países que mais publicaram sobre abordagens de ML aplicadas ao AT

Países	Publicações
Índia	113
China	95
Estados Unidos	83
Reino Unido	37
Coréia do Sul	24
Brasil	20
Canadá	20
Taiwan	20
Austrália	17
Japão	16
Demais países	169

Fonte: Scopus (2019)

No gráfico da Figura 20 pode-se visualizar a participação percentual dos dez países com maior número de publicações, conforme a base Scopus.

Figura 20 – Países que mais publicaram abordagens de ML aplicadas ao AT



Fonte: Scopus (2019)

A Tabela 18 apresenta os dez artigos sobre abordagens de ML aplicadas ao AT mais citados, conforme a base Scopus, com número de citações e o FWCI.

Tabela 18 – Dez publicações com maior número de citações sobre ML aplicado ao AT

<b>Título</b>	<b>Autores</b>	<b>Fonte</b>	<b>Ano</b>	<b>CC</b>	<b>FWCI</b>
<i>Textual analysis of stock market prediction using breaking financial news: The AZFin text system</i>	Schumaker R.P., Chen H.	ACM Information Systems	2009	312	9,69
<i>The use of data mining and neural networks for forecasting stock market returns</i>	Enke, D., Thawornwong, S.	Expert Systems with Applications	2005	217	4,57
<i>Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques</i>	Patel J., <i>et al.</i>	Expert Systems with Applications	2014	164	19,89
<i>Automated news reading: Stock price prediction based on financial news using context-capturing features</i>	Hagenau M., Liebmann M., Neumann D.	Decision Support Systems	2013	106	4,58
<i>Online sequential extreme learning machine with forgetting mechanism</i>	Zhao J., Wang Z., Park D.S.	Neurocomputing	2012	106	3,43
<i>Predicting stock market index using fusion of machine learning techniques</i>	Patel J., <i>et al.</i>	Expert Systems with Applications	2015	95	10,29
<i>Application of wrapper approach and composite classifier to the stock trend prediction</i>	Huang C.-J., Yang D.-X., Chuang Y.- T.	Expert Systems with Applications	2008	94	4,31
<i>Forecasting model of global stock index by stochastic time effective neural network</i>	Liao Z., Wang J.	Expert Systems with Applications	2010	91	7,04
<i>A quantitative stock prediction system based on financial news</i>	Schumaker R., Chen H.	Informat. Processing and Management	2009	90	3,70
<i>The adaptive selection of financial and economic variables for use with artificial neural networks</i>	Thawornwong S., Enke, D.	Neurocomputing	2004	82	0,99

Fonte: Scopus (2019)

Enke e Thawornwong (2004) examinaram se o uso de variáveis relevantes recentes melhoraria a previsão de retorno das ações. Dada a evidência de não linearidade nos mercados financeiros, as variáveis foram submetidas a redes neurais probabilísticas e *feed-forward* para a previsão de excedente de retorno futuro. Os resultados mostraram que redes neurais remodeladas, que usam informações relevantes recentes geram mais lucros e menos riscos que as estratégias *buy-and-hold*, regressão linear convencional, modelo de passeio aleatório e modelos de redes neurais que utilizam variáveis relevantes constantes.

A técnica de ganho de informação, usada em ML para mineração de dados, é utilizada por Enke e Thawornwong (2005) para avaliar as relações preditivas de variáveis financeiras e

econômicas. A partir de redes neurais alimentadas por tais variáveis, obtiveram lucros ajustados ao risco mais altos e superiores à estratégia *buy-and-hold*.

Huang, Yang e Chuang (2008) empregaram uma abordagem *wrapper* para selecionar o subconjunto de característica ótimo de um conjunto original de 23 índices técnicos associado a um processo de votação que combina diferentes algoritmos de classificação. Os resultados mostraram que a abordagem *wrapper* pode alcançar melhor desempenho que filtros comumente usados. Também constatou que o processo de votação supera classificadores simples como SVM, *K-Nearest Neighbor*, BP NN, DT e Regressão Logística.

Schumaker e Chen (2009a) utilizaram SVM e diversas representações textuais de dados para analisar notícias financeiras e prever o preço de ações no período de vinte minutos após a divulgação da notícia. A precisão na direção de preços atingiu 57% e o retorno de mais de 2%, o mais alto valor, então, para um sistema de negociação simulada.

Utilizando uma síntese de linguística e técnicas financeiras e estatísticas, Schumaker e Chen (2009b) compararam predições do Arizona Financial Text System (AZFinText) a fundos quantitativos e avaliações de especialistas. Os resultados mostraram que particionado por setores, os preços das ações apresentavam melhorias na medida de proximidade, erro quadrático médio e precisão direcional, superando também especialistas de mercado e ficando em quinto lugar comparado aos dez melhores fundos quantitativos.

Liao e Wang (2010) utilizaram uma abordagem *Stochastic Time Effective Neural Network* sobre um modelo de índice de ações globais. Assumem que os investidores tomam decisões de investimento baseados em dados históricos, ponderado no tempo, associado a um movimento Browniano para introduzir um efeito de movimento randômico, enquanto mantém a tendência original, resultando em predições com alta correlação com os valores reais.

Zhao, Wang e Park (2012) propõem a abordagem *Online Sequential Extreme Learning Machine with Forgetting Mechanism* que melhora os efeitos de aprendizado pelo descarte de dados desatualizados, com tempo de treinamento menor e precisão melhorada comparada às abordagens anteriores.

Hagenau, Liebmann e Neumann (2013) utilizaram métodos de *data mining* utilizando características mais expressivas para representação textual que abordagens anteriores. Uma robusta seleção de características semanticamente relevantes associada a tipos de características complexas permitem aumentar a precisão da classificação e reduzir o *overfitting*.

Patel et al. (2014) comparam ANNs, SVM, Random Forrest e Naive-Bayes utilizando duas formas de entrada de dados, o primeiro computando dez parâmetros técnicos a partir de dados de mercado diários, e o segundo representa esses parâmetros como dados determinísticos de tendências. Os resultados mostraram que a *Random Forest* apresentou melhor desempenho para a primeira forma de entrada de dados, e todas as abordagens apresentaram melhora na segunda forma.

Patel *et al.* (2015) realizaram experimentos de predição para 1 a 10 dias e 15 a 30 dias à frente, comparando abordagem em um estágio com abordagens em dois estágios, sendo *Support Vector Regression* no primeiro e uma combinação com ANN, ou *Random Forest* ou *Support Vector Regression* no segundo. Os resultados mostraram que os modelos híbridos desempenham melhor que os de estágio único.

## 2.6 *Deep Learning* aplicado à Negociação por Algoritmos

### 2.6.1 *Contexto*

Avanços recentes em DL têm se mostrado promissores na extração de informações sobre dados não classificados, prescindindo do trabalho intensivo da Engenharia de Características, e sendo também aplicados com sucesso em campos como o reconhecimento da fala, reconhecimento de imagens e processamento da linguagem natural (BENGIO; COURVILLE; PASCAL, 2012; LECUN; BENGIO; HINTON, 2015).

O DL também apresenta a vantagem de poder sobrepor camadas de representação de características para criar redes neurais profundas (DNN) com maior capacidade de modelar estruturas complexas de dados. Porém, para obter resultados satisfatórios e maior eficiência de aprendizado, as redes neurais dependem da redução da dimensionalidade e de uma cuidadosa seleção das variáveis de entrada e dos parâmetros de rede, como a taxa de aprendizado, o número de camadas ocultas e o número de nós em cada camada (LÄNGKVIST; KARLSSON; LOUTFI, 2014; CHONG; HAN; PARK, 2017).

A previsão de preços de ações, por sua vez apresenta alto grau de complexidade até mesmo para analistas especializados e investidores institucionais. Ademais, a identificação de padrões está sujeita a mudanças no comportamento dos agentes que competem pela obtenção de resultados ao longo do tempo (TAKEUCHI; LEE, 2013).

Não obstante, observa-se uma tendência na pesquisa em ML e reconhecimento de padrões para a aplicação da topologia não-linear do DL à predição em séries temporais financeiras, utilizando abordagens como as Redes Neurais Convolucionais (CNN) e em especial as Redes Neurais Recorrentes (RNN), por meio de seus modelos, tais como: *Long Short-Term Memory* (LSTM), *Gated Recurrent Unit* (GRU), *Deep Belief Network* (DBN) e *Stacked Autoencoders* (BAO; YUE; RAO, 2017).

### 2.6.2 Bibliometria

As mesmas bases foram pesquisadas para levantar a produção científica sobre abordagens de DL aplicadas ao AT, com a mesma expansão dos termos ligados ao AT do levantamento sobre ML. Os critérios de busca são mostrados no Quadro 7:

Quadro 7 – Critérios de busca sobre abordagens de DL aplicadas ao AT

Atributo	Critério
Expressão de Busca	("Deep Learning") AND ("Algorithmic Trading" OR "High Frequency Trading" OR "Systematic Trading" OR "Quantitative Trading" OR "Automated Trading" OR "Stock Market" OR "Stock Trading" OR "Stock Price" OR "Trading Systems")
Período	2000 a 2019
Idioma	Inglês
Tipo de Publicação	Artigos publicados em periódicos e conferências
Exclusão de Domínios de Pesquisa	Artes e Humanidades, Psicologia, Medicina, Agricultura e Biociências, Química e Geociências.

Fonte: autor

O número de documentos retornados pelas bases é apresentado na Tabela 19.

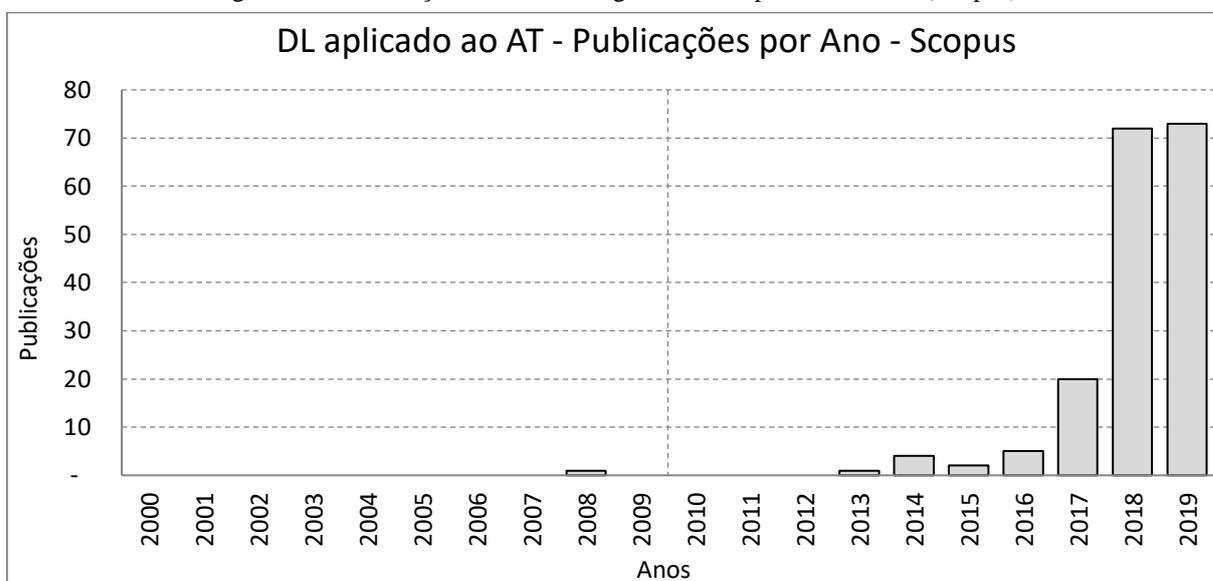
Tabela 19 – Publicações sobre abordagens de DL aplicadas ao AT

Base	Campo	Publicações
Scopus	Títulos, Resumos e Palavras-chave	178
Web of Science	Tópicos	103
ACM Digital Library	Títulos, Resumos e Palavras-chave	20
IEEE Xplore	All-Metadata	107

Fonte: Scopus (2019), Web of Science (2019), ACM Digital Library (2019), IEEE Xplore (2019)

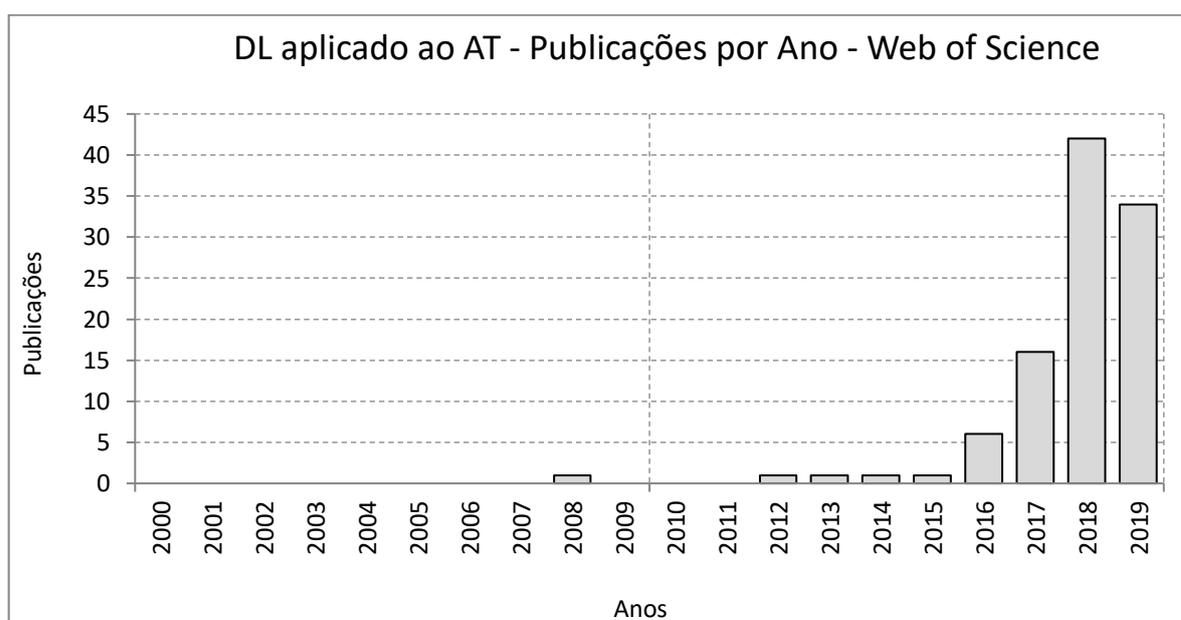
O crescimento de publicações utilizando abordagens de DL aplicadas ao AT é mostrado na Figura 21, conforme a base Scopus, e na Figura 22, conforme Web of Science.

Figura 21 – Publicações sobre abordagens de DL aplicadas ao AT (Scopus)



Fonte: Scopus (2019)

Figura 22 – Publicações sobre abordagens de DL aplicadas ao AT (Web of Science)



Fonte: Web of Science (2019)

Nota-se o surgimento dos primeiros documentos em meados da década de 2010. A partir de 2017 o crescimento se acentua, seguida de um crescimento significativo em 2018. Em 2019,

novamente deve-se considerar que a busca foi limitada até 30 de setembro, ainda assim o patamar elevado se mantém.

Na Tabela 20 são apresentados os crescimentos percentuais das publicações nas bases Scopus e Web of Science, nos períodos de 5 anos e 1 ano.

Tabela 20 – Crescimento das publicações sobre abordagens de DL aplicadas ao AT

<b>Bases</b>	<b>Períodos</b>	<b>2 anos</b>	<b>1 ano</b>
		<b>2016 a 2018</b>	<b>2017 a 2018</b>
Scopus		1.340 %	260 %
Web of Science		600 %	163 %

Fonte: Scopus (2019) e Web of Science (2019)

A Tabela 21 lista os dez países que mais publicaram abordagens de DL aplicadas ao AT. A China encabeça a lista seguida de Estados Unidos e Índia.

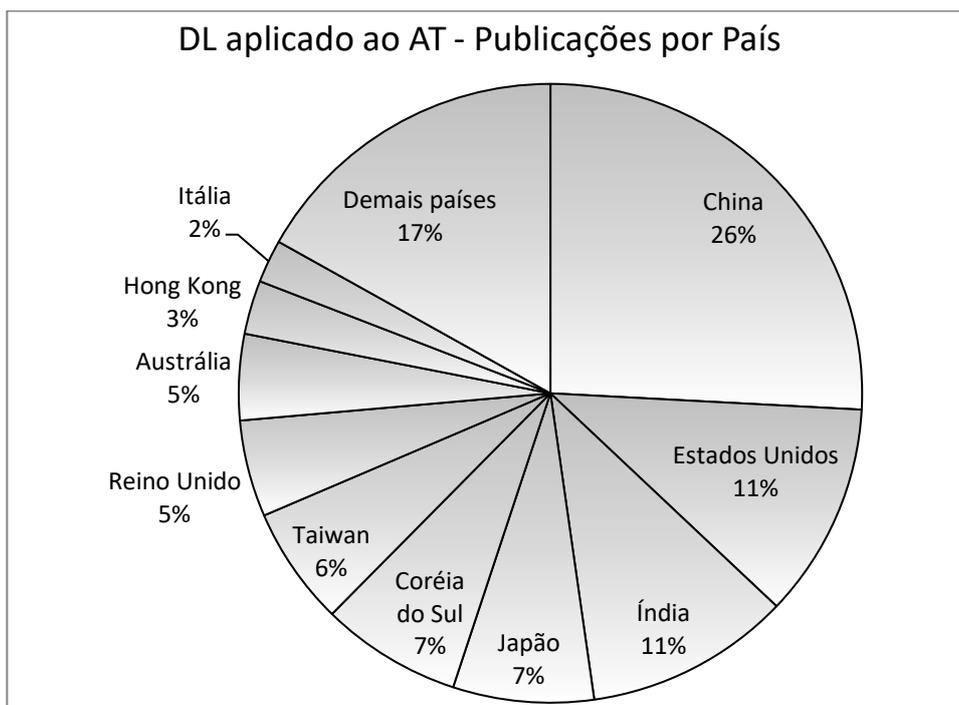
Tabela 21 – Dez Países que mais publicaram sobre abordagens de DL aplicadas ao AT

<b>Países</b>	<b>Publicações</b>
China	46
Estados Unidos	20
Índia	19
Japão	13
Coréia do Sul	13
Taiwan	11
Reino Unido	9
Austrália	8
Hong Kong	5
Itália	4
Demais países	30

Fonte: Scopus (2019)

O gráfico da Figura 23 apresenta a participação percentual dos dez países com maior produção científica sobre o tema, conforme a base Scopus.

Figura 23 – Países que mais publicaram abordagens de DL aplicadas ao AT



Fonte: Scopus (2019)

A Tabela 22 apresenta os vinte artigos mais citados dentre os 178 artigos resultantes da busca sobre a base Scopus, sobre abordagens de DL aplicadas ao AT.

Tabela 22 – Artigos sobre abordagens de DL aplicados ao AT mais citados

<b>Título</b>	<b>Autores</b>	<b>Fonte</b>	<b>Ano</b>	<b>CC</b>	<b>FWCI</b>
<i>Deep learning for event-driven stock prediction</i>	Ding X., et al.	IJCAI Artificial Intelligence	2015	132	22,23
<i>Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies</i>	Chong E., Han C., Park F.C.	Expert Systems with Applications	2017	80	16,77
<i>A deep learning framework for financial time series using stacked autoencoders and long-short term memory</i>	Bao W., Yue J., Rao Y.	PLoS ONE	2017	74	14,62
<i>Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models</i>	Kim H.Y., Won C.H.	Expert Systems with Applications	2018	28	11,12
<i>Big Data: Deep Learning for financial sentiment analysis</i>	Sohangir S., et al.	Journal of Big Data	2018	27	9,61
<i>Decision support from financial disclosures with deep neural networks and transfer learning</i>	Kraus M., Feuerriegel S.	Decision Support Systems	2017	25	5,47

<b>Título</b>	<b>Autores</b>	<b>Fonte</b>	<b>Ano</b>	<b>CC</b>	<b>FWCI</b>
<i>Forecasting stock prices from the limit order book using convolutional neural networks</i>	Tsantekidis A., et al.	2017 IEEE 19th Business Informatics	2017	21	16,26
<i>Artificial neural networks architectures for stock price prediction: Comparisons and applications</i>	Di Persio L., Honchar O.	Journal Circuits, Systems and Signal Processing	2016	19	2,27
<i>Listening to chaotic whispers: A deep learning framework for news-oriented Stock trend prediction</i>	Hu Z., et al.	WSDM 2018 - 11th ACM Web Search Data Mining	2018	18	26,08
<i>An innovative neural network approach for stock market prediction</i>	Pang X. et al.	Journal of Supercomputing	2018	18	13,61
<i>Predicting stock market trends by recurrent deep neural networks</i>	Yoshihara A., et al.	Computer Science	2014	18	1,42
<i>Stock price prediction using LSTM, RNN and CNN-sliding window model</i>	Selvin S. et al.	2017 Computing, Communications and Informatics	2017	17	8,96
<i>Deep learning for stock market prediction from financial news articles</i>	Vargas M.R., Lima B.S.L.P., Evsukoff A.G.	2017 IEEE Comput Intelligenc Virtual Environ Measurmt Systems Applic	2017	17	10,72
<i>Stock prediction using deep learning</i>	Singh R., Srivastava S.	Multimedia Tools and Applications	2016	16	3,14
<i>Financial time-series data analysis using deep convolutional neural networks</i>	Chen J.-F., et al.	2016 7th Cloud Computing and Big Data	2017	16	6,91
<i>A stock decision support system based on DBNs</i>	Zhu C., Yin J., Li Q.	Computational Information Systems	2014	14	1,02
<i>Deep neural networks for bot detection</i>	Kudugunta S., Ferrara E.	Information Sciences	2018	13	4,86
<i>Query-less: Predicting task repetition for nextgen proactive search and recommendation engines</i>	Song Y., Guo Q.	25th International WWW2016	2016	12	2,97
<i>Forecasting stock market crisis events using deep and statistical machine learning techniques</i>	Chatzis S.P., et al.	Expert Systems with Applications	2018	10	4,00
<i>Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations</i>	Gunduz H., Yaslan Y., Cataltepe Z.	Knowledge-Based Systems	2017	10	1,49

Fonte: Scopus (2019)

Os autores mais citados dentre os 178 artigos selecionados são apresentados na Tabela 23, contendo também o número de publicações, a contagem de citações, a razão do número de citações pelo número de publicações e o índice-h.

Tabela 23 – Autores sobre abordagens de DL aplicados ao AT mais citados

<b>Autor</b>	<b>Instituição Afiliação</b>	<b>País</b>	<b>Publicações</b>	<b>Citações</b>	<b>C/P</b>	<b>Índice-h</b>
Khoshgoftaar, T.	Florida Atlantic U.	EUA	610	11.114	18,2	53
Gabbouj, M.	Tampere U. Techn.	Finlândia	599	7.373	12,3	37
Liu, T-Y.	Microsoft Resrch Asia	China	187	5.986	32,0	35
Liu, T.	Harbin Inst Technology	China	246	3.831	15,6	27
Tefas, A.	Aristotle U. Thessaloniki	Grécia	314	3.500	11,1	29
Park, F. C.	Seoul National University	Coréia do Sul	158	3.479	22,0	29
Chang, V.	Teesside University	Reino Unido	250	2.811	11,2	28
Ferrara, E.	U. Southern California	EUA	100	2.778	27,8	29
Yin, J.	Dongguan U. Technology	China	262	2.145	8,2	23
Zhang, Y.	Hunan University	China	89	1.792	20,1	21
Soman, K. P.	Amrita Sch.Engineering	Índia	447	1.766	4,0	16
Wang, D.	Florida Atlantic U.	EUA	61	1.759	28,8	20
Iosifidis, A.	Aarhus Universitet	Dinamarca	123	1.210	9,8	18
Bian, J.	Microsoft Research Asia	China	42	1.116	26,6	15
Song, Y.	Google LLC	EUA	28	1.044	37,3	17
Liu, X.	Peking University	China	100	1.022	10,2	18
Chatzis, S.P.	Cyprus U. Technology	Chipre	88	921	10,5	15
Feuerriegel, S.	ETH Zürich	Suíça	60	520	8,7	14

Fonte: Scopus (2019)

### 2.6.3 Revisão Descritiva da Literatura

Para o estudo das frentes de pesquisa das abordagens de DL aplicadas ao AT, foram analisados os títulos, palavras-chave e resumos dos 50 artigos mais citados resultantes da busca da análise bibliométrica sobre as quatro bases: Scopus, Web of Science, ACM e IEEE Xplore. O critério de ordenação de artigos considerou a soma de citações nestas bases e o número de bases em que foram listados, buscando-se ponderar a seleção não apenas pelo número de citações, mas também pela presença nas bases consultadas.

A seleção dos documentos foi realizada mediante leitura dos resumos, obedecendo aos critérios da Tabela 24.

Tabela 24 – Critérios de seleção sobre abordagens de DL aplicadas ao AT

<b>Tipo de Critério</b>	<b>Critério</b>
Inclusão	Conteúdo se concentra em abordagens de DL aplicadas ao AT Conteúdo apresenta aspectos relevantes sobre abordagens de DL aplicadas ao AT
Exclusão	Conteúdo fora da área de interesse ou marginalmente relacionado ao tema de busca Documento não disponível para consulta online Documento duplicado

Fonte: autor

Os artigos resultantes foram analisados quanto a características como: dados de entrada, objetivos de saída, abordagens e desempenho.

### **Dados de Entrada**

Os dados de entrada mais utilizados são os dados de mercado, compostos pelas informações de alta frequência das variações dinâmicas dos livros de ordens e dos negócios realizados, contendo ativo, preço, quantidade e natureza (compra ou venda) e horários de execução (*timestamps*), apresentando-se na forma de séries temporais não lineares, não estacionárias e transformadas em retornos no tratamento de normalização ou padronização dos dados, apropriadas à abordagem por DL (CHONG; HAN; PARK, 2017; KIM; WON, 2018).

As notícias, em especial as de impacto econômico, podem se beneficiar dos métodos de interpretação de texto, processamento natural da linguagem, análise de sentimento e captura semântica, extraídas de quantidades imensas de dados, adequadas aos métodos de DL como CNN e LSTM (DING *et al.*, 2015; VARGAS; LIMA; EVSUKOFF, 2017; HU *et al.*, 2018; SOHANGIR; WANG; POMERANETS; KHOSHGOFTAAR, 2018).

Os indicadores de Análise Técnica são produzidos por softwares especializados que transformam os dados de mercado em características e padrões secundários, como médias móveis, formação de figuras, retas de suporte e resistência, entre outros, que são utilizados por algumas abordagens de DL (MURPHY, J., 1999; YOSHIHARA *et al.*, 2014; KHARE; DAREKAR; GUPTA; ATTAR, 2017).

A frequência de ocorrência de fontes de dados de entrada e a classificação por tipo são apresentadas na Tabela 25. Observe-se que um artigo pode apresentar mais de uma fonte de dados em sua pesquisa.

Tabela 25 – Fontes de dados das abordagens de DL aplicadas ao AT

<b>Fontes de Dados</b>	<b>Tipo de Dado</b>	<b>Número de Artigos</b>
Preços de Mercado	Dados de Mercado	45
Notícias	Dados Alternativos	10
Indicadores de Análise Técnica	Dados Analíticos	5
Balanços e Demonstrativos	Dados Fundamentais	4
Redes Sociais	Dados Alternativos	1
Volatilidade	Dados de Mercado	1

Fonte: Resultado da Pesquisa

### Objetivos de Saída

O desempenho de uma estratégia de AT é avaliado por meio da variável de saída, ou objetivo de saída (*target output*), comparado ao desempenho das abordagens de referência (*baseline methods*) que podem ser as tradicionais (sem técnicas de aprendizado) ou as de ML. A Tabela 26 mostra os objetivos de saída das abordagens analisadas e sua frequência de ocorrência:

Tabela 26 – Objetivos de saída das abordagens de DL aplicadas ao AT

<b>Objetivo de Saída (<i>Target Output</i>)</b>	<b>Número de Artigos</b>
Preço do Ativo	31
Direção da Tendência	10
Indicação de Compra ou Venda	4
Detecção de Crise	1
Detecção Falência	1
Escolha de Ativo	1
Volatilidade	1
Polaridade do Sentimento	1

Fonte: Resultado da Pesquisa

O objetivo de saída mais comum é o Preço do Ativo, em que a abordagem busca prever um preço alvo, ou um intervalo de preços, a ser atingido dentro de um determinado prazo (CHONG; HAN; PARK, 2017). A direção da tendência busca apenas indicar se os preços apresentarão comportamento de alta ou baixa, sem especificar valores ou faixas de valores

(VARGAS; LIMA; EVSUKOFF, 2017). As indicações de compra ou venda, buscam prever os momentos para assumir posições compradas ou e vendidas em razão da ocorrência de configurações características da análise técnica, como por exemplo, após formação de vales e topos (YOSHIHARA et al., 2014).

As abordagens que visam à identificação de crises buscam nos dados de mercado, divulgações corporativas e notícias uma indicação de risco sistemático. A indicação de falência busca nos dados informações sobre riscos e chances de falência de uma empresa, enquadrando-se na predição de riscos não sistemáticos (CHATZIS *et al.*, 2018). Por meio de uma criteriosa seleção de ativos, o gestor de carteiras (*portfolio manager*) estabelece o conjunto de ativos que maximiza seu desempenho ajustado ao risco que sua carteira pode assumir, cujo processo pode ser automatizado utilizando-se métodos de DL (SONG; GUO, 2016). As indicações de volatilidade desempenham papel crítico na precificação de derivativos, gestão de risco de carteiras e estratégias de proteção (*hedge*) (KIM; WON, 2018).

### Ocorrência das Abordagens

A Tabela 27 mostra as abordagens encontradas nos artigos e suas frequências de ocorrência. Um artigo pode eventualmente utilizar mais de uma abordagem. A LSTM e a CNN são as abordagens de DL mais frequentes.

Tabela 27 – Frequência de Abordagens de DL em artigos de AT

<b>Abordagens</b>	<b>Abreviação</b>	<b>Número de Artigos</b>
<i>Long Short-Term Memory</i>	LSTM	21
<i>Convolutional Neural Network</i>	CNN	20
<i>Deep Neural Network</i>	DNN	10
<i>Gated Recurrent Unit</i>	GRU	3
<i>Recurrent Neural Network</i>	RNN	3
<i>Stacked Autoencoders</i>	SAE	2
<i>Restricted Boltzmann Machine</i>	RBM	2
<i>Deep Belief Network</i>	DBN	1
<i>Recurrent Convolutional Neural Network</i>	RCNN	1
Outras – DL	DL	6
Outras – ML	ML	5
Outras – Tradicionais (não-ML)	não-ML	3

Fonte: Resultado da Pesquisa

Entre as abordagens de DL classificadas como outras, foram identificadas: *Word2Vec*, *Wavelet Transforms* e *Autoencoders*. As abordagens de ML encontradas são: SVM, *Principal Component Analysis* (PCA), Função Base Radial (RBF) sobre Redes Neurais e GA. As abordagens tradicionais (não-ML) identificadas nos artigos são essencialmente estatísticas, sem recursos de aprendizado, como: modelo de Heterocedasticidade Condicional Auto-Regressiva Generalizada (GARCH), e Médias Móveis Ponderadas Exponencialmente (EWMA).

A Tabela 28 mostra a ocorrência de abordagens de DL por objetivo de saída. Esta forma de visualização permite identificar as abordagens mais utilizadas para cada objetivo de saída.

Tabela 28 – Abordagens de DL aplicadas ao AT por objetivo de saída

Abordagem de DL	LSTM	CNN	DNN	GRU	RNN	SAE	RBM	RCNN	DBN
<b>Objetivo de Saída</b>									
Preço	15	11	6	2	3	2			
Direção	4	6	2	1			1	1	
Ponto de Compra/Venda	1	1	1						1
Seleção de Ativos		1							
Detecção de Falência		1							
Detecção de Crise			1						
Polaridade do Sentimento							1		
Volatilidade	1								
<b>Total das Abordagens</b>	<b>21</b>	<b>20</b>	<b>10</b>	<b>3</b>	<b>3</b>	<b>2</b>	<b>2</b>	<b>1</b>	<b>1</b>

Fonte: Resultado da Pesquisa

As estratégias orientadas à previsão de preços indicam utilização mais frequentemente da LSTM, seguida da CNN, com uma frequência menor para a DNN no sentido estrito, isto é, excluindo LSTM e CNN. As estratégias direcionais se concentram também na CNN e LSTM.

### Desempenho das Abordagens

Dos 50 artigos selecionados na revisão da literatura, 47 apresentam uma avaliação positiva do desempenho do DL comparativamente às abordagens de ML ou tradicionais (não-ML), seja quanto à precisão ou quanto à eficiência. Três artigos avaliaram o DL negativamente, ou com desempenho inferior às demais, conforme Tabela 29.

Tabela 29 – Avaliação geral de desempenho do DL aplicado ao AT

<b>Avaliação</b>	<b>Frequência</b>
Positiva	47
Negativa	3

Fonte: Resultado da Pesquisa

Dentre as abordagens mais frequentes de DL, o LSTM e o CNN apresentam 19 avaliações de desempenho favoráveis, com duas desfavoráveis para LSTM e uma para CNN. O desempenho por abordagem é apresentado na Tabela 30.

Tabela 30 – Avaliação de desempenho das abordagens de DL aplicadas ao AT

<b>Abordagens</b>	<b>Abreviação</b>	<b>Positiva</b>	<b>Negativa</b>
<i>Long Short-Term Memory</i>	LSTM	19	2
<i>Convolutional Neural Network</i>	CNN	19	1
<i>Deep Neural Network</i>	DNN	9	1
<i>Gated Recurrent Unit</i>	GRU	3	
<i>Recurrent Neural Network</i>	RNN	3	
<i>Restricted Boltzmann Machine</i>	RBM	2	
<i>Deep Belief Network</i>	DBN	1	
<i>Recurrent Convolutional Neural Network</i>	RCNN	1	

Fonte: Resultado da Pesquisa

### **Comparativo das abordagens LSTM e CNN**

No Quadro 8 apresenta-se a comparação das abordagens de DL de aplicação mais frequente em AT: LSTM e CNN, identificadas nos 50 artigos selecionados. São descritos os conceitos gerais de cada abordagem, suas aplicações gerais, aplicações específicas em AT, tipos de estratégias de AT, ocorrência da abordagem, objetivos de saída e avaliações de desempenho.

Quadro 8 – Comparativo das abordagens de DL mais frequentes

<b>Abordagens de DL</b>	<b><i>Long Shorty-Term Memory (LSTM)</i></b>	<b><i>Convolutional Neural Network (CNN)</i></b>
Conceitos Gerais	Baseadas em RNNs, possuem a capacidade de reter a memória de dados passados. Atuam com algoritmos de aprendizado baseados em gradiente. (HOCHREITER; SCHMIDHUBER, 1997)	Baseadas em Redes Neurais Multi-camadas, treinadas em algoritmo de <i>backpropagation</i> (LECUN; BENGIO, 1995).
Aplicações Gerais	Classificação, processamento e previsão sobre séries de dados. (WIERSTRA; SCHMIDHUBER; GOMEZ, 2005)	Reconhecimento de padrões visuais a partir de pixels de imagens com mínimo processamento (LECUN; BENGIO, 1995).
Aplicações em AT	Aplicável a dados sequenciais, não lineares, como sequências temporais como séries de preços, comportamento dinâmico dos livros de ofertas, volatilidade histórica, indicadores de análise técnica como médias móveis e índices de força relativa (BAO; YUE; RAO, 2017; KRAUS; FEUERRIEGEL, 2017; SELVIN <i>et al.</i> , 2017; KIM; WON, 2018).	Aplicável a dados não-estacionários, não lineares, característicos da microestrutura do mercado, como os livros de ordens limitadas durante um pregão eletrônico (DING <i>et al.</i> , 2015; CHEN <i>et al.</i> , 2016; DI PERSIO; HONCHAR, 2016; TSANTEKIDIS <i>et al.</i> , 2017; VARGAS; LIMA; EVSUKOFF, 2017).
Tipos de Estratégias	Previsão de preços de ativos, direção de tendências, recomendações de compra e venda (KRAUS; FEUERRIEGEL, 2017; SELVIN <i>et al.</i> , 2017; KIM; WON, 2018; PANG <i>et al.</i> , 2018).	Previsão de preços de ativos por eventos gerados por notícias, direção de tendências, previsão recomendações de compra e venda, escolha de ativos, previsão de falência (DING <i>et al.</i> , 2015; GUNDUZ; YASLAN; CATALTEPE, 2017; VARGAS; LIMA; EVSUKOFF, 2017).
Ocorrência nos artigos selecionados	21 artigos	20 artigos
Objetivos de Saída	Preço, Direção, Ponto de Compra ou Venda, Volatilidade.	Preço, Direção, Ponto de Compra ou Venda, Seleção de Ativos, Indicador de Falência.
Desempenho	19 avaliações positivas e 2 negativas	19 avaliações positivas e 1 negativa

Fonte: Resultado da Pesquisa

As CNNs, abordagem de grande frequência entre os artigos selecionados, se caracterizam pela especialização de redes neurais que atuam sobre dados armazenados em matrizes de grandes dimensões. Têm utilização na previsão em finanças porque preservam as características de tendências sem perder as informações da microestrutura dos mercados, aproveitando a totalidade dos dados e eventos dos LOBs, extraindo de forma mais precisa os padrões recorrentes em cada intervalo de tempo (TSANTEKIDIS *et al.*, 2017).

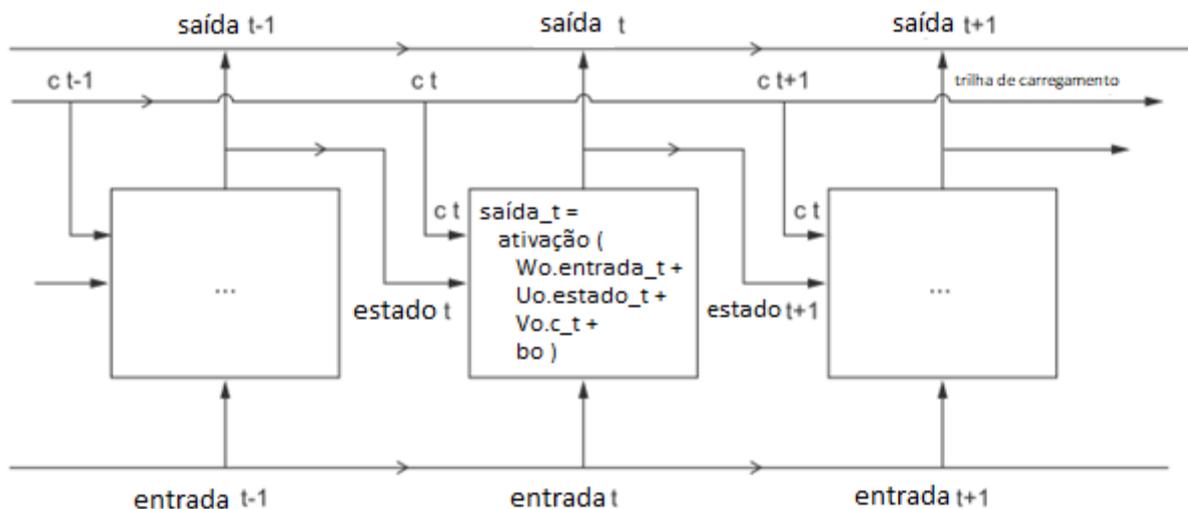
Ding *et al.* (2015) propuseram o uso de CNNs na previsão de preços de ações dirigida por eventos extraídos de notícias, para modelar as influências de curto e longo sobre o movimento de preços. Aproveitam os recentes avanços em poder computacional e em tecnologia de processamento de linguagem natural para obter representações de eventos

estruturados. No modelo de predição, buscam capturar a influência de notícias sobre períodos maiores que um dia, tratando-as como sequências de eventos diários usando uma CNN para realizar a composição semântica sobre a sequência de eventos de entrada e uma combinação de camadas para extrair as características globais mais representativas. Em seguida uma *Feed Forward Neural Network* (FFNN) associa as características globais às tendências de preços por meio de camadas ocultas e camadas de saída compartilhadas.

As FFNN não possuem memória dos dados passados, consideram apenas os dados de entrada correntes. Por sua vez, as RNNs conseguem memorizar dados de entrada recentes, embora percam tal capacidade em sequências de dados longas. Ainda que sejam adequadas a modelagem de séries temporais, seu aprendizado é dificultado pelo *vanishing gradient*, com variações nos pesos cada vez menores, impedindo a rede neural de continuar o aprendizado (BAO; YUE; RAO, 2017; JITHESH; SAGAYARAJ; SRINIVASA, 2017).

A LSTM, um caso particular de RNN, pode memorizar dados de sequências temporais longas. Seus nós fecham circuitos de realimentação recorrentes, de forma que os gradientes da função de perda sejam constantes e se propaguem sem o risco de desaparecerem (*vanishing gradients*) ou divergirem (*exploding gradients*), mantendo a capacidade de aprendizado da rede e tornando o treinamento mais eficiente. Os pesos nos circuitos recorrentes são condicionados ao contexto, controlados por uma unidade adjacente, conforme Figura 24 (HOCHREITER; SCHMIDHUBER, 1997; KIM; WON, 2018).

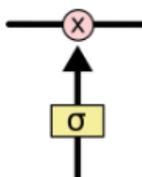
Figura 24 – Sequência de módulos da LSTM



A LSTM apresenta ainda a capacidade de capturar relações não lineares dos sinais temporais de prazos mais longos dos preços das ações (KRAUS; FEUERRIEGEL, 2017). Diante de tais características, justifica-se assim a preferência por este modelo entre os artigos selecionados.

As comportas (*gates*) são estruturas de chaveamento de dados baseadas em um nó de rede neural ativado pela função sigmoide, que retorna valores entre zero e um, alterando o sinal original por uma operação de multiplicação vetorial, conforme Figura 25 (KRAUS; FEUERRIEGEL, 2017; SELVIN *et al.*, 2017).

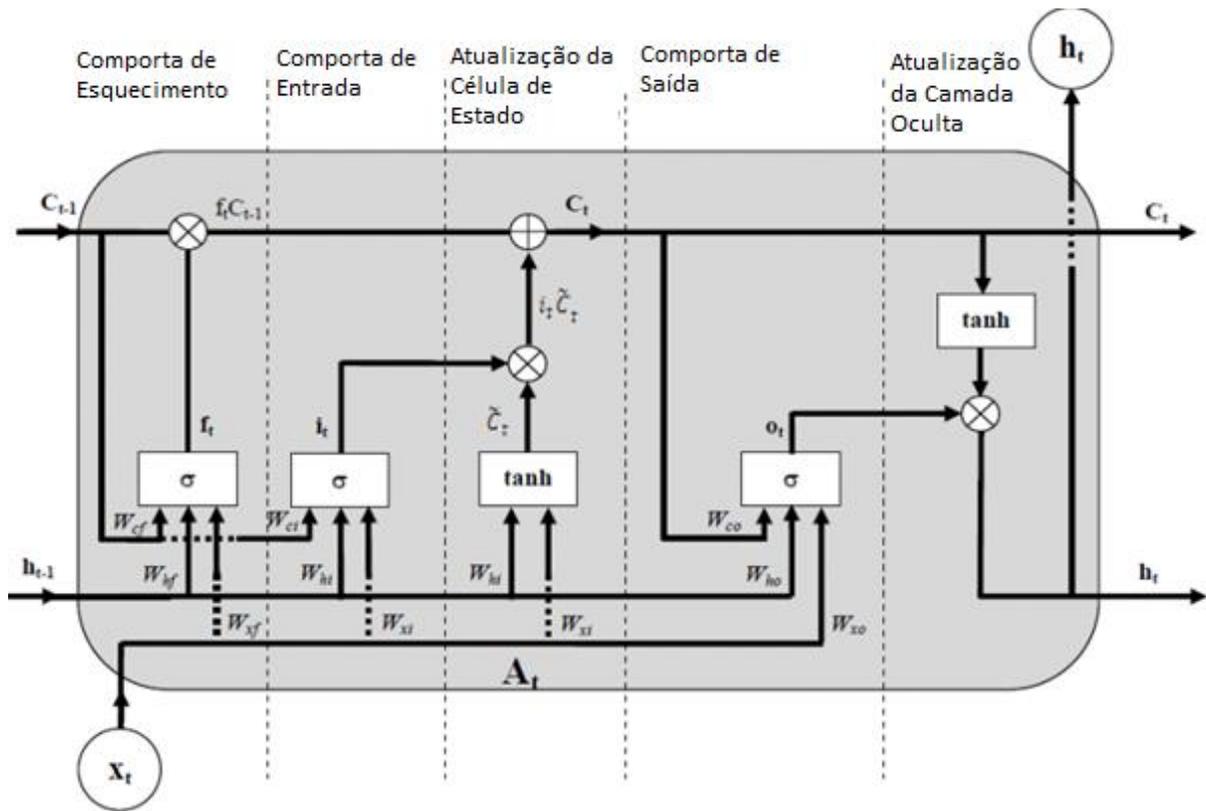
Figura 25 – Estrutura de comporta sigmoide de unidade LSTM



Fonte: Olah (2015)

O conceito central da LSTM é a célula de estado, que compreende a linha que transporta o sinal  $C_{t-1}$ ,  $f_i C_{t-1}$  e  $C_t$ . Funciona como uma correia de transmissão, permitindo que os dados percorram toda a estrutura, recebendo ou eliminando informações controladas pelas comportas (Figura 26). A LSTM possui comportas para controlar o conteúdo e o estado da unidade. A comporta de esquecimento (*Forget gate*), decide qual informação será eliminada. A comporta de entrada (*Input gate*) escolhe quais valores da atualização da célula de estado (*Cell state update*) serão atualizados. Esta célula traz os vetores candidatos a atualização da unidade LSTM ativados por meio de uma função de tangente hiperbólica, que retorna valores entre -1 e 1. A comporta do sinal de saída (*Output gate*) utiliza uma sigmoide para escolher quais os dados do estado atual que seguirão adiante. Na atualização da camada oculta (Hidden layer update) o sinal de carregamento,  $C_t$ , é ativado por uma função tangente hiperbólica e combinado por multiplicação ao sinal de estado da comporta de saída (HOCHREITER; SCHMIDHUBER, 1997; JITHESH; SAGAYARAJ; SRINIVASA, 2017; KIM; WON, 2018).

Figura 26 – Detalhamento da estrutura interna de unidade LSTM



Fonte: Jithesh, Sagayaraj e Srinivasa (2017)

### 3 MÉTODO

Este trabalho utiliza o método *Design Science Research* (DSR) em Sistemas de Informações para a condução da pesquisa aplicada. Este método compreende princípios, práticas e procedimentos para a produção e avaliação de um artefato (PEFFERS *et al.*, 2007).

O *Design Science* tem sua origem na engenharia e na ciência do artificial. (SIMON, 1996). Busca criar ideias, práticas, técnicas e produtos por meio da qual a análise, implementação do projeto, gestão e uso de sistemas de informação possam ser realizados de forma eficaz e eficiente (DENNING, 1997; TSICHRITZIS, 1998; HEVNER *et al.*, 2004).

Peffers *et al.* (2007) apresentam as diretrizes para a pesquisa em DSR aplicadas a Sistemas de Informações agrupadas em seis atividades:

#### 1 – Identificação do Problema e Motivação

Nesta atividade define-se o problema de pesquisa e se justifica o valor da solução proposta. O artefato deve oferecer uma solução eficaz para o problema. A justificativa motiva o pesquisador e a audiência a buscarem soluções, a compreender os resultados e o raciocínio utilizado no entendimento do problema (PEFFERS *et al.*, 2007).

#### 2 – Definição dos Objetivos

Infer-se o objetivo a partir da especificação do problema e sua viabilidade. Os recursos necessários incluem o conhecimento do estado dos problemas, das possíveis soluções atuais e sua eficácia. Os objetivos podem ser quantitativos, com indicadores que quantifiquem uma melhora de desempenho, ou qualitativos, que se concentrem em atributos que possam evidenciar uma melhor solução para o problema (PEFFERS *et al.*, 2007).

#### 3 – Projeto e Desenvolvimento

O artefato pode ser um constructo, um modelo, um método ou uma instanciação. O artefato deve ser eficaz quanto ao uso dos meios disponíveis para alcançar seus propósitos, obedecendo às leis no ambiente do problema (HEVNER *et al.*, 2004).

Em termos conceituais, o artefato pode ser qualquer objeto projetado no qual uma contribuição de pesquisa seja incorporada ao projeto. Deve-se também determinar a funcionalidade desejada do artefato e sua arquitetura, utilizando-se o conhecimento da teoria disponível e aplicável (PEFFERS *et al.*, 2007).

#### 4 – Demonstração

A viabilização e entendimento do funcionamento do artefato em atividade apropriada demonstra a resolução do problema (PEFFERS *et al.*, 2007).

#### 5 – Avaliação

A avaliação deve comparar os objetivos do artefato com os resultados observados na demonstração, podendo incluir qualquer evidência empírica apropriada ou prova lógica. Pode assumir várias formas, dependendo da natureza do local do problema e do artefato, como a comparação da funcionalidade do artefato com os objetivos da solução, utilizando medidas quantitativas de desempenho, como orçamentos, produção, pesquisas de satisfação, feedback do cliente e simulações; como também medidas de desempenho do sistema como o tempo de resposta ou a disponibilidade (PEFFERS *et al.*, 2007). Artefatos de Tecnologia da Informação (TI) podem ser avaliados em termos de funcionalidade, consistência e outros atributos relevantes de qualidade (HEVNER *et al.*,2004).

A utilidade pode ser demonstrada por meio de métodos descritivos, como argumentos fundamentados, obtidos da base de conhecimentos. Pode ser avaliada por meio de métodos analíticos, como análise estática e da arquitetura. A eficácia pode ser verificada por meio de testes funcionais (HEVNER *et al.*,2004).

#### 6 – Comunicação

A pesquisa deve ser efetivamente apresentada a audiências acadêmicas e orientadas a tecnologia e à gestão (HEVNER *et al.*,2004). A comunicação deve destacar a relevância do problema, a utilidade, ineditismo e eficácia do artefato (PEFFERS *et al.*, 2007).

## 4 RESULTADOS E DISCUSSÃO

A discussão dos resultados será apresentada de acordo com as seis atividades do método DSR, em conformidade com as proposições de Peffers *et al.* (2007).

### 4.1 Identificação do Problema e Motivação

As pesquisas em DL, com resultados importantes em diversas áreas do conhecimento, têm sido pouco explorada no mercado de capitais no segmento de AT, e menos ainda no mercado de ações brasileiro, conforme resultados da revisão da literatura apresentada na Fundamentação Teórica, sugerindo uma lacuna da pesquisa recente a ser investigada (LECUN, BENGIO; HINTON, 2015; GOODEFELLOW; BENGIO; COURVILLE, 2016; HOCHREITER; SCHMIDHUBER, 1997; BAO; YUE; RAO, 2017; CAVALCANTE *et al.*, 2016).

A motivação deste trabalho se fundamenta na pesquisa de abordagens de DL no estado da arte, aplicáveis ao mercado de capitais, que cooperem com a liquidez das operações e a redução das incertezas aos investidores, tendo como diferencial o direcionamento ao mercado de ações brasileiro, vis a vis ao desenvolvimento observado no mercado de ações de países desenvolvidos (HENDERSHOTT; JONES; MENKVELD, 2011; NUTI *et al.*, 2011; HARVARD UNIVERSITY - IACS, 2017; MIT TECHNOLOGY REVIEW, 2017).

### 4.2 Definição dos Objetivos

Propõe-se implementar uma abordagem de DL aplicada à negociação de ações baseada em algoritmos sobre ativos listados na bolsa de valores brasileira, B3, na forma de um artefato de software que utilize um modelo apropriado ao tratamento de séries de preços de ações.

### 4.3 Projeto e Desenvolvimento

As atividades de projeto e desenvolvimento foram estruturadas em: levantamento da teoria disponível e aplicável, e produção do artefato de software, a seguir apresentados:

#### a) Levantamento da teoria disponível e aplicável

O método DSR propõe a utilização do conhecimento da teoria disponível e aplicável para a determinação da funcionalidade do artefato e sua arquitetura (PEFFERS *et al.*, 2007).

Considerando-se que a essência da pesquisa científica é a produção de conhecimento e sua literatura é a manifestação desse conhecimento (OKUBO, 1997), a realização de análises bibliométricas possibilitam a mensuração da produção científica, enquanto as revisões da

literatura permitem a identificação das teorias existentes relativas ao fenômeno pesquisado, buscando as mais promissoras no intuito de transformar o conhecimento existente e, possivelmente, gerar conhecimento novo. Um levantamento bem executado permite identificar os modelos no estado da arte e compreender seus conceitos (CAUCHICK-MIGUEL; FLEURY, 2012). Nesse sentido, utiliza-se a Revisão Descritiva da Literatura, que tem por objetivo determinar a extensão para a qual um conjunto de estudos suporta ou revela padrões ou tendências com relação a proposições, teorias, metodologias ou descobertas existentes. Para cumprir este objetivo e garantir a generalidade dos resultados, a revisão descritiva coleta dados e analisa a frequência de assuntos, autores ou métodos na literatura existente, extrai dados de interesse específico para o estudo, como ano de publicação, métodos de pesquisa, técnicas de coleta de dados e robustez dos resultados. Assim procedendo, pode-se afirmar que os resultados da revisão descritiva representam o estado da arte em um domínio da pesquisa (PARÉ et al., 2014).

Por esta razão, a produção científica foi investigada de forma ampla por meio de análises bibliométricas e uma revisão descritiva da literatura, em que foram evidenciados o interesse recente da comunidade de pesquisa e os modelos no estado da arte.

Os motores de busca na literatura utilizaram as plataformas Scopus da Elsevier e Web of Science da Clarivate Analytics, que indexam periódicos internacionais de diferentes bases bibliográficas e editoras com artigos indexados e provenientes de periódicos que possuem Journal Citation Report (JCR) e SCImago Journal Rank (SJR) respectivamente (CAUCHICK-MIGUEL et al., 2017); além das bases ACM Digital Libray e IEEE Xplore. Este conjunto de plataformas e bases bibliográficas se adequam ao propósito da pesquisa em engenharia e ciência da computação (KITCHENHAM; CHARTERS, 2007; KITCHENHAM et al., 2010).

Os artigos sobre abordagens de DL aplicados ao AT, obtidos mediante Revisão Descritiva da Literatura (PARÉ et al., 2014), foram selecionados por critérios como contagem de citações, fatores de impacto dos periódicos, índice-h dos autores e pertinência ao tema, limitados até a data de 30 de setembro de 2019. Foram identificadas duas abordagens de DL mais frequentes, CNN e LSTM. A análise dos artigos e a comparação das abordagens resultou na escolha do modelo LSTM por sua capacidade de extração de características das relações não lineares dos sinais temporais dos preços das ações, da seletividade na retenção das informações passadas relevantes e descarte das não relevantes, como ruídos, e por evitar o *vanishing gradient* e o *exploding gradient*, resultantes da característica de memorização extensa, porém controlada e ajustada por comportas com funções sigmóides e tangentes hiperbólicas (HOCHREITER; SCHMIDHUBER, 1997; XIONG; NICHOLS; SHEN, 2016; KRAUS; FEUERRIEGEL, 2017;

SELVIN et al., 2017; FISCHER; KRAUSS, 2018; SHEN et al., 2018). Informações mais detalhadas constam da Revisão Descritiva da Literatura, no capítulo de Fundamentação Teórica, com uma análise dos dados de entrada, dos objetivos de saída, dos modelos mais frequentes, dos modelos CNN e LSTM, do funcionamento de redes LSTM e dos compartimentos de uma célula LSTM.

#### **b) Artefato de software**

O projeto consiste na instanciação de um artefato viável e de propósito definido, entendido como a implementação de um modelo que demonstra a sua viabilidade, possibilitando a avaliação da sua adequação aos propósitos pretendidos e a compreensão do seu funcionamento (HEVNER *et al.*, 2004). O artefato se apresenta na forma de um programa de computador composto de modelo lógico e código fonte que implementa o modelo LSTM, uma abordagem de DL, aplicado ao AT. O código fonte, está publicamente disponível na plataforma GitHub no endereço: <https://github.com/CamiloIlzo/Dados-de-Mercado-B3>, no arquivo “Modelo LSTM-B3.py”.

O programa utiliza a linguagem de código aberto Python v. 3.7.4, a plataforma de distribuição Anaconda v. 1.9.7, o ambiente integrado de desenvolvimento Spyder v. 3.3.6; e as bibliotecas: Numpy v. 1.16.5 para processamento matemático, Pandas v. 0.25.1 para operações de análise de dados, Matplotlib v. 3.1.1 para apresentação gráfica, Scikit-learn v. 0.21.3 (PEDREGOSA *et al.*, 2011) para pré-processamento e padronização de dados e Tensorflow v. 1.15 para a modelagem e treinamento da rede LSTM. A configuração do ambiente de desenvolvimento segue as orientações dos sites Python.org, Anaconda.com, TensorFlow.org.

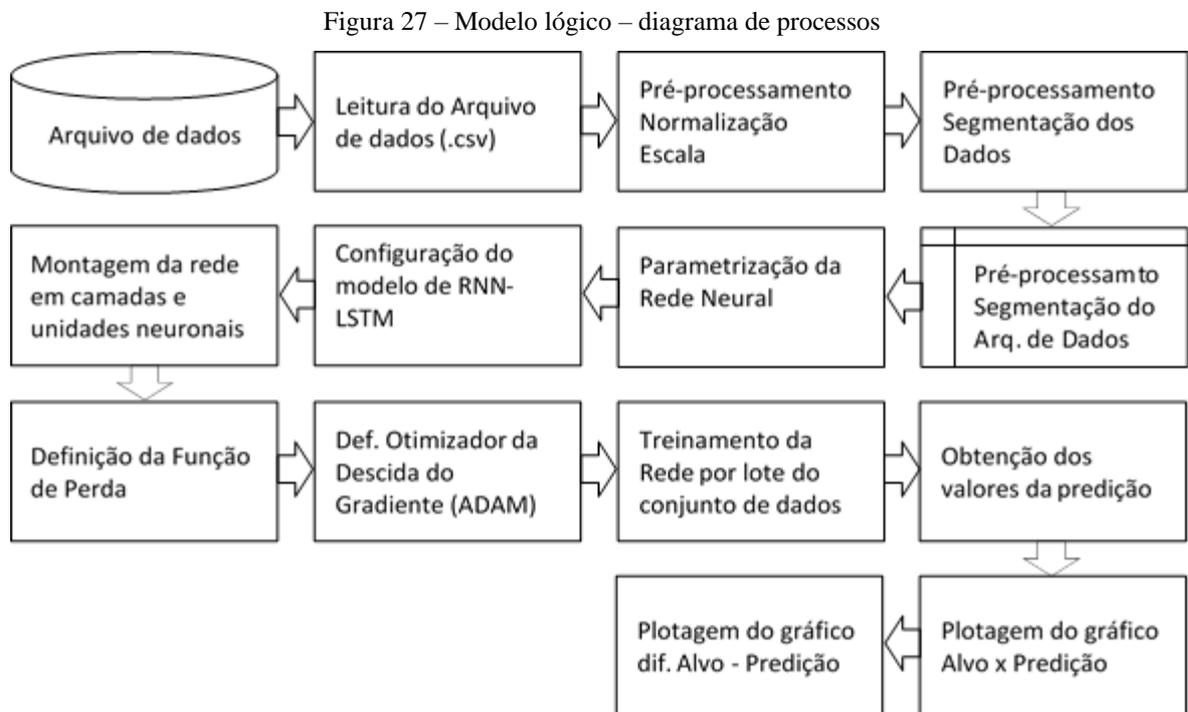
As parametrizações e o uso de bibliotecas de funções da abordagem LSTM utilizam configurações e recomendações de Karpathy (2019), Olah (2015), Rohrer (2019), Choudhury (2019) e Heinz (2019), a modelagem da rede LSTM projetada por Palai (2019) e disponibilizadas em regime código aberto (*open source*) no repositório GitHub, respeitando os direitos de propriedade intelectual.

O modelo foi executado em notebook Lenovo, modelo Ideapad S145, com 8Gb de memória RAM, processador Intel Core i5-8265U, e sistema operacional Microsoft Windows 10 Home Edition.

Em adição, foi desenvolvido um software em linguagem Python, registrado no INPI, conforme Anexo, para facilitar a interação com usuários, destinado à seleção de parâmetros, execução do modelo LSTM e exibição dos gráficos.

#### 4.4 Demonstração

As atividades de demonstração da viabilidade do artefato de software como solução do problema desta pesquisa, foram estruturadas de acordo com a Figura 27 e apresentadas a seguir.



Fonte: Resultado da Pesquisa

Inicialmente são declaradas as bibliotecas para a implementação do modelo: a biblioteca Numpy para a configuração do lote de processamento e carga das matrizes, Pandas para importação e manipulação de dados, Scikit-learn para a padronização da escala dos dados (MinMaxScaler), TensorFlow para modelagem da rede LSTM, como número de camadas, função de custo e treinamento, e Matplotlib para a plotagem de gráficos.

```

# Importação de bibliotecas de programas
import numpy as np
import pandas as pd
import sklearn
import sklearn.preprocessing
import matplotlib.pyplot as plt
import tensorflow as tf
  
```

### a) Dados de Entrada

Os dados de mercado foram obtidos por meio de acesso público ao endereço de disponibilização de arquivos do site da B3 (B3, 2019). Os arquivos compreendem os anos de 2008 a 2018, no formato COTAHIST\_Aaaaa.txt, onde ‘aaaa’ se refere ao ano, contendo as cotações dos papéis negociados. Foram realizados ajustes e adaptações na estrutura de dados dos arquivos, para lidar, por exemplo, com o desdobramento de ações, evitando-se distorções de escala.

Como critério de escolha dos ativos, foram identificadas as cinco ações mais líquidas do índice Ibovespa em 30 de setembro de 2019: ITUB4 (Itaú Unibanco PN), VALE3 (Vale ON), BBDC4 (Bradesco PN), PETR4 (Petrobras PN) e B3SA3 (B3 ON). ITUB4 foi desconsiderada em razão da fusão com o Banco Unibanco, anteriormente listada como ITAU4. B3SA3, que no início do período era listada como BVMF3, não figurava entre as cinco mais líquidas em 2008. Ao longo da década de 2010, o Bradesco adquiriu alguns bancos, como BERJ e HSBC, razão pela qual optou-se por desconsiderar BBDC4. Cabe observar que, em estudos posteriores, mais aprofundados e de natureza quantitativa, recomenda-se ajustar os efeitos dos eventos corporativos, como proventos e demais fatores.

Assim, foram extraídos os ativos PETR4 e VALE3, ajustados apenas quanto aos desdobramentos, e gerados os arquivos “PETR4.csv” e “VALE3.csv”, disponibilizados publicamente no repositório GitHub no endereço: <https://github.com/CamiloIlzo/Dados-de-Mercado-B3>.

### b) Leitura do Arquivo de Dados

Os arquivos no formato csv (colunas separadas por vírgulas) com 2719 linhas de dados cada, PETR4.csv e VALE3.csv, são lidos e seus dados transferidos para uma variável dataframe da biblioteca Pandas e são descartados os valores não válidos (N/A).

```
# Leitura do arquivo de dados

cotacoes = pd.read_csv('G:\Meu Drive\Data\PETR4.csv', index_col = 0)

cotacoesOHLC = cotacoes.copy()
cotacoesOHLC = cotacoesOHLC.dropna()
cotacoesOHLC = cotacoesOHLC[['Abert', 'Max', 'Min', 'Fech']]
```

### c) Pré-processamento: Padronização da escala

No Pré-processamento, inicialmente ocorre a Padronização de Escala do conjunto de dados. As redes neurais se beneficiam dessa padronização pois frequentemente utilizam funções de ativação como tangente hiperbólica ou sigmoide que retornam valores nos intervalos  $[-1,1]$  ou  $[0,1]$  respectivamente. Utiliza-se a função `MinMaxScaler`, que estabelece um intervalo de valores entre um mínimo e um máximo, neste caso entre zero e um, preservando a funcionalidade mesmo para valores de desvios-padrão muito pequenos e entradas de valor zero em conjuntos de dados esparsos (HEINZ, 2019; PALAI, 2019; SCIKIT-LEARN, 2020).

```
# Função de padronização de escala do conjunto de dados
def normescala(df):
    escalaminmax = sklearn.preprocessing.MinMaxScaler()
    df['Abert'] = escalaminmax.fit_transform(df.Open.values.reshape(-1,1))
    df['Max'] = escalaminmax.fit_transform(df.High.values.reshape(-1,1))
    df['Min'] = escalaminmax.fit_transform(df.Low.values.reshape(-1,1))
    df['Fech'] = escalaminmax.fit_transform(df['Fech'].values.reshape(-1,1))
    return df

# Normalização do dataframe
cotacoesOHLC_norm = cotacoesOHLC.copy()
cotacoesOHLC_norm = normescala(cotacoesOHLC_norm)
```

### d) Pré-processamento: Segmentação dos dados

Na sequência, a Segmentação dos Dados divide os registros em partes, sendo reservados 80% para o treinamento, outros 10% para a validação, e os 10% restantes para os testes. Assim, das 2719 linhas de dados, 2160 são destinadas ao treinamento, outras 271 destinadas à validação e as 271 restantes destinadas aos testes. O tamanho da sequência de dados foi definido para 20 pontos de dados.

```
valid_pct = 10
test_pct = 10
sequenc = 20

def dadoscarga(cotacoes, sequenc):
    matrizcotacoes = cotacoes.as_matrix()
    dadoscotacoes = []
    for cont in range(len(matrizcotacoes) - sequenc):
        dadoscotacoes.append(matrizcotacoes[cont: cont + sequenc])
    dadoscotacoes = np.array(dadoscotacoes);
```

```

valid_tam = int(np.round(valid_pct/100*dadoscotacoes.shape[0]));
test_tam = int(np.round(test_pct/100*dadoscotacoes.shape[0]));
trein_tam = dadoscotacoes.shape[0] - (valid_tam + test_tam);
x_trein = dadoscotacoes[:trein_tam,-1,:];
y_trein = dadoscotacoes[:trein_tam,-1,:];
x_valid = dadoscotacoes[trein_tam:trein_tam+valid_tam,-1,:];
y_valid = dadoscotacoes[trein_tam:trein_tam+valid_tam,-1,:];
x_test = dadoscotacoes[trein_tam+valid_tam:,-1,:];
y_test = dadoscotacoes[trein_tam+valid_tam:,-1,:];
return [x_trein, y_trein, x_valid, y_valid, x_test, y_test]

```

```

x_trein, y_trein, x_valid, y_valid, x_test, y_test = dadoscarga(cotacoesOHLC_norm,
sequenc)

```

### e) Parametrização da rede neural

Nesta fase configuram-se os hiperparâmetros. O tamanho do lote é o número de pontos de dados utilizado pelo modelo para computar o erro no processamento direto em avanço sobre a rede neural (feedforward), antes da retro propagação dos erros e ajuste de pesos. As épocas são o número de vezes que o treinamento do modelo será realizado sobre o conjunto de dados de treinamento (SINGH, 2019). São 4 entradas de dados OHLC (Abertura, Mínimo, Máximo, Fechamento), composta de duas camadas de 200 unidades neuronais cada e 4 saídas. A taxa de aprendizado fixada em 0,001, tamanho do lote (batch size) de 50 e número de épocas igual a 100. No tratamento dos dados de entrada e saída (placeholders), configura-se a matriz bidimensional para o armazenamento dos dados de entrada X, e o vetor saída da rede neural Y. Os dados de entrada do treinamento são randomizados e a função de obtenção do próximo lote (batch) configura o índice da época incrementando de um tamanho de lote.

```

# parametros
passosequenc = sequenc-1
numentradas = 4
numneurons = 200
numsaidas = 4
numcamadas = 2
txaprendizado = 0.001
tamlote = 50
numepocas = 100

trein_tam = x_trein.shape[0]
test_tam = x_test.shape[0]
tf.compat.v1.reset_default_graph()
X = tf.compat.v1.placeholder(tf.float32, [None, passosequenc, numentradas])

```

```

y = tf.compat.v1.placeholder(tf.float32, [None, numsaidas])
indsaida = 0;
matrizrandomiz = np.arange(x_trein.shape[0])
np.random.shuffle(matrizrandomiz)

def proxlote(tamlote):
    global indsaida, x_trein, matrizrandomiz
    inicia = indsaida
    indsaida += tamlote
    if indsaida > x_trein.shape[0]:
        np.random.shuffle(matrizrandomiz)
        inicia = 0
        indsaida = tamlote
    term = indsaida
    return x_trein[matrizrandomiz[inicia: term]], y_trein[matrizrandomiz[inicia: term]]

```

#### f) Configuração do modelo RNN-LSTM

Utiliza-se o método do TensorFlow `tf.contrib.rnn.BasicLSTMCell`, composto de 200 unidades neuronais, com duas camadas e função ativação Linear Exponencial Unit (ELU), dada por  $f(x) = x$ , para  $x \geq 0$ , e  $f(x) = a \cdot (e^x - 1)$ , para  $x < 0$ . As camadas de células neurais são combinadas para formar células recorrentes multicamadas (`rnn.MultiRNNCell`). Para simular a RNN sobre os passos de tempo da entrada utiliza-se a função `dynamic_rnn()` que retorna as ativações de saída e o último estado oculto na forma de tensores.

```

camadasrede = [tf.contrib.rnn.BasicLSTMCell(num_units=numneurons, activation=tf.nn.elu)
               for layer in range(numcamadas)]
neuronmulticam = tf.contrib.rnn.MultiRNNCell(camadasrede)
saidasrnn, states = tf.nn.dynamic_rnn(neuronmulticam, X, dtype=tf.float32)

```

#### g) Montagem da rede em camadas

As Camadas da Rede Neural são montadas por meio do método `tf.layers.dense`, implementando a operação de ativação da soma dos produtos (`dot`) da matriz de pesos pelo vetor de vieses do tensor `layer`. Os valores de saída do treinamento (`outputs`) resultam de uma composição de pesos e vieses sobre os dados de entrada 'x' da rede neural.

```

saidasrnnempilh = tf.reshape(saidasrnn, [-1, numneurons])
saidasempilh = tf.layers.dense(saidasrnnempilh, numsaidas)
saidas = tf.reshape(saidasempilh, [-1, passosequenc, numsaidas])
saidas = saidas[:, passosequenc-1, :]

```

## h) Definição da Função de Perda

A Função de Perda (Custo), ou Erro Quadrático Médio (MSE), avalia a diferença entre os resultados (*outputs*) e os valores esperados de validação 'y' (*validation*). O objetivo do algoritmo de treinamento é minimizar a Função de Custo. Após o processamento de um lote de dados (*batch*), os erros são aplicados sobre os pesos da rede neural de acordo com a taxa de aprendizado, da saída para a entrada, no procedimento chamado de retro propagação (*backpropagation*).

```
perda = tf.reduce_mean(tf.square(saidas - y))
```

## i) Otimizador da descida do gradiente (ADAM)

Para otimizar a atualização dos pesos da rede neural durante o treinamento iterativo no processo de minimização da função de custo, ou descida do gradiente, utiliza-se o otimizador *Adaptive Moment Estimation* (ADAM) por sua facilidade de implementação, eficiência computacional e por ser apropriado a grandes conjuntos de dados de características não estacionárias com gradientes esparsos ou que apresentem sinal de ruído de alta intensidade (KINGMA, 2015; RUDER, 2017; BROWNLEE, 2019).

```
otimizad = tf.compat.v1.train.AdamOptimizer(learning_rate=txaprendizado)
treinamtoproc = otimizad.minimize(perda)
```

## j) Treinamento da rede

Executa-se a iteração pelas épocas, que são os ciclos de avanço e retro propagação da rede neural. Para cada época e lote de dados, executa-se o otimizador da função de custo/perda e as predições são armazenadas no vetor de saída.

```
with tf.compat.v1.Session() as sesstrein:
    sesstrein.run(tf.compat.v1.global_variables_initializer())
    for linhadados in range(int(numepocas*trein_tam/tamlote)):
        lote_x, lote_y = proxlote(tamlote) # fetch the next training batch
        sesstrein.run(treinamtoproc, feed_dict={X: lote_x, y: lote_y})
        if linhadados % int(5*trein_tam/tamlote) == 0:
```

```

trein_errquadmed = perda.eval(feed_dict={X: x_trein, y: y_trein})
valid_errquadmed = perda.eval(feed_dict={X: x_valid, y: y_valid})
print('%0.2f Epocas: EQM treino/validacao = %0.6f/%0.6f'%(
    linhados*tamlote/trein_tam, trein_errquadmed, valid_errquadmed))
testpredicao_y = sesstrein.run(saidas, feed_dict={X: x_test})
testpredicao_y.shape

```

### k) Obtenção dos valores da predição

Os valores resultantes são armazenados na matriz *comp* para a plotagem dos gráficos de Predição versus Alvo, e da diferença entre Predição e Alvo.

```

comp = pd.DataFrame({'Col1':y_test[:,3], 'Col2':testpredicao_y[:,3]})

```

### l) Plotagem dos gráficos “Alvo x Predição” e de acurácia “Alvo – Predição”

Executa-se a plotagem do gráfico “Alvo x Predição” por meio do método *plt* da biblioteca Matplotlib. A linha de “Alvo” representa os preços históricos de fechamento obtidos da B3. A linha de “Predição” representa a saída do modelo LSTM resultante do processo de treinamento. O eixo das abscissas corresponde aos últimos 271 dias de pregão do arquivo de dados reservados para os testes. O eixo das ordenadas foi padronizado pelo critério MinMaxScaler, em que de cada preço é subtraído o valor mínimo da série de preços e o resultado é dividido pela diferença entre os valores máximo e mínimo, resultando em valores entre zero e um, otimizando o processamento da rede LSTM. Em seguida executa-se o gráfico de acurácia “Alvo – Predição”, com as diferenças entre os valores Alvo e Predição.

```

plt.figure(figsize=(10,5))
plt.plot(comp['Col1'], color='blue', label='Alvo')
plt.plot(comp['Col2'], color='red', label='Predicao')
plt.legend()
plt.show()

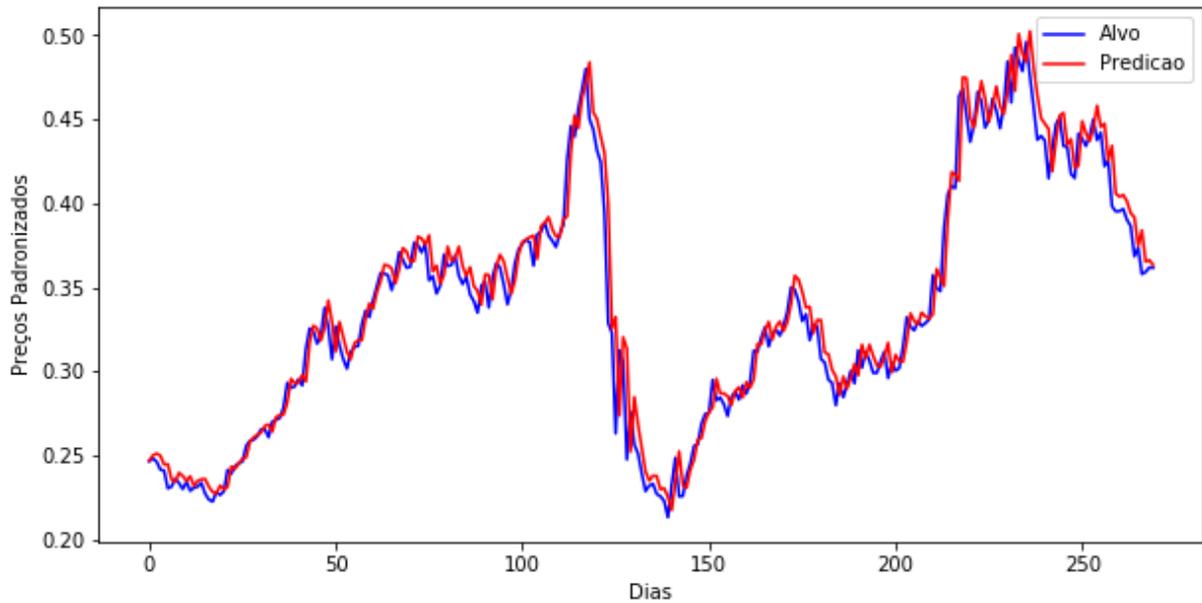
comp = pd.DataFrame({'Col':testpredicao_y[:,3] - y_test[:,3]})

plt.figure(figsize=(10,5))
plt.plot(comp['Col'], color='blue', label='Diferença')
plt.legend()
plt.show()

```

O gráfico resultante do processamento, com as séries de preços padronizadas Alvo e de Predição de PETR4 é apresentado na Figura 28.

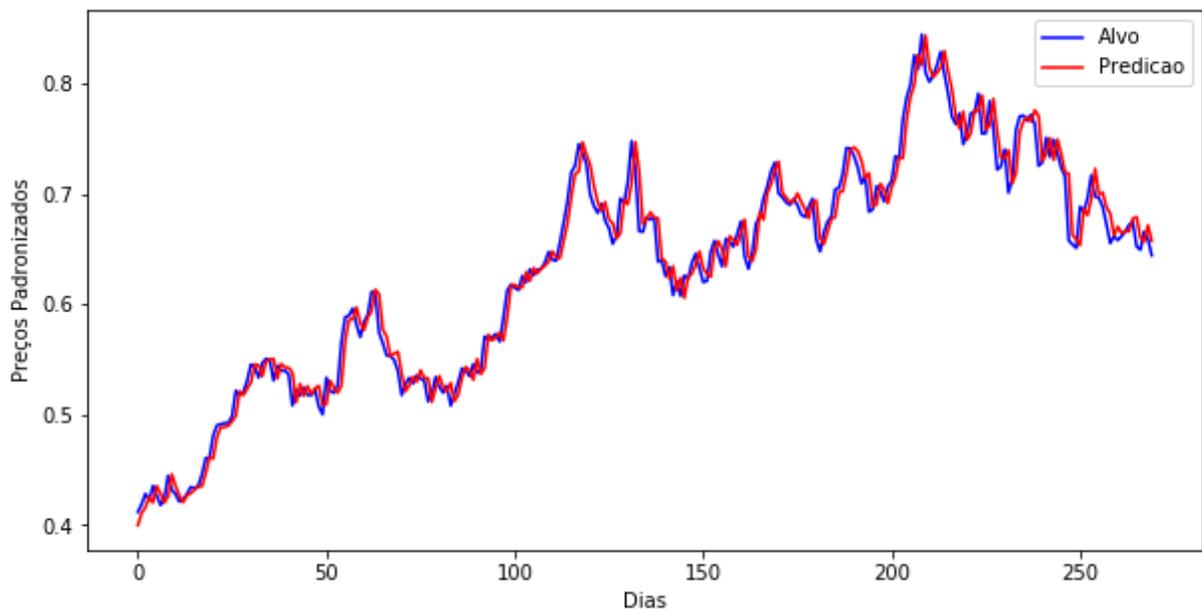
Figura 28 – PETR4: Alvo x Predição



Fonte: Resultado da Pesquisa

O gráfico de Alvo e de Predição de VALE3 é apresentado na Figura 29:

Figura 29 – VALE3: Alvo x Predição



Fonte: Resultado da Pesquisa

Os logs de execução contêm as mensagens de processamento do modelo LSTM, constando as iterações, as épocas e os erros quadráticos médios de treinamento e validação. Foram gravados no formato texto (.txt) e estão disponíveis publicamente no repositório GitHub no endereço: <https://github.com/CamiloIlzo/Dados-de-Mercado-B3>, sob os nomes: “PETR4-Log de treinamento.txt” e “VALE3-Log de treinamento.txt” e apresentados no Apêndice.

#### 4.5 Avaliação

*Design* do artefato: Utiliza-se o método analítico para conduzir a análise estática e da arquitetura para avaliação da funcionalidade e consistência do artefato, atributos qualitativos relevantes de um artefato de TI (HEVNER et al., 2004). Pela análise estática destacam-se as propriedades da rede LSTM, como a adequação à predição de séries de dados temporais longas, a retenção seletiva de informações passadas relevantes e descarte das não relevantes, e a capacidade de evitar o *vanishing gradient* e o *exploding gradient*, resultantes da característica de memorização extensa, porém controlada e ajustada por comportas. A análise da arquitetura se expressa no detalhamento da estrutura e funcionamento da rede e da célula LSTM, apresentados na Fundamentação Teórica (HOCHREITER; SCHMIDHUBER, 1997; XIONG; NICHOLS; SHEN, 2016; KRAUS; FEUERRIEGEL, 2017; SELVIN et al., 2017; FISCHER; KRAUSS, 2017; SHEN et al., 2018).

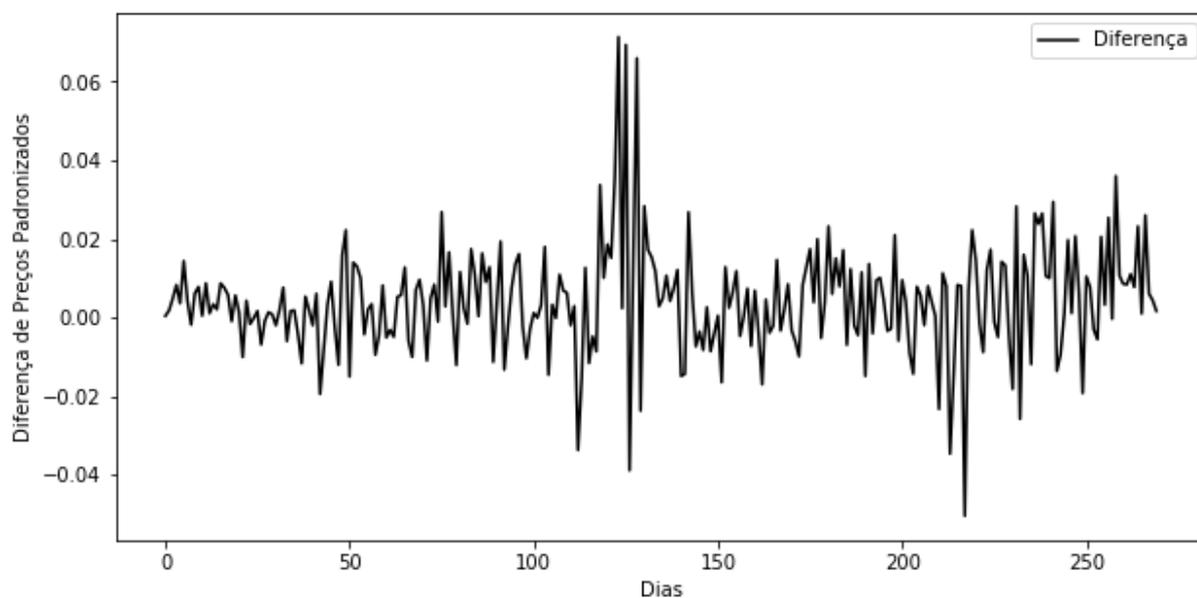
Relevância do problema, Utilidade e Ineditismo do Artefato: O artefato contribui para a pesquisa sobre DL em finanças aplicada ao mercado de capitais brasileiro, lacuna da pesquisa identificada em revisões da literatura, implementando uma abordagem no estado da arte, por meio de artefato de software viável da rede LSTM.

Eficácia: Nos gráficos de Alvo e Predição, observa-se a aderência dos valores da predição aos valores alvo nos períodos em que os preços apresentam formação de tendência. Nos momentos de reversão de preços e de acentuada volatilidade, a predição apresenta um atraso no reconhecimento do novo padrão, identificados nos gráficos de acurácia nos pontos de maior discrepância entre os preços padronizados de Alvo e de Predição.

Embora a inspeção visual possa parecer indicar boa capacidade de predição, deve-se levar em consideração uma investigação cuidadosa quanto à possibilidade de *overfitting* e efeitos de escala.

O gráfico de acurácia, com as diferenças entre os valores padronizados Alvo e de Predição de PETR4 é apresentado na Figura 30.

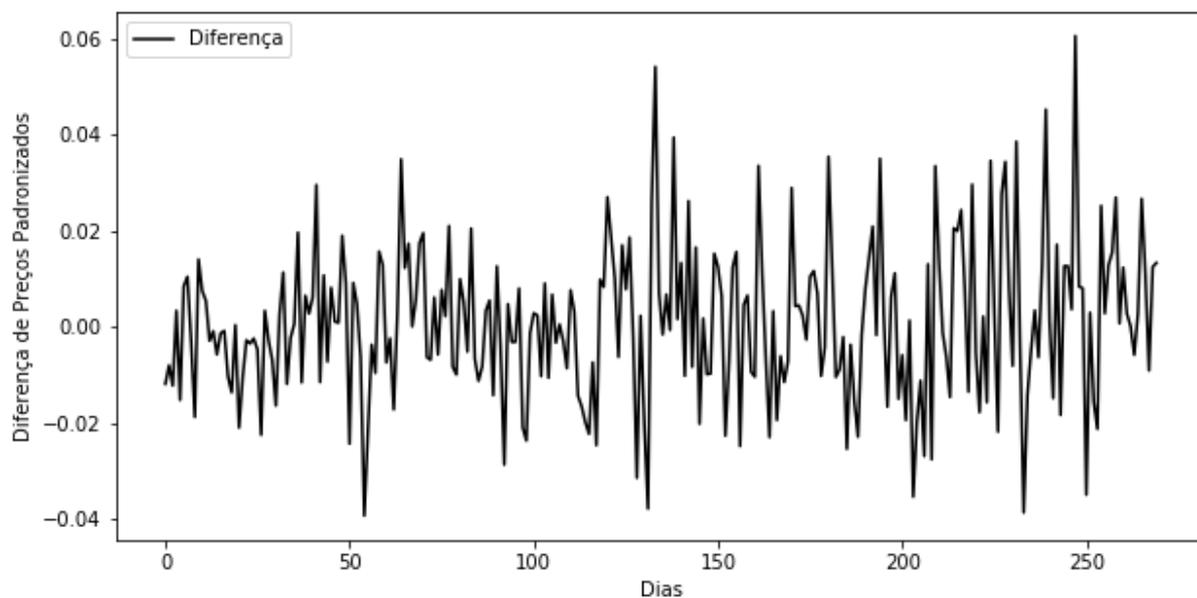
Figura 30 – Acurácia de PETR4: Alvo – Predição



Fonte: Resultado da Pesquisa

Na Figura 31 apresenta-se o gráfico de acurácia de VALE3:

Figura 31 – Acurácia de VALE3: Alvo – Predição



Fonte: Resultado da Pesquisa

As estatísticas de dispersão fornecem uma medida do grau de acurácia obtido pelo artefato e apresentadas na Tabela 31.

Tabela 31 – Medidas de Dispersão da Acurácia

	<b>PETR4</b>	<b>VALE3</b>
Média	0,004099	0,000778
Variância	0,000193	0,000259
Desvio Padrão	0,013906	0,016096
Coeficiente de Determinação $R^2$ (Alvo vs. Predição)	0,962369	0,975725

Fonte: Resultado da Pesquisa

Considerando-se os critérios de relevância do problema, do design, utilidade, ineditismo, e eficácia, confirmada pelos resultados obtidos, avalia-se que o artefato cumpre com o objetivo proposto, contribuindo para a pesquisa sobre DL em finanças, implementando modelo no estado da arte, por meio de artefato de software viável do modelo LSTM aplicado ao mercado de capitais brasileiro e justificando a escolha do modelo.

#### 4.6 Comunicação

Este trabalho foi elaborado visando públicos da comunidade científica, das áreas de engenharia de produção, gestão, tecnologia da informação e finanças. Para o cumprimento da atividade, a pesquisa deve ser apresentada e aprovada perante banca de avaliação de dissertação de mestrado em gestão e tecnologia em sistemas produtivos. A aprovação e disponibilização na forma impressa e digital representa a comunicação formal da pesquisa. O artefato apresentado será objeto de artigo a ser encaminhado a um periódico qualificado e com linha de pesquisa adequada para o tema.

## 5 CONCLUSÃO

Este trabalho identificou uma abordagem de *Deep Learning* aplicada ao *Algorithmic Trading* mediante revisão descritiva da literatura, seleção dos artigos pertinentes e análise das abordagens aplicáveis, resultando na escolha da rede LSTM, que foi implementada como artefato na forma de software para negociação de ações na bolsa de valores brasileira B3.

Os dados de entrada utilizados são preços de fechamento diários de um período de dez anos, de 2008 a 2018, e padronizados para evitar vieses de escala. Os resultados evidenciaram a aderência entre os preços padronizados Alvo e de Predição nos momentos de manutenção de tendência, apresentando maior discrepância nos momentos de reversão de preços ou de alta volatilidade. As variâncias da acurácia para PETR4 e VALE3 foram 0,000193 e 0,000259 respectivamente. Todavia, a interpretação de tais resultados merece cautela, com uma cuidadosa investigação quanto à possibilidade de *overfitting*.

Reconhece-se, entretanto, as limitações desta pesquisa quanto à capacidade de processamento disponível, limitando o conjunto de dados aos preços de fechamento. Para pesquisas futuras, sugere-se a utilização de preços intradiários e dados dos livros de oferta, com o devido dimensionamento da capacidade de processamento. Recomenda-se também utilizar ações de outros setores; outros ativos, tais como contratos futuros, commodities e moedas; e outras fontes de dados, como indicadores da análise técnica, fundamentalista, macroeconômica e comportamental.

Quanto ao tipo de rede, um possível desdobramento, seria a aplicação da abordagem *Gated Recurrent Unit* (GRU), um tipo de RNN, com características semelhantes à LSTM quanto à predição de séries temporais, porém com arquitetura mais simples e menos operações sobre tensores. Pode-se realizar também um estudo sobre o ajuste de hiperparâmetros da rede neural.

Vale destacar a aplicabilidade desse modelo em outras frentes de pesquisa. Em artigo sobre o efeito da temperatura na disseminação do coronavírus, Bukhari e Jameel (2020), pesquisadores do MIT, recomendaram a utilização de DL, em especial a LSTM, para estudos futuros, como modelo no estado da arte para predição de eventos em séries temporais de dados.

## REFERÊNCIAS

ACM Digital Library. Disponível em: <<https://dlnext.acm.org>>. Acesso em 30 set. 2019.

ALPAYDIN, E. Machine Learning. **Advanced Review**. Wiley, v. 3, pp. 195-203, 2011.

B3. Dados Históricos. Disponível em <[http://www.bmfbovespa.com.br/pt\\_br/servicos/market-data/historico/mercado-a-vista/series-historicas/](http://www.bmfbovespa.com.br/pt_br/servicos/market-data/historico/mercado-a-vista/series-historicas/)>. Acesso em 14 nov. 2019.

BAO, W.; YUE, J.; RAO, Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. **PLoS ONE**. 2017.

BELLMAN, R. E. **An Introduction to Artificial Intelligence: Can Computers Think?** Boyd & Fraser Publishing Company, 1978.

BENGIO, Y.; COURVILLE, A.; PASCAL, V. Representation learning: a review and new perspectives. **ArXiv e-prints**, 2012.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. **IEEE Pattern Analysis Machine Intelligence**. 2013

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. **Journal of Machine Learning Research**. 2012.

BLACK, B.; GILSON, R. **Venture capital and the structure of capital markets. Banks versus stock markets**. Journal of Financial Economics, Vol.47, Issue 3, pp.243-277, 1998.

BOYD, S. et al. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. **Foundations and Trends in Machine Learning**. v. 3, n. 1, pp. 1-122, 2010.

BREIMAN, L. Random Forests. **Machine Learning**, v. 45, pp. 5-32. 2001.

BROGAARD, J.; HENDERSHOTT, T.; HUNT, S.; YSUSI, C. High Frequency Trading and the Execution Costs of Institutional Investors. **The Financial Review**. v. 49, 2014.

BROGAARD, J.; HENDERSHOTT, T.; RIORDAN, R. High Frequency Trading and Price Discovery. **The Review of Financial Studies**. v. 27, pp. 2267-2306, 2014.

BROGAARD, J.; HENDERSHOTT, T.; RIORDAN, R. High Frequency Trading and the 2008 short-sale ban. **Journal of Financial Economics**. v. 124, 2017.

BROWNLEE, J. Machine Learning Mastery. Disponível em: <<https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>>. Acesso em 14 Nov. 2019.

BRYMAN, A.; BELL, E. **Business Research Methods**. 3 ed. Oxford University Press, 2011.

BUKHARI, Q.; JAMEEL, Y. Will coronavirus pandemic diminish by summer? **MIT. SSRN**, 2020.

CAUCHICK-MIGUEL, P. A.; FLEURY, A. **Metodologia De Pesquisa em Engenharia de Produção**. 2ª.ed.

Capítulo 2 – Planejamento do Projeto de Pesquisa e Definição do Modelo Teórico. Elsevier, 2012.

CAUCHICK-MIGUEL, P. A.; MARTINS, R. A. **Metodologia De Pesquisa em Engenharia de Produção**. 2ª.ed. Capítulo 3 – Abordagens Quantitativa e Qualitativa. Elsevier, 2012.

CAUCHICK-MIGUEL, P. A.; CAMPOS, L. M. S.; JABBOUR, C. J. C.; JABBOUR, A. B. L. S. **Elaboração de Artigos Acadêmicos: estruturas, métodos e técnicas**. 1ª ed. – Rio de Janeiro: Elsevier, 2017.

CAVALCANTE, C. C. et al. Computational Intelligence and Financial Markets: A Survey and Future Directions. **Expert Systems with Applications**. v 55, 2016.

CHABOUD, A. P.; CHIQUOINE, B.; HJALMARSSON, E.; VEGA, C. Rise of the Machines: Algorithmic Trading in the Foreign Exchange Market. **The Journal of Finance**. v. 69, n. 5, 2014.

CHAN, E. P. **Quantitative trading: how to build your own algorithmic trading business**. Wiley, 2009.

CHANG, C. C., LIN C.-J. LIBSVM: A Library for support vector machines. **ACM Transactions on Intelligent Systems Technology**, v.2, n3, 2011.

CHATZIS, S. P. et al. Forecasting stock market crisis events using deep and statistical machine learning techniques. **Expert Systems with Applications**. 2018

CHOLLET, F. **Deep Learning with Python**. Manning Publications Co., 2018.

CHONG, E.; HAN, H.; PARK, F. C. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. **Expert Systems with Applications**, v. 83, pp. 187-205, 2017.

CHOUDHURY, A. Disponível em: <<https://blog.usejournal.com/@aniruddha.choudhury94>>. Acesso em 14 nov. 2019.

CRESWELL, J. W. Research Design. **Qualitative, Quantitative, and Mixed Methods Approaches**. SAGE Publications, 2014.

DAMODARAN, A. **Damodaran on Valuation. Security Analysis for Investment and Corporate Finance**. 2nd ed. John Wiley & Sons, 2012.

DAMODARAN, A. **Investment Valuation. Tools and Techniques for Determining the Value of Any Asset**. 3rd ed. New Jersey: John Wiley & Sons, 2012.

DEAN, J., et al. Large scale distributed deep networks. **Advances in Neural Information Processing Systems**. 2012.

DENNING, P. J. A New Social Contract for Research, **Communications of the ACM**. February 1997, pp. 132-134. 1997.

DIAMOND, P. The Role of a Stock Market in a General Equilibrium Model with Technological Uncertainty. **The American Economic Review**. Vol. 57, No. 4 Sep. 1967, pp. 759-776.

DING, X.; ZHANG, Y., LIU, T.; DUAN, J. Deep learning for event-driven stock prediction. **IJCAI Artificial Intelligence**. 2015

DONG, C.; LOY, C. C.; HE, K.; TANG, X. Image Super-Resolution Using Deep Convolutional Networks. **IEEE Pattern Analysis Machine Intelligence**. 2016.

DRESCH, A.; LACERDA, D. P.; ANTUNES JR., J. A. V. **Design Science Research. Método de Pesquisa Avançado da Ciência e Tecnologia**. Porto Alegre: Bookman, 2015.

DUNIS, C.; WILLIAMS, M. Applications of Advanced Regression Analysis for Trading and Investment. Cap 1 in DUNIS, C.; LAWS, J.; NAIM, P. **Applied Quantitative Methods for Trading and Investments**. Wiley, 2003.

DUNIS, C.; JALILOV, J. Neural Network Regression and Alternative Forecasting Techniques for Predicting Financial Variables, **Neural Network World**, 2, 113–139, 2002.

EASLEY, D.; PRADO, M. M. L.; O'HARA, M. The Microstructure of the 'Flash Crash'. Flow toxicity, Liquidity Crashes and the Probability of Informed Trading. **The Journal of Portfolio Management**, v. 37, n. 2, p. 118-128, 2011.

ELTON, E.; GRUBER, M. **Modern Portfolio Theory and Investment Analysis**. 5 ed. New York: Wiley, 1995.

ENKE, D.; THAWORNWONG, S. The use of data mining and neural networks for forecasting stock market returns. **Expert Systems with Applications**. 2005

ERHAN, D.; BENGIO, Y.; COURVILLE, A.; MANZAGOL, P-A.; VINCENT, P. Why does unsupervised pre-training help deep learning? **Journal of Machine Learning Research**. 2010.

FARABET, C.; COUPRIE, C.; NAJMAN, L.; LECUN, Y. Learning Hierarchical Features for Scene Labeling. **IEEE Pattern Analysis Machine Intelligence**. 2013.

FAMA, E. Efficient Capital Markets: A Review of Theory and Empirical Work. **The Journal of Finance**, 1970.

FAWCETT, T. An introduction to ROC analysis. **Pattern Recognition Letters**. v. 27, 2005.

FILIPPINI, R. Operations management research: some reflections on evolution, models and empirical studies in OM. **International Journal of Operations and Production Management**, v. 17, n.7, p. 655-70, 1997.

FISCHER, T.; KRAUSS, C. Deep learning with Long Short-Term Memory Networks for Financial Market Predictions. **European Journal of Operation Research**. v. 270. 2017.

FIX TRADING. FIX Protocol. Disponível em: <[www.fixtrading.org](http://www.fixtrading.org)>. Acesso em 07 jun. 2018.

GADRE-PATWARDHAN, S.; KATDARE V. V.; JOSHI M. R. A Review of Artificially Intelligent Applications in the Financial Domain. Cap. 1 in DUNIS, C. L.; MIDDLETON, P. W.; KARATHANASOPOULOS, A.; THEOFILATOS, K. **Artificial Intelligence in Financial Markets**. Palgrave MacMillan, 2016.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow**. O'Reilly, 2017.

GOODEFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016.

GUO, X.; LAI, T. L.; SHEK, H.; WONG, S. P. **Quantitative trading - Algorithms, Analytics, Data, Models**,

**Optimization.** CRC Press. Taylor & Francis Group, 2017.

HAGENAU, M.; LIEBMANN, M., NEUMANN, D. Automated news reading: Stock price prediction based on financial news using context-capturing features. **Decision Support Systems.** 2013.

HAGSTROMER, B.; NORDÉN, L. The diversity of high frequency traders. **Journal of Financial Markets.** v.16, 2013.

HARVARD UNIVERSITY - IACS (INSTITUTE for APPLIED COMPUTATIONAL SCIENCE). **Symposium Data, Dollars, and Algorithms: The Computational Economy.** Publ. 19-jan-2017. Disponível em: <[iacs.seas.harvard.edu/event/annual-iacs-symposium-data-dollars-and-algorithms-computational-economy?delta=0](https://iacs.seas.harvard.edu/event/annual-iacs-symposium-data-dollars-and-algorithms-computational-economy?delta=0)>. Acesso em 19-set-2018.

HASBROUCK, J.; SAAR, G. Low-latency trading. **Journal of Financial Markets.** v. 16, 2013.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning. Data Mining, Inference, and Prediction.** 2 ed. Springer, 2009.

HEINZ, S. A simple deep learning model for stock price prediction using TensorFlow. Disponível em: <<https://medium.com/mlreview/a-simple-deep-learning-model-for-stock-price-prediction-using-tensorflow-30505541d877>>. Acesso em 14 nov. 2019.

HEIZER, J.; RENDER, B.; MUNSON, C. **Principles of Operations Management. Sustainability and Supply Chain Management.** 10<sup>th</sup> ed. Pearson, pp. 42-43. 2017.

HENDERSHOTT, T.; JONES, C. M.; MENKVELD, A. J. Does Algorithmic Trading Improve Liquidity? The **Journal of Finance.** v. 66 n. 1, 2011.

HENDERSHOTT, T.; RIORDAN, R. Algorithmic Trading and the Market for Liquidity. **Journal of financial and quantitative analysis.** v. 48, n. 4, p. 1001–1024, 2013.

HEVNER, A. R.; MARCH, S. T.; PARK, J.; RAM, S. Design Science in Information Systems Research. **MIS Quarterly.** v. 28, n. 1, pp. 75-105, Mar. 2004.

HILPISCH, Y. Algorithmic Trading in Less Than 100 Lines of Python Code. 2018. Disponível em: <[www.Oreilly.Com/Learning/Algorithmic-Trading-In-Less-Than-100-Lines-Of-Python-Code](http://www.Oreilly.Com/Learning/Algorithmic-Trading-In-Less-Than-100-Lines-Of-Python-Code)>. Acesso em 19 Set. 2018.

HJALMARSSON, E.; BENOS, E.; BRUGLER, J.; ZIKES, F. Interactions among High Frequency Traders. **Journal of Financial and Quantitative Analysis.** v.52 n.4, 2017.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. **Neural Computation,** 1997.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer Feedforward Networks Are Universal Approximators, **Neural Networks,** 2, 359–366, 1989.

HU, Z.; LIU, W.; BIAN, J.; LIU, X.; LIU, T-Y. Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction. **ACM Web Search and Data Mining,** Feb. 2018.

HUANG, C.-J.; YANG, D.-X.; CHUANG, Y.-T. Application of wrapper approach and composite classifier to the stock trend prediction. **Expert Systems with Applications.** 2008.

HULL, J. **Options, Futures, and Other Derivatives**. 10th ed. Pearson, 2018.

IEEE Spectrum. Hedge Funds Look to Machine Learning, Crowdsourcing for Competitive Advantage. 2017. Disponível em: <<https://spectrum.ieee.org/tech-talk/computing/software/fintech-trends-hedge-funds-look-to-machine-learning-crowdsourcing-for-competitive-advantage>>. Acesso em 18 abr. 2019.

IEEE Xplore. Acesso em 30 set. 2019. Disponível em <<https://ieeexplore.ieee.org/>> (2019).

JACOBS, F. R.; CHASE, R. B. **Operations and Supply Chain Management**. 15 ed. McGraw-Hill, pp. 24-25, 2018.

JI, S.; XU, W.; YANG, M.; YU, K. 3D Convolutional neural networks for human action recognition. **IEEE Pattern Analysis Machine Intelligence**. 2013

JIA, Y. et al. Caffe: Convolutional Architecture for Fast Feature Embedding. **ACM Conference on Multimedia**. 2014

JITHESH, V.; SAGAYARAJ, M. J.; SRINIVASA, K. G. LSTM recurrent neural networks for high resolution range profile-based radar target classification. **2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)**. IEEE Xplore. 13 jul. 2017.

JOHNSON, B. **Algorithmic Trading & DMA – An introduction to direct access trading strategies**. London: 4MyelomaPress, 2010.

JONES, N. The Learning Machines. Weekly Journal of Science. **Nature**. Vol. 505, Jan., p. 146, 2014.

KAHNEMAN, D. **Thinking Fast and Slow**. Farrar Straus Giroux, 2011

KARATHANASOPOULOS, A.; MIDDLETON, P. W.; THEOFILATOS, K.; GEORGOPOULOS, E. GETrader: A New Standalone Tool for Constructing Trading Strategies with Gene Expression Programming. Cap. 4 in DUNIS, C. L.; MIDDLETON, P. W.; KARATHANASOPOULOS, A.; THEOFILATOS, K. **Artificial Intelligence in Financial Markets**. Palgrave MacMillan, 2016.

KARPATY, A. Disponível em: <<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>>. Acesso em 14 nov. 2019.

KAUFMAN, P. J. **Trading Systems and Methods**. 5 ed. Wiley, 2013.

KELLEHER, J. D.; MAC NAMEE, B.; D'ARCY, A. **Fundamentals of Machine Learning for Predictive Data Analytics. Algorithms, Worked Examples, and Case Studies**. MIT Press, 2015.

KEARNS, M.; NEVMYVAKA, Y. Machine Learning for Market Microstructure and High-Frequency Trading. In: EASLEY, D.; PRADO, M. M. L.; O'HARA, M. (Orgs.) **High-Frequency Trading: New Reality for Traders, Markets and Regulators**. Risk Books, 2013.

KHARE, K.; DAREKAR, O.; GUPTA, P.; ATTAR, V. Z. Short Term Stock Price Prediction using Deep Learning. **2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, Proceedings - RTEICT 2017**, pp. 482-486, 2017.

KIM, H. Y.; WON, C.H. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. **Expert Systems with Applications**. v. 103, pp. 25-37. 2018

KINGMA, D. P. Adam: A Method for Stochastic Optimization. arXiv:1412.6980v9 [cs.LG]. **Conference Paper ICLR**, 2015.

KIRILENKO, A. A.; LO, A. W. Moore's Law versus Murphy's Law: Algorithmic Trading and Its Discontents. **Journal of Economics Perspectives**. v.27 n.2, 2013.

KIRILENKO, A. A.; KYLE, A. S.; SAMADI, M.; TUZUN, T. The Flash Crash: High-Frequency Trading in an Electronic Market. **The Journal of Finance**. v. 72, n. 3, 2017.

KISSELL, R. **The Science of Algorithmic Trading and Portfolio Management**. Elsevier, 2014.

KITCHENHAM, B. A.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. version 2.3. **EBSE Technical Report – Keele University – University of Durham**, 2007.

KITCHENHAM, B. A. *et al.*; Systematic literature reviews in software engineering – A tertiary study. **Information and Software Technology**, v. 52, pp. 792-805, 2010.

KRAJEWSKI, L. J.; MALHOTRA, M. K.; RITZMAN, L. P. **Operations Management. Processes and Supply Chains**. 11<sup>th</sup> ed. Pearson, pp. 23, 2016.

KRAUS, M.; FEUERRIEGEL, S. Decision support from financial disclosures with deep neural networks and transfer learning. **Decision Support Systems**, v. 104, pp. 38-48, 2017.

KRIESEL, D. **A Brief Introduction to Neural Networks**. 2005. Disponível em:

<http://creativecommons.org/licenses/by-nd/3.0/>>. Acesso em 12 abr. 2019.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. **Communications of the ACM**. Vol. 60 No. 6, 2012.

KRUGER, J.; DUNNING, D. Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments. **Journal of Personality and Social Psychology**. v. 77, n. 6, pp. 1121-1134. 1999.

KURZWEIL, R. **The Age of Intelligent Machines**. MIT Press. 1990.

LAI, T. L.; GAO, P. Transaction costs and stochastic control in dynamic portfolio management. Working paper, Department of Statistics, **Stanford University**, 2016.

LÄNGKVIST, M.; KARLSSON, L.; LOUTFI, A. A review of unsupervised feature learning and deep learning for time-series modeling. **Pattern Recognition Letters**. v. 42, pp. 11-24. 2014.

LECUN, Y.; BENGIO, Y. Convolutional Networks for Images, Speech, and Time-Series. **The handbook of brain theory and neural networks**. MIT Press, 1995.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep Learning. **Nature**. Vol. 521, pp.436-444, 2015.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition **Proceedings of the IEEE**. v. 86, n. 11, 1998.

LIAO, Z.; WANG, J. Forecasting model of global stock index by stochastic time effective neural network. **Expert Systems with Applications**. 2010.

LITJENS, G. et al. A Survey on Deep Learning in Medical Image Analysis. **Medical Image Analysis**. 2017

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully Convolutional Networks for Semantic Segmentation. **IEEE Pattern Analysis Machine Intelligence**. 2015.

LIU, O. Python Tutorial on Linear Regression with Batch Gradient Descent. Publ. 09-feb-2016. Disponível em: <<http://ozzieliu.com/2016/02/09/gradient-descent-tutorial>>. Acesso em 14-out-2018.

MARCH, S. T.; SMITH, G. Design and Natural Science Research on Information Technology, **Decision Support Systems**. December 1995, pp. 251-266. 1995.

MARKOV, A. A. Extension of the Limit Theorems of Probability Theory to a Sum of Variables Connected in a Chain. **Probabilistic Systems**, v.1. John Wiley and Sons, 1971.

MENKVELD, A. J. High Frequency Trading and the New Market Makers. **Journal of Financial Markets**. v. 16, 2013

MIDDLETON, P. W.; THEOFILATOS K.; KARATHANASOPOULOS A. Trading the FTSE100 Index: 'Adaptive' Modelling and Optimization Techniques. Cap. 2 in DUNIS, C. L.; MIDDLETON, P. W.; KARATHANASOPOULOS, A.; THEOFILATOS, K. **Artificial Intelligence in Financial Markets**. Palgrave MacMillan, 2016.

MIT TECHNOLOGY REVIEW. As Goldman Embraces Automation, Even the Masters of the Universe Are Threatened. Nanette Byrnes. 7 fev. 2017. Disponível em: <[www.technologyreview.com/s/603431/as-goldman-embraces-automation-even-the-masters-of-the-universe-are-threatened/](http://www.technologyreview.com/s/603431/as-goldman-embraces-automation-even-the-masters-of-the-universe-are-threatened/)>. Acesso em 7 jun. 2018.

MITCHELL, T. **Machine Learning**. McGraw-Hill, 1997.

MURPHY, K. P. **Machine Learning – A Probabilistic Perspective**. London: MIT Press, 2012.

MURPHY, J. **Technical Analysis of the Financial Markets**. New York: Penguin Putnam, 1999.

NATENBERG, S. **Option Volatility & Pricing. Advanced Trading Strategies and Techniques**. McGraw-Hill, 1994.

NGIAM, J. et al. Multimodal deep learning Machine Learning, **28th International Conference on Machine Learning**. 2011.

NUTI, G.; MIRGHAEMI, M.; TRELEAVEN, P.; YINGSAEREE, C. Algorithmic trading. **Computer-IEEE Computer Society**, v. 44, n. 11, p. 61-69, 2011.

NVIDIA Disponível em: [https://www.blogs.nvidia.com/wp-content/uploads/2016/07/Deep\\_Learning\\_Icons\\_R5.PNG.png](https://www.blogs.nvidia.com/wp-content/uploads/2016/07/Deep_Learning_Icons_R5.PNG.png). Acesso em 12-Jul-2019.

OLAH, C. Understanding LSTM Networks, [online] Disponível em: <<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>>. Acesso em 14 nov. 2019.

O'HARA, M. High frequency market microstructure. **Journal of Financial Economics**. v. 116, 2015.

OKUBO, Y. **Bibliometric Indicators and Analysis of Research Systems: Methods and Examples**. Paris: OECD Publishing, 1997.

PALAI, U. Algorithmic Trading. Disponível em: <<https://github.com/umeshpalai/AlgorithmicTrading-MachineLearning>>. Acesso em 14 nov. 2019.

PARÉ, G.; TRUDEL, M.-C.; JAANA, M.; KTSIOU, S. Synthesizing information systems knowledge: A typology of literature reviews. **Information and Management**. v. 52, n. 2, p 183-199, 2014.

PATEL, J.; SHAH, S.; THAKKAR, P.; KOTTECHA, K. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. **Expert Systems with Applications**. 2014.

PATEL, J.; SHAH, S.; THAKKAR, P.; KOTTECHA, K. Predicting stock market index using fusion of machine learning techniques. **Expert Systems with Applications**. 2015.

PEFFERS, K.; TUUNANEN, T.; ROTHENBERGER, M. A.; CHATTERJEE, S. A Design Science Research Methodology for Information Systems Research. **Journal of Management Information Systems**, v. 24:3, pp. 45-77, DOI: 10.2753/MIS0742-1222240302, 2007.

PEARSON, K. The Problem of the Random Walk. **Nature**. v.72, p.294, 1905.

PEDREGOSA, F. *et al.* Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**. v. 12., pp. 2825-2830, 2011.

PERLIN, M. S.; RAMOS, H. P. GetHFDData: A R package for downloading and aggregating high frequency trading data from Bovespa. **Revista Brasileira de Finanças** (Online), Rio de Janeiro, 14, No. 3, pp. 443-478, Jul. 2016.

PHILLIPS, S. J.; ANDERSON, R. P.; SCHAPIRE, R. E. Maximum entropy modeling of species geographic distributions. **Ecological Modelling**. v. 190, 2005.

PRADO, M. L. **Advances in Financial Machine Learning**. John Wiley & Sons, 2018.

RAZAVIAN, A. S.; AZIZPOUR, H.; SULLIVAN, J.; CARLSSON, S. CNN features off-the-shelf: An Astounding Baseline for Recognition. **IEEE Conference on Computer Vision and Pattern Recognition**. 2014.

REVERRE, S. **The Complete Arbitrage Deskbook**. McGraw-Hill, 2001.

ROHRER, B. Disponível em: <<https://brohrer.github.io/blog.html>>. Acesso em 14 nov. 2019.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**. Vol. 65, No. 6, 1958.

ROSS, S. A.; WESTERFIELD, R. W.; JAFFE, J. F. **Administração Financeira – Corporate Finance**. São Paulo: ATLAS, 2011.

ROSS, S. A.; WESTERFIELD, R. W.; JORDAN, B. D.; LAMB, R. **Fundamentos da Administração Financeira – 9ª ed.** Porto Alegre: AMGH Editora, 2013.

RUDER, S. An Overview of Gradient Descent Optimization Algorithms. **arXiv:1609.04747v2 [cs.LG]**. 2017.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning Representations by Back-Propagating Errors. **Nature**, 1986.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence - A Modern Approach**. 3rd. ed. Pearson, 2016

SAMUEL, A. L. Some Studies in Machine Learning Using the Game of Checkers. **IBM Journal of Research and Development**. Vol. 3, p. 210-229. 1959.

SCHMIDHUBER, J. Deep learning in neural networks. **Neural Networks**. Vol. 61, p.85-117, 2014.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. FaceNet: A unified embedding for face recognition and clustering. **IEEE Computer Vision and Pattern Recognition**. 2015.

SCHUMAKER, R.P.; CHEN, H. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. **ACM Information Systems**. 2009a.

SCHUMAKER, R.P.; CHEN, H. A quantitative stock prediction system based on financial news. **Information Processing and Management**. 2009b.

SCHUSTER, M. et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. **Cornell University-ArXiv**. 2016.

SCOPUS – ELSEVIER Disponível em: <[www.scopus.com](http://www.scopus.com)>. Acesso em 30 set. 2019.

SELVIN, S.; VINAYAKUMAR, R.; GOPALAKRISHNAM, E. A.; MENON, V. K.; SOMAN, K. P. Stock Price Prediction using LSTM, RNN, and CNN-Sliding Window Model. **International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017**.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding Machine Learning. From Theory to Algorithms**. New York: Cambridge University Press, 2014.

SHARPE, W.; ALEXANDER, G.; BAILEY, J. **Investments**. 6th ed. New Jersey: Prentice Hall, 1999.

SHEN, G.; TAN, Q.; ZHANG, H.; ZENG, P.; XU, J. Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions. **8<sup>th</sup> International Congress of Information and Communication Technology (ICICT-2018). Procedia Computer Science**. v. 131. 2018.

SHILLER, R. J. Rhetoric and Economic Behavior. Conversation, Information, and Herd Behavior. **The American Economic Review**. v. 85, n. 2, pp. 181-185. 1995.

SILVER, D. *et al.*; Mastering the Game of Go without Human Knowledge, **Nature**, Vol 550, 19 October 2017 pp. 354-360; Macmillan Publishers Ltd, Springer Nature – 2017

SCIKIT-LEARN Disponível em: <<http://scikit-learn.org/>>. Acesso em 13 jan 2020.

SIMON, H. A. **The Sciences of the Artificial**. 3 ed., Cambridge: MIT Press, 1996.

SINGH, R.; SRIVASTAVA, S. Stock prediction using deep learning. **Multimedia Tools and Applications**. 2016.

SINGH, D. Neural Network in Python: Introduction, Structure and Trading Strategies. **QuantInsti**. 2019.

SOHANGIR, S.; WANG, D.; POMERANETS, A.; KHOSHGOFTAAR, T. M. Big Data: Deep Learning for financial sentiment analysis. **Journal of Big Data**, 2018.

SONG, Y.; GUO, Q. Query-less: Predicting task repetition for nextgen proactive search and recommendation engines. **Proceedings of the 25th International Conference on World Wide Web**. 2016.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. **Journal of Machine Learning Research**. v. 15, pp. 1929-1958. 2014

STEVENSON, W. J. **Operations Management**. 12<sup>th</sup> ed. McGraw-Hill, pp.4-6, 2015.

SURYANSH, S. Gradient Descent: All You Need to Know. Publ. Mar 2012. Disponível em: <<http://hackernoon.com/gradient-descent-aynk-7cbe95a778da>>. Acesso em 14-out-2018.

TALEB, N. **Dynamic Hedging. Managing Vanilla and Exotic Options**. New York: John Wiley & Sons, 1997.

TAKEUCHI, L.; LEE, Y-Y. Applying Deep Learning to Enhance Momentum Trading Strategies in Stocks. **Stanford.edu**, 2013.

THALLER, R. H. Mental Accounting Matters. **Journal of Behavioral Decision Making**. v. 12, pp. 183-206. 1999

THAWORNWONG, S.; ENKE, D. The adaptive selection of financial and economic variables for use with artificial neural networks. **Neurocomputing**. 2004.

TRELEAVEN, P.; GALAS, M.; LALCHAND, V. Algorithmic Trading Review. **Communications of the ACM**. v. 56, n. 11, 2013.

TSANTEKIDIS, A. et al. Forecasting stock prices from the limit order book using convolutional neural networks. **2017 IEEE 19th Business Informatics**, 2017.

TSICHRITZIS, D. **The Dynamics of Innovation, in Beyond Calculation: The Next Fifty Years of Computing**. Denning, P. J.; METCALFE, R. M. (eds.), pp. 259-265, New York: Copernicus Books, 1998,

TURING, A. M. Computing Machinery and Intelligence. **Mind**. Vol 49. p. 433-460. 1950.

TVERSKY, A.; KAHNEMAN, D. Judgment under Uncertainty: Heuristics and Biases. **Science**. v. 184, n. 4157, pp. 1124-1131, 1974.

VARGAS, M. R.; LIMA, B. S. L. P.; EVSUKOFF, A. G. Deep learning for stock market prediction from financial news articles. **2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications, CIVEMSA 2017**, p. 60-65, 2017.

VEDALDI, A.; LENC, K. MatConvNet: Convolutional neural networks for MATLAB. **ACM 2015 Multimedia Conference**. 2015.

VIDYAMURTHY, G. **Pairs Trading. Quantitative Methods and Analysis**. New Jersey: Wiley, 2004.

VINCENT, P.; LAROCHELLE, H.; LAJOIE, I.; BENGIO, Y.; MANZAGOL, P-A. Stacked denoising

autoencoders: Learning Useful Representations in a Deep Network. **Journal of Machine Learning Research**. 2010.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. **IEEE Computer Vision and Pattern Recognition**. 2001

WIERSTRA, D.; SCHMIDHUBER, J.; GOMEZ, F. J. Evolino: Hybrid Neuroevolution/Optimal Linear Search for Sequence Learning. **Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)**, Edinburgh: pp. 853–858, 2005.

WILMOTT, P. **Derivatives. The Theory and Practice of Financial Engineering**. Chichester: Wiley, 1998.

WITTEN, I. H.; FRANK, E.; HALL, M. A.; PAL, C. J. **Data Mining: Practical Machine Learning Tools and Techniques**. Waikato Environment for Knowledge Analysis. 4 ed. Cambridge: Morgan Kaufmann, 2016.

WEB OF SCIENCE – CLARIVATE ANALYTICS Disponível em: < [www.webofknowledge.com](http://www.webofknowledge.com)>. Acesso em 30 set. 2019.

XIONG, R.; NICHOLS, E. P.; SHEN, Y. Deep Learning Stock Volatility with Google Domestic Trends. **Stanford.edu arXiv:1512.04916v3 [q-fin.CP]**. 2016.

YOSHIHARA, A.; FUJIKAWA, K.; SEKI, K.; UEHARA, K. Predicting stock market trends by recurrent deep neural networks. **Computer Science**. 2014

ZHANG, G.; PATUWO, B. E.; HU, M. Y. Forecasting with Artificial Neural Networks: The State of The Art, **International Journal of Forecasting**, 14, 35–62, 1998.

ZHAO, J.; WANG, Z.; PARK, D. S. Online sequential extreme learning machine with forgetting mechanism. **Neurocomputing**. 2012

## APÊNDICE – Logs de execução

Nas listagens abaixo são apresentados os passos da execução do treinamento da implementação da abordagem LSTM, constando as iterações, as frações das épocas e os erros quadráticos médios de treinamento e validação.

### PETR4:

```
Outputs de treinamento: 2.159 preços de PETR4
Iter 0 Epoca 0.0 MSE: Treino 0.072873 Valid 0.017208
Iter 43 Epoca 1.0 MSE: Treino 0.000937 Valid 0.000337
Iter 86 Epoca 2.0 MSE: Treino 0.000726 Valid 0.000238
Iter 129 Epoca 3.0 MSE: Treino 0.000703 Valid 0.000242
Iter 172 Epoca 4.0 MSE: Treino 0.000659 Valid 0.000246
Iter 215 Epoca 5.0 MSE: Treino 0.000696 Valid 0.000235
Iter 258 Epoca 6.0 MSE: Treino 0.000558 Valid 0.000193
Iter 301 Epoca 7.0 MSE: Treino 0.000492 Valid 0.000178
Iter 344 Epoca 8.0 MSE: Treino 0.000522 Valid 0.000193
Iter 387 Epoca 9.0 MSE: Treino 0.000456 Valid 0.000174
Iter 430 Epoca 10.0 MSE: Treino 0.000416 Valid 0.000149
Iter 473 Epoca 11.0 MSE: Treino 0.000397 Valid 0.000141
Iter 516 Epoca 11.9 MSE: Treino 0.000539 Valid 0.000164
Iter 559 Epoca 12.9 MSE: Treino 0.000342 Valid 0.000133
Iter 602 Epoca 13.9 MSE: Treino 0.000341 Valid 0.000132
Iter 645 Epoca 14.9 MSE: Treino 0.000323 Valid 0.000120
Iter 688 Epoca 15.9 MSE: Treino 0.000331 Valid 0.000116
Iter 731 Epoca 16.9 MSE: Treino 0.000343 Valid 0.000112
Iter 774 Epoca 17.9 MSE: Treino 0.000309 Valid 0.000113
Iter 817 Epoca 18.9 MSE: Treino 0.000268 Valid 0.000107
Iter 860 Epoca 19.9 MSE: Treino 0.000255 Valid 0.000106
Iter 903 Epoca 20.9 MSE: Treino 0.000245 Valid 0.000101
Iter 946 Epoca 21.9 MSE: Treino 0.000304 Valid 0.000114
Iter 989 Epoca 22.9 MSE: Treino 0.000228 Valid 0.000093
Iter 1032 Epoca 23.9 MSE: Treino 0.000219 Valid 0.000093
Iter 1075 Epoca 24.9 MSE: Treino 0.000246 Valid 0.000092
Iter 1118 Epoca 25.9 MSE: Treino 0.000233 Valid 0.000094
Iter 1161 Epoca 26.9 MSE: Treino 0.000192 Valid 0.000083
Iter 1204 Epoca 27.9 MSE: Treino 0.000209 Valid 0.000093
Iter 1247 Epoca 28.9 MSE: Treino 0.000240 Valid 0.000102
Iter 1290 Epoca 29.9 MSE: Treino 0.000181 Valid 0.000077
Iter 1333 Epoca 30.9 MSE: Treino 0.000190 Valid 0.000080
Iter 1376 Epoca 31.9 MSE: Treino 0.000243 Valid 0.000099
Iter 1419 Epoca 32.9 MSE: Treino 0.000184 Valid 0.000075
Iter 1462 Epoca 33.9 MSE: Treino 0.000163 Valid 0.000070
Iter 1505 Epoca 34.9 MSE: Treino 0.000178 Valid 0.000073
Iter 1548 Epoca 35.8 MSE: Treino 0.000146 Valid 0.000065
```

```
Iter 1591 Epoca 36.8 MSE: Treino 0.000163 Valid 0.000071
Iter 1634 Epoca 37.8 MSE: Treino 0.000166 Valid 0.000067
Iter 1677 Epoca 38.8 MSE: Treino 0.000227 Valid 0.000094
Iter 1720 Epoca 39.8 MSE: Treino 0.000158 Valid 0.000077
Iter 1763 Epoca 40.8 MSE: Treino 0.000140 Valid 0.000063
Iter 1806 Epoca 41.8 MSE: Treino 0.000142 Valid 0.000062
Iter 1849 Epoca 42.8 MSE: Treino 0.000155 Valid 0.000063
Iter 1892 Epoca 43.8 MSE: Treino 0.000183 Valid 0.000073
Iter 1935 Epoca 44.8 MSE: Treino 0.000135 Valid 0.000059
Iter 1978 Epoca 45.8 MSE: Treino 0.000140 Valid 0.000062
Iter 2021 Epoca 46.8 MSE: Treino 0.000142 Valid 0.000057
Iter 2064 Epoca 47.8 MSE: Treino 0.000125 Valid 0.000052
Iter 2107 Epoca 48.8 MSE: Treino 0.000128 Valid 0.000058
Iter 2150 Epoca 49.8 MSE: Treino 0.000122 Valid 0.000054
Iter 2193 Epoca 50.8 MSE: Treino 0.000119 Valid 0.000053
Iter 2236 Epoca 51.8 MSE: Treino 0.000130 Valid 0.000053
Iter 2279 Epoca 52.8 MSE: Treino 0.000125 Valid 0.000058
Iter 2322 Epoca 53.8 MSE: Treino 0.000136 Valid 0.000057
Iter 2365 Epoca 54.8 MSE: Treino 0.000119 Valid 0.000054
Iter 2408 Epoca 55.8 MSE: Treino 0.000117 Valid 0.000051
Iter 2451 Epoca 56.8 MSE: Treino 0.000116 Valid 0.000050
Iter 2494 Epoca 57.8 MSE: Treino 0.000131 Valid 0.000053
Iter 2537 Epoca 58.8 MSE: Treino 0.000118 Valid 0.000051
Iter 2580 Epoca 59.7 MSE: Treino 0.000131 Valid 0.000052
Iter 2623 Epoca 60.7 MSE: Treino 0.000158 Valid 0.000076
Iter 2666 Epoca 61.7 MSE: Treino 0.000147 Valid 0.000063
Iter 2709 Epoca 62.7 MSE: Treino 0.000130 Valid 0.000058
Iter 2752 Epoca 63.7 MSE: Treino 0.000135 Valid 0.000055
Iter 2795 Epoca 64.7 MSE: Treino 0.000117 Valid 0.000052
Iter 2838 Epoca 65.7 MSE: Treino 0.000152 Valid 0.000073
Iter 2881 Epoca 66.7 MSE: Treino 0.000155 Valid 0.000070
Iter 2924 Epoca 67.7 MSE: Treino 0.000119 Valid 0.000055
Iter 2967 Epoca 68.7 MSE: Treino 0.000107 Valid 0.000047
Iter 3010 Epoca 69.7 MSE: Treino 0.000152 Valid 0.000068
Iter 3053 Epoca 70.7 MSE: Treino 0.000112 Valid 0.000049
Iter 3096 Epoca 71.7 MSE: Treino 0.000121 Valid 0.000052
Iter 3139 Epoca 72.7 MSE: Treino 0.000114 Valid 0.000057
Iter 3182 Epoca 73.7 MSE: Treino 0.000115 Valid 0.000050
Iter 3225 Epoca 74.7 MSE: Treino 0.000121 Valid 0.000057
Iter 3268 Epoca 75.7 MSE: Treino 0.000112 Valid 0.000049
Iter 3311 Epoca 76.7 MSE: Treino 0.000126 Valid 0.000054
Iter 3354 Epoca 77.7 MSE: Treino 0.000181 Valid 0.000116
Iter 3397 Epoca 78.7 MSE: Treino 0.000109 Valid 0.000048
Iter 3440 Epoca 79.7 MSE: Treino 0.000106 Valid 0.000046
Iter 3483 Epoca 80.7 MSE: Treino 0.000171 Valid 0.000081
Iter 3526 Epoca 81.7 MSE: Treino 0.000124 Valid 0.000053
Iter 3569 Epoca 82.7 MSE: Treino 0.000150 Valid 0.000057
Iter 3612 Epoca 83.6 MSE: Treino 0.000111 Valid 0.000056
Iter 3655 Epoca 84.6 MSE: Treino 0.000114 Valid 0.000049
```

```
Iter 3698 Epoca 85.6 MSE: Treino 0.000107 Valid 0.000046
Iter 3741 Epoca 86.6 MSE: Treino 0.000107 Valid 0.000049
Iter 3784 Epoca 87.6 MSE: Treino 0.000108 Valid 0.000045
Iter 3827 Epoca 88.6 MSE: Treino 0.000103 Valid 0.000045
Iter 3870 Epoca 89.6 MSE: Treino 0.000149 Valid 0.000063
Iter 3913 Epoca 90.6 MSE: Treino 0.000106 Valid 0.000045
Iter 3956 Epoca 91.6 MSE: Treino 0.000106 Valid 0.000052
Iter 3999 Epoca 92.6 MSE: Treino 0.000121 Valid 0.000054
Iter 4042 Epoca 93.6 MSE: Treino 0.000137 Valid 0.000057
Iter 4085 Epoca 94.6 MSE: Treino 0.000133 Valid 0.000052
Iter 4128 Epoca 95.6 MSE: Treino 0.000121 Valid 0.000048
Iter 4171 Epoca 96.6 MSE: Treino 0.000104 Valid 0.000047
Iter 4214 Epoca 97.6 MSE: Treino 0.000140 Valid 0.000056
Iter 4257 Epoca 98.6 MSE: Treino 0.000109 Valid 0.000053
Iter 4300 Epoca 99.6 MSE: Treino 0.000119 Valid 0.000049
```

### VALE3:

Outputs de treinamento: 2.159 preços de VALE3)

```
Iter 0 Epoca 0.0 MSE: Treino 0.108656 Valid 0.049541
Iter 43 Epoca 1.0 MSE: Treino 0.001113 Valid 0.000811
Iter 86 Epoca 2.0 MSE: Treino 0.000881 Valid 0.000736
Iter 129 Epoca 3.0 MSE: Treino 0.000917 Valid 0.000661
Iter 172 Epoca 4.0 MSE: Treino 0.000930 Valid 0.000634
Iter 215 Epoca 5.0 MSE: Treino 0.000854 Valid 0.000610
Iter 258 Epoca 6.0 MSE: Treino 0.000703 Valid 0.000592
Iter 301 Epoca 7.0 MSE: Treino 0.000663 Valid 0.000574
Iter 344 Epoca 8.0 MSE: Treino 0.000577 Valid 0.000483
Iter 387 Epoca 9.0 MSE: Treino 0.000758 Valid 0.000533
Iter 430 Epoca 10.0 MSE: Treino 0.000567 Valid 0.000518
Iter 473 Epoca 11.0 MSE: Treino 0.000637 Valid 0.000461
Iter 516 Epoca 11.9 MSE: Treino 0.001135 Valid 0.000741
Iter 559 Epoca 12.9 MSE: Treino 0.000516 Valid 0.000409
Iter 602 Epoca 13.9 MSE: Treino 0.000426 Valid 0.000359
Iter 645 Epoca 14.9 MSE: Treino 0.000470 Valid 0.000409
Iter 688 Epoca 15.9 MSE: Treino 0.000410 Valid 0.000335
Iter 731 Epoca 16.9 MSE: Treino 0.000484 Valid 0.000358
Iter 774 Epoca 17.9 MSE: Treino 0.000394 Valid 0.000336
Iter 817 Epoca 18.9 MSE: Treino 0.000383 Valid 0.000301
Iter 860 Epoca 19.9 MSE: Treino 0.000372 Valid 0.000330
Iter 903 Epoca 20.9 MSE: Treino 0.000317 Valid 0.000304
Iter 946 Epoca 21.9 MSE: Treino 0.000420 Valid 0.000291
Iter 989 Epoca 22.9 MSE: Treino 0.000302 Valid 0.000259
Iter 1032 Epoca 23.9 MSE: Treino 0.000392 Valid 0.000270
Iter 1075 Epoca 24.9 MSE: Treino 0.000269 Valid 0.000229
Iter 1118 Epoca 25.9 MSE: Treino 0.000246 Valid 0.000224
Iter 1161 Epoca 26.9 MSE: Treino 0.000372 Valid 0.000232
```

```
Iter 1204 Epoca 27.9 MSE: Treino 0.000219 Valid 0.000190
Iter 1247 Epoca 28.9 MSE: Treino 0.000207 Valid 0.000186
Iter 1290 Epoca 29.9 MSE: Treino 0.000208 Valid 0.000172
Iter 1333 Epoca 30.9 MSE: Treino 0.000201 Valid 0.000178
Iter 1376 Epoca 31.9 MSE: Treino 0.000221 Valid 0.000160
Iter 1419 Epoca 32.9 MSE: Treino 0.000211 Valid 0.000159
Iter 1462 Epoca 33.9 MSE: Treino 0.000179 Valid 0.000147
Iter 1505 Epoca 34.9 MSE: Treino 0.000215 Valid 0.000158
Iter 1548 Epoca 35.8 MSE: Treino 0.000170 Valid 0.000139
Iter 1591 Epoca 36.8 MSE: Treino 0.000215 Valid 0.000148
Iter 1634 Epoca 37.8 MSE: Treino 0.000169 Valid 0.000131
Iter 1677 Epoca 38.8 MSE: Treino 0.000163 Valid 0.000129
Iter 1720 Epoca 39.8 MSE: Treino 0.000165 Valid 0.000126
Iter 1763 Epoca 40.8 MSE: Treino 0.000158 Valid 0.000127
Iter 1806 Epoca 41.8 MSE: Treino 0.000165 Valid 0.000122
Iter 1849 Epoca 42.8 MSE: Treino 0.000160 Valid 0.000123
Iter 1892 Epoca 43.8 MSE: Treino 0.000379 Valid 0.000245
Iter 1935 Epoca 44.8 MSE: Treino 0.000167 Valid 0.000139
Iter 1978 Epoca 45.8 MSE: Treino 0.000167 Valid 0.000125
Iter 2021 Epoca 46.8 MSE: Treino 0.000177 Valid 0.000124
Iter 2064 Epoca 47.8 MSE: Treino 0.000150 Valid 0.000117
Iter 2107 Epoca 48.8 MSE: Treino 0.000191 Valid 0.000137
Iter 2150 Epoca 49.8 MSE: Treino 0.000155 Valid 0.000123
Iter 2193 Epoca 50.8 MSE: Treino 0.000150 Valid 0.000116
Iter 2236 Epoca 51.8 MSE: Treino 0.000179 Valid 0.000153
Iter 2279 Epoca 52.8 MSE: Treino 0.000164 Valid 0.000127
Iter 2322 Epoca 53.8 MSE: Treino 0.000163 Valid 0.000115
Iter 2365 Epoca 54.8 MSE: Treino 0.000165 Valid 0.000124
Iter 2408 Epoca 55.8 MSE: Treino 0.000149 Valid 0.000113
Iter 2451 Epoca 56.8 MSE: Treino 0.000240 Valid 0.000181
Iter 2494 Epoca 57.8 MSE: Treino 0.000230 Valid 0.000146
Iter 2537 Epoca 58.8 MSE: Treino 0.000163 Valid 0.000113
Iter 2580 Epoca 59.7 MSE: Treino 0.000157 Valid 0.000115
Iter 2623 Epoca 60.7 MSE: Treino 0.000153 Valid 0.000110
Iter 2666 Epoca 61.7 MSE: Treino 0.000168 Valid 0.000115
Iter 2709 Epoca 62.7 MSE: Treino 0.000151 Valid 0.000111
Iter 2752 Epoca 63.7 MSE: Treino 0.000143 Valid 0.000105
Iter 2795 Epoca 64.7 MSE: Treino 0.000175 Valid 0.000141
Iter 2838 Epoca 65.7 MSE: Treino 0.000233 Valid 0.000203
Iter 2881 Epoca 66.7 MSE: Treino 0.000163 Valid 0.000111
Iter 2924 Epoca 67.7 MSE: Treino 0.000160 Valid 0.000110
Iter 2967 Epoca 68.7 MSE: Treino 0.000147 Valid 0.000121
Iter 3010 Epoca 69.7 MSE: Treino 0.000144 Valid 0.000106
Iter 3053 Epoca 70.7 MSE: Treino 0.000186 Valid 0.000123
Iter 3096 Epoca 71.7 MSE: Treino 0.000175 Valid 0.000116
Iter 3139 Epoca 72.7 MSE: Treino 0.000140 Valid 0.000112
Iter 3182 Epoca 73.7 MSE: Treino 0.000141 Valid 0.000109
Iter 3225 Epoca 74.7 MSE: Treino 0.000244 Valid 0.000135
Iter 3268 Epoca 75.7 MSE: Treino 0.000142 Valid 0.000116
```

```
Iter 3311 Epoca 76.7 MSE: Treino 0.000140 Valid 0.000103
Iter 3354 Epoca 77.7 MSE: Treino 0.000138 Valid 0.000103
Iter 3397 Epoca 78.7 MSE: Treino 0.000209 Valid 0.000164
Iter 3440 Epoca 79.7 MSE: Treino 0.000144 Valid 0.000106
Iter 3483 Epoca 80.7 MSE: Treino 0.000142 Valid 0.000111
Iter 3526 Epoca 81.7 MSE: Treino 0.000162 Valid 0.000140
Iter 3569 Epoca 82.7 MSE: Treino 0.000149 Valid 0.000128
Iter 3612 Epoca 83.6 MSE: Treino 0.000234 Valid 0.000158
Iter 3655 Epoca 84.6 MSE: Treino 0.000140 Valid 0.000106
Iter 3698 Epoca 85.6 MSE: Treino 0.000138 Valid 0.000103
Iter 3741 Epoca 86.6 MSE: Treino 0.000218 Valid 0.000126
Iter 3784 Epoca 87.6 MSE: Treino 0.000141 Valid 0.000107
Iter 3827 Epoca 88.6 MSE: Treino 0.000150 Valid 0.000108
Iter 3870 Epoca 89.6 MSE: Treino 0.000168 Valid 0.000130
Iter 3913 Epoca 90.6 MSE: Treino 0.000141 Valid 0.000106
Iter 3956 Epoca 91.6 MSE: Treino 0.000144 Valid 0.000122
Iter 3999 Epoca 92.6 MSE: Treino 0.000158 Valid 0.000110
Iter 4042 Epoca 93.6 MSE: Treino 0.000166 Valid 0.000119
Iter 4085 Epoca 94.6 MSE: Treino 0.000143 Valid 0.000122
Iter 4128 Epoca 95.6 MSE: Treino 0.000190 Valid 0.000117
Iter 4171 Epoca 96.6 MSE: Treino 0.000141 Valid 0.000108
Iter 4214 Epoca 97.6 MSE: Treino 0.000145 Valid 0.000107
Iter 4257 Epoca 98.6 MSE: Treino 0.000144 Valid 0.000107
Iter 4300 Epoca 99.6 MSE: Treino 0.000136 Valid 0.000106
```

## ANEXO – Certificado de Registro de Programa de Computador - INPI



REPÚBLICA FEDERATIVA DO BRASIL  
MINISTÉRIO DA ECONOMIA

INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL  
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS INTEGRADOS

## Certificado de Registro de Programa de Computador

Processo Nº: **BR512020001209-4**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de 22/06/2019, em conformidade com o §2º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

**Título:** LSTM applied to Financial Time Series

**Data de publicação:** 22/06/2019

**Data de criação:** 01/06/2019

**Titular(es):** NAPOLEÃO VERARDI GALEGAL; CAMILO ILZO SHIMABUKURO

**Autor(es):** NAPOLEÃO VERARDI GALEGAL; CAMILO ILZO SHIMABUKURO

**Linguagem:** PYTHON

**Campo de aplicação:** FN-03; IF-07; MT-04; MT-06

**Tipo de programa:** AP-01; IA-01; SM-01

**Algoritmo hash:** SHA-512

**Resumo digital hash:**

fb4f92bb8fe51db80ff725d35041af931cc194a8404fbb5703f72e45fe7ffddccc38c54b627be20f992d9cbb073bd941ecab6ec74404409b201764ce49e8a66

**Expedido em:** 07/07/2020

**Aprovado por:**

Helmar Alvares

Chefe da DIPTO - Portaria/INPI/DIRPA Nº 09, de 01 de julho de 2019