

# **Aplicação de Redes Neurais e Fatores de Prêmio de Risco no Mercado de Ações**

**Vitor Grando Eller**

**Orientador: Raul Ikeda Gomes da Silva**

**Inspira, 2020**

## Resumo

Esse estudo busca entender o impacto da combinação de redes neurais e fatores de prêmio de risco na previsibilidade do índice S&P500, e também de compreender a diferença de performance de topologias distintas dentro desse processo. Além disso, elucida as vantagens de se utilizar uma estratégia de retreinamento quando da criação de modelos que lidam com séries temporais, e, principalmente, séries de preços de ativos, mostrando a melhor performance obtida por modelos que utilizam desta técnica. Por fim, discute a viabilidade do uso do modelo de barreira tripla proposto por Lopez de Prado (2018) em seu livro, e a eficácia de um modelo classificador em comparação ao modelo regressor.

**Palavras-chave:** Redes Neurais, Fator de Prêmio de Risco, LSTM. Precificação de Ativos.

## Sumário

1.	INTRODUÇÃO .....	4
2.	METODOLOGIA .....	6
2.1.	ESTRUTURAS DE DADOS .....	6
2.1.1.	AÇÕES .....	6
2.1.2.	FATORES DE PRÊMIO .....	7
2.1.3.	A BARREIRA TRIPLA.....	7
2.1.4.	DADOS DE ENTRADA E VARIÁVEIS PREVISTAS .....	9
2.2.	PREPARAÇÃO DOS DADOS .....	11
2.3.	ESTRUTURA DAS REDES.....	11
2.3.1.	MODELO COM LONG SHORT TERM MEMORY (LSTM) .....	12
2.3.2.	MODELO SEM LONG SHORT TERM MEMORY (LSTM).....	13
2.4.	OS MÉTODOS DE ANÁLISE.....	13
3.	RESULTADOS OBTIDOS.....	15
3.1.	MODELO REGRESSOR .....	15
3.2.	MODELO CLASSIFICADOR (BARREIRA TRIPLA) .....	19
4.	CONCLUSÃO .....	21
5.	BIBLIOGRAFIA.....	22

## 1. Introdução

Apesar da teoria proposta por Kendall defender que os movimentos dos preços do mercado são imprevisíveis (Kendall & Hill, 1953), existe uma mobilização histórica entre os estudiosos para construir modelos que sejam capazes de capturar possíveis tendências nos retornos. O dispêndio de tempo em busca de tal modelo é explicado pela possibilidade de se obter uma maior previsibilidade durante o investimento, visando uma possível maximização de retorno com um menor risco.

Vale citar o Modelo de Precificação de Ativos (*Capital Asset Pricing Model* ou CAPM) (Bodie, et al., 2013), modelo desenvolvido no passado e base de estudos teóricos até hoje, que sugere que o retorno em excesso de uma carteira está diretamente relacionado ao beta da mesma. A partir deste, Fama e French propuseram o Modelo de 3 Fatores (1992), onde defendiam que, além do beta de mercado, o retorno em excesso seria explicado também por outros dois fatores relacionados à valores fundamentais das companhias.

Sabe-se que ambos são modelos lineares<sup>1</sup>, logo podem não ser capazes de capturar todas as sinergias existentes entre as variáveis de entrada e saída do sistema. Essa situação foi contornada durante o final do século XX, aliando o desenvolvimento de novas técnicas de Redes Neurais com os modelos de precificação de ativos (Fadlalla & Lin, 2001).

As Redes Neurais são sistemas computacionais que possuem como objetivo emular o processo de pensamento humano. Diferente dos modelos lineares, elas possuem a capacidade de examinar os dados após uma abstração de seus ruídos e distorções, sendo assim capaz de extrair relações não-lineares entre as variáveis preditivas (Haykin, 1994).

Desta forma, inúmeras pesquisas foram feitas para compreender a usabilidade desta tecnologia no mercado financeiro. Wanjawa e Muchemi (2014) propuseram um modelo baseado em Redes Neurais Artificiais (*Artificial Neural Networks* ou ANN) para prever o preço de ações nas bolsas, Pantazopoulos (1998) utilizou Sistemas Neuro-Fuzzy para fazer estimativas de dados financeiros e criar estratégias de trading, e Batres-Estrada (2015) avaliou a usabilidade de redes com Multicamadas de Perceptrons (*Multi Layer Perceptron* ou MLP) em séries temporais financeiras de múltiplas variáveis.

Houve uma evolução no âmbito computacional que acarretou o desenvolvimento de técnicas capazes de analisar uma quantia maior de dados e com múltiplos níveis de abstração, sem a necessidade de supervisão humana. O Deep Learning trouxe inovações que

---

<sup>1</sup> Modelos Lineares de regressão são aqueles que apresentam coeficientes de regressão  $\beta_j$  em uma forma linear onde os valores previstos correspondentes são funções lineares de tal coeficiente. (<http://leg.ufpr.br/~fernandomayer/aulas/ce074/modelos-lineares.html>)

possibilitaram a automatização da extração de representações complexas entre os dados, passando por uma estrutura hierarquizada de camadas, as quais recebem a ajuda de um algoritmo de retropropagação para compreender como a máquina deve alterar seus parâmetros internos para computar as representações das próximas camadas (Geron, 2019).

Este desenvolvimento trouxe à tona diversas técnicas de implementação do Deep Learning, e com elas surgiram diversos estudos que tinham como objetivo compreender qual delas se firmava como a mais eficaz, além é claro de realizar comparações com os Modelos Lineares (Fadlalla & Lin, 2001) (Vijay & Rakesh, 1998) (Refenes, et al., 1994) (Huang, et al., 2007). Porém, em todos os casos os estudos foram conduzidos em poucas ações e em amostras temporais com cenários econômicos constantes e bem definidos, além de utilizarem técnicas que, como exemplificado por Gamboa (Gamboa, 2017), não se encontram entre as mais propícias para o trabalho com séries temporais.

Nos últimos anos houve o desenvolvimento de uma nova técnica de *Deep Learning*, as Redes Neurais Recorrentes (RNN). Tais redes incorporam a habilidade de analisar conexões ao longo de séries temporais, lançando mão de uma *Long Short Term Memory* (LSTM). Dado que eventos que ocorrem hoje podem impactar um ativo somente no futuro, as LSTMs tornam as RNNs interessantes para o desafio de se prever o retorno de uma ação, sendo inclusive citadas por Alberg e Lipton (2018).

## 2. Metodologia

### 2.1. Estruturas de Dados

#### 2.1.1. Ações

O primeiro passo para desenvolver qualquer modelo é entender a estrutura básica de dados atrelada ao mercado. Nesse estudo são utilizados dados referentes ao fechamento do mercado, representados diariamente por dados de Abertura, Máximo, Mínimo, Fechamento e Volume.

Pensando em uma ação, a Abertura se refere ao preço do ativo no momento de abertura do mercado. Já o Máximo e Mínimo, indicam os valores de máximo e mínimo do ativo ao longo das variações intradia. O Fechamento indica o valor do ativo no momento de fechamento do mercado. Por fim, o Volume mostra quantas unidades do determinado ativo foram transacionadas ao longo do dia. Um exemplo desses dados pode ser visto na Tabela 1.

<b>Data</b>	<b>Abertura</b>	<b>Máximo</b>	<b>Mínimo</b>	<b>Fechamento</b>	<b>Volume</b>
Jan 23, 2020	R\$ 30,80	R\$ 31,25	R\$ 30,19	R\$ 31,23	14.887.900
Jan 22, 2020	R\$ 31,10	R\$ 31,21	R\$ 30,85	R\$ 31,02	11.883.700
Jan 21, 2020	R\$ 31,72	R\$ 31,84	R\$ 31,02	R\$ 31,02	7.708.800
Jan 20, 2020	R\$ 31,70	R\$ 32,14	R\$ 31,52	R\$ 31,98	5.121.200
Jan 17, 2020	R\$ 31,30	R\$ 31,85	R\$ 31,30	R\$ 31,85	13.844.600
Jan 16, 2020	R\$ 31,20	R\$ 31,33	R\$ 30,93	R\$ 31,22	6.655.700

*Tabela 1 – Amostra histórica de dados da PETR3.*

Para a análise foram selecionadas as vinte e seis empresas presentes no índice *S&P500* desde sua criação, além do próprio valor do índice.

<b>MO</b>	<b>AEP</b>	<b>FL</b>	<b>GE</b>
<b>BA</b>	<b>CAT</b>	<b>HON</b>	<b>HPQ</b>
<b>CVX</b>	<b>KO</b>	<b>IP</b>	<b>JNJ</b>
<b>DTE</b>	<b>XOM</b>	<b>MCD</b>	<b>MRK</b>
<b>GT</b>	<b>IBM</b>	<b>MRO</b>	<b>PG</b>
<b>MMM</b>	<b>ARNC</b>	<b>CNP</b>	<b>ED</b>
	<b>DIS</b>	<b>UTX</b>	

*Tabela 2 – Símbolo (Ticker) dos Ativos utilizados como variáveis preditivas*

### 2.1.2. Fatores de Prêmio

A fim de compreender o impacto causado pelos fatores de prêmio de risco na predição de valores futuros de um ativo foram utilizados quatro fatores diferentes. Esses correspondem aos fatores propostos por Fama e French em seu estudo inicial, além do fator de retorno livre de risco do mercado. Todas as variáveis possuem periodicidade diária.

O *Market Beta* é uma medida da volatilidade de um portfólio quando posta frente à volatilidade do mercado como um todo. Já o *Size Beta* compara portfólios compostos por ações de empresas com baixo valor de mercado com portfólios compostos por empresas com elevado valor de mercado. O *Value Beta* faz o comparativo entre a razão do valor contábil e o valor de mercado das empresas, comparando portfólios onde essa razão é alta com portfólios de razão baixa. Por fim, o *Risk Free* representa a taxa de juros livre de um investimento livre de risco.

Data	Risk Free	Market Beta	Value Beta	Size Beta
Jan 23, 2020	0,006	0,08	-0,16	-0,05

*Tabela 3 - Exemplo de Valores para os Fatores de Prêmio*

### 2.1.3. A Barreira Tripla

Uma estratégia utilizada em modelos financeiros de Machine Learning é utilizar o sinal do retorno de um ativo como variável discreta a ser prevista, transformando o regressor em um classificador. Virtualmente, a construção desse sinal, representada pela equação 2, é dada pelo retorno  $r_{0,T}$ , mostrado na equação 1, da ação após uma janela fixa de  $T$  dias e considera um valor mínimo de retorno  $\partial$ .

$$r_{0,T} = \frac{p_T - p_0}{p_0} \quad (1)$$

$$y_i = \begin{cases} -1 & \text{if } r_{0,T} < -\partial \\ 0 & \text{if } |r_{0,T}| \leq \partial \\ 1 & \text{if } r_{0,T} > \partial \end{cases} \quad (2)$$

Porém, como elucidado por Lopez de Prado (2018), esta abordagem desconsidera cenários de alta volatilidade, onde o limiar  $\partial$  pode ser atingida antes do final da janela T. Por esse motivo Lopez propõe uma segunda estratégia para a classificação de ativos: o modelo da Barreira Tripla.

Neste, a classificação  $y_i$  é dada pela primeira barreira atingida dentro do espaço de tempo T. As duas barreiras horizontais são definidas a partir do valor mínimo de retorno  $\partial$ , enquanto a barreira vertical une estas duas no instante T.

Caso a barreira horizontal superior seja atingida primeiro, dá-se à  $y_i$  o valor de 1, já na situação da barreira horizontal inferior ser atingida primeiro, dá-se à  $y_i$  o valor de -1. Para o caso de a barreira vertical ser atingida, o qual representa o caso de nenhuma barreira horizontal ser atingida durante a janela (tempo), existem duas alternativas: atribuir o valor de 0, ou atribuir o sinal do retorno naquele instante, como na estratégia anterior. A Figura 2 demonstra a implementação deste modelo, todo o código está disponível em um repositório no GitHub<sup>2</sup>.

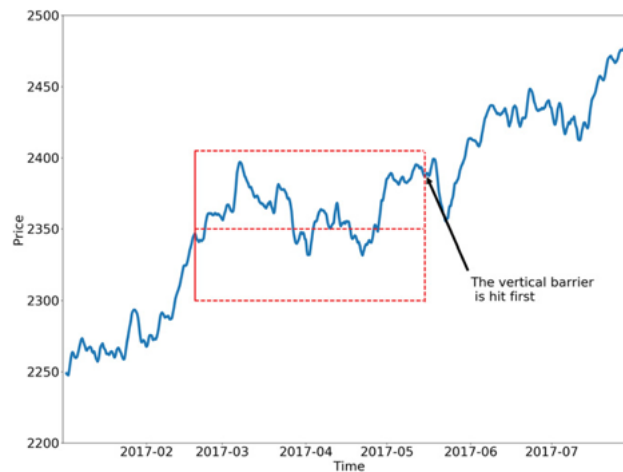


Figura 1 - Exemplo de Aplicação do modelo de Barreira Tripla

<sup>2</sup> <https://github.com/VFermat/NeuralNetworkFactors>



```

def calculateSignal(row, windowSize):
    if row.isna().values.any():
        return 0
    actualPrice = row.P
    pastKeys = [f'P-{i}' for i in range(1, windowSize + 1)] + ['P']
    futureKeys = [f'P+{i}' for i in range(1, windowSize + 1)]
    vol = row[pastKeys].ewm(span=windowSize).std()
    superiorBarrier = actualPrice + 1*vol
    inferiorBarrier = actualPrice - 1*vol
    try:
        supBreakpoint = list(row[futureKeys] > superiorBarrier).index(True)
    except:
        supBreakpoint = -1
    try:
        infBreakpoint = list(row[futureKeys] < inferiorBarrier).index(True)
    except:
        infBreakpoint = -1
    if infBreakpoint == -1 and supBreakpoint == -1:
        return 0
    elif infBreakpoint < supBreakpoint:
        return 1
    else:
        return 2

def buildBarriers(df, stock, windowSize=10):
    temp = df[[stock]]
    for i in range(1, windowSize + 1):
        temp[f'P+{i}'] = temp[stock].shift(-i)
        temp[f'P-{i}'] = temp[stock].shift(i)

    temp['P'] = temp[stock]
    return temp.apply(calculateSignal, axis=1, args=(windowSize, ))

```

*Figura 2 – Implementação da Barreira Tripla.*

#### 2.1.4. Dados de Entrada e Variáveis Previstas

Para testar as redes e o impacto das variáveis nas previsões foram criados dois modelos de *input* de dados, um utilizando os fatores de risco e ativos descritos na seção 2.1, e outro contendo apenas os ativos. Também foram criadas duas estruturas de variável de previsão, uma delas qualitativa e outra quantitativa. A qualitativa trata do valor nominal do *S&P500*, enquanto a quantitativa representa a classificação extraída do modelo de Barreira Tripla. Exemplos de *inputs* podem ser vistos na Tabela 4.

		1970-02-02	1970-02-03	1970-02-04
<b>Ações</b>	<b>MMM</b>	6,273438	6,4375	6,304688
	<b>MO</b>	0,35026	0,359375	0,354167
	<b>AEP</b>	28,25	28,375	28,25
	<b>ARNC</b>	6,149269	6,254685	6,160982
	<b>BA</b>	0,73251	0,748971	0,716049
	<b>CAT</b>	3,041667	3,104167	3,0625
	<b>CNP</b>	10,46725	10,572625	10,642875
	<b>CVX</b>	2,875	2,851563	2,875
	<b>KO</b>	0,830729	0,858073	0,859375
	<b>ED</b>	6,75	6,6875	6,75
	<b>DTE</b>	21,5	21,5	21,75
	<b>XOM</b>	1,75	1,785156	1,78125
	<b>FL</b>	8,40625	8,625	8,71875
	<b>GE</b>	0,732422	0,727414	0,727414
	<b>GT</b>	13,5	14,0625	13,75
	<b>HON</b>	3,773754	3,813478	3,714169
	<b>HPQ</b>	0,356122	0,353017	0,348582
	<b>IBM</b>	17,15	17,1625	17,1625
	<b>IP</b>	8,166913	8,351825	8,44428
	<b>JNJ</b>	1,152778	1,145833	1,095486
	<b>MRO</b>	5,447205	5,592187	5,612899
	<b>MCD</b>	0,320988	0,333333	0,341564
	<b>MRK</b>	1,423611	1,520833	1,479167
	<b>PG</b>	1,71875	1,75	1,753906
	<b>DIS</b>	0,735793	0,735793	0,746065
	<b>UTX</b>	1,058594	1,027344	1
<b>Fatores de Risco do S&amp;P500</b>	<b>Mkt-RF</b>	0,82	1,26	-0,74
	<b>SMB</b>	-0,64	-0,16	-0,31
	<b>HML</b>	-0,13	-0,3	0,73
	<b>RF</b>	32	32	32

Tabela 4 - Exemplo de Input com os Fatores de Prêmio

## 2.2. Preparação dos Dados

Dada as diferenças de grandeza entre os preços de diferentes ações, e o preço das ações ao longo do tempo, foi necessário aplicar uma normalização das variáveis, tanto preditivas quanto preditas.

## 2.3. Estrutura das Redes

Utilizando os módulos Keras<sup>3</sup> e SciPy<sup>4</sup> do python foram dois modelos principais de redes, cada um com uma topologia específica, tais topologias são explicitadas nas subseções abaixo. Para cada modelo foram construídas duas redes diferentes: um regressor e um classificador. O regressor foi utilizado quando a variável predita era o valor nominal do *S&P500*, enquanto o classificador foi utilizado para o modelo de barreira tripla.

Além disso, os modelos foram replicados para que fossem estudados os impactos da estratégia de retreinamento. A estratégia de retreinamento consiste em incrementar a estrutura padrão de treinamento de um modelo, dando novas informações para a rede aos poucos. Por sua vez, a estratégia padrão consiste em treinar a rede de uma só vez.

Na estratégia, a rede é treinada primeiro com o conjunto de aprendizagem, e inicia então o processo de teste. A partir disso, a rede passa a receber sucessivos treinamentos depois de uma janela temporal pré-definida. Isso serve para que a rede possa absorver novas variações e padrões apresentados pelas variáveis.

Essa técnica torna-se válida por estar-se trabalhando com um modelo de Série Temporal, onde cada amostra está atrelada à um momento no tempo. Com isso, entende-se que, ao se passar  $n$  amostras de teste, pode-se absorver as informações apresentadas nessa janela para refinar os parâmetros das redes na previsão da amostra  $n + 1$

---

<sup>3</sup> Keras é uma biblioteca de redes neurais *open-source* escrita em Python. Extremamente modular, funciona como um *wrapper* para bibliotecas de mais baixo nível. (<https://keras.io>)

<sup>4</sup> SciPy é uma biblioteca *open-source* escrita em Python e voltada à computação científica. (<https://www.scipy.org>)

### 2.3.1. Modelo com Long Short Term Memory (LSTM)

As LSTMs podem analisar conexões dentro de séries temporais. Essa estrutura é melhor detalhada na Figura 3. Para este modelo foi escolhido utilizar duas LSTMs em paralelo, fortalecendo assim a análise histórica. Dado que a saída destas redes é em forma de vetor, e o valor buscado é um numeral único, faz-se necessário o uso de ao menos uma *Dense*<sup>5</sup> para redimensionar a saída. Além disso, também foi utilizada uma rede extra de *Denses* na saída da LSTM. A topologia é apresentada pela Figura 4.

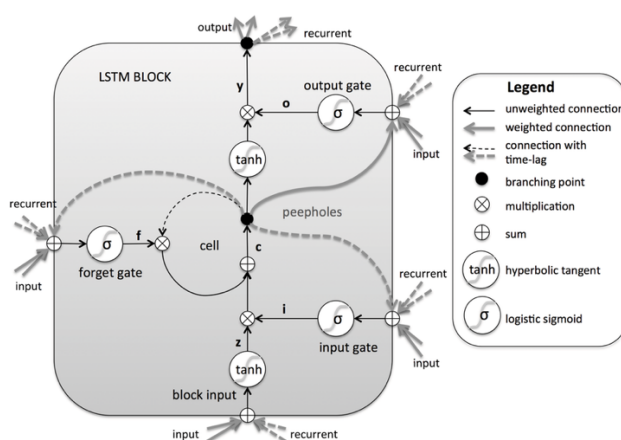


Figura 3 – Detalhamento de uma LSTM

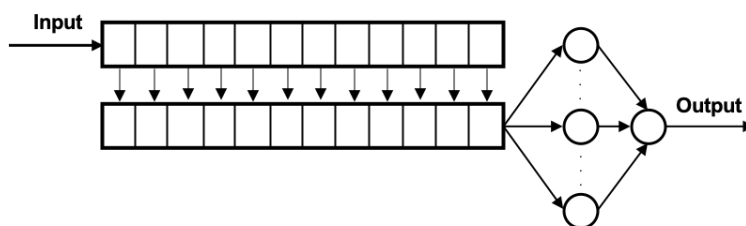


Figura 4 – Exemplificação do modelo de rede utilizado pela LSTM

<sup>5</sup> A *Dense* é a camada responsável por implementar a função de ativação nas informações inseridas, consolidando-as em uma saída. (<https://keras.io/layers/core>)

### 2.3.2. Modelo sem Long Short Term Memory (LSTM)

O modelo sem LSTMs consiste apenas em inúmeras camadas constituídas por *Denses*. A camada de entrada possui um número de neurônios equivalente ao número de variáveis preditivas; as camadas intermediárias, chamadas de *Hidden Layers*, possuem um número maior de *Denses*; a camada de saída consolida os cálculos em um valor único, o qual é a variável predita.

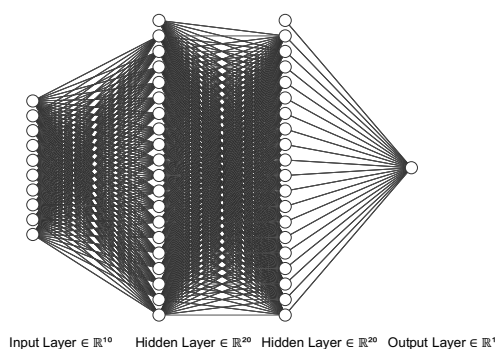


Figura 5 - Exemplificação do modelo composto apenas por denses

## 2.4. Os Métodos de Análise

Existem inúmeros métodos para se medir a eficácia de um modelo preditivo. Uma medida muito utilizada é o erro médio quadrático (*Mean Squared error* ou MSE) (2009). No estudo essa técnica foi utilizada com as redes regressoras para entender quão próximas as predições estavam dos resultados reais do mercado, tendo assim como objetivo a minimização de tal estimador. Por ser amplamente utilizada em estudos estatísticos, as bibliotecas utilizadas nas construções dos modelos possuem métodos prontos de cálculo.

É válido citar a necessidade de se calcular o MSE nos valores não normalizados, para que seja possível comparar os modelos com modelos que não necessitam da normalização para o processamento. Para efeitos de comparação entre os modelos descritos neste estudo por outro lado, pode-se utilizar o MSE calculado sobre os valores normalizados.

Além do modelo regressor, foi construído também um classificador que tem como resultado um valor discreto, e não contínuo como o regressor. Portanto, não é possível utilizar o MSE para analisar a eficácia do modelo. Para o classificador foi então utilizado uma métrica de acurácia, que calcula o percentual de acerto do modelo.

Além disso, a viabilidade da utilização dos modelos em situações reais tem de ser analisada,

dado sua possível utilização no mercado de ações. Para isso é possível usufruir da técnica do *Backtesting*.

Durante esse teste é possível visualizar como uma estratégia se comportaria frente ao mercado, e assim, se ela possui alguma vantagem se comparada aos métodos tradicionais. Para isso faz-se necessário a definição de um *benchmark*, utilizado como comparativo para com a estratégia criada. É comum ao mercado definir este como algo que possua correlação com a carteira de investimentos. Logo, pelo fato de o modelo prever o índice *S&P500*, é válido defini-lo como base comparativa.

Como comparativo foi construída uma estratégia sintética de *Buy & Hold* do índice, com o intuito de descobrir qual seu resultado histórico, e poder assim entender a aplicabilidade da rede e da estratégia.

### 3. Resultados Obtidos

#### 3.1. Modelo Regressor

Os primeiros testes realizados foram comparativos entre as duas redes escolhidas. Tais testes buscavam entender qual topologia possuía melhor performance a níveis preditivos, e se aproxima mais da curva histórica do *S&P500* (Figura 6).



*Figura 6 - Evolução Histórica do S&P500 (USD x Ano)*

Para isso as Redes foram treinadas com dados históricos até o início do ano de 2009. Com isso garantiu-se que ela passasse por diferentes momentos do mercado, podendo capturar informação em todos eles. Com as Redes treinadas foi possível trabalhar em cima da amostra de testes, que corresponde à valores do início do ano de 2009 até o meio do ano de 2019. As redes criadas, e suas características principais, podem ser visualizadas na Tabela 5.

Ao analisar os resultados percebe-se que existe uma diferença significativa entre os resultados apresentados pelas Redes com Retreinamento e as Redes sem o mesmo. Tal fato leva a crer que essa técnica é mais efetiva na previsão, principalmente no longo prazo, o que pode ser percebido após analisar as figuras comparativas entre as previsões e o resultado real do índice.

Foram calculados os MSEs de todas as amostras, além de uma representação visual e comparativa entre o gráfico histórico do mercado e as previsões feitas pelas Redes. Tais resultados podem ser observados abaixo, na Figura 7 e na Tabela 5.

<b>Identificador</b>	<b>Topologia</b>	<b>Retreinamento?</b>	<b>Fatores de Risco?</b>	<b>MSE Normalizado</b>	<b>MSE</b>
R1	LSTM	Sim	Sim	0,0002653	2476,6948
R2	LSTM	Sim	Não	0,0003317	2762,5841
R3	LSTM	Não	Sim	0,0068881	57273,2978
R4	LSTM	Não	Não	0,0564872	469683,1300
R5	Dense	Sim	Sim	0,0003292	2737,3691
R6	Dense	Sim	Não	0,0002532	2356,6111
R7	Dense	Não	Sim	0,01151	95701,0128
R8	Dense	Não	Não	0,00804	66854,7444

*Tabela 5 - Redes Criadas para Testes*

De forma diferente, a comparação entre modelos que possuem as variáveis de fatores de risco, com exceção ao modelo com LSTM, não apresentam grandes diferenças em seus resultados, tendo inclusive desempenho abaixo do modelo sem os fatores quando a topologia utilizada foi a *Dense*. Por fim, vale citar a proximidade dos resultados das diferentes topologias nas situações onde o retreinamento foi utilizado. Quando os fatores de risco foram utilizados, a LSTM apresentou um MSE 24,03% menor que o resultado da Dense. Já quando os fatores não foram utilizados, a LSTM apresentou um MSE 23,66% maior.



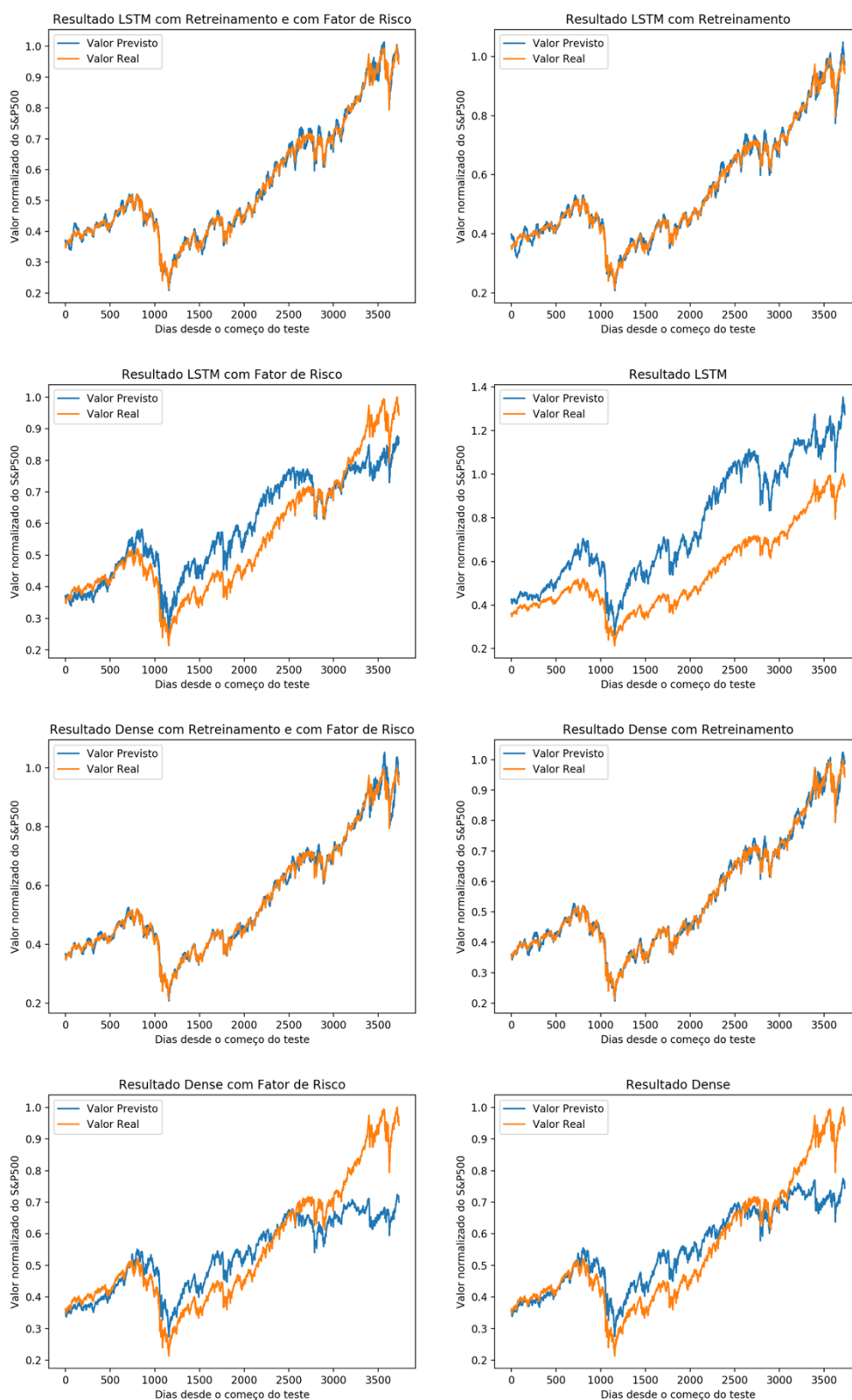


Figura 7 - Resultados das Redes Criadas

Para poder compreender o impacto causado pelo Modelo durante a aplicação de uma estratégia real de *Trading* é necessário simular com ambiente real. Essa simulação é realizada durante o *Backtesting* utilizando uma estratégia hipotética e comparando-a ao benchmark escolhido, que, como dito anteriormente, é o retorno do *S&P500*, ilustrado pelas Figuras 8 e 9. Dada a melhor performance dos modelos quando utilizado o retreinamento, a técnica foi definida como padrão para o *Backtesting*. Com isso, o número de redes testadas foi reduzido para quatro, as quais podem ser vistas na Tabela 6.

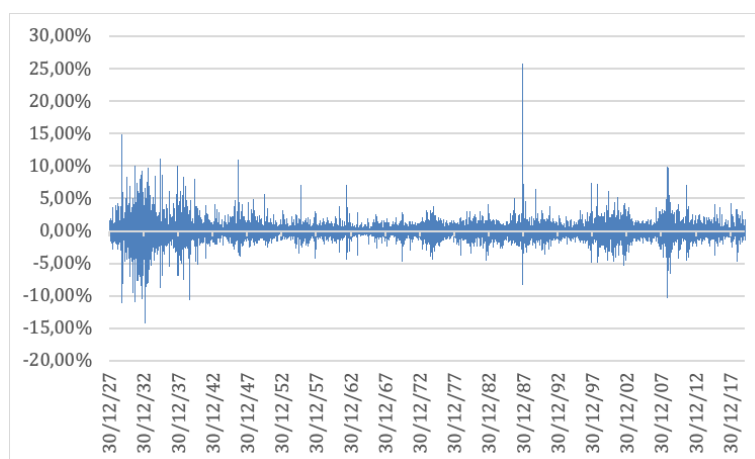


Figura 8 - Retorno diário do S&P500 desde seu início

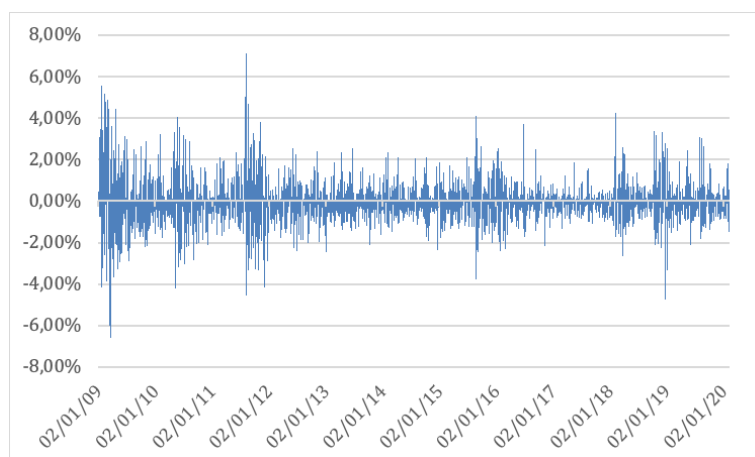


Figura 9 - Retorno diário do S&P500 desde 2009

Durante a simulação é importante a observação de algumas métricas que indicam boa performance da estratégia, e consequentemente do modelo. São elas: *hitting ratio*, que indica o percentual de acerto quanto ao sinal da variação do ativo; número de transações, P&L bruto, que juntos demonstram um panorama geral da performance da estratégia; e o retorno bruto médio, um dos balizadores do comparativo com o *benchmark*.

Identificador	Topologia	Fatores de Risco como variáveis?
B1	LSTM	Sim
B2	LSTM	Não
B3	Dense	Sim
B4	Dense	Não

*Tabela 6 - Redes utilizadas no Backtesting*

	B1	B2	B3	B4
Hitting Ratio	52,02%	56,70%	58,62%	57,79%
Número de Transações	223	97	145	154
P&L Bruto	737,39	530,29	501,67	287,57
Retorno Médio	0,13%	0,23%	0,29%	0,05%

*Tabela 7 - Métricas obtidas durante o Backtesting*

Percebe-se que, apesar de um melhor resultado bruto apresentado, as redes que utilizam os fatores de risco como variáveis não apresentam evidência de melhor performance quanto à taxa de acerto de direção do mercado.

### 3.2. Modelo Classificador (Barreira Tripla)

Neste modelo as redes têm como variável predita a classificação da Barreira Tripla para o dia analisado.

Assim como no modelo regressor, foram criadas redes com características distintas para observar o comportamento do retreinamento e dos fatores de prêmio de risco na eficácia das redes. As variantes podem ser encontradas na Tabela 5. De mesma forma, todas elas foram treinadas com informações que englobam o período de tempo do início da operação das ações até o início do ano de 2009, sendo depois testadas no espaço temporal entre 2009 e 2019. Os resultados obtidos podem ser encontrados na Tabela 8.

É importante ressaltar que este não é um modelo classificador binário, já que são três as possibilidades de classificação, como explicitado na seção 2.1.3. Este fato faz com que a linha de base comparativa seja de 33,3%, mostrando que as melhores performances são de modelos

que não apresentam retreinamento. Isso pode ser explicado pelo contexto do teste realizado, o qual ocorreu em um período de altas subsequentes do índice *S&P500*, podendo ter desbalanceado as predições.

<b>Identificador</b>	<b>Acurácia</b>
R1	33,95%
R2	35,27%
R3	47,52%
R4	54,15%
R5	29,80%
R6	30,18%
R7	33,41%
R8	47,33%

*Tabela 8 - Resultados das Redes Classificadoras*

#### 4. Conclusão

Investidores e pesquisadores continuam buscando uma abordagem eficaz para redes neurais no mercado de ações, dado o potencial retorno de uma aplicação bem-sucedida. Porém, o comportamento aparentemente aleatório do mercado faz dessa uma tarefa complexa.

Uma abordagem proposta pela literatura é a implementação de teorias econômicas clássicas utilizando um ferramental computacional mais moderno. Uma destas teorias é a composição de fatores de prêmio de risco, que, em teoria proposta por Fama e French, tendem a explicar os retornos em excessos das ações. Utilizar tais variáveis deveria, portanto, melhorar a performance de um modelo comum de redes neurais preditivas e classificativas.

O que é visto, porém, é que isso não ocorre na prática, com os modelos que utilizam fatores de prêmio tendo performance equiparável com seus pares que não utilizam a variável extra. Contudo, estes resultados não devem ser interpretados como uma falha da combinação das redes neurais com os fatores de prêmio, mas evidência da eficiência de mercados maduros, como o americano, objeto deste estudo.

Além disso, foi testada a hipótese de que a utilização da Barreira Tripla, proposta por Lopez de Prado, melhoraria a assertividade das previsões das redes. Porém os resultados obtidos não foram suficientes para validá-la. Mas assim como os resultados apresentados no parágrafo acima, isso não deve ser interpretado como falha do modelo, podendo ser resultado de um conjunto de teste e treinamento não favorável ou problemas na parametrização do delta utilizado na equação de classificação, sendo assim necessária a realização de novos estudos.

Por fim, mostrou-se também a eficiência do processo de retreinamento das redes no panorama preditivo. Estratégias de processamento de dados e de melhorias no treinamento de redes neurais são objeto de diversos estudos, dado que potencializam os resultados obtidos. Levando em consideração a natureza histórica das informações tratadas, o retreinamento se torna justo, à medida que em um processo natural de uso o volume de informações abertas para treinamento cresce ao passar do tempo.

## 5. Bibliografia

- Alberg, J. & Lipton, Z. C., 2018. Improving Factor-Based Quantitative Investing by Forecasting Company Fundamentals.
- Alessio, E., Carbone, A., Castelli, G. & Frappietro, V., 2002. Second-order moving average and scaling of stochastic time series. *The European Physical Journal B - Condensed Matter and Complex Systems*, Volume 27, pp. 197-200.
- Batres-Estrada, B., 2015. Deep Learning for Multivariate Financial Time Series.
- Fama, E. F. & French, K. R., 1992. The Cross-Section of Expected Stock Returns. *The Journal of Finance*, Issue 47, p. 427.
- Fadlalla, A. & Lin, C., 2001. An Analysis of the Applications of Neural Networks in Finance. *Interfaces*, 31(4), pp. 112-122.
- Gamboa, J., 2017. Deep Learning for Time-Series Analysis.
- Geron, A., 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2nd Edition ed. s.l.:O'Reilly Media, Inc..
- Haykin, S., 1994. *Neural Networks: A Comprehensive Foundation*. s.l.:Prentice Hall PTR.
- Huang, W. et al., 2007. Neural Networks in Finance and Economics Forecasting. *International Journal of Information Technology & Decision Making*, 6(1), pp. 113-140.
- Kendall, M. G. & Hill, A. B., 1953. The Analysis of Economic Time-Series - Part I: Prices. *Journal of the Royal Statistical Society. Series A (General)*..
- Fama, E. F. & French, K. R., 1992. The Cross-Section of Expected Stock Returns. *The Journal of Finance*, Issue 47, p. 427.
- Wanjawa, B. W. & Muchemi, L., 2014. *ANN Model to Predict Stock Prices at Stock Exchange Markets*, s.l.: s.n.
- Pantazopoulos, K. et al., 1998. Financial Prediction and Trading Strategies Using Neurofuzzy Approaches. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 28(4), pp. 520-531.
- Batres-Estrada, B., 2015. Deep Learning for Multivariate Financial Time Series.
- Gamboa, J., 2017. Deep Learning for Time-Series Analysis.
- Alberg, J. & Lipton, Z. C., 2018. Improving Factor-Based Quantitative Investing by Forecasting Company Fundamentals.
- Fadlalla, A. & Lin, C., 2001. An Analysis of the Applications of Neural Networks in Finance. *Interfaces*, 31(4), pp. 112-122.

- Vijay, S. D. & Rakesh, B., 1998. A Comparison of Linear Regression and Neural Network Methods for Predicting Excess Returns on Large Stocks. *Annals of Operations Research*, Volume 78, pp. 127-163.
- Refenes, A. N., Zapranis, A. & Francis, G., 1994. Stock Performance Modeling Using Neural Networks: A Comparative Study with Regression Models. *Neural Networks*, 7(2), pp. 375-388.
- Alessio, E., Carbone, A., Castelli, G. & Frappietro, V., 2002. Second-order moving average and scaling of stochastic time series. *The European Physical Journal B - Condensed Matter and Complex Systems*, Volume 27, pp. 197-200.
- Huang, W. et al., 2007. Neural Networks in Finance and Economics Forecasting. *International Journal of Information Technology & Decision Making*, 6(1), pp. 113-140.
- Geron, A., 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2nd Edition ed. s.l.:O'Reilly Media, Inc..
- Haykin, S., 1994. *Neural Networks: A Comprehensive Foundation*. s.l.:Prentice Hall PTR.
- Lopez de Prado, M., 2018. *Advances in Financial Machine Learning*. s.l.:John Wiley & Sons.
- Bodie, Z., Kane, A. & Marcus, A. J., 2013. *Investments*. 10 ed. s.l.:MCGRAW-HILL PROFESSI.
- Wang, Z. & Bovik, A. C., 2009. Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. *IEEE Signal Processing Magazine*, 26(1), pp. 98-117.