

# Otimização de Portfólio Futuro Baseado em Aprendizagem Profunda e Algoritmo Evolutivo Multiobjetivo

Elaine Pinto Portela<sup>1</sup>, Omar Andres Carmona Cortes<sup>2</sup>

<sup>1</sup>Departamento de Computação – Instituto Federal do Maranhão (IFMA)  
Av. Getulio Vargas, 04 – 65.030-005 – São Luis – MA – Brasil

elainepportela@gmail.com, omar@ifma.edu.br

**Abstract.** *The stock market is essential to economic growth since it provides investors the opportunity of rising heritage and provides financial resources to companies. However, its dynamic behavior is a challenge for both humans and algorithms. Thus, this work proposes a regression model using a deep neural network (LSTM) to discover patterns in historical stock data in order to predict future stock values. Afterwards, we optimize a portfolio using a multiobjective evolutionary algorithm named MOEA/D. This work used data from two famous indexes: S&P500 and Dow Jones. Results indicate that our model associated with the MOEA/D can suggest portfolios returning between 14% and 22% in the SP500 index. Also, between 4% and 10% in the Dow Jones. Both according to the risk that an investor is willing to face.*

**Resumo.** *O mercado de ações é importante para o crescimento econômico, pois fornece a oportunidade para investidores formar patrimônio e fornece recursos para o crescimento de organizações. Porém, o comportamento dinâmico do mercado é um desafio tanto para humanos quanto para algoritmos. Assim, este trabalho propõe um modelo regressivo usando uma rede neural profunda (LSTM) para encontrar padrões nos dados históricos de ações e tentar prever os valores esperados para o futuro. Em seguida, faz-se a otimização do portfólio utilizando um algoritmo evolutivo multiobjetivo chamado MOEA/D. Os dados utilizados nos testes são dados históricos de dois índices conhecidos: o S&P 500 e o Dow Jones. Os resultados indicam que o modelo de predição associado ao MOEA/D pode sugerir portfólios com retorno entre 14% e 22% no índice S&P 500, e entre 4% e 10% no índice Dow Jones dependendo do risco que o investidor esta disposto a assumir.*

## 1. Introdução

O Mercado de Ações vem ganhando destaque entre os brasileiros, resultado de uma modernização tecnológica [Pereira et al. 2013]. Assim, o índice IBOVESPA apresentou um aumento de pequenos investidores de 3% para 20% entre 1998 e 2012 [Economatica 2015]. O que resulta em uma expansão na economia [Bovespa 2000]. No entanto, devido a natureza volátil das ações negociadas, caracterizada como uma série temporal aleatória ou um processo estocástico não estacionário [Morettin e Toloí 2011], a montagem do portfólio ideal se torna mais difícil. Logo é notável a importância de estudos que auxiliem a montagem de um portfólio futuro que se aproxime da fronteira ótima. Assim, este trabalho propõe a utilização de uma rede neural profunda, a LSTM

[Hochreiter e Schmidhuber 1997] para prever o valor futuro de um ativo e um algoritmo multiobjetivo, MOEA/D [Zhang e Li 2008] para recomendar portfólios.

O ponto alto de auxiliar o mercado de ações é aproximar o detentor do recurso (investidor) com quem precisa do investimento para crescer (empresa), resultando em uma maior expansão da capacidade de produção e uma maior expansão no desenvolvimento econômico [Bovespa 2000]. No entanto, devido a natureza volátil das ações negociadas, caracterizada como uma série temporal aleatória ou um processo estocástico não estacionário [Morettin e Toloí 2011], e ao aumento no volume de informações a escolha dos ativos para a montagem do portfólio se torna desafiante.

Nesse contexto, este artigo está dividido em: a Seção 2 mostra os trabalhos relacionados; a Seção 3 apresenta a literatura necessária; a Seção 4 detalha os experimentos na previsão usando a LSTM e a otimização de portfólios usando o MOEA/D; finalmente, a Seção 5 apresenta as conclusões e os futuros trabalhos.

## 2. Trabalhos Relacionados

Em [Björklund e Uhlin 2017] faz-se a previsão de valores e a otimização de portfólios solicitada pela empresa Söderberg & Partners (S&P). Para a previsão dos valores foi utilizado um modelo de rede neural *feed-forward* e para a otimização de portfólios foram utilizados processos estocásticos. Na mesma linha [Weng 2017] propõe um sistema, utilizando rede neural, de auxílio a tomada de decisão prevendo os movimentos de curto prazo, tendência e preço das ações. Em [Heaton et al. 2016] investiga-se a utilização da aprendizagem profunda para problemas de classificação e previsão financeira, detectando uma seleção de investimentos que cumpra os objetivos propostos, que é chamado no artigo de *deep portfolios*. A Tabela 1 faz um sumário de alguns trabalhos e as técnicas de IA utilizadas em previsões no mercado de ações.

**Tabela 1. Estudos sobre a previsão do Mercado de Ações.**

Autores	Objetivo	Método
[Enke e Mehdiyev 2013]	Preço	Seleção de recursos + Agrupamento fuzzy + Rede neural fuzzy
[Niaki e Hoseinzade 2013]	Direção do Mercado	<i>Feature selection</i> + Rede neural recursiva
[Patel et al. 2015]	Preço	Regressão por SVM + Rede neural + <i>Random Forest</i>
[Chen e Chen 2016]	Direção do Mercado	Redução de dimensionalidade + <i>template matching</i>
[Chiang et al. 2016]	Sinalização	Otimização por nuvem de partículas + Rede neural
[Chourmouziadis e Chatzoglou 2016]	Otimização de Portfólio	Sistema Fuzzy
[Qiu et al. 2016]	Retorno	Rede neural + Algoritmo genético + <i>Simulated annealing</i>
[Arévalo et al. 2016]	Preço	Rede neural profunda
[Zhong e Enke 2017]	Direção do Mercado	Redução de dimensionalidade + Rede neural
[Chong et al. 2017]	Retorno	Rede neural profunda

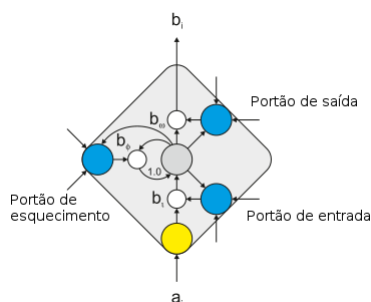
Na linha mais tradicional, em [Xiao et al. 2018] é investigada a otimização de portfólios propondo uma nova versão do MOEA/D baseada em distância de referência que pode resolver problemas de 2 a 4 objetivos. No trabalho de [Chan et al. 1999] investiga-se a previsão dos retornos das ações para a determinação dos portfólios. São usados vários modelos para a previsão da covariância de retorno, como o *Full Covariance Matrix*, *Factor Models*, *Constant Covariance Model*. Já na escolha do melhor portfólio foi utilizada a técnica de portfólio global de variação mínima, assim o portfólio mais estável é o melhor

portfólio. Em [Krokhmal et al. 2003] pesquisa-se a otimização do portfólio utilizando Valor em Risco Condicional (CVaR - *Conditional Value-at-Risk*) como parâmetro visando maximizar o retorno esperado.

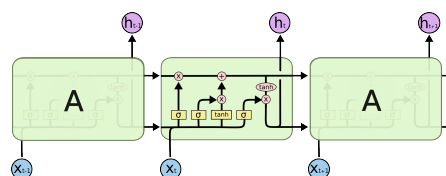
### 3. Referencial Teórico

#### 3.1. Aprendizado Profundo e as Redes LSTM

Simplificando, a principal diferença entre redes neurais profundas e as tradicionais ou rasas é que a primeira possui mais camadas intermediárias, conexões mais complexas e uma quantidade maior de neurônios. Em outras palavras a diferença básica esta na arquitetura da rede, que pode alterar consideravelmente a performance da mesma, sendo quatro as principais arquiteturas de redes neurais profundas: Recursivas, Recorrentes, Convolucionais e Não-supervisionadas [Patterson e Gibson 2017]. Neste trabalho é utilizada uma rede neural recursiva conhecida como LSTM do inglês *Long Short-Term Memory*. Por ter sido proposta como uma solução para o problema de dependências de longo prazo, a LSTM apresenta o comportamento de aprender informações de longo prazo como padrão. A LSTM foi projetada para modelar sequências temporais e as suas dependências de longo prazo de maneira mais precisa do que as redes neurais recorrentes padrão, possuindo uma estrutura de cadeia de quatro camadas ao invés de apenas uma [Sak et al. 2014]. Para isso ela conta com dois principais componentes: a célula de memória (o carrrossel de erro constante) presente durante as interações nas quatro camadas, e portões que podem modificar a célula de memória, como mostrado na Figura 1.



**Figura 1. A célula de memória e os portões da LSTM**



**Figura 2. A unidade da rede LSTM com as quatro camadas**

A arquitetura da rede LSTM (Figura 2) é formada por bloco de memórias onde ficam as células de memória que são responsáveis por solucionar o problema do Gradiente descendente a um baixo custo computacional [Sak et al. 2014]. Cada bloco de memória possui três portões para controlar a mudança da célula de memória, utilizando as funções *sigmoide* e *tanh*, o primeiro é o portão de entrada (*input layer*) que decide quais valores atualizar, prevenindo que informações irrelevantes alterem o estado da rede. O portão de “esquecimento” (*forget gate*) descarta as informações que não são mais relevantes. O terceiro é o portão de saída (*output gater*) que usa uma função para filtrar a versão do estado da célula de memória e então expõe como saída.

O primeiro passo da rede é decidir o que será esquecido da célula de memória pelo portão de esquecimento. A decisão é feita pela função sigmoide que gera um número entre 0 e 1, em que 1 sígnica manter e 0 destruí-la. Depois, decide-se qual nova informação

será adicionada na célula de memória. Primeiro é usada uma função sigmoide no portão de entrada decidindo qual valor vai atualizar. E então é utilizada a função *tanh* que cria um vetor de novos valores candidatos a serem adicionados no estado. No terceiro passo é preciso atualizar o antigo valor da célula de memória ( $C_t$ ) pelos novos valores que foram decididos no passo anterior, primeiro multiplicando o antigo valor ( $C_{t-1}$ ) pelos valores que precisam ser esquecidos ( $f_t$ ) do passo 1 e depois adicionam-se os valores que foram decididos ( $i_t * C_t$ ) no passo 2. No quarto passo é aplicado um filtro no estado da rede para retornar a saída. Primeiro, o valor passará por uma função sigmoide que representa a multiplicação do vetor de pesos da função ( $W_o$ ) com a soma do valor de saída da iteração anterior ( $h_{t-1}$ ) com a entrada dessa iteração ( $x_t$ ) que é somada ao vetor de Bias ( $b_o$ ). O último passo é passar o resultado anterior por uma função *tanh* para convertê-lo em valores entre -1 e 1.

### 3.2. Algoritmo Evolutivo Multiobjetivo Baseado em Decomposição (MOEA/D)

Alguns problemas reais possuem um conjunto de objetivos conflitantes entre si. Esses problemas são conhecidos como multiobjetivo ou multicritério, dependendo do caso. O problema tratado neste trabalho possui dois objetivos conflitantes entre si que são a maximização dos lucros e a minimização dos riscos. Esse conflito entre as funções faz com que o problema não possua apenas uma solução ótima, mas sim um conjunto de soluções não-dominadas denominado de Pareto ótimo ou fronteira de Pareto.

O Algoritmo Evolutivo Multiobjetivo Baseado em Decomposição (MOEA/D - *Multiobjective Evolutionary Algorithm based on Decomposition*) foi proposto por [Zhang e Li 2008] em 2008. A ideia base do algoritmo é decompor o problema em  $N$  subproblemas de otimização, sendo que a cada geração a população de cada subproblema é composta pelas melhores soluções. As relações de vizinhança desses subproblemas serão definidas pela distância entre os seus coeficientes de agregação. Assim cada subproblema será otimizado levando em consideração apenas as informações dos subproblemas vizinhos. Assim, possui uma complexidade computacional menor a cada geração quando comparado a outros como o NSGA-II [Deb et al. 2002].

O MOEA/D possui alguns componentes, um dos mais importantes é a Estratégia de Decomposição, pois é ela que aponta quais são as técnicas que são capazes de decompor um problema de otimização multiobjetivo, determinando um vetor de pesos. Três opções de decomposição podem ser utilizadas: Soma Ponderada (ou *Weighted Sum*), Tchebycheff ou Limite de Interseção (ou *Boundary Intersection*) [Zhang e Li 2008], cada uma seguindo um conjunto de regras. Outro componente do MOEA/D é a função de agregação sendo responsável em gerar os subproblemas a partir do vetor de pesos. Ainda, outro componente é a Estratégia de Escalonamento Objetivo que especifica como serão resolvidas as diferenças nas faixas de valores das funções objetivo. E também a atribuição da vizinhança que define o relacionamento de vizinhança entre os subproblemas.

Outro componente importante é a pilha de variação, responsável por gerar novos valores que sejam candidatos a solução. Outro é a estratégia de solução que define quais soluções candidatas devem ser mantidas ou descartadas a cada iteração. Para tratar os pontos que violam as restrições do problema utiliza-se o método de tratamento de restrições. E o critério de término, determina o ponto de parada do algoritmo. O Algoritmo 1 [Campelo et al. 2018] mostra o pseudo código do MOEA/D. Detalhes sobre seu funcionamento podem ser encontrados em [Zhang e Li 2008] e [Campelo et al. 2018].

---

**Algoritmo 1: Algoritmo MOEA/D**

---

**Input:** Função multiobjetivo  $f(\cdot)$ ; Funções constraints  $g(\cdot)$ ; Dados de entrada  
**Result:** População de saída, Parâmetros da função que gerou a população

```
1 t = 0 ;
2 run = TRUE ;
3 Gerar população inicial  $X^{(t)}$  aleatoriamente no domínio do problema; Gerar pesos  $\Lambda$ ;
5 while run do
6     Definir ou atualizar os vizinhos ;
7     Copiar solução definida  $X^{(t)}$  em  $X'^{(t)}$  ;
8     for cada variação de  $v \in V$  do
9          $X'^{(t)} = v(X'^{(t)})$ ;
10    Avaliar soluções em  $X^{(t)}$  e  $X'^{(t)}$  ;
11    Definir a próxima população  $X^{(t+1)}$  ;
12    Atualizar run;
13     $t = t + 1$  ;

return  $X^{(t)}, f(X^{(t)})$  ;
```

---

### 3.3. Mercado de Ações: Risco e Retorno

A otimização de portfólios pode ser definida como encontrar a melhor combinação de um grupo de ativos, aqui consideradas as ações, que permitam uma relação de custo benefício entre risco e retorno financeiro. O risco e o retorno estão diretamente ligados, ou seja, quanto maior o risco, maior é a probabilidade de retorno e de perda, fazendo que com que a otimização de portfólios seja multiobjetivo.

O risco pode ser definido como a probabilidade de um ativo financeiro perder valor ou não ter o retorno esperado. Uma forma de quantificar o risco é através do *Value at Risk* (VaR) que aponta a perda máxima de uma ativo em determinado período de tempo, com base em um grau de confiança. Estatisticamente, VaR é o valor crítico de 0.01 da distribuição da probabilidade de mudanças no valor do mercado [Duffie e Pan 1997], sendo que no seu cálculo considera-se o período (dia, mês ou ano), um nível de confiança (95%, 99% e 99.9%) e a perda máxima (montante ou porcentagem do que se espera perder). Existem três principais formas de calcular o VaR: o método Histórico, Paramétrico e de Monte Carlo [Harper 2019]. Dentre esses métodos os que levam em consideração o histórico dos preços das ações são o Histórico e o Paramétrico.

O método Histórico se baseia na organização dos retornos históricos assumindo que esses retornos se repetirão, em ordem dos piores para os melhores retornos. O cálculo é feito usando os 5%, 1% ou 0.01% dos piores retornos, no caso do nível de confiança em 95%, 99% e 99.9% respectivamente [Harper 2019]. O método Paramétrico ou variância-covariância se diferencia do método histórico pois leva em consideração a curva histórica, não necessitando a ordenação dos retornos. Para ser feito o cálculo do VaR é necessário primeiro escolher a amostra de retornos a ser trabalhada, levando em consideração o período desejado. Depois estima-se a média e o desvio padrão dos retornos da amostra e calcula-se o desvio que varia de acordo com o grau de confiança escolhido. O cálculo então será a média mais o desvio padrão multiplicado pelo desvio. O retorno em excesso ou *Alpha* são os retornos de investimentos de um ativo que excedem um *benchmark* específico ou um índice com um nível de risco similar, ou seja, desempenharam melhor que o mercado esperava. Assim, o retorno é calculado como o somatório de lucros e per-

das obtidos. Deste modo, a otimização de portfólio consiste em encontrar dentre todas as ações qual o valor ideal a ser investido em cada ação de modo a minimizar o risco e maximizar o retorno, delimitando-se uma Fronteira eficiente, na qual há uma relação de custo-benefício (risco-retorno).

## 4. Experimentos

### 4.1. Metodologia

O experimento é baseado em duas partes. Na primeira parte faz-se a predição do retorno futuro das ações baseado no histórico e usando uma rede LSTM. Em seguida utiliza-se o MOEA/D para otimizar o portfólio. Em outras palavras, desenvolve-se um modelo regressivo para a predição da bolsa de valores com uma determinada base de dados com as cotações diárias, para em seguida utilizá-los para construir portfólios otimizados selecionando uma porcentagem do capital disponível pelas ações disponíveis, auxiliando um investidor ou gerente de ativos.

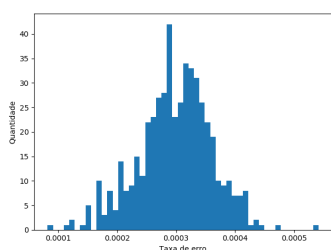
O experimento foi executado em um computador com Linux Mint 18.3 e um Intel Core i7-7500U 2.7 GHz, uma placa de vídeo NVIDIA GeForce 940 MX com 4 Gb de VRAM, e RAM de 8 Gb. Foram utilizadas as IDEs Pycharm para o desenvolvimento do código em Python, usando a biblioteca Keras para a rede neural; e o RStudio para o desenvolvimento do código em R para utilização do MOEA/D. A base de dados utilizada está disponível na plataforma Kaggle [Nugent 2018]. Foram utilizados os dados da S&P 500 que contém todas as ações e seus respectivos valores em uma série temporal diária referente a cotação da ação de 505 empresas durante cinco anos de 08/02/2013 até 07/02/2018. Foram usados também os dados do índice Dow Jones que contém 30 empresas com as cotações diárias no período de 12/05/2012 até 12/05/2017.

Como pré-processamento dos dados foi realizado a retirada dos valores da data e colocado índices na ordem das datas da base de dados, de forma a não prever a cronologia dos dados, visto que é uma série temporal sendo importante seguir a sequência dos valores. Em seguida foram adicionados os valores da média, variância e desvio padrão. E então foi feita a normalização por desvio padrão.

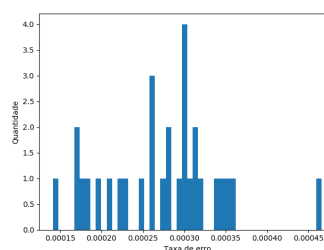
### 4.2. Parte I: Predição de valores

Nessa fase da experimentação foi escolhida uma arquitetura da rede LSTM composta por oito camadas: uma de entrada, uma de saída e seis camadas intermediárias. Após a entrada dos dados na rede, os mesmos são passados para a camada de *Embedding* que é responsável pela transformação de números positivos em vetores densos de tamanho fixo, sendo uma normalização dos dados recebidos. Então são passados para três camadas do tipo LSTM que são responsáveis pelo processamento. Baseados nos testes realizados, a configuração das funções de ativação escolhidas foram em duas camadas a função Sigmóide e em uma camada a função Tanh. Após as camadas LSTM, os dados são passados para uma camada de *Dropout* que realiza o descarte de alguns nós aleatórios de entrada. O *Dropout* foi utilizado porque resolve problemas de uma quantidade maior de camadas da *Deep Learning*, previne o *overfitting* e possibilita a combinação exponencial de arquiteturas de redes neurais diferentes [Srivastava et al. 2014]. A taxa de *dropout* utilizada foi de 0.25. A última camada antes da saída é a *Flatten*, usada para nivelar os resultados.

A otimização dos parâmetros da rede foi feita usando os algoritmos de Adam, AdaGrad, RMSProp e AdaDelta, sendo o Adam o que apresentou melhor desempenho. A taxa de aprendizado utilizada foi a padrão da biblioteca de 0.001 e a taxa de erro foi o RMSE que é a Raiz do Erro Quadrático Médio. O treinamento foi realizado utilizando o K-fold com K=3 como método de amostragem, dividindo a base de dados na proporção aproximada de 70/30, com o período de treinamento de 08/02/2013 a 08/02/2018. O treinamento é feito por empresa em 20 épocas. Após o teste foi gerado um arquivo “.csv” de previsão com as 505 ações do S&P 500 e das 30 ações do Dow Jones. A Figura 3 mostra o histograma com as taxas de erros para as 505 ações do S&P 500, que variou entre 0.0002 e 0.0003. Já para o Dow Jones, a Figura 4 mostra as taxas de erros das 30 empresas que compõem o índice, variando entre 0.00015 e 0.00045.



**Figura 3. Taxas de erro da rede para S&P 500**



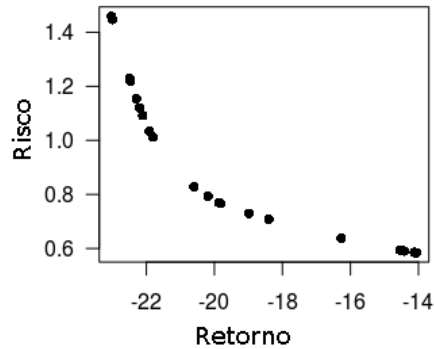
**Figura 4. Taxas de erro da rede para Dow Jones**

### 4.3. Parte II: Otimização de Portfólio

Na segunda fase da experimentação é necessário calcular o risco e retorno de cada ação, utilizando os valores futuros já previstos para serem utilizados como entrada no algoritmo MOEA/D. O retorno foi calculado somando as diferenças de cada dia, e o risco utilizado é o VaR com o método Histórico, que apresentou melhor resultado.

Na implementação do MOEA/D foram passados dois objetivos, maximizar o retorno e minimizar o risco. A biblioteca MOEA/D [Campelo et al. 2018] aceita apenas minimizações, então para atingir a maximização do retorno foi necessário multiplicá-lo por -1. A função de decomposição utilizada foi a padrão sugerida pelos autores do MOEA/D [Zhang e Li 2008], a *WS Weighted Sum*. O mesmo aconteceu na função de variação, foram utilizadas a *SBX Simulated Binary Crossover* e *Mutação Polinomial*. Finalmente, o critério de parada é de 2000 iterações, sendo que cada indivíduo é formado pela porcentagem do orçamento que deve ser investido no ativo, ou seja, um indivíduo é formado por 505 posições. As soluções ótimas propostas mostram a divisão do orçamento entre as ações, vide Figura 5 que mostra a fronteira de Pareto obtida para o S&P 500, logo um portfólio que possui um retorno de 16% tem um risco aproximado de 0.6. O algoritmo fornece um sumário de parâmetros gerados, como o tamanho da população que foi de 300 indivíduos (escolhido empiricamente) e em 2000 iterações foram encontrados 21% (63) pontos não dominados. É feita ainda uma sugestão de ponto ideal como um portfólio com 23% de retorno e um risco de 0.5.

A fim de facilitar a visualização de um dos portfólios da fronteira de Pareto sugerida, foi feito um teste com apenas 20 das 505 ações da S&P 500, Tabela 2. Dentre as ações mais recomendadas foram a EBAY e a EMN com uma sugestão do uso de um



**Figura 5. Fronteira de Pareto - S&P 500**

pouco mais de 99% do recurso do cliente. Já ações como a EL, ESRX e ETFC não foram recomendados.

**Tabela 2. Portfólio encontrado com 20 ações**

Ação	Porcentagem do recurso
DVN	0 %
DWDP	0 %
DXC	0.000000057843725547906%
EA	0.000000000000000735686007059297%
EBAY	87.4036672635803%
ECL	0.000000000000000286799040242216%
ED	0.0001486208850856%
EFX	0.000000000000000574876676082195%
EIX	0.000000000000000327285440871356%
EL	0%
EMN	12.5961567350221%
EMR	0.000000000000000152674459227347%
EOG	0.000000000000000533872440097732%
EQIX	0.00000224935876819509%
EQR	0.0000248876773559999%
EQT	0.00000069819906173266%
ES	0.000000115812679355857%
ESRX	0%
ESS	0.000000000000000100096466540419%
ETFC	0%

## 5. Conclusões

Este artigo apresentou uma pesquisa sobre a construção de um portfólio futuro usando uma rede neural profunda denominada LSTM para prever os valores futuros de ações, e um algoritmo evolutivo multiobjetivo chamado MOEA/D para sugerir portfólios ótimos com base nas previsões realizadas. Os resultados usando as bases de dados dos índices S&P 500 e Dow Jones mostrou que a rede LSTM alcançou previsões com baixo erro. E em seguida o MOEA/D apresentou fronteiras de Pareto contendo portfólios que poderiam ter gerado retornos entre 14% e 23% para o S&P e entre 4% e 10% para o Dow Jones. Trabalhos futuros incluem: (i) a aplicação dos algoritmos no índice Ibovespa; (ii) teste de outros algoritmos evolutivos multiobjetivos como o NSGA-II e SPEA2; (iii) teste de outras redes neurais profunda; e (iv) introdução de restrições ao problema, como por exemplo, restrições de orçamento.



## Referências

- Arévalo, A., Nino, J., Hernandez, G., e Sandoval, J. (2016). High-frequency trading strategy based on deep neural networks. volume 9773, pages 424–436.
- Björklund, S. e Uhlin, T. (2017). Artificial neural networks for financial time series prediction and portfolio optimization. Master's thesis, Department of Management and Engineering, Linköping University, Linköping, Sweden.
- Bovespa (2000). O mercado de capitais: Sua importância para o desenvolvimento e os entraves com que se defronta no Brasil. *Tendência - Consultoria Integrada*.
- Campelo, F., Batista, L., e Aranha, C. (2018). The MOEADr package - a component-based framework for multiobjective evolutionary algorithms based on decomposition. *Journal of Statistical Software*.
- Chan, L., Karceski, J., e Lakonishok, J. (1999). On portfolio optimization: Forecasting covariances and choosing the risk model. *Review of Financial Studies*, 12:937–74.
- Chen, T.-L. e Chen, F.-Y. (2016). An intelligent pattern recognition model for supporting investment decisions in stock market. *Information Sciences*, 346-347:261 – 274.
- Chiang, W.-C., Enke, D., Wu, T., e Wang, R. (2016). An adaptive stock index trading decision support system. *Expert Systems with Applications*, 59:195 – 207.
- Chong, E., Han, C., e Park, F. C. (2017). Deep learning networks for stock market analysis and prediction. *Expert Syst. Appl.*, 83(C):187–205.
- Chourmouziadis, K. e Chatzoglou, P. D. (2016). An intelligent short term stock trading fuzzy system for assisting investors in portfolio management. *Expert Systems with Applications*, 43:298 – 311.
- Deb, K., Pratap, A., Agarwal, S., e Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.
- Duffie, D. e Pan, J. (1997). An overview of value at risk. *The Journal of Derivatives*, 4(3):7–49.
- Economática (2015). Aumento da taxa dos pequenos investidos que entraram nas negociações da bolsa.
- Enke, D. e Mehdiyev, N. (2013). Stock market prediction using a combination of stepwise regression analysis, differential evolution-based fuzzy clustering, and a fuzzy inference neural network. *Intelligent Automation and Soft Computing*, 19.
- Harper, D. R. (2019). An introduction to value at risk (var). <https://www.investopedia.com/articles/04/092904.asp>.
- Heaton, J. B., Polson, N. G., e Witte, J. H. (2016). Deep learning for finance: deep portfolios: J. b. heaton, n. g. polson and j. h. witte. *Applied Stochastic Models in Business and Industry*, 33.
- Hochreiter, S. e Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Krokhmal, P., Palmquist, J., e Uryasev, S. (2003). Portfolio optimization with conditional value-at-risk objective and constraints. *Journal of Risk*, 4.

- Morettin, P. e Toloi, C. M. C. (2011). *Análise de séries temporais*. Edgard Blucher, ed Brasil, 3 edition.
- Niaki, S. T. A. e Hoseinzade, S. (2013). Forecasting s&p 500 index using artificial neural networks and design of experiments. *Journal of Industrial Engineering International*, 9.
- Nugent, C. (2018). S&p 500 stock data.
- Patel, J., Shah, S., Thakkar, P., e Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications*, 42(4):2162 – 2172.
- Patterson, J. e Gibson, A. (2017). *Deep Learning A Practitioner's Approach*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
- Pereira, I. L., Crispim, S. F., e Savóia, J. R. F. (2013). Valor de mercado como indicador de competitividade: identificação de empresas brasileiras com alto desempenho no longo prazo. *Revista das Faculdades Integradas Vianna Júnior*.
- Qiu, M., Song, Y., e Akagi, F. (2016). Application of artificial neural network for the prediction of stock market returns: The case of the japanese stock market. *Chaos, Solitons & Fractals*, 85:1 – 7.
- Sak, H., Senior, A., e Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 338–342.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., e Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.
- Weng, B. (2017). *Application of machine learning techniques for stock market prediction*. PhD thesis, Graduate Faculty of Auburn University, Auburn, Alabama.
- Xiao, J., Li, J.-J., Hong, X.-X., Huang, M.-M., Hu, X.-M., Tang, Y., e Huang, C.-Q. (2018). An improved moea/d based on reference distance for software project portfolio optimization. *Complexity*, 2018:1–16.
- Zhang, Q. e Li, H. (2008). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, 11:712 – 731.
- Zhong, X. e Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications*, 67:126 – 139.