

PREVISÃO DE AÇÕES COM MODELO HÍBRIDO LSTM E ALGORITMO GENÉTICO

Henrique Honorato Cruvinel ¹

¹Pontifícia Universidade Católica de Goiás, Goiânia, Goiás Brasil, henriquehebert008@gmail.com

RESUMO: Este artigo investigou o desenvolvimento de um modelo híbrido utilizando redes neurais LSTM em conjunto com algoritmo genético para previsão de preços de ações na bolsa de valores brasileira, focando nos dados da Petrobras. O estudo abordou a teoria do "dilema do caminho aleatório", aprimorou técnicas de pré-processamento de dados e utilizou métricas como RMSE, MSE, R^2 e MAPE para avaliar o desempenho do modelo. Os resultados demonstraram uma capacidade significativa de previsão, aproximando-se dos valores reais, e discutiram a integração do modelo com o MetaTrader 5 para automatização de decisões de investimento. Apesar dos desafios como a volatilidade do mercado, o estudo ofereceu contribuições importantes para melhorar a precisão das previsões financeiras e sugere possibilidades para futuras pesquisas na área.

Palavras chaves: *Rede Neural, Algoritmo Genético, Previsão de Ações, Algoritmo Híbrido, Previsão Financeira.*

ABSTRACT: This article investigated the development of a hybrid model using LSTM neural networks in conjunction with genetic algorithms for predicting stock prices on the Brazilian stock exchange, focusing on data from Petrobras. The study addressed the theory of the "random walk dilemma," enhanced data preprocessing techniques, and used metrics such as RMSE, MSE, R^2 , and MAPE to evaluate model performance. Results demonstrated significant predictive capability, approaching real values, and discussed the integration of the model with MetaTrader 5 for automated investment decisions. Despite challenges such as market volatility, the study provided valuable contributions to improving financial prediction accuracy and suggests avenues for future research in the field.

Keywords: *Neural Network, Genetic Algorithm, Stock Prediction, Hybrid Algorithm, Financial Forecasting.*

1. Introdução

O interesse por investimentos na bolsa de valores aumentou significativamente no Brasil devido à consolidação e segurança crescente do mercado financeiro. Em janeiro de 2022, a B3 (Brasil Bolsa Balcão) registrou um marco significativo com cinco milhões de contas de investidores ativos em renda variável (B3, 2022). Esse crescimento reflete um ambiente financeiro mais robusto e acessível para investidores individuais.

Neste contexto de expansão do mercado financeiro brasileiro, surge a necessidade de desenvolver métodos avançados para prever os preços das ações. Este estudo propõe a aplicação de um modelo híbrido que combina redes neurais artificiais (RNA), especificamente do tipo Long Short-Term Memory (LSTM), com algoritmo genético (AG). A escolha da LSTM é motivada por sua capacidade de lidar com séries temporais e capturar dependências de longo prazo, fundamentais para análises financeiras. O AG, por sua vez, será empregado para otimizar os hiperparâmetros da LSTM, visando maximizar o desempenho do modelo. Além disso, com base nas previsões geradas pelo treinamento da LSTM, será realizada automaticamente a compra de ações.

2. Referencial teórico

A previsão de movimentos no mercado financeiro enfrenta o desafio do "dilema do caminho aleatório" (random walk dilemma, RWD), conforme discutido por Fama e French (1988). A teoria sugere que os preços das ações seguem um padrão aleatório, o que limita a capacidade de previsão com base em dados históricos.

Segundo Vasco (2020), séries temporais são sequências ordenadas de valores ao longo do tempo, essenciais para previsões futuras, onde cada valor é influenciado pelos anteriores. O aprendizado de máquina utiliza redes neurais artificiais, como a LSTM, que são projetadas para lidar com sequências e memória de longo prazo (Hochreiter; Schmidhuber, 1997), sendo adaptadas para previsão de séries temporais financeiras.

A figura 1 mostra o funcionamento da arquitetura LSTM.

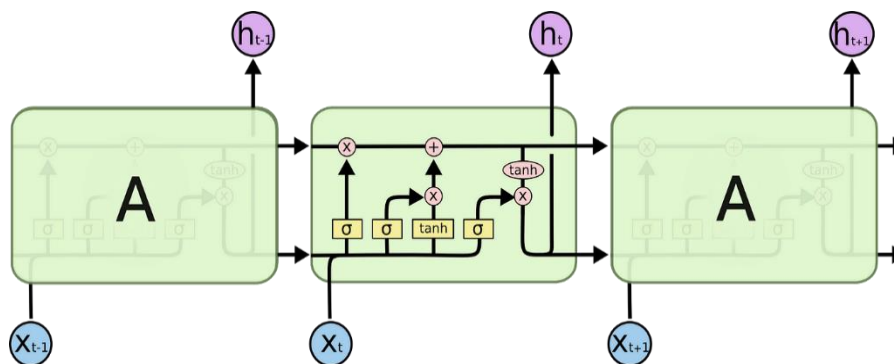


Figura 1: Arquitetura LSTM.

O algoritmo genético é uma das meta-heurísticas comumente utilizadas para otimização em problemas complexos, incluindo o aprendizado de máquinas (KURI e GALAVIZ, 2002). Neste projeto científico, pretende-se utilizar o algoritmo genético como otimizador para ajustar os pesos das camadas da rede neural. O desempenho de cada indivíduo será avaliado por meio da Raiz Quadrada do Erro Médio (Root Mean Squared Error, MSE), uma função de avaliação amplamente utilizada em tarefas de regressão. Abaixo encontra-se a formulação matemática do RMSE.

A qualidade dos dados e seu pré-processamento são fundamentais para o sucesso do modelo de previsão. Problemas como valores nulos, outliers e sazonalidades devem ser tratados adequadamente para evitar distorções nos resultados (Batista, 2003).

O software selecionado para a implementação da estratégia foi o Metatrader 5, uma plataforma de negociação eletrônica desenvolvida pela MetaQuotes Software. O Metatrader 5 é licenciado para corretoras, permitindo o envio de ordens de compra e venda, bem como a abertura e fechamento de posições na bolsa de valores brasileira B3. O Metatrader conta com uma linguagem de programação própria chamada MQL5, que é utilizada para desenvolver programas executados diretamente na plataforma Metatrader 5. No entanto, no algoritmo, será utilizado o Python para fazer a integração e acionar o Metatrader, permitindo que o software execute ordens de compra e venda com base na análise da rede neural (MQL5 Reference, 2022).

O objetivo deste projeto é desenvolver um algoritmo capaz de prever os preços dos ativos da bolsa de valores brasileira utilizando redes neurais artificiais em conjunto com algoritmos genéticos, visando melhorar a precisão das previsões financeiras.

3. Materiais e Métodos

O ativo da empresa Petrobras (Petróleo Brasileiro S.A) foi obtido através da plataforma de dados financeiros Yahoo Finance utilizando a biblioteca `yfinance` no ambiente de programação Python. Para o desenvolvimento do modelo de rede neural, definiu-se um intervalo temporal para o treinamento, compreendido entre as datas "2017-12-31" e "2023-07-21", em frequência diária.

A priori foram removidos os valores nulos e outliers da série temporal. Em seguida, verificou-se a normalidade da série pelo teste de Shapiro-Wilk e concluiu-se que a série não apresenta distribuição normal, além disso, foi constatado, por meio do Teste KPSS, que a série em análise não passa no teste de estacionariedade, sendo este um indicador apropriado para a determinação do caráter estacionário da série (MIOT, 2018).

Após a obtenção dos dados, realizou-se a preparação dos mesmos para o treinamento do modelo, onde foi selecionada a coluna correspondente aos preços de fechamento do ativo Petrobras. Esses preços de fechamento foram utilizados como variáveis dependentes para o treinamento da rede neural LSTM.

Com o objetivo de garantir uma adequada generalização do modelo, a divisão dos dados foi efetuada de forma a utilizar 80% do conjunto de preços de fechamento para o treinamento propriamente dito, enquanto os 20% restantes foram destinados para a etapa de testes.

Posteriormente, os conjuntos de dados de treinamento e teste foram submetidos ao processo de escalonamento utilizando o método Min-Max Scaler. Essa técnica tem como objetivo ajustar os valores dos dados para um intervalo entre zero e um, proporcionando uma melhor normalização dos dados (PEDREGOSA et al., 2011). Essa normalização é fundamental, uma vez que os dados originais podem apresentar comportamentos variados, os quais poderiam prejudicar o treinamento eficaz da rede neural.

O método de escalonamento Min-Max (Min-Max Scaler) é calculado pela fórmula apresentada abaixo, onde X representa o valor original que está sendo escalonado, X_{min} é o valor mínimo encontrado no conjunto de dados original, e X_{max} é o valor máximo encontrado no conjunto de dados original. O resultado *scaled* é o valor escalonado após a aplicação deste método, ajustando X para uma escala entre 0 e 1 com base nos limites mínimo e máximo do conjunto de dados. A fórmula do Min-Max Scaler é dada pela Eq. (1)

$$scaled = \frac{X_{max} - X_{min}}{X - X_{min}} \quad (1)$$

Com os conjuntos de dados agora separados e devidamente escalonados, procedeu-se à determinação do horizonte de previsão. Neste contexto, a LSTM (Long Short-Term Memory), uma rede neural capaz de armazenar informações de memória, foi utilizada para prever o preço da ação da Petrobras para o dia seguinte, baseado no horizonte de previsão determinado. Para essa predição, foi estabelecido um horizonte de sessenta dias, isto é, a rede neural vai prever os valores do dia seguinte baseado nos sessenta dias anteriores.

Na modelagem do Algoritmo Genético (AG), foram definidos os hiperparâmetros da rede neural LSTM como os "genes" a serem atualizados durante as gerações das populações. Esses hiperparâmetros incluem o número de unidades ocultas (hidden units), a densidade (dense) da rede, a taxa de aprendizagem (learning rate) e o número de épocas de treinamento (number epochs).

O algoritmo genético é utilizado para otimizar esses hiperparâmetros, buscando encontrar as melhores combinações que resultem em um desempenho ótimo da rede neural LSTM. A cada geração, o algoritmo genético realiza cruzamentos, mutações e seleção dos indivíduos mais aptos, permitindo uma exploração eficiente do espaço de busca de hiperparâmetros e convergência para soluções mais promissoras.

O treinamento da rede neural sem o auxílio do algoritmo genético pode levar a resultados variados, que vão desde ótimos desempenhos até resultados insatisfatórios. Através desse processo iterativo, o algoritmo genético busca melhorar a performance da rede neural LSTM, adaptando-se às características específicas dos dados e do problema em análise, com o objetivo de obter resultados mais precisos e confiáveis em relação às previsões da ação da Petrobras. A modelagem estabelecida para a rede neural pode ser observada usando código python apresentado na Figura 2.

A figura 2 ilustra a modelagem da rede neural LSTM

```

# Definir os parâmetros da rede com base no cromossomo
hidden_units = int(chromosome[0])
dense = chromosome[1]
learning_rate = chromosome[2]
num_epochs = int(chromosome[3])

print(hidden_units, dense, learning_rate, num_epochs)

# Criar o modelo da rede
model = Sequential()
model.add(LSTM(hidden_units, return_sequences = True, input_shape = (treinamento_x.shape[1],1)))
model.add(LSTM(hidden_units, return_sequences = False))
model.add(Dense(dense))
model.add(Dense(1))

# Treinar o modelo
model.compile(loss='mean_squared_error', optimizer=AdamLegacy(learning_rate=learning_rate))
model.fit(treinamento_x, treinamento_y, batch_size=1, epochs = 1)

```

Figura 2: Código implementado do LSTM.

A função de evolução do algoritmo genético retorna o *Root Mean Square Error* (RMSE) do treinamento da rede neural. Consequentemente, os melhores resultados são classificados no topo, uma vez que a variável "melhor fitness" armazena sempre o menor valor de RMSE encontrado. Os valores dos parâmetros do algoritmo genético são apresentados no **Quadro 1** a seguir:

Quadro 1: Parâmetros do AG.

Parâmetros	Valores
Número de genes	4
Tamanho da população	4
Taxa de mutação	0,05
Número de gerações	4

Os genes são definidos aleatoriamente no intervalo estabelecido no **Quadro 2**.

Quadro 2: Informações nos genes.

Genes	Intervalo
Unidades ocultas	(45, 60)
Densidade	(25,50)
Taxa de aprendizagem	(0,001; 0,01)
Número de épocas	(10,100)

Após a aplicação do algoritmo genético teve-se o resultado do melhor indivíduo encontrado com os seguintes hiperparâmetros da rede neural mostrados na **Quadro 3**.

Quadro 3: Hiperparâmetros da rede LSTM

Hiperparâmetros	Valores
Unidades ocultas	56
Densidade	29
Taxa de aprendizagem	0,007765305944367126
Número de épocas	80

Dessa forma, a rede neural LSTM foi treinada a partir desses valores para obter o melhor desempenho para avaliar as predições entre os valores reais e os valores obtidos.

A métrica de erro quadrático médio (RMSE) é definida pela Equação (2), em que n representa o número total de observações no conjunto de dados, y'_i é o valor previsto pelo modelo para a i -ésima observação e y_i é o valor real da i -ésima observação.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y'_i - y_i)^2} \quad (2)$$

A métrica de erro quadrático médio (MSE) é dada pela Equação (3), onde n é o número total de observações no conjunto de dados, y'_i representa o valor previsto pelo modelo para a i -ésima observação e y_i é o valor real da i -ésima observação.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y'_i - y_i)^2 \quad (3)$$

O coeficiente de determinação (R^2) é definido pela Equação (4), onde y_i representa o valor real da i -ésima observação, y'_i é o valor previsto pelo modelo para a i -ésima observação, y''_i é a média dos valores reais do conjunto de dados e n é o número total de observações no conjunto de dados.

$$R^2 = \frac{\sum_{i=1}^n (y_i - y'_i)^2}{\sum_{i=1}^n (y_i - y''_i)^2} \quad (4)$$

A métrica de erro percentual absoluto médio (MAPE) é definida pela Equação (5), onde n representa o número total de observações no conjunto de dados, y_i é o valor real da i -ésima observação e y'_i é o valor previsto pelo modelo para a i -ésima observação.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - y'_i}{y_i} \right| \times 100 \quad (5)$$

4. Resultados e Discussão

A Figura 3 mostra os dados do ativo na fase de treino e os valores utilizados e obtidos na fase de teste.

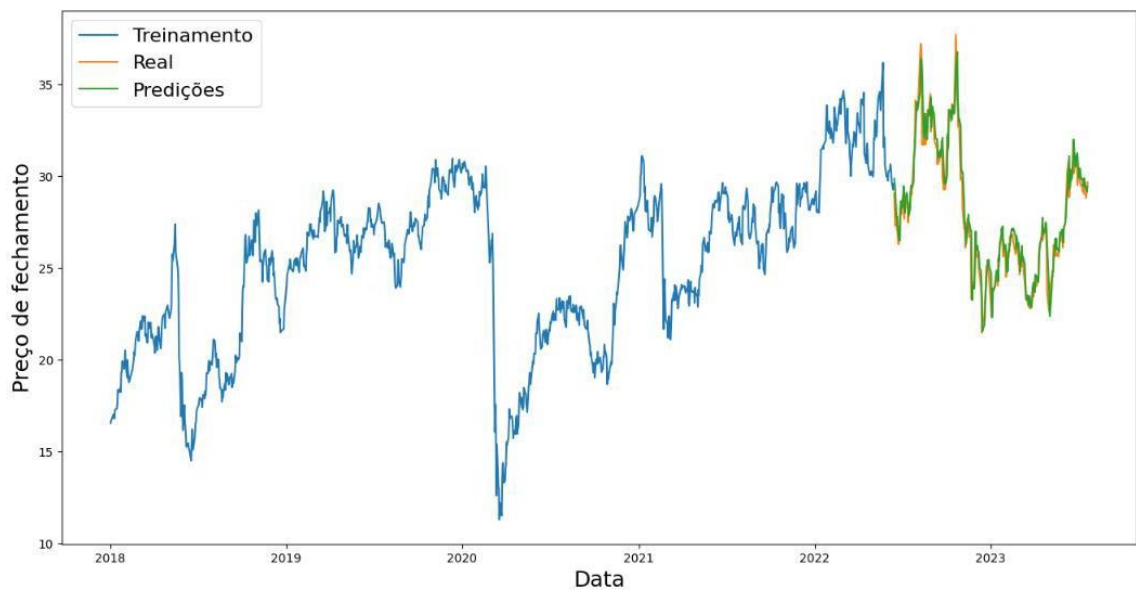


FIGURA 3: comparação dos valores reais e dos valores obtidos no teste.

No **Quadro 4** são apresentadas as métricas da fase de teste.

Quadro 4: Métricas obtidas na fase de teste.

Métrica	valor
RMSE	0,8527745215627146
MSE	0,7272243846265168

R^2	0,9416870804169537
MAPE	2,23%

Em seguida, são definidos os dados dos últimos sessenta dias do ativo para tentar prever o preço do dia subsequente. Como o modelo foi treinado até o dia 21/07/2023, ele será utilizado para prever o preço de fechamento do dia 22/07/2023. A seguir, os resultados do teste no **Quadro 5**:

Quadro 5: Resultados do teste

Dia	Fechamento real	Fechamento previsto
22/07/2032	29,68	29,96

O modelo apresentou um desempenho consideravelmente sólido, aproximando-se significativamente do valor real de fechamento. É importante ressaltar que, no contexto de previsões financeiras, é comum encontrar certa margem de erro, pois os mercados podem ser influenciados por uma série de fatores imprevisíveis.

Ainda assim, o resultado obtido demonstra uma eficiência promissora do modelo híbrido desenvolvido, sugerindo que sua capacidade de acompanhar o comportamento do ativo da Petrobras com os dados de fechamento treinados é notável.

Portanto, embora o modelo tenha apresentado um pequeno desvio em sua previsão, sua capacidade de se aproximar significativamente do valor real é encorajadora. A pesquisa e o desenvolvimento contínuos nesse campo são cruciais para aprimorar ainda mais as previsões financeiras e contribuir para uma tomada de decisão mais informada e eficiente no mercado de capitais.

No presente estudo, avaliamos a capacidade preditiva do modelo em relação aos valores reais. Entretanto, para uma análise abrangente, é essencial considerar o comportamento das previsões realizadas com base nos dados de treinamento. Dessa forma, buscamos determinar a percentagem de ganho sobre perda do ativo e a proporção de acertos em relação à direção da ação (ou seja, se a ação teve um movimento ascendente ou descendente).

A avaliação desses parâmetros permite uma compreensão mais completa da eficácia do modelo proposto e auxilia na tomada de decisões informadas em relação à utilização dessas previsões no mercado financeiro. É essencial destacar que as conclusões deste estudo oferecem informações valiosas que podem contribuir significativamente para aperfeiçoar e aprimorar a

precisão das previsões feitas pelo modelo em análise.

Em seguida, foi realizada a integração do MetaTrader no algoritmo para automatizar a compra da ação baseada na regra de negócio estabelecida. Para este teste simples, definiu-se que a compra deve ser efetuada quando a previsão de fechamento for maior do que o preço de abertura.

A Figura 4 mostra a condição para efetuar a compra. Se a previsão for maior do que o preço de abertura, chama a função de compra:

```
|  
if previsao_de_preco > preço_abertura:  
    abrir_compra()  
else:  
    print("A previsão não indica uma compra.")
```

Figura 4: Condição para efetuar a compra da ação.

A Figura 5 ilustra a função de abrir compra da ação:

```
# Função para abrir uma ordem de compra  
def abrir_compra():  
    price = mt5.symbol_info_tick(symbol).ask  
    request = {  
        "action": mt5.TRADE_ACTION_DEAL,  
        "symbol": symbol,  
        "volume": lot_size,  
        "type": mt5.ORDER_TYPE_BUY,  
        "type_filling": mt5.ORDER_FILLING_FOK,  
        "price": price,  
        "magic": 234567,  
        "type_time": mt5.ORDER_TIME_GTC,  
        "deviation": 20,  
        "comment": "Ordem de Compra de Teste",  
    }  
  
    result = mt5.order_send(request)  
    if result.retcode == mt5.TRADE_RETCODE_DONE:  
        print(f"Ordem de compra aberta com sucesso! ID da ordem: {result.order}")  
    else:  
        print(f"Erro ao abrir ordem de compra: {result.retcode} ({mt5.last_error()})")
```

Figura 5: Algoritmo da ordem de compra.

Apesar dos resultados promissores obtidos com o modelo híbrido proposto, ainda existem desafios a serem enfrentados para aprimorar sua acurácia. Um ponto crucial a ser abordado é a busca por maneiras de aumentar a robustez do modelo em relação a diferentes cenários de mercado e condições econômicas. A volatilidade e imprevisibilidade do mercado financeiro

podem impactar significativamente as previsões, tornando essencial o desenvolvimento de estratégias adaptativas que levem em consideração uma variedade de fatores externos, como notícias econômicas, eventos geopolíticos e tendências macroeconômicas. Além disso, investigar a sensibilidade do modelo a diferentes intervalos de treinamento e teste pode ajudar a determinar sua capacidade de generalização ao longo do tempo, permitindo uma avaliação mais completa de sua performance em diferentes contextos de mercado.

No tocante à diversificação das abordagens, é recomendável que o projeto explore a aplicação de outros modelos de negociação de ações em conjunto com as redes neurais LSTM. A diversificação das técnicas pode proporcionar uma compreensão mais holística do comportamento do mercado e das diferentes variáveis que influenciam os preços das ações. Ao considerar múltiplas abordagens, o projeto pode explorar sinergias e combinações que resultem em previsões mais precisas e resilientes, oferecendo aos investidores uma visão mais completa para embasar suas decisões de investimento. Dessa forma, a exploração de diferentes modelos de negociação de ações pode enriquecer o projeto, contribuindo para um entendimento mais aprofundado dos desafios e oportunidades no campo da previsão financeira.

5. Conclusão

Em suma, este projeto científico teve como objetivo desenvolver um modelo híbrido utilizando redes neurais LSTM (Long Short-Term Memory) em conjunto com o algoritmo genético para prever os preços de ativos na bolsa de valores. O modelo foi aplicado especificamente aos preços de fechamento das ações da empresa Petrobras (Petróleo Brasileiro S.A.).

O modelo híbrido obteve resultados promissores, demonstrando um bom acompanhamento do comportamento do ativo da Petrobras durante o período de treinamento. Ao prever o preço de fechamento para o dia seguinte (22/07/2032), o modelo apresentou um valor próximo ao real (29,96 previsto versus 29,68 real), mostrando sua capacidade de se aproximar significativamente do valor verdadeiro.

Como conclusão geral, este estudo oferece informações valiosas para aperfeiçoar e aprimorar a precisão das previsões financeiras utilizando modelos híbridos baseados em redes neurais LSTM e algoritmos genéticos. A continuidade da pesquisa nesse campo é essencial para melhorar ainda mais as previsões e contribuir para uma tomada de decisão mais eficiente no mercado de capitais.

6. Referências Bibliográficas

B3. B3 atinge 5 milhões de contas de investidores em renda variável em janeiro. Disponível em: https://www.b3.com.br/pt_br/noticias/5-milhoes-de-contas-de-investidores.htm. Acesso em: 25 mar. 2022.

BATISTA, G. E. d. A. P. A. Pré-processamento de dados em aprendizado de máquina supervisionado. Tese de doutorado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, 2003.

FAMA, E. F.; FRENCH, K. R. Permanent and temporary components of stock prices. In: *Journal of Political Economics*, v. 96, n. 2, 1988, pp. 246–273. doi: 10.1086/261535.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long Short-Term Memory. In: *Neural Comput*, v. 9, n. 8, 1997, pp. 1735–1780. doi: 10.1162/neco.1997.9.8.1735.

JUNIOR, José R. F. Redes Neurais Recorrentes — LSTM. Disponível em: <https://medium.com/@web2ajax/redes-neurais-recorrentes-lstm-b90b720dc3f6>. Acesso em: 23 jul. 2023.

KURI, M. Á.; GALAVIZ, C. J. Algoritmos Genéticos. México: Instituto Politécnico Nacional, Universidad Nacional Autónoma de México, Fondo de Cultura Económica, 2002. 60 p. ISBN: 9681663837.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. In: *Nature*, v. 521, 2015, pp. 436–444. doi: 10.1038/nature14539.

MIOT, Hélio Amante. Correlation analysis in clinical and experimental studies. *Jornal Vascular Brasileiro*, v. 17, n. 4, p. 275-279, 2018.

MQL5 Reference - How to use algorithmic/automated trading language for MetaTrader 5. Disponível em: . Acesso em: 30 nov. 2022.

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

VASCO, Lucas Pimenta. Um estudo de redes neurais recorrentes no contexto de previsões no mercado financeiro. 2020. 114 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de São Carlos, São Carlos, 2020. Disponível em: <https://repositorio.ufscar.br/handle/ufscar/13730>. Acesso em: 31 mar. 2023.