



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

ÍTALO MIGUEL CASTOR DINIZ PINHEIRO

**RECOMENDAÇÕES PARA COMPRA E VENDA DE ATIVOS EM DAY
TRADING UTILIZANDO REDES NEURAIAS ARTIFICIAIS**

CAMPINA GRANDE - PB

2023

ÍTALO MIGUEL CASTOR DINIZ PINHEIRO

**RECOMENDAÇÕES PARA COMPRA E VENDA DE ATIVOS EM
DAY TRADING UTILIZANDO REDES NEURAS ARTIFICIAIS**

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador : Herman Martins Gomes

CAMPINA GRANDE - PB

2023

ÍTALO MIGUEL CASTOR DINIZ PINHEIRO

**RECOMENDAÇÕES PARA COMPRA E VENDA DE ATIVOS EM
DAY TRADING UTILIZANDO REDES NEURAS ARTIFICIAIS**

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Herman Martins Gomes

Orientador – UASC/CEEI/UFCG

Maxwell Guimarães de Oliveira

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 17 de NOVEMBRO de 2023.

CAMPINA GRANDE - PB

RESUMO

É notório o crescimento do ingresso de investidores individuais no mercado financeiro. Muitos desses investidores não conseguem discernir corretamente onde devem investir o seu dinheiro para obter uma maior rentabilidade. Neste contexto, este trabalho teve como objetivo desenvolver uma aplicação para auxiliar nas decisões de compra e venda de ativos por meio de uma rede neural treinada de forma supervisionada sobre dados extraídos das cotações dos ativos. A aplicação desenvolvida busca apresentar uma forma intuitiva para guiar as decisões diárias de investimento (day trading) no tocante às ações que devem permanecer, entrar ou sair da carteira de cada usuário. O princípio é prever a cotação do dia seguinte da ação, com recomendação de venda se o preço previsto for diminuir e de compra se o preço for aumentar. A aplicação informa também uma estimativa sobre quanto cada operação de compra ou venda produzirá de lucro. Para isso, foram feitos experimentos com três ativos diferentes que atendiam as principais movimentações do mercado e foi possível notar um acréscimo no valor inicial investido pelo usuário de aproximadamente 23% a 43% no montante final. A ferramenta desenvolvida poderá auxiliar tanto usuários iniciantes no mercado de ações quanto usuários mais experientes.

RECOMMENDATIONS FOR BUYING AND SELLING ASSETS IN DAY TRADING USING ARTIFICIAL NEURAL NETWORKS

ABSTRACT

It is notable the increase in the participation of individual investors in the financial market. Many of these investors struggle to accurately discern where to invest their money for higher returns. In this context, the aim of this work was to develop an application to assist in the buying and selling decisions of assets through a supervised neural network trained on data extracted from asset quotations. The developed application seeks to provide an intuitive way to guide daily investment decisions (day trading) regarding which stocks to hold, enter, or exit from each user's portfolio. The principle is to predict the price of the stock for the next day, with a recommendation to sell if the forecasted price is decreasing and to buy if the price is expected to increase. The application also provides an estimate of the profit generated by each buy or sell operation. To achieve this, experiments were conducted with three different assets that matched the major market movements, and it was possible to observe an increase in the initial investment by the user of approximately 23% to 43% in the final amount. The developed tool can assist both novice and experienced users in the stock market.

Recomendações para compra e venda de ativos em day trading utilizando redes neurais artificiais

Pinheiro C. D., Italo Miguel
Universidade Federal de Campina (UFCG)
Campina Grande, Paraíba
Brasil

italo.pinheiro@ccc.ufcg.edu.br

M. Gomes, Herman
Universidade Federal de Campina (UFCG)
Campina Grande, Paraíba
Brasil

hmg@computacao.ufcg.edu.br

ABSTRACT

It is notable the increase in the participation of individual investors in the financial market. Many of these investors struggle to accurately discern where to invest their money for higher returns. In this context, the aim of this work was to develop an application to assist in the buying and selling decisions of assets through a supervised neural network trained on data extracted from asset quotations. The developed application seeks to provide an intuitive way to guide daily investment decisions (day trading) regarding which stocks to hold, enter, or exit from each user's portfolio. The principle is to predict the price of the stock for the next day, with a recommendation to sell if the forecasted price is decreasing and to buy if the price is expected to increase. The application also provides an estimate of the profit generated by each buy or sell operation. To achieve this, experiments were conducted with three different assets that matched the major market movements, and it was possible to observe an increase in the initial investment by the user of approximately 23% to 43% in the final amount. The developed tool can assist both novice and experienced users in the stock market.

RESUMO

É notório o crescimento do ingresso de investidores individuais no mercado financeiro. Muitos desses investidores não conseguem discernir corretamente onde devem investir o seu dinheiro para obter uma maior rentabilidade. Neste contexto, este trabalho teve como objetivo desenvolver uma aplicação para auxiliar nas decisões de compra e venda de ativos por meio de uma rede neural treinada de forma supervisionada sobre dados extraídos das cotações dos ativos. A aplicação desenvolvida busca apresentar uma forma intuitiva para guiar as decisões diárias de investimento (*day trading*) no tocante às ações que devem permanecer, entrar ou sair da carteira de cada usuário. O princípio é prever a cotação do dia seguinte da ação, com recomendação de venda se o preço previsto for diminuir e de compra se o preço for aumentar. A aplicação informa também uma estimativa sobre quanto cada operação de compra ou venda produzirá de lucro. Para isso, foram feitos experimentos com três ativos diferentes que atendiam as principais movimentações do mercado e foi possível notar um acréscimo no valor inicial investido pelo usuário de aproximadamente 23% a 43% no montante final. A ferramenta desenvolvida poderá auxiliar tanto usuários iniciantes no mercado de ações quanto usuários mais experientes.

KEYWORDS

Rede neural, mercado financeiro, cotações dos ativos.

1. INTRODUÇÃO

Define-se por mercado financeiro o ambiente no qual é possível realizar negociações de ativos sem grandes intervenções externas, de modo que os participantes possam adquirir recursos e capitais, transferir títulos e comercializar moedas [1]. Uma das instituições que permite esse processo de comercialização é a Bolsa de Valores, a qual atua primordialmente facilitando a compra e venda de títulos emitidos por empresas de capital aberto e outros valores mobiliários, tais como ações e fundos imobiliários. É nesse contexto que surge a seguinte questão: Por que as pessoas optam por investir na Bolsa?

Uma pesquisa feita no site Reclame Aqui com 10.095 usuários em julho de 2021 indicou que cerca de 40% dos ouvidos investem para ter uma estabilidade financeira. Além disso, pode-se citar também o desejo por parte dos investidores mais jovens, que nesta pesquisa abrange cerca de 41,7% da amostra, de atingirem independência financeira [2]. Sendo assim, de modo geral, percebe-se que, ao investir em renda variável na Bolsa, as pessoas buscam o aumento do seu patrimônio.

Diante desse cenário atrativo, notou-se um aumento exorbitante do número de pessoas físicas entrando no mundo de investimentos de renda variável. Um levantamento feito pela Bolsa de Valores de São Paulo, a B3, indicou que o número de investidores na B3 cresceu cerca de 35% entre 2021 e 2022. Ao comparar com os dados de 2019, esse número cresceu aproximadamente 82% [3].

Nesse âmbito, um problema surge nesse cenário, pois ao passo que o número de pessoas cresce investindo na bolsa, paralelamente, cresce também dois grupos de investidores. O primeiro grupo compreende aqueles que querem entrar na bolsa de valores mas não sabem em qual ativo investir [4]. Pesquisas feitas pela Capital Research com mais de 43 mil usuários em 2020, indicam que cerca de 63% dos brasileiros não investem por falta de conhecimento na área [5]. Por fim, o segundo grupo compreende aqueles que entram na bolsa, mesmo sem saber, e acabam perdendo dinheiro por investir em um ativo “errado” devido à falta da análise correta.

Pensando nisso, a fim de auxiliar esses pequenos investidores nesse aspecto, este trabalho tem o objetivo de desenvolver uma aplicação de suporte à tomada de decisão que

indique o momento certo de comprar ou vender um ativo com base em análises diárias. Nota-se que no mercado existem diversas aplicações voltadas para um gerenciamento dos ativos que o investidor possui e até mesmo trazendo gráficos de altas e baixas desses. Todavia, para aquele que está aprendendo a investir e que ainda não tem suficiente maturidade no assunto para entender o que todos aqueles gráficos e números realmente significam, não existe uma aplicação que lhe dê as direções necessárias para que ele possa ver o seu dinheiro aumentando. Sendo assim, a aplicação AssetRight é uma proposta nova que irá disponibilizar na sua primeira versão para o usuário uma recomendação de compra ou venda de uma ou mais cotações de um determinado ativo em conjunto com o descritivo da lucratividade que este terá ao seguir as instruções dadas pelo modelo.

A predição é realizada por meio de um modelo de rede neural treinado de forma supervisionada com dados extraídos do site Yahoo Finance. Este, por sua vez, disponibiliza todas as informações necessárias para que possa ser feito uma análise gráfica detalhada das cotações dos ativos desde o seu surgimento até o dado momento. A análise gráfica ou técnica, como é chamada no mercado, é uma das estratégias de escolha de ativos mais utilizada por grandes investidores e foi popularizada por Charles Dow, fundador do The Wall Street Journal, um dos jornais mais respeitáveis de publicações sobre economia [6]. Neste cenário, surge mais uma pergunta: O pequeno investidor saberá fazer análises técnicas? Em sua grande maioria não. É justamente nesse ponto que entra a rede neural treinada com os dados do site supracitado, uma vez que, o usuário apenas precisará pesquisar o ativo que ele deseja na aplicação e o modelo irá prever se o ativo terá sua cotação oscilando para cima ou para baixo.

2. FUNDAMENTAÇÃO

2.1 Trabalhos Relacionados

Existem muitos trabalhos que envolvem a predição de ativos utilizando redes neurais sejam elas recorrentes ou artificiais. Em geral, redes neurais generalizam bem devido à sua capacidade de aprender padrões nos dados e também por terem uma grande flexibilidade de aprendizagem em relações complexas entre entrada e saída esperada. Tendo em vista esses aspectos, redes neurais têm uma alta capacidade de adaptação para diferentes ativos e espaços de tempo.

Uma das redes neurais mais utilizadas para previsão dos preços dos ativos é a LSTM ou *Long Short-Term*, pois esta pode armazenar informações passadas que irão desempenhar um papel significativo na previsão do preço futuro de uma determinada ação [7]. Esta rede neural foi utilizada por Moghar e Hamiche [8] para prever justamente os preços de fechamento, que é a proposta desse trabalho, de ações da Google e Nike. Eles perceberam que os resultados de testes não estavam equivalentes às movimentações originais, porém, que as tendências puderam ser seguidas e que foi possível mapear os preços de abertura em ambos os ativos.

Outro trabalho, conduzido por Xavier [9], fez uso da média móvel para tentar prever o movimento dos preços das ações de quatro empresas diferentes não citadas. Os dados foram coletados do Yahoo Finance, o qual será a mesma fonte de dados

para esse trabalho. Todavia, foi possível perceber que utilizar a média móvel não produziu resultados satisfatórios na pesquisa e que portanto, foi uma estratégia descartada pelo autor.

Devido ao custo computacional mais elevado para treinamento e ao fato de já existirem muitas pesquisas utilizando a rede LSTM para a previsão das cotações, este trabalho optou-se por adotar uma rede neural relativamente mais simples e fácil de treinar e configurar, denominada de *Multilayer Perceptron* ou simplesmente MLP. Esta se mostrou bastante promissora quando comparada com a LSTM no trabalho de Roger Achkar[10]. Naquela pesquisa, os testes foram realizados em diferentes conjuntos de dados, como ações do Google, Facebook e Bitcoin. Os resultados foram satisfatórios para MLP, visto que obteve uma precisão de previsão tão quanto a LSTM e ainda com um tempo de processamento mais curto.

Uma análise feita por Halid El Himdi e Hassan Oukhouya[11] comparou também a MLP com outras redes neurais tradicionais. Tal pesquisa mostrou que a MLP obteve melhores resultados de RMSE (*Root Mean-Square Error*) em relação a LSTM. Sendo assim, essas pesquisas supracitadas mostram que a rede neural tratada neste trabalho é uma boa opção para prever ativos se saindo tão bem quanto ou até mesmo melhor em alguns casos de outras redes neurais convencionais.

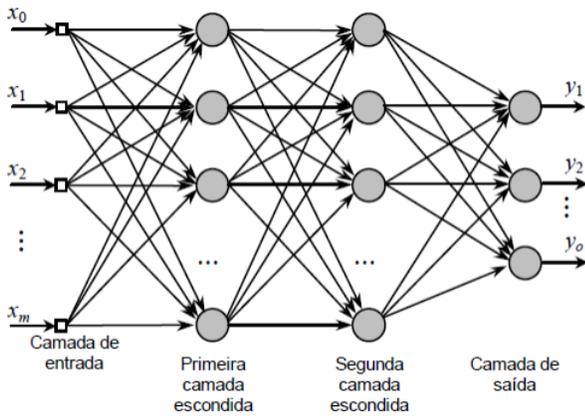
2.2 Multilayer Perceptron (MLP)

MLP - *Multilayer Perceptron* - é um modelo de rede neural formada pelo encadeamento de camadas de neurônios que supera as limitações do seu modelo precursor, o Perceptron, proposto por Frank Rosenblatt nas décadas de 1950 e 1960. O Perceptron de simples camada não era capaz de aprender problemas não linearmente separáveis [12]. A arquitetura MLP contém uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. A camada oculta tem uma função de ativação não linear, como a tangente hiperbólica ou uma função retificada linear - ReLU [13].

Um maior detalhamento das diferentes camadas de uma MLP é apresentado a seguir (Figura 1):

- Camada de entrada: recebe os dados de entrada. O foco é receber e disponibilizar os dados para próxima etapa;
- Camada interna, também conhecida por camada oculta: possui neurônios totalmente conectados aos neurônios da camada anterior e posterior. Tais conexões garantem a essa rede uma grande quantidade de parâmetros ajustáveis para fins de treinamento;
- Camada de saída: constituída por um (no caso de regressão de um valor contínuo, como é o caso deste trabalho) ou mais neurônios (mais adequada a problemas de classificação, em que cada neurônio associa-se a uma classe).

Figura 1: Ilustração da estrutura de uma MLP

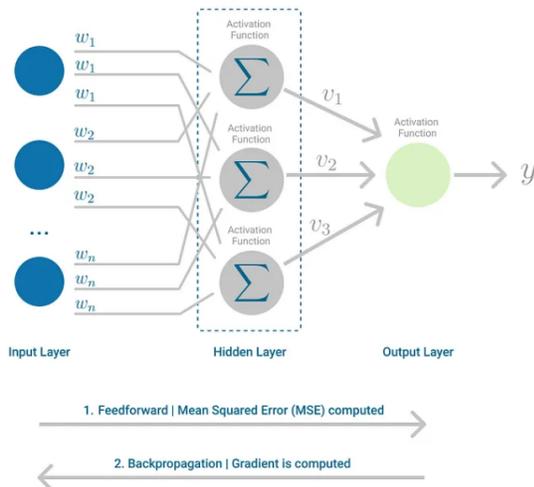


fonte: BOCCATO, Levy. Perceptron de múltiplas camadas e retropropagação de erro, 2019. Disponível em: https://www.dca.fee.unicamp.br/~lboccatto/topico_6.2_redes_neurais.pdf. Acesso em: 4 nov. 2023.

A MLP faz parte da categoria de redes neurais feedforward, sendo assim, durante a fase de propagação para frente (*forward propagation*) os dados vão passando por cada uma das três camadas e os neurônios presentes na camada oculta irão receber uma combinação ponderada das saídas de camadas anteriores e aplicar uma função de ativação sobre elas. Ao aplicar esta função nas entradas ponderadas, é favorecida a captura de informações não lineares no mapeamento entre entradas e saídas.

No final da etapa de propagação para frente ocorre o cálculo da função de perda que, no caso deste trabalho e para a maioria dos problemas de regressão, é utilizado o Erro Médio Quadrático. Então, ocorre a segunda etapa mais importante de uma MLP que é a retropropagação do gradiente (*Backpropagation*). Nessa fase, na camada de saída é calculado o gradiente da função de perda em relação aos pesos das conexões e então é propagado até a camada de entrada novamente promovendo um ajuste de pesos na conexão entre os neurônios com o fito de minimizar a função de perda [14].

Figura 2: Ilustração do treinamento da MLP [14]



Nesse contexto, identificam-se diversas vantagens ao utilizar uma MLP para prever ativos. A primeira deve-se à sua flexibilidade para modelar padrões complexos capturando interdependência nos dados devido a suas múltiplas camadas. Outro aspecto importante é o fato das camadas ocultas poderem aprender características intermediárias nos dados como a detecção de tendências do mercado. E por fim, um fator relevante é a sua facilidade de implementação bem como de treinamento. Em conclusão, a rede MLP, apesar de ser considerada relativamente simples, atende a priori aos objetivos iniciais deste trabalho.

3. METODOLOGIA

3.1 Obter dados do Yahoo Finance

Inicialmente, foram importados os dados de três empresas separadamente para avaliar como a rede neural se comportaria nos três principais tipos de movimentações do mercado. Sendo assim, foram coletados dados de uma empresa com ações em alta (PETR4), outra em baixa (MGLU3) e outra com suas ações lateralizadas (TAEE11) diretamente do Yahoo Finance, que é um site que disponibiliza diversas informações financeiras e dados históricos de empresas voltadas para aquelas que têm interesse em acompanhar o mercado financeiro. Para esse trabalho, o interesse maior era utilizar a API deste site para baixar apenas os dados relacionados às cotações da empresa escolhida durante o tempo.

Para isso, era necessário primeiramente formatar os códigos de cada uma das ações para que pudessem atender a formatação exigida pelo site para coleta de dados (PETR4.SA, MGLU3.SA, TAEE11.SA). Após isso, definiu-se o período de aquisição referente aos últimos 2 anos. A API do Yahoo Finance retornou 500 dias de cada uma das ações com os preços de abertura, os preços mais altos e baixos atingidos nos dias, os de fechamento, o volume de negociação da empresa no dia, a distribuição de dividendos e o estoque diário (Figura 3).

Figura 3: Ilustração dos dados importados da PETR4

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2021-10-20 00:00:00-03:00	11.833823	12.055680	11.699871	11.930100	56063300	0.0	0.0
2021-10-21 00:00:00-03:00	11.662197	11.796149	11.268713	11.490571	101567700	0.0	0.0
2021-10-22 00:00:00-03:00	11.344061	11.620337	10.787323	11.377549	138384700	0.0	0.0
2021-10-25 00:00:00-03:00	11.620338	12.269167	11.574291	12.156146	135830600	0.0	0.0
2021-10-26 00:00:00-03:00	12.055680	12.235678	11.971961	12.038937	57865600	0.0	0.0
...
2023-10-16 00:00:00-03:00	36.400002	36.799999	35.959999	36.680000	49626600	0.0	0.0
2023-10-17 00:00:00-03:00	36.700001	37.669998	36.630001	37.669998	68814800	0.0	0.0
2023-10-18 00:00:00-03:00	37.990002	38.610001	37.910000	38.520000	85931700	0.0	0.0
2023-10-19 00:00:00-03:00	38.360001	38.470001	37.919998	38.340000	56016600	0.0	0.0
2023-10-20 00:00:00-03:00	38.689999	38.860001	37.389999	37.849998	66242400	0.0	0.0

3.2 Pré-processamento

Na etapa de preparação dos dados para o modelo, primeiramente foi selecionada a coluna de interesse para esse trabalho, contendo as cotações de fechamento diário, uma vez que, o modelo objetiva prever justamente o preço de fechamento. Após essa etapa, os dados foram divididos em 80% para treinamento (400 dias) e 20% para testes (100 dias).

Ademais, com o fito de aumentar a interpretabilidade dos dados pelo modelo, estes foram normalizados entre 0 e 1

utilizando o algoritmo MinMaxScaler da biblioteca scikit-learn¹. É importante destacar que primeiro foram normalizados os dados de treinamento e, a partir do escalonador criado destes dados, os dados de testes foram normalizados. O objetivo dessa separação é justamente simular as condições reais evitando o vazamento de informações do conjunto de testes na etapa de pré-processamento.

Logo em seguida, foi montado um array a partir dos dados de treinamento para que o nosso modelo pudesse utilizar os 60 dias anteriores para prever o preço da ação no dia seguinte. No final, foram obtidos vetores de 2 dimensões com 60 cotações e 1 preço real (saída desejada). A técnica da janela deslizante foi adotada para a obtenção desses vetores. Por fim, os dados foram encapsulados numa estrutura de array de 3 dimensões, uma exigência da biblioteca Keras², adotada para a simulação da rede neural.

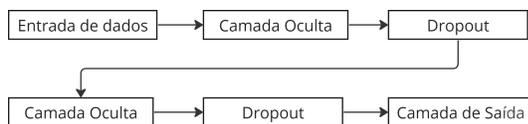
3.3 Montando a Rede Neural

Como dito anteriormente, criar uma MLP é simples. Portanto, neste trabalho fez uso da classe Sequential da biblioteca Keras para facilitar a montagem da rede neural. Inicialmente foram criadas duas camadas densas ocultas que representam a parte mais interna do modelo com 10.000 neurônios e 60 neurônios respectivamente, aplicando a estas, uma função de ativação Tangente hiperbólica para calcular as saídas dos neurônios e introduzindo a não linearidade na rede. Além disso, entre cada uma dessas camadas e a de saída foi adicionada uma camada de Dropout a fim de evitar o overfitting desativando 20% dos neurônios durante a etapa de treinamento. Por fim, foi adicionada mais uma camada densa com um único neurônio que representa a camada de saída (Figura 5). Nesta, foi utilizada a função de ativação linear.

Na fase de compilação do modelo, aplicou-se o otimizador Adam para o ajuste dos pesos durante o treinamento e como função de perda, utilizou-se o Erro Médio Quadrático, devido a sua alta aplicabilidade em problemas de regressão para diminuição de erro entre as saídas previstas e valores reais, que no caso representam nesse trabalho os preços previstos de cada ação e o preço real.

Na etapa de treinamento, foi definido um tamanho de lote de 32 com vistas à aceleração do treinamento para que o resultado seja rapidamente exibido para o usuário da aplicação. Por fim, foram definidas 100 épocas para a realização do treinamento da rede neural.

Figura 5: Estrutura do modelo



¹ Scikit-learn: Machine Learning in Python. (2011). Disponível em: <https://scikit-learn.org/stable/index.html>

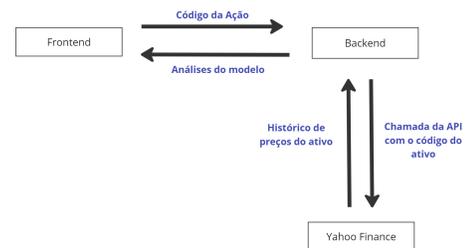
² Keras. Disponível em: <https://keras.io>. Acesso em 2 de novembro de 2023.

3.4 Backend da aplicação

Para construir o backend da aplicação, decidiu-se utilizar o framework FastAPI³, uma vez que permite a construção de uma API de forma muito simples utilizando Python, que neste trabalho, foi a linguagem utilizada para o pré-processamento de dados e construção da rede MLP. A partir do uso deste framework foi possível concentrar todo tratamento de dados em uma parte do código com apenas uma linguagem.

No backend, após receber o código da ação, ocorre a chamada da API do Yahoo Finance que retorna os dados de cada empresa e então é feito todo o tratamento e previsão descrito nas seções anteriores (Figura 6). Com isso, em um endpoint é retornado um json para o frontend com os dados das cotações reais, os dados de previsões em uma simulação de 100 dias, o valor previsto para o futuro (um dia após o dia atual) e o valor da métrica de erro (RMSE), para aqueles que possuem algum conhecimento mais especializado possam se aprofundar mais na avaliação da precisão do modelo criado.

Figura 6: Estrutura geral da solução proposta



3.5 Front End da aplicação

Para criar o frontend da aplicação⁴, que possibilita ao usuário analisar os seus ativos, foi utilizado o framework React Js⁵ (Figura 7). No frontend pode-se digitar o código da ação bem como o valor inicial a ser investido. Com base nisso, a aplicação retornará para ele, um gráfico equivalente às cotações do determinado ativo ao longo do tempo com uma simulação de previsão durante 100 dias para que este possa analisar visualmente se a IA está conseguindo seguir bem o mercado. Além disso, também é calculado o valor de lucro a partir do valor inicial a ser investido proposto caso o usuário tivesse seguido a IA durante os 100 dias da simulação.

Para o cálculo do lucro, o algoritmo leva em consideração se o preço previsto para o dia posterior é maior ou menor do que o preço real do dia atual. Caso seja maior, indica-se uma compra para o usuário, caso não, é indicado uma venda. Em situações em que o preço previsto é igual ao preço atual, a aplicação indicará para o usuário esperar uma movimentação do mercado como citadas anteriormente. Tendo em vista isso, os lucros serão gerados a partir de compras na baixa e venda na alta. No entanto, vale ressaltar que o lucro não se dá necessariamente apenas com o acerto do valor previsto, uma vez que, o valor previsto pode ser diferente do real, mas se o modelo apenas acertar o “lado” para que o mercado irá, o usuário irá sim ter um

³ FastAPI. Disponível em: <https://fastapi.tiangolo.com>

⁴Aplicação. Disponível em: <https://github.com/italomiguelpinheiro/asset-right>

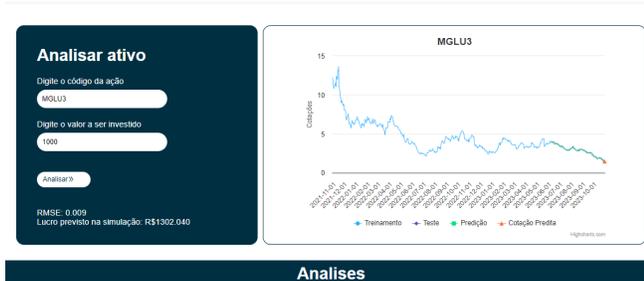
⁵ ReactJs. Disponível em: <https://pt-br.reactjs.org/>

lucro naquela determinada operação. Por fim, a aplicação apenas indica uma compra caso ainda esteja dentro do orçamento previamente informado pelo usuário no investimento inicial, para que assim, esse não fique no negativo e tenha um prejuízo muito maior caso o modelo erre a predição (Figura 8).

Figura 7: Parte inicial da aplicação



Figura 8: Parte de análises da aplicação



Análises	
Cotação Atual 1.440	Indicação Compra▲
Predição Para Amanhã 1.480109	Lucro na Venda 46.109

4. RESULTADOS E DISCUSSÕES

Como dito anteriormente, um dos objetivos do trabalho era observar como o modelo se comportaria mediante as três principais movimentações do mercado: alta, baixa e lateralização. Diante disso, as figuras 9, 10 e 11 representam justamente a atuação do modelo preditivo a partir dos 100 dias finais.

Figura 9: Gráfico das cotações da PETR4 com as previsões do modelo

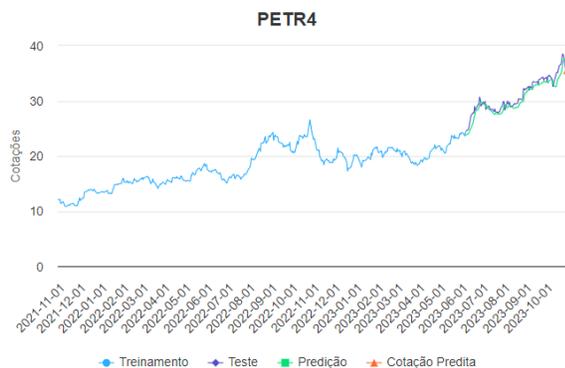
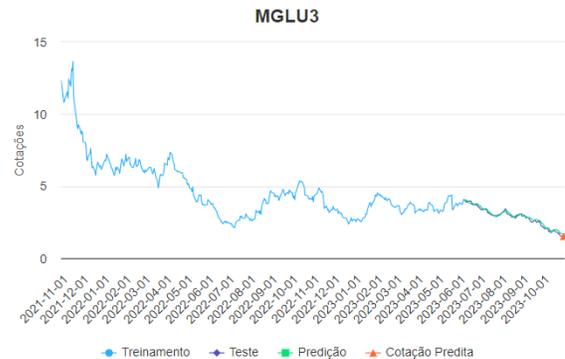


Figura 10: Gráfico das cotações da TAEE11 com as previsões



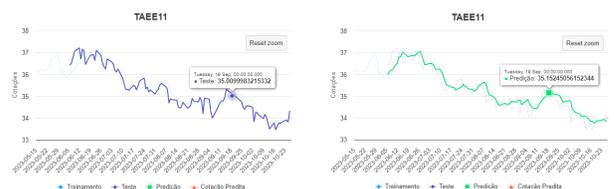
Figura 11: Gráfico das cotações da MGLU3 com previsões do modelo



Nos gráficos apresentados nas figuras anteriores, a linha azul clara representa as cotações das ações que foram utilizadas no treinamento, a linha azul escura indica a parte das cotações que foram aplicadas para os dados de testes, que neste trabalho representam 100 dias. A linha verde indica justamente a predição do modelo durante esses 100 dias e por fim, o triângulo laranja é a predição para o dia futuro a partir do dia atual. Diante do exposto, pode-se notar que o modelo conseguiu seguir as tendências do mercado de maneira satisfatória, uma vez que, nos três casos, a linha de predição e de teste estão relativamente próximas.

Além de uma avaliação qualitativa, o usuário pode analisar quantitativamente o quão bem está indo o modelo passando o mouse por cima das linhas para ver o quão perto ou não dos preços reais o modelo está conseguindo prever (Figura 12).

Figura 12: Imagem dos gráficos simulando o mouse do usuário



A fim de facilitar a visualização dessa mesma análise quantitativa neste trabalho, as tabelas 1, 2 e 3 apresentam os preços de fechamento reais e os preços previstos pelo modelo para as ações TAE11, PETR4 e MGLU3, respectivamente.

Tabela 1: Tabela de previsão de preços da TAE11

TAE11		
Data	Fechamento	Predição
2023-06-05	35.752144	36.163498
2023-06-06	36.424877	35.934654
2023-06-07	36.522373	36.031681
2023-06-09	37.058605	36.096500
2023-06-12	37.204853	36.352123
...
2023-10-19	33.770000	33.689754
2023-10-20	33.759998	33.730141
2023-10-23	33.889999	33.757385
2023-10-24	33.889999	33.799263

Tabela 2: Tabela de previsão de preços da PETR4

PETR4		
Data	Fechamento	Predição
2023-06-02	24.572840	23.800659
2023-06-05	24.835022	24.008909
2023-06-06	25.359388	24.260511
2023-06-07	26.100733	24.623026
2023-06-09	27.375483	25.151066
...
2023-10-19	38.340000	37.282166
2023-10-20	37.849998	37.584751
2023-10-23	35.349998	37.506527
2023-10-24	35.880001	36.337261

Tabela 3: Tabela de previsão de preços da MGLU3

MGLU3		
Data	Fechamento	Predição
2023-06-02	4.059999	4.034008
2023-06-05	3.890000	4.057829
2023-06-06	3.990000	3.972172
2023-06-07	3.880000	3.958044
2023-06-09	3.970000	3.840865

...
2023-10-19	1.610000	1.806863
2023-10-20	1.539999	1.739827
2023-10-23	1.509999	1.659856
2023-10-24	1.409999	1.611359

Analisando cada tabela, fica claro a proximidade em que o modelo está conseguindo prever os preços de fechamento. Em diversos casos o erro que o modelo comete é da ordem de centavos, o que é muito favorável para os investidores, sejam eles iniciantes ou não, conseguirem ter uma noção dos seus trades de forma mais precisa.

Como dito anteriormente, a aplicação também disponibiliza nos resultados para o usuário o RMSE - *Root Mean-Square Error* ou Raiz Quadrada do Erro-médio - para aqueles que entendem um pouco mais das métricas avaliativas de modelos de redes neurais, ou até mesmo, que possuem um conhecimento prévio de cálculos matemáticos que envolvem erros. A fim de simplificação, centralizamos o RMSE dos três casos testados neste trabalho na tabela 4.

Tabela 4: Tabela com os valores de RMSE de cada ação

AÇÃO	RMSE
PETR4	0.661
TAE11	0.025
MGLU3	0.060

Nota-se na tabela acima os baixos valores de RMSE nas três ações de teste neste trabalho. Vale a pena destacar que dos três, o maior RMSE foi o da PETR4, o que provavelmente ocorreu devido ao fato desta ação possuir maior instabilidade na curva. Esse comportamento acabou confundindo um pouco o modelo em alguns momentos e acabou refletindo nesse resultado.

Como mencionado anteriormente, a aplicação fornece uma estimativa de lucro hipótese do usuário ao seguir as recomendações ao longo dos 100 dias de previsão. A efeito de simulação, foi inserido um valor inicial de R\$1000,00 para analisar quanto de lucro o usuário teria no final e os resultados estão expressos na tabela 5, para cada uma das ações consideradas.

Tabela 5: Tabela com os valores de lucro em 100 dias de operação

AÇÃO	LUCRO (R\$)	LUCRO (%)
PETR4	R\$437.56	43.7%
TAE11	R\$354.18	35.4%
MGLU3	R\$232.30	23.2%

Fica nítido o bom desempenho do modelo, uma vez que, ao aplicar uma quantia de R\$1000,00 nas ações citadas nesse

trabalho e seguindo o modelo durante a simulação de 100 dias, o montante final do usuário representou acréscimos entre 23.2% e 43.7% do valor inicial disponível, a depender do ativo.

É importante destacar que a PETR4 mesmo tendo o maior RMSE rendeu o maior lucro projetado. Isso ocorreu devido ao fato de ser uma ação em tendência de alta. Como dito anteriormente, o lucro é resultante não do acerto da predição do preço de fechamento, basta apenas o modelo acertar a direção do movimento da ação. Supondo como exemplo que a ação custasse R\$10.00 e o modelo previu que no dia futuro a ação estaria custando R\$20.00 mas que na verdade ela chegou a um preço de R\$40.00, mesmo assim o algoritmo iria indicar uma compra e o lucro não seria de apenas R\$10.00, mas sim de R\$30.00. Além disso, por estar em tendência de alta, os aportes seriam cada vez maiores, o que contribuiu para que o lucro do usuário investindo nessa ação fosse maior em relação às outras. Por fim, o caso inverso ocorreu com a MGLU3 que obteve o menor lucro, uma vez que, esse ativo estava em tendência de baixa.

5. CONCLUSÃO

A quantidade de pessoas atuando no mercado de ativos tem crescido por inúmeros fatores e algo que permeia na mente de cada investidor, principalmente daqueles que estão iniciando nesse universo, é a incerteza de qual ativo será o melhor investimento, ou então, quando comprar e vender no momento certo. Neste trabalho foi abordada essa temática e desenvolvida uma aplicação para dar suporte a esses investidores por meio de uma rede neural artificial do tipo *Multilayer Perceptron*.

Nota-se que o modelo proposto neste trabalho conseguiu atender muito bem as três tendências principais do mercado. Ao final das simulações, pode-se perceber que este teve uma alta proximidade entre os preços reais e os previstos. Essa equivalência favoreceu para que o algoritmo implementado no frontend trouxesse lucros significativos a partir de um certo valor inicial e deixou claro para os usuários bons pontos de entrada e saída.

Não foi o único propósito do trabalho criar um modelo que fosse capaz de prever bons momentos de compra e venda, mas também tornar palpável para qualquer usuário, incluindo leigos, um auxílio na tomada de decisão. Dessa forma, após realizar todas as análises e predições, a aplicação proposta traz indicativos importantes e de forma acessível para que o usuário possa se sentir mais seguro e confiante no momento de investir, além de evidenciar também, a lucratividade que este terá ao seguir o modelo.

Em suma, o trabalho atingiu o objetivo desejado, mas há incrementos possíveis para a aplicação, a exemplo de uma maior exatidão na predição dos valores futuros. Em trabalhos futuros, planeja-se adicionar uma maior robustez à rede neural da aplicação a partir da experimentação de outras arquiteturas, como a rede ESN (*Echo State Network*) [15]. Ademais, para estudos subsequentes é sugerido implementar no algoritmo de cálculo de lucro os descontos de impostos e de taxas de corretoras na qual o usuário opera no mercado para que este possa ter uma noção mais precisa de quanto seria seu lucro líquido.

Por fim, vale ressaltar que este trabalho representa uma etapa inicial de um esforço para facilitar o acesso ao mercado financeiro com reais possibilidades de ganhos. Espera-se que este

trabalho possa auxiliar o investidor no seu processo de tomada de decisão

6. REFERÊNCIAS

- [1] REIS, Tiago. “**Tudo sobre o Mercado Financeiro: o que é e como ele funciona?**”. Suno, 2022. Disponível em: <https://www.suno.com.br/guias/mercado-financeiro/>. Acesso em: 10, dez. de 2022.
- [2] NOVELO, Rafael. “**Bolsa de valores: por que as pessoas investem?**”. Politize, 2021. Disponível em: <https://www.politize.com.br/bolsa-de-valores-por-que-as-pessoas-investem/>. Acesso em: 10, dez. de 2022
- [3] MACIEL, Camile. “**Investimento de pessoa física em renda variável cresce 35%**”. Agência Brasil, 2022. Disponível em: <https://agenciabrasil.ebc.com.br/economia/noticia/2022-11/investimento-de-pessoa-fisica-em-renda-variavel-cresce-35#:~:text=Levantamento%20da%20B3%2C%20bolsa%20de%4%2C6%20milh%C3%B5es%20neste%20ano> Acesso em: 11, dez. de 2022.
- [4] ROSA, Eduardo. “**Pesquisa Reclame AQUI: Maioria dos brasileiros ainda não investe seu dinheiro**”. Reclame Aqui, 2021. Disponível em: <https://noticias.reclameaqui.com.br/noticias/pesquisa-reclame-aqui-maioria-dos-brasileiros-ainda-nao-inve-4158/> Acesso em 12, dez. de 2022..
- [5] RODRIGUES, Edilma. “**63% dos brasileiros não investem por falta de conhecimento**”. Cantarino Brasileiro, 2020. Disponível em <https://cantarinobrasileiro.com.br/blog/63-dos-brasileiros-nao-investem-por-falta-de-conhecimento/> Acesso em: 12, dez.
- [6] “**Análise Técnica de ações: aprenda a interpretar gráficos e índices**”. Infomoney, 2022. Disponível em: <https://www.infomoney.com.br/guias/analise-tecnica/>. Acesso em 18, set. de 2023.
- [7] MWITI, Derrick. **Using a keras long short-term memory (LSTM) model to predict stock prices**, 2018. Acessível em: <https://www.kdnuggets.com/2018/11/keras-long-short-term-memory-lstmmodel-predict-stock-prices.html>. Acesso em 10 de Agosto, 2023.
- [8] A. Moghar, M. Hamiche. **Stock market prediction using lstm recurrent neural network**. Procedia Comput. Sci 170 (2020) 1168–1173, Elsevier.
- [9] XAVIER, Alexandre. **Predicting stock prices with LSTM | neurónio | medium** [internet], 2019, Disponível em: <https://medium.com/neuronio/predicting-stock-prices-with-lstm349f5a0974d4>. Acesso em 10 de Agosto de 2023.
- [10] R. Achkar, F. Elias-Sleiman, H. Ezzidine and N. Haidar, “**Comparison of BPA-MLP and LSTM-RNN for Stocks Prediction**,” 2018 6th International Symposium on Computational and Business Intelligence (ISCBI), Basel, Switzerland, 2018, pp. 48-51, doi: 10.1109/ISCBI.2018.00019. Acesso em 29 de Outubro de 2023.
- [11] Oukhouya, H.; El Himdi, K. **Comparing Machine Learning Methods—SVR, XGBoost, LSTM, and MLP— For**

- Forecasting the Moroccan Stock Market.** Comput. Sci. Math. Forum 2023, 7, 39. <https://doi.org/10.3390/IOCMA2023-14409>. Acesso em 29 de Outubro de 2023.
- [12] ROSENBLATT, Frank (1957). **"The Perceptron—a perceiving and recognizing automaton"**. Report 85-460-1. Cornell Aeronautical Laboratory. Acesso em 29 de Outubro de 2023.
- [13] Data Science Academy. Deep Learning Book, Capítulo 10. Disponível em <https://www.deeplearningbook.com.br/as-10-principais-arquiteturas-de-redes-neurais/>. Acesso em 29 de Outubro de 2023.
- [14] BENTO, Carolina. **Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis,** 2021. Disponível em <https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141>. Acesso em 29 de Outubro de 2023.
- [15] Jaeger, H. **The “echo state” approach to analyzing and training recurrent neural networks.** GMD Report 148, GMD – German National Research Institute for Computer Science, 2001. <http://www.gmd.de/People/Herbert.Jaeger/Publications.html>. Acesso em 2 de Novembro de 2023.