

Yuri Gandra Santos

Aplicações de Redes Neurais na Previsão de Operações Day Trade

Formiga - MG

2023

Yuri Gandra Santos

Aplicações de Redes Neurais na Previsão de Operações Day Trade

Monografia do trabalho de conclusão de curso apresentado ao Instituto Federal Minas Gerais - Campus Formiga, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais

Campus Formiga

Ciência da Computação

Orientador: Mário Luiz Rodrigues Oliveira

Coorientador: Rodrigo Menezes Sobral Zacaroni

Formiga - MG

2023

Santos, Yuri Gandra

S237a Aplicações de redes neurais na previsão de operações Day Trade / Yuri
Gandra Santos – Formiga : IFMG, 2023.
57p. : il. color.

Orientador: Prof. Mário Luiz Rodrigues Oliveira

Coorientador: Me. Rodrigo Menezes Sobral Zacaroni

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação)
Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – *Campus*
Formiga.

1. Robô de investimento. 2. Redes neurais. 3. Mercado financeiro. 4. Day
Trade. 5. Inteligência artificial. I. Oliveira, Mário Luiz Rodrigues. II.
Zacaroni, Rodrigo Menezes Sobral. III. Título.

CDD 004



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS
Campus Formiga
Diretoria de Ensino
Docência Área Acadêmica de Computação
Rua São Luiz Gonzaga, s/n - Bairro São Luiz - CEP 35570-000 - Formiga - MG
- www.ifmg.edu.br

YURI GANDRA SANTOS

Aplicações de Redes Neurais na Previsão de Operações *Day Trade*

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Minas Gerais - Campus Formiga, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

APROVADO em: 22 de novembro de 2023.

BANCA EXAMINADORA

Prof. Mário Luiz Rodrigues Oliveira (orientador, IFMG)

Prof.^a Rodrigo Menezes Sobral Zacaroni (co-orientador, IFMG)

Prof.^o Marlon Jesus Lizarazo Urbina (IFMG)

Prof.^o Danton Diego Ferreira (UFLA)



Documento assinado eletronicamente por **Mario Luiz Rodrigues Oliveira, Professor**, em 22/11/2023, às 12:11, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Marlon Jesus Lizarazo Urbina, Professor Substituto**, em 22/11/2023, às 12:14, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Danton Diego Ferreira, Usuário Externo**, em 22/11/2023, às 12:15, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Rodrigo Menezes Sobral Zacaroni, Fiscal de Contrato Substituto(a)**, em 22/11/2023, às 12:23, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **1736821** e o código CRC **18E97037**.

23211.002045/2021-71

1579644v1

Dedico este trabalho à minha família. Pai, mãe, irmãs e madrasta

Agradecimentos

Primeiramente, gostaria de agradecer a Deus pela conquista, ao meu pai por todo apoio e parceria de sempre, minha família pelo carinho e conforto que me passavam mesmo que de longe e aos meus orientadores pela paciência e os ensinamentos fornecidos.

“Nós nos tornamos o que pensamos a maior parte do tempo, e esse é o segredo mais estranho.” (Earl Nightingale)

Resumo

Esta monografia abordou o desenvolvimento e análise de quatro modelos de redes neurais: Autoencoders, Redes neurais convolucionais, Redes neurais *Long Short-Term Memory* (LSTM) e as Redes neurais *multilayer* perceptron (MLP), visando avaliar seu desempenho em operações de *day trade*. Os resultados destacaram a complexidade da implementação de redes neurais no ambiente financeiro, com cada modelo apresentando características únicas e *trade-offs* significativos. O estudo ressaltou a importância de abordagens adaptativas e cuidadosas na escolha e implementação de redes neurais em sistemas de negociação, reconhecendo a inexistência de uma solução universal. No trabalho esses modelos foram implementados, testados e avaliados utilizando métricas como lucro obtido, número de negociações, porcentagem de negociações com lucro, fator de lucro e fator de recuperação. Para treinamento dos modelos foi utilizado o período de 01/2018 até 01/2023, com 70% dos dados coletados destinados a treinamento e 30% para teste. Utilizou-se 5 *features* como entrada para cada uma das redes implementadas, sendo: preço de abertura, preço de fechamento, máxima, mínima e uma média móvel de 21 períodos. Ao fim do trabalho concluí-se que o melhor modelo de rede neural dentre as analisadas foi o autoencoder. Por obter o maior lucro e na maior parte dos testes os melhores resultados de acordo com as métricas adotadas. Isso não indica que as outras são ruins ou não podem ser utilizadas no mercado financeiro. Apenas que para o tempo gráfico de 15 minutos o modelo que apresentou o melhor resultado foi o autoencoder e que nesse cenário ele é a melhor opção.

Palavras-chave: Robô de Investimento, Redes Neurais, Mercado Financeiro, Day Trade, Inteligência Artificial.

Abstract

This monograph addressed the development and analysis of four neural network models: Autoencoders, Convolutional Neural Networks, Long Short-Term Memory (LSTM) Neural Networks, and Multilayer Perceptron (MLP) Neural Networks, aiming to evaluate their performance in day trading operations. The results highlighted the complexity of implementing neural networks in the financial environment, with each model exhibiting unique characteristics and significant trade-offs. The study emphasized the importance of adaptive and careful approaches in choosing and implementing neural networks in trading systems, recognizing the absence of a universal solution. In this work, these models were implemented, tested, and evaluated using metrics such as profit obtained, number of trades, percentage of profitable trades, profit factor, and recovery factor. The models were trained using data from January 2018 to January 2023, with 70% of the collected data allocated for training and 30% for testing. Five features were used as input for each implemented network: opening price, closing price, high, low, and a 21-period moving average. At the end of the study, it was concluded that the best neural network model among those analyzed was the autoencoder. It achieved the highest profit and, in most tests, the best results according to the adopted metrics. This does not indicate that the other models are poor or cannot be used in the financial market, but rather that for the 15-minute time frame, the autoencoder was the most effective model and the best option in this scenario.

Keywords: Investment Robot, Neural Network, Financial Market, Day Trade, Artificial intelligence.

Lista de ilustrações

Figura 1 – Estrutura de uma rede neural. Fonte: Autor	22
Figura 2 – Estrutura de uma unidade LSTM. Fonte: (ACADEMY, 2022)	26
Figura 3 – Camada de Pooling. Fonte: (ACADEMY, 2022)	28
Figura 4 – Funcionamento Autoencoder. Fonte: (ACADEMY, 2022)	30
Figura 5 – Reta da Regressão Linear. Fonte: (DAMÁSIO, s.d.)	34
Figura 6 – Estrutura da Aplicação. Fonte: Autor	38
Figura 7 – Predito x Real - Rede Neural MLP (Zoom). Fonte: Autor	41
Figura 8 – Predito x Real - Rede Neural MLP. Fonte: Autor	41
Figura 9 – Predito x Real - Rede Neural LSTM. Fonte: Autor	43
Figura 10 – Predito x Real - Rede Neural LSTM (Zoom). Fonte: Autor	43
Figura 11 – Predito x Real - Rede Neural Convolutacional. Fonte: Autor	45
Figura 12 – Predito x Real - Rede Neural Convolutacional (Zoom). Fonte: Autor	45
Figura 13 – Predito x Real - Autoencoder. Fonte: Autor	47
Figura 14 – Predito x Real - Autoencoder (Zoom). Fonte: Autor	48
Figura 15 – Comparação entre os Modelos. Fonte: Autor	49
Figura 16 – Autoencoder M5. Fonte: Autor	52
Figura 17 – LSTM M5. Fonte: Autor	53

Lista de tabelas

Tabela 1 – Termos utilizados no ambiente de <i>day trade</i>	18
Tabela 2 – Tabela de exemplos de estratégias no <i>day trade</i> . Fonte: Autor	18
Tabela 3 – Tabela de resultados da rede neural MLP	40
Tabela 4 – Tabela de resultados da rede neural LSTM	42
Tabela 5 – Tabela de resultados da rede neural convolucional	44
Tabela 6 – Tabela de resultados do Autoencoder	46
Tabela 7 – Tabela de resultados do Autoencoder M5	51
Tabela 8 – Tabela de resultados rede LSTM M5	53

Sumário

	Lista de tabelas	11
1	INTRODUÇÃO	14
1.1	Justificativa	15
1.2	Objetivos	16
1.2.1	Objetivo Geral	16
1.2.2	Objetivos Específicos	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Mercado Financeiro	17
2.1.1	<i>Day trade</i>	18
2.1.2	<i>Robô Trader</i>	19
2.2	Inteligência Artificial	19
2.2.1	Rede Neural Perceptron Multi-Camadas (MLP)	20
2.2.2	Redes Neurais de Longa Memória de Curto Prazo:	24
2.2.3	Redes Neurais Convolucionais	27
2.2.4	Autoencoders	29
2.3	Regressão Linear	32
3	MATERIAIS E MÉTODOS	35
3.1	Materiais	35
3.2	Métodos	36
3.3	Estrutura do Sistema	37
3.4	Servidor Python	38
3.5	Cliente MQL5	38
3.6	Avaliação dos Modelos	38
4	RESULTADOS E DISCUSSÕES	40
4.1	Rede Neural MLP	40
4.2	Rede Neural LSTM	42
4.3	Rede Neural Convolucional	44
4.4	Autoencoder	46
4.5	Comparação Entre os Modelos	48
4.6	Discussões sobre os resultados	49
5	CONCLUSÃO	55
5.1	Trabalhos Futuros	55

REFERÊNCIAS	56
--------------------------	-----------

1 Introdução

O avanço tecnológico dos últimos anos vem transformando vários setores da sociedade, e o mercado financeiro não é uma exceção. Com a atual disponibilidade em obter dados em tempo real, poder computacional, além da constante melhoria dos algoritmos que utilizam aprendizado de máquina. As aplicações visando o mercado financeiro têm aumentado cada vez mais, fazendo com que as análises gráficas e tomadas de decisões fiquem cada vez mais precisas e automatizadas. Desta forma, as operações de *day trade*, categorizadas por operações de compra ou venda de ativos realizadas em um único dia na bolsa de valores (BTG Pactual Digital, s.d.), se destacam como um ótimo ambiente para a aplicação de técnicas avançadas, como redes neurais.

As redes neurais foram desenvolvidas inspirando-se no funcionamento do cérebro humano, imitando a maneira em que os neurônios biológicos enviam sinais uns para os outros (IBM, s.d.). Esses modelos computacionais têm demonstrado habilidades impressionantes em tarefas complexas de processamento de dados e reconhecimento de padrões. No mercado financeiro, a capacidade das redes neurais de lidar com grandes volumes de informações e identificar relações entre variáveis torna-se particularmente atrativa para investidores e *traders* que buscam ganhos consistentes em operações de curto prazo.

Esta pesquisa se propõe a investigar quais são os melhores modelos de redes neurais que se destacam no contexto das operações de *day trade* no mercado financeiro. A avaliação desses modelos abrangerá diversas métricas que são utilizadas para avaliar robôs no mercado financeiro. Além disso, serão analisadas as arquiteturas mais adequadas para a representação dos dados financeiros, considerando a natureza sequencial e temporal das informações.

Ao compreender quais modelos de redes neurais apresentam desempenho superior nas operações de *day trade*, os *traders* e investidores poderão tomar decisões embasadas em análises mais precisas e consistentes, otimizando suas estratégias e potencialmente aumentando seus retornos financeiros. Ademais, a pesquisa contribuirá para a evolução do campo de finanças computacionais, ao explorar o potencial das redes neurais como ferramentas indispensáveis na tomada de decisões em um ambiente tão dinâmico e desafiador como o mercado financeiro.

No decorrer deste trabalho, serão abordados os fundamentos teóricos das redes neurais e do *day trade*, seguidos pela revisão da literatura que explora as aplicações dessas tecnologias no contexto financeiro. A metodologia adotada para avaliação dos modelos será detalhadamente descrita, assim como os conjuntos de dados utilizados e as métricas de desempenho empregadas. Por fim, os resultados obtidos serão analisados e comparados

entre si de acordo com cada modelo de rede, visando identificar os modelos mais áptos para serem utilizados no mercado. Culminando em conclusões que delineiam as implicações práticas e as perspectivas futuras dessa pesquisa.

Em resumo, esta monografia visa explicar sobre a interseção entre redes neurais e operações de *day trade* no mercado financeiro, desvendando quais modelos se destacam como os mais eficazes nesse contexto. Ao fazê-lo, busca-se contribuir para uma compreensão mais profunda das possibilidades e limitações das técnicas de aprendizado de máquina no ambiente desafiador das finanças, promovendo avanços significativos no campo das estratégias de investimento e tomada de decisões no curto prazo.

1.1 Justificativa

A crescente interseção entre tecnologia e finanças tem gerado um ambiente de intensa competição e inovação no mercado financeiro, onde a busca por vantagem competitiva é uma constante. As operações de *day trade*, caracterizadas por sua agilidade e curto prazo, são um cenário particularmente desafiador e lucrativo para *traders* e investidores. No entanto, a volatilidade e a complexidade dos mercados financeiros exigem abordagens analíticas sofisticadas que vão além das estratégias tradicionais.

Nesse contexto, as redes neurais emergem como uma ferramenta promissora, capaz de processar grandes volumes de dados, identificar padrões não-lineares e adaptar-se às mudanças do mercado em tempo real. A adoção de modelos de redes neurais nas operações de *day trade* pode oferecer uma série de benefícios, como aprimorar a precisão das previsões de preços, otimizar estratégias de entrada e saída, e mitigar riscos associados à tomada de decisões baseada em informações incompletas ou subjetivas (GAZONI, 1999).

No entanto, apesar do potencial das redes neurais, a aplicação eficaz desses modelos no contexto do *day trade* ainda carece de um entendimento aprofundado e sistemático. A seleção adequada de arquiteturas de redes neurais, a escolha de métricas de avaliação de desempenho apropriadas e a compreensão das peculiaridades dos dados financeiros são desafios cruciais a serem enfrentados. Portanto, a realização desta monografia visa preencher essa lacuna ao investigar, de maneira estruturada e rigorosa, quais são os melhores modelos de redes neurais que se destacam nas operações de *day trade* no mercado financeiro (GAZONI, 1999).

A relevância desta pesquisa reside na sua capacidade de contribuir significativamente para o avanço das estratégias de investimento e tomada de decisões no mercado financeiro. Ao compreender quais modelos de redes neurais oferecem melhor desempenho no *day trade*, os *traders* e investidores podem se beneficiar de *insights* mais precisos e confiáveis, aumentando suas chances de sucesso e reduzindo potenciais perdas. Além disso, a pesquisa pode também enriquecer o campo das finanças computacionais, oferecendo novas

perspectivas sobre como as tecnologias de aprendizado de máquina podem ser aproveitadas em ambientes dinâmicos e voláteis como o mercado financeiro.

Portanto, esta monografia busca não apenas aprimorar o conhecimento teórico sobre a aplicação de redes neurais no *day trade*, mas também proporcionar orientações práticas que possam ser aplicadas de maneira tangível por profissionais e instituições atuantes no mercado financeiro. Ao explorar essa interseção entre tecnologia e finanças, esta pesquisa almeja contribuir para a eficiência e a eficácia das estratégias de investimento no contexto das operações de *day trade*.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo principal deste trabalho é realizar uma comparação entre os modelos de redes neurais: Autoencoder, *Long Short-Term Memory* (LSTM), Convolutacional e *multilayer perceptron* (MLP). Analisando como elas comportam-se no mercado financeiro, com foco específico em operações de *day trade*, identificando as que possuem maior relevância neste meio a partir de métricas amplamente utilizadas na avaliação de robôs em plataformas como MQL5 forum ([METAQUOTES, 2000](#)) e SMARTBOT ([SMARTTBOT, 2011](#)).

1.2.2 Objetivos Específicos

1. Desenvolver estratégias lucrativas de negociação para operações no mercado de *day trade*;
2. Identificar as diferenças de funcionamento entre quatro modelos de redes neurais: Autoencoders, Redes neurais convolucionais, Redes neurais LSTM e as Redes neurais MLP;
3. Identificar, dentre os modelos de redes neurais: Autoencoders, Redes neurais convolucionais, Redes neurais LSTM e as Redes neurais MLP, o mais indicado para realizar previsões em operações no mercado de *day trade*, e
4. Utilizar as métricas: fator de lucro, lucro obtido, fator de recuperação, número de operações e porcentagem de negociações com lucro para analisar a eficácia do modelos de redes neurais: Autoencoders, Redes neurais convolucionais, Redes neurais LSTM e as Redes neurais MLP, ao realizar previsões em operações no mercado de *day trade*.

2 Fundamentação Teórica

Para dar início ao presente trabalho de conclusão de curso foi feito um estudo sobre o mercado financeiro focando principalmente em operações de *day trade*. Analisando seu funcionamento, os termos utilizados nesse meio e estratégias de operação conhecidas nessa modalidade de investimento.

Com esse conhecimento formado, foram feitas pesquisas para entender e identificar diferentes modelos de redes neurais que se adequavam bem no mercado. De forma a ter uma variedade de amostras, possibilitando a comparação entre elas. O que auxiliou a identificar os pontos fracos e fortes de cada uma. Além disso, foi necessário um estudo sobre o funcionamento da regressão linear, já que foi aplicada sobre cada modelo escolhido para ser feita as previsões.

Decidido os modelos de rede a serem comparados, foram escolhidas as linguagens Python e MQL5 para serem feitas as implementações. Para fazer a comunicação entre as aplicações em duas linguagens diferentes foi utilizado uma arquitetura de cliente/servidor via sockets.

De forma a compreender o tema tratado neste trabalho, é preciso entender alguns princípios e noções introdutórias, que estão detalhados a seguir.

2.1 Mercado Financeiro

O mercado financeiro é um sistema que permite a alocação eficiente de recursos financeiros e a negociação de ativos financeiros entre diferentes participantes. Ele desempenha um papel crucial na economia global, fornecendo mecanismos para captação de recursos, investimentos e gerenciamento de riscos. Existem vários componentes do mercado financeiro, incluindo o mercado de ações, o mercado de títulos, o mercado de câmbio e o mercado de derivativos ([MIRA, 2022](#)).

O mercado de ações é uma parte fundamental do mercado financeiro, onde os investidores compram e vendem ações de empresas listadas em bolsas de valores. As ações representam a propriedade parcial de uma empresa e os investidores podem lucrar com a valorização das ações e com os dividendos distribuídos pelas empresas ([INVESTIMENTOS, 2023b](#)).

Existem também alguns importantes termos utilizados no mercado que, para melhor entendimento do trabalho, se encontram detalhados na tabela 1. Estes termos são constantemente utilizados tanto nesta área de investimento quanto no presente trabalho. Segue a tabela com alguns dos termos mais utilizados para dar um melhor entendimento

referente ao assunto abordado nesta pesquisa.

Tabela 1 – Termos utilizados no ambiente de *day trade*

Termo	Significado
<i>day trade</i>	Compra e venda de ativos no mesmo dia de negociação.
Scalping	Estratégia de <i>day trade</i> que busca pequenos lucros em negociações rápidas.
Stop Loss	Preço pré-definido para vender uma posição e limitar as perdas.
Take Profit	Preço pré-definido para vender uma posição e realizar o lucro.
Alavancagem	Uso de capital emprestado para aumentar o tamanho da posição.
Gap	Diferença entre os preços de abertura de dois dias consecutivos.
Liquidez	Facilidade de compra/venda de um ativo sem afetar seu preço.
Candlestick (Candle)	Gráfico que mostra a variação de preço de um ativo em um período de tempo específico.
Short Selling	Venda de ativos emprestados na expectativa de recomprá-los a um preço mais baixo.
Volume de Negociação	Quantidade de ativos negociados em um determinado período de tempo.
Análise Técnica	Método de análise que se baseia em gráficos e indicadores para prever movimentos de preços.
Análise Fundamentalista	Método de análise que avalia os fundamentos financeiros de uma empresa para tomar decisões de investimento.
Gap de Fuga (Breakaway Gap)	Um gap que indica uma mudança significativa na direção do preço.
Overbought (Sobrevendido)	Indica que um ativo pode estar sobrecomprado (overbought) ou sobrevendido (oversold).
Ponto de Pivô	Nível de preço que pode servir como suporte ou resistência.
Timeframe	Tempo gráfico em que está se operando no mercado.

2.1.1 Day trade

O *day trade* é uma modalidade específica de negociação no mercado de ações. Nesse tipo de operação, os *traders* compram e vendem ativos financeiros, como ações, em um mesmo dia, buscando lucrar com as flutuações de curto prazo nos preços. O *day trade* envolve posições que são abertas e fechadas no mesmo dia de negociação, o que significa que os *traders* não mantêm posições abertas durante a noite. O objetivo, na verdade, é lucrar com as movimentações diárias de cada papel. A volatilidade, nesse caso, é uma característica do mercado que o investidor procura aproveitar para obter ganhos (INFOMONEY, 2020).

A análise técnica é uma ferramenta essencial para os *traders* de *day trade*. Ela envolve o estudo dos gráficos de preços, indicadores técnicos e padrões de mercado para tomar decisões de compra e venda. Os *traders* usam análise técnica para identificar pontos de entrada e saída ideais para suas operações (INVESTIMENTOS, 2023a).

Existem várias estratégias de negociação em *day trade*, cada uma com suas características e riscos. Alguns exemplos incluem scalping, em que os *traders* buscam pequenos lucros com várias negociações ao longo do dia, e o day trading de tendência, em que os *traders* buscam lucrar com movimentos de tendência de curto prazo. Segue um exemplo na tabela 2 dos diferentes tipos de estratégia que podem ser aplicadas no *day trade*, de acordo com o perfil de cada investidor.

Estratégia	Descrição.
Contra Tendência	Este tipo de estratégia visa lucrar com a reversão ou correção das tendências do mercado.
A Favor da Tendência	Esta estratégia visa lucrar com fortes movimentos de grandes volumes em uma direção enquanto esse movimento persistir.
Scalping	Esta estratégia consiste em realizar operações muito curtas que geram pouco lucro, porém, oferecem pouco risco.

Tabela 2 – Tabela de exemplos de estratégias no *day trade*. Fonte: Autor

O *day trade* é uma atividade de alto risco devido à natureza volátil dos mercados de curto prazo. Há uma grande dificuldade em entender os padrões do mercado e prever seus movimentos futuros devido a diversos fatores econômicos, políticos e até mesmo emocionais. Por conta disso, pequeno é o número de investidores que obtiveram sucesso no mercado financeiro (ESPERANDIO, 2020). Portanto, o gerenciamento de riscos é fundamental. Isso inclui definir limites de perda, dimensionar adequadamente as posições e diversificar os ativos negociados.

Enfim o *day trade* é uma modalidade emocionante e potencialmente lucrativa no mercado financeiro, mas requer um profundo entendimento dos aspectos técnicos e dos riscos envolvidos. É fundamental que os *traders* estejam bem preparados, usem estratégias eficazes e pratiquem o gerenciamento de riscos para ter sucesso nessa empreitada. Apesar da complexidade para se operar no *day trade*, ainda assim é válido dedicar um tempo para estudo de mercado e estratégias eficientes para serem utilizadas nesse meio. Uma vez que o *day trade* proporciona uma chance de alta lucratividade e, além disso, permite alavancagem, ou seja, é possível investir um valor acima do que o investidor possui.

2.1.2 Robô Trader

Os robôs *traders* são consultores financeiros automatizados que tem função de gerenciar o dinheiro do investidor e realizar operações de compra ou venda no mercado financeiro, com base no conjunto de regras (algoritmos) feito pelo programador de acordo com uma estratégia que ele acha válida a ser implementada para análise do mercado.

Para (CHAVES, 2020) o robô trader é perfeito para os *day traders*, ao programar uma estratégia em que o investidor considere recorrente no mercado, não será necessário que o investidor permaneça durante todo o pregão esperando pelo padrão da estratégia, ele pode deixar com que o robô a busque e faça as operações necessárias.

Por se tratar de um método inovador e prático, os robôs no mercado financeiro são considerados uma grande tendência, além disso, são capazes de fazer o que um investidor faz e muitas das vezes até melhor por não terem sentimentos como sono, fome, e nem se distraírem dentre outros fatores que podem fazer o investidor perder o foco em uma operação e consequentemente sair no prejuízo (SMARTTBOT, 2020).

2.2 Inteligência Artificial

A inteligência artificial (IA) é um campo da ciência da computação que se concentra no desenvolvimento de sistemas e algoritmos capazes de realizar tarefas que normalmente exigiriam inteligência humana. Essas tarefas incluem o aprendizado, o raciocínio, o reconhecimento de padrões, a resolução de problemas complexos e a tomada de decisões (COSSETTI, 2018).

Ela utiliza uma variedade de técnicas e abordagens, sendo o aprendizado de máquina uma das mais proeminentes. Nesse contexto, os computadores são treinados com grandes conjuntos de dados para aprender a identificar padrões e fazer previsões ou tomar decisões com base nesses padrões. As redes neurais artificiais, que se inspiram no funcionamento do cérebro humano, são um exemplo de tecnologia frequentemente usada para esse fim (FIA, 2023).

Além disso, a IA tem aplicações amplas e impactantes em diversos setores, como saúde, finanças, transporte, manufatura, entretenimento e muitos outros. Exemplos notáveis incluem assistentes virtuais, carros autônomos, diagnósticos médicos baseados em IA e sistemas de recomendação em plataformas de streaming.

2.2.1 Rede Neural Perceptron Multi-Camadas (MLP)

Redes neurais artificiais (RNAs) são modelos computacionais inspirados no funcionamento do cérebro humano. Elas são um componente central na área de aprendizado de máquina e inteligência artificial. As RNAs são compostas por camadas de unidades chamadas neurônios artificiais ou unidades de processamento. Cada neurônio recebe entradas, realiza cálculos e gera uma saída que é transmitida para a próxima camada de neurônios (ACADEMY, 2022).

O funcionamento das RNAs é caracterizado pelo aprendizado supervisionado, onde são alimentados com um conjunto de dados de treinamento que inclui entradas e saídas desejadas. Durante o treinamento, a RNA ajusta seus pesos e parâmetros internos para minimizar a diferença entre as saídas calculadas e as saídas desejadas. Isso permite que as RNAs generalizem e façam previsões ou classifiquem novos dados com base no que aprenderam durante o treinamento (ACADEMY, 2022).

As RNAs podem ter várias camadas, dando origem a redes neurais profundas, também conhecidas como redes neurais profundas (*deep neural networks* - DNNs). Essas DNNs são notáveis por sua capacidade de representar e aprender características complexas e abstratas em dados, tornando-as especialmente eficazes em tarefas como reconhecimento de imagem, processamento de linguagem natural e jogos.

Além das DNNs, existem muitas arquiteturas de RNAs especializadas, como redes neurais convolucionais (CNNs) para processamento de imagens e redes neurais recorrentes (RNNs) para sequências de dados temporais. Cada uma delas é adaptada para tarefas específicas.

Estrutura das Redes Neurais MLPs

Neurônios Artificiais

Os neurônios artificiais, também conhecidos como perceptrons ou unidades de

processamento, são a unidade fundamental nas redes neurais MLPs. Cada neurônio é projetado para simular, de maneira simplificada, o funcionamento de um neurônio biológico. Eles desempenham um papel essencial no processamento de informações em redes neurais, sendo responsáveis por realizar operações matemáticas nas entradas que recebem (TECH, s.d.).

Cada neurônio artificial recebe um conjunto de entradas, que podem representar dados ou informações vindas de neurônios anteriores na rede. O neurônio, em seguida, aplica uma função de ativação a essas entradas ponderadas para produzir uma saída. Os pesos associados a cada entrada são ajustáveis e desempenham um papel crítico na determinação da importância das diferentes entradas para a saída do neurônio (TECH, s.d.).

Camadas

As redes neurais MLPs têm pelo menos três camadas: a camada de entrada, camada oculta e a camada de saída.

A camada de entrada é a primeira camada da rede neural e recebe os dados de entrada que serão processados pela rede. Esses dados podem ser representações numéricas de características, como pixels de uma imagem, valores de sensores, palavras de um texto, etc. A principal função da camada de entrada é passar os dados de entrada para as camadas ocultas da rede, onde ocorrerão as operações de processamento. O número de neurônios na camada de entrada geralmente corresponde à dimensionalidade dos dados de entrada. Cada neurônio representa uma característica específica dos dados (ACADEMY, 2022).

As camadas ocultas são o coração da rede neural e desempenham um papel crucial no processamento e na extração de características dos dados de entrada. Essas camadas recebem as entradas da camada anterior (ou de camadas anteriores, no caso de redes profundas) e aplicam transformações lineares e não lineares a esses dados. As transformações lineares são realizadas por meio de pesos associados a cada conexão entre os neurônios das camadas, e as transformações não lineares são introduzidas por funções de ativação, como a função ReLU (*Rectified Linear Unit*). O número e a arquitetura das camadas ocultas podem variar dependendo do problema e da complexidade da tarefa. Redes profundas (com várias camadas ocultas) são capazes de aprender representações mais complexas dos dados (ACADEMY, 2022).

A camada de saída é a última camada da rede neural e produz a saída final da rede, que pode ser um valor contínuo (regressão) ou uma probabilidade associada a diferentes classes (classificação). A quantidade de neurônios na camada de saída depende da natureza da tarefa. Por exemplo, em um problema de classificação binária, geralmente há um neurônio de saída, enquanto em um problema de classificação multiclasse, há um neurônio para cada classe. A função de ativação na camada de saída pode variar de acordo com a

tarefa. Por exemplo, em problemas de classificação binária, pode ser usada uma função sigmoide, enquanto em problemas de classificação multiclasse, é comum usar uma função softmax para calcular as probabilidades das diferentes classes (ACADEMY, 2022).

Segue na figura 1 o exemplo da estrutura de uma rede neural MLP.

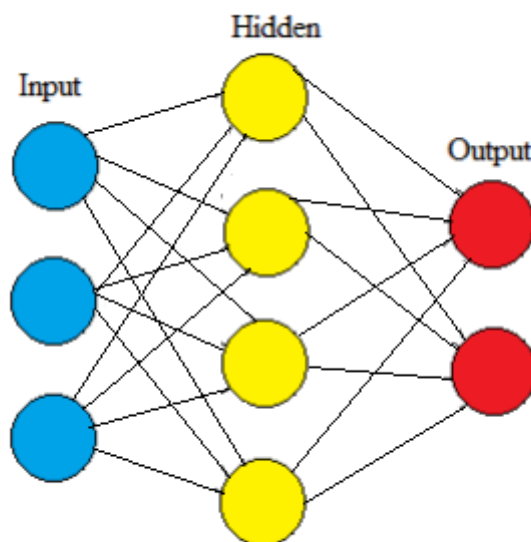


Figura 1 – Estrutura de uma rede neural. Fonte: Autor

Conexões Ponderadas

Cada conexão ponderada entre neurônios é um componente crucial nas redes neurais, pois esses pesos determinam a força e a importância de cada relação. Durante o treinamento de uma rede neural, o objetivo é encontrar os valores ideais para esses pesos, de modo a otimizar o desempenho da rede em uma tarefa específica (JR., s.d.).

A otimização dos pesos é geralmente realizada por meio de algoritmos como o gradiente descendente. Esses algoritmos analisam a diferença entre as previsões da rede e os valores desejados (ou rótulos) nos dados de treinamento. Com base nessa diferença, os pesos das conexões são ajustados de forma a minimizar o erro. Esse processo é repetido ao longo de muitas iterações até que a rede alcance um desempenho aceitável na tarefa, tornando-se capaz de fazer previsões precisas.

Os pesos das conexões ponderadas desempenham um papel fundamental na capacidade da rede de aprender e generalizar a partir dos dados de treinamento. Ajustar os pesos corretamente é um processo delicado, e a rede deve equilibrar a capacidade de se ajustar aos dados de treinamento sem perder a capacidade de generalização para novos dados não vistos. Quando os pesos são ajustados de maneira eficaz, a rede é capaz de capturar padrões e relações complexas nos dados, tornando-a útil em uma ampla gama de tarefas, desde reconhecimento de fala até processamento de imagens e previsões financeiras.

Essas conexões são demonstradas na figura acima (Figura 1), onde cada aresta na imagem possui um peso de acordo com sua importância no modelo.

Função de Ativação

Cada neurônio aplica uma função de ativação às suas entradas ponderadas para determinar sua saída. Funções comuns incluem a função sigmoide 2.1, que também é conhecida como função logística, é uma função matemática amplamente utilizada em diversos campos, incluindo a matemática, a estatística e o aprendizado de máquina (ACADEMY, 2022).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.1)$$

Nesta equação, x é a variável de entrada da função sigmoide, e é a constante matemática conhecida como número de Euler. A função sigmoide tem várias propriedades importantes, suas principais são:

- Limitação dos valores de saída: A função sigmoide mapeia qualquer valor de x para um valor no intervalo de 0 a 1. Isso significa que, independentemente do valor de entrada, a saída da função sigmoide estará sempre contida no intervalo $[0, 1]$.
- Comportamento suave e diferenciável: A função sigmoide é suave e diferenciável em todo o seu domínio. Isso a torna adequada para uso em algoritmos de otimização, como os usados no treinamento de redes neurais artificiais.
- Problemas de regressão logística: A função sigmoide é fundamental na regressão logística, um modelo estatístico amplamente utilizado para problemas de classificação binária.

A função ReLU (Rectified Linear Unit) é outra função de ativação amplamente utilizada em redes neurais e outras arquiteturas de aprendizado de máquina. Sua forma geral é simples, como mostrado na equação 2.2.

$$f(x) = \max(0, x) \quad (2.2)$$

Em outras palavras, a função ReLU retorna x se x for positivo e zero caso contrário. Aqui estão algumas das principais características da função ReLU:

- Simplicidade: A função ReLU é computacionalmente eficiente e fácil de implementar, pois envolve apenas uma operação de comparação e uma operação de retorno máximo.

- **Aprendizado de representações mais discriminativas:** A não-linearidade introduzida pela função ReLU ajuda a rede a aprender representações mais discriminativas dos dados, o que pode melhorar o desempenho em tarefas de aprendizado de máquina.
- **Esparsidade:** A função ReLU gera ativações esparsas, pois muitas das saídas são zero. Isso pode tornar a rede mais eficiente computacionalmente, economizando recursos.

Por fim, temos a função linear 2.3. Às vezes chamada de função de identidade, é uma das funções mais simples em matemática e não introduz não-linearidade em seus dados.

$$f(x) = x \tag{2.3}$$

A função linear basicamente retorna o valor de entrada x sem fazer nenhuma transformação. Aqui estão algumas características importantes da função linear:

- **Simplicidade extrema:** A função linear é a mais simples entre todas as funções de ativação. Ela não realiza nenhum tipo de transformação nos dados de entrada.
- **Uso em camadas de saída:** Embora a função linear seja raramente usada como função de ativação nas camadas ocultas de uma rede neural. Ela é comumente usada na camada de saída de modelos de regressão linear simples, onde o objetivo é prever valores numéricos contínuos.

Em conclusão, as redes neurais artificiais representam uma poderosa ferramenta na área de aprendizado de máquina, devido à sua capacidade de emular aspectos do funcionamento do cérebro humano. Sua versatilidade e capacidade de generalização tornam-se essenciais para uma ampla gama de aplicações em diversos setores, impulsionando a pesquisa e o desenvolvimento contínuo da inteligência artificial e do aprendizado de máquina. À medida que essas redes continuam a evoluir e se aprimorar, espera-se que desempenhem um papel ainda mais significativo na solução de problemas complexos e na criação de sistemas cada vez mais inteligentes.

2.2.2 Redes Neurais de Longa Memória de Curto Prazo:

As Redes Neurais Recorrentes (RNNs) são uma classe especial de redes neurais profundas projetadas para lidar com dados sequenciais ou temporais. Diferentemente das redes neurais tradicionais, as RNNs têm conexões recorrentes que lhes permitem manter uma memória interna, tornando-as especialmente adequadas para tarefas que envolvem sequências, como processamento de linguagem natural (NLP), séries temporais, reconhecimento de fala e muito mais. No entanto as RNNs podem sofrer de problemas

de gradiente estourando durante o treinamento devido às conexões recorrentes. Isso pode dificultar o treinamento de RNNs profundas. Além disso, as RNNs têm memória limitada, o que significa que podem ter dificuldade em lembrar informações relevantes de sequências longas (ACADEMY, 2022).

As Redes Neurais de Longa Memória de Curto Prazo (LSTMs) são uma variante especializada de redes neurais recorrentes (RNNs) projetadas para superar as limitações das RNNs tradicionais no processamento de sequências longas e temporais. As LSTMs foram introduzidas para capturar e aprender relações de longo prazo em dados sequenciais, o que as torna eficazes em uma ampla gama de aplicações.

Estrutura das Redes LSTM

Unidade LSTM

O bloco de construção básico de uma LSTM é a unidade LSTM. Ela é composta por três portões principais: o portão de entrada (*input gate*), o portão de esquecimento (*forget gate*) e o portão de saída (*output gate*). Esses portões controlam o fluxo de informações na unidade LSTM, permitindo a aprendizagem de dependências a longo prazo (ACADEMY, 2022).

A rede LSTM é composta por várias unidades LSTM, também conhecidas como células LSTM, empilhadas em camadas. Cada uma das portas das unidades LSTM possuem suas características, elas estão detalhadas:

- Porta de Entrada (*Input Gate*): Esta porta controla a informação que entra na célula LSTM. Ela decide quais informações da entrada atual devem ser armazenadas na célula.
- Porta de Esquecimento (*Forget Gate*): Esta porta decide quais informações na célula LSTM devem ser esquecidas ou descartadas. Isso ajuda a manter a relevância das informações ao longo do tempo.
- Porta de Saída (*Output Gate*): A porta de saída determina qual parte das informações na célula LSTM será transmitida como saída do passo atual.

Na figura 2 é representada esta estrutura.

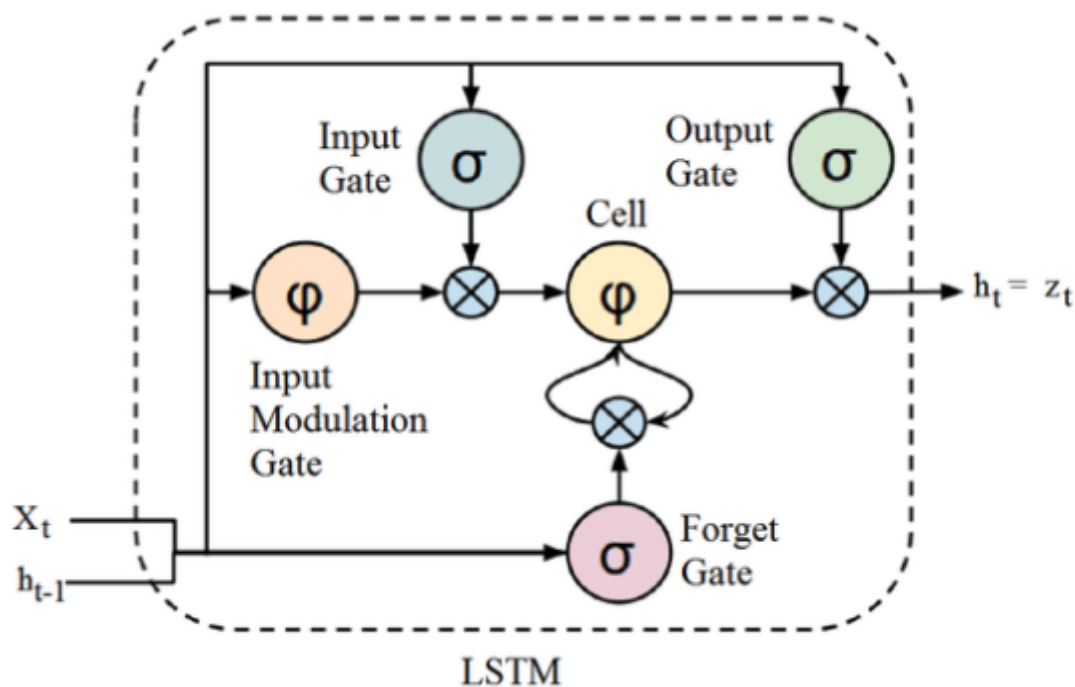


Figura 2 – Estrutura de uma unidade LSTM. Fonte: (ACADEMY, 2022)

Memória de Célula

Uma característica fundamental das LSTMs é a presença de uma memória de célula (*cell state*), representada na figura 2 que pode armazenar informações ao longo do tempo e permite que informações relevantes sejam mantidas e propagadas através de sequências longas.

Propagação no Tempo

Semelhante às Redes Neurais Recorrentes (RNNs), as Redes LSTM (*Long Short-Term Memory*) também se propagam no tempo para processar sequências passo a passo. No entanto, o que diferencia as LSTMs das RNNs tradicionais é a maneira como elas lidam com a passagem do tempo e atualizam suas informações internas (COSTA, 2019).

As LSTMs têm a capacidade de aprender e manter memórias de longo prazo em sua memória de célula. Isso permite que elas capturem relações de dependência a longo prazo nas sequências. Ao se propagar no tempo, uma LSTM atualiza tanto seu estado oculto quanto sua memória de célula com base na entrada atual e nas informações retidas a partir de etapas anteriores.

O estado oculto representa o que a rede sabe sobre o passado até o momento atual, enquanto a memória de célula é responsável por reter e atualizar informações relevantes

para o futuro da sequência. Essa combinação de mecanismos permite que as LSTMs controlem o fluxo de informações ao longo das etapas temporais, decidindo o que manter e o que descartar, e, assim, lidar com desafios, como o desvanecimento do gradiente, que é um problema comum em RNNs tradicionais.

Empilhamento de Camadas

A estrutura típica de uma rede LSTM envolve empilhar várias camadas de células LSTM umas sobre as outras, permitindo que a rede capture representações cada vez mais complexas da sequência de entrada. A saída da última camada LSTM é geralmente alimentada em uma camada densa (fully connected) ou em outra estrutura de saída adequada para a tarefa específica (COSTA, 2019).

Concluindo, as Redes Neurais de Longa Memória de Curto Prazo (LSTMs) representam uma evolução significativa das redes neurais recorrentes tradicionais e são altamente eficazes na modelagem de sequências temporais complexas. Sua capacidade de capturar informações de longo prazo as torna uma ferramenta poderosa para uma ampla gama de aplicações em ciência de dados e aprendizado de máquina. Compreender o funcionamento das LSTMs é fundamental para explorar seu potencial em seu TCC e em projetos futuros relacionados a dados sequenciais.

2.2.3 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (CNNs) são uma classe de redes neurais profundas projetadas para processar dados que têm uma estrutura de grade, como imagens e vídeos. Elas revolucionaram a visão computacional e têm sido amplamente adotadas em diversas aplicações de reconhecimento de padrões (ACADEMY, 2022).

As CNNs são capazes de aprender automaticamente características que são invariantes a translações, tornando-as ideais para tarefas de visão computacional, onde a posição dos objetos em uma imagem pode variar.

Elas também são capazes de aprender características em diferentes níveis de abstração, permitindo a detecção de características simples, como bordas, até características mais complexas, como texturas e objetos. Além disso, embora tenham sido desenvolvidas originalmente para tarefas de visão computacional, as CNNs também podem ser usadas com sucesso na identificação de padrões em uma variedade de tipos de dados, não se restringindo exclusivamente a imagens e vídeos. Tendo espaço em diferentes áreas como: análise de áudio, processamento de linguagem natural e identificação de padrões em séries temporais (RODRIGUES, 2018).

Estrutura das Redes CNN

Camadas Convolucionais

As Redes Neurais Convolucionais (CNNs) empregam um projeto fundamental baseado em camadas convolucionais. Essas camadas aplicam filtros ou kernels a pequenas regiões da entrada, realizando uma operação de convolução. Esse processo é crucial para extrair características relevantes das imagens ou conjuntos de dados.

Os filtros são matrizes ponderadas que deslizam pela entrada, permitindo a detecção de padrões locais em diferentes partes da imagem. Essa técnica é especialmente poderosa, pois as CNNs são capazes de aprender automaticamente quais padrões e características são importantes para a tarefa em questão. Por exemplo, durante o treinamento, uma CNN pode aprender a identificar bordas, texturas, formas e padrões, dependendo das demandas da aplicação (RODRIGUES, 2018).

O uso de camadas convolucionais permite que as CNNs capturem características hierárquicas e locais. Isso significa que as camadas iniciais detectam características mais simples, enquanto as camadas mais profundas compõem essas características em padrões mais complexos. Por exemplo, uma CNN pode detectar padrões na camada inicial, combinar essas padrões para identificar formas e, em camadas ainda mais profundas, combinar essas formas para reconhecer padrões completos.

Camadas de Pooling

Após as camadas convolucionais, as Redes Neurais Convolucionais (CNNs) frequentemente incorporam camadas de pooling, que desempenham um papel essencial no processamento de imagens e na extração de características mais significativas das representações geradas pelas camadas convolucionais (ACADEMY, 2022). Podemos ver um exemplo de funcionamento dessa camada na figura 3.

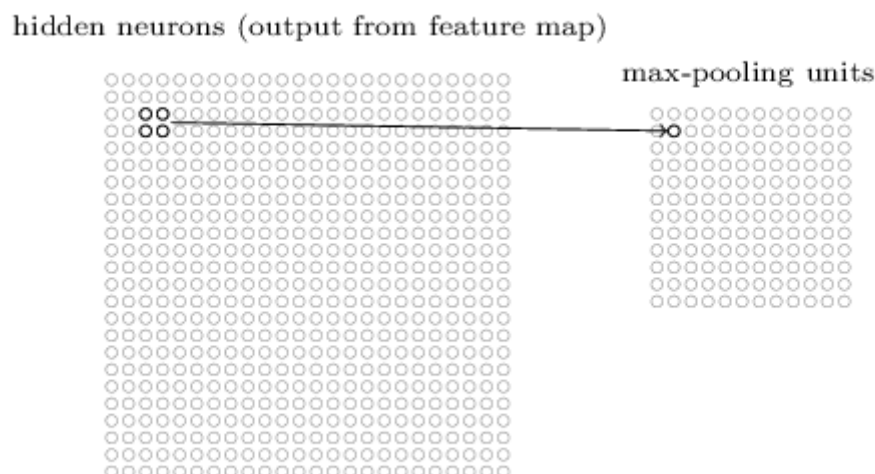


Figura 3 – Camada de Pooling. Fonte: (ACADEMY, 2022)

As camadas de pooling, em geral, operam reduzindo a dimensionalidade dos mapas

de características obtidos após a convolução. Isso é alcançado por meio de operações como o max-pooling, média-pooling ou outras variantes. O max-pooling, por exemplo, seleciona o valor máximo de um grupo de valores em uma região local da representação da imagem. Isso ajuda a preservar as características mais relevantes da imagem, reduzindo a quantidade de informações redundantes (RODRIGUES, 2018).

Essa redução de dimensionalidade é vantajosa por várias razões. Primeiro, ela reduz a carga computacional durante o treinamento e a inferência, o que torna as CNNs mais eficientes em termos de recursos. Em segundo lugar, o pooling ajuda a tornar as representações mais invariantes, o que é crucial para o sucesso da rede em tarefas de classificação e detecção (RODRIGUES, 2018).

Além disso, as camadas de pooling também auxiliam na obtenção de características mais abstratas e gerais a partir das características extraídas pelas camadas convolucionais. Isso é fundamental, pois as representações mais abstratas costumam ser mais úteis para tarefas de alto nível, como identificação de objetos ou reconhecimento de padrões complexos.

Camadas Totalmente Conectadas

As camadas totalmente conectadas, também conhecidas como camadas densas, desempenham um papel crítico nas redes neurais convolucionais (CNNs). Elas são responsáveis por agregar informações das camadas anteriores e são frequentemente usadas para realizar a classificação final ou fazer previsões com base nas características extraídas das camadas convolucionais e de pooling.

Essas camadas são compostas por neurônios que estão conectados a todos os neurônios da camada anterior. Essa conectividade total permite que as informações das características extraídas anteriormente sejam combinadas e processadas para gerar uma representação final que é utilizada para realizar a tarefa específica da rede, como classificar uma imagem em categorias ou fazer previsões com base em dados de entrada (RODRIGUES, 2018).

Por fim, as Redes Neurais Convolucionais (CNNs) são uma ferramenta fundamental na visão computacional e em várias outras áreas. Sua capacidade de aprender características hierárquicas a partir de dados de grade torna-as altamente eficazes na extração de informações.

2.2.4 Autoencoders

Os Autoencoders constituem uma categoria especial de redes neurais amplamente aplicadas em tarefas de aprendizado não supervisionado, e destacam-se particularmente na redução de dimensionalidade e extração de características. A arquitetura dos Autoencoders é projetada com o propósito de comprimir informações em uma representação concisa, conhecida como código latente no qual informações cruciais são encapsuladas. Em seguida,

por meio de um processo de decodificação, essa representação latente é transformada de volta aos dados originais, permitindo que informações relevantes sejam recuperadas com uma perda mínima (ACADEMY, 2022).

Portanto, os Autoencoders desempenham um papel fundamental na análise de dados, na eliminação de redundâncias e na extração de características significativas, facilitando a compreensão e o processamento de informações complexas em várias aplicações, como reconhecimento de padrões, compressão de dados e geração de imagens.

Estrutura dos Autoencoders

Segue na figura 4 o funcionamento do autoencoder.

Inicialmente, os dados de entrada são inseridos na rede. O encoder é responsável por comprimir os dados de entrada em uma representação de menor dimensionalidade chamada de código latente. Ele consiste em camadas que gradualmente reduzem a dimensionalidade, capturando características importantes dos dados.

O código latente é uma representação compacta e rica em características dos dados originais, onde informações essenciais são encapsuladas. A fase do decoder é simétrica ao encoder e é responsável por reconstruir os dados de entrada a partir do código latente. Ele utiliza camadas que expandem gradualmente a dimensionalidade, restaurando os dados de forma a se aproximarem dos dados de entrada originais.

Por fim, os dados de saída reconstruídos pelo decoder representam uma versão aproximada dos dados de entrada originais.

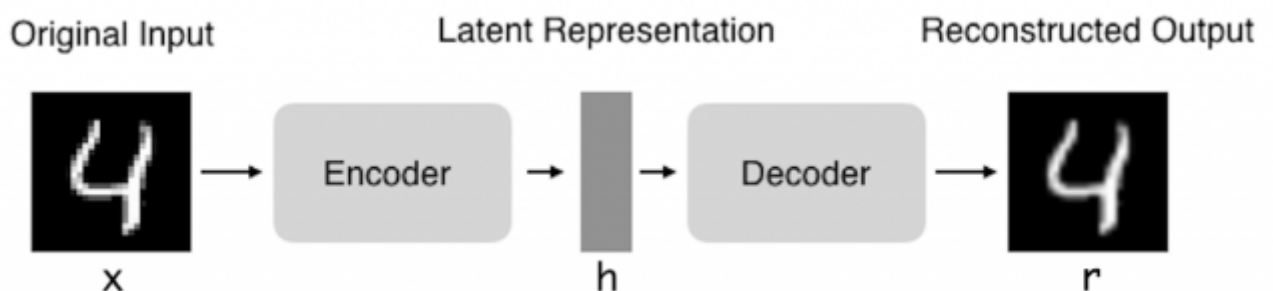


Figura 4 – Funcionamento Autoencoder. Fonte: (ACADEMY, 2022)

Encoder

A fase do encoder dentro da arquitetura de um autoencoder desempenha uma função crucial, concentrando-se na transformação dos dados de entrada em uma representação de menor dimensionalidade, frequentemente denominada como o código latente. Esta parte do

autoencoder é construída com camadas cuidadosamente projetadas que operam em conjunto para reduzir gradualmente a dimensionalidade dos dados originais. O processo ocorre em etapas sucessivas, com cada camada contribuindo para uma redução incremental no número de características representadas, resultando em um código latente mais compacto, onde informações essenciais são sintetizadas e mantidas de maneira eficaz (ACADEMY, 2022).

Essa etapa de codificação é um elemento crucial do autoencoder, pois permite a representação eficiente de dados complexos em um espaço de características mais gerenciável, proporcionando benefícios significativos em diversas aplicações, como reconhecimento de padrões, compactação de informações e aprendizado de representações.

Decoder

O decoder representa a segunda fase fundamental do autoencoder e desempenha o papel de reverter o processo anterior, ou seja, é encarregado de reconstruir os dados de entrada com base no código latente previamente criado. Essa parte do autoencoder, frequentemente concebida de forma simétrica em relação ao encoder, envolve camadas projetadas para gradualmente expandir a dimensionalidade do código latente, de modo a aproximar os dados de saída reconstruídos aos dados de entrada originais. Através de um processo de decodificação, essas camadas agem de maneira progressiva para desfazer a compressão realizada pelo encoder, restaurando a riqueza das informações contidas nos dados originais (ACADEMY, 2022).

Dessa forma, o decoder desempenha o papel de possibilitar que os autoencoders recuperem informações valiosas, tornando-os uma ferramenta versátil em tarefas como reconstrução de dados, geração de imagens e restauração de informações em várias aplicações de aprendizado de máquina e processamento de dados.

Função de Custo

A função de custo ou erro é usada para medir a diferença entre os dados de entrada e a saída reconstruída. O objetivo durante o treinamento do autoencoder é minimizar essa função de custo, fazendo com que o código latente capture as características mais importantes dos dados. Uma escolha comum é a Média do Erro Quadrático (Mean Squared Error - MSE), que é calculada como a média das diferenças ao quadrado entre os valores de entrada e os valores de saída (reconstruídos) para cada amostra de treinamento. A fórmula para esse cálculo está representada a seguir 2.4. Onde X_i representa uma amostra de entrada, $Enc(X_i)$ é a codificação gerada pela camada de codificação do autoencoder, $Dec(Enc(X_i))$ é a reconstrução da amostra de entrada e N é o número total de amostras de treinamento.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \text{Dec}(\text{Enc}(\mathbf{x}_i))\|^2 \quad (2.4)$$

Os Autoencoders são uma ferramenta poderosa para redução de dimensionalidade, remoção de ruído, geração de dados e extração de características. Eles são uma parte importante do campo de aprendizado profundo e têm aplicações significativas em várias áreas.

2.3 Regressão Linear

A regressão linear é uma técnica estatística fundamental amplamente utilizada para modelar a relação entre uma variável dependente (ou resposta) e uma ou mais variáveis independentes (ou preditores). Ela é uma das ferramentas mais simples e poderosas na análise estatística e tem uma ampla gama de aplicações em diversas áreas, incluindo ciências sociais, econômicas, biológicas, engenharia e muitas outras.

Além de sua simplicidade, a regressão linear é valiosa porque fornece interpretações intuitivas dos parâmetros do modelo, permitindo uma compreensão clara das relações entre variáveis. No entanto, é importante notar que, em cenários mais complexos, a regressão linear pode ser estendida para incluir múltiplas variáveis independentes ou ser combinada com outras técnicas estatísticas para modelar relações mais intrincadas. Assim, a regressão linear continua sendo uma base essencial para análise e modelagem de dados em muitos campos da ciência e da pesquisa.

Porém, para determinar se é apropriado usar regressão linear para resolver um problema específico, é importante considerar alguns critérios e pressupostos. A regressão linear é uma técnica estatística que pressupõe certas condições sobre os dados e a relação entre as variáveis (DAMÁSIO, s.d.). Aqui estão os principais critérios que um problema deve atender para que a regressão linear seja adequada:

- **Relação Linear:** A regressão linear assume que existe uma relação linear aproximada entre a variável dependente (a variável que você está tentando prever) e as variáveis independentes (variáveis preditoras). Isso significa que uma mudança em uma variável independente causa uma mudança proporcional na variável dependente.
- **Independência das Observações:** As observações (amostras) usadas no conjunto de dados devem ser independentes umas das outras. Isso significa que os valores de uma observação não devem depender dos valores de outras observações.
- **Homocedasticidade:** A variância dos erros (ou resíduos) deve ser constante em todas as faixas de valores das variáveis independentes. Em outras palavras, a dispersão

dos erros não deve aumentar ou diminuir à medida que as variáveis independentes mudam.

- **Normalidade dos Erros:** A distribuição dos erros (ou resíduos) deve ser aproximadamente normal. Isso significa que os erros devem seguir uma distribuição normal com média zero.
- **Linearidade dos Coeficientes:** Os coeficientes da regressão (os pesos atribuídos às variáveis independentes) devem ser lineares. Isso significa que as variáveis independentes devem ser combinadas linearmente para prever a variável dependente.
- **Ausência de Multicolinearidade:** As variáveis independentes não devem estar altamente correlacionadas entre si. A multicolinearidade pode dificultar a interpretação dos coeficientes e tornar os resultados instáveis.
- **Variáveis Numericamente Mensuráveis:** A regressão linear é mais apropriada para variáveis numéricas e mensuráveis. Embora seja possível usar regressão linear com variáveis categóricas, isso geralmente requer técnicas adicionais, como codificação de variáveis dummy.
- **Amostra Representativa:** A amostra de dados usada deve ser representativa da população que você está tentando fazer inferências. Amostras enviesadas podem levar a resultados não representativos.
- **Finalidade da Análise:** Considere a finalidade da análise. A regressão linear é adequada para previsão, modelagem de relações e análise de impacto, mas pode não ser a melhor escolha para todas as tarefas. Outros métodos, como árvores de decisão ou redes neurais, podem ser mais adequados para problemas específicos.

O modelo de regressão linear simples descreve a relação entre uma variável dependente (Y) e uma única variável independente (X) como uma linha reta. A equação de um modelo de regressão linear simples está representada na fórmula 2.5.

$$Y = \beta_0 + \beta_1 X + \epsilon \quad (2.5)$$

Onde Y é a variável dependente, X é a variável independente. B_0 é o intercepto (valor de Y quando X é zero), B_1 é o coeficiente de regressão (a inclinação da linha) e ϵ representa o erro aleatório (DAMÁSIO, s.d.).

Na imagem a seguir 5 vemos a distribuição das variáveis e a reta da regressão linear gerada. Sempre lembrando que alterando os valores de B_0 e B_1 determinamos diferentes retas a serem geradas.

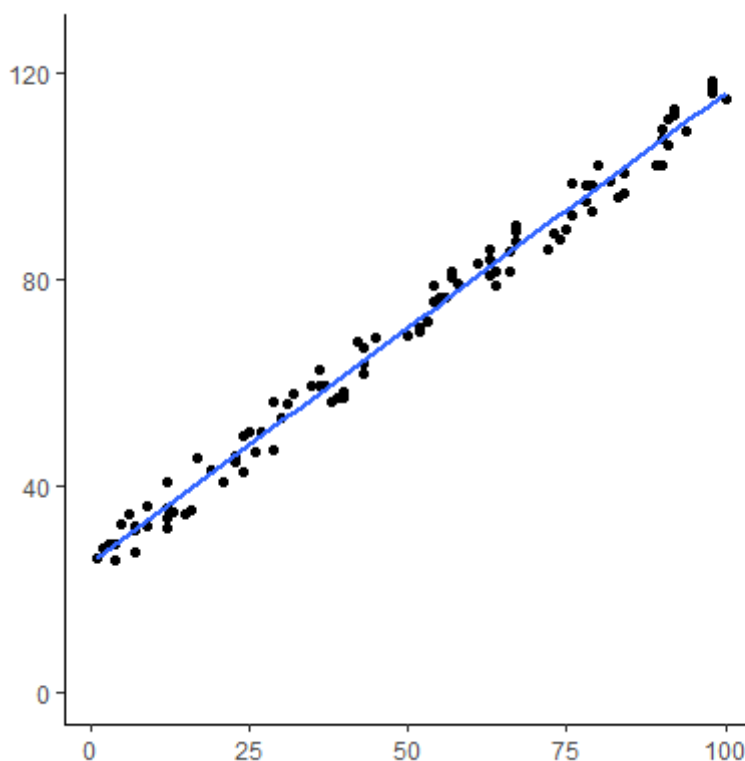


Figura 5 – Reta da Regressão Linear. Fonte: (DAMÁSIO, s.d.)

O objetivo da regressão linear é encontrar os valores de B_0 e B_1 que minimizam a soma dos quadrados dos resíduos ϵ entre os valores observados e os valores previstos pelo modelo.

A qualidade de um modelo de regressão linear é geralmente avaliada através de medidas como o coeficiente de determinação (R^2), que indica a proporção da variabilidade na variável dependente explicada pelas variáveis independentes.

Em resumo, a regressão linear é uma técnica poderosa para modelar relacionamentos entre variáveis e fazer previsões com base em dados observados. É amplamente usada na análise estatística e fornece *insights* valiosos em muitas áreas de pesquisa e aplicação. No entanto, é importante entender seus pressupostos e limitações ao aplicar a técnica. Já que a regressão linear assume uma relação linear entre as variáveis, o que pode não ser verdadeiro em todos os casos. Em algumas situações, modelos mais complexos podem ser necessários

3 Materiais e Métodos

Esta seção contém a metodologia escolhida para o desenvolvimento do presente trabalho e as técnicas e tecnologias utilizadas durante sua elaboração.

3.1 Materiais

No desenvolvimento deste trabalho foram utilizadas ferramentas de operação no mercado financeiro, linguagens de programação e bibliotecas. A seguir os materiais utilizados:

- Python: é uma linguagem de programação de alto nível, interpretada e de propósito geral. É conhecida por sua simplicidade e legibilidade de código, tornando-a ideal para iniciantes e utilizada em uma ampla variedade de aplicações, como desenvolvimento web, análise de dados, automação de tarefas e inteligência artificial. Python suporta uma vasta comunidade de desenvolvedores, oferece uma extensa biblioteca padrão e é multiplataforma, executando-se em sistemas operacionais como Windows, macOS e Linux (PYTHON, 2023). Utilizou-se a versão 3.10
- MQL5: MetaQuotes Language 5 é uma linguagem de programação utilizada principalmente para desenvolver sistemas de negociação automatizados (robôs) e indicadores personalizados na plataforma de negociação MetaTrader 5 (MT5). Ela oferece recursos avançados, como orientação a objetos, análise técnica avançada e acesso a dados de mercado em tempo real. Com o MQL5, os traders podem automatizar estratégias de negociação, criar indicadores personalizados e testar suas estratégias em dados históricos. É uma linguagem poderosa para traders e desenvolvedores que desejam personalizar e automatizar suas operações nos mercados financeiros (MetaQuotes Ltd, 2023a).
- Metatrader5: O MetaTrader 5 (MT5) é uma plataforma de negociação amplamente adotada, oferecendo recursos abrangentes para traders e investidores. Além de permitir a negociação em diversos mercados, incluindo forex, ações, commodities e criptomoedas, o MT5 suporta a criação de robôs de negociação e indicadores personalizados usando a linguagem de programação MQL5. Ele também oferece análise técnica avançada, ferramentas de análise fundamental, testador de estratégias e uma comunidade ativa onde os usuários podem acessar uma variedade de recursos e aplicativos. Com segurança de alto nível e execução rápida de ordens, o MT5 é uma escolha popular tanto para traders individuais quanto para corretores no mercado financeiro global (MetaQuotes Ltd, 2023b).

- Tensorflow: é uma biblioteca de código aberto para aprendizado de máquina e inteligência artificial desenvolvida pelo Google. Ela oferece uma estrutura flexível para a criação e treinamento de modelos de aprendizado profundo, como redes neurais, facilitando tarefas como classificação, reconhecimento de padrões, processamento de linguagem natural e muito mais. O TensorFlow é altamente escalável e suporta tanto CPUs quanto GPUs, o que o torna adequado para uma variedade de plataformas. Além disso, ele fornece ferramentas e recursos para visualização de dados, depuração de modelos e distribuição em ambientes de produção, tornando-o uma escolha popular para projetos de aprendizado de máquina e pesquisa em inteligência artificial ([TensorFlow, 2023](#)).

3.2 Métodos

Segue a lista dos métodos utilizados no trabalho:

- Desenvolvimento dos modelos de rede neural: para a aplicação foram desenvolvidos 4 modelos de redes neurais utilizando a linguagem de programação python. Foi utilizada a biblioteca do tensorflow para a implementação destes modelos, cada um com sua especialidade. Para o Autoencoder foi desenvolvido um modelo com a camada de entrada o encoder que codifica os dados de entrada e o decoder que decodifica essas informações para saída, a rede neural MLP é constituída por 3 camadas: entrada, oculta e de saída. Já para a LSTM são 1 uma camada de entrada 2 ocultas e 1 de saída. A rede convolucional é formada por uma camada de entrada convolucional, uma de pooling e a camada de saída.
- Treinamento das redes desenvolvidas: utilizou-se para treinamento dos modelos 70% dos dados obtidos e 30% para teste. O período utilizado para testes foi de 01/01/2018 até 01/01/2023. As *features* utilizadas para predição foram os dados mais utilizados para análise no mercado: preço de abertura, máxima do candle, mínima do candle e uma média móvel de 21 períodos. Além disso, o valor a ser previsto definido para os modelos foi o preço de fechamento do candle. As redes neurais foram treinadas no computador próprio com as seguintes configurações: 8GB de RAM, processador i5 7ª geração, placa de video integrada e um SSD 500GB.
- Desenvolvimento da estratégia de negociação utilizada para operar no mercado: a estratégia escolhida e desenvolvida funciona avaliando a informação prevista e comparando com o valor atual de mercado. Caso essa predição esteja pelo menos 250 pontos a cima do valor atual é feita uma operação de compra entendendo que o mercado tende a subir no mínimo 250 pontos no próximo candle. O mesmo é feito quando se trata de uma operação de venda, ao receber o resultado da predição e

verificar que a predição foi de pelo menos 250 pontos negativos é feita a venda no mercado esperando que o mercado caia pelo menos 250 pontos.

- Aplicação em python para enviar os resultados das predições para o Metatrader 5: na aplicação foi criado um socket em python que tem a função de um servidor ele recebe uma entrada com as *features* que foram utilizadas para treinar a rede neural e logo em seguida disponibiliza estas informações para o modelo para assim ser feita a predição. Com a predição realizada verifica-se qual será o resultado de acordo com a estratégia estabelecida e logo depois é enviado a resposta para o cliente que solicitou essa predição, esse resultado é um sinal de compra, venda ou não executar nenhuma operação.
- Aplicação em MQL5 que recebe as informações do python: essa aplicação tem o objetivo de obter os dados do Metatrader 5. Especificamente os dados referentes às *features* que foram utilizadas para treinar os modelos. Obtendo estas informações essa aplicação funciona como um cliente consumindo os serviços do servidor em python, estas informações adquiridas são enviadas para o servidor e ele processa esses dados retornando se é uma boa oportunidade de compra, venda ou se não é um bom momento para executar nem uma operação e de acordo com essa resposta do servidor é executada alguma operação na plataforma Metatrader 5.

3.3 Estrutura do Sistema

A estrutura da aplicação desenvolvida é dividida em duas partes, na primeira parte foi desenvolvido um servidor em python que tem o objetivo de processar os dados para predizer o preço de fechamento do próximo candle e avalia qual a melhor operação a ser tomada, compra, venda, ou manter sem alteração. Nesta parte é possível setar qual o modelo de rede a ser utilizado e o timeframe que será utilizado para aplicar a estratégia.

A segunda parte da aplicação é o cliente em MQL5, ele consome os serviços do servidor em python, envia para o servidor os valores atuais das *features* utilizadas no treinamento. Dada a resposta do servidor, a aplicação em MQL5 contata o Metatrader e é realizada a compra ou venda no mercado. Esta estrutura pode ser observada na figura 6.

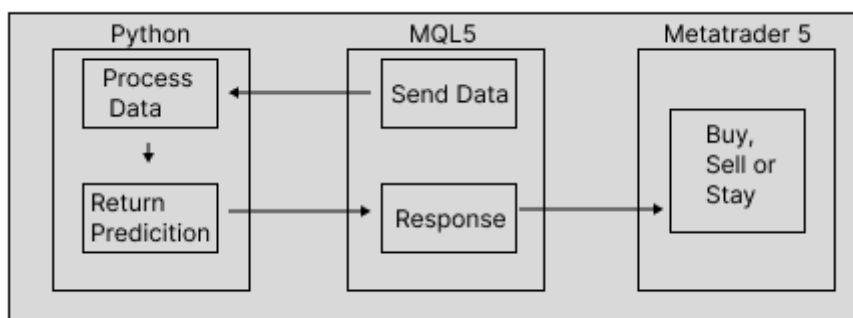


Figura 6 – Estrutura da Aplicação. Fonte: Autor

3.4 Servidor Python

Após treinados os modelos de rede neural podemos selecionar no servidor qual rede será utilizada e iniciar a aplicação. Nesta aplicação foi desenvolvida um servidor via socket apenas local, ou seja, não é possível utilizá-lo em outra máquina. Este servidor tem o papel de processar as informações recebidas, prever o valor de fechamento do próximo candle e por fim fazer uma validação de acordo com a estratégia que foi adotada. Esta estratégia funciona da seguinte maneira, é enviado um sinal de compra se o preço predito for pelo menos 250 pontos acima do atual ou de venda caso contrário. Se nem um desses cenários ocorrerem não é feita nenhuma operação.

3.5 Cliente MQL5

Em MQL5 foi feita a implementação do cliente, a aplicação que irá consumir os recursos do servidor. A função do cliente neste trabalho é obter e enviar as informações (features) que o servidor utiliza para fazer as previsões. Além disso, recebendo a resposta do servidor a aplicação executa a compra ou venda do ativo na plataforma Metatrader 5 atribuindo a essa operação os valores de take profit e stop loss de acordo com a banca do usuário. Vale lembrar que o servidor pode também retornar um valor nulo, se não for identificado nem uma oportunidade para alguma operação.

3.6 Avaliação dos Modelos

Utilizou-se um período de 6 meses operando no ativo Mini Índice para a avaliação dos modelos. As métricas escolhidas foram as comumente aplicadas para avaliação de *Expert Advisors* que podem ser vistas no fórum do MQL5 e em outras plataformas relacionadas a automatização de estratégias de investimento. Segue o detalhamento sobre elas:

Lucro: Diferença positiva entre a receita total gerada e as perdas.

DrawDown: Redução máxima do capital da conta de negociação a partir do saldo inicial até o nível mais baixo durante um determinado período de tempo

% Negociações com lucro: Porcentagem de negociações com lucro.

% Negociações com perda: Porcentagem de negociações com perdas.

Total de Negociações: Número total de negociações feitas durante o período de teste.

Fator de Lucro: Avalia a eficácia de um sistema de negociação automatizado ou estratégia de negociação. Ele fornece uma relação entre o lucro total obtido pelo EA e as perdas totais incorridas durante um determinado período de tempo. Um fator de lucro maior (>1) indica que o EA está gerando mais lucro do que perda.

Fator de Recuperação: O fator de recuperação é calculado como a relação entre o lucro acumulado e o maior rebaixamento (drawdown) durante um período de tempo específico. É uma métrica que avalia a eficácia da estratégia em recuperar perdas.

Máximo de Ganhos Consecutivos: Número de operações com lucro consecutivas.

Máximo de Perdas Consecutivas: Número de operações com prejuízo consecutivas.

4 Resultados e discussões

Nesta seção, serão apresentados os principais resultados da pesquisa realizada neste trabalho e inicia-se a discussão dos achados. Os resultados aqui apresentados foram obtidos por meio de análise dos dados coletados, que abrangem várias áreas, desde mercado financeiro até inteligência artificial. Esses resultados são cruciais para a compreensão do problema de pesquisa proposto e têm implicações significativas para identificação dos modelos que melhor se saem no mercado. Nas próximas seções, será aprofundado a interpretação desses resultados, relacionando-os aos objetivos do estudo e às teorias discutidas na revisão da literatura.

4.1 Rede Neural MLP

Analisando a rede neural MLP que utiliza três camadas (entrada, oculta e saída), sendo que a camada de entrada é formada por 64 neurônios totalmente conectados, a camada oculta 32 e a de saída apenas 1 neurônio. Foi feita a avaliação deste modelo durante o período de 6 meses informado, os resultados do teste da rede neural estão resumidos na tabela 3:

Lucro	DrawDown	% lucro	% perda	Total	Fator Lucro	Ganhos Seguidos	Perdas Seguidos	Fator Recuperação
950.0	-350.0	51.85	48.15	511	1.07	8	8	2.71

Tabela 3 – Tabela de resultados da rede neural MLP

Analisando o modelo em questão percebe-se que foi obtido um resultado positivo durante o período de teste, finalizando com 950 reais de lucro. Lembrando que como se trata de uma simulação não foi aplicado a esse resultado as taxas de operações que são cobradas ao fazer entradas no mercado.

O *Draw down* obtido pela estratégia foi de -350 reais, ou seja, a maior redução de capital do usuário que utilizou esse modelo durante esse período de teste foi de 350 reais. O modelo apresenta 51.85% de operações saindo com lucro e 48.15% de operações com prejuízo.

Durante esse período foram feitas 511 operações no mercado, o fator de lucro da estratégia é de 1.07, que não indica um resultado tão expressivo e o fator de recuperação é de 2.71. Além disso, este modelo obteve um total de 8 negociações com lucro consecutivas e 8 operações consecutivas com prejuízo.

Na figura 7, observa-se que a diferença entre as previsões e o valor real dos candles de fechamento não é tão expressiva. Porém, como se trata de um modelo de regressão

linear percebe-se que as previsões quando ocorre uma alteração na tendência do mercado são um pouco atrasadas.

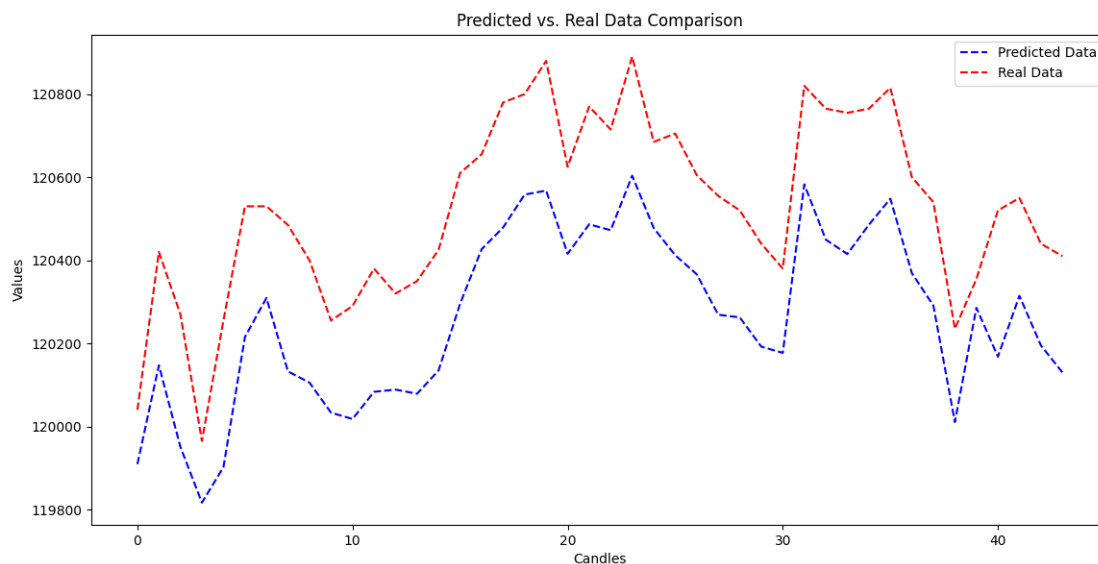


Figura 7 – Predito x Real - Rede Neural MLP (Zoom). Fonte: Autor

Em uma escala maior, com todo o período testado. É perceptível que mesmo com o pequeno atraso na predição conforme ocorre a mudança de tendência, as previsões seguem o sentido do mercado real. Como mostrado na figura 8

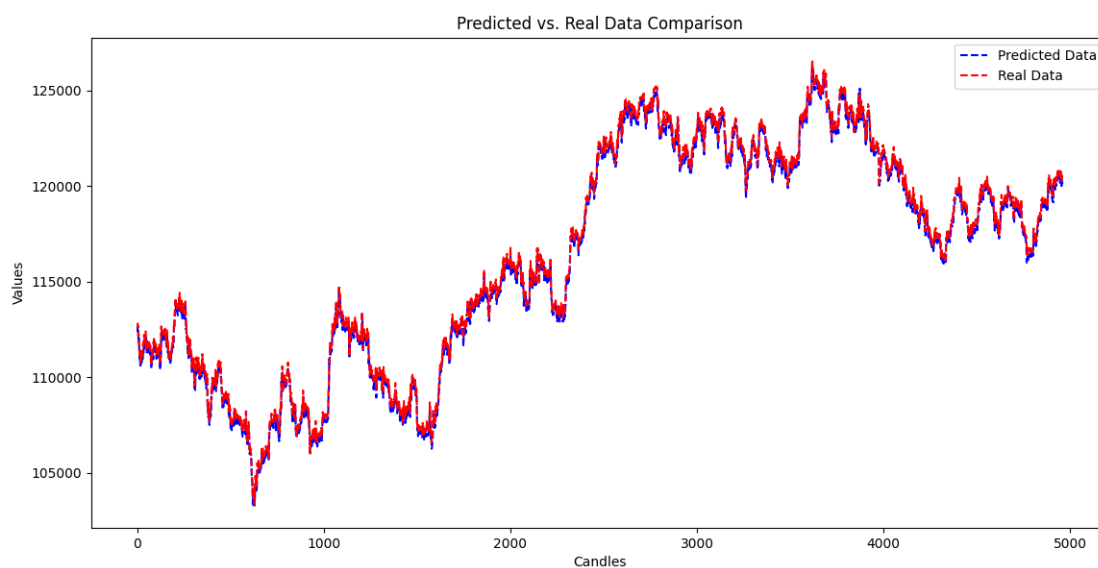


Figura 8 – Predito x Real - Rede Neural MLP. Fonte: Autor

4.2 Rede Neural LSTM

A rede neural LSTM é composta de 4 camadas, sendo que 3 delas são camadas LSTM. A camada de entrada com 64 neurônios, a próxima 32 e a última camada LSTM 16, por fim a camada de saída com apenas 1 neurônio. Podemos observar os resultados deste modelo na tabela 4.

Lucro	DrawDown	% lucro	% perda	Total	Fator Lucro	Ganhos Seguidos	Perdas Seguidos	Fator Recuperação
400.0	-550.0	51.04	48.96	384	1.04	6	8	0.72

Tabela 4 – Tabela de resultados da rede neural LSTM

Avaliando o modelo LSTM durante esse período foi obtido um resultado positivo, porém menor que o anterior. O lucro deste modelo de rede foi de 400 reais durante o período de 6 meses, com um *draw down* de -550 reais, ou seja, em algum momento o usuário teve um rebaixamento de 550 reais de seu capital.

A porcentagem de negociações com lucro foi de 51.05%, enquanto que as negociações com perda foram de 48.95%. No decorrer de 384 operações realizadas. O fator de lucro da estratégia é de 1.04, não indicando um grande resultado. O fator de recuperação utilizando este modelo está abaixo de 1 mostrando que a estratégia em questão com este modelo tem uma certa dificuldade para se recuperar de grandes perdas, além disso esse modelo possui um total de 6 ganhos consecutivos no mercado e 8 negociações com perda consecutivas.

Analisando a figura 9 é perceptível que as predições utilizando o modelo não se distanciam muito dos valores reais, porém, mesmo com essa precisão das predições o modelo não teve o melhor resultado.

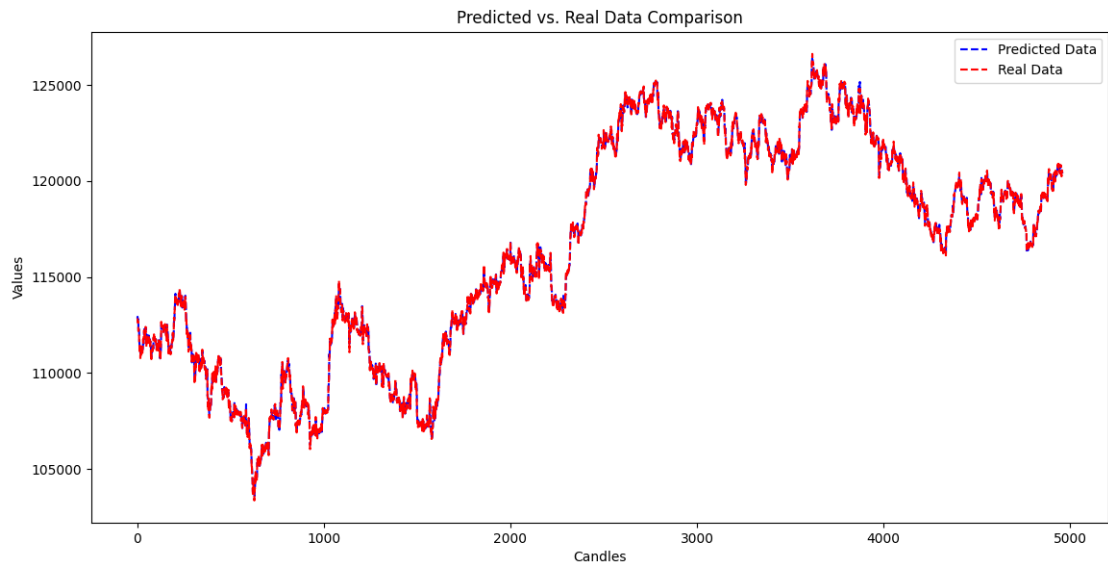


Figura 9 – Predito x Real - Rede Neural LSTM. Fonte: Autor

De mesma forma, reduzindo a escala em um intervalo de dias do periodo de teste. É evidente ver que as predições estão bem próximas da realidade como mostra a figura 10, contudo o modelo em questão aparenta não ter se saído muito bem com a estratégia estabelecida e os resultados não refletiram a precisão das predições.

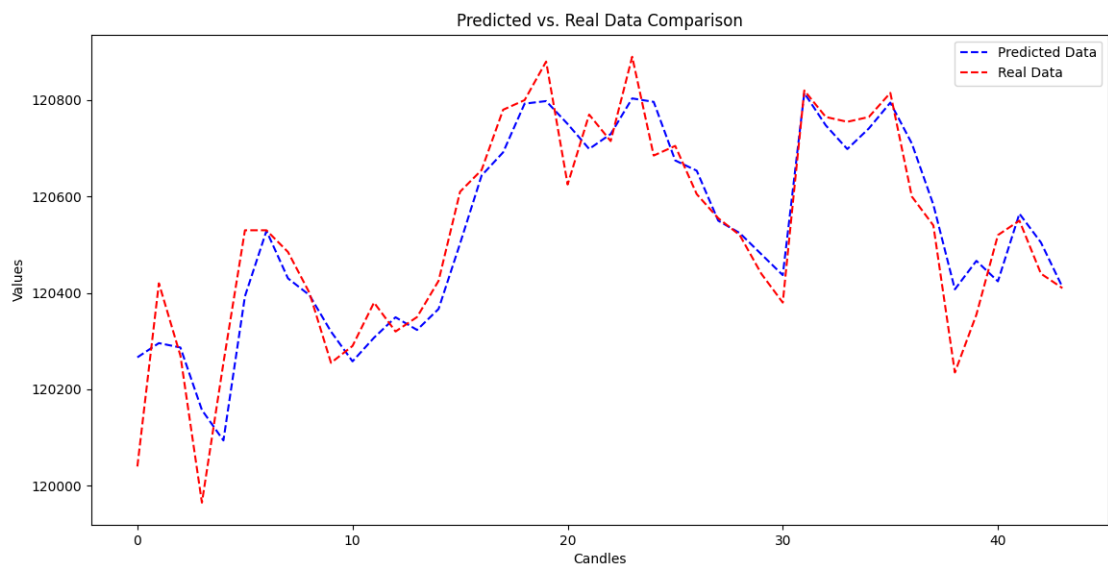


Figura 10 – Predito x Real - Rede Neural LSTM (Zoom). Fonte: Autor

4.3 Rede Neural Convolutacional

Neste modelo há no total 4 camadas, a primeira camada (camada de entrada) é a camada de convolução 1D e tem um total de 32 neurônios. A segunda camada é a de pooling 1D, ela realiza uma operação de pooling máxima global na saída da camada anterior. Ajudando a reduzir a dimensionalidade dos recursos extraídos, essa camada não possui nem um neurônio. A terceira camada é uma camada densa (totalmente conectada) que também possui 32 neurônios e por fim a camada de saída com apenas 1 neurônio. Os resultados obtidos por esse modelo são representados na tabela 5.

Lucro	DrawDown	% lucro	% perda	Total	Fator Lucro	Ganhos Seguidos	Perdas Seguidos	Fator Recuperação
1250.0	-700.0	52.9	47.1	431	1.12	8	8	1.78

Tabela 5 – Tabela de resultados da rede neural convolutacional

Percebe-se que o lucro obtido após o periodo de teste foi maior que o dos modelos anteriores, porém esse modelo se mostra um pouco mais volátil em relação aos anteriores, observando que o *draw down* chegou a -700 reais, dito isso, sabemos que em determinado momento o usuário chegou ao valor de 700 reais negativo.

A porcentagem de operações com lucro deste modelo chegou a 52.9% enquanto que a porcentagem das operações com prejuízo foram de 47.1%. O fator de lucro desta estratégia chegou no valor de 1.12, um valor ligeiramente melhor que o dos modelos vistos anteriormente e o fator de recuperação de 1.78 que é um valor considerado na média para os robôs de investimento em geral. Por fim, esse modelo apresentou um número de 8 operações com lucro seguidas e o mesmo valor para operações seguidas com prejuízo.

A figura 11 representa o gráfico da diferença entre o que foi previsto e o valor real dos candles. Podemos reparar que mesmo com os diferentes resultados finais, a precisão entre as predições e o valor real em cada modelo são bem próximas da realidade.

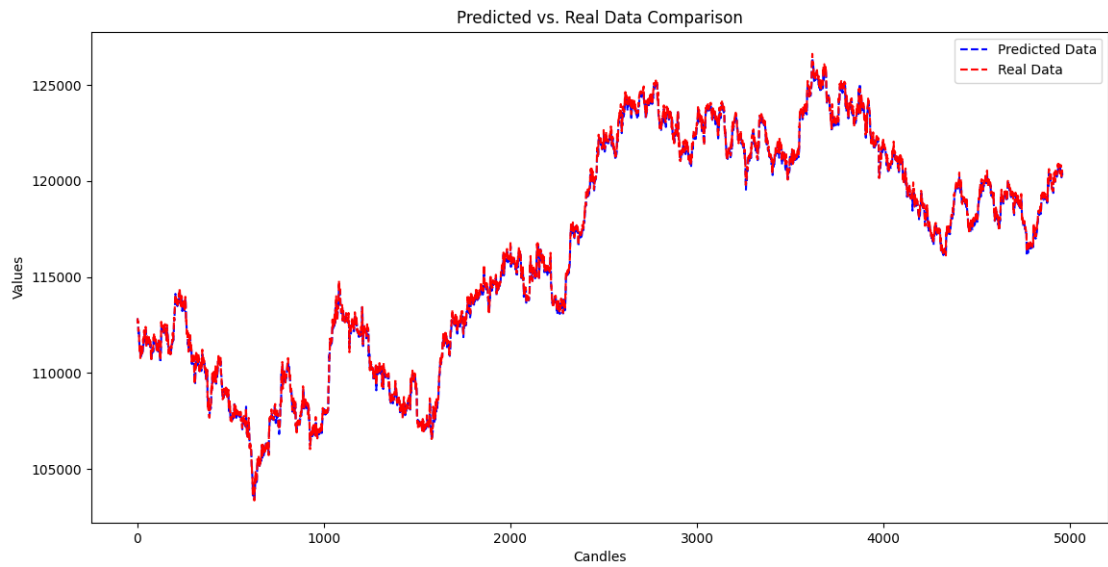


Figura 11 – Predito x Real - Rede Neural Convolutacional. Fonte: Autor

Em uma escala menor de alguns dias desse período testado observa-se a proximidade dos resultados previstos para o real (12), sempre tendo a questão de um pequeno atraso nas mudanças de tendencia do mercado porém é perceptível que as predições são sempre precisas e estando sempre bem próximas da realidade, o que contribui para os resultados positivos do modelo.

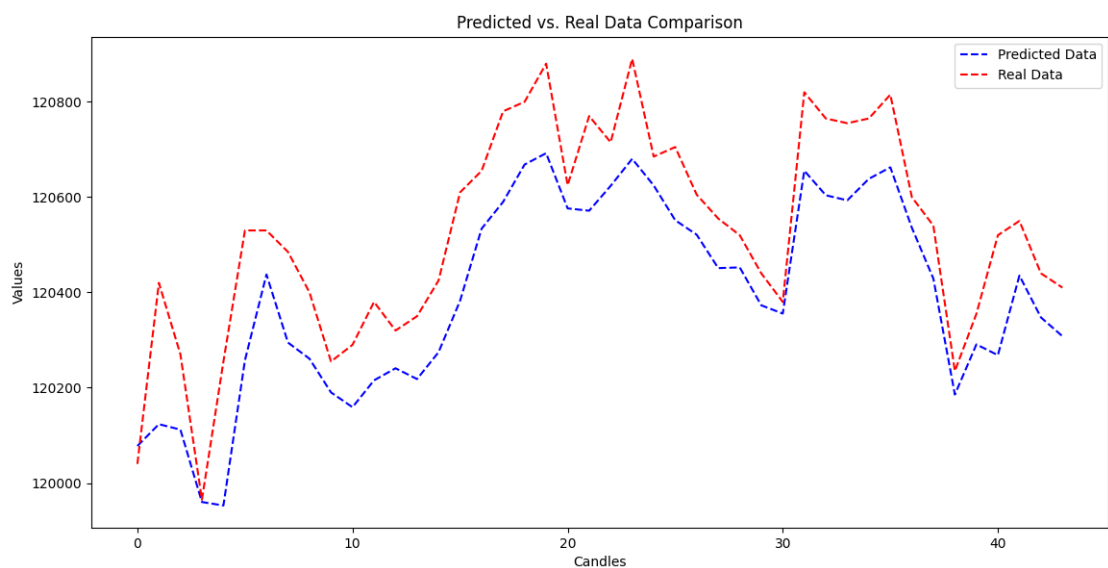


Figura 12 – Predito x Real - Rede Neural Convolutacional (Zoom). Fonte: Autor

4.4 Autoencoder

Este modelo implementado possui 3 camadas, a primeira camada representa a entrada da rede neural. A segunda camada deste modelo é densamente conectada e possui 32 neurônios, esta camada é responsável por mapear a entrada para um espaço de características mais compacto (codificação). Por fim, na terceira camada que também é densamente conectada e possui o número de neurônios igual ao número de recursos na entrada (32 neurônios), reconstrói a saída a partir da representação codificada e o resultado da predição é a saída reconstruída pelo autoencoder após passar por um processo de regressão linear. Segue os resultados obtidos por esse modelo na tabela 6.

Lucro	DrawDown	% lucro	% perda	Total	Fator Lucro	Ganhos Seguidos	Perdas Seguidos	Fator Recuperação
1250.0	-250.0	52.45	47.55	509	1.10	10	8	5

Tabela 6 – Tabela de resultados do Autoencoder

Percebe-se que também foi atingido um resultado positivo utilizando este modelo e observa-se que o lucro obtido foi idêntico ao lucro obtido utilizando o modelo analisado anteriormente. Porém é evidente que este modelo é mais constante que o modelo anterior, que apresenta um *draw down* de apenas -250 reais. Indicando que o maior rebaixamento de capital do usuário durante o período treinado foi de 250 reais.

A porcentagem de negociações com lucro obtida foi de 52.45% e a porcentagem de negociações com prejuízo foi de 47.55% em um total de 509 operações realizadas. O fator de lucro da estratégia foi de 1.10, não indicando uma grande eficiência do robo, porém o fator de recuperação desta estratégia é bem interessante, o valor de 5 no fator de recuperação indica que mesmo que sofra uma grande sequência de perdas essa estratégia utilizando este modelo de rede tem uma grande capacidade de recuperar o valor perdido. Ele alcançou uma sequência de 10 negociações com lucro seguidas e o número de 8 negociações com prejuízo seguidas.

Segue na figura 13 o resultado do que foi predito utilizando este modelo e o valor real do preço. Percebe-se o mesmo padrão anteriormente analisado, que mesmo se tratando de diferentes modelos e tendo diferentes resultados as predições continuam sempre bem próximas da realidade.

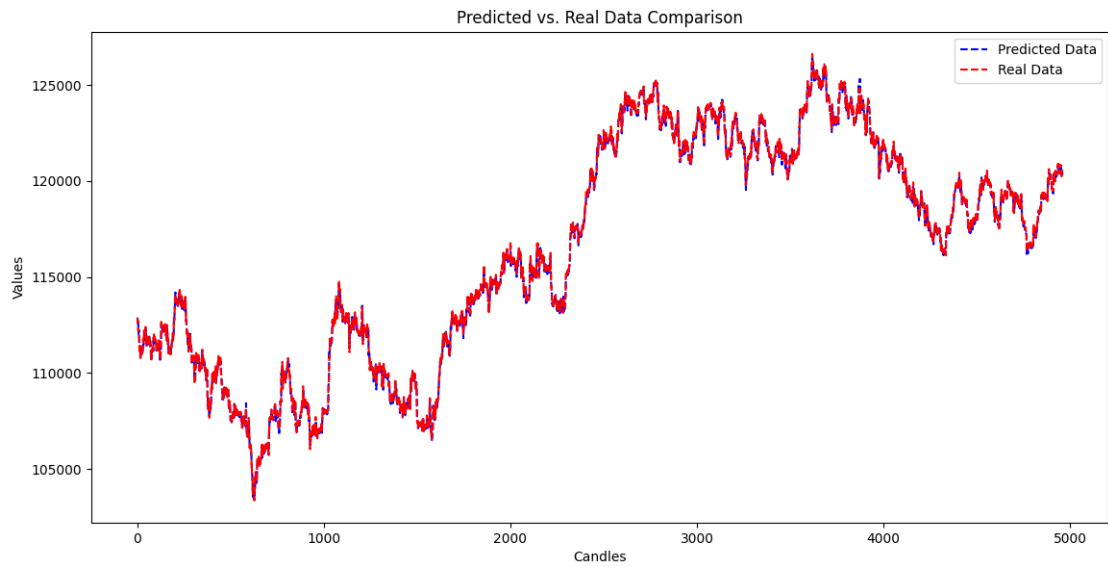


Figura 13 – Predito x Real - Autoencoder. Fonte: Autor

Diminuindo a escala destes resultados para alguns dias é evidente que da mesmo forma que os modelos anteriores os valores previstos, se comparados com os reais, estão muito próximos. Sempre um pouco abaixo dos valores reais mas é perceptível no gráfico que os valores previstos estão sempre acompanhando a mesma direção em que os valores reais estão. E isso abre a possibilidade de serem feitas as operações de acordo com os valores previstos. Segue a figura com estes resultados [14](#).

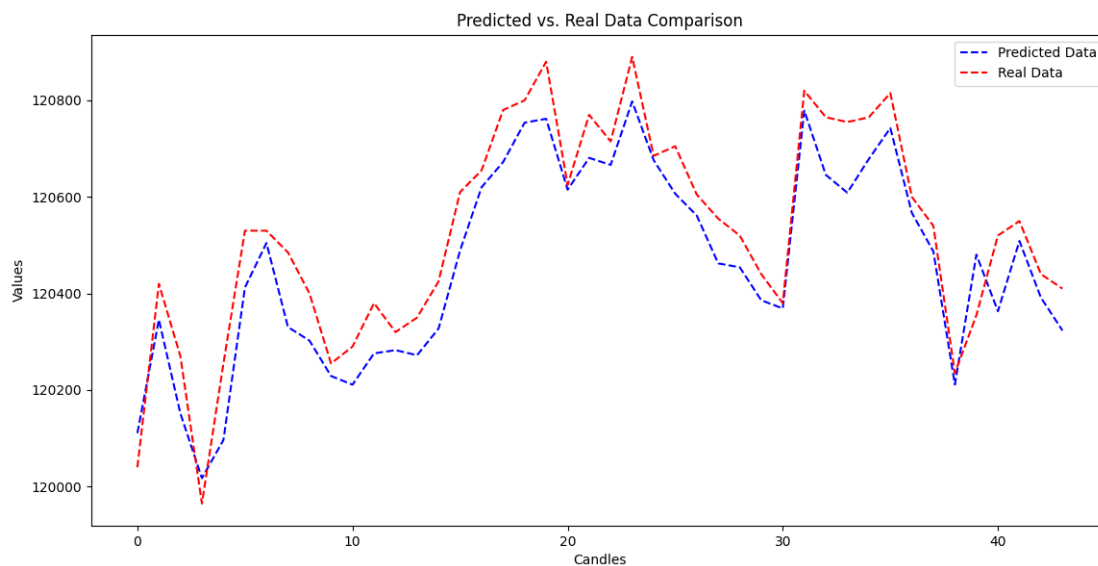


Figura 14 – Predito x Real - Autoencoder (Zoom). Fonte: Autor

4.5 Comparação Entre os Modelos

Analisando os quatro modelos de negociação, observa-se algumas diferenças e semelhanças em seus desempenhos. Todos os modelos apresentaram resultados positivos durante o período de teste, gerando lucros para os usuários. No entanto, há algumas nuances importantes a serem consideradas.

Em relação ao lucro obtido, o modelo de rede neural convolucional e o autoencoder tiveram resultados idênticos, ambos gerando um lucro de 1250 reais. Isso indica uma estabilidade nos resultados, mas, ao analisarmos o *Draw down*, vemos que o autoencoder teve um desempenho mais consistente, com um rebaixamento de capital significativamente menor (-250 reais) em comparação ao modelo de rede convolucional (-700 reais). Sugerindo que o autoencoder é menos arriscado e pode ser preferível para investidores que buscam uma abordagem mais segura.

Por outro lado, o modelo que utiliza a rede neural MLP obteve um valor de lucro ligeiramente inferior aos modelos citados acima. No entanto, ele apresentou um *Draw down* menor que o da rede convolucional (-350 reais), indicando menor volatilidade e risco associado.

A rede LSTM, embora tenha obtido um lucro menor (400 reais), teve um *Draw down* ainda mais significativo (-550 reais), o segundo maior rebaixamento obtido nos resultados. Tornando-o menos atraente em termos de consistência.

Na figura 15 é mostrada a contribuição de cada modelo para o resultado total

(100%) de acordo com algumas métricas utilizadas para avaliar as estratégias.

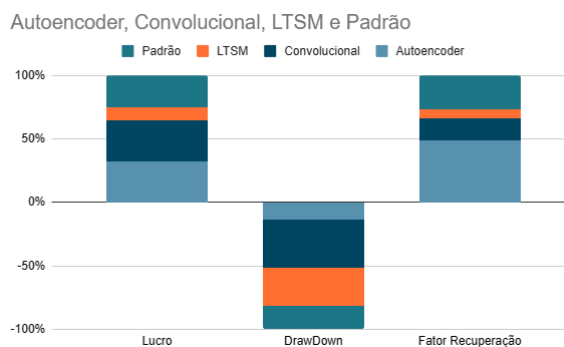


Figura 15 – Comparação entre os Modelos. Fonte: Autor

Analisando a imagem é evidente que o modelo que mais se destaca é o autoencoder. Possuindo um dos maiores valores de lucros durante o período testado, o menor rebaixamento e um fator de lucro consideravelmente maior que os outros modelos, ele claramente se destaca entre os demais. Além disso, fica claro que dado a estratégia utilizada no desenvolvimento a rede LSTM não se adaptou muito bem e não foi capaz de obter um resultado ao mesmo nível dos outros modelos.

A escolha do modelo ideal depende do perfil de risco e dos objetivos do investidor. Se a estabilidade é prioridade, o autoencoder pode ser uma escolha sólida. Para investidores dispostos a aceitar mais risco em busca de maiores lucros, o modelo convolucional pode ser uma opção a considerar. No entanto, é essencial considerar não apenas o lucro, mas também a volatilidade e o *Draw down* ao tomar decisões de investimento. Além disso, é importante lembrar que resultados passados não garantem o desempenho futuro, e a escolha de um modelo deve ser feita com base em uma análise completa e consideração dos fatores de risco.

4.6 Discussões sobre os resultados

Nesta seção de discussão sobre os resultados obtidos com os quatro modelos de negociação avaliados e testados, é fundamental analisar algumas considerações importantes. Os resultados positivos de lucro em todos os modelos indicam que, em um ambiente de teste, essas estratégias demonstraram a capacidade de gerar retornos financeiros. No entanto, a discussão a seguir se concentrará em aspectos-chave, como a consistência, o risco e o desempenho geral dos modelos.

Consistência versus Volatilidade:

Os resultados obtidos evidenciam a existência de um trade-off notável entre consistência e volatilidade nos modelos analisados. Os modelos de rede MLP e autoencoder

se destacaram por apresentar resultados mais consistentes, evidenciando uma capacidade superior de gerenciar riscos, como ilustrado pela ocorrência de *Draw downs* menores. Por outro lado, a rede convolucional, embora tenha gerado o maior lucro, também exibiu o maior *Draw down*, tornando-a mais suscetível a oscilações e, portanto, mais volátil e arriscada. No caso da rede LSTM, embora tenha obtido lucro, seu *Draw down* significativo aponta para uma notável inconsistência em seu desempenho ao longo do tempo.

Esses achados destacam a importância de considerar cuidadosamente a relação entre consistência e volatilidade ao escolher um modelo de negociação, ponderando os benefícios potenciais em termos de lucro contra os riscos associados à volatilidade.

Fator de Lucro:

O fator de lucro é uma métrica importante para avaliar a eficácia das estratégias de negociação. Embora todos os modelos tenham apresentado fatores de lucro acima de 1.0, a rede convolucional se destacou com um valor de 1.12. Isso indica que, em média, ele gera mais lucros do que prejuízos. No entanto, essa métrica deve ser interpretada com cautela, pois a volatilidade pode afetar significativamente o desempenho geral.

Além disso, é importante ressaltar que os resultados passados, mesmo quando medidos por métricas como o fator de lucro, não garantem resultados futuros. O mercado financeiro pode ser muito imprevisível e a adaptação contínua e a gestão de riscos são essenciais para o sucesso a longo prazo. Portanto, a avaliação de uma estratégia de negociação deve ser um processo abrangente, considerando uma gama de métricas e fatores, bem como a capacidade da estratégia de se ajustar às mudanças nas condições do mercado.

Fator de Recuperação:

O fator de recuperação mede a capacidade da estratégia de recuperar perdas após um período de desempenho ruim. Uma alta taxa de recuperação indica que a estratégia é resiliente e capaz de se recuperar de perdas anteriores, o que é um sinal positivo. Por outro lado, um fator de recuperação baixo pode indicar que a estratégia tem dificuldade em se recuperar de perdas, o que pode ser preocupante para investidores e traders.

O autoencoder possui um fator de recuperação de 5. Isso indica que, após um período de perdas, essa estratégia tem uma notável capacidade de recuperação, o que é muito positivo e foi o maior resultado obtido entre os modelos testados neste tempo gráfico. A rede MLP e a convolucional possuem um fator de recuperação entre 1 e 3. Este valor indica uma capacidade moderada de recuperação após perdas. A rede LSTM possui um fator de recuperação de 0.72. Esse valor é relativamente baixo, sugerindo que a estratégia tem mais dificuldade em se recuperar de perdas anteriores.

Consecutividade de Negociações:

A análise das sequências de negociações consecutivas é crucial para entender

a consistência de um modelo. Todos os modelos tiveram um número semelhante de negociações com lucro consecutivas, variando entre 6 e 10, o que sugere que eles têm a capacidade de manter um bom desempenho por várias operações. No entanto, é importante observar que todos também tiveram negociações com perda consecutivas, o que destaca a importância da gestão de risco para evitar grandes rebaixamentos.

Comparações entre outros tempos gráficos:

Testando os modelos em outros *timeframes* também foram obtidos alguns resultados relevantes. Mostrando que mesmo um modelo que não se saiu bem em determinado tempo gráfico, não funciona ou devemos descartá-lo. Um exemplo disso é o modelo do Autoencoder, testando sua performance em um gráfico de 5 minutos seus resultados no mesmo período de tempo testado no tempo gráfico de 15 minutos foram bem menores, estando abaixo de quase todos os outros modelos se formos levar em consideração o lucro obtido, como é mostrado na tabela 7.

Lucro	DrawDown	% lucro	% perda	Total	Fator Lucro	Ganhos Seguidos	Perdas Seguidos	Fator Recuperação
450.0	-100.0	51.1	49.9	395	1.04	6	7	4.5

Tabela 7 – Tabela de resultados do Autoencoder M5

Percebe-se que a maioria das métricas analisadas no tempo gráfico de 5 minutos para o modelo do autoencoder foram piores, isso se deve pela estratégia utilizada. Pois as previsões continuam próximas da realidade, porém dada a estratégia utilizada, dependendo do modelo, ele não se adequa ao padrão buscado pela estratégia e não produz resultados expressivos. Podemos analisar a precisão das previsões no gráfico 16 que retrata um pequeno período de tempo dentro dos 6 meses analisados.

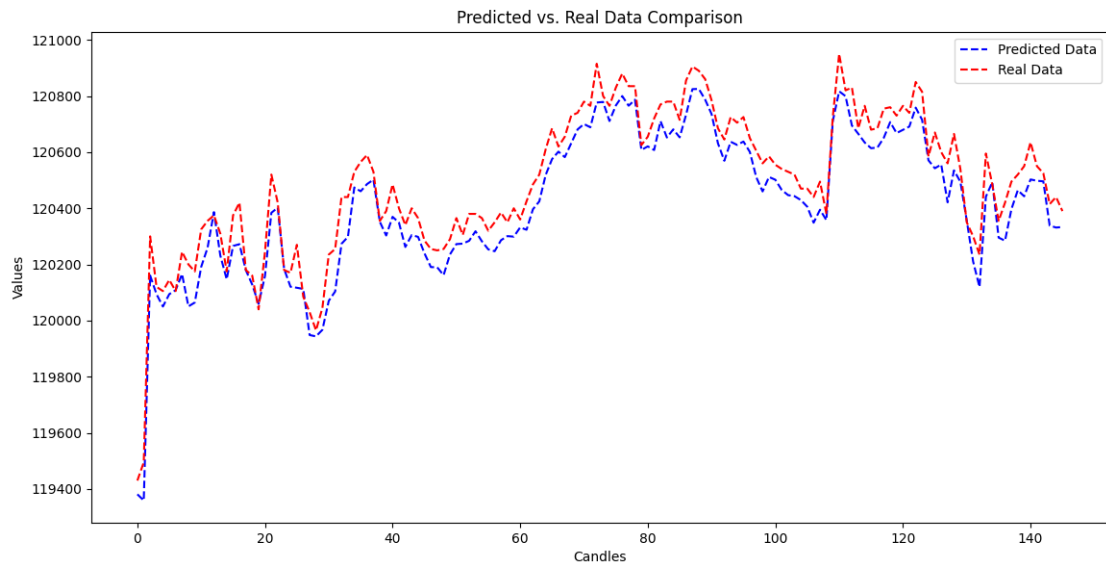


Figura 16 – Autoencoder M5. Fonte: Autor

É um fato que as previsões se mantêm bem próximas da realidade, porém devido a estratégia adotada os resultados nesse timeframe não se saíram tão bem quanto no tempo gráfico de 15 minutos.

Essa diferença de resultados se encontra na estratégia de operação adotada, pelo fato de que cada modelo atua de uma forma, alguns tem mais variações nos resultados nas previsões, outros tem menos variações. Dado esse fator, percebe-se que o autoencoder sempre tende a ter resultados seguindo o padrão do mercado sem grandes modificações nos preços de fechamento na passagem de um candle para outro. E como no tempo gráfico de 5 minutos o espaço para a movimentação dos preços é menor, o modelo do autoencoder algumas vezes não consegue acompanhar uma alavancagem do mercado ou uma queda e faz uma operação atrasada que consequentemente pode gerar o prejuízo, o que explica sua performance abaixo da média em relação ao gráfico de 15 minutos.

Em contrapartida, temos a rede neural LSTM que no gráfico de 15 minutos, entre todos os modelos analisados teve os menores resultados. Porém algo muito interessante ocorre com esse modelo quando utilizado no tempo gráfico de 5 minutos. Conseguimos obter ótimos resultados com a estratégia, como mostra a tabela 8. Os resultados obtidos utilizando esta rede no tempo gráfico de 5 minutos foram os melhores até o momento. Podemos ver que durante o período de teste obtivemos o maior valor de lucro entre todos os outros modelos e um drawdown extremamente pequeno. Dando a entender que é um modelo lucrativo e pouco volátil, o que torna este robô muito confiável, de pouco risco e com um retorno interessante.

Lucro	DrawDown	% lucro	% perda	Total	Fator Lucro	Ganhos Seguidos	Perdas Seguidos	Fator Recuperação
1600.0	-200.0	55.7	44.3	278	1.26	6	6	8

Tabela 8 – Tabela de resultados rede LSTM M5

A característica que faz com que essa rede se destaque neste tempo gráfico é que tanto a estratégia utilizada, quanto o modelo de rede se dão muito bem no tempo gráfico escolhido. O modelo de rede LSTM, mesmo estando no tempo gráfico de 5 minutos e a diferença dos preços de fechamento de um candle para outro não sendo tão grande, consegue pegar alguns movimentos de alavancagem ou queda do mercado a tempo, sendo capaz de sair no lucro em várias operações deste tipo. Isso ocorre pois em suas previsões existe uma variação maior nos preços de fechamento em relação ao autoencoder, pelas características da rede ele se sai melhor nesse tempo gráfico. Mas por se tratar de um modelo que utiliza da regressão linear, da mesma forma em que nos outros modelos avaliados, quando ocorre uma mudança de tendência do mercado o modelo muitas das vezes não identifica a tempo essa mudança e sofre com algumas perdas.

No gráfico 17, das previsões feitas comparando com os valores reais do mercado durante alguns dias do período de 6 meses testado. Percebe-se que, se comparado com o gráfico do autoencoder, existe uma variação bem maior nas previsões para cada candle mas sempre seguindo a mesma direção do mercado, o que fez o modelo obter ótimos resultados.

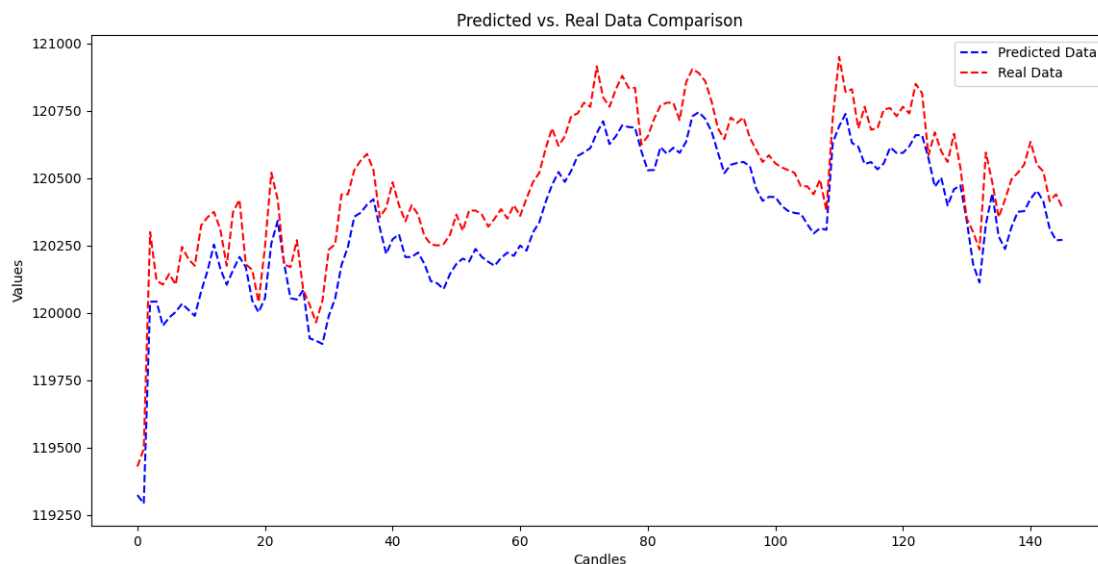


Figura 17 – LSTM M5. Fonte: Autor

Considerações Individuais:

Cada modelo possui suas próprias características, e a escolha entre eles deve levar em

consideração o perfil de risco e os objetivos do investidor. Investidores mais conservadores podem preferir os modelos de rede MLP ou autoencoder devido à sua menor volatilidade, enquanto investidores dispostos a assumir mais riscos podem se sentir atraídos pelo modelo de rede convolucional, que teve o maior fator de lucro. O modelo LSTM, com um lucro menor e um *Draw down* significativo, não apresenta pontos fortes para os investidores no tempo gráfico avaliado.

A avaliação e escolha de um modelo de negociação não devem ser baseadas apenas no lucro, mas também na gestão de risco, consistência e objetivos pessoais. Além disso, monitorar e ajustar constantemente esses modelos é fundamental para adaptar-se às mudanças nas condições do mercado.

5 Conclusão

Ao longo deste trabalho de conclusão de curso os pontos de foco foram no desenvolvimento e análise detalhada e comparativa de quatro modelos de redes neurais diferentes, aplicados ao contexto do mercado financeiro. O objetivo central era compreender o funcionamento dessas redes neurais e avaliar seu desempenho em termos de lucratividade, consistência e risco e expandir a usabilidade dessas tecnologias para o mercado financeiro.

Cada um dos modelos apresentados revelou suas características únicas e *trade-offs* significativos. Alguns com maior volatilidade e indicando um risco maior, como é o caso das redes convolucionais e LSTM e outros mais estáveis e seguros como o autoencoder e a rede padrão.

Nas análises para o tempo gráfico de 15 minutos concluiu-se que o modelo que se mostrou mais interessante e obteve resultados promissores foi o autoencoder, por ter obtido um lucro razoável no período de testes, o menor *draw down* entre os modelos avaliados e os melhores valores entre as outras métricas de avaliação adotadas. Porém isso não indica que os outros modelos são ruins ou não devem ser utilizados, mas dada a estratégia de negociação escolhida e o tempo gráfico utilizado o autoencoder foi o que obteve a melhor performance.

5.1 Trabalhos Futuros

A aplicação desenvolvida pode receber algumas modificações consideráveis como trabalhos futuros a fim de aprimorar algumas funcionalidades ou até mesmo melhorar os resultados obtidos com este estudo. Segue as principais nesta seção.

Foram desenvolvidas, treinadas e testadas as redes neurais utilizando-se de diferentes modelos de rede neural que processavam os dados e por fim era utilizado a técnica de regressão linear para finalizar a predição. Podemos ver que em determinados momentos as predições ficam atrasadas em relação aos valores reais devido à forma que a regressão linear funciona. Para ajustar esse atraso, poderia ser adicionado mais uma rede neural que analisa o resultado desta predição juntamente com alguns outros dados do mercado para corrigir esta ligeira defasagem.

No presente trabalho desenvolvido, o servidor é apenas local e só pode ser utilizado pelo computador que está o rodando. Uma melhoria interessante seria colocar a aplicação em um servidor público. Fazendo com que várias pessoas tenham acesso aos dados disponibilizados pelo servidor e possibilitando maiores estudos e testes nesta área.

Referências

- ACADEMY, D. S. *Deep Learning Book*. 2022. Disponível em: <<https://www.deeplearningbook.com.br/>>. Citado 11 vezes nas páginas 10, 20, 21, 22, 23, 25, 26, 27, 28, 30 e 31.
- BTG Pactual Digital. *O que é Day Trade*. s.d. <<https://www.btgpactualdigital.com/como-investir/artigos/day-trade/o-que-e-day-trade>>. Acesso em: 08 maio 2023. Citado na página 14.
- CHAVES, A. *Melhores Robôs de Investimento em 2021*. 2020. Acesso em: 01 ago. 2021. Disponível em: <<https://beincrypto.com.br/aprender/melhores-robos-de-investimentos/>>. Citado na página 19.
- COSSETTI, M. C. *O que é inteligência artificial?* 2018. <<https://tecnoblog.net/responde/o-que-e-inteligencia-artificial/>>. Acesso em: 15 out. 2023. Citado na página 19.
- COSTA, R. *Projeto de Graduação - Desenvolvimento de um Sistema de Gerenciamento de Energia para Microgrids*. Monografia (Monografia) — Universidade Federal do Espírito Santo (UFES), 2019. Citado 2 vezes nas páginas 26 e 27.
- DAMÁSIO, B. F. *O que é regressão linear simples?* s.d. <<https://psicometriaonline.com.br/o-que-e-regressao-linear-simples-2/>>. Acesso em: 10 out. 2023. Citado 4 vezes nas páginas 10, 32, 33 e 34.
- ESPERANDIO, C. *Só 0,1% lucra mais de R\$ 100 por dia com day-trade de ações*. 2020. Disponível em: <<https://economia.uol.com.br/columnas/econoweb/2020/09/15/day-trade-vale-a-pena.htm>>. Citado na página 19.
- FIA. *Inteligência Artificial: o que é, como funciona e exemplos*. 2023. <<https://fia.com.br/blog/inteligencia-artificial/>>. Acesso em: 15 out. 2023. Citado na página 20.
- GAZONI, P. L. *Redes Neurais: Aplicações em Finanças*. Monografia (Projeto de Iniciação Científica) — Escola de Administração de Empresas de São Paulo Fundação Getúlio Vargas, 1999. Citado na página 15.
- IBM. *Redes neurais: o que são e como funcionam*. s.d. <<https://www.ibm.com/br-pt/topics/neural-networks>>. Acesso em: 08 maio 2023. Citado na página 14.
- INFOMONEY. *Guia do Day Trade: como comprar e vender ações no mesmo dia*. 2020. Acesso em: 30 jul. 2021. Disponível em: <<https://www.infomoney.com.br/guias/day-trade/>>. Citado na página 18.
- INVESTIMENTOS, T. *Análise Técnica*. 2023. <<https://artigos.toroinvestimentos.com.br/trading/analise-tecnica>>. Acesso em: 15 out. 2023. Citado na página 18.
- INVESTIMENTOS, T. *Mercado de Ações: O que é, como funciona e como investir*. 2023. Disponível em: <<https://artigos.toroinvestimentos.com.br/bolsa/mercado-de-acoes>>. Citado na página 17.

- JR., J. G. de C. *Modelos Conexionistas - Redes Neurais*. s.d. <https://www.professores.uff.br/joseflaviofeiteira/wp-content/uploads/sites/126/2017/09/Gomes_1.pdf>. Acesso em: 15 out. 2023. Citado na página 22.
- METAQUOTES. *MQL5 Fórum*. 2000. <<https://www.mql5.com/>>. Acesso em: 15 out. 2023. Citado na página 16.
- MetaQuotes Ltd. *Manual MQL5*. 2023. <<https://www.mql5.com/pt/docs>>. Acesso em: 24 setembro 2023. Citado na página 35.
- MetaQuotes Ltd. *Plataforma de negociação MetaTrader 5*. 2023. <<https://www.metatrader5.com/pt/trading-platform>>. Acesso em: 24 setembro 2023. Citado na página 35.
- MIRA, E. Mercado financeiro: o que o iniciante precisa saber. *Forbes*, 2022. Disponível em: <<https://forbes.com.br/forbes-money/2022/10/eduardo-mira-mercado-financeiro-o-que-o-iniciante-precisa-saber/>>. Citado na página 17.
- PYTHON. *What is Python*. 2023. <<https://www.python.org/doc/essays/blurb/>>. Acesso em: 24 setembro 2023. Citado na página 35.
- RODRIGUES, D. A. *DEEP LEARNING E REDES NEURAIAS CONVOLUCIONAIS: RECONHECIMENTO AUTOMÁTICO DE CARACTERES EM PLACAS DE LICENCIAMENTO AUTOMOTIVO*. Monografia (Monografia) — CENTRO DE INFORMÁTICA UNIVERSIDADE FEDERAL DA PARAÍBA, 2018. Citado 3 vezes nas páginas 27, 28 e 29.
- SMARTTBOT. *SmarttBot*. 2011. <<https://smarttbot.com/>>. Acesso em: 15 out. 2023. Citado na página 16.
- SMARTTBOT. *Descubra o que é algo trading e como você pode lucrar com ele*. 2020. Acesso em: 06 ago. 2021. Disponível em: <<https://smarttbot.com/trader/o-que-e-algotrading/>>. Citado na página 19.
- TECH, D. *Neurônios artificiais*. s.d. <<https://didatica.tech/neuronios-artificiais/>>. Acesso em: 15 out. 2023. Citado na página 21.
- TensorFlow. *Introdução ao TensorFlow*. 2023. <<https://www.tensorflow.org/learn?hl=pt-br>>. Acesso em: 24 setembro 2023. Citado na página 36.