

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS TIMÓTEO**

Maycon Arthuso de Carvalho

**ESTUDO DE CASO: USO DE RNAS EM MINI ÍNDICE BOVESPA EM
GRÁFICO ATEMPORAL NA BOLSA DE VALORES BRASILEIRA**

Timóteo

2023

Maycon Arthuso de Carvalho

ESTUDO DE CASO: USO DE RNAS EM MINI ÍNDICE BOVESPA EM GRÁFICO ATEMPORAL NA BOLSA DE VALORES BRASILEIRA

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Douglas Nunes de Oliveira

Timóteo

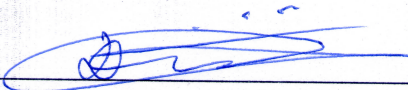
2023

Maycon Arthuso de Carvalho

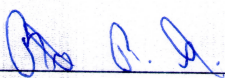
**ESTUDO DE CASO: USO DE RNAs EM MINI ÍNDICE BOVESPA EM GRÁFICO
ATEMPORAL NA BOLSA DE VALORES BRASILEIRA**

Trabalho de Conclusão de Curso
apresentado ao Curso de Engenharia de
Computação do Centro Federal de Educação
Tecnológica de Minas Gerais, campus Timóteo,
como requisito parcial para obtenção do título de
Engenheiro de Computação.

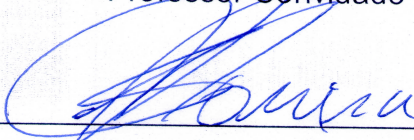
Trabalho aprovado. Timóteo, 15 de março de 2023:



Prof. Me. Douglas Nunes de Oliveira
Orientador



Prof. Dr. Bruno Rodrigues Silva
Professor Convidado



Prof. Me. Luciano Nascimento Moreira
Professor Convidado

Timóteo
2023

Dedico a todos
que me auxiliaram nessa caminhada.

Agradecimentos

Agradeço muito aos meus pais, minha irmã e meus amigos, além da instituição CE-FET e os professores que me deram todo o suporte e motivação nessa etapa da minha vida acadêmica e profissional.

*“Os grandes navegadores
devem sua reputação aos temporais e tempestades”.*
Epicuro

Resumo

Um dos meios mais clássicos para se realizar negociações financeiras é através da bolsa de valores, onde teve seu início por volta do século XV. A bolsa de valores brasileira tem como principal índice de ações o Ibovespa, que se tornou uma grande referência para o mercado por ser um indicador de desempenho médio dos ativos de maior negociabilidade e representatividade no mercado brasileiro. O mini índice Bovespa é um ativo que acompanha as oscilações do índice principal, representando 20% do índice cheio Ibovespa. Investidores estão expostos a grandes riscos investindo em ativos pela bolsa de valores, podendo assim ocasionar grandes perdas monetárias. Tendo isso em vista, este trabalho é um estudo de caso utilizando como objeto o mini índice Bovespa, aliado com o uso de redes neurais artificiais, para auxiliar nas decisões de abertura e fechamento de operações financeiras em um mesmo dia durante o período de execução de alguns dias. Foi utilizado 5 variações de redes neurais, com diferentes topologias e funções de ativação, todos em modelos *Perceptron* Multi-Camadas. Nos dados de entrada, utilizaram-se alguns indicadores técnicos de gráfico Média Móvel Simples, Média Móvel Exponencial, ADX, Estocástico e o HiLo Activator, sendo esse último utilizado como principal sinal de ativação para abrir posições de compra e venda. Para cada uma das 5 topologias de redes neurais foi implementado 2 modos com uma diferença na forma como os dados de entrada foram utilizados. Posteriormente todas as redes neurais foram treinadas e utilizadas para realizar negociações simultaneamente, em ambiente real utilizando uma conta de demonstração, em um mesmo período de 13 dias. Além disso, foi utilizada uma versão do algoritmo de negociação sem uma rede neural (*setup*) no mesmo período citado, usando os mesmos parâmetros. Após o período de negociações, foram coletados dados das negociações de cada modelo com rede neural e do modelo sem a rede neural para comparação. Os resultados coletados foram que alguns modelos obtiveram lucro, atingindo ganhos monetários positivos e valores acima da versão do algoritmo sem utilização de rede neural, porém alguns modelos obtiveram resultados monetários negativos e assim menos eficientes quanto comparado com a versão sem a utilização de inteligência artificial. Dessa forma, foi possível observar que alguns algoritmos que utilizaram redes neurais, obtiveram resultados monetários superiores ao algoritmo que não utilizou uma inteligência artificial.

Palavras-chave: bolsa de valores, mini índice bovespa, séries atemporais, Renko, redes neurais artificiais, negociações financeiras.

Abstract

One of the most classic ways of carry out financial negotiations is through the stock exchange, where it began around the 15th century. The main stock index of the Brazilian stock exchange is the Ibovespa, which has become a mainly benchmark for the market as it is an average performance indicator for the most tradable and representative assets in the Brazilian market. The Bovespa mini index is an asset that follows the fluctuations of the main index, in addition to representing 20% of the full Ibovespa index. Investors are exposed to great risks by investing in assets through the stock exchange, this being able to cause large monetary losses. With that in mind, this work is a case study the Bovespa mini index as a object, combined with the use of artificial neural networks, to assist in the decisions of opening and closing financial operations on the same day during the execution period of a few weeks. It was used 5 variations of neural networks, with different topologies and activation functions, all in Perceptron Multi-Layer models. In the input data, some technical indicators of Simple Moving Average, Exponential Moving Average, ADX, Stochastic and HiLo Activator were used, the latter being used as the main activation signal to open buy and sell positions. For each of the 5 neural network topologies, 2 modes were implemented with a difference in the way the input data was used. Subsequently, all neural networks were trained and used to perform the processing simultaneously, in a real environment using a demo account, in the same period of 13 days. In addition, a version of the trading algorithm without a neural network was used in the same period cited, using the same parameters. After the negotiation period, data from the negotiations of each model with neural network and the model without the neural network were collected for comparison. The results collected were that some models obtained profit, reaching positive monetary gains and above the version of the algorithm without the use of a neural network, but some models obtained negative monetary results and thus less efficient when compared to the version without the use of artificial intelligence. Thus, it was possible to observe that the algorithms that used neural networks, obtained higher monetary results than the algorithm that did not use an AI.

Keywords: stock exchange, bovespa mini index, timeless series, Renko, artificial neural networks, financial negotiations.

Lista de ilustrações

Figura 1 – Gráfico da função de ativação Sigmoid	18
Figura 2 – Gráfico da função de ativação Tangente hiperbólica	18
Figura 3 – Gráfico da função de ativação ReLU	19
Figura 4 – Representação de um <i>candle</i> de alta e de baixa	22
Figura 5 – Gráfico de <i>candlestick</i>	23
Figura 6 – Gráfico de <i>candlestick</i> com tempo de 1 minuto	24
Figura 7 – Gráfico <i>Renko</i>	24
Figura 8 – Gráfico do indicador estocástico	28
Figura 9 – HiLo Activator Indicador de um Gráfico Ibovespa Semanal	29
Figura 10 – Mini índice Bovespa	30
Figura 11 – Exemplo de <i>setup</i> operacional	32
Figura 12 – Exemplo de gráfico com Stop Loss	33
Figura 13 – Exemplo de gráfico com Stop Gain	34
Figura 14 – Exemplo de gráfico com Trailing Stop antes de mudar a referência	34
Figura 15 – Exemplo de gráfico com Trailing Stop depois de mudar a referência	35
Figura 16 – Metatrader 5	36
Figura 17 – Tela do testador de estratégias	36
Figura 18 – Tela do Profit Pro	37
Figura 19 – Procedimentos metodológicos para a construção do modelo	39
Figura 20 – Aba comum da biblioteca Renko 2.0 Offline	41
Figura 21 – Aba de configurações da biblioteca Renko 2.0 Offline	41
Figura 22 – Exemplo de sinal de compra	42
Figura 23 – Exemplo de sinal de venda	42
Figura 24 – Diagrama de topologia da MLP	44
Figura 25 – Fluxo de etapas para realização das negociações	48
Figura 26 – Configurações usadas para conversão e criação do gráfico <i>Renko</i>	51
Figura 27 – Trecho do gráfico <i>WIN\$N</i> em forma temporal	51
Figura 28 – Trecho do gráfico <i>WIN\$N</i> em forma atemporal	52
Figura 29 – Exemplo das principais configurações para coleta de dados	52
Figura 30 – Exemplo dos parâmetros para coleta de dados	53
Figura 31 – Exemplo de execução da coleta de dados	53
Figura 32 – Valores das cotações temporais do mini índice no período de 30/05/2022 a 15/06/2022	64
Figura 33 – Valores das cotações do mini índice em forma de gráfico <i>Renko</i> no período de 30/05/2022 a 15/06/2022	65
Figura 34 – Gráfico temporal de um trecho dia 15 junho de 2022	66
Figura 35 – Gráfico <i>Renko</i> do dia 15 junho de 2022	66
Figura 36 – Parâmetros de configuração para operar as negociações	67
Figura 37 – Valores monetários obtidos nas negociações	70

Figura 38 – Quantidade de negociações de venda e compra	71
Figura 39 – Porcentagem de negociações com lucro e prejuízo	72

Lista de tabelas

Tabela 1 – Código e vencimento	30
Tabela 2 – Classificação das amostras por intervalo de pontos	44
Tabela 3 – Topologia da RNA 1	45
Tabela 4 – Topologia da RNA 2	45
Tabela 5 – Topologia da RNA 3	46
Tabela 6 – Topologia da RNA 4	46
Tabela 7 – Topologia da RNA 5	46
Tabela 8 – Parâmetros do indicador Média Móvel Exponencial	49
Tabela 9 – Parâmetros do indicador ADX	49
Tabela 10 – Parâmetros do indicador Estocástico	49
Tabela 11 – Parâmetros do indicador HiLo	50
Tabela 12 – Classificação da operação por intervalo de pontos	50
Tabela 13 – Treinamento e resultado variação 1 RNA	57
Tabela 14 – Treinamento e resultado variação 2 RNA	58
Tabela 15 – Treinamento e resultado variação 3 RNA	59
Tabela 16 – Treinamento e resultado variação 4 RNA	61
Tabela 17 – Treinamento e resultado variação 5 RNA	62
Tabela 18 – Lucros obtidos nas negociações somente das RNAs	70
Tabela 19 – Negociações emitidas por modelo	72

Sumário

1	INTRODUÇÃO	13
1.1	Objetivos	15
1.1.1	Objetivos Específicos	15
2	REVISÃO BIBLIOGRÁFICA	16
2.1	Trabalhos correlatos	16
2.2	Redes Neurais Artificiais	16
2.2.1	Funções de ativação	17
2.2.1.1	Sigmoide	17
2.2.1.2	Tangente hiperbólica	18
2.2.1.3	ReLU	19
2.2.1.4	Softmax	19
2.2.2	Funções de otimização	20
2.2.2.1	SGD	20
2.2.2.2	AdaGrad	20
2.3	Balanceamento de dados	20
2.4	Séries temporais	21
2.5	Candlesticks	21
2.5.1	Formação temporal e atemporal	23
2.6	Mercado financeiro de <i>Day Trade</i>	25
2.7	Indicadores técnicos	25
2.7.1	Média Móvel Simples	25
2.7.2	Média Móvel Exponencial	26
2.7.3	Índice de Movimento Direcional	26
2.7.4	Estocástico	27
2.7.5	HiLo Activator	28
2.8	Mini Índice Bovespa	29
2.9	Técnica de negociação	31
2.9.1	Setup Operacional	31
2.9.2	Stop Loss	32
2.9.3	Stop Gain	33
2.9.4	Trailing Stop	34
2.10	Plataformas de negociações	35
2.10.1	Metatrader	35
2.10.2	Profit Pro	37
2.11	Frameworks e bibliotecas	37
2.11.1	Flask	38
2.11.2	Tensorflow	38

3	PROCEDIMENTOS METODOLÓGICOS	39
3.1	Classificação da pesquisa	39
3.2	Revisão bibliográfica	39
3.3	Indicadores, plataforma de negociação e estratégia	40
3.4	Obtenção dos dados de entrada	40
3.4.1	Transformação do gráfico	40
3.4.2	Classificação e armazenamento dos dados	41
3.4.3	Uso de DLL para coleta de dados	43
3.5	Modelo da RNA	43
3.5.1	Topologia da RNA 1	45
3.5.2	Topologia da RNA 2	45
3.5.3	Topologia da RNA 3	45
3.5.4	Topologia da RNA 4	46
3.5.5	Topologia da RNA 5	46
3.6	Treinamento e validação da previsão do modelo	47
3.7	Construção da API	47
3.8	Usar modelos em conta de demonstração	47
3.9	Análise dos resultados	48
4	DESENVOLVIMENTO	49
4.1	Obtenção dos dados	49
4.2	Balanceamento e classificação dos dados coletados	55
4.3	Implementação das RNAs	56
4.3.1	Variação 1 da RNA	56
4.3.2	Variação 2 da RNA	57
4.3.3	Variação 3 da RNA	58
4.3.4	Variação 4 da RNA	59
4.3.5	Variação 5 da RNA	61
4.4	Construção da API	62
4.5	Negociações usando as RNAs	63
4.5.1	Negociação sem o uso de IA	67
4.5.2	Negociação com o modelo 1 da RNA	68
4.5.3	Negociação com o modelo 2 da RNA	68
4.5.4	Negociação com o modelo 3 da RNA	68
4.5.5	Negociação com o modelo 4 da RNA	69
4.5.6	Negociação com o modelo 5 da RNA	69
4.5.7	Comparativo geral	69
5	CONCLUSÃO	73
	REFERÊNCIAS	76

1 Introdução

*“A resposta certa, não importa nada:
o essencial é que as perguntas estejam certas”.*

Mário Quintana

Um dos meios mais tradicionais de se realizar negociações financeiras é através da bolsa de valores. A sua origem não é exata e alguns escritores a localizam “nos *emporium* dos gregos, outros nos *collegium mercatorum* dos romanos, ou nos *funduks* (bazares) dos palestinos” (RUDGE; CAVALCANTE, 1996). As primeiras bolsas de valores com características modernas surgiram com o desenvolvimento do comércio, por volta do século XV. Mas foi somente em Burges, na Bélgica, em 1487, que a palavra bolsa adquiriu seu sentido comercial e financeiro, quando mercadores e comerciantes se reuniam na casa do senhor Van der Burse para realizar negócios, onde o brasão da casa era um desenho de três bolsas. A partir do século XIX, com o aumento dos empreendimentos comerciais, a bolsa foi se tornando mais voltada para a parte financeira através do comércio de ações. Com isso, teve um grande aumento na quantidade de transações realizadas e também foi decisivo para a especialização da bolsa voltada para as mercadorias e outras voltadas para mobiliários (GOMES, 1997).

No Brasil, inicialmente a Bovespa teve sua origem em agosto de 1890, por um grupo de negociantes liderados por Emilio Rangel Pestana. A BM&F teve sua origem em janeiro de 1986 e, no início da década de 90, tornando-se o principal centro de negociação de derivativos da América Latina. Em maio de 2008 foi realizada a junção dessas duas instituições BM&F (bolsa de derivativos) e a Bovespa (bolsa de ações) chamando-se BM&FBovespa, estava nascendo uma das maiores Bolsa de Valores do mundo (ALMEIDA, 2015). Em 2017, a BM&FBovespa realizou uma união com Cetip, assim rebatizada para o nome atual chamada B3, sigla para Brasil, Bolsa e Balcão (B3, 2020b).

Quando são efetuadas aplicações na bolsa de valores, os investidores desejam ter o máximo de retorno e o mínimo de prejuízo nas suas negociações, assim comprar ações com preço baixo e vender com preço alto é o objetivo que os investidores almejam alcançar. Por isso, é de grande importância fazer essas negociações no momento certo (ALMEIDA, 2015).

O Ibovespa é o índice de desempenho das ações mais importante do mercado brasileiro, sendo ele uma representação de carteira teórica de ações e *units* negociados na B3, cuja quantidade pode variar conforme as regras de composição. Por ser um indicador do desempenho médio dos preços dos ativos de maior negociabilidade e representatividade no mercado brasileiro, esse índice tornou referência para a rentabilidade de fundos de ações e para o desempenho da Bolsa (B3, 2020a). O mini índice Bovespa, assim como o Ibovespa, são contratos futuros de compra ou venda de uma estimativa do Índice Bovespa para uma data futura, com um preço determinado (GÓMEZ, 2019), ou seja, é um ativo que envolve negociações de contratos para uma data futura, que tem como base o sobe e desce da Ibovespa. Os mini contratos Bovespa são chamados assim porque eles representam 20% de um contrato de ín-

dice cheio (GÓMEZ, 2019), com isso o investidor pode começar a investir nesses ativos com valores menores.

Para se efetuar negociações na bolsa de maneira mais profissional, pode utilizar algumas técnicas para fazer suas tomadas de decisão, entre as principais são a análise técnica, a análise fundamentalista e a análise de fluxo. A análise fundamentalista tem o objetivo de fazer o processamento de informações obtidas da empresa em questão, partindo do entendimento da conjuntura macroeconômica onde a companhia está inserida, procedimentos operacionais e o valor esperado de sua ação (ALMEIDA, 2015). Enquanto a análise técnica, segundo (MURPHY, 1986) “é o estudo da ação do mercado, principalmente através da utilização de gráficos, com a finalidade de prever futuras tendências de preços”. Por fim, a análise de fluxo, também conhecida como *Tape Reading*, é uma técnica onde os investidores acompanham o fluxo de negociações, de compra e de venda, na bolsa de valores, visando identificar a tendência dos preços no mercado e seguir esse movimento para realizar negócios (TORO, 2021). Com isso, analisar as informações obtidas é de grande valia para ter resultados positivos em investimentos.

Embora já existem técnicas para auxiliar nas tomadas de decisões, realizar previsões dos valores das ações ainda é um problema não solucionado de forma satisfatório para os investidores, muito pelo caráter de grandes oscilações que o mercado da bolsa de valores possui. Milhares de investidores estão expostos a grandes riscos durante os investimentos em ativos que variam tanto seu valor ao longo do tempo, podendo ocasionar em grandes perdas monetárias. Segundo (KUMAR; JAIN; SINGH, 2020) a previsão do mercado de ações é uma tarefa altamente exigente e desafiadora devido à sua natureza não paramétrica, volátil, complexa, dinâmica e caótica dos preços das ações.

Com os avanços da tecnologia, a Inteligência Artificial tem crescido no cotidiano das pessoas e no mercado. Uma das subáreas desse campo, a Inteligência Computacional, especialmente as Redes Neurais Artificiais (RNA), que tem recebido muita atenção nos campos de economia e de finanças. Uma RNA é um modelo de aproximação universal que pode mapear funções não-lineares usando múltiplas camadas de neurônios não-lineares (CYBENKO, 1989). A principal vantagem dessa técnica é a capacidade de explorar a tolerância de sistemas para dados de incerteza, imprecisão e de verdades parciais (MARTINEZ et al., 2009).

Contudo, há outras pessoas no mercado financeiro que duvidam da habilidade das RNAs de “decifrar” o mercado financeiro. Eles observam a inteligência artificial como uma ferramenta que proporciona uma vantagem sistemática, podendo fornecer uma visão sobre o risco e quando devem ocorrer esse risco. Mesmo que essa ferramenta forneça ao investidor uma certa vantagem temporária, assim que a base dela acabar, outros investidores a terão, assim a vantagem desaparece (DATAFLOQ, 2020).

Os mercados financeiros não obedecem regras estacionárias. A prova disso é como o mercado reagiu à pandemia do coronavírus, onde se o desempenho passado permanecesse com o desempenho atualmente, o mercado na totalidade deveria estar em queda livre. Mas o que foi visto é que certos setores estão sofrendo devido à pandemia, enquanto outros estão prosperando. Os mercados financeiros flutuam com base em eventos sociais, políticos e

econômicos, assim uma IA não pode prever com precisão a fazer escolhas inteligentes com base no mercado atual. Isso não significa que o aprendizado de máquina não tenha lugar na previsão dos mercados financeiros, isso significa há limites para o ponto que as previsões possam ser realizadas com precisão (DATAFLOQ, 2020).

Alguns trabalhos realizados nessa área já enfrentaram esses desafios, como o caso do trabalho de (SILVA, 2021), que propôs um estudo de caso utilizando o mini dólar, aplicando redes neurais, para definir momentos de compra e venda desse ativo, objetivando lucros nas negociações. O trabalho em questão relata que as redes neurais usadas geraram lucros maiores que a não utilização dos mesmos. Além disso, foram propostos várias outras direções de trabalhos futuros, onde este trabalho irá conduzir a aplicação do ativo mini índice Bovespa na rede neural.

1.1 Objetivos

Considerando o contexto e os problemas descritos anteriormente e as melhorias propostas pelo trabalho de (SILVA, 2021), este trabalho visa realizar um estudo de caso aplicando Redes Neurais Artificiais para realizar a previsão de movimentação do Mini Índice Bovespa negociado na bolsa de valores brasileira, avaliando o desempenho visando obter lucros monetários.

1.1.1 Objetivos Específicos

- Permitir comunicação da plataforma de negociação no modo *backtest* com a rede neural, sanando assim um dos problemas que Silva (2021) se deparou na coleta de dados.
- Aplicar o modelo de Rede Neural Artificial em gráficos não temporais.

2 Revisão Bibliográfica

“Um bom começo é a metade”.
Aristóteles

Este capítulo pretendem esclarecer de forma mais detalhada os principais conceitos relacionados com o presente trabalho e trabalhos correlatos. As fontes de pesquisa utilizadas foram o Google Acadêmico, Portal de Periódico da Capes, sites oficiais de universidades brasileira e sites conceituados relacionados ao mercado financeiro.

2.1 Trabalhos correlatos

No trabalho de Silva (2021) foi realizado um caso de estudo usando o mini dólar aplicado redes neurais artificiais, para definir os momentos de compra e venda do ativo. Foi realizado uma comparação entre cinco variações de *Multilayer Perceptrons* com diferentes topologias. As diferentes redes neurais foram testadas sendo extraídos relatórios dos resultados, onde alguns modelos apresentaram resultados animadores. Segundo o autor, a utilização de alguns dos modelos de redes neurais geraram lucros superiores do que a não utilização dos mesmos, durante o período analisado. A comparação se deu através do uso do *setup* operacional, que são estratégias, baseado em análise técnica, composto por um conjunto de critérios que definem se uma operação será executada ou não em um determinado momento.

Enquanto no trabalho de (MARRETTI, 2019), foi apresentado, usando técnicas de aprendizado de máquina, a inferência de resultados em ações do índice Bovespa no mercado de ações brasileiro, com o objetivo de obter estratégias operacionais com retornos acima do *benchmark buy-and-hold*. Foram realizadas diferentes estratégias de construção de modelos de aprendizado de máquina com diferentes heurísticas computacionais, obtendo resultados positivos para alguns desses modelos.

No trabalho de (POMMERANZENBAUM, 2014) apresenta um modelo de predição de valores sobre as principais séries de preços do índice Bovespa máxima do dia (*high*), mínima do dia (*low*), fechamento (*close*) e a ordem que compra e venda (*order*), utilizando redes neurais artificiais. Foram obtidas séries temporais, submetidos a métodos clássicos de processamento de séries temporais, para entrada das redes neurais. Foi criado um ambiente de simulação para efetuar as negociações usando a rede neural.

2.2 Redes Neurais Artificiais

Trabalhos em redes neurais artificiais, também conhecidas como redes neurais, desde o reconhecimento de que o cérebro humano processa informações de uma forma completamente diferente de um computador digital. O cérebro é um computador muito complexo, não linear e paralelo (HAYKIN, 2001).

De uma forma mais geral, uma rede neural é uma máquina projetada para modelar uma tarefa particular ou uma função que seria realizada pelo cérebro (HAYKIN, 2001). Possuem a capacidade de adquirir e revisar o conhecimento, baseado em informações, e podem também ser definidos como um conjunto de unidades de processamento, composto por neurônios artificiais, interligados por um grande número de interconexões, as mesmas representadas por matrizes de pesos sinápticos (SILVA; SPATTI; FLAUZINO, 2010). O neurônio é a unidade básica de processamento.

As redes neurais de *perceptron* de multicamadas, ou MLP, são camadas que “consiste de um conjunto de unidades sensoriais (nós de fonte) que constituem a camada de entrada, uma ou mais camadas ocultas de nós computacionais e uma camada de saída de nós computacionais.” (HAYKIN, 2001). Elas são caracterizadas pela grande variedade de aplicações em diversos tipos de problemas nas mais diferentes áreas do conhecimento, sendo uma das ferramentas mais versátil quanto a sua aplicabilidade (SILVA; SPATTI; FLAUZINO, 2010).

O processo de treinamento das MLP é feito de forma supervisionada usando o algoritmo de retropropagação de erro, o *back-propagation*, baseado na regra aprendizagem por correção do erro. Basicamente a aprendizagem se dá por dois passos: o de propagação, que é quando o vetor de entrada é aplicado aos nós sensoriais da rede e seu efeito é propagado, camada por camada; e o passo de retropropagação, onde os pesos sinápticos são todos ajustados de acordo com uma regra de correção do erro. A resposta real da rede é subtraída da resposta desejada, para produzir um sinal de erro. Este erro é então propagado para trás da rede, contra as conexões sinápticas, ajustando os pesos para obter uma resposta real da rede mais próxima da resposta desejada (HAYKIN, 2001).

2.2.1 Funções de ativação

Uma função de ativação recebe o sinal de entrada e gera um sinal de saída, mas leva em conta algum tipo de limite. Existem muitas funções de ativação que podem ter esse efeito (RASHID, 2016). A função de ativação é uma transformação não linear nos dados de entrada, fazendo com que as redes neurais possam resolver problemas mais complexos. Abaixo é descrito as funções que foram usadas neste trabalho.

2.2.1.1 Sigmoides

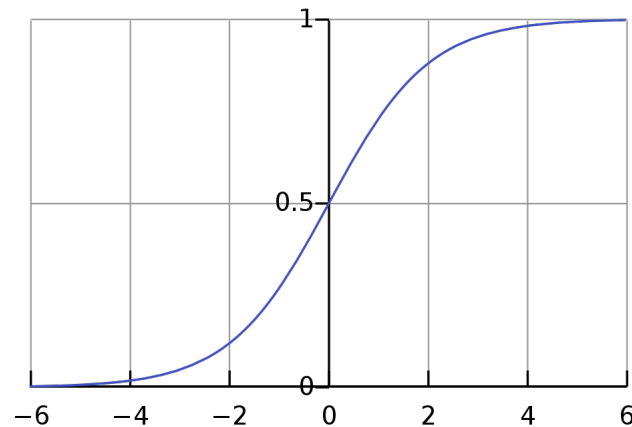
A função de ativação sigmoide é amplamente utilizada em redes neurais. Também chamada de função logística, ela é continuamente diferenciável e não linear. Os valores de saída dessa função ficam entre o intervalo de $[0,1]$, que é útil para realizar cálculos que devem ser interpretados como probabilidades (AGGARWAL et al., 2018).

Sua equação é dada por:

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

O gráfico que representa essa função está na figura 1 abaixo.

Figura 1 – Gráfico da função de ativação Sigmoide



Fonte: (WIKIPEDIA, 2022b)

2.2.1.2 Tangente hiperbólica

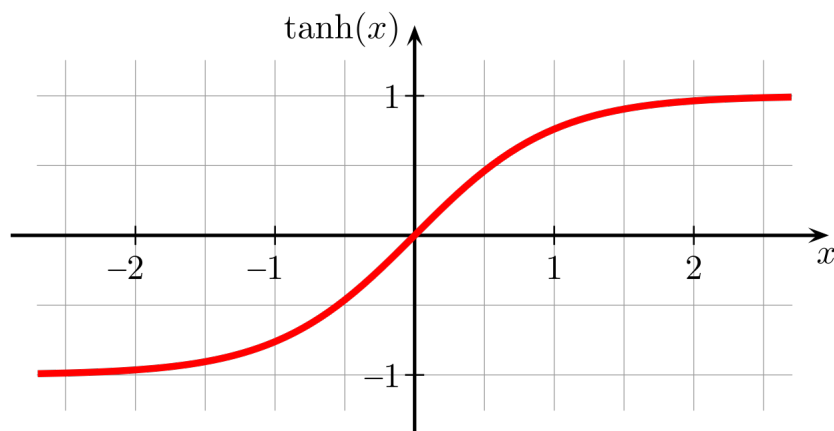
A função tangente hiperbólica tem um caráter similar a função sigmoide, exceto que ela para é verticalmente reescalada para valores entre $[-1,1]$ e com isso são considerados valores de saída positivos e negativos (AGGARWAL et al., 2018). Essa faixa de valores tende a fazer a saída da camada mais ou menos centrada no valor 0 no início do treinamento, o que geralmente ajuda a acelerar a convergência (GÉRON, 2019).

Sua equação é dada por:

$$y(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.2)$$

O gráfico que representa essa função está na figura 2 abaixo.

Figura 2 – Gráfico da função de ativação Tangente hiperbólica



Fonte: (WIKIPEDIA, 2022c)

2.2.1.3 ReLU

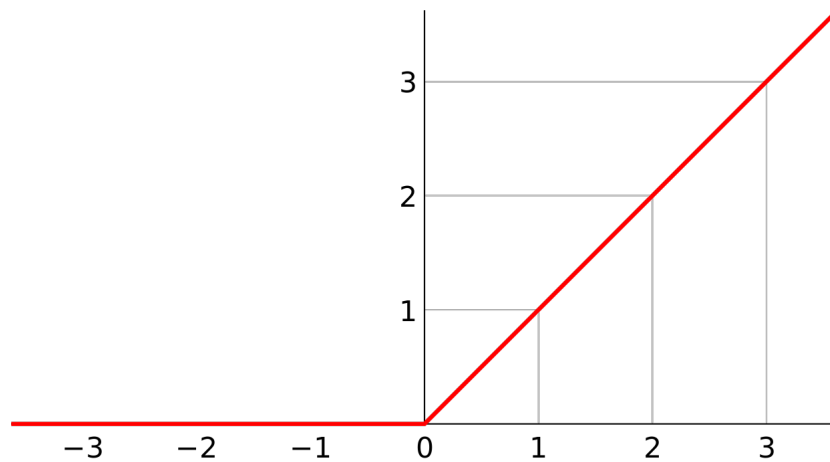
Nos últimos anos, várias funções de ativação lineares vem se tornando mais populares, a ReLU é uma delas (AGGARWAL et al., 2018). ReLU é o acrônimo para Rectified Linear Unit, a função é linear e contínua mas não diferenciável em $x = 0$ e é derivável em 0 e para $x < 0$. Na prática ela funciona muito bem e tem a vantagem de ser rápida para ser calculada. Como ela não possui valor máximo de saída, isso ajuda com a reduzir problemas durante o treino com o Gradiente Descendente (GÉRON, 2019).

Sua equação é dada por:

$$y(x) = \max\{x, 0\} \quad (2.3)$$

O gráfico que representa essa função está na figura 3 abaixo.

Figura 3 – Gráfico da função de ativação ReLU



Fonte: (WIKIPEDIA, 2022a)

2.2.1.4 Softmax

A função transforma as saídas para cada classe para valores entre $[0, 1]$ e também garantirá que as probabilidades estimadas estejam entre $[0, 1]$ e quem somam 1 (GÉRON, 2019). A sua saída é probabilística.

Sua equação é dada por:

$$y(x)_j = \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}} \quad \text{para } j = 1, \dots, k \quad (2.4)$$

É útil pensar nesses valores de j como valores para saída de neurônios, onde as entradas as entradas são $j_1 \dots j_k$ (AGGARWAL et al., 2018).

2.2.2 Funções de otimização

Treinar uma rede neural muito profunda pode ser muito lento, porém com o uso de um otimizador mais rápido que o otimizador Gradiente Descente, podemos ganhar mais velocidade no treinamento (GÉRON, 2019). Embora as redes neurais presentes nesse trabalho não sejam consideradas redes neurais profundas, o uso dessa técnica tende a trazer ganho para a fase de treinamento. A seguir serão descritos os otimizadores usados neste trabalho.

2.2.2.1 SGD

A função SGD, que significa *Stochastic Gradient Descent*, que apenas escolhe uma amostra aleatoriamente do conjunto de dados de treinamento em cada época e calcula os gradientes com base nessa única amostra. O que é um ponto positivo, uma vez que torna algoritmo muito mais rápido pois possui poucos dados para calcular por época, que também o torna vantajoso para treinamento com grandes conjuntos de dados (GÉRON, 2019). Cada gradiente local pode ser calculado de forma eficiente, o que torna a descida do gradiente mais rápida ao custo da precisão do cálculo do gradiente (AGGARWAL et al., 2018).

Uma propriedade interessante dessa função de otimização é que embora não tenha um bom desempenho na etapa de treinamento da rede neural, em comparação a versão do Gradiente Descente, ele geralmente tem um desempenho comparável, ou até as vezes melhor, nos dados de teste (AGGARWAL et al., 2018). Devido a sua natureza aleatória, a função SGD é muito menos regular que a Gradiente Descente, dado que a função irá fazer saltos para cima e para baixo diminuindo apenas em média, em vez de ir diminuindo suavemente. Com o tempo, a função chegará próximo do mínimo mas continuará oscilando (GÉRON, 2019).

2.2.2.2 AdaGrad

A função AdaGrad diminui a taxa de aprendizado mas o faz mais rápido para dimensões mais ingênuas do que para dimensões com inclinações mais suaves. Essa característica é chamada de taxa de aprendizagem adaptativa, o que ajuda a encontrar atualizações de valores mais diretamente para o ótimo global (GÉRON, 2019).

Essa função normalmente tem boa performance para problemas quadráticos simples, mas muitas vezes, o treinamento da rede pode parar cedo demais, pois a taxa de aprendizado é reduzida tanto que o algoritmo acaba parando antes de atingir o ótimo global (GÉRON, 2019).

2.3 Balanceamento de dados

Um conjunto de dados é considerado desbalanceado quando existe uma desproporção entre a quantidade de amostras de uma ou mais classes em relação às demais classes. Como, por exemplo, em estudos de uma doença rara em uma população, onde o número de pessoas portadores da doença é muito menor do que o número de pessoas que não portam a doença,

criando assim uma grande diferença entre a classe de pessoas que portam a doença e as que não portam a doença (BARELLA, 2015).

Nessas situações onde não existe uma proporção de quantidade de amostras por classe, os algoritmos de aprendizado de máquina tradicionais não têm conseguido bons resultados em classificar os dados. Isso ocorre pois, apesar de classificar corretamente os dados com classes de maior proporção, as classes que contém menos amostras não são classificadas corretamente. Em grande parte das vezes, essas classes minoritárias são de maior interesse (BARELLA, 2015).

Uma das abordagens usadas em balanceamento de dados é o pré-processamento de dados, onde tem o objetivo de balancear a distribuição de quantidade de amostras na distribuição original dos dados. Tais mecanismos incluem a subamostragem do conjunto majoritário ou sobreamostragem do conjunto minoritário. A subamostragem (ou *undersampling*) remove elementos da classe majoritária, podendo ser de forma aleatória ou por meio de algum critério de seleção, a fim de balancear as classes. A sobreamostragem (ou *oversampling*) adiciona amostras na classe minoritária, podendo ser a replicação dessa classe de forma aleatória ou informativa, ou gerar dados artificialmente (BARELLA, 2015).

2.4 Séries temporais

Uma série temporal é constituída por um conjunto de dados ou variáveis ordenadas no tempo, ou qualquer variável que desempenhe a função de índice (NASCIMENTO et al., 2017). O conjunto de dados ordenados no tempo pode ser discreto como o número de casos notificados de uma doença específica, ou contínua como o registro dos valores de temperatura e umidade ao longo do dia (LATORRE; CARDOSO, 2001).

Uma das razões de interesse na análise de séries temporais é adquirir conhecimento de padrões a medida que se desenvolve em um período de tempo, feito através de análise de dados passados. Posteriormente os modelos criados são utilizados para projetar o que é provável de ocorrer em um curto período. Os resultados das análises podem ser utilizados, principalmente, nas áreas de Economia e Finanças (OLIVEIRA, 2007).

2.5 Candlesticks

Entre as muitas ferramentas disponíveis para o investidor, encontra-se a análise de gráficos de *candlestick*. A tradução do termo *candlestick* significa “candelabro” e seu nome se deve ao formato como são dispostos os preços dos ativos no gráfico, dando a cada período um formato semelhante de uma vela (ou *candle*, em inglês), que pode possuir um “pavio” na parte inferior ou superior (DEBASTIANI, 2007).

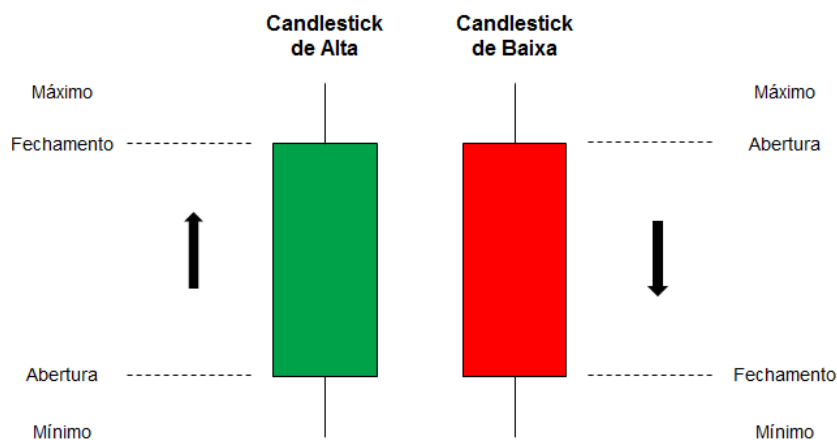
Ao analisar o mercado, podemos usar vários intervalos de tempo no estudo das negociações, podendo ser um dia, uma semana ou períodos de minutos dentro de um dia. No final de cada dia de pregão, cada ativo tem cinco preços divulgados, que são:

1. Preço de abertura, o preço pelo qual foi fechado o primeiro negócio do dia.

2. Preço de encerramento, o preço pelo qual foi fechado o último negócio do dia.
3. Preço mínimo, o preço mais baixo que foi negociado naquele dia.
4. Preço máximo, o preço mais alto que foi negociado naquele dia.
5. Preço médio, divisão do volume financeiro no dia pela quantidade de títulos negociados.

O processo de montagem do gráfico consiste em representar cada pregão como uma “vela”, onde o “corpo da vela” dá o espaço ocupado entre o preço de abertura e de fechamento, e os preços máximo e mínimo formam os “pavios” acima e abaixo do “corpo da vela”, se forem preços diferentes de abertura e fechamento. Os pavios também são chamados de “sombra”. A cor que representa o corpo do *candle* é definida conforme os preços do período definido no gráfico. Caso o preço de fechamento for maior que o de abertura, o *candle* comumente será branco ou verde. Caso o preço de fechamento for menor que o de abertura, ele comumente assumirá a cor preto ou vermelho (DEBASTIANI, 2007). As cores dos *candles* podem ser alteradas a gosto da pessoa, mas as cores citadas anteriormente são as mais tradicionais. A figura 4 a seguir mostra um *candle* de alta e um *candle* de baixa e a figura 5 a seguir representa um gráfico de *candles*. De todos os preços divulgados durante o pregão, somente o preço médio não é utilizado na análise do gráfico.

Figura 4 – Representação de um *candle* de alta e de baixa



Fonte: (INVESTIDOR, 2021)

Figura 5 – Gráfico de *candlestick*

Fonte: (INVESTIDOR, 2021)

Note que no gráfico de *candlestick* é muito mais rico em informação do que se fosse plotado somente o preço de fechamento. Nesse gráfico é possível ver a evolução dos preços durante o pregão, os pontos de maior euforia marcado por *candles* mais longos, a variação dos preços ao longo do pregão. A cor dos *candles* que diferencia altas e baixas, o que dá a noção de “correção de preço”, durante um período. No início do pregão, os preços podem não iniciar com os mesmos valores do pregão anterior. Caso isso ocorra, forma-se uma lacuna de preços em que não houve negociação, chamado *gap* (DEBASTIANI, 2007).

2.5.1 Formação temporal e atemporal

Existem duas categorias de gráficos usados para se operar na bolsa de valores, os gráficos temporais e os atemporais. Nos gráficos temporais, o operador acompanha a formação dos *candles* com base em um tempo determinado por ele, esse tempo pode ser 1 minuto, 5 minutos, 10 minutos, 30 minutos, 1 dia, 1 semana, 1 mês, entre outros. Isso significa que a cada período uma vela é formada com seus valores de preços mínimo, máximo, abertura e de fechamento (CLEAR, 2018). A figura 6 a seguir, mostra um gráfico de tempo de 1 minuto do ativo mini índice Bovespa.

Figura 6 – Gráfico de *candlestick* com tempo de 1 minuto

Fonte: (TRADINGVIEW, 2021)

Uma categoria de gráfico que vem crescendo no mercado é o gráfico atemporal. Esse caso, os *candles* não usam como base o fator tempo para ocorrer a sua formação. A sua formação é dada pela flutuação do valor de um ativo ou contrato, tendo seus *candles* formados pela quantidade de pontos ou *tickers* (CLEAR, 2018). Existem vários gráficos atemporais, mas neste trabalho forcaremos no gráfico *Renko*.

O gráfico *Renko*, tem como formação de *candle* a partir do fechamento do anterior. Se o *candle* anterior for positivo e a variação do mercado for para cima, 2 *tickers* são necessários para formar um *candle*. Para formar um *candle* negativo, deve ter uma variação de 5 *tickers* para baixo. A figura 7 a seguir, mostra o gráfico *Renko* para o ativo mini índice Bovespa.

Figura 7 – Gráfico *Renko*

Fonte: (TRADINGVIEW, 2021)

2.6 Mercado financeiro de *Day Trade*

Segundo (CHAGUE; DE-LOSSO; GIOVANNETTI, 2020), *Day Trade* é a atividade de comprar e vender um mesmo ativo financeiro no mesmo dia na mesma quantidade. O objetivo de lucrar com pequenas oscilações do mercado, tendo como principais vantagens a agilidade nas operações e a possibilidade de operar alavancado ¹ (CLEAR, 2021a).

Mercado futuro é um ambiente da bolsa de valores onde são negociados contrato de compra e venda de mercadorias como produtos alimentícios, moedas, commodities, juro e o próprio Índice Bovespa. Os contratos serão realizados em um data futura (CLEAR, 2021b). Ao operar nesse mercado, o investidor terá resultados diários, ou seja, o preço do ativo será ajustado diariamente, tendo assim a possibilidade de acompanhar a diferença entre o valor negociado e o valor de mercado no dia (MODALMAIS, 2021).

2.7 Indicadores técnicos

Os indicadores técnicos são fórmulas matemáticas, que usam valores de preço e/ou volume, plotadas no gráfico de um ativo, onde a própria plataforma utilizada pelo investidor ou pelo *trader* traça automaticamente. O seu objetivo é fornecer informações de modo a auxiliar a tomada de decisão sobre como e quando fazer transações. É uma ferramenta primordial para se realizar a análise técnica (METAQUOTES, 2021b), (FUTURA, 2020).

Conforme as suas propriedades funcionais, é possível classificar os indicadores de dois tipos, os indicadores de tendência e os osciladores. Os indicadores de tendência monitoram o preço de um ativo para determinar se há divergência ou convergência em relação à tendência. Os indicadores osciladores formam linhas que acompanham ou não o gráfico de preços, oscilando entre um intervalo fixo. Também indicam possíveis pontos de reversão de tendência (FUTURA, 2020). Abaixo são apresentados os indicadores utilizados nesse trabalho.

2.7.1 Média Móvel Simples

O indicador média móvel mostra o valor médio dos preços em um período de tempo. Quando os preços variam, a sua média móvel também varia, podendo aumentar ou diminuir seu valor. Pode ser calculada para qualquer conjunto de dados sequenciais, como preços de abertura e fechamento, máximas e mínimas de preço, volume de negociação ou outros indicadores.

A média móvel não foi projetada para fornecer entradas no mercado diretamente no fundo e a saída exatamente em um topo. Ela permite comprar logo após os preços atingirem um fundo e vender logo após os preços atingirem o seu topo.

A média móvel simples é calculada através da soma dos preços de fechamento ao longo de um certo número de períodos individuais. Este valor é então dividido pelo número de

¹ Operar alavancado é a realização de operações de compra e venda de ativos com valores superiores aos que possui na conta da corretora, assim há a possibilidade de multiplicar os ganhos mas também as perdas. Esse valor é emprestado pela própria corretora e deve ser devolvido quando receber o retorno do investimento.

tais períodos. É dado pela seguinte fórmula:

$$MMS(p) = \frac{\sum_{i=1}^p F_{i-1}}{p} \quad (2.5)$$

onde:

MMS representa o valor da média móvel simples

p representa o número de períodos desejado

F_t representa o preço do fechamento do ativo no instante t

2.7.2 Média Móvel Exponencial

A única coisa que faz com que a média móvel simples se diferencie da média móvel exponencial, é o comportamento dos coeficientes, onde na variação exponencial, tem um peso maior aos preços mais recentes, enquanto na média móvel simples os coeficientes possuem o mesmo peso.

A média móvel exponencial é calculada pela soma de uma determinada parte do preço de fechamento atual, no valor anterior da média. A fórmula é dada da forma a seguir:

$$MME(p) = F_i \cdot \left(\frac{2}{p+1} \right) + MME(F_{i-1}) \cdot \left(1 - \left(\frac{2}{p+1} \right) \right) \quad (2.6)$$

onde

MME representa o valor da média móvel exponencial;

p representa o número de períodos desejado;

F_t representa o preço do fechamento do ativo no instante t ;

2.7.3 Índice de Movimento Direcional

O índice de movimento direcional, ou ADX, é um indicador utilizado para medir a força de uma tendência. O índice foi desenvolvido em 1978 por J. Welles Wilder, em seu livro *New Concepts In Technical Trading Systems* com o intuito de analisar a tendência de preços de ativos do segmento de *commodities* (INVESTIDOR, 2013).

Esse indicador é composto por duas linhas, o índice direcional positivo (+DI) e o índice direcional negativo (-DI). Cada um deles indica se a tendência está em alta ou queda. Quando +DI começar a subir acima do -DI, sugere que a tendência dos preços é de alta, e o inverso é verdadeiro, quando a linha do -DI está acima do +DI, sugere que a tendência é de queda. A escala do índice é de 0 a 100. Quando as linhas se encontram até o nível de 20 pontos não há uma clara tendência, somente quando é ultrapassado os 40 pontos, que segure uma forte tendência (INVESTIDOR, 2013). A fórmula é dada a seguir:

$$ADX(p) = MA \cdot \left(\frac{(+DI) - (-DI)}{(+DI) + (-DI)} \right) \cdot 100 \quad (2.7)$$

onde

ADX representa o resultado do índice;

p representa o período;

$+DI$ representa a direção positiva;

$-DI$ representa a direção negativa;

$MA(x)$ representa a função de média móvel

2.7.4 Estocástico

O indicador estocástico compara o preço de fechamento de um ativo em relação à sua faixa de preço ao longo de um determinado período. Ele é apresentado por duas linhas. A linha principal é chamada %K, sendo normalmente apresentada em forma sólida e a segunda linha é chamada %D, sendo ela uma média móvel de %K e representada pontilhadamente. A fórmula de %K é dada por:

$$\%K(p) = 100 \cdot \left(\frac{F_t - L_p}{H_p - L_p} \right) \quad (2.8)$$

onde

%K representa a curva do estocástico %K

p representa o período;

F_t representa o preço de fechamento;

H_p representa o preço máximo no período;

L_p representa o preço mínimo no período;

A fórmula de %D, temos:

$$\%D(p) = MMS \cdot (\%K, p) \quad (2.9)$$

onde

%D representa a curva do estocástico %D

MMS representa a média móvel simples;

Na figura 8 a seguir, é representada o indicador Estocástico em um ativo.

Figura 8 – Gráfico do indicador estocástico



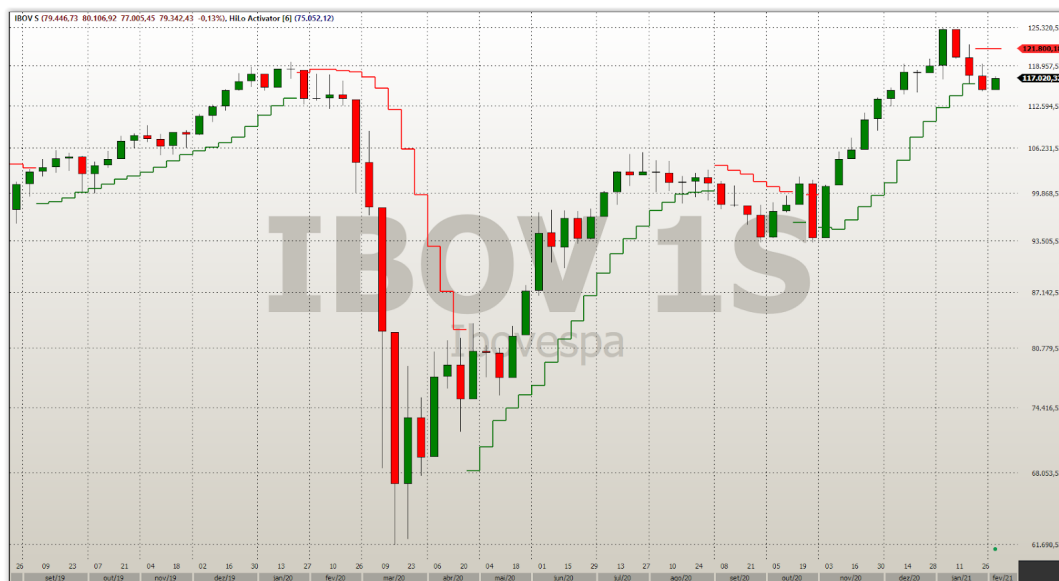
Fonte: (METAQUOTES, 2021b)

2.7.5 HiLo Activator

O *HiLo Activator* é um indicador que segue tendências. Se baseia na média móvel das máximas e mínimas dos últimos períodos. Seu nome vem de *High* (alta) e *Low* (baixa) em inglês. Esse indicador aponta em qual momento devemos fazer uma negociação de compra ou de venda. Também sinaliza possíveis reversões de tendência, de baixa para alta ou de alta para baixa. Esse indicador não apresenta bom desempenho em mercados sem tendência ou tendência lateral (MOORE, 2021).

A figura 9 representa o indicador HiLo aplicado no Ibovespa. Quando o preço do ativo está acima do *HiLo*, representado pela escadinha em cor verde, o ativo está com tendência de alta então é o momento de comprar. Quando o preço estiver abaixo do indicador, representado pela escadinha em cor vermelho, o ativo está com tendência de baixa, assim é o momento para vender. O momento para se fazer entradas no ativo é quando o preço do ativo vêm abaixo do *HiLo* e passa a ficar acima (muda de vermelho para verde), gerando compras, ou quando o preço vem acima do *HiLo* (muda de verde para vermelho) e passa a ficar abaixo, gerando venda. As saídas das operações acontecem nas viradas de direção (MOORE, 2021).

Figura 9 – HiLo Activator Indicador de um Gráfico Ibovespa Semanal



Fonte: (MOORE, 2021)

2.8 Mini Índice Bovespa

Ibovespa Futuro é um contrato derivativo negociado no mercado futuro. Esse contrato é realizado baseado na expectativa do seu valor dada em pontos, que o índice terá em uma data futura. Portanto, esse investimento tem seus ganhos ou perdas atrelados ao índice, sempre tendo uma data de vencimento e uma expectativa de pontos para essa data (CLEAR, 2021b). Considere o seguinte cenário:

Supondo que o Índice Bovespa no dia 1 de agosto de 2022 esteja em 125.000 pontos. Nesse dia foi negociado um contrato futuro usando esse índice para a data de 1 de setembro de 2022, com uma expectativa de 129.000 pontos. Para saber se foi um bom negócio ou não, no dia que o contrato vencer, se observa a pontuação atingida no dia do vencimento. Caso a pontuação nesse dia seja menor a 129.000 pontos, o investidor terá perda, pois pagou por um contrato de valor superior. Contudo, caso a pontuação do índice no dia do vencimento for maior que 129.000 pontos, o investidor realizou um bom negócio.

É possível negociar o contrato cheio e o minicontrato futuro de Ibovespa. O contrato cheio é calculado a partir dos pontos do índice, e cada ponto corresponde a R\$ 1,00, sendo que o lote mínimo para se investir é de 5 lotes (CLEAR, 2021b). Assim caso a pontuação do índice Bovespa no dia seja 127.000 pontos, o contrato cheio será de R\$ 127.000,00. Conforme dito anteriormente, esse valor pode ser alto para um investidor comum, e para isso existe o contrato futuro mini de Ibovespa. Esse mini contrato tem o valor de R\$ 0,20 por ponto no índice, ou seja, 20% do índice e tendo o lote mínimo de 1 mini contrato (CLEAR, 2021b). Na figura 10 a seguir é possível visualizar a variação dos pontos do mini índice no intervalo de agosto de 2020 a agosto de 2021.

Figura 10 – Mini índice Bovespa



Fonte: (TRADINGVIEW, 2021)

Durante o horário de negociação, o ativo tem o código formado por três letras indicando o código do minicontrato de índice, seguindo de uma letra seguindo de dois algarismos. No caso do mini índice Bovespa o código é o WIN, seguido pelo mês da letra correspondente ao mês de vencimento do contrato, seguindo a tabela 1 abaixo (B3, 2020a).

Tabela 1 – Código e vencimento

Vencimento	Letra
Fevereiro	G
Abril	J
Junho	M
Agosto	Q
Outubro	V
Dezembro	Z

Fonte: (B3, 2020a)

Os contratos futuros de índice possuem vencimentos a cada dois meses, apenas nos meses pares do ano. O vencimento ocorre sempre na quarta-feira mais próxima do dia 15 do mês de vencimento. E por fim, a terceira e última parte do código são os dois últimos algarismos do ano de vencimento (B3, 2020a). Por exemplo, um contrato com o código WINQ22 representa um mini índice Bovespa com o vencimento para o mês de agosto do ano de 2022.

2.9 Técnica de negociação

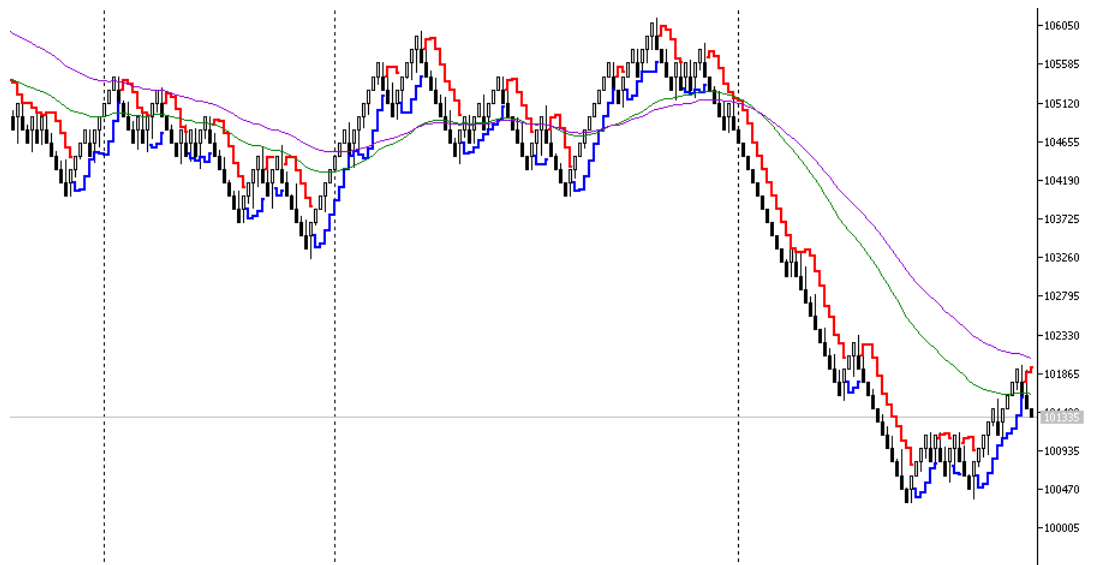
O *Day Trade* é uma das estratégias de negociação em curtíssimo prazo. Por ser bastante dinâmica, existe bastante concentração e rapidez do operador, no qual muitas vezes, passa o dia acompanhando o mercado para identificar boas oportunidades de operar (GENIAL, 2021a). Uma das técnicas que são de grande importância para *traders* poderem operar no mercado financeiro de modo a obter resultado positivos e assertivos é o *setup* operacional. O *Setup* é a maneira de como se operar no mercado, ou seja, são o desenvolvimento de estratégias pré-definidas para os auxiliam para tomar decisões de compra ou venda de um ativo (PINHEIRO, 2021).

2.9.1 Setup Operacional

Um *setup* é o conjunto de técnicas e parâmetros definidos pelo operador, nesse caso o *trader*, para executar operações de compra e venda um ativo ou derivativo financeiro. A definição dos parâmetros para se realizar compras e vendas pode ser realizadas de diversas formas. Nele será definido, por exemplo, o gatilho de entrada para que abra uma operação, o *stop loss*, *stop gain* e os indicadores a serem usados. A configuração do *setup* é feita com base na análise gráfica, de modo que caso o gráfico fizer uma formação gráfica específica, o *setup* faz uma operação conforme configurado (GENIAL, 2021a).

Traders de alto desempenho possuem estratégias bem definidas, porém a unanimidade entre esse público, diz que cada operador tem seu próprio jeito de operar. Eles constroem o *setup* de acordo com às suas necessidades, logo é uma ferramenta muito pessoal. Existem *traders* que aceitam tomar mais riscos em troca de um maior potencial de retorno ou aqueles que operam grandes quantias financeiras (GENIAL, 2021a), (PINHEIRO, 2021). O *HiLo* é um *setup* muito utilizado pelos *traders* para fazerem operações.

A figura 11 abaixo, é um exemplo de *setup* operacional utilizando o *HiLo*. As tendências de alta nos *candles* do gráfico Renko são marcados por linhas da cor azul e as tendências de queda são marcadas pelas linhas vermelhas. A linha roxa representa a média móvel de 66 períodos e a linha verde representa a média móvel de 44 períodos.

Figura 11 – Exemplo de *setup* operacional

Fonte: (METAQUOTES, 2021b)

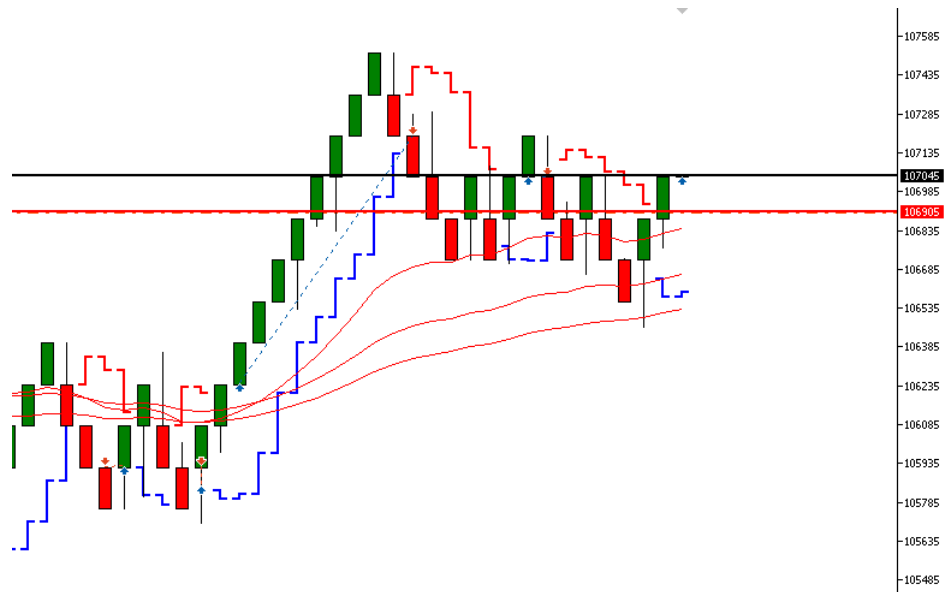
2.9.2 Stop Loss

Perdas em bolsas de valores nem sempre pode ser evitado, principalmente em operações de alta volatilidade como as de *Day Trade*. É necessário estar preparado para os resultados negativos. Para isso, podemos usar um mecanismo de proteção de perdas muito elevadas, o *Stop Loss*. Ou seja, a principal função do *Stop Loss* é evitar a perda de dinheiro indefinida (GENIAL, 2021c).

Stop Loss, em tradução livre, “interromper ou parar a perda”, é um mecanismo que executa uma ordem automaticamente para fechar uma posição quando o limite de perda definido pelo *trader* é alcançado. Depois que o operador estabeleceu o limite, ele será usado para encerrar a posição caso o preço atinja o limite. Para quem está posicionado em uma operação de compra, o encerramento consiste na venda do ativo ou produto financeiro e para esta posicionado em uma operação de venda, o mecanismo gera uma ordem de compra (GENIAL, 2021c).

A figura 12 a seguir, ilustra um exemplo de *Stop Loss* que criada quando uma ordem de compra foi emitida. A linha contínua em vermelho representa o *Stop Loss* e a linha em preto representa o valor de abertura da ordem de compra. Caso o valor do ativo chegue na linha vermelha, a operação será fechada automaticamente em prejuízo.

Figura 12 – Exemplo de gráfico com Stop Loss



Fonte: (METAQUOTES, 2021b)

2.9.3 Stop Gain

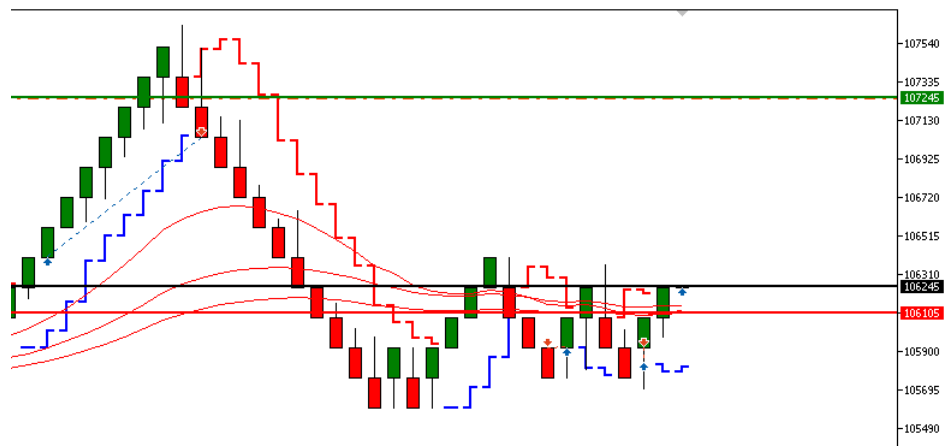
Operar em mercado de renda variável envolve riscos, principalmente quem opera em curtíssimo prazo como no *Day Trade*, pois a volatilidade dos preços durante o dia pode fazer uma operação de lucro se reverter em segundos (GENIAL, 2021b).

O *Stop Gain* é um mecanismo de proteção utilizado em bolsa de valores que encerra uma operação enquanto está com lucro, definindo um limite de ganhos pelo *trader* nas operações. Essa ordem é realizada automaticamente, sem a ação do operador. Essa operação pode ser lançada tanto para sair de uma posição de compra quanto uma posição de venda, trazendo uma segurança maior para o *trader* mesmo ele sem esta acompanhando o mercado (GENIAL, 2021b).

Com isso, o *Stop Gain* se torna uma forma de garantir que uma operação saia com lucro e prevenir prejuízos. Porém, ele também corre o risco de perder o lucro obtido até o momento por não encerrar a posição, uma vez que o lucro e o prejuízo só acontece quando a operação se encerra (GENIAL, 2021b).

A figura 13 a seguir, ilustra um exemplo de *Stop Gain* que criada quando uma ordem de compra foi emitida. A linha contínua em verde representa o *Stop Gain*, a linha vermelha é a *Stop Loss* e a linha em preto representa o valor de abertura da ordem de compra. Caso o valor do ativo chegue na linha verde, a operação será fechada automaticamente em lucro.

Figura 13 – Exemplo de gráfico com Stop Gain



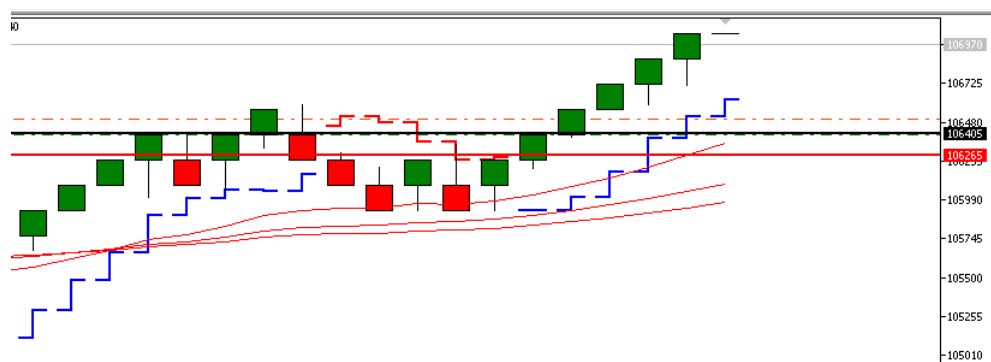
Fonte: (METAQUOTES, 2021b)

2.9.4 Trailing Stop

É uma técnica de gestão de risco em que o operador determina um nível para que o *Stop Loss* acompanhe o preço atual do mercado, seguindo um valor absoluto ou uma porcentagem. É uma variação do *Stop Loss* só que em vez de um preço de parada fixo, o *trailing Stop* segue o preço de mercado a uma distância predefinida durante uma tendência. Com isso pode maximizar o lucro e minimizar o potencial de perda. Esse mecanismo não é usado como uma ordem de entrada, os *traders* usam quando já possuem uma posição em aberta que já possui lucro e o mercado ainda está se movendo (NIKOLAEV, 2020).

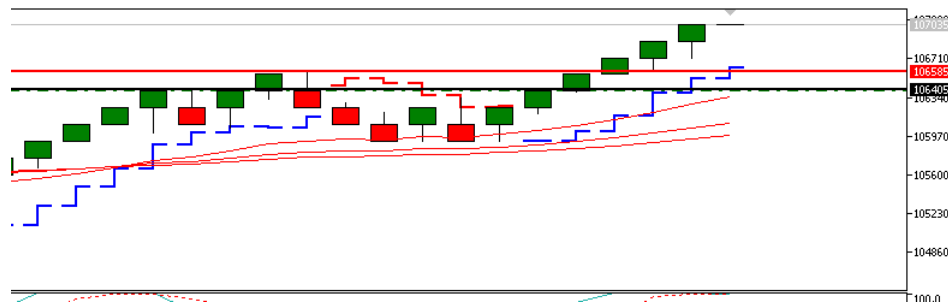
As figuras 14 e a 15 a seguir, ilustram um exemplo de *Trailing Stop* que criada quando uma ordem de compra foi emitida. Na figura 14, a linha contínua em vermelho representa o *Stop Stop* que até então está posicionada em 106265 pontos. Depois de mais alguns minutos de negociação, a linha vermelha é movida 320 pontos para a posição de 106585 pontos. Essa mudança de posição do *Stop Loss* se deu pois o valor do ativo naquele momento teve um aumento de 106970 pontos para 107035 pontos, assim para ter uma margem de lucro maior e minimizando o prejuízo, a linha vermelha foi movida para cima no gráfico.

Figura 14 – Exemplo de gráfico com Trailing Stop antes de mudar a referência



Fonte: (METAQUOTES, 2021b)

Figura 15 – Exemplo de gráfico com Trailing Stop depois de mudar a referência



Fonte: (METAQUOTES, 2021b)

2.10 Plataformas de negociações

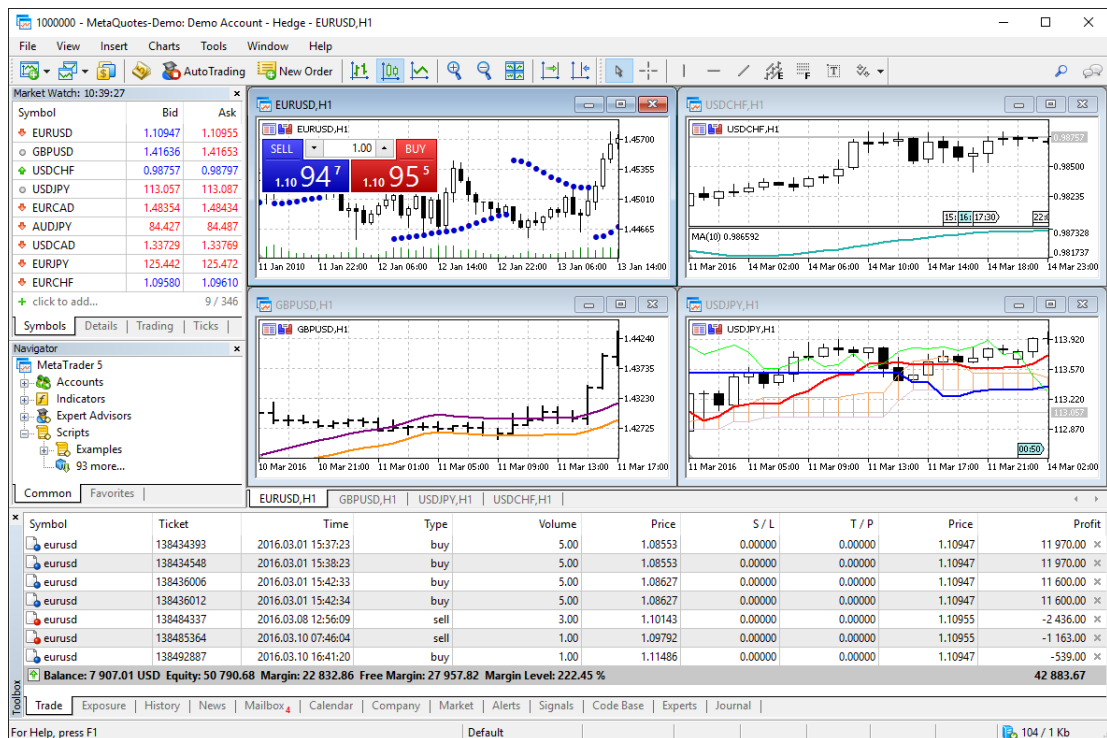
A plataforma de investimento é um ambiente online, disponibilizado por corretora de investimentos ou por bancos digitais, que visam possibilitar que os investidores tenham acesso a diversos produtos, desde renda fixa à renda variável, e possam realizar aplicações por conta própria. Ela também oferece informações e serviços úteis aos investidores, para que os auxiliem em tomadas de decisões (MODALMAIS, 2021). Existem várias opções de plataformas para os investidores, nessa sessão será abordada alguns exemplos.

2.10.1 Metatrader

É uma plataforma que oferece funções financeiras avançadas para fazer negócios, bem como ferramentas para se realizar análise técnica e fundamentalista. A ferramenta também cria a possibilidade de negociar usando robôs e sinais de negociação. A plataforma oferece um ambiente de desenvolvimento na linguagem MQL5, especializado para programação algorítmica. A IDE MQL5 permite que *traders* e programadores desenvolvam, depurem, testem e otimizem seus próprios robôs de negociação (METAQUOTES, 2021a). A figura 16 a seguir mostra a tela inicial do Metatrader 5.

A MetaQuotes Language 5 (MQL5) é uma linguagem de programação orientada a objetos de alto nível baseada na linguagem C++. Todos os aplicativos de negociação são escritos em MQL5. A sua principal vantagem frente a outras linguagens de programação é a sua especialização de negociação, visto que ela possui uma grande gama de funções integradas para análise técnica e permite a criação de aplicações para análise automatizada de negociação (METAQUOTES, 2021a).

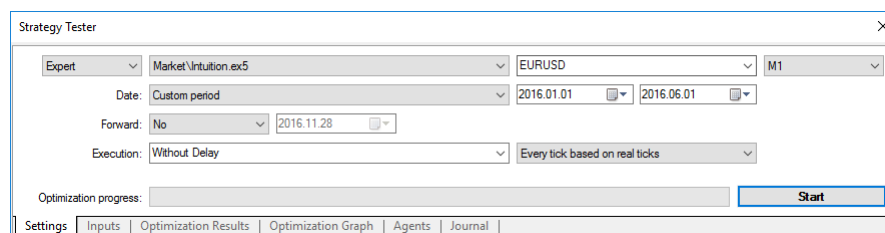
Figura 16 – Metatrader 5



Fonte: (METAQUOTES, 2021a)

A IDE MQL5 possui um testador de estratégia integrado, que permite aos usuários avaliar os seus robôs antes de usá-los em uma conta real. As operações são baseadas no histórico de preços dos ativos financeiros, realizando as operações virtuais segundo a sua programação. Com esses testes permite avaliar a eficiência do robô assim como simular comportamento durante a negociação ao vivo (METAQUOTES, 2021a). A figura 17 a seguir é a tela de configurações do testador de estratégia do Metatrader 5.

Figura 17 – Tela do testador de estratégias



Fonte: (METAQUOTES, 2021a)

No Metatrader existem dois sistemas de contabilidade de posição da conta de negociação: o Netting e o Hedging. Pelo sistema Netting, é permitido somente uma posição aberta para um ativo ao mesmo tempo. Assim, caso já tenha uma posição aberta, a execução de uma nova ordem aumenta a posição já existente caso seja do mesmo tipo. Se caso a ordem for de tipo diferente a que está aberta, o volume da negociação total pode diminuir, ser encerrado ou

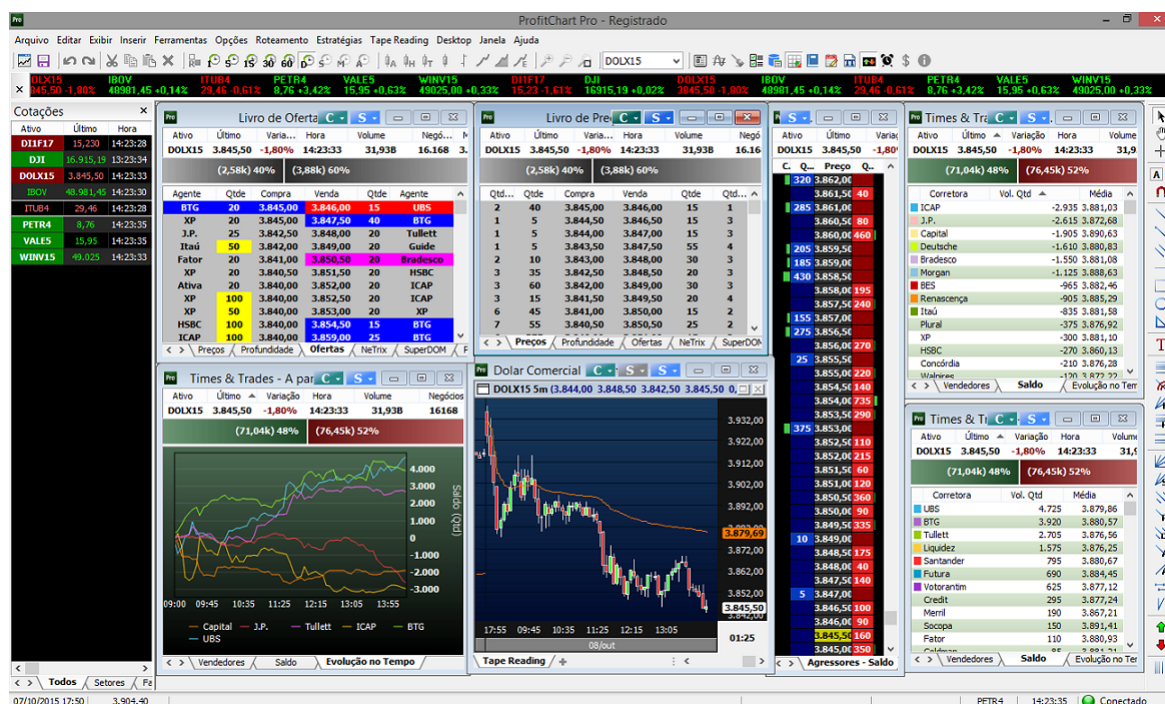
reverter a posição (METAQUOTES, 2021a).

Pelo novo sistema Hedging é possível abrir múltiplas posições de compra e/ou venda em um mesmo ativo ao mesmo tempo. Assim, quando tem uma posição aberta e executa uma nova ordem, uma nova posição é adiciona no mesmo ativo, sem alterar a posição anterior. Esse sistema abre a possibilidade de uma nova estratégia de negociação baseado no chamado “bloqueio”, uma vez que caso o preço do ativo mover contra o *trader*, ele poderá abrir uma posição na direção oposta (METAQUOTES, 2021a).

2.10.2 Profit Pro

O Profit Pro é uma plataforma da empresa Nelogica utilizada principalmente por *traders*, que possibilita o acesso a recursos avançados de análise técnica e possui diversas ferramentas para se realizar operações *Day Trade*. Através dela é possível definir estratégias, acompanhar o mercado e fazer o gerenciamento de operações e posicionamento nos mais diversos ativos. Assim como o Metatrader, o Profit Pro é integrada a diversas corretoras e é uma das conceituadas e populares do mercado. A figura 18 a seguir ilustra uma configuração de tela da plataforma Profit Pro.

Figura 18 – Tela do Profit Pro



Fonte: (NELOGICA, 2021)

2.11 Frameworks e bibliotecas

Para a construção de softwares e diversas aplicações, o uso de frameworks e bibliotecas são de grande importância para um desenvolvimento com melhor desempenho, segurança e agilidade. No desenvolvimento web existem uma enorme gama de ferramentas para a

construção de aplicações robustas, rápidas e seguras. Algumas ferramentas utilizadas nesse trabalho serão descritas abaixo.

2.11.1 Flask

O Flask é um micro-framework multiplataforma escrito em linguagem de programação Python e disponível sobre a licença de código aberto, que provê um modelo simples para desenvolvimento web. Um micro-framework são frameworks modularizados que possuem uma estrutura muito mais simples quando comparados aos frameworks convencionais (ANDRADE, 2019).

O Flask foi lançado em 2010 e desenvolvido por Armin Ronacher, destinado a pequenas aplicações com requisitos mais simples. Embora tenha um núcleo simples, ele permite acrescentar novos pacotes que permite a sua expansão para um projeto que possua os recursos necessários para sua execução. Suas principais características são a simplicidade de desenvolvimento de uma aplicação, rapidez no desenvolvimento onde o desenvolvedor se preocupa somente com o necessário para o projeto sem a precisar realizar muitas configurações e possibilita aplicações robustas, que embora seja um micro-framework, permite criar softwares robustos, já que ele é totalmente personalizável (ANDRADE, 2019).

2.11.2 Tensorflow

O Tensorflow é uma plataforma de código aberto para criar projetos de *machine learning*. Ele possui um ecossistema abrangente e flexível de ferramentas, bibliotecas e recursos da comunidade, que permite aos pesquisadores e desenvolvedores criar e implantar aplicativos com tecnologia de aprendizado de máquina simplificada, independentemente da linguagem ou da plataforma utilizada (TENSORFLOW, 2021).

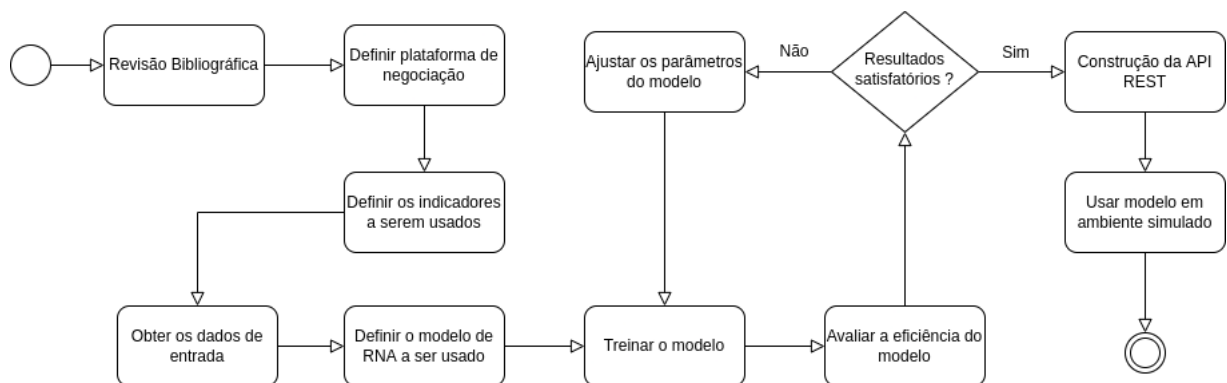
A ferramenta apresenta diversos níveis de abstração para as mais diversas necessidades, podendo assim o usuário focar seus esforços no modelo e na lógica ao invés de realizar toda a implementação manualmente. Usando a API Keras de alto nível, é possível criar e treinar modelos. Também é possível a API *Distribution Strategy* para treinamento distribuído em diferentes configurações de hardware sem mudar as configurações do modelo (TENSORFLOW, 2021).

3 Procedimentos Metodológicos

“O período de maior ganho em conhecimento e experiência é o período mais difícil da vida”.
Dalai Lama

Este capítulo destina-se a descrever de forma mais detalhada as etapas dos procedimentos metodológicos a serem realizadas neste trabalho. A figura 19 ilustra o fluxo dos procedimentos metodológicos deste trabalho.

Figura 19 – Procedimentos metodológicos para a construção do modelo



Fonte: Elaborado pelo autor

O modelo da rede neural será construído usando dados de gráficos atemporais, utilizando os mesmos indicadores que o trabalho de Silva (2021), para todas as topologias de rede. O gráfico atemporal que será usado é o gráfico *Renko* descrito na seção 2.5.1.

3.1 Classificação da pesquisa

Segundo (WAZLAWICK, 2014), este trabalho pode ser classificado, quanto aos objetivos, como uma pesquisa exploratória, pois visa examinar fenômenos do mercado financeiro através de um dos muitos ativos disponíveis na bolsa de valores, no caso o mini índice Bovespa, visando realizar previsões de preço do ativo para encontrar o melhor momento de compra e de venda. Quanto aos procedimentos técnicos, este trabalho pode ser classificado como pesquisa experimental, pois há um foco no estudo do ativo de mini índice e nos indicadores, objetivando a busca de melhores resultados para a rede neural.

3.2 Revisão bibliográfica

O primeiro item do procedimento metodológico é a realização de uma revisão bibliográfica descrita no Capítulo 2. As fontes de informações em que foram consultadas foram o

Portal de Periódico da CAPES, o *Google Scholar*, livros da área de rede neural e canais e sites voltados ao mercado financeiro.

3.3 Indicadores, plataforma de negociação e estratégia

Conforme foi dito anteriormente, os modelos terão os mesmos indicadores do trabalho de Silva (2021). Os indicadores que escolhidos foram a Média Móvel Simples, Média Móvel Exponencial, ADX e o Estocástico, além do HiLo Ativador usado para definir momentos de compra e venda. De acordo com Silva (2021), esses indicadores foram escolhidos por suas características principais de forma que trabalham melhor juntos.

Além dos indicadores citados, serão utilizados dados de valores de preço de abertura, preço de fechamento, preço máximo e preço mínimo. Também serão usados dados como volume de negociação, a amplitude do *candle* pegando os valores de abertura e fechamento, um sinal indicando se o preço de fechamento foi superior ao de abertura e por fim as médias móveis exponenciais de 21, 44 e 66 períodos.

Para a realização da codificação foi escolhida a plataforma Metatrader 5, devido aos seus extensos recursos de funcionalidades e de criação de robôs para fazer negociações, permitindo a automatização de operações.

Para realizar as negociações na plataforma, foram utilizadas estratégias de *Day Trade* para emitir as ordens de compra e venda do ativo financeiro. O *setup* escolhido foi o mesmo utilizado por Silva (2021), que utilizará o indicador HiLo para indicar mudanças de tendências.

3.4 Obtenção dos dados de entrada

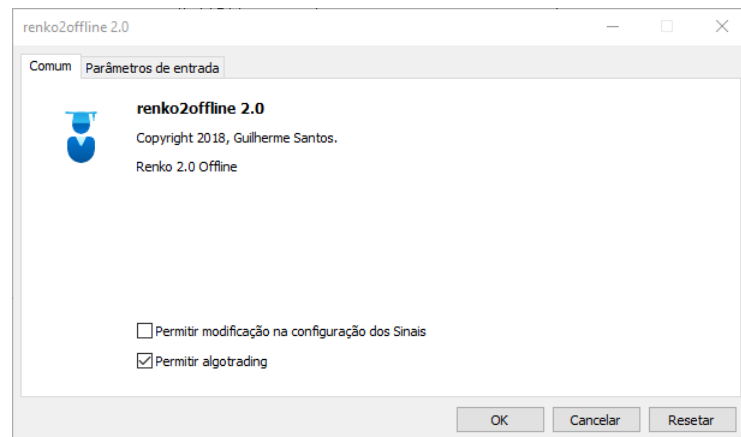
Para poder realizar o treinamento da RNA, é preciso obter uma base de dados para classificá-la para inserir na rede. Para obter tais dados, será implementado um algoritmo na plataforma Metatrader usando a linguagem MQL5, o qual irá fazer a leitura dos dados de cada indicador e das informações já citadas anteriormente e armazená-los em arquivos de texto.

3.4.1 Transformação do gráfico

O mini índice Bovespa possui um *ticker* onde ele armazena os valores dos *candles* de vários meses de negociação, o *WIN\$N*. Esse *ticker* não é possível fazer negociações, somente registra as informações dos *candles* de negociações do ativo que está vigente e das anteriores. Esse *ticker* será usado para coleta de dados, uma vez que possui um histórico considerável, e o ativo que está sendo negociável no momento, guarda somente as informações de negociações de no máximo dois meses e depois é trocado para outro *ticker*, conforme a tabela 1 de vencimentos dos ativos de mini índice Bovespa.

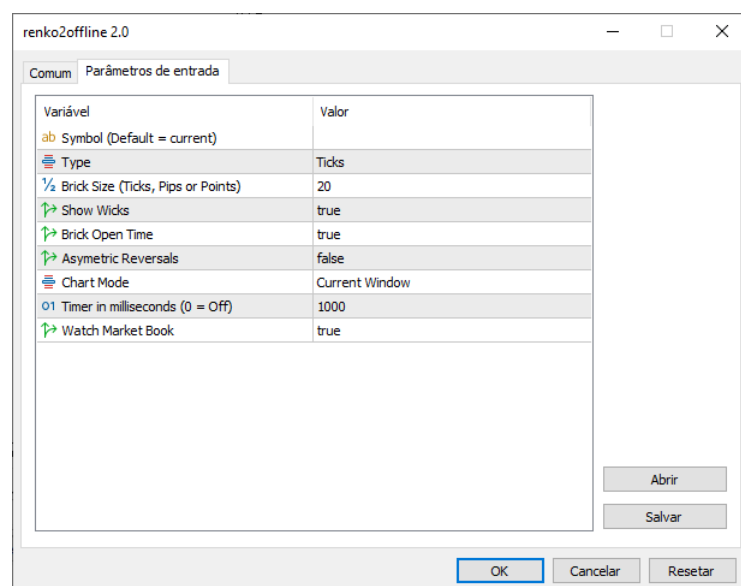
Para realizar essa transformação do gráfico temporal para o gráfico atemporal *Renko*, foi utilizado uma biblioteca *open source* disponibilizada na comunidade oficial da linguagem MQL5. A biblioteca usada é a “Renko 2.0 Offline” criado pela autor Guilherme Santos (SANTOS, 2018). As configurações do uso dessa biblioteca segue nas figuras 20 e 21 abaixo.

Figura 20 – Aba comum da biblioteca Renko 2.0 Offline



Fonte: Autor

Figura 21 – Aba de configurações da biblioteca Renko 2.0 Offline



Fonte: Autor

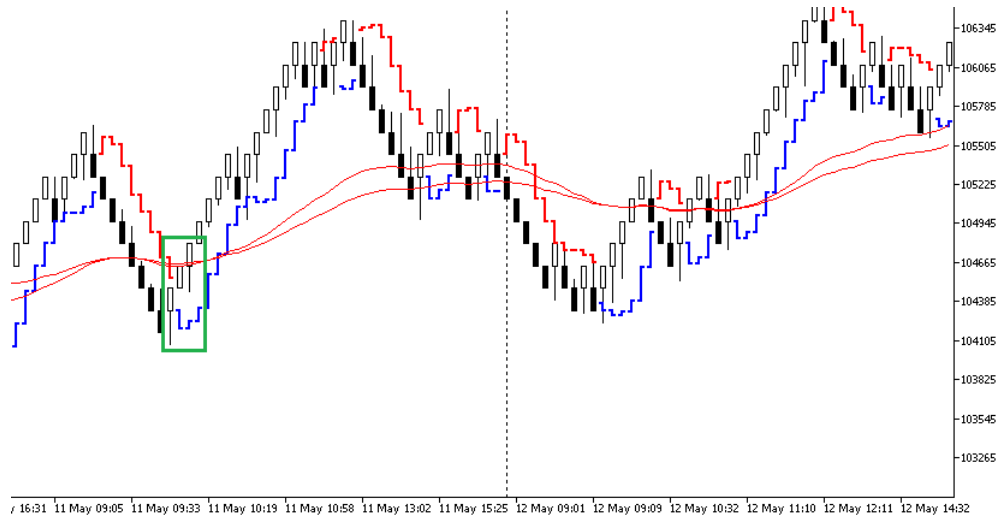
Essa mesma biblioteca será usada quando for usar o modelo em ambiente real utilizando uma conta de demonstração, porém agora não mais no ativo *WIN\$N* mas sim no ativo em negociação no momento da execução dos modelos. Esse ativo também deverá ser transformada em atemporal e os seus *candles* serão formados a medida que as negociações no mercado da bolsa de valores forem ocorrendo.

3.4.2 Classificação e armazenamento dos dados

A classificação se o sinal é do tipo venda ou do tipo compra, se dará pela saída do indicador HiLo. O sinal de compra se caracteriza pela troca de tendência de baixa para alta, indicada pelo indicador HiLo, e no momento seguinte ela for confirmada, conforme ilustrado na

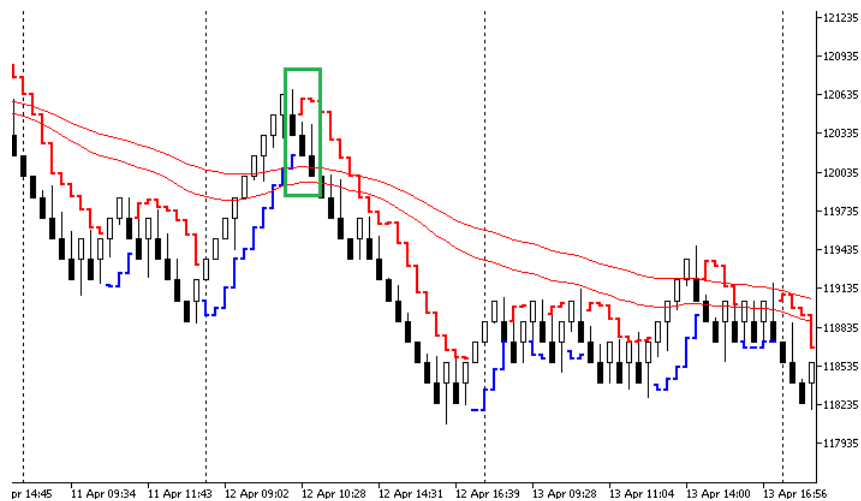
figura 22. Já o sinal de venda é caracterizado pela troca de tendência de alta para baixa e no momento seguinte for confirmada, conforme ilustrado na figura 23.

Figura 22 – Exemplo de sinal de compra



Fonte: (METAQUOTES, 2021a)

Figura 23 – Exemplo de sinal de venda



Fonte: (METAQUOTES, 2021a)

Os dados de compra e os dados de venda serão armazenados em arquivos de texto separados. Foram criados dois tipos diferentes (chamados nesse trabalho como “modo 1” e “modo 2”) de coleta e armazenamento de dados nos arquivos de texto, usando os indicadores e dados descritos anteriormente. Para cada modo de coleta e armazenamento, foram gerados um arquivo de texto com somente dados de compra e um com somente os dados de venda. Como são dois modos de armazenamento de dados, no final da etapa de coleta de dados, gerou quatro arquivos de texto com os dados de entrada para alimentar a rede.

Para o “modo 1” foram coletados e armazenados dados referentes ao indicador Esto-

cástico, ao indicador ADX, aos preços de abertura, fechamento, preço máximo e preço mínimo, a multiplicação do volume negociado com o sinal indicando se o preço de fechamento foi superior ao de abertura, informações do indicador HiLo, as médias móveis exponenciais de 21, 44 e 66 períodos totalizando uma entrada de 84 dados.

Para o “modo 2” também foram coletados e armazenados dados referentes ao indicador Estocástico, ao indicador ADX, aos preços de abertura, fechamento, preço máximo e preço mínimo. A diferença do segundo modo é que o sinal indicando se o preço de fechamento foi superior ao de abertura agora é multiplicado pela amplitude e as informações referentes ao volume de negociação são armazenadas sem essa multiplicação. E por fim são armazenadas as informações do indicador HiLo, as médias móveis exponenciais de 21, 44 e 66 períodos totalizando uma entrada de 89 dados.

Para cada topologia de rede proposta, será usada esses dois modos de captura e armazenamento de dados nos arquivos de texto, que serão usados como dados de entrada para treinamento das redes. O uso desses modos tem como objetivo usar diferentes dados de entrada e treinamento nas redes neurais e analisar os resultados obtidos.

3.4.3 Uso de DLL para coleta de dados

DLL é uma sigla para *Dynamic Link Library*, se trata de uma biblioteca dinâmica, desenvolvido pela Microsoft sendo arquivos de sistemas essenciais para o Windows. Neles pode conter dados, código ou recursos que podem ser acessados por um ou mais programas simultaneamente. Eles são compostas de sub-rotinas armazenadas em disco, podendo ser carregada na memória quando uma aplicação usa seus recursos. Uma aplicação pode fazer o uso de vários desses arquivos (PEREIRA, 2013).

Uma das limitações mencionada por Silva (2021) foi que a plataforma Metatrader não permite que o algoritmo realize negociações HTTP durante o modo de testes em bases históricas, permitindo somente durante negociações em tempo real. Com isso, a simulação das negociações utilizando a rede precisou ser realizada durante o período que houvesse negociações.

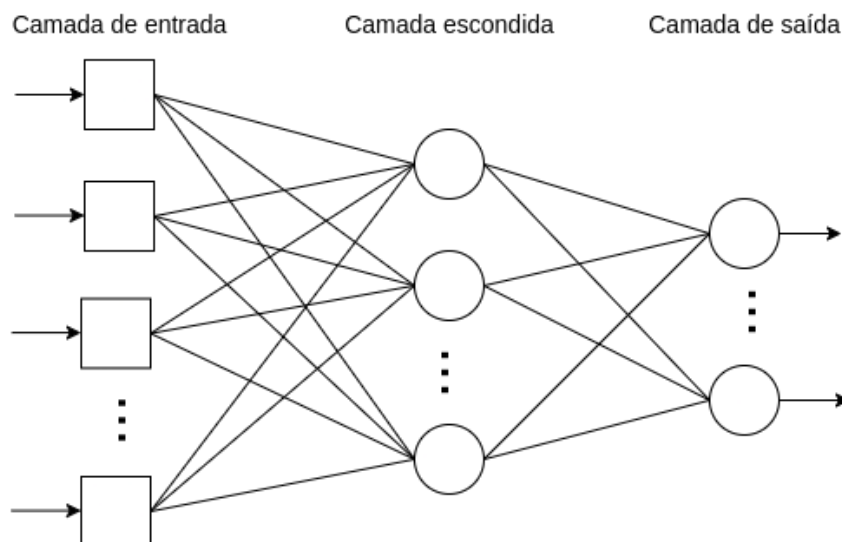
Para contornar o problema descrito acima, foi utilizado a DLL *Wininet.dll* provida em sistema operacional Windows. Essa biblioteca irá permitir que o algoritmo faça requisições HTTP durante o modo de testes, utilizando as funções dessa biblioteca e usando os recursos do sistema operacional (METAQUOTES, 2014).

3.5 Modelo da RNA

Dos modelos existentes de redes neurais artificiais, foi escolhido o *Multilayer perceptron* com o algoritmo de *backpropagation* para o treinamento da rede. A topologia da rede será de uma camada entrada, com as entradas sendo os valores dos indicadores e os demais dados descritos na seção anterior, uma ou mais camadas escondidas e uma camada de saída. A quantidade de neurônios em cada camada irá variar conforme o uso e a topologia da rede que

será utilizada, porém, a quantidade de neurônios de saída sempre serão oito. A figura 24 a seguir representa a topologia do modelo de MLP proposta com todos os neurônios conectados.

Figura 24 – Diagrama de topologia da MLP



Fonte: Elaborado pelo autor

As amostras foram classificadas automaticamente conforme o ganho financeiro da operação. Utilizou-se oito classes segundo a quantidade de pontos ganhos no ativo mini índice, resultando em um modelo MLP com 8 neurônios na camada de saída. As oito classes estão organizadas por pontos conforme descrito na tabela 2 abaixo.

Tabela 2 – Classificação das amostras por intervalo de pontos

Classe	Intervalo de pontos
1	Até 50
2	Entre 51 e 100
3	Entre 101 e 150
4	Entre 151 e 200
5	Entre 201 e 250
6	Entre 251 e 300
7	Entre 301 e 350
8	Maior que 351

Fonte: Elaborado pelo autor

Uma posição de compra e venda usando a rede neural será somente aberta caso a RNA classifique os dados como classe 4, 5, 6, 7 ou 8, ou seja, caso a RNA efetue uma previsão que o ganho será de pelo menos 151 pontos. Assim, o robô só irá abrir uma posição de venda ou compra caso a tendência do indicador *HiLo* e se o maior valor de saída do neurônio seja maior que 150 pontos.

Com a intenção de obter um melhor aprendizado, para cada topologia de RNA, serão treinadas duas redes onde uma será especializada em compra e outra em venda. Para cada

topologia de rede, há duas formas de dados de entrada usando os indicadores e demais dados descritos anteriormente.

3.5.1 Topologia da RNA 1

A primeira topologia de rede neural é construída por um MLP contendo uma camada de entrada de dados, uma camada escondida e a camada de saída conforme a tabela 3 abaixo. A função de otimização foi a Adagrad, função de cálculo de erro é a função Erro Médio Quadrático (MSE) e por fim as métricas a serem avaliadas pelo modelo durante o treinamento e teste são a MSE e a precisão.

Tabela 3 – Topologia da RNA 1

Camada	Unid. modo 1	Unid. modo 2	Função de ativação	Usa bias
Entrada	84	89	Não possui	Não
Escondida	46	48	Softmax	Sim
Saída	8	8	Tangente Hiperbólica	Sim

Fonte: Elaborado pelo autor

3.5.2 Topologia da RNA 2

A segunda topologia de rede neural é construída por um MLP contendo uma camada de entrada de dados, duas camadas escondidas e a camada de saída conforme a tabela 4 abaixo. A função de otimização foi a Adagrad, função de cálculo de erro é a função Erro Médio Quadrático (MSE) e por fim as métricas a serem avaliadas pelo modelo durante o treinamento e teste são a MSE e a precisão.

Tabela 4 – Topologia da RNA 2

Camada	Unid. modo 1	Unid. modo 2	Função de ativação	Usa bias
Entrada	84	89	Não possui	Não
Escondida 1	46	48	Tangente Hiperbólica	Sim
Escondida 2	27	28	Tangente Hiperbólica	Sim
Saída	8	8	Sigmoid	Sim

Fonte: Elaborado pelo autor

3.5.3 Topologia da RNA 3

A terceira topologia de rede neural é construída por um MLP contendo uma camada de entrada de dados, duas camadas escondidas e a camada de saída conforme a tabela 5 abaixo. A função de otimização foi a Gradiente Descendente, função de cálculo de erro é a função Erro Médio Quadrático (MSE) e por fim as métricas a serem avaliadas pelo modelo durante o treinamento e teste são a MSE e a precisão.

Tabela 5 – Topologia da RNA 3

Camada	Unid. modo 1	Unid. modo 2	Função de ativação	Usa bias
Entrada	84	89	Não possui	Não
Escondida 1	46	48	ReLu	Sim
Escondida 2	27	28	ReLu	Não
Saída	8	8	Sigmoid	Não

Fonte: Elaborado pelo autor

3.5.4 Topologia da RNA 4

A quarta topologia de rede neural é construída por um MLP contendo uma camada de entrada de dados, três camadas escondidas e a camada de saída conforme a tabela 6 abaixo. A função de otimização foi a Gradiente Descendente, função de cálculo de erro é a função Erro Médio Quadrático (MSE) e por fim as métricas a serem avaliadas pelo modelo durante o treinamento e teste são a MSE e a precisão.

Tabela 6 – Topologia da RNA 4

Camada	Unid. modo 1	Unid. modo 2	Função de ativação	Usa bias
Entrada	84	89	Não possui	Não
Escondida 1	46	48	Tangente Hiperbólica	Sim
Escondida 2	27	28	ReLu	Sim
Escondida 3	17	18	Tangente Hiperbólica	Não
Saída	8	8	Sigmoid	Não

Fonte: Elaborado pelo autor

3.5.5 Topologia da RNA 5

A quinta topologia de rede neural é construída por um MLP contendo uma camada de entrada de dados, três camadas escondidas e a camada de saída conforme a tabela 7 abaixo. A função de otimização foi a Adagrad, função de cálculo de erro é a função Erro Médio Quadrático (MSE) e por fim as métricas a serem avaliadas pelo modelo durante o treinamento e teste são a MSE e a precisão.

Tabela 7 – Topologia da RNA 5

Camada	Unid. modo 1	Unid. modo 2	Função de ativação	Usa bias
Entrada	84	89	Não possui	Não
Escondida 1	46	48	Sigmoid	Sim
Escondida 2	27	28	Tangente Hiperbólica	Sim
Escondida 3	17	18	ReLu	Sim
Saída	8	8	Softmax	Sim

Fonte: Elaborado pelo autor

3.6 Treinamento e validação da previsão do modelo

Com a base de dados de entrada devidamente balanceada, os dados de entrada foram separadas em dados de treino e dados de teste, na razão de 70% dos dados para treino e 30% para teste. Os modelos foram treinados usando a base de treino e os valores dos pesos serão armazenados em um novo arquivo gerado pelo próprio TensorFlow, no qual armazenará somente os melhores pesos obtidos. O treinamento irá ser realizada para cada topologia descrita na seção 3.5 e para cada modo de entrada de dados.

Para fazer a validação da previsão do modelo, foi criado um algoritmo em linguagem Python, no qual irá ler a base de dados de teste, e irá usar a função *predict* que a ferramenta TensorFlow disponibiliza para gerar a previsão dos dados.

Com o resultado obtido pela fase de validação da predição, foi gerado um percentual de amostras de acertos e erros fazendo o cálculo de erro de classificação das amostras previstas. Caso o percentual do classificação de acertos seja baixo, serão feitos ajustes na rede e refeito o processo de treinamento e classificação dos erros até obter uma classificação satisfatória.

3.7 Construção da API

Com as redes devidamente treinadas, obtendo bons resultados e os arquivos gerados pelo TensorFlow contendo os pesos da rede, será construído uma API Web, utilizando o framework Flask, construído em linguagem de programação Python. Essa API será construída de forma que consiga receber e enviar requisições HTTP e executar as previsões. A previsão se dará pelo uso da biblioteca TensorFlow, importando os pesos salvos de cada modelo já criado e treinado anteriormente e enviar dados para o Metatrader para auxiliar na possível abertura de posição de compra ou venda do ativo.

A escolha do framework Flask se deve pela disponibilidade de integração da ferramenta TensorFlow na construção de uma API e por ser de uso simples, prática e rápida para a construção de aplicações Web. Uma vez que esse framework seja uma abordagem minimalista, suportando várias requisições por segundo mantendo um bom desempenho.

A API terá somente uma interface para processar os dados. Essa interface será responsável por receber os dados vindo do Metatrader, via requisição HTTP do tipo GET passando os parâmetros dos indicadores e demais informações na URL, e então devolver a resposta ao Metatrader com qual neurônio de saída da rede obteve o maior valor de previsão. Essa informação será processada pelo robô de negociação para decidir se irá abrir uma posição no ativo naquele momento ou não.

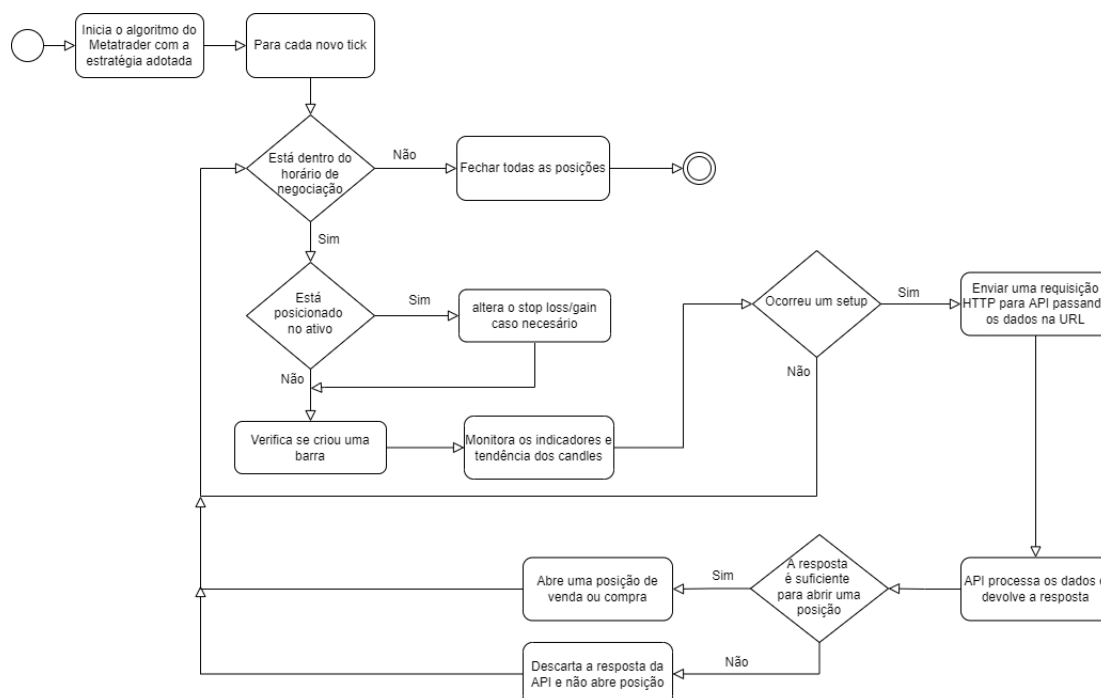
3.8 Usar modelos em conta de demonstração

Com os modelos das redes neurais treinadas, obtendo bons resultados durante a etapa de validação e previsão do modelo, e a construção da API já realizada, o próximo passo é a criação do robô usando o Metatrader para realizar as negociações em tempo real.

Para a construção do robô foi usada a linguagem usada na plataforma Metatrader 5, a MQL5. O algoritmo do robô deve conseguir realizar a leitura dos indicadores e das demais informações já descritas em questão, identificar pontos que podem ser interessantes para uma possível abertura de posição de compra/venda, usando o HiLo como referência, enviar requisições HTTP do tipo GET passando os parâmetros na URL, receber e interpretar a resposta obtida das requisições GET que serão enviadas da API para o Metatrader, definir que irá abrir uma posição no ativo conforme o valor recebido da API, abrir e fechar posições de negociação de acordo técnica de *Day Trade*.

Para uma melhor representação, a figura 25 ilustra o fluxo de informação para a realização das negociações descrito acima.

Figura 25 – Fluxo de etapas para realização das negociações



Fonte: Elaborado pelo autor

3.9 Análise dos resultados

Ao final, será analisado os resultados utilizando as redes neurais treinadas durante o processo de compras e vendas do ativo, verificando será possível obter ganhos financeiros com os modelos propostos.

4 Desenvolvimento

*“Há apenas um bem, o saber;
e apenas um mal, a ignorância”.*
Sócrates

Este capítulo destina-se a descrever de forma mais detalhada as etapas de desenvolvimento do presente trabalho.

4.1 Obtenção dos dados

Para obter os dados que serão usados no treinamento da rede, é necessário adicionar os indicadores descritos na seção 3.3. Alguns dos indicadores escolhidos, a plataforma Meta-trader já disponibiliza na sua biblioteca de indicadores e basta importá-los para o algoritmo do robô. Os indicadores que já se encontram pronto para uso na plataforma são a Média Móvel Simples, Média Móvel Exponencial, ADX e o Estocástico. Porém, outro importante indicador, o indicador HiLo, não se encontra nessa biblioteca, sendo assim foi necessário a codificação desse indicador conforme descrito na seção 2.7.5 usando a função “iCustom” do Metatrader, que possibilita a criação de indicadores técnicos personalizados. A seguir nas tabelas 8, 9, 10 e 11, serão descritos os parâmetros passados para cada indicador para a coleta dos dados.

Tabela 8 – Parâmetros do indicador Média Móvel Exponencial

Parâmetro	Valor
Períodos	21, 44, 66

Fonte: Elaborado pelo autor

Tabela 9 – Parâmetros do indicador ADX

Parâmetro	Valor
Períodos	9

Fonte: Elaborado pelo autor

Tabela 10 – Parâmetros do indicador Estocástico

Parâmetro	Valor
Período para %K	5
Período para %D	3
Valor da desaceleração	3
Tipo de Média	Média Móvel Simples
Método de cálculo	Preços Mínimo/Máximo

Fonte: Elaborado pelo autor

Tabela 11 – Parâmetros do indicador HiLo

Parâmetro	Valor
Período	3
Tipo de Média	Média Móvel Simples

Fonte: Elaborado pelo autor

Após todos os indicadores descritos anteriormente estarem disponíveis para uso, um algoritmo escrito em MQL5 na plataforma Metatrader, foi implementado visando ler os valores dos indicadores quando o indicador HiLo indicasse possíveis momentos de negociação. Além das informações dos indicadores, também foi implementado uma classificação da pontuação de ganho da operação relacionando com os intervalos de pontuação das oito classes do modelo da rede neural, conforme descritos na tabela 2. Para cada classe é adicionado aos dados de entrada o número “0,995” ou “0,005”, seja ela uma operação de compra ou de uma operação de venda. Os valores e intervalos estão descritos na conforme a tabela 12 abaixo. O valor 0,005 representa uma aproximação para o valor 0, que implicaria que o neurônio de saída não foi ativado e o valor 0,995 representa uma aproximação para o valor 1, que implicaria que o neurônio foi ativado.

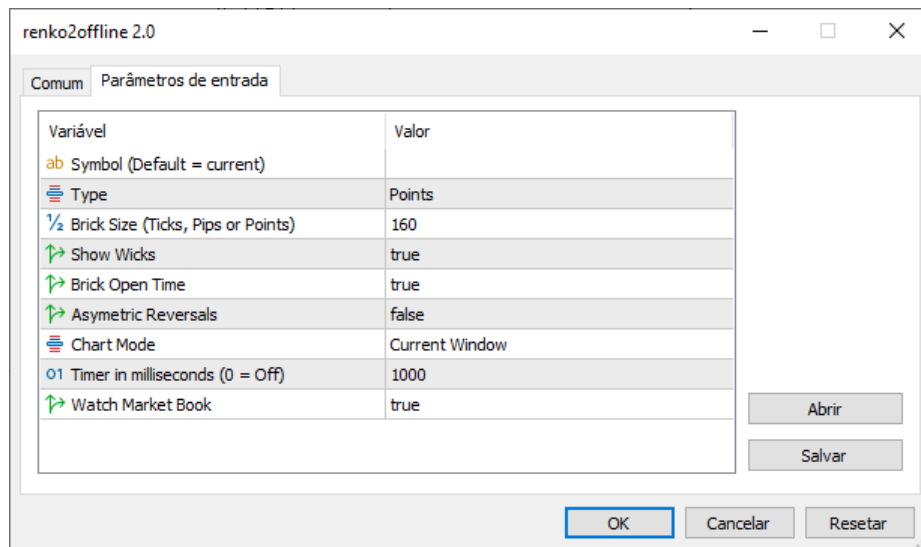
Tabela 12 – Classificação da operação por intervalo de pontos

Intervalo de pontos	Valores
até 50 pontos	0.995, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005
51 a 100 pontos	0.005, 0.995, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005
101 a 150 pontos	0.005, 0.005, 0.995, 0.005, 0.005, 0.005, 0.005, 0.005
151 a 200 pontos	0.005, 0.005, 0.005, 0.995, 0.005, 0.005, 0.005, 0.005
201 a 250 pontos	0.005, 0.005, 0.005, 0.005, 0.995, 0.005, 0.005, 0.005
251 a 300 pontos	0.005, 0.005, 0.005, 0.005, 0.005, 0.995, 0.005, 0.005
301 a 350 pontos	0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.995, 0.005
mais de 351 pontos	0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.995

Fonte: Elaborado pelo autor

Para cada linha adicionada, há somente um valor de “0.995” representando que para aquela operação obteve uma diferença de pontos daquele intervalo de valores. Por exemplo, caso a operação obteve um lucro com até 50 pontos de variação ou qualquer prejuízo, o primeiro valor adicionado aos dados de entrada será “0,995” e os demais sete valores adicionados serão “0,005”. Caso outra operação obteve lucro e tenha uma diferença de 160 pontos, os valores adicionados aos dados de entrada serão os que estão no intervalo de 151 a 200 pontos.

Com todos os indicadores e dados de entrada prontos, o próximo passo é conversão do gráfico temporal para gráfico atemporal *Renko*. Para isso, foi utilizado a biblioteca “Renko 2.0 Offline” descrita na seção 3.4.1. As configurações usadas estão descritas na figura 26 abaixo.

Figura 26 – Configurações usadas para conversão e criação do gráfico *Renko*

Fonte: (METAQUOTES, 2021a)

A conversão usando a biblioteca “Renko 2.0 Offiline” foi primeiramente usada no ativo histórico *WIN\$N* descrito na seção 3.4.1. Para fazer a transformação do gráfico, uma vez que o *WIN\$N* guarda os dados dos *candles* em forma temporal. A figura 27 mostra um trecho do ativo *WIN\$N* para um trecho do período da manhã do dia 13 de abril de 2022, em forma temporal, como os dados dos *candles* na forma em que foram realizadas as negociações.

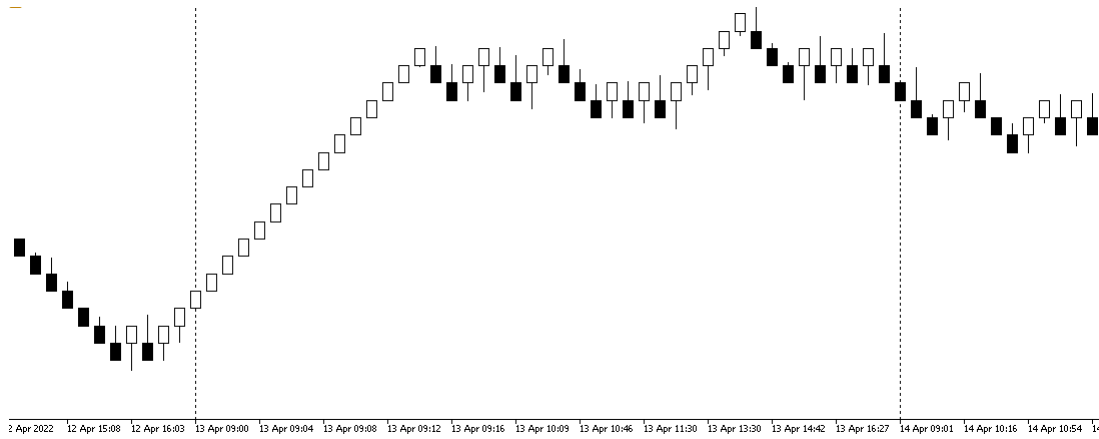
Figura 27 – Trecho do gráfico *WIN\$N* em forma temporal

Fonte: (METAQUOTES, 2021a)

A transformação foi realizada para criar um gráfico contendo as informações, as mesmas informações do gráfico *WIN\$N*, só que agora em forma atemporal usando a técnica de formação de *candle* atemporal do gráfico *Renko*. A figura 28 mostra o mesmo período de dados da figura 27, porém agora já transformada em gráfico atemporal *Renko*. Enquanto a conformação dos *candles* no gráfico temporal, para a visualização da figura 27, a cada 1 minuto, o gráfico atemporal *Renko* se dá pela variação de pontos do gráfico temporal, não mais impor-

tando em que momento o *candle* foi criado. Para a criação desse novo gráfico, foi definida a quantidade de 160 pontos do gráfico temporal para a criação de um novo *candle*.

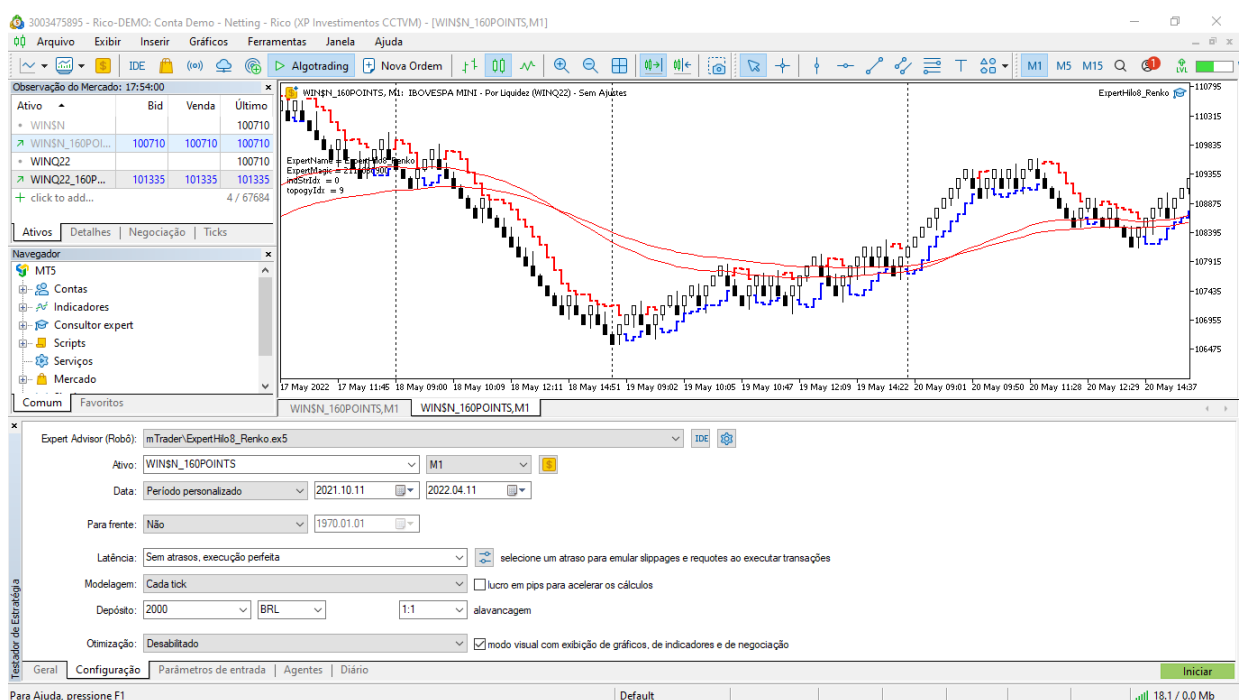
Figura 28 – Trecho do gráfico *WIN\$N* em forma atemporal



Fonte: (METAQUOTES, 2021a)

O gráfico usado para a coleta dos dados será o gráfico *Renko* transformado pela biblioteca “Renko 2.0 Offline”. As figuras 29, 30 e 31 mostram as telas do Metatrader em modo Testador de Estratégia. Na figura 29 podemos visualizar algumas configurações usadas como o robô usado, o ativo histórico criado pela biblioteca e nomeado de *WIN\$N_160POINTS*, o período no qual o robô irá executar o algoritmo, a latência desejada do envio da ordem entre outras configurações.

Figura 29 – Exemplo das principais configurações para coleta de dados



Fonte: Elaborado pelo autor

Na figura 30 é possível visualizar os parâmetros do algoritmo e dos indicadores para realizar a coleta dos dados.

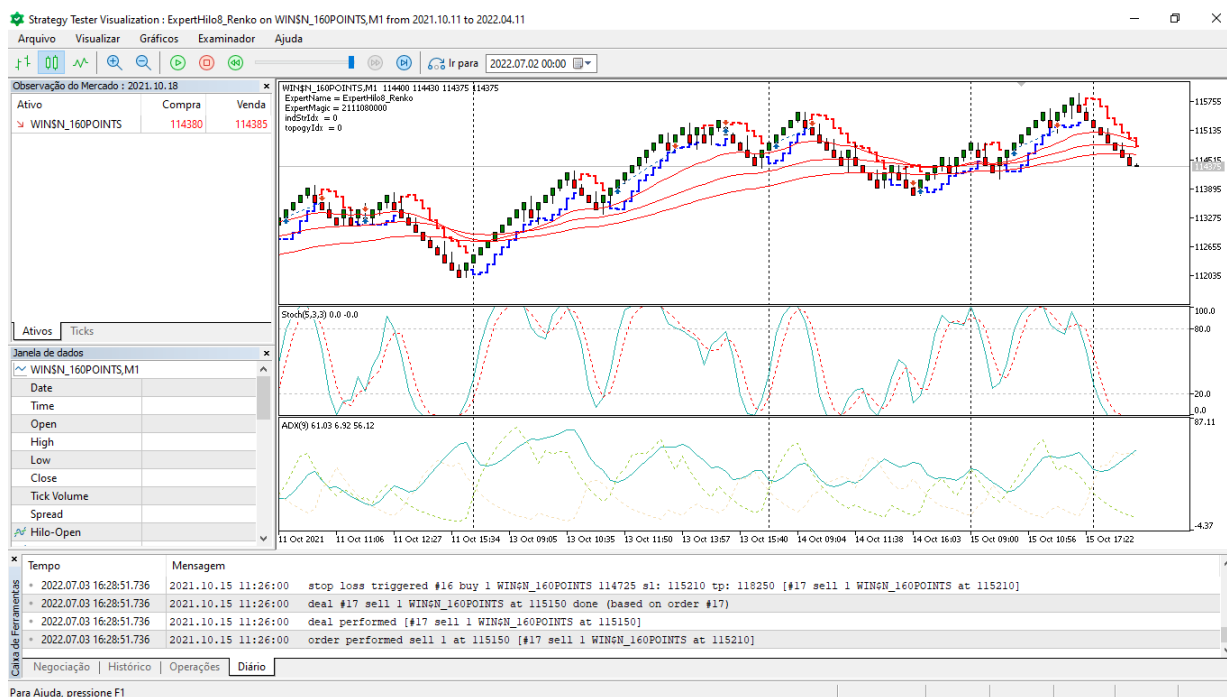
Figura 30 – Exemplo dos parâmetros para coleta de dados

Variável	Valor	Iniciar	Passo	Parar	Passos
☒ Nro de Contratos	1	1	0.1	10.0	
☒ Ativo Renko	WINSN_160POINTS				
☒ Ativo B3	WINSN_160POINTS				
☒ Time Frame	1 Minute	current		1 Month	
☒ Precisão de normalização	5	0.5	0.05	5.0	
==> Outros					
☒ Tipo de média	Exponential	Simple		Linear weighted	
☒ Modo operacao	Modo Gerar dados	Modo operar		Modo Gerar dados	
☒ Habilitar IA	false	false		true	
☒ operarRetIA	3	3	1	30	
☒ Indicadores Modo String	0	0	1	10	
☒ Nro Topologia IA	0	1	1	10	
==> Operacional					
☒ Stop Loss	140	8.5	0.85	85.0	
☒ Stop Gain	1000	50	5.0	500.0	
☒ TrailStop Hilo Habil	true	false		true	
☒ TrailStop Hilo Dist	40	2	0.2	20.0	
☒ TrailStop Pontos Habil	true	false		true	
☒ TrailStop Pontos Dist	140	7	0.7	70.0	
☒ TrailStop Pontos GainStop	50	2.5	0.25	25.0	
☒ segueTendencia	true	false		true	
==> Hilo					
☒ Período Hilo	3	3	1	30	
☒ Modo Hilo	Modo 2	Modo 1		Modo 2	
==> Horário de operação					
☒ Hr e Mn p inicio de ordens (HHMM)	900	900	1	9000	
☒ Hr e Mn limite p ordens (HHMM)	1725	1725	1	17250	
☒ Hr e Mn fechar ordens e posições (HHMM)	1740	1740	1	17400	

Fonte: Elaborado pelo autor

Na figura 31 abaixo mostra o algoritmo de coleta de dados em execução com os indicadores a mostra na tela.

Figura 31 – Exemplo de execução da coleta de dados



Fonte: Elaborado pelo autor

Uma vez que foi construída 2 redes neurais diferentes, uma especializada em compra e outra em venda, os dados coletados foram salvos em 2 arquivos de texto distintos, um para cada tipo de modo de coleta de dados, no caso 2 modos, totalizando 4 arquivos de texto. Conforme o indicador HiLo, os seguintes dados serão salvos em cada linha dos arquivos, para no primeiro momento, os mesmos dados e posteriormente uma pequena alteração. Os dados salvos para o modo 1 são:

1. Os últimos 5 *candles* para o Estocástico rápido
2. Os últimos 5 *candles* para Estocástico lento
3. Os últimos 4 *candles* do ADX principal
4. Os últimos 4 *candles* do ADX com indicador de direção positiva
5. Os últimos 4 *candles* do ADX com indicador de direção negativa
6. Os últimos 5 *candles* com o valor do preço de abertura
7. Os últimos 5 *candles* com o valor do preço de máximo
8. Os últimos 5 *candles* com o valor do preço de mínimo
9. Os últimos 5 *candles* com o valor do preço de fechamento
10. Os últimos 5 *candles* com o valor do volume de negociação vezes o sinal se o preço de fechamento for menor que o de abertura
11. Os últimos 4 *candles* para o HiLo com a média móvel alta
12. Os últimos 4 *candles* para o HiLo com a média móvel baixa
13. Os últimos 11 *candles* para o valor do sinal do HiLo
14. Os valores de 6 *candles* aleatórios para a média móvel de 21 períodos
15. Os valores de 6 *candles* aleatórios para a média móvel de 44 períodos
16. Os valores de 6 *candles* aleatórios para a média móvel de 66 períodos
17. Valor do ano, mês e dia do *candle* no formato AAAAMMDD
18. Valor da hora e minuto do *candle* no formato HHMM
19. Os 8 valores da classificação da operação conforme a tabela 12

Os dados para o segundo se começa coletando os dados da mesma forma do item 1 ao item 9 da lista acima. Porém, a alteração da coleta do modo 2 se dá da seguinte forma:

10. Os últimos 5 *candles* com o valor de amplitude vezes o sinal se o preço de fechamento for menor que o de abertura

11. Os últimos 5 *candles* com o valor do volume de negociação sem nenhuma multiplicação

Os demais dados coletados no segundo modo respeitam os itens 11 a 19 do primeiro modo de coleta de dado, formando assim 5 informações a mais coletadas nesse modo além da mudança de coleta descrita acima.

4.2 Balanceamento e classificação dos dados coletados

Para realizar o treinamento de todas as variações das redes em todos os modos, foram utilizados a mesma massa de dados, coleta com o algoritmo de obtenção de dados descrito na seção anterior 4.1. O intervalo de dados usado para a coleta foi de 01/08/2021 até 20/04/2022, totalizando 258 amostras para a rede de compra e 262 amostras para a rede de venda.

Analizando os dados coletados e realizando a contagem de dados por classe conforme a tabela 2, tanto para amostras dos dados de compra como os dados de venda, foi possível observar que a massa de dados estava muito desbalanceada, tendo poucos dados em cada classe, assim como a massa de dados na totalidade. As amostras coletadas estavam concentradas na primeira classe, que corresponde a classe “até 50 pontos”, com aproximadamente 84% para os dados de compra e 88% para os de venda. O restante das amostras estão distribuídas entre as demais classes.

Com base no cenário descrito, foi necessário fazer o balanceamento da massa de dados antes de realizar o treinamento. Para fazer esse balanceamento, foi implementado um algoritmo onde foi realizada a simples replicação, sem alteração, das amostras em todas as classes até que todas ficassem com a mesma quantidade de dados. Para saber o limite de dados a serem gerados por classe, foi utilizado como referência a quantidade de amostras da classe que continha mais dados, no caso a classe “até 50 pontos”.

Com os dados devidamente balanceamento e com uma amostra de dados maior, foi realizado a separação dessa massa de dados em dados de treino e dados de teste de forma aleatória. Para realizar essa separação foi utilizado uma função da biblioteca “scikit-learn”, conforme o código abaixo.

```
from sklearn.model_selection import train_test_split

treino_x, teste_x, treino_y, teste_y = train_test_split(
    dados_entrada,
    dados_saida,
    test_size=0.3,
    shuffle=True
)
```

O código acima retorna 4 listas de dados separados de forma aleatória na proporção de 70% da quantidade de dados para as listas correspondentes aos dados de treino e 30%

correspondentes aos dados de teste. Os dados foram esbaralhados antes da separação das massas de dados em treino e teste. Essa separação foi realizada para cada treinamento de todas as variações de rede neural nos 2 modos, usando os dados de treino para realizar o treino e os dados de teste para validar a taxa de acerto da rede.

4.3 Implementação das RNAs

Depois coletar os dados de entrada para treinar a rede neural, o próximo passo é a implementação e treinamento das redes neurais, conforme descrito na seção 3.5.1 a 3.5.5. Foram construídas 5 variações de redes neurais em 2 diferentes modos e de 2 formas separadas, sendo uma especializada em compra e outra venda. Ao final de cada treinamento, os melhores pesos foram salvos em disco por uma função do *Tensorflow* e carregar esses dados posteriormente para a API.

Todas as variações de topologias, modos de redes neurais, funções de ativação e funções de otimização foram decididas de forma empírica, buscando uma maior diversificação nas construções da rede e obter bons resultados.

Durante os treinamentos foi utilizado um notebook Acer com processador Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz 2.40GHz, com memória RAM de 8GB, GPU integrada Intel(R) HD Graphics 5500 e sistema operacional Windows 10. Nos momentos de treinamento da rede, a máquina foi exclusivamente utilizada para este fim, executando processos do sistema operacional e do treinamento.

4.3.1 Variação 1 da RNA

A implementação da primeira variação da rede neural seguiu a configuração descrita na seção 3.5.1, sendo que a construção foi realizada em 2 modos de quantidade de neurônios em cada camada. A estrutura foi construída em 3 camadas, sendo uma de entrada, uma camada escondida e a outra camada de saída.

No trecho de código a seguir, segue a descrição de como foi construída a primeira variação de rede neural usando linguagem de programação Python e a biblioteca Tensorflow.

```
from tensorflow.keras import layers, Sequential, optimizers, losses
```

```
modelo = Sequential([
    layers.InputLayer(
        input_shape=(qtd_entradas,)
    ),
    layers.Dense(
        qtd_camada_escondida_1,
        bias_initializer='Ones',
        activation='softmax'
    ),
])
```

```

        layers.Dense(
            qtd_saidas,
            bias_initializer='Ones',
            activation='tanh'
        )
    ])

modelo.compile(
    loss=losses.MSE,
    optimizer=optimizers.Adagrad(learning_rate=0.75),
    metrics=['mean_squared_error', 'accuracy']
)

```

As redes foram treinadas e os resultados foram obtidos conforme a tabela 13 abaixo.

Tabela 13 – Treinamento e resultado variação 1 RNA

Modo	Tipo de rede	Quantidade épocas	Precisão de acerto
1	Compra	2000	86,59 %
1	Venda	2000	95,67 %
2	Compra	2000	95,02 %
2	Venda	2000	92,99 %

Fonte: Elaborado pelo autor

4.3.2 Variação 2 da RNA

A implementação da segunda variação da rede neural seguiu a configuração descrita na seção 3.5.2, sendo que a construção foi realizada em 2 modos de quantidade de neurônios em cada camada. A estrutura foi construída em 4 camadas, sendo uma de entrada, duas camadas escondidas e a camada de saída.

No trecho de código a seguir, segue a descrição de como foi construída a primeira variação de rede neural usando linguagem de programação Python e a biblioteca Tensorflow.

```

from tensorflow.keras import layers, Sequential, optimizers, losses

modelo = Sequential([
    layers.InputLayer(
        input_shape=(qtd_entradas,)
    ),
    layers.Dense(
        qtd_camada_escondida_1,
        bias_initializer='Ones',
        activation='tanh'
    )
])

```

```

    ),
    layers.Dense(
        qtd_camada_escondida_2,
        bias_initializer='Ones',
        activation='tanh'
    ),
    layers.Dense(
        qtd_saidas,
        bias_initializer='Ones',
        activation='sigmoid'
    )
])

modelo.compile(
    loss=losses.MSE,
    optimizer=optimizers.Adagrad(learning_rate=0.75),
    metrics=['mean_squared_error', 'accuracy']
)

```

As redes foram treinadas e os resultados foram obtidos conforme a tabela 14 abaixo.

Tabela 14 – Treinamento e resultado variação 2 RNA

Modo	Tipo de rede	Quantidade épocas	Precisão de acerto
1	Compra	2000	98,08 %
1	Venda	2000	98,70 %
2	Compra	2000	98,66 %
2	Venda	2000	98,35 %

Fonte: Elaborado pelo autor

4.3.3 Variação 3 da RNA

A implementação da terceira variação da rede neural seguiu a configuração descrita na seção 3.5.3, sendo que a construção foi realizada em 2 modos de quantidade de neurônios em cada camada. A estrutura foi construída em 4 camadas, sendo uma de entrada, duas camadas escondidas e a camada de saída.

No trecho de código a seguir, segue a descrição de como foi construída a primeira variação de rede neural usando linguagem de programação Python e a biblioteca Tensorflow.

```

from tensorflow.keras import layers, Sequential, optimizers, losses

modelo = Sequential([
    layers.InputLayer(

```

```

        input_shape=(qtd_entradas,)
    ),
    layers.Dense(
        qtd_camada_escondida_1,
        bias_initializer='Ones',
        activation='relu'
    ),
    layers.Dense(
        qtd_camada_escondida_2,
        use_bias=False,
        activation='relu'
    ),
    layers.Dense(
        qtd_saidas,
        use_bias=False,
        activation='sigmoid'
    )
])

modelo.compile(
    loss=losses.MSE,
    optimizer=optimizers.SGD(learning_rate=0.5),
    metrics=['mean_squared_error', 'accuracy']
)

```

As redes foram treinadas e os resultados foram obtidos conforme a tabela 15 abaixo.

Tabela 15 – Treinamento e resultado variação 3 RNA

Modo	Tipo de rede	Quantidade épocas	Precisão de acerto
1	Compra	2000	95,78%
1	Venda	2000	97,94%
2	Compra	2000	95,59%
2	Venda	2000	94,84%

Fonte: Elaborado pelo autor

4.3.4 Variação 4 da RNA

A implementação da quarta variação da rede neural seguiu a configuração descrita na seção 3.5.4, sendo que a construção foi realizada em 2 modos de quantidade de neurônios em cada camada. A estrutura foi construída em 5 camadas, sendo uma de entrada, três camadas escondidas e a camada de saída.

No trecho de código a seguir, segue a descrição de como foi construída a primeira variação de rede neural usando linguagem de programação Python e a biblioteca Tensorflow.

```
from tensorflow.keras import layers, Sequential, optimizers, losses

modelo = Sequential([
    layers.InputLayer(
        input_shape=(qtd_entradas,)
    ),
    layers.Dense(
        qtd_camada_escondida_1,
        bias_initializer='Ones',
        activation='tanh'
    ),
    layers.Dense(
        qtd_camada_escondida_2,
        bias_initializer='Ones',
        activation='relu'
    ),
    layers.Dense(
        qtd_camada_escondida_3,
        use_bias=False,
        activation='tanh'
    ),
    layers.Dense(
        qtd_saidas,
        use_bias=False,
        activation='sigmoid'
    )
])

modelo.compile(
    loss=losses.MSE,
    optimizer=optimizers.SGD(learning_rate=0.5),
    metrics=['mean_squared_error', 'accuracy']
)
```

As redes foram treinadas e os resultados foram obtidos conforme a tabela 16 abaixo.

Tabela 16 – Treinamento e resultado variação 4 RNA

Modo	Tipo de rede	Quantidade épocas	Precisão de acerto
1	Compra	2000	98,28%
1	Venda	2000	98,14%
2	Compra	2000	97,70%
2	Venda	2000	97,73%

Fonte: Elaborado pelo autor

4.3.5 Variação 5 da RNA

A implementação da quinta variação da rede neural seguiu a configuração descrita na seção 3.5.5, sendo que a construção foi realizada em 2 modos de quantidade de neurônios em cada camada. A estrutura foi construída em 5 camadas, sendo uma de entrada, três camadas escondidas e a camada de saída.

No trecho de código a seguir, segue a descrição de como foi construída a primeira variação de rede neural usando linguagem de programação Python e a biblioteca Tensorflow.

```

from tensorflow.keras import layers, Sequential, optimizers, losses

modelo = Sequential([
    layers.InputLayer(
        input_shape=(qtd_entradas,)
    ),
    layers.Dense(
        qtd_camada_escondida_1,
        bias_initializer='Ones',
        activation='sigmoid'
    ),
    layers.Dense(
        qtd_camada_escondida_2,
        bias_initializer='Ones',
        activation='tanh'
    ),
    layers.Dense(
        qtd_camada_escondida_3,
        bias_initializer='Ones',
        activation='relu'
    ),
    layers.Dense(
        qtd_saidas,
        bias_initializer='Ones',
        activation='softmax'
    )
])

```

```

    )
])

modelo.compile(
    loss=losses.MSE,
    optimizer=optimizers.Adagrad(learning_rate=0.75),
    metrics=['mean_squared_error', 'accuracy']
)

```

As redes foram treinadas e os resultados foram obtidos conforme a tabela 17 abaixo.

Tabela 17 – Treinamento e resultado variação 5 RNA

Modo	Tipo de rede	Quantidade épocas	Precisão de acerto
1	Compra	2000	99,04%
1	Venda	2000	98,55%
2	Compra	2000	97,89%
2	Venda	2000	92,57%

Fonte: Elaborado pelo autor

4.4 Construção da API

Para fazer a comunicação entre o algoritmo implementado na plataforma Metatrader e as RNAs já criadas, foi necessário a construção de uma API REST, desenvolvida na linguagem de programação Python, executada na máquina local simultaneamente com o Metatrader. Sendo assim, as requisições que exigiam banda de rede seriam as informações vinda do ativo e as abertura e fechamento de posições no ativo.

A implementação da API foi feita usando o framework Flask, usando o gerenciador de pacotes e bibliotecas Anaconda e usando o framework Tensorflow. Após instalados as ferramentas e bibliotecas necessárias e criado o ambiente para a codificação, foi criada uma aplicação com três camadas que recebe requisições HTTP do tipo GET especificado na porta 8080.

A primeira camada da aplicação seria a *Controller*, que recebe as requisições HTTP do tipo GET vindo do Metatrader e passa os parâmetros vindo dessa requisição para a segunda camada. A segunda camada é a *Service* que contém as regras de negócio associadas as redes neurais e faz a comunicação com a terceira camada, que é a camada *Model*, onde está descrito as topologias, importação dos pesos já obtidos da fase de treinamento.

A camada *Service* faz algumas validações de qual RNA deve ser executada, passa os parâmetros para a camada *Model*, que faz os processamento e predição dos dados usando a função *predict* da biblioteca Tensorflow. O resultado dessa função é tratado pela camada *Service* e posteriormente enviado para a camada de *Controller* e o por fim o retorno da requisição para o Metatrader.

A interface que retorna o valor da predição dos dados de entrada da rede recebe como parâmetros (*query params*) número da topologia da rede, se a venda da rede é compra (+1) ou venda (-1), o modo da rede (1 ou 2) e por fim os valores dos indicadores na seção 4.1. A seguir é ilustrado um exemplo de requisição de usando o método *CURL*, usando uma rede com 84 parâmetros de entrada dos indicadores.

```
curl --location --request GET 'http://localhost/expertia/experthilo7_renko?
topogyIdx=1&positiontype=-1&indStrIdx=1&indicadores=84;0.6786;0.8516;0.9432;
0.7953;0.4403;0.8245;0.8634;0.7263;0.4705;0.2054;0.3271;0.3557;0.3913;0.396;
0.3365;0.4206;0.5258;0.4073;0.2183;0.2729;0.2424;0.3030;160.00;160.00;0.00;
-160.00;-320.00;405.00;320.00;160.00;0.00;-160.00;0.00;160.00;0.00;-160.00;
-615.00;0.00;320.00;160.00;0.00;-160.00;-34031.20;0.00;0.00;0.00;39297.40;
295.0000;160.0000;0.0000;-86.6667;53.3333;0.0000;-258.3333;-418.3333;-1;-1;
1;1;1;-1;-1;-1;-1;-1;-1;174.4655;191.9120;200.2135;224.6584;459.4984;
563.2107;373.8435;391.2315;413.5878;445.1678;589.9471;662.5530;502.7335;
518.2022;540.7361;569.4495;678.9931;739.3138;20210503;915'
--header 'X-resp-format:json'
```

A resposta que essa requisição retorna em formato JSON com o número de neurônio da rede neural que foi ativado, conforme ilustrado abaixo.

```
{
  'stOut': '3'
}
```

4.5 Negociações usando as RNAs

Com a construção da API REST, as redes neurais implementadas e treinadas e o algoritmo na plataforma Metatrader concluídos, conforme descrito nas seções anteriores, o próximo passo é a realização das negociações. Uma conta em uma corretora de valores foi criada e utilizada em modo DEMO (conta em modo de demonstração), que oferecesse suporte a plataforma Metatrader e pudesse efetuar negociações em tempo real.

Para realizar as negociações foi utilizado a plataforma Metatrader 5 com a conta DEMO da corretora de valores Rico. Para realizar as negociações usando a IA, foi utilizado a conta DEMO, e para as realizar as negociações sem a IA, para obter parâmetros comparativos, foi utilizado a ferramenta Testador de Estratégia da plataforma Metatrader, permitindo assim obter os resultados das negociações em poucos segundos.

O tempo de execução de todas as negociações foi iniciado na semana do dia 30/05/2022 e encerrando no dia 15/06/2022, totalizando 13 dias úteis de negociações, durante o horário de 9h00 às 17h40. O ativo de mini índice Bovespa utilizado foi definido conforme a tabela 1,

uma vez que o código do mini índice a ser utilizado depende do mês e o ano em que ele irá vencer. Outro critério foi a escolha pelo ativo que contém a maior liquidez nas negociações, que seria o que estivesse com a data de vencimento mais próximo em relação ao período definido para as negociações. Conforme os critérios descritos, o código do ativo utilizado para as negociações foi o *WINM22*. No período descrito para realizar as negociações, o ativo obteve o seguinte comportamento de negociações descrito na figura 32 abaixo, com um tempo de formação de *candle* de 30 minutos.

Figura 32 – Valores das cotações temporais do mini índice no período de 30/05/2022 a 15/06/2022



Fonte: (METAQUOTES, 2021a)

Com o código do ativo definido, foi criado um gráfico com a versão atemporal (gráfico Renko) da versão do ativo, com o critério de formação de um novo *candle* de 160 pontos de variação do ativo *WINM22*. Para a criação do gráfico *Renko* foi utilizado novamente a biblioteca “Renko 2.0 Offline”, conforme a figura 26 descrito na seção 4.1. O gráfico atemporal gerado pode ser visualizado na figura 33 abaixo.

Figura 33 – Valores das cotações do mini índice em forma de gráfico *Renko* no período de 30/05/2022 a 15/06/2022



Fonte: (METAQUOTES, 2021a)

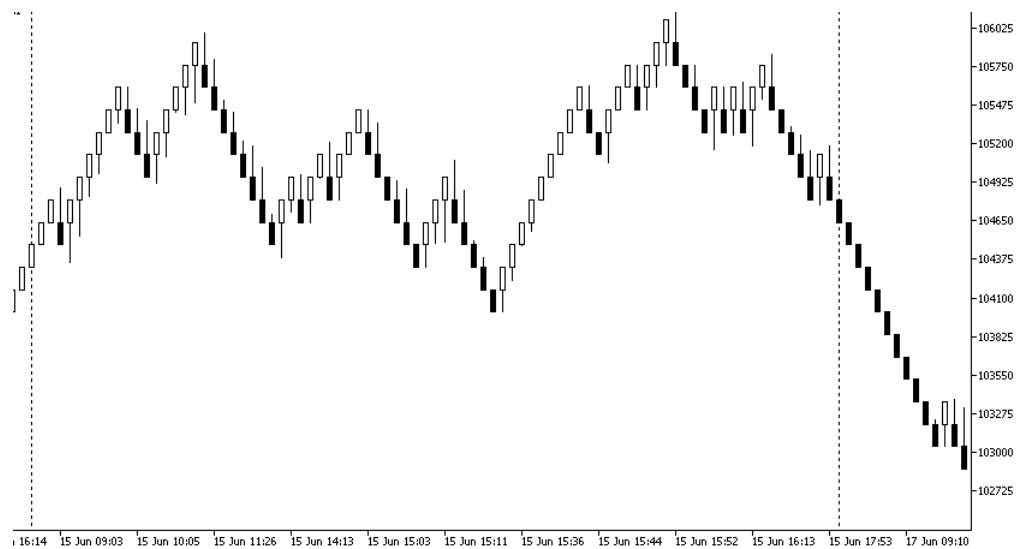
Com a criação desse gráfico atemporal também foi criado um ativo fictício para representar o gráfico, nomeado como *WINM22_160POINTS*. O gráfico foi criado para que o algoritmo escrito no Metatrader pudesse ler os dados dos *candles* que estão sendo formados em tempo de execução, agora de acordo com a variação de pontos no ativo *WINM22* e não mais com o tempo decorrido das negociações. Porém, esse ativo fictício não existe na bolsa de valores, então o ativo utilizado para abrir e fechar as posições foi o *WINM22*. Assim, o algoritmo criado no Metatrader irá ler os *candles* no gráfico atemporal *Renko* e irá realizar negociações no gráfico temporal *WINM22*, uma vez que somente o ativo temporal existe e é possível realizar negociações.

A figura 34 a seguir mostra o gráfico de *candles* temporal de um trecho do dia 15 de junho de 2022, o último dia das negociações. Usando a biblioteca “Renko 2.0 Offiline” o mesmo dia em forma de gráfico *Renko* é ilustrado na figura 35, agora podendo ser ilustrado o dia inteiro em questão não mais como um trecho.

Figura 34 – Gráfico temporal de um trecho dia 15 junho de 2022



Fonte: (METAQUOTES, 2021a)

Figura 35 – Gráfico *Renko* do dia 15 junho de 2022

Fonte: (METAQUOTES, 2021a)

Com o gráfico *Renko* devidamente criado, foi importado o algoritmo para esse gráfico com as seguintes configurações ilustradas na figura 36 abaixo. Conforme descrito, o ativo *WINM22_160POINTS* não existe na bolsa de valores e por isso foi usado o *WINM22* para negociações. Os valores das variáveis “Indicadores Modo String” e “Nro Topologia IA” que tem os valores alterados para os valores de forma a utilizar todas as topologias e modos descritos na seção 3.5. Os demais valores de variáveis se mantiveram os mesmos para todas os modelos de RNAs.

Figura 36 – Parâmetros de configuração para operar as negociações

Variável	Valor
==> Geral	
Nro de Contratos	1
Ativo Renko	WINM22_160POINTS
Ativo B3	WINM22
Time Frame	1 Minute
Precisão de normalização	5
==> Outros	
Tipo de média	Exponential
Modo operacao	Modo operar
Habilitar IA	true
operarRetIA	3
Indicadores Modo String	2
Nro Topologia IA	1
==> Operacional	
Stop Loss	140
Stop Gain	1000
TrailStop Hilo Habil	true
TrailStop Hilo Dist	40
TrailStop Pontos Habil	true
TrailStop Pontos Dist	140
TrailStop Pontos GainStop	50
segueTendencia	true
==> Hilo	
Período Hilo	3
Modo Hilo	Modo 2
==> Horário de operação	
Hr e Mn p inicio de ordens (HHMM)	900
Hr e Mn limite p ordens (HHMM)	1725
Hr e Mn fechar ordens e posições (HHMM)	1740

Fonte: (METAQUOTES, 2021a)

Os valores dos parâmetros escolhidos para realizar as negociações foram 140 pontos para o *stop loss*, 1000 pontos para o *stop gain*, 40 pontos de distância para o *trailing stop*, habilitado a opção de seguir a tendência do ativo e 1 mini contrato de *WINM22*.

Para utilizar o algoritmo do Metatrader para realizar negociações, é precisa habilitar a opção “Algotrading”, que permite a execução de operações de compra e venda nos mercados financeiros usando robôs de negociação. Na plataforma, os robôs também são referidos como Experts. Para executar vários robôs simultaneamente abrindo e fechando posições no mesmo ativo independentemente, foi necessário mudar o sistema da de contabilidade de posição conta da correta para “Hedging”, conforme descrito os conceitos na seção 2.10.1.

Após deixar os algoritmos fazendo as negociações, foram obtidos descritos nas seções a seguir. Para primeiro modelo de RNA foram obtidos os dados para os dois modos de entrada de dados. Para obter os resultados, foram coletados os dados gerados do histórico de negociações da plataforma Metatrader.

4.5.1 Negociação sem o uso de IA

Sem fazer o uso da consulta da API onde foi construído alguns modelos de rede neural, um dos casos de teste é verificar os lucros obtidos sem usar uma inteligência artificial e usar como comparador para os algoritmos com o uso das redes neurais.

Após executar o algoritmo de negociação no mesmo período que os algoritmos com o uso de rede neural foram realizados, os dados de lucro e negociações foram coletados.

Para esse modo, sem uso de IA, foi obtido um lucro de R\$ 58,00 contendo um total de 35 negociações, sendo elas 30 negociações com abertura de posição de venda e 5 negociações com abertura de posição de compra. Das 30 negociações de venda, 10 tiveram lucro e 20 foram com prejuízo. E das 5 negociações de compra, 3 delas foram com lucro e 2 delas foram com prejuízo.

Com isso, podemos notar que os ganhos nas operações que obtiveram lucro, tiveram valores nominais maiores que os negociações com prejuízo, resultando em um valor de lucro positivo no final do período da data de teste.

4.5.2 Negociação com o modelo 1 da RNA

Para o primeiro modo da primeira RNA foi obtido um lucro de R\$ 395,00 contendo um total de 18 negociações, sendo 17 negociações emitida pela RNA especializada em venda e 1 negociação de compra emitida pela RNA de compra. Dessas 17 negociações, 8 delas tiveram lucro e 9 foram com prejuízo. A única negociação emitida pela rede de compra obteve lucro.

Para o segundo modo da primeira RNA obteve um lucro de R\$ 460,00 com um total de 25 negociações, sendo 22 emitidas pela RNA especializada em venda e 3 negociações emitidas pela RNA especializada em compra. Das 22 negociações de venda, 13 delas obtiveram lucro e 9 delas obtiveram prejuízo. Das 3 negociações de compra, 2 delas obtiveram lucro e uma obteve prejuízo.

O modelo obteve lucro nos dois modos com valores nominais de lucro maiores que os valores nominais de prejuízo, gerando resultado positivo na totalidade para a rede.

4.5.3 Negociação com o modelo 2 da RNA

Para o primeiro modo da segunda RNA foi obtido um prejuízo de -R\$ 16,00 contendo um total de 7 negociações, sendo 6 negociações emitida pela RNA especializada em venda e 1 negociação de compra emitida pela RNA de compra. Dessas 6 negociações, 4 delas tiveram lucro e 2 foram com prejuízo. A única negociação emitida pela rede de compra obteve prejuízo.

Para o segundo modo da segunda RNA foi obtido um lucro de R\$ 45,00 com um total de 14 negociações, sendo 11 emitidas pela RNA especializada em venda e 3 negociações emitidas pela RNA especializada em compra. Das 11 negociações de venda, 8 delas obtiveram lucro e 3 delas obtiveram prejuízo. Das 3 negociações de compra, 2 delas obtiveram lucro e uma obteve prejuízo.

4.5.4 Negociação com o modelo 3 da RNA

Para o primeiro modo da terceira RNA foi obtido um lucro de R\$ 331,00 contendo um total de 25 negociações, sendo 23 negociações emitida pela RNA especializada em venda e 2 negociações de compra emitida pela RNA de compra. Dessas 23 negociações, 10 delas

tiveram lucro e 13 foram com prejuízo. Das 2 negociações emitidas pela rede de compra, uma obteve prejuízo e a outra lucro.

Para o segundo modo da terceira RNA foi obtido um lucro de R\$ 414,00 com um total de 31 negociações, sendo 28 emitidas pela RNA especializada em venda e 3 negociações emitidas pela RNA especializada em compra. Das 28 negociações de venda, 15 delas obtiveram lucro e 13 delas obtiveram prejuízo. Das 3 negociações de compra, somente 1 dela obteve lucro e 2 obtiveram prejuízo.

Assim como o primeiro modelo de RNA, a também obteve lucro nos dois modos de rede, porém agora tendo mais negociações realizadas nos dois modos da rede. A quantidade de negociações com lucro e prejuízo nessa rede são bem próximas, sendo assim, novamente, os valores nominais das negociações com lucros maiores que as negociações com prejuízo.

4.5.5 Negociação com o modelo 4 da RNA

Para o primeiro modo da quarta RNA foi obtido um prejuízo de -R\$ 71,00 contendo um total de 9 negociações, sendo 5 negociações emitida pela RNA especializada em venda e 4 negociações de compra emitida pela RNA de compra. Dessas 5 negociações, 2 delas tiveram lucro e 3 foram com prejuízo. Das 4 negociações emitidas pela rede de compra, duas obtiveram prejuízo e as outras obtiveram lucro.

Para o segundo modo da quarta RNA foi obtido um prejuízo de -R\$ 24,00 com um total de 5 negociações, sendo que todas as ordens emitidas pela RNA foram da rede especializada em compra. De todas as negociações, 2 delas obtiveram lucro e 3 delas obtiveram prejuízo.

De todas os modelos de rede criadas e executadas, essa foi que obteve somente prejuízo nos dois modos, com poucas negociações realizadas. Embora algumas negociações tiveram lucro, não foi com valores suficientes para que a rede obtivesse lucro.

4.5.6 Negociação com o modelo 5 da RNA

Para o primeiro modo da quinta RNA foi obtido um prejuízo de -R\$ 30,00 contendo um total de 2 negociações, todas elas emitidas pela RNA especializada em compra. Dessas negociações realizadas, as 2 obtiveram prejuízo.

Para o segundo modo da quinta RNA foi obtido um lucro de R\$ 346,00 com um total de 18 negociações, sendo 17 emitidas pela RNA especializada em venda e 1 negociações emitidas pela RNA especializada em compra. Das 17 negociações de venda, 7 delas obtiveram lucro e 10 delas obtiveram prejuízo. Da única negociação de compra realizada obteve prejuízo.

4.5.7 Comparativo geral

Esta sessão tem o objetivo de fazer um comparativo geral entre todos os resultados obtidos das negociações com o algoritmo usando redes neurais com as negociações realizadas sem rede neural com o *setup HiLo*, no período definido.

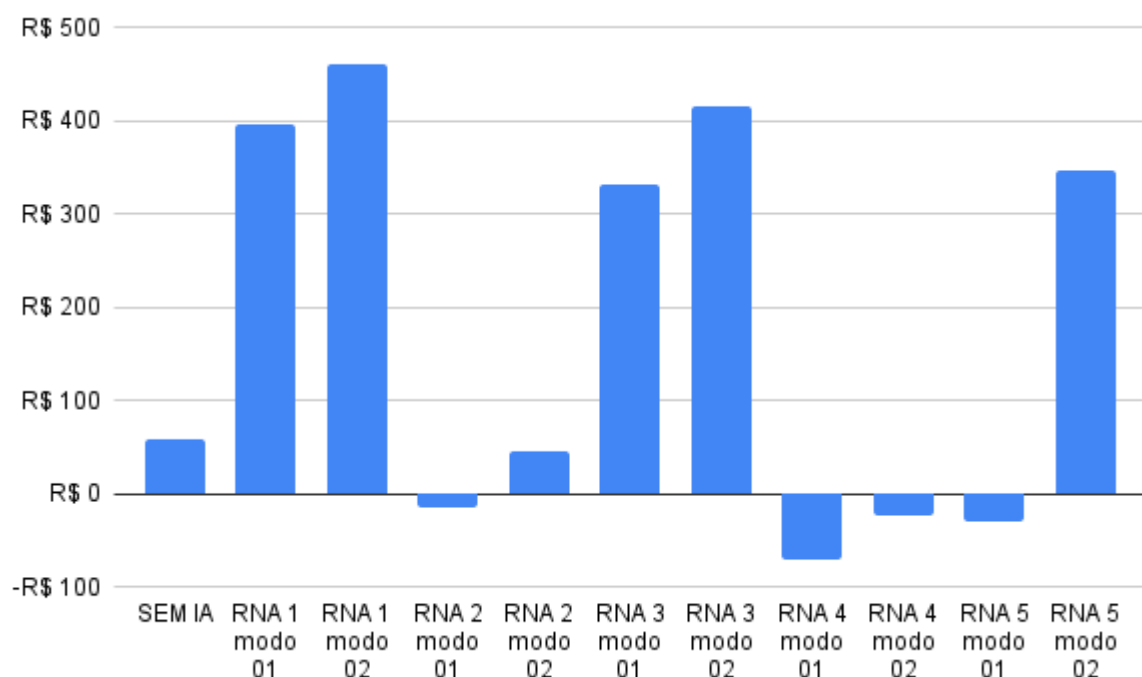
Na tabela 18 e na figura 37 apresentam os resultados os valores monetários obtidos em todos os modelos de negociações e comparando com os resultados com o do *steup HiLo* nomeado como “Sem IA”. Analisando os valores obtidos, podemos notar que os resultados foram diversos, algumas configurações de rede obtiveram somente lucro, outras lucro e prejuízo e por fim somente prejuízo. Para os resultados que obtiveram lucro, a maioria obteve valores, usando IA, nominalmente maiores em comparação ao *setup HiLo* (sem uso de IA). E para os casos que obtiveram prejuízo, os valores foram nominalmente menores em comparação aos valores que obtiveram lucro.

Tabela 18 – Lucros obtidos nas negociações somente das RNAs

Modelos	Modo 1	Modo 2
Modelo 1	R\$ + 395,00	R\$ + 460,00
Modelo 2	R\$ - 16,00	R\$ + 45,00
Modelo 3	R\$ + 331,00	R\$ + 414,00
Modelo 4	R\$ - 71,00	R\$ - 24,00
Modelo 5	R\$ - 30,00	R\$ + 346,00
Sem IA	R\$ + 58,00	

Fonte: Elaborado pelo autor

Figura 37 – Valores monetários obtidos nas negociações



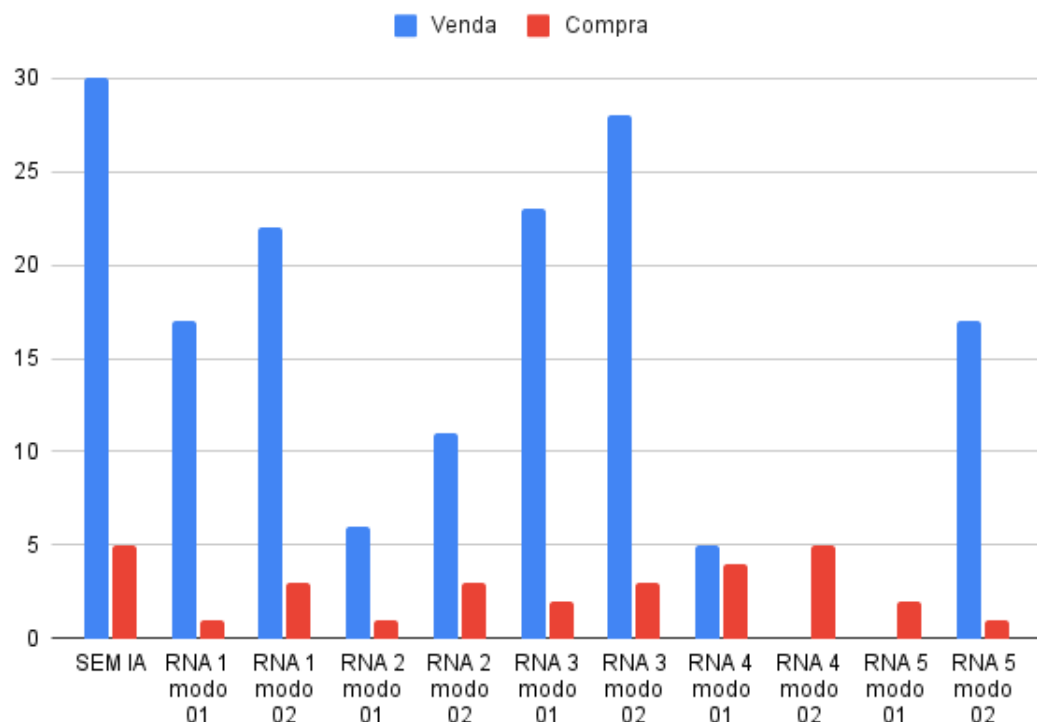
Fonte: Elaborado pelo autor

Analisando somente o modo 01 das RNAs, eles obtiveram uma variação de resultado de lucro e prejuízo maior que o modo 02, onde o segundo modo obteve lucro em todas os modelos, com exceção do modelo 04. Com os resultados monetários obtidos dos lucros e prejuízos, é possível observar que o uso da RNAs aumentou nominalmente os lucros, quando

obteve lucro, e quando obteve prejuízos, os valores não foram da mesma magnitude que os lucros, sendo assim prejuízos monetários menores para os modelos e período executados.

Observando agora a quantidade de negociações realizadas em cada modelo de RNA e com o *setup* ilustrada na figura 38, é possível notar a diferença entre a quantidade entre negociações realizadas, sendo que usou o *setup* realizou o maior volume de operações quanto para compra quanto para venda (no caso da venda, realizou a mesma quantidade que a RNA 4 modo 2). Associando a quantidade de negociações realizadas com a figura 37, uma menor quantidade de negociações, no caso modos modelos RNA 1, RNA 3 (nos dois modos) e RNA 5 modo 02, obtiveram lucro maiores que a do *setup*, sendo que o modelo sem IA foi o que mais abriu operações. Com isso, os modelos usando RNAs citados obtiveram, em média, uma maior magnitudes nos lucros das negociações em comparação as negociações realizadas do modelo *setup* (Sem IA).

Figura 38 – Quantidade de negociações de venda e compra



Fonte: Elaborado pelo autor

A tabela 19 e a figura 39 retratam a eficiência dos modelos, com e sem o uso de redes neurais, em obter resultados com lucro e prejuízos para as negociações. A tabela demonstra a quantidade de negociações com lucro e prejuízos emitidas por modelo separadas em operações de compra e venda. A figura demonstra, para cada modelo, a porcentagem total de negociações de lucro e de prejuízo em relação à quantidade total de negociações emitidas. Conforme podemos observar, a maioria dos modelos concentrou os resultados dentro de uma faixa de porcentagem, tendo valores variados de porcentagem maior de lucro e prejuízo entre

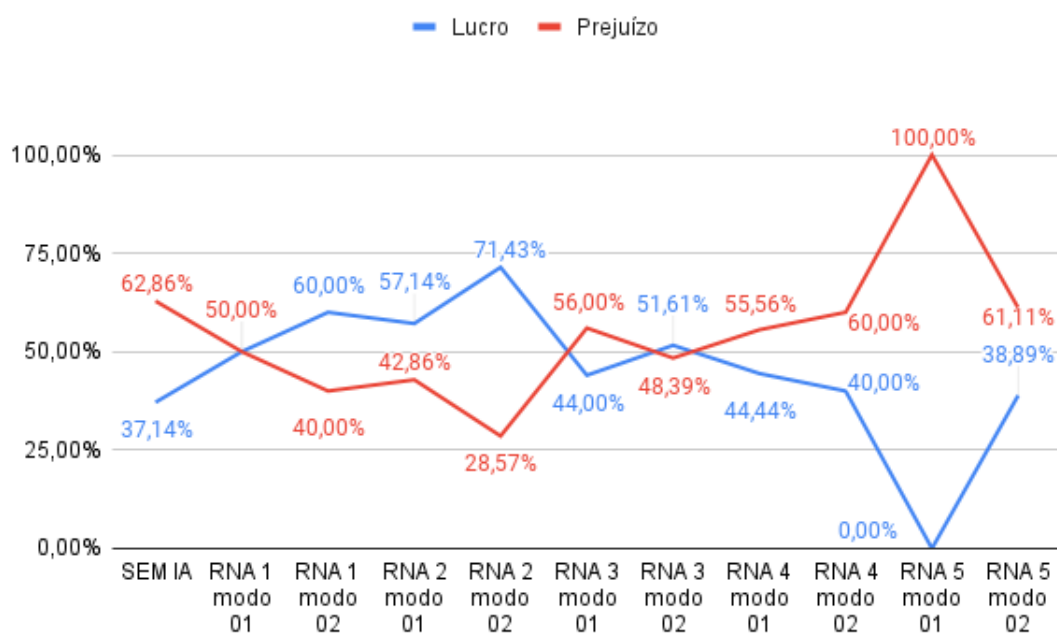
os modelos.

Tabela 19 – Negociações emitidas por modelo

Modelos	Total de negociações	Quantidade venda lucro	Quantidade compra lucro	Quantidade venda prejuízo	Quantidade compra prejuízo
Sem IA	35	10	3	20	2
Modelo 1 Modo 01	18	8	1	9	0
Modelo 1 Modo 02	25	13	2	9	1
Modelo 2 Modo 01	7	4	0	2	1
Modelo 2 Modo 02	14	8	2	3	1
Modelo 3 Modo 01	25	10	1	13	1
Modelo 3 Modo 02	31	15	1	13	2
Modelo 4 Modo 01	9	2	2	3	2
Modelo 4 Modo 02	5	0	2	0	3
Modelo 5 Modo 01	2	0	0	0	2
Modelo 5 Modo 02	18	7	0	10	1

Fonte: Elaborado pelo autor

Figura 39 – Porcentagem de negociações com lucro e prejuízo



Fonte: Elaborado pelo autor

5 Conclusão

*“As palavras fogem quando precisamos delas e
sobram quando não pretendemos usá-las.”
Carlos Drummond de Andrade*

Conforme já descrito na seção de introdução, fazer previsões de ativos na bolsa de valores ainda é um problema a ser solucionado de forma satisfatória, além de também considerar os riscos quando está exposto a esse tipo de mercado. Esses fatores se dão muito pelo caráter desse mercado de possuir grandes oscilações que pode ocasionar grandes perdas monetárias aos investidores. Diante esse cenário, o objetivo principal é realizar um estudo de caso aplicando RNA para realizar previsão do Mini Índice Bovespa visando lucros monetários.

Uma consideração a se fazer sobre esse trabalho, se refere a base de dados históricos para o treino das redes neurais. Com o uso da biblioteca “Renko 2.0 Offline”, criou uma base de dados na versão atemporal considerando, aproximadamente, os últimos 12 meses que foi o intervalo de dados que a base histórica tinha informação para *candles* de 1 minuto. Isso não impediu o treinamento das redes neurais, mas foi utilizado uma base de dados menor em comparação com uma base de dados temporal.

Uma segunda consideração é em relação à base de dados atemporal. Como houve uma conversão dos dados temporais para atemporais, o critério para formação dos *candles* agora sendo por pontos, gerou uma base com menos dados que uma base temporal, assim gerando menos *candles* para que o algoritmo do *Metatrader* ler e tomar decisões.

Os resultados desse trabalho serão descritos a seguir. O primeiro objetivo foi criar uma comunicação através de requisições HTTP na plataforma *Metatrader* durante o modo de testes de estratégias utilizando bases históricas. O problema foi resolvido ao usar a DLL *Wininet.dll*, descrito na seção 3.4.3, podendo assim realizar simulações das negociações (uso da conta de demonstração) utilizando as redes neurais, em horários que a bolsa de valores está fechada e não há negociações com o ativo trabalhado. Assim, solucionando uma das limitações citadas por Silva (2021).

O segundo objetivo foi aplicar o modelo de rede neural em gráficos atemporais usando o ativo mini índice Bovespa. Os dados coletados de lucro e negociações de cada rede neural, descrito de forma geral na seção 4.5.7, trouxe resultados com algumas redes neurais com lucros e outras com prejuízo. Das que geraram lucros expressivos podemos observar bons resultados como a topologia 1 e a topologia 3 na totalidade e o modo 2 da topologia 5.

Comparando os resultados das negociações usando RNA com a sem IA, podemos observar que em algumas variações de rede neural, o *setup* obteve lucros maiores, como na RNA 4 e na RNA 2 nos dois modos, e na RNA 5 modo 1. Nos demais casos, o algoritmo usando uma IA obteve valores monetários maiores. Um fator importante a ser observado também é a quantidade de negociações realizadas por cada modelo, ilustrado na tabela 18, que acabou

sendo o modelo sem uma IA.

Porém, também foram obtidos modelos de redes neurais que não abriram tantas negociações, como o caso da RNA 2 modo 1, RNA 4 modo 1 e modo 2 e a RNA 5 modo 1, que inclusive foi a única que não abriu negociação de venda. Isso evidencia a importância de ter realizados vários modelos de redes neurais com diferentes topologias e diferentes funções de ativação, podendo assim cobrir resultados mais diversos para um possível caminho de aprimoramento das redes criadas nesse trabalho.

Considerando somente a quantidade de negociações realizadas com lucro, o modelo que obteve a maior quantidade de negociações com lucro foi a RNA 3 modo 2 com 16 negociações com lucro (que representa 51,61% do total de negociações). Em segundo lugar foi RNA 1 modo 2 com 15 negociações com lucro (representando 60,00%). O terceiro lugar ficaria com o modelo sem IA com 13 negociações com lucro (representando 37,14%). Para esses três modelos a rede especializada em vendas obteve a maior quantidade de acertos, sendo bem maiores que nas redes de compra.

Analisando os resultados do modo 1 e do modo 2 das RNAs, de forma geral, o modo 2 se saiu melhor do que o modo 1, obtendo lucros maiores como no caso das RNA 1, RNA 2, RNA 3 e da RNA 5, que inclusive obteve uma reversão de prejuízo para lucro. Enquanto na RNA 4 que só teve prejuízos, o modo 2 obteve um prejuízo menor que o modo 1. Essa diferença tem como hipótese a forma como as informações sobre um sinal se preço de fechamento for menor que o de abertura e o volume de negociação, aliado com um novo dado no modo 2, a informação sobre a amplitude, foram trabalhadas através de operações matemáticas simples.

A quantidade de operações de venda, bem superiores à quantidade de operações de compra em todas as redes, pode ser explicado devido à tendência do gráfico no período de execução dos testes nos 13 dias. Nesse momento, gráfico estava com uma tendência de baixa, aliado com um dos parâmetros do algoritmo de negociação para respeitar a tendência do ativo, as ordens de venda foram abertas em maiores quantidades com expectativa de que nos próximos momentos o preço do ativo se desvalorizasse.

Como foi possível observar, alguns modelos obtiveram lucros, alguns superiores até mesmo ao modelo *setup* atingindo assim o objetivo geral de obter lucros monetários usando redes neurais com gráficos atemporais. Outros modelos, no entanto, não atingiram esse objetivo, tendo como hipótese os parâmetros utilizados, topologias e funções de ativação que não obtiveram uma boa combinação para obter lucro e em alguns casos não geraram quantidade de negociações relevantes, indicando que ainda há um espaço para melhoras.

Com o término desse trabalho, ainda existem possibilidades de trabalhos futuros para continuar e completar esse estudo de caso. Uma direção de caminho seria utilizar outros indicadores além dos utilizados neste trabalho como Bandas de Bollinger e o RSI. Outra sugestão seria alterar os parâmetros dos indicadores usados tentando outros valores, alinhado com estudo com outra possibilidade conversor de gráfico temporal para atemporal.

Uma outra direção de trabalho seria um estudo para analisar quais seriam os melhores dados de entrada para serem utilizados na coleta e no treinamento das redes neurais.

Outro estudo seria a aplicação da metodologia usada nesse trabalho, mas agora para a forma temporal desse ativo, variando valores de tempos de formação dos *candles* buscando melhores resultados. Complementando esse caminho, seria a utilização de outro tipo e configuração de *setup* que use outros critérios para emitir ordens de compra e venda.

Referências

AGGARWAL, C. C. et al. Neural networks and deep learning. *Springer*, Springer, v. 10, p. 978–3, 2018. Citado nas páginas 17, 18, 19 e 20.

ALMEIDA, A. J. S. MODELO DE PREDIÇÃO PARA o MERCADO ACIONÁRIO BASEADO NA LÓGICA FUZZY. 2015. Disponível em: <<http://tedebc.ufma.br:8080/jspui/handle/tede/289>>. Citado nas páginas 13 e 14.

ANDRADE, A. P. d. *O que é Flask?* 2019. Acessado em 02 de novembro 2021. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-flask>>. Citado na página 38.

B3. *Futuro de Ibovespa* / B3. 2020. Acessado em 19 de julho de 2021. Disponível em: <https://www.b3.com.br/pt_br/produtos-e-servicos/negociacao/renda-variavel/>. Citado nas páginas 13 e 30.

B3. *Histórico* - B3. 2020. Acessado em 19 de julho de 2021. Disponível em: <<https://ri.b3.com.br/pt-br/b3/historico/>>. Citado na página 13.

BARELLA, V. Técnicas para o problema de dados desbalanceados em classificação hierárquica. *ICMC-USP*, 2015. Acessado em 12 de fevereiro 2022. Disponível em: <https://www.teses.usp.br/teses/disponiveis/55/55134/tde-06012016-145045/publico/VictorHugoBarella_dissertacao_revisada.pdf>. Citado na página 21.

CHAGUE, F.; DE-LOSSO, R.; GIOVANNETTI, B. SSRN Scholarly Paper, *Day Trading for a Living?* 2020. Disponível em: <<https://papers.ssrn.com/abstract=3423101>>. Citado na página 25.

CLEAR. *Gráficos para operar na Bolsa: principais tipos e como utilizá-los*. 2018. Acessado em 02 de setembro 2021. Disponível em: <<https://blog.clear.com.br/graficos-para-operar-na-bolsa/>>. Citado nas páginas 23 e 24.

CLEAR. *Day Trade* / *Compra e venda de ativos no curto prazo*. 2021. Acessado em 20 de agosto de 2021. Disponível em: <<https://www.clear.com.br/>>. Citado na página 25.

CLEAR. *Ibovespa Futuro: saiba como funciona o mercado futuro do Índice*. 2021. Acessado em 20 de agosto de 2021. Disponível em: <<https://blog.clear.com.br/ibovespa-futuro/>>. Citado nas páginas 25 e 29.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. v. 2, n. 4, p. 303–314, 1989. ISSN 1435-568X. Disponível em: <<https://doi.org/10.1007/BF02551274>>. Citado na página 14.

DATAFLOQ. *Why Machine Learning and Artificial Intelligence will Never be Enough*. 2020. Acessado em 27 de outubro 2021. Disponível em: <<https://datafloq.com/read/why-machine-learning-artificial-intelligence-never-enough-completely-crack-stock-market/9309>>. Citado nas páginas 14 e 15.

DEBASTIANI, C. A. *Candlestick: um método para ampliar lucros na Bolsa de Valores*. [S.l.]: Novatec Editora, 2007. Citado nas páginas 21, 22 e 23.

FUTURA, N. *Principais Indicadores da Análise Gráfica, confira o artigo!* 2020. Acessado em 20 de outubro 2021. Disponível em: <<https://www.novafutura.com.br/principais-indicadores-da-analise-grafica/>>. Citado na página 25.

- GENIAL, I. *Setup trader: qual o segredo por trás de uma estratégia eficaz?* 2021. Acessado em 10 de fevereiro 2022. Disponível em: <<https://blog.genialinvestimentos.com.br/setup-trader-qual-o-segredo/>>. Citado na página 31.
- GENIAL, I. *Stop gain: o que é, como funciona e quais as vantagens?* 2021. Acessado em 10 de fevereiro 2022. Disponível em: <<https://blog.genialinvestimentos.com.br/stop-gain-o-que-e-como-funciona-e-vantagens/>>. Citado na página 33.
- GENIAL, I. *Stop loss: o que é, como funciona e como usar?* 2021. Acessado em 11 de fevereiro 2022. Disponível em: <<https://blog.genialinvestimentos.com.br/stop-loss/>>. Citado na página 32.
- GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. [S.l.]: "O'Reilly Media, Inc.", 2019. Citado nas páginas 18, 19 e 20.
- GOMES, F. R. A bolsa de valores brasileira como fonte de informações financeiras. v. 2, n. 2, 1997. ISSN 1981-5344. Disponível em: <<https://periodicos.ufmg.br/index.php/pci/article/view/23238>>. Citado na página 13.
- GÓMEZ, N. *Mini índice bovespa: Saiba o que é e para que serve*. 2019. Disponível em: <<https://www.euqueroinvestir.com/mini-indice-bovespa/>>. Citado nas páginas 13 e 14.
- HAYKIN, S. *Redes neurais: princípios e prática*. Bookman Editora, p. 183, 2001. Citado nas páginas 16 e 17.
- INVESTIDOR, B. d. *Tudo sobre o gráfico de candlestick*. 2021. Acessado em 22 de agosto de 2021. Disponível em: <<https://www.bussoladoinvestidor.com.br/grafico-de-candlestick/>>. Citado nas páginas 22 e 23.
- INVESTIDOR, B. do. *Índice de Movimento Direcional: a Força da Tendência*. 2013. Acessado em 20 de outubro 2021. Disponível em: <<https://www.bussoladoinvestidor.com.br/indice-de-movimento-direcional/>>. Citado na página 26.
- KUMAR, G.; JAIN, S.; SINGH, D. U. Stock market forecasting using computational intelligence: A survey. v. 28, 2020. Citado na página 14.
- LATORRE, M. d. R. D. d. O.; CARDOSO, M. R. A. Análise de séries temporais em epidemiologia: uma introdução sobre os aspectos metodológicos. v. 4, p. 145–152, 2001. ISSN 1415-790X, 1980-5497. Disponível em: <<http://www.scielo.br/j/rbepid/a/KM9MndgpCGSnjSNDddSydCG/abstract/?lang=pt>>. Citado na página 21.
- MARRETTI, R. B. L. Simulação de negociações em instrumentos do índice IBOVESPA utilizando machine learning. 2019. Disponível em: <<http://tede.mackenzie.br/jspui/handle/tede/4258>>. Citado na página 16.
- MARTINEZ, L. C. et al. From an artificial neural network to a stock market day-trading system: A case study on the BM and BOVESPA. In: *2009 International Joint Conference on Neural Networks*. [S.l.: s.n.], 2009. p. 2006–2013. ISSN: 2161-4407. Citado na página 14.
- METAQUOTES. *MetaTrader 5 Multi-Asset Trading Platform*. 2021. Acessado em 21 de agosto de 2021. Disponível em: <<https://www.metaquotes.net/en/metatrader5>>. Citado nas páginas 35, 36, 37, 42, 51, 52, 64, 65, 66 e 67.
- METAQUOTES. *Uso de indicadores técnicos - Gráficos, análise técnica e fundamental - Ajuda para o MetaTrader 5*. 2021. Acessado em 20 de outubro 2021. Disponível em: <https://www.metatrader5.com/pt/terminal/help/charts_analysis/indicators>. Citado nas páginas 25, 28, 32, 33, 34 e 35.

METAQUOTES, S. C. *Utilizando Winlnet.dll para a troca de dados entre plataformas via internet*. 2014. Acessado em 12 de março 2021. Disponível em: <<https://www.mql5.com/pt/articles/73>>. Citado na página 43.

MODALMAIS. *Contrato Futuro: o que é e tudo o que você precisa saber para operar*. 2021. Acessado em 20 de agosto de 2021. Disponível em: <<https://www.modalmais.com.br/blog/o-que-e-contrato-futuro>>. Citado na página 25.

MODALMAIS. *Plataformas de investimentos*. 2021. Acessado em 21 de agosto de 2021. Disponível em: <<https://www.modalmais.com.br/guia-de-plataformas/plataforma-de-investimento-como-escolher>>. Citado na página 35.

MOORE, M. *HiLo Activator: como usar e aproveitar as tendências nos gráficos*. 2021. Acessado em 23 de outubro 2021. Disponível em: <<https://portaldotrader.com.br/blog/analise-tecnica/hilo-activator-como-usar-e-aproveitar-as-tendencias-nos-graficos/>>. Citado nas páginas 28 e 29.

MURPHY, J. J. *Technical analysis of the futures markets*. New York Institute of Finance: New York, p. 34, 1986. Citado na página 14.

NASCIMENTO, L. A. et al. GERAÇÃO AUTOMATIZADA DE SÉRIES TEMPORAIS DE DADOS POSICIONAIS UTILIZANDO o IBGE-PPP. v. 23, p. 150–165, 2017. ISSN 1413-4853, 1982-2170. Disponível em: <<http://www.scielo.br/j/bcg/a/TJcrZyxZBF94fZhQbT4Dz7B/abstract/?lang=pt>>. Citado na página 21.

NELOGICA. *ProfitPro*. 2021. Acessado em 29 de outubro 2021. Disponível em: <<https://www.nelogica.com.br/produtos/profit-pro>>. Citado na página 37.

NIKOLAEV, K. *Ordem Trailing Stop - O que é e como funciona?* 2020. Acessado em 12 de fevereiro 2022. Disponível em: <<https://blog.earn2trade.com/pt/ordem-trailing-stop-como-funciona/>>. Citado na página 34.

OLIVEIRA, M. A. d. *Aplicação de redes neurais artificiais na análise de séries temporais econômico-financeiras*. 2007. text, 2007. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/12/12139/tde-31012008-112504/>>. Citado na página 21.

PEREIRA, A. P. *O que é um arquivo com extensão DLL?* 2013. Acessado em 12 de março 2021. Disponível em: <<https://www.tecmundo.com.br/o-que-e/44592-o-que-e-um-arquivo-com-extensao-dll-.htm>>. Citado na página 43.

PINHEIRO, L. *Setup do trader: como criar uma estratégia para operar no mercado*. 2021. Acessado em 29 de outubro 2021. Disponível em: <<https://blog.nelogica.com.br/setup-do-trader-como-criar-uma-estrategia-para-operar-no-mercado/>>. Citado na página 31.

POMMERANZENBAUM, I. R. *Redes neurais artificiais na predição das principais séries do Índice ibovespa e suas aplicações em sistemas automatizados de negociações*. 2014. Disponível em: <<http://pee.ufrj.br/teses/textocompleto/2014033101.pdf>>. Citado na página 16.

RASHID, T. *Make your own neural network*. [S.l.]: CreateSpace Independent Publishing Platform, 2016. Citado na página 17.

RUDGE, L.; CAVALCANTE, F. As bolsas de valores: estrutura e funcionamento. *Mercado de capitais*, v. 3, p. 169–204, 1996. Citado na página 13.

SANTOS, G. *Renko 2.0 Offline*. 2018. Acessado em 02 de julho 2021. Disponível em: <<https://www.mql5.com/pt/code/20254>>. Citado na página 40.

SILVA, I. N. D.; SPATTI, D. H.; FLAUZINO, R. A. Redes neurais artificiais para engenharia e ciências aplicadas-curso prático. *São Paulo: Artliber*, 2010. Citado na página 17.

SILVA, L. H. Estudo de caso: Aplicando rnas na cotação do contrato futuro de dólar comercial na bolsa de valores brasileira. 2021. Citado nas páginas 15, 16, 39, 40, 43 e 73.

TENSORFLOW. *TensorFlow*. 2021. Acessado em 02 de novembro 2021. Disponível em: <<https://www.tensorflow.org/?hl=pt-br>>. Citado na página 38.

TORO. *Tape Reading: o que é e como realmente funciona na prática?* 2021. Acessado em 02 de setembro de 2021. Disponível em: <<https://blog.toroinvestimentos.com.br/tape-reading-o-que-e-na-pratica>>. Citado na página 14.

TRADINGVIEW. *Bovespa Index-Mini Futures*. 2021. Acessado em 21 de agosto de 2021. Disponível em: <<https://br.tradingview.com/chart/?symbol=BMFBOVESPA%3AWIN1!>> Citado nas páginas 24 e 30.

WAZLAWICK, R. *Metodologia de pesquisa para ciência da computação*. [S.l.]: Elsevier Brasil, 2014. v. 2. Citado na página 39.

WIKIPEDIA. *Ramp function*. 2022. Acessado em 10 de fevereiro 2022. Disponível em: <https://en.wikipedia.org/wiki/Ramp_function>. Citado na página 19.

WIKIPEDIA. *Sigmoid function*. 2022. Acessado em 10 de fevereiro 2022. Disponível em: <https://en.wikipedia.org/wiki/Sigmoid_function>. Citado na página 18.

WIKIPEDIA. *Tangente hiperbólica*. 2022. Acessado em 10 de fevereiro 2022. Disponível em: <https://es.wikipedia.org/wiki/Tangente_hiperbólica>. Citado na página 18.