

**FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS –
FATECS
CURSO ENGENHARIA DA COMPUTAÇÃO**

João Alexandre Mendonça Marson
21605033

**Previsão de tendências de preços na bolsa de valores, utilizando redes
neurais profundas**

BRASÍLIA
2020

João Alexandre Mendonça Marson

Previsão de tendências de preços na bolsa de valores, utilizando redes neurais profundas

Trabalho de Conclusão de Curso (TCC) apresentado como um dos requisitos para a conclusão do curso de Engenharia Computação do UniCEUB– Centro Universitário de Brasília

Orientador (a): **Prof. MsC. Francisco Javier de Obaldia Diaz**

BRASÍLIA
2020

João Alexandre Mendonça Marson

Previsão de tendências de preços na bolsa de valores, utilizando redes neurais profundas

Trabalho de Conclusão de Curso (TCC) apresentado como um dos requisitos para a conclusão do curso de Engenharia da Computação do UniCEUB – Centro Universitário de Brasília

Orientador: **Prof. MsC. Francisco Javier de Obaldia Diaz**

Brasília, 2020.

BANCA EXAMINADORA

Prof. MsC. Francisco Javier de Obaldia Diaz
Orientador

Prof. MsC. Flávio Antônio Klein
Examinador

Prof. MsC. William Roberto Malvezzi
Examinador

Previsão de tendências de preços na bolsa de valores, utilizando redes neurais profundas

João Alexandre Mendonça Marson¹, Prof. MsC. Francisco Javier de Obaldia Diaz², Prof. MsC Flávio Antônio Klein³, Prof MsC William Roberto Malvezzi⁴

RESUMO

Neste trabalho aplicam-se técnicas de treinamento e criação de redes neurais profundas, utilizando uma abordagem que leva em conta o conhecimento disponível na área de análise de bolsas de valores. O processo é descrito, desde seu início, com a mineração e limpeza de dados. Explica-se alguns indicadores técnicos utilizados por analistas de mercado e como organiza-se as estruturas de dados. Alimentam-se esses para três modelos baseados em literatura. Demonstram-se procedimentos para melhorar os resultados dos modelos. Conclui-se que os resultados não são tão simples de serem verificados e quais mudanças geraram a melhor acurácia.

Palavras-chave: Redes neurais profundas, aprendizado profundo, previsão de mercado

Abstract: In this work it is applied training techniques and the creation of deep neural models using knowledge available in the field of technical analysis. The process is described from its beginning, going through data mining and cleaning. It is explained a few of the indicators used by market analysts, as well as how to organize the data. The data is fed into three models created based on available literature. It is shown how to improve these models results. And it is concluded that results are not easy to be verified and which changes optimized the accuracy the most.

keywords: Deep neural networks, deep learning, market forecast

¹ UniCEUB, aluno.

² UniCEUB, orientador.

³ UniCEUB, primeiro examinador.

⁴ UniCEUB, segundo examinador.

1 INTRODUÇÃO

Previsão de tendências de ativos financeiros de mercados são o foco de muitos estudos, mas devido a sua complexidade e imprevisibilidade, tem sido uma tarefa árdua para estatísticos e economistas.

Utiliza-se a teoria do mercado eficiente, entendendo que o preço de uma mercadoria reflete toda a informação sobre um ativo financeiro em um dado momento. Um mercado que fosse totalmente eficiente indicaria que é impossível ganhar do mercado, uma vez que qualquer informação disponível já estaria incorporada nos preços.

50,57% cenário entende-se que o preço é o meio de transmissão de informação sobre mudanças na relativa importância de mercadorias, em virtude de seus fatores de produção, oferta e demanda, até reflexos de ações políticas e catástrofes mundiais, mas que a propagação dessas informações não podem ocorrer instantaneamente pela sua impossibilidade física.

Vendo o preço como peça central nessa dinâmica, faz todo o sentido aplicar tecnologias de aprendizado de máquina para prever oscilações de mercado, com base em históricos, associando e ligando essas informações que inicialmente estão desconectadas entre si.

Entende-se que a estratégia tradicional é utilizar algoritmos de Machine Learning, aprender com dados históricos para gerar análises automáticas. Mas isso exige muitos dados para treinar o modelo, e ainda um esforço na categorização das informações.

Além disso, sabe-se que mercados diferentes se comportam de formas diferentes, e já é sabido que quando se utiliza machine learning para diferentes estratégias de investimento, como scalping, day-trade ou swing, os dados coletados serão categorizados de formas diferentes.

Então torna-se importante ter um sistema que seja automatizado, que possa ser treinado com mais facilidade para diferentes cenários.

A partir dessas considerações, a direção trilhada é a que demonstra uma mudança na forma de análise, até então. Enquanto as metodologias citadas até agora pressupõem tabelas e dados, ignora-se o histórico já existente em estratégias de investimento, sejam pelo método de velas, conforme descrito em (NISON 1991) ou com alguns indicadores econômicos dando sinais de movimento, como em (MURPHY, 1986) e (WILDER, 1978).

Baseado nessas pesquisas, replica-se o modelo apresentado com a proposta de implementação descrita neste trabalho.

2 REVISÃO BIBLIOGRÁFICA

A base de dados desse estudo fundamenta-se na implementação dos indicadores técnicos descritos em *Technical Analysis of the Financial Markets* de John J. Murphy e em *New Concepts In Technical*

Trading Systems de Welles Wilder. Esses indicadores têm função de indicar tendências e de reversões dos preços.

Vale observar que em (SEZER,2018) utiliza-se gráficos para alimentar uma rede neural convolucional (CNN-Convolutional Neural Network), o que é contraditório, uma vez que se ‘força’ a inteligência artificial a aprender padrões e informações que poderiam ser aprendidos pelos números. Detalha-se que essa alternativa tem sua valia quando analisados os processos de normalização de dados, uma vez ser muito mais simples o processo de normalizar imagens do que dados brutos.

Considerando também a dificuldade para acesso a equipamento de ponta para treinamento de inteligência, divergir da iniciativa de usar gráficos é decisão que irá facilitar o processo, já que os esforços iniciais de criação de imagens exige mais poder de processamento e acesso a mais memória.

Pelo demonstrado em (LIM,2020) pode-se ainda pensar em utilizar uma CNN sendo alimentada por uma série no tempo, análogo a uma imagem com duas dimensões, em que um eixo representa os dados, e o outro o tempo.

Apesar de considerar essa decisão, a ideia de usar séries de tempo em redes convolucionais gera uma série de problemas, uma vez que a análise de séries temporais não tem absolutamente nada de trivial em seu processo (NIELSEN, 2019). Por isso decide-se seguir os passos de (YANG et al, 2019), na sua proposta de passar agrupamentos de quinze dias com nove indicadores técnicos de um grupo de empresas. Mostra-se o processo de transformação desses dados para verificar suas alterações sempre que possível.

Nota-se em outro estudo que a utilização de indicadores técnicos específicos aumenta a acurácia de um sistema (ZHOU,2020). No exemplo, a Convergência e Divergência de Médias Móveis (MACD-Moving Average Convergence Divergence) aumenta a acurácia em até 20%, indicando que o resultado depende dos indicadores selecionados.

Levando em consideração que nem todo indicador funciona em qualquer mercado (PRADO,2013), entende-se que o processo de treinamento de uma rede neural deve ser capaz de identificar esses padrões, dependendo de seu mercado.

A utilização de redes neurais profundas convolucionais, não é por acaso. Em relação aos primeiros métodos utilizados de IA desde 2010, esse tipo de rede tem uma acurácia maior (TOVAR,2020).

Mesmo outras formas de IA ainda conseguem resultados melhores do que as melhores técnicas de econometria para casos de mercado (KAMALOV,2020).

Entende-se que, de alguma forma, o preço está relacionado a fatores de produção, economia, política e natureza (FRIEDMAN,1947). Dessa forma, o método de redução de dimensionalidade tem como objetivo, identificar padrões em dados, sem considerar

a relação inicial entre si (GATTO,2020).

Usa-se de guia as informações em (SEZER,2018), tanto para auxiliar na checagem dos cálculos dos indicadores, quanto para pensar sobre as facetas necessárias para se avaliar a performance dos modelos.

2.1 CONCEITOS BÁSICOS

Nesta seção definem-se os conceitos que serão utilizados no campo de mercado de ações, informações iniciais sobre as camadas de inteligência, descrição dos três tipos de modelos a serem utilizados e sobre a forma de manipulação dos dados.

2.1.1 DADOS DA BOLSA

As bolsas de valores repassam algumas informações referentes as transações que ocorrem durante um período de tempo. Essas transações podem ser de compra ou venda de um ativo. Denomina-se essa transação de trade. Consideram-se apenas os dados diários de fechamento, em vez de utilizar informações de intraday, que seriam referentes às oscilações do ativo durante o dia.

A decisão de utilizar os dados de fechamento é comum para a grande maioria dos indicadores técnicos, especialmente aqueles que se utiliza aqui.

Para cada dia analisado, são as seguintes informações disponibilizadas pela bolsa:

1. Data: Dia do pregão
2. Símbolo: Código do ativo
3. Abertura: Preço da primeira trade do dia.
4. Alta: Preço da trade mais cara do dia.
5. Baixa: Preço da trade mais baixa do dia
6. Close: Preço da última trade do dia.
7. Volume: Cálculo da multiplicação do número de ativos negociados pelo valor dos ativos.

A partir desses, pode-se calcular indicadores em uma janela de tempo definida.

2.1.2 INDICADORES TÉCNICOS

Indicadores são entendidos como uma técnica heurística e são aperfeiçoados conforme sua aplicação e estudo. Esses valores são calculados a partir das informações brutas disponíveis.

Nesta situação, utilizam-se 6 indicadores, sendo que um deles será decomposto em seus 4 sub indicadores, sendo um total de 9 indicadores descritos a seguir.

A maioria desses indicadores referem-se ao grupo de osciladores, que são especialmente úteis em auxiliar decisões em mercados financeiros, dando uma percepção de força aplicada a uma série que se reflete em uma tendência.

Em (MURPHY, 1986) observam-se três importantes usos dos osciladores:

- 1.) O oscilador tem sua maior utilidade quando seu valor chega a um extremo, indicando que o mercado está *comprado* se estiver na parte superior, e *vendido* quando estiver na parte inferior. Qualquer um dos casos indica que a tendência de preço está vulnerável.
- 2.) A divergência entre um oscilador e o preço aponta a virada da tendência.
- 3.) Ao cruzar no seu ponto mediano apresenta sinais de mudanças de direção nas tendências do preço, geralmente sendo o primeiro sinal.

Então é relevante lembrar qual informação o oscilador disponibiliza, quando está em seus extremos, ou sua informação diverge do preço, ou quando ele cruza seu ponto mediano.

2.1.2.1 RSI - RELATIVE STRENGTH INDEX

O RSI, conhecido como Índice de Força Relativa, é um dos indicadores mais comuns.

Define-se como **dia de ganho**, quando preço de abertura for menor que o fechamento, indicando que o preço subiu durante o dia. Temos o inverso definido como **dia de perda**, ou seja, o preço caiu.

Com essa definição, calcula-se a força relativa usando os últimos 14 dias. Esse valor de 14 é arbitrário e pode ser alterado, mantendo-se este, uma vez que é utilizado como padrão.

Com as informações de abertura e fechamento define-se quais dias dessa janela de 14 são dias de ganho e dias de perda e separa-se em dois grupos. Em cada um deles, será tirada a média referente à soma das variações diárias de preço de fechamento, e depois dividi-se o grupo de ganho pela de perda.

Isto é, conforme fórmula a seguir, para o total de dias *n* de ganho representado por 'g', e para o total de dias *m* de perda representado por 'p', faz-se uma somatória das diferenças de ganho e perda, tira-se sua média, e divide-se um pelo outro.

$$\frac{\sum_{i=1}^n (g_i.fechamento - g_i.abertura)/n}{\sum_{i=1}^m (p_i.abertura - p_i.fechamento)/m}$$

A partir desse cálculo, normaliza-se o valor para que o oscilador fique entre 0 e 100:

$$RSI = 100 - \frac{100}{1 + RS}$$

O RSI então é plotado em um eixo vertical de 0 a 100, considerando-se movimentos acima de 70 como comprados e abaixo de 30 vendidos.

Para entender melhor a relação entre o RSI e o preço de um ativo, observa-se o gráfico representado na fig 1.

Na parte superior, apresenta-se os valores de fechamento, e indica-se uma linha em vermelho e

outra em verde.

Na parte inferior, apresenta-se o RSI com duas retas de tendência, uma de baixa e outra de alta. Nota-se que, enquanto a baixa vai tendo picos de RSI menores, o preço de fechamento continua chegando no mesmo valor em vermelho, isto é, precisando de cada vez menos força para se manter no mesmo patamar.

Quando a tendência altera para a alta, nota-se que o preço ultrapassa o valor vermelho e atinge ao verde. Isso acontece porque havia força suficiente para esse aumento do preço de fechamento.

Essa é apenas uma das interpretações desse indicador. Recomenda-se sempre (MURPHY,1986) para um aprofundamento dessa informação.

Figura 1. Exemplo de MACD



Fonte: investing.com

2.1.2.2 LARRY WILLIAMS %R - WR

Este indicador foi criado pela percepção de que, em tendências de alta, o fechamento tende a ficar mais acima da faixa de oscilação do preço, enquanto que, em tendências de baixa, o fechamento tende a ficar abaixo da faixa de oscilação do preço.

Esse cálculo abrange um período n , em que

utiliza-se o fechamento f para o dia i e calcula-se, preço máximo $MaxN$ e o preço mínimo $MinN$ do período.

Divide-se a diferença do máximo com o fechamento do dia, pela diferença do máximo e do mínimo do período.

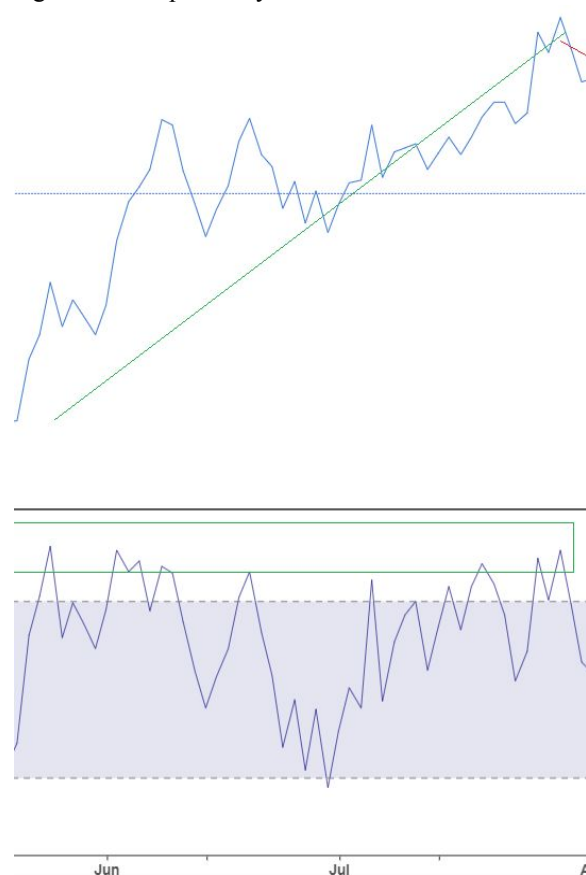
Multiplica-se o resto por -100, uma vez que a diferença de um preço maior que o máximo vai ser negativa, quando deveria ser o contrário, para padronizar o comportamento de um oscilador.

$$WR = \frac{(Maxn - fi)}{(Maxn - Minn)} * -100$$

A plotagem do Larry Williams não é diferente de outros osciladores, mas a sua interpretação em relação à tendência ocorre **apenas** quando o indicador atinge o extremo superior ou inferior.

No gráfico da Figura 2, observa-se o %R, representado na parte inferior, acima de sua banda, indicando a tendência de alta. No gráfico da Figura 3, nota-se o mesmo %R, indicando a tendência de baixa.

Figura 2 Exemplo Larry Williams 1



Fonte: investing.com

Figura 3 Exemplo Larry Williams 2



Fonte: investing.com

2.1.2.4 MACD - MOVING AVERAGE CONVERGENCE / DIVERGENCE

O indicador “convergência e divergência de médias móveis”, assim como o RSI, combina o princípio do oscilador com a comparação de duas médias móveis.

O MACD é representado graficamente por duas linhas e também, em alguns casos, um gráfico de barras representando a diferença entre essas.

Uma das linhas, chamada de linha do MACD, é a diferença entre duas médias móveis exponenciais dos fechamentos de preço em 12 e 26 dias:

$$MACD = MME12 - MME26$$

Enquanto a linha do sinal é uma média exponencial do próprio MACD de 9 dias.

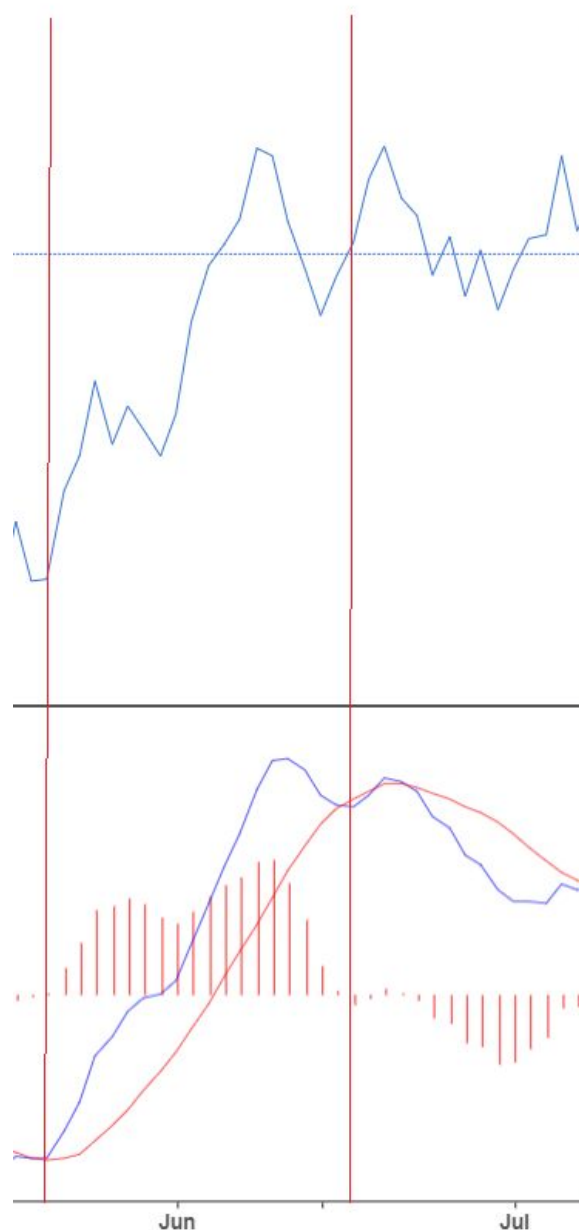
$$S = MMACD9$$

Opcionalmente, representa-se a diferença entre esses dois cálculos num histograma. Isso demonstra a força naquele instante.

$$H = MACD - S$$

Pode-se ver, no gráfico da Figura 4, como o MACD pode ser interpretado para auxiliar a estratégia de trading. Nota-se que o cruzar das linhas indica o início e o fim de uma tendência de alta.

Figura 4 MACD



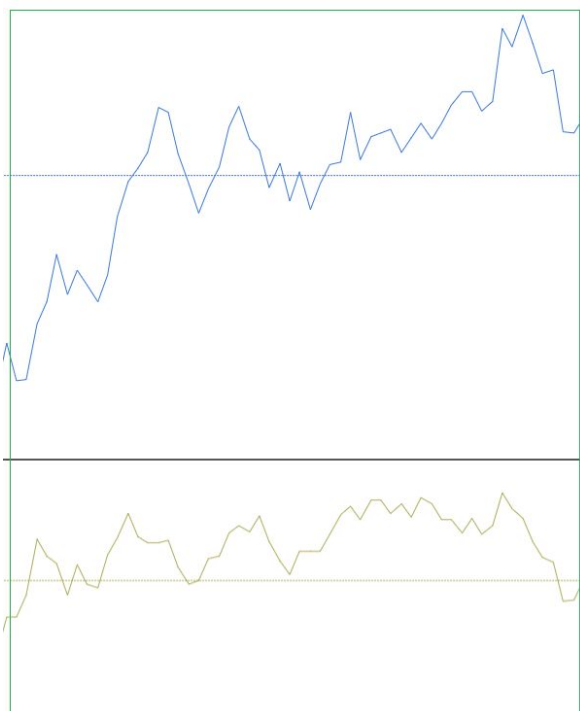
Fonte: investing.com

2.1.2.5 MTM - Momentum

O conceito de momento é a análise de um indicador mais básico. Esse mede a velocidade de mudança e é calculado com a diferença do preço de fechamento de hoje, com o de N dias atrás.

Observa-se na Figura 5, como o momento pode ajudar a encontrar pontos de tendência, mostrando como o preço irá se comportar.

Figura 5 Momentum



Fonte: inveting.com

2.1.2.6 DMI - Directional Movement Index

Em (MURPHY,1986) o indicador DMI é apresentado com um cálculo mais complexo que se torna facilitado pelo uso de computadores. Registra-se que esse foi descrito primeiramente por Welles Wilder em 1978.

O DMI consiste de 3 valores, que representam a pressão de venda (-DM), a pressão de compra (+DM) e a diferença entre essas (DX).

A média dessa diferença é ADX, enquanto o ADXR vai ser a taxa de variação de ADX.

Como mencionado anteriormente, esse índice será feito em 4 partes descritas a seguir.

2.1.2.6.1 TR - True Range

O True Range entra nos cálculos também e ele é o máximo entre 3 operações:

- 1.) Alta Atual - Baixa Atual
- 2.) Alta atual - Fechamento Anterior
- 3.) Baixa atual - Fechamento Anterior

Sempre o valor absoluto dessas operações que deve ser levado em consideração para o escolher o True Range do dia em questão.

Deve-se fazer uma Média Móvel de 14 dias com os valores para termos o ATR, Average True Range, que será utilizado para as transformações:

2.1.2.6.2 +DI - Plus Directional Indicator

Às vezes escrito como PDI, é o Plus Directional Indicator. Seu cálculo é feito a partir da média móvel de 14 dias de +DM dividido pelo ATR. +DM é calculado da subtração do preço alto de hoje pelo

preço alto de *n* dias atrás.

2.1.2.6.3 -DI - Minus Directional Indicator

Também escrito como MDI, o Minus Directional Indicator é feito a partir da média móvel de 14 dias de -DM dividido pelo ATR. -DM é calculado com o preço baixo de ontem pelo preço baixo de hoje.

2.1.2.6.4 ADX - Average Directional Movement Index

O DX vai ser o absoluto da diferença entre +DI e -DI dividido pelo absoluto da soma de +DI e -DI:

$$DX = \frac{|+DI - -DI|}{|+DI + -DI|}$$

A partir do valor de DX, faz-se um média móvel (também de 14 dias) para ter-se o ADX.

2.1.2.6.4 ADXR - Avg. Directional Movement Rating

ADXR vai ser a taxa de mudança de dois dias do ADX, isto é, a média do ADX de ontem e hoje.

Figura 6 Curvas para +DMI, -DMI e ADX



Fonte: dthelp.dynamictrend.com

Na Figura 6 apresenta-se como todos esses indicadores trabalham juntos para auxiliar na avaliação de força de uma tendência de mercado. Nota-se que o ADX acompanha as retas de subida e descida, encontrando um ponto ideal de entrada e saída do mercado.

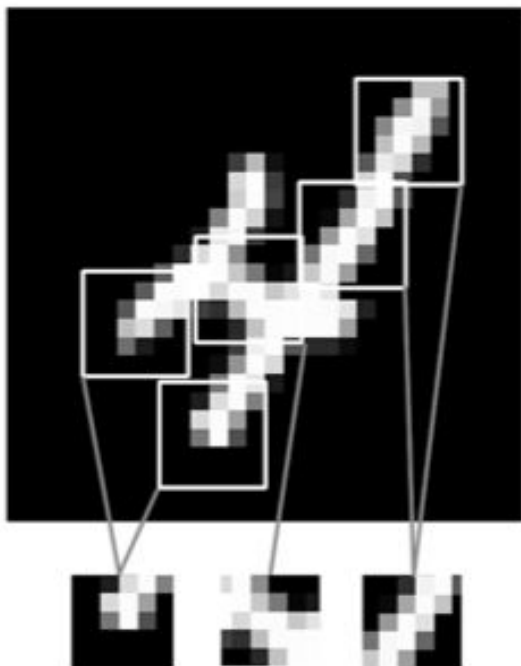
2.2. CAMADAS DE REDES NEURAIS

Em todos os estudos citados sobre redes neurais profundas foram descritas novas estratégias para empilhar camadas de processamento, de forma que se abstrai a implementação de sua complexidade, reduzindo-se o problema a otimização de hiperparâmetros da aplicação.

2.2.1 CONVOLUÇÃO

No contexto mais clássico, uma camada convolucional vai encontrar particularidades em pequenas janelas 2D dos dados.

Figura 7 Convolução em ação



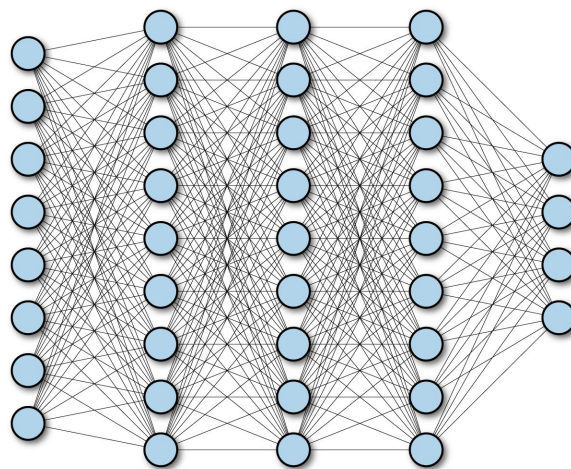
Fonte: DURR et all 2020

No caso de imagens, é possível retirar dessas pequenas janelas, padrões que indiquem, por exemplo, beiradas de objetos ou letras, conforme Figura 7.

2.2.2 FULLY CONNECTED

Esse caso particular ocorre quando se colocam camadas densas sequenciais, em que todos os neurônios de uma camada estão ligados a todos os neurônios da próxima camada, conforme demonstrado na Figura 8.

Figura 8 Fully Connected



Fonte: towardsdatascience.com

A função desta camada é diametralmente oposta ao processo de convolução, já que as redes totalmente conectadas têm por objetivo encontrar padrões que englobam todos os dados de entrada, em vez de procurar padrões em pequenos espaços de um grupo de dados.

Faz-se sua utilização depois de extrair os padrões particulares de uma informação, geralmente para diminuir o número de saídas ou para aumentar o número de combinações de dados disponíveis.

2.2.3 MAX POOLING

A camada Max Pooling tem função similar à da camada convolucional, mas em vez de utilizar transformação linear, usa-se a função máximo, que acaba por minimizar o número de parâmetros.

Isso deve ser feito porque ao conectar-se duas camadas convolucionais, ocorrem aumentos expressivos de parâmetros e o Max Pooling é uma das formas de manter isso sob controle.

2.2.4 DROPOUT

A camada Dropout é uma das mais utilizadas e mais efetivas no processo de regular as técnicas de redes neurais (GHOTRA,2018). Quando se aplica o dropout, aleatoriamente muda-se valores dos resultados para zero. Isso faz com que muito dos padrões encontrados sejam removidos, de forma que removem-se falsos positivos do processo de análise.

3 METODOLOGIA DO TRABALHO

Esse estudo tem como objetivo verificar os resultados de aplicação de inteligência artificial para previsão de preços de bolsa de valores. A natureza da pesquisa baseia-se em conceitos de inteligência artificial para previsão de preços. A abordagem do problema é quantitativa, pois serão verificadas as métricas de acurácia para avaliar o resultado.

A pesquisa é plenamente descritiva, com os

procedimentos descritos nesta seção, sua devida execução na seção 4, e a interpretação das conclusões na seção 6.

Toda a base de conhecimento deste estudo advém de artigos acadêmicos, mas a execução se baseia de informações de documentações de bibliotecas de software e até livros práticos de implementação de código.

O processo de coleta de dados é feito por uma biblioteca; seu processo de limpeza e preparação estão melhores descritos na seção 4.2

A linguagem escolhida para o estudo é Python, uma vez que é o campo que tem sido vanguarda nas áreas de ciência de dados, utilizando Anaconda para melhor gerenciar e organizar as ferramentas, acompanhado de Jupyter Notebook para facilitar a compreensão dos passos.

Utilizou-se a biblioteca YFinance para capturar dados referente à bolsa de São Paulo e a listagem das empresas do índice Bovespa para escolher quais ativos foram selecionados.

Os dados que serão coletados são os entendidos como os mais fundamentais para análise da bolsa: abertura, máxima, mínima, fechamento. Apesar do volume ser um valor fundamental, não é utilizado nos cálculos de nenhum indicador.

Essa pesquisa concentra-se nas decisões tomadas para a construção dos modelos descritos em três autores, como a seguir:

YANG utiliza uma construção de dados que gera matrizes 9x15, representando 9 indicadores técnicos em 15 dias. Converte-se esses dados num formato de 9x15x1, com o objetivo de reduzir a relação de posição entre os indicadores, isto é, como não usa uma imagem, utiliza essa quebra em canais para alimentar os dados em diferentes ordens, o que não afeta o resultado.

SEZER, para não correr o risco de alimentar informações duplicadas, utiliza uma janela -no caso de 15 dias-, e a cada período pula os dados utilizados para gerar a matriz anterior. Por exemplo se forem considerados 150 dias de dados, serão geradas apenas 10 matrizes. SEZER também propõe alimentar os dados de 5 em 5 anos, e depois verificar seus resultados com o ano seguinte de testes, dessa forma demonstra que os dados antigos, ou mais novos, resultam em uma mudança de acurácia.

Em LUCA, a janela é de 30 ao invés de de 15.

Os três modelos têm estruturas parecidas entre si, em que todos combinam camadas de convolução, com max pooling, sempre terminando com uma última camada fully connected, que resulta nas categorias de classificação.

Será utilizada a forma mais simples de classificação, espelhando-se as decisões na verificação que o preço de fechamento em 20 dias é maior ou menor do que o do dia atual (ZHOU,2020).

Todos os dados serão normalizados por Mínimo/Máximo de 0 a 1, considerando que os features da entrada tem peso igual.

Portanto nesse trabalho cria-se um modelo inicial baseado em LUCA, utilizando a mesma sequência de camadas e um valor mínimo para os hiperparâmetros disponíveis.

A partir desses resultados, modela-se 3 grandes modelos baseados nos autores apresentados:

LUCA

Idêntico ao modelo inicial, mas com aumentos de capacidade em suas camadas.

SEZER

Parecido com o modelo de LUCA, mas utilizando-se sequências de camadas de forma diferente.

YANG

Ainda parecido com os modelos anteriores, mas inclui-se uma camada antes da entrada dos dados, em que se transformam os dados iniciais de 15x9, isto é, onde se todos os indicadores são alimentados simultaneamente, para 1x15x1, em 9 camadas separadas. O objetivo principal dessa mudança é que a ordem dos indicadores apresentada não altere o resultado.

Em todos os casos, incluem-se funções de penalidade L2, funções de ativação relu (rectified linear). Posteriormente, usando keras tuner, tenta-se otimizar os valores dos hiperparâmetros disponíveis (GULLI, 2017).

Na última etapa, altera-se a janela de informações alimentada ao modelo LUCA.

4. APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

4.1 O MODELO INICIAL

Primeiramente segue-se as ideias de desenvolvimento que são apresentadas em https://keras.io/guides/sequential_model/ (data de acesso: 21/novembro/2020). Aqui se apresenta a forma direta de fazer um modelo mais simples de uma ótica mais abstrata, de forma que seja possível controlar algumas variáveis do processo, para refinar o sistema proposto.

Apresenta-se na Tabela 1 o sumário do modelo:

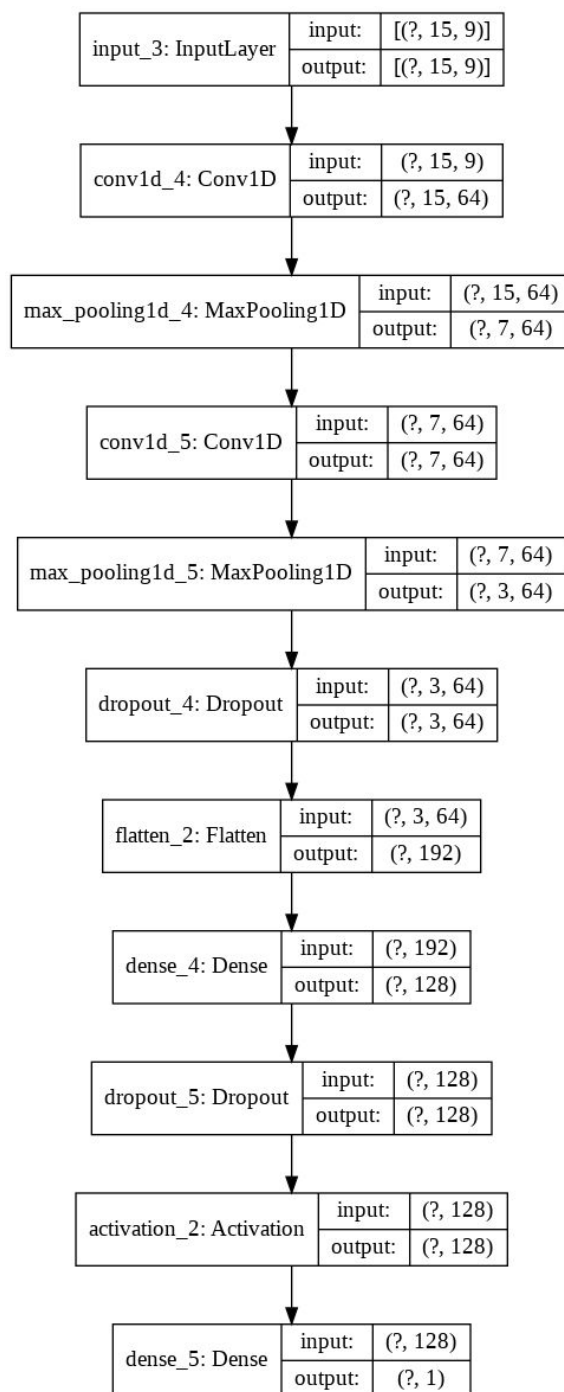
Tabela 1 Sumário modelo inicial.

Camada	Nº parâmetros
Conv1D	640
MaxPooling1D	0
Conv1D	4160
MaxPooling	1
Dropout	0
Flatten	0
Dense	24704

Dropout	0
Activation	0
Dense	129
Total de parâmetros	29633
Parâmetros treináveis	29633
Parâmetros Não Treináveis	0

Apresenta-se uma representação gráfica do modelo mostrando o processo de transformação dos dados de cada camada na figura 8.

Fig 8 Modelo Inicial



A camada inicial representa a estrutura de dados esperada na entrada do modelo. Input $(?, 15, 9)$ indica uma entrada do tipo 15×9 , em que ? representa o tamanho do lote de informações alimentado de cada vez. Como esse valor pode variar no treinamento, ele não é definido.

Nota-se na camada seguinte a transformação de 15×9 para 15×64 , implicando que a camada inicial de convolução gerou 7 vezes o número de features inicialmente alimentadas ao modelo.

Na sequência, o dado se transforma de 15×64 para 7×64 na primeira camada de MaxPooling, indicando um corte da metade das features identificadas anteriormente. A próxima convolução demonstra que

não é possível gerar novas features, a partir dos dados em questão. Enquanto a segunda MaxPooling consegue reduzir as features de 7x64 para 3x64. A camada de Dropout não altera o formato dos dados, ela apenas irá alterar um percentual dos valores de entrada para 0. A camada Flatten vai transformar nosso dado em um vetor único, pois isso é necessário uma vez que a camada Dense não consegue trabalhar com dimensão maior que 1. Verifica-se como as features mudam de 3x64 para o resultado do número de dados $3 \times 64 = 192$. Na sequência tem-se uma Dense Fully Connected de 128 neurônios, o que demonstra uma perda de dados, uma vez que não existe uma camada que esteja fazendo essa redução. Antes de ativar a função do modelo, apresenta-se outra camada de Dropout, terminando com uma última camada com apenas duas saídas, dando o sinal esperado para os dados de entrada.

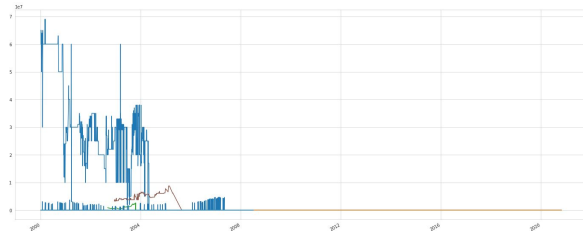
O modelo foi feito utilizando o mínimo possível de complexidade, sendo que, uma vez que no modelo original nenhuma das camadas tem função de ativação, exceto aquela antes de sua saída final, conforme documentação oficial do TensorFlow (https://www.tensorflow.org/tutorials/keras/overfit_and_underfit), (21/Novembro/2020), onde serão adotadas estratégias específicas para refinar o modelo proposto

4.2 FLUXO DE DADOS

Os dados da bolsa foram capturados usando a biblioteca YFinance e salvos em um banco de dados sequencial simples, SQLite3, interfaceado com a biblioteca sqlalchemy.

Primeiro observam-se os dados de fechamento de todas as empresas capturadas, conforme gráfico da Figura 9.

Fig 9. Dados bolsa 2000-2020



Observa-se na escala logarítmica do gráfico, a necessidade de limpeza dessas informações.

Como exemplo, verificamos qual é o maior valor de fechamento em nossos dados, e percebe-se que a CIA Hering SA, representada pelo ativo HGTX3 tem preços de fechamento superiores a 9.000.000, o que representa um erro dos dados, uma vez que nunca se teve uma ação custando tanto.

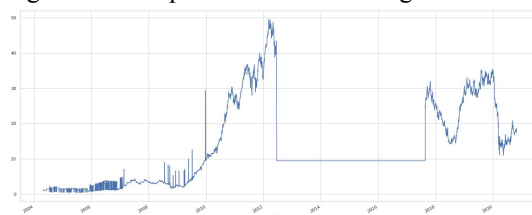
Plota-se o gráfico do ativo em questão para verificar a qual período se referem os valores, conforme figura 9.1

Fig 9.1 Dados históricos hering



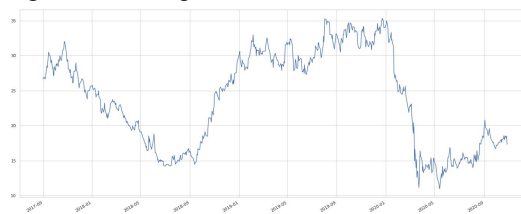
Num processo manual, verifica-se quais dados precisam ser removidos para que a escala do gráfico esteja dentro de um valor mais comum, no caso da Hering, ao remover os dados anteriores a maio/2014 verifica-se na figura 9.2 o resultado.

Fig 9.2: Dados pós-Maio/2014 Hering



A variação dos dados chega a um máximo próximo de 50, mas verifica-se uma reta horizontal entre 2012 e 2018, o que obriga a fazer um novo corte nos dados, desta vez em setembro/2017, conforme figura 9.3.

Fig 9.3: Dados após setembro/2017



Esse processo se repete para as outras empresas selecionadas.

Quando verificam-se os dados individuais de cada empresa em gráficos separados, observam-se algumas inconsistências em seus preços, conforme figura 10, e é possível identificar alguns pontos marcados em vermelho:

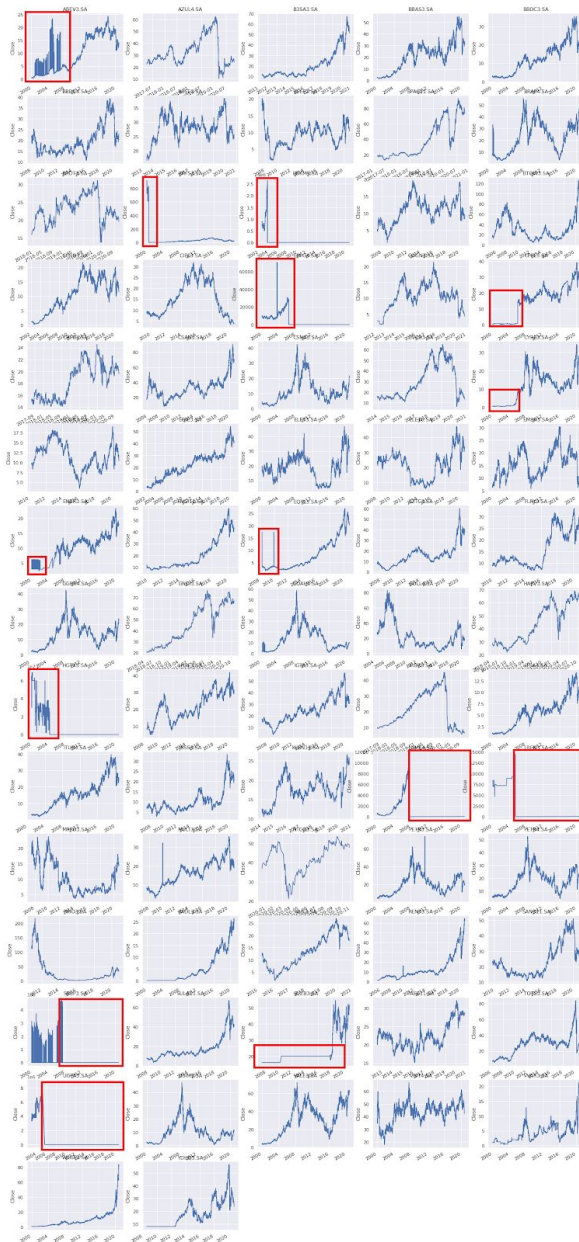


Fig 10 Gráficos individuais.

Nota-se, na figura 11, o resultado da remoção dos dados indicados pelos quadrados vermelhos da figura 10.

Cada empresa que continha preços acima de 100 para o fechamento foi analisada individualmente e teve parte de seus dados removidos.

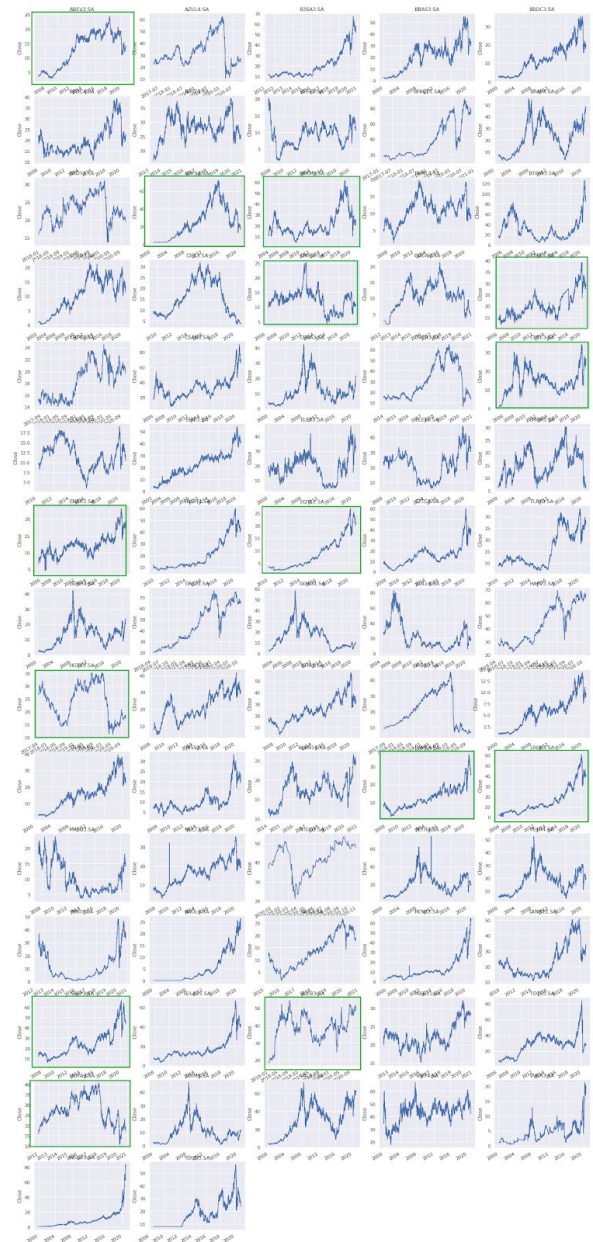


Fig 11 Gráficos Individuais corrigidos

O gráfico da figura 9 com os dados corrigidos fica com uma aparência uniforme, conforme a figura 12.

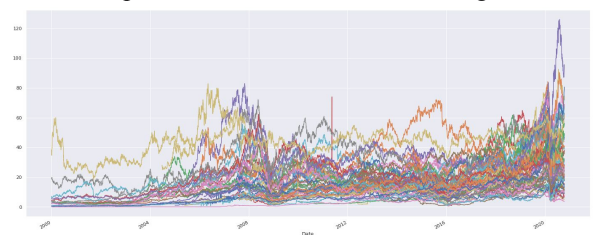


Fig 12 Dados uniformes

Esses dados foram feitos a partir de 826 empresas listadas no YFinance, gerando um total de 438750 dados. Lembra-se que 438 mil não é o total de dados de entrada para os modelos. Como foram usadas janelas de dados, essa informação é dividida pelo números de dias de cada parte, por exemplo, se

dividido por 15 dias, tem-se 29250, ou se dividido por 120 apenas 3656.

Com os dados limpos e organizados, são salvos no formato HDF (Hierarchical Data Formats) e comprimidos com 7z. Uma vez carregado no projeto, pode-se manipular as informações de cada grupo com mais agilidade. Esse controle é fundamental para automatizar o processo descrito em (SEZER,2018), em que se usa uma janela de cinco anos para treinar os dados e o próximo ano da sequência para testes. Pode-se também filtrar as informações utilizando o mesmo parâmetro de qualificação, que seleciona aquelas empresas negociadas com mais volume.

O resultado desse corte inicial é uma matriz 15x9 com os valores brutos dos indicadores, conforme figura 13.

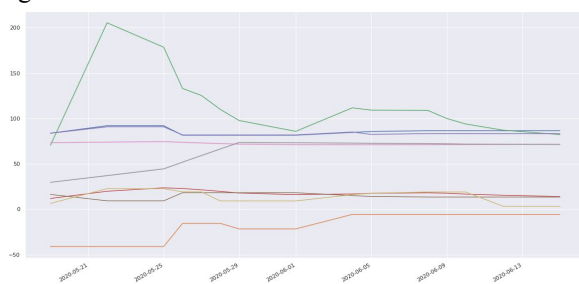


Fig 13 Exemplo de entrada bruta

A matriz 15x9 gerada é normalizada com mínimo e máximo entre 0 e -1 e resulta no gráfico da Fig 14.

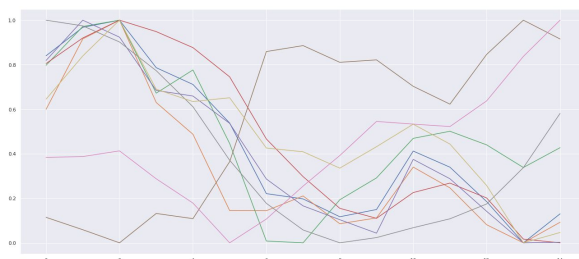


Fig 14 Exemplo de entrada normalizada.

Como utilizam-se dados desde 2000, consegue-se construir os datasets dessa forma.

Os dados de 2000-2004 representam aqueles que serão utilizados para treinar o modelo, enquanto os dados de 2005 em diante serão utilizados como testes para o resultado do modelo treinado.

4.3.1 DADOS DO MODELO MULTI CANAIS

Apesar da simplicidade para alterar o modelo, é fundamental notar que a entrada mudou. Uma vez que se tem 15 indicadores, mas 9 canais de entrada, o formato muda de 15x9 para 9x15x1.

Apesar do exemplo inicial, encontra-se fundamento em (CHAKRABORTY et al, 2020) com a aplicação de agrupamentos de dados de sensores de saúde para o diagnóstico de pacientes.

Portanto, é importante incluir a opção de retornar

dados em mais de uma dimensão, para que seja possível redistribuir os dados de entrada de acordo com o modelo.

4.4 RESULTADOS

Com o modelo inicial em mãos e uma estrutura para fornecimento de dados conforme descrito, chegou-se aos seguintes resultados.

4.4.1 RESULTADOS INICIAIS

O primeiro modelo retornou apenas 49.72% de acurácia nos dados de testes. Apesar do treinamento em 20 épocas ter alcançado valores de 76% e 75% para a acurácia em treinamento e validação.

A seguir aumentou-se a capacidade de processamento, criando 3 novos modelos:

O modelo de LUCA, em 1000 épocas, chegou a 76% e 75% de acurácia com os dados de validação, mas como o modelo inicial, sua predição com dados novos ficou em 50.96%. Dobrando a capacidade de cada camada e triplicando o número de parâmetros, treinou-se novamente o modelo, conseguindo 77% e 75% de acurácia com os dados de treinamento e validação. O resultado em dados novos foi de 51.13%.

A partir daí, foram colocadas algumas funções para auxiliar no processo da distribuição de peso das camadas, usando a penalidade L2. No modelo proposto atingiu-se 51.34% com novos dados.

O modelo de SEZER não tem resultados muito diferentes do modelo de LUCA, atingindo 76% e 77% de acurácia em treinamento e validação, com apenas 50,80% de acerto com dados novos. Aumentou-se a capacidade das camadas, apenas para conseguir menos acurácia em treinamento e validação, 74% e 73%, e apenas 49,95% com dados novos. Tentou-se aumentar os Dropouts e as ativações das camadas para diminuir o *overfit*, para conseguir 76% e 77% de acurácia no treinamento, e 50.89% com dados novos.

O terceiro modelo, YANG, aquele em que se usa uma camada de reshape, que transforma os dados iniciais 1x15x9 para 1x15x1 em nove sub camadas separadas, alcançou acurácia de 73.30% e 74%, mas com 50.21% com dados novos. Ampliou-se a capacidade do modelo aumentando as conexões nas camadas densas e usando mais filtros nas camadas de convolução. A acurácia ficou em 73.24% e 72,91%, e o resultado em dados novos atingiu 50.24%.

4.4.2 RESULTADOS DA MANIPULAÇÃO DE DADOS

Replicando as estruturas de SEZER de 2020 até 2004, nota-se que a utilização de dados mais recentes aumentou a acurácia em treinamento, refletindo marginalmente com dados novos.

Na figura 15, as linhas laranja e azul, bem no topo, apresentam valores de teste melhores do que as linhas cinza e verde, demonstrando que o grupo de dados mais antigo apresenta um resultado inferior.

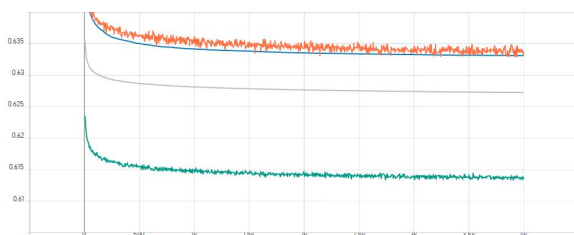


Fig 15 Acurácia de Validação 2000-2005 e 2015-2020.

Não existe nenhum resultado real na taxa de acurácia com dados novos, uma vez que a acurácia do modelo melhorou menos de 0.03% com as mudanças de dados.

Replicou-se, então, a ideia de YANG de fazer treinamento deixando os dados darem overlap entre si. Apesar da acurácia de treinamento e validação passar dos 90%, o resultado em dados novos é de apenas 5,6% em média para cada um dos modelos.

A ideia de LUCA utiliza uma janela de 30 dias, ao invés de 15 e aumentou a acurácia do modelo LUCA atingindo 52.06%, o de YANG para 52,00% e SEZER.

4.4.3 RESULTADOS COM JANELA DE DADOS

Os artigos originais de SEZER, YANG e LUCA utilizam janelas de dados de 11, 15 e 30 dias, outros como (EAPEN,2019) demonstram que janelas de 50 dias performam melhor.

Portanto, alimentou-se o modelo LUCA original com novos valores, que variam de 15 a 120 dias, resultando na relação da linha azul tracejada, apresentada na figura 16.

O aumento da janela de dados diminui o número de entradas do modelo proposto, com isso em mente, aumentam-se os dados alimentados ao modelo para uma janela de tempo maior; dez anos são utilizados para treinamento, entre 2008 e 2018, e o restante é usado como testes, 2019 e 2020. Esse resultado é representado na figura 16 pela linha sólida verde.

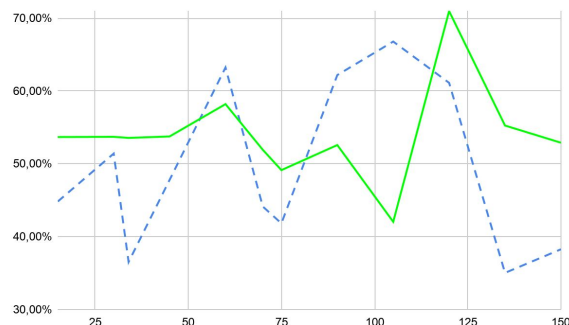


Fig 16 Variação de acurácia por janela de tempo

4.4.4 RESULTADOS HIPERPARÂMETROS

Seguindo a metodologia, onde verificamos quais os números ideais para alguns hiperparâmetros dos modelos, chegamos às conclusões a seguir.

Em toda camada de convolução, foram incluídas duas variáveis, uma para o total de filtros da convolução, e outra para a taxa de penalidade utilizada no processo de normalização L2. Os filtros variaram de 32 a 1024 e as taxas de penalidade entre os valores 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7 e 1e-8.

Em todas as camadas de dropout, foram colocados valores 1e-1, 2e-1, 5e-1, 7e-1, 1e-2, 2e-2, 5e-2 ou 7e-2 para serem utilizados como taxa de drop.

Todas as camadas densas, exceto a última, têm seu número de neurônios entre 32 e 1024 e seu regularizador L2 nos valores 1e-2, 1e-3, 1e-4, 1e-5 ou 1e-6.

Treinando os modelos com seus hiperparâmetros ideais, atingiu-se 51,57% com o modelo LUCA, 51,27% com o SEZER e 50,78% com YANG.

5 CONCLUSÃO

Conclui-se que o processo de classificação de dados financeiros de bolsas é um problema de resolução não trivial. Se os resultados de (YANG,2019) demonstram aproximadamente 60% de acurácia para uma classificação tripla, pressupõe-se que o mesmo seria possível para uma classificação binária de oscilação de preço, ignorando-se o detalhe de que o estudo original utilizou os ganhos e perdas das operações, para classificar os dados em situação de compra, venda ou segura, em vez de uma comparação simples de aumento de preço, dentro de um número de dias.

Demonstrou-se a necessidade de que os mesmos dados sejam alimentados para a rede várias vezes e que é necessário deixar separadas as janelas de tempo da série. Uma vez que, ao inserir dados duplicados o modelo passa a encontrar atalhos para o treinamento, tornando a acurácia com dados novos inferior a 6%.

Concluiu-se que o processo de escolha de hiperparâmetros auxilia as melhorias dos resultados de redes neurais e que, de todas as manipulações que foram feitas nos modelos, os melhores resultados ocorreram da manipulação dos dados, mais especificamente, da janela de dados que é alimentada para o modelo, chegando a uma melhoria de quase 20%, em relação à alimentação original.

Não foram combinados métodos múltiplos de otimização (inclusão de pesos com uma janela maior, ou mesmo procura de hiperparâmetros com essa mesma janela).

6 CONSIDERAÇÕES FINAIS

A utilização de inteligências artificiais para resolver problemas é uma realidade sem volta. Apesar da facilidade de criação de

modelos complexos, conclui-se sobre a importância do processo de tratamento de dados, demonstrando-se que foi possível aumentar a otimização de um modelo, apenas pela alteração da formatação de seus dados de entrada.

Percebe-se como uma forma de solução ingênua combinar diferentes aspectos de diversos projetos de inteligências, mas faz-se necessário um fluxo de trabalho, onde seja possível gerenciar as várias etapas do processo, já que se encontrou otimização dos resultados de um modelo, apenas pelo processo de mudança de seus dados iniciais.

Entende-se como curioso o resultado referente a variação da acurácia por janela de tempo, uma vez que não se esperava uma oscilação tão grande de acurácia baseado nessas alterações. No processo de criação dos indicadores, optou-se por algumas janelas de tempo para criação de médias e sinais. Pressupõe-se que essas decisões afetem diretamente a facilidade de interpretação dos dados, em relação a janela, implicando que o processo de manipulação de dados deveria fazer parte das primeiras camadas do modelo, tornando esses valores hiperparâmetros, que podem ser otimizados.

Neste estudo não se tratou os dados como uma série temporal, portanto, uma das formas de melhorar a performance de todos os modelos é analisar e implementar formas de análise que são relacionadas ao tempo, conforme (NIELSEN,2019).

Destaca-se que também é possível converter séries temporais em imagens, utilizando Campos Gramianos Angulares de Soma/Diferença (BAI,2020) ou mesmo em Campos de Transição Markovianos (XIA, 2018), dessa forma, seria possível utilizar plenamente camadas de convolução para literal interpretação de dados.

Observa-se que a análise de séries temporais é um vasto campo de estudo da estatística, e que tal análise também pode ser feita por meio de modelos de alisamento exponencial. Recomenda-se o capítulo nove do livro *Time Series: Theory and Methods*, sobre *Model Building and Forecasting with ARIMA Processes*.

AGRADECIMENTOS

Ao meu orientador, Prof. Me. Francisco Javier de Obaldia Diaz, pela orientação e confiança em mim depositadas.

A todos os professores do curso de Engenharia da Computação do Centro Universitário de Brasília, por todo conhecimento transmitido durante essa jornada.

Ao meu companheiro, Arthur Nobrega Rocha Xavier, pelo suporte e pela paciência demonstrada durante as horas em que me dediquei à elaboração do trabalho.

Aos meus pais, João Marson e Janine Mendonça Marson, sem os quais nada seria possível.

A todos que, de forma direta ou indireta, colaboraram para a realização e finalização deste estudo.

REFERÊNCIAS

BROCKWELL, Peter J., DAVIS, Richard A. **Time Series: Theory and Methods**.1991

CHAKRABORTY, Sabyasachi; AICH, Satyabrata; JOO, Moon-il; SAIN, Mangal; KIM, Hee-Cheol. **A Multichannel Convolutional Neural Network Architecture for the Detection of the State of Mind Using Physiological Signals from Wearable Devices**. 2020

CHOLLET, François. **Keras**. (2020) disponível em <https://keras.io/> Acesso em 21/Novembro/2020

DURR, Oliver; SICK, Beate; MURINA, Elvis **Probabilistic Deep Learning: With Python, Keras and Tensorflow Probability** 2020

FRIEDMAN, Milton. **Lerner on the Economics of Control**. *Journal of Political Economy*. 1947

GATTO, Laurent. **An Introduction to Machine Learning with R**. 2020

GHOTRA, Manpreet S.; DUA, Rajdeep. **Keras Deep Learning Cookbook**. 2018

GULLI, Antonio; PAL, Sujit. **Deep Learning with Keras**. 2017

KAMALOV, Firuz; IKHLAAS, Gurrib. **Machine learning based forecasting of significant daily returns in foreign exchange markets**. 2020

LIM, Bryan; STEFAN, Zohren. **Time Series Forecasting With Deep Learning: A Survey**. 2020.

MURPHY, John J. **Technical Analysis of the Financial Markets**. 1986

NIELSEN, Aileen. **Practical Time Series Analysis: Prediction with Statistics and Machine Learning** 2019

PRADO, Hércules; FERNEDA, Edilson; MORAIS, Luis; LUIZ, Alfredo; MATSURA, Eduardo. **On the Effectiveness of Candlestick Chart Analysis for the Brazilian Stock Market**. 2013

SEZER, Omer. OZBAYOGLU, Murat. **Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach** 2018

TOVAR, Wilfredo. **Deep Learning Based on Generative Adversarial and Convolutional Neural Networks for Financial Time Series Predictions**. 2020

WILDER, Welles. **New Concepts in Technical Trading Systems** (1978)

YANG, Zhenhua.; HAO, Kuangrong; Cai, Xin; CHEN, Lei; REN, Lihong. **Prediction of Stock Trading Signal Based on Multi-indicator Channel Convolutional Neural Networks**. 2019

ZHOU, Qiao; N, Liu. **A Stock Prediction Model Based on DCNN**. 2020