

RECONHECIMENTO DE PADRÕES DE *CANDLESTICKS* UTILIZANDO DEEP LEARNING

Gabriel Vinicius Boin Freitas

Orientador: Marcelo Cohen

PUC-RS Online

`gabriel.freitas@dicasdabolsa.com.br`

2023, Março

Resumo

Com o avanço do *machine learning* e especialmente das redes neurais convolucionais, que compõem o *deep learning*, na segunda metade da década de 2010, muitos autores tentaram elaborar como reconhecer automaticamente padrões de *candlestick* em um gráfico, de forma probabilística.

Alguns realizam *encoding* das séries temporais que compõem um *candlestick* e então utilizam redes neurais convolucionais para classificar os padrões e, outros, utilizam a representação *2d* do próprio gráfico de *candles* para ser analisado pelo *deep learning*, mas, o mais assertivo, que aparece já no começo de 2023, é utilizar a implementação determinística algorítmica para tal feita.

Mesmo assim, nada impede ou desabona a utilização das técnicas de inteligência artificial para descoberta de novos padrões e construção de incrementos à teoria dos *candlesticks*.

O artigo aplica *transfer learning*, *data augmentation*, *dropout* e *batch normalization* para treinamento de rede neural convolucional de referência (pré-treinada) VGG19, e consegue resultado de acerto na classificação de 68%, superior ao artigo de referência utilizado para comparação que obteve taxa de acerto de 62%, porém, inferior ao resultado obtido em artigos que utilizam *encoding* do gráfico de *candlestick* antes do treinamento da *CNN*.

A utilização de uma rotulagem automática dos padrões de *candle*, através de algoritmo determinístico e posterior treinamento de *CNN* utilizando-se a mesma metodologia deste artigo pode aumentar a assertividade e utilidade do modelo.

Palavras-chaves: candlestick, image recognition, pattern recognition, financial vision.

Sumário

Sumário	2
1 Introdução	2
2 Trabalhos Relacionados	5
3 Metodologia	8
4 Resultados	15
5 Discussão	15
6 Conclusão e Trabalhos Futuros	17
Referências	17

1 Introdução

Em cada segundo, em todo planeta, existem inúmeras transações de compra e venda de ativos em bolsas de valores.

Os preços de ações de empresas nos mercados de capitais variam constantemente, pois, refletem, nas ordens de compra e venda dos investidores, as expectativas do mercado em relação ao valor da empresa que aquela ação representa uma parcela.

Poder prever corretamente a tendência, de subida ou descida, no preço de uma ação, pode gerar ganhos financeiros e diminuir riscos de investimentos.

Muitas são as formas de tentar entender os preços dos ativos listados na bolsa.

Basicamente, existem 2 tipos:

- **análise fundamentalista**, que analisa vários indicadores pertinentes à empresa e ao mercado e precifica o valor justo de um ativo, de forma que as flutuações no preço de compra e venda das ações reflitam sentimentos diferentes dos investidores quanto a algumas variáveis discricionárias na precificação dos ativos;
- **análise técnica**, onde indicadores baseados em análise de séries temporais auxiliam a tomada de decisões de compra e venda dos ativos, entendendo que o preço segue certa lógica cíclica do mercado.

Pensando no segundo grupo de técnicas, do Japão, no final do século XX, para todo o mundo, difundiu-se a técnica de *candlestick*.

O gráfico de *candlestick*, candelabro, em inglês, nasceu na forma de se registrar as variações dos preços no mercado de arroz japonês.

Analisando estes gráficos, foram descobertos padrões no conjunto de *velas* que indicavam maior chance de subida ou descida nos preços dos arroz nos próximos pregões.

O *papa* destes conceitos é um dos livros mais famosos sobre o assunto é (NISON, 1994). Neste livro, ele conceitua o que são os *candles*, velas, em inglês, e os candelabros, *candlesticks*.

Candlesticks são gráficos que vão mostrar como variam os preços negociados de um ativo na bolsa de valores.

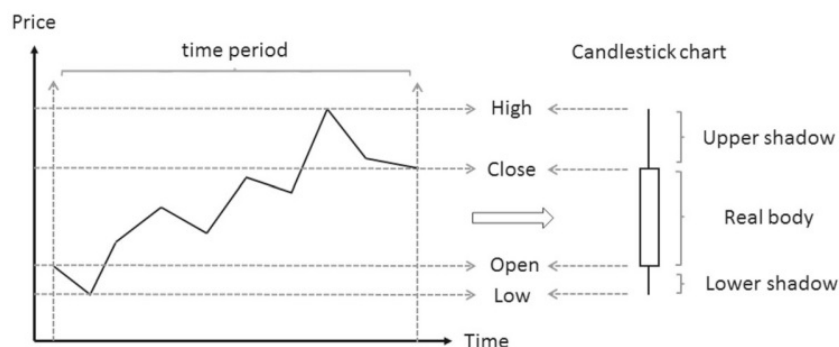


Figura 1 – Como é formado um *Candle*. Seu corpo é composto pelo preço de abertura e fechamento, enquanto que o preço mais alto e mais baixo do pregão fazem parte de suas sombras, superior e inferior, respectivamente. Ele representa, de forma compacta, o que ocorreu no período. O normal é ele representar um dia do pregão, para determinado ativo. Fonte: (CHEN; TSAI, 2020)

Cada *candle* é um período de pregão, normalmente um dia, mas podem ser criados *candlesticks* onde cada *candle* represente 15 minutos do pregão, ou onde cada *candle* represente um mês do pregão, a lógica para criação será igual e o importante é o período que cada *candle* representa ser uniforme, conforme apresentado na figura 1.

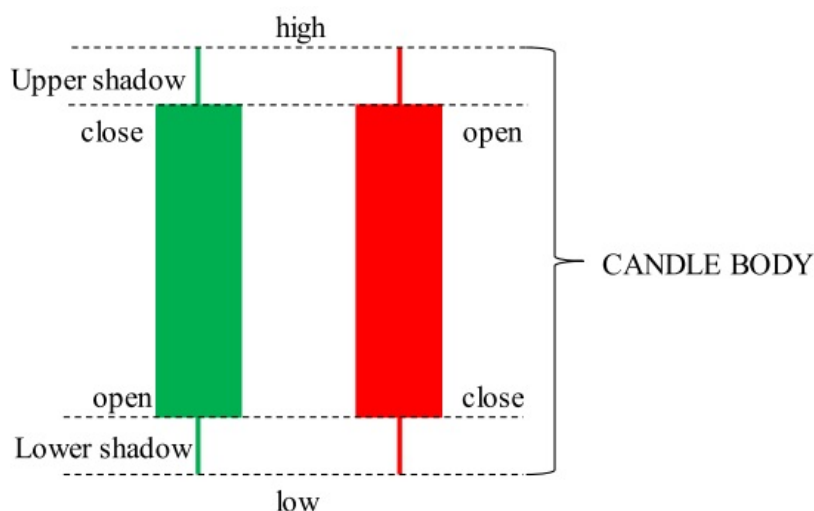


Figura 2 – Exemplos de *candles* de alta ou baixa. O de alta em verde e o de baixa em vermelho. Fonte: (CHEN; TSAI, 2020)

Um *candle* é formado por alguns valores do pregão (se forem *candles* diários) ou período de tempo que ele representa: preço de abertura do pregão, preço mais baixo do pregão, preço de fechamento do pregão e preço mais alto do pregão.

Além disto, embaixo de cada *candle*, deve-se ter um gráfico de barras (uma barra por *candle*) que apresente o volume negociado no pregão.

Caso o pregão tenha tido alta nos preços, o candle é branco, do contrário, ele deve ser preto. Estas cores advém do *Yin Yang*.

Normalmente, nos softwares de mercado, se adotam o verde para *candles* de alta e o vermelho, para os de baixa.

O nome *candle* se dá por o gráfico formado pelos vários *candles* se parecer com um candelabro, *candlestick*.

A figura 2 apresenta exemplos de *candles* de alta (verde) e baixa (vermelho).

Pattern (duration in candles)		
<i>Low reliability</i>	<i>Fair reliability</i>	<i>High reliability</i>
Doji (1)	BreakAway (5)	Closing Marubozu (1)
Gap Side Side White	Doji Star (2)	ConcealBabysWall (4)
Three Crows (3)		
Hammer (1)	Dragonfly Doji (1)	Counter Attack (2)
Hanging Man (1)	Engulfing (2)	Dark Cloud Cover (2)
Harami (2)	Gravestone Doji (1)	Evening Doji Star (3)
Harami Cross (2)	Hikkake (3)	Evening Star (3)
High Wave (1)	Hikkake Mod (3)	Falling Three Methods (3)
Inverted Hammer (1)	Homing Pigeon (2)	Identical Three Crows (3)
Long Legend Doji (1)	Ladder Bottom (5)	In Neck (2)
Rickshaw Man (1)	Long Line (1)	Kicking By Length (2)
Separating Lines (2)	Matching Low (2)	Kicking (2)
Shooting Star (1)	On Neck (2)	Maruzobu (1)
ShortLine (1)	Piercing Line (2)	Mat Hold (5)
SpinningTop (1)	StalledPattern (3)	Morning Doji Star (3)
Takuri (1)	Stick Sandwich (3)	Morning Star (3)
Three Line Strike (3)	Tasuki Gap (3)	Rising Three Methods (3)
Thrusting (2)	Three River (3)	Three Black Crows (3)
	Three Stars in the South (3)	Three Inside Up (3)
	Three White Soldiers (3)	Three Outside Up (3)
	Tri-Star (3)	Upside Gap Two Crows (3)
	Two Crows (3)	
	Upside Gap Three Methods (3)	

Figura 3 – Exemplos de padrões de *candlestick* com seu grau de certeza e quantos períodos os compõem. Fonte: (CAGLIERO; FIOR; GARZA, 2023)

Conjuntos de *candles* são nomeados por (NISON, 1994) e vários padrões são identificados, como tendo efeitos de uma possível alta ou baixa, nos períodos que os seguem. A figura 3 apresenta vários padrões famosos.

Conforme será apresentado na seção de trabalhos relacionados, há varias iniciativas para se identificar automaticamente os padrões de *candlestick* nos gráficos, de forma a facilitar a análise do investidor que não possui conhecimento profundo sobre o assunto, mas que quer obter um prognóstico utilizando tal técnica, informando se o mercado irá subir ou cair.

Além disto, automatizar a identificação dos padrões de *candlestick* ajuda a criar modelos mais sofisticados de *trading* por robôs e pode contribuir para validar e incrementar a própria teoria que cunhou os padrões de *candlestick*.

2 Trabalhos Relacionados

Em 2015, (WANG; OATES, 2015), inaugura a utilização das técnicas *Gramian Angular Summation/Difference Fields* GASF e GADF; e *Markov Transition Fields (MTF)*¹, para transformar uma série temporal em imagens e utilizar estas imagens, depois de padronizadas, na sua classificação, utilizando uma CNN², fazendo a aplicação em 20 bases de dados conhecidas e obtendo bons resultados com o método.

Conseguiu um menor MSE ³ em 12% a 48%, em comparação às abordagens de comparação, além de ter vencido como sendo o melhor modelo em 9 das 20 bases usadas.

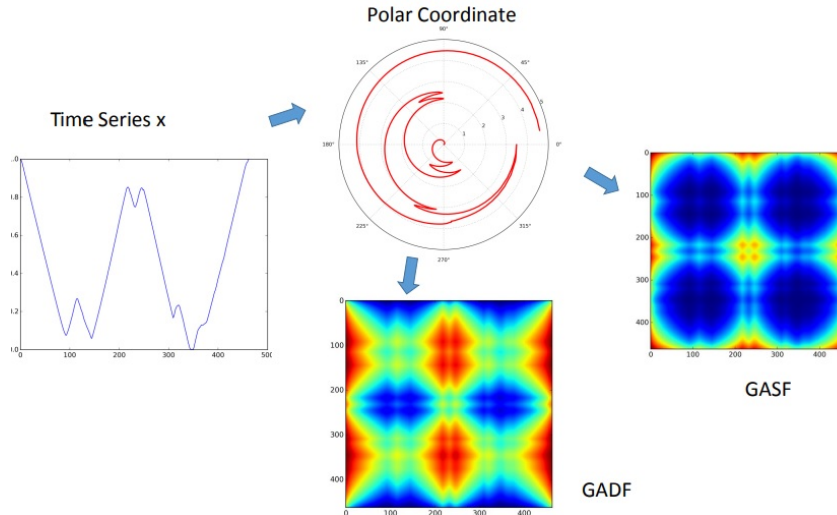


Figura 4 – *Encoding* usando o *GASF* / *GADF*. Primeiro, se interpola os valores da série temporal em um gráfico de coordenadas polar. Num segundo momento, utilizando fórmulas, se calcula o *GASF* e *GADF* e se constrói duas matrizes. Ao final, se cria um *heatmap* para cada uma das matrizes geradas. Estas imagens geradas serão enviadas para uma CNN. Fonte: (WANG; OATES, 2015)

Em 2018, (HATAMI; GAVET; DEBAYLE, 2018), apresenta como transformar uma série temporal em uma imagem utilizando *Recurrence Plot*⁴.

Esta imagem é padronizada e então passa por uma rede neural convolucional, para classificar cada série temporal. São utilizadas as mesmas bases de dados que (WANG;

¹ *Gramian Angular Summation / Difference Fields* GASF e GADF são utilizados para transformar uma série temporal em imagem da seguinte forma: Primeiro, se interpola a série em um gráfico polar, utilizando-se de arco cossenos da série transformada no intervalo de $[-1a1]$ ou de $[0a1]$, de forma em que o tempo seja o raio no gráfico. Em um segundo momento, se calcula o valor de *GASF/GADF*, utilizando suas fórmulas. Depois disto, é criado um *heatmap* para as 2 matrizes geradas. Também é feito o *encoding* das séries temporais utilizando-se da técnica de *Markov Transition Fields (MTF)*. (Esta técnica é apresentada na figura 4).

² CNN é acrônimo de *Convolutional Neural Network*, rede neural convolucional.

³ *MSE* quer dizer *Mean Square Error*, erro quadrático médio. Verifica-se a diferença entre o valor previsto e o valor esperado/conhecido, eleva-se ao quadrado, soma-se todos os desvios e se calcula a média.

⁴ *Recurrence Plot*: Em um primeiro passo, a série temporal (uma dimensão: o valor x e o tempo) é transformada em uma matriz de coordenadas (x,y) , onde x é cada valor no tempo t , e y , o valor em $t+1$. No segundo passo, é construída a imagem a partir de um *heatmap* da distância entre cada ponto (x,y) criado na matriz. A técnica é ilustrada na figura 5.

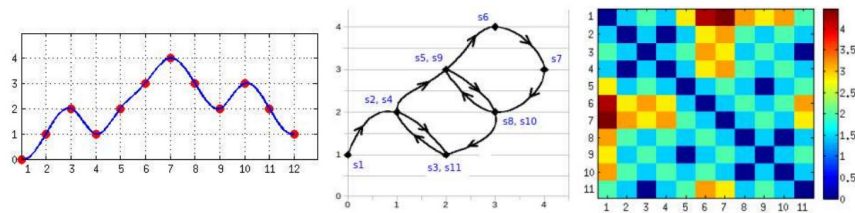


Figura 5 – *Encoding* usando o Recurrence Plot. Em um primeiro passo, a série temporal (uma dimensão: o valor x e o tempo) é transformada em uma matriz de coordenadas (x,y) , onde x é cada valor no tempo t , e y , o valor em $t+1$. No segundo passo, é construída a imagem a partir de um *heatmap* da distância entre cada ponto (x,y) criado na matriz. Fonte: (HATAMI; GAVET; DEBAYLE, 2018)

OATES, 2015) e com a nova forma de gerar as imagens, a metodologia se apresentou mais assertiva, obteve 10 vitórias nas mesmas 20 bases.

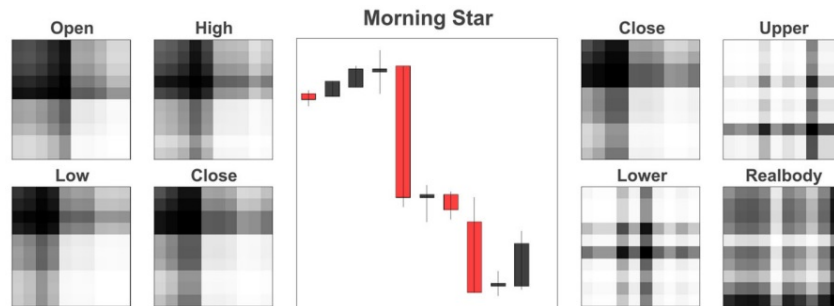


Figura 6 – *Encoding* das séries temporais que formam o *candlestick*. Exemplo de um padrão *morning star* com *encoding*. Do lado esquerdo do padrão de *candlestick*, *OHLC* e do lado direito, *CULR*. Fonte: (CHEN; TSAI, 2020)

No artigo (CHEN; TSAI, 2020), é utilizada a técnica de *Gramian Angular Fields*, utilizada em 2015, para fazer o *encoding* em imagens de 8 padrões de *candlestick*.

É utilizada uma janela temporal de 10 candles para a criação de 8 imagens que representam 2 conjuntos de séries temporais que compõem a evolução de 10 *candles OHLC*⁵ ou *CULR*⁶.

Ambos conjuntos de 4 imagens são usados para treinar a rede neural convolucional, um de cada vez, e se verifica que a utilização do *CULR* é mais performática para a classificação dos padrões de *candles*. 92,42% de acurácia média.

Neste artigo, o autor cunha o termo *Financial Vision*, como técnica para se enxergar padrões financeiros pela máquina. A figura 6 apresenta esta técnica de *encoding* para um padrão de *candlestick*, *morning star*.

⁵ *OHLC* : Opening, high, low, closing prices. Preço de abertura, maior preço, menor preço e preço de fechamento do pregão.

⁶ *CULR*: Closing prices, upper shadow, lower shadow, real-body. Preço de fechamento do pregão, tamanho da sombra superior, tamanho da sombra inferior e tamanho do corpo do *candle*, que é a diferença entre o preço de abertura e fechamento.

Ainda, em 2020, (CHEN et al., 2020), utiliza técnica de ataque adversário⁷ no modelo criado em seu artigo anterior para demonstrar que o modelo de rede neural convolucional⁹, realmente aprendeu a reconhecer os 8 padrões de candlestick propostos.

Pela publicação (SHAH, 2020), o autor apresenta código em *Python* de geração automática dos gráficos de *candlestick* para um conjunto pré-definido de ativos, com *labelling* automático das imagens em duas classes: *Up* (padrões de alta) e *Down* (padrões de baixa).

O código utiliza a biblioteca *fastai* do *Python* para realizar *data augmentation*, reduzindo *overfitting* e o autor utiliza *transfer learning* da rede profunda pré-treinada *resnet34*¹⁰.

Ele consegue acurácia de 62% no reconhecimento de padrões de alta ou baixa de *candles*, porém, sem classificá-los em suas nomenclaturas originais.

O autor utiliza *heatmap* em gráficos candlestick onde os padrões de *alta* ou *baixa* foram identificados, de forma a iluminar a região onde um padrão relevante acontece.

No trabalho de (BIROGUL; TEMÜR; KOSE, 2020), é utilizada a tecnologia *YOLO*¹¹, para identificar padrões de *candlestick*, fornecendo um sinal de compra ou venda do ativo, que junto com outros indicadores da análise técnica de preços de ativos, consegue-se performar corretamente e de forma automática no mercado de ações em 83,93% dos casos estudados.

Neste trabalho, também não são classificados os *candlesticks* em suas várias classes, mas sim, apenas, padrão de alta ou baixa.

Em 2021, no artigo (XU, 2021), utiliza o *encoding* dos *candlesticks* baseados na técnica de *GAF*, similar ao realizado em (CHEN; TSAI, 2020), porém, utiliza técnicas mais simples de *machine learning* para a classificação dos padrões de *candlestick*, concluindo que *XGBoost*¹² obteve melhores resultados do que a aplicação de uma rede neural convolucional profunda.

Utilizando *YOLO*, também, (PATHAK; TAMBOLI; TAMBOLI, 2022) treinam modelo com padrões pré-definidos de *candlestick* e obtém bons resultados. Não especificam a acurácia no artigo, porém, apresentam como utilizar um software de *labelling*¹³ - *LabelImg*

⁷ Ataque adversário é uma técnica que tenta enganar a rede utilizando-se de imagens que contém *pixels* proposadamente modificados de forma que ativem certos nodos específicos que farão a rede errar a classificação. Se a rede aprendeu de forma superficial um problema de classificação, poderá cair mais facilmente neste tipo de ataque, caso contrário, é possível utilizar tal técnica para mostrar que a rede aprendeu *features*⁸ robustas no tratamento do problema de classificação em questão.

⁹ *CNNs* e modelos de *deep learning* em geral têm sua explicabilidade mais complexa do que modelos de *Machine Learning* mais simples. Redes neurais são chamadas de caixas pretas por muitos autores. Existem muitas técnicas de visualização das camadas intermediárias que podem mitigar tal situação.

¹⁰ *Resnet34* é uma *CNN* de referência, pré-treinada, para classificação de imagens. Ela utiliza técnica de aprendizado residual em sua arquitetura. Mais informações podem ser obtidas em <https://www.kaggle.com/datasets/pytorch/resnet34>.

¹¹ *YOLO: you only look once*, é uma arquitetura de *CNN* pré-treinada que consegue detectar e classificar imagens de forma muito rápida porque não faz em 2 estágios a tarefa, utiliza mascaras pré determinadas para a detecção, de forma a reduzir o problema. Ela tem como objetivo detectar e classificar as imagens praticamente em tempo real.

¹² *XGBoost* é um algoritmo de aprendizado de máquina, que não utiliza rede neural profunda. Mais informações em <https://xgboost.readthedocs.io/en/stable/>.

¹³ *Labelling*, que quer dizer rotular, é a técnica de se escolher a área da imagem onde está presente determinado padrão e colocar uma classe correspondente a ela, para ser possível a realização do treinamento supervisionado em seguida.

Software, para poder classificar os *candlesticks* antes de treinar o modelo *YOLO*.

Em 2022, (CHEN; TSAI, 2022) utilizam *YOLO* junto com sua técnica de transformar o padrão *candlestick* em um conjunto de 4 imagens que o representa, (CHEN; TSAI, 2020), e obtém bons resultados, 88,35% de acurácia média na detecção em tempo real dos 8 padrões de *candlestick* propostos em 2020.

Similar ao que fazem em 2020, (CHEN et al., 2022) utilizam ataques adversários no modelo proposto em (CHEN; TSAI, 2022) para aumentar a explicabilidade deste.

(HUANG, 2022) cria um novo passo no *encoding* utilizando *GAF*, onde ele transforma em uma imagem *RGB*¹⁴, que consegue evitar certas redundâncias verificadas no *encoding GAF* simples.

Ele utiliza *data augmentation* nas séries temporais utilizando o modelo *ARIMA*¹⁵ junto com uma rede neural convolucional e depois, através de *YOLO* treinado com os dados *aumentados*, consegue fazer a classificação dos padrões de *candlestick*.

Em seu livro de janeiro de 2023, (KAABAR, 2023) apresenta, de forma detalhada e com o código em *Python*, como reconhecer os padrões clássicos de *candlestick* de forma totalmente algorítmica, e, por tanto, determinista.

Além disto, ele inclui alguns novos padrões modernos em que analisa serem úteis para a teoria dos *candlesticks*.

Ele analisa a força dos vários padrões de *candlestick* em diversos ativos e mercados e apresenta diversas estratégias automatizadas de *trading*, utilizando os modelos determinísticos de reconhecimento de padrões de *candles* e outros indicadores financeiros utilizados nas previsões de mercado.

No artigo de janeiro de 2023 de (CAGLIERO; FIOR; GARZA, 2023), os autores utilizam biblioteca¹⁶ implementada em *Python*, que também identifica os padrões de *candlestick* de forma determinística, com algoritmos, e a partir deste reconhecimento e agregando mais indicadores financeiros à análise, utilizam modelos clássicos de machine learning para prever se o mercado vai subir ou descer com as várias configurações de padrões de *candlestick* junto com indicadores financeiros técnicos.

3 Metodologia

Em primeiro lugar, foi criado um *notebook* no *Google Colab* para poder refazer, passo a passo, o que foi feito em (SHAH, 2020).

Foi escolhido este trabalho por apresentar um código didático, limpo e por se entender que um modelo que reconheça os padrões de alta e baixa, sem utilizar classes previamente criadas pela teoria dos *candles*, poderia ser interessante, já que a modelagem pode identificar mais classes e padrões do que o olho humano e sua teoria.

¹⁴ *RGB* quer dizer *Red, Green and Blue*, vermelho, verde e azul, e são os canais que uma imagem possui ao ser apresentada na tela de um computador. Para cada um dos canais, existe uma matriz com um número de 0 a 255 para os *pixels* que representam a imagem. Com isto, é possível a criação das cores no vídeo.

¹⁵ *ARIMA Autoregressive integrated moving average*, média móvel integrada auto regressiva é uma técnica para análise de séries temporais.

¹⁶ Biblioteca: *Python-based Technical Analysis Library*: <https://github.com/quantopian/ta-lib>

Já se tem documentado no levantamento bibliográfico, livro que aborda de forma determinística o problema.

(SHAH, 2020) utilizou conceitos como *data augmentation*, *transfer learning*, uma rede pré-treinada e conhecida, a *resnet34*, e não utilizou interpolações ou imagens derivadas daquela que um gráfico de *candlestick* em *2d* apresenta, apresentando à rede convolucional a imagem tal qual o humano vê, o que ajuda a exploração deste campo de novos possíveis padrões de *candlestick*.

Foram utilizadas, no primeiro momento, todas as parametrizações e ativos selecionados por (SHAH, 2020).

A janela que ele decidiu para criar os *candlesticks* foi de 15 dias, sendo cada *candle*, um dia.

Ele classificou os *candlesticks* em *Up* ou *Down*, caso nos próximos 5 pregões juntos, houvesse um aumento ou declínio superior a 3,70%.

Caso não houvesse movimento de alta ou baixa, o *script* não gerava o *candlestick*.

A ideia seria classificar futuros *candlesticks* em *Up* ou *Down* apenas se a porcentagem indicada pelo modelo para determinada classe atingisse um *threshold*¹⁷ alto, já que não havia, no treinamento, classe neutra.

Para aumentar a velocidade, foi contratado o pacote do *Google Colab Pro*, utilizado acelerador via *Tensor Processor Unit - TPU* e o aumento da memória RAM disponível para 32,5 GB. Para fins de registro à posteridade, o valor da assinatura mensal *PRO*, que dá direito a 100 unidades de processamento por mês, acesso a mais memória RAM que o normal, o padrão seria 12 GB, e possibilidade de utilização de *GPUs* e *TPUs* foi de R\$ 58,00.

Anteriormente, em testes, cada época estava demorando 12 minutos para rodar e o ambiente travava por falta de memória RAM, depois do upgrade, cada época é processada em um minuto e o ambiente chegou a consumir no máximo 25 GB de RAM, tendo memória de sobra.

Seria inviável realizar os experimentos sem o ambiente contratado da nuvem, com os recursos atuais disponíveis no equipamento local do autor. O preço pareceu justo pela entrega de performance conseguida.

epoch	train_loss	valid_loss	error_rate	time
0	1.387497	1.033692	0.493023	01:24
1	1.275627	1.050920	0.483721	01:04
2	1.096563	0.818640	0.460465	01:10
3	0.967336	0.793957	0.460465	01:08

Figura 7 – Treinamento inicial.

Primeiro, o modelo de (SHAH, 2020) foi treinado com 4 épocas e com a métrica de treinamento *error rate*.

Foi conseguida uma *error rate* de 46% no primeiro ciclo, melhor que os 48% conseguido por ele em sua publicação. A figura 7 ilustra este passo.

¹⁷ *Threshold* pode ser traduzido como "limiar".

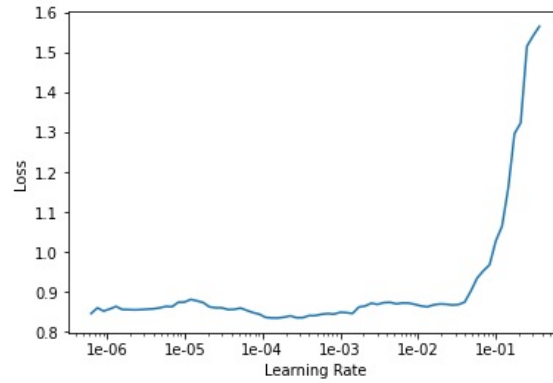


Figura 8 – Buscando *learning rate* ótima.

Foram descongelados os pesos da rede e então utilizada a procura da melhor *learning rate*¹⁸, se plotando o gráfico de *loss vs learning rate* (figura 8) e verificado que a taxa ideal estaria entre 1e-4 e 1e-3.

epoch	train_loss	valid_loss	error_rate	time
0	0.832339	0.824531	0.418605	01:28
1	0.824472	0.803320	0.483721	01:28
2	0.788142	0.770923	0.451163	01:30
3	0.727796	0.675664	0.374419	01:29
4	0.664248	0.669214	0.362791	01:29

Figura 9 – Segundo ciclo de treinamento.

Foi rodado, então, novo ciclo de treinamento, com este range de *learning rate* e com 5 épocas. Conseguiu-se uma *error rate* de 36% já neste segundo ciclo, superior a relatada no final do artigo do autor, de 38% (figura 9).

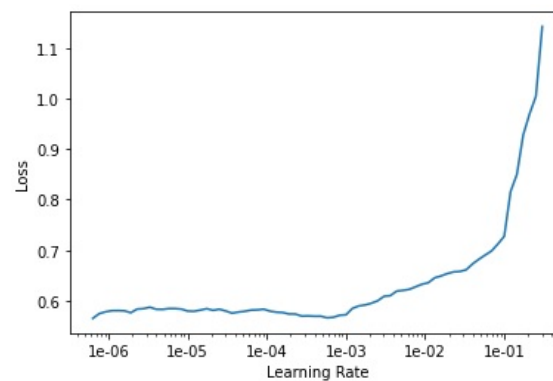


Figura 10 – Buscando *learning rate* ótima depois do segundo ciclo.

¹⁸ *Learning rate* é a taxa de aprendizado, parâmetro bastante importante em uma CNN, e vai determinar o tamanho do passo do algoritmo para encontrar o mínimo da função de custo que ele busca minimizar.

epoch	train_loss	valid_loss	error_rate	time
0	0.572904	0.648913	0.332558	01:29
1	0.548967	0.728142	0.362791	01:26
2	0.573034	0.796656	0.420930	01:18
3	0.564277	0.672823	0.341860	01:18
4	0.519322	0.738442	0.330233	01:20
5	0.480038	0.620905	0.295349	01:19
6	0.419189	0.561065	0.253488	01:18
7	0.356243	0.501201	0.202326	01:17
8	0.296227	0.523095	0.209302	01:19
9	0.252419	0.511463	0.197674	01:20

Figura 11 – Terceiro ciclo de treinamento.

Continuou-se o treinamento, repetindo o passo de identificar o melhor range para a *learning rate*, e então sendo rodado um ciclo de treinamento com 10 épocas, tal qual realizado pelo autor. A melhor *learning rate* ainda estaria entre $1e-4$ e $1e-3$. O resultado deste ciclo foi uma *error rate* de 19%, porém, um desequilíbrio entre a *train loss* e a *valid loss* 0,252419 vs 0,511463, indicando um possível *overfitting* (figuras 10 e 11).

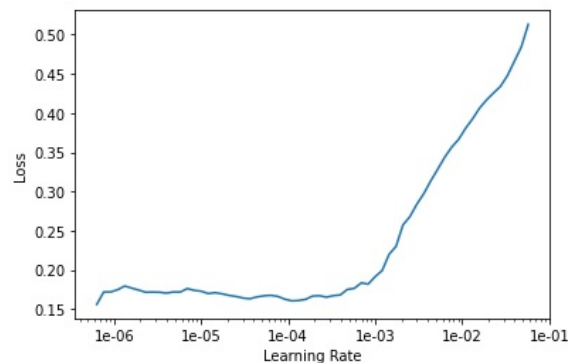


Figura 12 – Buscando *learning rate* ótima depois do terceiro ciclo.

epoch	train_loss	valid_loss	error_rate	time
0	0.170199	0.548781	0.218605	01:21
1	0.180584	0.602361	0.216279	01:25
2	0.180240	0.629916	0.227907	01:31
3	0.168816	0.609869	0.227907	01:25
4	0.148105	0.614111	0.225581	01:23

Figura 13 – Quarto ciclo de treinamento.

Continuou-se procurando a melhor *learning rate* novamente e então executado um

ciclo de treinamento com 5 épocas.

A *learning rate* utilizada foi de no máximo $1e-4$ e obteve-se uma *error rate* de 22,55%, maior do que a anterior e com o modelo mais *overfittado*, sendo *train loss* e a *valid loss* 0,148105 vs 0,614111 (figuras 12 e 13).

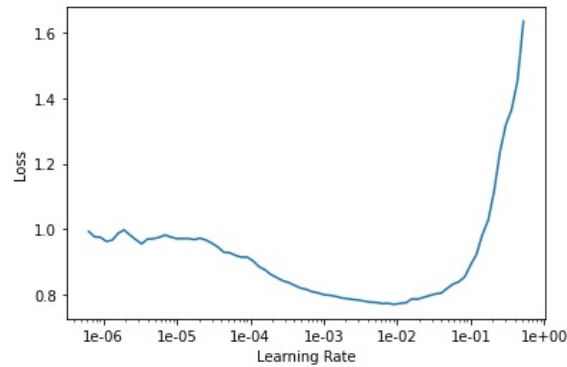


Figura 14 – Buscando *learning rate* ótima depois do quarto ciclo.

epoch	train_loss	valid_loss	error_rate	time
0	0.782334	1.786913	0.469767	03:00
1	0.873269	41.905727	0.458140	02:59
2	0.882582	0.845861	0.502326	02:52
3	0.850927	0.754046	0.500000	02:53
4	0.820880	0.689208	0.483721	02:51

Figura 15 – Quinto ciclo de treinamento.

Tal qual (SHAH, 2020) realizou, foi salva uma primeira versão do modelo e então rodado um novo *data augmentation* para começar novo ciclo de treinamento com 5 épocas e *learning rate* descoberta em $1e-3, 1e-2$, conseguindo-se uma *error rate* de 48,37% , mas com um equilíbrio maior entre *train loss* e a *valid loss* 0,820880 vs 0,689208 (figuras 14 e 15).

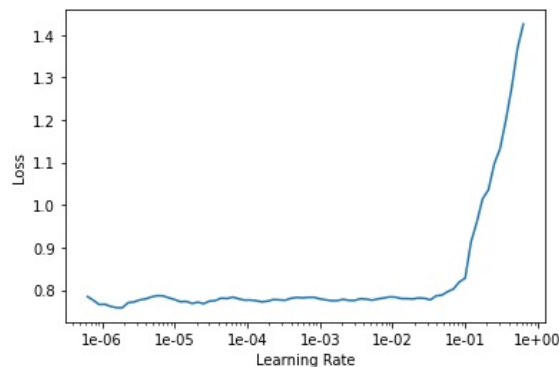


Figura 16 – Buscando *learning rate* ótima depois do quinto ciclo.

epoch	train_loss	valid_loss	error_rate	time
0	0.770445	0.687980	0.453488	02:49
1	0.755654	0.686953	0.453488	02:52
2	0.746153	0.687028	0.446512	02:51
3	0.753909	0.686692	0.462791	02:50
4	0.762665	0.687047	0.467442	02:51

Figura 17 – Sexto ciclo de treinamento.

Mais um ciclo foi rodado com 5 épocas e *learning rate* no intervalo de $1e-6$ a $1e-5$, obtendo-se o resultado de *error rate* de 46%, superior ao que o autor havia chegado em sua publicação, de 38% (figuras 16 e 17). O modelo foi salvo novamente. As previsões foram realizadas utilizando-se o *threshold* de 75% para poder classificar em uma das classes.

A classificação ficou zerada com este *threshold*, diferente do que o autor apresentou.

Diminuiu-se o *threshold* para 50% (foi se diminuindo de 5 em 5 pontos percentuais até haver classificações).

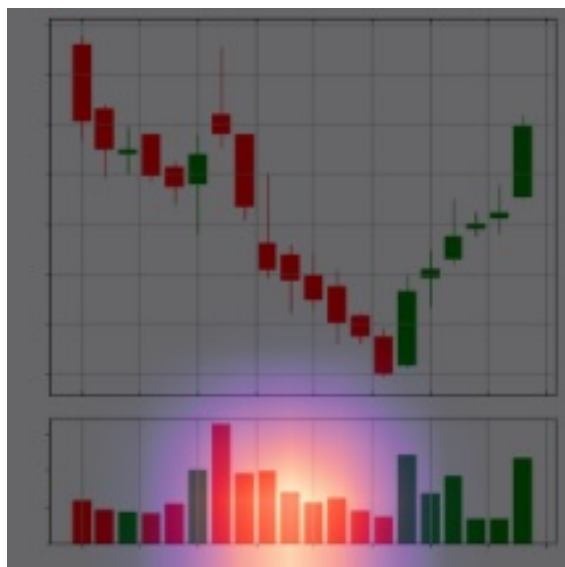


Figura 18 – Modelo se baseando no volume de *trading* para a classificação.

Verificou-se que o modelo aprendeu a classificar entre padrões *Up* e *Down* de forma pouco assertiva tendo como pontos principais para determinação da classe, não o padrão de *candlestick* observado, mas sim, o padrão de volume e suas cores observadas.

A figura 18 apresenta esta situação, iluminando quais os *pixels* que mais ativam a rede para a realização da classificação. Foram gastas 7 unidades de processamento do *Google Colab Pro* neste experimento.

Foi repetido o experimento apenas fazendo o primeiro *data augmentation* e colocando-se 20 épocas em um ciclo.

Conseguiu-se *error rate* de 40,93%, com equilíbrio entre validação e teste.

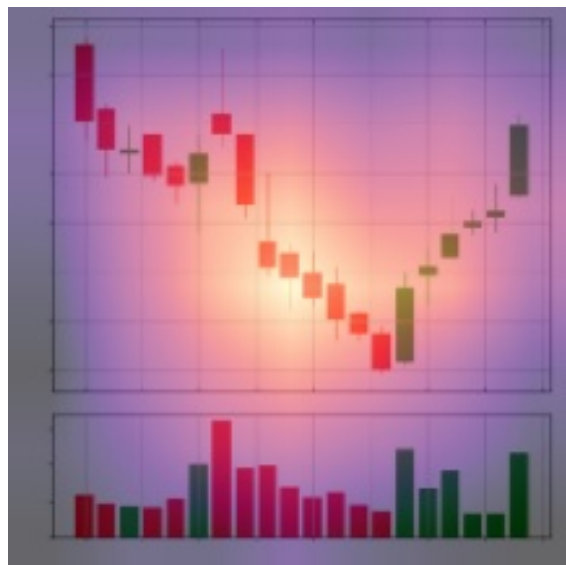


Figura 19 – Modelo reconhece padrão de *candlestick* de alta.

Percebeu-se que na época 16 a taxa de erro era de 39,77% com equilíbrio maior entre validação e teste.

Este resultado ficou próximo do mencionado no artigo original, sendo possível utilizar o *threshold* de 75% e funcionando o *heatmap* proposto pelo autor para mostrar que o padrão de *candle* foi aprendido(figura 19). Neste experimento foi gasta apenas 1 unidade de processamento.

A ideia para o próximo experimento foi executar o mesmo que o anterior, porém, com os dados de todas ações que compõem o índice Bovespa em dezembro de 2022.

A janela de 15 dias para a criação dos vários *candlesticks* foi revista, alterada para 5, visto que o maior valor em dias que compõe os padrões na bibliografia é de 5 pregões.

Além disto, foram consideradas as tendências de alta ou queda, quando nos 5 primeiros dias depois do surgimento do padrão ocorressem aumento ou queda acumulada maior ou igual a 15%, devido à alta volatilidade do mercado acionário brasileiro.

Foi coletado o período de 2022, garantindo informações de todas as ações que constam no índice e se esperando uma boa amostra de *candlesticks* de altas e baixas. Os resultados foram semelhantes aos conseguidos no experimento 2.

Em um quarto experimento, tentou-se aumentar o tamanho da base de dados, considerando a magnitude de 10% para classificação de altas ou baixas dos ativos que compõe o *ibovespa* em 2022.

Tentou-se utilizar magnitude de 5%, porém, excedeu-se a capacidade de memória RAM disponível no ambiente pela quantidade de casos encontrados, que gerou demasiada quantidade de imagens.

Investiu-se 10 unidades de processamento e 2 horas de espera na tentativa de magnitude 5%.

O resultado com a magnitude 10% foi muito próximo do conseguido com 15% e também com o conseguido na base original, com os ativos escolhidos por (SHAH, 2020),

mesmo tendo-se rodado 40 épocas em 2 ciclos de treinamentos, a *error rate* ficou em torno de 42%. O modelo *iluminou* o padrão de *candlestick*, porém é pouco assertivo também.

Foi tentado, em uma variação do experimento anterior, utilizar-se de *dropout*¹⁹ e de um treinamento mais lento, com paradas para utilizar-se da função de descoberta das melhores *learning rates* em cada estágio do treinamento, mas obteve-se o mesmo resultado.

Em um quinto experimento, utilizou-se a rede neural de referência *VGG19*²⁰ com *batch normalization*²¹ e *dropout* de 80%.

A rede foi treinada de forma lenta, com busca pela melhor *learning rate* a cada 10 épocas, sendo que o primeiro ciclo possuiu 16 épocas.

Houveram 5 ciclos de treinamento e cada um demorou, em média, uma hora. Foram consumidas 30 unidades de processamento do *Google Colab*.

Chegou-se a uma *error rate* de **32%**, desempenho superior à *error rate* do artigo de (SHAH, 2020), de 38%.

O modelo conseguiu iluminar a área esperada do padrão de *candle*.

Todos códigos utilizados no trabalho estão no endereço:

<https://github.com/gabrielfea/candlestick>

4 Resultados

Aplicando-se a metodologia descrita, conseguiu-se resultado superior ao de (SHAH, 2020), ele teve uma taxa de acerto de 62%, enquanto neste trabalho obteve-se 68% e utilizando os ativos do índice Bovespa.

A matriz de confusão e um exemplo de uma instância de *candlestick* da classe *Up* corretamente prevista com *threshold* de 0,75 e com a região correta ativada é vista na figura 20.

O modelo atingiu acurácia de 65,43% para a classe *Down* e 70,32% para a classe *Up*, tendo acurácia média de 67,88%.

5 Discussão

Foi verificado, através do trabalho, que a utilização do modelo proposto por (SHAH, 2020) com base de dados maior e com a rede de referência *vgg19* conseguiu superar os resultados por ele alcançados.

Mesmo superando o resultado de (SHAH, 2020), 68% de assertividade não é um resultado excepcional, visto que outros trabalhos conseguiram resultados muito superiores utilizando-se de técnicas de *encoding*, conforme a seção de trabalhos relacionados apresenta.

¹⁹ *Dropout* é uma técnica de regularização dos dados que visa evitar o overfitting durante o treinamento de uma *CNN*.

²⁰ *VGG19* é uma rede neural de referência tal qual existe a *RESNET34*. Documentação sobre *VGG19* pode ser obtida através do link: <https://www.mathworks.com/help/deeplearning/ref/vgg19.html>.

²¹ *Batch normalization* é uma técnica de regularização no treinamento que visa suavizar o aprendizado e evitar o *overfitting* do modelo, que ocorre quando ele é bom para os dados de teste e ruim na generalização, quando da validação.

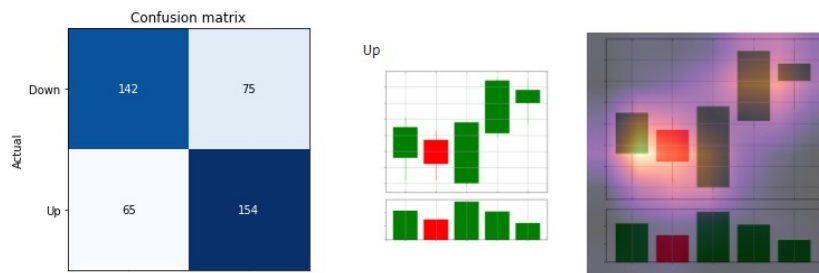


Figura 20 – Conforme visto na matriz de confusão, que é outra forma de apresentar a taxa de acerto de 68%, o modelo teve uma acurácia para a classe *Down* de 65,43% e para a classe *Up*, de 70,32%, totalizando uma acurácia média de 67,88%. Na imagem do meio é vista uma amostra de instância que é prevista para classe *Up* corretamente, utilizando-se de um *threshold* de 0,75. Para identificar se o modelo está se baseando realmente na imagem do padrão de *candlestick*, é iluminada a parte de interesse do modelo, que bate exatamente com o padrão de *candle*.

A *resnet34* e a *vgg19* podem não ter as *features* necessárias em suas estruturas para a melhor distinção dos padrões de *candle* nas classes propostas, sem *encoding*: *Up* e *Down*.

Utilizar o *encoding* proposto por (CHEN; TSAI, 2020) parece facilitar o reconhecimento dos padrões para as redes de referência e ser o caminho mais assertivo para tal intenta.

A utilização de ambiente com recursos superiores com a assinatura do *Google Colab Pro* foi fundamental para a possibilitação de realização dos experimentos.

O problema de pesquisa levantado desde o começo no artigo foi resolvido, conforme a revisão bibliográfica, de forma determinística, através de algoritmos, porém, toda a evolução levantada em como se utilizar redes neurais convolucionais para classificação dos padrões de *candlestick*, através dos *encodings* de séries temporais em imagens, foi bastante educativo a este pesquisador, pois foi possível entender e conhecer tais ferramentas matemáticas.

A tendência, agora, nesta seara, deve ser a utilização de técnicas de *Machine Learning* como redes adversárias ou outras tecnologias em prol da descoberta de novos padrões não triviais de *candlestick* não previstos na literatura, pois a máquina pode enxergar padrões com maior janela de tempo do que os humanos.

Para classificação nos padrões pré-existentes descritos por (NISON, 1994), o modelo determinístico sempre será mais assertivo.

Dos modelos probabilísticos levantados, que realizam a classificação dos padrões de *candles* automaticamente, a abordagem de (CHEN; TSAI, 2020) foi decisiva.

É elucidador entender que com o *encoding* utilizado por ele e lançando a mão de técnicas menos complexas de classificação, conforme verificado por (XU, 2021), se mostra mais assertivo o reconhecimento dos padrões.

Foi levantado que explorar a tecnologia *YOLO* é um bom caminho no aprimoramento da classificação probabilística dos *candles*.

A utilização do *machine learning* para prever o temperamento do mercado, tendo como base os sinais determinísticos passados pelos padrões automaticamente encontrados via algoritmos e os sinais de outros indicadores financeiros parece ser o caminho da utilização de inteligência artificial neste contexto.

6 Conclusão e Trabalhos Futuros

Junto com a classificação e identificação automática dos padrões de *candlestick* e com o cálculo de diversos indicadores técnico-analíticos sobre a evolução dos preços de um ativo, pode ser um trabalho interessante a utilização de processamento de linguagem natural em fóruns especializados da web, para geração de sinal ou indicador que represente a expectativa do mercado e seus *players* em relação a um ativo, a ser agregado em um modelo de previsão de tendências do mercado.

Um trabalho oportuno, no que tange classificação probabilística de *candlesticks*, seria a repetição do trabalho de (CHEN; TSAI, 2022) no contexto da bolsa de valores brasileira, para verificação da assertividade atingida.

O artigo foi construído pela curiosidade do autor em como detectar automaticamente os padrões de *candlestick* em gráficos, com a ajuda da máquina.

Esta dúvida foi respondida amplamente ao longo de sua elaboração e com isto, da-se realizado o objetivo de tal feita.

A utilização de uma rotulagem automática dos padrões de *candle*, através de algoritmo determinístico e posterior treinamento de *CNN* utilizando-se a mesma metodologia deste artigo pode aumentar a assertividade e utilidade do modelo.

Agradecimentos

Agradeço, imensamente, à minha mulher, por sempre apoiar meus estudos e, também, a todo corpo docente da PUC-RS Online, que fez esta especialização em Ciência de Dados e Inteligência Artificial ser tão interessante, completa e agradável.

Graças a esta pós graduação, consegui os conhecimentos necessários para trabalhar na área de soluções *analytics* da empresa na qual colaboro, sendo que desde a metade do curso já conquistei tal posição, já fazendo a entrega de um projeto piloto neste final de ano de 2022 com modelo de previsão 400% melhor do que o que existia anteriormente.

Este artigo encaminha a solução de uma vontade/anseio/busca que me inquietava há 4 anos, dede que criei o site www.dicasdabolsa.com.br, cursando a primeira pós graduação que fiz na PUC-RS Online, Tecnologia para Negócios: DS, AI e Big Data. Sempre quis ter esta funcionalidade no site.

Referências

BIROGUL, S.; TEMÜR, G.; KOSE, U. Yolo object recognition algorithm and “buy-sell decision” model over 2d candlestick charts. *IEEE access*, IEEE, v. 8, p. 91894–91915, 2020. Citado na página 7.

CAGLIERO, L.; FIOR, J.; GARZA, P. Shortlisting machine learning-based stock trading recommendations using candlestick pattern recognition. *Expert Systems with Applications*, Elsevier, p. 119493, 2023. Citado 2 vezes nas páginas 4 e 8.

CHEN, J.-H. et al. Explainable deep convolutional candlestick learner. *arXiv preprint arXiv:2001.02767*, 2020. Citado na página 7.

CHEN, J.-H.; TSAI, Y.-C. Encoding candlesticks as images for pattern classification using convolutional neural networks. *Financial Innovation*, SpringerOpen, v. 6, n. 1, p. 1–19, 2020. Citado 5 vezes nas páginas 3, 6, 7, 8 e 16.

CHEN, J.-H.; TSAI, Y.-C. Dynamic deep convolutional candlestick learner. *arXiv preprint arXiv:2201.08669*, 2022. Citado 2 vezes nas páginas 8 e 17.

CHEN, J.-H. et al. Explainable digital currency candlestick pattern ai learner. In: IEEE. *2022 14th International Conference on Knowledge and Smart Technology (KST)*. [S.l.], 2022. p. 91–96. Citado na página 8.

HATAMI, N.; GAVET, Y.; DEBAYLE, J. Classification of time-series images using deep convolutional neural networks. In: SPIE. *Tenth international conference on machine vision (ICMV 2017)*. [S.l.], 2018. v. 10696, p. 242–249. Citado 2 vezes nas páginas 5 e 6.

HUANG, H. Fast candlestick patterns detection with limited training samples using rgb gramian angular field and yolo-lite-v1. *Stanford University*, 2022. Citado na página 8.

KAABAR, S. *Mastering Financial Pattern Recognition*. [S.l.]: O'RELLY, 2023. Citado na página 8.

NISON, S. *Beyond candlesticks: New Japanese charting techniques revealed*. [S.l.]: John Wiley & Sons, 1994. v. 56. Citado 3 vezes nas páginas 2, 4 e 16.

PATHAK, L.; TAMBOLI, S.; TAMBOLI, A. Detection and analyzation of candlestick pattern using yolo object recognition. *International Research Journal of Engineering and Technology (IRJET)*, 2022. Citado na página 7.

SHAH, S. *Identifying Candlestick Patterns using Deep Learning*. Shaan Shah, 2020. Disponível em: <<https://towardsdatascience.com/identifying-candlestick-patterns-using-deep-learning-b7d706726874>>. Citado 6 vezes nas páginas 7, 8, 9, 12, 14 e 15.

WANG, Z.; OATES, T. Imaging time-series to improve classification and imputation. In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 2015. p. 3939–3945. Citado 2 vezes nas páginas 5 e 6.

XU, C. Image-based candlestick pattern classification with machine learning. In: *2021 6th International Conference on Machine Learning Technologies*. [S.l.: s.n.], 2021. p. 26–33. Citado 2 vezes nas páginas 7 e 16.