INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Como construir um modelo híbrido de previsão para o S&P500 usando um modelo VECM com um algoritmo LSTM?

Tiago Miguel Dias da Gama Lobo de Sousa Lopes

Mestrado em Economia Monetária e Financeira

Orientadora:
Doutora Diana E. Aldea Mendes, professora associada,
ISCTE - Instituo Universitário de Lisboa

junho, 2021

CIÊNCIAS SOCIAIS
E HUMANAS

Como construir um modelo híbrido de previsão para o S&P500 usando um modelo VECM com um algoritmo LSTM?

Tiago Miguel Dias da Gama Lobo de Sousa Lopes

Mestrado em Economia Monetária e Financeira

Orientadora:
Doutora Diana E. Aldea Mendes, professora associada,
ISCTE - Instituo Universitário de Lisboa

Junho, 2021

Ao amor da minha vida: a minha querida esposa

Agradecimento

Finalizar este marco da minha vida académica e profissional é em grande medida graças a todas as pessoas que me apoiaram e me permitiram alcançá-lo e que ao meu lado estiveram em todo este percurso.

Em primeiro lugar quero agradecer à minha companheira de todas as aventuras e desafios, a minha amada esposa Marta, que sem o seu apoio e incentivo ao longo de vários anos e, em especial agora nesta última etapa, não teria sido possível de alcançar com a mesma determinação, vontade, alegria, rapidez e sucesso. Apesar de neste ano difícil e marcante, não só pela crise pandémica, mas porque também perdeu uma das pessoas mais importantes da sua vida, o seu amado irmão Paulo, que também a mim muito triste me fez, fomos capazes de avançar com coragem, apesar da dor, e concluir esta etapa da melhor maneira possível.

Agradeço também o apoio dos meus professores que me instruíram e aconselharam ao longo deste mestrado, pela partilha da sua experiência e conhecimento, em particular, os professores Sérgio Lagoa, Emanuel Leão e Bhimjee Diptes.

Um especial agradecimento à minha estimada orientador Diana Mendes, que incansavelmente esteve disponível para me ajudar e responder às minhas questões, assim como, aconselhar neste percurso novo e desconhecido com vários desafios. Desde aprender a programar em Python, entender as complexidades relacionadas com a aprendizagem automatizada (*machine learning*), em especial o LSTM e as redes neuronais, a construir um modelo híbrido, a relacionar esses tópicos com a área do curso, entre muitos outros que foram sendo resolvidos.

A todos o meu

Muito Obrigado!

Resumo

A previsão de séries financeiras faz parte do processo de decisão das políticas monetárias por parte dos bancos centrais. Mendes, Ferreira e Mendes (2020) propõem um modelo híbrido que junta um VECM (modelo vetorial corretor de erro) com um algoritmo de aprendizagem profunda o LSTM (memória de longo curto-prazo) para uma previsão multivariada do índice acionista norte-americano S&P500, utilizando-se as séries do Nasdaq, Dow Jones e as taxas de juro dos bilhetes do tesouro americano a 3 meses no mercado secundário, com dados semanais, entre 19/04/2019 e 17/04/2020. Nesta dissertação, replicou-se esse artigo e construiu-se um modelo híbrido semelhante com a mesma finalidade e obteve-se um erro de previsão MAPE 86% inferior (4% *versus* 28%), mesmo incluindo a crise da COVID-19. Analisou-se o período sem crise e obteve-se um MAPE de 1.9%. Verificou-se que o vazamento de dados entre os períodos de teste e treino é um problema que prejudica os resultados. Comparou-se diferentes formas de construir o modelo híbrido variando o número de desfasamentos e de épocas de treino no LSTM, verificou-se o impacto de logaritmizar as séries, e comparou-se com modelos de referência (LSTM univariado/multivariado). Além disso, testou-se a causalidade à Granger entre os períodos com forte intervenção por parte da FED (décadas de 70 e 80, e crise da COVID-19 em fevereiro de 2020), concluindo-se que a variação das taxas de juro causam à Granger os retornos dos índices acionistas analisados, invertendo-se essa relação causal fora desses períodos.

Palavras-chave: previsão multivariada; VECM; LSTM; modelos híbridos; aprendizagem profunda; aprendizagem automatizada; S&P500; Dow Jones; Nasdaq; Bilhetes do Tesouro americano a 3 meses; política monetária; crise COVID-19; previsão índice acionista; montagem de modelos; causalidade à Granger; dados semanais.

Abstract

The forecasting of financial series is part of the decision-making process of monetary policies by central banks. Mendes, Ferreira and Mendes (2020) proposed a hybrid model that combines a VECM (Vector Error Correction Model) with a deep learning algorithm LSTM (Long Short-Term Memory) for a multivariate forecast of the U.S. stock index S&P500, using Nasdaq, Dow Jones and U.S. treasury bills for 3 months yields of the secondary market series, with weekly data, between 19/04/2019 and 17/04/2020. In this dissertation, this article was replicated, and a similar hybrid model was constructed with the same purpose and an 86% lower MAPE forecast error was obtained (4% *versus* 28%), even including the COVID-19 crisis. The time period without the crisis was analyzed and a MAPE of 1.9% was obtained. It was found that data leakage between the test and training periods is a problem that impairs the results. Different ways of constructing the hybrid model were compared by varying the number of lags and training epochs in LSTM, the impact of using the log-series was verified, and benchmarking with univariate and multivariate LSTM was made. In addition, granger causality was tested between the time periods with strong intervention by the FED (1970s and 1980s, and the COVID-19 crisis in February 2020) concluding that the changes in yields Granger cause the stock indices returns. In contrast, this causal relationship outside these time periods was the opposite, with the indices returns causing the changes in yields.

Keywords: multivariate forecasting; VECM; LSTM; hybrid models; deep learning; machine learning; S&P500; Dow Jones; Nasdaq; US Treasury Bills 3 months; monetary policy; COVID-19 crisis; stock index forecasting; model ensembling; Granger causality; weekly data.

Índice

Lista de figuras

Lista de Tabelas

Introdução

A pergunta geral de investigação: *Como construir um modelo híbrido de previsão para índices bolsistas usando métodos econométricos tradicionais juntamente com algoritmos de 'deep learning'?* visa proporcionar conhecimento agregado e exemplificativo de como se pode construir e usar modelos híbridos para previsão usando métodos econométricos tradicionais, tais como, modelos multivariados, VAR (autorregressivo vetorial), com modelos mais avançados e modernos de algoritmos de *Machine Learning* (ML, termo genérico para todos os algoritmos que melhoram automaticamente com a experiência), mais especificamente, *deep learning* (termo genérico para modelos de redes neuronais artificiais (ANN) mais complexos com múltiplas camadas internas), tal como, as redes neuronais recorrentes (RNN), em particular o LSTM (*Long-Short Term Memory*). Não se pretende abordar todos os métodos exaustivamente, mas antes, analisar como se pode juntar diferentes métodos num único modelo preditivo, de forma relativamente simples, e exemplificar com a criação de um modelo híbrido que inclui 1 modelo de cada categoria (1 modelo econométrico tradicional: VECM, e 1 algoritmo de *deep learning*: LSTM) para a previsão do preço do índice acionista S&P500. A escolha deste índice prende-se com o facto de ser um índice frequentemente usado como referência em artigos científicos com o objetivo de facilitar a comparação, além de ser o principal índice usado como referência global do mercado acionista norte americano na construção de portfolios de ações na indústria financeira.

As sub-perguntas de investigação consideradas são:

1.1. Quais as variáveis mais significativas na previsão do preço do S&P500?

Perceber *apriori*, i.e., antes de construir um modelo final, quais os fatores mais determinantes na previsão do preço do índice, permite usar um menor número de variáveis, excluindo as irrelevantes, além de se poupar tempo e recursos computacionais no desenvolvimento dos algoritmos de *deep learning* (Secção 1.4).

1.2. Qual a importância da estrutura (construção das camadas internas) de um algoritmo de *deep learning* (LSTM) nos resultados da previsão?

Analisou-se a literatura e procurou-se responder qual a importância das conexões, isto é, será que eliminar conexões não significativas (*pruning*) melhora ou mantem os resultados? Se a acurácia melhora ou se mantém, significa que encontrar uma estrutura certa é suficiente, e

permite poupar recursos computacionais e de tempo por se reduzir o número de conexões antes da fase de treino, sem prejuízo na qualidade da previsão (Secção 1.5).

1.3. Como unir os diferentes modelos ou metodologias usadas num único modelo de previsão?

Pesquisou-se que opções existem na montagem de modelos (Secção 1.6) e enquadrou-se nessas opções o tipo de montagem usada no modelo construído (Secção 2.6).

1.4. Pode um modelo híbrido ser capaz de produzir uma previsão que desafie a hipótese de eficiência dos mercados?

Segundo a hipótese de eficiência dos mercados, não é possível desenvolver uma estratégia de negociação, com ganhos anormais (i.e. superiores aos retornos do mercado alvo) consistentemente ao longo do tempo, pois o preço já tem descontado toda a informação disponível em cada momento, tornando o mercado eficiente. Assim, poder-se-á testar esta hipótese, por ver se uma estratégia baseada no modelo híbrido conseguiria obter ganhos anormais. Pesquisou-se na literatura por exemplos que testaram esta hipótese (Secção 1.7).

Os modelos VAR com um mecanismo de correção de erro (modelos VECM - *Vector Error Correction Model*), têm sido muito usados para a previsão de variáveis económicas e financeiras, com o fim de serem integradas em análises das políticas económicas e monetárias. Concernente à política monetária, são muito usados para prever e modelar o impacto de variações das taxas de juro em diversas variáveis tanto económicas como financeiras (Secção 1.2). Nesta dissertação, faz-se uma pequena análise ao efeito de causalidade entre a variação das taxas de juro e os retornos dos índices acionistas considerados, ao se aplicar o teste de causalidade à Granger (Secções 2.3, 3.1.2 e 3.2). Além disso, procurou-se também perceber se o uso de um algoritmo de *deep learning* (LSTM) no contexto de um modelo híbrido com um modelo VECM, ajuda ou não a melhorar a previsão final, ou se usado isoladamente se obtém melhores resultados, tanto usado para uma previsão com dados univariados como com dados multivariados (Secções 2.5, 3.1.4, 3.1.5, 3.2).

Os algoritmos de *deep learning* têm tido, nos últimos anos, um aumento crescente da sua popularidade e aplicações tão diversas como na tradução de línguas, na detenção de fraudes financeiras, ou na deteção de e-mails spam (Secção 1.3). São algoritmos mais indicados para categorizar variáveis ou para captar efeitos não lineares em séries temporais (p.e. financeiras). Também facilitam a automatização de processos e tarefas complexas num espaço mais curto de tempo. Pelo que é uma mais valia ter conhecimento sobre estes algoritmos e ser capaz de os usar.

A preferência pelo estudo de modelos híbridos vem da constatação do seu maior uso e sucesso em competições de previsão, como, por exemplo, na edição de 2018 da competição M4 (Secção 1.1),

o modelo vencedor foi um modelo híbrido. Esta preferência é devido a estes modelos apresentarem melhor acurácia na previsão de categorias e apresentarem maior flexibilidade na análise da informação, pelo facto de modelos individuais só captarem parte das características de uma série e, com diferentes modelos consegue-se captar várias características em simultâneo, além de que isso também permite a análise de mais informação com sistemas em paralelo (diferentes informações são analisadas ao mesmo tempo por diferentes modelos), poupando tempo.

A escolha por um índice bolsista teve como principais fatores preferenciais o maior número de variáveis disponíveis e de diferentes tipos, facilidade de obtenção de dados em quantidade, para diferentes periodicidades e por um longo período de tempo, serem mercados regulados e negociados em bolsa com dados uniformizados, as séries embora não sendo estacionárias, tendem a apresentar tendência de longo prazo, o que favorece o uso de um LSTM.

O principal objetivo desta dissertação é aprender a criar um modelo híbrido a partir de um modelo já existente na literatura e aplicá-lo na previsão do S&P500 e comparar os resultados obtidos, inclusive com modelos mais simples. Entender quais os fatores mais relevantes para a previsão do índice bolsista estudado e encontrar uma metodologia relativamente simples para criar modelos híbridos também são objetivos pretendidos. Paralelamente, perceber se existe vantagem em usar esse modelo num contexto de análise da política monetária, nomeadamente, no impacto da variação das taxas de juro.

O artigo escolhido foi o de Mendes, Ferreira e Mendes (2020) e o modelo híbrido usado é um modelo VECM para prever a tendência de longo prazo, e um LSTM para captar efeitos não lineares nos resíduos do modelo anterior, sendo a previsão final o resultado da adição dos outputs de cada modelo.

Segue-se o CAPÍTULO 1 com a revisão da literatura, onde se responde às sub-perguntas de investigação com base em artigos científicos e estudos existentes. No CAPÍTULO 2 é apresentada a metodologia da construção do modelo híbrido como aplicação de alguns conceitos referidos anteriormente. Posteriormente, no CAPÍTULO 3, apresentam-se e discutem-se os resultados obtidos. Por fim, conclui-se com um resumo das principais conclusões e sugere-se melhorias e investigação adicional das questões levantadas nesta dissertação.

CAPÍTULO 1

Revisão da Literatura

1.1. Artigos base

O artigo principal que se escolheu replicar e aplicar os conceitos e técnicas aprendidas ao dar resposta à pergunta de investigação é o de Mendes, Ferreira e Mendes (2020). Este artigo analisa a dinâmica do preço do maior índice acionista americano o S&P500 (SPX) e faz uma previsão do seu preço no longo-prazo (7 anos). É usado um modelo VECM (*Vector Error Correction Model*) para captar a tendência de longo prazo, e um algoritmo de *deep learning*, um LSTM (*Long-Short Term Memory*), para captar efeitos não lineares presentes nos resíduos do modelo anterior. Além do S&P500, foram usadas as séries do preço dos índices Dow Jones 30 (DJI) e NASDAQ 100 (NDQ), juntamente com a taxa de juro dos bilhetes do tesouro americano (*treasury bills*) a 3 meses (TB3M). Os dados são semanais, sendo o período de treino entre 05/02/1971 e 12/04/2013 e o período de teste entre 19/04/2013 e 17/04/2020, sendo o período entre 19/04/2013 a 12/04/2019 usado para validação de Hiper parâmetros na fase de treino do LSTM. O período de teste abrangeu a queda de março no preço dos índices causada pela crise pandémica associada ao SARS-Cov-2, pelo que o erro de previsão, medido pelo MAPE (*Mean Absolute Percentage Error*), foi de 28.19%, um valor bastante elevado. No artigo são propostas melhorias, tais como, ajustar o período a prever, usar dados diários ou mudar a filosofia de treino do LSTM.

A importância de se usar modelos híbridos ou combinados na previsão é destacada por Makridakis, Spiliotis e Assimakopoulos (2020) por salientar que é das poucas coisas onde há consenso na área da previsão por se obterem menores erros de previsão e por consequência melhor acurácia. Este artigo consiste na apresentação dos resultados da competição M4 a nível mundial, bem como, das regras gerais, conclusões e comparações com as edições anteriores. A competição consistiu em prever 100 mil séries retiradas da base de dados ForeDeCk (900 mil séries), com dados micro e macroeconómicos, financeiros, demográficos, industriais, e com diferentes periodicidades: horária, diária, semanal, mensal, trimestral e anual. A seriação da performance dos modelos foi baseada numa média ponderada de duas medidas de erro de previsão (após relativizar os erros com os de um passeio aleatório): MAPE e MASE. Participaram 61 modelos, sendo que o vencedor foi um modelo híbrido (ML com alisamento exponencial), e apenas 17 modelos tiveram melhor acurácia que o *benchmark* ARIMA, sendo 13 modelos combinados (mais que um modelo usado para a previsão final) e 4 modelos puramente estatísticos. Todos os modelos da competição previram melhor que os *benchmark* de *Machine Learning* (MLP, RNN), mostrando que estes modelos isoladamente têm uma performance

fraca, mas em combinação com outros modelos, especialmente estatísticos, têm bons resultados, o que serve de motivação para a construção de modelos híbridos. Como sugestão de melhorias o artigo indica a necessidade de se perceber melhor qual a forma mais eficaz de escolher os métodos a combinar e como otimizar os pesos destes na previsão final.

1.2. Modelos VECM

Os modelos VECM são um dos modelos econométricos mais populares (Mills & Markellos, 2008, citado em Mendes, Ferreira e Mendes, 2020). A origem e interesse por estes modelos é bem descrita por Mendes, Ferreira e Mendes (2020):

> "*Esses modelos derivam da ideia de que séries temporais comuns podem ter uma dependência de longo prazo numa tendência estocástica. Um grande avanço nesta área veio do teorema da representação de Granger (Engle & Granger, 1987), que mostra precisamente que uma relação de cointegração pode ser representada pelo modelo de correção de erros (ECM)*."

Vários modelos VECM foram usados para modelar as relações existentes entre muitas variáveis económicas e financeiras, como, por exemplo, na análise do efeito da política monetária no crescimento da economia do Reino Unido (Agbonlahor, 2014), ou no nível de preços da Zona Euro (Holtemoller, 2004), na modelação das principais variáveis macroeconómicas da economia dos EUA (Anderson, Hoffman e Rasche, 2002), na previsão das taxas de juro de longo prazo do Brasil (Lima, Luduvice e Tabak, 2006), ou na previsão de taxas de câmbio (Zhang, Lowinger e Tang, 2007).

Segundo Kliensen (1999), existem dois tipos de modelos usados pelo banco central americano (FED) usados no processo de políticas monetárias: modelos estruturais[1] e modelos de previsão. Dentro dos modelos de previsão, também conhecidos como modelos de séries temporais, baseiam-se na existência de correlação estatística entre observações correntes e passadas das variáveis em estudo. Este autor confirma que o modelo mais popular é o VAR/VECM e compara-os com os modelos estruturais, que ao contrário destes, os VAR consideram todas as variáveis como endógenas, isto é, são simultaneamente determinadas e, portanto, têm uma equação para cada variável no modelo. Ou seja, não assumem uma relação comportamental única como uma equação de consumo, investimento ou procura de moeda, como é assumida por modelos estruturais. Em termos de capacidade de previsão, os modelos de previsão são geralmente melhores do que os modelos estruturais. No entanto, o autor afirma que estes não são úteis para avaliar alternativas de política monetária, como por

[1] Modelos de equilíbrio geral da economia, tipicamente novo-Keynesianos. O modelo atualmente usado é o FRB/US. Bernanke, (2020) usou apenas o modelo da FED, o FRB/US, para simular o impacto das políticas monetárias não convencionais, especialmente o *Quantitative Easing* (QE), e *Forward-looking*, na economia norte-americana, confirmando a relevância ainda atual desse modelo.

exemplo, analisar o que aconteceria ao crescimento do PIB real e à inflação se a taxa dos fundos federais (taxa de juro de referência) variasse 25 pontos base.

Então como pode um modelo de previsão ser usado ou útil no processo de definição da política monetária? Segundo Reifschneider et al. (1997) (citado por Kliensen, 1999), estes modelos são usados nesse processo de três formas: primeiro, é criada uma previsão base de como se espera que a economia se comporte de 1 a 2 anos (previsões trimestrais, sendo as principais variáveis previstas o PIB real e a inflação) mantendo a mesma taxa de juro de referência. Esta previsão é atualizada todos os meses entre reuniões da FED. Segundo, são feitos pressupostos sobre variáveis exógenas, por exemplo, como variará o preço do petróleo, onde também são usados modelos de previsão. Nesta fase existe uma discussão e afinação das previsões entre os economistas até se chegar a uma convergência/consenso e produzir a previsão final. Terceiro, são feitas análises de sensibilidade do impacto de variações da taxa de juro de referência, do preço de índices acionistas, de impostos, nas principais variáveis objetivo. Por fim, é apresentada à comissão da FED o cenário base (da primeira fase), sem alteração de políticas, e os cenários simulados (da segunda e terceira fases).

Ben Bernanke, presidente da FED entre 2006 a 2014, no seu artigo (Bernanke, Boivin e Eliasz, 2004) confirma que os modelos VAR são muito usados para estudar os efeitos das inovações de política monetária na economia, e salienta a sua importância na determinação dessas políticas, além de afirmar que geralmente dão resultados empíricos plausíveis das dinâmicas existentes nas principais variáveis económicas. Apresenta algumas limitações destes modelos, especificamente concernentes com o limite na quantidade de informação, *aka* variáveis e número de desfasamentos, que é possível de incluir e estimar, mesmo que na realidade os bancos centrais usem centenas de séries e variáveis na sua análise global. Por isso, propõe o uso de modelos FAVAR (*Factor-Augmented VAR*) onde junta os novos conhecimentos criados na área da análise de fatores preditivos num contexto de grandes dados, com as vantagens dos modelos VAR. Ou seja, a investigação em modelos dinâmicos com fatores concluiu que a informação contida num grande número de séries pode ser resumida por um reduzido número de fatores ou índices. Assim, propõe usar esses fatores como inputs para o VAR e, com isso, evita-se a necessidade de incluir demasiadas séries (o que tem um limite reduzido, devido ao aumento dos graus de liberdade ser limitado pelo número de observações). O resultado foi que o modelo FAVAR conseguiu captar a informação relevante que permitiu estimar os efeitos de variações na política monetária numa gama alargada de variáveis económicas. Este artigo evidencia o uso de modelos VAR e tentativas de os melhorar na sua capacidade preditiva.

Assim, procura-se nesta dissertação, obter um modelo de previsão, nomeadamente um modelo híbrido usando um LSTM, que melhore a capacidade preditiva de um modelo VECM, o que ajuda na modelação de valores futuros das variáveis em estudo, e não construir um modelo para avaliar alternativas de política monetária. De notar que na segunda fase do processo de definição da política

monetária, descrita anteriormente, a convergência/consenso necessária para se chegar a uma previsão final, é semelhante ao que acontece na construção de um modelo híbrido (automatizado), onde vários modelos são analisados, tornando-se necessário determinar uma forma de se usar toda essa informação para se chegar a uma previsão final, podem ser escolhidos os melhores modelos e rejeitados os restantes, ou ser feita uma ponderação das várias previsões individuais, ou usar o output de um modelo como input para outro (ver Secção 1.6). É neste contexto de montagem de modelos, VECM com LSTM, e obtenção da melhor previsão, para o índice acionista S&P500, que se insere esta dissertação.

1.3. Enquadramento da investigação realizada (2005 a 2019)

De forma a se ter uma ideia da investigação feita e existente sobre modelos de previsão que usam modelos híbridos de ML (*Machine Learning*), Sezer, Gudelek e Ozbayoglu (2020), bem como Kumar, Sarangi e Verma (2021) fazem uma revisão sistemática da literatura.

Em Sezer, Gudelek e Ozbayoglu (2020), apenas 16 artigos dos 152 analisados incluem em simultâneo a previsão do S&P 500, ou de ações constituintes, com modelos que incluem ou são baseados em LSTM, conforme Tabela 1. Um deles apenas prevê a volatilidade e outro usa dados de *order book* que sai fora do âmbito desta dissertação. Dos 11 artigos que usam o índice S&P500 com um modelo LSTM para previsão, apenas 8 têm potencial para serem comparados com o modelo híbrido a ser construído (dos 11, 1 é apenas sobre volatilidade, nos outros 2 o índice é apenas usado para ajudar a prever o preço de ações individuais). De referir que este artigo apenas cobre artigos entre 2005 e 2019, sendo que em 2020/21 saíram mais artigos, onde alguns deles serão analisados posteriormente.

Desde 2015 que começou um aumento do número de artigos publicados que usam algoritmos de ML na previsão de índices bolsistas, tendo um pico em 2017 (linha laranja na Figura 1). Significa que é agora uma área em franca expansão com muito potencial por desenvolver. O algoritmo mais usado é o RNN (*recurrent neural network*) do qual o LSTM faz parte, sendo o mais usado dentro dos RNN (Figura 2). De notar que o LSTM é usado para captar relações de longo prazo e, por isso, mais usado em séries que apresentam tendência. Na tabela acima, não são usados modelos LSTM para séries de Forex e muito pouco para *commodities* pois estas têm uma forte reversão à média, com tendências laterais, mas são usados na previsão de índices bolsistas, ações e cryptomoedas que apresentam uma forte tendência ascendente de longo prazo.

Tabela 1. Contagem do número de artigos citados em Sezer, Gudelek e Ozbayoglu (2020) por tema que incluem a série S&P500 e o algoritmo de Deep Learning LSTM

Classe	Tema	# artigos	Inclui					
			S&P 500 index	S&P 500 stocks	LSTM c/ outros	Só LSTM	S&P 500 c/ LSTM	Index c/ LSTM
Stock price	Stock price forecasting using only raw time series data.	16		5	6	2	2	
	Stock price forecasting using various data.	18		1	6	3		
	Stock price forecasting using text mining techniques for feature extraction.	9	2	4	3	1		
	Subtotal	*43*	*2*	*10*	*15*	*6*	*2*	*0*
Index	Index forecasting using only raw time series data.	21	9		3	6	6	6
	Index forecasting using various data.	13	5	1	6		1	1
	Subtotal	*34*	*14*	*1*	*9*	*6*	*7*	*7*
Commodities	Commodity price forecasting.	7	2		1			
Volatility	Volatility forecasting.	7	1		1	2	1	1
Forex	Forex price forecasting.	18	4					
Cryptocurrency	Cryptocurrency price prediction.	2			2			
Trend	Trend forecasting using only raw time series data.	11	5	3	1	1	2	1
	Trend forecasting using technical indicators & price data & fundamental data.	9	2		1	2		
	Trend forecasting using text mining techniques.	15	4	4	2	2	3	2
	Trend forecasting using various data.	6		1	1	2	1	
	Subtotal	*41*	*11*	*8*	*5*	*7*	*6*	*3*
	TOTAL	152	34 22%	19 13%	33 22%	21 14%	16 11%	11 7%

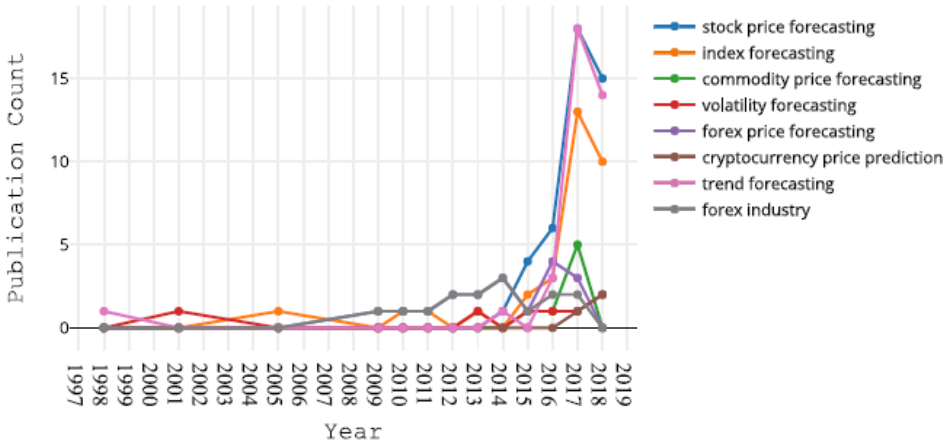

Figura 1. Contagem do número de publicações por temas (Sezer, Gudelek e Ozbayoglu, 2020)

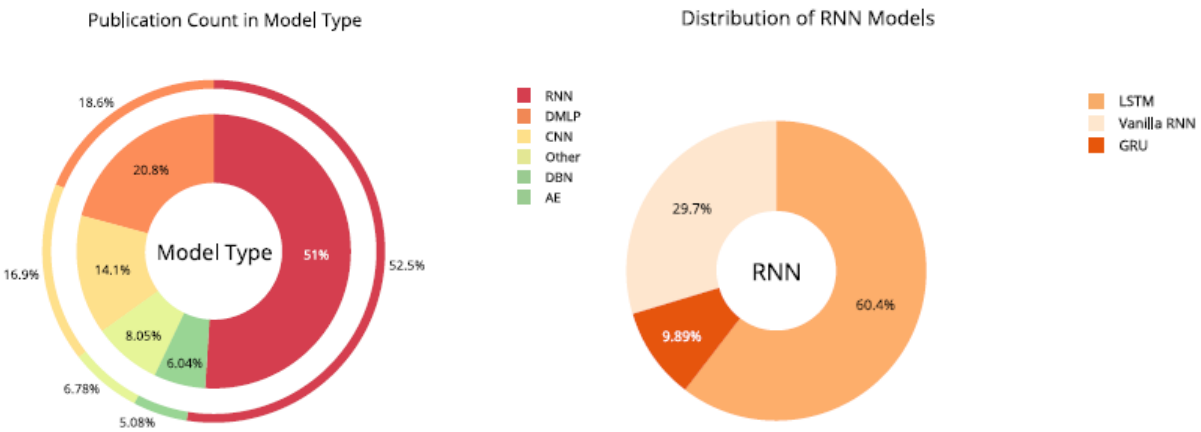

Figura 2. Algoritmos de *Machine Learning* mais usados (esquerda) e dentro dos RNN (direita) (Sezer, Gudelek e Ozbayoglu, 2020)

Kumar, Sarangi e Verma (2021), ilustra o processo de se construir um modelo de previsão do preço de ações, na figura abaixo, que inclui 9 etapas, desde a recolha dos dados, e sua preparação, até à aplicação de técnicas de ML (Figura 3). Segundo este, as redes neuronais no seu conjunto (NN, ANN, CNN, RNN) são os algoritmos de ML mais usados, seguidos do SVM (*Support Vector Machine*). O parâmetro de performance mais usado é a acurácia (*Accuracy*, da matriz de confiança nos modelos de classificação) seguido do MSE e RMSE. Os jornais mais usados para publicações relacionadas são o IEEE e o Springer (Figura 4).

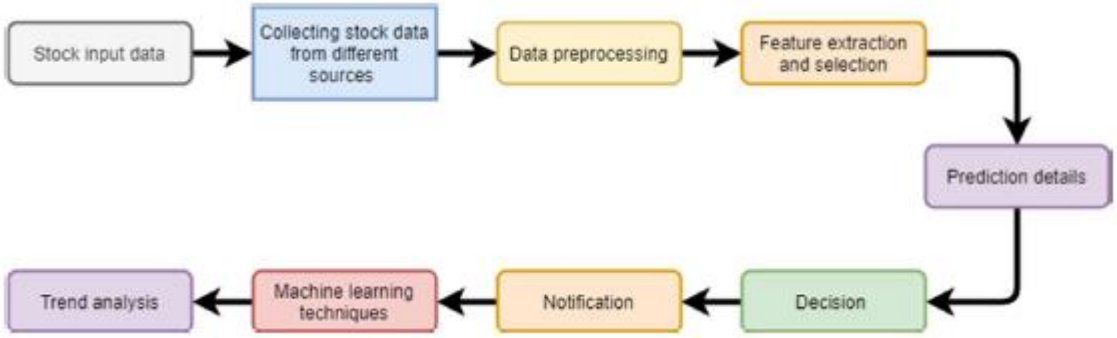

Figura 3. Processo de previsão de séries do mercado acionista (Kumar, Sarangi e Verma, 2021)

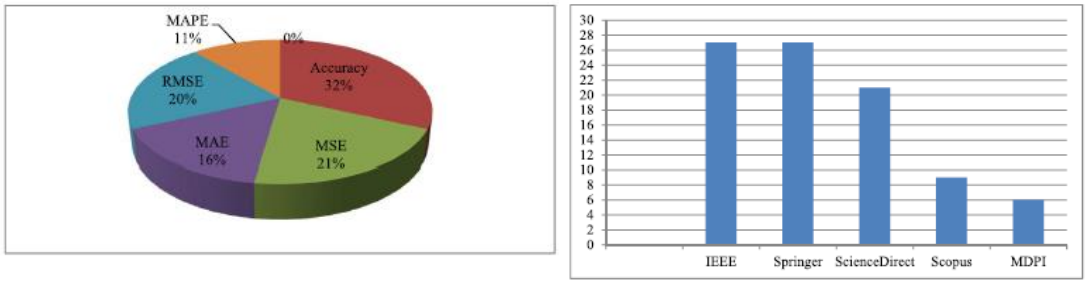

Figura 4. Medidas de performance mais usadas (esquerda) e jornais com mais publicações da área (direita) (Kumar, Sarangi e Verma, 2021)

1.4. Seleção de variáveis

Para responder à sub-pergunta de investigação 1.1 (*Quais as variáveis mais significativas na previsão do preço do S&P500?*) foram encontrados artigos que chegaram a algumas conclusões semelhantes, especialmente na importância da volatilidade como fator mais relevante na previsão de ações ou índices acionistas, mesmo quando obtida por diferentes tipos de transformação ou indicador usado para capturar a sua dinâmica temporal.

Nevasalmi (2020) introduz um novo modelo de classificação baseado na distribuição multinomial, afirmando ser uma novidade por não existirem estudos prévios que a usassem para os retornos de ações, e que dá mais ênfase a prever variações absolutas grandes em vez de variações pequenas ou ruído à volta de zero. O intervalo de tempo usado foi entre 12/02/1990 e 05/10/2018, e a série financeira é o índice S&P500. Foram usados 5 algoritmos de ML: *k-Nearest Neighbor classifier* (KNN),

Gradient Boosting using J-terminal node regression trees (GB), *Random Forest* (RF), *Neural Networks*, *Support Vector Machines* (SVM). O método de montagem com melhor performance foi o GB, e o fator mais importante para prever o retorno do dia seguinte foi o índice de volatilidade implícita do S&P500 baseado no mercado de opções, o VIX. Foram selecionados os 6 tipos de fatores considerados mais relevantes todos com 3 desfasamentos, escolhidos de entre os melhores resultados do GB, RF e PCA (*Principal Component Analysis*). Além do VIX, os restantes 5 foram: *spread* entre as *yields* empresariais com *rating* AAA (da Moody's, EUA) e os bilhetes do tesouro (*US Treasury Bills*) a 10 anos, oscilador estocástico, Williams %R, indicador MACD (*Moving Average Convergence Divergence*), *spread* entre o preço máximo e mínimo do dia, retornos passados do índice S&P500 e do DAX.

Yang, Zhai e Tao (2020) usam um algoritmo CNN (*Convolution Neural Network*) para seleção de fatores e um LSTM para a previsão final. Pretendem prever a direção do preço do S&P500 usando como fatores outros índices acionistas como o NASDAQ e o Dow Jones, além de uma série de indicadores técnicos e suas fórmulas. O intervalo de tempo usado foi entre 04/01/2010 e 29/12/2017, os dados foram retirados do Yahoo finance. O optimizador do LSTM foi o Adam. Conclui o artigo que o modelo híbrido (CNN + LSTM) é o que tem melhores resultados especialmente se os índices usados como fatores forem ordenados descendentemente consoante a sua correlação de Pearson e os indicadores técnicos forem transformados em sinais de tendência determinística antes de serem enviados para o CNN.

Dionisio et al (2011) apresentam a métrica informação mútua para medir a interdependência não linear entre índices acionistas e indicadores macroeconómicos e financeiros. A principal vantagem apontada é que esta métrica não necessita de pressupostos quanto à distribuição probabilística ou às especificações na modelação da dependência. E, portanto, esta métrica pode ser usada como fator ou em substituição do coeficiente de correlação de Pearson na ordenação dos fatores mais relevantes. Este artigo estuda a relação histórica entre o preço do PSI20 e variáveis macroeconómicas e financeiras. Conclui que as variáveis financeiras como a *dividend yield* e o rácio rendimento/preço (P/E), são mais relevantes que as variáveis macroeconómicas para explicar o retorno adicional em relação ao índice base, como também concluído por Fama e French (1993).

Isfan, Menezes e Mendes (2010) apresentam o coeficiente de Hurst como métrica útil para medir a memória de longo-prazo existente numa série financeira e, consequentemente, poder ser usado como uma estimativa da sua previsibilidade. Variando entre 0 e 1, um valor de 0.5 indica que a série é um passeio aleatório, valores acima indica que é uma série com memória longa, ou persistência, e valores abaixo uma série com tendência para regressão à média, ou anti persistência.

1.5. Optimização da estrutura das camadas internas de uma Rede Neuronal

Respondendo à sub-pergunta 1.2 (*Qual a importância da estrutura (construção das camadas internas) de um algoritmo de deep learning (LSTM) nos resultados da previsão?*), Lee, Ajanthan e Torr (2019) propõem uma metodologia de corte das camadas internas de uma rede neuronal, a que chamam de SNIP (*single-shot network pruning*). Apresentam uma nova abordagem que poda uma determinada rede uma vez na inicialização antes do treino. Para tal, introduziram um critério de saliência baseado na sensibilidade das conexões que identifica ligações estruturalmente importantes na rede para a tarefa dada. Isto elimina a necessidade de pré-treino e reduz a complexidade da poda, o que torna o modelo robusto às variações de arquitetura. Após a poda, a rede neuronal reduzida é treinada da forma padrão. Este método obtém redes extremamente reduzidas com praticamente a mesma acurácia que as versões de base em todas as arquiteturas testadas. São usados métodos de normalização da variância para inicializar os pesos, de modo a variância permanecer a mesma em toda a rede. Garantindo isso, mostraram empiricamente que a nova medida proposta de saliência calculada na inicialização é robusta às variações da arquitetura. Com esta metodologia de corte ou poda consegue-se reduzir o número de conexões, e por consequência, o número de pesos a estimar, na ordem dos 95% com pouca ou nenhuma perda de performance. Outra vantagem é que com esta poda da rede neuronal deixa de conseguir memorizar fatores gerados aleatoriamente como acontece com redes completas, evitando-se ter de reduzir o número de épocas (treinos sucessivos, iterações) de forma a evitar o efeito de *overfitting* como se pode ver na Figura 5.

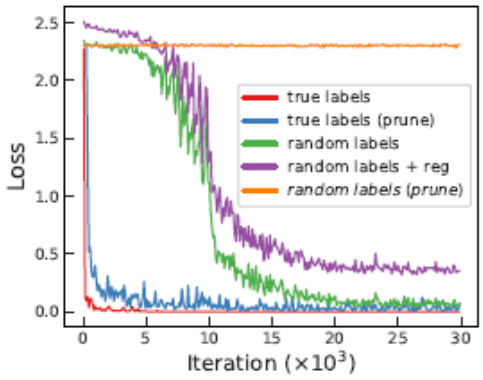

Figura 5. O Modelo reduzido com poda por SNIP não ajusta fatores aleatórios (Lee et al., 2019)

1.6. Métodos de montagem

Para dar resposta à sub-pergunta de investigação 1.3 (*Como unir os diferentes modelos ou metodologias usadas num único modelo de previsão?*), é necessário definir o que é um método de montagem e quais existem.

Segundo Pedamkar (2021) um método de montagem no âmbito de ML é definido como um sistema multimodal no qual diferentes classificadores ou algoritmos e técnicas são estrategicamente combinados num modelo preditivo final. O método de montagem também ajuda a reduzir a variação dos dados previstos, minimiza o enviesamento no modelo preditivo e classifica e prevê as estatísticas de problemas complexos com maior precisão. São enunciados e explicados os principais 4 tipos de montagem (sequencial, paralela, homogénea e heterogénea), bem como as principais 4 classificações

técnicas usadas (*Bagging, Boosting, Stacking* e *Random Forest* (RF)), cujas definições e explicações técnicas podem ser lidas nesse artigo. Com o mesmo intuito, o livro de Zhou (2012) também contém mais especificações e detalhes desses e de outros métodos de montagem, tais como, sistema de votação, *clustering*, sistemas de corte, classificadores dinâmicos, etc.

Em contraste, Kourentzes, Barrow e Petropoulos (2019) apresentam um sistema inovador que é um sistema intermédio entre selecionar apenas o melhor modelo previsional e usar uma combinação do output de todos os modelos na previsão final. Chamam a este modelo de *Pooling*, e é baseado numa heurística para selecionar automaticamente o número ótimo de modelos a usar. É uma técnica relativamente simples que pode ser usada para reduzir o número de modelos a usar na previsão final sem perda significativa de acurácia.

Smyl (2020) explica como foi feito o modelo vencedor da competição M4 pelo próprio criador. Consiste num modelo híbrido e não apenas numa montagem de um conjunto de modelos, como dito pelo autor, pela forma como os parâmetros dos 2 modelos usados são obtidos. Em vez de apenas montar os modelos e fazer a média da previsão de cada modelo, ou serem modelos sequenciais (o modelo seguinte usa o output do modelo anterior), como acontece com a maioria dos métodos de montagem, o autor esclarece que o seu modelo é verdadeiramente híbrido no sentido que os parâmetros de ambos os modelos são obtidos em simultâneo pelo mesmo método de estimação, o *Stochastic Gradient Descent* (com *backpropagation*). No âmbito desta dissertação, a intenção de construir modelos híbridos não é tão restritiva e inclui os modelos conjugados. Mas neste artigo e no artigo da competição M4, Makridakis, Spiliotis e Assimakopoulos (2020), é feita essa distinção. O primeiro modelo é um alisamento exponencial das séries (ES) e o segundo o LSTM numa estrutura comum (modelo híbrido). O objetivo do modelo ES é captar as principais características das séries individuais, nomeadamente, a sazonalidade e a tendência, de forma eficaz, e o LSTM captar as tendências não lineares e as relações entre as séries. Este modelo híbrido tem 3 elementos: (i) dessazonalização e normalização adaptativa, (ii) geração de previsões e (iii) montagem. Na parte da montagem, como os valores iniciais (obtidos aleatoriamente) dos pesos das conexões num LSTM influenciam a previsão final, foi criada uma montagem de vários desses modelos onde se obtém a média destes, de forma a melhorar a acurácia final. O intervalo de valores que foi determinado que maximiza a acurácia foi o de montar entre 6 e 9 desses modelos. Num nível de montagem superior, foi realizada uma montagem com modelos estimados paralelamente, cada um estimado apenas numa subamostra da população, chamada montagem de especialistas, onde apenas os N melhores modelos são usados. Uma metodologia de montagem mais simples, usada nos dados mensais e anuais, é similar ao *bagging*, onde em vez de se escolher o melhor modelo para cada série, estimam-se todos os modelos e no final é feita a média das previsões. Estes modelos LSTM diferem dos anteriores pois são

variações dos Hiper parâmetros dentro de uma mesma amostra em vez de variações dos pesos das conexões com os mesmos Hiper parâmetros em subamostras diferentes.

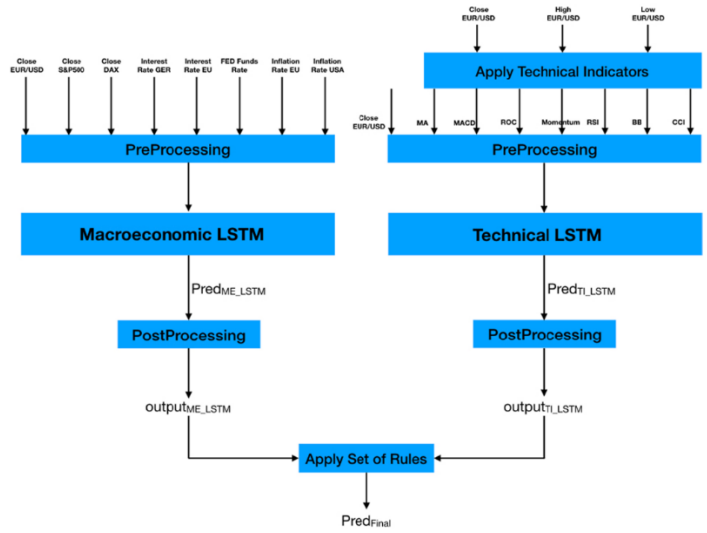

Figura 6. Modelo híbrido com um LSTM para fatores macroeconómicos (esquerda) e outro LSTM para fatores técnicos (direita) (Yildirim et al., 2021)

Yildirim, Toroslu e Fiore (2021) apresenta um modelo híbrido com uma estrutura onde separa os modelos LSTM de acordo com o tipo de fatores usados, isto é, um LSTM usa fatores macroeconómicos e o outro usa fatores técnicos (Figura 6). Apesar deste modelo híbrido ter sido aplicado a uma divisa (euro/dólar), a ideia de se usar diferentes algoritmos de acordo com o tipo de fatores pode ser replicada.

1.7. Estratégias de negociação

Obter uma resposta para a sub-pergunta de investigação 1.4 (*Pode um modelo híbrido ser capaz de produzir uma previsão que desafie a hipótese de eficiência dos mercados?*) implica testar a validade da hipótese dos mercados eficientes (EMH) (Fama, 1970).

Segundo Nevasalmi (2020), a validade da EMH é tipicamente testada numa simulação de negociação em condições reais. A capacidade de gerar lucros para além da estratégia passiva de apenas comprar e manter a posição, e levando em conta os custos de transação, é vista como uma violação do EMH. Todos os métodos de ML considerados neste artigo foram capazes de vencer a estratégia passiva mesmo após contabilizar o custo de transação de 0,1%. O modelo com melhor desempenho, o GB, produziu retornos 80% superiores à estratégia passiva (Figura 7). A previsibilidade dos modelos foi mais elevada quando houve maior volatilidade nos mercados, como nas alturas da crise financeira (2008) e da crise da dívida soberana europeia (2011), que, segundo este artigo, está de acordo com a literatura recente.

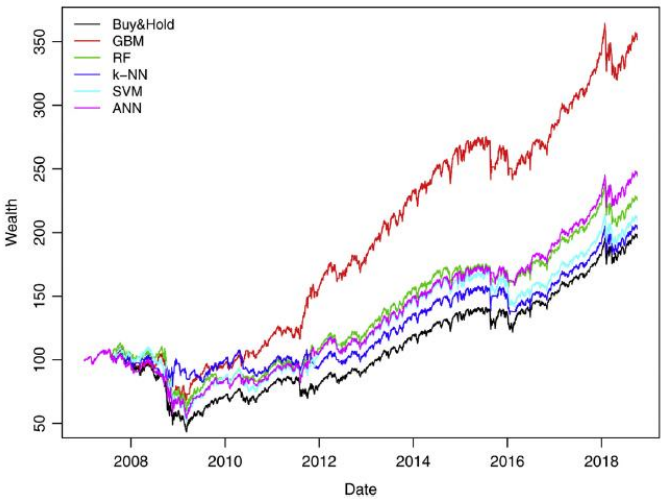

Figura 7. Séries dos retornos acumulados dos modelos testados versus estratégia passiva (Buy&Hold) (Nevasalmi, 2020)

No mesmo sentido, Saifan et al (2020) demonstraram que diferentes métodos de montagem de ML produziram todos melhores resultados que o *benchmark* (Figura 8). Usaram o ETF do S&P500 (SPY) para treinar os modelos mas aplicaram a previsão em ações individuais. O período usado foi entre 2006 e 2015, sendo que a previsão foi entre 2010 e 2015. Usaram 89 fatores preditivos, que incluíram 2 tipos de indicadores técnicos (ATR – *Average True Range* - e *Bollinger Bands*, de notar serem ambos formas de incorporar o comportamento dinâmico da volatilidade) e diferentes retornos passados. Não pré-selecionaram os fatores mais relevantes, usando todos. O treino foi realizado com base nas 1000 observações passadas do ETF no início de cada mês (havendo, assim, treinos recorrentes), sendo a previsão feita diariamente durante o mês seguinte e até novo treino. Adotaram quatro variantes, uma entre pré-escolher aleatoriamente as 36 ações no início do período de treino (2010) e escolher de forma automática com base na capitalização de mercado (primeiras 100 ações acima de 100 milhões de US dólares) e no PE (*Price to Earnings* < 10), mas também entre usar apenas um classificador por cada tipo de algoritmo (com um critério probabilístico) ou dois classificadores (média dos dois sem critério probabilístico). Foram testados os seguintes algoritmos de ML: *Gradient Boosting Classifier* (GBC), *Random Forest Classifier* (RFC), *Extremely Randomized Trees Classifier* (ETC). O melhor algoritmo foi o GBC levando em conta o rácio Sharpe. Uma conclusão interessante é que o aumento de estimadores (cada árvore) e a redução da sua profundidade (número de ramos) aumenta a performance do modelo final.

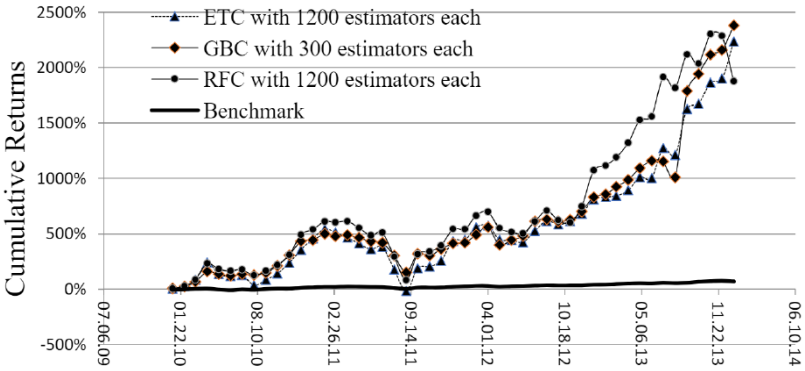

Figura 8. Retornos acumulados dos algoritmos de ML da simulação de negociação, com dois classificadores cada e seleção automática das ações (Saifan et al., 2020)

Normalmente o modelo que prevê é testado quanto à sua previsibilidade e só depois é testado e otimizado para negociação. Chalvatzis e Hristu-Varsakelis (2020) apresentam uma solução híbrida com o objetivo de produzir um modelo que otimize a negociação em simultâneo com a previsão a partir de um modelo LSTM. Isso é feito por se obter a distribuição da previsão e dos resultados desta e decidir a compra ou venda de acordo com o percentil do retorno previsto a partir dessa distribuição. Os resultados deste modelo são comparados com outros algoritmos (RF e XGB) e com modelos de outros artigos, nas métricas MAPE, MAE, RMSE. O período de treino é de 01/01/2005 a 04/01/2010 e o de teste de 04/01/2010 a 20/12/2019 e a simulação das negociações é feita com base no ETF do S&P500 (SPY). Em termos da EMH, é violada a hipótese de eficiência, com todos os algoritmos testados a apresentarem melhores resultados que a estratégia passiva (*Buy and Hold*), mesmo sem vendas a descoberto (*short selling*), com pelo menos o dobro da rentabilidade acumulada, sendo que o LSTM obteve melhor rácio Sharpe.

É interessante de notar que estes resultados foram obtidos mesmo sem o uso de modelos híbridos, o que sugere que ainda se poderia melhorar mais os resultados destes modelos seguindo esta abordagem de otimizar a negociação junto com a previsão.

1.8. Outros artigos relacionados

Yujun, Yimei e Jianhua (2020) apresentam uma comparação entre modelos de vários algoritmos (SVM, KNN, *Bayesian ARD regression*, LSTM) e usam três tipos de métricas: RMSE, MSE e R2. Os modelos sugeridos são modelos híbridos que juntam o LSTM a outra metodologia que usa uma decomposição de frequências, chamada EMD (*Empirical Mode Decomposition*) e EEMD (*Ensemble Empirical Mode Decomposition*). Como usam dados do S&P500 até maio de 2020, o período coincide com o período que se usará na parte experimental desta dissertação, e como também abrange o período da pandemia, os erros são mais comparáveis.

Dionisio, Menezes and Mendes (2007) encontraram sinais de efeitos de memória longa não só na dimensão do nível de preços, mas também na volatilidade dos retornos. Também encontraram evidências de que os retornos de todos os índices acionistas apresentam uma integração fracionada (em vez de unitária como se pressupõe habitualmente por simplificação). Indicando que as séries originais dos preços dos índices apresentam uma integração aproximadamente de 1.5. Este resultado indica que os métodos tradicionais de cointegração, assentam numa premissa diferente da realidade ao pressupor que a integração é unitária, e que o equilíbrio de longo-prazo é estacionário. Como tal, é proposto que se use um modelo de cointegração fracionada.

Nesse sentido, o capítulo 5 do livro de Prado (2018), explica como diferenciar uma série fracionariamente de forma a torná-la estacionária e, assim, perder o mínimo possível de memória, uma vez que advoga que fazer as primeiras diferenças elimina toda a memória existente. Apresenta uma solução e código em Python (embora incompleto) de como, não só calcular as diferenças fracionadas, mas encontrar o valor mínimo do parâmetro de diferenciação 'd' de forma a se obter uma série estacionária. O principal intuito do autor é o de usar o resultado desta diferenciação como variável previsional nos algoritmos de ML.

Uma questão que se coloca habitualmente é como avaliar os modelos de forma a determinar qual é o melhor, uma vez que existem múltiplas métricas e rácios que os ordenam de forma diferente. Emrouznejad, Rostami-Tabar e Petridis (2016) propõem um modelo multiplicativo a que chamam de DEA (*Data Envelopment Analysis*) para ordenar os melhores modelos previsionais com base em várias métricas do erro de previsão. Aplicaram o conceito aos primeiros 22 modelos da competição M3 (edição anterior à M4 referida anteriormente em Makridakis, Spiliotis e Aimakopoulos (2020) ao agregar 5 dessas métricas (RMSE, MAE, MAPE, sMAPE e MASE) para obter um único valor de

pontuação (DEA score) final, onde o melhor modelo apresenta uma pontuação de 1 e os restantes modelos valores iguais ou inferiores até 0. Esta técnica pode ser usada para ordenar os melhores modelos, mas obriga a calcular todas as métricas necessárias ou a tê-las disponíveis, o que raramente acontece quando se quer comparar com modelos de outros artigos. Além de que exige cálculos mais avançados e complexos, não dá para comparar DEA scores calculados separadamente, em contraponto com métricas mais simples como o rácio de Sharpe ou o de Sortino que levam em conta a relação entre o risco e o retorno de uma estratégia, sendo facilmente comparáveis e são sobejamente usados em artigos científicos. Mas é uma boa técnica para combinar o resultado de várias métricas em simultâneo, desde que usada num mesmo estudo de comparação entre modelos e no mesmo período de tempo.

O livro de Jansen (2020) apresenta uma perspetiva estratégica, compreensão conceptual e ferramentas práticas para aplicar ML ao processo de negociação e investimento. Além de ser dos livros mais recentes e completos, fornece exemplos em código Python de todas as ideias e conceitos abordados que auxiliam na implementação de modelos com ML. Por exemplo, o Capítulo 19 apresenta o processo de como construir um modelo previsional do S&P500 que usa o algoritmo LSTM. A previsão foi do ano de 2019 com treino de 2015 a 2018. O RSME foi de 22 e o IC (coeficiente de informação mútua) de 0.98 (Figura 9). No entanto, padece de um problema, infelizmente muito comum, que é o *data leakage*, ou vazamento de dados do período de teste para o período de treino, nomeadamente devido à normalização dos dados ser feita em toda a série em vez de apenas ao período de treino. Esse erro faz com que a escala dos valores do período de teste, que devia de ser desconhecida para o modelo, esteja implícita na escala do período de treino (Soni, 2019). A consequência é as previsões serem melhores no período de treino e/ou teste do que aconteceria se não houvesse vazamento de dados.

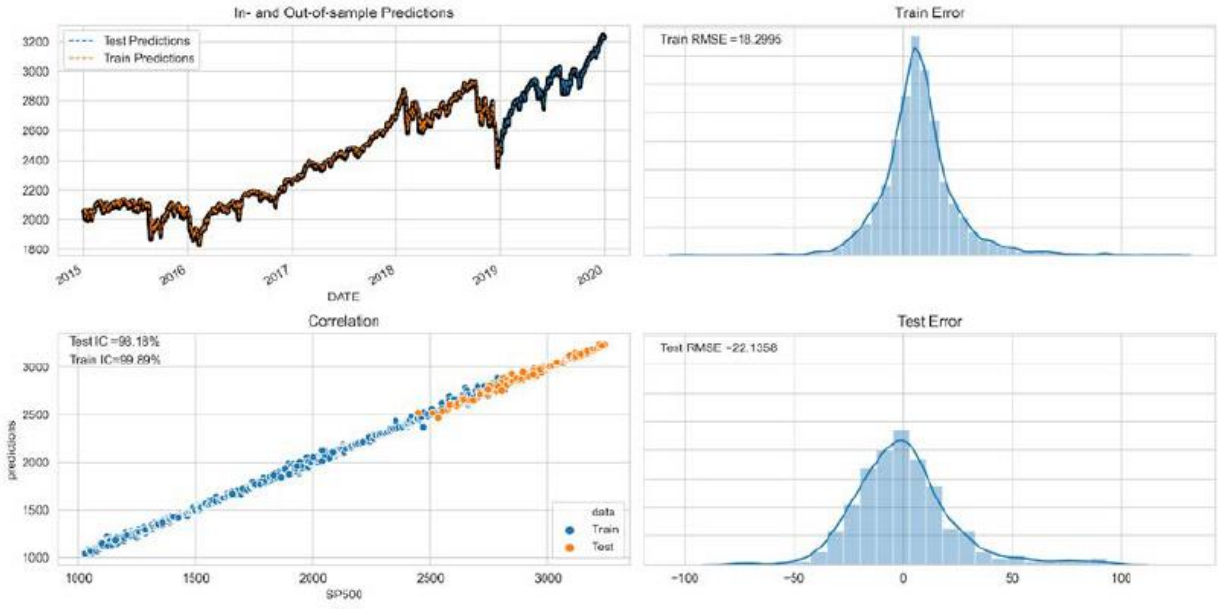

Figura 9. Gráficos com o resultado da previsão ao S&P500 no Capítulo 19 de Jansen (2020)

CAPÍTULO 2

Metodologia

2.1. Escolha dos dados e séries

Um vez escolhido o tipo de modelo híbrido a construir, com base no artigo de Mendes, Ferreira e Mendes (2020), as séries e o período escolhidos são iguais, para ser possível comparar resultados.

Assim, as séries usadas são o índice acionista norte americano S&P500 (SPX) para ser previsto pelos índices acionistas Dow Jones 30 (DJI) e Nasdaq 100 (NDQ) juntamento com as *Yields* dos bilhetes do tesouro americano a 3 meses no mercado secundário (TB3M). O período usado para treinar os modelos é de 05/02/1971 a 12/04/2013, o período de teste de 19/04/2013 a 17/04/2020, para o VECM, sendo o período entre 19/04/2013 a 12/04/2019 usado para validação de Hiper parâmetros do LSTM, e o restante para teste.

Os dados dos índices acionistas foram retirados da base de dados Stooq[2], por apresentar dados com melhor qualidade e com mais dados históricos em comparação a outras bases de dados gratuitas. A série TB3M foi retirada da base de dados do Fred St. Louis da reserva federal americana[3] (série DTB3). Ambas as bases de dados foram acedidas através da biblioteca pandas datareader, da linguagem Python, usando o Jupyter notebook. Foram retirados dados com periodicidade diária, e a estes foram excluídas as datas com valores ausentes, seja por o mercado estar fechado seja porque não existiam numa das outras séries, sendo retirados apenas 2 valores em 12414. De seguida os dados foram transformados em dados semanais sendo o valor de fecho do preço de sexta-feira usado como valor de fecho da semana. As séries finais são compostas por 2568 semanas, com valores de fecho, abertura, máximo, mínimo e volume de transações de cada semana. Finalmente, uniformizou-se as séries dos índices por criar a base 100 no valor de fecho de 05/02/1971.

2.2. Análise Exploratória dos dados

Ao conjunto dos dados, foram retirados apenas os valores de fecho de cada série para futura modelação. A estes dados foram criadas as séries do logaritmo dos índices e respetivos retornos (log-retornos) pelas primeiras diferenças, incluindo a série TB3M (excetuando a aplicação do logaritmo).

Foram elaborados gráficos das séries em nível e também dos log-retornos. Também foram aplicados vários testes estatísticos, como testes de estacionariedade ADF (Augmented Dickey-Fuller),

[2] pandas_datareader.stooq

[3] pandas_datareader.fred; 3-Month Treasury Bill: Secondary Market Rate

PP (Phillips-Perrom) e KPSS (Kwiatkowski-Phillips-Schmidt-Shin)[4], correlograma com gráficos das funções de autocorrelação e autocorrelação parcial[5], medidas estatísticas descritivas como média, mediana, máximo, mínimo, desvio padrão, 1º, 2º e 3º quartis, e matriz de correlação de Pearson[6]. Apenas aos retornos: coeficiente de curtose (*kurtosis*), enviesamento (*skewness*), teste de normalidade, gráfico quantil/quantil em comparação com a distribuição normal[7] (*QQ-plot*). O nível de significância padrão é de 0.05, salvo indicação contrária.

Esta análise introdutória serve para confirmar a natureza dos dados consoante a literatura os descreve (Sewell, 2011), com factos estilizados, onde as séries dos índices acionistas não são estacionárias (a média e a variância não são constantes - apresentando raiz unitária) mas os seus retornos tendem a ser, apresentam tendência não linear (estocástica) de longo-prazo, as distribuições dos retornos são leptocúrticas (apresentam caudas gordas), isto é, maior presença de valores extremos em comparação com a distribuição normal (medido pelo valor da curtose, numa distribuição normal tem o valor de 3, valores maiores implicam distribuições leptocúrticas), apresentam variância com *clusters*, isto é, sequências de elevada variância seguidas por sequências de baixa variância e vice-versa, têm efeito de alavancagem onde a variância é maior para retornos negativos do que para os positivos.

Por fim, os dados foram divididos nos períodos de treino e teste para o modelo VECM, sendo o período de teste encurtado para permitir criar o período de validação, de forma a incluir uma parte das previsões feitas pelo VECM, sendo o restante para teste do LSTM, como indicado na Secção 2.1.

2.3. Causalidade à Granger

De modo a perceber a interdependência entre as séries, foram efetuados testes de causalidade à Granger (Brooks, 2014, pp. 334-335) aos log-retornos das séries dos índices e às primeiras diferenças da série TB3M.

Este teste[8] mede a significância estatística da correlação entre o valor corrente de uma variável e os desfasamentos de outra variável, e caso exista, diz-se que o desfasamento X causa à Granger Y. Diz-se à Granger porque correlação entre variáveis não significa que exista causalidade na realidade. Estes testes ajudam a avaliar as relações que possam existir entre as variáveis estudadas, principalmente, entender que efeitos existem, se existirem, entre a evolução das taxas de juro e os índices acionistas e, se possível, relacionar com a política monetária seguida, especialmente durante o período de

[4] statsmodels.tsa.stattools.adfuller; arch.unitroot.PhillipsPerron; statsmodels.tsa.stattools.kpss
[5] statsmodels.graphics.tsaplots.plot_acf; statsmodels.graphics.tsaplots.plot_pacf
[6] pandas.DataFrame.describe; pandas.DataFrame.corr
[7] scipy.stats.kurtosis; scipy.stats.skew; scipy.stats.normaltest; scipy.stats.probplot
[8] statsmodels.tsa.stattools.grangercausalitytests

elevada inflação nas décadas de 70 e 80, com consequente política monetária restritiva, e durante o período da crise da COVID-19, onde houve uma redução abrupta das taxas de juro, em resultado de uma política monetária expansionista.

O teste foi feito a cada duas variáveis para os primeiros 24 desfasamentos de uma delas em relação ao valor corrente da outra. Assim, foram feitos 288 (12x24) testes em cada período de treino (a todo o período, durante e sem as décadas de 70 e 80) e de validação, 144 (12x12) testes em cada período de teste (a todo o período e à parte sem crise da COVID-19, até 12 *lags*) e 24 (12x2) testes no período da crise da COVID-19 (até ao segundo *lag*). A diminuição do número total de *lags* analisados no período de teste foi devido ao menor número de observações disponíveis, não existindo a quantidade necessária para se proceder ao teste com 24 *lags*.

2.4. Modelo VECM

A metodologia usada para encontrar e estimar o modelo VECM é semelhante ao artigo de referência (Mendes, Ferreira e Mendes, 2020), onde é usado um modelo VECM multivariado com 4 séries, em que é preciso determinar qual o número ótimo de desfasamentos a usar para as séries (todas as séries têm o mesmo desfasamento), o número de relações de cointegração existentes entre elas, se existirem, e qual a componente determinística a usar, se pretendido, de acordo com as características das séries a modelar, nomeadamente, se tem ou não constante, tendência linear, e/ou ambas dentro ou fora das relações de cointegração[9].

Para determinar o número ótimo de desfasamentos ou *lags*, pode ser usado o critério de informação de Akaike (AIC) ou o Bayesiano (BIC). O artigo de referência usou o AIC. O BIC tende a dar resultados mais parcimoniosos, pelo que foi o usado nesta dissertação[10]. Foi usado o teste de Johansen para determinação do número de vetores de cointegração existentes[11].

Após a escolha do melhor modelo VECM a usar, procedeu-se à sua estimação e fez-se a previsão para 7 anos (366 semanas). Recolheu-se os resíduos, analisou-se a sua estacionariedade (ADF e PP)[4], homocedasticidade (se a variância é constante, teste ARCH de Engel)[12], autocorrelação (independência dos resíduos, teste de LJung-Box)[13] e normalidade (se seguem uma distribuição normal)[7].

Em termos teóricos, um modelo VECM é um modelo VAR com correção de erro que permite isolar as relações de cointegração quando existem. Um modelo VAR (Autorregressivo Vetorial) consiste na

[9] 'nc': sem constante; 'co': constante irrestrita (tendência estocástica), fora da cointegração; 'ci': constante dentro da cointegração; 'lo' tendência linear, fora da cointegração; 'li': tendência linear dentro da cointegração; e combinações: 'colo', 'coli', 'cili', 'cilo'.

[10] statsmodels.tsa.vector_ar.var_model.VAR.select_order

[11] statsmodels.tsa.vector_ar.vecm.select_coint_rank

[12] statsmodels.stats.diagnostic.het_arch

[13] statsmodels.stats.diagnostic.acorr_ljungbox

construção de equações para cada variável com desfasamentos (*lags*) seus e das outras variáveis onde os respetivos coeficientes são estimados em simultâneo, resultando num sistema de equações de variáveis endógenas (Mills, 2019, capítulo 13). Assim, este modelo dinâmico (por incluir *lags*) multivariado, com *p lags*, pode ser escrito da seguinte forma (até *n* variáveis, $y_t = (y_{1t}, \dots, y_{nt})$):

$$y_t = c + \sum_{i=1}^{p} A_i y_{t-i} + u_t \quad (1)$$

Onde:

y_t é o vetor das *n* variáveis no período atual *t*

A_i é a matriz $n \times n$ dos coeficientes do *lag i* com cada variável

c é um vetor com as constantes irrestritas ou *drifts* (opcional, foi incluído no modelo estimado).

u_t representa a componente vetorial dos erros ou inovações (ruídos brancos[14])

Se existir correlação entre diferentes variáveis, então os termos de erro também vão ser correlacionados. Pelo que um modelo VAR constrói recursivamente os seus erros de forma a estes serem independentes para cada duas regressões consecutivas.

O modelo VAR da equação (1) pode ser estimado pelo método dos mínimos quadrados (OLS) desde que todas as equações tenham o mesmo número de *lags*, o que se pressupõe, sendo um VAR(*p*). Assume-se que todas as variáveis são estacionárias ou integradas de ordem 0. De forma a poder ser estimado é preciso escolher o número de *lags*. Essa escolha é feita empiricamente através de um procedimento sequencial de teste. Considerando o modelo (1) com uma matriz de covariância dos erros $\Omega_p = E(u_t u_t')$, a sua estimação é dada por:

$$\widehat{\Omega}_p = (T-p)^{-1} \widehat{U}_p \widehat{U}_p' \quad (2)$$

Onde $\widehat{U}_p = (\hat{u}_{p,1}', \dots, \hat{u}_{p,n}')'$ é a matriz dos resíduos obtidos pela estimação OLS de um VAR(*p*), e $\hat{u}_{p,r}' = (\hat{u}_{r,p+1}', \dots, \hat{u}_{r,T}')'$ é o vetor dos resíduos da equação *r* (sendo *T* o tamanho da amostra, *p* observações são perdidas devido aos *lags*, sendo o expoente ' a respetiva transposta). Para encontrar o número ótimo de *lags*, minimiza-se o valor de um critério de Informação, como o AIC (Akaike) ou o BIC (Bayesiano) multivariados, com $p = 0,1, \dots, p_{max}$:

$$MAIC = \log\left|\widehat{\Omega}_p\right| + (2 + n^2 p) T^{-1} \quad (3)$$

$$MBIC = \log\left|\widehat{\Omega}_p\right| + n^2 p T^{-1} \ln T \quad (4)$$

Se as variáveis não têm tendência estocástica, sendo estacionárias, significa que são integradas de ordem 0, ou $I(0)$. Se tiverem tendência estocástica, mas após a aplicação do operador de primeira diferença deixarem de ter, passando a serem estacionárias, significa que são integradas de primeira ordem, ou $I(1)$. Se duas variáveis x e y têm tendência estocástica (ou seja, não são estacionárias) mas

[14] Média nula (E(u_t)=0) e *iid* (independentes e identicamente distribuídos), ou seja, com matriz de covariância positiva e invariante no tempo ($E(u_t u_t') = \sum u$).

a componente residual ($\hat{\varepsilon}$) obtida de um modelo por combinação linear, isto é, $y_t = \beta_0 + \beta_1 x_t + \varepsilon_t$, já não tem, então y fica com a tendência do x, e as variáveis x e y dizem-se cointegradas. Se ambas forem $I(1)$, então diz-se que x e y são cointegradas de ordem 1. Ou seja, se uma combinação linear de variáveis não-estacionárias, mas integradas da mesma ordem, resulta num processo estacionário, então diz-se que as variáveis são cointegradas (os erros não têm tendência estocástica). Isto significa que, embora as variáveis passeiem de forma estocástica, estas ficam próximas uma da outra e convergem para um equilíbrio de longo prazo. O ajustamento da sua posição faz-se através de um mecanismo de correção do erro (EC - *error correction*).

Sejam as variáveis x e y cointegradas de ordem 1, então:

$$\Delta y_t = \beta_0 + \beta_1 \Delta x_t + \beta_2 (y_{t-1} - \gamma x_{t-1}) + \varepsilon_t \quad (5)$$

Onde $(y_{t-1} - \gamma x_{t-1})$ é o termo da correção do erro e β_2 a velocidade de ajustamento (como y muda em resposta ao desequilíbrio), ambos compondo o vetor de cointegração que mantém x e y juntos quando passeiam de forma aleatória. Como as variáveis são cointegradas de ordem 1, o termo da correção do erro é estacionário, pelo que se pode usar o método OLS para a sua estimação.

Colocando o modelo (5) na forma matricial para várias variáveis endógenas, ficamos com um modelo VECM, ou um modelo VAR com incorporação dos vetores cointegrantes (*r*), na forma:

$$\Delta y_t = c + \Pi y_{t-i} + \sum_{i=1}^{p-1} \varphi_i \Delta y_{t-i} + u_t \quad (6)$$

Onde: $\Pi = -I + \sum_{i=1}^{p} A_i$ e $\varphi_i = -\sum_{j=i+1}^{p} A_j$

Π corresponde à matriz de informação de longo prazo e pode ser reescrita na forma $\Pi = \alpha\beta'$, onde α representa a velocidade de ajustamento ao desequilíbrio e β representa a matriz de coeficientes de longo prazo (os vetores cointegrantes *r*). Esta representação é conhecida como o *teorema de representação de Granger*, e, segundo este, quando:

- *r* = *n*: as variáveis em níveis são estacionárias e pode usar-se um VAR(*p*).
- *r* = 0: $\Pi = 0$ e não existe qualquer relação cointegrante no sistema, pelo que se pode usar um VAR(*p*-1) em primeiras diferenças.
- 0 < *r* < *n*: as variáveis são cointegradas e existem *r* vetores cointegrantes.

Para se poder estimar corretamente um modelo VECM é, assim, necessário determinar o número de relações de cointegração. Determinar a característica cointegrante (λ) é o mesmo que determinar quantos vetores cointegrantes existem em β, isto é, quantas colunas são nulas em α (ou equivalente, o número de colunas linearmente independentes existentes em Π). A metodologia de Johansen foi usada para determinar o valor de r. Esta usa dois tipos de testes estatísticos, um baseado na estatística do traço ($\eta_r = -T \sum_{i=r+1}^{n} \log(1 - \lambda_i)$) e o outro na estatística do valor próprio máximo ($\epsilon_r = -T \log(1 - \lambda_{r+1}), \text{com } r = 0,1, \dots, \text{n} - 1$), que testam a hipótese de que há quando muito *r* vetores cointegrantes com *r* < *n*. No caso da estatística do traço, *r* é o número de raízes abaixo do qual as

restantes estatísticas são significativas, ou seja, o valor *r* é selecionado se a última estatística significativa for η_{r-1} rejeitando a hipótese alternativa da existência de *n* – *r* + 1 raízes unitárias em Π. Esta estatística mede, assim, a importância dos coeficientes de ajustamento α sobre os vetores próprios a serem potencialmente omitidos. No caso da estatística do valor próprio máximo, testa-se se *r* + 1 pode ser rejeitado em favor de *r* raízes. Os valores críticos destas estatísticas foram providenciadas por Johansen e Juselius (1990).

Em suma, a metodologia de Johansen permite determinar os parâmetros necessários para se estimar um modelo VECM e pode ser resumida da seguinte forma: primeiro testar a ordem de integração das variáveis (testes de raiz unitária, ADF, PP, KPSS às séries), depois escolher o número adequado de *lags* (p.e. AIC ou BIC), escolher o modelo adequado (com ou sem constante, e/ou tendência), determinar a característica da matriz Π (número de vetores cointegrantes, testes η_r ou ϵ_r), por fim testar para exogeneidade fraca (teste de independência dos resíduos, p.e. LJung-Box, se os resíduos são independentes significa que não existe informação relevante nestes e a estimação do VECM é válida e o modelo está bem especificado).

2.5. LSTM

As redes neuronais artificiais (ANN), como o nome indica, originaram-se da ideia de imitar o processo de aprendizagem do cérebro de forma artificial. No entanto, as redes neuronais profundas (*deep neural networks*) advêm de poderem ser usadas no contexto da aprendizagem de máquina (ML, *machine learning*) e permitir aprender de forma genérica múltiplos níveis de complexidade somente a partir dos dados. O grande desenvolvimento destas redes veio pela criação do algoritmo de retro-propagação (*back-propagation*) a par do conceito matemático do gradiente descendente usados para treinar estas redes junto com o aumento exponencial da capacidade computacional, que permitiu aplicar muitos conceitos e algoritmos anteriormente criados mas que não eram tecnicamente viáveis dado as restrições existentes na tecnologia da época, além de originarem um ressurgimento da investigação e de novas soluções. O livro de Goodfellow, Bengio e Courville (2016) é considerado a referência em *deep learning*, e é a principal fonte usada nesta secção.

Algoritmos de *deep learning* existem desde a década de 1940 (conhecidos como *cybernetics*, entre 1940s-60s, e por *connectionism*, entre 1980s-90s), mas só desde 2006 que são reconhecidos por esse nome e têm visto um crescente interesse e aplicações por parte da indústria das tecnologias de informação. Este vai e vem no interesse pelas redes neuronais está relacionado com os problemas técnicos e matemáticos subjacentes ao seu funcionamento, sendo que o interesse renasce quando novos desenvolvimentos tecnológicos ou novos conceitos matemáticos são descobertos. É o caso em relação ao aparecimento do LSTM por Hochreiter e Schmidhuber (1997), que veio melhorar e

completar a capacidade de memorizar dados de muitos períodos atrás (cerca de 1000) e reduzir os longos tempos de treino, problemas dos anteriores algoritmos de RNN (*Recurrent Neural Networks*), que por sua vez vieram trazer soluções às ANN por incorporarem memória de dados passados (curto prazo) e permitir resolver problemas relacionados com a dependência temporal e ordem sequencial com que os dados originais possam estar associados.

Uma ANN costuma ser organizada com múltiplos processadores ou neurónios (ou células, círculos na Figura 10) a trabalhar em paralelo numa mesma camada, e se tiverem várias camadas ocultas (camada a cinzento na Figura 10) são considerados *deep learning*. A primeira camada recebe os dados como inputs (camada azul na Figura 10) e estes são usados por todos os neurónios dessa camada passando o seu output para todos os neurónios da próxima camada, e assim sucessivamente até à camada final de output (camada laranja na Figura 10).

As redes neuronais para serem treinadas precisam de essencialmente 3 ingredientes: uma função de custo (*loss function*) a minimizar, um optimizador para essa função (algoritmo de optimização global), e um algoritmo de aprendizagem para "ensinar" cada neurónio, isto é, ajustar o seu peso em cada época de treino de forma que os erros de previsão diminuem (a função custo converta para o seu mínimo). Isto foi feito nesta dissertação utilizando o erro quadrático médio para a função de custo, o algoritmo ADAM como optimizador, e a *back-propagation* como metodologia de treino padrão. O papel do optimizador da função de custo é encontrar o mínimo global, quando possível (se não apenas encontra o mínimo local), da função custo num contexto multidimensional. Para ilustrar de forma simples, basta pensar numa função do tipo $y = x^2$, uma parábola, e usando o conceito de gradiente descendente, é possível obter o mínimo da função por calcular a derivada em qualquer ponto inicial e gradualmente ir-se deslocando no sentido de se aproximar da derivada com valor 0. O tamanho do passo a deslocar nesse sentido, é chamada de taxa de aprendizagem (*learning rate*) e é um parâmetro a definir *apriori*, antes do treino, por isso sendo considerado um hiper-parâmetro, em contra ponto aos parâmetros das funções e pesos usados dentro da rede neuronal e estimados durante o treino. O sentido habitual do fluxo de informação de uma rede neuronal é da camada de input à output e, no final, gerar os erros (função de custo), mas o algoritmo de *back-propagation*, como o nome indica, permite que a informação circule no sentido inverso para que se calcule o gradiente com base na função de custo e ao se atualizar os pesos ou parâmetros dos neurónios, através do optimizador, se ir obtendo cada vez mais um menor erro médio em cada época de treino, e assim, conseguir treinar a rede. Ou seja, enquanto o *back-propagation* é usado para calcular o gradiente, o optimizador é usado para treinar ou usar o gradiente para atualizar os parâmetros dos neurónios dentro da rede neuronal. E é na parte do optimizador que os problemas surgem.

Uma RNN é recorrente pois usa a mesma função para todo o input enquanto o output de cada neurónio depende, não só dos inputs dos dados ou dos neurónios da camada anterior, mas também

do neurónio anterior na própria camada (Figura 12). Este processo permite memorizar informação passada, mas o seu treino apresenta graves falhas relacionadas com a explosão ou anulação do gradiente quando a RNN é treinada com séries longas (uma sequência de 10 ou 20 já causa o problema, impedindo a aprendizagem da rede). Este problema pode ser exemplificado pensando nos pesos das conexões ou neurónios que são transferidos de um para os outros dentro da mesma camada (h_t na Figura 12), e que permite guardar a memória de um neurónio para o outro. Se h_t for multiplicado, por exemplo, múltiplas vezes consigo próprio, o resultado será um valor cada vez mais próximo de 0 se o valor inicial for inferior a 1, ou explodir para um valor muito grande se o valor inicial for superior a 1. O habitual é aproximar-se de 0 e não contribuir para a aprendizagem da rede que acaba só por conseguir aprender relações de muito curto-prazo.

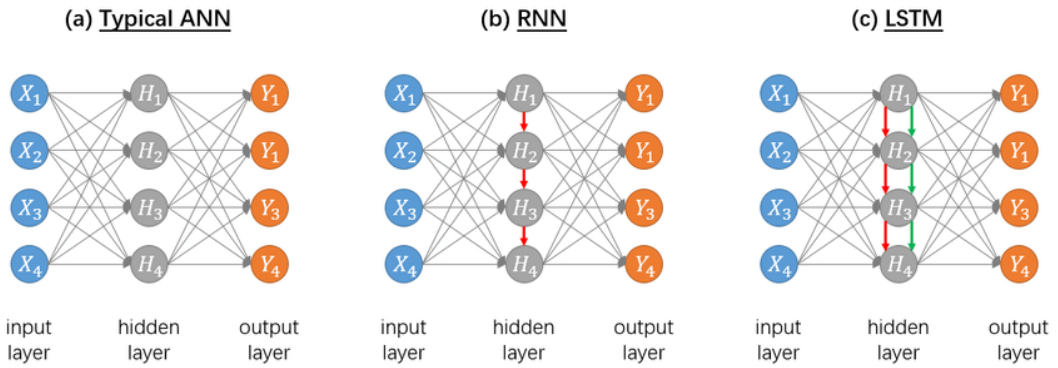

Figura 10. Representação genérica das diferentes estruturas das redes neuronais abordadas (ANN, RNN e LSTM) (Ma *et al.*, 2019)

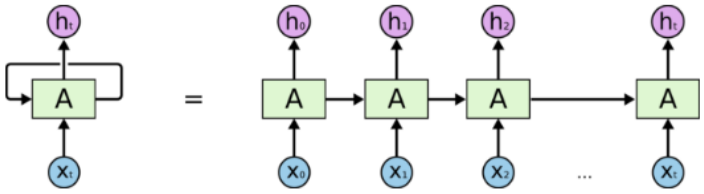

Figura 12. Célula de uma RNN e respetiva sequência na análise dos dados numa camada (Mittal, 2019)

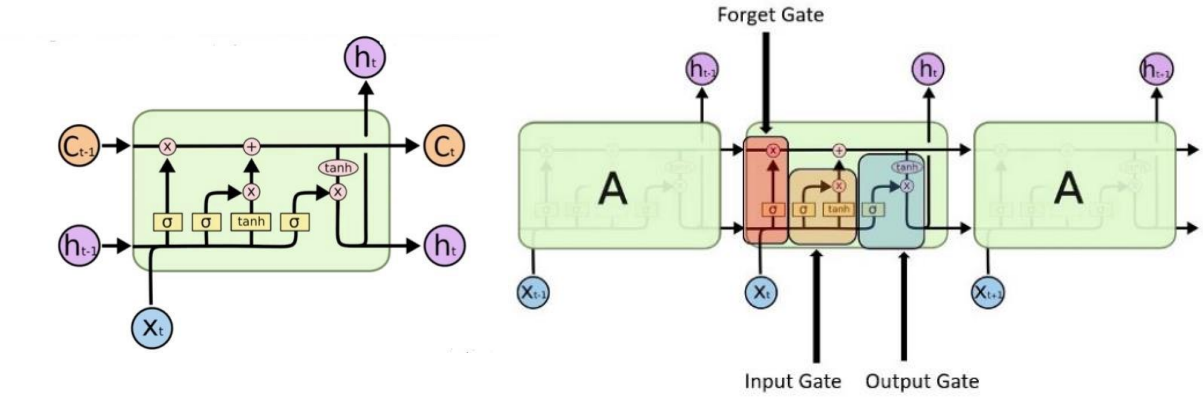

Figura 11. Célula do LSTM (esquerda) e o fluxo de informação ao longo do tempo (direita) (Mittal, 2019)

O LSTM surgiu como resposta ao problema da anulação do gradiente nas RNN ao se querer memorizar séries longas. O LSTM é composto por células que apresentam 3 partes: uma para esquecer a informação irrelevante (*Forget Gate*) do anterior estado da célula, outra para atualizar com nova informação (*Input Gate*), e a terceira para passar a informação atualizada para o passo temporal seguinte (*Output Gate*). Assim como os neurónios de uma RNN, as células de um LSTM também têm o estado oculto (*h*) que representa a memória de curto-prazo, mas acresce o estado da célula (não oculto, *C*, caminho das setas verdes na Figura 10(c)) que representa a memória de longo prazo. Assim, enquanto uma RNN tem uma recorrência na camada, um LSTM acrescenta uma recorrência dentro da própria célula.

O LSTM usa em cada célula funções de ativação que são comuns a toda a camada, pretendem captar padrões não lineares e facilitam a *back-propagation* porque têm funções de derivada que se relacionam com os inputs, sendo as usadas por defeito a sigmoid (função logística, σ), para a recorrência interna da célula, que condiciona os valores a um intervalo entre 0 e 1, e a tangente hiperbólica (*tanh*), para a transformação do output da célula e input para a recorrência da camada, que transforma os valores numa escala de -1 a 1. Estas funções padecem do mesmo mal das RNN no sentido de o gradiente tender a se anular, apesar com menor frequência. No sentido de melhorar esse problema, a função de ativação *ReLu* (*Rectified Linear Unit*) que toma valores sempre positivos (Figura 13), apresenta melhores resultados e é a mais usada na área, mesmo assim, não elimina por completo o problema da anulação do gradiente em alguns valores de input próximos de zero. Foi esta função de ativação, *ReLu*, usada nesta dissertação.

O optimizador usado, ADAM (Kingma and Ba, 2015), diminui em muito o problema do gradiente desaparecer. É um algoritmo baseado no gradiente de primeira ordem (primeira derivada) de funções custo estocásticas, baseado em estimativas adaptativas de momentos de ordem inferior. O ADAM é um dos mais recentes algoritmos de optimização de última geração e dos mais usados por muitos utilizadores de aprendizagem automática. O primeiro momento do gradiente normalizado pelo segundo momento dá a direção da atualização. Este optimizador mitiga o decaimento rápido do gradiente por usar uma média exponencial da raiz quadrada dos gradientes passados, diminuindo, assim, a dependência da atualização do parâmetro ao último gradiente ou a poucos passados (Sanghvirajit, 2020).

A Figura 11 ilustra o processo que ocorre dentro de uma célula LSTM. C_{t-1} é o estado anterior da célula, h_{t-1} o output anterior da célula e x_t o input atual da célula. C_t e h_t são o estado atual da célula (memória de longo prazo) e o seu output (memória de curto prazo) respetivamente.

A primeira camada é a unidade de esquecimento ($f_i^{(t)}$), onde é determinada qual a informação do estado anterior da célula que é esquecida e qual a que será memorizada. Primeiro é feita uma transformação linear de h_{t-1} e x_t com pesos e viéses antes de serem passados para a função sigmoid. Na sigmoid os valores são comprimidos entre 0 e 1, onde 0 significa que a informação é toda esquecida e 1 que é toda memorizada.

Na unidade de estado ou de input ($C_i^{(t)}$), estão compreendidas três etapas, na primeira, a dos inputs externos ($g_i^{(t)}$), é feita uma transformação semelhante à unidade anterior onde é feita uma transformação linear de h_{t-1} e x_t com pesos e viéses antes de serem passados para a função sigmoid. Na segunda etapa é feita nova transformação linear idêntica de h_{t-1} e x_t mas desta vez passados para a função *ReLu*. Por fim, a informação é atualizada somando a multiplicação das anteriores transformações com a multiplicação da informação do estado anterior da célula com o output da unidade de esquecimento, formando o novo estado da célula (C_t).

A unidade de output, através da porta de output ($q_i^{(t)}$), tem a capacidade de desligar o output da célula $h_i^{(t)}$. O output da célula usa a função *ReLu* depois de fazer transformação linear idêntica às anteriores de h_{t-1} e x_t. A porta de output ($q_i^{(t)}$) faz a mesma transformação a h_{t-1} e x_t mas passa-a para uma função sigmoid. Por fim, o output final ($h_i^{(t)}$) resulta da multiplicação dessas funções.

Em termos matemáticos, as partes constituintes da célula do LSTM são assim representadas:

- Unidade de esquecimento (*Forget Gate*), para a célula *i* em *j* células e período temporal *t*:

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right) \quad (7)$$

  onde: $x^{(t)}$é o vetor de inputs, $h^{(t)}$é o vetor corrente da camada oculta contendo os outputs de todas das células do LSTM, b_i^f é o viés, U^f é a matriz dos pesos dos inputs, W^f é a matriz dos pesos da recorrência das unidades de esquecimento, e σ é a função sigmoid.

- Unidade de Input (ou de estado), que apresenta um auto circuito (*self loop*):

$$C_i^{(t)} = f_i^{(t)} C_i^{(t-1)} + g_i^{(t)} ReLu\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right) \quad (8)$$

  Sendo a etapa dos inputs externos ($g_i^{(t)}$) calculada de forma análoga à unidade de esquecimento, mas com os seus próprios parâmetros:

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right)$$

- Unidade de output:

$$h_i^{(t)} = ReLu(C_i^{(t)})\, q_i^{(t)} \quad (9)$$

  Onde: $$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right)$$

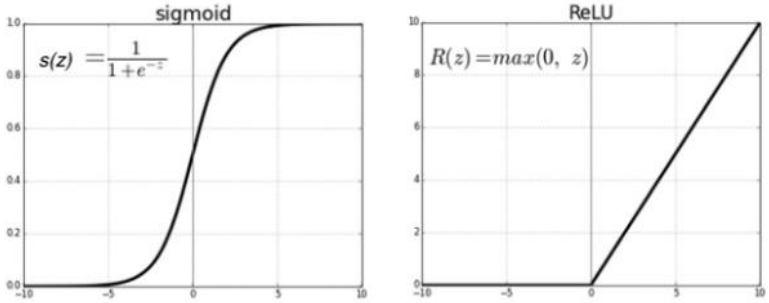

Figura 13. Funções de ativação: sigmoid (esquerda) versus ReLu (direita) (Chaudhary, 2020)

A estrutura final usada para o LSTM foi a mesma usada no artigo de referência: quatro camadas, a de inputs com as 4 séries (e respetivos *lags*), duas ocultas com 50 células cada e uma de output com apenas uma célula.

Antes de introduzir os valores das séries no LSTM foi necessário proceder à normalização dos dados de treino para ficarem numa escala de 0 a 1. Para isso usou-se o algoritmo MinMaxScaler[15] que usa os valores mínimos e máximos para transformar os valores na escala pretendida, mas apenas no período de treino. Essa nova escala é depois aplicada aos dados de validação e de teste, que poderão ter valores inferiores a 0, ou superiores a 1, caso existam novos mínimos ou máximos. Ao se aplicar a escala dos dados de treino aos períodos de validação e teste, evita-se passar informação que deve ser desconhecida para o modelo no período de treino, evitando o vazamento de dados (*data leakage*). Se o algoritmo fosse aplicado a toda a série, sem separação entre períodos, a escala obtida teria o valor 1 no valor máximo, que no caso de ocorrer durante o período de teste, a escala obtida continha informação implícita sobre o máximo futuro, que passaria para o período de treino, informação que não deveria de ser passada. Por isso se diz que houve um vazamento de dados do período de teste para o período de treino, não que o valor do máximo passasse diretamente (porque o valor em si não é usado durante o treino), mas esse valor passaria a estar implícito na escala usada no período de treino (que teria um valor máximo < 1, em vez de 1, ou um valor mínimo > 0, em vez de 0).

De seguida as observações das séries são organizadas em grupos de 4 ou 20, consoante o número de *lags* escolhido, para todos os dados de treino, e para cada série, ficando com um formato tridimensional com uma dimensão da amostra igual à original menos o número de *lags*.

A taxa de aprendizagem (*learning rate*) foi de 0.001, a escolha aleatória dos dados foi desativada por se tratar de dados com sequência temporal, o número de épocas de treino foi de 30, o tamanho do *batch* foi de 32 (o algoritmo usa conjuntos de 32 valores de cada vez de forma a usar a memória computacional de forma mais eficiente, até percorrer todos os valores das séries, existindo 69 *batchs* em cada época, 68 completos e o último com os valores restantes para os 2195 dados amostrais para os 4 *lags*, e para os 2179 dados amostrais para os 20 *lags*), a função de ativação usada foi a *ReLu*, o optimizador usado foi o ADAM e a previsão foi feita para o período seguinte.

[15] sklearn.preprocessing.MinMaxScaler

2.6. Montagem de modelos e Previsão final

Os modelos híbridos foram montados pela simples adição dos resíduos previstos pelo LSTM à previsão feita pelo VECM durante o período de teste, sendo uma montagem sequencial.

Para comparar resultados foram criados os seguintes modelos de referência:

- NAIVE: previsão igual ao valor anterior ao período de teste;
- Médias móveis de 4 e 20 períodos;
- LSTM univariado com 4 e 20 *lags* (dois modelos);
- LSTM multivariado com 4 e 20 *lags* (dois modelos);
- Previsão do VECM para 7 anos, sendo usado a parte correspondente ao período de teste.

Aos resíduos dos modelos de referência foi calculado os valores das métricas de erro MAPE% (erro absoluto médio em percentagem), MAE (erro absoluto médio), RMSE (raiz do erro quadrático médio) e RMSE% (RMSE em percentagem). Foram ainda realizados outros cálculos, nomeadamente o teste ADF com 1 desfasamento (valor-p para estacionariedade) e a correlação entre os valores preditos pelos modelos e os valores atuais da série do S&P500. Os mesmos cálculos destas métricas foram feitos para os modelos híbridos de forma a se comparar os resultados. Além disso, ainda se comparou estas métricas entre o período com e sem a crise da COVID-19 (até 14/02/2020 inclusive).

Foram criados os seguintes modelos híbridos (VECM +LSTM):

- VECM + LSTM com 4 *lags* com os inputs originais das séries, mas normalização com vazamento de dados (modelo mais aproximado ao do artigo de referência);
- VECM + LSTM com 4 e 20 *lags* com os inputs originais das séries (dois modelos);
- VECM + LSTM com 4 e 20 *lags* com os inputs do logaritmo dos índices acionistas e série original TB3M (dois modelos);

A todos os modelos com LSTM foi selecionado a época com menor erro no período de validação medido pelo MAE, obtendo-se os pesos idênticos aos que se obteriam usando a técnica denominada de *Early stopping*, ou paragem precoce, que consiste em terminar o período de treino (ou terminar na época) antes dos erros do período de validação começarem a subir, de modo a se evitar o *overfitting*[16], duplicando, assim, o número de modelos com LSTM a avaliar.

Desta forma, pretende-se analisar, além do impacto da crise da COVID-19, o impacto do número de *lags*, da logaritmização das séries, do vazamento de dados, da hibridização de modelos, e do número de épocas a treinar na capacidade preditiva dos modelos.

[16] Ajustamento exagerado do modelo aos dados de treino que resultaria numa pior previsão do modelo quando confrontado com dados novos fora da amostra, evidenciando perda da capacidade de generalização das relações existentes nos dados.

CAPÍTULO 3

Resultados e Discussão

3.1. Resultados

3.1.1. Análise Exploratória dos dados

Após aplicação da metodologia enunciada no capítulo anterior, os resultados são apresentados de seguida e as tabelas, figuras e outros pormenores secundários são apresentados no ANEXO I.

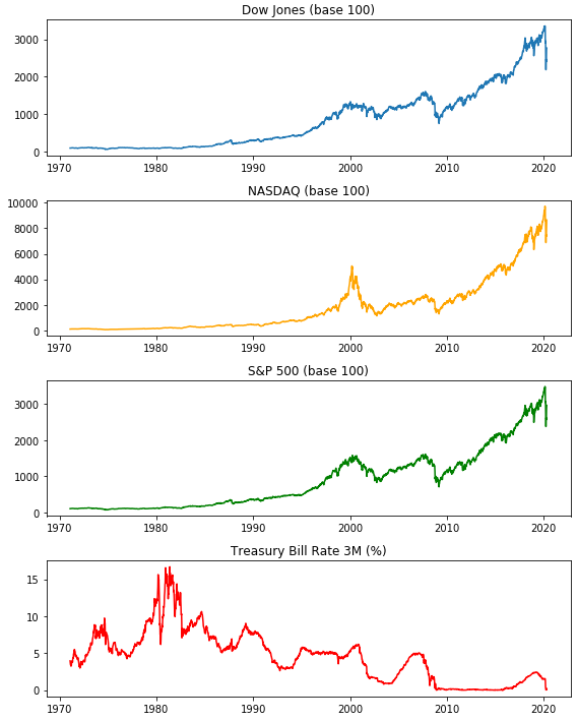

Figura 14. Gráfico do nível das séries usadas (base100) de 05/02/1971 a 17/04/2020

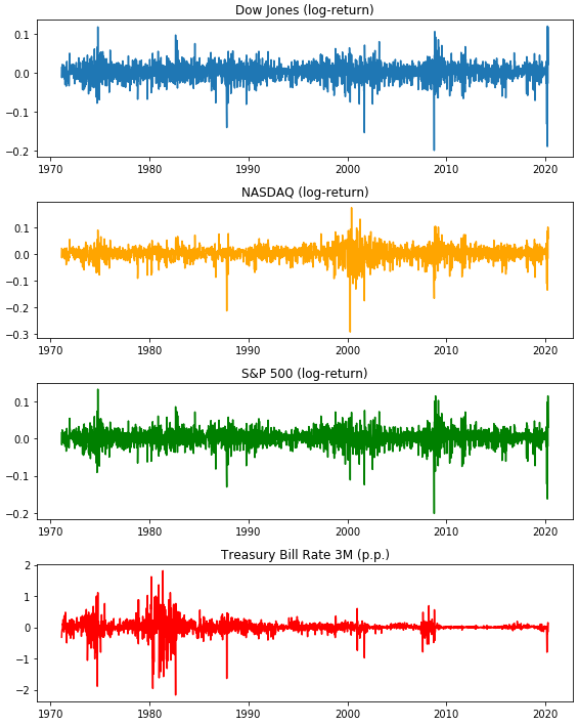

Figura 15. Gráfico dos log-retornos das séries usadas

Tabela 2. Estatísticas descritivas das séries em nível (esquerda) e dos log-retornos (direita)

	SPX	DJI	NDQ	TB3M
count	2568.000000	2568.000000	2568.000000	2568.000000
mean	876.430440	844.857488	1767.217909	4.577465
std	799.899772	792.218180	2018.496908	3.421421
min	64.310000	65.890000	55.200000	-0.010000
25%	158.075000	129.052500	245.250000	1.630000
50%	594.440000	539.430000	1018.960000	4.810000
75%	1340.155000	1263.612500	2460.492500	6.420000
max	3487.220000	3353.650000	9731.176000	16.680000

	SPX	DJI	NDQ	TB3M
count	2567.000000	2567.000000	2567.000000	2567.000000
mean	0.001320	0.001301	0.001738	-0.001496
std	0.022893	0.023129	0.027616	0.225498
min	-0.201000	-0.200000	-0.292000	-2.170000
25%	-0.011000	-0.011000	-0.011000	-0.050000
50%	0.003000	0.003000	0.003000	0.000000
75%	0.014000	0.014000	0.017000	0.050000
max	0.132000	0.121000	0.174000	1.810000

A Tabela 2 apresenta algumas estatísticas descritivas. O Nasdaq teve um retorno total superior, 3x maior que os outros dois índices, e 2x mais desvio padrão, durante o período considerado (1971 a

2020), mas a média e o desvio padrão dos retornos são semelhantes. O SPX e o DJI tiveram um comportamento semelhante em termos de retorno total e desvio padrão, tendo uma correlação máxima de 1 entre si e de 0.98 com o NDQ. A correlação dos retornos é ligeiramente inferior mas positiva e elevada na mesma. A correlação entre a TB3M e os índices é negativa e elevada (-0.67) nas séries em nível mas inexistente (0.0) nos retornos (matrizes de correlação linear na Figura 24).

A destacar está a constatação da veracidade de todos os factos estilizados em relação às séries financeiras. As séries dos índices apresentam uma tendência crescente e a TB3M decrescente (gráfico das séries em nível na Figura 14). As séries não têm distribuição normal (gráfico probabilístico de normalidade na Figura 23), a curtose é muito superior a 3 (valor da distribuição normal), isto é, as distribuições dos retornos são leptocúrticas, os seus retornos não são simétricos, existindo um enviesamento à esquerda (sendo maior em ambas as métricas na TB3M, Tabela 8), os efeitos de alavancagem estão presentes, a variância é maior nas quedas, existem *clusters* de volatilidade (gráfico dos log-retornos na Figura 15 e respetivo histograma na Figura 22).

As séries em nível não são estacionárias, ou seja, apresentam uma raíz unitária, segundo todos os testes aplicados (ADF, PP, KPSS, Tabela 7). Já os retornos das séries apresentam estacionariedade fraca, também com todos os testes a concordar. Assim, as séries em estudo são integradas de ordem 1, pois as primeiras diferenças tornam as séries estacionárias (Tabela 8).

3.1.2. Causalidade à Granger

Na Figura 16 podem ser vistos os *lags* estatisticamente significativos para um alfa de 0.05, para cada par de séries no período de treino (12/04/1971 a 12/04/2013). As restantes figuras estão no Anexo I.B.

O resultado global mostra que existe uma causalidade bidirecional entre a série TB3M e os índices. Entre o SPX e o NDQ existe causalidade bidirecional no *lag* 7, e unidirecional (do SPX para o NDQ) nos restantes *lags* entre 4 e 17. Existe causalidade unidirecional do DJI para o SPX a partir do *lag* 11. Em suma, olhando apenas para as variáveis que causam o SPX, os *lags* 3 a 17 do TB3M, os *lags* 11 a 24 do DJI, e o *lag* 7 do NDQ causam à Granger o SPX. Assim verifica-se que, das séries analisadas, a série do TB3M e do DJI são as mais relevantes para a previsão do SPX.

Lags **estatisticamente significativos com** α **= 0.05 (X causa Y)**

```
Y=SPX,X=SPX,Lags=[]
Y=DJI,X=SPX,Lags=[]
Y=NDQ,X=SPX,Lags=[4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]
Y=TB3M,X=SPX,Lags=[3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
Y=SPX,X=DJI,Lags=[11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
Y=DJI,X=DJI,Lags=[]
Y=NDQ,X=DJI,Lags=[4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
Y=TB3M,X=DJI,Lags=[6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]
Y=SPX,X=NDQ,Lags=[7]
Y=DJI,X=NDQ,Lags=[]
Y=NDQ,X=NDQ,Lags=[]
Y=TB3M,X=NDQ,Lags=[6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
Y=SPX,X=TB3M,Lags=[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]
Y=DJI,X=TB3M,Lags=[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]
Y=NDQ,X=TB3M,Lags=[3, 4, 5, 6, 7, 8, 9, 10, 11]
Y=TB3M,X=TB3M,Lags=[]
```

	SPX_x	DJI_x	NDQ_x	TB3M_x
SPX_y	1.0000	0.0023	0.0316	0.0030
DJI_y	0.0736	1.0000	0.0995	0.0023
NDQ_y	0.0078	0.0013	1.0000	0.0128
TB3M_y	0.0003	0.0097	0.0007	1.0000

Valores mínimos do p-value para todos os *Lags* de cada combinação de 2 séries

Rejeitar não causalidade se
P-value <= 0.05 (então X causa Y)

Figura 16. Causalidade à Granger entre os log-retornos das séries no período de treino

Subdividindo o período de treino em dois, durante as décadas de 70 e 80, onde existiu uma política monetária agressiva de combate à inflação na década de 80 (Figura 26), e excluindo-as (de 01/01/1990 a 12/04/2013), os resultados diferem na medida em que a causalidade bidirecional entre TB3M e os índices é quebrada apenas no DJI (é causado por mas não causa) durante essas décadas, e na sua totalidade após esse período. Ou seja, excluindo as décadas de 70 e 80, apenas os índices causam a TB3M e o SPX é causado unicamente pelos *lags* 7, 11, e 14 a 24 do DJI. Durante essas décadas, o SPX é causado por todas as outras séries:

Durante o período de validação (de 19/04/2013 a 12/04/2019) não foi encontrada nenhuma relação de causalidade à Granger para o SPX (Figura 27).

No período de teste (de 19/04/2019 a 17/04/2020), voltou a ser encontrada uma relação causal bidirecional entre a TB3M e os índices, sendo a principal diferença em relação ao período de treino que o SPX é agora causado pelos *lags* 1, 6 e do 9 ao 12 do NDQ e não pelos do DJI (Figura 28). Também a causalidade de TB3M para o SPX passou a ser para todos os *lags* desde o 1 ao 12, e não apenas a partir do *lag* 3. De forma a entender o impacto da crise da COVID-19 nestes resultados, subdividiu-se em período de teste sem a crise e a parte com a crise.

Na parte do período de teste sem crise (até 14/02/2020), a causalidade bidirecional entre a TB3M e os índices quebra-se, sendo os índices que causam a TB3M, mas foi encontrada uma causalidade bidirecional entre os índices. Os *lags* 8 a 12 do DJI e *lags* 1, 4 ao 12 do NDQ causam o SPX (Figura 29). Durante a crise da COVID-19 (a partir de 21/02/2020), os resultados mantêm-se na causalidade bidirecional entre os índices, mas invertem-se na causalidade unidirecional que agora passa a TB3M a causar os índices. Assim, o SPX é causado pelos *lags* 1 e 2 do NDQ e 2 do DJI e TB3M (Figura 30).

3.1.3. Modelo VECM

Em contraste com o enunciado no artigo Mendes, Ferreira e Mendes (2020) onde encontraram 3 relações de cointegração (metodologia de Johansen) para 2 desfasamentos (critério AIC), os resultados obtidos foram 0 relações de cointegração para as séries em níveis (metodologia de Johansen, para nível de significância de 0.10), e o número ótimo de *lags* foi de 8 pelo AIC e 1 pelo BIC. Mas se os mesmos testes forem aplicados ao logaritmo das séries, os resultados alteram-se na medida em que foi encontrada 1 relação de cointegração, e 2 *lags* pelo BIC, mantendo-se os 8 pelo AIC.

Pelo que foi **escolhido o modelo VECM com 2 desfasamentos, 1 cointegração e constante fora da relação de cointegração** para as séries em logaritmo (VECM(2)_1co) (parâmetros nas Figura 35 a Figura 37 no Anexo I.C). A escolha da constante irrestrita (a deriva de um passeio aleatório) prende-se pelo facto das séries dos índices apresentarem tendência estocástica, pois têm raiz unitária (Tabela 7). A relação de cointegração encontrada foi entre o SPX que é explicado pelo DJI e pela TB3M, sendo estatisticamente não significativo com o NDQ (Figura 37).

Os resíduos do modelo apresentam estacionariedade, pelos 3 testes (ADF, PP e KPSS, Figura 33). Os resíduos do SPX e do DJI não são autocorrelacionados até ao desfasamento 5. No entanto, existe autocorrelação nos resíduos das restantes séries (NDQ e TB3M), consoante teste de Ljung-box de independência dos resíduos (para 10 *lags*). Para um alfa de 0.01, a conclusão não se altera (Figura 31 e Figura 32). Aplicando o teste de Engle para a Heterocedasticidade autorregressiva condicional (ARCH), os resíduos de todas as séries não são homocedásticos, ou seja, a sua variância não é constante (Figura 33), evidenciando a existência de efeitos não lineares na variância.

Efetuando a previsão para 7 anos (366 semanas), o modelo VECM escolhido apresenta melhor previsão que o modelo VECM apresentado no artigo replicado (linhas laranjas da Figura 17).

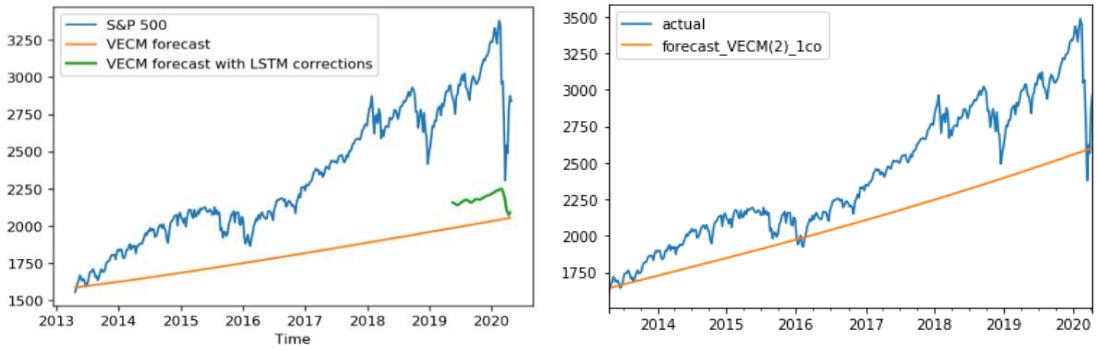

Figura 17. Previsão do modelo do artigo (Mendes, Ferreira e Mendes, 2020) (esquerda) e do VECM(2)_1co (direita)

3.1.4. LSTM

Todos os LSTM criados, à exceção do modelo com vazamento de dados (Figura 42), tiveram um treino estável por ser convergente, diminuindo os erros tanto do período de treino como o de validação ao longo das épocas, convergindo mais depressa para os com 4 *lags* do que para os com 20 *lags*. O número de épocas usado, 30, mostrou-se excessivo para todos os LSTM dos modelos híbridos por aumentar o erro durante o período da validação (Figura 18), o que evidencia a existência de *overfitting*, espelhado no maior erro das previsões dos modelos treinados com 30 épocas (Tabela 10) em relação aos modelos treinados até à melhor época (antes dos erros do período de validação começarem a subir).

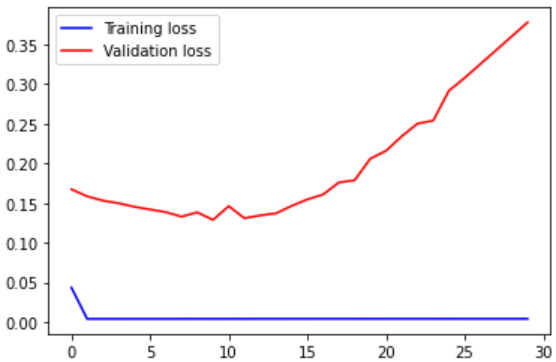

Figura 18. Curvas dos erros por época durante os períodos de treino e validação: modelo híbrido com 20 *lags*

As restantes curvas de aprendizagem, com os erros do período de teste e de validação no final de cada época podem ser vistas no Anexo I.D.

3.1.5. Montagem de modelos e Previsão final

As medidas de erro de todos os modelos em simultâneo estão na Tabela 10 para todo o período de teste, ordenados pelo MAPE% na Tabela 11, para o período sem COVID-19 na Tabela 12, e ordenados pelo MAPE% na Tabela 13, todas no Anexo I.E.

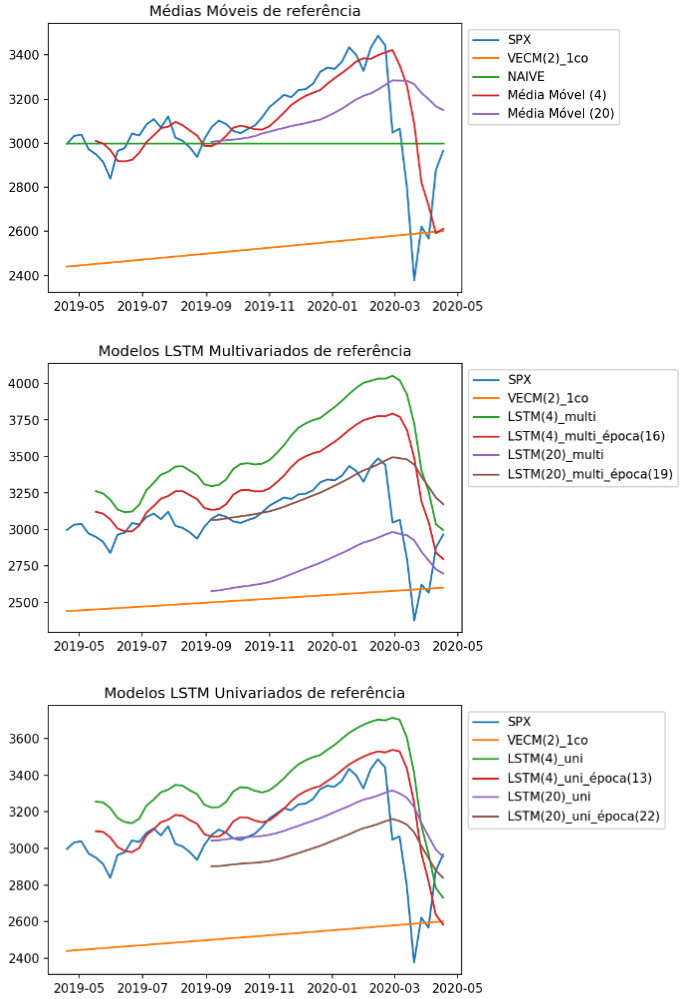

Figura 19. Gráficos da previsão dos modelos de referência

Foram usadas médias móveis para comparar os modelos mais complexos com um procedimento simples que apenas alisa os valores passados para se ter uma noção de como diferem disso. Os modelos *benchmark* são os modelos LSTM multivariados, pois analisam as mesmas séries que os modelos híbridos. Os modelos LSTM univariados foram criados para se ter a noção de como um modelo multivariado se compara com um modelo univariado. Entre os modelos univariados, o com menor MAPE% foi o com 4 lags e treinado até à época 13 (4.7%), e o melhor com 20 lags foi o treinado até à época 30 (5.7%). Já nos multivariados, os melhores modelos foram os que tiveram um treino encurtado, na época 16 para os com 4 *lags* (8.39%) e na época 19 nos com 20 *lags* (6.14%), sendo o melhor modelo o com 20 *lags* (Figura 19 e Tabela 9). Comparando os LSTM univariados com os multivariados, os modelos univariados foram melhores que os multivariados, mas se analisarmos apenas o período sem a crise da COVID-19, o melhor modelo multivariado com 20 lags é o modelo que conseguiu o menor valor de MAPE% entre todos os modelos analisados e até que a média móvel de 4 (1.97%) (que obteve o menor valor de MAPE entre todos os modelos no período completo de teste, 3.7%). Nesse período sem a COVID, o melhor modelo com 4 *lags*, foi novamente o univariado (2.31%). Em relação ao valor de referência NAIVE, previsão igual ao valor do último período conhecido, a crise da COVID, ao criar uma correção no preço, diminuiu os erros associados ao NAIVE, que retirando essa crise, passa do 3º melhor resultado entre os modelos de referência para a 6ª posição. *Generalizando, os modelos com menor treino (<30) tendem a obter melhores resultados, entre os univariados e os multivariados, os univariados são melhores para 4 lags e os multivariados para 20 lags.*

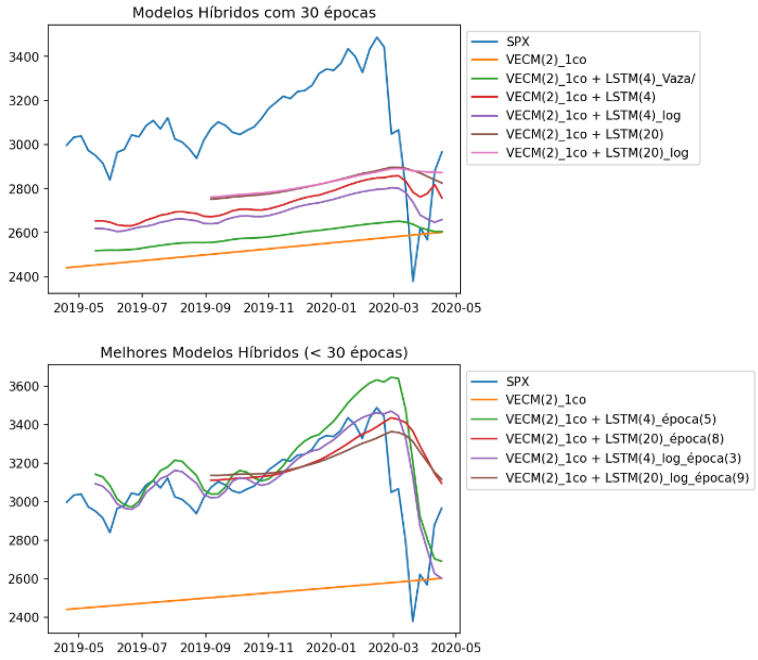

Figura 20. Gráficos da previsão dos modelos híbridos

É, assim, evidente, pela observação tanto dos gráficos (Figura 20) como das métricas de avaliação das previsões (Tabela 3 à Tabela 6), que os melhores resultados obtidos foram os dos modelos onde o número de épocas durante o treino do LSTM foi mais curto, isto é, inferior a 30 épocas (o que de certa forma é expectável, pois o conjunto de dados tem um número relativamente baixo de observações). Para os modelos híbridos, bastam até 9 épocas ou menos, para treinar os modelos, e acima disso, o risco de *overfitting* aumenta, com os erros de previsão na fase de teste a aumentarem quanto mais treino houver a partir daí, mesmo que durante o treino o erro do período de treino possa diminuir.

Tabela 3. Métricas de avaliação das previsões dos modelos híbridos com 4 *lags* e 30 épocas e respetivos modelos de referência, ordenados pelo mape%, no período de teste com e sem COVID-19 (até 14/02/2020)

Período de Teste Modelos	mape%	mae	rmse	rmse%	acf1	corr
Média Móvel (4)	3.7429	108.6917	169.0513	6.2854	0.6479	0.6749
NAIVE	5.5138	170.5628	230.7686	7.6259	0.8568	-0.0000
VECM(2)_1co + LSTM(4)	12.4340	390.2723	414.9227	13.0522	0.8481	0.3224
VECM(2)_1co + LSTM(4)_log	13.3841	421.0698	447.0110	14.0225	0.8247	0.4911
LSTM(4)_multi	14.3620	435.6693	506.1210	17.3557	0.8122	0.5507
VECM(2)_1co + LSTM(4)_vaza/	16.5080	520.4303	551.3086	17.2555	0.8444	0.2681
VECM(2)_1co	18.5415	596.3743	639.4035	19.6283	0.8573	-0.2795
Período de Teste não COVID **Modelos**	**mape%**	**mae**	**rmse**	**rmse%**	**acf1**	**corr**
Média Móvel (4)	1.9753	61.6957	68.1575	2.1984	0.5762	0.9211
NAIVE	4.6728	152.0820	204.8337	6.1486	0.8869	-0.0000
LSTM(4)_multi	11.5606	366.6429	390.1278	12.1875	0.8626	0.9261
VECM(2)_1co + LSTM(4)	13.4497	426.7526	439.5117	13.7143	0.8197	0.9263
VECM(2)_1co + LSTM(4)_log	14.5120	460.3017	473.1341	14.7687	0.8266	0.9374
VECM(2)_1co + LSTM(4)_vaza/	17.9843	569.9906	583.6914	18.2406	0.8517	0.9282
VECM(2)_1co	21.4531	697.3532	706.6242	21.6124	0.8412	0.9640

Entre os modelos híbridos criados a partir do logaritmo das séries dos índices acionistas, as medidas de erro são muito aproximadas, não sendo um procedimento essencial para que um LSTM capte melhor as relações nos dados. Os modelos dos logaritmos dos índices apenas apresentaram uma ligeira melhoria para os melhores modelos com 4 *lags* (3.97 versus 5.20, Tabela 4), não apresentando nos restantes modelos.

Entre o número de desfasamentos a pôr como input no LSTM dos modelos híbridos ser de 4 ou 20, os modelos com 4 *lags* têm menores erros (melhor modelo com MAPE% de 3.97 versus 6.05, Tabela

Tabela 4. Métricas de avaliação das previsões dos modelos híbridos com 4 *lags* e melhores épocas e respetivos modelos de referência, ordenados pelo mape%, no período de teste com e sem COVID-19 (até 14/02/2020)

Período de Teste Modelos	mape%	mae	rmse	rmse%	acf1	corr
Média Móvel (4)	3.7429	108.6917	169.0513	6.2854	0.6479	0.6749
VECM(2)_1co + LSTM(4)_log_época(3)	3.9669	113.8389	187.1198	6.9366	0.6788	0.6360
VECM(2)_1co + LSTM(4)_época(4)	5.2094	152.5272	233.1271	8.4306	0.7465	0.6128
NAIVE	5.5138	170.5628	230.7686	7.6259	0.8568	-0.0000
LSTM(4)_multi_época(16)	8.3949	250.8054	333.5891	11.8739	0.7787	0.5607
VECM(2)_1co	18.5415	596.3743	639.4035	19.6283	0.8573	-0.2795
Período de Teste não COVID **Modelos**	**mape%**	**mae**	**rmse**	**rmse%**	**acf1**	**corr**
Média Móvel (4)	1.9753	61.6957	68.1575	2.1984	0.5762	0.9211
VECM(2)_1co + LSTM(4)_log_época(3)	1.9844	60.7305	79.2743	2.6444	0.6780	0.8743
VECM(2)_1co + LSTM(4)_época(4)	2.9052	90.4782	115.5134	3.7484	0.7248	0.8637
NAIVE	4.6728	152.0820	204.8337	6.1486	0.8869	-0.0000
LSTM(4)_multi_época(16)	5.7120	181.3727	205.3756	6.4194	0.8055	0.9213
VECM(2)_1co	21.4531	697.3532	706.6242	21.6124	0.8412	0.9640

4 e Tabela 6), e as previsões variam mais (semelhantes à média móvel 4), por terem maior sensibilidade aos novos dados do que os com 20 *lags*, onde as previsões são mais alisadas (semelhantes à média móvel 20) (Figura 21). Mas a maior diferença está na correlação entre as previsões e os valores atuais do S&P500, onde os modelos com 4 *lags* têm uma correlação forte e positiva (>0.60 para os com melhor época), enquanto os modelos com 20 *lags* não apresentam correlação (≈ 0).

Analisar o impacto que o problema do vazamento de dados pode ter nos resultados é revelador da importância de o evitar, mesmo que habitualmente ocorra por desconhecimento ou involuntariamente. O modelo híbrido com vazamento obteve os piores resultados a seguir à previsão da tendência de longo prazo do VECM, independentemente de o período escolhido abranja a crise da COVID-19 ou não (MAPE: 16.5% versus 17.98% sem COVID) (Tabela 3).

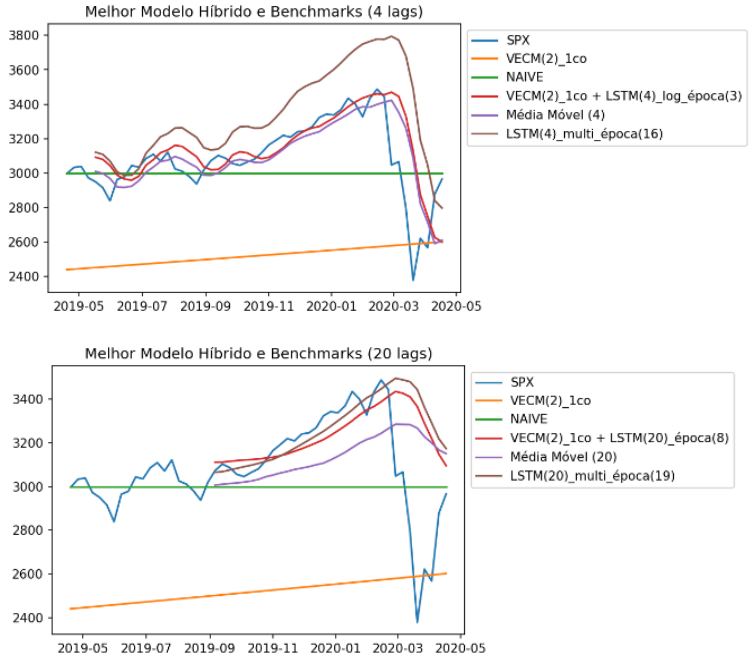

Figura 21. Gráficos da previsão dos melhores modelos híbridos de cada *lag* em comparação com os respetivos modelos de referência

Comparando os resultados para o período de teste com o período sem a crise da COVID -19 (Tabela 11 versus Tabela 13), para os modelos híbridos, os erros de previsão caiem consideravelmente com a retirada do efeito da crise. Outro efeito é que os modelos com 20 *lags* que não apresentavam correlação com o S&P500, passam a apresentar correlações superiores a 0.90, e os restantes modelos também a viram aumentar para esses valores (Tabela 13). Outra alteração, foi o melhor

modelo híbrido no período de teste ter sido o com 4 *lags* e melhor época (3) e o melhor modelo no período sem crise ter sido o com 20 *lags* e melhor época (8) com um MAPE de 3.97% versus 1.93%, respetivamente. O melhor modelo no período sem crise (20 *lags*) também foi até melhor que a média móvel 4, o que não aconteceu com o melhor modelo no período completo de teste (4 *lags*). Mas no período de teste o modelo com 4 *lags* foi melhor que todos os modelos *benckmark* (LSTM multivariados), o que não aconteceu no período do modelo com 20 *lags* onde o melhor modelo globalmente foi o LSTM multivariado (época 19) (Figura 21).

Tabela 5. Métricas de avaliação das previsões dos modelos híbridos com 20 *lags* e 30 épocas e respetivos modelos de referência, ordenados pelo mape%, no período de teste com e sem COVID-19 (até 14/02/2020)

Período de Teste Modelos	mape%	mae	rmse	rmse%	acf1	corr
NAIVE	5.5138	170.5628	230.7686	7.6259	0.8568	-0.0000
Média Móvel (20)	7.2745	213.7474	282.0899	10.5034	0.8681	-0.1287
VECM(2)_1co + LSTM(20)_log	11.8437	375.5849	405.2324	12.6783	0.8602	-0.1661
VECM(2)_1co + LSTM(20)	11.9530	378.8985	406.1308	12.7087	0.8569	-0.0589
LSTM(20)_multi	13.8138	436.0077	459.6909	14.5171	0.8619	0.0759
VECM(2)_1co	18.5415	596.3743	639.4035	19.6283	0.8573	-0.2795
Período de Teste não COVID **Modelos**	**mape%**	**mae**	**rmse**	**rmse%**	**acf1**	**corr**
Média Móvel (20)	4.2159	138.8459	155.1770	4.6579	0.7873	0.9518
NAIVE	4.6728	152.0820	204.8337	6.1486	0.8869	-0.0000
VECM(2)_1co + LSTM(20)_log	13.0946	426.9406	438.7410	13.3456	0.8312	0.9590
VECM(2)_1co + LSTM(20)	13.1883	429.7678	440.4484	13.4104	0.8249	0.9603
LSTM(20)_multi	15.6862	507.3372	508.7715	15.7164	0.5050	0.9631
VECM(2)_1co	21.4531	697.3532	706.6242	21.6124	0.8412	0.9640

Tabela 6. Métricas de avaliação das previsões dos modelos híbridos com 20 *lags* e melhores épocas e respetivos modelos de referência, ordenados pelo mape%, no período de teste com e sem COVID-19 (até 14/02/2020)

Período de Teste Modelos	mape%	mae	rmse	rmse%	acf1	corr
NAIVE	5.5138	170.5628	230.7686	7.6259	0.8568	-0.0000
VECM(2)_1co + LSTM(20)_época(8)	6.0494	169.9855	286.6732	11.0329	0.8554	0.0822
LSTM(20)_multi_época(19)	6.1425	169.7500	314.3144	12.1060	0.8679	0.0292
VECM(2)_1co + LSTM(20)_log_época(9)	6.1619	175.3228	273.4045	10.4867	0.8511	0.0895
Média Móvel (20)	7.2745	213.7474	282.0899	10.5034	0.8681	-0.1287
VECM(2)_1co	18.5415	596.3743	639.4035	19.6283	0.8573	-0.2795
Período de Teste não COVID **Modelos**	**mape%**	**mae**	**rmse**	**rmse%**	**acf1**	**corr**
LSTM(20)_multi_época(19)	1.2059	39.2908	44.7087	1.3622	0.5396	0.9589
VECM(2)_1co + LSTM(20)_época(8)	1.9253	62.9650	70.1536	2.1247	0.7621	0.9314
VECM(2)_1co + LSTM(20)_log_época(9)	2.4191	79.1787	88.9104	2.6829	0.8103	0.9261
Média Móvel (20)	4.2159	138.8459	155.1770	4.6579	0.7873	0.9518
NAIVE	4.6728	152.0820	204.8337	6.1486	0.8869	-0.0000
VECM(2)_1co	21.4531	697.3532	706.6242	21.6124	0.8412	0.9640

3.2. Discussão

No período de treino foi encontrada uma relação causal bidirecional entre os retornos do SPX e do DJI, mas no período de teste essa relação perdeu-se e passou a ser entre os retornos do SPX e do NDQ. Este resultado pode ser justificado pela transformação da economia dos EUA, que passou de uma economia essencialmente industrial, cujas empresas com maior capitalização bolsista estavam listadas tanto no SPX como no DJI, para uma economia com maior peso das empresas tecnológicas, cujas empresas com maior capitalização bolsista estão listadas tanto no SPX como no NDQ.

É de frisar a relação causal entre a TB3M e os índices, uma vez que reflete a evolução das políticas monetárias em cada período. Os resultados mostram que quando a política monetária tem um impacto grande nos mercados, como no caso durante o combate à inflação onde as taxas de juro bateram máximos históricos no período analisado (Figura 25), e no caso da crise da COVID-19, onde houve uma redução abrupta das taxas de juro (Figura 29), surge uma relação de causalidade (podendo ser bidirecional ou não), onde a TB3M passa a causar os índices. A quebra da causalidade bidirecional entre a TB3M e os índices no período excluindo as décadas de 70 e 80, evidencia que a razão plausível da variação das taxas de juro de referência causarem os retornos dos índices foi devido à importância da política monetária no contexto de elevada inflação e subida das taxas de juro para a combater. Em períodos mais calmos, apenas existe uma relação unidirecional, onde os índices causam a TB3M, talvez explicado por serem maioritariamente os agentes de mercado a definirem o preço de ambos os ativos (existindo um equilíbrio entre ambos, ou seja, se a rentabilidade de um tipo de ativo sobe em demasia em relação ao outro ou vice-versa tende a existir uma correção e aproximação dessas rentabilidades entre si), de forma que se torna evidente uma relação de equilíbrio de longo prazo no preço de ambos (num curto espaço de tempo, como no período de validação, não se encontrou essa relação), onde a variação do preço das ações precede a variação das taxas de juro. Durante períodos onde há uma *intervenção forte* por parte da reserva federal, subindo ou reduzindo as taxas de juro de referência, seja por ser num relativo curto espaço de tempo (como no caso da crise da COVID-19), ou num período mais alargado de tempo, mas persistentemente, atingindo valores de juros historicamente elevados (como no caso das décadas de 70 e 80), cria um forte impacto na rentabilidade dos ativos de dívida. O mercado tende, então, a corrigir o diferencial de retorno por variar o preço das ações. No caso de uma descida dos juros, o relativo excesso de retorno das ações, é reduzido pelo aumento do preço das ações e vice-versa em períodos de subida dos juros, originando a relação causal da TB3M aos índices.

Esta relação próxima no comportamento destas séries favorece a análise de relações de cointegração, e, a se confirmarem, justifica o uso de um modelo VECM. Na verdade, foi encontrada uma relação de cointegração onde o DJI e a TB3M explicam o SPX no longo prazo, o que está de acordo com os resultados obtidos no período de treino pelos testes de causalidade à Granger.

No período de teste, a média móvel 4 foi a que conseguiu o menor valor de MAPE% (3.74%) entre todos os modelos analisados, por ter um período curto (4) alisa pouco os dados sendo mais reativa à evolução do S&P500, por isso não deve ser dada muita relevância a esse resultado levando em conta que apenas se está a prever um valor à frente, no caso de se querer prever vários períodos à frente, dificilmente uma média móvel teria melhor resultado que os modelos criados.

O aumento considerável da correlação entre as previsões dos modelos híbridos e o S&P500 quando se retira o efeito da crise da COVID-19 (>0.90), juntamente com o decréscimo acentuado de todas as métricas de erro medidas (modelo de referência LSTM multivariado passou de um MAPE de 6.14% para 1.21% sem COVID, e o melhor modelo híbrido com 4 *lags* de 3.97% para 1.98%, e com 20 *lags* de 6.05% para 1.92%), reflete a natureza inesperada e imprevisível desta crise pandémica. Além disso, reflete também que os modelos híbridos, especialmente os com 4 *lags*, mostraram melhor resiliência ou robustez ao aparecimento da crise, do que os modelos de referência, pela variação dos erros ter sido menor que a variação dos erros dos modelos *benckmark* (LSTM multivariados).

Criar modelos a partir do logaritmo das séries é boa prática porque reduz a assimetria das séries, alterando a escala, e tende a linearizar as relações entre as séries, evidenciando-se no maior número de relações de causalidade à Granger (obtida a partir da correlação linear entre *lags* passados de outra série e valor atual da série em análise). Não parece afetar a performance do LSTM, o que indica que este capta bem os efeitos tanto lineares como não lineares entre as séries.

A normalização dos dados é um passo muito importante, pois altera em muito os resultados finais. No artigo de referência os inputs do LSTM foram normalizados pelos valores máximos e mínimos de toda a série, independentemente da separação entre períodos de treino, validação e teste (Figura 38). Isso causa vazamento de dados dos períodos de teste para o período de treino (*data leakage*). Embora o erro obtido no período de treino seja menor, no período de teste é bem maior, o que evidencia a pior *performance* do modelo em relação ao que poderia ser se os dados fossem corretamente normalizados (ver medidas de erro Tabela 10).

Também ficou evidente que a técnica de escolher a época com menor erro no período de validação durante o treino (*early stopping*), é uma técnica eficaz para se evitar o ajustamento em excesso dos pesos do modelo ao período de teste (*overfitting)* e se conseguir manter bons resultados de previsão com dados fora da amostra de treino, provando a melhor capacidade de generalização das relações existentes nos dados.

O MAPE não é uma medida de erro indicada de se usar durante a fase de treino dos LSTM porque se os resíduos do VECM forem nulos, não pode ser calculada (por se dividir por 0) ou obtém-se valores muito elevados, se forem muito próximos de 0. Em termos de comparabilidade com o artigo de referência, essa métrica pode ser usada, porque o seu valor é calculado com base no valor original da série do S&P500, e as previsões foram transformadas de volta a essa escala de valores.

Conclusões

O objetivo global desta dissertação foi sobejamente cumprido. Conseguiu-se construir um modelo híbrido entre um VECM e um LSTM e aplicar na previsão do S&P500 com resultados satisfatórios, ao mesmo tempo que foi justificada a escolha tanto dos parâmetros como das variáveis, com evidências estatísticas da sua utilidade e significância, até mesmo no contexto de diferentes políticas monetárias.

Demonstrou-se estatisticamente, através dos testes de causalidade à Granger para diferentes períodos, a influência que a política monetária tem nas séries estudadas. Concluindo-se que quando existiu uma atuação forte por parte da FED, tanto nas décadas de 70 e 80 (período com elevada inflação e combate pela FED com o aumento das taxas de juro) como durante a crise da COVID-19 (com a queda abrupta das taxas de juro) isso contribuiu para causar os retornos do S&P500 e restantes índices usados.

O modelo VECM foi capaz de captar a tendência de longo prazo entre o S&P500, o Dow Jones e as taxas de juro dos bilhetes do tesouro americano a 3 meses, por revelar uma relação de cointegração entre elas, pela metodologia de Johansen, e confirmar os resultados dos testes de causalidade à Granger, onde o S&P500 é causado por essas variáveis.

O algoritmo LSTM conseguiu corrigir os resíduos do VECM e captar os efeitos não lineares existentes, reduzindo-os consideravelmente de um MAPE de 18.5% para um de 4.0%. Sendo ainda mais eficaz no período sem a crise da COVID-19 onde o MAPE do melhor modelo foi de 1.9%.

Uma boa especificação dos modelos é crucial para se obterem melhores resultados. Ao mudar apenas o modelo VECM para um com constante irrestrita, o que está em harmonia com o comportamento das séries (por apresentarem raiz unitária), obteve-se um erro MAPE 10 p.p. (pontos percentuais) inferior ao do artigo de referência (28%). Ao se normalizar os dados corretamente, e evitar-se o vazamento de dados, diminui-se o MAPE em mais 6 p.p.. Por se diminuir o número de épocas, e evitar-se o *overfitting*, o MAPE foi reduzido em mais 7 p.p.. O efeito da logaritmização das séries não alterou significativamente os resultados, mas se for usado o modelo com menor MAPE, este foi reduzido em mais 1 p.p.. No total conseguiu-se reduzir o MAPE em 24 p.p., ou em 86%.

Na globalidade, estes modelos tendem a superar os modelos de referência (LSTM multivariados). Dada a previsão alargada (7 anos) realizada no modelo VECM, e se ter apenas usado uma previsão para o período seguinte no LSTM, os resultados são satisfatórios, pois conseguiu-se melhorar em muito o modelo do artigo de referência, mesmo no contexto de uma crise pandémica, onde os efeitos são

imprevisíveis por serem novos e únicos, o que aumentou em muito os erros de previsão, em comparação com o período sem essa crise.

Usar períodos que não incluam a crise pandémica e, com isso, retirar o seu efeito exógeno e imprevisível, pode melhorar os resultados. Por outro lado, obter um modelo robusto por se captar a tendência de muito longo prazo, também é benéfico ao reduzir o impacto de eventos extremos e inesperados no curto-prazo. Mas mais que acertar numa previsão, o objetivo é captar a dinâmica existente entre as variáveis, e com isso, prever a tendência dos seus valores futuros. Pelo modelo híbrido alcançado, penso que esse objetivo foi cumprido, ao demonstrar que a tendência de muito longo prazo é captada pelo modelo VECM, e a dinâmica de mais curto-prazo modelada pelo LSTM. Os modelos com 4 *lags* mostraram-se mais robustos à crise da COVID-19, com menor variação do erro de previsão (MAPE de 3.97% versus 1.98% sem COVID), em comparação com os modelos com 20 *lags* (6.05% versus 1.92%) e até mais com o *benchmark* (6.14% versus 1.21%).

As respostas às sub-perguntas de investigação são:

1.1. Quais as variáveis mais significativas na previsão do preço do S&P500?

Foi revelado, pela revisão da literatura, que a volatilidade do S&P500 (p.e. medida pelo VIX) é o fator mais relevante na previsão do índice. Outras variáveis importantes também são o *spread* entre as *yields* empresariais com rating AAA (da Moody's, EUA) e os bilhetes do tesouro (US Treasury Bills) a 10 anos, oscilador estocástico, Williams %R, indicador MACD (Moving Average Convergence Divergence), *spread* entre o preço máximo e mínimo do dia, retornos passados do índice S&P500 e do DAX de acordo com Nevasalmi (2020). Dionisio et al (2011) concluiu que as variáveis financeiras como a *dividend yield* e o rácio rendimento/preço (P/E) são as mais relevantes.

1.2. Qual a importância da estrutura (construção das camadas internas) de um algoritmo de *deep learning* (LSTM) nos resultados da previsão?

Ao se aplicar a metodologia de corte ou poda, proposta por Lee, Ajanthan e Torr (2019), consegue-se reduzir o número de conexões, e por consequência, o número de pesos a estimar, na ordem dos 95% com pouca ou nenhuma perda de acurácia. Isso demonstra que, mais importante que ter todas as conexões, é ter as que mais contribuem para a previsão. Assim, é possível obter modelos com LSTM muito mais pequenos e fáceis de treinar e, com isso, ainda eliminar o risco de se treinar em excesso (*overfitting*) e perder capacidade de generalização.

1.3. Como unir os diferentes modelos ou metodologias usadas num único modelo de previsão?

Apesar de existir inúmeros métodos de montar diferentes modelos num único, a simples adição dos resultados de previsão dos resíduos do VECM pelo LSTM com a previsão do nível do S&P500 pelo VECM, mostrou ser suficiente e satisfatória. Esta simples metodologia pode ser aplicada para modelos treinados em sequência, como no caso desta dissertação.

1.4. Pode um modelo híbrido ser capaz de produzir uma previsão que desafie a hipótese de eficiência dos mercados?

Pela leitura de outros artigos científicos, é possível conseguir construir estratégias que violem a hipótese dos mercados eficientes, tendo sido encontrados modelos que obtiveram retornos muito acima da média de mercado (Secção 1.7).

Recomenda-se a realização de mais estudos no sentido de analisar como se poderá usar este tipo de modelos híbridos num contexto de avaliar o efeito de políticas monetárias (uma alteração da taxa de juro de referência que impacto tem nas restantes variáveis). Se o objetivo for o de prever o efeito a médio/longo prazo, prever com o LSTM apenas para o período seguinte, não serve o propósito. Assim, é recomendável a realização de mais experiências onde se use um VECM e um LSTM para se prever múltiplos períodos à frente, e com isso avaliar a sua utilidade no contexto referido.

Outra sugestão de trabalho futuro é adicionar novas variáveis ao LSTM, como, por exemplo, dados do VIX, ou informação categórica do mês ou semana do ano, entre outras, e perceber quais contribuem mais para a previsão final, e analisar se os resíduos do modelo VECM se mantêm relevantes ou não em junção com essas outras variáveis. Posteriormente, também é possível desenvolver estratégias de negociação, com decisões de compra e venda, a partir do output dos modelos híbridos desenvolvidos e testar se os resultados obtidos desafiam a hipótese dos mercados eficientes.

Usar dados com menor frequência que a semanal, como dados diários, permite a um LSTM ter mais dados para treinar e, possivelmente, captar melhor ou com mais detalhe, as relações existentes. Isso também pode ser feito como trabalho de investigação futuro, além de se poder testar diferentes arquiteturas do LSTM, como variar o número de camadas e/ou células em cada uma, ou alterar a filosofia de montagem de modelos híbridos e combinar com outros tipos de algoritmos de *Machine Learning*, como as árvores de decisão aleatórias (*Random Forest*) ou o PCA (*Principal Component Analysis*), na seleção e escolha das variáveis previsionais mais relevantes antes de serem usadas como inputs no VECM e/ou LSTM, por exemplo.

Referências Bibliográficas

Agbonlahor, O. (2014) 'The Impact of Monetary Policy on the Economy of the United Kingdom: a Vector Error Correction Model (VECM)', *European Scientific Journal*, 10(16), pp. 1857–7881. Disponível em: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.877.8639&rep=rep1&type=pdf.

Anderson, R. G., Hoffman, D. L. and Rasche, R. H. (2002) 'A vector error-correction forecasting model of the U.S. economy', *Journal of Macroeconomics*, 24(4), pp. 613–614. doi: 10.1016/S0164-0704(02)00070-8.

Bernanke, B., Boivin, J. and Eliasz, P. (2004) *Measuring the Effects of Monetary Policy: A Factor-Augmented Vector Autoregressive (FAVAR) Approach*. Cambridge, MA. doi: 10.3386/w10220.

Bernanke, B. S. (2020) 'The New Tools of Monetary Policy American Economic Association Presidential Address', *American Economic Review*, 110(4), pp. 943–83. Disponível em: https://www.aeaweb.org/articles/pdf/doi/10.1257/aer.110.4.943.

Brooks, C. (2014) *Introdutory Econometrics for Finance*. 3rd edn. Cambridge University Press.

Chalvatzis, C. and Hristu-Varsakelis, D. (2020) 'High-performance stock index trading via neural networks and trees', *Applied Soft Computing Journal*. Elsevier B.V., 96, p. 106567. doi: 10.1016/j.asoc.2020.106567.

Chaudhary, M. (2020) *Activation Functions: Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax*, *Medium2*. Available at: https://medium.com/@cmukesh8688/activation-functions-sigmoid-tanh-relu-leaky-relu-softmax-50d3778dcea5.

Dionisio, A., Menezes, R. and Mendes, D. A. (2007) 'On the integrated behaviour of non-stationary volatility in stock markets', *Physica A: Statistical Mechanics and its Applications*, 382(1), pp. 58–65. doi: 10.1016/j.physa.2007.02.008.

Dionisio, A. T. *et al.* (2011) 'Linear and Nonlinear Dependence Models of Stock Market Returns', *SSRN Electronic Journal*, pp. 1–19. doi: 10.2139/ssrn.668001.

Emrouznejad, A., Rostami-Tabar, B. and Petridis, K. (2016) 'A novel ranking procedure for forecasting approaches using Data Envelopment Analysis', *Technological Forecasting and Social Change*. The Authors, 111(August), pp. 235–243. doi: 10.1016/j.techfore.2016.07.004.

Fama, E. F. (1970) 'Efficient Capital Markets: A Review of Theory and Empirical Work', *The Journal of Finance*, 25(2), p. 383. doi: 10.2307/2325486.

Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*, *Angewandte Chemie International Edition, 6(11), 951–952.* Massachusetts Institute of Technology. Available at: www.deeplearningbook.org.

Hochreiter, S. and Schmidhuber, J. (1997) 'Long Short-Term Memory', *Neural Computation*, 9(8), pp. 1735–1780. doi: 10.1162/neco.1997.9.8.1735.

Holtemoller, O. (2004) 'A monetary vector error correction model of the Euro area and implications for monetary policy', *Empirical Economics*, 29(3). doi: 10.1007/s00181-004-0198-4.

Isfan, M., Menezes, R. and Mendes, D. A. (2010) 'Forecasting the portuguese stock market time series by using artificial neural networks', *Journal of Physics: Conference Series*, 221. doi: 10.1088/1742-6596/221/1/012017.

Jansen, S. (2020) *Machine learning for algorithmic trading*. Second Edi. Packt Publishing.

Johansen, S. and Juselius, K. (1990) 'Maximum Likelihood Estimation and Inference on Cointegration — With Applications To the Demand for Money', *Oxford Bulletin of Economics and Statistics*, 52(2), pp. 169–210. doi: 10.1111/j.1468-0084.1990.mp52002003.x.

Kingma, D. P. and Ba, J. L. (2015) 'Adam: A method for stochastic optimization', *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15.

Kliensen, L. K. (1999) *Models and Monetary Policy: More Science Than Art?*, *Federal Reserve Bank of St. Louis*. Available at: https://www.stlouisfed.org/publications/regional-economist/january-1999/models-and-monetary-policy-more-science-than-art.

Kourentzes, N., Barrow, D. and Petropoulos, F. (2019) 'Another look at forecast selection and combination: Evidence from forecast pooling', *International Journal of Production Economics*, 209, pp. 226–235. doi: 10.1016/j.ijpe.2018.05.019.

Kumar, D., Sarangi, P. K. and Verma, R. (2021) 'A systematic review of stock market prediction using machine learning and statistical techniques', *Materials Today: Proceedings*. Elsevier Ltd., (xxxx). doi: 10.1016/j.matpr.2020.11.399.

Lee, N., Ajanthan, T. and Torr, P. H. S. (2019) 'SNIP: single-shot network pruning based on connection sensitivity', in *Conference ICLR*.

Lima, E. J. A., Luduvice, F. and Tabak, B. M. (2006) *Forecasting Interest Rates: an application for Brazil*. Available at: https://ideas.repec.org/p/bcb/wpaper/120.html.

Ma, J. *et al.* (2019) 'Spatiotemporal Prediction of PM2.5 Concentrations at Different Time Granularities Using IDW-BLSTM', *IEEE Access*, 7, pp. 107897–107907. doi: 10.1109/ACCESS.2019.2932445.

Makridakis, S., Spiliotis, E. and Assimakopoulos, V. (2020) 'The M4 Competition: 100,000 time series and 61 forecasting methods', *International Journal of Forecasting*. Elsevier B.V., 36(1), pp. 54–74. doi: 10.1016/j.ijforecast.2019.04.014.

Mendes, D. A., Ferreira, N. R. B. and Mendes, V. M. P. (2020) 'Comparative multivariate forecast performance for the G7 Stock Markets: VECM Models vs deep learning LSTM neural networks', in *Conference CARMA*, pp. 163–171. doi: 10.4995/carma2020.2020.11616.

Mills, T. C. (2019) *Applied Time Series Analysis: A Practical Guide to Modeling and Forecasting*, *Elsevier*.

Mittal, A. (2019) *Understanding RNN and LSTM*, *Medium*. Disponível em: https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e.

Nevasalmi, L. (2020) 'Forecasting multinomial stock returns using machine learning methods', *Journal of Finance and Data Science*. Elsevier Ltd, 6, pp. 86–106. doi: 10.1016/j.jfds.2020.09.001.

Pedamkar, P. (2021) *Ensemble Techniques*. Disponível em: https://www.educba.com/ensemble-techniques/ (Accessed: 22 March 2021).

Prado, M. L. de (2018) *Advances in Financial Machine Learning*. John Wiley & Sons Inc.

Saifan, R. *et al.* (2020) 'Investigating Algorithmic Stock Market Trading using Ensemble Machine Learning Methods', *Informatica*, 44(3). doi: 10.31449/inf.v44i3.2904.

Sanghvirajit (2020) *A Complete Guide to Adam and RMSprop Optimizer*, *Analytics Vidhya*. Disponível em: https://medium.com/analytics-vidhya/a-complete-guide-to-adam-and-rmsprop-optimizer-75f4502d83be.

Sewell, M. (2011) 'Characterization of financial time series', *Rn*, 11(01), p. 01.

Sezer, O. B., Gudelek, M. U. and Ozbayoglu, A. M. (2020) 'Financial time series forecasting with deep learning: A systematic literature review: 2005–2019', *Applied Soft Computing Journal*. Elsevier B.V., 90, p. 106181. doi: 10.1016/j.asoc.2020.106181.

Smyl, S. (2020) 'A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting', *International Journal of Forecasting*. Elsevier B.V., 36(1), pp. 75–85. doi: 10.1016/j.ijforecast.2019.03.017.

Soni, D. (2019) *Data Leakage in Machine Learning*, *towards data science, medium*. Disponível em: https://towardsdatascience.com/data-leakage-in-machine-learning-10bdd3eec742.

Yang, C., Zhai, J. and Tao, G. (2020) 'Deep Learning for Price Movement Prediction Using Convolutional Neural Network and Long Short-Term Memory', *Mathematical Problems in Engineering*, 2020, pp. 1–13. doi: 10.1155/2020/2746845.

Yildirim, D. C., Toroslu, I. H. and Fiore, U. (2021) 'Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators', *Financial Innovation*, 7(1), p. 1. doi: 10.1186/s40854-020-00220-2.

Yujun, Y., Yimei, Y. and Jianhua, X. (2020) 'A hybrid prediction method for stock price using LSTM and ensemble EMD', *Complexity*, 2020. doi: 10.1155/2020/6431712.

Zhang, S., Lowinger, T. C. and Tang, J. (2007) 'The Monetary Exchange Rate Model: Long-run, Short-run, and Forecasting Performance', *Journal of Economic Integration*, 22(2), pp. 397–406. Disponível em: https://www.jstor.org/stable/23001103.

Zhou, Z.-H. (2012) *Ensemble Methods Foundations and Algorithms*. Edited by R. Herbrich and T. Graepel. CRC Press, Taylor & Francis Group, LLC.

ANEXO I

Apresentação Gráfica dos Resultados

A. Análise Exploratória dos dados

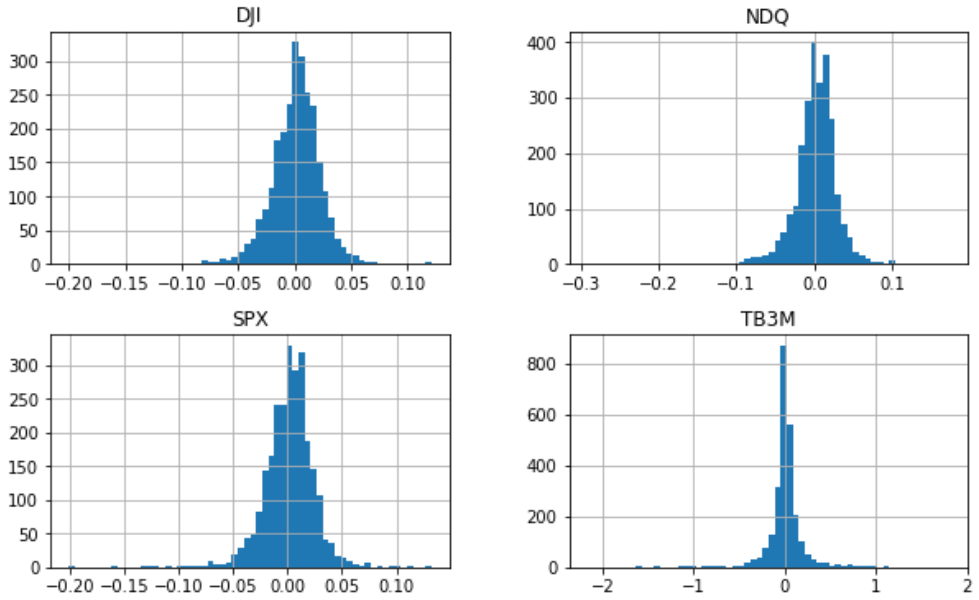

Figura 22. Histogramas dos log-retornos de cada série

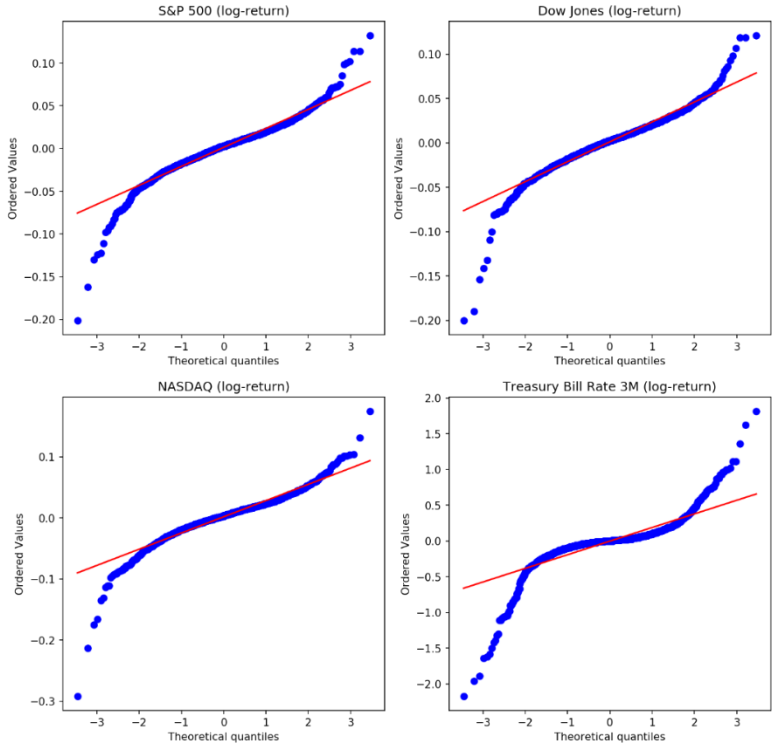

Figura 23. Gráfico quantil/quantil da distribuição dos log-retornos em comparação à distribuição normal

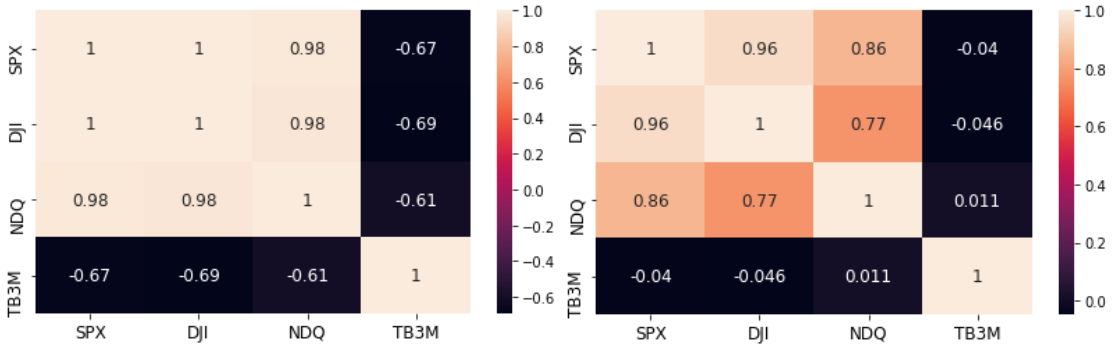

Figura 24. Matrix de correlação de Pearson das séries em nível (esquerda) e dos log-retornos (direita)

Tabela 7. Testes de Estacionariedade para as Séries em nível

Séries em nível	ADF (valor-p)	PP (valor-p)	KPSS (valor-p)
SPX	0.997	0.997	0.000
DJI	0.998	0.995	0.000
NDQ	0.999	0.999	0.000
TB3M	0.624	0.434	0.000
Hipótese nula (H0)	**Existe raiz unitária**	**Existe raiz unitária**	Fracamente Estacionário
Hipótese Alternativa (H1)	Fracamente Estacionário	Fracamente Estacionário	**Existe raiz unitária**

Tabela 8. Testes de estacionariedade, normalidade e valores de kurtosis e enviesamento dos retornos das séries

Primeiras diferenças	ADF (valor-p)	PP (valor-p)	KPSS (valor-p)	Normal Test (valor-p)	Kurtosis	Enviesamento (skewness)
SPX (log)	0.000	0.000	0.10	0.000	6.44	-0.69
DJI (log)	0.000	0.000	0.10	0.000	7.29	-0.76
NDQ (log)	0.000	0.000	0.10	0.000	9.55	-1.08
TB3M	0.000	0.000	0.10	0.000	20.29	-1.36
Hipótese nula (H0)	Existe raiz unitária	Existe raiz unitária	**Fracamente Estacionário**	Distribuição Normal		
Hipótese Alternativa (H1)	**Fracamente Estacionário**	**Fracamente Estacionário**	Existe raiz unitária	**Não normalidade**		

B. Causalidade à Granger

Lags estatisticamente significativos com $\alpha = 0.05$ (X causa Y)

```
Y=SPX,X=SPX,Lags=[]
Y=DJI,X=SPX,Lags=[12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
Y=NDQ,X=SPX,Lags=[]
Y=TB3M,X=SPX,Lags=[7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
Y=SPX,X=DJI,Lags=[2, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
Y=DJI,X=DJI,Lags=[]
Y=NDQ,X=DJI,Lags=[]
Y=TB3M,X=DJI,Lags=[]
Y=SPX,X=NDQ,Lags=[1, 7]
Y=DJI,X=NDQ,Lags=[1, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24]
Y=NDQ,X=NDQ,Lags=[]
Y=TB3M,X=NDQ,Lags=[6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
Y=SPX,X=TB3M,Lags=[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 20, 23, 24]
Y=DJI,X=TB3M,Lags=[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15]
Y=NDQ,X=TB3M,Lags=[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 23]
Y=TB3M,X=TB3M,Lags=[]
```

	SPX_x	DJI_x	NDQ_x	TB3M_x
SPX_y	1.0000	0.0033	0.0382	0.0004
DJI_y	0.0008	1.0000	0.0047	0.0008
NDQ_y	0.0510	0.2316	1.0000	0.0011
TB3M_y	0.0041	0.0642	0.0032	1.0000

Valores mínimos do p-value para todos os Lags de cada combinação de 2 séries

Rejeitar não causalidade se
P-value <= 0.05 (então X causa Y)

Figura 25. Causalidade à Granger entre os log-retornos das séries no período de teste durante as décadas de 70 e 80

Lags estatisticamente significativos com $\alpha = 0.05$ (X causa Y)

```
Y=SPX,X=SPX,Lags=[]
Y=DJI,X=SPX,Lags=[]
Y=NDQ,X=SPX,Lags=[4, 5, 6, 7, 11]
Y=TB3M,X=SPX,Lags=[5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]
Y=SPX,X=DJI,Lags=[7, 11, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
Y=DJI,X=DJI,Lags=[]
Y=NDQ,X=DJI,Lags=[5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
Y=TB3M,X=DJI,Lags=[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15]
Y=SPX,X=NDQ,Lags=[]
Y=DJI,X=NDQ,Lags=[]
Y=NDQ,X=NDQ,Lags=[]
Y=TB3M,X=NDQ,Lags=[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
Y=SPX,X=TB3M,Lags=[]
Y=DJI,X=TB3M,Lags=[]
Y=NDQ,X=TB3M,Lags=[]
Y=TB3M,X=TB3M,Lags=[]
```

	SPX_x	DJI_x	NDQ_x	TB3M_x
SPX_y	1.0000	0.0013	0.2245	0.2021
DJI_y	0.0532	1.0000	0.1144	0.0692
NDQ_y	0.0307	0.0024	1.0000	0.1839
TB3M_y	0.0089	0.0252	0.0000	1.0000

Valores mínimos do p-value para todos os Lags de cada combinação de 2 séries

Rejeitar não causalidade se
P-value <= 0.05 (então X causa Y)

Figura 26. Causalidade à Granger entre os log-retornos das séries no período de treino excluindo as décadas de 70 e 80

Lags estatisticamente significativos com $\alpha = 0.05$ (X causa Y)

```
Y=SPX,X=SPX,Lags=[]
Y=DJI,X=SPX,Lags=[]
Y=NDQ,X=SPX,Lags=[17]
Y=TB3M,X=SPX,Lags=[]
Y=SPX,X=DJI,Lags=[]
Y=DJI,X=DJI,Lags=[]
Y=NDQ,X=DJI,Lags=[]
Y=TB3M,X=DJI,Lags=[]
Y=SPX,X=NDQ,Lags=[]
Y=DJI,X=NDQ,Lags=[]
Y=NDQ,X=NDQ,Lags=[]
Y=TB3M,X=NDQ,Lags=[19, 20]
Y=SPX,X=TB3M,Lags=[]
Y=DJI,X=TB3M,Lags=[]
Y=NDQ,X=TB3M,Lags=[]
Y=TB3M,X=TB3M,Lags=[]
```

	SPX_x	DJI_x	NDQ_x	TB3M_x
SPX_y	1.0000	0.2056	0.3839	0.3386
DJI_y	0.1058	1.0000	0.2472	0.4193
NDQ_y	0.0458	0.0861	1.0000	0.4086
TB3M_y	0.1539	0.0589	0.0299	1.0000

Valores mínimos do p-value para todos os *Lags* de cada combinação de 2 séries

Rejeitar não causalidade se
P-value <= 0.05 (então X causa Y)

Figura 27. Causalidade à Granger entre os log-retornos das séries no período de validação

Lags estatisticamente significativos com α = 0.05 (X causa Y)

```
Y=SPX,X=SPX,Lags=[]
Y=DJI,X=SPX,Lags=[]
Y=NDQ,X=SPX,Lags=[1, 9, 10, 11, 12]
Y=TB3M,X=SPX,Lags=[1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Y=SPX,X=DJI,Lags=[]
Y=DJI,X=DJI,Lags=[]
Y=NDQ,X=DJI,Lags=[]
Y=TB3M,X=DJI,Lags=[1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Y=SPX,X=NDQ,Lags=[1, 6, 9, 10, 11, 12]
Y=DJI,X=NDQ,Lags=[1]
Y=NDQ,X=NDQ,Lags=[]
Y=TB3M,X=NDQ,Lags=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Y=SPX,X=TB3M,Lags=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Y=DJI,X=TB3M,Lags=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Y=NDQ,X=TB3M,Lags=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Y=TB3M,X=TB3M,Lags=[]
```

	SPX_x	DJI_x	NDQ_x	TB3M_x
SPX_y	1.0000	0.2793	0.0005	0.0
DJI_y	0.2238	1.0000	0.0427	0.0
NDQ_y	0.0003	0.0711	1.0000	0.0
TB3M_y	0.0000	0.0000	0.0000	1.0

Valores mínimos do p-value para todos os Lags de cada combinação de 2 séries

Rejeitar não causalidade se
P-value <= 0.05 (então X causa Y)

Figura 28. Causalidade à Granger entre os log-retornos das séries no período de teste

Lags estatisticamente significativos com α = 0.05 (X causa Y)

```
Y=SPX,X=SPX,Lags=[]
Y=DJI,X=SPX,Lags=[9, 10, 11, 12]
Y=NDQ,X=SPX,Lags=[4, 6, 7, 8, 9, 10, 11, 12]
Y=TB3M,X=SPX,Lags=[10, 11, 12]
Y=SPX,X=DJI,Lags=[8, 9, 10, 11, 12]
Y=DJI,X=DJI,Lags=[]
Y=NDQ,X=DJI,Lags=[8, 9, 10, 11, 12]
Y=TB3M,X=DJI,Lags=[10, 11, 12]
Y=SPX,X=NDQ,Lags=[1, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Y=DJI,X=NDQ,Lags=[9, 10, 11, 12]
Y=NDQ,X=NDQ,Lags=[]
Y=TB3M,X=NDQ,Lags=[10, 11, 12]
Y=SPX,X=TB3M,Lags=[]
Y=DJI,X=TB3M,Lags=[12]
Y=NDQ,X=TB3M,Lags=[]
Y=TB3M,X=TB3M,Lags=[]
```

	SPX_x	DJI_x	NDQ_x	TB3M_x
SPX_y	1.0	0.0	0.0	0.0915
DJI_y	0.0	1.0	0.0	0.0003
NDQ_y	0.0	0.0	1.0	0.2161
TB3M_y	0.0	0.0	0.0	1.0000

Valores mínimos do p-value para todos os Lags de cada combinação de 2 séries

Rejeitar não causalidade se
P-value <= 0.05 (então X causa Y)

Figura 29. Causalidade à Granger entre os log-retornos das séries no período de teste sem crise da COVID-19

```
Y=SPX,X=SPX,Lags=[]
Y=DJI,X=SPX,Lags=[2]
Y=NDQ,X=SPX,Lags=[1, 2]
Y=TB3M,X=SPX,Lags=[]
Y=SPX,X=DJI,Lags=[2]
Y=DJI,X=DJI,Lags=[]
Y=NDQ,X=DJI,Lags=[1, 2]
Y=TB3M,X=DJI,Lags=[]
Y=SPX,X=NDQ,Lags=[1, 2]
Y=DJI,X=NDQ,Lags=[1, 2]
Y=NDQ,X=NDQ,Lags=[]
Y=TB3M,X=NDQ,Lags=[]
Y=SPX,X=TB3M,Lags=[2]
Y=DJI,X=TB3M,Lags=[2]
Y=NDQ,X=TB3M,Lags=[2]
Y=TB3M,X=TB3M,Lags=[]
```

	SPX_x	DJI_x	NDQ_x	TB3M_x
SPX_y	1.0000	0.0088	0.0001	0.0
DJI_y	0.0043	1.0000	0.0057	0.0
NDQ_y	0.0000	0.0021	1.0000	0.0
TB3M_y	0.3251	0.3476	0.2787	1.0

Valores mínimos do p-value para todos os Lags de cada combinação de 2 séries

Rejeitar não causalidade se
P-value <= 0.05 (então X causa Y)

Figura 30. Causalidade à Granger entre os log-retornos das séries no período de teste da crise da COVID-19

C. Modelo VECM

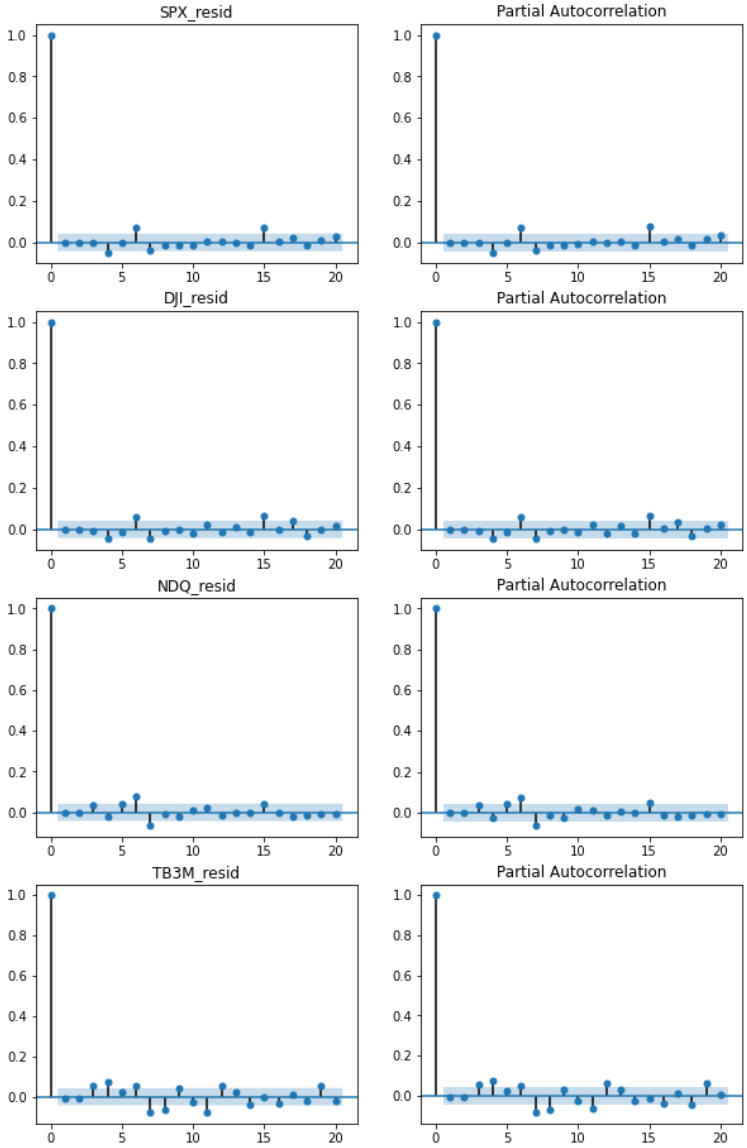

Figura 31. Gráfico da autocorrelação (direita) e da autocorrelação parcial (esquerda) para os resíduos do modelo VECM(2) para cada série

```
teste de Ljung-Box de independência dos resíduos:
SPX_resid :  [0.89082203 0.98907469 0.99652199 0.28704037 0.41526691 0.01402427
 0.00654929 0.01065474 0.01513796 0.02252448]
DJI_resid :  [0.88345896 0.98867057 0.97751064 0.29940695 0.37865593 0.05213767
 0.02248907 0.03771504 0.06001522 0.07225272]
NDQ_resid :  [9.02223432e-01 9.92416745e-01 3.79257273e-01 3.79304188e-01
 1.35114137e-01 1.48444637e-03 6.14126487e-05 1.34325564e-04
 1.84374790e-04 3.38357993e-04]
TB3M_resid :  [8.07881578e-01 8.73382812e-01 8.87686576e-02 1.00832578e-03
 1.33361274e-03 2.36021004e-04 2.16223007e-06 1.32693246e-07
 6.17945487e-08 7.84716275e-08]
```

Figura 32. Valores-p do teste de independência dos resíduos Ljung-Box para cada desfasamento (1 a 10) de cada série

```
SPX_resid
  ADF pvalue: 0.0
  Phillips Perron p-value: 0.0
  KPSS p-value: 0.1
DJI_resid
  ADF pvalue: 0.0
  Phillips Perron p-value: 0.0
  KPSS p-value: 0.1
NDQ_resid
  ADF pvalue: 0.0
  Phillips Perron p-value: 0.0
  KPSS p-value: 0.1
TB3M_resid
  ADF pvalue: 0.0
  Phillips Perron p-value: 0.0
  KPSS p-value: 0.1
```

```
Teste ARCH de heterocedasticidade:
SPX_resid :  5.084002044354739e-45
DJI_resid :  9.254173705354767e-31
NDQ_resid :  5.47668573956977e-27
TB3M_resid :  2.8778890961176013e-80
```

Figura 33. Valores-p dos testes de estacionariedade (esquerda) e de heterocedasticidade com 5 *lags* (direita) aos resíduos do modelo VECM(2)

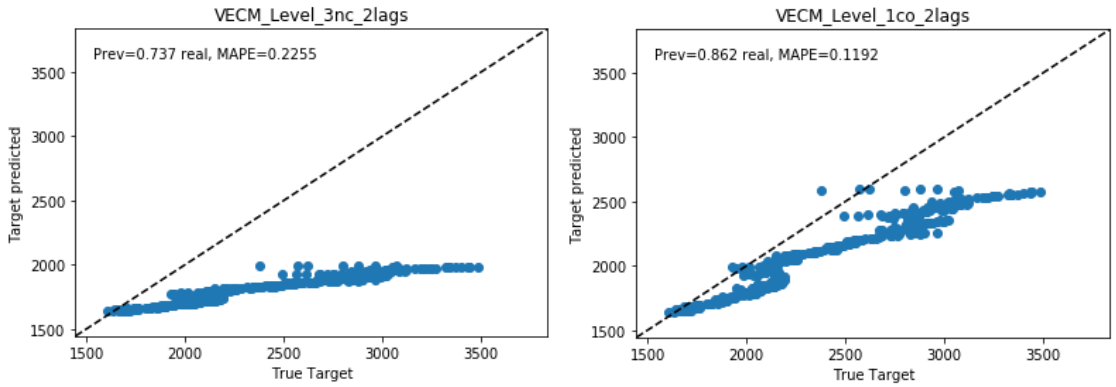

Figura 34. Previsões do VECM(2)_3nc (esquerda) e do VECM(2)_1co (direita) versus valores reais (com o Beta da regressão e o MAPE)

Det. terms outside the coint. relation & lagged endog. parameters for equation SPX

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0120	0.004	3.307	0.001	0.005	0.019
L1.SPX	-0.0437	0.093	-0.470	0.639	-0.226	0.139
L1.DJI	-0.0200	0.075	-0.266	0.790	-0.168	0.128
L1.NDQ	0.0048	0.035	0.136	0.892	-0.064	0.073
L1.TB3M	0.0004	0.002	0.206	0.837	-0.004	0.004
L2.SPX	-0.1079	0.092	-1.172	0.241	-0.288	0.073
L2.DJI	0.0776	0.075	1.033	0.301	-0.070	0.225
L2.NDQ	0.0598	0.034	1.744	0.081	-0.007	0.127
L2.TB3M	-0.0036	0.002	-1.746	0.081	-0.008	0.000

Det. terms outside the coint. relation & lagged endog. parameters for equation DJI

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0141	0.004	3.874	0.000	0.007	0.021
L1.SPX	-0.0067	0.093	-0.073	0.942	-0.189	0.175
L1.DJI	-0.0594	0.075	-0.789	0.430	-0.207	0.088
L1.NDQ	0.0004	0.035	0.012	0.991	-0.068	0.069
L1.TB3M	-0.0009	0.002	-0.421	0.674	-0.005	0.003
L2.SPX	-0.1119	0.092	-1.216	0.224	-0.292	0.068
L2.DJI	0.0811	0.075	1.080	0.280	-0.066	0.228
L2.NDQ	0.0675	0.034	1.970	0.049	0.000	0.135
L2.TB3M	-0.0043	0.002	-2.101	0.036	-0.008	-0.000

Figura 35. Equações VECM dos parâmetros do SPX (esquerda) e do DJI (direita)

Det. terms outside the coint. relation & lagged endog. parameters for equation NDQ

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0142	0.004	3.179	0.001	0.005	0.023
L1.SPX	0.0145	0.114	0.127	0.899	-0.209	0.238
L1.DJI	0.0566	0.092	0.614	0.539	-0.124	0.237
L1.NDQ	-8.375e-05	0.043	-0.002	0.998	-0.084	0.084
L1.TB3M	-0.0015	0.002	-0.588	0.556	-0.006	0.003
L2.SPX	-0.0714	0.113	-0.634	0.526	-0.292	0.149
L2.DJI	0.0628	0.092	0.683	0.495	-0.117	0.243
L2.NDQ	0.0674	0.042	1.606	0.108	-0.015	0.150
L2.TB3M	-0.0011	0.003	-0.456	0.648	-0.006	0.004

Det. terms outside the coint. relation & lagged endog. parameters for equation TB3M

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.1135	0.038	2.965	0.003	0.038	0.188
L1.SPX	0.4108	0.977	0.421	0.674	-1.503	2.325
L1.DJI	0.1038	0.791	0.131	0.896	-1.446	1.654
L1.NDQ	-0.2958	0.368	-0.804	0.421	-1.017	0.425
L1.TB3M	0.0838	0.021	3.916	0.000	0.042	0.126
L2.SPX	1.0990	0.967	1.136	0.256	-0.796	2.994
L2.DJI	-0.6064	0.789	-0.769	0.442	-2.153	0.940
L2.NDQ	-0.1659	0.360	-0.461	0.645	-0.871	0.540
L2.TB3M	0.0588	0.021	2.742	0.006	0.017	0.101

Figura 36. Equações VECM dos parâmetros do NDQ (esquerda) e do TB3M (direita)

Loading coefficients (alpha) for equation SPX

	coef	std err	z	P>|z|	[0.025	0.975]
ec1	0.0190	0.006	2.986	0.003	0.007	0.031

Loading coefficients (alpha) for equation DJI

	coef	std err	z	P>|z|	[0.025	0.975]
ec1	0.0226	0.006	3.555	0.000	0.010	0.035

Loading coefficients (alpha) for equation NDQ

	coef	std err	z	P>|z|	[0.025	0.975]
ec1	0.0226	0.008	2.897	0.004	0.007	0.038

Loading coefficients (alpha) for equation TB3M

	coef	std err	z	P>|z|	[0.025	0.975]
ec1	0.2034	0.067	3.044	0.002	0.072	0.334

Cointegration relations for loading-coefficients-column 1

	coef	std err	z	P>|z|	[0.025	0.975]
beta.1	1.0000	0	0	0.000	1.000	1.000
beta.2	-1.0807	0.103	-10.484	0.000	-1.283	-0.879
beta.3	0.0096	0.080	0.121	0.904	-0.147	0.166
beta.4	-0.0437	0.008	-5.653	0.000	-0.059	-0.029

Relação de cointegração:

SPX = alfa x [beta.2(DJI) + beta.3(NDQ) + beta.4(TB3M)]

SPX = 0.019 x [1.0807(DJI) + 0.0437(TB3M)]

Figura 37. Coeficientes alpha do VECM (esquerda) e relação de cointegração (direita)

D. LSTM

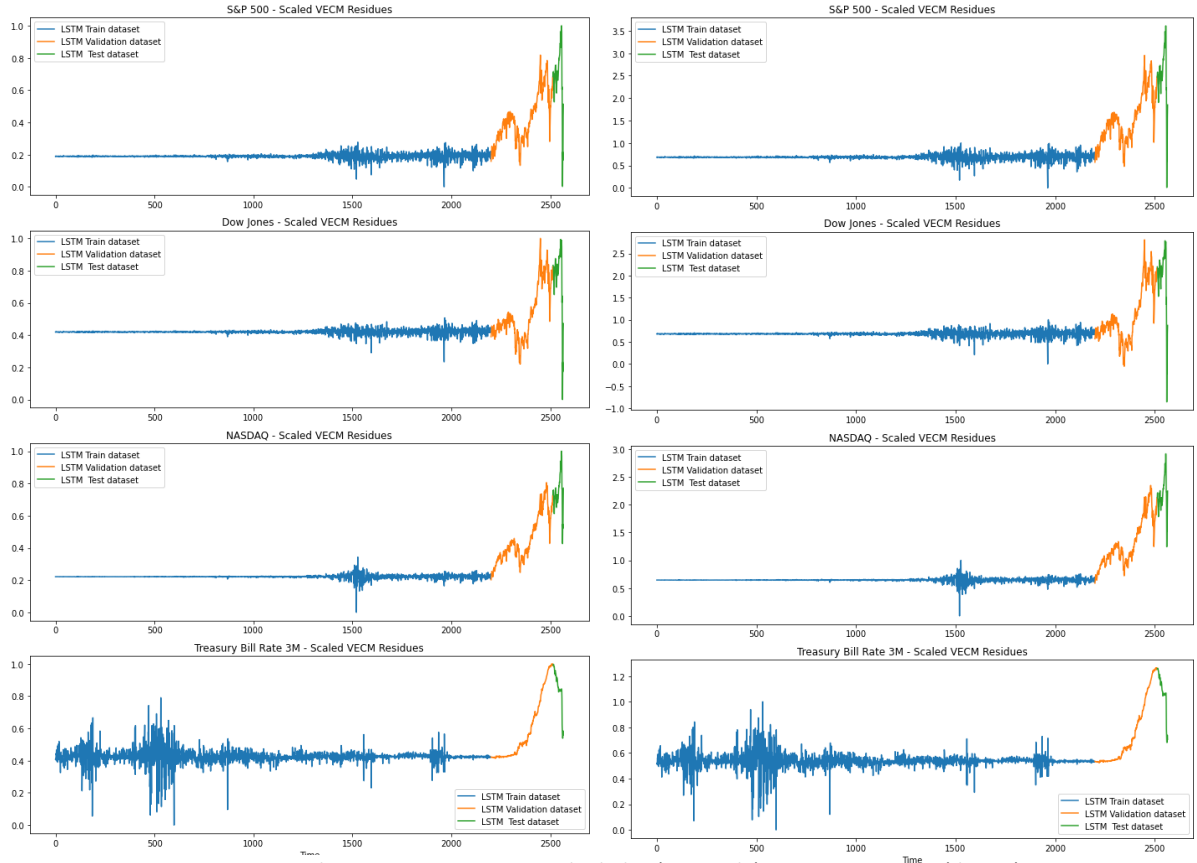

Figura 38. Inputs do LSTM com vazamento de dados (esquerda) e sem vazamento (direita)

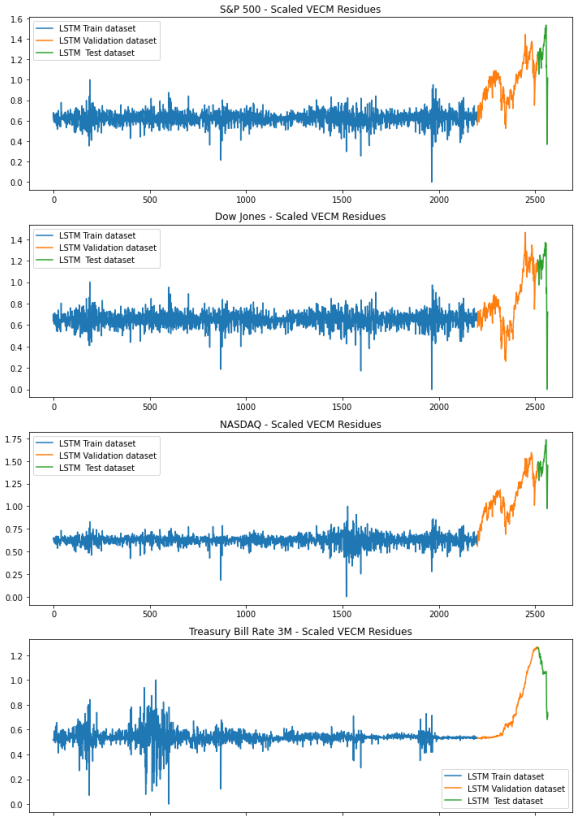

Figura 39. Inputs do LSTM sem vazamento de dados do logaritmo das séries (exceto TB3M)

Curvas de aprendizagem para os LSTM de referência:

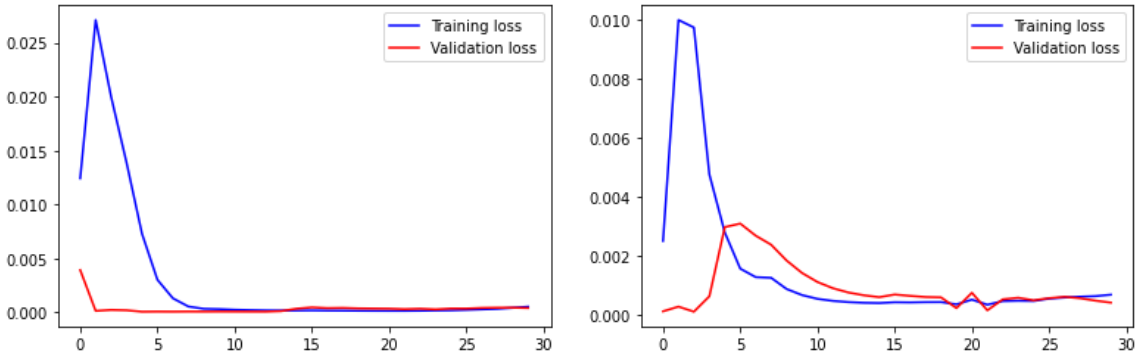

Figura 40. Curvas dos erros por época durante os períodos de treino e validação: LSTM univariado com 4 *lags* (esquerda) e 20 *lags* (direita)

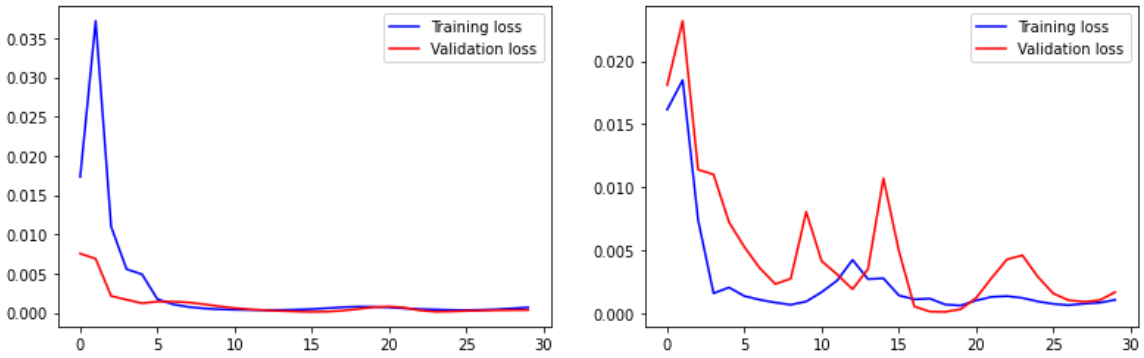

Figura 41. Curvas dos erros por época durante os períodos de treino e validação: LSTM multivariado com 4 *lags* (esquerda) e 20 *lags* (direita)

Curvas de aprendizagem para os LSTM dos modelos híbridos:

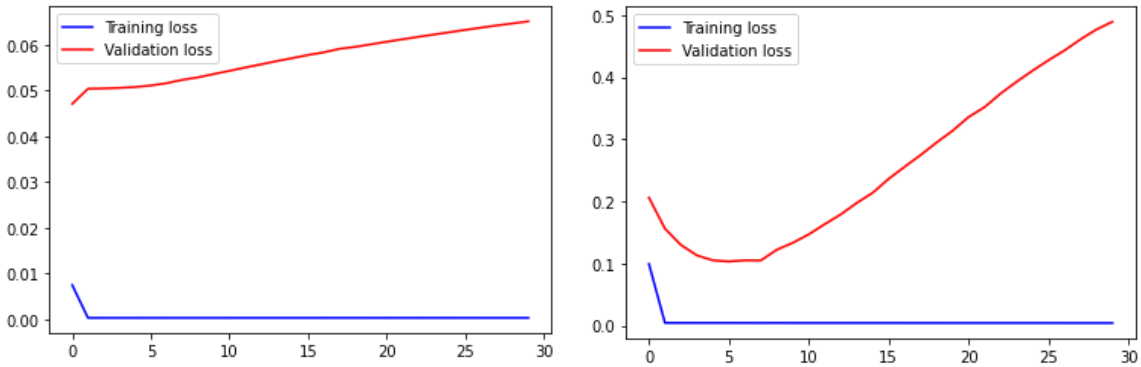

Figura 42. Curvas dos erros por época durante os períodos de treino e validação: modelo híbrido com 4 *lags* e vazamento de dados (esquerda) e sem vazamento (direita)

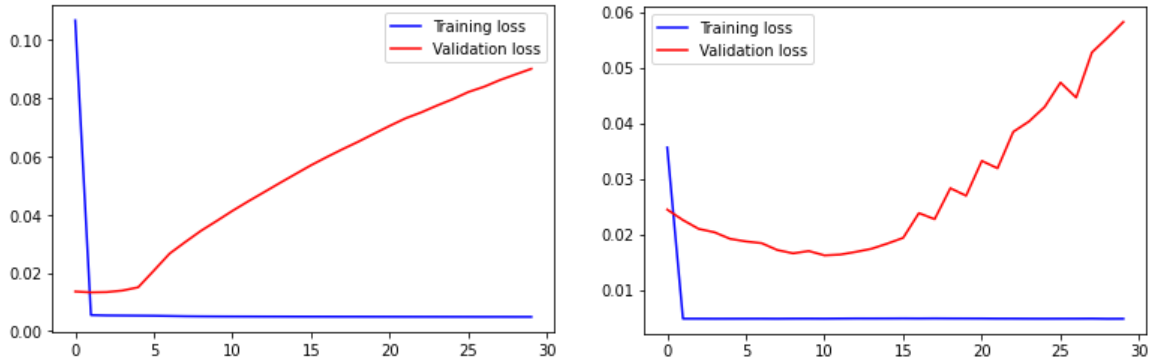

Figura 43. Curvas dos erros por época durante os períodos de treino e validação: modelo híbrido dos logaritmos com 4 *lags* (esquerda) e 20 *lags* (direita)

E. Montagem de modelos e Previsão final

Tabela 9. Métricas de avaliação das previsões dos modelos de referência, ordenados pelo mape%, período de teste com e sem COVID-19 (até 14/02/2020)

Período de Teste Modelos	mape%	mae	rmse	rmse%	acf1	corr
Média Móvel (4)	3.7429	108.6917	169.0513	6.2854	0.6479	0.6749
LSTM(4)_uni_época(13)	4.7041	135.3484	220.3388	8.1641	0.6860	0.5981
NAIVE	5.5138	170.5628	230.7686	7.6259	0.8568	-0.0000
LSTM(20)_uni	5.7017	165.7239	243.1290	9.2077	0.8253	0.3164
LSTM(20)_multi_época(19)	6.1425	169.7500	314.3144	12.1060	0.8679	0.0292
Média Móvel (20)	7.2745	213.7474	282.0899	10.5034	0.8681	-0.1287
LSTM(20)_uni_época(22)	8.2148	252.0052	279.8991	9.5690	0.8341	0.2559
LSTM(4)_multi_época(16)	8.3949	250.8054	333.5891	11.8739	0.7787	0.5607
LSTM(4)_uni	9.3852	280.2216	331.6933	11.7403	0.6845	0.6015
LSTM(20)_multi	13.8138	436.0077	459.6909	14.5171	0.8619	0.0759
LSTM(4)_multi	14.3620	435.6693	506.1210	17.3557	0.8122	0.5507
VECM(2)_1co	18.5415	596.3743	639.4035	19.6283	0.8573	-0.2795
Período de Teste não COVID Modelos	mape%	mae	rmse	rmse%	acf1	corr
LSTM(20)_multi_época(19)	1.2059	39.2908	44.7087	1.3622	0.5396	0.9589
Média Móvel (4)	1.9753	61.6957	68.1575	2.1984	0.5762	0.9211
LSTM(4)_uni_época(13)	2.3183	71.3554	93.5879	3.0960	0.6374	0.9015
LSTM(20)_uni	2.9606	97.9162	115.0750	3.4441	0.7363	0.9561
Média Móvel (20)	4.2159	138.8459	155.1770	4.6579	0.7873	0.9518
NAIVE	4.6728	152.0820	204.8337	6.1486	0.8869	-0.0000
LSTM(4)_multi_época(16)	5.7120	181.3727	205.3756	6.4194	0.8055	0.9213
LSTM(4)_uni	7.2779	227.2119	237.8866	7.6791	0.6403	0.9023
LSTM(20)_uni_época(22)	7.5061	244.9525	253.8517	7.7138	0.7583	0.9543
LSTM(4)_multi	11.5606	366.6429	390.1278	12.1875	0.8626	0.9261
LSTM(20)_multi	15.6862	507.3372	508.7715	15.7164	0.5050	0.9631
VECM(2)_1co	21.4531	697.3532	706.6242	21.6124	0.8412	0.9640

Tabela 10. Medidas de avaliação das previsões de todos os modelos no período de teste

Período de Teste Modelos	mape%	mae	rmse	rmse%	acf1	corr
NAIVE	5.5138	170.5628	230.7686	7.6259	0.8568	-0.0000
Média Móvel (4)	3.7429	108.6917	169.0513	6.2854	0.6479	0.6749
Média Móvel (20)	7.2745	213.7474	282.0899	10.5034	0.8681	-0.1287
VECM(2)_1co	18.5415	596.3743	639.4035	19.6283	0.8573	-0.2795
LSTM(4)_uni	9.3852	280.2216	331.6933	11.7403	0.6845	0.6015
LSTM(4)_uni_época(13)	4.7041	135.3484	220.3388	8.1641	0.6860	0.5981
LSTM(20)_uni	5.7017	165.7239	243.1290	9.2077	0.8253	0.3164
LSTM(20)_uni_época(22)	8.2148	252.0052	279.8991	9.5690	0.8341	0.2559
LSTM(4)_multi	14.3620	435.6693	506.1210	17.3557	0.8122	0.5507
LSTM(4)_multi_época(16)	8.3949	250.8054	333.5891	11.8739	0.7787	0.5607
LSTM(20)_multi	13.8138	436.0077	459.6909	14.5171	0.8619	0.0759
LSTM(20)_multi_época(19)	6.1425	169.7500	314.3144	12.1060	0.8679	0.0292
VECM(2)_1co + LSTM(4)_vaza/	16.5080	520.4303	551.3086	17.2555	0.8444	0.2681
VECM(2)_1co + LSTM(4)	12.4340	390.2723	414.9227	13.0522	0.8481	0.3224
VECM(2)_1co + LSTM(20)	11.9530	378.8985	406.1308	12.7087	0.8569	-0.0589
VECM(2)_1co + LSTM(4)_log	13.3841	421.0698	447.0110	14.0225	0.8247	0.4911
VECM(2)_1co + LSTM(20)_log	11.8437	375.5849	405.2324	12.6783	0.8602	-0.1661
VECM(2)_1co + LSTM(4)_época(4)	5.2094	152.5272	233.1271	8.4306	0.7465	0.6128
VECM(2)_1co + LSTM(4)_log_época(3)	3.9669	113.8389	187.1198	6.9366	0.6788	0.6360
VECM(2)_1co + LSTM(20)_época(8)	6.0494	169.9855	286.6732	11.0329	0.8554	0.0822
VECM(2)_1co + LSTM(20)_log_época(9)	6.1619	175.3228	273.4045	10.4867	0.8511	0.0895

Tabela 11. Medidas de avaliação das previsões de todos os modelos no período de teste, ordenadas pelo MAPE%

Período de Teste Modelos	mape%	mae	rmse	rmse%	acf1	corr
Média Móvel (4)	3.7429	108.6917	169.0513	6.2854	0.6479	0.6749
VECM(2)_1co + LSTM(4)_log_época(3)	3.9669	113.8389	187.1198	6.9366	0.6788	0.6360
LSTM(4)_uni_época(13)	4.7041	135.3484	220.3388	8.1641	0.6860	0.5981
VECM(2)_1co + LSTM(4)_época(4)	5.2094	152.5272	233.1271	8.4306	0.7465	0.6128
NAIVE	5.5138	170.5628	230.7686	7.6259	0.8568	-0.0000
LSTM(20)_uni	5.7017	165.7239	243.1292	9.2077	0.8253	0.3164
VECM(2)_1co + LSTM(20)_época(8)	6.0494	169.9855	286.6732	11.0329	0.8554	0.0822
LSTM(20)_multi_época(19)	6.1425	169.7501	314.3144	12.1060	0.8679	0.0292
VECM(2)_1co + LSTM(20)_log_época(9)	6.1619	175.3228	273.4045	10.4867	0.8511	0.0895
Média Móvel (20)	7.2745	213.7474	282.0899	10.5034	0.8681	-0.1287
LSTM(20)_uni_época(22)	8.2148	252.0052	279.8991	9.5690	0.8341	0.2559
LSTM(4)_multi_época(16)	8.3949	250.8054	333.5891	11.8739	0.7787	0.5607
LSTM(4)_uni	9.3852	280.2216	331.6933	11.7403	0.6845	0.6015
VECM(2)_1co + LSTM(20)_log	11.8437	375.5849	405.2324	12.6783	0.8602	-0.1661
VECM(2)_1co + LSTM(20)	11.9530	378.8985	406.1308	12.7087	0.8569	-0.0589
VECM(2)_1co + LSTM(4)	12.4340	390.2723	414.9227	13.0522	0.8481	0.3224
VECM(2)_1co + LSTM(4)_log	13.3841	421.0698	447.0110	14.0225	0.8247	0.4911
LSTM(20)_multi	13.8138	436.0076	459.6909	14.5171	0.8619	0.0759
LSTM(4)_multi	14.3620	435.6693	506.1209	17.3557	0.8122	0.5507
VECM(2)_1co + LSTM(4)_vaza/	16.5080	520.4303	551.3086	17.2555	0.8444	0.2681
VECM(2)_1co	18.5415	596.3743	639.4035	19.6283	0.8573	-0.2795

Tabela 12. Medidas de avaliação das previsões de todos os modelos no período de teste não COVID-19 (até 14/02/2020)

Período de Teste não COVID Modelos	mape%	mae	rmse	rmse%	acf1	corr
NAIVE	4.6728	152.0820	204.8337	6.1486	0.8869	-0.0000
Média Móvel (4)	1.9753	61.6957	68.1575	2.1984	0.5762	0.9211
Média Móvel (20)	4.2159	138.8459	155.1770	4.6579	0.7873	0.9518
VECM(2)_1co	21.4531	697.3532	706.6242	21.6124	0.8412	0.9640
LSTM(4)_uni	7.2779	227.2119	237.8866	7.6791	0.6403	0.9023
LSTM(4)_uni_época(13)	2.3183	71.3554	93.5879	3.0960	0.6374	0.9015
LSTM(20)_uni	2.9606	97.9162	115.0750	3.4441	0.7363	0.9561
LSTM(20)_uni_época(22)	7.5061	244.9525	253.8517	7.7138	0.7583	0.9543
LSTM(4)_multi	11.5606	366.6429	390.1278	12.1875	0.8626	0.9261
LSTM(4)_multi_época(16)	5.7120	181.3727	205.3756	6.4194	0.8055	0.9213
LSTM(20)_multi	15.6862	507.3372	508.7715	15.7164	0.5050	0.9631
LSTM(20)_multi_época(19)	1.2059	39.2908	44.7087	1.3622	0.5396	0.9589
VECM(2)_1co + LSTM(4)_vaza/	17.9843	569.9906	583.6914	18.2406	0.8517	0.9282
VECM(2)_1co + LSTM(4)	13.4497	426.7526	439.5117	13.7143	0.8197	0.9263
VECM(2)_1co + LSTM(20)	13.1883	429.7678	440.4484	13.4104	0.8249	0.9603
VECM(2)_1co + LSTM(4)_log	14.5120	460.3017	473.1341	14.7687	0.8266	0.9374
VECM(2)_1co + LSTM(20)_log	13.0946	426.9406	438.7410	13.3456	0.8312	0.9590
VECM(2)_1co + LSTM(4)_época(4)	2.9052	90.4782	115.5134	3.7484	0.7248	0.8637
VECM(2)_1co + LSTM(4)_log_época(3)	1.9844	60.7305	79.2743	2.6444	0.6780	0.8743
VECM(2)_1co + LSTM(20)_época(8)	1.9253	62.9650	70.1536	2.1247	0.7621	0.9314
VECM(2)_1co + LSTM(20)_log_época(9)	2.4191	79.1787	88.9104	2.6829	0.8103	0.9261

Tabela 13. Medidas de avaliação das previsões de todos os modelos no período de teste não COVID-19 (até 14/02/2020), ordenadas pelo MAPE%

Período de Teste Modelos	mape%	mae	rmse	rmse%	acf1	corr
LSTM(20)_multi_época(19)	1.2059	39.2908	44.7087	1.3622	0.5396	0.9589
VECM(2)_1co + LSTM(20)_época(8)	1.9253	62.9650	70.1536	2.1247	0.7621	0.9314
Média Móvel (4)	1.9753	61.6957	68.1575	2.1984	0.5762	0.9211
VECM(2)_1co + LSTM(4)_log_época(3)	1.9844	60.7305	79.2743	2.6444	0.6780	0.8743
LSTM(4)_uni_época(13)	2.3183	71.3554	93.5879	3.0960	0.6374	0.9015
VECM(2)_1co + LSTM(20)_log_época(9)	2.4191	79.1787	88.9104	2.6829	0.8103	0.9261
VECM(2)_1co + LSTM(4)_época(4)	2.9052	90.4782	115.5134	3.7484	0.7248	0.8637
LSTM(20)_uni	2.9606	97.9162	115.0750	3.4441	0.7363	0.9561
Média Móvel (20)	4.2159	138.8459	155.1770	4.6579	0.7873	0.9518
NAIVE	4.6728	152.0820	204.8337	6.1486	0.8869	-0.0000
LSTM(4)_multi_época(16)	5.7120	181.3727	205.3756	6.4194	0.8055	0.9213
LSTM(4)_uni	7.2779	227.2119	237.8866	7.6791	0.6403	0.9023
LSTM(20)_uni_época(22)	7.5061	244.9525	253.8517	7.7138	0.7583	0.9543
LSTM(4)_multi	11.5606	366.6429	390.1278	12.1875	0.8626	0.9261
VECM(2)_1co + LSTM(20)_log	13.0946	426.9406	438.7410	13.3456	0.8312	0.9590
VECM(2)_1co + LSTM(20)	13.1883	429.7678	440.4484	13.4104	0.8249	0.9603
VECM(2)_1co + LSTM(4)	13.4497	426.7526	439.5117	13.7143	0.8197	0.9263
VECM(2)_1co + LSTM(4)_log	14.5120	460.3017	473.1341	14.7687	0.8266	0.9374
LSTM(20)_multi	15.6862	507.3372	508.7715	15.7164	0.5050	0.9631
VECM(2)_1co + LSTM(4)_vaza/	17.9843	569.9906	583.6914	18.2406	0.8517	0.9282
VECM(2)_1co	21.4531	697.3532	706.6242	21.6124	0.8412	0.9640