Previsão do preço de um produto

Aplicações de Inteligência Artificial

–

Universidade de Aveiro

2022/2023

João Martins (NMec 93304)
Mestrado Integrado em Engenharia Computacional
Departamento de Física
Aveiro, Portugal
joao.paul@ua.pt

Miguel Marques (NMec 98532)
Licenciatura em Engenharia Computacional
Departamento de Física
Aveiro, Portugal
miguel.rosas@ua.pt

***Resumo*—A previsão de preço ajuda empresas quando elas têm a necessidade de aumentar ou descer os preços para corresponder à demanda dos comsumidores.**
No nosso trabalho iremos prever o preço de um produto de uma empresa de telecomunicações, mais concretamente o Huawei Y9S 128GB desbloqueado. O modelo que teve melhor desempenho foi o ARIMA com um *RMSE* de 0.286.

I. Introdução

O desenvolvimento de novas tecnologias assim como novas descobertas na área de Inteligência Artificial permitiram o surgimento de novos métodos para a previsão. Tendo em conta o nosso objetivo de prever o preço de um produto, nós iremos utilizar modelos de Inteligência Artificial tais como: LSTM univariado e multivariado, para além destes iremos também utilizar o modelo tradicional ARIMA.

II. Análise do conjunto de dados

O nosso dataset tem 11 séries temporais para o preço do Huawei Y9S 128GB ao longo de um período de tempo em diferentes empresas.

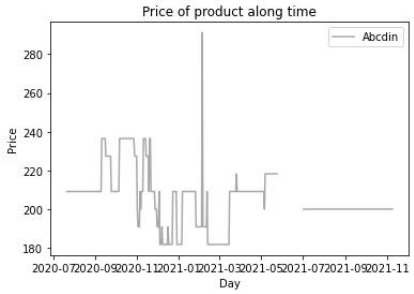

Figura 1. Valores do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Abcdind

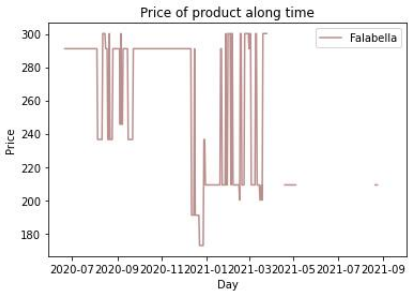

Figura 2. Valores do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Falabella

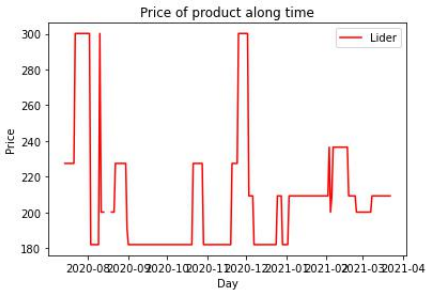

Figura 3. Valores do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Lider

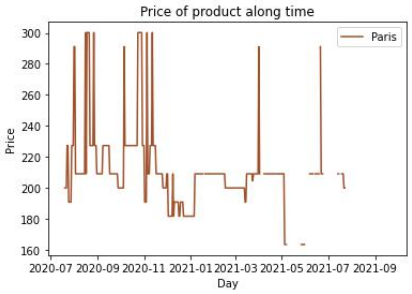

Figura 4. Valores do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Paris

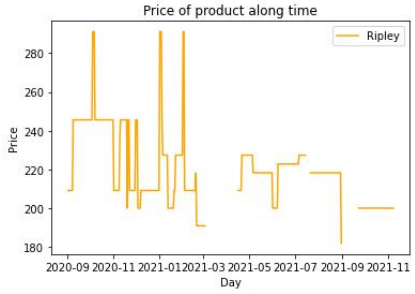

Figura 5. Valores do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Ripley

Figura 6. Valores do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Claro

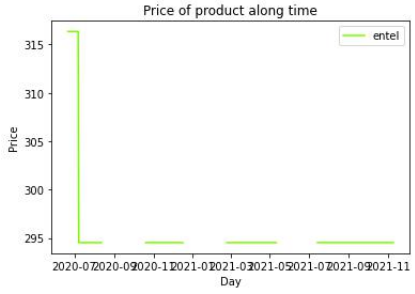

Figura 7. Valores do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Entel

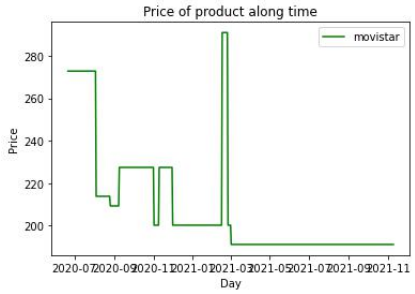

Figura 8. Valores do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Movistar

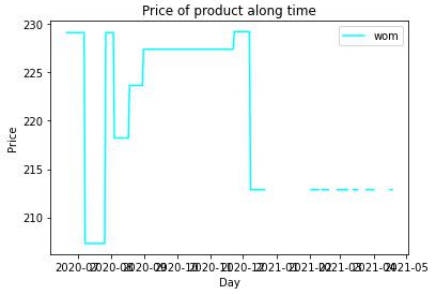

Figura 9. Valores do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Wom

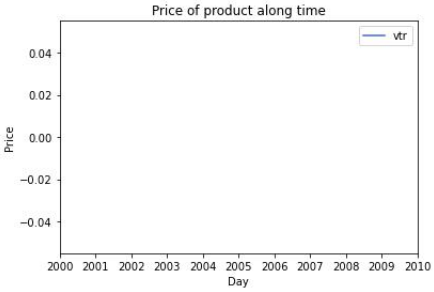

Figura 10. Valores do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Vtr

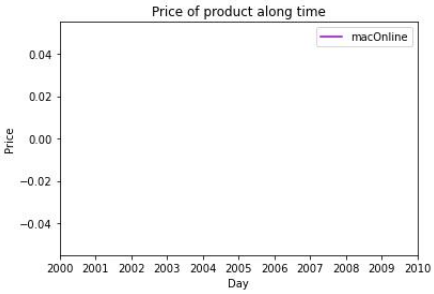

Figura 11. Valores do preço do Huawei Y9S 128GB ao longo do tempo para a empresa macOnline

Como podemos ver através das figuras 10 e 11, as empresas Vtr e macOnline não tem preço para o Huawei Y9S 128GB ao longo do tempo, relativamente às restantes empresas, podemos verificar que há espaços na série temporal onde têm valores em falta.

III. Pré-processamento de dados

A. *Univariada*

Neste caso iremos utilizar apenas uma série temporal, mais concretamente a série temporal da empresa Abcdin para prever o preço do produto ao longo do tempo. Como foi mencionado na II, as nossas séries temporais têm valores em falta, neste caso como estamos a tratar de modelos univariados.

O tratamento de dados pode ser dividido em 3 casos:

- Valores em falta no início da série temporal: Removemos os valores da série.

- Valores em falta no meio da série temporal: Fazemos a interpolação dos valores.
- Valores em falta no final da série temporal: Fazemos a extrapolação dos valores.

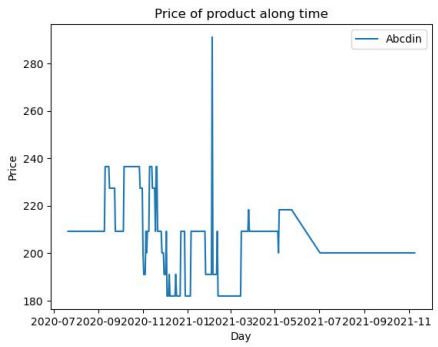

Figura 12. Gráfico do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Abcdin após interpolação

Se compararmos a figura 12 com a figura 1 verificamos que já não temos valores em falta ao longo da série.

B. Multivariada

Neste caso iremos utilizar valores do preço do Huawei Y9S 128GB de várias empresas para prever o preço deste produto para a empresa Abdcdin.

Como previamente, as séries temporais têm valores em falta, para além disto duas das empresas, Vtr e macOnline respetivamente não têm valores para o preço do produto, pelo que não iremos considerar estas duas empresas. O processamento de dados foi semelhante ao da *Univariada*, exceto o tratamento de dados dos valores em falta no ínicio da série temporal. Neste caso nós analisamos qual das empresas tinha o maior número de valores em falta, para que todas as séries temporais começassem ao mesmo tempo desta, removemos os valores de todas as empresas até aquele dia.

IV. Descrição de modelos utilizados

A. LSTM (Long-short-term-memory)

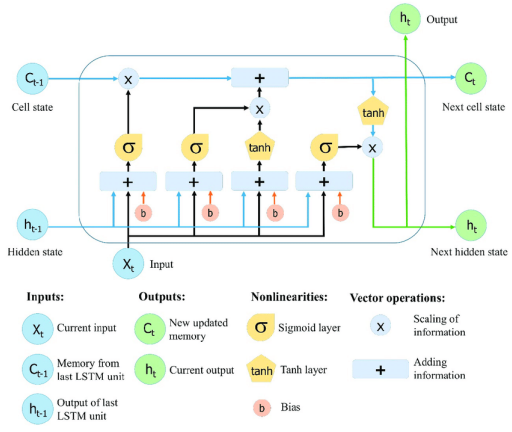

Figura 13. Estrutura do modelo LSTM

Um modelo de LSTM é composto por um *cell state*, um *hidden state*, um *input gate*, um *output gate* e um *forget gate*.

A *cell state* é responsável pela memória a longo prazo, isto é feito através de um vetor que guarda a sequência de inputs até ao momento, a *cell state* é atualizada através do *forget gate* e *input gate*.

O *hidden state* é responsável pela memória a curto prazo, isto também é feito através de um vetor.

O *forget gate* irá determinar a informação que será descartada, isto é feito através do input atual e do *hidden state* anterior. Com estes parâmetros de entrada no *gate* e uma função *sigmoid* que irá calcular um valor entre 0 e 1 para cada elemento, os valores que estão próximos de 0 serão descartados da *cell state* enquanto que os valores próximos de 1 serão mantidos.

O *input gate* é utilizado para controlar o fluxo de informação para a *cell state*. Este *gate* utiliza os valores do input atual e o *hidden state* anterior para calcular o peso para a nova informação que poderá ser adicionada à *cell state*, caso a função *sigmoid* ou a função *tanh* calcula um valor igual a 0, então esta informação terá um peso de igual a 0. Isto significa que a informação não é relevante e não será adicionada à *cell state*.

O *output gate* atualiza o *hidden state* e com base neste *hidden state* e usando uma função por exemplo *softmax* podemos calcular o valor para a observação atual. Este gate tem como parâmetros o input atual e o *hidden state* anterior que terão um peso calculado pela função *sigmoid* e tem também a *cell state* que terá um um peso calculado pela função *tanh*, através da multiplicação dos pesos calculados através destas duas funções selecionamos a informação relevante para o *hidden state*

B. ConvLSTM

Este modelo combina dois tipos de neural networks usando CNN para extrair *features* espaciais e LSTM para capturar dependências temporais. A arquitetura deste modelo em questão também consegue manter e atualizar o estado interno semelhante às LSTM. No nosso modelo utilizamos a dimensão do *kernel* será igual a dimensão das *subsequences*, ou seja a dimensão das partições que fizemos à nossa sequência original, nós iremos considerar duas subsequências.

C. ARIMA (AutoRegressive Integrated Moving Average)

Este é um modelo estatístico o qual combina três componentes, *autoregression*, *integration* e *moving average*

A componente *AR (autoregression)* modela a dependência das observações ao longo do tempo, esta é modelada como uma combinação linear dos *"p"lagged* valores anteriores

A componente *I (integration)* é utilizada para colocar a série temporal estacionária, isto significa que as propriedades da série são constantes ao longo do tempo. A ordem de integração *"d"* representa a quantidade de vezes que os dados têm que ser derivados para colocar a série estacionária.

A componente *MA (moving average)* modela a dependência entre a observação atual e o termo de erro aleatório. Esta componente modela o erro assumindo que é uma combinação dos *"q"lagged* termos de erro.

V. Modelos Utilizados

Na nossa construção dos modelos variamos os hiperparâmetros para calcular quais dentro do conjunto que definimos otimizam o modelo de modo a melhorar a sua performance.

Nós separamos o nosso conjunto de dados originais em múltiplas sequências de dimensão *n*, onde n representa o número de observações anteriores a serem consideradas para a previsão de um valor, a cada sequência está associada um valor *y* que corresponde ao valor real seguinte da sequência. Separamos o conjunto de sequências de dados em um conjunto de treino e teste, o conjunto de dados de treino ficou com os primeiros 70% dos dados enquanto os de teste os últimos 30%.

A. Vanilla LSTM

Este é modelo *LSTM* que contêm apenas uma *layer*, este tipo de modelo *LSTM* têm apenas um hyperparâmetro que é o número de observações anteriores a ter em questão para a previsão de um valor. O número de observações a considerar irá variar desde 3 até a 19 com um passo de 4.

B. Stacked LSTM

Nesta variação do modelo *LSTM* temos como hyperparâmetros o número de steps, este irá variar entre 3 e 11 com um passo de 4, para além disto, temos quantidade de hidden layers, a quantidade de *hidden layers* irá variar de 2 até 5.

C. ConvLSTM

Neste variante do modelo *LSTM*, que têm como hyperparâmetros o número de layers que será 4, o tamanho de filtros que será dimensão [1,2], a quantidade de observações que irá variar no intervalo [6,14] com um passo de 4 por fim temos a quantidade de filtros que irá variar desde 16 até 48 com um passo de 16.

D. Multiple Input Series LSTM

Neste modelo de LSTM iremos variar o número de observações desde 3 até 19 com um passo de 4, iremos também variar o número de *hidden layers* de 2 até 5. Este é um modelo do tipo multivariado em que iremos considerar os dados de múltiplas empresas.

E. AutoRegressive Integrated Moving Average (ARIMA)

Neste modelo consideramos duas abordagens, um com vista na análise e processamento do dados e outra de grid search. Na primeira determinamos o número de diferenciações baseado o estudo das estacionariedade das mesmas tendo em conta o teste ***Augmented Dickey–Fuller test (ADF)*** e o ***Kwiatkowski-Phillips-Schmidt-Shin (KPSS)***.

P-values	ADF	KPSS	Stationarity
Original	0.0110	0.0403	YES
First Diff	2.2731e-11	0.1	NO
Second Diff	7.5831e-14	0.0417	YES

E com o auxílio dos gráficos da ***Autocorrelation Function (ACF)*** ee da ***Partial Autocorrelation Function (PACF)***

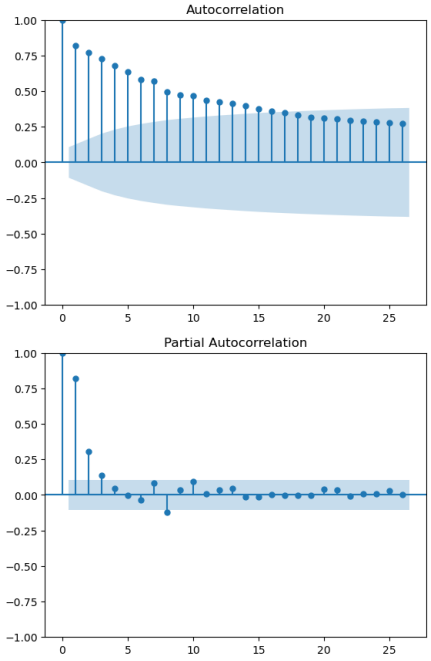

Figura 14. Gráficos para os dados originais da ACF e PCF ,respetivamente

Deste modo, para o nosso modelo manual, determinamos que os parâmetros da nossa (p,d,q) seriam (6,0,2) que desempenha a mesma função que um modelo ARMA(6,2).

SARIMAX Results

Dep. Variable:	data	No. Observations:	333
Model:	SARIMAX(6, 0, 2)	Log Likelihood	-1205.445
Date:	Tue, 24 Jan 2023	AIC	2428.891
Time:	22:30:40	BIC	2463.164
Sample:	07-21-2020 - 06-18-2021	HQIC	2442.558
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.5226	0.039	13.442	0.000	0.446	0.599
ar.L2	1.2140	0.061	19.887	0.000	1.094	1.334
ar.L3	-0.4160	0.057	-7.242	0.000	-0.529	-0.303
ar.L4	-0.1720	0.083	-2.075	0.038	-0.334	-0.010
ar.L5	-0.1063	0.058	-1.834	0.067	-0.220	0.007
ar.L6	-0.0422	0.071	-0.592	0.554	-0.182	0.098
ma.L1	0.0140	0.080	0.175	0.861	-0.143	0.171
ma.L2	-0.9860	0.052	-18.845	0.000	-1.088	-0.883
sigma2	80.2816	0.002	3.8e+04	0.000	80.277	80.286

Figura 15. Sumário do modelo utilizado

Podemos verificar que a maior parte dos coeficientes são significantes.

Para o modelo optimizado foram utilizados valores de p e q de 0 a 8 e valores de d de 0 a 2. Foram testadas todas as possíblidades e o melhor modelo obtido foi o de valores de (p,d,q) de (3,1,1).

Ambos modelos apresentam resíduos com distribuíção normal e aparência random podendo-se concluir que se trata de white noise.

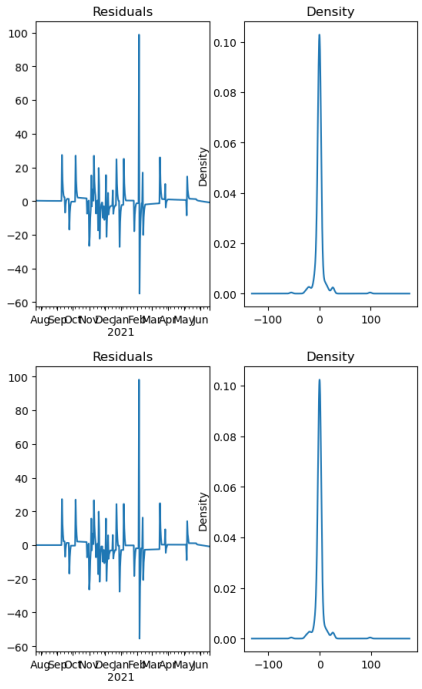

Figura 16. Gráficos dos resíduos para o modelo manual e para o modelo optimizado, respetivamente

VI. Resultados Obtidos

Para avaliar a performance do nossos modelos utilizamos a métrica *RMSE (Root Mean Square Error)* entre o valores reais e os valores previstos.

A. Vanilla LSTM

Na tabela I do apêndice, podemos verificar que o melhor hiperparâmetro, é o número de observações igual a 3 com um (RMSE) de 0.4057078 para os dados de teste, para os dados de treino obtivemos um RMSE de 8.7531412. Neste modelo usamos então 3 observações anteriores para prever o preço numa instância de tempo.

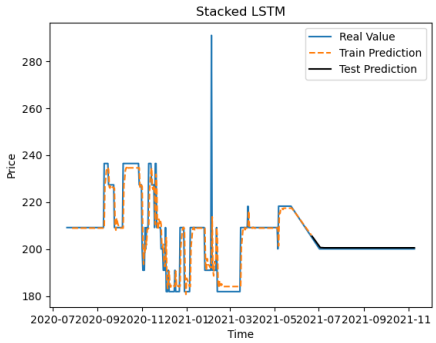

Figura 17. Gráfico do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Abcdin para os valores reais e os previstos para os dados de treino e de teste

B. Stacked LSTM

Na tabela A do apêndice, podemos concluir que o modelo que obteve a melhor performance foi aquele que teve o número de observações igual a 7 e o número de *layers* igual a 2, para os dados de teste obtivemos um *RMSE* de 0.5106161 e para os dados de treino um valor de 8.1799280.

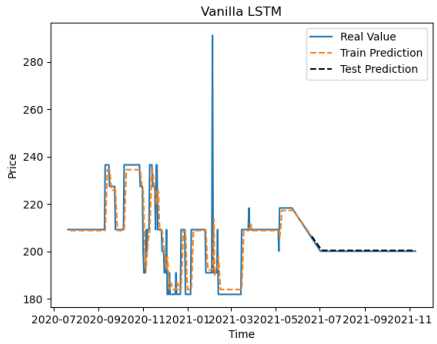

Figura 18. Gráfico do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Abcdin para os valores reais e os previstos para os dados de treino e de teste

Analisando os resultados obtidos na tabela A do apêndice, podemos verificar que aumentar a complexidade da rede neuronal nem sempre resulta em melhores resultados, isto é evidenciado na maneira em que conforme aumentamos o número de *layers* a performance do modelo nem sempre é melhor.

C. ConvLSTM

Analisando os resultados da tabela A do apêndice, verificamos que os melhores hiperparâmetros são o número de observações igual a 6 e o número de filtros igual a 48. O valor obtido para *RMSE* no conjunto de dados de teste foi de 0.4352977 enquanto que no conjunto de dados de treino foi de 8.7584340.

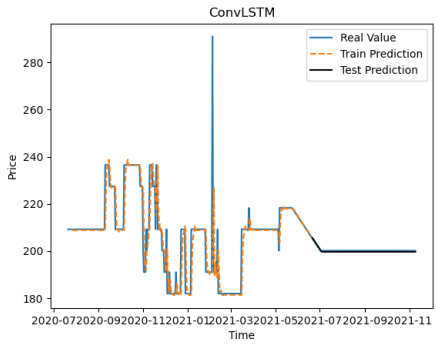

Figura 19. Gráfico do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Abcdin para os valores reais e os previstos para os dados de treino e de teste

D. Multiple Input Series LSTM

Analisando os resultados da tabela A do apêndice, concluímos que os melhores hiperparâmetros são 3 para o número

de observações e 3 *layers*. Para o conjunto de dados de teste obtivemos o RMSE igual a 14.8781837 e para os dados de treino obtivemos 22.7813317.

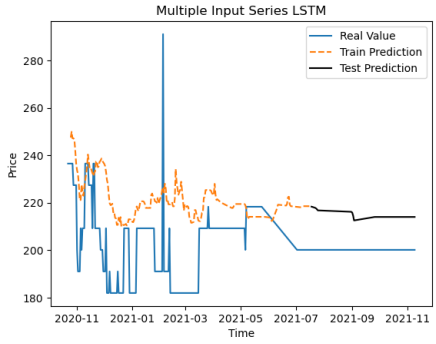

Figura 20. Gráfico do preço do Huawei Y9S 128GB ao longo do tempo para a empresa Abcdin para os valores reais e os previstos para os dados de treino e de teste

Podemos verificar que o valores preditos estão bastante distanciados dos valores reais, isto é esperado uma vez que as previsões deste modelo foram influenciadas por os preços deste produto em outras empresas.

E. AutoRegressive Integrated Moving Average (ARIMA)

Para os dois modelos propostos foram obtidos valores de RMSE de 0.286 para o modelo manual e de 0.648 para o modelo optimizado para os dados de teste.Os parâmetros optimizados obtiveram pior preformance devido à escolha da série temporal não estacionária possibilitada pela gama dos valores de d .

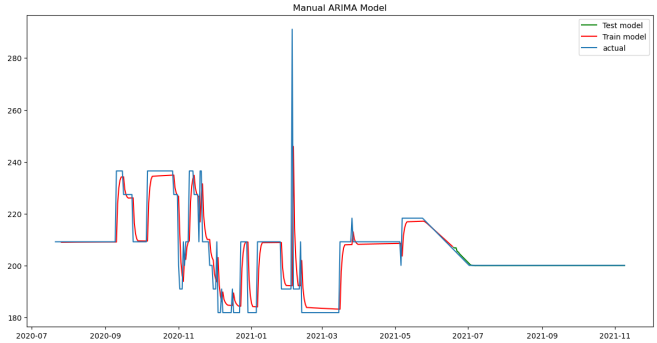

Figura 21. Gráfico dos valores previstos para treino e teste do modelo manual

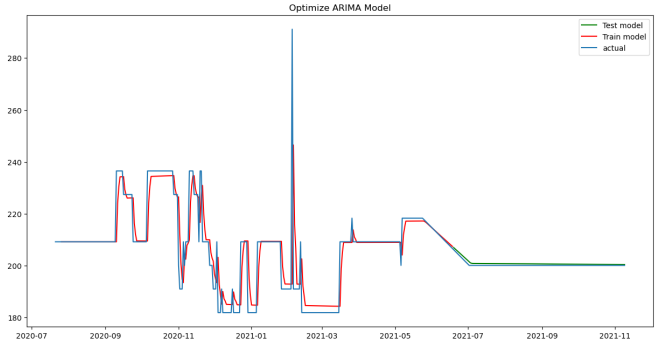

Figura 22. Gráfico dos valores previstos para treino e teste do modelo optimizado

Para o caso dos modelos univariados podemos concluir que a previsão feita por estes foi satisfatória, verificamos que nestes modelos houve uma enorme descrepância entre o erro de treino e de teste, isto é devido à volatilidade dos dados de treino que é bastante superior à volatilidade no conjunto de dados de teste. O modelo que obteve a melhor performance foi o modelo ARIMA calculado manualmente, com um *RMSE* de 0.286.

VII. Conclusão

Tendo em conta o objetivo deste trabalho em prever o preço do produto Huawei Y9S 128GB, podemos dizer que o realizamos com sucesso, implementamos vários modelos de redes neuronais em que variámos os hiperpârametros determinar uma melhor performance nos mesmos. Também implementamos um modelo clássico ARIMA que obteve bons resultados.

Apesar de termos obtido um *RMSE* menor nos métodos univariados comparado aos métodos multivariados, o método multivariado é mais adequado, no contexto em que nos permite estar a frente dos nossos competidores, neste caso a alterações que as outras empresas possam fazer ao preço Huawei Y9S 128GB.

No trabalho [4], é feita a previsão do preço de *stocks*, os autores obtiveram valores de (RMSE) menores, analisando a Tabela 1 deste mesmo trabalho verificamos que o *RMSE* para o modelo *LSTM* este num intervalo de [0.0004928, 0,0031980] e o modelo *BI-LSTM* teve um erro no intervalo de [0.0003568, 0.0007167] conforme o número de epochs. Este resultados são melhores que os obtidos nos nossos modelos, o que nos mostra que podemos fazer um *tuning* dos hiperparâmetros para melhorar a performance dos nossos modelos.

No âmbito deste trabalho para futuras melhorias temos um *tuning* dos hiperparâmetros para melhorar a performance dos modelos já implementados, podemos também implementar o método de *ensemble* para combinar múltiplos métodos de previsão de valores o que pode resultar numa melhor performance e por fim aplicar os modelos às restantes séries temporais para obter estatísticas gerais a nível do desempenho dos diferentes modelos.

REFERÊNCIAS

[1] https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

[2] https://www.machinelearningplus.com/time-series/vector-autoregression-examples-python/

[3] https://www.researchgate.net/figure/The-structure-of-the-Long-Short-Term-Memory-LSTM-neural-network-Reproduced-from-Yan_fig8_334268507

[4] M. A. Istiake Sunny, M. M. S. Maswood and A. G. Alharbi, "Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model,"2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 2020, pp. 87-92, doi: 10.1109/NILES50944.2020.9257950. https://ieeexplore.ieee.org/abstract/document/9257950

APÊNDICE

Vanilla LSTM

Tabela de resultados obtidos para o modelo Vanilla LSTM.

Nº de Observações	Erro de Teste
3	0.4057078
7	5.7989957
11	4.0195729
15	0.4945385
19	0.7524928

Tabela I

TABELA DOS RESULTADOS OBTIDOS PARA DIFERENTES CONJUNTOS DE HIPERPARÂMETROS DO MODELO VANILLA LSTM

Stacked LSTM

Tabela de resultados obtidos para o modelo Stacked LSTM.

Nº de Observações	Nº de Layers	Erro de Teste
3	2	0.8924409
3	3	1.8829684
3	4	2.2032056
3	5	4.3266740
7	2	0.5106161
7	3	1.6125049
7	4	10.730208
7	5	4.0044854
11	2	10.3015627
11	3	3.38171713
11	4	3.69523715
11	5	8.82932722

Tabela II

TABELA DOS RESULTADOS OBTIDOS PARA DIFERENTES CONJUNTOS DE HIPERPARÂMETROS DO MODELO STACKED LSTM

ConvLSTM

Tabela de resultados obtidos para o modelo ConvLSTM.

Nº de Observações	Nº de Filtros	Erro de Teste
6	16	0.5014070
6	32	0.61048852
6	48	0.43529770
10	16	0.54374184
10	32	0.78692476
10	48	3.51020589
14	16	1.07709612
14	32	0.79989192
14	48	0.91520242

Tabela III

TABELA DOS RESULTADOS OBTIDOS PARA DIFERENTES CONJUNTOS DE HIPERPARÂMETROS DO MODELO CONVLSTM

Multiple Input Series LSTM

Tabela de resultados obtidos para o modelo Multiple Input Series LSTM.

Nº de Observações	Nº de Layers	Erro de Teste
3	2	18.4018476
3	3	14.8781837
3	4	19.3864420
3	5	15.442695
7	2	17.716297
7	3	18.886586
7	4	15.978045
7	5	16.446645
11	2	22.104686
11	3	15.978473
11	4	15.751593
11	5	24.919147