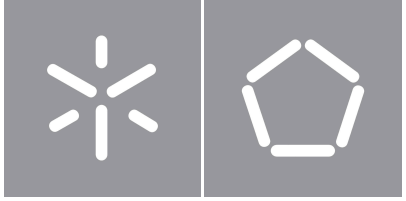


Universidade do Minho
Escola de Engenharia

Francisco Correia Franco

IA explicável para problemas de séries temporais

janeiro 2024



Universidade do Minho
Escola de Engenharia

Francisco Correia Franco

IA explicável para problemas de séries temporais

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho efetuado sob a orientação de
Paulo Jorge Freitas de Oliveira Novais

janeiro 2024

Direitos de Autor e Condições de Utilização do Trabalho por Terceiros

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho:



CC BY-NC-SA

https://creativecommons.org/licenses/by-nc-sa/4.0/

Agradecimentos

Em primeiro lugar, expresso a minha profunda gratidão ao meu orientador, Paulo Novais, pela orientação dedicada, pelo incentivo constante e pela sua notável sabedoria, que foram fundamentais para o desenvolvimento desta dissertação. Sem a sua orientação experiente, este projeto não teria alcançado o mesmo nível de excelência.

Gostaria também de agradecer ao Pedro Oliveira, investigador do projeto AIM4WATER, cuja ajuda e disponibilidade foram inestimáveis ao longo deste percurso. A sua orientação e partilha de conhecimento foram cruciais para a concretização deste projeto, e sou grato pela generosidade demonstrada em partilhar a sua expertise.

À minha querida mãe e à minha família, dedico um agradecimento especial. O vosso amor incondicional e o vosso apoio constante foram a força propulsora que me impulsionou nos momentos desafiantes. Agradeço por serem a minha base sólida e por tornarem esta jornada possível.

À Tuna Universitária do Minho, que me acolheu e se tornou uma segunda casa e refúgio durante estes anos, expresso a minha gratidão. A camaradagem, a música e o espírito de união foram fontes de inspiração que enriqueceram a minha experiência académica.

Não poderia deixar de agradecer ao Ivan e à Carol, por serem exemplos inspiradores em diferentes áreas da vida. As vossas trajetórias e valores foram referências importantes, motivando-me a perseguir a excelência em várias vertentes da minha vida.

E por fim, um agradecimento muito especial aos meus amigos de disquete, companheiros que me acompanharam ao longo destes 5 anos. A partilha de risos, desafios e conquistas foi crucial para manter a motivação e a alegria durante esta jornada.

Este projeto foi parcialmente apoiado por Fundos Nacionais através da agência de financiamento portuguesa, FCT - Fundação para a Ciência e a Tecnologia, dentro do projeto AIM4WATER, com a referência 2022.06822.PTDC (https://doi.org/10.54499/2022.06822.PTDC).

Declaração de Integridade

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Universidade do Minho, Braga, janeiro 2024

Assinado por: **Francisco Correia Franco**
Num. de Identificação: 14817391
Data: 2024.01.29 15:03:16+00'00'

Francisco Correia Franco

Resumo

Ao longo dos últimos anos, a aplicação de modelos de **Machine Learning (ML)** tem desempenhado um papel fundamental na nossa sociedade, ajudando a obter conhecimentos sobre grandes quantidades de dados. No entanto, existe uma preocupação quanto à clareza da forma como estes modelos chegam às suas conclusões. Por outras palavras, existe uma "*blackbox*"sobre a forma como estes modelos chegam às suas conclusões, quer sejam problemas de classificação ou de regressão. Atualmente, a explicabilidade tornou-se uma peça fundamental no desenvolvimento e aplicação de modelos de previsão para séries temporais. Em contextos onde a tomada de decisões é sensível, requerem fundamentação sólida e os dados são maioritariamente séries temporais, modelos com explicabilidade oferecem vantagens significativas. A transparência nas previsões não só ajuda a identificar possíveis falhas, mas também facilita a adoção desses modelos em ambientes críticos, promovendo uma integração mais eficaz da **Inteligência Artificial (IA)** no dia a dia.

Deste modo, esta dissertação visa a implementação de técnicas de explicabilidade num modelo **Long-Short Term Memory (LSTM)**, com o objetivo de aumentar a interpretabilidade deste modelo de *black-box*. Os dados utilizados para treinar o modelo, e efetuar o estudo da explicabilidade, são relativos a emissões de vários poluentes na cidade do Porto. Desta forma, cada *feature* é uma série temporal que contribui para prever as próximas 24 horas do **Dióxido de Nitrogénio (NO_2)**.

O modelo foi treinado inicialmente sozinho e depois foi inserida uma camada de *attention* no início. Desta forma, foi possível extrair explicabilidade *ante-hoc* da camada de *attention* e ainda aplicar estratégias *post-hoc*, como o **Shapley Additive Explanations (SHAP)** e *feature permutation*, ao modelo com e sem *attention*. Através dos resultados obtidos, é possível perceber que o modelo foi capaz de identificar dependências temporais e relações não lineares nos dados que foram cruciais para o "raciocínio"e tomada de decisão destes. Estas relações apenas foram possíveis de observar graças aos métodos de explicabilidade implementados, que, pela sua diversidade e convergência nos resultados, transmitem segurança e confiança na sua veracidade.

Palavras-chave Deep Learning, Explicabilidade, Machine Learning, Transparência, Séries Temporais.

Abstract

Over the last few years, the application of **ML** models has played a fundamental role in our society, helping to gain insight into large amounts of data. However, there is concern about the clarity of how these models reach their conclusions. In other words, there is a "*blackbox*"about how these models reach their conclusions, whether they are classification or regression problems. Nowadays, explainability has become a key element in the development and application of forecasting models for time series. In contexts where decision-making is sensitive, requires solid reasoning and the data is mostly time series, models with explainability offer significant advantages. Transparency in predictions not only helps to identify possible biases or failures, but also facilitates the adoption of these models in critical environments, promoting a more effective integration of artificial intelligence into everyday life.

Therefore, this dissertation aims to implement explainability techniques in a **LSTM** model, in order to increase the interpretability of this *blackbox* model. The data used to train the model and carry out the explainability study refers to emissions of various polutants in the city of Porto. Thus, each *feature* is a time series of one of the gases measured and is used to predict the the next 24 hours of NO_2 concentrations.

The model was initially trained alone and then a *attention* layer was inserted at the beginning. In this way, it was possible to extract *ante-hoc* explainability from the *attention* layer and also apply *post-hoc* strategies, such as **SHAP** and *feature permutation*, to the models with and without *attention*. From the results obtained, it can be seen that the model was able to identify temporal dependencies and non-linear relationships in the data that were crucial for their "reasoning"and decision-making. These relationships were only possible to observe thanks to the explainability methods implemented, which, due to their diversity and convergence in the results, convey security and confidence in their veracity.

Keywords Deep Learning, Explainability, Machine Learning, Time Series, Transparency.

Conteúdo

Lista de Abreviações e Acrónimos x

1 Introdução 1

1.1 Enquadramento e Motivação . . . 1
1.2 Objetivos . . . 5
1.3 Estrutura do Documento . . . 6

2 Estado da Arte 7

2.1 Explicabilidade: Conceitos e Técnicas . . . 7
2.1.1 Explicabilidade Ante-hoc . . . 8
2.1.2 Explicabilidade Post-hoc . . . 10
2.2 Explicabilidade para séries temporais . . . 16
2.2.1 Princípios para previsão de séries temporais . . . 16
2.2.2 ARIMA . . . 20
2.2.3 Exponencial smoothing . . . 21
2.2.4 Permutation Importance . . . 24
2.2.5 Mecanismos de Atenção . . . 25
2.2.6 LIME . . . 26
2.2.7 SHAP . . . 27
2.3 Deep Learning para séries temporais . . . 29
2.3.1 RNN . . . 29
2.3.2 GRU . . . 30
2.3.3 LSTM . . . 32
2.4 Revisão de leitura . . . 33
2.4.1 Análise Crítica . . . 37

3 Materiais e Métodos **39**

3.1 Exploração dos Dados . . . 39

3.1.1 Análise CO . . . 41

3.1.2 Análise NO2 . . . 44

3.1.3 Análise O3 . . . 46

3.2 Preparação dos Dados . . . 48

3.2.1 Feature Engineering . . . 48

3.2.2 *Missing Timestamps* e *Missing Values* . . . 49

3.2.3 Correlação e seleção de variáveis . . . 50

3.3 Análise temporal dos dados . . . 52

3.3.1 Anomalias . . . 52

3.3.2 Decomposição: Tendência e Sazonalidade . . . 53

3.3.3 Autocorrelação . . . 56

3.4 Tecnologias . . . 58

4 Experiências **59**

4.1 Concepção dos Modelos Candidatos . . . 59

4.2 Métricas de Avaliação . . . 62

4.3 Pipeline e Tecnologias . . . 63

5 Resultados e Discussão **65**

5.1 Seleção do modelo . . . 65

5.2 Técnicas de Explicabilidade . . . 67

5.2.1 Attention . . . 67

5.2.2 Shapley Values . . . 71

5.2.3 Feature Permutation . . . 77

5.3 Comparação de Resultados . . . 78

6 Conclusão **81**

Lista de Figuras

1 Participantes em conferências de IA. Fonte:[4] 2
2 Explicabilidade vs Performance.Fonte:[18] 3
3 Explicabilidade BlackBox.Fonte:[3] 4

4 Abordagens de **Explainable Artificial Intelligence (XAI)**. Fonte:[16] 11
5 Modelo substituto do **Local Interpretable Model Agnostic Explanations (LIME)**. Fonte:[50] 13
6 Gráfico de *Feature Importance* e F Scores. Fonte:[17] 14
7 LRP *Backpropagation*.Fonte:[55] 15
8 Exemplos de Séries Temporais.Fonte:[63] 18
9 Exemplo de **AutoCorrelation Function (ACF)**.Fonte:[63] 19
10 Exemplo de **ACF** com *White Noise*.Fonte:[63] 20
11 Métodos do cálculo das pontuações de atenção. Fonte:[91] 25
12 Exemplo de explicação do lime usando um modelo linear para um problema de classificação.Fonte:[15] 27
13 *Feature importance extraído do modelo interpretável criado pelo* **LIME**. Fonte:[15] . . . 27
14 Plot de explicação de valores de shap para duas mulheres do conjunto de dados de cancro do colo do útero. Fonte:[63] 28
15 Arquitetura de um bloco de uma **Recurrent Neural Network (RNN)**. 30
16 Arquitetura de um bloco de **Gated Recurrent Unit (GRU)**. Fonte:[102] 31
17 Arquitetura de um bloco de **LSTM**. Fonte:[102] 32

18 Série Temporal do **Monóxido de Carbono (CO)** 42
19 Distribuição CO 43
20 CO médio mensal 44
21 Série Temporal do NO_2 44

22 Distribuição NO_2 . . . 45
23 NO_2 médio mensal . . . 46
24 Série Temporal do **Ozono (O_3)** . . . 47
25 Distribuição O_3 . . . 47
26 O_3 médio mensal . . . 48
27 Matriz de Correlação . . . 51
28 Anomalias do NO_2 com espaço temporal de 1 mês . . . 53
29 Anomalias do CO com espaço temporal de 1 mês . . . 53
30 Decomposição sazonal do NO_2 . . . 54
31 Decomposição sazonal do CO . . . 55
32 Decomposição sazonal do O_3 . . . 55
33 Gráfico de autocorrelação do CO . . . 56
34 Gráfico de autocorrelação do NO_2 . . . 57
35 Gráfico de autocorrelação do O_3 . . . 58
36 Pipeline das experiências . . . 64
37 Valores reais vs Valores previstos . . . 66
38 Pipelines de previsão com Attention e sem Attention. Fonte [128] . . . 68
39 Valores reais vs Valores previstos . . . 70
40 Pesos de atenção atribuídos à sequência de input. . . . 70
41 Progressão dos pesos de atenção das *features* ao longo dos *timesteps*. . . . 72
42 Heatmap da soma dos os valores de importância por timestep . . . 73
43 Shap Values nos *timesteps* 1, 24 e 36. . . . 73
44 Shap Values nos *timesteps* 18, 20 e 22. . . . 74
45 Soma da importância de todos os timesteps . . . 75
46 Heatmap da soma dos valores de importância por *timestep* . . . 75
47 Soma da importância de todos os *timesteps* . . . 76
48 *Shap Values* nos *timesteps* 1, 24 e 36. . . . 77
49 Feature Importance Permutation . . . 78

Lista de Tabelas

1 Modelos Interpretáveis. Fonte:[17] 9
2 Combinações dos padrões da Figura 8 19
3 Classificação de modelos de exponencial smoothing. Fonte: [80]. 22
4 Tipos de explicabilidade usada nos artigos 38
5 Constituição do Dataset 40
6 Análise estatística 40
7 Escala de nível de index da qualidade do ar. Fonte: [114] 41
8 Número de correlações superiores a 0.45 52
9 Hiperparâmetros para otimizar o modelo 62
10 Resultados do top-5 dos modelos candidatos. 66
11 Performance do modelo com e sem Atention 70
12 Mapeamento das *input features* da Figura 40. 71
13 Top 3 *features* mais importantes 79

Siglas

CO Monóxido de Carbono.

NH_3 Amoníaco.

NO Monóxido de Nitrogénio.

NO_2 Dióxido de Nitrogénio.

O_2 Oxigénio.

O_3 Ozono.

PM_{10} Matéria Particular 10.

$PM_{2.5}$ Matéria Particular 2.5.

SO_2 Dióxido de Enxofre.

ACF AutoCorrelation Function.

ADF Augmented Dickey-Fuller.

ANN Artifical Neural Network.

AQGs Air Quality Guidelines.

AR Auto-Regression.

ARIMA Auto Regressive Integrated Moving Average.

Att-LSTM Attention-Long-Short Term Memory.

AUC Area under de ROC curve.

BPTT Back-Propagation Through Time.

CNN Convolutional Neural Network.

DL Deep Learning.

DNN Deep Neural Network.

EEA European Economic Area.

ES Exponential Smoothing.

FNN Feed-forward Neural Network.

GPDR General Data Protection Regulation.

GRU Gated Recurrent Unit.

IA Inteligência Artificial.

IDH Isocitrate Dehydrogenase.

kNN k-Nearest Neighbour.

KPSS Kwiatkowski-Phillips-Schmidt-Shin.

L1 Lasso.

L2 Ridge.

LIME Local Interpretable Model Agnostic Explanations.

LOESS Regressão Local.

LRP Layer-Wise Relevance Propagation.

LSTM Long-Short Term Memory.

MA Moving Average.

MAE Mean Absolute Error.

MIT Massachusetts Institute of Technology.

ML Machine Learning.

NN Neural Network.

NOx Óxidos de Azoto.

rCBV relative Cerebral Blood Volume.

ReLU Unidade Linear Retificada.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Network.

ROC Receiver Operating Characteristic.

SES Single Exponencial Smoothing.

SHAP Shapley Additive Explanations.

SVM Support Vector Machine.

SVR Support Vector Regression.

tanh Tangente Hiperbólica.

TCN Temporal Convolutional Network.

VAR Vector Autoregression.

VECM Vector Error Correction Model.

VOCs Compostos Orgânicos Voláteis.

WHO World Health Organization.

XAI Explainable Artificial Intelligence.

XOR Exclusive Or.

Capítulo 1

Introdução

A presente dissertação surge por meio da unidade curricular de Dissertação em Engenharia Informática, na Universidade do Minho, com o título "IA explicável para problemas de séries temporais". Este capítulo está dividido em 3 secções. Na secção 1.1 é feita uma introdução ao tema desta dissertação começando por explorar a origem e interesse na área de **IA** e **ML**, bem como a sua crescente notoriedade nos últimos anos. Além disso, é falado ainda da sua aplicação em diversas áreas do mercado e da saúde e da evolução dos modelos mais simples para modelos mais complexos, como redes neuronais. Posteriormente são apresentados os problemas e limitações dos modelos blackbox e apresentada a explicabilidade como solução deste problema. Na secção 1.2 são apresentados os objetivos a atingir no final desta dissertação. Por fim, na secção 1.3, é descrita a estrutura deste documento com todos os temas abordados em cada capítulo.

1.1 Enquadramento e Motivação

Nos últimos anos tem-se observado um aumento significativo no recurso a **IA** e **ML** para resolução de problemas em várias áreas da ciência e indústria, como *retail*, fabricação, educação, energia, finanças, segurança, saúde, etc. [1]. O aumento do poder computacional, juntamente com as grandes quantidades de dados disponibilizadas na internet permitiram desenvolver diversos modelos de **ML**. Estes modelos são capazes de concretizar grandes quantidades de tarefas com tempos de performance cada vez melhores e resultados aplicáveis em cenários reais [2]. Este crescimento é notado de forma variada na industria, financiamentos e aderência em conferências. Muitos actores industriais globais da Amazon à Zalando concentraram agora esforços internacionais em **IA**. A União Europeia tem como objetivo trazer um total de 20 bilhões de euros para a investigação em **IA** no futuro em ambos os sectores, público e privado, sendo a saúde um dos alvos centrais por ser um tópico que abrange muitas pessoas [3]. As conferências de **IA** indicam fortemente o nível de interesse académico e da industria nesta àrea. Estas não cresceram

apenas em número, mas também em prestígio. A Figura 1 mostra o número de participantes em várias conferências desde 1984 a 2019 [4].

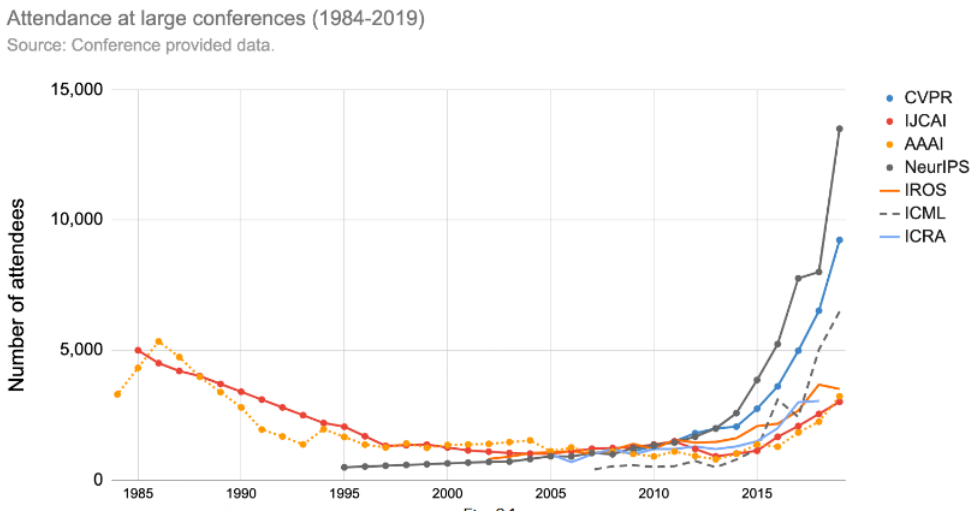


Figura 1: Participantes em conferências de IA. Fonte:[4]

Mas **IA** e **ML** não são termos recentes. Em 1957, Frank Rosenblatt criou o Perceptron, o primeiro protótipo do que conhecemos atualmente como **Neural Network (NN)**. O Perceptron tinha duas camadas de processamento que reconheciam padrões simples [5, 6]. Mais tarde, professores do **Massachusetts Institute of Technology (MIT)** descobriram que este não era nem capaz de aprender uma simples função **Exclusive Or (XOR)**, o que levou ao inverno da **IA**, época em que o interesse e financiamento caíram e não houve avanços na investigação nesta àrea [5, 6, 7].

A crescente aplicação de redes neuronais profundas ou **Deep Neural Network (DNN)** na medicina para deteção de doenças [8, 9, 10], na previsão do preço das ações [11], etc. fez os especialistas questionarem-se se as decisões tomadas pela máquina eram de facto a abordagem mais correta e porquê. Apesar do grande poder de processamento e dos modelos conseguirem adquirir valores de precisão e erro quase perfeitos, um grande problema é não conseguirmos saber de que forma estes modelos chegam às decisões alcançadas [12].

Para modelos de aprendizagem automática utilizados, por exemplo, em electrónica ou serviços de tradução online, a transparência e explicabilidade não são um requisito fundamental, desde que o desempenho destes sistemas seja suficientemente bom [2].

Quando os sistemas computacionais que são usados para resolver problemas de **IA** são complexos e opacos, isto é, o processo de tomada de decisão não é claro e intuitivo para os humanos, são denominados de modelos de *BlackBox* [13]. Esta descrição está diretamente ligada à intuição metafórica de que o comportamento de um sistema pode ser explicado quando "olhamos para dentro"para perceber porque

e como trabalha [14]. Quando estas perguntas permanecem sem resposta, então é impossível perceber o funcionamento das máquinas, o que leva os utilizadores a dúvidar das suas decisões [15].

Desta forma, podemos ter dois tipos distintos de modelos, transparentes e opacos. Os modelos transparentes típicos incluem **k-Nearest Neighbour (kNN)**, árvores de decisão, aprendizagem baseada em regras, rede Bayesiana, entre outros [16]. As decisões a partir destes modelos são frequentemente transparentes, embora a transparência, como propriedade, não seja suficiente para garantir que um modelo seja facilmente explicável [17]. Os modelos típicos opacos incluem Random Forest, **Artifical Neural Network (ANN)**, **Support Vector Machine (SVM)**, entre outros [18]. Embora estes modelos alcancem frequentemente uma elevada precisão, não são transparentes devido à sua complexidade [16]. É possível analisar estas relações entre interpretabilidade e performance na Figura 2, onde modelos com melhor performance são menos interpretáveis devido à sua complexidade.

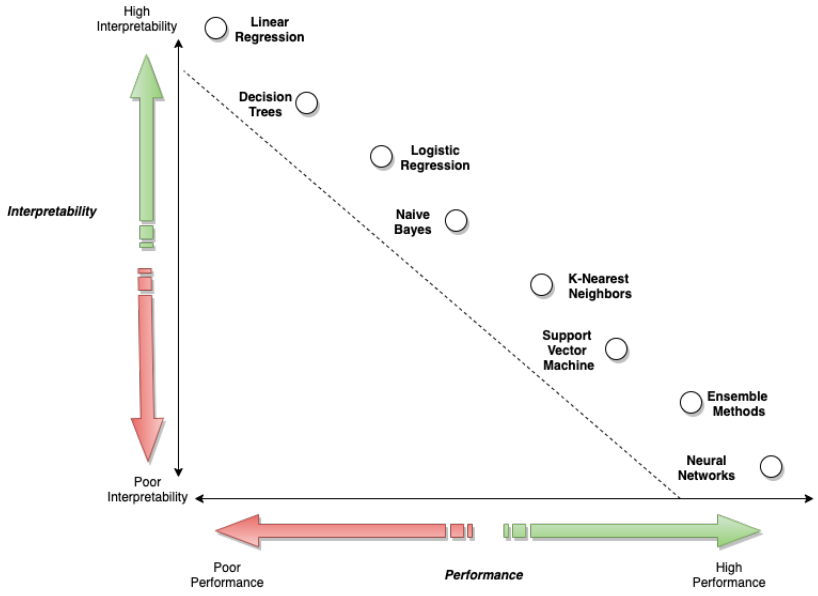


Figura 2: Explicabilidade vs Performance.Fonte:[18]

Heider afirma que as pessoas procuram por explicações para melhorar a sua compreensão de alguém ou algo para que assim seja possível derivar um modelo estável que pode ser utilizado para previsão e controlo [19]. Esta hipótese é apoiada por investigações que sugerem que as pessoas tendem a fazer perguntas sobre eventos ou observações que consideram anormais ou inesperadas do seu próprio ponto de vista [20].

Recentemente, a noção de **XAI** tem assistido a um ressurgimento, após ter abrandado desde a explosão do trabalho de explicação em sistemas de peritos [21]. Este ressurgimento é impulsionado por provas de que muitas aplicações de **IA** não têm a formalização do problema completa [22], ou não são apropriados de todo, devido a preocupações éticas e a uma falta de confiança em nome dos seus utilizadores

[21].

A interpretabilidade pode ajudar o programador e os utilizadores a adquirir conhecimentos adicionais sobre o problema e desenvolver a confiança no modelo de **ML** [22], por exemplo, no caso da medicina. Tal como um radiologista, que escreve um relatório a explicar as suas conclusões, um sistema de **IA** de apoio deve explicar em pormenor as suas decisões, em vez de apenas fornecer o diagnóstico ao perito [2]. A Figura 3 demonstra exatamente o problema dos modelos *BlackBox*. Sem uma explicação para os resultados, nem forma de saber o algoritmo de decisão, o utilizador final levanta sempre muitas questões sobre a precisão da previsão e confiabilidade da mesma. As explicações servem para responder a essas perguntas [23].

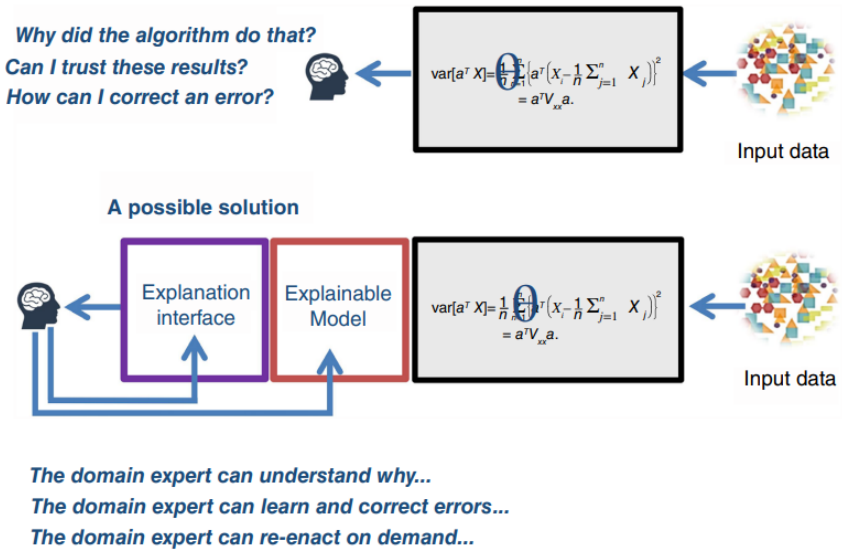


Figura 3: Explicabilidade BlackBox.Fonte:[3]

Para concluir, as razões mais importates para a necessidade de modelos de **XAI** são:

- **Conhecimento Científico:** A explicabilidade facilita aos investigadores a replicação dos seus projetos e experiências, fornecendo evidências sólidas para justificar as suas descobertas. Facilita ainda na comunicação dentro da comunidade cientifica e para o público em geral, permitindo a estes perceber de forma mais clara e apreciar a importância e relevância do seu trabalho [22, 24];
- **Segurança:** Nos dias de hoje em que humanos e máquinas com **IA**, convivem diariamente, é crucial garantir a segurança dos que interagem diretamente ou indiretamente com estas tecnologias [25]. Os operadores e engenheiros de segurança podem compreender melhor os factores que influenciam as decisões da **IA**, permitindo intervenções proactivas quando necessário. Ao fornecer explicações claras para as recomendações da **IA**, a **XAI** promove uma cultura de segurança,

capacita os operadores humanos para tomarem decisões informadas e contribui para a fiabilidade global e para a gestão do risco nos processos de fabrico [26]. Também no mercado de ações, ao compreender as razões por detrás de uma determinada previsão, os investidores podem avaliar melhor os potenciais riscos e benefícios de um determinado investimento, e podem tomar decisões mais informadas sobre a forma de gerir as suas carteiras [23, 27];

- **Confiança:** Ao aumentar a visibilidade da forma como os modelos de **IA** funcionam e o raciocínio subjacente às suas previsões, a **XAI** promove um nível mais elevado de confiança [25]. Nos veículos autónomos, perceber como estes visualizam o ambiente à sua volta e como reagem aos obstáculos do meio, ajuda-nos, como utilizadores, a ter confiança na **IA**, uma vez que a nossa vida e a de mais pessoas está em risco [28]. Outro exemplo é o uso de **IA** em grandes plantações. Para confiar nas tomadas de decisão de um modelo que toma conta de plantações é crucial saber como este toma as decisões para irrigação, fertilização, previsão do tempo, etc. Más decisões podem levar a perda de vários hectares de alimento [27, 29];

- **Questões éticas e legais:** Em alguns casos, a explicabilidade pode ser necessária por razões legais ou éticas. Por exemplo, na União Europeia, o **General Data Protection Regulation (GPDR)** exige que seja dada aos indivíduos uma explicação para qualquer tomada de decisão automatizada que os afete significativamente [27, 28, 30].

1.2 Objetivos

De acordo com o enquadramento elaborado no ponto anterior, esta dissertação tem como objetivos:

1. Investigar, na literatura existente, as aplicações de modelos de explicabilidade para séries temporais;

2. Explorar, entender, visualizar e analisar os dados relativos a séries temporais.

3. Estudar o problema de séries temporais e, de acordo com as suas características, efetuar o processamento dos dados;

4. Desenvolver modelos candidatos de **Deep Learning (DL)**, nomeadamente **LSTM**, para previsão de séries temporais;

5. Estudo da explicabilidade do melhor modelo candidato concebido;

6. Analisar e discutir os resultados obtidos.

Após a elaboração destas tarefas, é esperado que, do melhor modelo concebido, seja possível extrair as caraterísticas mais relevantes dos dados de séries temporais de forma a estudar a perceber o sentido da previsão obtida.

1.3 Estrutura do Documento

Este documento foi organizado em diversos capítulos para facilitar a leitura e compreensão. O Cap. 1 faz uma breve introdução ao tema da dissertação, com uma breve história e evolução da **IA**, finalizando com os problemas atuais e como estes fomentam o seu desenvolvimento. De seguida o Cap. 2 apresenta conceitos relevantes para o estado da arte da dissertação, relativos à explicabilidade dos modelos de **IA** e séries temporais. Ainda é feita uma revisão de literatura e análise crítica de modo a investigar o trabalho feito até ao momento. Posteriormente no Cap. 3 são apresentados os materiais e métodos utilizados para a realização desta dissertação. Este capitulo foca-se na exploração e análise estatística e temporal do *dataset* escolhido para realizar o estudo da explicabilidade, bem como o tratamento dos dados para a realização das experiências. No Cap. 4 são configuradas as experiências, bem como o processo de concepção dos modelos e a identificação das métricas de avaliação. Os resultados são posteriormente apresentados no capitulo 5, onde é feita uma análise individual sobre os resultados de todas as técnicas de explicabilidade aplicadas, bem como uma comparação entre elas. Por fim no Cap. 6 é feita uma conclusão face ao trabalho desenvolvido neste documento e uma reflexão sobre o trabalho a desenvolver no futuro.

Capítulo 2

Estado da Arte

Neste capítulo, é feito um levantamento dos conceitos fundamentais referentes ao estado de arte desta dissertação. Inicialmente, em 2.1 serão explicados os conceitos gerais de explicabilidade, métodos e abordagens diferentes para vários tipos de modelos. Além disso, serão abordados e explicados os modelos intrinsecamente explicáveis *ante-hoc*, métodos de explicabilidade *post-hoc*, a dimensão da explicabilidade local e global, a especificidade das abordagens dos modelos específicos e a abrangência da explicabilidade dos modelos agnósticos. Porteriormente, em 2.2 é feita uma introdução aos modelos *ante-hoc* explicáveis para séries temporais **Auto Regressive Integrated Moving Average (ARIMA)** e Exponencial Smoothing, e seguidamente os métodos de explicabilidade *post-hoc* para modelos de **DL**, nomeadamente **LIME**, **SHAP** e mecanismos de atenção. Em 2.3, são estudados os modelos de **DL** para previsão de séries temporais. Por fim, em 2.4, é feita uma revisão da literatura que explora o tema de explicabilidade aplicado a séries temporais e uma análise crítica sobre estes.

2.1 Explicabilidade: Conceitos e Técnicas

A explicabilidade pode ter diferentes níveis de profundidade, podemos querer explicar resultados de forma mais superfícial, ou perceber todos os cálculos feitos pelo modelo para as previsões obtidas [31]. O nível de detalhe que o comportamento de um modelo é explicado, dá-se o nome de granularidade [31]. Instituições como os hospitais, por exemplo, podem ter mais interesse em perceber as decisões da **IA** num todo e não apenas para pacientes individuais [2]. É importante no entanto não confundir granularidade com a explicabilidade local e global, uma vez que este conceito pode ser aplicado a qualquer um deles [25].

A capacidade de compreender o comportamento geral de um modelo de **ML** através de uma série de *inputs* é conhecida como explicabilidade global [32]. Esta proporciona uma ampla compreensão do processo de tomada de decisão do modelo e pode ser útil para identificar tendências ou padrões nos

dados [32]. Neste caso, a granularidade pode referir-se ao nível de detalhe em que o processo global de tomada de decisão de um modelo é explicado [25]. Uma possibilidade seria incluir uma explicação de alto nível da abordagem geral do modelo para fazer previsões, bem como uma explicação mais detalhada dos seus preconceitos e limitações específicas [25].

Por outro lado, no caso da explicabilidade local, esta refere-se à capacidade de explicar as previsões de um modelo de **ML** para um *input* específico ou grupo de *inputs* [33]. Este tipo de explicabilidade fornece uma compreensão detalhada de como o modelo está a tomar decisões para um caso particular, e pode ser útil para identificar erros ou enviesamentos no comportamento do modelo [31]. Neste caso, a granularidade pode referir-se ao nível de detalhe em que as razões para a previsão de um modelo para uma instância específica são explicadas. A granularidade pode variar desde uma explicação de alto nível, como a identificação das tendências ou padrões gerais que o modelo está a detectar nos dados, até uma explicação mais detalhada, como a identificação das características específicas ou variáveis de entrada que são mais importantes para a previsão [25].

Em geral, a explicabilidade local é mais detalhada e específica, mas pode não fornecer tantas informações sobre o comportamento global do modelo, enquanto a explicabilidade global fornece uma compreensão mais ampla do comportamento do modelo, mas pode não ser tão detalhada [33]. É frequentemente vantajoso utilizar uma combinação de técnicas de explicabilidade local e global com níveis de granularidade diferentes, tanto para estudar o comportamento de um modelo de **ML**, como para tornar a sua explicação fácil e útil [21].

Como falado anteriormente, os modelos distinguem-se entre transparentes e opacos. Essa distinção está diretamente relacionada com o tipo de explicabilidade que pode ser aplicada em cada um [34]. Dependendo da altura em que o método de explicabilidade é aplicado, podemos diferenciar entre construir modelos interpretáveis (explicabilidade *ante-hoc*) ou aplicar o método de explicabilidade *post-hoc* [16].

2.1.1 Explicabilidade Ante-hoc

A explicabilidade *ante-hoc* refere-se a métodos que são incorporados na conceção de um modelo de **ML** desde o início, com o objetivo de tornar o modelo mais interpretável e transparente [34]. Estes métodos são frequentemente incorporados na arquitetura ou no processo de formação do modelo e destinam-se a proporcionar uma compreensão mais intuitiva do processo de tomada de decisão do modelo [17]. A Tabela 1 apresenta uma visão geral de alguns tipos de modelos interpretáveis e as suas propriedades.

A linearidade, monotonicidade e interação entre *features*, são propriedades que ajudam a definir o nível de interpretabilidade destes modelos [17]. Um modelo é linear se a associação entre *feature*

Algoritmo	Linear	Monotono	Interação	Tarefa
Regressão Linear	Sim	Sim	Não	Reg
Regressão Logística	Não	Sim	Não	Class
Árvore de decisão	Não	Algumas	Sim	Reg, Class
Rule-Fit	Sim	Não	Sim	Reg, Class
Naive Bayes	Não	Sim	Não	Class

Tabela 1: Modelos Interpretáveis. Fonte:[17]

e alvo for modelada de forma linear. As restrições de monotonicidade asseguram que a relação entre uma *feature* e o alvo vai sempre na mesma direcção ao longo de toda a gama da *feature* [35]. Desta forma, um aumento do valor da *feature* leva sempre a um aumento ou sempre a uma diminuição do alvo [35]. A monotonicidade é útil para a interpretação de um modelo, porque facilita a compreensão de uma relação [17]. Quanto às interações, alguns modelos podem automaticamente incluir interações entre as *features* para prever o resultado [36]. É possível incluir interações em qualquer tipo de modelo, criando manualmente *features* de interacção [37]. Estas podem melhorar o desempenho preditivo, mas interações demasiado numerosas ou demasiado complexas podem prejudicar a interpretabilidade [17, 37].

Para além de estudar as propriedades, é necessário explorar o *modus operandi* destes modelos para conseguir explicar as suas decisões. Os modelos são considerados interpretáveis pois permitem facilmente repartir todos os seus componentes e ver como estes interagem entre si [36]. Além disso, estudar os parâmetros do modelo ajuda a perceber como este chega à decisão final, por exemplo, analisar os pesos de cada *feature* num modelo de regressão linear ajuda a explicar porque é que certos *inputs* geraram outros *outputs* [38].

À medida que a complexidade de um modelo aumenta, torna-se mais difícil repartir os seus componentes e, consequentemente, explicar os seus resultados. As **DNN**, por exemplo, são modelos altamente complexos, com muitas camadas e um grande número de parâmetros, que as pode tornar difíceis de interpretar [6]. Contudo, existem várias técnicas que podem ser utilizadas para fornecer uma interpretação *ante-hoc* destes modelos, o que significa que podem ser aplicadas durante o processo de treino, para compreender como é que este está a fazer previsões [39]. Algumas destas técnicas são:

- **Métodos de regularização:** Estes métodos acrescentam restrições ao modelo durante o processo de treino, tais como decadência de peso ou regularização **Lasso (L1)**/**Ridge (L2)** [40], o que pode tornar os parâmetros do modelo mais interpretáveis ao forçar os parâmetros para soluções

mais simples. Os primeiros métodos de restrições incluem a aplicação de *sparsity* que encoraja um modelo a utilizar menos *features* para uma previsão, ou a imposição de restrições monotónicas semânticas para permitir uma relação monotónica entre *features* e previsões [22]. Desta forma, consegue-se uma maior compreensão para o utilizador, mas o desempenho do modelo é significativamente reduzido [22].

- **Adversarial training:** Esta técnica implica acrescentar ruído ou perturbações à entrada durante o treino, de modo a tornar o modelo mais robusto e generalizar melhor [41]. A ideia por detrás desta abordagem é que ao regularizar o modelo com estas perturbações de entrada, o modelo teria de se concentrar nas características mais informativas e assim tornar o modelo mais interpretável [42].

- **Integrated gradients:** Esta técnica é uma forma de atribuir cada *feature* de um *input* com uma pontuação de importância para a previsão final, o que permite obter as importâncias da *feature* de uma forma ponta a ponta, ou seja, desde o *input* até à previsão, em oposição às técnicas que precisam de ser aplicadas depois de o modelo ter sido treinado [43].

É válido observar que estas técnicas não são mutuamente exclusivas, e muitas técnicas de interpretação de modelos combinam múltiplos métodos, a fim de proporcionar uma compreensão mais completa do comportamento do modelo [44]. As **DNN** são também menos interpretáveis do que os modelos mais simples, e os resultados das técnicas de interpretação podem ser difíceis de interpretar, especialmente quando se lida com dados altamente dimensionais e complexos [6].

2.1.2 Explicabilidade Post-hoc

A explicabilidade *post-hoc* refere-se a métodos que são aplicados após a formação de um modelo de **ML**, com o objetivo de fornecer uma explicação das previsões do modelo [21]. Estes métodos são tipicamente mais flexíveis e podem ser aplicados a uma vasta gama de modelos, mas podem não fornecer tanta perceção do funcionamento interno do modelo como as técnicas *ante-hoc* [21]. As técnicas *post-hoc* fornecem ainda explicações locais para uma decisão específica e tornam-na reprodutível a pedido (em vez de explicar todo o comportamento do sistema). Um exemplo representativo são as **LIME**, que é um sistema de modelos agnósticos onde o objetivo é identificar um modelo interpretável sobre a representação interpretável que seja localmente fiel ao classificador [15].

Os métodos *post-hoc* podem ser categorizados em *model agnostic* e *model-specific* [16]. No contexto de **XAI**, as abordagens *model agnostic* referem-se a métodos que podem ser utilizados para explicar as

previsões de qualquer modelo de **ML**, independentemente da sua arquitetura específica ou detalhes de implementação [45]. Estes métodos são geralmente mais flexíveis e podem ser aplicados a uma vasta gama de modelos, mas podem não ser tão poderosos como as abordagens model-specific [16]. Estas são concebidas para serem utilizadas com um tipo específico de modelo de aprendizagem de máquina, exigindo uma compreensão mais profunda da arquitetura do modelo, e podem fornecer explicações mais detalhadas e precisas das previsões do modelo [3].

As técnicas *model-specific* são frequentemente utilizadas em domínios onde é importante compreender o raciocínio subjacente às previsões do modelo, tais como diagnóstico médico, avaliação do risco de crédito, classificação da imagem e detecção de fraude [16]. Contudo, podem ser computacionalmente dispendiosas e podem não ser bem dimensionadas para grandes conjuntos de dados [22].

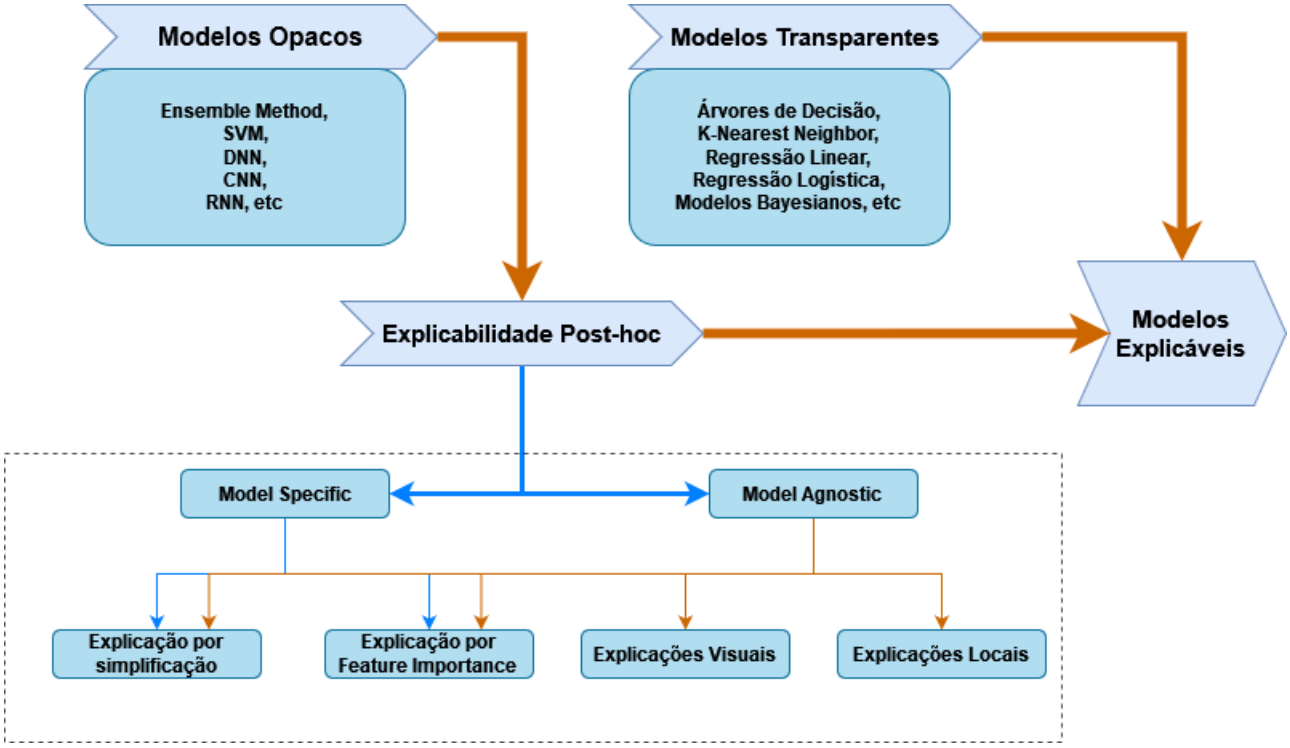


Figura 4: Abordagens de **XAI**. Fonte:[16]

A Figura 4 apresenta a ontologia da taxonomia **XAI**. Os modelos transparentes podem ser facilmente explicados, enquanto os modelos opacos exigem abordagens *post-hoc* para os tornar interpretáveis. As categorias de abordagens *post-hoc* estão ilustradas em conformidade.

Todos os termos mencionados até agora apenas tratam da dimensão da explicabilidade e em que altura (antes ou depois do treino) esta é obtida. Posteriormente, são apresentados exemplos de técnicas concretas que são aplicadas aos modelos para explicar as suas previsões.

Modelos Substitutos

Uma técnica de explicabilidade *post-hoc* é a substituição de modelos mais complexos por modelos mais simples e fáceis de interpretar [46]. Modelos simples tais como modelos lineares ou árvores de decisão pouco profundas são intrinsecamente interpretáveis, de modo que a explicação das suas previsões se torna uma tarefa trivial [47]. Um modelo substituto, é um modelo simplificado que é especificamente concebido para aproximar ou imitar o comportamento do modelo original, mas com características mais simples e interpretáveis [46]. Estes podem ser utilizados para explicar as previsões do modelo original e para fornecer explicações mais interpretáveis para as suas decisões, mas não tem de ser treinado com as mesmas entradas e saídas do modelo original [2, 16].

Uma das formas de obter um modelo substituto é através da extração de regras. A extração de regras é o processo de extrair regras de decisão de uma árvore de decisão, tornando o processo de decisão mais explícito e interpretável para o utilizador final [48]. Existem vários métodos de extração de regras de árvores de decisão, tais como algoritmos de cobertura, que procuram um conjunto de regras que abranja todas as instâncias na mesma folha, ou a divisão da árvore de decisão num conjunto de regras conjuntivas ou disjuntivas [48]. Estas regras de decisão são tipicamente declarações *if-then* que testam o valor de uma *feature* de entrada. Por exemplo, uma regra de decisão numa árvore de decisão para um problema de classificação binária pode ser "se a *feature* X for superior a algum limiar, então a decisão da árvore é 'classe A', senão a decisão é 'classe B'"[49]. Depois de extraídas, as regras podem ser apresentadas ao utilizador num formato legível por humanos, o que facilita a compreensão de como a árvore de decisão está a fazer as suas previsões [48]. Além disso, a extração de regras permite compreender como a árvore de decisão lida com diferentes casos e como melhorá-la ou utilizá-la como guia para tomar decisões em situações semelhantes [49].

Para modelos mais complexos, tais como **DNN** ou **RNN**, que contêm várias camadas de transformações não lineares, encontrar o que os faz chegar exatamente às suas previsões é uma tarefa mais difícil [39]. Uma abordagem para explicar as previsões de modelos complexos é, a nível local, aproximá-los com uma simples função de substituição ou por um modelo substituto, ambos interpretáveis [2, 16].

O algoritmo para treinar modelos locais de substituição pode ser descrito nos seguintes pontos:

1. Selecionar a instância de interesse para a qual se deseja ter uma explicação da previsão da *Black-Box*.

2. Perturbar o conjunto de dados e obter as previsões da *BlackBox* para estes novos pontos.

3. Pesar as novas amostras de acordo com a sua proximidade com a instância de interesse.

4. Treinar um modelo ponderado e interpretável no conjunto de dados com as variações.

5. Explicar a previsão através da interpretação do modelo local.

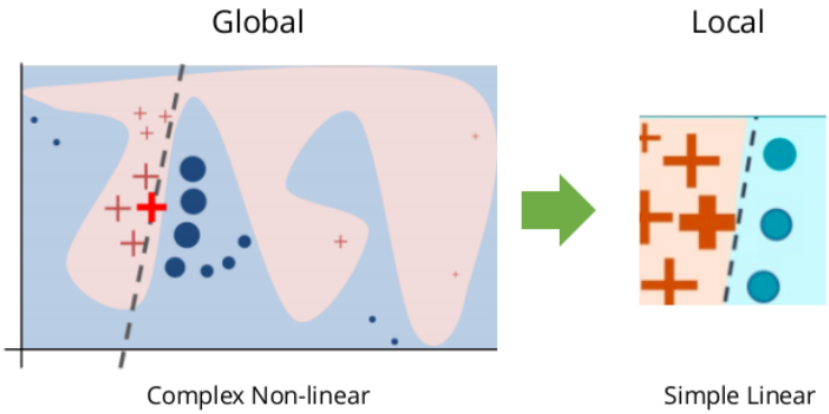


Figura 5: Modelo substituto do **LIME**. Fonte:[50]

Na Figura 5, podemos observar como o **LIME** utilizou uma função linear (reta tracejada) para representar o modelo global para um conjunto de dados local. Desta forma, é mais fácil interpretar os resultados do modelo global pela análise do modelo substituto mais simples.

Técnicas orientadas às features

A explicabilidade orientada às *features* é um tipo de interpretabilidade que se concentra na compreensão de como o modelo está a utilizar *features* de entrada individuais para fazer as suas previsões [15, 51]. O objetivo da explicabilidade orientada às *features* é determinar que *features* são importantes para o modelo, como o modelo utiliza essas *features*, e como o modelo é afetado por alterações às *features* individuais [40]. Este tipo de explicabilidade envolve tipicamente a análise do comportamento do modelo em relação aos *inputs* individuais, tais como o cálculo das pontuações de *feature importance*, a visualização do comportamento do modelo na vizinhança de pontos de entrada específicos, ou a análise da relação entre os *inputs* e as previsões do modelo [40]. Uma forma de perceber quais *features* têm mais relevância na hora de decisão do modelo é fazer pequenas variações locais a essa *feature* específica e avaliar de que forma o resultado final é alterado [52]. Desta forma é possível perceber, até um certo nível, o porquê de uma previsão ser tão diferente, ou não, de outra [52]. Por exemplo, se uma *feature* não tem muita relevância para a tomada de decisão, então uma grande variação dessa *feature* em dois inputs diferentes poderia gerar o mesmo *output* caso os outros parâmetros não tivessem grandes alterações [40]. Outros métodos orientados a *feature* são por exemplo, *feature importance* [40], **SHAP** [51] e **LIME** [15].

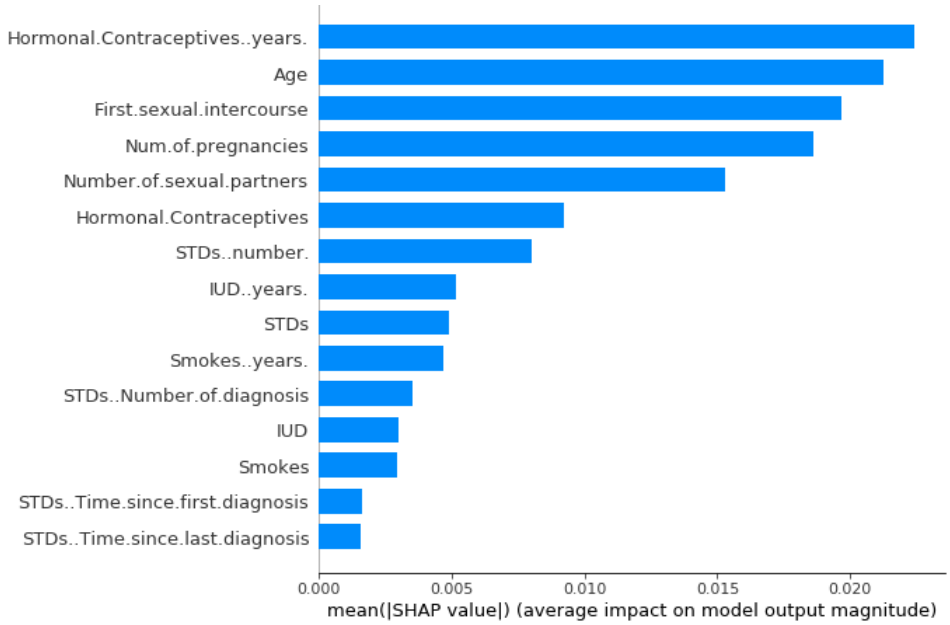


Figura 6: Gráfico de *Feature Importance* e F Scores. Fonte:[17]

Na Figura 6 podemos observar um gráfico da distribuição do impacto médio de cada *feature*, calculado pelo **SHAP** para o output de um determinado modelo. É de se esperar que as *features* com valores mais altos de importância tenham mais peso no *output* final dado pelo modelo.

Técnicas para ANN

Existem ainda algumas técnicas que, apesar de serem específicas às **ANN**, são agnósticas a qualquer tipo de **ANN**, como por exemplo Maximização de ativação e métodos baseados na propagação. A maximização de ativação é um método de interpretação *post-hoc* a nível de modelo que é utilizado para visualizar a representação aprendida por **DNN** [22]. Esta técnica consiste em encontrar o *input* que maximiza a ativação de um neurónio específico ou de um conjunto de neurónios numa **ANN** [53]. Esta abordagem é utilizada para compreender as *features* que são mais importantes para um neurónio específico, fornecendo um *input* que ativa mais esse neurónio [53]. Além disso, pode ser usado para gerar imagens que ativam um neurónio específico, chamadas "ativações de neurónios"[54].

Outra abordagem são os métodos baseados na propagação. Estes são um tipo de métodos de *feature importance* que utilizam os gradientes da saída do modelo em relação às *features* de entrada para estimar a importância de cada *feature* [55]. Os métodos baseados na propagação são particularmente adequados para as **DNN**. A ideia principal por detrás destes métodos é propagar os gradientes da saída de volta através da rede às *features* de entrada [43]. Apesar destes métodos serem específicos para **ANN**, são muito populares por serem capazes de lidar com não-linearidades [55, 56]. Ainda sobre os métodos

de propagação temos a **Layer-Wise Relevance Propagation (LRP)**, que é uma framework aplicável às estruturas gerais das rede neuronais artificiais, incluindo **DNN**, **LSTM**, e classificadores Fisher Vector [57]. A **LRP** explica as decisões individuais de um modelo através da propagação da previsão desde a saída até à entrada, utilizando regras de redistribuição local [43, 56]. Para tarefas de reconhecimento de imagem, o procedimento resulta num "heatmap"que indica quais os pixels da imagem que são importantes para uma determinada predição de rede neural [55, 58].

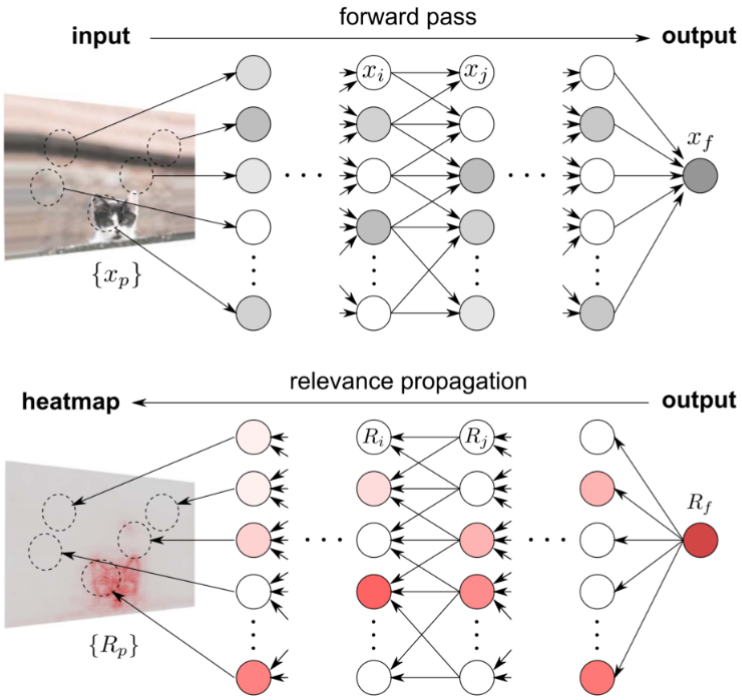


Figura 7: LRP *Backpropagation*.Fonte:[55]

Na Figura 7, podemos ver de que forma o **LRP** destaca os pixeis mais importantes da imagem a ser ingerida pela *backpropagation*, através da ativação dos neurónios da rede. Apesar de parecidos, o **LRP** é um método baseado na propagação que utiliza o algoritmo de *backpropagation* para calcular as pontuações de relevância de cada *feature* de entrada, enquanto que a maximização da ativação é uma técnica que maximiza a ativação de um neurónio específico, têm objetivos diferentes e, por conseguinte, proporcionam diferentes perspetivas [59].

Por fim, toda a informação obtida pelas técnicas mencionadas nesta secção deve ser organizada e apresentada aos utilizadores de forma a que seja fácil interpretar o seu significado. Desta forma, surgem os *heatmaps* ou *Saliency maps*, que são formas de visualização muito poderosas que nos ajudam a compreender as previsões dos modelos de **ML** [55, 58]. Todos os tipos e formas de atingir modelos mais interpretáveis podem e devem ser utilizados em conjunto para atingir maior compreensão do

funcionamento e tomada de decisões dos modelos [44].

2.2 Explicabilidade para séries temporais

Nesta secção é feita uma introdução aos conceitos de séries temporais. Além disso, é feita uma introdução aos modelos *ante-hoc* explicáveis para séries temporais **ARIMA** e *Exponencial Smoothing*, seguido de métodos de explicabilidade *post-hoc* para modelos de **DL**, nomeadamente **LIME**, **SHAP**, *Feature Permutation Importance* e mecanismos de atenção.

2.2.1 Princípios para previsão de séries temporais

Antes de se abordar as técnicas de explicabilidade para previsão de séries temporais, é importante perceber o que são séries temporais e os seus princípios. As séries temporais são um conjunto de observações com sequência cronológica sob uma variável de interesse [60]. Muitas empresas colecionam dados em intervalos diários, semanais, mensais, anuais, etc, mas qualquer intervalo de tempo é válido para fazer previsões. Além disso, os dados coletados podem ser instantâneos, como a temperatura de um líquido medida a uma certa hora, cumulativos, como o total de vendas de uma loja num mês, ou estatísticos de forma a representar a atividade de uma variável num intervalo de tempo [61]. A previsão de séries temporais são muito importantes em processos de decisão e planeamento em áreas como Gestão de operações, Marketing, Financiamento e Gestão de Riscos, Economia, Controlo de processo Industrial e Demografia [62, 63].

Apesar da vasta gama de problemas que requerem previsões, existem apenas dois tipos de técnicas de previsão. Um destes tipos é o método quantitativo. Neste método as previsões baseiam-se em modelos matemáticos e estatísticos que utilizam dados históricos [64]. Estes métodos incluem técnicas tais como **Moving Average (MA)**, *exponencial smoothing*, e modelos **ARIMA**. Estes métodos são tipicamente utilizados para fazer previsões sobre dados numéricos, tais como vendas ou preços de ações, e são frequentemente considerados mais objectivos e precisos, mas podem não ter em conta factores externos ou eventos inesperados [63, 64].

O outro tipo de técnica é o método qualitativo, que se baseia no julgamento especializado e na intuição, em vez de modelos matemáticos. Estes métodos incluem técnicas tais como Delphi, construção de cenários, e sondagens de opinião de peritos [65]. Estes métodos são tipicamente utilizados para fazer previsões sobre dados não-numéricos, tais como o sentimento do consumidor, as tendências do mercado ou quando há muitos poucos dados históricos. Podem ser mais flexíveis e adaptáveis, mas podem ser

menos fiáveis e precisos[63]. Na prática, é muitas vezes mais eficaz utilizar uma combinação de métodos quantitativos e qualitativos para fazer previsões uma vez que pode ajudar a fornecer uma imagem mais abrangente e precisa do futuro [64].

Existem ainda vários principios básicos das séries temporais que é necessário entender para se poder contruir os modelos para obter melhores previsões, como por exemplo as componentes. As componentes de uma séria temporal apontam padrões nos dados ao longo do tempo. Estas componentes são tendência, sazonalidade, ciclo e aleatório [66].

- **Tendência**: Existe uma tendência quando há um aumento ou diminuição a longo prazo dos dados, não tem de ser linear. Por vezes, pode acontecer uma mudança de direção e a tendência passar de crescente para decrescente e vice-versa. Esta componente pode ser extraída usando o modelo de crescimento logístico, ou pode ser extraída dos dados usando a técnica **MA** [63].

- **Sazonalidade**: Um padrão sazonal ocorre quando uma série temporal é afetada por fatores sazonais tais como a época do ano ou o dia da semana. Esta tem sempre um período fixo e conhecido. Pode ser modelada com séries de Fourier [63].

- **Ciclo**: Um ciclo ocorre quando os dados apresentam subidas e descidas que não são de uma frequência fixa. Estas flutuações são geralmente devidas a condições económicas, e estão frequentemente relacionadas com o "ciclo económico". A duração destas flutuações é normalmente de pelo menos 2 anos [63].

- **Aleatória**: Esta componente surge quando há flutuações aleatórias nos dados que não parecem ser muito previsíveis, nem há padrões fortes que ajudem a desenvolver o modelo de previsão [63].

A Figura 8 apresenta 4 exemplos de séries temporais e as possíveis combinações de padrões representados na Tabela 2. Na série temporal de vendas de casa podemos observar um padrão repetitivo com picos anuais, o que é um indicativo de sazonalidade. Além disso podemos observar que a cada 6 anos há uma quebra de vendas. Uma vez que esse padrão tem um espaço temporal maior que o anterior, o comportamento é avaliado como cíclico. Já nas séries temporais de contratos de tesouro existe uma tendência decrescente. Uma vez que há poucos dados observados, pouco se pode concluir em relação a sazonalidade, mas podesse deduzir que os valores continuação a diminuir se seguirem o mesmo comportamento dos valores observados. O gráfico da produção de eletricidade mostra uma série com tendência crescente e sazonalidade representada pelos vários picos consecutivos com o mesmo intervalo de tempo entre eles. Por fim, a última série de preço de fecho de stock da Google, não se consegue perceber nenhum padrão repetitivo, nem tendência pelo que a série é categorizada como irregular.

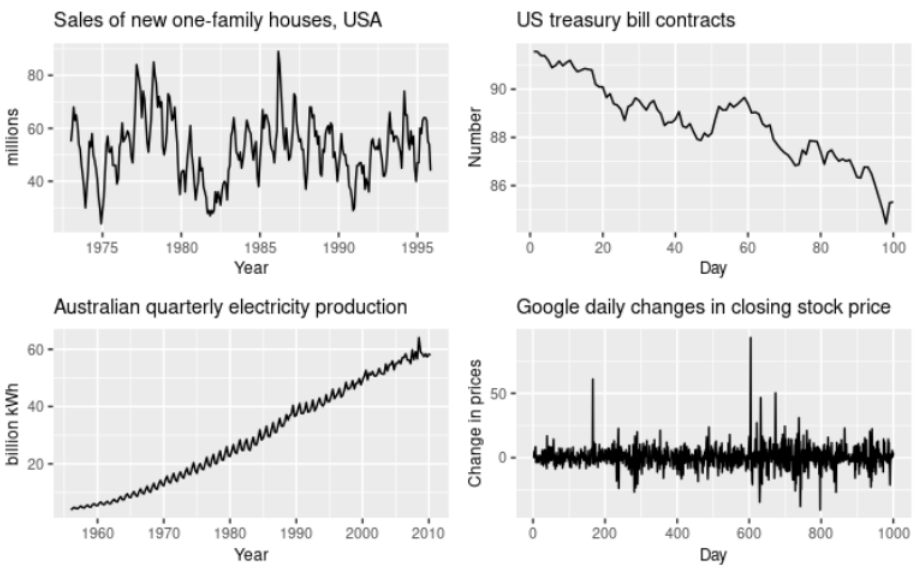


Figura 8: Exemplos de Séries Temporais.Fonte:[63]

É comum confundir o comportamento cíclico com o comportamento sazonal, mas na realidade são bastante diferentes. Se as flutuações não são de uma frequência fixa, então são cíclicas; se a frequência é imutável e associada a algum aspeto do calendário, então o padrão é sazonal . Em geral, a duração média dos ciclos é mais longa do que a duração de um padrão sazonal, e as magnitudes dos ciclos tendem a ser mais variáveis do que as magnitudes dos padrões sazonais [63]. Muitas séries temporais incluem tendência, ciclos e sazonalidade. Ao escolher um método de previsão, teremos primeiro de identificar os padrões das séries cronológicas nos dados, e depois escolher um método que seja capaz de captar os padrões corretamente [64].

As propriedades estatísticas das séries temporais também influênciam a sua categorização. Se e média, variância e auto-covariância não se alteram com o tempo, então a diz-se estacionária [67]. As séries temporais estacionárias são particularmente úteis para a previsão, porque são mais previsíveis do que as séries temporais [68]. Uma série temporal pode ser testada para a estacionaridade usando testes estatísticos, tais como o teste **Augmented Dickey-Fuller (ADF)** [69] ou o teste **Kwiatkowski-Phillips-Schmidt-Shin (KPSS)** [70]. Estes testes examinam se a média e a variância da série temporal são constantes ao longo do tempo e se existe uma raiz unitária (que é uma característica não estacionária) nos dados [67]. Caso os testes mostrem que a série não é estacionária, é possível transformá-la numa série estacionária através de um processo chamado diferenciação [71]. Este processo pode ser utilizado para remover tendências e sazonalidade dos dados, tornando-os mais adequados para previsão.

	Tendência	Sazonalidade	Ciclo	Irregular
Vendas de casas a novas famílias monoparentais, EUA	Não	Sim (anual)	Sim (6-10 anos)	Não
Contrato de títulos de tesouro, EUA	Sim (decrescente)	Não	Talvez (pouco tempo de amostra)	Não
Produção de eletricidade trimestral, Austrália	Sim (crescente)	Sim	Não	Não
Mudança diária no preço de fecho do stock, Google	Não	Não	Não	Sim

Tabela 2: Combinações dos padrões da Figura 8

A autocorrelação é outro princípio das séries temporais. Enquanto a correlação normalmente mede a extensão de uma relação linear entre duas variáveis, a autocorrelação mede a relação linear entre os valores de *lag* de uma série temporal [72]. Estas variáveis representam o valor de uma série temporal num ponto temporal anterior. Por exemplo, numa série com *lag* de 1, a observação atual é comparada com a observação anterior, e numa série com *lag* de 2, a observação atual é comparada com a observação que ocorreu há dois períodos de tempo atrás [63, 71]. Estas comparações são feitas através da inclusão das variáveis de *lag* como variáveis explicativas num modelo de regressão. Por exemplo, num modelo ARIMA, os valores passados das séries são utilizados como variáveis explicativas. As variáveis de atraso são também utilizadas na análise de modelos de séries temporais, tais como **Vector Autoregression (VAR)** e **Vector Error Correction Model (VECM)** para examinar a relação entre as diferentes variáveis ao longo do tempo [73]. Esta correlação entre as variáveis de atraso pode ser analisada em gráficos **ACF** como os das Figuras 9 e 10.

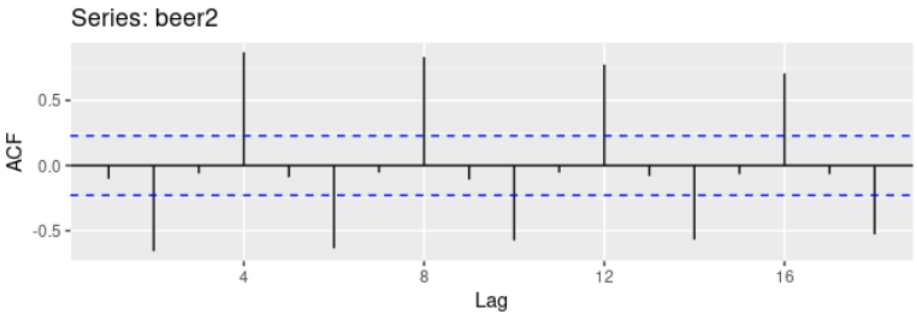


Figura 9: Exemplo de **ACF**.Fonte:[63]

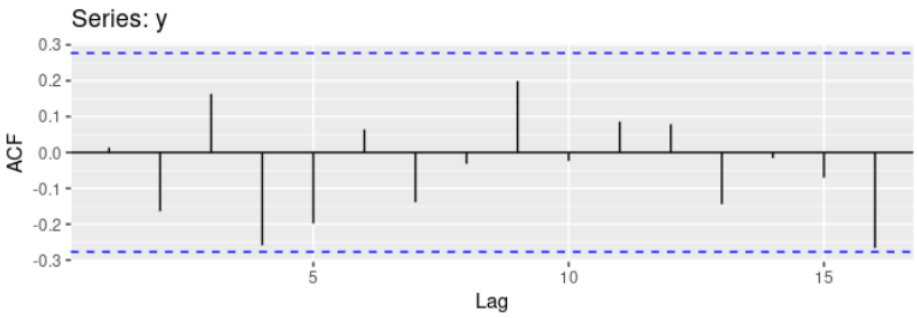


Figura 10: Exemplo de **ACF** com *White Noise*.Fonte:[63]

Na Figura 9 podemos observar que em $lag = 4$ há um pico positivo, e este pico repete-se de 4 em 4 tempos e em $lag = 2$ há um pico negativo, repetindo-se também de 4 em 4 tempos. Este fenómeno deve-se a um padrão de sazonalidade nos dados. As linhas azuis representam os limites segundo os quais as correlações têm relevência. No caso da Figura 10, esta série é chamada de white noise, uma vez que todos os valores de correlação estão abaixo das linhas azuis e por isso não têm quase nenhuma interferência nos dados futuros [63]. Depois de analisar os diferentes comportamentos de uma série temporal, nos próximos subcapitulos são apresentadas algumas técnicas de explicabilidade.

2.2.2 ARIMA

ARIMA é um modelo estatístico que é utilizado para analisar e prever dados de séries temporais, tais como dados sobre indicadores económicos, crescimento populacional, ou padrões meteorológicos [74]. O modelo tem em conta tanto os valores passados de uma série temporal como quaisquer tendências ou sazonalidade nos dados. **ARIMA** é composto por 3 componentes: **Auto-Regression (AR)**, Integração e **MA** [63, 75].

AR refere-se a um modelo estatístico em que o valor de uma variável num determinado momento é função dos seus próprios valores passados [76]. Nestes modelos, prevemos a variável de interesse utilizando uma combinação linear de valores passados da variável. O termo **AR** indica que se trata de uma regressão da variável contra si mesma [76]. Por exemplo, um modelo **AR** de ordem 1, muitas vezes escrito como AR(1), é um modelo em que o valor atual de uma série temporal é assumido como uma função linear do seu valor anterior mais algum termo de erro aleatório [77]. Um modelo AR(p) é um modelo em que o valor atual de uma série temporal é assumido como uma função linear dos "p"valores passados da série, mais um termo de erro aleatório. A ordem de **AR** é representada pelo parâmetro "p"no modelo [63].

Um modelo **AR** de ordem p, pode ser escrito da seguinte forma:

$$Y_t = C + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \varepsilon_t \tag{2.1}$$

Por outro lado, **MA** refere-se a um modelo estatístico em que o valor de uma variável num determinado momento é uma função da média de um número fixo de termos de erro passados ou residuais [76]. Por outras palavras, um modelo de **MA** assume que o valor atual de uma série temporal está relacionado com um número de erros passados ou residuais, em vez de valores passados da série [76]. A ordem de **MA** é representada pelo parâmetro "q"no modelo. Por exemplo, um modelo de **MA** de ordem 1, muitas vezes escrito como MA(1), é um modelo em que o valor atual de uma série é assumido como uma função linear do termo de erro no passo de tempo anterior [77]. Um modelo MA(q) é um modelo em que o valor atual de uma série é suposto ser uma função linear dos erros q passados ou residuais [63].

Um modelo **MA** de ordem p, pode ser escrito da seguinte forma:

$$Y_t = C + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + ... + \theta_p \varepsilon_{t-p} \tag{2.2}$$

Desta forma um modelo de ordem p com as 3 componentes pode ser escrito:

$$Y'_t = C + \phi_1 Y'_{t-1} + \phi_2 Y'_{t-2} + ... + \phi_p Y'_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + ... + \theta_p \varepsilon_{t-p} + \varepsilon_t \tag{2.3}$$

em que Y'_t representa a série diferenciada. A estes modelos dá-se o nome de ARIMA(p,d,q).

Estes modelos são facilmente interpretáveis pois apenas é necessário analisar o valor dos 3 parâmetros e de que forma estes contribuem para a previsão da série. Por exemplo, um alto valor de *p* indica que o modelo está a colocar muito peso nos valores passados da série, o que pode indicar que a série está altamente correlacionada com os seus valores passados. Um valor elevado de *d* indica que o modelo é responsável por uma não-estacionariedade na série, o que pode indicar que a série tem uma tendência ou um componente sazonal. Um valor elevado de *q* indica que o modelo está a colocar muito peso nas flutuações de curto prazo da série, o que pode indicar que a série tem muito ruído ou volatilidade [78].

2.2.3 Exponencial smoothing

Historicamente, a **Exponential Smoothing (ES)** descreve uma classe de métodos de previsão. Existem diversos métodos que se enquadram na família de exponencial smoothing, tendo cada um deles a propriedade de que as previsões são combinações ponderadas de observações passadas, sendo as observações recentes dadas relativamente mais peso do que as observações mais antigas [79].

A **ES** é uma família de modelos de previsão de séries temporais que se baseiam na ideia de utilizar uma média móvel ponderada para prever valores futuros [80]. Estes modelos são amplamente utilizados para a previsão de séries temporais porque são simples de compreender, fáceis de implementar, e muitas vezes fornecem bons resultados [81]. O modelo utiliza um parâmetro de alisamento (alfa) que controla o peso dado à observação mais recente[63].

A classificação dos modelos de exponencial smoothing é feita com base na tendência e sazonalidade [82]. Uma vez que a tendência é uma combinação do nível (l) e termo de crescimento (b), todas as combinações criam 5 tipos de tendência [82]. Seja T_h a tendência para os próximos *h* períodos de tempo e ϕ o parâmetro de amortecimento $(0 < \phi < 1)$:

$$Nenhum : T_h = l \tag{2.4}$$

$$Aditivo : T_h = l + bh \tag{2.5}$$

$$Additivo\ amortecido : T_h = l + (\phi + \phi^2 + ... + \phi^h)b \tag{2.6}$$

$$Multiplicativo : T_h = lb^h \tag{2.7}$$

$$Multiplicativo\ amortecido : T_h = lb^{\phi+\phi^2+...+\phi^h} \tag{2.8}$$

Se adicionar-mos também a componente sazonal de forma aditiva ou multiplicativa teremos então 15 combinações possíveis como mostra a Tabela 3.

Tendência	**Sazonalidade**		
	N (Nehuma)	**A (Aditiva)**	**M (Multiplicativa)**
N (Nenhuma)	N,N	N,A	N,M
A (Adtiva)	A,N	A,A	A,M
Ad (Aditiva Amortecida	Ad,N	Ad,A	Ad,M
M (Multiplicativa)	M,N	M,A	M,M
Md (Multiplicadiva Amortecida)	Md,N	Md,A	Md,M

Tabela 3: Classificação de modelos de exponencial smoothing. Fonte: [80].

Os modelos mais usados são o (N,N), também conhecido como **Single Exponencial Smoothing (SES)**, o (A,N) ou Holt's Linear Method e o (A,A) e (A,M), conhecido como Holt's Winter Exponential Smoothing Aditivo e Multiplicativo, respetivamente [80].

O **SES** é utilizado para previsões de curto alcance no futuro, isto é, para prever dados até um mês no futuro [83]. O modelo assume que os dados não variam muito além de uma média estável, ou seja,

não há tendência ou crescimento. Uma forma de representar este modelo em componentes é a seguinte [63]:

$$Previsao : Y'_{t+h|t} = l_t \tag{2.9}$$

$$Alisamento : l_t = \alpha Y_t + (1 - \alpha) l_{t-1} \tag{2.10}$$

O Holt's Linear Method é utilizado para séries que apresentem tendência e por isso usa dois parâmetros de alisamento, α e β [84]. Para este método é necessário calcular para cada período um novo nível e tendência. O nível (l) é o valor estimado amortecido dos dados no final de cada período. A tendência é o valor estimado amortecido da média do crescimento (b) no final de cada período [84]. Este modelo é representado da seguinte forma [83]:

$$Previsao : Y'_{t+h|t} = l_t + hb_t \tag{2.11}$$

$$Nivel : l_t = \alpha Y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \tag{2.12}$$

$$Tendencia : b_t = \beta(l_t + l_{t-1}) + (1 - \beta) b_{t-1} \tag{2.13}$$

O último método, conhecido como Holt-Winter's além de usar os parâmetros α e β, usa ainda outro parâmetro para amortecer a sazonalidade, γ [85]. Há ainda uma nova variável m que denota a periodicidade da sazonalidade. Por exemplo $m = 4$, indica sazonalidade trimestral e $m = 12$ indica sazonalidade anual. A sazonalidade pode ser adicionada às equações anteriores de forma aditiva (A,A) ou multiplicativa (A,M) [63, 83]. Por exemplo, se todos os anos em dezembro, uma loja aumenta as vendas em 1 milhão de euros, então todos os anos podemos adicionar esse valor às previsões para dezembro (sob o respetivo valor anual médio) para contar esta variação sazonal. Neste caso estamos perante sazonalidade aditiva e as respetivas fórmulas são:

$$Previsao\ Aditiva : Y'_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)} \tag{2.14}$$

$$Nivel\ Aditivo : l_t = \alpha(Y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}) \tag{2.15}$$

$$Tendencia : b_t = \beta(l_t + l_{t-1}) + (1 - \beta) b_{t-1} \tag{2.16}$$

$$Sazonalidade\ Aditiva : s_t = \gamma(Y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \tag{2.17}$$

Se em vez de um aumento fixo de 1 milhão de euros, tivermos um aumento segundo um fator, por exemplo aumento de 40% anual, então estamos perante sazonalidade multiplicativa e calcula-se da seguinte forma:

$$Previsao\ Multiplicativa : Y'_{t+h|t} = (l_t + hb_t)s_{t+h-m(k+1)} \tag{2.18}$$

$$Nivel\ Multiplicativa : l_t = \alpha(\frac{Y_t}{s_{t-m}}) + (1-\alpha)(l_{t-1} + b_{t-1}) \tag{2.19}$$

$$Tendencia : b_t = \beta(l_t + l_{t-1}) + (1-\beta)b_{t-1} \tag{2.20}$$

$$Sazonalidade\ Multiplicativa : s_t = \gamma(\frac{Y_t}{l_{t-1} + b_{t-1}}) + (1-\gamma)s_{t-m} \tag{2.21}$$

A variável *k* representa a parte inteira de $\frac{h-1}{m}$ que assegura que a estimativa dos indices sazonais utilizados para previsão são os do último ano dos dados.

Ao interpretar um modelo de exponencial smoothing, é importante considerar os valores dos parâmetros e como estes se relacionam com as características dos dados [82]. Por exemplo, um valor elevado do parâmetro de amortecimento α indica que o modelo está a colocar mais peso nas observações mais recentes, o que pode indicar que a série é altamente volátil. O mesmo raciocínio vale para β e γ. É importante notar que os valores desses parâmetros dependem das caraterísticas da série e os melhores valores são atingidos por processos de tentativa e erro [82].

2.2.4 Permutation Importance

Introduzindo agora técnicas *post-hoc*, nesta secção começamos por abordar a *Permutation Importance*. A *Permutation Importance* de *features* é um método de explicabilidade utilizado em **ML** para avaliar a influência relativa e individual das *features* no desempenho de previsão de um modelo [17]. Esta técnica consiste em reorganizar sistematicamente os valores de uma *feature* específica, mantendo as restantes *features* inalteradas e, em seguida, avaliar a alteração resultante na métrica de desempenho do modelo, como a *accuracy* ou o **Mean Absolute Error (MAE)**. Esta ligação com a métrica e performance torna fácil as interpretações dos seus resultados uma vez que uma queda substancial no desempenho após a permutação indica que a *feature* é crucial para as previsões do modelo. No entanto, em alguns casos

é preferível perceber o que a alteração de uma *feature* influência no *output* do modelo sem considerar como afeta a performance [17]. Esta abordagem serve múltiplos objectivos, incluindo a *feature selection*, onde ajuda a identificar e a dar prioridade às variáveis com maior impacto, reduzindo assim potencialmente a dimensionalidade do conjunto de dados e aumentando a eficiência computacional. Além disso, a *Permutation Importance* de *features* contribui significativamente para a explicabilidade do modelo. Ao elucidar a importância relativa de cada *feature*, fornece às partes interessadas informações sobre o processo de tomada de decisão do modelo [86]. A interpretabilidade oferecida pela *Permutation Importance* de *features* alinha-se com a procura crescente de modelos de **ML** transparentes e responsáveis [87]. A *Permutation Importance* também é uma técnica agnóstica, o que permite a sua aplicação universal, independentemente do modelo utilizado. Ainda assim, a sua utilização encontra limitações quanto ao *dataset* em que é utilizada, uma vez que em *features* com correlações elevadas, o valor da *Permutation Importance* dessas *features* será baixo [88].

2.2.5 Mecanismos de Atenção

Os mecanismos de atenção funcionam ao permitir que uma rede neuronal se concentre seletivamente em certas partes do *input* ao atribuir um peso, ou pontuação de atenção, a cada elemento da entrada [89]. Esta pontuação representa a importância ou relevância de cada elemento em relação à tarefa em questão [90]. O mecanismo de atenção consiste tipicamente em dois componentes principais: uma camada de atenção, e um vetor de contexto [91].

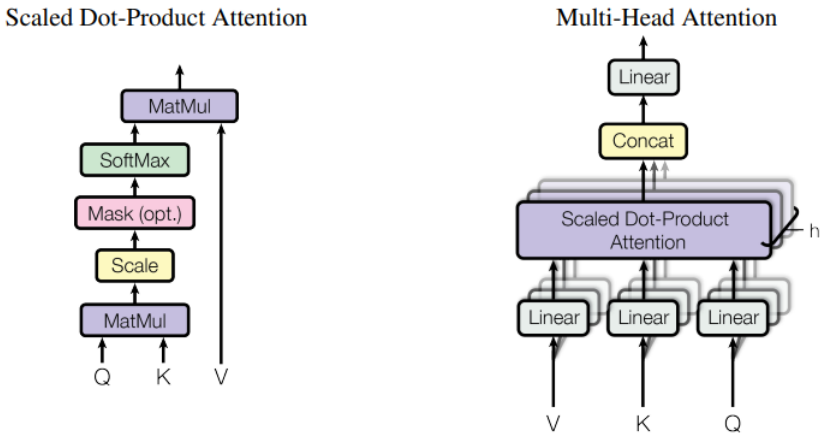


Figura 11: Métodos do cálculo das pontuações de atenção. Fonte:[91]

A camada de atenção é responsável pelo cálculo das pontuações de atenção para cada elemento da entrada. Pode utilizar métodos diferentes para calcular as pontuações de atenção, tais como *dot product*

ou *multi-head attention* [91]. As arquiteturas destas abordagens estão representadas na Figura 11. Em geral utiliza o *input*, uma *query* e uma chave, e devolve as pontuações de atenção. O vetor de contexto é uma soma ponderada do *input*, onde os pesos são as pontuações de atenção. É utilizado para resumir a informação mais importante na entrada, e é passado para as camadas seguintes da rede [91]. Existem várias camadas de atenção que computam as pontuações de forma diferente. A melhor camada a utilizar depende do tipo de dados que compõem a série temporal e do objetivo da previsão [92].

No contexto de séries temporais as camadas de *attention* são muito úteis para identificar os *timesteps* mais importantes. Ao atribuir um peso mais elevado aos *timesteps* mais relevantes e menos peso aos *timesteps*, é possível perceber a *feature importance* ao longo do tempo. Em séries multivariadas, só temos a evolução da importância de uma *feature* ao longo do tempo, mas também podemos retirar informação sobre como *timesteps* passados de uma *feature* afetam a variável de previsão, ajudando a perceber novas relações entre estas [92].

Visualizar cada uma das pontuações computadas num *heatmap*, é uma forma simples de compreender em que partes do *input* a rede se está a focar e como as pontuações de atenção mudam com o tempo[93]. Mecanismos de atenção estão também no *core* dos transformadores , que podem detetar variáveis globalmente importantes para o problema de previsão, temporais padrões persistentes, e eventos significativos que conduzem a mudanças significativas na dinâmica temporal[89].

2.2.6 LIME

Outra técnica de explicabilidade é o **LIME**, que tem como objetivo identificar um modelo interpretável sobre a representação interpretável que é localmente fiel ao classificador ou regressor [15]. Por outras palavras, a **LIME** treina um modelo substituto interpretável para aprender o comportamento local das previsões de um modelo global de *BlackBox* [15]. Seja *g* um modelo interpretável de uma classe de modelos interpretáveis G, *f* a função que representa o modelo a ser explicado, $(\pi_x)Z$ a medida de aproximação da instância Z a *x*, para definir localidade a volta de *x*, $\mathcal{L}(f, g, \pi_x)$ a medida de quão não-confiável *g* é na aproximação de *f* à localidade π_x e $\Omega(g)$ a medida de complexidade de explicabilidade de *g* [51, 94], a explicabilidade de **LIME** é calculada pela seguinte equação:

$$\xi(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2.22}$$

Na Figura 12, a complexa função de decisão *f* é representado pelo fundo azul/rosa, que não pode ser bem aproximado por um modelo linear. A cruz vermelha destacada é o exemplo a ser explicado. O **LIME** cria amostras e obtém previsões usando *f*, que são pesadas pela proximidade à instância que está a ser

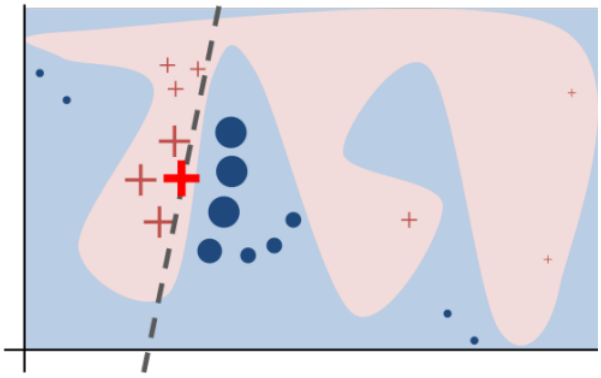


Figura 12: Exemplo de explicação do lime usando um modelo linear para um problema de classificação.Fonte:[15]

explicada. A linha tracejada é a explicação aprendida (g) que é fiel localmente (mas não globalmente).

Uma vez treinado o modelo interpretável, a **LIME** utiliza-o para estimar a *feature importance*, interpretando os coeficientes do modelo linear ou o caminho de decisão da árvore [15]. Desta forma, a **LIME** gera uma medida de *feature importance* que visa explicar como o modelo está a utilizar características de entrada individuais para fazer as suas previsões. Este cálculo da *feature importance* é uma abordagem para atingir a explicabilidade integrada nos métodos orientados às *features* [51].

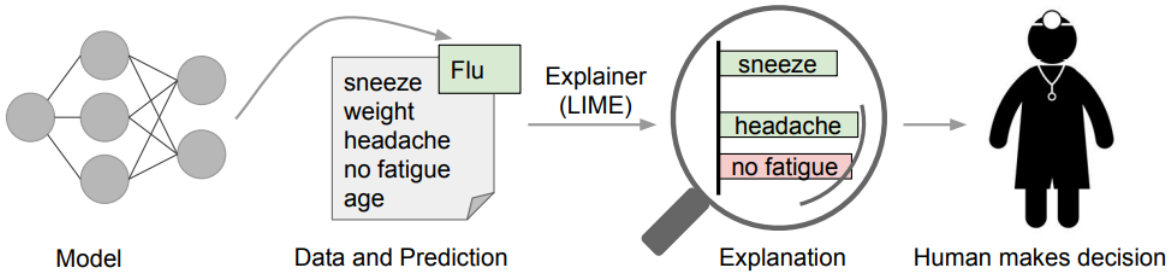


Figura 13: *Feature importance extraído do modelo interpretável criado pelo* **LIME**. Fonte:[15]

Na Figura 13, podemos ver o processo de extração da *feature importance* após o cálculo do modelo interpretável. Desta forma é possível, a nível local, explicar as previsões do modelo complexo inicial e ajudar as pessoas na tomada de decisões.

2.2.7 SHAP

SHAP é um método para interpretar e compreender as previsões de qualquer classificador ou regressor. Baseia-se no conceito de valores *shapley* da teoria do jogo cooperativo [95], e fornece uma explicação da contribuição de cada característica para a previsão final. Este utiliza uma técnica chamada amostragem

para estimar a diferença média nas previsões que resultam de uma *feature* estar presente ou ausente no *input* [51].

Para cada *feature*, **SHAP** gera um conjunto de amostras que incluem o valor original da *feature* e um conjunto de outros valores de *features* amostrados a partir da sua correspondente distribuição marginal [96]. Para cada amostra, é calculada a previsão do modelo, e é calculada a diferença entre a previsão com a *feature* presente e a previsão com a *feature* ausente [51].

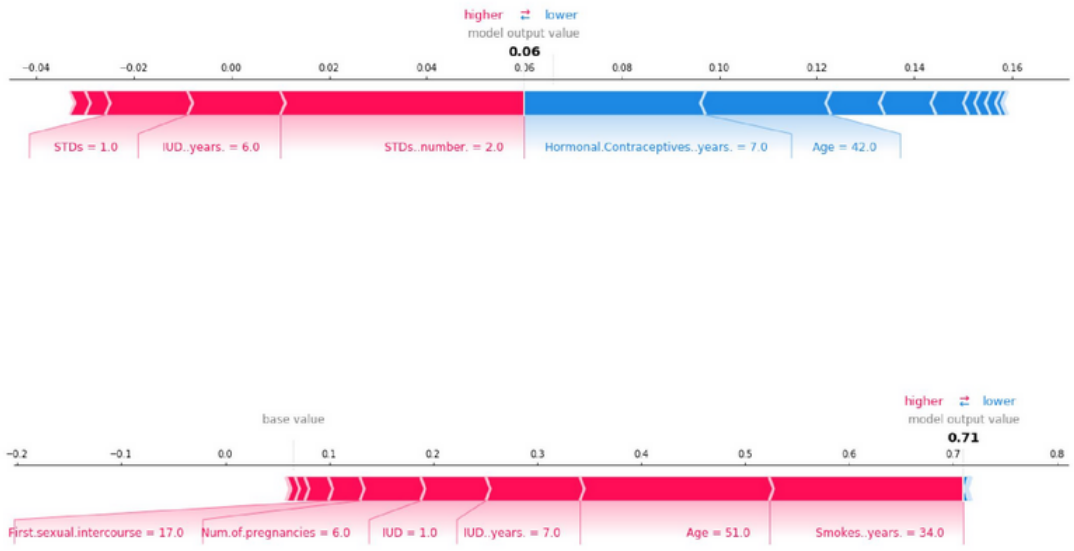


Figura 14: Plot de explicação de valores de shap para duas mulheres do conjunto de dados de cancro do colo do útero. Fonte:[63]

Na Figura 14 podemos observar os *shap values* para as várias *features* de um problema de previsão de cancro da mama em mulheres. A primeira mulher, representada pelo primeiro gráfico, tem um baixo risco previsto de 0,06. Os efeitos de aumento do risco, tais como as DSTs, são compensados por efeitos decrescentes, como a idade. A segunda mulher, representada pelo segundo gráfico, tem um elevado risco previsto de 0,71. A idade de 51 e 34 anos de tabagismo aumentam o risco de cancro previsto.

Atualmente existem várias versões **SHAP** disponíveis na biblioteca de *Python*, no entanto todos eles calculam os *shap values* através de um método baseado em *Feature Permutation*. A versão de *kernel* do **SHAP**, *KernelSHAP*, tal como a maioria das técnicas baseados em permutações, não tem em conta a dependência das *features*, pelo que, muitas vezes, dá demasiada importância a pontos de dados improváveis. Já o *TreeSHAP*, a versão para modelos de árvores, resolve este problema. No entanto, dependência de previsões condicionais esperadas produz valores de importância das *features* não intuitivos, uma vez que, *features* sem impacto nas previsões podem receber um valor de importância diferente de zero [87].

A análise e a previsão de séries temporais, como qualquer outra área que possa beneficiar da **IA**, precisa de incorporar mecanismos que ofereçam transparência e explicabilidade dos seus resultados. No entanto, em **DL**, a utilização destes mecanismos para séries temporais não é um tarefa fácil, devido à dependência temporal dos dados. No entanto, ainda é possível tirar bom proveito de **LIME** e **SHAP** no contexto de séries temporais. Podemos utilizar estes mecanismos para explicar como o modelo chega às previsões em pontos no tempo diferentes [97]. Uma forma de o fazer, é calcular a *feature importante* em cada *time step* e comparar a importância de cada *feature* ao longo do tempo. Desta forma, é possível identificar quais *features* são mais importantes e como essa importância varia em diferentes alturas temporais.

2.3 Deep Learning para séries temporais

Em **DL**, apesar de se utilizar vários tipos de **ANN** para previsão de séries temporais, há que destacar o poder de precisão das **RNN**, **GRU** e em específico as **LSTM**, pois estes modelos são capazes de detetar e aprender padrões não lineares e dependências temporais nos dados, como é demonstrado nesta secção.

2.3.1 RNN

Uma **RNN** é um tipo de **ANN** concebida para processar sequências de dados [98]. Este tipo de redes são particularmente úteis para tarefas que envolvem dados de séries temporais, processamento de linguagem natural, reconhecimento de voz e outras aplicações em que a ordem e o contexto dos pontos de dados são importantes [99]. A principal caraterística das **RNN** é a presença de conexões recorrentes. Essas conexões permitem que a rede mantenha um *hidden state* que captura informações sobre os *timesteps* anteriores na sequência [100]. A cada *timestep* a **RNN** recebe um novo *input* e combina-o com o *hidden state* anterior, de forma a produzir um *output* e atualizar o *hidden state*. O *hidden state* atualizado é então usado para calcular o *timestep* seguinte [99].

Na equação 2.23 é possível observar a fórmula para calcular o *hidden state*. Neste caso a função de ativação utilizada é a **Tangente Hiperbólica (tanh)**, mas não é exclusiva. W_{xh} representa o peso do *input* atual x_t e W_{hh} representa o peso do *hidden state* anterior h_{t-1}. Na Figura 15, podemos ver de que forma são calculados os *hidden states* num bloco de **RNN**.

$$Hidden\ State : ht = tanh(W_{xh}x_t + W_{hh}h_{t-1}) \tag{2.23}$$

Uma das desvantagens significativas das **RNN** para a previsão de séries temporais é o problema

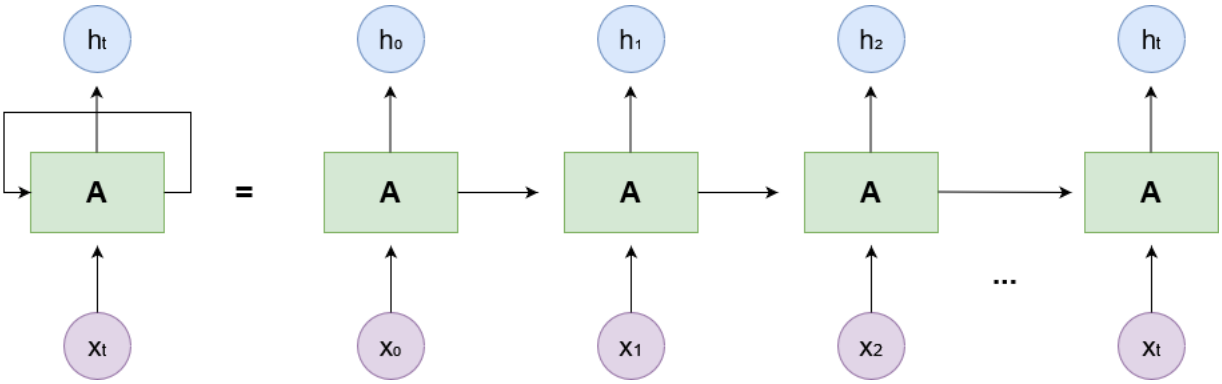


Figura 15: Arquitetura de um bloco de uma **RNN**.

dos *vanishing gradients* [98]. Ao calcular gradientes para os parâmetros da rede durante o treino, os gradientes podem tornar-se muito pequenos, aproximando-se de zero, à medida que se propagam para trás no tempo [101]. Esta diminuição dos gradientes indica que o modelo tem dificuldade em aprender as dependências de longo prazo nos dados, que são cruciais para uma previsão que se aproxime da realidade das séries cronológicas [99, 102].

Para resolver este problema, foram desenvolvidas várias arquitecturas avançadas de **RNN**, **LSTM** e a **GRU**. Estas arquitecturas incluem mecanismos de *gating* que ajudam a mitigar o problema dos *vanishing gradients*, além de permitem que as **RNN** captem de forma mais eficaz as dependências de longo alcance, tornando-as mais adequadas para tarefas de previsão de séries temporais [103, 104].

2.3.2 GRU

As **GRU** são uma arquitetura desenvolvida a partir das **RNN** para resolver o problema dos *vanishing gradients*. A solução foi a introdução de mecanismos de portas que controlam o fluxo de informação na rede [103]. Estes mecanismos de controlo ajudam este modelo a captar as dependências de longo alcance e a atenuar o problema dos *vanishing gradients* que podem ocorrer nas **RNN** tradicionais. As **GRU** possuem duas portas. Uma *update gate* que determina quanto do *hidden state* anterior deve ser transportado para o *timestep* atual, isto é, decide qual a informação a reter e qual a informação a descartar do estado anterior. Uma *reset gate* que decide quanto do *hidden state* passado deve ser esquecido ao calcular o estado atual, ou seja, controla a relevância do *hidden state* passado para o *timestep* atual [105]. Assim como nas **RNN**, o *hidden state* é atualizado com o valor do novo input para gerar um *candidate state* [105]. As portas calculam um novo *hidden state* final, como uma combinação do *hidden state* anterior e o *candidate state*, com informação relevante para passar ao próximo *timestep* [106]. Na Figura 16 podemos observar a arquitetura de um bloco de **GRU**.

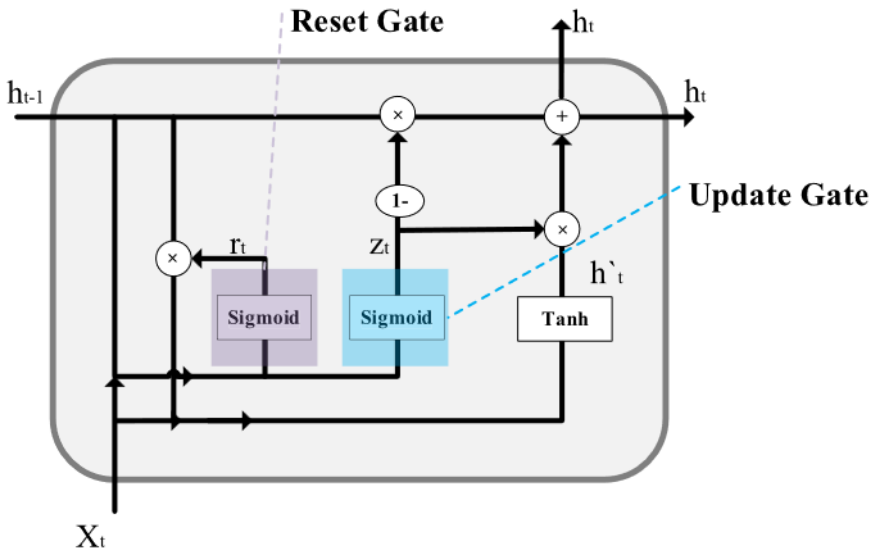


Figura 16: Arquitetura de um bloco de **GRU**. Fonte:[102]

O *update gate* e o *reset gate* estão representados pelas fórmulas 2.24 e 2.25, respetivamente. Como podemos ver na fórmula 2.26, o *input* x_t e a informação do *timestep* anterior h_{t-1} são transformados linearmente, e as diferentes matrizes W são multiplicadas à direita, respetivamente. Em seguida, o *reset gate* r_t e $W_{hh}h_{t-1}$ são multiplicados em conjunto e finalmente a informação do *timestep* atual é obtida através da função **tanh**. O *output* do bloco de **GRU** é então a soma da informação filtrada do *timestep* anterior com a nova informação obtida no *timestep* atual que se pode observar na fórmula 2.27.

$$Update\ Gate : Z_t = \sigma(W_{zh}h_{t-1} + W_{zx}x_t + b_z) \tag{2.24}$$

$$Reset\ Gate : r_t = \sigma(W_{rh}h_{t-1} + W_{rx}x_t + b_r) \tag{2.25}$$

$$Hidden\ State : h'_t = tanh(W_{xh}x_t + r_t W_{hh}h_{t-1}) \tag{2.26}$$

$$New\ Hidden\ State : h_t = Z_t h_{t-1} + (1 - z_{t)}h'_t \tag{2.27}$$

Apesar da melhoria na performance em relação as **RNN**, as **GRU** podem ainda ter dificuldade em captar padrões muito complexos e não lineares em dados de séries temporais.

2.3.3 LSTM

As **LSTM** são capazes de capturar dependências a longo prazo nos dados, utilizando uma célula de memória, que pode reter informação durante um período de tempo prolongado, resolvendo o problema dos *vanishing gradients* [103, 107]. A sua principal caraterística é a sua capacidade de reter ou esquecer informações de forma seletiva. A célula tem três portões que controlam o fluxo de informação: o *input gate*, o *forget gate*, e o *output gate*. O *input gate* controla o fluxo de nova informação para dentro da célula, o *forget gate* controla o fluxo de informação para fora da célula, e o *output gate* controla o fluxo de informação para fora da célula e para a célula seguinte da série [89, 108]. Podemos ver uma representação da arquitetura de uma **LSTM** na Figura 17.

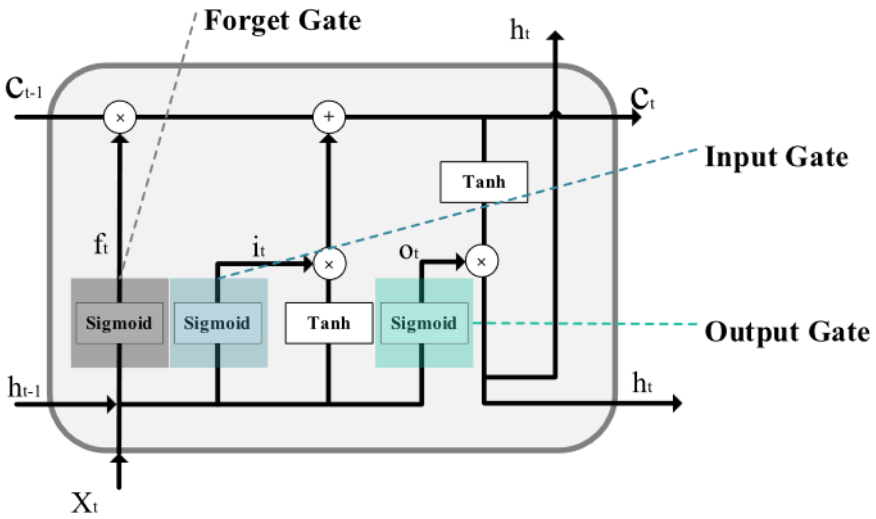


Figura 17: Arquitetura de um bloco de **LSTM**. Fonte:[102]

Para previsão de séries temporais, a rede é apresentada com um conjunto de dados históricos das séries temporais, e é treinada para prever o próximo valor na série. A rede é treinada utilizando uma variante de *backpropagation* chamada **Back-Propagation Through Time (BPTT)**, que permite que o erro se propague de volta através de toda a série de células **LSTM**. As **LSTM** são particularmente boas no processamento de dados de séries temporais com dependências a longo prazo, tais como dados financeiros, dados meteorológicos, etc [109]. Além disso, também são boas no processamento de dados com múltiplas escalas temporais e dados em falta ou ruidosos [89, 108].

$$Forget\ Gate: f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \tag{2.28}$$

$$Input\ Gate : i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \tag{2.29}$$

$$Output\ Gate : o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \tag{2.30}$$

$$Cell\ State : c_t = f_t \times c_{t-1} + i_t \times tanh(W_o \times [h_{[t-1]}, x_t] + b_o) \tag{2.31}$$

$$Hidden\ State : h_t = o_t \times tanh(c_t) \tag{2.32}$$

As fórmulas 2.28, 2.29 e 2.30, representam o *forget gate*, *input gate* e *output gate*, respetivamente. Na fórmula 2.31, a variável c_t representa a informação obtida desde o ínicio do treino até ao *timestep* atual. Essa informação é então utilizada na fórmula 2.32 que vai definir quanta da informação atual será passada para o próximo *timestep*.

2.4 Revisão de leitura

Nesta secção é feita uma revisão sobre alguns artigos que exploram o tema de explicabilidade aplicado a séries temporais, bem como uma análise crítica sobre estes.

Os autores em [110] desenvolveram um modelo explicável de **RNN** com alto desempenho diagnóstico para a previsão não invasiva de genótipos **Isocitrate Dehydrogenase (IDH)** em gliomas usando a ressonância magnética de perfusão. Estudos anteriores utilizaram o **relative Cerebral Blood Volume (rCBV)**, o que reflecte vascularização tumoral, para prever o genótipo **IDH** dos gliomas, porque se sabe que a mutação **IDH** está associada à angiogénese tumoral nos gliomas. O estudo demonstrou que um modelo de **RNN**, que aprende padrões sequenciais, pode distinguir estes grupos sobrepostos através da utilização de curvas de intensidade de sinal de susceptibilidade *T2** bruto, e multidimensionais. Estas curvas são obtidas a partir da ressonância magnética de perfusão, levando a um desempenho de diagnóstico melhorado e generalizado, utilizando um conjunto de testes invisíveis. O modelo também proporciona interpretabilidade ao demonstrar quais as características temporais cruciais para a previsão de genótipos IDH baseados em antecedentes molecular-biológicos obtidos usando mecanismos de atenção. O desempenho do modelo foi avaliado através do cálculo da precisão, sensibilidade, e especificidade do conjunto de teste. Além disso, **Receiver Operating Characteristic (ROC)** foi realizada utilizando probabilidades sigmóides para obter curvas **ROC** e calcular a **Area under de ROC curve (AUC)**. Para construir

um modelo **DL**, construíram também um modelo baseado em **LSTM** para a previsão do estado 1p/19q com modificação do hiperparâmetro. Foi utilizada uma camada de Atenção, que consistia de uma única **Feed-forward Neural Network (FNN)**, para visualizar e interpretar as características temporais. O mecanismo de atenção permitiu-lhes examinar que passos temporais das sequências de entrada eram críticos para o modelo alcançar uma classificação, visualizando os pesos da **FNN** como um heatmap. Este modelo de previsão alcançou um elevado desempenho de diagnóstico com interpretabilidade e reprodutibilidade para o conjunto de teste. O alto desempenho deve-se não só ao facto do modelo ser baseado em **LSTM**, que lhe permite exibir um elevado desempenho na aprendizagem sequencial, mas também porque optimizaram o modelo adicionando uma camada convolucional. Esta camada extra permitiu extrair características regionais semânticas e remover redundância de dados sequenciais de entrada altamente correlacionados, levando assim a uma melhor captura de correlações temporais, bem como a camada de atenção para a interpretabilidade do modelo.

Em [44] o objetivo principal da investigação era determinar se os métodos **XAI** existentes eram transferíveis para modelos de previsão de séries temporais financeiras. Começaram por contruir uma série temporal multivariada do S&P 500 de forma diária, de 2 de Dezembro de 1984 a 28 de Maio de 2021, excluindo fins de semana, abrangendo 9197 dias. Posteriormente, a tendência foi removida através da decomposição sazonal e da tendência, utilizando a **Regressão Local (LOESS)**. O teste Dickey-Fuller aumentado confirmou que a série temporal estava estacionária antes de ter sido submetida a uma normalização global. O estudo utilizou três arquiteturas de redes neuronais recorrentes diferentes como modelos para investigar métodos de explicabilidade: um **RNN** padrão, um **GRU**, e um **LSTM**. Os hipermarâmetros foram optimizados utilizando o *optimizer adam*, minimizando o erro quadrado médio no conjunto de validação. Além disso, os autores utilizaram depois 4 métodos para explicar os 3 resultados do modelo: Ablação, Gradientes Integrados, Ruído Adicionado e Permutação. A ablação pode ser considerada uma forma mais simples de gradientes integrados, pois é uma técnica que remove informação de uma característica, e a mudança na previsão é medida. Os gradientes integrados tornaram-se um método popular e intuitivo para medir as correlações entre as características de entrada e a previsão correta. Foi adicionado um nível definido de ruído (1%) a todas as entradas numa determinada célula, para determinar quais as células, e por extensão quais os dias, mais importantes para a previsão. O foco dos métodos de permutação foi determinar qual a característica que contribui mais para a previsão, para cada arquitectura **RNN**. Curiosamente, todos os três modelos atribuíram a prioridade mais baixa ao volume de stock negociado, enquanto que não houve uma única característica com a maior prioridade entre os três modelos. Todos os quatro métodos apresentados forneceram resultados complementares apesar de

testarem diferentes aspetos da rede, proporcionando um meio robusto para compreender os resultados de cada um dos três modelos. Coletivamente, estes resultados forneceram uma visão de onde os modelos dão importância às características de entrada e revelaram estratégias contrastantes de afinação da previsão entre arquiteturas. Os resultados sugerem que os princípios que formam a base dos métodos de explicabilidade são aplicáveis na previsão de séries temporais financeiras e podem proporcionar uma compreensão das suas arquiteturas complexas.

No trabalho desenvolvido em [111], os autores propuseram-se a comparar extensivamente o desempenho de vários métodos de interpretabilidade baseados na saliência entre diversas arquiteturas neuronais, incluindo **RNN**, **Temporal Convolutional Network (TCN)**, e *Transformers*, numa nova referência de dados de séries temporais sintéticas. Os autores propõem e relatam múltiplas métricas para avaliar empiricamente o desempenho dos métodos de saliências para detetar a importância das características. Os mapas produzidos pela **Convolutional Neural Network (CNN)** podem distinguir pixeis informativos correspondentes a características reveladoras em etapas informativas de tempo. No entanto, os mapas produzidos pela **TCN** ficam aquém de distinguir características importantes dentro de um dado passo de tempo. Os métodos de saliência, incluindo os métodos baseados no gradiente, e os métodos baseados na perturbação, não produzem interpretações de alta qualidade quando aplicados a dados de séries temporais multivariadas. No entanto, podem produzir mapas precisos quando as séries multivariadas são representadas como imagens ou como séries temporais univariadas. Ou seja, quando os domínios temporal e de características são combinados numa série temporal multivariada, os métodos de saliência decompõem-se em geral. Através de experiências, os autores observaram que os métodos geralmente identificam passos temporais salientes, mas não conseguem distinguir características importantes versus características não importantes, dentro de um dado passo temporal. Com base nesta observação, é proposta então uma abordagem de redimensionamento da saliência temporal em duas etapas, para adaptar os métodos de saliência existentes aos dados das séries temporais. Esta abordagem conduziu a melhorias substanciais na qualidade dos mapas de saliência produzidos por diferentes métodos. No primeiro passo, é calculada a pontuação de relevância temporal para cada *timestep* através do cálculo da mudança total dos valores de saliência, se essa etapa temporal for mascarada. No segundo passo, em cada *timestep*, cuja pontuação de relevância temporal esteja acima de um determinado limiar α, é calculada a pontuação de relevância temporal para cada característica calculando a mudança total nos valores de saliência, se essa característica for mascarada. Ao escolher um valor adequado para α, o segundo passo pode ser executado em alguns passos de tempo altamente relevantes, de forma a reduzir a complexidade computacional global do método. Depois, a pontuação de importância final (tempo, ca-

racterística) é o produto do tempo associado e da pontuação de relevância da característica. Desta forma, os mapas produzidos no final mostraram uma qualidade superior em termos de explicabilidade tanto a a nível temporal como entre *features*.

Em [94] as previsões de vendas de uma companhia feitas pelo modelo **Support Vector Regression (SVR)** foram interpretadas utilizando métodos de explicabilidade **LIME** e **SHAP** para compreender a influência das características nas previsões a nível local. O modelo de melhor desempenho baseado no menor erro percentual absoluto médio foi interpretado utilizando **SHAP** e **LIME**. Dez instâncias de previsão do conjunto de dados de teste do modelo com melhor desempenho foram escolhidas e a explicação local para cada uma delas foi obtida utilizando tanto o **SHAP**, como o **LIME**. A fim de avaliar a qualidade das explicações produzidas pelos dois métodos de explicabilidade, foi realizado um estudo de avaliação humana. O estudo consistiu em selecionar 3 grupos de pessoas com diversos graus universitários para responder a um questionário sobre as previsões feitas pelo modelo. Um grupo consistia num questionário com explicações de **LIME**, outro com explicações de **SHAP** e último sem explicações. No final do questionário todos os 3 grupos foram apresentados com uma questão para avaliar a sua experiêcia numa escala de Likert de 0-5 e sugerir possiveis melhorias. Nos grupos com explicações **LIME** e **SHAP** ainda havia questões para avaliar as explicações na mesma escala de Likert, ainda foram pedidos para sugerir possiveis melhorias às explicações e o nível de confiança das mesmas. No grupo sem explicações, foram pedidos que dissessem se confiavam nas previsões e sugestões de explicações que poderiam ser úteis. O estudo concluiu que **LIME** e **SHAP** ajudaram os humanos na compreensão do processo de tomada de decisão do modelo subjacente e melhoraram a confiança no modelo. A partir da comparação quantitativa e qualitativa das explicações **LIME** e **SHAP**, parece que os avaliadores humanos têm preferência pelas explicações **LIME** em detrimento de **SHAP**. A diferença foi, no entanto, insignificante, mas talvez um aumento do tamanho da amostra pudesse ajudar a solidificar ainda mais a comparação. Uma vez que as explicações foram avaliadas utilizando uma avaliação humana que envolveu utilizadores finais, as explicações poderiam no futuro ser adicionalmente avaliadas utilizando uma avaliação fundamentada na aplicação que envolve peritos do domínio, executando tarefas específicas para o caso de utilização das explicações.

Um sistema hibrido entre **ARIMA** e uma **ANN** é utilizado em [74]. A metodologia proposta para o sistema híbrido consiste em duas etapas. Na primeira etapa, um modelo **ARIMA** é utilizado para analisar a parte linear do problema. Na segunda etapa, é desenvolvido uma **ANN** para modelar os resíduos do modelo **ARIMA**. Uma vez que o modelo **ARIMA** não pode capturar a estrutura não linear dos dados, os resíduos do modelo linear conterão informações sobre a não linearidade. O modelo híbrido

explora a característica única e a força do modelo **ARIMA** e do modelo **ANN** na determinação de padrões diferentes. Assim, torna-se vantajoso modelar padrões lineares e não lineares separadamente utilizando modelos diferentes e depois combinar as previsões para melhorar o desempenho global da previsão. Para problemas complexos que têm estruturas de correlação linear e não linear, o método de combinação mostrou ser uma forma eficaz de melhorar o desempenho da previsão. Os resultados empíricos com três conjuntos de dados reais sugerem claramente que o modelo híbrido é capaz de superar o desempenho de cada modelo componente utilizado isoladamente.

2.4.1 Análise Crítica

Após leitura é possível notar a utilização de vários métodos para explicar as previsões dos modelos. Dos modelos de **DL** as **RNN** e **LSTM** sobressaem pela sua grande capacidade de detetar dependências temporais, mesmo em séries multivariadas, no entanto necessitam de estar acompanhadas de mecanismos de atenção ou outras ferramentas de explicabilidade como **LIME** e **SHAP**. Apesar de estas não serem ideais a captar as dependências temporais dos dados, ainda assim ajudam as pessoas na compreenssão da tomada de decisão dos modelos e dão alguma transparência à *blackbox*. Ao contrário do **LIME** e **SHAP**, os métodos de saliência têm grande poder de explicação visual, mas captam apenas relações temporais das *features*.

Em [110], a camada de atenção melhora a capacidade do modelo de se concentrar seletivamente em elementos específicos dentro da série temporal multivariada, permitindo uma extração de características mais eficaz e o reconhecimento de dependências temporais. Mesmo que desta forma já seja possivel extrair os valores dos pesos de atenção de cada *feature*, em cada *timestep*, este é o único método utilizado. Este tipo de explicabilidade é global a todos os dados de treino e de caráter ante-hoc por ser adquirida durante as fase de treino.

No estudo em [44], a série multivariada é injetada em 3 modelos distintos e posteriormente são aplicados 4 métodos diferentes em cada um dos modelos. A exploração de vários modelos e técnicas diferentes busca, não só aumentar a precisão da previsão, mas aumentar a confiança das previsões e tornar os modelos mais transparentes uma vez que os vários métodos convergiram para os mesmos resultados.

Em [111], os 3 modelos treinados são Transformer, **RNN** e **TCN** e são utilizados permutação e gradientes como técnicas de explicabilidade. Assim como em [44], são usados vários modelos e técnicas para comparação de resultados, no entanto este difere-se por aprofundar nas explicações. A aplicação de um método de duas fases que primeiro calcula a importância temporal do *timestep* e depois a relevência

da *feature* no mesmo *timestep*, permite-nos ter um conhecimento maior sobre as possíveis interações não lineares, captadas apenas por modelos complexos, entre as várias *features* de uma séria multivariada.

[94] é o único dos artigos que explora a explicabilidade post-hoc e local, com utilização de **LIME** e **SHAP**. No entanto, o modelo utilizado foi o **SVR** devido ao reduzido número de dados. Este modelo não é o mais adequado, uma vez que modelos clássicos de **ML** não são capazes de capturar dependências temporais e comportamentos não lineares entre os dados. Ainda assim, o estudo comprovou que quando aplicados e apresentados de forma correta, os resultados dos modelos transmitem mais confiança mesmo para pessoas que não têm conhecimento do caso de estudo.

Por fim, a abordagem híbrida apresentado em [74], mostrou-se poderoso no desempenho das previsões de séries temporais univariadas, no entento, não é aplicado nenhuma técnica de explicabilidade à **ANN**, correspondente à parte não linear do treino. A única explicabilidade que é possivel extrair no problema é das variáveis do modelo ARIMA utilizado para modelar a parte linear dos dados.

Artigo	Ante-hoc	Post-hoc	Global	Local
[110]	✓	X	✓	X
[44]	X	✓	✓	✓
[111]	X	✓	✓	✓
[94]	X	✓	✓	✓
[74]	✓	X	✓	X

Tabela 4: Tipos de explicabilidade usada nos artigos

A Tabela 4 foi contruída a partir dos tipos de explicabilidade utilizada nos artigos. Como se pode observar, nenhum dos artigos analisados nesta secção implementou métodos de explicabilidade que conseguissem cobrir todos os campos. Desta forma, para esta dissertação todos os métodos implentados respondem em conjunto às necessidades de conseguir interpretar um modelo tanto a nível global ou local às pevisões, e durante (*ante-hoc*) ou após (*post-hoc*) o treino.

Capítulo 3

Materiais e Métodos

Este capítulo foca-se na exploração, análise e transformação dos dados de forma a serem corretamente aplicados aos modelos candidatos concebidos. O capítulo começa com uma análise estatística dos dados, observação e distribuição dos dados na secção 3.1. Após esta análise, na secção 3.2, os dados são tratados tendo em conta valores e *timesteps* em falta, *feature engineering* e seleção de varáveis através da análise de correlação. Por fim, na secção 3.3, como etapa importante no estudo de séries temporais, é feita uma análise temporal dos dados onde é feita uma procura de possíveis anomalias ao longo da série, estudo da tendência, sazonalidade e *white noise* através de decomposição e gráficos de autocorrelação.

3.1 Exploração dos Dados

Como processo essencial no *Data Mining*, a exploração dos dados permite analisar, visualizar e avaliar os dados levando a uma melhor compreensão dos mesmos para realizar o processamento mais adequado e adquirir melhores resultados. Os vários parâmetros, que compoêm a série temporal multivariada em estudo, dizem respeito à qualidade do ar na cidade do Porto e foram adquiridos no *Open Weather Map*. A inclusão de várias variáveis de concentrações de poluentes, relacionados à qualidade do ar, contribuem para uma análise abrangente e detalhada do ambiente atmosférico da região. Sendo uma série multivariada, esta oferece uma perspetiva mais completa, permitindo uma compreensão mais profunda dos padrões e tendências associados à qualidade do ar na cidade do Porto.

O *dataset* é composto por vários parâmetros, incluindo data e hora da observação, oito colunas de poluentes e um grande número de observações, que totalizam 18.215 registros. Cada um dos poluentes representa a sua própria série temporal tornando assim o dataset numa série temporal multivariada. Estes dados abrangem um período considerável, com o primeiro registro datado de 25/11/2020 (01:00h) e o último de 31/12/2022 (23:00h). É importante observar que, em séries temporais, a ordem e a temporalidade das observações desempenham um papel fundamental na análise. O fato de as observações

Atributo	Descrição	Tipo de Dados	Unidade
Data	Registo de data e hora	Timestamp	Data e Hora
CO	Valor da medição do CO	Double	$\mu g/mg^3$
Monóxido de Nitrogénio (NO)	Valor da medição do NO	Double	$\mu g/mg^3$
NO_2	Valor da medição do NO_2	Double	$\mu g/mg^3$
O_3	Valor da medição do O_3	Double	$\mu g/mg^3$
Dióxido de Enxofre (SO_2)	Valor da medição do SO_2	Double	$\mu g/mg^3$
Matéria Particular 2.5 ($PM_{2.5}$)	Valor da medição do $PM_{2.5}$	Double	$\mu g/mg^3$
Matéria Particular 10 (PM_{10})	Valor da medição do PM_{10}	Double	$\mu g/mg^3$
Amoníaco (NH_3)	Valor da medição do NH_3	Double	$\mu g/mg^3$

Tabela 5: Constituição do Dataset

ocorrerem a cada hora permite a análise de tendências, sazonalidades e variações ao longo do tempo, tornando este dataset muito adequado para o problema em questão. É possível observar uma descrição dos dados na Tabela 5

Além disso, foi realizada uma análise estatística aos vários poluentes, onde foram avaliados os valores mínimo, máximo, mediana, média, desvio padrão, *skewness* (mede a assimetria da curva da função de distribuição de probabilidade) e *kurtosis* (carateriza o achatamento da curva da função de distribuição de probabilidade) [112]. É possível observar esta análise na Tabela 6.

	CO	NO	NO_2	O_3	SO_2	$PM_{2.5}$	PM_{10}	NH_3
Min	133.51	0	0.89	0	0.37	0.5	0.59	0
Max	1388.55	84.94	97.33	246.05	20.98	125.18	154.17	69.92
Mediana	240.33	0.14	6.60	60.08	1.97	5.31	8.39	1.65
Média	275.86	1.88	10.49	57.89	2.45	8.22	12.25	2.83
Desvio Padrão	129.53	6.43	10.16	31.35	1.75	8.35	12.88	4.20
Skewness	2.96	5.93	2.21	0.088	2.89	2.83	4.08	4.20
Kurtosis	11.79	42.12	6.35	0.20	13.12	13.30	27.29	37.71

Tabela 6: Análise estatística

De acordo com a escala de nível de index da qualidade do ar do Open Weather, escala representada na Tabela 7, na Tabela 6 podemos observar que todos os poluentes com exceção do CO, NO e NH_3

ultrapassam, em algum momento, pelo menos o primeiro nível de qualidade, com o SO_2 chegando ao index 2, o NO_2 ao index 3 e o $PM_{2.5}$, PM_{10} e O_3 chegando ao index 5. No entanto, de acordo com as **Air Quality Guidelines (AQGs)** da **World Health Organization (WHO)**, apenas o $PM_{2.5}$ ultrapassa a média anual recomendada [113]. Analisando os valores mínimos, máximos, médios e medianas das variáveis, quase todas apresentam valores de média e mediana muito mais próximos do valor mínimo do que do máximo, podendo ser um indicativo de que o *dataset* apresenta *outliers* ou que a distribuição dos dados é mais concentrada nos valores baixos. Os *outliers* e distribuição são avaliados posteriormente ainda neste capitulo.

Nome Qualitativo	**Index**	**Concentração em** $\mu g/mg^3$					
		SO_2	NO_2	PM_{10}	$PM_{2.5}$	O_3	CO
Bom	1	[0; 20]	[0; 40]	[0; 20]	[0; 10]	[0; 60]	[0; 4400]
Razoável	2	[20; 80]	[40; 70]	[20; 50]	[10; 25]	[60; 100]	[4400; 9400]
Moderado	3	[80; 250]	[70; 150]	[50; 100]	[25; 50]	[100; 140]	[9400; 12400]
Pobre	4	[250; 350]	[150; 200]	[100; 200]	[50; 75]	[140; 180]	[12400; 15400]
Muito Pobre	5	$\geq$350	$\geq$200	$\geq$200	$\geq$75	$\geq$ 180	$\geq$15400

Tabela 7: Escala de nível de index da qualidade do ar. Fonte: [114]

Nas seguintes secções será feita uma análise mais detalhada dos valores apresentados na Tabela 6 do CO, NO_2 e O_3. Estas 3 variáveis foram selecionadas pelo seu elevado número de emissões e impacto na saúde e no ambiente [115]. Com o objetivo de verificar a conformidade dos dados com uma distribuição normal (ou Gaussiana), recorreu-se ao teste de *Kolmogorov-Smirnov*. Este teste envolve a formulação de duas hipóteses: a primeira, chamada hipótese nula, postula que a amostra segue uma distribuição Gaussiana, enquanto a segunda, denominada hipótese alternativa, que sugere que a amostra não adere a uma distribuição Gaussiana. A avaliação entre essas hipóteses baseia-se no valor *p*. Se este valor for superior a 0.05, a hipótese nula é aceite, indicando que a distribuição dos dados segue uma distribuição Gaussiana. No entanto, se o valor p for menor ou igual a 0.05, a hipótese nula é rejeitada, concluindo que a distribuição dos dados não é normal.

3.1.1 Análise CO

A escolha para a seleção desta variável, deu-se não só pelo facto de ser a que apresenta o valor mais elevado na Tabela 6, mas também pelo facto de ser um gás extremamente perigoso para a saúde humana,

uma vez que é tóxico, inodor, incolor, e insipido. Além disso, a sua emissão é completamente derivada da atividade humana, ou seja, não ocorre naturalmente na natureza [116].

As medições de CO na atmosfera apresentam uma notável variabilidade, com valores que se estendem desde um mínimo de 133 $\mu g/mg^3$ até um máximo de 1388 $\mu g/mg^3$. Esta ampla gama de valores sugere que ocorrem variações substanciais nos níveis de CO ao longo do tempo. A Figura 18, que representa a variação de CO ao longo do tempo, mostra estas grandes variações nas medições de CO, havendo picos nas concentrações desta variável quase diariamente.

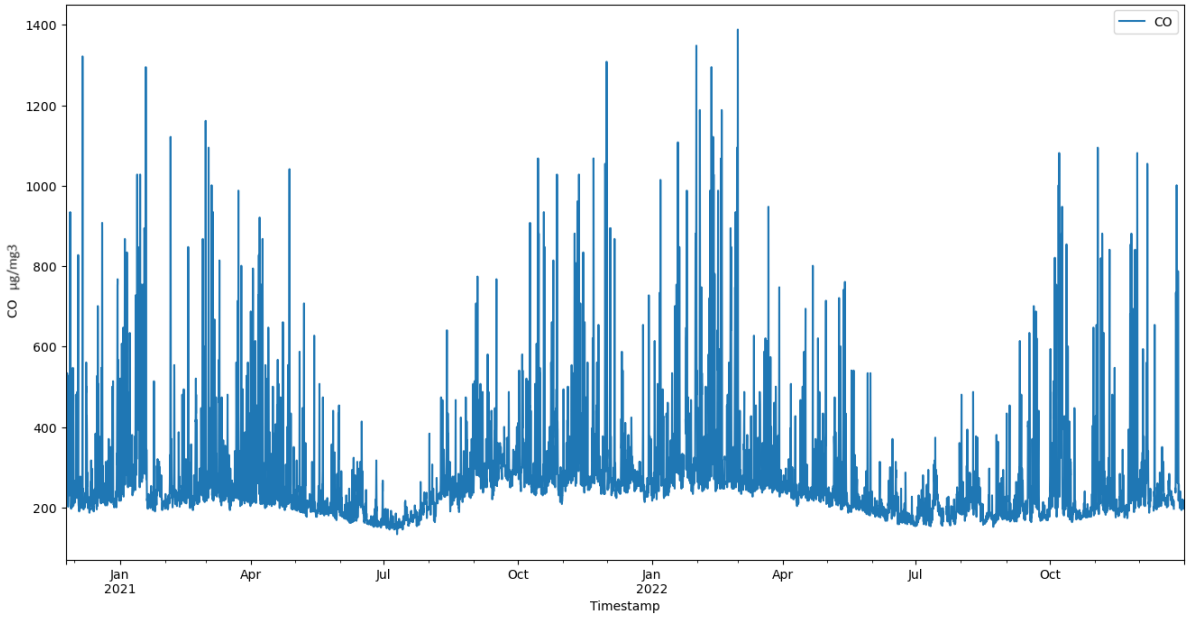


Figura 18: Série Temporal do CO

A média e mediana têm respetivamente os valores 275.86 e 240.33, 6, logo podemos concluir que metade das medições está abaixo da média uma vez que mediana < média. Esta diferença sugere que a distribuição das medições pode apresentar uma possível assimetria à direita. Em outras palavras, existem valores mais elevados que puxam a média para cima, enquanto a maioria das medições se concentra em valores mais baixos. O desvio padrão de 129,53 $\mu g/mg^3$, 6, é relativamente alto em relação à média, o que sugere uma dispersão considerável dos dados em torno da média. Esse desvio padrão mais elevado indica que as medições estão amplamente espalhadas, o que pode ser resultado de flutuações significativas nos níveis de CO ao longo do tempo, como se pode confirmar na Figura 18. A alta dispersão sugere que, apesar da média representar um valor central, os valores individuais podem variar consideravelmente, refletindo a complexidade das condições ambientais e as influências sazonais ou episódicas que afetam os níveis de CO na atmosfera.

Os valores da *skewness* e da *kurtosis* também fornecem informações valiosas sobre a distribuição

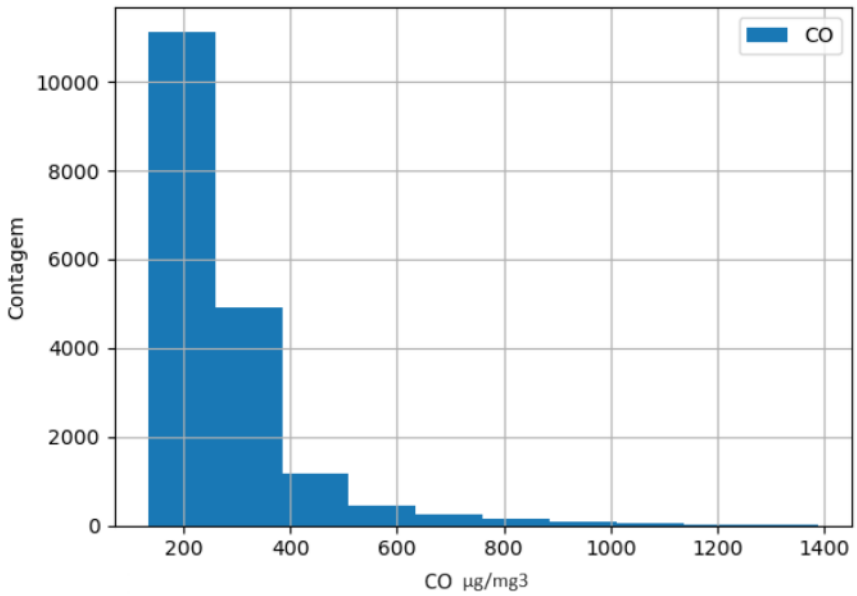


Figura 19: Distribuição CO

das medições. A *skewness* de 2,96 $\mu g/mg^3$, 6, que é positiva, indica que a cauda da distribuição está orientada para a direita, confirmando a suspeita de assimetria nesta direção. Esta assimetria significa que há uma tendência de valores mais elevados a afastar-se da média. A *kurtosis* de 11,79 $\mu g/mg^3$, 6, que é maior do que a *kurtosis* de uma distribuição normal (que tem uma *kurtosis* de 3), indica que a distribuição é mais leptocúrtica, ou seja, tem caudas mais alongadas. Desta forma, a distribuição das medições de CO apresenta valores mais extremos em comparação com uma distribuição normal. Estes valores extremos são consistentes com a vasta gama de valores observados, incluindo o valor máximo de 1388, como se pode observar no histograma da ditribuição das medições do CO na Figura 19. Além disso, relativamente ao tipo de distribuição, o CO não apresenta uma distribuição normal uma vez que o valor *p* dado pelo teste *Kolmogorov-Smirnov* é menor que 0.05.

Além das análises efetuadas, na Figura 20, podemos ver a série temporal do CO pela média mensal. A partir deste gráfico é possível retirar que as medições são mais altas nos meses de Inverno e tendem a diminuir no Verão. O mês onde se detetou mais emissões de CO foi em feveriro de 2022 e o mês com menos emissões foi julho de 2021. Os valores mais elevados no Inverno podem estar relacionados com o facto de que com o frio e a chuva as pessoas utilizem mais meios de transporte pessoais ou recorram a formas de aquecimento que emitem maiores quantidades de CO. Apesar do aumento que se faz notar nos meses frios, o mês de dezembro parece não seguir esta tendência, tendo sempre valores mais baixos em relação à sua vizinhança.

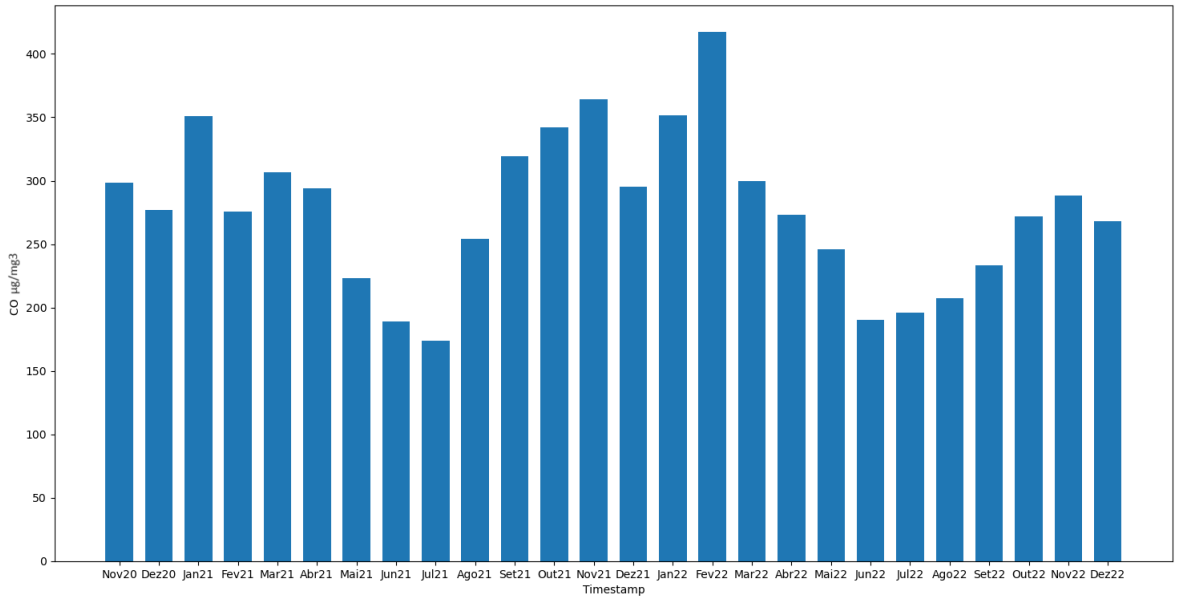


Figura 20: CO médio mensal

3.1.2 Análise NO2

Os dois principais poluentes precursores da formação de O_3 são **Óxidos de Azoto (NOx)** (emitidos principalmente pela queima de fósseis) e os **Compostos Orgânicos Voláteis (VOCs)** (emitidos tanto pela queima de combustíveis como pela evaporação da vegetação e combustíveis armazenados). A escolha do NO_2 sobre o NO, deveu-se principalmente ao pela breve análise efetuada aos valores de skewness e kurtosis da Tabela 6 [117].

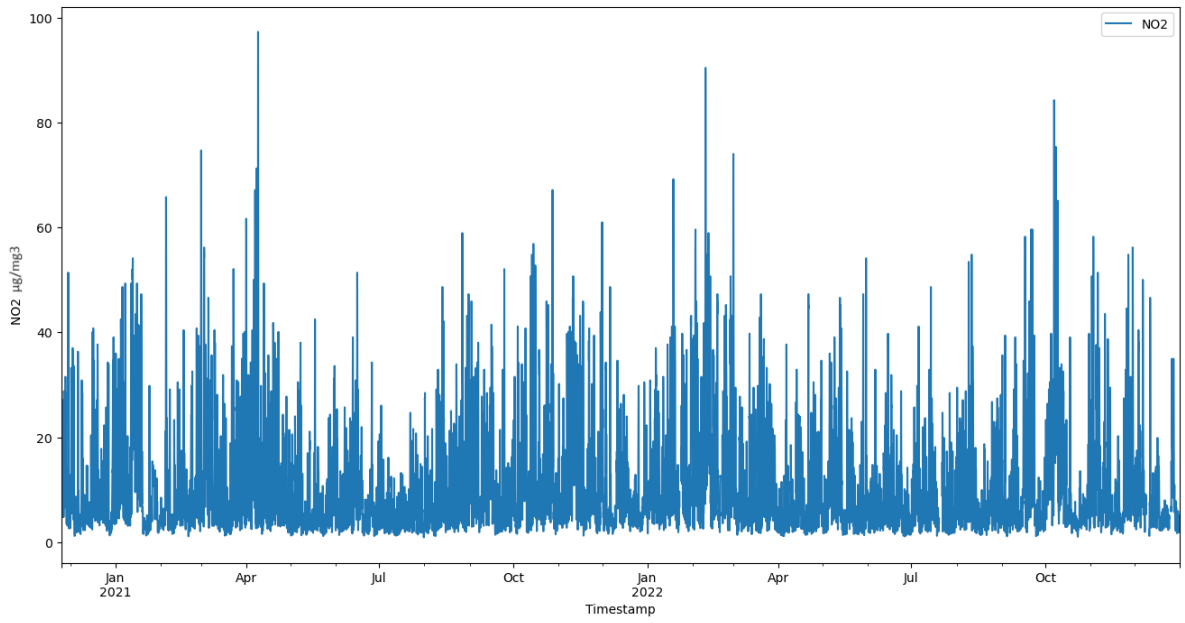


Figura 21: Série Temporal do NO_2

Analisando com mais detalhe o componente NO_2, na Figura 21 podemos ver a série temporal das suas medições. É possível observar 3 picos nas medições em abril de 2021 e fevereiro e outubro de 2022. O valor mínimo registado de 0.89 $\mu g/mg^3$, 6, indica a presença mínima de NO_2 e, no extremo oposto, o valor máximo de 97.33 $\mu g/mg^3$, 6, sugere a ocorrência de eventos ou condições excecionais, resultando em níveis de NO_2 muito elevados, possivelmente associados a fontes específicas de emissões ou condições atmosféricas particulares.

A mediana, situada em 6.60 $\mu g/mg^3$, 6, indica, assim como no CO, que a maioria dos valores está agrupada em níveis mais baixos. No entanto, a média de 10.49 $\mu g/mg^3$, 6, é influenciada pela assimetria positiva (*skewness*) de 2.21 $\mu g/mg^3$, 6, indicando que existem valores mais elevados que estão a puxar a média para cima. Todos estes valores sugerem uma distribuição assimétrica, com alguns eventos ou condições elevando ocasionalmente os níveis de NO_2. O desvio padrão de 10.16 reflete uma considerável dispersão dos dados em relação à média, indicando variabilidade significativa nas medições.

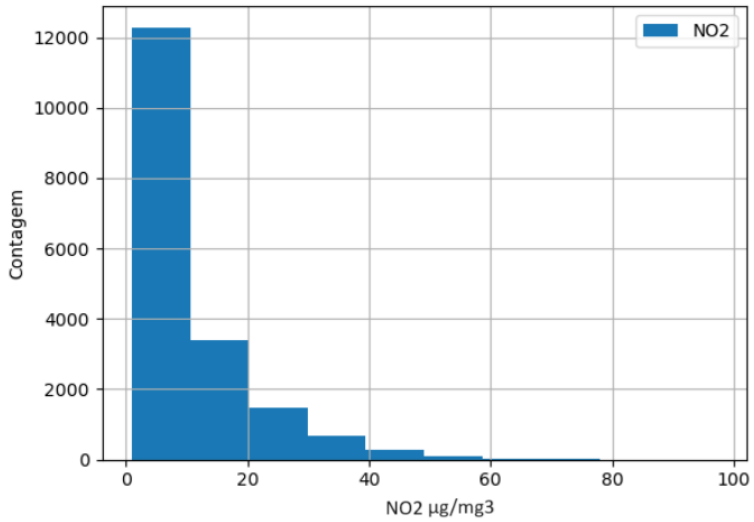


Figura 22: Distribuição NO_2

Através da Figura 22, podemos ver que a distribuição não é balanceada, com a maioria das medições localizadas à esquerda do histograma e com uma longa cauda à direita. Este gráfico vai de encontro à *skewness* de 2.21 $\mu g/mg^3$, 6, e à *kurtosis* de 6.35 $\mu g/mg^3$, 6, superior ao valor padrão de 3, que indica uma distribuição mais leptocúrtica, com cauda mais pesada. Uma vez que parte das emissões de NO_2 advém das mesmas fontes das emissões de CO, observamos o mesmo comportamento de aumento e redução de emissões do NO_2 em relação ao CO ao longo do ano. Relativamente ao tipo de distribuição, o teste *Kolmogorov-Smirnov* indicou um valor de *p* menor que 0.05, provando assim que a distribuição não é Gaussiana.

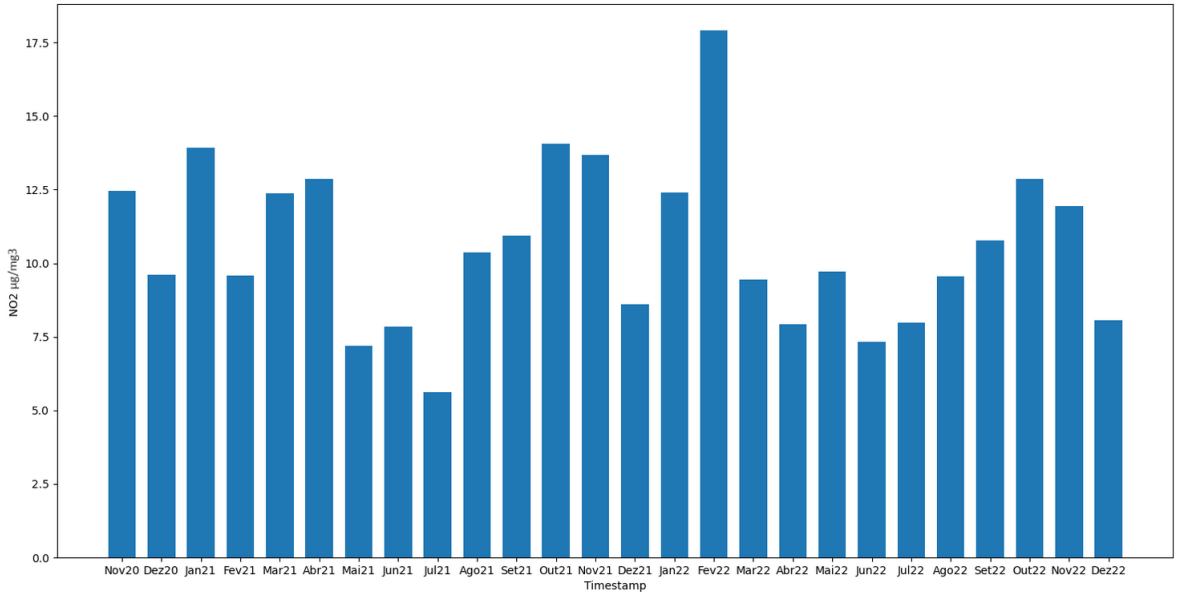


Figura 23: NO_2 médio mensal

Mais uma vez os meses frios detêm as medições mais altas e os meses mais quentes registam medições mais reduzidas, porém esta diferença não é tão notável como no CO. Na Figura 23 retiramos que o mês com a maior quantidade de emissões foi fevereiro de 2022 e o mês com menos emissões foi julho de 2021. Além disso, os meses de dezembro voltam a registar valores mais baixos em relação à sua vizinhança.

3.1.3 Análise O3

Como mencionado na secção anterior, o O_3 é formado nas camadas mais baixas da atmosfera através da interação do NO_2 com **VOCs** [117]. A interação entre estes compostos ajudou na decisão das variáveis selecionadas para uma análise mais profunda, na espetativa de extrair conhecimento dos dados sobre as suas relações.

Para analisar o O_3 começamos por observar a série temporal das suas medições na Figura 24. Este componente é o único que apresenta valores mínimos de 0, ocorrendo várias vezes ao longo do tempo. Os máximos foram registados nos meses de julho sendo a medição de 246.05 $\mu g/mg^3$, 6, a mais elevada registada em julho de 2022. À semelhança das *features* analisadas anteriormente, a mediana bastante mais baixa que o máximo, sugere que a maioria dos dados estão agrupados em níveis mais baixos do O_3. Por outro lado, neste caso a mediana de 60.08 $\mu g/mg^3$, 6, é superior à média de 57.89 $\mu g/mg^3$, 6, sugerindo uma distribuição relativamente simétrica e equilibrada, sem influências significativas de valores extremos. O desvio padrão de 31.35 $\mu g/mg^3$, 6, revela uma variabilidade significativa nas medições.

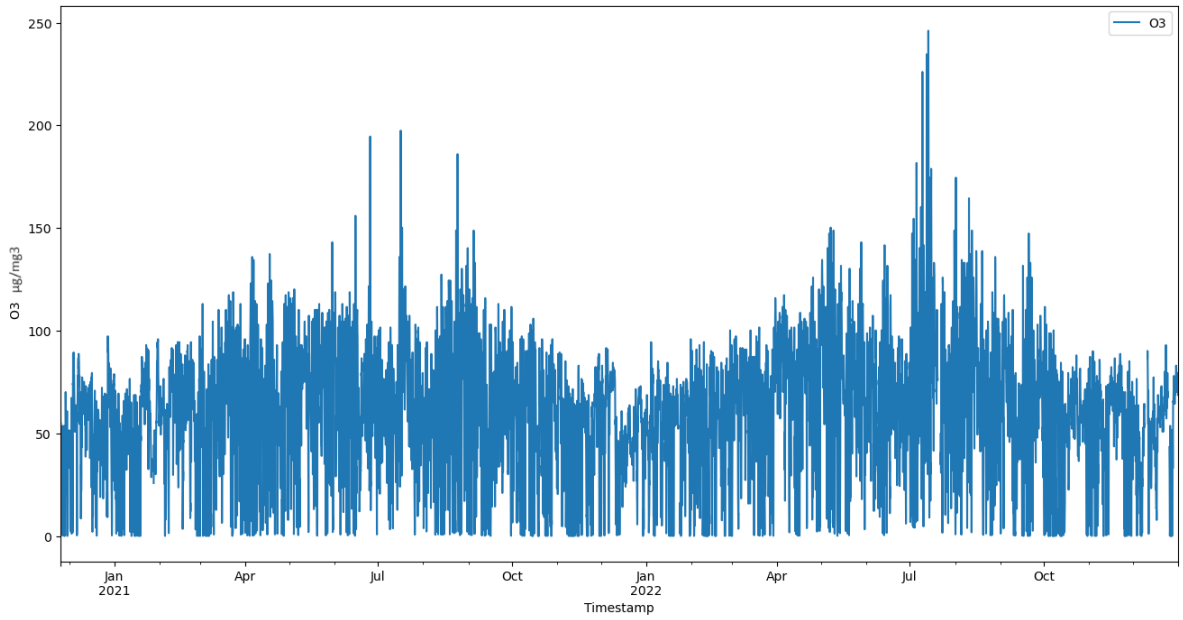


Figura 24: Série Temporal do O_3

A *skewness* próxima de 0 $\mu g/mg^3$, 6, demonstra não haver uma tendência para valores mais baixos ou mais altos, sustentando a hipótese de que a distribuição é mais simétrica em relação às anteriores. A *kurtosis* com valor 0.20 $\mu g/mg^3$, 6, é menor do que 3, o que indica que a distribuição se assemelha à forma de uma distribuição normal, sem caudas pesadas ou presença significativa de valores extremos, como podemos confirmar na Figura 25. No entanto, o teste *Kolmogorov-Smirnov* confirma que de facto esta *feature* também não segue uma distribuição Gaussiana.

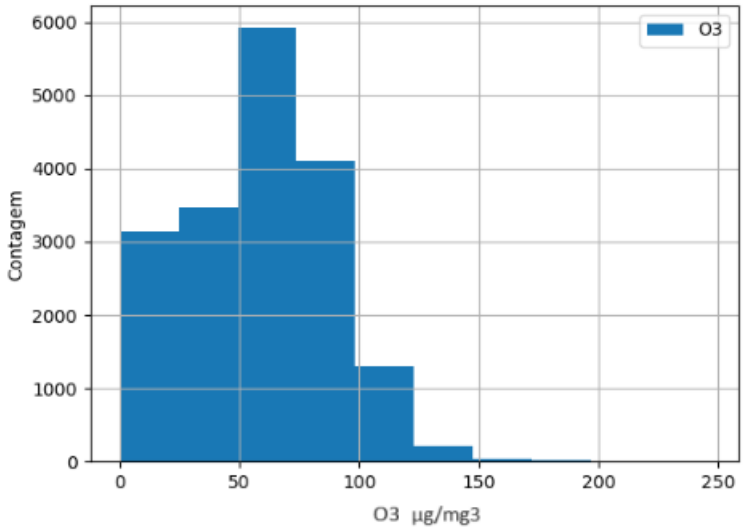


Figura 25: Distribuição O_3

Em relação aos valores médios mensais, Figura 26, estes mostram que o mês com medições mais

elevadas foi julho de 2022, que também contém o valor máximo de toda a série. O mês de novembro de 2021 é o que apresenta a menor média no entanto este mês apenas contém dados relativos aos últimos 5 dias. Por esse motivo, foi considerado como mínimo o mês de novembro de 2022. Podemos ainda ver um comportamento sazonal, com os meses mais quentes contendo medições mais elevadas e os meses frios contendo medições mais reduzidas. Este comportamento é contrário ao do CO e NO_2 sendo um indicativo de que estas *features* têm uma correlação negativa.

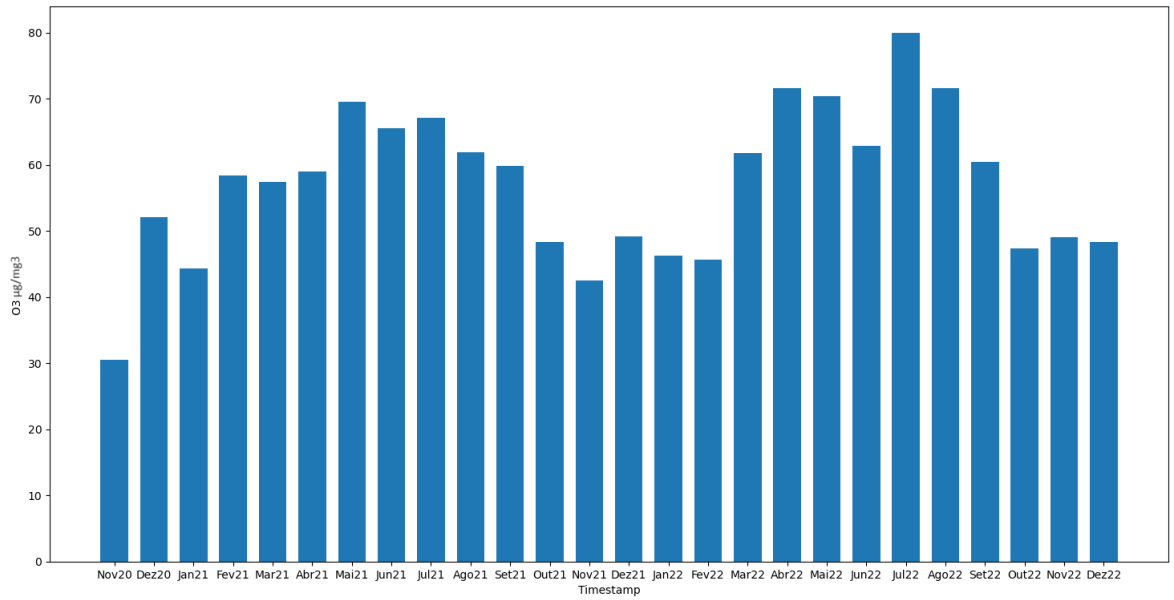


Figura 26: O_3 médio mensal

3.2 Preparação dos Dados

Nesta secção, é efetuado o tratamento de dados adequado para séries temporais e para posteriormente serem consumidos pelo modelo de previsão. Primeiramente é feito *feature engineering* do *datetime*. De seguida serão tratados os *timesteps* e valores em falta. Por fim, é feita uma análise da correlação das variáveis e a seleção das variáveis mais importantes para o caso de estudo.

3.2.1 Feature Engineering

Como mencionado anteriormente, cada medição está associada a um *timestamp* com um intervalo de 1 hora entre cada medição. No entanto a API disponibiliza o *timestamp* em Unix. A representação Unix de timestamps é uma contagem de segundos desde uma data de referência, geralmente 1º de janeiro de 1970, e é amplamente utilizada em sistemas de computador para registrar datas e horas. Converter

esses valores em um formato de data e hora comum é essencial para que os dados sejam interpretáveis e para permitir análises mais precisas. Apesar de haver *timesteps* em falta, a maior sequência em falta não supera as 24h, pelo que um *timestamp* a cada hora seria o mais adequado.

Após essa transformação foi ainda feito reindexação pelo *timestamp*. A reindexação dos dados pelo *timestamp* garante que os dados estejam organizados de forma uniforme, com uma entrada correspondente a cada hora no período de 25/11/2020 01:00h até 31/12/2022 23:00h. Desta forma, foi criada uma estrutura de dados regular que é fundamental para análises de séries temporais, uma vez que as análises de tendências, sazonalidades e padrões temporais dependem de uma sequência temporal consistente.

3.2.2 Missing Timestamps e Missing Values

Ao observar o dataset não foi identificada nenhuma entrada que contesse valores em falta, no entanto foram identificados saltos temporais, isto é, ausência de timestamps. Isto foi resultado da reindexação, pois ao criar uma entrada para cada hora, foram criadas novas linhas para horas em que não havia recolha de dados, ficando agora com valores em falta nas várias séries temporais. Esta é uma ocorrência comum quando se lida com séries temporais, pois os dados nem sempre são registrados de forma contínua, resultanto em valores ausentes nas várias séries. Por forma a resolver este problema, esses valores em falta foram preenchidos com recurso à biblioteca *Matrix Profile*. A *Matrix Profile* é uma abordagem importante e útil na análise de séries temporais, pois permite identificar padrões e anomalias de forma eficaz. Para compreendermos melhor o seu funcionamento, é importante destacar que a *Matrix Profile* é, essencialmente, um vetor que armazena informações sobre a distância euclidiana normalizada em Z, entre uma subsequência e a subsequência mais próxima na série temporal em análise. Esta métrica de distância é fundamental para a identificação de semelhanças e divergências nos dados, e a normalização em Z ajuda a garantir que as comparações sejam justas, independentemente da escala dos valores. Em 3.1 está um excerto de código com os passos para calcular a *Matrix Profile*.

Listing 3.1: Cálculo da matrix profile

```
1 # Step 1: Calculate the matrix profile
2 def calculate_matrix_profile(data, window_size):
3     n = len(data)
4     matrix_profile = []
5
6     for i in range(n - window_size + 1):
```

```
7           subsequence = data[i:i + window_size]
8           distance_profile = calculate_distance_profile(subsequence,
                data)
9           matrix_profile.append(min(distance_profile))
10
11      return matrix_profile
12
13  # Step 2: Calculate the distance profile for a subsequence
14  def calculate_distance_profile(subsequence, data):
15      n = len(data)
16      distance_profile = []
17
18      for i in range(n - len(subsequence) + 1):
19          subsequence_candidate = data[i:i + len(subsequence)]
20          distance = calculate_distance(subsequence,
                subsequence_candidate)
21          distance_profile.append(distance)
22
23      return distance_profile
24
25  # Step 3: Calculate the distance between two subsequences
26  def calculate_distance(subsequence1, subsequence2):
27      # Implementar uma métrica de distância como por exemplo a
            distância Euclidiana
```

Desta forma optou-se por inserir os dados em falta tendo em conta a média dos 24 *timestamps* anteriores ao valor em falta. No final o dataset passou de 18.215 para 18.407 *timestamps*, totalizando 192 *timesteps* inseridos, cerca de 8 dias. É importante referir que estes *timestamps* não se encontravam seguidos no tempo e que a maior sequência de *timestamps* em falta era cerca de 24.

3.2.3 Correlação e seleção de variáveis

De forma a selecionar a nossa *target feature* e filtrar os atributos menos relevantes, foi realizado um estudo de correlação. Como a distribuição dos dados não segue uma distribuição normal/gaussiana, tal como referido ao longo da secção 3.1, foi aplicado o método da correlação de *Spearman*. [118]. O método de

correlação de *Spearman* é uma técnica estatística que avalia a relação monotônica entre duas variáveis. Em vez de se basear na linearidade, como no caso da correlação de *Pearson*, o método de *Spearman* avalia a monotonicidade, ou seja, se, em geral, quando uma variável aumenta, a outra também aumenta, ou se, quando uma variável aumenta, a outra tende a diminuir. A correlação de *Spearman* é frequentemente usada quando as variáveis não seguem uma distribuição normal ou quando há a presença de outliers nos dados. Além disso, é menos sensível a valores extremos em comparação com a correlação de *Pearson*, tornando-a uma escolha robusta em várias situações. Assumindo que qualquer *feature* podia ser o *target*, percorreu-se uma a uma e registou-se a sua relação com todos os outros atributos, contabilizando como correlação forte ou moderada apenas o par que tinha um valor superior a 0.45 em módulo.

A Tabela 8 foi construída através da matriz na Figura 27 e representa o número de correlações fortes ou moderadas que cada *feature* tem com as restantes. É possível concluir que o NO_2 é a *feature* que partilha mais correlções fortes com outras variáveis e portanto foi escolhida para ser a variável a prever. Como apenas 5 das restantes variáveis têm correlação significante, as variáveis que não atingiram o valor mínimo de 0.45 em módulo, NO e SO_2 nomeadamente, foram ignoradas e não serão relevantes na continuidade do estudo. Este processo de seleção da variável alvo, com base na correlação ajuda a concentrar o estudo em *features* que têm uma relação mais substancial com o resultado desejado, permitindo uma análise mais precisa e focalizada.

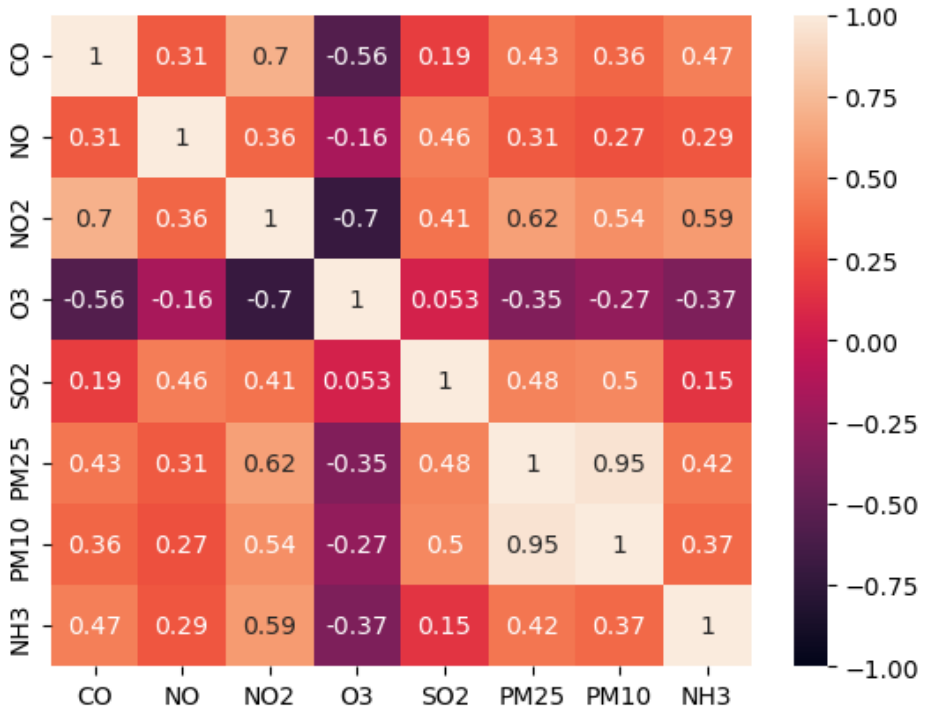


Figura 27: Matriz de Correlação

CO	NO	NO_2	O_3	SO_2	$PM_{2.5}$	PM_{10}	NH_3
3	1	5	2	3	3	3	2

Tabela 8: Número de correlações superiores a 0.45

3.3 Análise temporal dos dados

Para além da exploração comum a todos os tipos de dados, as séries temporais, pela sua natureza, requerem uma análise extra à sua vertente temporal. Deste modo, nesta secção são identificados as anomalias, isto é, as subséries que mais divergem em relação a toda toda a série. De seguida é feito um estudo quanto à tendência e sazonalidade dos dados de forma a detetar padrões no seu comportamento. Por fim, é feito um estudo da auto correlação para ver a influência que os dados do passado têm nos dados do presente e futuro.

3.3.1 Anomalias

Para analisar as anomalias foi utilizada a *Matrix Profile* mencionada em 3.2.2 Ao procurar os valores mais elevados no vetor da *Matrix Profile*, é possível realçar as subsequências que se destacam na série temporal [119]. Nas Figuras 28 e 29 estão representados 3 gráficos. O primeiro gráfico representa a *Matrix Profile* da série temporal do terceiro gráfico, em que as linhas vermelhas representam as subsequências com a distância euclidiana mais elevada de toda a série, e a azul as subsequências com a distância euclidiana mais baixa, ou seja, a vermelho temos as anomalias. A *Matrix Profile*, também pode ser visualizada no segundo gráfico onde temos no eixo vertical o valor calculado da distância euclidiana e no eixo horizontal a subsequência em questão. Neste gráfico as anomalias estão assinaladas com uma estrela, indicando os picos mais elevados da sequência. Estes picos estão representados no terceiro gráfico, na série temporal, com a subsequência assinalada como anomalia destacada com uma coloração diferente.

Nas Figuras 28 e 29, são apresentadas as três subsequências mais distintas nas séries temporais analisadas. É notável que, em ambos os casos, as duas maiores anomalias, destacadas a vermelho e roxo, correspondem aos meses de dezembro de 2020 e dezembro de 2021. Este comportamento anómalo desses meses, em relação aos demais meses de inverno, já havia sido previamente reportado. Especificamente, o mês de dezembro destaca-se como único, apresentando valores médios mais baixos. Estes valores médios inferiores podem estar relacionados a condições climáticas específicas ou a padrões sazonais. Além disso, a terceira anomalia identificada em ambos os casos parece estar relacionada com

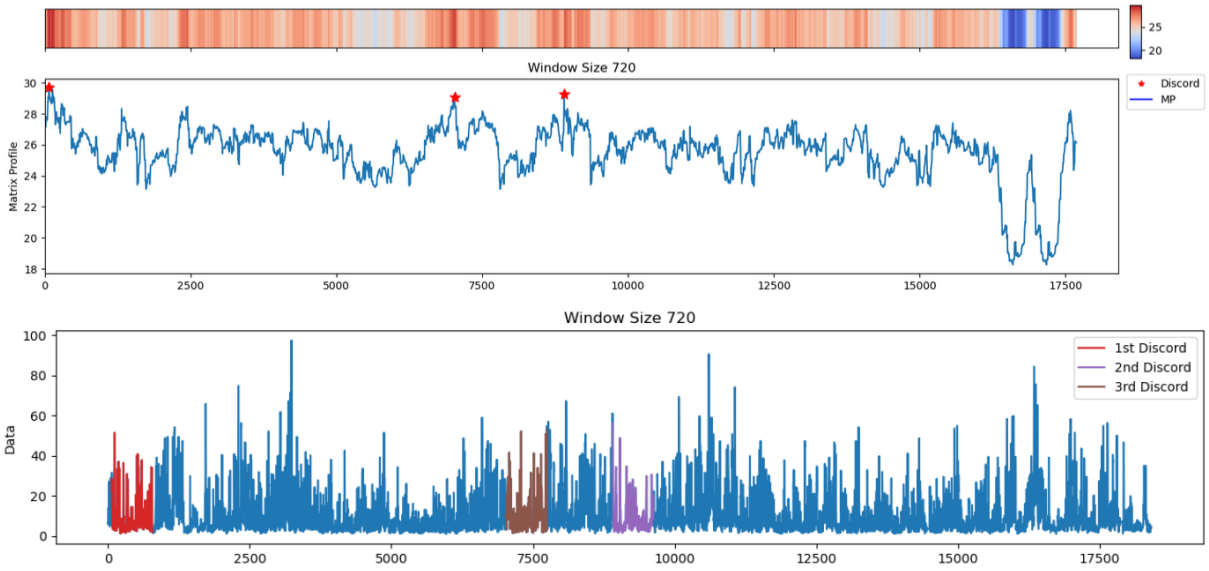


Figura 28: Anomalias do NO_2 com espaço temporal de 1 mês

a imposição e o levantamento de restrições em resposta à pandemia de Covid-19, que estava a decorrer em Portugal nessa época. Esta correlação de acontecimentos destaca como a *Matrix Profile* pode ser valiosa na identificação de eventos não apenas sazonais, mas também relacionados a fatores externos, como crises de saúde pública, o que é crucial para uma análise mais abrangente das séries temporais.

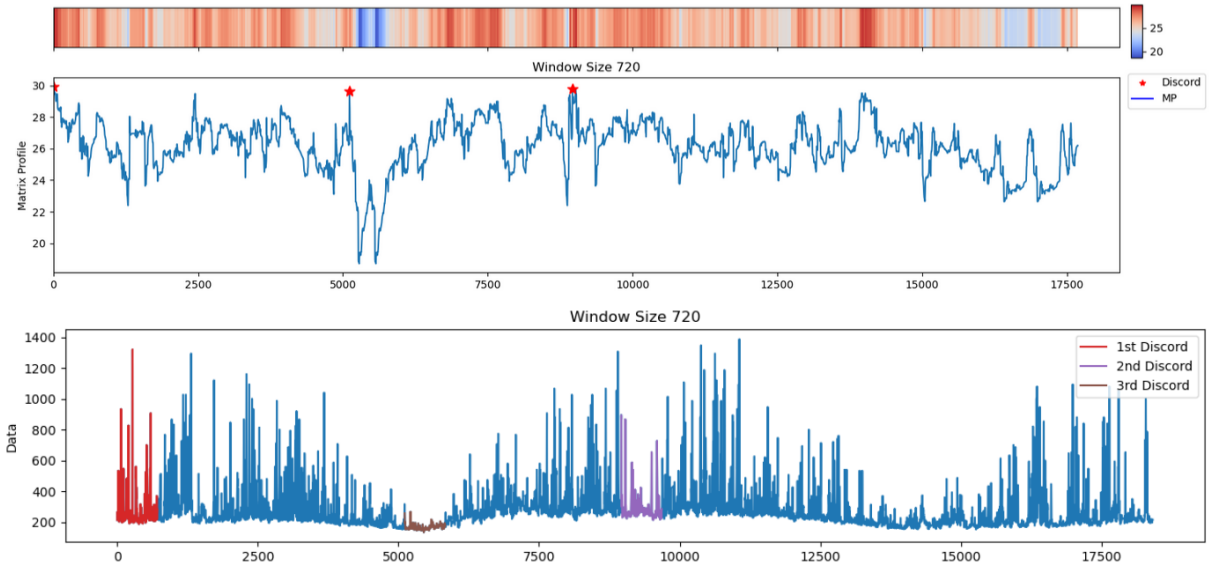


Figura 29: Anomalias do CO com espaço temporal de 1 mês

3.3.2 Decomposição: Tendência e Sazonalidade

Neste subcapitulo foi efetuada a decomposição das séries em tendência e sazonalidade utilizando a biblioteca *Matrix Profile*. À primeira vista, ao analisar as Figuras 18 e 20, podemos afirmar que esta série

não tem nenhuma tendência uma vez que o valor médio mantêm-se constante ao longo de toda a sua extensão. Outra coisa que podemos notar é que parece haver alguma sazonalidade anual com baixas na concentração de CO nos meses de Primavera/Verão e concentrações mais altas no resto do ano. Já para o NO_2, nas Figuras 21 e 23 não é tão claro tirar conclusões sem uma análise mais detalhada. Mesmo que na Figura 23 pareça haver um padrão da sazonalidade como no CO, este comportamento não é tão transparente.

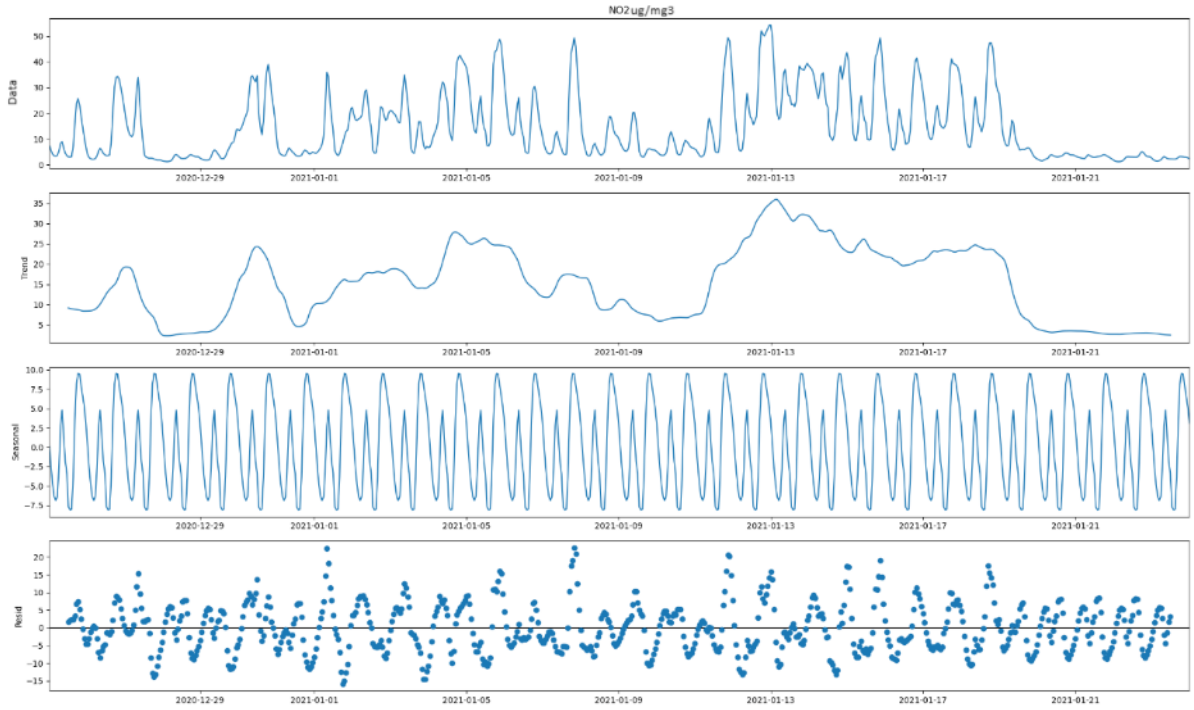


Figura 30: Decomposição sazonal do NO_2

As Figuras presentes nesta secção contêm 4 gráficos que representam a decomposição da série temporal nos vários componentes. O gráfico de tendência, como indica o nome, representa a componente da tendência da série temporal e o gráfico da sazonalidade representa a componente sazonal. O gráfico com os valores residuais é obtido ao calcular a diferença entre a série original, representada pelo primeiro gráfico em cada figura, e a soma das componentes de tendência e sazonalidade. Os valores residuais representam a restante variação sem as componentes de tendência e sazonalidade e incluem *random noise* e flutuações irregulares.

As Figuras 30, 31 e 32 ilustram a decomposição sazonal dos gases CO, NO_2 e O_3, respetivamente, no intervalo de 1 mês (25/12/2020 - 24/01/2021). Ao reduzir a série para este número de *timesteps* é aparente que o comportamento das três é muito parecido, como já era esperado pela correlação forte que o CO e O_3 têm com o NO_2, 0.7 e -0.7 respetivamente.

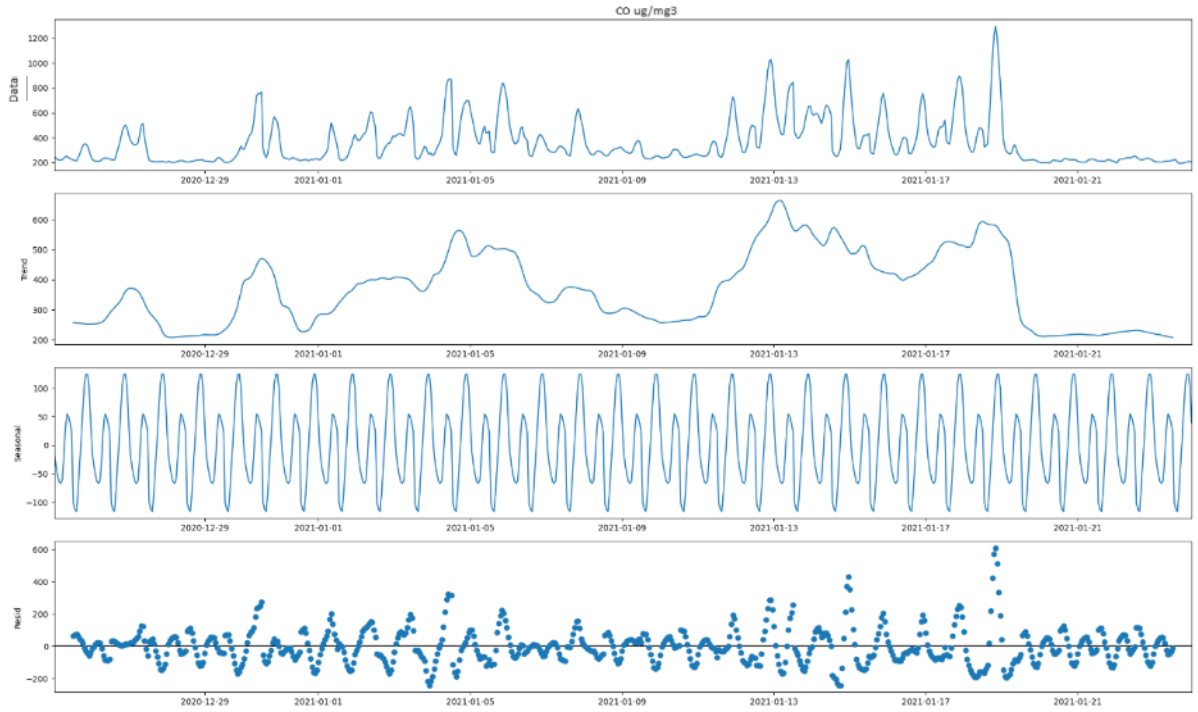


Figura 31: Decomposição sazonal do CO

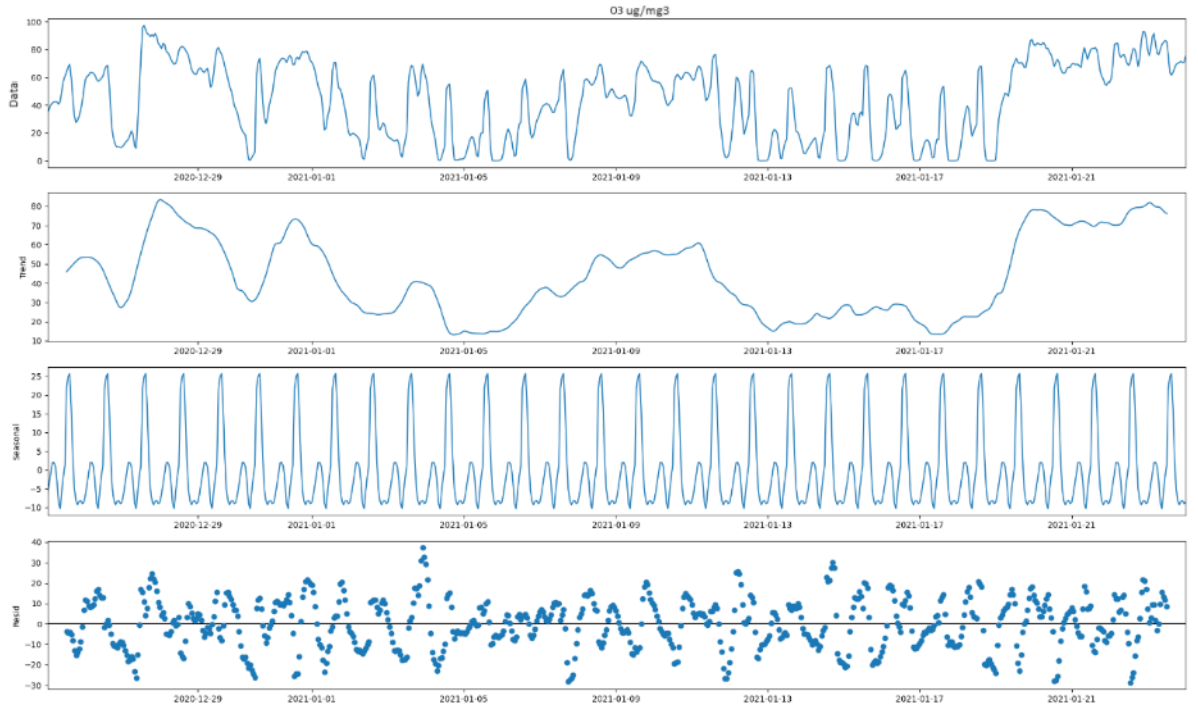


Figura 32: Decomposição sazonal do O_3

Ainda numa análise aos dados, conseguimos observar uma redução nas concentrações de CO e NO_2 a partir de 21 de janeiro e um aumento do O_3. Esta variação atípica aos dados passados, pode justificar-se pela imposição de novas medidas de prevenção sobre a prorrogação do estado de emergência

que estava em vigor em Portugal devido à pandemia da Covid-19 [120]. A componente da tendência das duas é quase idêntico, e como se tinha concluído anteriormente, estas séries não apresentam tendência crescente nem decrescente, pelo menos neste intervalo. Em relação à sazonalidade, as duas apresentam um padrão diário, indicando dois picos nas suas concentrações, todos os dias. Em séries temporais com frequência de 1 hora é possível observar 3 tipos de sazonalidade, diária, semanal e anual. No entanto, nestes casos não é identificável nenhum tipo de padrão semanal na sazonalidade. Uma vez que todas as variáveis apresentam este mesmo comportamento tanto na tendência como na sazonalidade, podemos classificar as séries como *trend stationary* [63].

3.3.3 Autocorrelação

Tendo em conta os modelos candidatos implementados na secção 4, não foram feitas análises aos gráficos de autocorrelação parcial das séries temporais, uma vez que a única finalidade deste tipo de gráficos é perceber se a série temporal em questão pode ser modelada num modelo **AR**. Os gráficos de autocorrelação de séries temporais permitem-nos extrair ainda mais informação acerca da tendência e sazonalidade, bem como perceber se a nossa série contém *white noise*, isto é, se tem um fator de aleatoriedade elevado. Num gráfico de autocorrelação, o *white noise* é identificado quando os pesos dos valores dos *timesteps* passados estão maioritariamente abaixo de um *threshold*, representado pela zona sombreada. Nestes gráficos os *timesteps* passados são identificados como *lag* com a ordem correspondente à sua posição temporal, ou seja, o *timestep* imediatamente anterior ao atual é designado *lag* de ordem 1, dois *timesteps* atrás é designado *lag* de ordem 2, e assim sucessivamente.

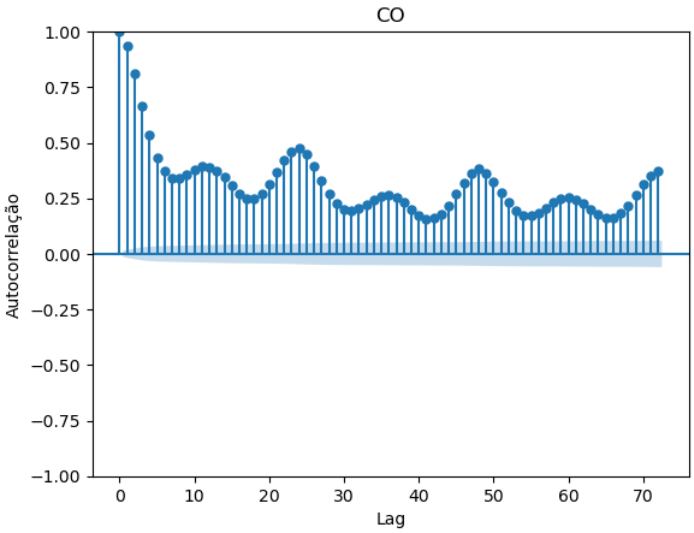


Figura 33: Gráfico de autocorrelação do CO

Ao analisar as Figuras 33, 34 e 35, a primeira carateristica que podemos identificar é a ausência de *white noise*, uma vez que para séries temporais *white noise* esperamos valores de autocorrelação próximos de 0 ou dentro da área sombreada. Analisando a progressão do gráfico, facilmente identificamos a sazonalidade diária que havia sido observada na secção 3.3.2. Além disso, é possível perceber que ao afastarmo-nos do *timestep* atual, a relação entre eles vai sendo cada vez menor, indicando que a série não é estacionária e apresenta tendência. No entanto, ao analisar *lags* mais elevados, os valores de autocorrelação estabilizam em torno de uma média. Desta forma, e com suporte da decomposição efetuada na secção 3.3.2, podemos afirmar que as séries não têm tendência e são caraterizadas como *trend stationary*.

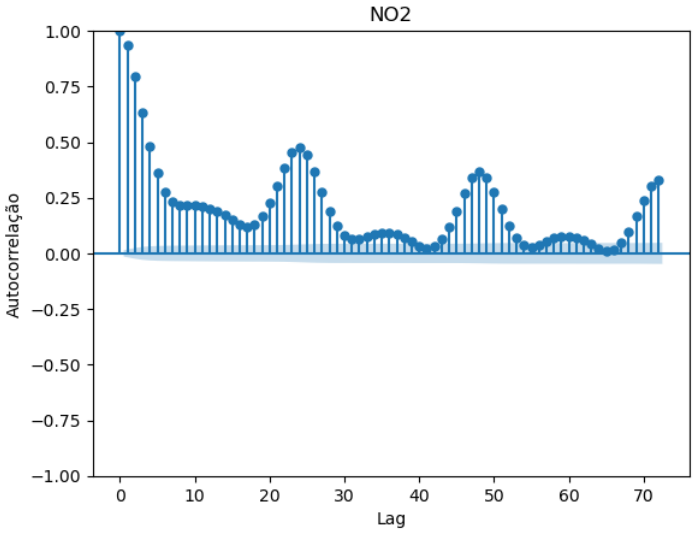


Figura 34: Gráfico de autocorrelação do NO_2

Na Figura 35 notamos um comportamento diferente dos outros. Os valores de correlação oscilam constantemente entre positivos e negativos. Valores negativos significam correlação inversa, ou seja, se no timestep atual temos uma medição do O_3 acima da média, é de se esperar que nos timesteps com correlação negativa a medição do O_3 seja abaixo da média. A autocorrelação negativa pode indicar uma tendência para as séries temporais apresentarem padrões ou ciclos alternados, em que valores elevados são seguidos de valores baixos e vice-versa.

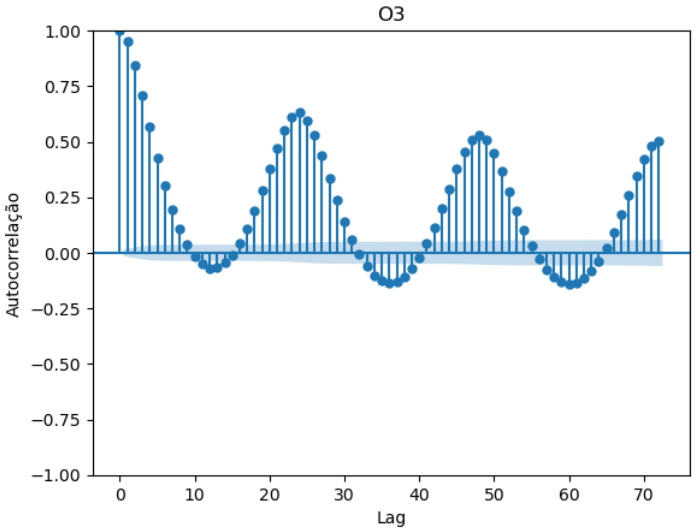


Figura 35: Gráfico de autocorrelação do O_3

3.4 Tecnologias

Ao longo desta dissertação, o código fonte utilizado nas etapas de exploração e tratamento de dados, bem como na conceção, treino e avaliação dos modelos candidatos de **ML**, foi desenvolvido tirando partido da linguagem de programação Python (versão 3.10). Durante o desenvolvimento, foram incluídas algumas bibliotecas como *pandas*, *matplotlib*, *tensorflow*, *numpy*, *seaborn*, *scipy*, *sklearn*, *collections*, *shap* e *Matrix Profile*. Além disso, os ambientes utilizados para apoiar a criação e execução dos *scripts* foram o *Spyder* IDE (https://www.spyder-ide.org/) e o *Google Colaboratory* (https://colab.google/). O *fit* dos modelos candidatos ocorreu no *Google Colaboratory*, que providenciou um GPU de *back-end* do *Google Compute Engine* (tipo T4) com 12 GB de RAM. Por fim, os resultados das experiências foram armazenados em ficheiros csv, através da comunicação efetuada entre o Google *Colaboratory* e o *Google Drive*.

Capítulo 4

Experiências

Neste capítulo são apresentadas todas as abordagens, decisões e técnicas efetuadas relativamente às experiências efetuadas para a seleção do melhor modelo candidato. Em primeiro lugar, é efetuado o processo de implementação dos modelos candidatos, na secção 4.1. Na secção 4.2 são apresentadas as métricas utilizadas para avaliar o desempenho dos diferentes modelos candidatos e selecionar o melhor. Por fim, é apresentada a pipeline seguida para a execução das experiências na secção 4.3.

4.1 Concepção dos Modelos Candidatos

De forma a selecionar o melhor modelo candidatado, as experiências desenvolvidas tiveram por base as **LSTM**. A escolha deste modelo de **DL** teve por base dois motivos. O primeiro é ser um modelo de *BlackBox* e a sua imprevisibilidade torna ainda mais cativante o uso de técnincas de explicabilidade para a sua interpretabilidade. O outro motivo, foi que dos modelos de **DL** este é o mais adequado para problemas de previsão de séries temporais, como relatado na secção 2.3. A implementação dos vários modelos candidatos **LSTM** foram efetuados no *SpyderIDE*, com recurso à biblioteca do *Python tensorflow.keras*. Esta implementação consistiu ainda numa abordagem dinâmica, que permite ao utilizador selecionar vários hiperparâmetros aquando da inicialização dos modelos com a finalidade de conseguirmos selecionar os melhores.

Na arquitetura do modelo foram utilizadas as classes *Input*, **LSTM**, *Dense*, *Dropout* e *Model* do *tensorflow*. O primeiro passo da construção do modelo é o estabelecimento de uma *seed*, com valor 91195003, para garantir a reprodutibilidade dos resultados. Seguidamente foi criado um tensor que representa os *Inputs* a serem injetados na arquitetura principal para se obterem os *outputs* no final. A arquitetura principal da rede é conseguida num ciclo em que em cada iteração é instanciada uma camada da rede com base nos parâmetros fornecidos pelo utilizador. Estes parâmetros são atribuídos às variáveis *timesteps*, *features*, *layers*, *neurons*, *activation* e *dropout*. Os *timesteps* representam o número

de registos da série temporal, que pertencem à sequência que serão utilizados para efetuar a previsão, isto é, se quisermos prever o nível de NO_2 para as próximas 24 horas com base nos ultimos 36 registos, o número de *timesteps* será 36. Relativamente às *features*, estas representam o número de colunas que o *dataset* de treino possui. A variável *layers* representa o número de camadas **LSTM** que a rede possui, sendo também o valor pelo qual o ciclo itera. O número de neurónios utilizados nas várias camadas da rede neuronal é indicado em *neurons*, a função de ativação na última camada da rede é a variável *activation* e o *dropout* indica a percentagem de neurónios que serão desativados da rede por forma a generalizar o modelo e prevenir *overfitting*. Para a compilação do modelo, na função de *loss* foi utilizado **Root Mean Squared Error (RMSE)** e o *optimizer* utilizado, foi o *Adam*, respetivamente.

Em 4.1 está um excerto do código explicado no último parágrafo para ajudar a visualizar a arquitetura.

Listing 4.1: Construção da LSTM

```
1
2  def build_lstm_model(features, timesteps=7, layers=2, neurons=64,
3  activation='relu', dropout_rate=0.5, seed=91195003):
4      tf.random.set_seed(seed)
5
6      inputs = Input(shape=(timesteps, features))
7
8      for i in range(layers):
9          #if first LSTM layer
10         if i == 0:
11             #if also the last LSTM layer then return_sequences=False
12             if i+1 == layers:
13                 x = LSTM(neurons/2, return_sequences=False)(inputs)
14             #it has more layers! So return_sequences=True
15             else:
16                 x = LSTM(neurons, return_sequences=True)(inputs)
17         #if last LSTM layer then return_sequences=False
18         elif i+1 == layers:
19             x = LSTM(neurons*2, return_sequences=False)(x)
20         #if not the last LSTM layer then return_sequences=True
21         else:
22             x = LSTM(neurons, return_sequences=True)(x)
```

```
23
24      x = Dense(neurons, activation=activation)(x)
25      x = Dropout(dropout_rate)(x)
26      outputs = Dense(1)(x)
27
28      lstmModel = Model(inputs=inputs, outputs=outputs, name='
            lstm_model')
29
30      lstmModel.compile(
31          loss = rmse,
32          optimizer= tf.keras.optimizers.Adam(),
33          metrics = ['mae', rmse]
34      )
35
36      return lstmModel
```

No que diz respeito ao treino, foram ainda utilizados os gráficos das *learning curves* com o intuito de analisar o desempenho da aprendizagem dos modelos, e principalmente para escolher o número de *epochs* mais adequado para evitar situações de under e *overfitting*. Para complementar, foram utilizadas as callbacks *ReduceLROnPlateau*, e *EarlyStopping*. Para avaliar a performance do modelo, o *dataset* passou por um processo de *cross-validation* com *TimeSeriesSplit*, de modo a não quebrar a sequência temporal dos dados. Este *cross-validator* foi configurado com *n_splits* igual a 4, e foi a partir do mesmo que foram criados os conjuntos de treino, teste, e validação. Por fim, é importante referir ainda que o processo de previsão seguiu uma abordagem *Multistep Recursive Forecasting*, que consiste na utilização subsequente das previsões feitas pelo modelo para a próxima previsão. As métricas foram calculadas utilizando a média dos erros de todas as previsões do *Multistep Recursive Forecasting*.

A Tabela 9 mostra todos os valores testados nos hiperparametros para otimizar o modelo. O resultado são 972 experiências que correspondem a todas as combinações possíveis com estes valores.

Finalmente, antes de inserir os dados para treino, todas as *features* são normalizadas para valores entre -1 e 1, uma vez que as camadas de **LSTM** apresentam na sua arquitetura a aplicação da função **tanh** [121].

Hiperparâmetros	Intervalo de valores
layers	[3,4,5,6]
neurons	[32,64,128]
activation	[relu, tanh, sigmoid]
dropout	[0.0, 0.25, 0.5]
timesteps	[12,24,36]
batch size	[5,10,20]
epochs	125

Tabela 9: Hiperparâmetros para otimizar o modelo

4.2 Métricas de Avaliação

Após a implementação do modelo e a otimização dos seus hiperparâmetros, o que levou à execução de várias experiências, foi necessário analisar os resultados obtidos e avaliar os melhores modelos candidatos. Uma vez que a previsão dos níveis de NO_2 se trata de um problema de regressão linear, foram utilizadas duas métricas de avaliação nesta dissertação, nomeadamente o **MAE** e o **RMSE**. Estas métricas são baseadas no cálculo dos erros absolutos das previsões, comparando os valores previstos com os valores reais, através da diferença absoluta entre os mesmos.

Ambas as métricas calculam a magnitude média do erro das previsões. No caso do **MAE**, é calculada a diferença média entre os valores reais e os previstos, enquanto o **RMSE** representa a raiz quadrada da média do quadrado dos erros, calculados pela diferença absoluta entre os valores reais e os valores previstos [122, 123]. Uma vez que os erros são elevados ao quadrado antes de ser calculada a média, o **RMSE** dá pesos mais elevados a erros maiores, o que torna esta uma boa métrica quando erros grandes são particularmente indesejados. Para encontrar um bom modelo de previsão, procuramos que estas métricas sejam o mínimo possíveis, de modo a selecionar os modelos com o erro mais baixo, estes foram ordenados pelo seu **RMSE**. Em seguida apresentam-se as fórmulas de cálculo das métricas nas equações 4.1 e 4.2, onde n representa o número de pontos do *dataset* [124].

$$\textbf{MAE} = \frac{\sum_{i=1}^{n} |y_i real - y_i previsto|}{n} \tag{4.1}$$

$$\textbf{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i real - y_i previsto)^2}{n}} \tag{4.2}$$

4.3 Pipeline e Tecnologias

Nesta secção é apresentada a pipeline utilizada para a execução das várias experiências de seleção do melhor modelo candidato. Tendo os dados finais num único documento .csv e todo o código necessário implementado, foi criada uma sequência de etapas necessárias para efetuar as experiências de forma organizada e eficiente. A pipeline criada consiste então em 3 etapas fundamentais descritas em seguida. A Figura 36 ilustra a sequência de etapas a pipeline.

1. **Upload do dataset e do código fonte do modelo:** O *dataset*, cuja sua análise e tratamento foi abordado na secção 3, e os ficheiros que contêm o código para a concepção dos modelos candidatos são os principais recursos deste capítulo. Desta forma, a primeira etapa consiste no armazenamento destes ficheiros num ambiente em que possam ser acedidos com facilidade para a execução das experiências. Para tal, foi utilizado o *Google Drive*, onde foi feito o *upload* do *dataset* no formato *.csv* e vários ficheiros no formato *.py* que incluem todo o código de implementação do modelo **LSTM**.

2. **Criação de Jupyter notebook no Google Colaboratory:** Nesta etapa foi utilizado o *Google Colaboratory* para a criação do notebook onde foi escrito o código, em *python*, para a validação e treino dos modelos. Sendo uma ferramenta da *Google*, criada a partir do *Google Colaboratory*, este notebook permite aceder ao *Google Drive* para importar os ficheiros com o código e *dataset*. Neste notebook o *dataset* foi devidamente preparado para poder ser ingerido nos modelos e estes foram criados e testados com todas as combinações das configurações representadas na Tabela 9.

3. **Execução das experiências e armazenamento de resultados:** Como etapa final da *pipeline* é aqui que todas as experiências são executadas através do código produzido no *notebook*, utilizando os recursos do *Google Colaboratory*. Uma vez que o número de experiências era demasiado elevado, isto resultou na geração de vários ficheiros de resultados. Estes ficheiros eram então gradualmente armazenados no *Google Drive* para poderem ser posteriormente acedidos para análise e seleção do melhor modelo.

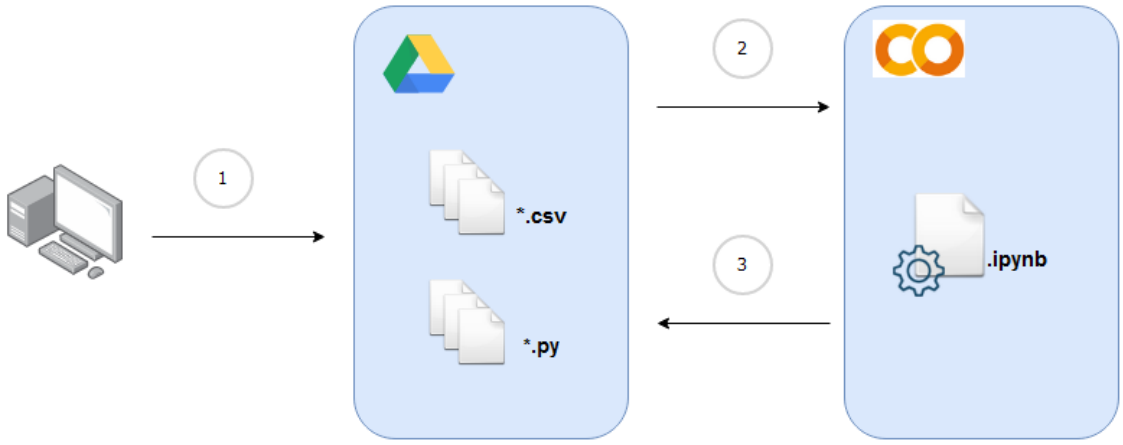


Figura 36: Pipeline das experiências

Capítulo 5

Resultados e Discussão

Este capítulo apresenta os resultados obtidos das experiências realizadas ao longo desta dissertação. É feita a análise dos diversos modelos candidatos bem como a seleção do melhor em 5.1, e são demonstradas as várias técnicas de explicabilidade aplicadas e o estudo dos seus resultados em 5.2. Depois de percorrer e entender as várias abordagens, é feita uma análise comparativa dos resultados de todas as técnicas, em 5.3.

5.1 Seleção do modelo

Nesta secção serão apresentados o top-5 dos melhores modelos candidatos resultantes das experiências realizadas, bem como uma discussão sobre os valores dos seus hiperparâmetros.

Na Tabela 10 podemos visualizar as 5 melhores modelos candidatos **LSTM**, ordenados por ordem crescente de **RMSE** e **MAE**, bem como os valores de cada um dos seus parâmetros. No topo da Tabela 10 temos então o modelo candidato selecionado, que será utilizado na continuidade desta dissertação para aplicação das técnicas de explicabilidade. Os valores calculados para o **RMSE** e **MAE** desnormalizados foram, respetivamente, 1.870 $\mu g/mg^3$ e 1.271 $\mu g/mg^3$. Olhando para as *layers* e *neurons*, o modelo possui 3 camadas **LSTM**, sendo que as primeiras duas possuem 128 neurónios cada e a última camada possui o dobro, ou seja 256 neurónios. A *dropout_rate* de 0.25 significa que na fase de treino o modelo teve de retirar $1/4$ dos neurónios. *dropout_rate* é uma técnica utilizada em redes neuronais durante a fase de treino para prevenir o *overfitting*. Durante cada iteração do treinamento, uma fração aleatória de neurónios em uma camada é temporariamente desativada ou eliminada, fazendo com que essas unidades não contribuem para o cálculo dos gradientes e atualização dos pesos. Essa eliminação temporária de neurónios ajuda a prevenir a coadaptação entre neurónios, forçando a rede a aprender representações mais robustas [125]. Este modelo obteve melhores resultados ao receber um *input* com 36 *timesteps* para prever os próximos 24 *timesteps*, uma vez que segue uma abordagem *multistep Recursive*

#	layers	neurons	dropout_rate	activation	timesteps	batch_size	MAE	RMSE
1	3	128	0.25	relu	36	10	1.271	1.870
2	3	64	0.25	tanh	24	10	1.352	1.940
3	4	32	0	tanh	24	10	1.366	1.974
4	3	64	0.5	tanh	24	5	1.368	1.974
5	3	128	0.25	relu	24	20	1.410	1.977

Tabela 10: Resultados do top-5 dos modelos candidatos.

Forecasting. Além disso a função de ativação e o *batch_size* são **Unidade Linear Retificada (ReLU)** e 10, respetivamente.

Comparando o top-5 dos melhores modelos candidatos percebemos uma relação entre as *layers*, e *neurons*, em que para obter boa performance, os modelos com mais neurónios têm menos *layers* e vice-versa. Esta relação é alusiva ao facto de que normalmente um modelo com muitos neurónios pode resultar num *overfitting* dos dados [126].

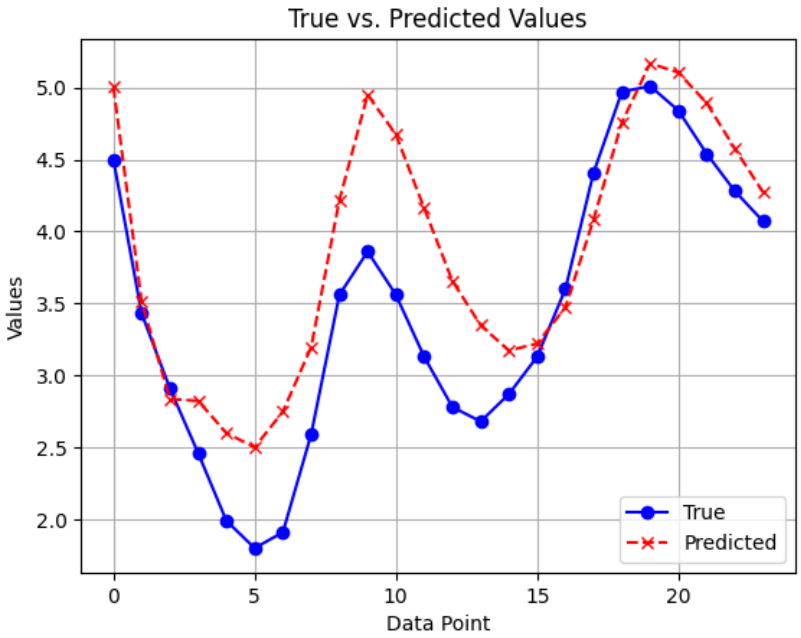


Figura 37: Valores reais vs Valores previstos

A função de ativação mais frequente é a **tanh**, no entanto notamos que os modelos com mais neurónios utilizam a função de ativação **ReLU**. Devemos notar que os modelos neste top-5 parecem convergir para *inputs* de 24 *timesteps*, mas o modelo que obteve melhores resultados recebe um input de 36 *timesteps*. Uma possível explicação tem a ver com a abordagem *multistep Recursive Forecasting*. Ao

fazer previsões para os 24 *timesteps* com base nos *timesteps* previstos anteriormente, com 36 *timesteps* damos ao modelo mais valores reais para fazer previsão. Os 3 modelos candidatos com melhor performance têm *batch_size* de 10, ou seja, que cada modelos recebe *batches* de 10 *inputs* para treino. Na Figura 37 podemos comparar os valores previstos pelo modelo com os valores reais. Podemos observar que o modelo acompanha a curva dos valores reais, sendo um indicativo de que este aprendeu o comportamento da sériee é capaz de a replicar, no entanto, os valores previstos são na sua maioria ligeiramente superiores com interseções entre as duas linhas.

5.2 Técnicas de Explicabilidade

Tendo selecionado o melhor modelo, a próxima fase é aplicar técnicas de explicabilidade para ajudar a interpretar as previsões feitas pelo modelo. Neste subcapítulo serão apresentadas as 3 técnicas implementadas e os seus resultados. Começamos por aplicar explicabilidade baseada em *Attention*, como uma abordagem *ante-hoc*, em 5.2.1. Na secção 5.2.2, numa abordagem *post-hoc*, recorreu-se à biblioteca *shap* para calcular os *shapley values* para cada *feature* em cada *timestep*.Por fim, em 5.2.3, também com foco em *post-hoc*, fez-se *feature permutation* para perceber o impacto de cada feature nas previsões. É importante referir que um dos objetivos desta dissertação é aplicar explicabilidade antes e depois do treino do modelo (*ante-hoc* e *post-hoc*).

5.2.1 Attention

A primeira estratégia para extrair explicabilidade e tornar o modelo mais interpretável foi feita com recurso a *attention*. Os mecanismos de atenção calculam os pesos de atenção, *attention weights*, dos *timesteps* de *input*. Estes pesos, que indicam a importância de cada *timestep* nas previsões, são posteriormente organizados num vetor de contexto a passados como *input* ao modelo de previsão [127]. Desta forma, o modelo passou a ser dividido em duas fases. Uma primeira camada de *Attention* que receberá o *input* original, ou seja, as séries temporais de cada *feature*. Na segunda fase, o nosso modelo que receberá como *input* o vetor de contexto da *attention* que contém a informação das *features* e *timesteps* mais relevantes para a previsão. Uma vez feita a previsão, podemos extrair o vetor de contexto da camada de Atenção e analisar os valores previstos com os pesos atribuidos às *feature* e *timesteps* do *input* da previsão [128]. A Figura 38 representa as duas arquiteturas para comparação.

A camada de *Attention* utilizada foi implementada de acordo com os objetivos do problema e a arquitetura.

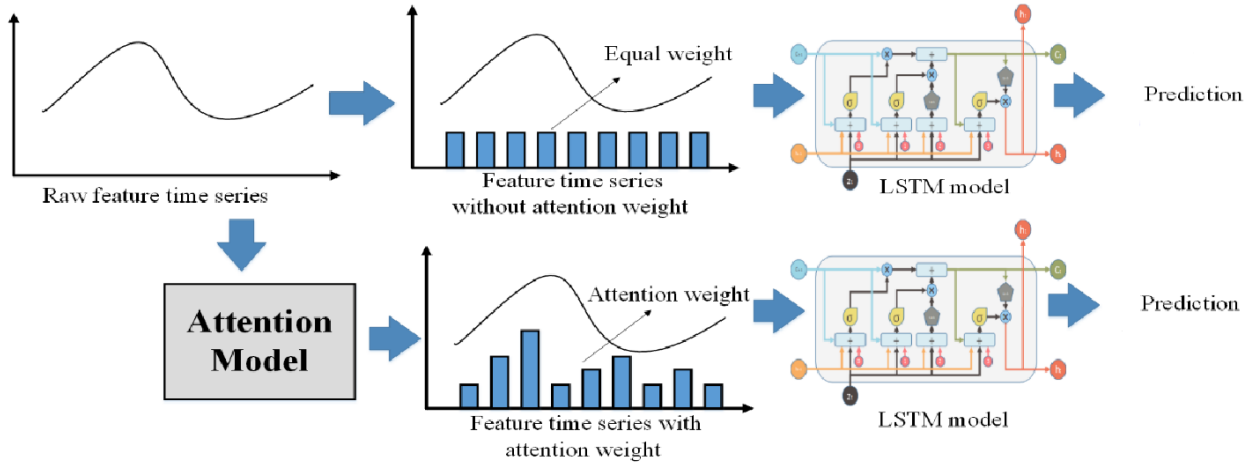


Figura 38: Pipelines de previsão com Attention e sem Attention. Fonte [128]

Listing 5.1: Construção da Attention

```
class Attention2(tf.keras.layers.Layer):
    def __init__(self, units):
        super(Attention2, self).__init__()
        self.units = units

    def build(self, input_shape):
        self.W = tf.keras.layers.Dense(self.units, activation='tanh')
            # Specify activation function
        self.V = tf.keras.layers.Dense(1, activation=None)  # No
           activation for the output layer
        super(Attention2, self).build(input_shape)

    def call(self, inputs):
        # Compute attention scores
        score = tf.nn.tanh(self.W(inputs))
        attention_weights = tf.nn.softmax(self.V(score), axis=1)

        # Apply attention weights to input
        context_vector = attention_weights * inputs
        #context_vector = tf.reduce_sum(context_vector, axis=1)

```

```
20          return context_vector
```

No excerto de código em 5.1 é possível ver a implementação da classe que representa a camada de atenção. É possível dividir o código em 3 fases de computação dos pesos:

1. **Cálculo das pontuações de atenção:** as pontuações de atenção são calculadas aplicando uma função de ativação *tanh* à transformação linear da sequência de *input* utilizando uma matriz de pesos aprendida *self.W(inputs)*. Este é um mecanismo baseado no conteúdo, porque se concentra no conteúdo da sequência de *input* ao calcular as pontuações de atenção.

2. **Cálculo dos Attention weights:** as pontuações de atenção calculadas são depois utilizadas para calcular os pesos de atenção através de uma função *softmax*, tornando-os numa distribuição de probabilidade sobre a sequência de *input*.

3. **Aplicação de pesos de atenção ao input:** os pesos de atenção são então aplicados à sequência de *input* através da multiplicação por elementos, destacando os elementos relevantes da entrada com base no seu conteúdo. O vetor de contexto final é obtido através de uma soma ponderada dos elementos de entrada utilizando estes pesos de atenção.

De forma a testar a viabilidade dos resultados da *attention*, o novo modelo **Attention-Long-Short Term Memory (Att-LSTM)** foi treinado com os mesmos dados da **LSTM** para verificar se os pesos de atenção calculados faziam sentido no contexto do problema e se não era necessário outra abordagem. Com base nos dados da Tabela 11, vemos uma depreciação na performance do modelo quando aplicamos a *attention*. Em ambas as métricas os valores do modelo com *attention* aumentaram em 0.6, tendo em atenção que um aumento de 0.6 do **RMSE** é mais significativo que o mesmo aumento na **MAE**, pela forma como ambas as métricas são calculadas. Tendo em conta que nossa variável de previsão varia entre 0.89 e 97.33 $\mu g/mg^3$, em 6 foi decidido que um aumento do erro de 0.6 não justifica a exclusão desta abordagem. Assim, podemos extrair o vetor de contexto da camada de atenção e analisar os pesos atribuídos a cada *feature* em cada *timestep*. Podemos ver a performance do modelo com attention na Figura 39, onde temos os valores reais e os valores previstos lado a lado. Assim como no modelo sem *Attention*, podemos ver que apesar do aumento do erro, o modelo com *Attention* acompanha a curva dos valores reais, no entento neste caso a a linha das previsões mantêm-se maioritariamente abaixo dos valores reais.

A Figura 40 representa o vetor de contexto estraído da *attention*. O eixo horizontal representa os *timesteps* sendo o *timestep* mais antigo o 0 e o mais recente o 35, ou seja, na série temporal queremos

	LSTM	ATT-LSTM
MAE	1.271	1,806
RMSE	1,870	2,456

Tabela 11: Performance do modelo com e sem Atention

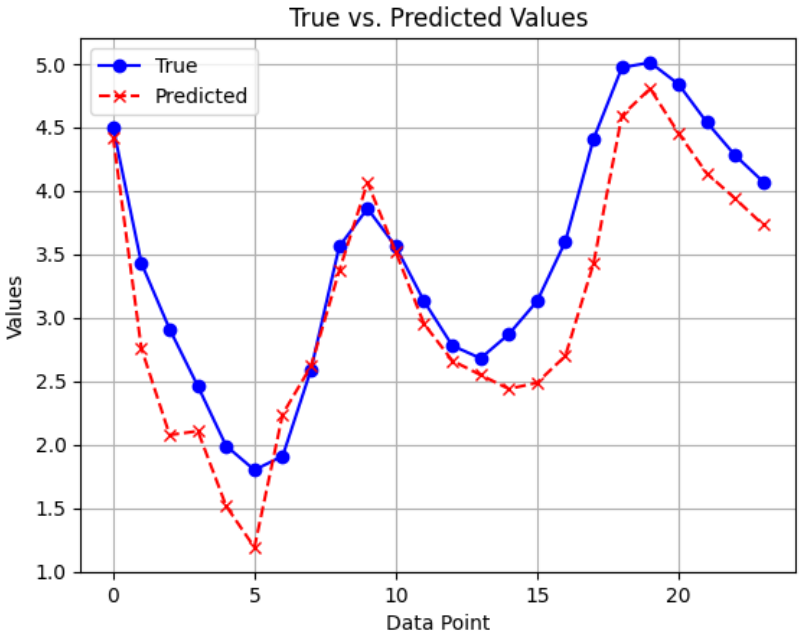


Figura 39: Valores reais vs Valores previstos

prever o *timestep* 36. As cores representam a intensidade dos pesos atribuídos, sendo amarelo o mais forte e azul mais fraco. No eixo vertical, cada linha representa uma *feature*. A Tabela 12 mapeia cada linha do eixo vertical com a respetiva *feature*.

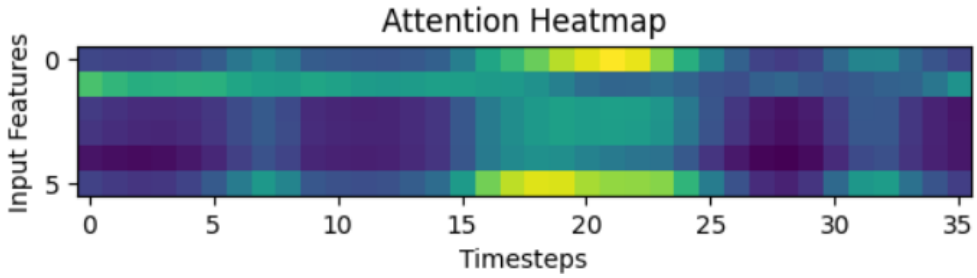


Figura 40: Pesos de atenção atribuídos à sequência de input.

A análise da imagem permite extrair que as duas *features* que a *attention* decidiu serem mais relevantes foram o CO e o NO_2, em torno do *timestep* 20 (15 horas no passado). Como vimos no capitulo 3, o CO tem uma correlação bastante elevada com a *feature* que queremos prever, então podemos esperar que seja de alguma forma relevante na sua previsão. No entanto, como as séries do CO e do NO_2 se

0	1	2	3	4	5
CO	O_3	$PM_{2.5}$	PM_{10}	NH_3	NO_2

Tabela 12: Mapeamento das *input features* da Figura 40.

comportam exatamente da mesma forma, era de se esperar que o *timestep* 12, (24 horas no passado) tivesse mais relevância, uma vez que identificamos sazonalidade diária em ambas as séries. Até mesmo para o NO_2, que é a própria variavel de previsão, isto não acontece. O mesmo acontece para o $PM_{2.5}$, PM_{10} e NH_3, embora a camada não dê tanta atenção, também dá mais relevância ao *timestep* 20. Já para a *feature* O_3, quanto mais longe no passado maior relevância esta variável parece ter. Apesar disso, e da elevada correlação de -0.7 que o O_3 também apresenta com o NO_2, nenhum *timestep* chega a ser tão relevante como acontece com o CO. Isto é uma demonstração de que correlação não implica atenção nem causalidade, pois a correlação apenas indica uma relação, mas não aponta para nenhuma direção de causa [129].

Apesar dos valores dos *attention weights* não parecerem ter relação com o que estudamos sobre as séries no capitulo 3, uma vez que apontam para *timesteps* diferentes dos esperados, é um facto que ainda assim o modelo com *attention* teve uma boa performance comparativamente ao modelo sem *attention*. Este performance é reconhecida porque, apesar da sazonalidade diária, podem haver pequenas variações nos dados no intervalo de 24 horas, ou até mesmo anomalias ou eventos que não sejam fáceis de identificar na exploração dos dados e que o modelo deteta e acha mais relevantes [106]. Na Figura 41, podemos ver a progressão dos pesos atribuídos às *features* ao longo dos *timesteps*.

5.2.2 Shapley Values

O segundo método de explicabilidade implementado foram *Shapley Values* com recurso à biblioteca **SHAP** do *Python*. O **SHAP** é uma abordagem unificada para explicar os resultados de qualquer modelo de **ML**. Neste caso, o *output* será um conjunto de dados em 3D que contém a importância de cada amostra, *timestep* e *features*, por esta ordem. Se somarmos o *output* na dimensão 0 (ou seja, no eixo da amostra), obtemos dados 2D que contêm a importância das *features* como colunas e os *timesteps* como linhas. Desta forma, o **SHAP** fornece a importância das *features* em função do intervalo de tempo. Este método foi aplicado primeiramente ao modelo sem *attention* e, posteriormente, ao modelo com *attention* de forma a ver como os pesos de atenção influênciam o modelo na tomada de decisão.

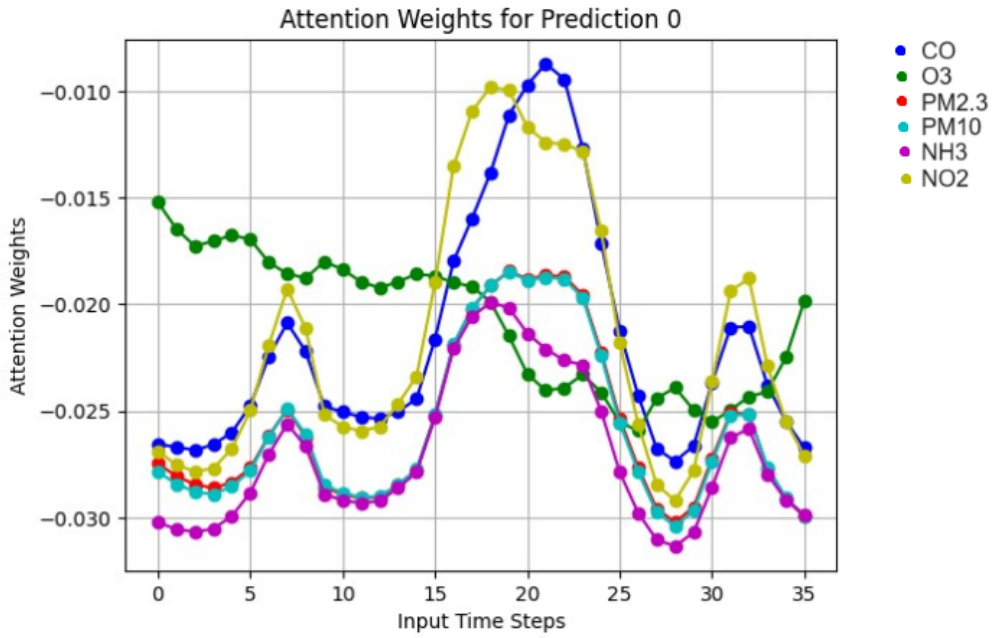


Figura 41: Progressão dos pesos de atenção das *features* ao longo dos *timesteps*.

LSTM

Começando pela **LSTM** sem *Attention*, foram escolhidos os 7224 *timestamps* mais recentes do conjunto de dados do treino para passar na classe *DeepExplainer*, uma vez que era o máximo de valores que a máquina, com os recursos disponíveis, conseguia processar ao mesmo tempo. A passagem de todos os dados do treino dará valores muito precisos, mas tornasse demasiado difícil de computar, uma vez que a complexidade deste método aumenta linearmente com o número de pontos de dados. Destes 7224, foram separados ainda os 7200 mais antigos para o *explainer* aprender e, após isso, foi pedido ao **SHAP** para computar os **SHAP** values dos 24 *timesteps* mais recentes. A classe *DeepExplainer* do **SHAP** foi concebida especificamente para computar os *shapley values* de modelos de **DL**. O resultado pode ser visto na Figura 42.

A Figura 42 representa um *heatmap* dos *shap values* de todas as *features*, em cada *timestep*. Esta representação 2D é conseguida após somar todos os valores, em módulo, da dimensão 0 (*samples*) do *output*. Os *timesteps* estão ordenados de baixo para cima, do mais recente para o mais antigo, e os valores de importância variam desde 0 até um máximo de 4.03. Como se pode observar pela matriz, a *feature* mais relevante para a previsão é o NO_2. Esta detém os 3 maiores valores de importância nos timesteps 33, 34 e 35 sendo estes 1.16, 2.84 e 4.03 respetivamente. Estes *timesteps* dizem respeito às 3 horas anteriores à previsão. Na hora anterior à previsão, a segunda *feature* mais relevante é o CO, sendo este ultrapassado pelo O_3 que passa a ser a segunda *feature* mais relevante a partir da segunda

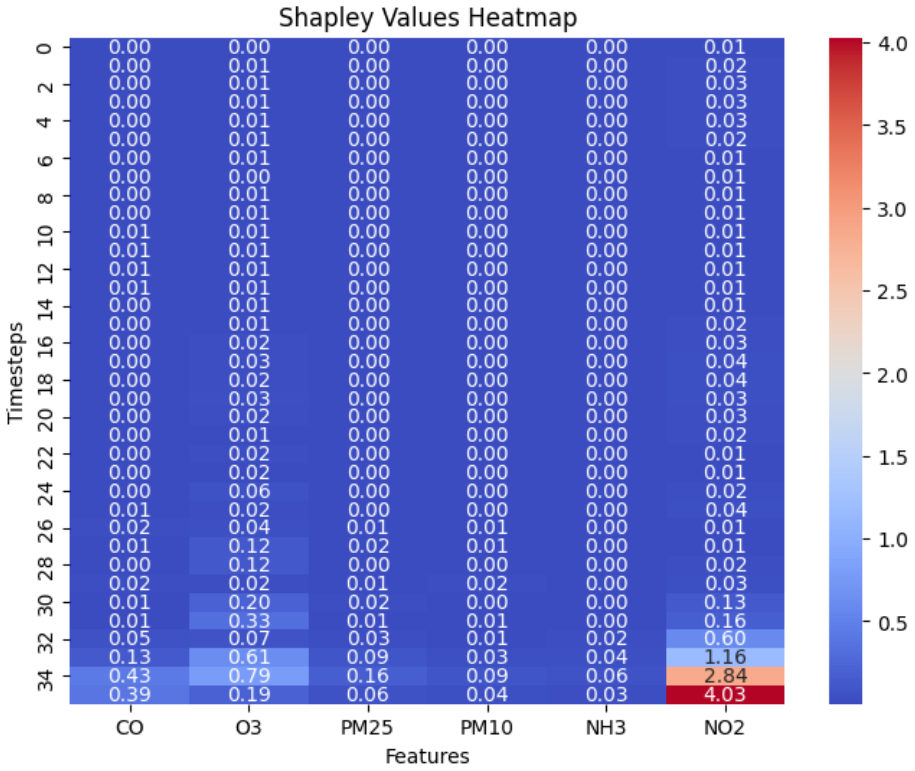


Figura 42: Heatmap da soma dos os valores de importância por timestep

hora anterior à previsão chegando mesmo a ser a mais importante ultrapassando a própria variável da previsão.

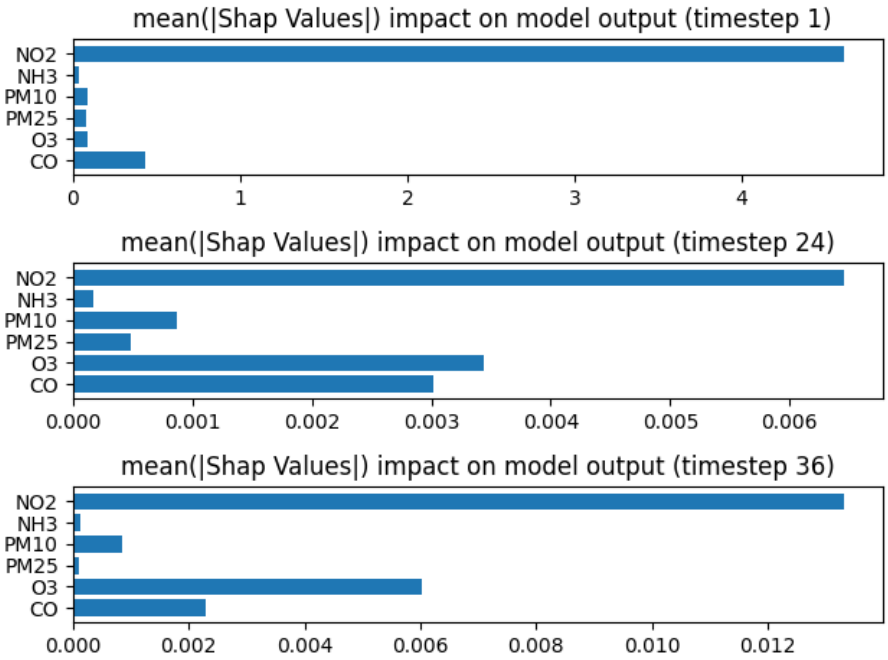


Figura 43: Shap Values nos *timesteps* 1, 24 e 36.

De uma forma geral, os valores tendem a diminuir conforme avançamos para o passado, mas não

parecem voltar a subir na marca das 24 horas (*timestep 12*). Da mesma forma que referimos em 5.2.1, apesar da sazonalidade diária, podem haver pequenas variações nos dados no intervalo de 24 horas, ou até mesmo anomalias ou eventos que não sejam fáceis de identificar na exploração dos dados, mas que o modelo deteta e acha mais relevantes. Na Figura 43, é possível observar de forma mais detalhada os *shap values* para os timesteps 1, 24 e 36. Os *shap values* do timestep 36 contém cerca do dobro da importância dos *shap values* no *timestep* 24.

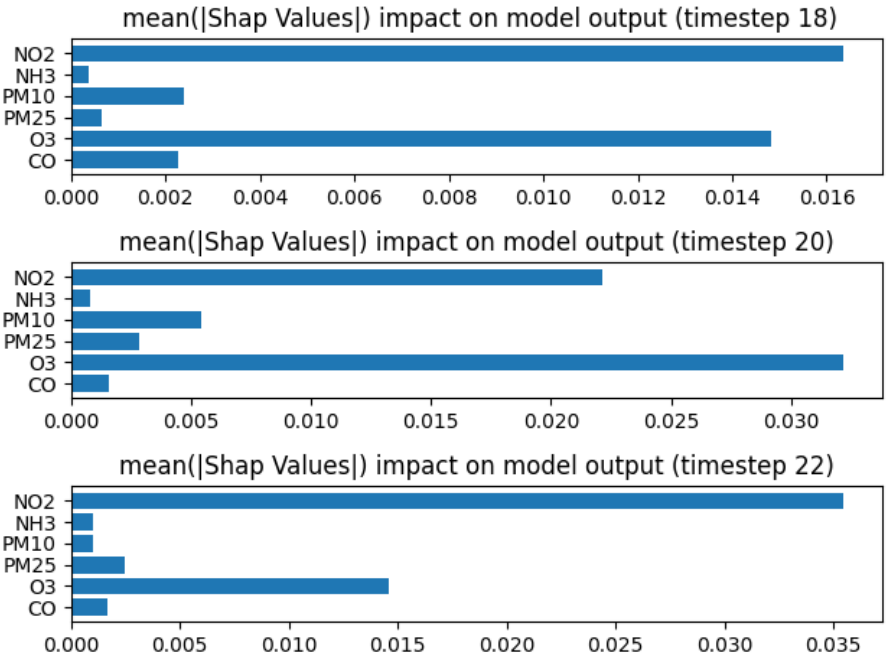


Figura 44: Shap Values nos *timesteps* 18, 20 e 22.

Além disso, para o NO_2 e O_3 parece haver uma subida nos valores de importância em torno do *timestep* 20, que corresponde a 15 horas no passado, e no *timestep* 3, 33 horas no passado, mas desta vez apenas para o NO_2. Para uma melhor análise, podemos observar os valores das variáveis nos *timesteps* 18, 20 e 22 na Figura 44. Mais uma vez no *timestep* 20, o O_3 volta a passar o NO_2 quanto à importância para a previsão, ficando com um valor quase igual no *timestep* 18. Este comportamento volta a repetir-se em relação ao observado com os pesos de atenção, sendo mais um indicativo de que as séries contém variações e anomalias que não são facilmente identificáveis apenas com análises estatísticas. O NO_2 é a *feature* mais importante para a previsão, sendo o seu valor cerca de 6 vezes superior à do O_3 que vem em segundo e logo atrás o CO, como podemos observar na Figura 45 .

De forma geral, na Figura 45 podemos ver a a importância total de todas as *features*, como a soma de todas os *shap values* em todos os *timesteps*. O NO_2 é a *feature* mais importante para a previsão, sendo o seu valor cerca de 6 vezes superior à do O_3 que vem em segundo e logo atrás o CO. De acordo

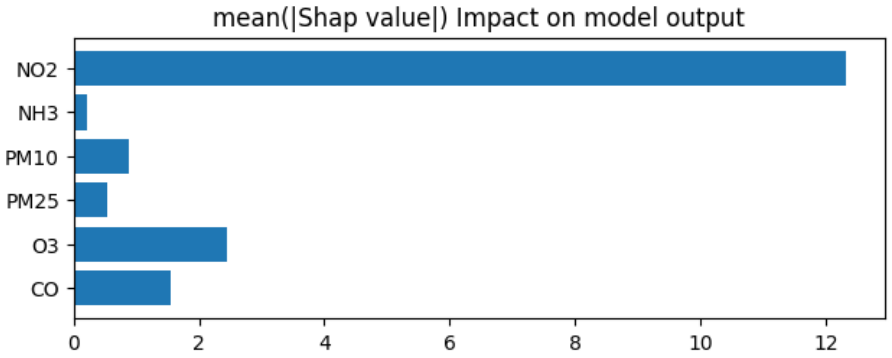


Figura 45: Soma da importância de todos os timesteps

com o *Shap*, as restantes *features* não têm praticamente influência na previsão do NO_2, com o NH_3 demonstrando sempre os menores valores na matriz. Desta forma, podemos concluir que estas variáveis podiam ser descartadas do modelo de previsão.

Att-LSTM

Para o modelo com *attention*, para obter resultados mais objetivos, os dados para o *explainer* foram retirados da mesma forma que na **LSTM** sem *Attention*. Os valores de **SHAP** computados para este modelo encontram-se na Figura 46.

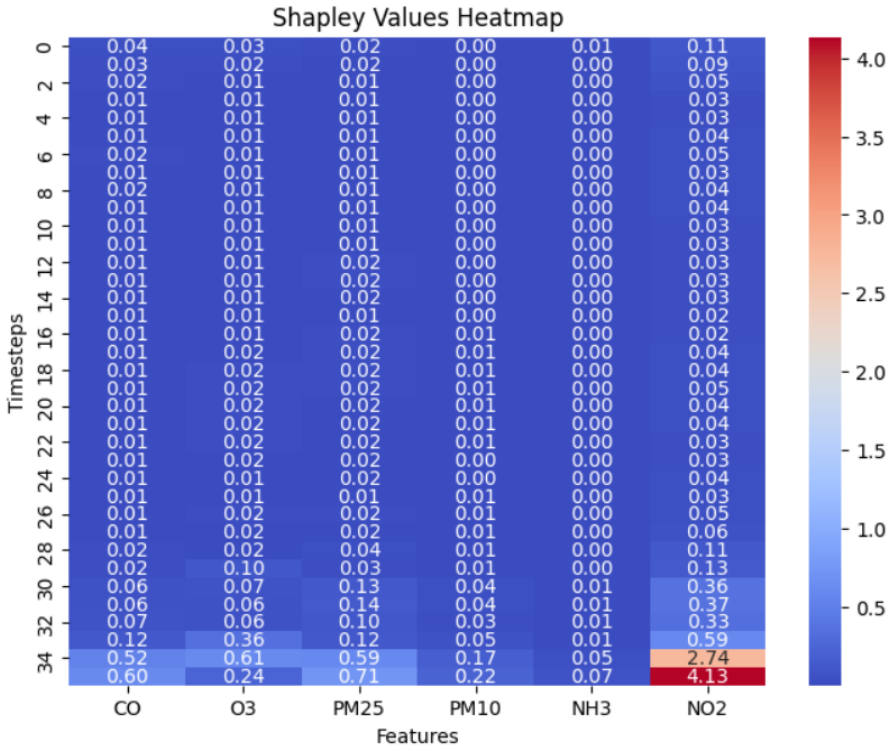


Figura 46: Heatmap da soma dos valores de importância por *timestep*

Assim como na **LSTM** sem *Attention*, o NO_2 contínua a ser a *feature* mais relevante, apresentando valores de shap muito similares aos da Figura 42 nos primeiros *timesteps*. A principal diferença entre os dois *heatmaps* é na importância do $PM_{2.5}$, que tem mais relevância neste modelo do que no anterior. Esta variável é a segunda mais importante na previsão na maioria dos *timesteps*, ao contrário do que acontece no modelo sem atenção, em que esta variável era praticamente descartável. Esta diferença mostra como a camada de atenção altera o treino do modelo e a forma como este aprende relações, eventos e dependências nos dados. Além disso, é posível verificar a importância total de todas as *features* em todos os *timesteps* na Figura 47. Neste modelo, assim como no modelo sem *Attention*, o NO_2 é a *feature* mais relevante, sendo cerca de 5 vezes mais importante do que a *feature* em segundo lugar. Esta posição é ocupada pelo $PM_{2.5}$, no entanto o O_3 e o CO têm valores de importância muito próximos do segundo lugar. Como *features* desprezáveis temos o PM_{10} e o NH_3 com o valor de importância mais baixo.

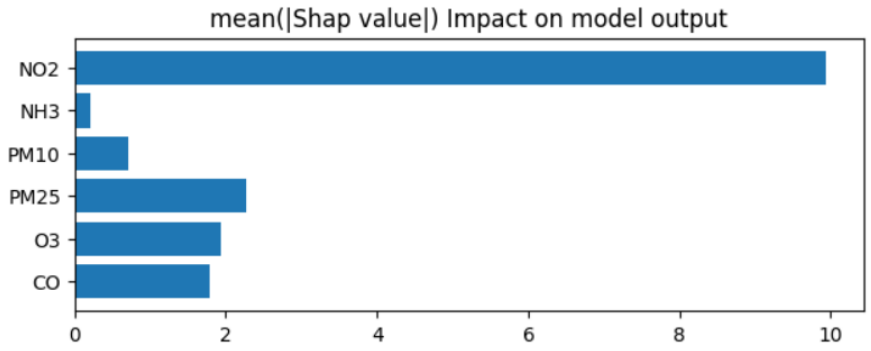


Figura 47: Soma da importância de todos os *timesteps*

Em relação à sazonalidade diária e à importânca, mais uma vez a marca das 24 horas, não parece ter maior relevância para a previsão em relação aos restantes *timesteps* voltando a haver um aumento dos valores de **SHAP** com a proximidade da marca das 36 horas. Na hora anterior à previsão a *feature* com maior relevância permanece o NO_2, com o $PM_{2.5}$ em segundo lugar com um valor 8 vezes menor ao do NO_2. Na marca das 24 horas, há um aumento da importância do $PM_{2.5}$, O_3 e CO em relação ao NO_2, no entanto esta ainda permanece como a *feature* mais importante para a previsão. Na marca de tempo de 36 horas antes, o $PM_{2.5}$ volta a diminuir mas o O_3 e CO permanecem com valores elevados em relação ao NO_2. Em nenhum ponto no tempo o NH_3 e o PM_{10} chegam a ser relevantes para a previsão, enquanto o NO_2 é a *feature* mais importante ao longo de todo o intervalo de tempo.

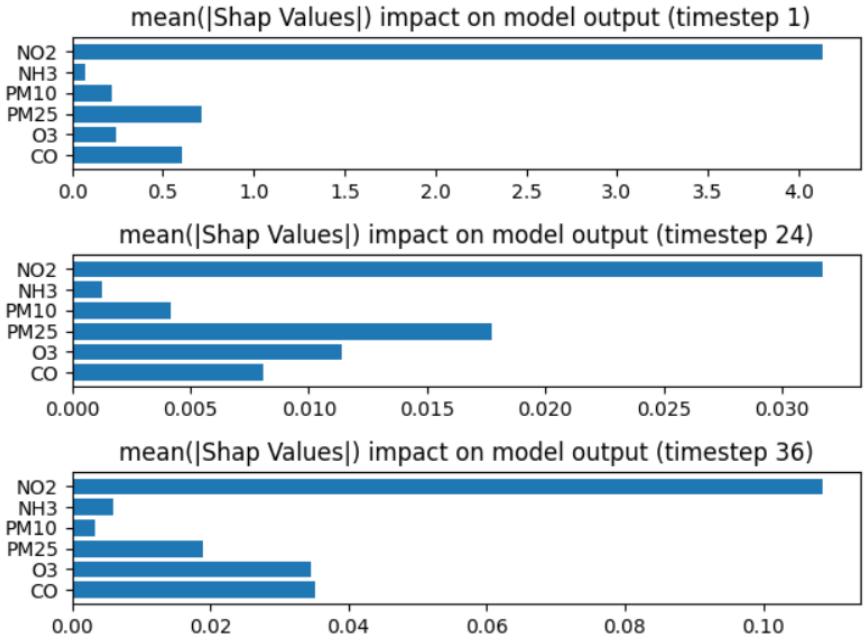


Figura 48: *Shap Values* nos *timesteps* 1, 24 e 36.

5.2.3 Feature Permutation

A última técnica aplicada no âmbito da explicabilidade, foi a *feature permutation*, ou *feature importance permutation*. Esta técnica é agnóstica aos modelos, isto é, pode ser aplicada a qualquer modelo de previsão [130]. O objetivo desta técninca é avaliar o impacto de cada *feature* na previsão, reorganizandp os valores apenas daquela *feature* e mantendo as restantes iguais [130]. O processo pode ser descrito nos seguintes passos:

1. **Previsão de linha de base:** calcular a precisão ou o erro da previsão de linha de base do modelo num conjunto de dados de validação, isto é, fazer uma previsão com dados reais sem qualquer alteração de forma a verificar a performance real. Esta performance é a nossa base para calcular os valores de importância para cada *feature* nos passos seguintes.

2. **Permutação:** para cada *feature*, reorganizar aleatoriamente e individualmente, os valores no conjunto de dados de validação, de forma a quebrar a sua relação com a variável de previsão.

3. **Previsão com dados reorganizados:** utilizar o conjunto de dados de validação modificado com a *feature* reordenada para efetuar previsões utilizando o modelo treinado.

4. **Avaliação do impacto:** medir a queda na precisão da previsão ou o aumento no erro da mesma em comparação com a linha de base. Uma queda significativa indica que a *feature* é importante para as previsões do modelo, e uma queda pequena sugere que é menos relevante.

O Figura 49 demonstra a importância de cada *feature* aquando das previsões dos dados de validação.

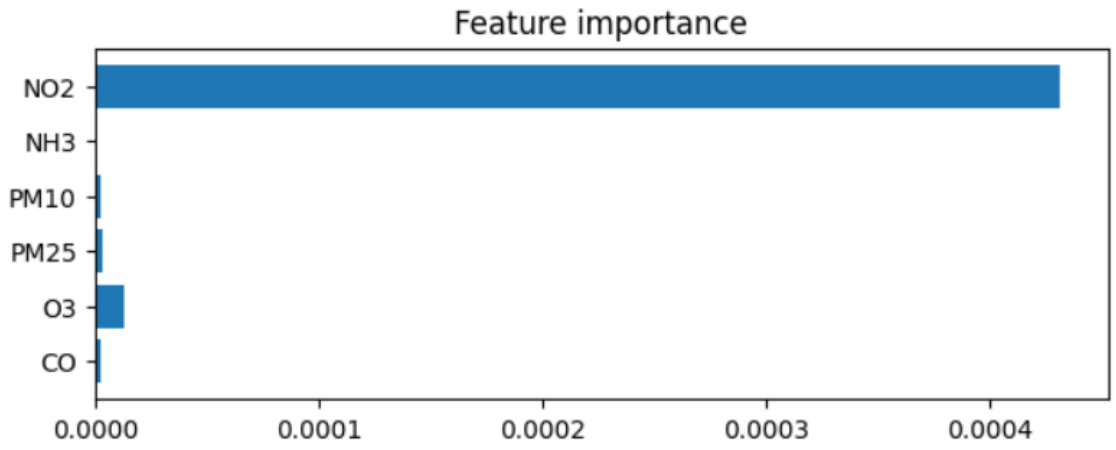


Figura 49: Feature Importance Permutation

Como explicado no passo 4 do processo de cálculo das importâncias, os valores do eixo horizontal são a diferença do erro de previsão sem permutação e com permutação. Quanto maior a diferença, maior a importância da variável. Ordenando as *features* por ordem decrescente de importância, temos em primeiro o NO_2 com uma diferença notável das restantes *features*. Em segundo lugar, temos o O_3, seguido $PM_{2.5}$, CO, PM_{10} e NH_3. Todas as *features* têm valores bastante baixos quando comparados com o NO_2, indicando que essas variáveis podem ser descartadas para as previsões. Apesar de ser uma técnica fácil de implementar e agnóstica aos modelos, é importante referir que os resultados não só estão diretamente relacionados à métrica de performance utilizada, como também dependem da *seed* utilizada no processo de permutação [17].

5.3 Comparação de Resultados

Após a análise dos resultados das várias técnicas aplicadas, é necessário analisar como é que estes se relacionam entre si e se estão de acordo com o que acontece no meio ambiente. A Tabela 13 apresenta as 3 *features* mais importantes pelos valores de **SHAP** no modelo sem e com *Attention* e os valores de *feature permutation*. Os 3 métodos admitem o NO_2 como a mais relevante para a previsão. Já para o segundo lugar o **SHAP** da **LSTM** sem *Attention* e a *feature permutation* consideram o O_3 como mais importante enquanto o **SHAP** aplicado à *attention* considera o O_3 como a terceira mais relevante, tendo o $PM_{2.5}$ como a segunda *feature* que mais interfere no resultado da previsão do NO_2. O **SHAP** no modelo com *attention* e a *feature permutation* têm as mesmas *features* no seu top 3, trocando a ordem do terceiro e segundo lugar. Já quando aplicado ao modelo sem *attention*, o **SHAP**, dá valores de importância mais

altos ao CO do que ao $PM_{2.5}$.

Posição	Shap	Att-Shap	Feature Permutation
1°	NO2	NO2	NO2
2°	O3	PM2.5	O3
3°	CO	O3	PM2.5

Tabela 13: Top 3 *features* mais importantes

Em relação aos *timesteps* mais importantes, ao analisar os pesos de atenção, na geralidade os *timesteps* que marcam as 14 a 18 horas no passado são os mais importantes, sendo apenas o O_3 mais relevante no marco das 36 horas. Já o **SHAP** considera as 6 horas anteriores à previsão como mais relevantes. No entanto, para o **SHAP** também notamos uma ligeira subida na importância das *features* entre as 14 e 18 horas.

Para perceber as previsões, não basta olhar para os valores e resultados obtidos por estas técnicas, é necessário ver como estes se aplicam no caso de estudo e se fazem sentido. Para isso é importante perceber como é que cada um dos gases representados nas *features* é emitido para a atmosfera e como reagem entre si. Os seguintes dados referentes a fontes de emissão e interações das *features* foram retirados do relatório da qualidade do ar na Europa em 2020 pela **European Economic Area (EEA)**.[131]

Começando pelo NO_2, as principais fontes de emissão deste composto são do setor de transporte, agricultura e industria, maioritarimante formado pela combustão de combustíveis fósseis a altas temperaturas [132]. Além disso, o NO_2 também é formado naturalmente quando ocorrem descargas elétricas das nuvens. As mudanças de temperatura drásticas no ar causadas pela descarga fazem as moléculas de nitrogénio e oxigénio presentes na atmosfera reagir e formar NO que mais tarde reage de novo com outras moléculas de oxigénio para formar NO_2 [133]. O CO tem como principais fontes o setor de transporte, industrial e residencial pela combustão incompleta de combustíveis fósseis. $PM_{2.5}$ e PM_{10} são partículas com diâmetros de 2.5 e 10 micrometro, respetivamente. Estas partículas podem ter origem numa vasta gama de fontes, tanto naturais como de origem humana. As fontes incluem emissões de veículos, processos industriais, actividades de construção e fontes naturais como poeiras e incêndios florestais [134]. Estas também podem ser formadas e transformadas através de reações atmosféricas entre o NO_2 e NH_3 na presença de água. A partir destes dados conseguimos perceber o porquê do CO e $PM_{2.5}$ terem uma importância um tanto elevada na previsão do NO_2. Todas as 3 variáveis provêm principalmente das mesmas fontes de emissão. O facto do $PM_{2.5}$ ser considerado mais importante que o CO em 2 das técnicas aplicadas pode ser justificado pelo facto da produção de $PM_{2.5}$ estar ligada às

reações atmosféricas provenientes do NO_2, tornando-as assim variáveis dependentes, enquanto o NO_2 e o CO não reagem diretamente de nenhuma forma. A última variável apresentada na tabela é o O_3, que contrastando com as restante, não tem origem em atividade humana. O O_3 forma-se naturalmente na estratosfera através da interação da radiação ultravioleta com as moléculas de **Oxigénio (O_2)**. No entanto, ao nível do solo a sua formação ocorre de forma diferente. Em vez de moléculas de oxigénio, o calor e radiação causam reações químicas entre os **VOCs** e o NO_2 para criar O_3. Uma vez que o sensor utilizado para fazer as medições está ao nível do solo, as medições do O_3 são maioritariamente influenciadas por esta reação química, fazendo com que neste caso, o O_3 seja quase 100% dependente dos níveis de NO_2 no local e, consequentemente, ser uma variável bastante importante quando queremos fazer previsões do NO_2, como indicado por cada uma das técnicas de explicabilidade aplicadas. Estas reações ocorrem de forma gradual ao longo do dia, verificando-se o pico de concentração de O_3 cerca de 5 horas após o pico de concentração de NO_2 [133].

Após a análise exploratória dos dados no capítulo 3, das métricas obtidas pelos modelos de previsão no capítulo 4 e da compreenssão das interrelações das *features*, podemos aumentar a nossa confiança nos resultados da Tabela 13. Apesar das 3 técnicas darem resultados diferentes, em conjunto ajudam a perceber a tomada de decisão e os valores obtidos pelos modelos.

Capítulo 6

Conclusão

A evolução do **ML**, em particular, do **DL** transformou o panorama da **IA** nos últimos anos. Desde o seu início, como um conceito inovador, até à sua presença generalizada em diversas áreas, com modelos cada vez mais complexos e poderosos. À medida que estes modelos se tornam mais complexos, compreender como tomam decisões, prever o seu comportamento e garantir que os seus resultados estão de acordo com os valores humanos não se tornou apenas uma questão de curiosidade académica, mas uma necessidade crítica em domínios que vão desde os cuidados de saúde às finanças, dos veículos autónomos ao processamento de linguagem natural. Nesta era avançada, a capacidade de compreender a *blackbox* e trazer transparência aos processos de tomada de decisão destes modelos tornou-se um imperativo, e é sobre esta base que esta dissertação é construída.

Esta dissertação, explorou várias abordagens para tratar a questão crítica da explicabilidade e interpretabilidade do modelo no contexto da análise de dados de séries temporais. Os dados em questão dizem respeito a uma série temporal multivariada, em que cada *feature* representa as medições dos níveis de CO, NH_3, O_3, $PM_{2.5}$, PM_{10} e NO_2, emitidos na cidade do Porto. O objetivo da dissertação consistiu em utilizar um modelo **LSTM**, para prever os níveis de NO_2 nas próximas 24 horas, e aplicar explicabilidade para interpretar as previsões. Foram utilizados diferentes técnicas de explicabilidade, desde a abordagem ante-hoc e específica das **ANN** fornecida pela camada de *attention*, até abordagens post-hoc baseadas na importância das *features*, como *shap* e *feature permutation importance*.

Começando com uma exploração estatística do conjunto de dados, foram calculadas métricas descritivas essenciais, como a média, que forneceu uma medida central dos dados, a mediana, que ofereceu uma perspectiva sobre a tendência central robusta, o desvio padrão, que quantificou a dispersão dos dados e *skewness* e *kurtosis*, que revelaram informações sobre a simetria e a forma da distribuição dos dados. A importância da visualização gráfica dos dados foi essencial durante a exploração estatística. Neste sentido, foram concebidos gráficos e histogramas, de forma a representar a distribuição dos dados, facilitando a identificação de padrões, outliers e a compreensão da forma geral da distribuição temporal.

Através dessa análise visual complementou as métricas estatísticas, proporcionando uma compreensão mais completa e intuitiva do comportamento dos dados. Além disso, a análise da correlação com o método de *Spearman* serviu para perceber as relações e filtrar as *features* para a previsão do NO_2. Esta primeira análise estatística lançou as bases para a análise temporal subsequente.

De modo a obter uma compreensão mais detalhada das variações inesperadas nos dados temporais, foi feita uma deteção de anomalias, bem como um estudo das componentes principais que compoem as séries temporais. A aplicação da *Matrix Profile* permitiu identificar padrões anómalos e eventos fora do comum, contribuindo para a robustez da análise ao destacar pontos de interesse que poderiam influenciar significativamente a interpretação dos resultados. Quanto à tendência, esta revelou a estabilidade dos dados em torno de um valor médio, enquanto a identificação da sazonalidade permitiu entender os padrões recorrentes ao longo do tempo, como variações sazonais diárias e anuais. A identificação de padrões de autocorrelação através dos *lags* nos gráficos **ACF** ajudou a compreender como as observações passadas influenciam as observações futuras. Estas observações adicionais sobre a dependência temporal das *features*, contribuiram para a modelação e interpretação mais precisa dos resultados.

O treino de modelos **LSTM** e a meticulosa afinação de parâmetros foram fundamentais para os objectivos desta dissertação. Ao afinar estes modelos, procurou-se melhorar a precisão da previsão e, simultaneamente, ao adicionar *attention*, explorar a forma como a arquitetura do modelo e os hiperparâmetros influenciam a explicabilidade do modelo.

A abordagem de explicabilidade *ante-hoc* adotada não foi apenas específica, mas também global, proporcionando uma visão holística e abrangente da interpretabilidade do modelo desde o início do processo de construção. A explicabilidade *ante-hoc* global, fornecida pela *attention*, visou compreender não apenas as interações locais entre as diferentes partes do input e as previsões finais em instâncias individuais, mas também entender os padrões gerais e influências globais ao longo do conjunto de dados. Ao extrair informações da camada de *attention* de forma global, foi possível identificar quais as partes do *input* eram consideradas mais relevantes ou influentes durante a tomada de decisão do modelo. Esta análise considerou as relações e padrões que emergem em uma escala mais ampla, permitindo uma compreensão mais profunda, não apenas das decisões do modelo em casos específicos, mas também da consistência e das tendências gerais em todo o conjunto de dados.

As técnicas *post-hoc*, por outro lado, permitiram inspecionar e interpretar as decisões do modelo após a sua implementação. O **SHAP** forneceu uma decomposição aditiva das previsões, destacando, não só a contribuição global de cada *feature*, mas também como cada *timestep* específico contribuiu para a variação nas previsões. Essa abordagem granular ajudou a compreender melhor a dinâmica temporal

das decisões do modelo num contexto local e a compreensão de como cada variável influenciou a saída do modelo em casos específicos. A *Feature Permutation Importance* foi apenas utilizada para avaliar a contribuição de cada *feature* sem avaliação da contribuição do *timestep*. Esta decisão foi tomada após se verificar que a diferença do cálculo e variação da importância de cada *timestep* para a previsão final estava na casa dos 10^6.

Foram utilizados gráficos de barras e *heatmaps* de forma a visualizar e melhor interpretar os resultados obtidos por cada uma das técnicas. Após análise, fez-se uma comparação dos resultados de todas as técnicas e foi contruído o top 3 das *features* mais relevantes em cada abordagem. O NO_2 liderou em todas as abordagens, sendo classificada como a *feature* mais importante para a previsão do seu próprio valor futuro. Em segundo lugar, ficou o O_3 nas técnicas *post-hoc* **SHAP** e *feature permutation*, enquanto na abordagem que inclui a técnica *ante-hoc*, esta posição é ocupada pelo $PM_{2.5}$. A última posição do top 3 foi a mais divergente, uma vez que cada abordagem indicou uma *feature* diferente. No entanto, a terceira posição de algumas abordagens corresponde à segunda posição de outras abordagens, com exceção do CO que aparece pela primeira vez no **SHAP**. A validação destes resultados foi feita, não só pelo estudo estatístico e temporal efetuado no conjunto de dados, mas também pela estudo das relações e transformações físicas e químicas que ocorrem entre estes gases na natureza. Este conhecimento sobre o domínio foi crucial para consolidar e validar os resultados finais.

A aplicação de diversos métodos de explicabilidade e a análise dos seus resultados em diferentes cenários reforçaram a robustez das conclusões, validando não só a consistência das explicações, como também permitindo uma compreensão mais abrangente do comportamento do modelo em condições variáveis. Esta robustez apenas é afirmada pela convergência dos resultados das várias técnicas, que se verificou não só nas *features*, mas principalmente na determinação dos *timesteps* mais relevantes para as previsões. Essa convergência nos dois cenários (com e sem *attention*) indicou uma estabilidade nas características críticas identificadas, independentemente da presença ou ausência da camada de *attention*, reforçando a validade das conclusões e proporcionando não só uma base sólida para a interpretação das decisões do modelo em diferentes contextos, mas também um aumento da confiança sobre o comportamento do modelo. É essencial reconhecer que a fiabilidade dos resultados da explicabilidade está intrinsecamente ligada ao contexto geral dos dados. Ao considerarmos as nuances do conjunto de dados, incluindo a qualidade, a quantidade e o conhecimento específico do domínio, conseguimos obter interpretações mais significativas e precisas. A compreensão dos aspectos específicos do domínio dos dados aumentou a credibilidade e a relevância dos resultados de explicabilidade, garantindo que não só eram válidos, mas também acionáveis em aplicações práticas.

Com o objetivo de fortalecer as explicações das previsões na temática da qualidade do ar, existem algumas medidas que podem ser consideradas para trabalho futuro. Em primeiro lugar, seria interessante arranjar dados relativos à temperatura, humidade e pluviosidade para a mesma região no mesmo intervalo de tempo. Uma vez que estas variáveis têm influência direta e indireta nas emissões e interações das componentes estudadas, estas novas relações poderiam fortalecer o poder de previsão do modelo, aumentando a sua capacidade de detetar, padrões, sazonalidade e tendências. Além disso, ajudariam a fortalecer e validar as explicações com conhecimento no domínio. Em segundo lugar, implementar um modelo de deteção de anomalias para o tratamento de *outliers*. O processo de tratar *outliers* de uma série temporal é mais complexo devido às variações estatísticas que ocorrem ao longo do tempo. Estas variações, devem-se principalmente às caraterísticas únicas de tendência e sazonalidade que este tipo de dados apresenta. Deste modo, é necessário um modelo adequado que detete *outliers* conforme estes padrões e assim tratá-los corretamente. Por fim, poderiam ser implementados outros tipos de técnicas de explicabilidade, como camadas de *attention* que se focam nas dependências temporais dos dados.

Bibliografia

[1] Xiaomin Mou. *Artificial Intelligence: Investment Trends and Selected Industry Uses*. EMCompass 71. http://hdl.handle.net/10986/32652. International Finance Corporation, 2019.

[2] Wojciech Samek e Klaus Robert Müller. *Towards Explainable Artificial Intelligence*. Vol. 11700 LNCS. Springer Verlag, 2019, pp. 5–22. DOI: 10.1007/978-3-030-28954-6_1.

[3] Andreas Holzinger et al. «Current advances, trends and challenges of machine learning and knowledge extraction: From machine learning to explainable AI». Em: vol. 11015 LNCS. Springer Verlag, 2018, pp. 1–8. ISBN: 9783319997391. DOI: 10.1007/978-3-319-99740-7_1.

[4] Raymond Perrault et al. «2019 annual report ar intelligence index Project Manager and Report Editor-in-Chief Steering Committee Steering Commitee Introduction Report Highlights Acknowledgements». Em: (2019).

[5] «perceptron». Em: ().

[6] Ajay Shrestha e Ausif Mahmood. *Review of deep learning algorithms and architectures*. 2019. DOI: 10.1109/ACCESS.2019.2912200.

[7] James Hendler. «A L e t t e r f r o m t h e E d i t o r We'd Like to Hear from You Avoiding Another AI Winter». Em: (2008). URL: www.computer.org/intelligent/author.htm..

[8] Kunal Nagpal et al. «Development and validation of a deep learning algorithm for improving Gleason scoring of prostate cancer». Em: *npj Digital Medicine* 2 (1 dez. de 2019). ISSN: 23986352. DOI: 10.1038/s41746-019-0112-2.

[9] Mark Sendak et al. «"The human body is a black box": Supporting clinical decision-making with deep learning». Em: Association for Computing Machinery, Inc, jan. de 2020, pp. 99–109. ISBN: 9781450369367. DOI: 10.1145/3351095.3372827.

[10] Ramy Hussein et al. «Epileptic Seizure Detection: A Deep Learning Approach». Em: (mar. de 2018). URL: http://arxiv.org/abs/1803.09848.

[11] Abdullah H. Al-Nefaie e Theyazn H. H. Aldhyani. «Predicting Close Price in Emerging Saudi Stock Exchange: Time Series Models». Em: *Electronics* 11.21 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11213443. URL: https://www.mdpi.com/2079-9292/11/21/3443.

[12] Riccardo Guidotti et al. «A Survey Of Methods For Explaining Black Box Models». Em: (fev. de 2018). URL: http://arxiv.org/abs/1802.01933.

[13] Jenna Burrell. «How the machine 'thinks': Understanding opacity in machine learning algorithms». Em: *Big Data and Society* 3 (1 jan. de 2016). ISSN: 20539517. DOI: 10.1177/2053951715622512.

[14] Carlos Zednik. «Solving the Black Box Problem: A Normative Framework for Explainable Artificial Intelligence». Em: (). URL: http://carloszednik.net/.

[15] Marco Tulio Ribeiro, Sameer Singh e Carlos Guestrin. «"Why should i trust you?"Explaining the predictions of any classifier». Em: vol. 13-17-August-2016. Association for Computing Machinery, ago. de 2016, pp. 1135–1144. ISBN: 9781450342322. DOI: 10.1145/2939672.2939778.

[16] Plamen P. Angelov et al. «Explainable artificial intelligence: an analytical review». Em: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11 (5 set. de 2021). ISSN: 19424795. DOI: 10.1002/widm.1424.

[17] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2ª ed. 2022. URL: https://christophm.github.io/interpretable-ml-book.

[18] «Model Explainability with AWS Artificial Intelligence and Machine Learning Solutions». Em: ().

[19] Michael Heider et al. «Assessing Model Requirements for Explainable AI: A Template and Exemplary Case Study». Em: *Artificial Life* (ago. de 2023), pp. 1–19. ISSN: 1064-5462. DOI: 10.1162/artl_a_00414. eprint: https://direct.mit.edu/artl/article-pdf/doi/10.1162/artl_a_00414/2155678/artl_a_00414.pdf. URL: https://doi.org/10.1162/artl_a_00414.

[20] Charles R. Berger e Richard J. Calabrese. «Some Explorations in Initial Interaction and Beyond: Toward a Developmental Theory of Interpersonal Communication». Em: *Human Communication Research* 1.2 (mar. de 2006), pp. 99–112. ISSN: 0360-3989. DOI: 10.1111/j.1468-2958.1975.tb00258.x. eprint: https://academic.oup.com/hcr/article-pdf/1/2/99/22344108/jhumcom0099.pdf. URL: https://doi.org/10.1111/j.1468-2958.1975.tb00258.x.

[21] Tim Miller. «Explanation in Artificial Intelligence: Insights from the Social Sciences». Em: (jun. de 2017). URL: http://arxiv.org/abs/1706.07269.

[22] Juri Zach. «Interpretability of deep neural networks». Em: ().

[23] Finale Doshi-Velez e Been Kim. «Towards A Rigorous Science of Interpretable Machine Learning». Em: (fev. de 2017). URL: http://arxiv.org/abs/1702.08608.

[24] James Woodward e Lauren Ross. «Scientific Explanation». Em: *The Stanford Encyclopedia of Philosophy*. Ed. por Edward N. Zalta. Summer 2021. Metaphysics Research Lab, Stanford University, 2021.

[25] Leilani H. Gilpin et al. «Explaining Explanations: An Overview of Interpretability of Machine Learning». Em: (mai. de 2018). URL: http://arxiv.org/abs/1806.00069.

[26] Henna Tiensuu et al. «Evidence-Based and Explainable Smart Decision Support for Quality Improvement in Stainless Steel Manufacturing». Em: *Applied Sciences* 11.22 (2021). ISSN: 2076-3417. DOI: 10.3390/app112210897. URL: https://www.mdpi.com/2076-3417/11/22/10897.

[27] Rich Caruana et al. «Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-Day Readmission». Em: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '15. Sydney, NSW, Australia: Association for Computing Machinery, 2015, pp. 1721–1730. ISBN: 9781450336642. DOI: 10.1145/2783258.2788613. URL: https://doi.org/10.1145/2783258.2788613.

[28] Jun Wang et al. «Safety of Autonomous Vehicles». Em: *Journal of Advanced Transportation* 2020 (2020). ISSN: 20423195. DOI: 10.1155/2020/8867757.

[29] Tanha Talaviya et al. «Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides». Em: *Artificial Intelligence in Agriculture* 4 (jan. de 2020), pp. 58–73. ISSN: 25897217. DOI: 10.1016/j.aiia.2020.04.002.

[30] *I (Legislative acts) REGULATIONS REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)*.

[31] Been Kim et al. «Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)». Em: (nov. de 2017). URL: http://arxiv.org/abs/1711.11279.

[32] Mark Ibrahim et al. «Global Explanations of Neural Networks: Mapping the Landscape of Predictions». Em: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '19. Honolulu, HI, USA: Association for Computing Machinery, 2019, pp. 279–287. ISBN: 9781450363242. DOI: 10.1145/3306618.3314230. URL: https://doi.org/10.1145/3306618.3314230.

[33] Gregory Plumb, Denali Molitor e Ameet Talwalkar. *Model Agnostic Supervised Local Explanations*. URL: https://github.com/GDPlumb/MAPLE..

[34] Tejaswini Pedapati et al. «Learning Global Transparent Models consistent with Local Contrastive Explanations». Em: *Advances in Neural Information Processing Systems*. Ed. por H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 3592–3602. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/24aef8cb3281a2422a59b51659f1ad2e-Paper.pdf.

[35] Paul-Christian Bürkner e Emmanuel Charpentier. «Modelling monotonic effects of ordinal predictors in Bayesian regression models». Em: *British Journal of Mathematical and Statistical Psychology* 73.3 (2020), pp. 420–451. DOI: https://doi.org/10.1111/bmsp.12195. eprint: https://bpspsychub.onlinelibrary.wiley.com/doi/pdf/10.1111/bmsp.12195. URL: https://bpspsychub.onlinelibrary.wiley.com/doi/abs/10.1111/bmsp.12195.

[36] Amanpreet Singh, Narina Thakur e Aakanksha Sharma. «A review of supervised machine learning algorithms». Em: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. 2016, pp. 1310–1315.

[37] Joseph D. Janizek, Pascal Sturmfels e Su-In Lee. «Explaining explanations: axiomatic feature interactions for deep networks». Em: *J. Mach. Learn. Res.* 22.1 (2021). ISSN: 1532-4435.

[38] Kim F. Nimon e Frederick L. Oswald. «Understanding the Results of Multiple Linear Regression: Beyond Standardized Regression Coefficients». Em: *Organizational Research Methods* 16.4 (2013), pp. 650–674. DOI: 10.1177/1094428113493929. eprint: https://doi.org/10.1177/1094428113493929. URL: https://doi.org/10.1177/1094428113493929.

[39] Amy McGovern et al. «Making the Black Box More Transparent: Understanding the Physical Implications of Machine Learning». Em: *Bulletin of the American Meteorological Society* 100.11 (2019), pp. 2175 –2199. DOI: 10.1175/BAMS-D-18-0195.1. URL: https://journals.ametsoc.org/view/journals/bams/100/11/bams-d-18-0195.1.xml.

[40] Harouna Soumare, Alia Benkahla e Nabil Gmati. «Deep learning regularization techniques to genomics data». Em: *Array* 11 (set. de 2021), p. 100068. ISSN: 25900056. DOI: 10.1016/j.array.2021.100068.

[41] Prasad Chalasani et al. «Concise Explanations of Neural Networks using Adversarial Training». Em: *Proceedings of the 37th International Conference on Machine Learning*. Ed. por Hal Daumé III e Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1383–1391. URL: https://proceedings.mlr.press/v119/chalasani20a.html.

[42] Gao Huang et al. «Deep Networks with Stochastic Depth». Em: (mar. de 2016). URL: http://arxiv.org/abs/1603.09382.

[43] Mukund Sundararajan, Ankur Taly e Qiqi Yan. «Axiomatic Attribution for Deep Networks». Em: (mar. de 2017). URL: http://arxiv.org/abs/1703.01365.

[44] Warren Freeborough e Terence van Zyl. «Investigating Explainability Methods in Recurrent Neural Network Architectures for Financial Time Series Data». Em: *Applied Sciences (Switzerland)* 12 (3 fev. de 2022). ISSN: 20763417. DOI: 10.3390/app12031427.

[45] Marco Tulio Ribeiro, Sameer Singh e Carlos Guestrin. *Model-Agnostic Interpretability of Machine Learning*. 2016. arXiv: 1606.05386 [stat.ML].

[46] Geoffrey Hinton, Oriol Vinyals e Jeff Dean. *Distilling the Knowledge in a Neural Network*. 2015. arXiv: 1503.02531 [stat.ML].

[47] Osbert Bastani, Carolyn Kim e Hamsa Bastani. *Interpreting Blackbox Models via Model Extraction*. 2019. arXiv: 1705.08504 [cs.LG].

[48] Morteza Mashayekhi e Robin Gras. «Rule Extraction from Decision Trees Ensembles: New Algorithms Based on Heuristic Search and Sparse Group Lasso Methods». Em: *International Journal of Information Technology & Decision Making* 16.06 (2017), pp. 1707–1727. DOI: 10.1142/S0219622017500055. eprint: https://doi.org/10.1142/S0219622017500055. URL: https://doi.org/10.1142/S0219622017500055.

[49] Josue Obregon e Jae Yoon Jung. «RuleCOSI+: Rule extraction for interpreting classification tree ensembles». Em: *Information Fusion* 89 (jan. de 2023), pp. 355–381. ISSN: 15662535. DOI: 10.1016/j.inffus.2022.08.021.

[50] Joseph Manu. *Interpretability Part 3: Opening the black box with lime and shap*. 2019. URL: https://www.kdnuggets.com/2019/12/interpretability-part-3-lime-shap.html.

[51] Scott M Lundberg, Paul G Allen e Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. URL: https://github.com/slundberg/shap.

[52] Rami Mochaourab et al. *Post-hoc Explainability for Time Series Classication: Toward a Signal Processing Perspective*.

[53] Dumitru Erhan et al. *Visualizing Higher-Layer Features of a Deep Network*. Technical Report 1341. P.O. Box 6128, Downtown Branch, Montreal, H3C 3J7, QC, Canada: Département d'Informatique et Recherche Opérationnelle, Université de Montréal, 2009.

[54] Yang Zhang et al. «An Adversarial Framework for Generating Unseen Images by Activation Maximization». Em: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.3 (2022), pp. 3371–3379. DOI: 10.1609/aaai.v36i3.20247. URL: https://ojs.aaai.org/index.php/AAAI/article/view/20247.

[55] Grégoire Montavon et al. «Explaining NonLinear Classification Decisions with Deep Taylor Decomposition». Em: (dez. de 2015). DOI: 10.1016/j.patcog.2016.11.008. URL: http://arxiv.org/abs/1512.02479http://dx.doi.org/10.1016/j.patcog.2016.11.008.

[56] Maximilian Kohlbrenner et al. «Towards Best Practice in Explaining Neural Network Decisions with LRP». Em: (out. de 2019). URL: http://arxiv.org/abs/1910.09840.

[57] Sebastian Bach et al. *Analyzing Classifiers: Fisher Vectors and Deep Neural Networks*. 2015. arXiv: 1512.00172 [cs.CV].

[58] Ihsan Ullah et al. *Explaining Deep Learning Models for Structured Data using Layer-Wise Relevance Propagation*. URL: https://simple.wikipedia.org/wiki/Clever_Hans.

[59] Guannan Li et al. «Interpretation and explanation of convolutional neural network-based fault diagnosis model at the feature-level for building energy systems». Em: *Energy and Buildings* 295 (2023), p. 113326. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2023.113326. URL: https://www.sciencedirect.com/science/article/pii/S037877882300556X.

[60] Philippe Esling e Carlos Agon. «Time-Series Data Mining». Em: *ACM Comput. Surv.* 45.1 (2012). ISSN: 0360-0300. DOI: 10.1145/2379776.2379788. URL: https://doi.org/10.1145/2379776.2379788.

[61] Wolfgang Aigner et al. «Visual Methods for Analyzing Time-Oriented Data». Em: *IEEE Transactions on Visualization and Computer Graphics* 14.1 (2008), pp. 47–60. DOI: 10.1109/TVCG.2007.70415.

[62] Cheryl L. Jennings Douglas C. Montgomery e Murat Kulahci. *Introduction to Time Series Analysis and Forecasting, Second Edition.* 2015.

[63] Rob J Hyndman e George Athanasopoulos. *Forecasting: principles and practice, 3rd edition,* 2021.

[64] Maximilian Zellner et al. «A survey of human judgement and quantitative forecasting methods». Em: *R. Soc. Open Sci.* (2021). DOI: 10.1098/rsos.201187. URL: http://doi.org/10.1098/rsos.201187.

[65] J.Morley English e Gerard L. Kernan. «The prediction of air travel and aircraft technology to the year 2000 using the Delphi method». Em: *Transportation Research* 10.1 (1976), pp. 1–8. ISSN: 0041-1647. DOI: https://doi.org/10.1016/0041-1647(76)90094-0. URL: https://www.sciencedirect.com/science/article/pii/0041164776900940.

[66] Andrew Harvey. «Chapter 7 Forecasting with Unobserved Components Time Series Models». Em: ed. por G. Elliott, C.W.J. Granger e A. Timmermann. Vol. 1. Handbook of Economic Forecasting. Elsevier, 2006, pp. 327–412. DOI: https://doi.org/10.1016/S1574-0706(05)01007-4. URL: https://www.sciencedirect.com/science/article/pii/S1574070605010074.

[67] Lajos Horváth, Piotr Kokoszka e Gregory Rice. «Testing stationarity of functional time series». Em: *Journal of Econometrics* 179.1 (2014), pp. 66–82. ISSN: 0304-4076. DOI: https://doi.org/10.1016/j.jeconom.2013.11.002. URL: https://www.sciencedirect.com/science/article/pii/S0304407613002327.

[68] Don M. Miller e Dan Williams. «Shrinkage estimators of time series seasonal factors and their effect on forecasting accuracy». Em: *International Journal of Forecasting* 19.4 (2003), pp. 669–684. ISSN: 0169-2070. DOI: https://doi.org/10.1016/S0169-2070(02)00077-8. URL: https://www.sciencedirect.com/science/article/pii/S0169207002000778.

[69] Rizwan Mushtaq. *TESTING TIME SERIES DATA FOR STATIONARITY*. URL: http://ssrn.com/abstract=1911068Electroniccopyavailableat:https://ssrn.com/abstract=1911068.

[70] Dongin Lcc e Peter Schmidt. *On the power of the KPSS test of stationarity against fractionally-integrated alternatives*. 1996, pp. 285–302.

[71] Jin Xiu e Yao Jin. «Empirical study of ARFIMA model based on fractional differencing». Em: *Physica A: Statistical Mechanics and its Applications* 377 (1 abr. de 2007), pp. 138–154. ISSN: 03784371. DOI: 10.1016/j.physa.2006.11.030.

[72] Kulwinder Singh Parmar e Rashmi Bhardwaj. «Statistical, time series, and fractal analysis of full stretch of river Yamuna (India) for water quality management». Em: *Environmental Science and Pollution Research* 22.1 (2015), pp. 397–414. ISSN: 1614-7499. DOI: 10.1007/s11356-014-3346-1. URL: https://doi.org/10.1007/s11356-014-3346-1.

[73] Agus Suharsono, Auliya Aziza e Wara Pramesti. «Comparison of Vector Autoregressive (VAR) and Vector Error Correction Models (VECM) for Index of ASEAN Stock Price». Em: *International Conference and Workshop on Mathematical Analysis and its Applications (ICWOMAA 2017)*. Vol. 1913. AIP Conf. Proc. AIP Publishing. 2017, pp. 020032–1–020032–9. ISBN: 978-0-7354-1605-5. DOI: 10.1063/1.5016666.

[74] G Peter Zhang. *Time series forecasting using a hybrid ARIMA and neural network model*. 2003, pp. 159–175. URL: www.elsevier.com/locate/neucom.

[75] Jan G. De Gooijer e Rob J. Hyndman. «25 years of time series forecasting». Em: *International Journal of Forecasting* 22 (3 2006), pp. 443–473. ISSN: 01692070. DOI: 10.1016/j.ijforecast.2006.01.001.

[76] Jatinder Kaur, Kulwinder Singh Parmar e Sarbjit Singh. «Autoregressive Models in Environmental Forecasting Time Series: A Theoretical and Application Review». Em: *Environmental Science and Pollution Research* 30.8 (2023), pp. 19617–19641. ISSN: 1614-7499. DOI: 10.1007/s11356-023-25148-9. URL: https://doi.org/10.1007/s11356-023-25148-9.

[77] Badi H. Baltagi e Qi Li. «Testing AR(1) against MA(1) disturbances in an error component model». Em: *Journal of Econometrics* 68.1 (1995), pp. 133–151. ISSN: 0304-4076. DOI: https://doi.org/10.1016/0304-4076(94)01646-H. URL: https://www.sciencedirect.com/science/article/pii/030440769401646H.

[78] A. Meyler, G. Kenny e T. Quinn. *Forecasting Irish Inflation using ARIMA Models*. http://mpra.ub.uni-muenchen.de/11359/. MPRA Paper No. 11359, posted 3. November 2008 14:34 UTC. 1998.

[79] Charles C. Holt. «Forecasting seasonals and trends by exponentially weighted moving averages». Em: *International Journal of Forecasting* 20.1 (2004), pp. 5–10. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2003.09.015. URL: https://www.sciencedirect.com/science/article/pii/S0169207003001134.

[80] *Springer Series in Statistics Forecasting with Exponential Smoothing The State Space Approach*.

[81] Baki Billah et al. «Exponential smoothing model selection for forecasting». Em: *International Journal of Forecasting* 22.2 (2006), pp. 239–247. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2005.08.002. URL: https://www.sciencedirect.com/science/article/pii/S016920700500107X.

[82] Everette S. Gardner. «Exponential smoothing: The state of the art—Part II». Em: *International Journal of Forecasting* 22.4 (2006), pp. 637–666. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2006.03.005. URL: https://www.sciencedirect.com/science/article/pii/S0169207006000392.

[83] Prajakta S Kalekar. *Time series Forecasting using Holt-Winters Exponential Smoothing*. 2004.

[84] Guckan YAPAR et al. «Modified Holt's linear trend method». Em: *Hacettepe Journal of Mathematics and Statistics* 47.5 (2018), pp. 1394–1403.

[85] Phillip G. Gould et al. «Forecasting time series with multiple seasonal patterns». Em: *European Journal of Operational Research* 191.1 (2008), pp. 207–222. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2007.08.024. URL: https://www.sciencedirect.com/science/article/pii/S0377221707008740.

[86] André Altmann et al. «Permutation importance: a corrected feature importance measure». Em: *Bioinformatics* 26.10 (abr. de 2010), pp. 1340–1347. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btq134. eprint: https://academic.oup.com/bioinformatics/article-pdf/26/10/1340/48851160/bioinformatics_26_10_1340.pdf. URL: https://doi.org/10.1093/bioinformatics/btq134.

[87] Pantelis Linardatos, Vasilis Papastefanopoulos e Sotiris Kotsiantis. «Explainable AI: A Review of Machine Learning Interpretability Methods». Em: *Entropy* 23.1 (2021). ISSN: 1099-4300. DOI: 10.3390/e23010018. URL: https://www.mdpi.com/1099-4300/23/1/18.

[88] Hiromasa Kaneko. «Cross-validated permutation feature importance considering correlation between features». Em: *Analytical Science Advances* 3.9-10 (2022), pp. 278–287. DOI: https://doi.org/10.1002/ansa.202200018. eprint: https://chemistry-europe.onlinelibrary.wiley.com/doi/pdf/10.1002/ansa.202200018. URL: https://chemistry-europe.onlinelibrary.wiley.com/doi/abs/10.1002/ansa.202200018.

[89] Thomas Rojat et al. «Explainable Artificial Intelligence (XAI) on TimeSeries Data: A Survey». Em: (abr. de 2021). URL: http://arxiv.org/abs/2104.00950.

[90] Yifan Hao e Huiping Cao. «A New Attention Mechanism to Classify Multivariate Time Series». Em: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (). DOI: 10.24963/ijcai.2020/277. URL: https://par.nsf.gov/biblio/10177556.

[91] Ashish Vaswani et al. «Attention Is All You Need». Em: (jun. de 2017). URL: http://arxiv.org/abs/1706.03762.

[92] Shun-Yao Shih, Fan-Keng Sun e Hung yi Lee. «Temporal pattern attention for multivariate time series forecasting». Em: *Machine Learning* 108.8 (2019), pp. 1421–1441. ISSN: 1573-0565. DOI: 10.1007/s10994-019-05815-0. URL: https://doi.org/10.1007/s10994-019-05815-0.

[93] Jun Hu e Wendong Zheng. «Multistage attention network for multivariate time series prediction». Em: *Neurocomputing* 383 (2020), pp. 122–137. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.11.060. URL: https://www.sciencedirect.com/science/article/pii/S0925231219316625.

[94] Rohit Saluja et al. «Towards a Rigorous Evaluation of Explainability for Multivariate Time Series». Em: (abr. de 2021). URL: http://arxiv.org/abs/2104.04075.

[95] Georgios Chalkiadakis, Edith Elkind e Michael Wooldridge. «Cooperative game theory: Basic concepts and computational challenges». Em: *IEEE Intelligent Systems* 27 (3 2012), pp. 86–90. ISSN: 15411672. DOI: 10.1109/MIS.2012.47.

[96] Ziqi Li. «Extracting spatial effects from machine learning model using local interpretation method: An example of SHAP and XGBoost». Em: *Computers, Environment and Urban Systems* 96 (2022), p. 101845. ISSN: 0198-9715. DOI: https://doi.org/10.1016/j.compenvurbsys.2022.101845. URL: https://www.sciencedirect.com/science/article/pii/S0198971522000898.

[97] Helena Aito. «Explainability of Time Series Models». Master of Engineering - Big Data Analytics. Arcada University of Applied Sciences, 2022.

[98] Jan Koutnik et al. «A Clockwork RNN». Em: *Proceedings of the 31st International Conference on Machine Learning*. Ed. por Eric P. Xing e Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 2014, pp. 1863–1871. URL: https://proceedings.mlr.press/v32/koutnik14.html.

[99] Hansika Hewamalage, Christoph Bergmeir e Kasun Bandara. «Recurrent Neural Networks for Time Series Forecasting: Current status and future directions». Em: *International Journal of Forecasting* 37.1 (2021), pp. 388–427. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2020.06.008. URL: https://www.sciencedirect.com/science/article/pii/S0169207020300996.

[100] Yao Ming et al. «Understanding Hidden Memories of Recurrent Neural Networks». Em: *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 2017, pp. 13–24. DOI: 10.1109/VAST.2017.8585721.

[101] Phong Le e Willem Zuidema. *Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs*. 2016. arXiv: 1603.00423 [cs.AI].

[102] Jingyang Wang et al. «NGCU: A New RNN Model for Time-Series Data Prediction». Em: *Big Data Research* 27 (2022), p. 100296. ISSN: 2214-5796. DOI: https://doi.org/10.1016/j.bdr.2021.100296. URL: https://www.sciencedirect.com/science/article/pii/S2214579621001131.

[103] Yao Ming et al. «Understanding Hidden Memories of Recurrent Neural Networks». Em: *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 2017, pp. 13–24. DOI: 10.1109/VAST.2017.8585721.

[104] Pedro Oliveira et al. «Forecasting Energy Consumption of Wastewater Treatment Plants with a Transfer Learning Approach for Sustainable Cities». Em: *Electronics* 10.10 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10101149. URL: https://www.mdpi.com/2079-9292/10/10/1149.

[105] Rajib Rana. *Gated Recurrent Unit (GRU) for Emotion Classification from Noisy Speech*. 2016. arXiv: 1612.07778 [cs.HC].

[106] Gang Li et al. «Anomaly-PTG: A Time Series Data-Anomaly-Detection Transformer Framework in Multiple Scenarios». Em: *Electronics* 11.23 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11233955. URL: https://www.mdpi.com/2079-9292/11/23/3955.

[107] Bruno Fernandes et al. «Long Short-Term Memory Networks for Traffic Flow Forecasting: Exploring Input Variables, Time Frames and Multi-Step Approaches». Em: *Informatica* 31.4 (2020), pp. 723–749. ISSN: 1822-8844. DOI: 10.15388/20-INFOR431.

[108] Sepp Hochreiter e Jürgen Schmidhuber. «Long Short-Term Memory». Em: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[109] Zahra Karevan e Johan A.K. Suykens. «Transductive LSTM for time-series prediction: An application to weather forecasting». Em: *Neural Networks* 125 (2020), pp. 1–9. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2019.12.030. URL: https://www.sciencedirect.com/science/article/pii/S0893608020300010.

[110] Seung Hong Choi Kyu Sung Choi e Bumseok Jeong. «Prediction of IDH genotype in gliomas with dynamic susceptibility contrast perfusion MR imaging using an explainable recurrent neural network». Em: (2019).

[111] Aya Abdelsalam Ismail et al. *Benchmarking Deep Learning Interpretability in Time Series Predictions*. URL: https://github.com/ayaabdelsalam91/TS-Interpretability-Benchmark.

[112] Hae-Young Kim. «Statistical notes for clinical researchers: Assessing normal distribution (2) using skewness and kurtosis». Em: *Restorative dentistry endodontics* 38 (fev. de 2013), pp. 52–54. DOI: 10.5395/rde.2013.38.1.52.

[113] World Health Organization. *WHO Global Air Quality Guidelines. Particulate Matter (PM 2.5 and PM 10), Ozone, Nitrogen Dioxide, Sulfur Dioxide and Carbon Monoxide*. Licence: CC BY-NC-SA 3.0 IGO. Geneva: World Health Organization, 2021.

[114] OpenWeather. *OpenWeatherMap Air Pollution API*. URL: https://openweathermap.org/api/air-pollution.

[115] Patrick L. Kinney. «Climate Change, Air Quality, and Human Health». Em: *American Journal of Preventive Medicine* 35.5 (2008). Theme Issue: Climate Change and the Health of the Public, pp. 459–467. ISSN: 0749-3797. DOI: https://doi.org/10.1016/j.amepre.2008.08.025. URL: https://www.sciencedirect.com/science/article/pii/S0749379708006909.

[116] Brian Widdop. «Analysis of carbon monoxide». Em: *Annals of Clinical Biochemistry* 39.4 (2002). PMID: 12117442, pp. 378–391. DOI: 10.1258/000456302760042146. eprint: https://doi.org/10.1258/000456302760042146. URL: https://doi.org/10.1258/000456302760042146.

[117] Patrick L. Kinney. «Interactions of Climate Change, Air Pollution, and Human Health». Em: *Current Environmental Health Reports* 5.1 (2018), pp. 179–186. ISSN: 2196-5412. DOI: 10.1007/s40572-018-0188-x. URL: https://doi.org/10.1007/s40572-018-0188-x.

[118] Leann Myers e Maria J. Sirois. «Spearman Correlation Coefficients, Differences between». Em: *Encyclopedia of Statistical Sciences*. John Wiley Sons, Ltd, 2006. ISBN: 9780471667193. DOI: https://doi.org/10.1002/0471667196.ess5050.pub2. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/0471667196.ess5050.pub2. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/0471667196.ess5050.pub2.

[119] Yan Zhu et al. «Matrix Profile XI: SCRIMP++: Time Series Motif Discovery at Interactive Speeds». Em: *2018 IEEE International Conference on Data Mining (ICDM)*. 2018, pp. 837–846. DOI: 10.1109/ICDM.2018.00099.

[120] Government of Portugal. *Comunicado de Conselho de Ministros*. Accessed: 29 de janeiro de 2024. Year. URL: https://www.portugal.gov.pt/pt/gc22/governo/comunicado-de-conselho-de-ministros?i=400.

[121] Samit Bhanja e Abhishek Das. *Impact of Data Normalization on Deep Neural Network for Time Series Forecasting*. 2019. arXiv: 1812.05519 [cs.LG].

[122] Alaa Sagheer e Mostafa Kotb. «Time series forecasting of petroleum production using deep LSTM recurrent networks». Em: *Neurocomputing* 323 (2019), pp. 203–213. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2018.09.082. URL: https://www.sciencedirect.com/science/article/pii/S0925231218311639.

[123] Ola Surakhi et al. «Time-Lag Selection for Time-Series Forecasting Using Neural Network and Heuristic Algorithm». Em: *Electronics* 10.20 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10202518. URL: https://www.mdpi.com/2079-9292/10/20/2518.

[124] Jian Cao, Zhi Li e Jian Li. «Financial time series forecasting model based on CEEMDAN and LSTM». Em: *Physica A: Statistical Mechanics and its Applications* 519 (2019), pp. 127–139. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2018.11.061. URL: https://www.sciencedirect.com/science/article/pii/S0378437118314985.

[125] Christian Garbin, Xingquan Zhu e Oge Marques. «Dropout vs. batch normalization: an empirical study of their impact to deep learning». Em: *Multimedia Tools and Applications* 79.19 (2020), pp. 12777–12815. ISSN: 1573-7721. DOI: 10.1007/s11042-019-08453-9. URL: https://doi.org/10.1007/s11042-019-08453-9.

[126] Xue Ying. «An Overview of Overfitting and its Solutions». Em: *Journal of Physics: Conference Series* 1168.2 (2019), p. 022022. DOI: 10.1088/1742-6596/1168/2/022022. URL: https://dx.doi.org/10.1088/1742-6596/1168/2/022022.

[127] Sangyeon Kim e Myungjoo Kang. *Financial series prediction using Attention LSTM*. 2019. arXiv: 1902.10877 [cs.LG].

[128] Xuan Zhang et al. «AT-LSTM: An Attention-based LSTM Model for Financial Time Series Prediction». Em: *IOP Conference Series: Materials Science and Engineering* 569.5 (2019), p. 052037. DOI: 10.1088/1757-899X/569/5/052037. URL: https://dx.doi.org/10.1088/1757-899X/569/5/052037.

[129] Isabelle M Guyon. «Practical Feature Selection: from Correlation to Causality». Em: *NATO ASI Mining Massive Data Sets for Security*. 2007. URL: https://api.semanticscholar.org/CorpusID:2937130.

[130] Christoph Molnar et al. «Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process». Em: *Explainable Artificial Intelligence*. Ed. por Luca Longo. Cham: Springer Nature Switzerland, 2023, pp. 456–479. ISBN: 978-3-031-44064-9.

[131] European Environment Agency. *Air quality in Europe — 2020 report*.

[132] European Environment Agency. *Air Pollution sources*.

[133] Suqin Han et al. «Analysis of the Relationship between O3, NO and NO2 in Tianjin, China». Em: *Aerosol and Air Quality Research* 11.2 (2011), pp. 128–139. DOI: 10.4209/aaqr.2010.07.0055. URL: https://doi.org/10.4209%2Faaqr.2010.07.0055.

[134] Fulvio Amato et al. «Urban air quality: The challenge of traffic non-exhaust emissions». Em: *Journal of Hazardous Materials* 275 (2014), pp. 31–36. ISSN: 0304-3894. DOI: https://doi.org/10.1016/j.jhazmat.2014.04.053. URL: https://www.sciencedirect.com/science/article/pii/S030438941400315X.