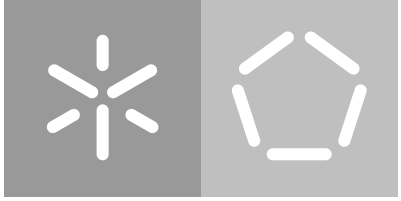


Universidade do Minho
Escola de Engenharia

Duarte Augusto Rodrigues Lucas

Utilização de modelos de Deep Learning para prever os parâmetros do reator do digestor de lamas anaeróbias: Um problema de séries temporais

Outubro, 2024



Universidade do Minho
Escola de Engenharia

Duarte Augusto Rodrigues Lucas

Utilização de modelos de Deep Learning para prever os parâmetros do reator do digestor de lamas anaeróbias: Um problema de séries temporais

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho efetuado sob a orientação do(a)
Paulo Jorge Freitas Oliveira Novais

Outubro, 2024

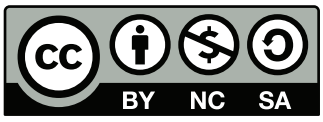
DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

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 RepositoriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International
CC BY-NC-SA 4.0

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

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.

Santo-Tirso, 30/10/2024

(Localização) (Data)

Assinado por: **Duarte Augusto Rodrigues Lucas**
Num. de Identificação: 14652490
Data: 2024.10.30 18:30:03 +0000

(Duarte Augusto Rodrigues Lucas)

Agradecimentos

Começo por agradecer ao meu orientador, Professor **Paulo Jorge Freitas Oliveira Novais**, pela orientação e formação académica, que demonstraram grande valor na realização desta dissertação e no meu desenvolvimento enquanto profissional.

Ao Professor **Pedro José Costa Oliveira**, pelo acompanhamento, conselhos e conhecimento partilhado desde o início até ao fim desta caminhada na realização da dissertação, bem como pela paciência, companheirismo, boa vontade e ajuda que sempre demonstrou sem hesitar, os quais foram essenciais nesta longa jornada.

À **Universidade do Minho**, pelos melhores cinco anos académicos, e a todos os professores que me acompanharam nesta jornada, partilhando o seu conhecimento e contribuindo para o meu crescimento.

Aos meus pais, **António Lucas** e **Alice Rodrigues**, por tudo o que fizeram por mim desde sempre, pelos sacrifícios, pela luta, por me darem a mão quando mais precisei e, acima de tudo, por me incutirem os valores que fizeram de mim a pessoa que sou hoje. Ao meu irmão, **Edgar Lucas**, por ser a minha luz, a minha companhia, por me mostrar que tudo tem solução e por me impulsionar a ambicionar sempre mais. Agradeço-lhe por trazer sempre o melhor de mim. A eles dedico tudo o que fiz e continuarei a fazer.

Aos meus avós paternos, **Joaquim Lucas** e **Maria Lucas**, e aos avós maternos, **Diamantino Rodrigues** e **Maria Alice**, agradeço pelos valores transmitidos, pela companhia nos melhores e nos piores momentos, e pelo amor e carinho que sempre recebi. São e serão as pessoas que representarão o expoente máximo do orgulho e amor, e farão sempre parte de todas as minhas conquistas.

À minha família, com especial menção à **Ana Luísa** e ao **Diogo Lucas**, pelo apoio constante, pela preocupação e por sempre demonstrarem alegria e otimismo.

Aos meus amigos que a universidade me proporcionou, especialmente ao **Francisco Novo**, **João Vieira**, **José Magalhães**, **Tiago Ribeiro**, **Francisco Izquierdo**, **Fábio Meira**, **Filipe Dias**, **Paulo Oliveira** e **António Luís**, por se terem tornado uma família nos bons e nos maus momentos, pelas memórias inesquecíveis, pela ajuda e por serem o melhor destes cinco anos académicos.

Aos amigos que sempre tiveram um papel crucial tanto na minha vida pessoal como na minha jornada académica, **Diogo Silva**, **Gabriel Vinhas**, **Leandro Brito**, **André Arantes** e **Sofia Cunha**, o meu grande agradecimento.

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).

A todos um sincero obrigado.

Resumo

A otimização dos digestores anaeróbios em reatores de lamas não só é essencial para garantir o tratamento eficiente dos afluentes, como também desempenha um papel fundamental na produção de energia renovável. O biogás gerado no processo de digestão anaeróbia, composto principalmente por percentagem de metano (%CH4), pode ser capturado e convertido em energia, contribuindo para a redução da dependência de fontes energéticas não renováveis. Desta forma, a gestão eficiente destes sistemas permite não só cumprir os padrões regulatórios de qualidade ambiental, mas também promover a sustentabilidade energética, maximizando a produção de biogás e, consequentemente, o aproveitamento de uma fonte de energia limpa e renovável.

A aplicação de técnicas avançadas, como o Deep Learning (DL) em conjunto com séries temporais, apresenta um grande potencial para melhorar a previsão e gestão de sistemas complexos. O DL, com a capacidade de aprender autonomamente a partir de dados complexos, aliado à análise de séries temporais, que permite a compreensão de padrões ao longo do tempo, oferecem uma valiosa ferramenta para otimizar a previsão e gestão de parâmetros nos digestores de lamas anaeróbias. Essa combinação de técnicas possibilita uma resposta mais precisa e adaptativa às variações nas condições operacionais, contribuindo para a eficiência global do sistema.

Deste modo, esta dissertação foca-se na implementação de diversos modelos de DL, incluindo Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Perceptron Multicamadas (MLP), Redes LSTNet (Long- and Short-term Time-series Network) e Redes Neurais Convolucionais Temporais (TCN), para a previsão dos valores de %CH4 e da concentração de biomassa ativa (Xc) resultantes da digestão anaeróbia realizada no digestor para os proximos dois dias, utilizando uma abordagem *multistep recursive forecaste*. Após a análise comparativa dos modelos candidatos, verificou-se que os modelos baseados em CNN alcançaram os melhores resultados na previsão tanto dos valores de %CH4 como de Xc, destacando-se pela sua eficácia e precisão, com um valor de *Root Mean Square Error* (RMSE) de 0.0667% e 1028.06 mgO2/L respetivamente.

Palavras-chave Aprendizagem profunda, Digestão anaeróbia, Estação de Tratamento de Águas Residuais, Inteligência artificial, Séries temporais

Abstract

The optimization of anaerobic digesters in sludge reactors is not only essential to ensure the efficient treatment of effluents, but also plays a key role in the production of renewable energy. The biogas generated in the anaerobic digestion process, composed mainly of methane (CH4), can be captured and converted into energy, helping to reduce dependence on non-renewable energy sources. In this way, the efficient management of these systems makes it possible not only to comply with regulatory environmental quality standards, but also to promote energy sustainability, maximizing biogas production and, consequently, the use of a clean and renewable energy source.

The application of advanced techniques such as Deep Learning (DL) in conjunction with time series has great potential for improving the prediction and management of complex systems. DL, with its ability to learn autonomously from complex data, combined with time series analysis, which allows patterns to be understood over time, offers a valuable tool for optimizing the prediction and management of parameters in anaerobic sludge digesters. This combination of techniques enables a more precise and adaptive response to variations in operating conditions, contributing to the overall efficiency of the system.

This work therefore focuses on the implementation of various DL models, including Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Multilayer Perceptron (MLP), Long- and Short-term Time-series Networks (LSTNet) and Temporal Convolutional Neural Networks (TCN), for predicting %CH4 values and the concentration of active biomass (Xc) resulting from the anaerobic digestion carried out in the digester. After comparative analysis of the candidate models, it was found that the CNN-based models achieved the best results in predicting both %CH4 and Xc values, standing out for their effectiveness and accuracy, with a *Root Mean Square Error* (RMSE) value of 0.0667% and 1028.06 mgO2/L respectively.

Keywords Anaerobic digestion, Artificial intelligence, Deep learning, Time series, Wastewater treatment plant

Índice

Lista de Figuras x

Lista de Tabelas xii

Siglas xiv

1 Introdução 1

1.1 Enquadramento e Motivação . . . 1

1.1.1 Tipos de Poluição . . . 2

1.1.2 ETAR . . . 6

1.1.3 Reator de Digestão Anaeróbica . . . 6

1.2 Objetivos . . . 7

1.3 Estrutura do Documento . . . 7

2 Estado de Arte 9

2.1 Poluição da Água . . . 9

2.1.1 ETAR . . . 10

2.1.2 Funcionamento de uma ETAR . . . 11

2.1.3 Reator de Digestão Anaeróbio . . . 13

2.1.4 Princípio de Funcionamento . . . 13

2.1.5 Transformação Biológica . . . 13

2.1.6 Eficiência na Redução de Volume . . . 13

2.1.7 Importância da Digestão Anaeróbia . . . 14

2.1.8 Parâmetros do Reator . . . 15

2.2 Inteligência Artificial . . . 15

2.2.1 Machine Learning . . . 17

2.2.2 Deep Learning . . . 21

2.2.3 Series Temporais . . . 23

2.2.4 Problemas de Regressão em Séries Temporais . . . 25

2.2.5 Modelos *Deep Learning* em Séries Temporais . . . 26

2.2.6 CNN . . . 30

2.2.7 GRU . . . 30
2.2.8 LSTNet . . . 32
2.2.9 TCN . . . 34
2.2.10 MLP . . . 36
2.3 Revisão da Literatura . . . 38
2.3.1 Análise Crítica sobre os Estudos . . . 41

3 Materiais e Métodos 43
3.1 Exploração dos Dados . . . 43
3.1.1 Análise do Dataset . . . 43
3.2 Preparação de Dados . . . 50
3.2.1 Tratamento dos Missing Values . . . 50
3.2.2 Tratamento dos Outliers . . . 51
3.2.3 Feature Engineering . . . 53
3.2.4 Análise de Correlação e Seleção de *Feature* . . . 54
3.3 Tecnologias . . . 55

4 Experiências 56
4.1 Implementação de Modelos . . . 56
4.1.1 LSTM . . . 57
4.1.2 CNN . . . 59
4.1.3 MLP . . . 61
4.1.4 LSTNet . . . 62
4.1.5 TCN . . . 64
4.1.6 GRUS . . . 67
4.2 Métricas de Avaliação . . . 68

5 Resultados e Discussão 69
5.1 %CH4 . . . 69
5.1.1 LSTM . . . 70
5.1.2 CNN . . . 71
5.1.3 MLP . . . 71
5.1.4 LSTNet . . . 72
5.1.5 TCN . . . 73
5.1.6 GRU . . . 73
5.1.7 Análise Comparativa . . . 74
5.2 Xc . . . 75
5.2.1 LSTM . . . 76
5.2.2 CNN . . . 76

5.2.3 MLP 77
5.2.4 LSTNet 78
5.2.5 TCN 78
5.2.6 GRU 79
5.2.7 Análise Comparativa 79

6 Conclusão e Trabalho Futuro **81**

Bibliografia **85**

Lista de Figuras

1.1 Tipos de Poluição, adaptado de [8] . . . 2
1.2 Mortes por fator de risco, retirado de [39] . . . 5

2.1 Ciclo Urbano, adaptado de [51] . . . 10
2.2 Passagem da água do meio ambiente para habitações, retirado de [51] . . . 11
2.3 Passagem da água das habitações para o meio ambiente, retirado de [51] . . . 11
2.4 Funcionamento Geral de uma Estações de Tratamento de Águas Residuais (ETAR), adaptado de [51] . . . 12
2.5 Princípio de Funcionamento, adaptado de [58] . . . 13
2.6 Subcampos da Inteligência Artificial (IA), adaptado de [91] . . . 17
2.7 Algoritmo de Machine Learning (ML) adaptado de: [88] . . . 18
2.8 Tipos de Aprendizagem, adaptado de [76] . . . 19
2.9 Estrutura de uma Rede Neuronal Artificial (RNA), adaptado de [100] . . . 22
2.10 Exemplo de comportamento sazonal. Retirado de: [101] . . . 24
2.11 Exemplo de arquitetura Recurrent Neural Network (RNN), adaptado de [111] . . . 27
2.12 Exemplo de arquitetura Long Short-Term Memory (LSTM), adaptado de [115] . . . 29
2.13 Exemplo de arquitetura Convolutional Neural Network (CNN),retirado de: [117] . . . 30
2.14 Estrutura exemplo GRU, adaptado de: [120] . . . 32
2.15 Estrutura exemplo LSTNet, adaptado de: [124] . . . 33
2.16 Estrutura exemplo Temporal Convolutional Network (TCN), retirado de: [126] . . . 35
2.17 Estrutura exemplo MLP, retirado de: [129] . . . 36

3.1 Histograma %Metano (CH4) . . . 46
3.2 Valores médios anuais de %CH4 para cada estação do ano. . . . 46
3.3 Valores médios mensais CH4 para cada ano. . . . 47
3.4 Histograma Concentração de Biomassa Ativa (Xc) . . . 48
3.5 Valores médios anuais de Xc para cada estação do ano. . . . 49
3.6 Valores médios mensais Xc para cada ano. . . . 50
3.7 *BoxPlot* de %CH4 antes do tratamento. . . . 51
3.8 *BoxPlot* de %CH4 após o tratamento. . . . 52
3.9 *BoxPlot* de Xc antes do tratamento. . . . 53

3.10 BoxPlot de Xc após o tratamento. 53
3.11 Matriz de correlação. 54

5.1 Comparação dos melhores resultados obtidos na previsão da %CH4. 74
5.2 Previsões de %CH4 para 2 *timesteps* futuros. 75
5.3 Comparação dos melhores resultados obtidos na previsão da Xc. 80
5.4 Previsões de Xc para 2 *timesteps* futuros. 80

Lista de Tabelas

3.1 Tipos de cada coluna 44
3.2 Estatística do parâmetro %CH4. 45
3.3 Estatística do parâmetro Xc. 47
3.4 *Features* do novo *dataset*. 54

4.1 Hiperparâmetros. 57

5.1 5 melhores resultados do modelo LSTM. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *epochs*, h - Mean Absolute Error (MAE), i - Root Mean Squared Error (RMSE), j - tempo(s). 70
5.2 5 melhores resultados do modelo CNN. Significado das letras: a - *layers*, b - *dropout_rate*, c - *activation*, d - *timesteps*, e - *batch_size*, f - *filters*, g - *kernel_size*, h- *pool_size*, i - *epochs*, j - MAE, k - RMSE, l - tempo(s). 71
5.3 5 melhores resultados do modelo Multilayer Perceptron (MLP). Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s). 71
5.4 5 melhores resultados do modelo Long Short-Term Network (LSTNet). Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *filters*, h - *kernel_size*, i - *epochs*, j - MAE, k - RMSE, l - tempo(s). 72
5.5 5 melhores resultados do modelo TCN. Significado das letras: a - *layers*, b - *dropout_rate*, c - *activation*, d - *timesteps*, e - *batch_size*, f - *kernel_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s). 73
5.6 5 melhores resultados do modelo Gated Recurrent Unit (GRU). Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s). 73
5.7 5 melhores resultados do modelo LSTM. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s). 76
5.8 5 melhores resultados do modelo CNN. Significado das letras: a - *layers*, b - *dropout_rate*, c - *activation*, d - *timesteps*, e - *batch_size*, f - *filters*, g - *kernel_size*, h- *pool_size*, i - *epochs*, j - MAE, k - RMSE, l - tempo(s). 76

5.9 5 melhores resultados do modelo MLP. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s). 77
5.10 5 melhores resultados do modelo LSTNet. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *filters*, h - *kernel_size*, i - *epochs*, j - MAE, k - RMSE, l - tempo(s). 78
5.11 5 melhores resultados do modelo TCN. Significado das letras: a - *layers*, b - *dropout_rate*, c - *activation*, d - *timesteps*, e - *batch_size*, f - *kernel_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s). 78
5.12 5 melhores resultados do modelo GRU. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s). 79

Siglas

AcoD Co-digestão anaeróbia

AG Algoritmo Genético

AG-LSTM Algoritmo Genético - Long Short-Term Memory

API Application Programming Interface

AR Águas Residuais

AutoML máquina automatizado

C3D Convolutional 3D

CH4 Metano

CNN Convolutional Neural Network

CNN 1D Redes Neurais Convolucionais 1D

CRISP-DM Cross-Industry Standard Process for Data Mi- ning

DA-LSTM Drift-Adaptive, Long Short-Term Memory

DA-LSTM-VSN Drift-Adaptive, Long Short-Term Memory, Visual Sensor Network

DBO5 Demanda Bioquímica de Oxigênio em 5 dias

DL Deep Learning

DNN Deep Neural Network

DQO Demanda Química de Oxigênio

ETA Estação de Tratamento de Águas

ETAR Estações de Tratamento de Águas Residuais

GBR Gradient Boosting Regression

GPU Graphics Processing Units

GRU Gated Recurrent Unit

HRT Tempo de Retenção Hidráulica

IA Inteligência Artificial

LSTM Long Short-Term Memory

LSTNet Long Short-Term Network

MAE Mean Absolute Error

ML Machine Learning

MLP Multilayer Perceptron

NLP Processamento de Linguagem Natural

OFR Taxa de Alimentação Orgânica

ReLU Rectified Linear Unit

RF Random Forest

RL Reinforcement Learning

RMSE Root Mean Squared Error

RNA Rede Neuronal Artificial

RNN Recurrent Neural Network

SHAP SHapley Additive exPlanations

SSL Semissupervisionado

SSV Sólidos Suspensos Voláteis

SVT Concentração de Sólidos Voláteis Totais

TCN Temporal Convolutional Network

TN Nitrogênio Total

TRH Tempo de Retenção Hidráulico

VSN Rede de Seleção de Variáveis

Xc Concentração de Biomassa Ativa

Capítulo 1

Introdução

Na atualidade, a Engenharia Informática desempenha um papel vital na abordagem de desafios ambientais complexos. Este estudo é fruto do compromisso com a disciplina de Dissertação em Engenharia Informática, na Universidade do Minho, intitulado de "Utilização de modelos de Deep Learning para prever os parâmetros do reator do digestor de lamas anaeróbias: Um problema de séries temporais". O presente trabalho revela em pormenor temas relacionados com Águas Residuais (AR) provenientes de ETAR, explorando conceitos avançados de Deep Learning (DL) e como eles se entrelaçam de maneira fundamental com as séries temporais, especialmente no contexto do digestor de lamas anaeróbias.

Este capítulo inicial está estruturado em três secções interligadas. Na secção 1.1, é fornecida uma visão abrangente do contexto e da motivação subjacentes a esta dissertação. Na secção 1.2, são delineados e descritos os objetivos que guiam este documento. Por fim, na secção 1.3, é apresentada a estrutura de todo o documento acompanhado de uma breve descrição de cada subtítulo.

1.1 Enquadramento e Motivação

A busca pela sustentabilidade ambiental é um dos maiores desafios do século XXI [1]. À medida que a população global continua a crescer, juntamente com a expansão da urbanização e industrialização, o impacto das atividades humanas no meio ambiente tornou-se cada vez mais evidente [2]. Por um lado, essa busca pela sustentabilidade ambiental trouxe benefícios notáveis em diversas áreas de estudo, como por exemplo estudos sobre mudanças climáticas e mitigação, e na vida quotidiana das pessoas, a mesma resultou em avanços significativos que afetaram positivamente setores como a ciência, tecnologia e qualidade de vida [3] [4]. Por outro lado, também desencadeou desafios complexos que ameaçam a harmonia dos ecossistemas e a qualidade de vida no planeta [3] [4]. Estes desafios incluem problemas como a perda de biodiversidade, a degradação ambiental e as mudanças climáticas, que exigem uma

ação urgente e coordenada para serem enfrentados [5].

Este conceito de sustentabilidade ambiental é fundamental num mundo que se encontra em constante evolução. A preocupação com os diversos tipos de poluição está intrinsecamente ligada a essa busca, uma vez que a poluição representa uma das principais ameaças à estabilidade dos ecossistemas e à qualidade de vida no planeta [6]. Ao longo do tempo é possível verificar que o crescimento contínuo da população global, juntamente com a expansão da urbanização e industrialização, têm contribuído para o aumento do impacto das atividades humanas no meio ambiente [6].

A complexidade do desafio da poluição está refletida em várias formas distintas. A poluição ambiental assume várias facetas, cada uma com implicações específicas para a saúde do planeta e para os seres que nela habitam [7].

1.1.1 Tipos de Poluição

Entre as principais categorias de poluição, podemos destacar a poluição do ar, do solo, sonora, da água, da luz, do lixo, da contaminação radioativa, da poluição térmica e por fim da poluição visual [8], como ilustrado na Figura 1.1.

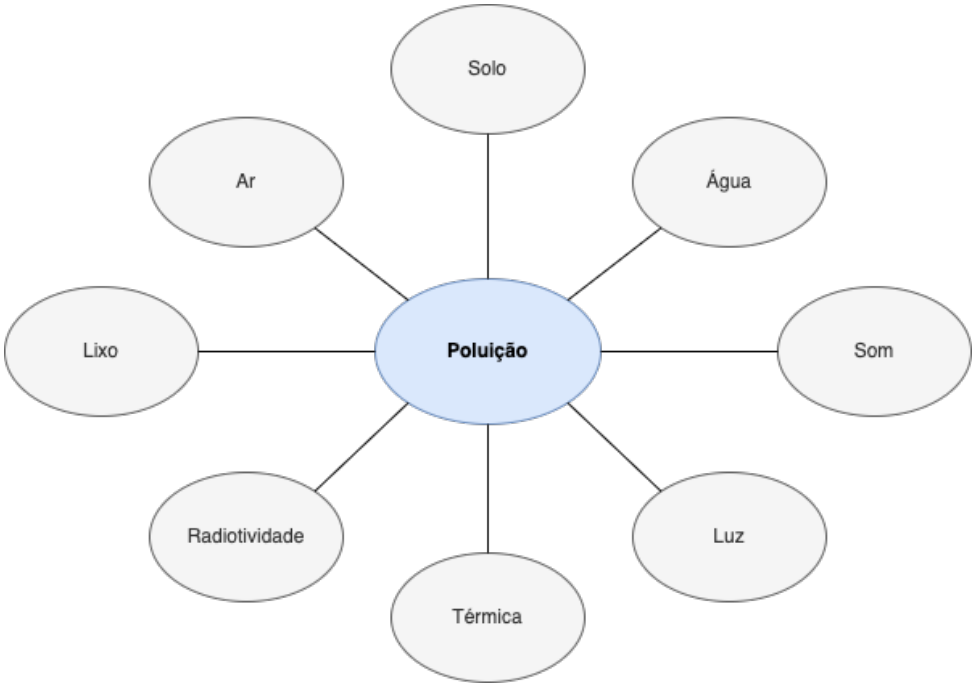


Figura 1.1: Tipos de Poluição, adaptado de [8]

- A poluição do ar é o resultado direto da industrialização e da queima de combustíveis fósseis que por sua vez provocam impactos significativos na qualidade de vida das pessoas, especialmente naquelas que residem em cidades densamente povoadas e fortemente poluídas [9]. A exposição crónica a poluentes atmosféricos, como dióxido de enxofre, óxidos de nitrogénio e partículas em suspensão, pode levar a uma série de problemas de saúde [10] [11]. A população urbana, em particular, enfrenta um aumento das taxas de doenças respiratórias, como asma, bronquite e cancro

nos pulmões, devido à inalação dessas partículas poluentes [12]. Além disso, a poluição atmosférica contribui para a má qualidade do ar nas cidades, resultando em dias de *smog* frequentes, que afetam a visibilidade, tornando a respiração difícil além de causar irritações nos olhos [13]. Além disso, a poluição do ar desempenha um papel importante nas mudanças climáticas, que foram mencionadas como um dos desafios da sustentabilidade ambiental.

- A poluição do solo, que resulta principalmente da deposição de resíduos tóxicos, produtos químicos agrícolas e poluentes industriais [14], tem um impacto significativo na fertilidade do solo, o que por sua vez compromete os alimentos nele cultivados [15]. Os alimentos cultivados em solos poluídos têm maior probabilidade de absorver resíduos tóxicos e produtos químicos, o que pode representar um risco significativo para a saúde humana [16]. A ingestão de alimentos contaminados por substâncias nocivas pode levar a uma série de problemas de saúde, incluindo intoxicações alimentares, diminuição das defesas imunológicas, retardo do crescimento intra-uterino, deficiências associadas à desnutrição e alta prevalência de taxas de cancro [17]. Além disso, a poluição do solo pode levar à contaminação de lençóis de água, o que afeta diretamente o abastecimento de água potável, uma vez que a água contaminada por produtos químicos e poluentes torna-se imprópria para consumo humano [17].

- A poluição sonora, provocada pela emissão de ruídos provenientes de fontes industriais, tráfego rodoviário, aeroportos e atividades humanas em áreas urbanas, impacta negativamente o bem-estar das pessoas e a qualidade do ambiente [18]. A exposição constante a altos níveis de ruído pode causar uma série de problemas de saúde, incluindo stress, distúrbios de sono, perda auditiva e aumento do risco de doenças cardiovasculares [19][20].

- A poluição da água é um problema significativo e preocupante, que resulta das descargas inadequadas e sem controle de resíduos industriais, esgotos e produtos químicos tóxicos [21]. Tem consequências devastadoras para a vida aquática, como a morte em massa de peixes e outros organismos aquáticos devido à exposição a substâncias poluentes [22]. Além disso, a poluição da água pode representar uma séria ameaça à saúde humana. A ingestão ou exposição de água contaminada por produtos químicos tóxicos, bactérias patogénicas e poluentes orgânicos pode resultar em numa série de problemas de saúde, incluindo doenças gastrointestinais, infeções, e até mesmo o desenvolvimento de cancro em certos casos [23] [24]. Além dos riscos diretos para a saúde, a poluição da água também afeta o abastecimento de água potável. A contaminação da água pode tornar fontes de água doce impróprias para consumo humano, o que por vezes leva à morte, principalmente de crianças [25].

- A poluição luminosa é o resultado do excesso de iluminação artificial em áreas urbanas e rurais. Este tipo de poluição pode provocar efeitos prejudiciais na vida selvagem e nos ciclos naturais [26] [27]. A luminosidade excessiva perturba os ritmos circadianos de muitas espécies, afetando a sua reprodução, migração e hábitos alimentares [26].

- A poluição radioativa manifesta-se quando as substâncias radioativas são libertadas no meio ambiente, como meio aquático, frequentemente como resultado de resíduos de atividades nucleares ou incidentes nucleares [28]. Essa forma de poluição é altamente perigosa, trazendo riscos significativos para a saúde humana e para o meio ambiente. A exposição a materiais radioativos pode resultar em condições graves, como o cancro, mutações genéticas e outros problemas de saúde de longo prazo [29], como ficou evidente ao longo dos anos após o pior desastre nuclear conhecido, o incidente de Chernobyl [30] [31].

- A poluição térmica está diretamente relacionada com o aumento da temperatura de corpos de água, geralmente causado pelas descargas de água aquecida em processos industriais e centrais elétricas [32]. Esta variação térmica afeta o bem estar de muitas espécies uma vez que condiciona a qualidade e temperatura da água que estes animais precisam, além de afetar a qualidade da água [32]. Por outro lado , a poluição térmica desempenha um papel relevante nas questões ambientais mais amplas, como o aquecimento global e as alterações climáticas [33]. O aumento da temperatura da água causado pela poluição térmica pode contribuir para o aumento da temperatura do ar nas áreas afetadas, com implicações no clima local e regional, podendo agravar os efeitos das alterações climáticas [34].

- A poluição visual, por vezes ignorada, está relacionada com a presença indesejada e excessiva de elementos visuais que interferem na estética e na qualidade do ambiente [35]. Este tipo de poluição é mais comum em áreas urbanas podendo incluir uma variedade de elementos, como *outdoors* excessivos, fios elétricos, edifícios desproporcionais, publicidade intrusiva, entre outros [35] [36]. A poluição visual pode ter diversos impactos, como causar desconforto psicológico e stress nas pessoas, uma vez que um ambiente visualmente poluído pode ser avassalador e perturbador para as mesmas [37]. Além disso, a presença excessiva de anúncios publicitários e elementos visuais muitas vezes tem como objetivo a procura de atenção do público de maneira agressiva, o que pode ser invasivo e incómodo [37].

A relação entre a busca pela sustentabilidade ambiental e a necessidade de enfrentar a poluição é clara. Para alcançar a sustentabilidade é essencial abordar esses tipos de poluição de maneira eficaz, adotando práticas sustentáveis, regulamentações ambientais mais rígidas e a promoção de tecnologias limpas [1]. A redução da poluição é fundamental para preservar ecossistemas saudáveis e garantir um futuro sustentável para as gerações presentes e futuras [1].

Contudo torna-se evidente que a poluição da água ocupa uma posição de destaque no âmbito da sustentabilidade ambiental, isso deve-se à sua necessidade na saúde humana, segurança alimentar e na qualidade do ambiente [38]. Entre os vários tipos de poluição, a contaminação da água sobressai devido ao seu potencial impacto devastador [25]. Como supramencionado, a poluição da água não é apenas uma preocupação ambiental, mas também uma questão de saúde pública de extrema relevância. Dados

alarmantes indicam que a poluição da água é responsável por um número significativo de mortes em todo o mundo, principalmente de crianças [25], superando muitas outras causas de mortalidade.

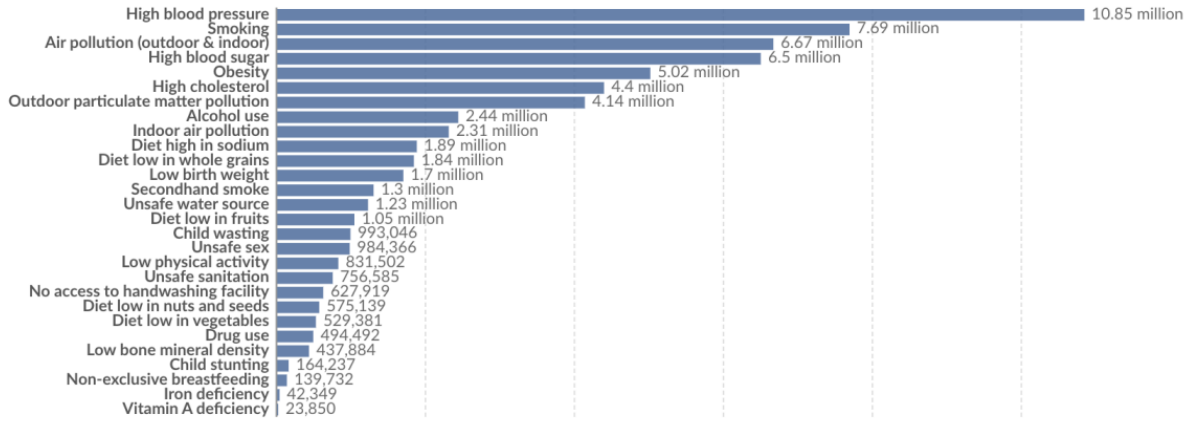


Figura 1.2: Mortes por fator de risco, retirado de [39]

A Figura 1.2 representa o numero de mortes por fator de risco, é possível analisar que a mortalidade devido ao consumo de água não potável atingiu a marca de 1.23 milhões de pessoas, posicionando-se entre as 14 principais causas de morte com maior letalidade a nível mundial.

Diante desses desafios críticos, foi necessário desenvolver esforços significativos para combater os efeitos prejudiciais da poluição ambiental e garantir a sustentabilidade de ecossistemas aquáticos e recursos hídricos. Esses esforços envolveram não apenas a regulamentação e a implementação de políticas ambientais rigorosas, mas também avanços significativos na tecnologia e pesquisa científica [40].

A consciencialização crescente sobre a importância da qualidade da água levou ao desenvolvimento de métodos de controlo mais avançados e à implementação de sistemas de tratamento de água cada vez mais eficazes. Entre os métodos mais utilizados estão a filtração por membranas, a ozonização e a utilização de bioreatores de membrana, que oferecem maior eficiência na remoção de poluentes e contaminantes emergentes, como resíduos farmacêuticos e microplásticos [41]. Além disso, a pesquisa em áreas como a recuperação de ecossistemas aquáticos degradados e a gestão sustentável dos recursos hídricos tem desempenhado um papel fundamental na abordagem desses desafios globais [42], o que levou a uma necessidade de uma gestão das águas poluentes. A gestão eficiente das AR desempenha um papel fundamental na busca pela sustentabilidade ambiental em todo o mundo, sendo essencial para preservar a qualidade de vida das atuais e futuras gerações [42]. Essa importância é substanciada por uma série de fatores interligados que afetam direta e indiretamente tanto o meio ambiente quanto a sociedade. Algumas considerações essenciais sobre a importância da gestão eficiente das AR são:

- **Preservação dos Recursos Hídricos:** A água é um recurso finito e essencial para a sobrevivência de todos os seres vivos. A gestão eficiente das AR é crucial para preservar a disponibilidade de água doce em rios, lagos e aquíferos. Ao tratar as AR e devolvê-las ao ambiente com padrões de qualidade aceitáveis, minimiza-se o esgotamento dos recursos hídricos, contribuindo para a sustentabilidade a longo prazo [43].

- **Proteção da Qualidade da Água:** AR frequentemente contêm uma variedade de poluentes, incluindo produtos químicos tóxicos, materiais orgânicos em decomposição e microrganismos patogénicos. A gestão eficaz das AR envolve a remoção ou redução desses poluentes, garantindo que a água tratada não prejudique ecossistemas aquáticos e não represente riscos para a saúde humana [42].

- **Prevenção de Doenças e Riscos à Saúde:** AR inadequadamente tratadas podem ser veículos de doenças transmitidas pela água, colocando em risco a saúde pública. A gestão adequada das AR inclui a eliminação de doenças patogénicas e a minimização dos riscos à saúde, reduzindo a incidência de doenças relacionadas à água [44].

Neste cenário, o tratamento de AR surge como uma solução eficaz para a redução da poluição da água e da prevenção de doenças e proteção do ecossistema. As ETAR desempenham um papel crítico na remoção de poluentes das AR antes que sejam devolvidas aos ecossistemas naturais ou utilizadas para consumo humano. O processo de tratamento envolve a eliminação de sólidos suspensos, a desinfeção de bactérias, a remoção de nutrientes em excesso e a degradação de substâncias químicas nocivas.

1.1.2 ETAR

Uma ETAR é a infraestrutura responsável por receber, tratar e limpar AR domésticas e industriais antes de serem devolvidas ao meio ambiente. Além de representar um papel crucial no ciclo urbano da água, é bastante importante na gestão e preservação dos recursos hídricos [45]. A principal função de uma ETAR é remover poluentes, sólidos suspensos, microrganismos patogénicos e substâncias químicas prejudiciais presentes nas AR. Posto isto, as ETAR desempenham um papel fundamental na prevenção da poluição da água, garantindo que a água tratada atenda aos padrões de qualidade aceitáveis antes de ser devolvida aos rios, lagos ou oceanos [45].

A importância das ETAR destaca-se especialmente em áreas urbanas densamente povoadas, onde a concentração de poluentes é significativa. A falta de tratamento adequado de AR resultaria na libertação direta de poluentes no meio ambiente, o que por usa vez intensificaria os problemas de poluição da água [45]. Além disso, num mundo cada vez mais consciente da escassez de recursos hídricos, o tratamento eficiente de AR possibilita a reutilização da água, não sendo necessário recorrer a recursos não renováveis [45].

1.1.3 Reator de Digestão Anaeróbica

Além das etapas convencionais de tratamento de AR mencionadas anteriormente, destaca-se um componente crucial no controlo responsável dos resíduos gerados: o tratamento de lamas. A gestão eficiente desses subprodutos do processo de purificação das AR é essencial para garantir a sustentabilidade ambiental das ETAR [46].

No âmbito do tratamento de lamas, surge uma tecnologia de grande importância, os reatores de digestão anaeróbica. Estes reatores desempenham um papel fundamental na transformação das lamas resultantes do tratamento de AR, utilizam um processo biológico sem a presença de oxigénio. Este processo anaeróbico não reduz apenas a quantidade de resíduos sólidos, mas também oferece benefícios ambientais e energéticos [47] [46].

1.2 Objetivos

Neste trabalho, foi delineado uma série de objetivos que visam explorar o uso de modelos de DL para aprimorar a gestão dos digestores de lamas anaeróbicas nas ETAR. Foi então estabelecido como objetivos principais os seguintes:

- Revisão da literatura existente sobre o uso de modelos de ML e DL para o digestor de lamas anaeróbicas em ETAR;
- Compreensão e visualização de dados relacionados a diferentes parâmetros presentes no digestor de lamas anaeróbicas das ETAR;
- Pré-processar os dados tendo em conta aspetos específicos de problemas de séries temporais: tratamento de valores em falta, uso de um validador cruzado específico, *feature engineering* e análise de correlação e *feature selection* .;
- Conceber modelos de DL para prever os parâmetros %CH4 e Xc no digestor de lodo das ETAR;
- Análise e discussão dos resultados obtidos.

1.3 Estrutura do Documento

O documento em questão foi estruturado em diversos capítulos para simplificar a leitura e compreensão dos conceitos abordados. Inicialmente, no primeiro capítulo, realiza-se uma contextualização do tema em análise, proporcionando uma visão sobre o domínio do problema, seguido de uma exposição da motivação subjacente a este trabalho. No Capítulo 2, são expostos os conceitos relevantes ao estado da arte desta dissertação, abrangendo a problemática da poluição da água e a aplicação de técnicas de IA, mais aprofundado na área do DL. Neste capítulo, é também realizada uma revisão bibliográfica com o propósito de investigar trabalhos prévios relacionados com a temática da dissertação, incluindo uma análise crítica dos mesmos. O capítulo subsequente, o Capítulo 3, tem como objetivo apresentar os materiais e métodos utilizados neste trabalho, focando-se na exploração dos *datasets* utilizados, apresentação dos *targets* de estudo e no tratamento e preparação dos mesmos para as experiências. Em seguida, surge o Capítulo 4, onde é apresentado o processo de conceção dos modelos de DL, bem como a identificação das métricas

de avaliação e da metodologia utilizada para efetuar todas as experiências. No Capítulo 5, são apresentados e analisados os resultados obtidos, onde se faz a análise e discussão dos mesmos, bem como a comparação entre os melhores modelos candidatos obtidos. Por fim, no Capítulo 6, é desenvolvida uma conclusão face a todo o trabalho elaborado nesta dissertação. Além disso, são apresentadas medidas relevantes para aplicação em trabalhos futuros, relacionadas com a temática desta dissertação.

Capítulo 2

Estado de Arte

Neste capítulo, procede-se a uma análise dos conceitos relacionados com o estado da arte desta dissertação. Inicialmente, são explorados diferentes conceitos, desde a Poluição da Água até às ETAR, em 2.1, esta secção aborda ainda o funcionamento de uma ETAR. Por seguinte, dá-se entrada a conceitos relacionados com IA, 2.2, que seguidamente são mencionados e explorados tópicos como ML, e DL. Apresenta-se também uma revisão de literatura, complementada com a análise crítica de alguns artigos de investigação incluídos na temática desta dissertação, em 2.3.

2.1 Poluição da Água

Num contexto global onde a necessidade e a importância da água são inquestionáveis, este recurso torna-se vital para o suporte da vida, desempenhando um papel crucial na saúde humana, na garantia de segurança alimentar e na preservação dos ecossistemas [48]. A crescente procura por água, aliada à ameaça persistente da poluição, destaca a urgência de abordar a gestão sustentável desse recurso essencial [48].

A poluição da água e os seus impactos, como supramencionado, representam uma ameaça significativa tanto à saúde pública quanto ao meio ambiente. A poluição da água não é apenas uma preocupação ambiental é uma questão crítica de saúde pública [22]. Neste cenário desafiador, a necessidade de tratamento de águas residuais emerge como uma resposta crucial para combater a poluição hídrica. As ETAR assumem uma posição proeminente nesse contexto. Ao remover poluentes, sólidos suspensos, microrganismos patogénicos e substâncias químicas prejudiciais das águas residuais, as ETAR desempenham um papel crítico na prevenção da poluição da água [49].

2.1.1 ETAR

As ETAR desempenham um papel vital no ciclo urbano da água, que representa uma infraestrutura essencial na gestão eficiente e responsável dos recursos hídricos [49]. Ao integrar-se nesse ciclo, as ETAR contribuem significativamente para a preservação da qualidade da água, promovendo a sustentabilidade ambiental e protegendo a saúde pública. Em áreas urbanas densamente povoadas, onde a concentração de poluentes é significativa, as ETAR destacam-se como defensoras-chave na prevenção direta da libertação de poluentes no meio ambiente [50]. A falta de tratamento adequado intensificaria os problemas de poluição da água, agravando os riscos para a saúde pública e o equilíbrio dos ecossistemas.

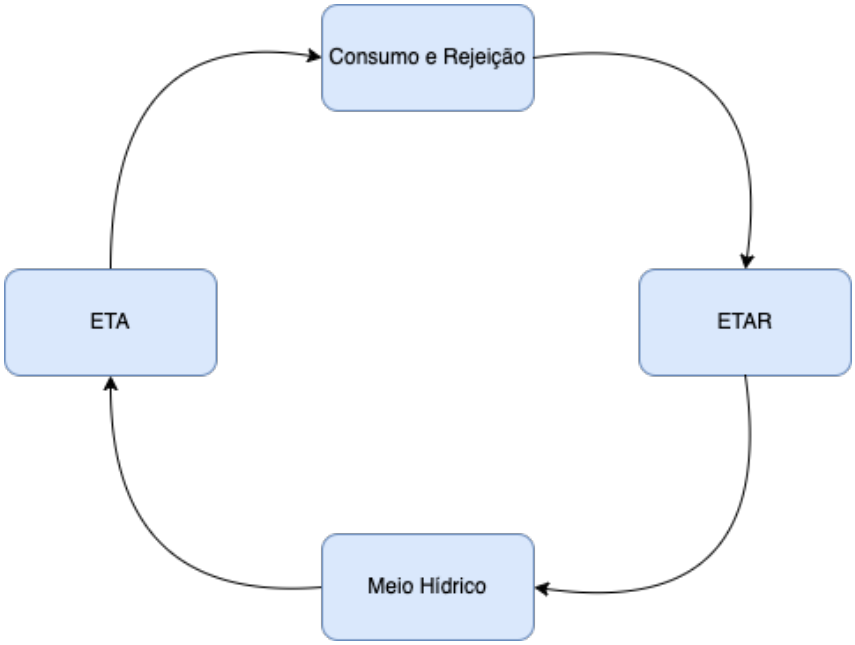


Figura 2.1: Ciclo Urbano, adaptado de [51]

Como ilustrado na Figura 2.1, o ciclo de vida da água envolve um conjunto de processos entre o meio hídrico, as Estação de Tratamento de Águas (ETA) e as ETAR. Esse ciclo dinâmico é fundamental para garantir a qualidade da água desde a sua captação até à devolução ao meio ambiente, assegurando a sustentabilidade, tanto do ecossistema quanto de todos os seres que dependem desse recurso vital [52].

No contexto do ciclo de vida da água, o processo inicia-se com a captação da mesma do meio hídrico, podendo esta ocorrer de duas formas: à superfície, isto é, em rios e albufeiras, ou no subsolo, através de furos ou poços [45]. O que pode levar a envolver atividades de elevação, como a bombagem, para transportar a água de altitudes mais baixas para altitudes superiores, algo muito comum em terrenos de casas antigas [45].

Posteriormente, a água é encaminhada para a ETA, onde se realiza uma correção abrangente das suas características físicas, químicas e bacteriológicas, tornando-a adequada para consumo humano [45]. Durante este processo, são utilizados diversos tratamentos essenciais, como gradagem para remoção de impurezas, coagulação e floculação, decantação, filtração e desinfeção [53] [54]. Adicionalmente, a gestão eficiente das lamas removidas na fase de tratamento é fundamental. A distribuição da água tratada para

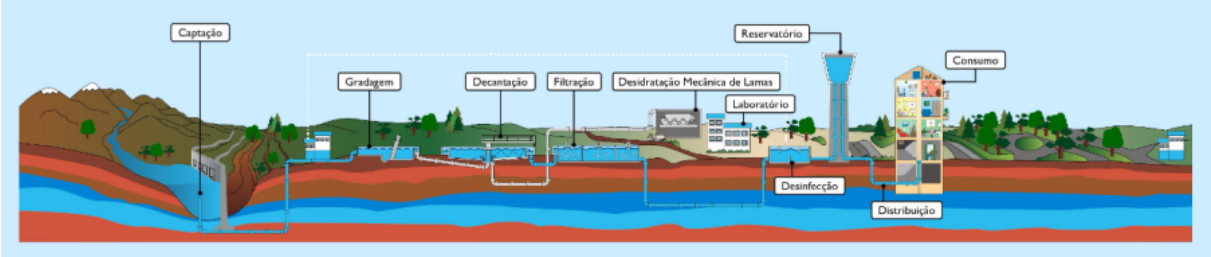


Figura 2.2: Passagem da água do meio ambiente para habitações, retirado de [51]

as zonas de consumo, mantendo a pressão e qualidade adequadas, constitui a etapa seguinte, seguida pela recolha das águas residuais provenientes do consumo humano e atividades produtivas, como se pode ver na Figura 2.2.

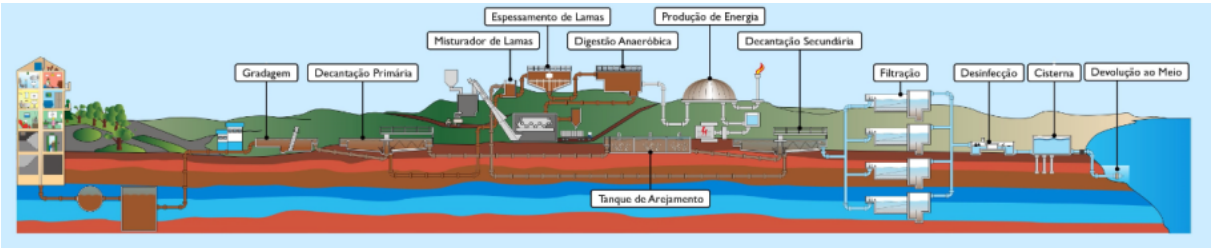


Figura 2.3: Passagem da água das habitações para o meio ambiente, retirado de [51]

As águas residuais são direcionadas para as ETAR, onde são tratadas para serem devolvidas à natureza em condições ambientalmente seguras através de um conjunto de processos que garantem a qualidade da água [55]. No processo de tratamento de águas residuais, as ETAR realizam etapas como o tratamento preliminar (gradagem), tratamento primário (separação sólido-líquida), tratamento secundário (tratamento biológico), tratamento terciário (desinfeção e remoção de nutrientes) e o tratamento de lamas [53]. Após a conclusão destes processos nas ETAR, parte da água tratada é reutilizada para atividades como rega e lavagem, enquanto o restante é devolvido ao meio ambiente, assegurando a reposição nos meios hídricos sem comprometer a saúde pública e a integridade dos ecossistemas e de todos os seres nela habitam [53], tal como visualizado na Figura 2.3.

Este ciclo dinâmico, como representado na Figura 2.1, destaca a importância dos sistemas de tratamento na garantia da qualidade da água, na preservação dos recursos hídricos e na proteção da saúde pública e do meio ambiente [52].

2.1.2 Funcionamento de uma ETAR

Tal como supramencionado, a operação de uma ETAR envolve uma série de processos sequenciais, cada um desempenhando um papel crucial na purificação eficiente das águas residuais, como demonstrado na Figura 2.4

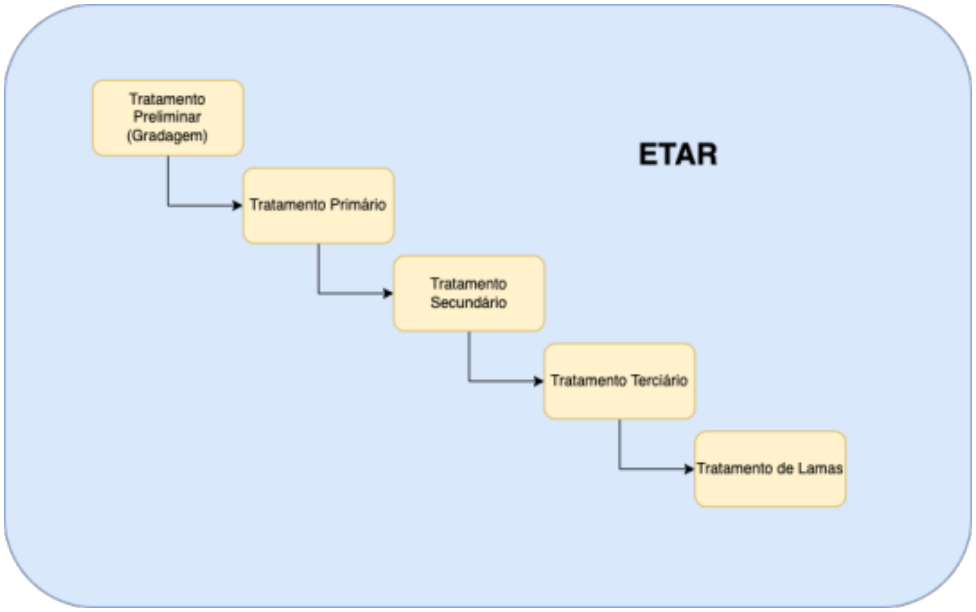


Figura 2.4: Funcionamento Geral de uma ETAR, adaptado de [51]

- **Tratamento Preliminar (Gradagem):** Nesta fase inicial, o tratamento preliminar, conhecido como gradagem, tem como objetivo a remoção dos sólidos de maiores dimensões presentes nas águas residuais que chegam à ETAR [56]. Esses sólidos podem incluir resíduos como papel higiénico, cotonetes, algodão, restos de comida, entre outros. A remoção desses elementos grosseiros é essencial para prevenir obstruções nos sistemas e garantir o funcionamento eficaz dos processos subsequentes [53].

- **Tratamento Primário:** No tratamento primário é realizada uma separação sólido-líquida que permite a remoção de uma quantidade significativa dos sólidos suspensos presentes na água. Os sólidos sedimentados no decantador primário, conhecidos como lamas primárias, são retirados e encaminhados para a linha de tratamento de lamas [53] [56].

- **Tratamento Secundário:** As águas residuais, após parcialmente tratadas, são submetidas a um tratamento biológico no estágio secundário. Neste processo as bactérias são introduzidas para digerir a matéria orgânica presente na água. As lamas biológicas formadas durante este processo depositam-se no fundo dos tanques, como resultado a água mais limpa permanece à superfície. Essas lamas biológicas são então também direcionadas para a linha de tratamento de lamas [53] [56].

- **Tratamento de Lamas:** As lamas resultantes dos estágios anteriores são tratadas adequadamente para o seu destino final. Esse tratamento pode envolver processos de espessamento e desidratação, preparando as lamas para o seu destino final ou para valorização energética (como combustível) e agrícola (como fertilizante). O tratamento de lamas é fundamental para lidar de maneira responsável com os resíduos gerados no processo de tratamento de águas residuais. É neste contexto que surge o reator de digestão anaeróbia [53] [56].

2.1.3 Reator de Digestão Anaeróbio

O reator do digestor anaeróbio, peça crucial no tratamento de lamas em ETAR, opera com base em princípios fundamentais que visam a decomposição eficaz da matéria orgânica, promovendo uma abordagem ambientalmente responsável [47]. O cerne do seu funcionamento reside na utilização do processo anaeróbio, no qual a decomposição ocorre na ausência de oxigénio [47].

2.1.4 Princípio de Funcionamento

O funcionamento do reator do digestor anaeróbio é focalizado no processo anaeróbio, no qual a ausência de oxigénio é crucial. Este ambiente anaeróbio propicia a atividade bacteriana específica que conduz à decomposição eficiente da matéria orgânica presente nas lamas [57]. A ausência de oxigénio é essencial para promover a ação de bactérias anaeróbias, desencadeando um processo de digestão que transforma a matéria orgânica complexa em componentes mais simples [57], tal como retratado na Figura 2.5.

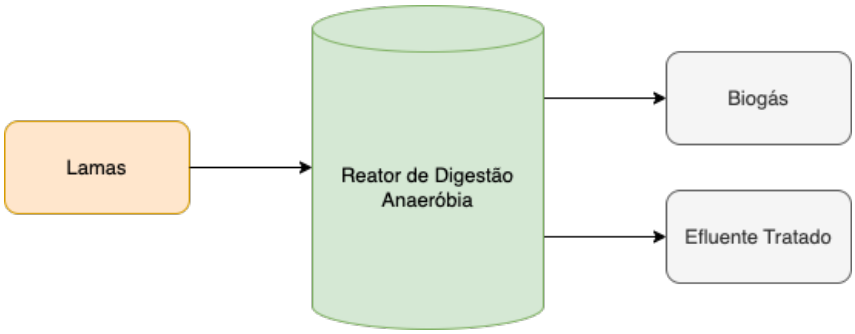


Figura 2.5: Princípio de Funcionamento, adaptado de [58]

2.1.5 Transformação Biológica

No ambiente anaeróbio do reator, as bactérias anaeróbias iniciam uma reação bioquímica, que converte a matéria orgânica complexa presente nas lamas em substâncias mais simples, como ácidos orgânicos e gases. Esta metamorfose biológica é essencial para aliviar a carga poluente das águas residuais, o que torna mais simples as complexas estruturas moleculares dos resíduos orgânicos [58].

Ao converter componentes complexos em formas mais simples, as bactérias anaeróbias desempenham um papel instrumental na redução da complexidade orgânica, tornando os resíduos mais acessíveis aos processos subsequentes de tratamento [58]. Esta transformação é um passo crucial na jornada de purificação das águas residuais, garantindo que a carga orgânica seja minimizada de forma eficiente, promovendo assim a eficácia e a sustentabilidade do tratamento global [58].

2.1.6 Eficiência na Redução de Volume

O reator do digestor anaeróbio desempenha um papel crucial na redução do volume das lamas, conferindo eficiência aos processos subsequentes de tratamento e disposição final [59]. Ao decompor a matéria

orgânica presente nas lamas, o reator desempenha um papel central na redução da presença de sólidos orgânicos [60]. Esta redução de volume não simplifica só a logística de controlo e tratamento das lamas, mas também promove uma gestão mais eficaz dos recursos [60]. A diminuição significativa dos resíduos sólidos não apenas alivia a carga sobre os sistemas de tratamento, mas também contribui para a minimização da pegada ambiental associada à gestão de resíduos sólidos [61].

2.1.7 Importância da Digestão Anaeróbia

A digestão anaeróbia, além de ser uma das maiores responsáveis pelaa redução expressiva da quantidade de resíduos sólidos presentes nas lamas, emerge como uma valiosa produtora de biogás [60] [62] [63]. Este subproduto, predominantemente composto por metano, não é apenas uma fonte de energia renovável, mas representa um elo essencial na transição para práticas mais sustentáveis [62]. Ao converter resíduos considerados inicialmente como problemáticos em recursos úteis, o biogás originado durante o processo de digestão anaeróbia contribui significativamente para atenuar o impacto ambiental associado à gestão de resíduos, e ajuda significativamente outras áreas [64]:

- **Setor Económico:** A utilização do biogás oferece uma série de benefícios económicos e ambientais [65]. Os produtores podem utilizar a energia térmica ou elétrica gerada pelo biogás para alimentar suas propriedades ou empreendimentos, reduzindo significativamente a dependência de eletricidade convencional ou lenha [66]. Além disso, ao fornecer o excedente para a rede elétrica, os produtores podem obter créditos, contribuindo para uma economia mais sustentável. A produção de biocombustíveis, provenientes dos reatores anaeróbios, desempenha um papel vital no abastecimento de veículos movidos a Gás Natural Veicular, promovendo assim uma alternativa mais ecológica no setor de transporte [67].

- **Proteção Ambiental:** A produção de biogás não transforma apenas resíduos em recursos úteis, mas também desempenha um papel crucial na proteção do meio ambiente. Os dejetos gerados na criação de animais, ao serem utilizados no processo de digestão anaeróbia, deixam de representar uma ameaça à saúde do solo, açudes, rios e lençóis freáticos [63]. Ao evitar a contaminação desses recursos hídricos, contribui-se diretamente para a preservação da biodiversidade e a manutenção do equilíbrio ambiental.

- **Sustentabilidade:** Ao adotar práticas sustentáveis, como a utilização de biofertilizantes provenientes do processo anaeróbio, o produtor contribui para a prevenção de enfermidades decorrentes da contaminação ambiental [68]. Essa abordagem holística, que une benefícios económicos, ambientais e de saúde, destaca a importância crescente do biogás como uma solução multifacetada para os desafios contemporâneos.

2.1.8 Parâmetros do Reator

Os reatores de digestão anaeróbia são geridos e monitorizados através de uma variedade de parâmetros cruciais para assegurar a eficiência do processo [69]. Alguns dos parâmetros chave incluem:

- **Tempo de Retenção Hidráulico (TRH):** Este parâmetro refere-se ao tempo que os substratos permanecem no reator, influenciando a taxa de degradação dos resíduos orgânicos [70]. O controle adequado do TRH é essencial para otimizar a eficiência da digestão anaeróbia.
- **pH:** O nível de acidez ou alcalinidade no reator é crítico para a atividade microbiana. Manter um pH adequado é essencial para garantir a estabilidade do processo, pois diferentes grupos de microrganismos têm faixas de pH ideais [71] [69].
- **Relação Carbono/Nitrogênio:** A proporção entre carbono e nitrogênio nos substratos afeta a eficácia da digestão. Um equilíbrio adequado entre esses elementos é fundamental para promover condições favoráveis à atividade microbiana [69].
- **Temperatura:** A temperatura é um fator crucial, pois influencia a velocidade das reações bioquímicas. Diferentes microrganismos têm faixas de temperatura ideais, e manter a temperatura controlada é vital para garantir um processo eficiente [70] [69].
- **Taxa de Alimentação Orgânica (OFR):** Refere-se à quantidade de substrato orgânico adicionada ao reator ao longo do tempo. Controlar a OFR é essencial para evitar sobrecarga do sistema e garantir uma digestão eficaz [72].
- **Concentração de Sólidos Voláteis Totais (SVT):** Este parâmetro fornece informações sobre a quantidade de matéria orgânica disponível para a digestão. Monitorizar a concentração de SVT ajuda a ajustar a alimentação e a manter um equilíbrio adequado no reator [73].
- **Taxa de Produção de Biogás:** A quantidade e a qualidade do biogás produzido são indicadores importantes da eficiência do processo. Monitorizar a taxa de produção de biogás permite ajustes operacionais para otimizar a produção de energia renovável, tais como, CH4 e Xc [74].

2.2 Inteligência Artificial

A IA refere-se à inteligência demonstrada por máquinas, no âmbito das ciências da computação, é principalmente caracterizada como o estudo de "agentes inteligentes", isto é, qualquer dispositivo capaz de perceber o ambiente ao seu redor e tomar ações que aumentem as suas probabilidades de atingir um determinado objetivo [75] [76]. A IA representa o culminar de décadas de avanços tecnológicos, que marcou uma era revolucionária na interseção entre as ciências da computação e a cognição humana [77]. Ao longo dos anos, foi constatada uma evolução notável, desde simples algoritmos até sistemas sofisticados capazes de aprendizagem autónoma [77]. Esta revolução tecnológica resulta de décadas de pesquisa e

desenvolvimento que atravessam diversas áreas, redefinindo fundamentalmente a forma como as pessoas interagem com o mundo à sua volta. Este crescimento acelerado é impulsionado por avanços tecnológicos, algoritmos mais avançados e um aumento massivo na disponibilidade de dados [78] [79].

A presença impactante IA estende-se para além das fronteiras, insere-se de forma integral em setores estratégicos. A IA revela-se multifacetada, desdobrando-se em aplicações que abraçam desde a automação industrial até à medicina e às decisões empresariais, conferindo uma nova dimensão ás operações quotidianas [79]. No domínio da saúde, foram realizados avanços notáveis, com sistemas de IA a desempenhar papéis cruciais na realização de diagnósticos rápidos e precisos [80]. Estes sistemas além de identificarem patologias de forma eficiente, também propõem tratamentos personalizados, o que por sua vez marca uma transição significativa na abordagem à medicina [81]. Adicionalmente, a capacidade de antecipar surtos epidemiológicos destaca a utilidade da IA na prevenção e gestão de crises de saúde [81]. No cenário económico, a presença de algoritmos avançados revelam-se importantes nas decisões financeiras, proporcionando análises de riscos detalhadas e contribuindo para a deteção eficaz de fraudes [82]. A IA tornou-se cada vez mais uma aliada indispensável para instituições financeiras, introduzindo uma camada adicional de precisão e eficiência nas operações diárias [83]. Para além dos setores já mencionados, a IA é um dos grandes responsáveis pela remodelação de setores tradicionais, como educação, agricultura e transporte, proporcionando melhorias significativas em eficiência e processos decisórios.e segurança, redefinindo as normas do setor [84] [85].

A integração da IA no quotidiano é inegavelmente tangível, manifestando-se numa panóplia de formas que transformam as rotinas diárias [86]. Desde assistentes virtuais, que descomplicam as tarefas diárias, até algoritmos de recomendação que personalizam meticulosamente a nossa experiência online, a presença da IA tece-se de modo intrincado no tecido do quotidiano [86]. Os assistentes virtuais, por exemplo, têm evoluído para serem verdadeiros colaboradores nas tarefas diárias, desde a marcação de compromissos, até à gestão de listas de afazeres. Estes agentes virtuais, impulsionados pela IA, não só simplificam as nossas atividades, mas também aprendem com os padrões de comportamento, aprimorando continuamente a sua utilidade [87]. Além disso, os algoritmos de recomendação, presentes em plataformas online, são uma expressão sofisticada da IA, moldando as interações digitais de forma personalizada. Desde sugestões de filmes, até produtos de compras online, estes algoritmos analisam os nossos gostos e preferências, proporcionando uma experiência mais cativante e relevante [88]. Tal como é possível observar na Figura 2.6, atualmente existe um leque enorme de áreas onde a IA está bastante presente [88].

Apesar das oportunidades únicas que a IA oferece, o seu avanço não está isento de desafios. O algorítmo pode tornar-se tendencioso, por exemplo, pode perpetuar discriminações existentes na sociedade [89]. A privacidade torna-se uma preocupação à medida que a IA processa e utiliza grandes volumes de dados pessoais [90]. Além disso, a automatização intensiva pode levar à substituição de empregos, suscitando questões sobre a equidade no acesso a oportunidades económicas [90].

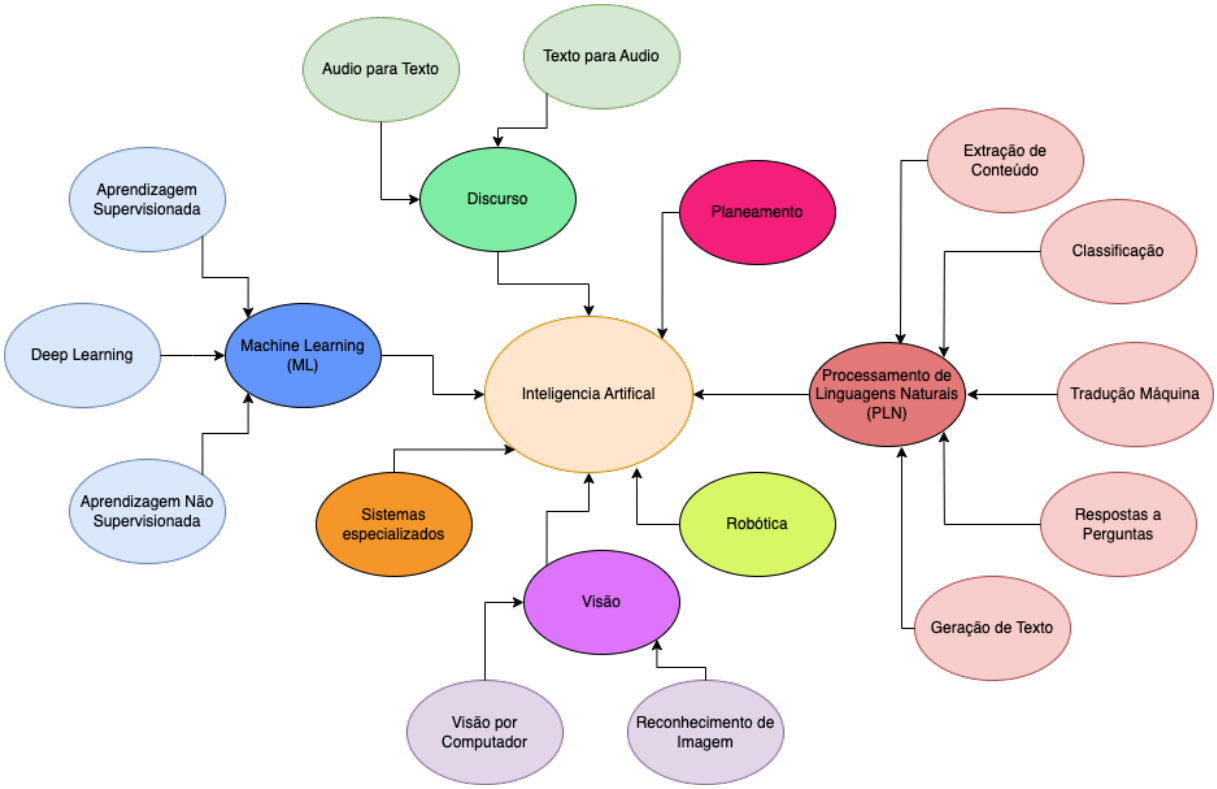


Figura 2.6: Subcampos da IA, adaptado de [91]

2.2.1 Machine Learning

O ML representa um subcampo crucial da IA. De um modo geral é a capacidade de dotar os computadores com a habilidade de aprender sem necessidade de programação explícita [76]. Tendo origem no estudo inicial sobre reconhecimento de padrões e teoria computacional de aprendizagem no âmbito da IA, o ML foca-se na criação de algoritmos capazes de aprender com dados e efetuar previsões [76]. Estes algoritmos transcendem as instruções de programas estáticos, tomando decisões e realizando previsões orientadas por dados, construindo modelos a partir de amostras de entrada [76].

Aplicado a uma diversidade de tarefas computacionais, o ML destaca-se em situações onde desenhar e programar algoritmos explícitos com desempenho eficaz se revela difícil ou impraticável [76]. Um dos vastos exemplos é a filtragem de e-mails, a deteção de intrusos em redes, a identificação de colaboradores maliciosos em violações de dados, o reconhecimento ótico de caracteres, bem como tarefas de aprendizagem para classificação e visão computacional [76].

Além disso, o ML foca-se numa abordagem centrada na aprendizagem a partir de dados, permitindo que os sistemas evoluam e melhorem o seu desempenho à medida que são expostos a conjuntos mais alargados de informações [88]. Diferentemente da programação tradicional, os modelos de aprendizagem máquina são concebidos para se adaptarem e aprimorarem com o tempo, representando um paradigma inovador no desenvolvimento de sistemas inteligentes [88].

Nos algoritmos de ML, é possível expressar a relação entre o input (entrada) e o output (saída) como a construção de um modelo. O processo de ML envolve alimentar o algoritmo com um conjunto de dados

de entrada, que inclui as características relevantes (input) e as saídas desejadas (output). O algoritmo utiliza esses dados para aprender padrões e relações, construindo assim um modelo que pode ser posteriormente utilizado para fazer previsões ou tomar decisões com base em novos dados não vistos, como demonstrado na Figura 2.7 [88]:

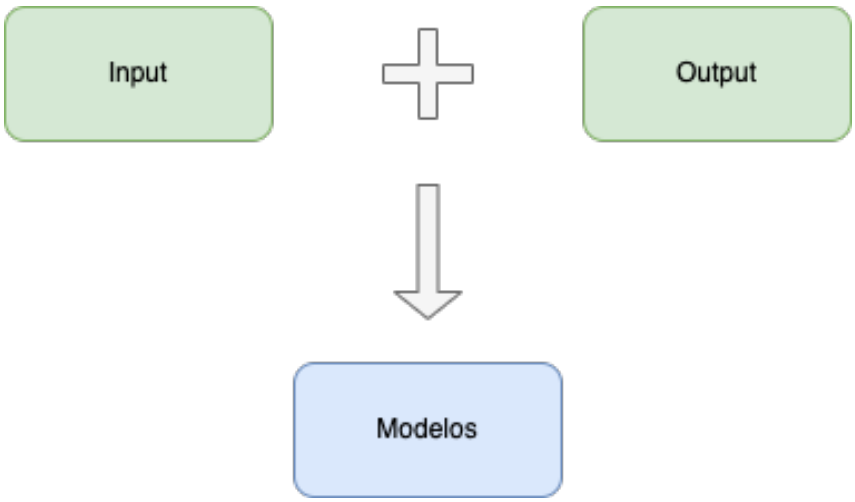


Figura 2.7: Algoritmo de ML adaptado de: [88]

É então possível reconhecer 3 princípios base:

- **Aprendizagem a partir de Dados:** O cerne do ML reside na capacidade de os algoritmos aprenderem padrões e informações úteis a partir dos dados fornecidos. Quanto mais dados relevantes são disponibilizados, maior é capacidade do modelo se ajustar e fazer previsões/tomar decisões [92].

- **Algoritmos e Modelos:** Diferentes algoritmos de ML são utilizados para tarefas específicas, como classificação, regressão, agrupamento, entre outras. Esses algoritmos formam modelos que representam as relações e padrões identificados nos dados [93].

- **Treino e Teste:** O processo de treino envolve alimentar o algoritmo com dados rotulados, permitindo que ele aprenda a fazer previsões ou tomar decisões. Posteriormente, o modelo é testado com dados não vistos para avaliar sua capacidade de generalização [94].

O campo de ML engloba diversas abordagens de aprendizagem, sendo notáveis os quatro principais tipos: aprendizagem supervisionada, aprendizagem não supervisionada, aprendizagem semi-supervisionada e aprendizagem por reforço, como visto na Figura 2.8. Cada um desses paradigmas desempenha um papel distintivo no desenvolvimento de algoritmos capazes de aprender a partir de dados.



Figura 2.8: Tipos de Aprendizagem, adaptado de [76]

2.2.1.1 Aprendizagem Supervisionada

A aprendizagem supervisionada representa um paradigma essencial no campo do ML, caracterizado pela utilização de casos de treino que incorporam informação acerca dos resultados desejados, permitindo estabelecer uma relação entre os valores pretendidos e os valores produzidos pelo sistema [76]. A vasta maioria dos algoritmos de ML adota este método, cerca de 70%, que se revela fundamental para a capacidade de aprendizagem do sistema [76].

Este modelo implica que os dados de entrada (x) e os resultados (y) estejam correlacionados, possibilitando ao algoritmo aprender uma função (f) que mapeia os dados nos resultados: y=f(x). A designação supervisionada advém do acompanhamento do processo de aprendizagem por um algoritmo que monitora o mapeamento, havendo então uma necessidade de intervenção humana neste processo. Esta abordagem é geralmente dividida em duas categorias principais [95]:

- **Classificação:** Quando os resultados são discretos, como por exemplo, categorias como preto, branco ou cinza [95].
- **Regressão:** Quando os resultados são contínuos, abrangendo variáveis como preço, temperatura, idade, entre outros [95].

No contexto da aprendizagem supervisionada, os algoritmos são treinados utilizando exemplos rotulados, onde as entradas são associadas a saídas desejadas conhecidas [95]. Por exemplo, um equipamento pode ter dados rotulados como "F"(falha) ou "R"(funciona). O algoritmo compara a sua saída com os resultados corretos, ajustando o modelo conforme necessário. Este tipo de aprendizagem é comum em aplicações onde dados históricos são utilizados para prever eventos futuros, como a deteção de transações fraudulentas em cartões de crédito ou a avaliação da propensão de um cliente de seguro a apresentar uma reclamação [95].

2.2.1.2 Aprendizagem Não Supervisionada

A aprendizagem não supervisionada é por sua vez um algoritmo utilizado com bastante frequência, representando cerca de 10% a 20% dos casos [76]. É um paradigma que no qual não são conhecidos resultados sobre os casos, apenas os enunciados dos problemas. Nesse contexto, torna-se imperativo escolher técnicas de aprendizagem que avaliem o funcionamento interno do sistema, explorando a estrutura e a distribuição dos dados [76].

Neste tipo de aprendizagem, existem dados de entrada (x), mas não há conhecimento prévio dos resultados correspondentes. O principal objetivo é modelar a estrutura subjacente ou a distribuição dos dados do problema, sem a orientação explícita de resultados conhecidos [96]. Existem três categorias principais de aprendizagem não supervisionada [96]:

- **Segmentação (Clustering):** Organiza os dados em grupos coerentes, agrupando elementos que compartilham características semelhantes. Por exemplo, agrupar clientes que preferem produtos biológicos, formando segmentos distintos [96].
- **Redução (Reduction):** Reduz o número de características em um conjunto de dados ou decompõe o conjunto em múltiplos componentes, simplificando a representação. Por exemplo, reduzir a dimensionalidade de um conjunto de dados ou decompor características complexas [96].
- **Associação:** Descobre regras que associam o comportamento demonstrado pelos dados, identificando relações intrínsecas. Como por exemplo, identificar padrões de compra, como pessoas que compram produtos biológicos geralmente não compram produtos de charcutaria [96].

A aprendizagem não supervisionada é valiosa em situações onde se pretende explorar a estrutura interna dos dados, sem depender de rótulos históricos [76]. Técnicas populares incluem mapas auto-organizáveis, mapeamento de vizinhos mais próximos, agrupamento k-means e decomposição de valor singular [76]. Esses algoritmos não apenas auxiliam na segmentação de dados transacionais, mas também são aplicados em tarefas como segmentação de tópicos de texto, recomendação de itens e deteção de valores atípicos nos dados [76].

2.2.1.3 Aprendizagem Semi-Supervisionada

A aprendizagem semi-supervisionada representa uma abordagem híbrida, combinando elementos da aprendizagem supervisionada e não supervisionada [76]. Este paradigma opera tanto em dados rotulados quanto não rotulados. Esta técnica desempenha um papel fundamental em situações em que há escassez de dados rotulados, mas uma abundância de dados sem rótulos. Esta estratégia oferece vantagens significativas, melhorando o desempenho do modelo ao incorporar informações de ambas as fontes de dados [76].

2.2.1.4 Aprendizagem por Reforço

O paradigma de aprendizagem por reforço é uma abordagem única, que se destaca pela sua capacidade de permitir que um sistema avalie a qualidade dos seus resultados, mesmo sem conhecimento prévio dos resultados desejados [76]. Diferentemente da aprendizagem supervisionada, que se baseia em avaliações de um supervisor, e da aprendizagem não supervisionada, que opera sem informação sobre os resultados, a aprendizagem por reforço envolve a avaliação crítica interna dos resultados produzidos pelo algoritmo [76].

A aprendizagem dá-se pela capacidade de crítica sobre os próprios resultados produzidos pelo algoritmo. Por exemplo, O Q-Learning, ao assumir a adoção de uma política ótima, utiliza-a para ajustar os valores das ações [97]. Por outro lado, o SARSA considera a política de controlo em execução e procede à atualização dos valores das ações [98].

Este tipo de aprendizagem destaca-se em diversas aplicações, sendo especialmente relevante em campos como robótica, jogos e navegação [99]. Neste paradigma, o algoritmo, inserido em um ambiente dinâmico, aprimora seus conhecimentos por meio de uma abordagem de tentativa e erro, identificando quais ações conduzem a recompensas mais substanciais.

Além das estratégias de aprendizagem mencionadas anteriormente e das distintas formas de abordar o processo, surge uma técnica proeminente que redefine os limites do ML: o DL.

2.2.2 Deep Learning

A área DL é uma subárea de ML, destaca-se pelo foco no treino de algoritmos, com especial ênfase nas RNA, visando a realização autónoma de tarefas complexas [76]. O termo ”deep”refere-se à característica distintiva dessas redes, compreendendo múltiplas camadas que transcendem as abordagens mais convencionais [76]. A característica fundamental do DL é a capacidade de aprender automaticamente representações hierárquicas e abstratas dos dados [76]. Essas representações são construídas em diferentes níveis de complexidade, permitindo que a rede aprenda características desde as mais simples, até as mais sofisticadas. A estrutura em camadas das redes neurais profundas possibilita a extração de características e padrões complexos, o que é especialmente útil em tarefas como reconhecimento de padrões, processamento de linguagem natural, visão computacional e outras aplicações complexas [76].

As RNA constituem a espinha dorsal do DL, sendo um modelo inspirado na estrutura neuronal do cérebro humano, composta por neurónios artificiais interligados [76]. A sua estrutura computacional complexa é organizada em camadas, incluindo uma camada de entrada, uma ou várias camadas ocultas e uma camada de saída. Dentro do panorama das RNA, destacam-se as Deep Neural Network (DNN), que transcendem as RNA tradicionais, caracterizadas pela presença de múltiplas camadas ocultas entre a camada de entrada e a camada de saída. Quanto mais camadas ocultas uma rede tiver, mais profunda elas são consideradas. Portanto, uma DNN é, por definição, uma RNA com mais de uma camada oculta [76]. Essa sobreposição de camadas confere às DNN uma estrutura hierárquica que simula níveis de abstração, sendo cada camada responsável por extrair atributos específicos dos dados [76]. Essa hierarquia de

conceitos é essencial para a aprendizagem, permitindo a realização de tarefas tanto de forma supervisionada quanto não supervisionada. A capacidade de extração hierárquica de características posiciona as DNN como pilares fundamentais para a realização de tarefas complexas no âmbito do DL [76].

2.2.2.1 RNA

Como supramencionado, as RNA são estruturas que procuram simular o funcionamento das redes neuronais presentes no cérebro humano [76]. A sua arquitetura básica é composta por diversas camadas interconectadas. Cada camada é composta por neurónios artificiais, e a interligação entre eles é realizada através de conexões sinápticas [76].

Na camada de entrada, os neurónios recebem os dados iniciais, que podem representar características de um objeto, valores numéricos, ou qualquer tipo de informação relevante para a tarefa em questão [100]. As camadas ocultas, por sua vez, são responsáveis por processar esses dados de forma a aprender padrões e relações mais complexas. Finalmente, na camada de saída, os neurónios produzem o resultado final da rede, que pode ser uma classificação, previsão ou qualquer outra saída desejada [100], tal como representado na Figura 2.9.

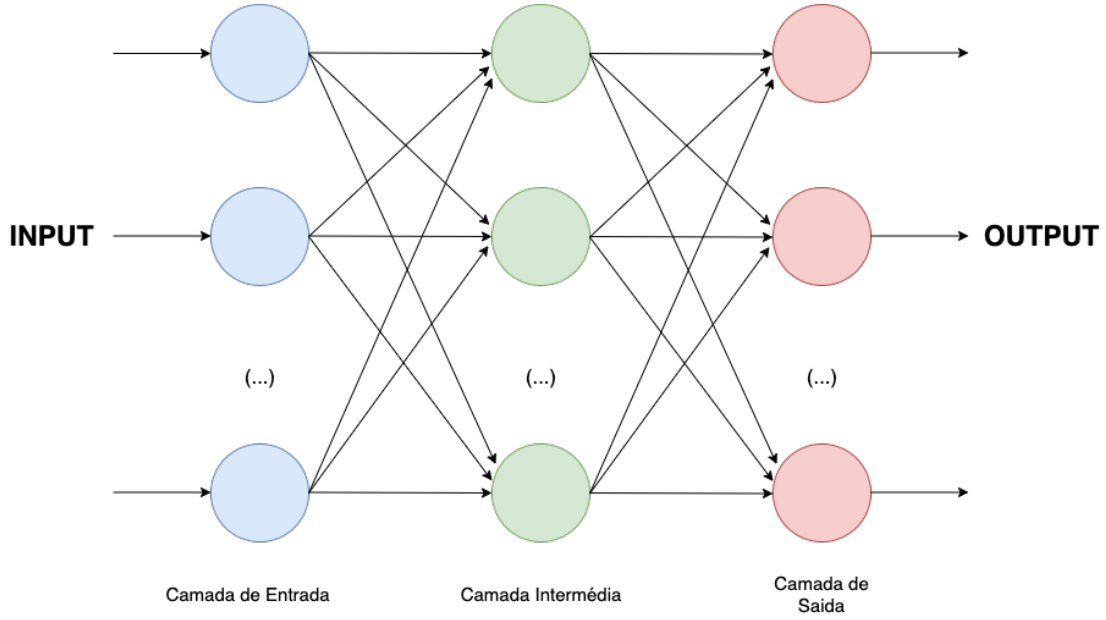


Figura 2.9: Estrutura de uma RNA, adaptado de [100]

A arquitetura em camadas permite que as RNA aprendam representações hierárquicas dos dados, extraindo características desde as mais simples até as mais abstratas. A cada conexão sináptica é atribuída a um peso que é ajustado durante o treino, permitindo que haja uma adaptação dinâmica [100]. A RNA realiza operações ponderadas nos inputs e aplica funções de ativação para produzir uma saída. A capacidade de extrair relações entre atributos e organizar dados em grupos através de técnicas, como *clustering*, destaca-se como uma das características distintivas das RNA [100].

O funcionamento das RNA envolve dois processos cruciais: propagação direta (*forward propagation*)

e aprendizagem por ajuste de pesos (*backpropagation*) [100]. Na propagação direta, os dados são introduzidos na camada de entrada e são processados pelas camadas ocultas até chegar à camada de saída. Cada neurónio realiza um cálculo ponderado das suas entradas, adiciona o *bias* e aplica uma função de ativação. A saída de cada neurónio serve como entrada para os neurónios da camada seguinte [100]. No processo de mecanismo de aprendizagem por ajuste de pesos, a diferença entre a saída prevista e a saída real (o erro) é retropropagada pela rede. Os pesos sinápticos são ajustados de acordo com o gradiente descendente, minimizando o erro. Esse processo é repetido iterativamente durante o treino, refinando os pesos para otimizar o desempenho da RNA [100].

No cenário da IA, a distinção entre ML e DL revela nuances cruciais. ML, como subconjunto da IA, destaca-se por treinar em conjuntos de dados menores, exigindo mais intervenção humana para correção [77]. Em contraste, o DL, subconjunto do ML, necessita de grandes volumes de dados, mas oferece uma aprendizagem autónomo e correlações não-lineares, exigindo Graphics Processing Units (GPU) especializadas para treino [77]. A escolha entre ambos depende da complexidade da tarefa, delineando um equilíbrio entre eficiência e profundidade na implementação de soluções de IA [77].

2.2.3 Series Temporais

As séries temporais representam conjuntos de dados onde as observações são registadas em intervalos temporais regulares [101]. Este formato único de dados captura a evolução de variáveis ao longo do tempo, fornecendo *insights* valiosos para análise e previsão.

As séries temporais possuem diversas características, tais como:

- **Ordem Temporal** : Trata-se da disposição sequencial das observações ao longo do tempo, onde a posição relativa de uma observação em relação às outras é crucial para a compreensão e interpretação dos dados [101].

- **Persistência Temporal:** Além da ordem sequencial, as séries temporais refletem uma persistência temporal, onde padrões passados têm impacto nas observações futuras. Essa inércia temporal é crucial para compreender a evolução dinâmica dos dados ao longo do tempo [101].

- **Ruído e Incerteza:** A presença de ruído e incerteza é uma característica inerente, tornando as séries temporais suscetíveis a flutuações imprevisíveis. Lidar com esse ruído é essencial para distinguir padrões significativos de variações aleatórias [101].

- **Comportamento Estocástico:** A natureza estocástica das séries temporais implica que, mesmo com padrões identificáveis, as observações futuras ainda possuem uma dose de aleatoriedade. Modelar esse comportamento estocástico é crucial para prever variações inesperadas [101].

- **Sazonalidade Explícita:** Além das tendências gerais, muitas séries temporais exibem sazonalidades explícitas relacionadas a padrões recorrentes em determinados períodos. A deteção e modelação desses ciclos sazonais são fundamentais para análises precisas [101].

- **Transições e Mudanças de Estado:** Séries temporais podem testemunhar transições e mudanças abruptas de estado. A capacidade de identificar e modelar essas mudanças é crucial para prever eventos, não apenas com base em padrões, mas também em alterações significativas nas condições [101].

- **Intervalos de Confiança Dinâmicos:** Contrariamente aos dados estáticos, as séries temporais exigem intervalos de confiança dinâmicos. A incerteza associada a uma previsão pode variar ao longo do tempo, exigindo abordagens que capturem essa dinâmica de confiança [101].

- **Interconexões Complexas:** As relações entre diferentes componentes de uma série temporal podem ser complexas e interdependentes. Descobrir e modelar essas interconexões é vital para uma compreensão holística e precisa da dinâmica temporal [101].

- **Resiliência a Outliers:** Séries temporais frequentemente apresentam pontos atípicos ou outliers. Desenvolver modelos que sejam resistentes a essas observações extremas é essencial para garantir a robustez das análises e previsões [101].

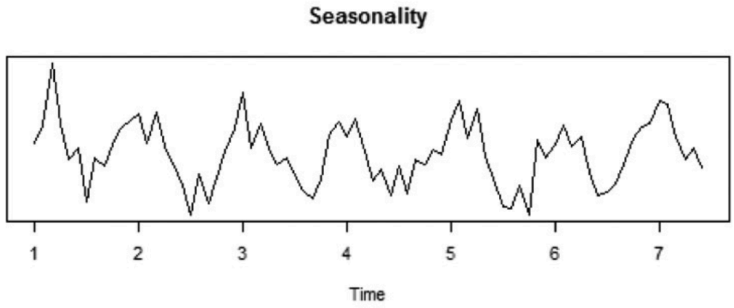


Figura 2.10: Exemplo de comportamento sazonal. Retirado de: [101]

A Figura 2.10 oferece uma representação visual clara do comportamento sazonal em séries temporais. Ao observar o gráfico, destacam-se padrões recorrentes que se repetem em intervalos regulares, sugerindo a presença de variações sazonais ao longo do tempo [101]. A linha de tendência na figura revela picos e vales que ocorrem em padrões previsíveis, indicando uma influência significativa de fatores sazonais. Essas flutuações podem ser associadas a fenómenos cíclicos, como estações do ano, feriados ou eventos recorrentes específicos. A consistência nas oscilações ao longo do eixo temporal destaca a periodicidade do comportamento sazonal. Essa regularidade na variação temporal fornece *insights* valiosos para compreender e antecipar padrões futuros.

A utilidade das séries temporais estende-se por uma variedade de domínios, proporcionando *insights* valiosos em aplicações distintas. Na área financeira, essas séries são essenciais para prever tendências de mercado, antecipar preços de ações e avaliar riscos financeiros, permitindo decisões estratégicas mais informadas [102]. No setor da saúde, a monitorização temporal de parâmetros médicos desempenha um papel crucial, detetando precocemente anomalias e otimizando tratamentos [103]. Por fim, na indústria

e manufatura, a monitorização temporal de processos de produção é vital para detetar falhas em tempo real, melhorando eficiência e qualidade [104].

2.2.4 Problemas de Regressão em Séries Temporais

A modelação de problemas de regressão em séries temporais desempenha um papel fundamental na análise preditiva, permitindo estimar e antecipar valores futuros em sequências temporais contínuas [105]. Esta área tem como objetivo principal decifrar a trajetória evolutiva de uma variável ao longo do tempo, adotando abordagens analíticas que exploram meticulosamente os padrões, tendências e flutuações inerentes aos dados históricos [105]. Em contraste com a previsão de eventos discretos, onde o foco recai sobre eventos específicos, a modelação de problemas de regressão lida com fenómenos contínuos, demandando, por conseguinte, uma análise mais aprofundada das intrincadas relações entre variáveis ao longo da sequência temporal [105].

Neste âmbito, é imperativo compreender a dinâmica temporal subjacente aos dados, uma vez que a evolução de variáveis contínuas pode ser influenciada por uma miríade de fatores interconectados [105]. A utilização de métodos especializados nesta área revela-se essencial para desvendar as complexidades intrínsecas das séries temporais, proporcionando uma visão mais clara e precisa das tendências emergentes. Dessa forma, a análise de padrões históricos, a identificação de tendências significativas e a interpretação de flutuações tornam-se elementos cruciais para o sucesso na previsão de problemas de regressão em séries temporais [105].

2.2.4.1 *Univariate vs. Multivariate*

Nas séries temporais *univariate*, a atenção foca-se exclusivamente numa única variável temporal, permitindo uma análise mais específica e direcionada dessa dimensão temporal singular [105]. Este destaque único é pretendido quando o objetivo é compreender as variações ao longo do tempo de uma única entidade, proporcionando uma visão mais clara e concisa da dinâmica temporal associada a essa variável específica [105]. Por outro lado, em séries temporais *multivariate*, a complexidade aumenta significativamente, uma vez que múltiplas *features* (fatores temporais) entram em jogo de forma simultânea [105]. Essa abordagem abraça a interconexão entre várias variáveis temporais, reconhecendo que as influências entre elas podem ser interdependentes e complexas. A incorporação de vários fatores temporais proporciona uma representação mais abrangente do ambiente temporal em questão [105].

2.2.4.2 *Single-Step vs. Multi-Step*

Um dos aspetos fundamentais nos problemas de previsão no contexto das series temporais é a dualidade entre os *single-step* e os *multi-step*. Em séries temporais *univariate*, a previsão de passo único concentra-se na estimativa do próximo ponto na sequência temporal. Essa abordagem mais pontual e imediata permite uma visão mais detalhada e específica da evolução da variável temporal em questão [106]. Por

outro lado, a previsão de *multi-step* em séries *univariate* antecipa vários pontos adiante na sequência, proporcionando uma visão mais abrangente e aprofundada das possíveis trajetórias futuras. Essa distinção é vital para ajustar a estratégia de previsão de acordo com a complexidade intrínseca de cada série temporal específica [106]. A escolha entre previsão de *single-step* e *multi-step* não é apenas uma questão técnica, mas uma decisão estratégica que considera a dinâmica temporal e a extensão da projeção necessária para atender aos objetivos analíticos [106].

2.2.4.3 *Recursive Multi Step Forecast vs Output Vector*

A abordagem *recursive multi-step* destaca-se pela sua capacidade de se adaptar dinamicamente às alterações na tendência temporal. Essa estratégia incorpora previsões anteriores à medida que avança na sequência temporal, permitindo ao modelo ajustar-se continuamente às nuances e flutuações presentes nos dados [107]. Essa flexibilidade dinâmica é crucial em contextos onde as condições podem evoluir rapidamente, proporcionando uma resposta ágil e ajustada às mudanças nas tendências temporais [107].

Por outro lado, a estratégia de *output vector* oferece uma abordagem mais holística ao considerar simultaneamente a dependência temporal entre os passos futuros [108]. Ao integrar a relação entre diferentes momentos na sequência temporal, essa técnica busca capturar padrões mais complexos e interrelações entre os pontos previstos. Essa visão abrangente é valiosa em cenários onde a interdependência entre os passos futuros é essencial para uma previsão precisa e contextualmente relevante [108].

2.2.5 Modelos *Deep Learning* em Séries Temporais

Atualmente, a inserção de modelos de DL na esfera das séries temporais representa uma nova vertente na análise preditiva. Esse paradigma emergente tem consolidado avanços notáveis na compreensão e previsão de dados temporais de natureza complexa, estabelecendo-se como uma ferramenta indispensável na era da informação [109]. Nesse contexto dinâmico, diversas arquiteturas de modelos despontam, impulsionando a precisão das previsões temporais. Diante dos desafios crescentes e da complexidade dos problemas enfrentados quotidianamente, as vantagens intrínsecas dessas técnicas tornam-se cada vez mais evidentes [109].

A capacidade intrínseca das técnicas de DL em processar grandes volumes de dados desempenha um papel crucial na análise de séries temporais [109]. Ao contrário das abordagens convencionais, que frequentemente enfrentam limitações na manipulação de quantidades expressivas de dados e na representação de relações complexas, as técnicas de DL oferecem uma oportunidade única de lidar com a dinâmica e a complexidade inerentes a esses conjuntos temporais [109].

Entre as arquiteturas de modelos mais destacadas, figuram as RNN, cuja estrutura permite a captura de dependências temporais de longo alcance; as CNN, especializadas em identificar padrões temporais em dados unidimensionais; e por fim as LSTNet e das TCN, ambas conhecidas pela sua capacidade de lidar com séries temporais complexas. Estas abordagens permitiram um melhor desempenho na modelação dos dados temporais e na previsão de valores futuros. [109].

2.2.5.1 *Recurrent Neural Network*

As RNN representam uma contribuição fundamental no campo da análise de séries temporais, destacando-se pelo seu papel crucial na captura de dependências temporais de longo alcance [110]. As mesmas representam uma classe especial de arquiteturas de redes neuronais projetadas para lidar de maneira eficaz com dados sequenciais [110]. Ao contrário das redes neuronais convencionais, as RNN têm a capacidade única de processar informações em sequências, permitindo a modelação de dependências temporais em conjuntos de dados. Esse aspeto torna as RNN particularmente adequadas para lidar com problemas que envolvem padrões temporais e contextos sequenciais [109].

O conceito fundamental por trás das RNN é a introdução de conexões recorrentes, que permitem que as informações sejam transmitidas de um passo de tempo para o próximo [110]. Essa capacidade de reter e utilizar informações anteriores em cada etapa subsequente é crucial para tarefas que exigem uma compreensão contextual do histórico da sequência [110].

A diferença crucial entre as RNN e as redes neurnais convencionais reside na capacidade de lidar com dados que têm uma natureza sequencial ou temporal [109]. Numa rede neuronal convencional, cada entrada é tratada de forma independente, sem levar em consideração a ordem ou relação com observações anteriores. Em contraste, as RNN incorporam uma memória interna que lhes permite manter um estado oculto, capturando informações do passado e influenciando as previsões futuras [109].

A capacidade das RNN de lidar com sequências temporais extensas não vem sem desafios. Um desses desafios é a propagação do gradiente ao longo de sequências temporais extensas, o que pode resultar em dificuldades de aprendizagem e na perda de informações cruciais[109]. Essa limitação levou à evolução e à evolução contínua de arquiteturas mais avançadas, visando superar as barreiras inerentes às RNN e otimizar a eficiência na modelação de séries temporais[109].

O funcionamento das RNN é essencialmente a operação através da aplicação iterativa de uma função aos inputs sequenciais. A característica recorrente dessas redes reside na consideração do output calculado previamente e do input atual durante o cálculo de cada novo output. O que confere à RNN a capacidade de aproveitar o contexto da sequência [111], como visto na Figura 2.11.

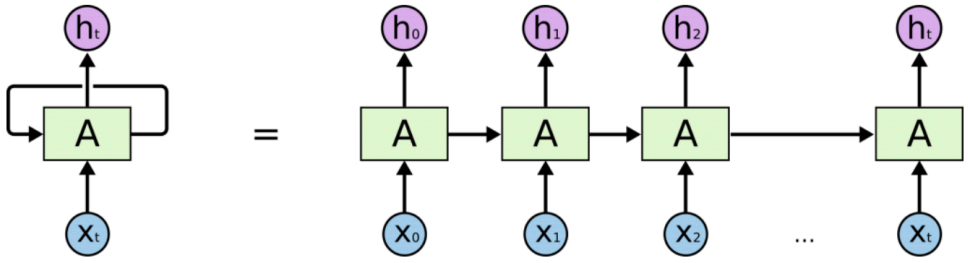


Figura 2.11: Exemplo de arquitetura RNN, adaptado de [111]

No funcionamento das RNN, essa capacidade de contextualização é potencializada pelas funções de

ativação. A função de ativação, como a tangente hiperbólica (tanh), desempenha um papel crucial na computação dos estados ocultos [111]. A fórmula geral para o cálculo do estado oculto h_t em uma RNN com função de ativação *tanh* é [111]:

$$h_t = \tanh(W_{hh} \cdot h_{t-1} + W_{xh} \cdot x_t) \tag{2.1}$$

Essa equação reflete a natureza recorrente da RNN, onde o estado oculto atual (h_t) é calculado considerando o estado oculto anterior (h_{t1}) e o input atual (x_t) [111]. A função *tanh* é responsável por comprimir os valores no intervalo entre -1 e 1, promovendo a estabilidade numérica durante o treino e permitindo que a rede lide eficientemente com a dinâmica temporal. A função *tanh* é valiosa para mitigar o problema do desaparecimento do gradiente, comum em arquiteturas recorrentes [111].

Outra função de ativação frequentemente utilizada é a Rectified Linear Unit (ReLU). Essa função introduz não linearidades ao definir como zero todos os valores negativos, o que pode acelerar o processo de aprendizagem e proporcionar uma maior capacidade de representação [111]. A simplicidade e eficácia da ReLU contribuem para torná-la uma escolha popular em arquiteturas de RNN [111]. No entanto, é importante destacar que, apesar das vantagens das RNN, essas redes podem enfrentar desafios, especialmente relacionados à propagação do gradiente ao longo de sequências temporais extensas.

2.2.5.2 LSTM

As LSTM são uma categoria especializada de RNN que desempenham um papel fundamental na análise e previsão de sequências temporais complexas [112]. Através dessa arquitetura foi projetada para superar desafios inerentes às RNN tradicionais, tais como a dificuldade na aprendizagem de dependências temporais de longo alcance e o problema do desvanecimento do gradiente. O conceito-chave por trás das LSTM é a incorporação de unidades de memória e mecanismos de controlo, o que por sua vez permite que a rede aprenda e retenha informações importantes ao longo de sequências temporais prolongadas [112, 113].

A estrutura de uma LSTM é composta por unidades específicas que facilitam a retenção e o acesso a informações ao longo de sequências temporais [114]. A unidade de uma LSTM é geralmente composta por três portas distintas e uma célula de memória, como visto na Figura 2.12:

- **Célula de Memória (Cell State):** A célula de memória é responsável por armazenar e transmitir informações ao longo do tempo. Ela é projetada para preservar informações relevantes e descartar as não essenciais. A célula de memória opera como uma transportadora, na qual as informações são adicionadas ou removidas conforme necessário [115].

- **Porta de Esquecimento (Forget Gate):** A porta de esquecimento determina quais as informações da célula de memória anterior devem ser mantidas ou esquecidas [115]. Ela utiliza uma função de ativação sigmoide para gerar valores entre 0 e 1, onde 0 significa "esquecer completamente"e 1 significa "manter completamente". Esses valores são multiplicados pela célula de memória anterior para decidir quais informações manter [115].

As LSTM incorporam um mecanismo de esquecimento essencial para superar o desafio do degradação do gradiente, um problema comum em RNN. Este mecanismo permite que as LSTM retenham informações relevantes ao longo do tempo, atualizando-as conforme necessário [115].

- **Porta de Entrada (Input Gate):** A porta de entrada regula a adição de novas informações à célula de memória [115]. Ela passa por duas etapas: uma porta sigmoide decide quais valores da entrada atual devem ser atualizados, enquanto uma função de ativação tanh cria um vetor de candidatos para atualização. Esses valores são multiplicados para calcular as informações a serem adicionadas à célula de memória [115].
- **Porta de Saída (Output Gate):** A porta de saída decide qual parte da célula de memória atual deve ser a saída da unidade LSTM. A saída é determinada utilizando uma função de ativação sigmoide para decidir quais as partes da célula de memória envia como saída, que passam por uma função de ativação tanh para normalização [115].

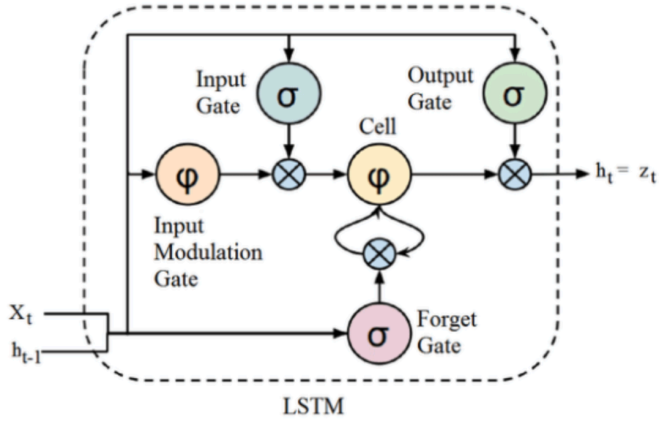


Figura 2.12: Exemplo de arquitetura LSTM, adaptado de [115]

A equação geral para a célula de memória em uma unidade LSTM é dada por [115]:

$$\text{Cell State}(t) = \text{Forget Gate}(t) \times \text{Cell State}(t-1) + \text{Input Gate}(t) \times \tanh(\text{Input}(t)) \tag{2.2}$$

A saída da unidade LSTM é então determinada multiplicando a célula de memória pelo resultado da porta de saída:

$$\text{Output}(t) = \text{Output Gate}(t) \times \tanh(\text{Cell State}(t)) \tag{2.3}$$

2.2.6 CNN

As CNN são uma classe de modelos de aprendizagem profunda que foram inicialmente desenvolvidos para processamento de imagens, mas a sua aplicabilidade foi estendida para várias outras áreas, incluindo séries temporais [116]. O que faz com que as redes destacam-se na extração de padrões e características de dados que têm uma estrutura de grade, como imagens ou, no caso de séries temporais, dados sequenciais [116].

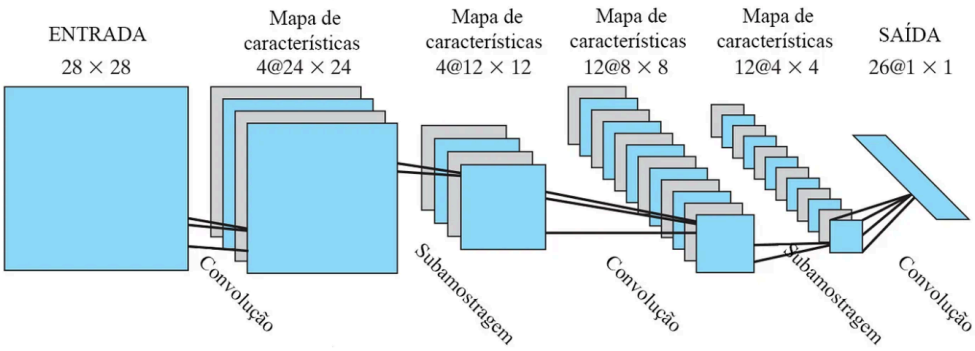


Figura 2.13: Exemplo de arquitetura CNN,retirado de: [117]

A arquitetura básica de uma CNN é composta por camadas de convolução, camadas de *pooling* e camadas totalmente conectadas [116], como representado na Figura 2.13. No contexto de séries temporais, as camadas de convolução são especialmente úteis para detetar padrões locais e sequenciais nos dados. As mesmas permitem que a rede aprenda automaticamente características relevantes, como mudanças de padrões ao longo do tempo [116].

A convolução em séries temporais envolve o uso de filtros que deslizam pela sequência, calculando produtos escalares locais em diferentes pontos. Por sua vez os filtros ajudam a identificar padrões específicos e características importantes na série temporal [109]. Além das camadas de convolução, as CNN para séries temporais frequentemente incluem camadas de pooling, que ajudam a reduzir a dimensionalidade dos dados, preservando as características mais importantes [109].

As CNN têm se mostrado eficazes em tarefas como previsão de séries temporais, classificação de padrões e deteção de anomalias. A sua capacidade de aprender representações hierárquicas e invariantes de deslocamento torna-as valiosas em cenários onde a posição relativa das características é crucial [109].

2.2.7 GRU

Os modelos GRU são um tipo de RNN que foram introduzidos para resolver alguns dos problemas enfrentados pelas RNN tradicionais, como a dificuldade em aprender dependências de longo prazo devido ao problema de degradação de gradiente [118]. As GRU têm ganho popularidade devido à sua simplicidade e eficiência em várias tarefas de processamento de sequência, como tradução automática, reconhecimento de fala, e previsão de séries temporais [118].

Uma GRU utiliza portas para controlar o fluxo de informações. As portas permitem que o modelo mantenha informações relevantes por longos períodos e esqueça informações desnecessárias [118]. Existem dois tipos principais de portas em GRU: a porta de reset (reset gate) e a porta de atualização (update gate) [118], como representado na Figura 2.14.

- **Porta de Reset (Reset Gate)**: A porta de reset decide a quantidade de informação anterior deve ser esquecida. É representada por um vetor r_t que é calculado da seguinte forma [119]:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{2.4}$$

  onde σ é a função sigmoide, W_r são os pesos treináveis, h_{t-1} é o estado oculto anterior, e x_t é a entrada atual.

- **Porta de Atualização (Update Gate)**: A porta de atualização determina a quantidade do estado oculto anterior deve ser levado em consideração para o próximo estado oculto. É representada por um vetor z_t que é calculado assim [119]:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{2.5}$$

2.2.7.1 Cálculo do Estado Intermediário (Candidate Hidden State)

O estado intermediário $\tilde{h}_t$ é calculado utilizando a porta de reset [119]:

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{2.6}$$

onde $*$ denota o produto elemento a elemento.

2.2.7.2 Cálculo do Novo Estado Oculto (Hidden State)

O novo estado oculto h_t é uma interpolação entre o estado oculto anterior h_{t-1} e o estado intermediário $\tilde{h}_t$ [119]:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{2.7}$$

A estrutura permite que as GRU lidem eficientemente com dependências de longo prazo, ajustando dinamicamente o quanto das informações passadas devem ser esquecidas ou retidas [119].

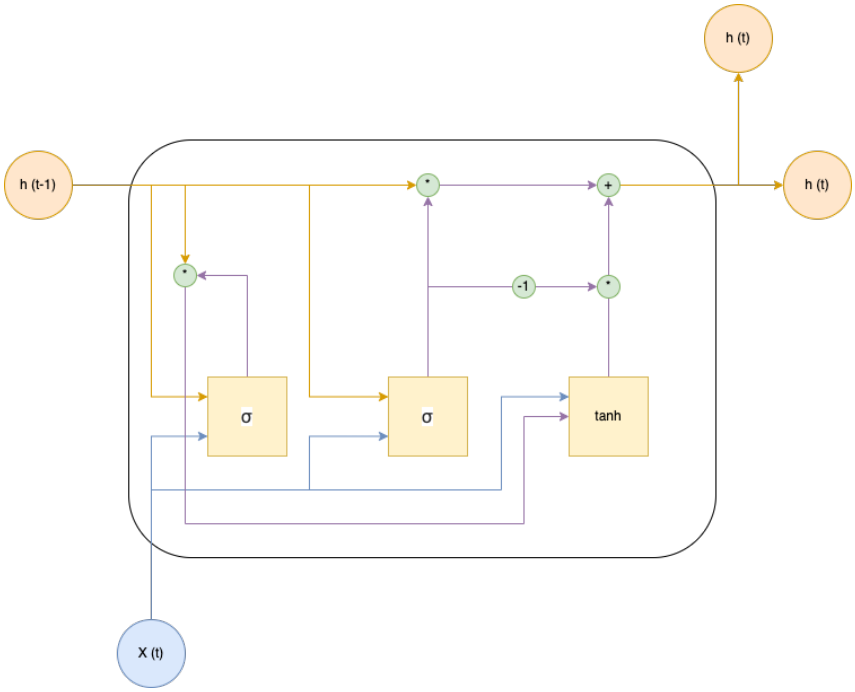


Figura 2.14: Estrutura exemplo GRU, adaptado de: [120]

As GRU são frequentemente comparadas com LSTM. Embora ambas as arquiteturas utilizem portas para controlar o fluxo de informações, as GRU são mais simples, com menos parâmetros, e costumam ser mais rápidas na fase de treino [121]. A principal diferença é que as LSTM têm uma porta de entrada adicional e uma célula de memória separada, enquanto as GRU combinam essas funções em uma estrutura mais compacta [121].

As GRU são amplamente utilizadas em várias aplicações de aprendizagem profunda que envolvem dados sequenciais [122]:

- **Processamento de Linguagem Natural (NLP)**: Tradução automática, modelação de linguagem, e análise de sentimento [122].
- **Reconhecimento de Fala**: Transcrição de fala em texto.
- **Previsão de Séries Temporais**: Previsão de preços de ações, demanda de energia, e análise de tendências de vendas [122].
- **Processamento de Vídeo**: Análise e síntese de vídeos, incluindo reconhecimento de ações e legendagem de vídeos [122].

2.2.8 LSTNet

O LSTNet é um modelo de rede neuronal projetado especificamente para previsão de séries temporais multivariadas [123]. A LSTNet combina as vantagens das CNN e das RNNs, em particular as LSTMs, para capturar padrões complexos e dependências de longo prazo em dados temporais [123].

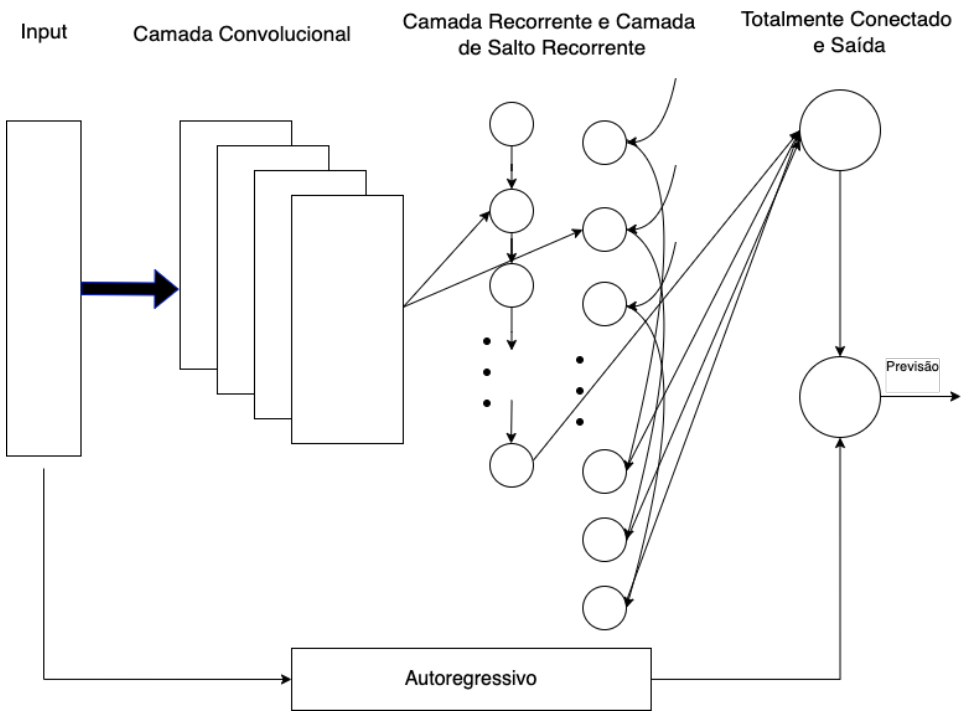


Figura 2.15: Estrutura exemplo LSTNet, adaptado de: [124]

O modelo LSTNet é composto por três principais componentes que trabalham em conjunto para capturar diferentes padrões das séries temporais: uma camada CNN, uma camada RNN e uma camada Skip-RNN. Além desses, há um componente adicional autoregressivo para melhorar a capacidade de previsão de curto prazo, como ilustrado na Figura 2.15:

- **CNN**: As CNN são usadas para extrair características locais de curto prazo das séries temporais [116]. A convolução é aplicada em cada dimensão da entrada *multi-step* para detetar padrões locais [123].

  A fórmula geral da convolução aplicada é:

  $$y_t = \sigma(W * x_t + b) \tag{2.8}$$

  onde $*$ denota a operação de convolução, W são os filtros convolucionais, x_t é a entrada e b é o viés.

- **RNN**: As RNN, especificamente LSTM ou GRU, como supramencionado são usadas para capturar dependências de longo prazo e a dinâmica temporal [110].

  A fórmula da atualização do estado oculto em uma LSTM é:

  $$h_t = \text{LSTM}(x_t, h_{t-1}) \tag{2.9}$$

  onde h_t é o estado oculto no tempo t e x_t é a entrada no tempo t.

- **Atividade Periódica e Desvio (Skip-RNN)**: Além das camadas CNN e RNN, a LSTNet introduz uma camada Skip-RNN para capturar periodicidades sazonais e padrões cíclicos. Essa camada salta intervalos fixos de tempo para modelar ciclos sazonais [123].

  A fórmula da Skip-RNN é:

$$h_t^{(\text{skip})} = \text{RNN}(x_{t-\text{skip}}, h_{t-1}^{(\text{skip})}) \tag{2.10}$$

  onde skip é o intervalo de tempo a ser saltado.

- **Componente Autoregressivo** : A LSTNet incorpora um componente autoregressivo para capturar a correlação linear direta nas séries temporais, o que é especialmente útil para previsões de curto prazo [123].

  Fórmula do componente AR:

$$\hat{y}_t = \sum_{i=1}^{p} \phi_i y_{t-i} \tag{2.11}$$

  onde ϕ_i são os coeficientes autoregressivos e y_{t-i} são os valores passados da série.

2.2.8.1 Fluxo do Modelo

- **Entrada**: A entrada é uma série temporal multivariada.
- **CNN**: As características locais são extraídas pela camada convolucional.
- **RNN**: As dependências de longo prazo são capturadas pela camada LSTM/GRU.
- **Skip-RNN**: As periodicidades sazonais são modeladas pela camada Skip-RNN.
- **AR**: As correlações lineares são modeladas pelo componente autoregressivo.
- **Saída**: A saída é uma previsão das séries temporais.

2.2.9 TCN

Os modelos TCN são uma abordagem recente para lidar com dados sequenciais [125]. Eles oferecem uma alternativa às RNN e suas variantes, como LSTM e GRU, para tarefas que envolvem séries temporais e outras sequências [125]. As TCN combinam a capacidade das CNN de extrair características locais com a habilidade de capturar dependências de longo prazo, proporcionando uma arquitetura eficiente e paralelizada [125], como representado na Figura 2.16.

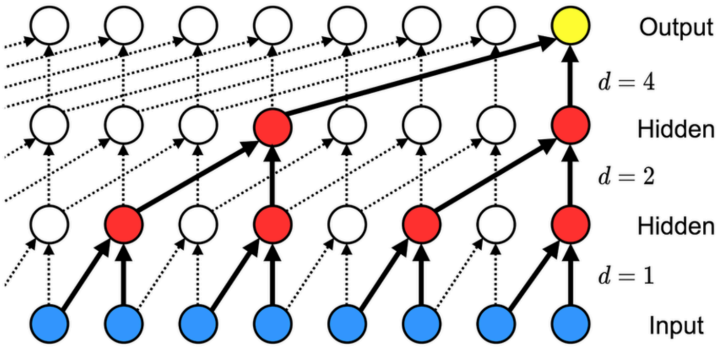


Figura 2.16: Estrutura exemplo TCN, retirado de: [126]

As TCN têm várias características que os diferenciam das RNN e fazem deles uma escolha robusta para o processamento de séries temporais [127]:

- **Convoluções Causais** : As convoluções causais garantem que a predição no tempo t dependa apenas dos valores anteriores (ou do próprio tempo t), evitando assim a "fuga de informação"do futuro para o passado [127]. A fórmula da convolução causal é:

$$y_t = \sum_{i=0}^{k-1} (W_i \cdot x_{t-i}) \tag{2.12}$$

  onde W_i são os filtros convolucionais e x_{t-i} são os valores de entrada.

- **Convoluções Dilatadas** : As convoluções dilatadas permitem expandir o campo de recepção sem aumentar o número de parâmetros, introduzindo lacunas (dilations) entre os elementos de uma janela de convolução [127]. A fórmula da convolução dilatada:

$$y_t = \sum_{i=0}^{k-1} (W_i \cdot x_{t-d \cdot i}) \tag{2.13}$$

  onde d é o fator de dilatação.

- **Convoluções em Múltiplas Camadas** : As TCNs utilizam múltiplas camadas convolucionais, permitindo a extração de características em diferentes níveis de abstração, ajudando a capturar tanto padrões locais quanto globais [127]. Em cada camada, a convolução dilatada aumenta exponencialmente o campo de receção, permitindo que o modelo capture dependências de longo prazo [127].

- **Residual Connections (Conexões Residuais)** :As conexões residuais ajudam a mitigar o problema de gradiente desaparecendo e permitem a construção de redes mais profundas. Essas conexões adicionam a entrada da camada diretamente à sua saída [127]. A fórmula das conexões residuais:

$$y = \text{ReLU}(x + F(x)) \tag{2.14}$$

onde $F(x)$ é a saída da camada convolucional.

2.2.9.1 Fluxo do Modelo

- **Entrada**: A entrada é uma série temporal *univariate* ou *multi-step*.
- **Convoluções Causais**: Garantem que as predições dependam apenas dos valores atuais e anteriores, respeitando a ordem temporal.
- **Convoluções Dilatadas**: Aumentam o campo de receção sem adicionar novos parâmetros, capturando dependências de longo prazo.
- **Múltiplas Camadas**: Camadas convolucionais em diferentes níveis de abstração para capturar padrões locais e globais.
- **Conexões Residuais**: Mantêm o gradiente e permitem redes mais profundas, preservando informações importantes.
- **Saída**: Previsão das séries temporais com dependências de curto e longo prazo capturadas.

2.2.10 MLP

Os modelos MLP são um tipo de RNA amplamente utilizado em diversas aplicações de ML, principalmente em tarefas de classificação e regressão [128].

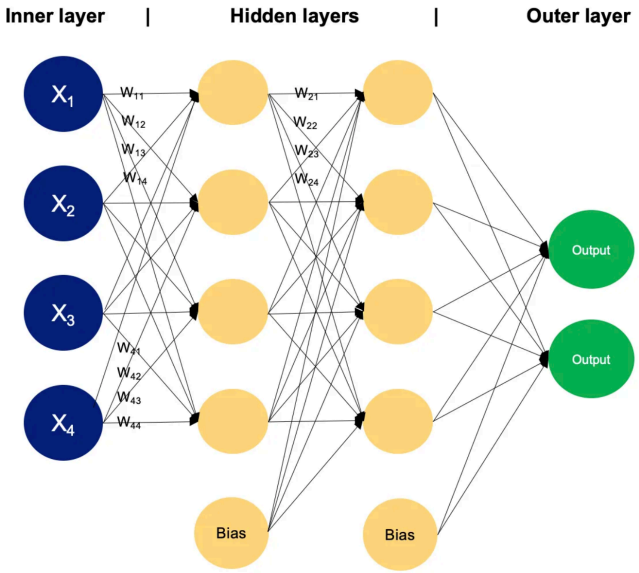


Figura 2.17: Estrutura exemplo MLP, retirado de: [129]

O modelo MLP é composto pelos seguintes componentes principais [130], a Figura 2.17 representa a constituição dos mesmo:

- **Camada de Entrada**: A camada de entrada é responsável por receber os dados iniciais. Cada neurónio nesta camada corresponde a uma característica (feature) do conjunto de dados. Por exemplo, se o conjunto de dados tem n variáveis independentes, a camada de entrada terá n neurónios [130].

  Matematicamente, a entrada $\mathbf{x}$ é representada como um vetor de dimensões $(n \times 1)$:

  $$\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$$

  onde x_i representa o valor de cada característica de entrada.

- **Camadas Ocultas**: As camadas ocultas são responsáveis por processar as entradas, aplicando transformações não-lineares para capturar as relações complexas entre as variáveis. Cada neurónio de uma camada oculta recebe entradas da camada anterior, realizando um calculo de uma combinação linear ponderada, sendo aplicada uma função de ativação para introduzir não-linearidade [130].

  A fórmula geral para o valor de ativação $a_j^{(l)}$ de um neurónio j na camada l é:

  $$a_j^{(l)} = \sigma\left(\sum_{i=1}^{n_{l-1}} w_{ji}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right)$$

  onde: - σ é a função de ativação (ex.: ReLU, Sigmóide, Tanh), - $w_{ji}^{(l)}$ é o peso entre o neurónio i da camada anterior $(l-1)$ e o neurónio j na camada l, - $a_i^{(l-1)}$ é a ativação do neurónio i na camada anterior, - $b_j^{(l)}$ é o valor do *bias* para o neurónio j na camada l, - n_{l-1} é o número de neurónios na camada $(l-1)$.

  Esta operação é realizada para cada neurónio em cada camada oculta. O número de camadas ocultas e de neurónios em cada uma é ajustável e depende da complexidade do problema [130].

- **Camada de Saída**: A camada de saída produz a previsão final do modelo, seja uma classificação ou um valor contínuo. O número de neurónios na camada de saída depende da natureza do problema. Por outro lado, para classificação binária, há geralmente um neurónio com uma função de ativação sigmoidal. Para problemas de classificação multiclasse, pode-se usar múltiplos neurónios com a função de ativação *softmax* [130].

  A fórmula para o valor $\hat{y}$ da previsão é:

  $$\hat{y} = \sigma\left(\sum_{j=1}^{n_L} w_j^{(L)} a_j^{(L-1)} + b^{(L)}\right)$$

onde: - σ é a função de ativação na camada de saída (ex.: Sigmóide, Softmax), - $w_j^{(L)}$ são os pesos da camada final, - $a_j^{(L-1)}$ são as ativações da última camada oculta, - $b^{(L)}$ é o *bias* da camada de saída.

- **Retropropagação de Erro (Backpropagation)**

  Durante o treino, a MLP ajusta os seus pesos através do processo de retropropagação de erro, que utiliza o gradiente descendente para minimizar a função de custo [131]. O gradiente da função de custo $J(\theta)$ em relação aos pesos $w_{ji}^{(l)}$ é calculado usando a regra da cadeia [131].

  A atualização dos pesos é dada por:

$$w_{ji}^{(l)} \leftarrow w_{ji}^{(l)} - \eta \frac{\partial J}{\partial w_{ji}^{(l)}}$$

  onde η é a taxa de aprendizagem e $\frac{\partial J}{\partial w_{ji}^{(l)}}$ é o gradiente da função de custo em relação ao peso $w_{ji}^{(l)}$.

2.2.10.1 Função de Ativação

A escolha da função de ativação é crítica para o desempenho da MLP. As funções mais comuns incluem:

- **ReLU (Rectified Linear Unit)**: $\sigma(x) = \max(0, x)$
- **Sigmóide**: $\sigma(x) = \frac{1}{1+e^{-x}}$
- **Tanh**: $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Cada função de ativação tem vantagens específicas dependendo do tipo de problema e do comportamento desejado no processamento da informação.

2.3 Revisão da Literatura

Este capítulo foca-se na revisão da literatura sobre a utilização de modelos de DL na previsão de parâmetros do reator do digestor anaeróbio de lamas, nomeadamente em problemas de séries temporais. A compreensão aprofundada desses estudos é essencial para orientar a presente dissertação. Serão apresentados resumos e críticas de cinco artigos selecionados, visando extrair conhecimentos fundamentais e identificar lacunas que possam orientar o desenvolvimento de soluções mais eficazes para este desafio específico.

O artigo [132], elaborada por Jeong et al., aborda o crescente interesse na Co-digestão anaeróbia (AcoD), devido ao aumento significativo na produtividade de biogás decorrente da utilização de diversos resíduos orgânicos, como resíduos alimentares e lodo de esgoto. O estudo propõe um modelo robusto de

AcoD para previsão de biogás usando DL, introduzindo uma arquitetura de DL híbrida denominada Drift-Adaptive, Long Short-Term Memory, Visual Sensor Network (DA-LSTM-VSN). Essa estrutura inclui uma rede de memória de curto prazo com atenção em duas fases, chamada Drift-Adaptive, Long Short-Term Memory (DA-LSTM), que é integrada a redes de seleção de variáveis. Para melhorar a previsibilidade do modelo, é realizada uma otimização de hiperparâmetros. A performance do modelo é validada utilizando métricas de erro, com base em dados de monitorização de longo prazo da AcoD ao longo de dois anos em uma ETAR. A comparação é feita com outros dois modelos baseados em DL (DA-LSTM e LSTM padrão). Além disso, é realizada uma análise da importância das características para investigar a contribuição relativa das variáveis de entrada na previsão de biogás. Os resultados destacam uma melhoria significativa na precisão do modelo ao incorporar DA no LSTM, aumentando o coeficiente de determinação (R^2) de 0.38 para 0.68. A combinação do DA-LSTM com Rede de Seleção de Variáveis (VSN) eleva ainda mais o R^2 para 0.76. O VSN contribui significativamente para a previsão de biogás, utilizando séries temporais descontínuas de dados de medição em variáveis orgânicas biodegradáveis durante a AcoD.

O artigo [133], desenvolvido por Salamattalab et al., apresenta a aplicação de uma RNA, mais concretamente uma LSTM, juntamente com um Algoritmo Genético (AG) para seleção de características. O objetivo é a previsão da produção de biogás em digestores anaeróbios em grande escala da ETAR de Tehran South (Irã), com uma produção de biogás de aproximadamente 30.000 Nm^3/d. Para simular as condições reais, o Tempo de Retenção Hidráulica (HRT) dos digestores anaeróbios, que é de 21 dias, foi utilizado como a janela de observação do modelo LSTM. Foram definidos três cenários diferentes para avaliar as previsões do modelo. No primeiro cenário, o modelo previu o biogás produzido utilizando características brutas do efluente e alcançou um coeficiente de determinação de R^2 = 0.84. O AG selecionou quatro dos onze parâmetros do efluente bruto, incluindo cargas de Demanda Bioquímica de Oxigênio em 5 dias (DBO5), Demanda Química de Oxigênio (DQO), Sólidos Suspensos Voláteis (SSV) e Nitrogênio Total (TN) (kg/d), como os dados mais informativos para o modelo. No segundo cenário, o modelo previu o biogás produzido usando dados dos fluxos de lodo espessado que entram nos digestores anaeróbios e obteve uma precisão ainda maior (R^2 = 0.89). Nesse cenário, o AG selecionou dois dos seis parâmetros dos fluxos de lodo, incluindo taxa de fluxo total (m^3 /d) e o teor médio de sólidos (w/w%). Por fim, no terceiro cenário, combinando os parâmetros dos dois cenários anteriores, a precisão das previsões do modelo aumentou ligeiramente (R^2 = 0,90). Os resultados indicam que a técnica de modelação Algoritmo Genético - Long Short-Term Memory (AG-LSTM) pode alcançar um desempenho confiável na previsão da produção de biogás em digestores anaeróbios de grande escala, incluindo o HRT no procedimento de modelação. Além disso, conclui-se que as características brutas do efluente afetam significativamente o comportamento do AD e podem ser usadas com sucesso como dados de entrada nos modelos de AD.

O estudo desenvolvido por Shen et al. [134], apresenta um estudo focado na previsão de parâmetros da digestão anaeróbia utilizando algoritmos de ML tanto para séries temporais como para dados não temporais. A pesquisa visa melhorar a previsao da produção de biogás e das suas características em escala industrial, ultrapassando limitações de estudos anteriores baseados em dados laboratoriais. Os autores adotam uma abordagem híbrida, comparando modelos não temporais, como o Gradient Boosting

Regression (GBR) e Random Forest (RF), com modelos de séries temporais, incluindo LSTM, CNN-LSTM e o DA-LSTM. Após otimizações de hiperparâmetros, os resultados mostram que o modelo GBR apresentou um excelente desempenho na previsão do teor de %CH4, com um valor de R^2 de 0.93 no conjunto de teste (RMSE = 1.11). O modelo DA-LSTM destacou-se na previsão da produção total de biogás, com um R^2 de 0.87 e um RMSE de 1048.14. Além disso, a temperatura foi identificada como a variável mais significativa para a produção de biogás, conforme revelado pela análise da importância das características e pelo método SHapley Additive exPlanations (SHAP), explicações das previsões feitas por ML. Este trabalho fornece uma base sólida para a aplicação de ML na monitorização e controlo em tempo real de projetos industriais de biogás, propondo uma abordagem eficiente para aumentar a estabilidade e a eficiência da produção energética em digestores anaeróbios.

Este estudo levado a cabo por Cheng et al. [135], aborda a importância da previsão da produção de metano na digestão anaeróbia com diferentes estratégias de pré-tratamento, visando um tratamento eficiente do lodo e a recuperação de recursos. Os modelos tradicionais de ML têm apresentado precisão limitada devido a desafios na otimização de parâmetros complexos e à escassez de dados. Neste trabalho, foi proposto um sistema integrado que utilizou um modelo de aprendizagem Semissupervisionado (SSL) combinado com aprendizagem de máquina automatizado (AutoML) em um *ensemble*. Considerando as correlações diretas entre o tipo de pré-tratamento e os substratos de digestão, o tipo de pré-tratamento é tratado como uma variável oculta. Os resultados demonstraram que o modelo AutoML superou os modelos ML convencionais (como Support Vector Regression - SVR, Extreme Gradient Boosting - XGB, etc.), evidenciado pelo valor de R^2 mais elevado. Além disso, a integração do SSL revelou ser uma mais-valia para melhorar a precisão das previsões, especialmente ao utilizar dados não rotulados. Isso resultou numa redução significativa do RMSE, de 11,3 para 9,7.A análise explicativa revelou a importância de diferentes variáveis, destacando a importância crucial do tempo de operação, seguido por proteínas, carboidratos, carência química de oxigénio e ácidos graxos voláteis. Além disso, a análise de componentes principais e correlação revelou relações interconectadas entre a concentração de substrato, comunidades microbianas e funções metabólicas na produção de metano, indicando que o aumento da concentração de substrato promoveu o enriquecimento de funções microbianas e metabólicas. Essas descobertas destacam as vantagens do SSL-AutoML na previsão da produção de metano em sistemas de digestão anaeróbia e elucidam as relações de dependência com variáveis-chave, oferecendo orientação valiosa para o pré-tratamento eficaz do lodo com o melhoração das capacidades de recuperação de recursos.

Como ultimo estudo [136], desenvolvido por Jia et al., aborda a importância da resposta imediata às perturbações do estado da digestão anaeróbia para evitar falhas no processo. No entanto, a monitorização frequente do estado e desempenho da digestão anaeróbia é desafiador.Assim, foram investigados modelos de DL para prever as variáveis de estado e de desempenho a partir dos dados recolhidos por sensores online. Entre estes, mediram-se parâmetros como o pH, a condutividade elétrica e o potencial de oxidação-redução, os quais foram processados como variáveis de entrada nos modelos de DL. Paralelamente, os dados de estado e de desempenho, obtidos em medições offline, foram utilizados como rótulos para treinar os modelos. O desempenho do modelo foi comparado para vários modelos de DL, como CNN,

LSTM e combinações. O modelo combinado de CNN e LSTM bidirecional foi robusto e bem generalizado na previsão das variáveis de estado e desempenho (R^2 = 0,978, RMSE = 0,031). O modelo combinado é uma excelente métrica virtual para monitorar o estado e desempenho da digestão anaeróbia a partir de sensores eletroquímicos.

2.3.1 Análise Crítica sobre os Estudos

Após uma leitura aprofundada e análise dos cinco artigos selecionados à aplicação de modelos de DL na previsão de parâmetros em digestores de lamas anaeróbias, é possível extrair conhecimentos relevantes para o campo. No geral, os estudos abordam de maneira abrangente questões cruciais, como a produção de biogás, seleção de características, uso de dados genômicos, estratégias de pré-tratamento e resposta a perturbações. No entanto, ao examinar os artigos em conjunto, algumas lacunas e áreas de melhoria se destacam.

Em [132], os autores oferecem uma descrição detalhada do tratamento de dados, incluindo a separação dos dados contínuos e descontínuos. A ausência de *cross-validation* é uma limitação, já que apenas uma divisão treino/teste (70/30) foi utilizada, e técnicas como *k-fold cross-validation* poderiam fortalecer a avaliação de generalização do modelo. O uso de *early stopping* para evitar *overfitting* é um ponto positivo, mas a inclusão de curvas de *loss* teria facilitado a análise desse risco. Além disso, a otimização de hiperparâmetros através da otimização *Bayesiana* foi bem implementada, assim como a seleção de variáveis por redes VSN, melhorando a interpretabilidade do modelo.

No que diz respeito ao estudo [133], é apresentado um processo rigoroso de preparação de dados, destacando-se a normalização e a eliminação de *outliers* como medidas essenciais para a qualidade do modelo. Foi usada uma divisão simples dos dados em treino e teste (80/20), a implementação de métodos como *k-fold cross-validation* poderia fornecer uma melhor avaliação da generalização do modelo. A abordagem para a seleção de hiperparâmetros via *random search* foi adequadamente explorada, e o uso de técnicas de regularização como *dropout* contribuiu para melhorar a robustez do modelo, reduzindo a variância sem comprometer significativamente o desempenho.

Os autores em [134] descrevem o tratamento de dados em que utilizam interpolação cúbica para lidar com valores ausentes, melhorando a precisão dos modelos preditivos. No entanto, não aplicaram técnicas de *cross-validation*, limitando-se a uma divisão treino-teste, o que pode afetar a generalização. O estudo focou-se em dados da planta de biogás, sem incluir variáveis externas, que poderiam melhorar a predição. A análise de *overfitting* foi brevemente mencionada, sem explorar muito o mesmo. Houve uma otimização robusta dos hiperparâmetros, tanto para modelos de séries temporais como não temporais, o que melhorou a precisão. A seleção de variáveis foi bem feita, com uso de SHAP, destacando a importância da temperatura na produção de biogás.

No artigo [135], os autores mencionam a normalização dos dados utilizando uma escala de 1 a 10. Além disso, não há menção ao uso de técnicas de *cross-validation* para prevenir *overfitting* e avaliar a generalização do modelo. Seria igualmente interessante a inclusão de gráficos, como curvas de *loss*, para

uma análise mais crítica sobre possíveis problemas de *overfitting* ou *underfitting*. Por outro lado, o artigo destaca a otimização automática de hiperparâmetros via AutoML, o que é positivo, mas faltam detalhes sobre o processo exato de otimização, o que teria enriquecido a análise.

No artigo [136], os autores não fornecem detalhes sobre o tratamento de dados realizado, como a gestão de valores nulos, outliers ou dados ausentes, o que teria sido importante. Além disso, não há menção ao uso de *cross-validation* para prevenir *overfitting* e avaliar a generalização do modelo, sendo que a inclusão de gráficos, como as curvas de *loss*, teria permitido uma análise mais crítica sobre *overfitting* ou *underfitting*. Embora o artigo apresente a divisão dos dados em conjuntos de treino e validação (70/30), a aplicação de técnicas de validação cruzada, como *K-fold cross-validation*, teria fornecido uma avaliação mais robusta da capacidade de generalização do modelo. Por outro lado, os autores destacam a otimização dos hiperparâmetros, o que é positivo, mas faltam detalhes sobre o processo utilizado para tal.

Analisando os estudos revistos, percebe-se que apenas dois deles apresentam um tratamento detalhado de dados, enquanto nos restantes há falta de informações importantes sobre o tratamento de dados, como a gestão de valores nulos ou *outliers*. Além disso, a ausência da técnica de *cross-validation* apropriado a series temporais é uma limitação em quatro dos cinco estudos, o que poderia ter oferecido uma avaliação mais robusta da generalização dos modelos. Dois estudos destacam a importância da otimização de parâmetros, utilizando métodos como *Bayesian Optimization* ou AutoML, embora muitas vezes faltem detalhes sobre o processo completo. Além disso, a análise de *overfitting* foi insuficiente ou ausente na maioria dos estudos, com poucos exemplos de ferramentas como curvas de *loss* que poderiam proporcionar uma visão mais clara sobre o desempenho do modelo. Finalmente, a seleção de variáveis, uma prática que pode aumentar significativamente a interpretabilidade dos modelos, foi realizada em apenas dois estudos, utilizando técnicas avançadas como redes VSN ou SHAP.

Capítulo 3

Materiais e Métodos

Este capítulo apresenta os materiais e métodos utilizados no desenvolvimento desta dissertação. Primeiramente, em 3.1, é detalhada a análise inicial realizada sobre os dados. Nesta mesma secção, são definidos os parâmetros da ETAR selecionados como alvos para as previsões dos modelos desenvolvidos posteriormente, especificamente %CH4 e Xc. Em seguida, a secção 3.2 descreve as técnicas desenvolvidas na preparação dos dados, culminando na construção dos conjuntos de dados específicos para cada caso de estudo abordado no capítulo seguinte. Finalmente, em 3.3, são apresentadas as tecnologias utilizadas para o desenvolvimento dos modelos desta dissertação.

3.1 Exploração dos Dados

Seguindo a estrutura Cross-Industry Standard Process for Data Mi- ning (CRISP-DM), a exploração e visualização de dados são etapas cruciais neste processo. Estas atividades permitem uma compreensão aprofundada dos dados disponíveis, a identificação de padrões e tendências, bem como a deteção de anomalias. A análise preliminar é essencial para fundamentar as decisões subsequentes no processo de modelação, assegurando a eficácia e precisão dos modelos desenvolvidos [137].

3.1.1 Análise do Dataset

Inicialmente, foi realizada uma análise dos tipos de dados presentes em cada coluna do dataset. Observou-se rapidamente a necessidade de padronizar os tipos de dados, de modo a garantir a consistência e permitir uma análise aprofundada. A Tabela 3.1 ilustra o tipo de dados iniciais. A padronização é essencial para evitar problemas de incompatibilidade e assegurar que todos os dados sejam comparáveis.

Features	Descrição	Tipo de dados
data	Data de medição	object
time lapse	Intervalo de tempo entre as observações	int64
Q(m3/d)	Fluxo de entrada	float64
T(°C)	Temperatura	float64
qgas produzido(m3/d)	Gás produzido	float64
qgas cogerador(m3/d)	Gás usado no cogerador	float64
E (kwh)	Energia	float64
pgas total	Pressão total do gás	float64
pgas.ch4	Pressão parcial de CH4	float64
pgas.co2	Pressão parcial de CO2	float64
%CO2	Percentagem de CO2 no gás	float64
%CH4	Percentagem de CH4 no gás	float64
VS	Sólidos voláteis	float64
X_I(mg/L)	Sólidos inorgânicos	float64
Xc(mgo2/L)	Demanda química de oxigênio	float64

Tabela 3.1: Tipos de cada coluna

Este conjunto de dados é composto por 1187 observações, registadas entre 01/01/2020 e 31/03/2023, com uma periodicidade diária. No entanto, existem 10 dias com valores em falta, o que cria pequenas lacunas na série temporal. Apesar disso, a maioria dos registos segue uma distribuição regular ao longo do tempo. A gestão destes dias ausentes será abordada num capítulo mais adiante, onde se explicará a metodologia adotada para mitigar o impacto dessas falhas e garantir a consistência da análise. Entre todas as *features* o %CH4 e a Xc são as que se destacam.

O %CH4 é um componente chave do biogás, resultante da digestão anaeróbia da matéria orgânica nas ETAR. A percentagem de metano no biogás é um indicador direto da eficiência do processo de digestão anaeróbia [138]. Monitorizar e otimizar a percentagem de metano no biogás permite maximizar a produção de energia renovável a partir de resíduos orgânicos, contribuindo para a sustentabilidade energética das ETAR [139] [138]. Além disso, a captura e utilização do metano produzido ajudam a reduzir as emissões de gases de efeito estufa, dado que o metano possui um potencial de aquecimento global significativamente superior ao do dióxido de carbono [138]. A qualidade do biogás, determinada pela percentagem de metano, influencia diretamente a sua aplicação como combustível. Um biogás com alta percentagem de metano é mais valioso e eficiente para a geração de eletricidade e calor, além de ser mais fácil de purificar para aplicações mais avançadas, como o biometano [139].

A Xc, expressa como Xc (mg O2/L), é um parâmetro essencial na avaliação da eficiência do tratamento biológico nas ETAR. O Xc representa a biomassa microbiana ativa no sistema de tratamento, sendo a sua concentração diretamente proporcional à capacidade do sistema em degradar a matéria orgânica presente nas águas residuais [139]. A monitorização de Xc é crucial para assegurar que o processo biológico está a funcionar de maneira otimizada. Além disso, a gestão eficaz da Xc permite um melhor controlo operacional das ETAR, pois parâmetros como a idade do lodo e a taxa de recirculação podem ser ajustados com base na medição de Xc, garantindo uma operação estável e eficiente do sistema de tratamento [139].

A escolha de %CH4 e Xc como *targets* proporciona uma abordagem integrada que abrange tanto a eficiência energética como a eficácia do tratamento biológico. Os parâmetros selecionados, ao serem

monitorizados e otimizados, não só melhoram o desempenho das ETAR, como também contribuem de forma significativa para a sustentabilidade ambiental e energética.

3.1.1.1 Análise CH4

Fazendo uma análise estatística, consoante o observado na Tabela 3.2 é possível observar que o valor mínimo de %CH4 é 61.5, enquanto o valor máximo é 75.1. A média do %CH4 é 65.73, o que representa o valor central dos dados. O desvio padrão é 1.34, indicando a variabilidade dos dados em torno da média. Um desvio padrão relativamente baixo sugere que a maioria dos valores de %CH4 estão próximos da média.

Total	Média	Desvio Padrão	Mínimo	Máximo	1ºQuartil	Mediana	3ºQuartil
1186	65.73	1.34	61.50	75.1	64.99	65.70	66.57

Tabela 3.2: Estatística do parâmetro %CH4.

O primeiro quartil (25%) é 64.99, a mediana (50%) é 65.7 e o terceiro quartil (75%) é 66.57. Estes valores fornecem uma visão sobre a distribuição dos dados, indicando que 50% dos valores estão entre 64.99 e 66.57.

Uma das melhores formas de avaliar a simetria e as características das caudas da distribuição é o cálculo da *skewness* e *kurtosis* [140]. O valor de *skewness* é 0.54, o que indica que a distribuição do %CH4 é ligeiramente assimétrica com uma cauda à direita. Isso sugere que existem alguns valores maiores que puxam a média para cima, embora a assimetria não seja muito acentuada. O valor de *kurtosis* é 2.27, o que indica que a distribuição do %CH4 é ligeiramente *platykurtic*, ou seja, menos pontiaguda do que a distribuição normal. O que leva a que os dados têm caudas mais curtas e uma concentração de valores mais uniforme em torno da média. Considerando o problema das séries temporais, é fundamental verificar a integridade dos *timestamps* e a presença de valores ausentes.

Para uma análise mais completa do parâmetro %CH4, foram gerados dois gráficos adicionais. Um gráfico que apresenta os dados agrupados por estação do ano e por ano, utilizando a média dos valores neste agrupamento. Esta abordagem permite compreender as variações do valor médio de %CH4 entre os anos de 2019 e 2022 para cada estação do ano, oferecendo uma visão detalhada das tendências sazonais e das possíveis influências climáticas sobre os níveis de %CH4, como visto na figura 3.1.

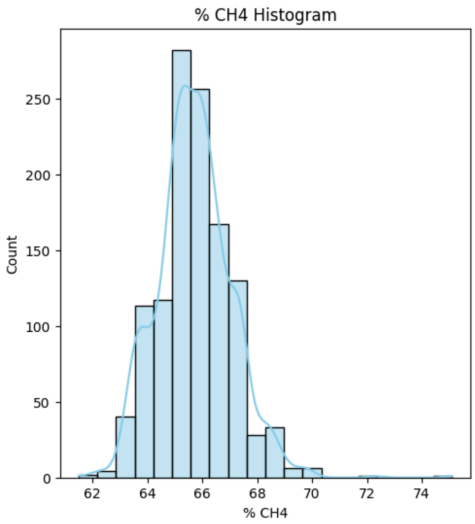


Figura 3.1: Histograma %CH4

Adicionalmente, foi gerado o gráfico que representa os valores médios mensais de cada ano. Este gráfico fornece uma análise mais granular das variações mensais, permitindo identificar padrões específicos dentro de cada ano e compará-los ao longo do período de estudo.

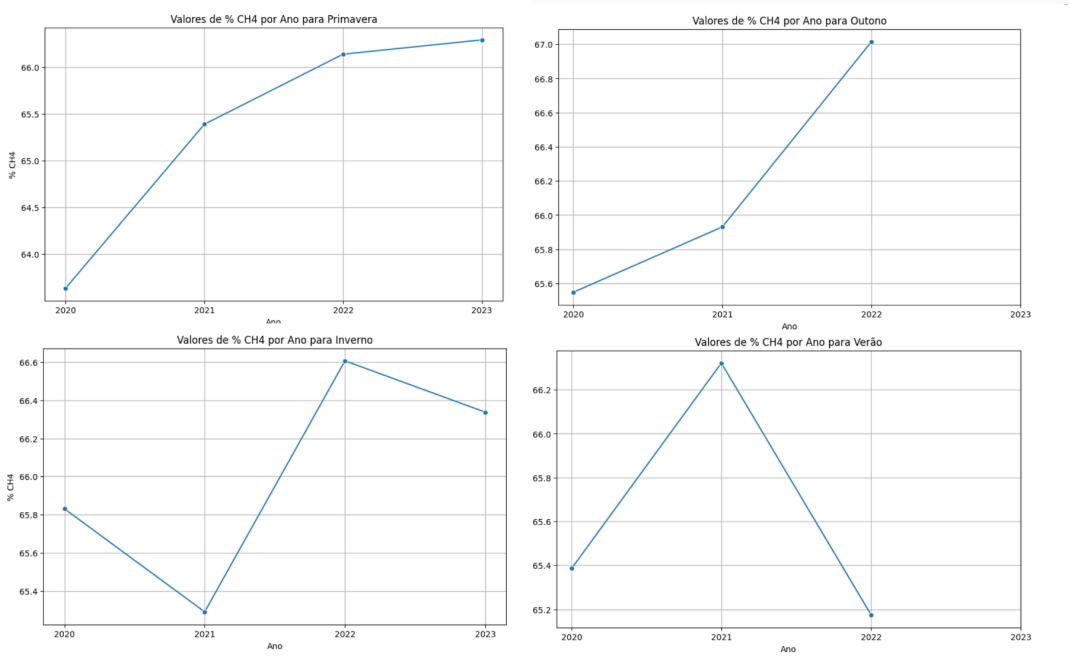


Figura 3.2: Valores médios anuais de %CH4 para cada estação do ano.

Na Figura 3.2 que apresenta os valores médios anuais de %CH4 para cada estação do ano, observa-se um comportamento particular ao longo dos anos. Em três das quatro estações (inverno, primavera e outono), verifica-se um pico acentuado no ano de 2022, seguido de uma tendência geral de aumento ao longo do período analisado. Esta tendência de crescimento sugere uma evolução significativa nos valores de %CH4 para estas estações. Em contrapartida, a estação do verão destaca-se como uma exceção, mostrando uma tendência oposta. Durante o mesmo intervalo de tempo, os valores médios de %CH4 no verão apresentam uma tendência de decréscimo, diferenciando-se claramente das outras estações analisadas.

Esta discrepância sugere que fatores sazonais específicos podem ter uma influencia na concentração de %CH4 de maneira diferenciada ao longo das estações do ano.

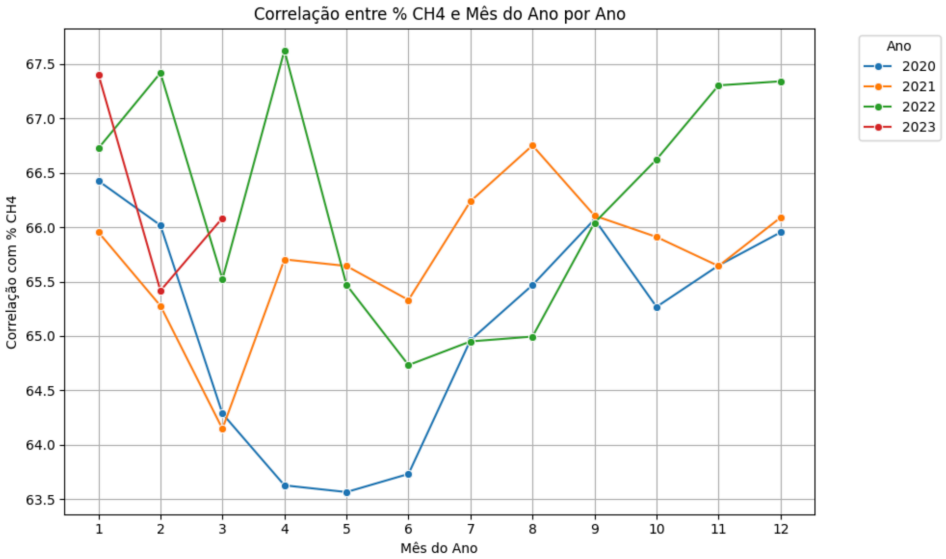


Figura 3.3: Valores médios mensais CH4 para cada ano.

Na Figura 3.3, observa-se que os valores de %CH4 tendem a ser mais elevados no início e no final de cada ano, enquanto há uma disparidade nos meses intermédios e entre os diferentes anos. De um modo geral, verifica-se um aumento progressivo nos valores de %CH4 ao longo dos anos, com uma tendência ascendente. Este padrão culmina no ano de 2023, que apresenta o valor médio de %CH4 mais alto de todo o período analisado. Este comportamento sugere uma acumulação gradual de %CH4 ao longo dos anos, possivelmente devido a fatores sazonais ou mudanças nas condições ambientais que influenciam a concentração deste parâmetro ao longo do tempo.

3.1.1.2 Análise Xc

De acordo com os dados apresentados na Tabela 3.3, o valor mínimo de Xc é 1731.97 mgo2/L, enquanto o valor máximo atinge 55343.6. A média de Xc é 24276.91 mgo2/L, representando o valor central da distribuição dos dados. O desvio padrão é 6026.76 mgo2/L, refletindo a dispersão dos dados em torno da média. Um desvio padrão relativamente alto sugere uma maior variabilidade nos valores de Xc em relação à média.

Total	Média	Desvio Padrão	Mínimo	Máximo	1ºQuartil	Mediana	3ºQuartil
1186	24276.91	6026.91	1731.97	55454.6	20030.63	23735.14	28923.90

Tabela 3.3: Estatística do parâmetro Xc.

O primeiro quartil (25%) é 20030.63, a mediana (50%) é 23735.14 e o terceiro quartil (75%) é 28923.90. Estes valores fornecem uma visão detalhada da distribuição dos dados, indicando que 50% dos valores de Xc estão compreendidos entre 20030.63 e 28923.90.

Recorrendo novamente aos dados de *skewness* e de *kurtosis*, o valor da *skewness* é 0.54. Este valor positivo indica que a distribuição de Xc é ligeiramente assimétrica com uma cauda à direita. O que leva a sugerir que existem alguns valores mais elevados que puxam a média para cima, embora a assimetria não seja muito acentuada. Com um valor de *kurtosis* de 2,27, podemos concluir que a distribuição de Xc é ligeiramente *platykurtic*. Isto significa que a distribuição tem uma menor concentração de dados em torno da média e caudas mais curtas em comparação com a distribuição normal. Consequentemente, há uma menor frequência de valores extremos, resultando numa distribuição mais uniforme e menos pontiaguda.

Novamente, recorrendo ao histograma ilustrado na Figura 3.4, é possível analisar o problema das séries temporais, relativamente ao Xc.

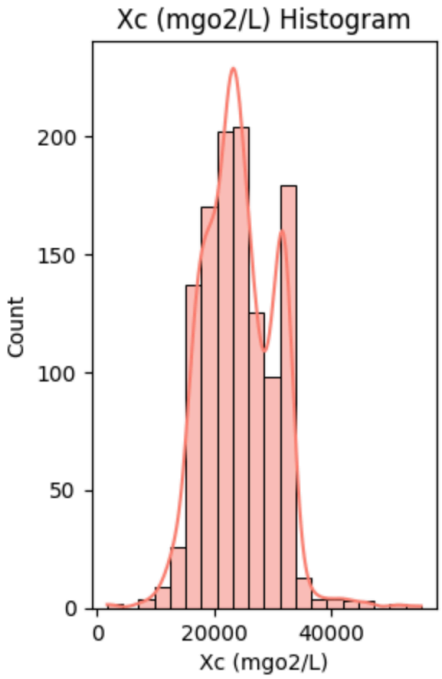


Figura 3.4: Histograma Xc

De forma similar para uma análise mais completa, como foi realizado com %CH4, foram criados dois gráficos adicionais, o primeiro gráfico exibe os dados divididos por estação do ano e por ano, recorrendo às médias dos valores em cada divisão. Já o segundo gráfico representa os valores médios mensais de Xc ao longo de cada ano.

Na Figura 3.5, é possível analisar os valores de Xc ao longo de três anos, distribuídos pelas quatro estações do ano. Observa-se um padrão claro nas quatro estações, caracterizado por um decréscimo dos valores de Xc ao longo dos anos 2020, 2021 e 2023. No entanto, no início do ano de 2024, há um crescimento evidente deste valor, especialmente nos meses correspondentes ao inverno e à primavera.

Adicionalmente, entre os anos de 2020 e 2023, a primavera destacou-se como a estação com o maior pico de descida dos valores de Xc. Nas restantes estações do ano, embora também se observe um decréscimo, este não é tão acentuado quanto o verificado na primavera. Estes padrões são cruciais para compreender as variações sazonais e interanuais de Xc.

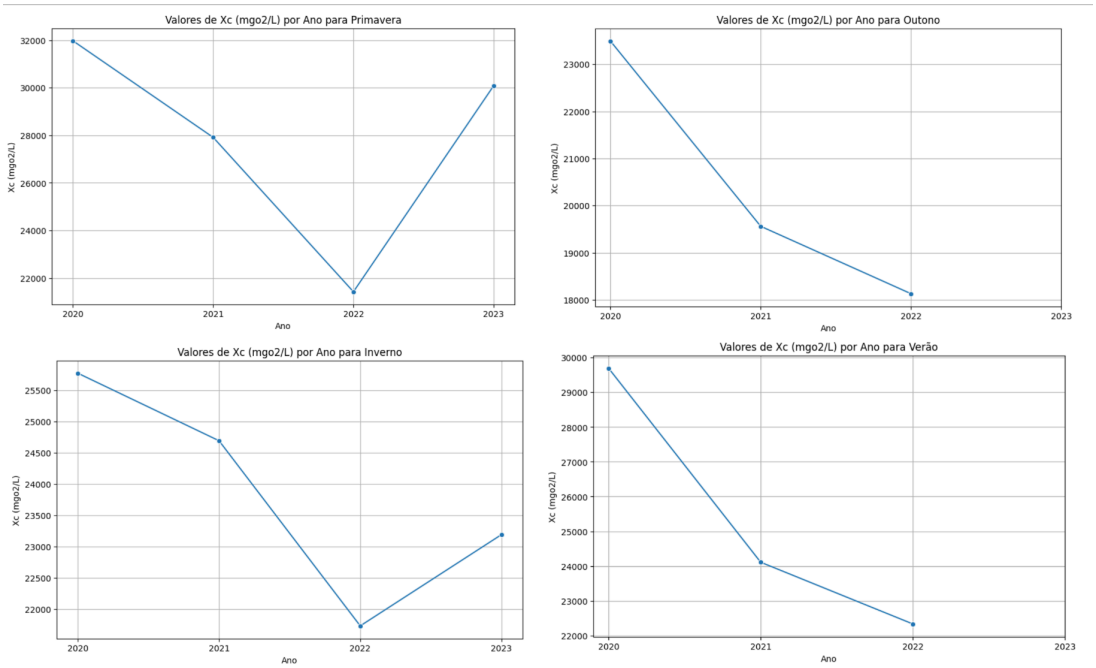


Figura 3.5: Valores médios anuais de Xc para cada estação do ano.

Em relação à Figura 3.6, é notório que os valores de Xc exibem um comportamento semelhante ao longo dos anos. Especificamente, observa-se um aumento súbito no início do ano, seguido de uma diminuição gradual ao longo do restante período. A exceção a este padrão é o ano de 2022, onde se verifica uma descida no início do ano, uma subida no meio do ano e uma nova descida para o final do ano.

Além disso, é importante destacar que tem havido um decréscimo percentual dos valores de Xc ao longo dos anos, apesar do comportamento geral se manter consistente. A exceção notável é o ano de 2023, que apresenta registos apenas até ao mês de março, exibindo valores significativos: menores que os de 2020, mas superiores aos de 2021 e 2022.

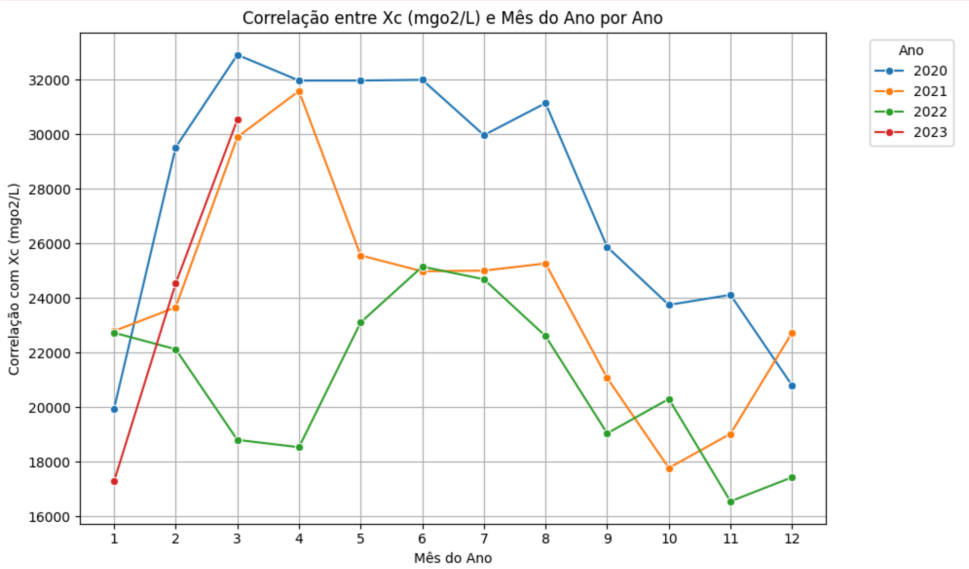


Figura 3.6: Valores médios mensais Xc para cada ano.

3.2 Preparação de Dados

A fase de preparação dos dados é crucial para garantir que os dados estejam adequadamente preparados para serem utilizados como input nos processos de treino e teste dos modelos concebidos. Um *dataset* bem preparado e modificado é essencial para obter modelos de previsão precisos e confiáveis. A preparação dos dados envolve várias etapas, cada uma com sua importância específica.

Deste modo, a preparação dos dados consistiu numa fase inicial de identificação e tratamento dos valores em falta, em 3.2.1. Posteriormente, foi realizado o tratamento de *outliers*, em 3.2.2. Por fim, foi realizado o processo de *feature engineeringe* e a análise de correlação, em 3.2.3 e em 3.2.4.

3.2.1 Tratamento dos Missing Values

Verificou-se a ausência de dados correspondentes a um período de 10 dias, o que resultou na falta de 10 linhas no conjunto de dados. Esta lacuna afeta a continuidade temporal do *dataset*, o que poderia comprometer as análises subsequentes. Para assegurar que a sequência temporal não fosse interrompida, foi necessário injetar essas linhas em falta, mantendo a integridade da estrutura temporal.

Após a inserção das linhas correspondentes aos dias em falta, surgiram valores em branco nas variáveis relacionadas com esses dias, aumentando a presença de valores em falta no *dataset*. Esses *missing values* poderiam prejudicar a qualidade dos modelos de previsão se não fossem devidamente tratados.

Para resolver esta questão, aplicou-se a interpolação linear, uma técnica comum em séries temporais. Este método preenche os valores ausentes ao estimar os dados com base nos pontos adjacentes, assumindo uma progressão linear entre os valores observados antes e depois das lacunas. Deste modo, preserva-se a tendência e a estrutura subjacente do *dataset*, garantindo a continuidade da informação e minimizando o impacto da ausência de dados.

3.2.2 Tratamento dos Outliers

Após o tratamento dos valores em falta, foi necessário prosseguir com uma análise aprofundada, especificamente para identificar e tratar possíveis *outliers* existentes no *dataset*. Os *outliers* são valores que se desviam significativamente da maioria dos dados, podendo distorcer as análises e os modelos preditivos se não forem adequadamente tratados [141].

Para identificar os *outliers*, utilizou-se o método de análise dos valores que ultrapassam os limites superiores e inferiores. Estes limites são calculados com base no intervalo interquartil (IQR), que é a diferença entre o terceiro quartil (Q3) e o primeiro quartil (Q1) dos dados. Os limites são definidos da seguinte forma [141], como ilustrado em 3.1 e 3.2:

$$LimiteInferior : Q1 - 1.5 * IQR \tag{3.1}$$

$$LimiteSuperior : Q3 + 1.5 * IQR \tag{3.2}$$

Valores que se encontram fora destes limites são considerados *outliers*. A ferramenta de apoio utilizada para a visualização e identificação destes *outliers* foi o *boxplot*, que é um gráfico que permite observar a distribuição dos dados, os quartis e os possíveis *outliers*. Foram realizados dois *boxplots*: um para o valor de %CH4 e outro para o valor de Xc.

3.2.2.1 %CH4

Para a análise de *outliers* no alvo %CH4, tal como mencionado anteriormente, foi utilizada a técnica do *boxplot*. A Figura 3.7 apresenta os *outliers* associados ao valor de %CH4, ou seja, os valores que ultrapassam os limites determinados. Esta visualização permite uma análise clara dos dados atípicos, contribuindo para uma melhor compreensão da distribuição dos valores.

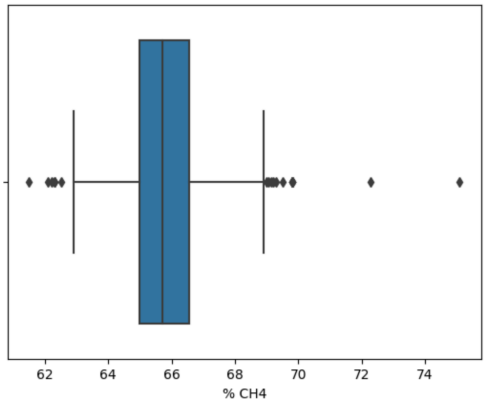


Figura 3.7: *BoxPlot* de %CH4 antes do tratamento.

Observa-se claramente a presença de diversos pontos que ultrapassam os limites superior e inferior. Os valores discrepantes à esquerda (valores menores) situam-se abaixo de aproximadamente 62% de %CH4, enquanto os valores discrepantes à direita (valores maiores) estão acima de aproximadamente 68% de %CH4, alguns alcançando até 75%.

É possível observar que a maior parte dos dados de %CH4 está concentrada em uma faixa estreita, entre cerca de 64% e 66%, como indicado pelo IQR. No entanto, a presença significativa de múltiplos valores discrepantes, sobretudo à direita, sugere a existência de alguns valores elevados de %CH4 no conjunto de dados.

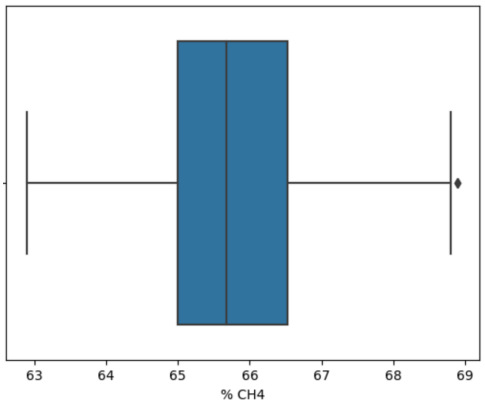


Figura 3.8: *BoxPlot* de %CH4 após o tratamento.

Após a observação dos valores fora dos limites e sua identificação, optou-se por aplicar a técnica de interpolação linear para tratar esses valores extremos. Esta abordagem consiste em substituir os *outliers* por valores estimados com base nos dados adjacentes, garantindo que a sequência temporal dos dados seja mantida.

Ao utilizar a interpolação, busca-se não apenas preservar a integridade do conjunto de dados, mas também melhorar a robustez e a validade das análises subsequentes, tal como visto na Figura 3.8.

3.2.2.2 Xc

De forma semelhante a %CH4, foi utilizado *boxplot* para a visualização de *outliers*, em relação ao *boxplot* da Figura 3.9, é possível observar que os *outliers* à esquerda estão abaixo de aproximadamente 10,000 mgO2/L, enquanto os *outliers* à direita estão acima de aproximadamente 40,000 mgO2/L, com alguns valores atingindo até cerca de 55,000 mgO2/L. A maioria dos dados de Xc está concentrada entre aproximadamente 20,000 mgO2/L e 30,000 mgO2/L, conforme indicado pelo intervalo interquartil (IQR). A presença de vários *outliers*, especialmente à direita, sugere que existem alguns valores elevados de Xc no conjunto de dados. Esses valores altos podem ser resultado de eventos anómalos ou picos específicos que merecem uma investigação adicional para compreender melhor suas causas e impactos.

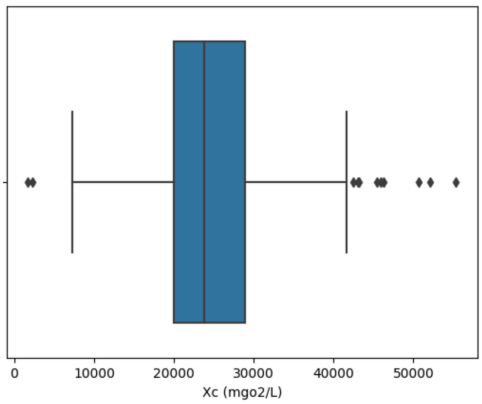


Figura 3.9: *BoxPlot* de Xc antes do tratamento.

O processo de tratamento de *outliers* identificados foi semelhante ao processo supramencionado em %CH4, foi também usado a técnica de interpolação linear para tratar os valores extremos, tal como visto na Figura 3.10

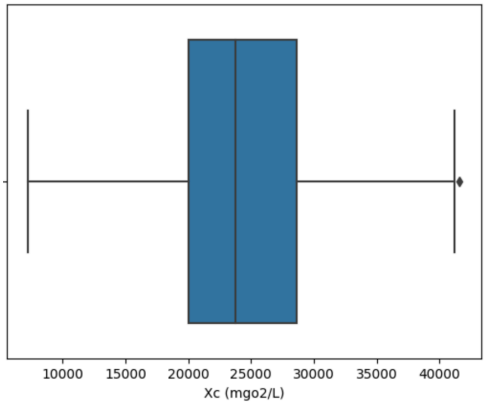


Figura 3.10: BoxPlot de Xc após o tratamento.

3.2.3 Feature Engineering

Após a deteção e tratamento de *outliers*, foi utilizada a técnica de *feature engineering*. *Feature engineering* refere-se ao processo de usar o conhecimento sobre os dados para criar novas variáveis que podem ajudar os modelos de ML e DL a obterem melhores previsões. Este processo envolve a criação, transformação e seleção de *features* a partir dos dados brutos [142]. A aplicação adequada desta técnica pode melhorar significativamente a performance dos modelos ao permitir que eles capturem de forma mais eficaz os padrões subjacentes nos dados. O que por sua vez permite uma melhoria da performance do modelo, redução de ruído e melhoria da interpretação [142].

No contexto desta dissertação, foram criadas novas *features* a partir da *feature* original data. Estas novas *features* incluem:

- **Ano**: Representa o ano da entrada.

- **Mês**: Representa o mês da entrada.
- **Dia**: Representa o dia da entrada.

Além disso, como já mencionado anteriormente, para uma análise mais profunda e detalhada, foi criada a *feature* estação, que indica a estação do ano correspondente a cada entrada. Esta *feature* foi determinada com base nas datas, permitindo uma análise sazonal dos dados. A inclusão da estação do ano poderia ser importante porque muitas variáveis ambientais e climáticas exibem variações sazonais significativas, é possível visualizar as *features* todas na Tabela 3.4.

Data	Q(m3/d)	T(ºC)	qgas produzido (m3/d)	qgas cogerador (m3/d)	E (kwh)	pgas.total	pgas.ch4	pgas.co2	%CO2	%CH4	VS	X_I(mg/L)	Xc (mgo2/L)	Ano	Mês	Dia	Estação do Ano
2020-01-01	36.60	35.2	323.563	333.125	532.167	0.221	0.146	0.074	33.3	66.1	11662.2	3784	13416	2020	1	1	4
2020-01-02	30.35	34.8	439.698	448.375	521.750	2.140	1.415	0.713	33.3	66.1	11662.2	3784	13416	2020	1	2	4
2020-01-03	28.42	35.0	295.292	310.813	481.354	2.140	1.412	0.714	33.4	66.0	11662.2	3784	13416	2020	1	3	4
2020-01-04	33.62	34.3	372.854	373.271	705.104	2.140	1.415	0.713	33.3	66.1	11662.2	3784	13416	2020	1	4	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Tabela 3.4: *Features* do novo *dataset*.

3.2.4 Análise de Correlação e Seleção de *Feature*

Por fim, foi realizada a análise de correlação de dados e a subsequente seleção de *features*. Para este propósito, optou-se pelo coeficiente de correlação de *Spearman*. Este método foi escolhido devido à sua capacidade de identificar relações monotônica.

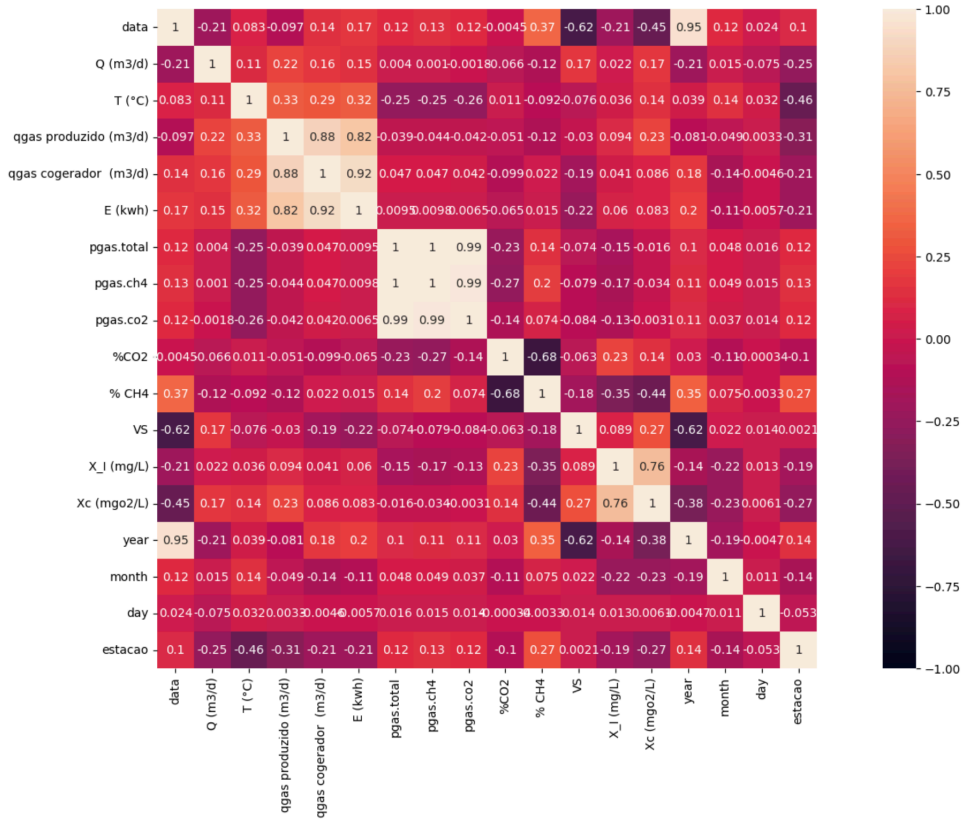


Figura 3.11: Matriz de correlação.

Na Figura 3.11, apresenta-se a matriz de correlação utilizando o coeficiente de *Spearman*, onde são exibidas as correlações entre todas as variáveis presentes no *dataset*.

Com base nos resultados da matriz de correlação, foram selecionadas as *features* mais adequadas para os valores de %CH4 e de Xc.

Para o parâmetro %CH4, as seguintes *features* foram selecionadas:

- %CO2: Apresenta um coeficiente de correlação de -0.68 com %CH4, indicando uma forte correlação negativa. Isto sugere que variações nos níveis de %CO2 estão fortemente associadas a variações em %CH4.
- Ano: Com um coeficiente de correlação de 0.35, esta *feature* tem uma correlação moderada com %CH4. Isto pode indicar tendências temporais ou mudanças ao longo dos anos que afetam os níveis de %CH4.
- X_I (mg/L): Também apresenta uma correlação de -0.35 com %CH4, indicando uma relação moderada e negativa.
- Xc: Com um coeficiente de correlação de -0.44, esta variável mostra uma correlação moderada com %CH4 e negativa.

Para o parâmetro Xc, as *features* selecionadas foram:

- Ano: Apresenta um coeficiente de correlação de -0.38, indicando uma forte correlação negativa.
- X_I (mg/L): Também com um coeficiente de correlação de 0.76, esta variável demonstra uma forte correlação, sugerindo uma relação significativa com Xc.
- %CH4: Com um coeficiente de correlação de -0.44, esta *feature* tem uma forte correlação negativa com Xc.

3.3 Tecnologias

Para a realização desta dissertação, foi utilizado *Python* 3.11.8 como linguagem principal de programação. Todo o trabalho foi realizado no ambiente de trabalho do *Kaggle* e do *Collab*, uma plataforma de ciência de dados que oferece um ambiente de computação em nuvem altamente eficiente, incluindo acesso a GPU.

As bibliotecas de *Python* utilizadas no desenvolvimento desta dissertação foram selecionadas pela sua robustez e eficiência na manipulação de dados, visualização e construção de modelos preditivos. As principais bibliotecas utilizadas foram *Pandas*, *Matplotlib*, *Seaborn*, *Scikit-learn*, *NumPy*, *TensorFlow*, etc. O uso combinado destas bibliotecas permitiu um fluxo de trabalho eficiente e eficaz, desde a exploração inicial dos dados até ao desenvolvimento e avaliação de modelos preditivos robustos.

Capítulo 4

Experiências

Todas as abordagens, decisões e técnicas consideradas nas experiências realizadas são apresentadas neste capítulo. Primeiro, é descrito o processo de implementação dos modelos candidatos, na secção 4.1. Em seguida, na secção 4.2, são apresentadas as métricas de avaliação utilizadas neste trabalho.

4.1 Implementação de Modelos

Considerando a aplicação de diferentes modelos na elaboração das experiências desta dissertação, este subcapítulo explora os recursos utilizados e as estratégias implementadas para configurar esses modelos, além de apresentar os valores dos hiperparâmetros definidos para os modelos. Os modelos escolhidos para prever Xc e %CH4, que envolvem a análise de séries temporais, são LSTM, CNN, GRU, LSTNet, TCN e MLP. É relevante salientar que todos esses modelos são fundamentados em algoritmos de DL.

A implementação dos modelos de DL foi realizada em *Python*, utilizando a biblioteca *TensorFlow* e a Application Programming Interface (API) *Keras*. A abordagem adotada para o desenvolvimento dos modelos foi dinâmica, o que permitiu a otimização de hiperparâmetros conforme as necessidades exigidas pela exploração de dados realizada, como ilustrado na Figura 4.1. O que leva a que, ao iniciar os modelos, vários hiperparâmetros podem ser ajustados, proporcionando maior flexibilidade e personalização no processo de treino.

Para avaliar o desempenho dos modelos durante o treino, foram utilizados gráficos das *learning curves*. Os mesmos foram essenciais para a escolha do número de *epochs* mais adequado em cada caso, garantindo assim um processo de treino eficiente, evitando tanto o *underfitting* como o *overfitting*. Além disso, os *datasets* utilizados passaram por um processo de *cross-validation* com o *TimeSeriesSplit*, de modo a não quebrar a sequência temporal dos dados. Este *cross-validator* foi configurado com 3 divisões (*splits*) para criar os conjuntos de treino, teste e validação, uma escolha justificada por ser uma técnica comum aplicada em séries temporais [143]. O uso do *TimeSeriesSplit* é crucial, pois preserva a ordem

cronológica dos dados e evita o vazamento de informações, assegurando uma avaliação mais realista da capacidade de generalização dos modelos. De forma a assegurar a reprodutibilidade dos resultados, foi definida uma seed com o valor 91195003 em todos os modelos. Esta prática garante que os resultados obtidos sejam consistentes ao longo de diferentes execuções.

Em relação às funções utilizadas, a função de *loss* selecionada foi o RMSE, e o *optimizer* escolhido foi o *Adam*, reconhecido pela sua eficiência e robustez em diversos cenários de aprendizagem. Além disso, o processo de previsão seguiu uma abordagem de *multi-step recursive forecasting*, utilizando 2 passos à frente (*multi-steps*). Isto significa que, na etapa de validação dos modelos candidatos, foram avaliadas previsões para 2 *timesteps* futuros de forma recursiva. Ou seja, após a previsão do primeiro *timestep*, este valor foi usado como entrada para prever o segundo *timestep*. As métricas de avaliação foram calculadas com base na média dos erros gerados nessas 2 previsões, permitindo uma análise mais robusta e facilitando a generalização dos modelos.

Na construção dos modelos, foram explorados diversos hiperparâmetros essenciais para otimizar o desempenho e a capacidade de generalização das redes neuronais, tal ilustrado na Tabela 4.1.

Hiperparâmetros	**Valores**
Layers	[3,4,5]
Neurons	[34,64,128]
Dropout Rate	[0.0,0.5]
Activation	['relu','tanh']
Timesteps	[7,14,21]
Batch Size	[5,10,20]
Filters	[16,32]
Kernel Size	[3,4,5]
Pool Size	[2,3]

Tabela 4.1: Hiperparâmetros.

4.1.1 LSTM

Na implementação do modelo baseado em LSTM, foram utilizadas diversas classes da biblioteca *TensorFlow*, nomeadamente *Input*, LSTM, *Dense*, *Dropout* e *Model*. O código seguinte, 4.1, apresenta um excerto desta implementação, cuja finalidade é construir a arquitetura do modelo LSTM.

Inicialmente, é definida a *seed* através do comando *tf.random.set_seed*(seed), assegurando a reprodutibilidade dos resultados durante o treino do modelo. Posteriormente, é criado um tensor que representa as entradas (*inputs*) do modelo, utilizando a classe *Input*. Este tensor é então fornecido à arquitetura principal da rede, que processa as entradas através de uma série de camadas LSTM e camadas densas, até se obterem as saídas (*outputs*) do modelo.

No que diz respeito aos parâmetros utilizados na construção do modelo, estes são representados pelas variáveis *timesteps*, *features*, *layers*, *neurons*, *activation* e *dropout_rate*. Cada um destes parâmetros desempenha um papel crucial na definição da estrutura e do comportamento do modelo LSTM:

Os parâmetros do modelo LSTM desempenham papéis específicos na sua configuração. Os *timesteps* indicam o número de registos temporais anteriores usados para prever valores futuros, enquanto as *features* definem a quantidade de variáveis de entrada no modelo. As *layers* referem-se ao número de camadas LSTM, permitindo a captura de padrões mais complexos, e os *neurons* determinam o número de neurónios em cada camada, influenciando a capacidade de treino do modelo. A função de ativação (*activation*) na camada densa final introduz não linearidades, e o *dropout rate* define a percentagem de neurónios desativados durante o treino para evitar *overfitting* e melhorar a generalização.

A arquitetura do modelo segue uma lógica em que a primeira camada LSTM recebe diretamente os inputs e, dependendo do número de camadas especificadas, o modelo pode devolver sequências de saída intermediárias (*return_sequences=True*) ou valores únicos (*return_sequences=False*). A última camada LSTM é seguida por uma camada densa, que aplica a função de ativação especificada, e por uma camada de *dropout* para regularização. Finalmente, a camada de saída do modelo é uma camada densa com uma única unidade, responsável por gerar a previsão final, 4.1.

O modelo é então compilado utilizando a função de perda RMSE e o otimizador *Adam*. Para além da função de perda, o modelo é configurado para reportar a métrica MAE e novamente o RMSE durante o processo de treino.

```
1   def build_lstm_model(features, timesteps=7, layers=2, neurons=64, activation='relu',
    ↪ dropout_rate=0.5, seed=91195003):
2       tf.random.set_seed(seed)
3
4       inputs = Input(shape=(timesteps, features))
5
6       for i in range(layers):
7           #if first LSTM layer
8           if i == 0:
9               #if also the last LSTM layer then return_sequences=False
10              if i+1 == layers:
11                  x = LSTM(neurons/2, return_sequences=False)(inputs)
12              #it has more layers! So return_sequences=True
13              else:
14                  x = LSTM(neurons, return_sequences=True)(inputs)
15          #if last LSTM layer then return_sequences=False
16          elif i+1 == layers:
17              x = LSTM(neurons*2, return_sequences=False)(x)
18          #if not the last LSTM layer then return_sequences=True
19          else:
20              x = LSTM(neurons, return_sequences=True)(x)
21
22      x = Dense(neurons, activation=activation)(x)
23      x = Dropout(dropout_rate)(x)
24      outputs = Dense(1)(x)
25
```

```
    lstmModel = Model(inputs=inputs, outputs=outputs, name='lstm_model')

    lstmModel.compile(
        loss = rmse,
        optimizer= tf.keras.optimizers.Adam(),
        metrics = ['mae', rmse]
    )

    return lstmModel
```

Listagem 4.1: Excerto de código da construção do modelo LSTM.

Após o desenvolvimento do código, procedeu-se à análise das *learning curves* para ajustar os hiperparâmetros em cada cenário e selecionar o número de *epochs* mais adequado. Este processo foi cuidadosamente executado para minimizar o risco de *overfitting*, garantindo que o modelo se generalizasse bem a novos dados, com o número de *epochs* de 75 para %CH4 e 50 para Xc.

Além disso, as *features* dos conjuntos de dados foram normalizadas para o intervalo de valores entre -1 e 1. Este intervalo foi escolhido especificamente para as experiências com LSTM, uma vez que as camadas LSTM utilizam a função de ativação *tanh*, cuja saída está naturalmente limitada a este domínio, o que melhora a eficiência e a convergência do modelo [144].

4.1.2 CNN

Na construção do modelo de CNN, adotaram-se procedimentos semelhantes aos aplicados nos outros modelos, como a definição da *seed* para assegurar a reprodutibilidade dos resultados e a configuração dos vetores de entrada. As camadas da arquitetura foram estruturadas até alcançar os *outputs*, conforme detalhado na listagem 4.2. Para esta implementação, foram utilizadas várias classes do *Keras*, tais como *Input*, *Conv1D*, *AveragePooling1D*, *Flatten*, *Dense*, *Dropout* e *Model*.

A principal estratégia na construção deste modelo centrou-se numa série de camadas de convolução, seguidas por uma camada *feed-forward*. Os hiperparâmetros que diferem mais significativamente dos modelos anteriormente discutidos são *filters*, *kernel_size* e *pool_size*. O parâmetro *filters* controla o número de filtros de saída em cada camada de convolução, *kernel_size* define o tamanho da janela de convolução, e *pool_size* especifica o tamanho da janela de *average pooling*, 4.1. Estes hiperparâmetros são cruciais para ajustar a capacidade do modelo em identificar padrões nos dados temporais, influenciando diretamente o desempenho da CNN na tarefa analisada.

```
def build_cnn_model(features, timesteps=7, layers=2, activation='relu', dropout_rate=0.5,
    ↪ filters=16, kernel_size=3,
                    pool_size=2, log=False, seed=91195003):
    tf.random.set_seed(seed)

    kernel_stepsize = kernel_size - 1
    total_kernel_stepsize = kernel_stepsize * layers
```

```
    if timesteps - total_kernel_stepsize <= 0:
        return None

    inputs = Input(shape=(timesteps, features))

    for i in range(layers):
        # if first CNN layer
        if i == 0:
            # and also the last
            if i + 1 == layers:
                x = Conv1D(filters=filters / 2, kernel_size=kernel_size, activation=activation
    ↪ ,
                           data_format='channels_last')(inputs)
                x = AveragePooling1D(pool_size=pool_size, data_format='channels_first')(x)
                x = Flatten()(x)
            # it has more layers
            else:
                x = Conv1D(filters=filters, kernel_size=kernel_size, activation=activation,
                           data_format='channels_last')(inputs)
                x = AveragePooling1D(pool_size=pool_size, data_format='channels_first')(x)
        # if last layer
        elif i + 1 == layers:
            x = Conv1D(filters=filters * 2, kernel_size=kernel_size, activation=activation,
                       data_format='channels_last')(x)
            x = AveragePooling1D(pool_size=pool_size, data_format='channels_first')(x)
            x = Flatten()(x)
        # if middle layers
        else:
            x = Conv1D(filters=filters, kernel_size=kernel_size, activation=activation,
    ↪ data_format='channels_last')(x)
            x = AveragePooling1D(pool_size=pool_size, data_format='channels_first')(x)

    x = Dense(filters)(x)
    x = Dropout(dropout_rate)(x)
    outputs = Dense(1)(x)

    cnnModel = Model(inputs=inputs, outputs=outputs, name='cnn_model')

    cnnModel.compile(
        loss=rmse,
        optimizer=tf.keras.optimizers.Adam(),
        metrics=['mae', rmse]
    )

    return cnnModel
```

Listagem 4.2: Excerto de código da construção do modelo CNN.

Após a implementação do modelo, procedeu-se à avaliação e seleção dos melhores intervalos de hiperparâmetros, bem como à determinação do número ideal de *epochs*, seguindo a mesma abordagem utilizada para o modelo discutido anteriormente neste capítulo, com o número de *epochs* de 70 para %CH4 e 120 para Xc. Para finalizar, os dados foram transformados para o intervalo [0,1], garantindo a normalização adequada antes de serem utilizados nas experiências com a CNN.

4.1.3 MLP

Na construção do modelo MLP, seguiu-se uma abordagem estruturada semelhante aos modelos anteriormente descritos. Iniciou-se com a definição da *seed* para garantir a reprodutibilidade dos resultados, seguida da configuração dos vetores de entrada (*inputs*) e do processo de *flattening* para transformar os dados temporais em uma forma adequada para as camadas densas, como visto na imagem seguinte.

O modelo foi projetado com múltiplas camadas densas, controladas pelos parâmetros *layers* e *neurons*. A primeira camada densa recebe os dados transformados, e dependendo da profundidade do modelo, pode aplicar diferentes configurações de neurónios. As camadas intermediárias utilizam metade dos neurónios da camada inicial, enquanto a última camada densa aplica o dobro dos neurónios, permitindo ao modelo capturar padrões mais complexos à medida que a rede se aprofunda, como visto na listagem 4.3. Além disso, foi integrada uma camada de *dropout* para reduzir o risco de *overfitting*, desativando aleatoriamente uma percentagem dos neurónios durante o treino. A camada de saída foi configurada com um único neurónio, destinada a produzir a previsão final. O modelo foi compilado utilizando a função de perda RMSE e o otimizador Adam.

```
1
2  def build_mlp_model(features, timesteps=7, layers=2, neurons=64, activation='relu',
       ↪ dropout_rate=0.5, log=False,
3                      seed=91195003):
4      tf.random.set_seed(seed)
5
6      inputs = Input(shape=(timesteps, features))
7      f = Flatten()(inputs)
8
9      for i in range(layers):
10         # if first Dense layer
11         if i == 0:
12             # and also the last
13             if i + 1 == layers:
14                 x = Dense(neurons, activation=activation)(f)
15             # it has more layers
16             else:
17                 x = Dense(neurons / 2, activation=activation)(f)
```

```
        # if last Dense layer
        elif i + 1 == layers:
            x = Dense(neurons * 2, activation=activation)(x)
        # if middle layers
        else:
            x = Dense(neurons, activation=activation)(x)

    x = Dropout(dropout_rate)(x)
    outputs = Dense(1)(x)

    mlpModel = Model(inputs=inputs, outputs=outputs, name='mlp_model')

    mlpModel.compile(
        loss=rmse,
        optimizer=tf.keras.optimizers.Adam(),
        metrics=['mae', rmse]
    )


    return mlpModel
```

Listagem 4.3: Excerto de código da construção do modelo MLP.

Mais uma vez, após a implementação do modelo, foi realizada a avaliação e seleção dos melhores intervalos de hiperparâmetros, assim como a escolha do número ideal de *epochs* 120 para %CH4 e 175 para Xc, seguindo a mesma metodologia aplicada aos modelos previamente discutidos neste capítulo. Para concluir, os dados foram normalizados no intervalo [0,1], assegurando a preparação adequada antes de serem utilizados nas experiências com a MLP.

4.1.4 LSTNet

Na construção do modelo LSTNet, combinou-se o uso de camadas de convolução e camadas LSTM, integrando duas abordagens populares de redes neurais para capturar padrões tanto locais quanto de longo prazo em séries temporais.

A implementação começou com a definição da *seed* para garantir a reprodutibilidade dos resultados. Em seguida, foi configurado os vetores de entrada através da camada *Input*, que define a estrutura dos dados de entrada, consistindo em *timesteps* e *features*. As camadas de convolução (CNN) foram então aplicadas para extrair características locais dos dados temporais, usando um número especificado de filtros (*filters*) e o tamanho da janela de convolução (*kernel_size*). Estas camadas foram ajustadas com base no número de camadas (*layers*) configuradas, permitindo capturar padrões a diferentes níveis de granularidade, tal como verificado na listagem 4.4.

Após as camadas de convolução, o modelo prosseguiu com as camadas LSTM, configuradas para capturar dependências de longo prazo na série temporal. O número de unidades *neurons* foi adaptado

de acordo com a profundidade da camada, variando para ajustar a capacidade de captura de padrões complexos. Em seguida, uma camada de *dropout* foi aplicada para reduzir o risco de *overfitting*.

Finalmente, a camada de saída foi configurada usando *Dense*, que fornece a previsão final do modelo. O modelo foi compilado com a função de perda RMSE e o otimizador Adam. Tal como os restantes modelos, o modelo oferece a opção de gerar um resumo detalhado da arquitetura e salvar uma visualização gráfica da rede, facilitando a análise e interpretação dos resultados.

```
1
2  def build_lstnet_model(features, timesteps=7, layers=3, num_channels=32, kernel_size=3,
       ↪ lstm_units=64, output_dim=1,
3                         neurons=64, dropout_rate=0.5, activation='relu', filters=32, seed
       ↪ =91195003, log=False):
4      tf.random.set_seed(seed)
5
6      kernel_stepsize = kernel_size - 1
7      total_kernel_stepsize = kernel_stepsize * layers
8      if timesteps - total_kernel_stepsize <= 0:
9          return None
10
11     inputs = Input(shape=(timesteps, features))
12
13     for i in range(layers):
14         # if first CNN layer
15         if i == 0:
16             # and also the last
17             if i + 1 == layers:
18                 x = Conv1D(filters=filters / 2, kernel_size=kernel_size, activation=activation
       ↪ ,
19                            data_format='channels_last')(inputs)
20             # it has more layers
21             else:
22                 x = Conv1D(filters=filters, kernel_size=kernel_size, activation=activation,
23                            data_format='channels_last')(inputs)
24         # if last layer
25         elif i + 1 == layers:
26             x = Conv1D(filters=filters * 2, kernel_size=kernel_size, activation=activation,
27                        data_format='channels_last')(x)
28         # if middle layers
29         else:
30             x = Conv1D(filters=filters, kernel_size=kernel_size, activation=activation,
       ↪ data_format='channels_last')(x)
31
32     for i in range(layers):
33         #if first LSTM layer
34         if i == 0:
```

```
            #if also the last LSTM layer then return_sequences=False
            if i+1 == layers:
                x = LSTM(neurons/2, return_sequences=False)(inputs)
            #it has more layers! So return_sequences=True
            else:
                x = LSTM(neurons, return_sequences=True)(inputs)
        #if last LSTM layer then return_sequences=False
        elif i+1 == layers:
            x = LSTM(neurons*2, return_sequences=False)(x)
        #if not the last LSTM layer then return_sequences=True
        else:
            x = LSTM(neurons, return_sequences=True)(x)

      # Camada Dropout
    x = Dropout(rate=dropout_rate)(x)

    # Camada de saída
    outputs = Dense(output_dim)(x)

    lstnet_model = Model(inputs=inputs, outputs=outputs, name='lstnet_model')

    lstnet_model.compile(
        loss='mean_squared_error',
        optimizer=Adam(),
        metrics=['mae']
    )

    return lstnet_model
```

Listagem 4.4: Excerto de código da construção do modelo LSTNet.

Após a implementação do modelo LSTNet, avaliou-se os melhores intervalos de hiperparâmetros e o número ideal de *epochs*, 150 para %CH4 e 300 para Xc, seguindo a mesma metodologia dos modelos anteriores. Por fim, os dados foram normalizados no intervalo [0,1] antes das experiências com o LSTNet.

4.1.5 TCN

Na construção do modelo TCN, foi utilizada uma abordagem focada em capturar padrões temporais de curto e longo prazo através de convoluções dilatadas, ajustadas para processar séries temporais de forma eficaz. Este modelo baseia-se na aplicação de convoluções temporais causais, permitindo que o modelo capture informações em várias escalas temporais de forma eficiente.

A implementação iniciou-se de forma semelahnte às anteriores, com a definição da *seed*, seguida da configuração dos vetores de entrada através da camada Input. A arquitetura do modelo foi construída com camadas convolucionais causais (TCN), ajustadas de acordo com o número de filtros (*filters*), o tamanho

da janela de convolução (*kernel_size*), e as dilatações, que permitem ao modelo expandir a sua visão temporal à medida que a profundidade aumenta. O uso de múltiplas pilhas (*stacks*) assegura que as convoluções podem captar padrões a diferentes níveis de granularidade.

A camada final é uma camada Dense, que gera a previsão a partir das características extraídas pelas camadas convolucionais. A regularização foi implementada através de uma camada *Dropout*, que ajuda a evitar o *overfitting*, como visto em 4.5. O modelo foi compilado utilizando a função de perda RMSE e o otimizador Adam. Assim como nos outros modelos, a arquitetura permite a visualização e o resumo detalhado do modelo, facilitando a análise do seu desempenho.

```
1
2  def build_TCN_model(features, timesteps=7, layers=2, activation='relu', dropout_rate=0.5,
       ↪ nb_filters=16, kernel_size=3,
3                      dilations=[1], nb_stacks = 2, padding='causal', kernel_initializer='
       ↪ glorot_uniform', log=False, seed=91195003):
4      tf.random.set_seed(seed)
5
6      inputs = Input(shape=(timesteps, features))
7      x = inputs
8
9      for i in range(layers): #  nb_stacks = 2
10
11         if i == 0:
12             if i + 1 == layers:
13                 x = TCN(nb_filters=int(nb_filters / 2),
14                         kernel_size=(kernel_size,),
15                         dilations=dilations,
16                         padding=padding,
17                         dropout_rate=dropout_rate,
18                         return_sequences=False,
19                         activation=activation,
20                         kernel_initializer=kernel_initializer) (x)
21
22
23             else:
24                 x = TCN(nb_filters=nb_filters,
25                         kernel_size=(kernel_size,),
26                         dilations=dilations,
27                         padding=padding,
28                         dropout_rate=dropout_rate,
29                         return_sequences=True,
30                         activation=activation,
31                         kernel_initializer=kernel_initializer) (x)
32
33         else:
34             if i + 1 == layers:
```

```
                x = TCN(nb_filters=nb_filters * 2,
                        kernel_size=(kernel_size,),
                        dilations=dilations,
                        padding=padding,
                        dropout_rate=dropout_rate,
                        return_sequences=False,
                        activation=activation,
                        kernel_initializer=kernel_initializer) (x)

            else:
                x = TCN(nb_filters=nb_filters,
                        kernel_size=(kernel_size,),
                        dilations=dilations,
                        padding=padding,
                        dropout_rate=dropout_rate,
                        return_sequences=True,
                        activation=activation,
                        kernel_initializer=kernel_initializer) (x)

    x = Dense(nb_filters, activation='softmax')(x)


    outputs = Dense(1)(x)


    TCNModel = Model(inputs=inputs, outputs=outputs, name='tcn_model')
    # TCNModel.summary()

    TCNModel.compile(
        loss=rmse,
        optimizer=tf.keras.optimizers.Adam(),
        metrics=['mae', rmse]
    )

    return TCNModel
```

Listagem 4.5: Excerto de código da construção do modelo TCN.

Mais uma vez, e como supramencionado, foi realizado uma avaliação dos melhores intervalos de hiperparâmetros e o número ideal de *epochs* 40 para %CH4 e 30 para Xc , foi foita a normalização dos dados no intervalo de tempo [0,1].

4.1.6 GRUS

Por fim, foi concebido o modelo GRU, no qual foram seguidos procedimentos semelhantes aos utilizados para os outros modelos discutidos. A construção do modelo iniciou-se com a definição da *seed* para assegurar a reprodutibilidade dos resultados e a configuração dos vetores de entrada. O modelo foi estruturado com camadas GRU, ajustando o número de camadas e neurónios para otimizar a captura de padrões temporais. Após as camadas GRU, foram adicionadas uma camada densa e uma camada de *dropout* para evitar *overfitting* e melhorar a generalização, como visto em 4.6.

```
1
2  def build_gru_model(features, timesteps=7, layers=2, neurons=64, activation='relu',
       ↪ dropout_rate=0.5, log=False,
3                      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 GRU layer
10         if i == 0:
11             # if also the last GRU layer then return_sequences=False
12             if i + 1 == layers:
13                 x = GRU(neurons / 2, return_sequences=False)(inputs)
14             # it has more layers! So return_sequences=True
15             else:
16                 x = GRU(neurons, return_sequences=True)(inputs)
17         # if last GRU layer then return_sequences=False
18         elif i + 1 == layers:
19             x = GRU(neurons * 2, return_sequences=False)(x)
20         # if not the last GRU layer then return_sequences=True
21         else:
22             x = GRU(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     gruModel = Model(inputs=inputs, outputs=outputs, name='gru_model')
29
30     gruModel.compile(
31         loss=rmse,
32         optimizer=tf.keras.optimizers.Adam(),
33         metrics=['mae', rmse]
34     )
35
36     if log:
```

```
37          gruModel.summary()
38          # save the model's plot
39          from pathlib import Path
40          Path('results').mkdir(parents=True, exist_ok=True)
41          plot_model(gruModel, 'results/gruModel.png', show_shapes=True)
42
43      return gruModel
```

Listagem 4.6: Excerto de código da construção do modelo GRU.

A avaliação dos hiperparâmetros e a seleção do número ideal de *epochs* 130 para %CH4 e 190 para Xc, foram realizadas conforme a metodologia aplicada aos modelos anteriores. Finalmente, os dados foram normalizados no intervalo [0,1] antes das experiências com o modelo GRU, garantindo uma preparação adequada e consistência na análise dos resultados.

4.2 Métricas de Avaliação

Após a implementação dos modelos e a otimização dos seus hiperparâmetros, um processo que envolveu várias experiências, torna-se essencial analisar os resultados obtidos para avaliar quais são os melhores candidatos. Esse procedimento permite determinar quais modelos, entre os testados, se destacam como os mais adequados. Assim, é seguido um processo rigoroso de seleção para identificar os modelos mais promissores.

Dado que a previsão dos *targets*, como %CH4 e Xc, constitui um problema de regressão linear, foram utilizadas duas métricas de avaliação neste estudo: o MAE e a RMSE. Ambas as métricas baseiam-se no cálculo dos erros absolutos das previsões, comparando os valores previstos com os valores reais, através da diferença absoluta entre eles.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.1}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.2}$$

Capítulo 5

Resultados e Discussão

Este capítulo apresenta os resultados obtidos a partir dos modelos discutidos no capítulo anterior, bem como a análise e discussão dos mesmos. No total, foram utilizados seis modelos distintos, cuja performance foi avaliada com base nas previsões realizadas.

O capítulo está estruturado de forma a apresentar os resultados relativos aos *targets* deste trabalho, nomeadamente o %CH4 e o Xc. Para isso, foram criadas duas secções principais: 5.1, dedicada ao CH4, e 5.2, referente ao Xc.

Cada uma destas secções está subdividida de acordo com os modelos de DL considerados, sendo que cada secção principal contém 6 subsecções, nas quais são analisados os resultados para cada modelo. Assim, as subsecções de 5.1 referem-se aos modelos aplicados ao %CH4, enquanto as subsecções de 5.2 tratam dos resultados para o Xc.

Por fim, em cada secção principal, existe uma subsecção adicional (5.1.7 e 5.2.7) onde é realizada uma análise comparativa entre os modelos, identificando o melhor modelo candidato para cada substância. Esta análise inclui ainda a apresentação de uma figura, na qual são visualizadas as previsões feitas pelo modelo selecionado.

5.1 %CH4

Nesta subsecção, são apresentados os resultados de todas as experiências realizadas para a previsão dos níveis de %CH4. Tal como referido no capítulo anterior, foram implementados seis modelos distintos de DL para esta tarefa.

Cada um dos seis modelos gerou um conjunto de previsões que foram sistematicamente analisadas. Desta forma, são apresentadas seis tabelas de resultados, correspondendo a cada um dos modelos utilizados, onde se destacam as métricas de desempenho mais relevantes para a avaliação das previsões de %CH4.

Após a apresentação individual dos resultados, é realizada uma análise comparativa que visa identificar o modelo que demonstrou o melhor desempenho em termos preditivos, para tal são apresentados os melhores resultados de cada modelo num gráfico. Nesta fase final, será indicado o modelo candidato mais adequado para a previsão de %CH4, baseado nas métricas avaliadas. Além disso, será apresentada uma figura que ilustra as previsões deste modelo vencedor, demonstrando a sua capacidade preditiva ao longo de dois *timesteps* futuros.

5.1.1 LSTM

Relativamente aos modelos LSTMs, a Tabela 5.1, sumariza os resultados para os melhores 5 modelos candidato.

a.	b.	c.	d.	e.	f.	g.	h.	i.	j.	k.
3	128	0.5	relu	7	10	75	0.1582	0.1673	80.91	
3	128	0.0	tanh	14	10	75	0.1997	0.2039	86.06	
5	128	0.5	tanh	7	5	75	0.2003	0.2134	240.90	
3	32	0.5	tanh	7	5	75	0.2258	0.2382	153.06	
4	128	0.5	relu	14	10	75	0.2465	0.2488	107.78	

Tabela 5.1: 5 melhores resultados do modelo LSTM. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s).

Analisando os resultados, a primeira experiência destaca-se como a melhor entre as cinco, com um MAE de 0.1582% e um RMSE de 0.1673%, ambos os menores valores obtidos. Além disso, esta experiência registou um tempo de processamento relativamente baixo de 80.91 segundos, o que a torna também uma das mais eficientes. Esta configuração utilizou 3 *layers*, 128 neurónios, uma taxa de *dropout* de 0.5 e a função de ativação *relu*, sugerindo que esta combinação proporcionou um ótimo equilíbrio entre a complexidade do modelo e a sua capacidade de generalização.

Comparando as demais experiências, observa-se uma certa homogeneidade nos parâmetros, como o número de *neurons* (128), que são constantes em quase todas as linhas, exceto pela quarta, que utilizou apenas 32 neurónios. Outros fatores, como o valor de *dropout* e os *timesteps*, revelam ser bastantes presentes entre os melhores resultados. Em termos de funções de ativação, os resultados não indicam uma clara superioridade de uma função sobre a outra. A *relu* foi utilizada tanto na melhor (0.1582 de MAE) quanto na pior experiência (0.2465 de MAE), e a *tanh* mostrou desempenhos variados.

5.1.2 CNN

De forma semelhante ao supramencionado, a Tabela 5.2 ilustra os melhores resultados dos modelos candidatos no âmbito da previsão de %CH4, com a utilização de modelos CNNs.

a.	b.	c.	d.	e.	f.	g.	h.	i.	j.	k.	l.
4	0.5	tanh	14	20	32	4	3	70	0.0611	0.0667	50.62
3	0.0	relu	14	20	32	5	2	70	0.0626	0.0675	32.70
3	0.5	relu	14	20	16	4	2	70	0.0792	0.0831	48.13
5	0.5	tanh	14	10	16	3	2	70	0.0819	0.0852	57.60
4	0.5	tanh	14	10	16	3	3	70	0.0915	0.0925	47.30

Tabela 5.2: 5 melhores resultados do modelo CNN. Significado das letras: a - *layers*, b - *dropout_rate*, c - *activation*, d - *timesteps*, e - *batch_size*, f - *filters*, g - *kernel_size*, h- *pool_size*, i - *epochs*, j - MAE, k - RMSE, l - tempo(s).

A primeira linha representa o melhor modelo candidato, com uma combinação de parâmetros que proporcionou um desempenho otimizado. Nesta experiência, foram usadas 4 camadas, uma taxa de *dropout* de 0.5 e a função de ativação *tanh*, que, juntamente com 32 *filters*, revelou um valor de MAE de 0.0611% e RMSE de 0.0667%. O tempo total de execução foi de 50,62 segundos.

Comparando com as outros modelos candidatos, nota-se que o segundo modelo apresenta uma taxa de *dropout* de 0.0 e função *relu*, o que resultou em uma diminuição significativo no tempo (32.70 s), mas um desempenho inferior (MAE de 0.0626% e RMSE de 0.0675%). O quarto modelo candidato, com 5 camadas e uma configuração semelhante à primeira, revelou o maior tempo de execução (57.60 s), porém não obteve erros menores. Assim, a combinação dos parâmetros no melhor modelo candidato destaca-se como a mais eficiente em termos de balanço entre tempo e precisão.

5.1.3 MLP

Relativamente aos modelos MLPs, a Tabela 5.3, sumariza os resultados para os melhores 5 modelos candidatos.

a.	b.	c.	d.	e.	f.	g.	h.	i.	j.
4	128	0.0	relu	7	5	120	0.1330	0.1419	70.55
3	32	0.5	tanh	14	5	120	0.1447	0.1512	64.24
3	128	0.5	relu	7	5	120	0.1464	0.1560	71.79
3	128	0.0	relu	7	20	120	0.1412	0.1564	39.90
4	32	0.0	tanh	7	20	120	0.1480	0.1583	36.60

Tabela 5.3: 5 melhores resultados do modelo MLP. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s).

Nesta análise, é possível observar que a primeira experiência apresentou o melhor desempenho geral, com o menor valor de MAE (0.1330%) e RMSE (0.1419%). Além disso, o tempo de execução dessa experiência foi de 70.55 segundos. Essa configuração utilizou 4 *layers*, 128 neurónios, uma taxa de *dropout*

de 0.0 e a função de ativação *relu*, sugerindo que a combinação de uma arquitetura mais profunda (4 *layers*) e a ativação *relu* sem *dropout* contribuiu para a melhoria do desempenho do modelo.

Verifica-se novamente uma certa homogeneidade no número de *timesteps*. No entanto, o segundo melhor modelo candidato, que utilizou apenas 32 neurónios e a função de ativação *tanh*, obteve um MAE e RMSE ligeiramente mais altos (0.1447% e 0.1512%, respetivamente) relativamente com o melhor modelo supramencionado, mas compensou com o menor tempo de treino (64.24 segundos). Esta variação poerá indicar que a redução no número de neurónios e a escolha da função de ativação podem afetar a precisão dos resultados, mas reduzir o tempo de execução. A função *relu*, presente em três das cinco experiências, foi novamente associada aos melhores resultados em termos de erro, sugerindo que ela poderá ser mais eficaz para este modelo de previsão de %CH4.

5.1.4 LSTNet

A Tabela 5.4 representa os melhores resultados dos modelos candidatos no âmbito da previsão de %CH4, com a utilização de modelos LSTNet.

a.	b.	c.	d.	e.	f.	g.	h.	i.	j.	k.	l.
3	32	0.5	relu	21	5	16	3	150	0.2198	0.2318	150.99
3	32	0.5	tanh	21	5	16	5	150	0.2547	0.2603	146.31
5	128	0.0	relu	14	20	32	3	150	0.2616	0.2632	69.75
3	32	0.5	relu	14	20	32	3	150	0.2701	0.2744	50.94
5	128	0.5	tanh	14	20	32	3	150	0.2811	0.2837	70.06

Tabela 5.4: 5 melhores resultados do modelo LSTNet. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *filters*, h - *kernel_size*, i - *epochs*, j - MAE, k - RMSE, l - tempo(s).

O primeiro modelo destaca-se como o melhor, com um MAE de 0.2198% e RMSE de 0.2318%. O tempo de execução foi de 150.99 segundos, sendo o segundo mais alto da tabela, o que indica que, apesar de não ser o mais rápido, este modelo foi o que obteve a melhor performance. Esta configuração utilizou 3 *layers*, 32 neurónios, uma taxa de *dropout* de 0.5 e a função de ativação *relu*.

Comparando as restantes experiências, observa-se que a terceira apresenta o menor tempo de treino (69.75 segundos), mas com um desempenho inferior em termos de erro (MAE de 0.2616 e RMSE de 0.2632), provavelmente devido ao uso de um número maior de *neurons* (128) e ausência de *dropout* (0.0), o que pode ter levado a uma sobreajustamento do modelo. Em termos de funções de ativação, a *relu* apresentou-se novamente superior em relação ao melhor modelo, visto que as duas experiências com a função *tanh* tiveram um desempenho inferior, com maiores valores de MAE e RMSE.

5.1.5 TCN

A Tabela 5.5 representa os melhores resultados dos modelos candidatos no âmbito da previsão de %CH4, com a utilização de modelos TCN.

a.	b.	c.	d.	e.	f.	g.	h.	i.	j.
3	0.0	tanh	7	20	3	40	0.2868	0.2953	45.93
3	0.0	relu	7	20	5	40	0.3138	0.3246	43.47
3	0.0	tanh	7	20	4	40	0.3216	0.3306	46.25
4	0.0	tanh	7	20	4	40	0.3561	0.3629	53.16
5	0.0	relu	7	10	3	40	0.3600	0.3666	70.37

Tabela 5.5: 5 melhores resultados do modelo TCN. Significado das letras: a - *layers*, b - *dropout_rate*, c - *activation*, d - *timesteps*, e - *batch_size*, f - *kernel_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s).

Relativamente aos melhores modelos candidato TCN, o primeiro modelo sobressai como a melhor entre as cinco, apresentando o menor MAE (0.2868%) e RMSE (0.2953%). Com um tempo de treino de 45.93 segundos, esta experiência combina uma configuração de 3 *layers*, taxa de *dropout* de 0.0 e a função de ativação *tanh*, juntamente com 7 *timesteps*, 20 de *batch size*, e 3 no *kernel size*, sendo esta combinação a mais eficiente para a tarefa de previsão de %CH4 com o modelo TCN.

Observa-se que o uso da função de ativação *tanh* está presente nos três resultados dos cinco melhores em termos de erro. Verifica-se que a última experiência, que também teve o maior tempo (70.37 segundos) representa também o pior valor de MAE e RMSE. Vale destacar que a homogeneidade nos parâmetros *timesteps* (7), *batch size* (20) e *dropout* mostra que essas variáveis não tiveram grande impacto no desempenho geral, mas sim a combinação entre a função de ativação e o número de *layers*.

5.1.6 GRU

Por fim e de forma semelhante ao supramencionado, a Tabela 5.6 ilustra os melhores resultados dos modelos candidatos no âmbito da previsão de %CH4, com a utilização de modelos GRU.

a.	b.	c.	d.	e.	f.	g.	h.	i.	j.
4	32	0.5	relu	14	5	130	0.0730	0.0739	148.18
5	64	0.5	tanh	21	5	130	0.1171	0.1241	176.69
5	32	0.0	tanh	7	5	130	0.1244	0.1259	168.65
5	64	0.5	relu	14	10	130	0.1319	0.1377	95.84
4	32	0.5	relu	21	10	130	0.1351	0.1447	92.89

Tabela 5.6: 5 melhores resultados do modelo GRU. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s).

Nesta análise é possível observar que a primeira experiência se destaca como a melhor entre as cinco apresentadas na tabela, apresenta um tempo de treino de 148.18 segundos. Este modelo candidato utilizou 4 *layers* e 32 neurónios, com uma taxa de *dropout* de 0.5 e a função de ativação *relu*, apresentando

um MAE de 0.0730% e um RMSE de 0.0739%. Estes valores de erro são os mais baixos entre os cinco melhores modelos candidatos, indicando uma melhor performance do modelo na previsão de %CH4 em relação às restantes.

Em termos de homogeneidade, podemos observar que a maioria dos modelos candidatos utiliza 5 *batch_size*, com exceção do quarto e do quinto, que utilizam 10 *batch_size*, o que pode ter contribuído para os resultados ligeiramente piores. Também é relevante destacar que a função de ativação *relu* foi utilizada em três das cinco experiências, enquanto a *tanh* foi utilizada nas outras duas.

5.1.7 Análise Comparativa

Após a execução de todos os modelos, foi necessário realizar uma análise comparativa dos resultados obtidos, conforme ilustrado na Figura 5.1. Os valores exibem os melhores resultados de previsão de cada modelo para o %CH4.

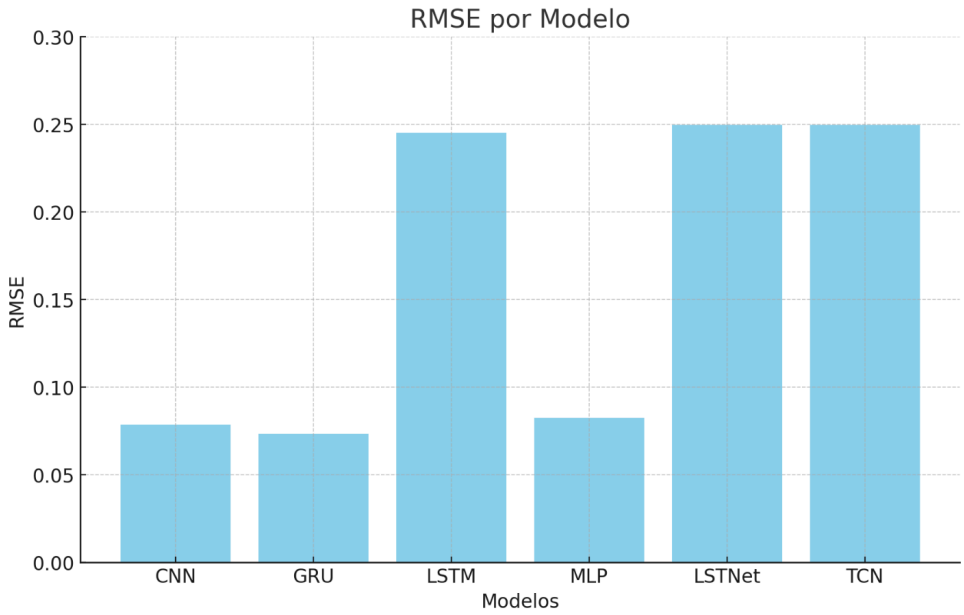


Figura 5.1: Comparação dos melhores resultados obtidos na previsão da %CH4.

Ao analisar os resultados do histograma, verifica-se que o modelo CNN obteve o melhor desempenho, apresentando o menor RMSE, com um valor de 0.0667%. Seguindo-se o modelo GRU, que obteve um RMSE de 0.0739, ambos os modelos destacam-se por terem erros relativamente baixos na previsão de %CH4.

Os modelos MLP e LSTM apresentaram desempenhos intermediários, com RMSEs de 0.1419 e 0.1673, respetivamente. Esses valores mostram que, apesar de serem modelos robustos, têm um erro maior em comparação com os modelos CNN e GRU.

No entanto, os modelos LSTNet e TCN tiveram desempenhos inferiores, com RMSEs significativamente mais elevados, sendo 0.2198 para o LSTNet e 0.2953 para o TCN. Esses valores indicam uma maior margem de erro na previsão de %CH4, sugerindo que esses modelos são menos adequados para esta tarefa em particular.

Por fim, após concluir que o melhor modelo candidato para a previsão de produção de %CH4 é o baseado em CNN, a Figura 5.2 apresenta uma ilustração das suas previsões. Embora durante a etapa de validação tenham sido considerados apenas dois *multi-steps*, a figura demonstra a capacidade do modelo em realizar previsões recursivas para quatro passos à frente (*multi-steps*), utilizando 14 *timesteps* de entrada. Este resultado destaca a robustez do modelo em prever séries temporais com horizonte de previsão alargado, evidenciando a sua aplicabilidade no contexto de otimização do reator digestor anaeróbio.

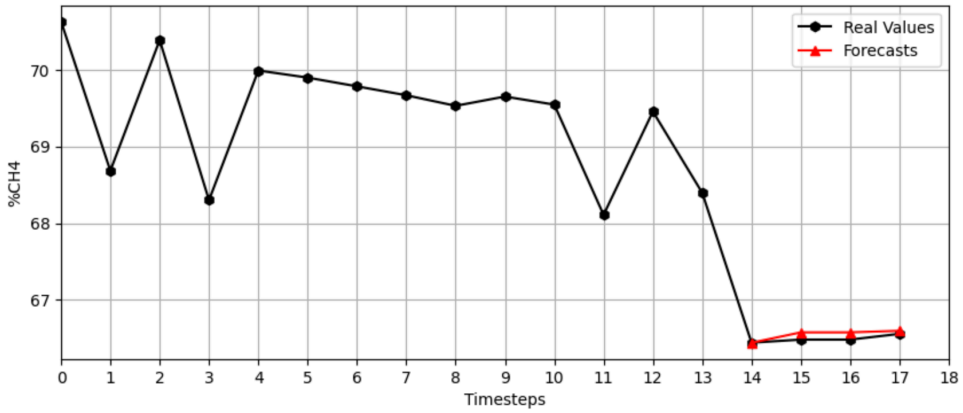


Figura 5.2: Previsões de %CH4 para 2 *timesteps* futuros.

5.2 Xc

Nesta subseção, assim como na anterior, são apresentados os resultados de todas as experiências realizadas para a previsão dos níveis de Xc. Novamente, foram implementados seis modelos distintos de DL para essa tarefa.

Cada um dos seis modelos gerou um conjunto de previsões que foram sistematicamente analisadas. Seguidamente são apresentadas novamente seis tabelas de resultados, correspondentes a cada um dos modelos utilizados, destacando-se as métricas de desempenho mais relevantes para a avaliação das previsões de Xc.

Após a apresentação individual dos resultados, será realizada uma análise comparativa entre os diferentes modelos, como feita na secção anterior.

5.2.1 LSTM

A Tabela 5.7 representa os melhores resultados dos modelos candidatos no âmbito da previsão de Xc, com a utilização de modelos LSTM.

a.	b.	c.	d.	e.	f.	g.	h.	i.	j.
3	64	0.5	tanh	7	10	50	1660.64	1712.58	70.28
3	64	0.0	relu	7	10	50	1696.25	1743.82	68.19
5	128	0.0	relu	7	10	50	1789.40	1805.24	102.88
3	64	0.5	relu	7	10	50	1856.09	1871.91	72.66
3	32	0.0	tanh	7	10	50	1797.51	1902.81	67.59

Tabela 5.7: 5 melhores resultados do modelo LSTM. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s).

A primeira experiência destaca-se com os menores valores de MAE (1660.64 mgO2/L) e RMSE (1712.58 mgO2/L), sendo também a terceira mais rápida (70.28 segundos). Esta configuração utilizou 3 *layers*, 64 neurónios, *dropout rate* de 0.5, e a função de ativação *tanh*.

Observa-se uma homogeneidade nos *timesteps* (7) e *batch size* (10), que são constantes em todas as experiências, sugerindo que esses hiperparâmetros desempenham um papel estabilizador no desempenho do modelo LSTM. No entanto, há variações no número de *layers* e no *dropout rate*, sendo que a terceira experiência, com 5 *layers* e 128 neurónios, resultou no maior tempo de execução (102.88 segundos) e apresentou os piores valores de erro (MAE de 1789.40 mgO2/L e RMSE de 1805.24 mgO2/L) em relação ao melhor modelo candidato. Este fato reforça que arquiteturas mais complexas nem sempre oferecem melhor desempenho. A segunda experiência, com *relu* e sem *dropout*, também teve um desempenho satisfatório, mas com maior erro comparado à primeira, o que destaca a eficiência da função *tanh* com *dropout*.

5.2.2 CNN

A Tabela 5.8 representa os melhores resultados dos modelos candidatos no âmbito da previsão de Xc, com a utilização de modelos CNN.

a.	b.	c.	d.	e.	f.	g.	h.	i.	j.	k.	l.
3	0.5	relu	21	20	16	3	3	120	1007.24	1028.06	38.98
5	0.0	tanh	21	20	16	4	3	120	1003.71	1032.79	40.39
5	0.5	tanh	21	20	32	3	2	120	1048.16	1081.17	46.48
4	0.5	relu	21	10	16	3	3	120	1071.69	1093.89	49.83
4	0.0	tanh	21	20	16	3	2	120	1066.74	1098.84	40.56

Tabela 5.8: 5 melhores resultados do modelo CNN. Significado das letras: a - *layers*, b - *dropout_rate*, c - *activation*, d - *timesteps*, e - *batch_size*, f - *filters*, g - *kernel_size*, h- *pool_size*, i - *epochs*, j - MAE, k - RMSE, l - tempo(s).

A primeira experiência destaca-se como a melhor entre as cinco, apresentando os menores valores de MAE (1007.24 mgO2/L) e RMSE (1028.06 mgO2/L). Esta configuração utilizou 3 *layers*, *dropout rate* de 0.5, a função de ativação *relu* e 21 *timesteps*, com um tempo de treino de 38.98 segundos, o mais baixo entre as cinco experiências. Estes resultados sugerem que essa combinação de hiperparâmetros proporcionou uma melhor eficiência na previsão de Xc.

Os resultados também revelam certa consistência nos parâmetros *timesteps* (21), *batch size* (20), e *filters* (16), presentes na maioria dos modelos candidatos, indicando que esses hiperparâmetros são relativamente homogêneos entre as melhores configurações. No entanto, variações nas funções de ativação (*relu* e *tanh*), *dropout rates* e o número de *layers* parecem ser os principais responsáveis pelas diferenças nos erros, com a função *relu* demonstrando ligeira vantagem no melhor desempenho, relativamente ao melhor modelo candidato. A experiência de menor tempo de treino (38.98 segundos), isto é o melhor modelo, também utilizou *relu* com 3 *layers*, reforçando o papel dessa configuração em otimizar o tempo de execução.

5.2.3 MLP

De forma semelhante aos restantes modelos, a Tabela 5.9 ilustra os melhores resultados dos modelos candidatos no âmbito da previsão de Xc, com a utilização de modelos MLP.

a.	b.	c.	d.	e.	f.	g.	h.	i.	j.
4	64	0.5	relu	21	20	175	1689.24	1767.91	46.81
5	32	0.5	relu	21	20	175	1879.50	1901.29	35.51
3	32	0.5	relu	21	10	175	1828.07	1921.74	46.97
5	32	0.0	relu	14	20	175	1899.52	1934.68	29.90
3	128	0.0	tanh	7	20	175	2115.91	2130.06	31.06

Tabela 5.9: 5 melhores resultados do modelo MLP. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s).

De forma semelhante aos modelos supramencionados, o primeiro modelo candidato destaca-se com os menores valores de MAE (1689.24 mgO2/L) e RMSE (1767.91 mgO2/L), além de um tempo de treino razoável de 46.81 segundos. Esta configuração utilizou 4 *layers*, 64 neurónios, *dropout rate* de 0.5, e a função de ativação *relu*.

Entre as cinco experiências, observa-se uma homogeneidade nos parâmetros *timesteps* (21) e *batch size* (20), que permanecem constantes em três das cinco experiências. No entanto, variações no número de *neurónios*, *dropout rate* e função de ativação revelam-se importantes para o desempenho. A quinta experiência, que utiliza a função *tanh* e 128 neurónios, apresenta os maiores valores de erro (MAE de 2115.91 mgO2/L e RMSE de 2130.06 mgO2/L), destacando a inferioridade desta configuração em relação às demais. A quarta experiência, com o menor tempo de treino (29.90 segundos), utiliza 5 *layers*, *relu* e sem *dropout*, demonstrando que essa configuração pode ser vantajosa para otimização de tempo, embora com menor precisão.

5.2.4 LSTNet

A Tabela 5.10 representa os melhores resultados dos modelos candidatos no âmbito da previsão de Xc, com a utilização de modelos LSTNet.

a.	b.	c.	d.	e.	f.	g.	h.	i.	j.	k.	l.
3	128	0.5	relu	21	10	16	3	300	10032.26	10039.63	80.38
3	64	0.5	tanh	21	10	16	4	300	10266.97	10272.78	80.82
4	32	0.0	tanh	21	5	16	5	300	10584.48	10589.80	178.48
5	32	0.0	tanh	12	20	32	3	300	1013.49	1081.85	65.13
4	32	0.0	relu	14	5	32	4	300	11020.07	11088.25	152.57

Tabela 5.10: 5 melhores resultados do modelo LSTNet. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *filters*, h - *kernel_size*, i - *epochs*, j - MAE, k - RMSE, l - tempo(s).

É possível observar que o modelo LSTNet atingiu o desempenho mais eficiente na primeira experiência. Essa configuração utilizou 3 *layers*, 128 *neurons*, *dropout rate* de 0.5 e a função de ativação *relu*, obtendo os menores valores de MAE (10032.26 mgO2/L) e RMSE (10039.63 mgO2/L), com um tempo de execução de 80.38 segundos.

Os *timesteps* foram constantes em 21 na maioria das experiências, exceto na terceira (24) e na quinta (14). A função de ativação *tanh* apareceu em três das cinco experiências, mas foi a função *relu* que produziu o melhor resultado. O número de *layers* variou de 3 a 5, mas não há uma correlação direta entre a quantidade de *layers* e a melhoria de desempenho.

Os valores de *batch size* variaram entre 5 e 20, enquanto os *filters* mantiveram-se em 16 ou 32, com algum impacto evidente na diferença de desempenho. O tempo de treino variou significativamente, com o menor tempo sendo de 65.13 segundos, enquanto o maior tempo foi de 178.48 segundos.

5.2.5 TCN

A Tabela 5.11 representa os melhores resultados dos modelos candidatos no âmbito da previsão de Xc, com a utilização de modelos TCN.

a.	b.	c.	d.	e.	f.	g.	h.	i.	j.
5	0.0	relu	7	20	4	30	2361.88	2386.93	53.60
3	0.0	relu	7	10	3	30	2828.48	2842.64	50.19
3	0.0	relu	21	20	4	30	2980.48	2988.78	39.07
3	0.0	relu	7	5	5	30	3029.69	3063.19	68.30
4	0.0	relu	7	20	3	30	2962.41	3076.16	42.78

Tabela 5.11: 5 melhores resultados do modelo TCN. Significado das letras: a - *layers*, b - *dropout_rate*, c - *activation*, d - *timesteps*, e - *batch_size*, f - *kernel_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s).

O melhor modelo candidato apresentou um MAE de 2361.88 mgO2/L e um RMSE de 2386.93 mgO2/L. Esta configuração usou 5 *layers*, sem *dropout*, com a função de ativação *relu*, 7 *timesteps*,

batch_size de 20, e *kernel_size* de 4. O tempo de execução foi de 53.60 segundos, o que, embora não seja o mais rápido, reflete a boa performance do modelo.

A análise restantes modelos revela padrões importantes sobre o impacto de diferentes hiperparâmetros no desempenho dos modelos. Quase todos os modelos utilizaram a função de ativação *relu* e *dropout* rate de 0.0, o que sugere uma consistência nesse aspeto, removendo a influência do *dropout* nas variações de desempenho. Em relação ao tempo de execução, o quarto modelo foi o mais lento, com 68.30 segundos, sugerindo que o aumento no *kernel size* e o uso de menos *timesteps* podem ter contribuído para esse maior tempo de treino. No entanto, ele também apresentou um desempenho razoável em termos de erros (MAE de 3029.69 mgO2/L e RMSE de 3063.19 mgO2/L).

5.2.6 GRU

Relativamente aos modelos GRUs, a Tabela 5.12, sumariza os resultados para os melhores 5 modelos candidatos.

a.	b.	c.	d.	e.	f.	g.	h.	i.	j.
3	32	0.0	relu	7	20	190	1611.25	1649.91	43.49
4	128	0.0	relu	21	20	190	1908.28	1922.78	63.94
3	64	0.0	tanh	21	20	190	1920.39	1938.66	55.72
3	32	0.0	relu	14	20	190	1936.11	1957.63	50.32
5	64	0.5	relu	21	20	190	2062.21	2070.91	68.28

Tabela 5.12: 5 melhores resultados do modelo GRU. Significado das letras: a - *layers*, b - *neurons*, c - *dropout_rate*, d - *activation*, e - *timesteps*, f - *batch_size*, g - *epochs*, h - MAE, i - RMSE, j - tempo(s).

A primeira experiência destaca-se como a melhor, com os menores valores de MAE (1611.25 mgO2/L) e RMSE (1649.91 mgO2/L). Esta configuração utilizou 3 *layers*, 32 neurónios, *dropout rate* de 0.0, e a função de ativação *relu*, com 7 *timesteps*. Este conjunto de parâmetros também proporcionou o menor tempo de execução (43.49 segundos), o que reforça o excelente desempenho tanto em precisão como em eficiência.

Comparando com os outros modelos candidatos,o segundo modelo, que usou 4 camadas e 128 neurónios, resultou num maior tempo de treino (63.94 segundos), além de erros MAE e RMSE mais elevados. Os modelos com funções *relu* tendem a apresentar melhores resultados em termos de precisão. Por outro lado, o quinto modelo, com *dropout rate* de 0.5, apresentou o maior tempo de execução (68.28 segundos) e os piores erros, sugerindo que a introdução de *dropout* pode ter impactado negativamente o desempenho do modelo GRU.

5.2.7 Análise Comparativa

Ao analisar os resultados do histograma ilutrado na Figura 5.3, verifica-se que o modelo CNN obteve o melhor desempenho, apresentando o menor RMSE, com um valor de 1028.06 mgO2/L. Seguindo-se o modelo GRU, que obteve um RMSE de 1649.91 mgO2/L.

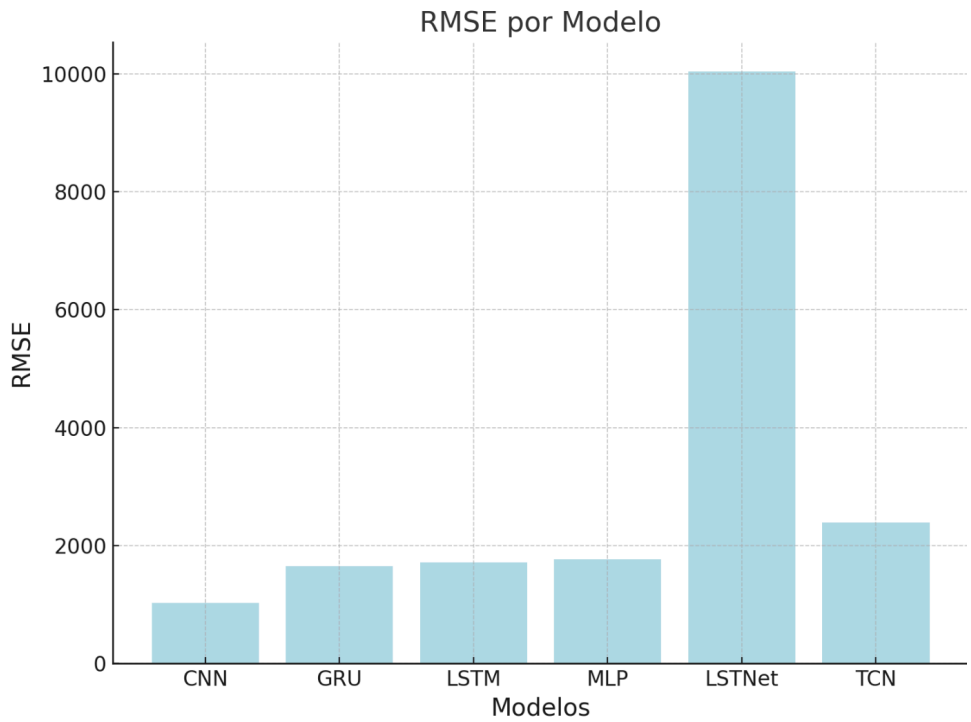


Figura 5.3: Comparação dos melhores resultados obtidos na previsão da Xc.

Os modelos MLP e LSTM apresentaram desempenhos intermediários, com RMSEs de 1767.91 mgO2/L e 1712.58 mgO2/L, respetivamente. Esses valores mostram que, apesar de serem modelos robustos, têm um erro maior em comparação com os modelos CNN e GRU.

No entanto, os modelos LSTNet e TCN tiveram desempenhos significativamente inferiores, com RMSEs muito mais elevados, sendo 10039.63 mgO2/L para o LSTNet e 2386.93 mgO2/L para o TCN. Estes resultados indicam uma margem de erro maior na previsão de Xc, sugerindo que estes modelos são menos adequados para esta tarefa específica.

Por fim, com a análise dos resultados obtidos, a Figura 5.4 ilustra as previsões do modelo para a evolução da percentagem de Xc ao longe de 2 *timesteps* futuros. Novamente como no parâmetro da secção anterior foram sido considerados apenas dois *multi-steps*, a figura demonstrou uma capacidade do modelo em realizar previsões recursivas para quatro passos à frente.

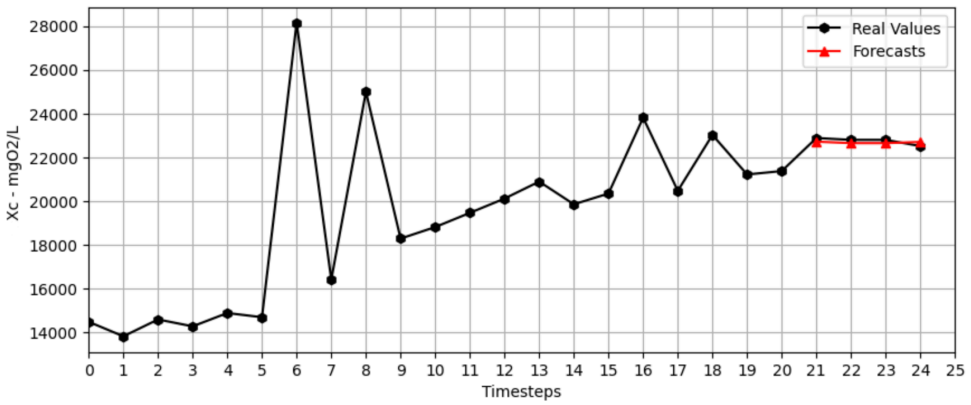


Figura 5.4: Previsões de Xc para 2 *timesteps* futuros.

Capítulo 6

Conclusão e Trabalho Futuro

A crescente crise ambiental está intimamente relacionada com as atividades humanas, sobretudo no que diz respeito à poluição ambiental. A utilização de agentes químicos e o aumento das emissões de poluentes têm impactos diretos e prejudiciais na saúde humana, com destaque para a poluição da água. A contaminação das águas, resultante de práticas industriais, agrícolas e domésticas, coloca em risco tanto o meio ambiente quanto a saúde pública. Neste contexto, as AR surgem como uma das principais fontes de poluição hídrica, exigindo soluções de tratamento eficazes. As ETAR desempenham um papel fundamental na mitigação deste problema. Assim, as primeiras etapas desta dissertação focaram-se na investigação da poluição ambiental e das técnicas atuais de prevenção, bem como na análise das soluções existentes para o tratamento de AR, explorando ainda a viabilidade de combinar conceitos de IA com esses tratamentos para alcançar soluções mais eficientes.

As ETAR, no âmbito do ciclo urbano da água, estão diretamente ligadas ao tratamento das AR, desempenhando um papel crucial na captação, tratamento e distribuição da água para o meio ambiente. Nessas instalações ocorrem diversos processos físicos, físico-químicos e biológicos, essenciais para a remoção ou controlo de substâncias contaminantes presentes nas AR. Um dos processos mais importantes no tratamento das AR é o reator de digestão anaeróbia, que não só contribui para a redução do volume de resíduos sólidos, como também permite o aproveitamento de biogás, uma fonte de energia renovável. Durante o desenvolvimento desta dissertação, verificou-se a importância de otimizar este processo, garantindo que a qualidade da água tratada atenda aos parâmetros estabelecidos e que o aproveitamento energético seja maximizado.

Neste sentido, uma das abordagens centrais desta dissertação foi a integração de tecnologias de ponta, com especial foco na aplicação da IA no processo de monitorização do reator anaeróbio numa ETAR. A análise dos dados recolhidos nas diferentes fases do tratamento da água revelou que a monitorização tradicional ainda não explora o potencial máximo do reator. Com o uso de IA, foi possível demonstrar que o processo de monitorização contínua pode resultar num maior aproveitamento dos recursos e numa

melhoria significativa da eficiência do reator, assegurando uma gestão mais eficaz da qualidade da água tratada.

A introdução de conceitos de IA, como ML e DL, mostrou-se uma solução promissora para alcançar resultados mais económicos, rápidos e adaptáveis. Estes modelos são capazes de generalizar, adaptando-se a diferentes dados, o que é especialmente relevante em sistemas complexos e dinâmicos como o tratamento de AR. O principal objetivo desta dissertação foi, então, a conceção e avaliação de modelos baseados em DL, para prever as concentrações de determinadas substâncias no reator das ETARs, permitindo um controlo mais eficiente por parte das ETARs. Tal controlo é essencial para garantir a qualidade da água e para maximizar o aproveitamento de energia renovável, atingindo assim uma solução mais sustentável e económica no tratamento de AR.

Inicialmente esta dissertação consistiu numa investigação aprofundada da área de aplicação, onde vários conceitos essenciais foram apresentados para guiar o desenvolvimento da dissertação. Em primeiro lugar, destacou-se o funcionamento de uma ETAR, com especial ênfase no reator de digestão anaeróbia, uma peça fundamental no tratamento das AR. O processo de digestão anaeróbia envolve a transformação biológica de matéria orgânica em biogás, sendo eficiente na redução do volume de resíduos e permitindo o aproveitamento de energia renovável. A digestão anaeróbia revelou-se crucial, não só pela sua capacidade de redução de resíduos, mas também pela sua contribuição na produção de CH4, um recurso energético valorizado. Para tal, diversos parâmetros do reator, como temperatura, pH e carga orgânica, são monitorizados e ajustados para maximizar a eficiência do processo. Seguidamente um dos capítulos deste documento descreve como os conceitos provenientes desta investigação inicial possibilitaram a escolha de duas substâncias de particular relevância para prever através de algoritmos de DL: o CH4 e o Xc. Estas substâncias foram selecionadas devido à sua importância na avaliação da eficiência do reator e no controlo da qualidade da água tratada. Paralelamente, foram abordados os conceitos fundamentais de DL, juntamente com a problemática das séries temporais, dado que os conjuntos de dados analisados nesta dissertação consistem precisamente em dados desse tipo. Desta forma, foram apresentados os algoritmos mais adequados para a previsão de séries temporais, todos eles baseados em RNA. Adicionalmente, para complementar esta investigação, foram analisados os cinco artigos mais representativos relacionados diretamente com o tema da presente dissertação. A análise crítica desses estudos revelou lacunas importantes, nomeadamente no que diz respeito à escassez de dados e a otimização de hiperparâmetros cuidada e precisa, aspetos fundamentais em tarefas de DL. Esta análise crítica permitiu, então, definir novos objetivos para complementar e aprimorar o objetivo principal desta dissertação.

Deste modo, foram realizadas tarefas como a preparação do *dataset*, a conceção dos modelos, a execução das experiências e o armazenamento dos resultados. A escolha do metano CH4 e da Xc como *targets* do estudo foi justificada com base na sua importância no processo de digestão anaeróbia e no tratamento biológico nas ETAR. O CH4, componente chave do biogás, é um indicador direto da eficiência da digestão anaeróbia, e a sua monitorização é essencial para maximizar a produção de energia renovável, contribuindo para a sustentabilidade energética das ETAR. A alta percentagem de metano no biogás melhora o seu valor como combustível, reduz as emissões de gases de efeito estufa e facilita a sua

purificação para usos avançados, como o biometano. Por outro lado, a Xc, é um parâmetro crucial para avaliar a eficiência do tratamento biológico, pois reflete a quantidade de biomassa microbiana responsável pela degradação da matéria orgânica nas AR. A monitorização de Xc permite otimizar o processo de tratamento, ajustando variáveis como a idade do lodo e a taxa de recirculação, garantindo assim uma operação mais estável e eficiente.

Em relação à exploração e manipulação dos dados, várias etapas cruciais foram realizadas para garantir a qualidade e fiabilidade dos mesmos. Inicialmente, procedeu-se ao tratamento dos *missing values*, assegurando que os dados em falta fossem devidamente imputados de forma a não comprometer a análise. Seguiu-se o tratamento dos *outliers*, com o objetivo de identificar e corrigir valores atípicos que poderiam distorcer os resultados dos modelos. Posteriormente, foi implementada a fase de *feature engineering*, onde novas variáveis foram criadas a partir das existentes, de modo a enriquecer a informação disponível e melhorar o desempenho dos modelos. Por fim, realizou-se uma análise de correlação e *feature selection*, permitindo selecionar as variáveis mais relevantes para a modelação, com base na sua relação com as variáveis-alvo, %CH4 e Xc, garantindo assim uma abordagem mais eficiente e focada. Estas etapas são fundamentais para o sucesso de qualquer projeto que envolva ML e DL. O tratamento adequado dos dados garante que o modelo trabalhe com informações corretas e consistentes, aumentando a precisão e eficiência. A *feature selection* e o ajuste de variáveis são essenciais para reduzir a complexidade do modelo e melhorar sua generalização, resultando em previsões mais robustas e confiáveis.

Por fim, todas as experiências foram realizadas de acordo com a configuração dos cenários experimentais previamente definidos. Os modelos candidatos, desenvolvidos com base em diversas arquiteturas, nomeadamente CNN, LSTM, TCN, LSTNet, GRU e MLP, foram testados utilizando a abordagem de previsão *multistep recursive forecast*. Esta abordagem permite que as previsões realizadas sejam usadas de forma recursiva como inputs para as previsões subsequentes, garantindo assim a continuidade do processo de aprendizagem.

Os resultados obtidos, que consideraram abordagens multivariadas para cada *target*, foram apresentados através de tabelas e figuras comparativas que ilustram a performance dos modelos em cada caso. Esta análise permitiu identificar o modelo candidato mais adequado para prever cada *target* específico. Consequentemente, para o %CH4, o modelo baseado na arquitetura CNN destacou-se, atingindo um RMSE de 0.0667. No caso do Xc, o melhor desempenho também foi alcançado por um modelo CNN, com um RMSE de 28.06 mgO2/L.

Relativamente às análises comparativas dos modelos, destacou-se que os dois modelos com melhores resultados foram o CNN e o GRU. A introdução de arquiteturas mais complexas, como o TCN e o LSTNet, conduziu a resultados menos favoráveis, o que sugere que, embora sejam técnicas mais avançadas, a sua maior complexidade não se traduziu numa melhoria de desempenho para este caso específico. De forma a avaliar com clareza o desempenho dos melhores modelos candidatos para cada variável-alvo, foram também apresentadas figuras que comparam as previsões recursivas com os valores reais. Através dessas comparações, verifica-se que as previsões geradas pelos modelos encontram-se bastante próximas dos valores reais, evidenciando a sua capacidade de previsão eficiente e consistente.

Relativamente às análises comparativas dos modelos, destacou-se que os dois modelos com melhores resultados foram o CNN e o GRU. A introdução de arquiteturas mais complexas, como o TCN e o LSTNet, conduziu a resultados menos favoráveis, o que sugere que, embora sejam técnicas mais avançadas, a sua maior complexidade não se traduziu numa melhoria de desempenho para este caso específico. Na literatura, tal como visto, modelos baseados em GRU e LSTM são frequentemente apontados como altamente eficazes para a modelação de séries temporais, devido à sua capacidade de lidar com dependências temporais de longo prazo. Contudo, nesta dissertação, os modelos mais complexos, como o TCN e o LSTNet, não superaram os resultados obtidos pelos GRU, o que pode indicar que, para o conjunto de dados e variáveis estudadas, a simplicidade das GRU foi suficiente para captar as dinâmicas temporais. De forma a avaliar com clareza o desempenho dos melhores modelos candidatos para cada variável-alvo, foram também apresentadas figuras que comparam as previsões recursivas com os valores reais. Através dessas comparações, verifica-se que as previsões geradas pelos modelos encontram-se bastante próximas dos valores reais, evidenciando a sua capacidade de previsão eficiente e consistente.

Complementarmente ao trabalho apresentado nesta dissertação, algumas propostas de investigação futura podem ser consideradas, com o intuito de aprofundar o estudo sobre o aproveitamento otimizado do reator digestor anaeróbio numa ETAR, utilizando técnicas de DL. Em termos de modelação, uma possível abordagem seria o uso de arquiteturas mais avançadas, como modelos híbridos que combinem RNA com técnicas de otimização, como AG ou otimização por enxame de partículas, para melhorar a capacidade de generalização e precisão das previsões.

Além disso, a integração de métodos de AutoML poderia automatizar o processo de escolha de hiperparâmetros e otimizar a seleção do modelo, facilitando a a realização de experiências com diferentes configurações. Por outro lado, uma maior disponibilidade de dados poderia ser alcançada através da implementação de sensores mais avançados para a monitorização contínua de parâmetros no reator anaeróbio, possibilitando a criação de *datasets* mais ricos e abrangentes. A expansão dos dados para incluir outras variáveis relacionadas ao processo de digestão anaeróbia também pode aumentar a qualidade das previsões.

Bibliografia

[1] J. Rockström, W. Steffen, K. Noone, Å. Persson, F. S. Chapin III, E. Lambin, T. M. Lenton, M. Scheffer, C. Folke, H. J. Schellnhuber et al. "Planetary boundaries: exploring the safe operating space for humanity". Em: *Ecology and society* 14.2 (2009).

[2] A. A. Bartlett. "Reflections on sustainability, population growth, and the environment". Em: *Population and Environment* 16 (1994), pp. 5–35. doi: 10.1007/BF02208001.

[3] N. K. Arora, T. Fatima, I. Mishra, M. Verma, J. Mishra e V. Mishra. "Environmental sustainability: challenges and viable solutions". Em: *Environmental Sustainability* 1 (2018), pp. 309–340. doi: 10.1007/s42398-018-00038-w.

[4] P. Ekins e D. Zenghelis. "The costs and benefits of environmental sustainability". Em: *Sustainability Science* 16 (2021), pp. 949–965. doi: 10.1007/s11625-021-00910-5.

[5] B. J. Cardinale, J. E. Duffy, A. Gonzalez, D. U. Hooper, C. Perrings, P. Venail e S. Naeem. "Biodiversity loss and its impact on humanity". Em: *Nature* 486.7401 (2012), pp. 59–67. doi: 10.1038/nature11148.

[6] B. Elleuch, F. Bouhamed, M. Elloussaief e M. Jaghbir. "Environmental sustainability and pollution prevention". Em: *Environmental Science and Pollution Research* 25 (2018), pp. 18223–18225. doi: 10.1007/s11356-017-0619-5.

[7] P. J. Landrigan, R. Fuller, N. J. Acosta, O. Adeyi, R. Arnold, A. B. Baldé, R. Bertollini, S. Bose-O'Reilly, J. I. Boufford, P. N. Breysse et al. "The Lancet Commission on pollution and health". Em: *The lancet* 391.10119 (2018), pp. 462–512.

[8] R. Mia, M. D. Selim, A. M. Shamim, M. Chowdhury, S. Sultana, M. Armin e H. Naznin. "Review on various types of pollution problem in textile dyeing & printing industries of Bangladesh and recommendation for mitigation". Em: *Journal of Textile Engineering & Fashion Technology* 5.4 (2019), pp. 220–226. doi: 10.15406/jteft.2019.05.00205.

[9] J. Xu, N. M. Niehoff, A. J. White, E. J. Werder e D. P. Sandler. "Fossil-fuel and combustion-related air pollution and hypertension in the Sister Study". Em: *Environmental Pollution* 315 (2022), p. 120401. doi: 10.1016/j.envpol.2022.120401.

[10] B. Brunekreef e S. T. Holgate. "Air pollution and health". Em: *The Lancet* 360.9341 (2002), pp. 1233–1242.

[11] H. Magsi. “Industrialization, environment and pollution”. Em: *The Diplomatic Insight* 7 (2014), pp. 24–36.

[12] S. M. Simkovich, D. Goodman, C. Roa, M. E. Crocker, G. E. Gianella, B. J. Kirenga e W. Checkley. “The health and social implications of household air pollution and respiratory diseases”. Em: *NPJ Primary Care Respiratory Medicine* 29.1 (2019), p. 12. doi: 10.1038/s41533-019-0126-x.

[13] I. Naureen, A. Saleem, S. Aslam, L. Zakir, A. Mukhtar, R. Nazir e S. Zulqarnain. “Potential Impact of Smog on Human Health”. Em: *Haya Saudi J Life Sci* 7.3 (2022), pp. 78–84. doi: 10.36348/sjls.2022.v07i03.002.

[14] E. Wołejko, A. Jabło□ska-Trypuć, U. Wydro, A. Butarewicz e B. Łozowicka. “Soil biological activity as an indicator of soil pollution with pesticides–a review”. Em: *Applied Soil Ecology* 147 (2020), p. 103356.

[15] P. Li, D. Karunanidhi, T. Subramani e K. Srinivasamoorthy. “Sources and consequences of groundwater contamination”. Em: *Archives of Environmental Contamination and Toxicology* 80 (2021), pp. 1–10. doi: 10.1007/s00244-020-00805-z.

[16] Y. Lu, S. Song, R. Wang, Z. Liu, J. Meng, A. J. Sweetman e T. Wang. “Impacts of soil and water pollution on food safety and health risks in China”. Em: *Environment International* 77 (2015), pp. 5–15. doi: 10.1016/j.envint.2014.12.010.

[17] S Khan, Q Cao, Y. Zheng, Y. Huang e Y. Zhu. “Health risks of heavy metals in contaminated soils and food crops irrigated with wastewater in Beijing, China”. Em: *Environmental pollution* 152.3 (2008), pp. 686–692. doi: 10.1016/j.envpol.2007.06.056.

[18] R. Sordello, F. Flamerie De Lachapelle, B. Livoreil e S. Vanpeene. “Evidence of the environmental impact of noise pollution on biodiversity: a systematic map protocol”. Em: *Environmental Evidence* 8.1 (2019), pp. 1–7. doi: 10.1186/s13750-019-0146-6.

[19] H. Slabbekoorn. “Noise pollution”. Em: *Current Biology* 29.19 (2019), R957–R960.

[20] O. Hahad, S. Kröller-Schön, A. Daiber e T. Münzel. “The cardiovascular effects of noise”. Em: *Deutsches Ärzteblatt International* 116.14 (2019), p. 245. doi: 10.3238/arztebl.2019.0245.

[21] L. J. Bao, K. A. Maruya, S. A. Snyder e E. Y. Zeng. “China’s water pollution by persistent organic pollutants”. Em: *Environmental Pollution* 163 (2012), pp. 100–108. doi: 10.1016/j.envpol.2011.12.022.

[22] D. S. Malik, A. K. Sharma, A. K. Sharma, R. Thakur e M. Sharma. “A review on impact of water pollution on freshwater fish species and their aquatic environment”. Em: *Advances in Environmental Pollution Management: Wastewater Impacts and Treatment Technologies* 1 (2020), pp. 10–28. doi: 10.26832/aesa-2020-aepm.

[23] M. Haseena, M. F. Malik, A. Javed, S. Arshad, N. Asif, S. Zulfiqar e J. Hanif. “Water pollution and human health”. Em: *Environmental Risk Assessment and Remediation* 1.3 (2017).

[24] R. D. Morris. “Drinking water and cancer”. Em: *Environmental Health Perspectives* 103.suppl 8 (1995), pp. 225–231. doi: 10.1023/A:1018444902486.

[25] J. Fawell e M. J. Nieuwenhuijsen. “Contaminants in drinking water: Environmental pollution and health”. Em: *British Medical Bulletin* 68.1 (2003), pp. 199–208. doi: 10.1093/bmb/ldg027.

[26] T. Longcore e C. Rich. “Ecological light pollution”. Em: *Frontiers in Ecology and the Environment* 2.4 (2004), pp. 191–198. doi: 10.1890/1540-9295(2004)002[0191:ELP]2.0.CO;2.

[27] K. Pothukuchi. “City light or star bright: a review of urban light pollution, impacts, and planning implications”. Em: *Journal of Planning Literature* 36.2 (2021), pp. 155–169. doi: 10.1177/0885412220986421.

[28] G. Hatra. “Radioactive pollution: An overview”. Em: *The Holistic Approach to Environment* 8.2 (2018), pp. 48–65.

[29] K. Kamiya, K. Ozasa, S. Akiba, O. Niwa, K. Kodama, N. Takamura e R. Wakeford. “Long-term effects of radiation exposure on health”. Em: *The Lancet* 386.9992 (2015), pp. 469–478.

[30] M. Hatch e E. Cardis. “Somatic health effects of Chernobyl: 30 years on”. Em: *European Journal of Epidemiology* 32 (2017), pp. 1047–1054. doi: 10.1007/s10654-017-0303-6.

[31] S. V. Fesenko, R. M. Alexakhin, S. A. Geras’ kin, N. I. Sanzharova, Y. V. Spirin, S. I. Spiridonov e P. Strand. “Comparative radiation impact on biota and man in the area affected by the accident at the Chernobyl nuclear power plant”. Em: *Journal of Environmental Radioactivity* 80.1 (2005), pp. 1–25. doi: 10.1016/j.jenvrad.2004.08.011.

[32] F. Verones, M. M. Hanafiah, S. Pfister, M. A. Huijbregts, G. J. Pelletier e A. Koehler. “Characterization factors for thermal pollution in freshwater aquatic environments”. Em: *Environmental Science Technology* 44.24 (2010), pp. 9364–9369.

[33] B. Nordell. “Thermal pollution causes global warming”. Em: *Global and Planetary Change* 38.3-4 (2003), pp. 305–312. doi: 10.1016/S0921-8181(03)00113-9.

[34] P. G. Whitehead, R. L. Wilby, R. W. Battarbee, M. Kernan e A. J. Wade. “A review of the potential impacts of climate change on surface water quality”. Em: *Hydrological Sciences Journal* 54.1 (2009), pp. 101–123. doi: 10.1623/hysj.54.1.101.

[35] P. Nami, P. Jahanbakhsh e A. Fathalipour. “the role and heterogeneity of visual pollution on the quality of urban landscape using GIS; case study: Historical Garden in City of Maraqeh”. Em: *Open Journal of Geology* 6.1 (2016), pp. 20–29. doi: 10.4236/ojg.2016.61003.

[36] A. Portella. *Visual pollution: advertising, signage and environmental quality*. Routledge, 2016. doi: 10.4324/9781315547954.

[37] S Banerjee. “A study of visual pollution and its effect on mental health”. Em: *Scholarly Research Journal for Interdisciplinary Studies* 4.30 (2017), pp. 4768–4771.

[38] C. J. Vörösmarty, P. B. McIntyre, M. O. Gessner, D. Dudgeon, A. Prusevich, P. Green, S. Glidden, S. E. Bunn, C. A. Sullivan, C. R. Liermann et al. "Global threats to human water security and river biodiversity". Em: *nature* 467.7315 (2010), pp. 555–561. doi: doi.org/10.1038/nature09440.

[39] *Clean Water - Our World*. https://ourworldindata.org/clean-water. Accessed: 2023-11-11.

[40] M. Petraru, M. Gavrilescu et al. "Pollution prevention, a key to economic and environmental sustainability". Em: *Environmental Engineering and Management Journal* 9.4 (2010), pp. 597–614.

[41] E. Obotey Ezugbe e S. Rathilal. "Membrane technologies in wastewater treatment: a review". Em: *Membranes* 10.5 (2020), p. 89. doi: 10.3390/membranes10050089.

[42] D. P. Loucks. "Sustainable water resources management". Em: *Water international* 25.1 (2000), pp. 3–10. doi: 10.1080/02508060008686793.

[43] Z. W. Kundzewicz. "Water resources for sustainable development". Em: *Hydrological Sciences Journal* 42.4 (1997), pp. 467–480. doi: 10.1080/02626669709492047.

[44] Z. Bekturganov, K. Tussupova, R. Berndtsson, N. Sharapatova, K. Aryngazin e M. Zhanasova. "Water related health problems in Central Asia—A review". Em: *Water* 8.6 (2016), p. 219. doi: 10.3390/w8060219.

[45] P. Kehrein, M. Van Loosdrecht, P. Osseweijer, M. Garfí, J. Dewulf e J. Posada. "A critical review of resource recovery from municipal wastewater treatment plants–market supply potentials, technologies and bottlenecks". Em: *Environmental Science: Water Research & Technology* 6.4 (2020), pp. 877–910. doi: 10.1039/C9EW00905A.

[46] M. C. Tomei, C. M. Braguglia, G. Cento e G. Mininni. "Modeling of anaerobic digestion of sludge". Em: *Critical Reviews in Environmental Science and Technology* 39.12 (2009), pp. 1003–1051. doi: 10.1080/10643380801977818.

[47] L. Appels, J. Baeyens, J. Degrève e R. Dewil. "Principles and potential of the anaerobic digestion of waste-activated sludge". Em: *Progress in energy and combustion science* 34.6 (2008), pp. 755–781. doi: 10.1016/j.pecs.2008.06.002.

[48] B. M. Popkin, K. E. D'Anci e I. H. Rosenberg. "Water, hydration, and health". Em: *Nutrition Reviews* 68.8 (2010), pp. 439–458. doi: 10.1111/j.1753-4887.2010.00304.x.

[49] X. Xiang, Q. Li, S. Khan e O. I. Khalaf. "Urban water resource management for sustainable environment planning using artificial intelligence techniques". Em: *Environmental Impact Assessment Review* 86 (2021), p. 106515. doi: 10.1016/j.eiar.2020.106515.

[50] R. I. Eggen, J. Hollender, A. Joss, M. Schärer e C. Stamm. "Reducing the discharge of micropollutants in the aquatic environment: the benefits of upgrading wastewater treatment plants". Em: (2014).

[51] *O ciclo urbano da agua*. https://www.adp.pt/pt/sobre-nos/o-que-fazemos/o-ciclo-urbano-da-agua/?id=28. Accessed: 2023-11-13.

[52] N. Tangsubkul, P. Beavis, S. J. Moore, S. Lundie e T. D. Waite. "Life cycle assessment of water recycling technology". Em: *Water Resources Management* 19 (2005), pp. 521–537. doi: 10.1007/s11269-005-5602-0.

[53] H. M. D. Monte, M. T. Santos, A. M. Barreiros e A. Albuquerque. *Tratamento de águas residuais: operações e processos de tratamento físico e químico*. 2016.

[54] S. F. M. Pinto. "Estudo do funcionamento da ETAR da TINTEX e contributo para a sua otimização". Tese de mestrado. 2019.

[55] H. Hamdhani, D. E. Eppehimer e M. T. Bogan. "Release of treated effluent into streams: A global review of ecological impacts with a consideration of its potential use for environmental flows". Em: *Freshwater Biology* 65.9 (2020), pp. 1657–1670. doi: 10.1111/fwb.13519.

[56] *Funcionamento de uma ETAR*. Porto Editora.

[57] M. A. Mir, A. Hussain e C. Verma. "Design considerations and operational performance of anaerobic digester: A review". Em: *Cogent Engineering* 3.1 (2016), p. 1181696. doi: 10.1080/23311916.2016.1181696.

[58] Y. Chen, J. J. Cheng e K. S. Creamer. "Inhibition of anaerobic digestion process: a review". Em: *Bioresource Technology* 99.10 (2008), pp. 4044–4064. doi: 10.1016/j.biortech.2007.01.057.

[59] A. C. Wilkie. "Anaerobic digestion: biology and benefits". Em: *Dairy manure management: treatment, handling, and community relations*. 2005, pp. 63–72.

[60] S. Mani, J. Sundaram e K. C. Das. "Process simulation and modeling: Anaerobic digestion of complex organic matter". Em: *Biomass and Bioenergy* 93 (2016), pp. 158–167. doi: 10.1016/j.biombioe.2016.07.018.

[61] R. A. D. Arancon, C. S. K. Lin, K. M. Chan, T. H. Kwan e R. Luque. "Advances on waste valorization: new horizons for a more sustainable society". Em: *Energy Science & Engineering* 1.2 (2013), pp. 53–71. doi: 10.1002/ese3.9.

[62] C. Priadi, D. Wulandari, I. Rahmatika e S. S. Moersidik. "Biogas production in the anaerobic digestion of paper sludge". Em: *APCBEE Procedia* 9 (2014), pp. 65–69. doi: 10.1016/j.apcbee.2014.01.012.

[63] J. B. Holm-Nielsen, T. Al Seadi e P. Oleskowicz-Popiel. "The future of anaerobic digestion and biogas utilization". Em: *Bioresource Technology* 100.22 (2009), pp. 5478–5484. doi: 10.1016/j.biortech.2008.12.046.

[64] C. Nzila, J. Dewulf, H. Spanjers, D. Tuigong, H. Kiriamiti e H. Van Langenhove. "Multi criteria sustainability assessment of biogas production in Kenya". Em: *Applied Energy* 93 (2012), pp. 496–506. doi: 10.1016/j.apenergy.2011.12.020.

[65] W. Xiaohua, D. Chonglan, H. Xiaoyan, W. Weiming, J. Xiaoping e J. Shangyun. "The influence of using biogas digesters on family energy consumption and its economic benefit in rural areas—comparative study between Lianshui and Guichi in China". Em: *Renewable and Sustainable Energy Reviews* 11.5 (2007), pp. 1018–1024. doi: 10.1016/j.rser.2005.08.001.

[66] R. Dalpaz, O. Konrad, C. C. da Silva Cyrne, H. P. Barzotto, C. Hasan e M. Guerini Filho. "Using biogas for energy cogeneration: An analysis of electric and thermal energy generation from agro-industrial waste". Em: *Sustainable Energy Technologies and Assessments* 40 (2020), p. 100774. doi: 10.1016/j.seta.2020.100774.

[67] A. Bušić, S. Kundas, G. Morzak, H. Belskaya, N. Marđetko, M. I. Šantek e B. Šantek. "Recent trends in biodiesel and biogas production". Em: *Food Technology and Biotechnology* 56.2 (2018), p. 152.

[68] T. Chakraborty e N. Akhtar. "Biofertilizers: prospects and challenges for the future". Em: *Biofertilizers: Study and Impact*. 2021, pp. 575–590.

[69] L. Rocha-Meneses, R. Zannerni, A. Inayat, M. Abdallah, A. Shanableh, C. Ghenai, ... e T. Kikas. "Current progress in anaerobic digestion reactors and parameters optimization". Em: *Biomass Conversion and Biorefinery* (2022), pp. 1–24. doi: 10.1007/s13399-021-02224-z.

[70] J. K. Kim, B. R. Oh, Y. N. Chun e S. W. Kim. "Effects of temperature and hydraulic retention time on anaerobic digestion of food waste". Em: *Journal of Bioscience and Bioengineering* 102.4 (2006), pp. 328–332. doi: 10.1263/jbb.102.328.

[71] H. Yu, J. Kim, C. Rhee, J. Shin, S. G. Shin e C. Lee. "Effects of different pH control strategies on microalgae cultivation and nutrient removal from anaerobic digestion effluent". Em: *Microorganisms* 10.2 (2022), p. 357. doi: 10.3390/microorganisms10020357.

[72] A. Babaee e J. Shayegan. "Effect of organic loading rates (OLR) on production of methane from anaerobic digestion of vegetable waste". Em: *Proceedings of the World Renewable Energy Congress*. Linköping, Sweden, 2011, pp. 8–13.

[73] D. Wu, L. Li, X. Zhao, Y. Peng, P. Yang e X. Peng. "Anaerobic digestion: A review on process monitoring". Em: *Renewable and Sustainable Energy Reviews* 103 (2019), pp. 1–12. doi: 10.1016/j.rser.2018.12.039.

[74] S. M. Sunny e K. Joseph. "Review on factors affecting biogas production". Em: *International Journal For Technological Research In Engineering* 5.9 (2018), pp. 3693–3697.

[75] I. Rudowsky. "Intelligent agents". Em: *Communications of the Association for Information Systems* 14.1 (2004), p. 14. doi: 10.17705/1CAIS.01414.

[76] P. Ongsulee. "Artificial intelligence, machine learning and deep learning". Em: *2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE)*. IEEE. 2017, pp. 1–6. doi: 10.1109/ICTKE.2017.8259629.

[77] Y. Lu. “Artificial intelligence: a survey on evolution, models, applications and future trends”. Em: *Journal of Management Analytics* 6.1 (2019), pp. 1–29. doi: 10.1080/23270012.2019.1570365.

[78] J. F. Arinez, Q. Chang, R. X. Gao, C. Xu e J. Zhang. “Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook”. Em: *Journal of Manufacturing Science and Engineering* 142.11 (2020), p. 110804. doi: 10.1115/1.4047855.

[79] D. E. O’Leary. “Artificial Intelligence and Big Data”. Em: *IEEE Intelligent Systems* 28.2 (2013), pp. 96–99. doi: 10.1109/MIS.2013.39.

[80] M. Y. Shaheen. “Applications of Artificial Intelligence (AI) in Healthcare: A Review”. Em: *ScienceOpen Preprints* (2021). doi: 10.14293/S2199-1006.1.SOR-.PPVRY8K.v1.

[81] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen e Y. Wang. “Artificial intelligence in healthcare: past, present and future”. Em: *Stroke and vascular neurology* 2.4 (2017). doi: 10.1136/svn-2017-000101.

[82] R. B. Asha e S. K. KR. “Credit Card Fraud Detection Using Artificial Neural Network”. Em: *Global Transitions Proceedings* 2.1 (2021), pp. 35–41. doi: 10.1016/j.gltp.2021.01.006.

[83] A. H. Huang e H. You. “Artificial Intelligence in Financial Decision-Making”. Em: *Handbook of Financial Decision Making*. 2023, p. 315. doi: 10.4337/9781802204179.00029.

[84] H. Mishra e D. Mishra. “Artificial Intelligence and Machine Learning in Agriculture: Transforming Farming Systems”. Em: *Res. Trends Agric. Sci* 1 (2023), pp. 1–16.

[85] B. George e O. Wooden. “Managing the Strategic Transformation of Higher Education Through Artificial Intelligence”. Em: *Administrative Sciences* 13.9 (2023), p. 196. doi: 10.3390/admsci13090196.

[86] R. S. Lee. *Artificial Intelligence in Daily Life*. Singapore: Springer, 2020, pp. 1–394.

[87] R. Rawassizadeh, T. Sen, S. J. Kim, C. Meurisch, H. Keshavarz, M. Mühlhäuser e M. Pazzani. “Manifestation of Virtual Assistants and Robots into Daily Life: Vision and Challenges”. Em: *CCF Transactions on Pervasive Computing and Interaction* 1 (2019), pp. 163–174. doi: 10.1007/s42486-019-00014-1.

[88] M. Mills. *Artificial Intelligence in Law: The State of Play 2016*. Thomson Reuters Legal Executive Institute, 2016.

[89] G. S. Nelson. “Bias in Artificial Intelligence”. Em: *North Carolina Medical Journal* 80.4 (2019), pp. 220–222.

[90] D. Acemoglu e P. Restrepo. “Automation and New Tasks: How Technology Displaces and Reinstates Labor”. Em: *Journal of Economic Perspectives* 33.2 (2019), pp. 3–30.

[91] S Shukla Shubhendu e J. Vijay. “Applicability of artificial intelligence in different fields of life”. Em: *International Journal of Scientific Engineering and Research* 1.1 (2013), pp. 28–35.

[92] T. M. Mitchell. “Machine Learning and Data Mining”. Em: *Communications of the ACM* 42.11 (1999), pp. 30–36.

[93] S. Ray. “A Quick Review of Machine Learning Algorithms”. Em: *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. IEEE. 2019, pp. 35–39. doi: 10.1109/COMITCon.2019.8862451.

[94] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen e J. S. Rellermeyer. “A Survey on Distributed Machine Learning”. Em: *ACM Computing Surveys (CSUR)* 53.2 (2020), pp. 1–33. doi: 10.1145/3377454.

[95] N. Burkart e M. F. Huber. “A Survey on the Explainability of Supervised Machine Learning”. Em: *Journal of Artificial Intelligence Research* 70 (2021), pp. 245–317. doi: 10.1613/jair.1.12228.

[96] M. Usama, J. Qadir, A. Raza, H. Arif, K.-L. A. Yau, Y. Elkhatib, ... e A. Al-Fuqaha. “Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges”. Em: *IEEE Access* 7 (2019), pp. 65579–65615. doi: 10.1109/ACCESS.2019.2916648.

[97] J. Clifton e E. Laber. “Q-learning: Theory and Applications”. Em: *Annual Review of Statistics and Its Application* 7 (2020), pp. 279–301. doi: 10.1146/annurev-statistics-031219-041220.

[98] W. Qiang e Z. Zhongli. “Reinforcement Learning Model, Algorithms and Its Application”. Em: *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*. IEEE. 2011, pp. 1143–1146. doi: 10.1109/MEC.2011.6025669.

[99] J. Kober, J. A. Bagnell e J. Peters. “Reinforcement Learning in Robotics: A Survey”. Em: *The International Journal of Robotics Research* 32.11 (2013), pp. 1238–1274. doi: 10.1177/0278364913495721.

[100] J. Hopfield. “Artificial neural networks”. Em: *IEEE Circuits and Devices Magazine* 4.5 (1988), pp. 3–10. doi: 10.1109/101.8118.

[101] A. T. Jebb e L. Tay. “Introduction to time series analysis for organizational research: Methods for longitudinal analyses”. Em: *Organizational Research Methods* 20.1 (2017), pp. 61–94. doi: 10.1177/109442811666.

[102] O. B. Sezer, M. U. Gudelek e A. M. Ozbayoglu. “Financial time series forecasting with deep learning: A systematic literature review: 2005–2019”. Em: *Applied soft computing* 90 (2020), p. 106181. doi: 10.1016/j.asoc.2020.106181.

[103] S. L. Zeger, R. Irizarry e R. D. Peng. “On time series analysis of public health and biomedical data”. Em: *Annu. Rev. Public Health* 27 (2006), pp. 57–79. doi: 10.1146/annurev.publhealth.26.021304.144517.

[104] D. Park, S. Kim, Y. An e J.-Y. Jung. “LiReD: A light-weight real-time fault detection system for edge computing using LSTM recurrent neural networks”. Em: *Sensors* 18.7 (2018), p. 2110. doi: doi.org/10.3390/s18072110.

[105] K. Stergiou, E. Christou e G Petrakis. “Modelling and forecasting monthly fisheries catches: comparison of regression, univariate and multivariate time series methods”. Em: *Fisheries Research* 29.1 (1997), pp. 55–95. doi: 10.1016/S0165-7836(96)00482-1.

[106] C. Yin e Q. Dai. “A deep multivariate time series multistep forecasting network”. Em: *Applied Intelligence* 52.8 (2022), pp. 8956–8974. doi: 10.1007/s10489-021-02899-x.

[107] S. B. Taieb e G. Bontempi. “Recursive multi-step time series forecasting by perturbing data”. Em: *2011 IEEE 11th International Conference on Data Mining*. IEEE. 2011, pp. 695–704. doi: 10.1109/ICDM.2011.123.

[108] S. B. Taieb, A. Sorjamaa e G. Bontempi. “Multiple-output modeling for multi-step-ahead time series forecasting”. Em: *Neurocomputing* 73.10-12 (2010), pp. 1950–1957. doi: 10.1016/j.neucom.2009.11.030.

[109] J. C. B. Gamboa. “Deep learning for time-series analysis”. Em: *arXiv preprint arXiv:1701.01887* (2017). doi: 10.48550/arXiv.1701.01887.

[110] M. Hüsken e P. Stagge. “Recurrent neural networks for time series classification”. Em: *Neurocomputing* 50 (2003), pp. 223–235. doi: 10.1016/S0925-2312(01)00706-8.

[111] N. Manchev e M. Spratling. “Target propagation in recurrent neural networks”. Em: *The Journal of Machine Learning Research* 21.1 (2020), pp. 250–282.

[112] B. Lim e S. Zohren. “Time-series forecasting with deep learning: a survey”. Em: *Philosophical Transactions of the Royal Society A* 379.2194 (2021), p. 20200209. doi: 10.1098/rsta.2020.0209.

[113] P. Oliveira, B. Fernandes, C. Analide e P. Novais. “Multi-step ultraviolet index forecasting using long short-term memory networks”. Em: *International Symposium on Distributed Computing and Artificial Intelligence*. Springer. 2020, pp. 187–197. doi: 10.1007/978-3-030-53036-5_20.

[114] K. Ma e H. Leung. “A novel LSTM approach for asynchronous multivariate time series prediction”. Em: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–7. doi: 10.1109/IJCNN.2019.8851792.

[115] A. Sagheer e M. Kotb. “Time series forecasting of petroleum production using deep LSTM recurrent networks”. Em: *Neurocomputing* 323 (2019), pp. 203–213. doi: 10.1016/j.neucom.2018.09.082.

[116] Z. Wang e T. Oates. “Encoding time series as images for visual inspection and classification using tiled convolutional neural networks”. Em: *Workshops at the twenty-ninth AAAI conference on artificial intelligence*. 2015.

[117] *Redes neuronais convolucionais*. https://medium.com/itau-data/redes-neurais-convolucionais-2206a089c715. Accessed: 2023-11-22.

[118] J. Chung, C. Gulcehre, K. Cho e Y. Bengio. “Empirical evaluation of gated recurrent neural networks on sequence modeling”. Em: *arXiv preprint arXiv:1412.3555* (2014). doi: 10.48550/arXiv.1412.3555.

[119] K. Cho. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. Em: *arXiv preprint arXiv:1406.1078* (2014).

[120] *Understanding Gated Recurrent Unit (GRU) in Deep Learning*. https://medium.com/@anishnama20/understandin gated-recurrent-unit-gru-in-deep-learning-2e54923f3e2. Accessed: 2023-11-30.

[121] R. Jozefowicz, W. Zaremba e I. Sutskever. “An empirical exploration of recurrent network architectures”. Em: *International conference on machine learning*. PMLR. 2015, pp. 2342–2350.

[122] I Sutskever. “Sequence to Sequence Learning with Neural Networks”. Em: *arXiv preprint arXiv:1409.3215* (2014).

[123] G. Lai, W.-C. Chang, Y. Yang e H. Liu. “Modeling long-and short-term temporal patterns with deep neural networks”. Em: *The 41st international ACM SIGIR conference on research & development in information retrieval*. 2018, pp. 95–104. doi: 10.1145/3209978.321000.

[124] *The framework of long-term and short-term timeseries network LSTNet*. https://medium.com/@anishnama20/unc gated-recurrent-unit-gru-in-deep-learning-2e54923f3e2. Accessed: 2023-12-5.

[125] C. Lea, R. Vidal, A. Reiter e G. D. Hager. “Temporal convolutional networks: A unified approach to action segmentation”. Em: *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*. Springer. 2016, pp. 47–54. doi: 10.1007/978-3-319-49409-8_7.

[126] K. Lee, J. Ray e C. Safta. “The predictive skill of convolutional neural networks models for disease forecasting”. Em: *Plos one* 16.7 (2021), e0254319. doi: 10.1371/journal.pone.0254319.

[127] S. Bai, J. Z. Kolter e V. Koltun. “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling”. Em: *arXiv preprint arXiv:1803.01271* (2018). doi: 10.48550/arXiv.1803.01271.

[128] I. Goodfellow. *Deep learning*. 2016.

[129] *A Deep Architecture: Multi-Layer Perceptron*. https://medium.com/@nlunge786/a-deep-architecture-multi-layer-perceptron-164bc5ff3842. Accessed: 2023-12-9.

[130] D. E. Rumelhart, G. E. Hinton e R. J. Williams. “Learning representations by back-propagating errors”. Em: *nature* 323.6088 (1986), pp. 533–536. doi: 10.1038/323533a0.

[131] R. Hecht-Nielsen. “Theory of the backpropagation neural network”. Em: *Neural networks for perception*. Elsevier, 1992, pp. 65–93. doi: 10.1016/B978-0-12-741252-8.50010-8.

[132] K. Jeong, A. Abbas, J. Shin, M. Son, Y. M. Kim e K. H. Cho. “Prediction of biogas production in anaerobic co-digestion of organic wastes using deep learning models”. Em: *Water Research* 205 (2021), p. 117697. doi: 10.1016/j.watres.2021.117697.

[133] M. M. Salamattalab, M. H. Zonoozi e M. Molavi-Arabshahi. “Innovative approach for predicting biogas production from large-scale anaerobic digester using long-short term memory (LSTM) coupled with genetic algorithm (GA)”. Em: *Waste Management* 175 (2024), pp. 30–41. doi: 10.1016/j.wasman.2023.12.046.

[134] R. Shen, P. Sun, J. Liu, J. Luo, Z. Yao, R. Zhang, J. Yu e L. Zhao. “Robust prediction for characteristics of digestion products in an industrial-scale biogas project via typical non-time series and time-series machine learning algorithms”. Em: *Chemical Engineering Journal* 498 (2024), p. 155582. doi: 10.1016/j.cej.2024.155582.

[135] X. Cheng, R. Xu, Y. Wu, B. Tang, Y. Luo, W. Huang, F. Wang, S. Fang, Q. Feng, Y. Cheng et al. “Predicting and Evaluating Different Pretreatment Methods on Methane Production from Sludge Anaerobic Digestion via Automated Machine Learning with Ensembled Semisupervised Learning”. Em: *ACS ES&T Engineering* (2023). doi: 10.1021/acsestengg.3c00368.

[136] R. Jia, Y.-C. Song, D.-M. Piao, K. Kim, C.-Y. Lee e J. Park. “Exploration of deep learning models for real-time monitoring of state and performance of anaerobic digestion with online sensors”. Em: *Bioresource Technology* 363 (2022), p. 127908. doi: 10.1016/j.biortech.2022.127908.

[137] R. Wirth e J. Hipp. “CRISP-DM: Towards a standard process model for data mining”. Em: *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. Vol. 1. Manchester. 2000, pp. 29–39.

[138] Y. Shen, J. L. Linville, M. Urgun-Demirtas, M. M. Mintz e S. W. Snyder. “An overview of biogas production and utilization at full-scale wastewater treatment plants (WWTPs) in the United States: challenges and opportunities towards energy-neutral WWTPs”. Em: *Renewable and Sustainable Energy Reviews* 50 (2015), pp. 346–362. doi: 10.1016/j.rser.2015.04.129.

[139] M Calero, V Godoy, C. G. Heras, E Lozano, S Arjandas e M. Martín-Lara. “Current state of biogas and biomethane production and its implications for Spain”. Em: *Sustainable Energy & Fuels* 7.15 (2023), pp. 3584–3602. doi: 10.1039/D3SE00419H.

[140] D. N. Joanes e C. A. Gill. “Comparing measures of sample skewness and kurtosis”. Em: *Journal of the Royal Statistical Society: Series D (The Statistician)* 47.1 (1998), pp. 183–189. doi: 10.1111/1467-9884.00122.

[141] V. Barnett e T. Lewis. “Outliers in statistical data”. Em: (1994).

[142] M. Kuhn e K. Johnson. *Feature engineering and selection: A practical approach for predictive models*. Chapman e Hall/CRC, 2019. doi: 10.1201/9781315108230.

[143] C. Bergmeir e J. M. Benítez. “On the use of cross-validation for time series predictor evaluation”. Em: *Information Sciences* 191 (2012), pp. 192–213. doi: 10.1016/j.ins.2011.12.028.

[144] L. Hou, J. Zhu, J. Kwok, F. Gao, T. Qin e T.-y. Liu. “Normalization helps training of quantized lstm”. Em: *Advances in Neural Information Processing Systems* 32 (2019).