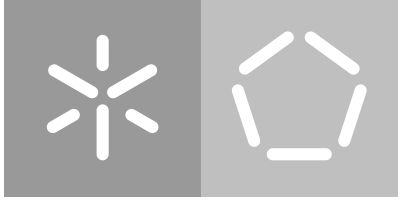
**Universidade do Minho**
Escola de Engenharia

Diogo Filipe Gigante da Silva

# Deteção de Anomalias em Estações de Tratamento de Águas Residuais

Junho, 2022



**Universidade do Minho**
Escola de Engenharia

Diogo Filipe Gigante da Silva

# Deteção de Anomalias em Estações de Tratamento de Águas Residuais

Dissertação de Mestrado
Mestrado em Engenharia Informática

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

Junho, 2022

# Agradecimentos

Quero agradecer primeiramente ao meu orientador, professor Paulo Jorge Freitas de Oliveira Novais, que me proporcionou a realização desta dissertação bem como, gostaria ainda de realçar todo o seu apoio e comprometimento ao longo dos dois anos de mestrado que me permitiram atingir ótimos resultados. Ainda de salientar, a sua constante disponibilidade para a resolução de qualquer eventualidade. Não podia também de deixar de referir o Pedro Oliveira, que foi incansável no decorrer deste projeto onde me apoiou e respondeu a todas as questões que surgiram, contribuindo imenso para o resultado final desta dissertação. Para os engenheiros Francisco Aguiar e Frederico Barros Lopes das Águas do Norte, um especial agradecimento pelo fornecimentos dos conjuntos de dados que foram utilizados no decorrer desta dissertação, bem como pelo todo o apoio prestado.

Menciono agora a minha família, especialmente os meus pais e namorada, que caminharam a meu lado ao longo destes dois anos de mestrado, fornecendo-me sempre suporte, força e motivação para concluir esta fase com sucesso. A eles, um enorme obrigado.

Finalmente, um grande agradecimento aos meus amigos que me apoiaram e ajudaram a levar esta jornada a bom porto.

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 DSAIPA/AI/0099/2019.

## 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.

Assinado por: **Diogo Filipe Gigante da Silva**
Num. de Identificação: 15366085
Data: 2022.06.30 21:04:20+01'00'

(Diogo Filipe Gigante da Silva)

# Resumo

A iminente escassez de recursos naturais e o constante aumento populacional tem assolado o presente século. Tal crescente habitacional contribui para uma concentração nos grandes centros urbanos e, consequentemente, um maior nível de poluição quer em contextos habitacionais como industriais. Nesta vertente, as Estações de Tratamento de Águas Residuais, desempenham um papel crucial no controlo do nível de qualidade da água que é reutilizada ou descarregada para o exterior. Estas instalações recebem ininterruptamente cargas de afluentes extremamente poluentes que são provenientes da rede pública de esgotos e que carecem de um tratamento faseado para a purificação das mesmas. Porém, para garantir a qualidade da água que é reaproveitada ou devolvida ao meio ambiente, é necessária monitorização contínua destas estações de forma a permitir o processo de tomada de decisão.

Posto isto, esta dissertação visa implementar modelos de *Machine Learning* com o intuito de detetar possíveis anomalias nas substâncias presentes na efluente destas infraestruturas. Assim sendo, são aplicados modelos como *Isolation Forest* (IF), *One Class Support Vector Machine* (OCSVM) e *Long Short-Term Memory Autoencoder* (LSTM-AE) para identificar os registos do Azoto Total, Nitratos e *pH* que possam ser anómalos. No caso em específico das LSTM-AE, são considerados três *thresholds* para classificar os registos, dos quais, dois utilizam valores estáticos e um consiste em valores dinâmicos.

De entre os melhores modelos candidatos, no global, os modelos de IF e OCSVM alcançaram resultados superiores aos modelos baseados em LSTM-AE. No que diz respeito aos *thresholds*, as abordagens com valores estáticas de forma geral, atingiram resultados ligeiramente superiores. Em suma, os vários cenários aplicados permitiram concluir que os modelos concebidos conseguiram detetar as várias anomalias presentes nas substâncias referidas.

**Palavras-chave:** Controlo Analítico, Deteção de Anomalias, Estações de Tratamento de Águas Residuais, *Machine Learning*

# Abstract

The imminent scarcity of natural resources and the constant population increase have plagued the present century. Such populational growth contributes to a concentration in large urban centres and, consequently, a higher pollution level in housing and industrial contexts. In this regard, the Wastewater Treatment Plants play a crucial role in controlling the water quality that is reused or discharged abroad. These installations receive uninterrupted loads of extremely polluting affluents from the public sewage system that need a phased treatment to purify them. However, to guarantee the quality of the water reused or discharged into the environment, continuous monitoring of these facilities is necessary to allow the decision-making process.

That said, this dissertation aims to implement Machine Learning models to detect possible anomalies in the substances present in the effluent of these infrastructures. Therefore, models such as Isolation Forest (IF), One-Class Support Vector Machine (OCSVM) and Long Short-Term Memory Autoencoder (LSTM-AE) are applied to identify the records of the Total Nitrogen, Nitrates and pH that may be anomalous. In the specific case of LSTM-AE, three thresholds are considered to classify the records, of which two use static values, and one consists of dynamic values.

Among the best candidate models, overall, the IF and OCSVM models achieved superior results to the models based on LSTM-AE. Regarding thresholds, the approaches with static values generally achieved slightly better results. The various scenarios applied allowed us to conclude that the designed models could detect various anomalies in the substances mentioned.

**Keywords:** Analytical Control, Anomaly Detection, Machine Learning, Wastewater Treatment Plants

# Índice

# Lista de Figuras

# Lista de Tabelas

# Siglas

| | |
|---|---|
| AGV | Ácidos Gordos Voláteis |
| API | Application Programming Interface |
| AR | Águas Residuais |
| ARD | Águas Residuais Domésticas |
| ARI | Águas Residuais Industriais |
| ARIMA | Autoregressive Integrated Moving Average |
| ART | Águas Residuais Tratadas |
| ARU | Águas Residuais Urbanas |
| AUC-ROC | Area Under The Curve - Receiver Operating Characteristics |
| | |
| C | Carbono |
| CBO | Carência Bioquímica de Oxigénio |
| COT | Carbono Orgânico Total |
| CPV | Cumulative Percent Variance |
| CQO | Carência Química de Oxigénio |
| CRISP-DM | Cross Industry Standard Process for Data Mining |
| CRUD | Create, Read, Update and Delete |
| CUA | Ciclo Urbano da Água |
| | |
| DBN | Deep Belief Networks |
| DI | Departamento de Informática |
| DL | Deep Learning |
| DS | Data Science |
| DT | Decision Trees |
| | |
| EDA | Exploratory Data Analysis |
| ETA | Estações de Tratamento de Água |
| ETAR | Estação de Tratamento de Águas Residuais |
| ETARs | Estações de Tratamento de Águas Residuais |

| | |
|---|---|
| F | Fósforo |
| FN | False Negative |
| FP | False Positive |
| FPR | False Positive Rate |
| H | Hidrogénio |
| IA | Inteligência Artificial |
| IDE | Integrated Development Environment |
| iF | Isolation Forest |
| iT | Isolation Tree |
| IVL | Índice Volumétrico de Lamas |
| LSTM | Long-Short Term Memory Networks |
| LSTM-AE | Long-Short Term Memory Auto Encoders |
| MAE | Mean Absolute Error |
| MEI | Mestrado em Engenharia Informática |
| ML | Machine Learning |
| N | Nitrogénio |
| O | Oxigénio |
| OCSVM | One-Class support vector machines |
| OD | Oxigénio Dissolvido |
| PCA | Principal Components Analysis |
| pH | Potencial Hidrogeniônico |
| REST | Representational State Transfer |
| RF | Random Forests |
| RNN | Recurrent Neural Networks |
| S | Enxofre |
| SFT | Sólidos Fixos Totais |

SS Sólidos Suspensos
SSF Sólidos Suspensos Fixos
SST Sólidos Suspensos Totais
SSV Sólidos Suspensos Voláteis
ST Sólidos Totais
SV Sólidos Voláteis
SVM Support Vector Machine
SVT Sólidos Voláteis Totais

TN True Negative
TP True Positive
TPR True Positive Rate

UM Universidade do Minho

VLE Valores limite de emissão

Capítulo 1

# Introdução

O corrente tema proposto para a dissertação, Deteção de Anomalias em Estações de Tratamento de Águas Residuais, será desenvolvido no âmbito da Unidade Curricular de Dissertação em Engenharia Informática do Mestrado em Engenharia Informática (MEI) realizado no Departamento de Informática (DI) na Universidade do Minho (UM). Por conseguinte, o primário objetivo deste projeto, passa por a aplicação de áreas da ciência de computação, nomeadamente Machine Learning (ML) e Data Science (DS), com o intuito de melhorar e fornecer informação relevante à gestão das Estações de Tratamento de Águas Residuais (ETARs).

Posto isto, no presente capítulo, a secção 1.1 focar-se-á no enquadramento bem como na motivação que sustem este tema. No ponto 1.2, serão delineados os objetivos essenciais à elaboração deste tema. De seguida, na secção 1.3, a respetiva metodologia de trabalho adjacente ao desenvolvimento desta dissertação será descrita. Por fim, a secção 1.4 caracteriza a estrutura do documento bem como os tópicos inerentes aos vários capítulos.

## 1.1 Enquadramento e Motivação

Nesta dissertação estão incorporadas várias áreas, tornando-se assim, de extrema importância, enquadrar as mesmas no tema a ser abordado.

A sustentabilidade é uma área muito marcante das últimas décadas, surgindo em 1987 como um conceito político no documento Bruntland Report [1], esta conceção remonta para uma preocupação com os recursos naturais e o bem-estar das futuras gerações [2]. Com a população mundial de 7.8 mil milhões em 2020 e uma projeção de 9.8 mil milhões para 2050, a preservação de recursos naturais torna-se uma tarefa exaustiva e complexa [3].

O aquecimento global e as mudanças climáticas são duas das principais causas do drástico decréscimo de recursos hídricos em todo o mundo [4]. As Águas Residuais (AR), após tratadas, são um recurso

hídrico extremamente importante, especialmente em países onde a disponibilidade deste bem-essencial é escassa [5].

No que a Portugal diz respeito, na Figura 1.1 é possível verificar que já em Outubro de 2018 existiam valores baixos de armazenamento subterrâneo, alguns inferiores ao 20 percentil na região sul, sendo que este cenário em Abril de 2019 se estendeu por todo o país o que pode provocar escassez de água nas regiões afetadas [6].

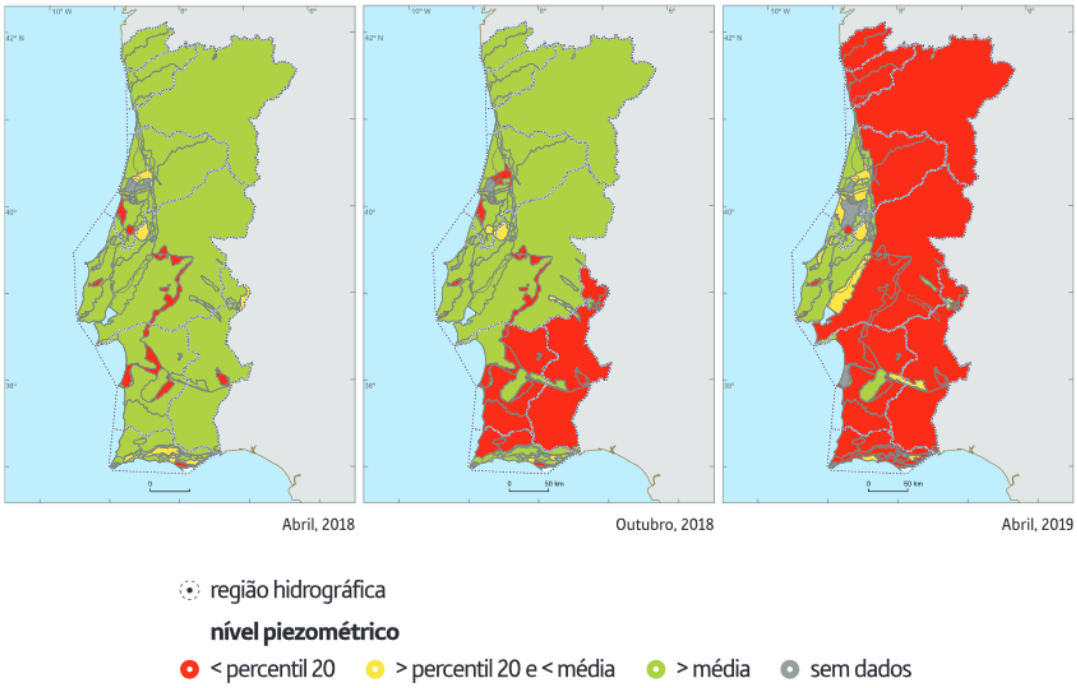



Figura 1.1: Evolução do armazenamento subterrâneo em Portugal (extraído do Relatório do Estado do Ambiente Portugal 2019[1])

Para colmatar estes problemas, é necessária uma ótima gestão da água e, neste sentido, as ETARs fornecem um serviço de extrema importância para a comunidade. Estas infraestruturas tem um papel crucial para remover todos os micróbios e contaminantes presentes nos esgotos [7] permitindo que esta água seja reutilizada para diversos fins, tais como, urbanos, industriais e agrícolas [8]. Para garantir a qualidade da água, existem inúmeros sensores ao longo das várias fases de tratamento nas ETARs, porém, não existe um sistema inteligente que consiga extrair informações relevantes dos dados nem detetar eficientemente anomalias durante todo o processo que possam acrescentar valor a estes sensores.

Este sistema inteligente é o grande foco da dissertação e é nele que é introduzida outra grande área, o ML. Esta área é um ramo da Inteligência Artificial (IA) na qual os sistemas computacionais aprendem diretamente de exemplos, dados e experiências [9].

Posto isto, a motivação para a realização desta dissertação advém do elevado interesse em ML que foi cultivado em unidades curriculares do MEI. Neste período formativo, houve um grande foco em IA, mais especificamente, DS e ML. Após experienciar um contato mais próximo com esta área, foi notório o enorme

[1] https://sniambgeoviewer.apambiente.pt/GeoDocs/geoportaldocs/rea/REA2019/REA2019.pdf

espectro aplicacional que pode ser reinventado e revolucionado. Posto isto, emergiu a oportunidade de realizar esta dissertação, permitindo aumentar o conhecimento nesta área.

Sendo o campo de atuação desta dissertação as ETARs, que representam um meio de tratamento para a poluição aquática e contribuem para a sustentabilidade ambiental, o interesse foi instantâneo.

### 1.1.1 Poluição Ambiental

Ao mencionar sustentabilidade, é imprescindível referir a poluição ambiental. A drástica crise ambiental em que vivemos é causada por diversas alterações ambientais e ecológicas que resultaram de um processo económico e tecnológico comandado pelos seres humanos ao longo do corrente século [10]. É factual a enorme evolução socioeconómica, cientifica e tecnológica levada a cabo neste período, porém, as consequências que causaram no nosso planeta são irrefutáveis [11].

O problema que estamos a enfrentar é de extrema complexidade sendo por isso imperativo mencionar as principais causas do mesmo. Embora existam diversas visões, é lógico considerar que não existe uma única causa para o estado atual, mas sim um conjunto de fatores, dos quais se destacam [12]:

- **Aumento Populacional** → o crescimento exponencial da população mundial está implicitamente correlacionado com uma maior necessidade de bens essenciais à existência humana. Este grande aumento verificado nos últimos tempos resulta numa maior exploração dos recursos naturais, tais como, a água e as florestas [13].

- **Tecnologia** → a evolução da tecnologia tem como principal consequência os danos no ambiente. O desenvolvimento de aparelhos tecnológicos cada vez mais sofisticados causa grande impacto no nosso planeta, tais como, acidificação dos solos, poluição do ar e poluição da água o que, entre várias consequências, provoca a extinção de espécies bem como o degelo das calotas polares e consequente aumento do nível do mar [14].

- **Desflorestação** → a conversão de terras florestais em terrenos agrícola e pastagens, os incêndios florestais e, naturalmente, a extração de madeira são os principais fatores que causam a desflorestação e consequentemente problemas relacionados com uma aceleração na erosão do solo, aumento de cargas sedimentares nos rios e aumento do efeito de estufa [15].

- **Desenvolvimento Industrial** → o grande aumento da industrialização resulta numa maior exploração de recursos naturais bem como na libertação de enormes quantidades de poluentes causando a poluição do ar e consequente acidificação das águas e solos [16].

- **Urbanização** → a crescente urbanização significa obrigatoriamente uma maior concentração populacional num menor espaço, o que resulta na necessidade de, por exemplo, construção de estradas e edifícios, circulação de mais veículos e aumento de lixo urbano e do consumo hídrico que provoca, entre várias consequências, a escassez hídrica [17].

Posto isto, existem obviamente diversas causas para a situação atual. Estas e outras causas podem ser classificadas de uma forma geral através de dois essenciais grupos de poluição, nomeadamente [18]:

- **Poluição natural** → este tipo de poluição é menos comum, porém, ocasionalmente acontecem fenómenos naturais como tremores de terra, inundações, ciclones, etc.
- **Poluição produzida pelo homem** → ao contrário da poluição natural, a poluição produzida pelo homem é, infelizmente, extremamente comum. As atividades humanas, tais como as previamente mencionadas, são o maior contributo para a poluição do planeta.

Assim sendo, estas duas formas de poluição causam diversas consequências no nosso planeta sendo, grande parte delas, irreversíveis. Estas consequências podem ser agrupadas entre os seguintes cinco principais tipos de poluição [19]:

- **Poluição do Ar** → este tipo de poluição é causado por gases e partículas suspensas no ar que causam grandes concentrações de toxicidade e consequentemente grandes impactos na saúde humana. Como exemplo disso, a World Health Organization, estima que todos os anos, cerca de 2.4 milhões de pessoas morrem por motivos relacionados diretamente com a poluição do ar [20].
- **Poluição Aquática** → a contaminação aquática surge através de, essencialmente, descargas de esgotos e químicos nocivos. Este tipo de poluição tem um enorme impacto nos seres vivos aquáticos, nos solos e, em casos extremos, pode até aumentar o risco de cancro nos seres humanos [21].
- **Poluição do Solo** → na poluição do solo, a contaminação advém de resíduos de produtos químicos como hidrocarbonetos e metais pesados. A presença deste metais pesados nos solos, entre diversas consequências, é extremamente tóxica para os seres humanos. Já os hidrocarbonetos, podem causar problemas cardiovasculares e cancerísnos [22].
- **Poluição Biológica** → a poluição biológica ocorre através da contaminação com germes de micróbios. Um exemplo interessante deste tipo de poluição, é a introdução de uma espécie denominada de caranguejo chinês que se estende por grande parte do Noroeste Europeu, como por exemplo, no estuário do Tejo, em Portugal, e começou a causar danos ás paredes de defesa contra inundação através da sua atividade escavadora [23].
- **Poluição Radioativa** → esta poluição advém de resíduos nucleares, como por exemplo, mineração de urânio. Esta variante de poluição é extremamente nociva para o meio ambiente, podendo causar nos seres humanos diversos problemas, tais como, deformidades crónicas, problemas respiratórios e de circulação sanguínea, cancro e leucemia [24].

#### 1.1.1.1 Inteligência Artificial na Poluição Ambiental

A IA é uma área muito abrangente que se foca no desenvolvimento de sistemas computacionais capazes de realizar tarefas, que em tempos seriam realizadas apenas por humanos, de forma mais eficaz. São inúmeras as áreas onde a IA está presente e este número está em constante ascensão, tais como, Agricultura e Automóvel [25].

No que a poluição ambiental diz respeito, já existem alguns sistemas de IA que visam proteger o meio ambiente, entre outros:

- **Sistema capaz de prever os níveis de poluição do ar** → na Universidade de Loughborough uma equipa criou um sistema que prevê os níveis de poluição do ar com horas de antecedência. Este sistema tem como principal objetivo combater as várias milhões de mortes anuais devido à poluição do ar, tornando o ar das comunidades mais limpo e, consequentemente, melhorando a vida da população mundial no futuro [26].

- **IA para sustentabilidade agrícola** → a monitorização de plantações e solos ajuda a maximizar a produção agrícola e a reduzir o impacto no meio ambiente [27]. É possível com dispositivos e sensores de monitorização, quando conectados ás plantações, consultar parâmetros como hidratação, nutrição e possíveis doenças em tempo real [28]. Estes dados são então utilizados para determinar vários padrões, como por exemplo, de irrigação, recomendando os melhores ciclos de irrigação para prevenir desperdícios hídricos e, através de informações nutricionais, reduzir o uso de pesticidas prejudiciais [28].

- **IA para preservação da água** → os níveis de poluição nos oceanos são mais elevados que nunca e a preservação destes recursos tornou-se uma necessidade do imediato. Sistemas capazes de detetar a fonte dos poluentes encontrados na água através de modelos de ML são aplicados para descobrir e rastrear a fonte industrial que expeliu os poluentes [29]. Outra aplicação da IA que está a dar largos passos para ajudar a reduzir este problema é a deteção de descargas ilegais de efluentes industriais e despejos de sólidos pela população [30].

### 1.1.2 Ciclo Urbano da Água

De todos os recursos naturais do planeta, é indiscutível que a água é o recurso imprescindível para o seu bom funcionamento porque sem a mesma, não existiria vida na Terra. Este recurso é indispensável sendo que a vida humana e maior parte da natureza depende da disponibilidade de água doce para subsistir porém, apenas 2.5% de toda a água do planeta terra é doce e, grande parte desta, está retida em calotas de gelo [31].

A qualidade da água está diretamente relacionada com o vida humana, ou seja, as atividades humanas, aumentos da industrialização e urbanização diminuem a pureza da água.

Com a mentalização do aumento da poluição aquática e escassez dos recurso hídricos, existe globalmente, cada vez mais, uma preocupação com a reaproveitação da água sendo o Ciclo Urbano da Água (CUA) o exemplo perfeito desta reutilização, ilustrado na Figura 1.2.

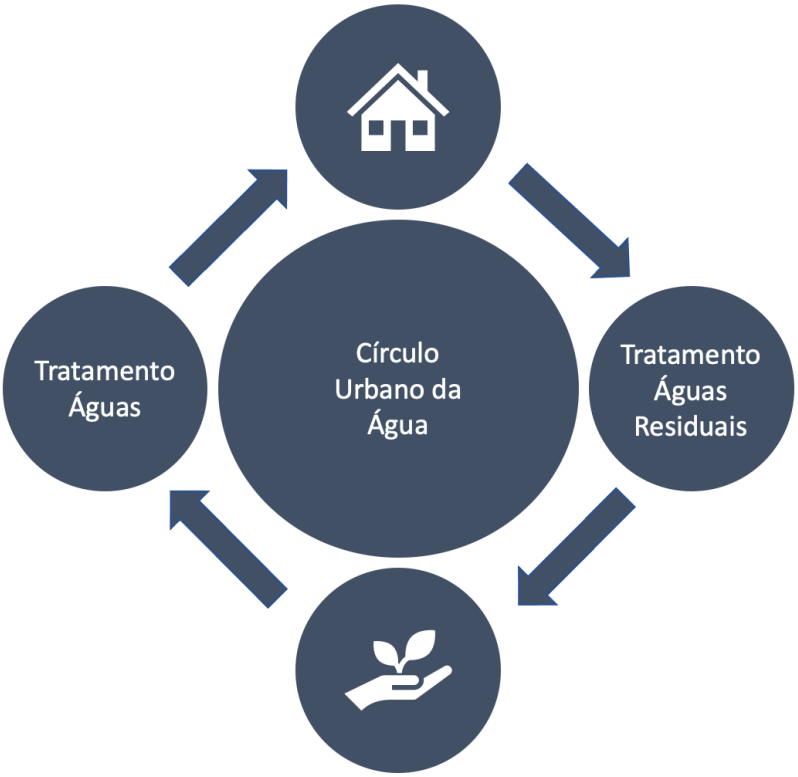



Figura 1.2: Ciclo Urbano da Água

O CUA corresponde ao percurso realizado pela água desde que é absorvida da natureza até ser utilizada para diversos fins e posteriormente devolvida à sua origem. Este ciclo consiste em oito essenciais etapas, nomeadamente [32]:

- **Captação da água** → como o nome indica, inicialmente, é captada água da natureza presente no subsolo, rios ou à superfície.
- **Estações de Tratamento de Água (ETA)** → na segunda etapa, feita a captação da água no passo anterior, a mesma é encaminhada para a ETA onde são realizados tratamentos que a tornam segura para a utilização humana.
- **Armazenamento de Água** → realizado o tratamento à água, esta é transportada para vários reservatórios onde é armazenada até seguir para a rede de distribuição.
- **Rede de distribuição e consumo** → na quarta etapa, é realizada a distribuição da água armazenada nos reservatórios referidos no passo anterior. Esta distribuição é efectuada recorrendo a uma rede complexa de tubagens garantindo a devida pressão e qualidade para a sua utilização.

- **Rede de recolha e transporte de AR** → após o transporte para consumo e consequente utilização, as águas agora denominadas de AR são recolhidas e encaminhadas através de redes de saneamento para as ETARs.
- **Estação de Tratamento de Águas Residuais (ETAR)** → na sexta etapa, as AR passam por várias etapas de tratamento nas ETAR até poderem ser devolvidas com segurança ao ambiente.
- **Reutilização de água** → feito o tratamento na ETAR, a água pode ser reutilizada para diversos fins como, por exemplo, industriais e agrícolas.
- **Devolução da água** → na ultima etapa, para concluir o ciclo, a água é devolvida à natureza de forma a permitir a reposição de água nos meios hídricos sem comprometer a saúde pública.

### 1.1.3 Estações de Tratamento de Águas Residuais

Como referido previamente na secção 1.1, as ETARs são cruciais na remoção de micróbios e contaminantes presentes nos esgotos. Em Portugal, é possível reaproveitar as AR das Estações de Tratamento Municipais para diversos fins, sejam eles agro-culturais ou industriais. Porém, estas AR aplicadas para irrigação do solo ou indústria, acabarão por entrar em contacto com humanos ou animais [33][34].

Posto isto, antes das águas serem distribuídas, passam por quatro essenciais tratamentos nas ETARs, nomeadamente, tratamento preliminar, primário, secundário e terciário que possibilitam uma reutilização segura e estão distribuídos pelas várias fases de tratamento ilustradas na Figura 1.3 [6][35].


Figura 1.3: Estação de Tratamento de AR

#### 1.1.3.1 Tratamento preliminar

A primeira fase de tratamento consiste num conjunto de operações físicas para remover da água residual materiais grosseiros, areias e gorduras. Com estas remoções, protegem-se os tratamentos seguintes, bem como são evitadas obstruções dos circuitos hidráulicos e contaminações das águas e lamas. Os sistemas que estão presentes nesta etapa são os seguintes [35]:

- **Gradagem** → sistema de grelhas instado em canais por onde circula a água residual. Estas grelhas retém os objetos sólidos com maiores dimensões que, após armazenados em contentores são encaminhados para destinos adequados.
- **Tamisação** → tal como no sistema anterior, o seu objetivo é reter objetos sólidos porém, como possui uma malha mais fina, consegue remover sólidos de menores dimensões pelo que deve ser utilizado como forma de complementar a gradagem.
- **Desarenação** → neste sistema, são retiradas as areias do afluente. Este processo pode ser realizado de várias formas como, por exemplo, num tanque em que as areias assentem por gravidade no fundo do mesmo, sendo removidas e encaminhadas para um classificador de areias.
- **Remoção de óleos e gorduras** → o último sistema presente nesta fase tem, como o nome indica, o objetivo de retirar gorduras e óleos existentes. Normalmente, recorre-se à injeção de um fluxo de ar ascendente no seio do afluente o que leva à acumulação de gorduras na superfície que são posteriormente removidas.

#### 1.1.3.2 Tratamento primário

Concluída a primeira fase surge o tratamento primário. Esta etapa pode ser constituída por processos físicos e químicos e tem como objetivo remover os sólidos facilmente sedimentáveis. É expectável que, com este tratamento, o total das partículas sólidas em suspensão seja reduzido, no mínimo, em 50%. O tratamento nesta fase é realizado através de decantação ou flotação [35]:

- **Decantação** → tem como principal objetivo retirar os sólidos em suspensão. Esta operação é realizada num decantador onde a água permanece o tempo suficiente para permitir que as partículas suspensas sedimentem no fundo, sendo também removidas as escumas que se acumulam à superfície dos decantadores.
- **Flotação** → esta operação surge como complemento à anterior onde são removidos sólidos de dimensões tão baixas que não é viável a sua separação por ação da gravidade. Para realizar esta extração, é injetado um fluxo de ar ascendente no tanque, sendo estes sólidos arrastados para a superfície juntamente com as bolhas de ar onde são, por fim, recolhidos e encaminhados para tratamento.

#### 1.1.3.3 Tratamento secundário

A terceira fase, denominada de tratamento secundário, é constituída por processos biológicos que visam retirar a matéria orgânica biodegradável existente no afluente que não foi retirada no tratamento primário. Neste tratamento a água residual é colocada em contacto com microrganismos que vão metabolizar essa matéria orgânica. Esta é a fase de tratamento para a qual existe a maior variedade de sistemas, podendo ser de biomassa fixa, hídricos, suspensa e sistemas combinados [35].

#### 1.1.3.4 Tratamento terciário

Após o tratamento secundário, a última fase, denominada de tratamento terciário ou de afinação, complementa as etapas anteriores de tratamento. Nesta etapa a exigência de qualidade é uma prioridade visto que se trata do passo final. Posto isto, o seu objetivo passa pela remoção de determinados poluentes que se mantém na água após terem passado pelos tratamentos anteriores, como partículas dificilmente decantáveis, microrganismos patogénicos, nutrientes entre outros compostos através de filtração seguida de desinfeção com recurso a radiação ultravioleta [35].

#### 1.1.3.5 Tratamento extra

Para além dos quatro tratamentos realizados na fase líquida, são necessários outros processos para a matéria sólida que foi retirada das AR. No que diz respeito ás lamas geradas pela ETAR, é realizado um espessamento onde o volume das mesmas é reduzido para diminuir as dimensões. Feito isto, é realizada a fase de estabilização de forma a evitar o potencial de putrefacção, remover microrganismos patogénicos e eliminar odores. Por fim, ocorre a desidratação com o intuito de retirar o máximo de água presente nas lamas refletindo-se num menor custo de transporte das lamas para o destino final [35].

Uma das características menos positivas das ETARs prende-se com a produção de odores resultantes da degradação da matéria orgânica presente no afluente. De forma a minimizar esta emissão são utilizadas tecnologias como biofiltros, sistemas de absorção e sistemas de lavagem química. [35]

## 1.2 Objetivos

O tema de estudo para esta dissertação é a deteção de anomalias em ETARs. Sendo estas infraestruturas suscetíveis a falhas para as quais, as manutenções são muito dispendiosas, é crucial uma boa análise para a deteção e possível prevenção das mesmas. Desta forma, esta dissertação assentará essencialmente nos seguintes objetivos:

- Levantamento da literatura existente no âmbito da ciência dos dados no campo das ETARs;
- Levantamento das possibilidade de deteção de anomalias em ETARs;
- Compreensão dos dados através de fontes heterogéneas;

- Tratamento dos dados utilizando técnicas que lidam com problemas de séries temporais;
- Desenvolvimento de modelos de ML para deteção de anomalias;
- Analisar e validar todos os modelos desenvolvidos;
- Discussão dos resultados obtidos.

## 1.3 Metodologia de Trabalho

Nesta dissertação, de uma forma global, é necessária toda uma pesquisa e compreensão do tema em estudo com uma subsequente aprendizagem para desenvolver uma solução concisa.

Assim sendo, as etapas para a realização desta dissertação são:

- Desenvolver uma investigação sobre o problema e descrição das suas características.
- Realizar uma análise das soluções que possam existir e dos possíveis pontos que são importantes para integrar no projeto, conduzindo a uma aquisição de conhecimentos sobre os trabalhos realizados no mesmo domínio.
- Desenvolver as várias etapas inerentes à resolução do problema, tendo em conta os resultados das etapas anteriores, elaborando todos os requisitos necessários para a solução e um esquema conceptual da arquitetura a ser utilizada com recurso à metodologia Cross Industry Standard Process for Data Mining (CRISP-DM) como modelo de trabalho.

Como referido, nesta dissertação, a metodologia de trabalho empregue é a metodologia CRISP-DM, com o intuito de formular e refletir sobre soluções para o problema em questão [36].

A metodologia CRISP-DM, é um processo para mineração de dados que foi desenvolvido em 1996. Esta metodologia é uma das mais utilizadas e define o ciclo de vida do projeto, dividindo-o em seis fases, ilustrado na Figura 1.4 [37]:

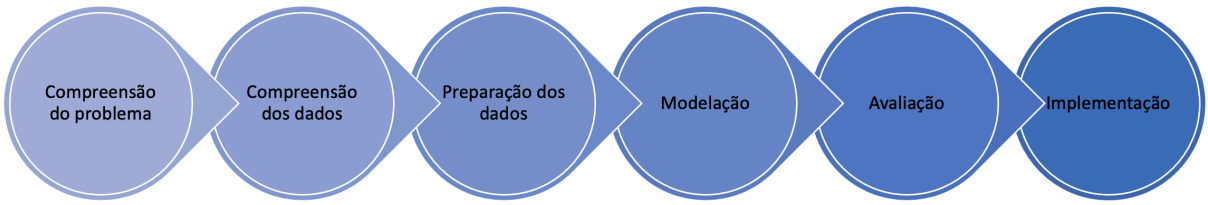



Figura 1.4: CRISP-DM

1. **Compreensão do problema** → Nesta primeira fase, é essencial, como o nome indica, compreender o problema em questão bem como analisar e fundamentar objetivos que levem à sua resolução.

2. **Compreensão dos dados** → A segunda fase consiste na mitigação dos dados de forma a extrair o máximo conhecimento possível sobre os mesmos, ou seja, é realizada toda a investigação, documentação e organização necessária de forma a facilitar a sua futura utilização.

3. **Preparação dos dados** → Realizada a fase de compreensão dos dados, nesta etapa é possível começar com o trabalho mais técnico, nomeadamente, limpeza dos dados, junção de dados, filtragem dos dados, entre outros.

4. **Modelação** → Na quarta fase desta metodologia, são criados modelos que visam resolver o problema em questão, sendo os parâmetros dos mesmos suscetíveis a otimização.

5. **Avaliação** → Construído o modelo na fase anterior, é agora necessário verificar se este modelo cumpre com os requisitos e atinge os objetivos previamente definidos. Nesta fase, é realizado todo este processo de extrema relevância.

6. **Implementação** → Após criação e avaliação do modelo, a última fase visa implementar o modelo no contexto para o qual foi definido, de forma a solucionar o problema inicial.

## 1.4 Estrutura do Documento

A estrutura deste documento de dissertação centraliza-se em seis capítulos envolventes. No presente capítulo 1, é realizada uma breve introdução referindo os tópicos da poluição ambiental, CUA e ETARs que constituem a motivação para a realização deste tema. Após isso, são apontados os objetivos inerentes à conclusão da dissertação, a metodologia de trabalho utilizada no desenvolvimento da dissertação e, por fim, é apresentada uma breve descrição da estrutura do documento.

No capítulo 2, é realizado o trabalho de investigação, denominado de estado de arte, onde é retratada a grande área desta dissertação, o ML, com a definição do seu modelo genérico, a sua importância bem como os paradigmas existentes. Posto isto, é realizado uma introdução ao ML para deteção de anomalias seguido de uma ponte entre esta sub-área de ML com as ETARs. Nesta vertente são caracterizadas as AR, definidas possíveis anomalias nas mesmas e, por fim, uma revisão à literatura existente na comunidade científica sobre este tema.

O terceiro capítulo, capítulo 3, retrata os materiais e métodos, onde é explicado o processo de recolha e armazenamento de dados bem como a consequente análise exploratória e manipulação dos mesmos.

Já no capítulo 4, estão descritas as experiências consideradas no desenvolvimento e respetiva avaliação dos modelos de ML.

No capítulo 5, como o nome indica, são apresentados e analisados todos os resultados obtidos ao longo das diversas experiências.

Por fim, e de forma a concluir, no capítulo 6, é exposta uma conclusão sobre o trabalho realizado nos capítulos anteriores bem como uma análise ao trabalho futuro.

Capítulo 2

# Estado de Arte

Este capítulo apresenta a análise do estado de arte com o propósito de identificar os artigos disponíveis na literatura inerentes ao tema desenvolvido nesta dissertação, bem como a identificação e análise dos trabalhos desenvolvidos neste âmbito.

A metodologia utilizada para elaborar os sub-capítulos consistiu numa análise de artigos de conferências e revistas nas base de dados do *Google Scholar*, *Science Direct*, *Research Gate*, entre outros. Dos vários termos pesquisados, destacam-se: *Anomaly detection in wastewater treatment plants*, *Fault detection in wastewater treatment plants*, *Anomalies in a wastewater treatment plant* e *Machine learning for anomaly detection*.

Posto isto, na secção 2.1 é mencionada uma análise geral de ML onde é mencionada a importância da mesma, o seu modelo genérico bem como os paradigmas existentes. Na secção 2.2 é retratada a aplicação de ML para deteção de anomalias. Já na secção 2.3 são caracterizadas as AR bem como as possíveis anomalias. Por último, na secção 2.4, é realizada a análise de trabalho relevante na área.

## 2.1 Machine Learning

A aprendizagem é um processo de mudança e aprimoramento de comportamentos por meio de exploração de novas informações no tempo. Quando esta definição é executada por máquinas, é denominada de ML [38].

O campo cientifico do ML, é um ramo da IA que, tal como definido pelo pioneiro Tom M. Mitchell: "Machine Learning é o estudo de algoritmos de computador que permitem que programas de computador sejam melhorados automaticamente através da experiência"[39].

Nesta área os algoritmos são constantemente referidos porque estabelecem a ponte entre as máquinas e o conhecimento. Pode-se pensar em algoritmos como um conjunto de instruções que um programa

especifica e que um computador processa [40]. Ou seja, de forma similar aos humanos que após realizarem tarefas continuadamente começam a aprender formas de a executarem mais eficazmente, os algoritmos de ML, por exemplo, após visualizarem vários exemplos de um mesmo objeto, conseguem identifica-lo num cenário completamente diferente porque aprenderam através de experiências anteriores [39][41].

### 2.1.1 Importância de Machine Learning

ML tem revolucionado diversas indústrias e esta influência continuará nas próximas décadas a revolucionar o mundo tal como o conhecemos [42]. Como referido por Andrew Ng: "Tal como a eletricidade transformou quase tudo há 100 anos, hoje eu tenho dificuldade em pensar numa indústria que não ache que a IA transformará nos próximos anos"[43].

Alguns exemplos de aplicabilidade desta área são:

- **Reconhecimento de discurso** → sistemas de reconhecimento de discurso mais sofisticados utilizados hoje em dia, como por exemplo, o *Google Translate*, tiram partido de modelos de ML para realizar a deteção dos mais variados idiomas [44].

- **Veículos autónomos** → modelos de ML são extremamente aplicados para conduzir veículos autonomamente, tais como, carros (como por exemplo Tesla Cars e Google Driver Less Cars) e drones. Maioritariamente, estas técnicas são aplicadas através de câmaras e sensores estrategicamente posicionadas nestes sistemas [45].

- **Filtragem de *emails* (*spam*)** → outra aplicabilidade desta área, passa por classificar *emails*, ou seja, os modelos "memorizam"*emails* classificados como *spam* pelo utilizador e, quando um novo email é recebido, o modelo vai compará-lo com *emails* de *spam* prévios e classifica-lo como indesejado ou não [46].

- **Campo médico** → numa área tão importante para todos nós como a medicina, também existem diversos progressos em projetos a decorrer para melhorar o dia a dia de todos nós. Um grande exemplo disso, é o TRISS (Trauma & Injury Severity Score) que foi criado em 1987 e, com base em dados introduzidos calcula a probabilidade de sobrevivência do paciente [47][48].

- ***Web* e Redes Sociais** → modelos classificativos e de *clustering* são utilizados por empresas como Facebook para analisar emoções positivas e negativas, bem como, realizar campanhas de marketing. Outro exemplo prende-se com a Google e Yahoo que utilizam estes modelos para encontrar similaridades entre páginas web [49].

- **Métodos Bayesian** → o teorema de Bayes é um dos métodos mais populares para calcular probabilidades condicionais dado um determinado conjunto de hipóteses. Isto pode ser usado para resolver problemas complexos de DS e problemas analíticos. Estes métodos são utilizados para,

por exemplo, gerar as sugestões da Netflix para novos filmes e séries que nos possam interessar bem como nos corretores textuais presentes nos dispositivos móveis [50].

- **Robótica** → a área de ML está naturalmente ligada ao futuro e é imprescindível mencionar a robótica. Os modelos tem ajudado imenso a levar os *robots* ao próximo nível, como por exemplo, nos variados sistemas de condução autónoma existentes em praticamente todos os meios de transporte atuais [51].

- **Deteção de Anomalias** → por último, e como o próprio nome indica, a área de ML tem facilitado imenso a deteção de anomalias em vários contextos como, por exemplo, deteção de intrusões, deteção de falhas e fraudes [52].

### 2.1.2 Modelo Genérico de Machine Learning

Como referido no ponto 2.1.1, o ML resolve diversos problemas que requerem a aprendizagem por parte da máquina, sendo que este processo de obtenção de conhecimento pode ser representado por um modelo genérico. A Figura 2.1 ilustra as principais componentes desse mesmo modelo [43]:


Figura 2.1: Modelo genérico de Machine Learning

Posto isso, conseguimos descrever o processo de implementação do modelo genérico com as seguintes etapas [53]:

- **Recolha e tratamento dos dados** → neste primeiro passo, de extrema importância, os dados são recolhidos e tratados para um formato em que possam ser utilizados como *input* para o algoritmo. Na maior parte das ocasiões, os dados ascendem aos milhares de registos e contém bastante ruído, tornando esta etapa crucial na obtenção de um bom modelo.

- **Feature Engineering** → obtidos os dados no passo anterior, é necessário validar quais os atributos são realmente relevantes para o problema em questão, removendo aqueles que não acrescentam valor ao processo de aprendizagem.

- **Escolha do algoritmo** → existem diversos algoritmos de ML porém, nem todos são adequados para determinados problemas. A seleção do melhor algoritmo para uma determinada situação é uma tarefa complexa, mas imperativa para obter os melhores resultados.

- **Definição de parâmetros** ⟶ realizada a escolha de algoritmo, na maioria dos casos, é necessário definir, inicialmente, os valores mais apropriados para os parâmetros.
- **Treino** ⟶ com os parâmetros definidos é fornecido ao modelo o subconjunto dos dados que serão utilizados para realizar o processo de aprendizagem.
- **Avaliar o desempenho** ⟶ a última fase, antes da implementação do sistema num contexto real, consiste em testar o modelo com o subconjunto dos dados que ele desconhece porque não os utilizou para treinar. Com este teste, é possível, através de diversas métricas de avaliação de performance dos modelos, comparar os valores atuais com os previstos, de forma a perceber o quão eficaz foi o processo de aprendizagem.

### 2.1.3 Paradigmas de Machine Learning

Como mencionado previamente no ponto 2.1.2, o modelo genérico de ML segue um processo de implementação com diversas etapas, sendo que, o grande intuito, foca-se na aquisição de conhecimento, sobre os dados em questão, por parte do algoritmo [38].

Em função do resultado desejado do algoritmo, é possível classificá-lo em diversas categorias. Nesta secção são mencionados os três principais tipos de aprendizagem de ML, nomeadamente, aprendizagem supervisionada, não supervisionada e por reforço. Posto isto, na Figura 2.2, estão ilustrados estes paradigmas de ML, bem como alguns exemplos de modelos que pertencem a cada um destes grupos [54].

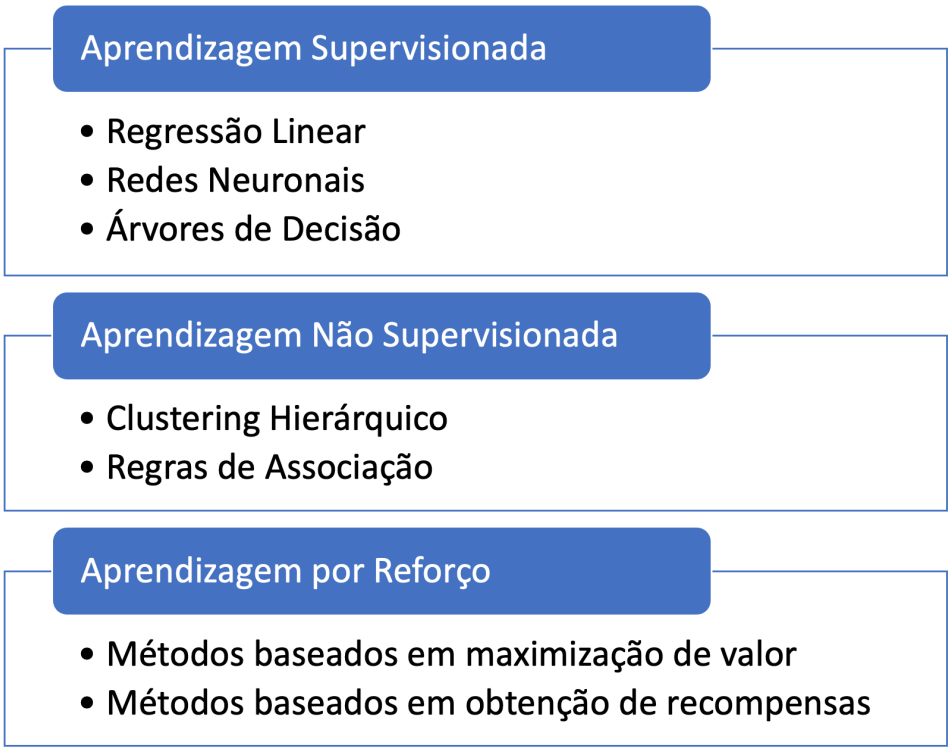



Figura 2.2: Paradigmas de Machine Learning

#### 2.1.3.1 Aprendizagem Supervisionada

A aprendizagem supervisionada é caracterizada por dados de treino classificados [55]. Tal como mencionado pela sua designação, esta aprendizagem pode ser interpretada, metaforicamente, como se um supervisor instruí-se o sistema sobre as classes associadas aos dados de treino [56].

Tipicamente, os algoritmos deste tipo, focam-se na realização de um mapeamento entre variáveis de entrada $x$ e variáveis de saída $y$ que, por exemplo, providencia a previsão de valores de saída para dados de entrada desconhecidos. Este mapeamento é realizado através da procura de uma função que melhor descreva os dados, ou seja, como ilustrado na Figura 2.3, os dados são utilizados como *input* para o algoritmo onde, é encontrada a função que represente os dados de saída com base nos dados de entrada. Realizada esta fase, os dados são processados e o *output* é retornado.

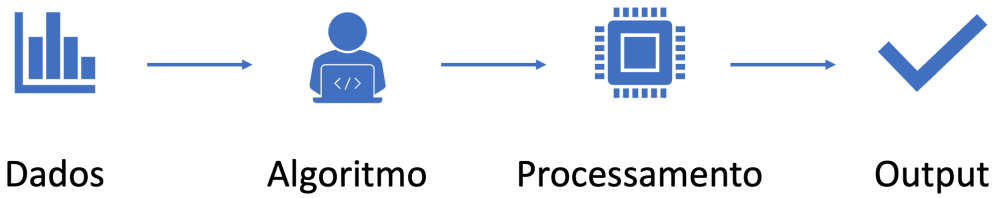



Figura 2.3: Processo geral aprendizagem supervisionada

Posto isto, os algoritmos de aprendizagem supervisionada permitem ás máquinas aprender o comportamento humano e realizar estes mapeamentos entre valores de entrada-saída de uma forma mais rápida e persistente do que os humanos [55]. Alguns dos algoritmos mais comuns são [57]:

- Nearest Neighbor
- Regressão Linear
- Redes Neuronais Artificiais
- Árvores de Decisão
- Naive Bayes
- Support Vector Machine (SVM)

No que diz respeito ás suas aplicabilidades, estes algoritmos são usados com provas dadas em áreas como, por exemplo:

- **Visão computacional** → Em visão por computador existem aplicações como, por exemplo, reconhecimento de objetos que, a partir de um grande conjunto de dados, visam identificar um objeto num contexto diferente. Os veículos autónomos são um dos exemplos mais conhecidos [58].

- **Reconhecimento de discurso** → No reconhecimento do discurso o algoritmo aprende a voz do utilizador e é depois capaz de a reconhecer. Exemplo disso são aplicações como a *Siri* da *Apple* que utilizam esta tecnologia [59].

- **Deteção de *spam*** → A deteção de *spam* em caixas de correio eletrónicas é uma prática muito comum. O *Yahoo* e *Outlook*, por exemplo, utiliza um algoritmo que aprende as diferentes palavras chave utilizadas no corpo de texto da mensagem e, consequentemente, classifica-o com base em conhecimento adquirido como *spam* ou não [60].

- **Bioinformática** → A bioinformática consiste no armazenamento de informação biológica de humanos, como por exemplo, impressões digitais, íris, entre outros. Um grande exemplo é os dispositivos móveis que são capazes de aprender a nossa informação biológica e utilizá-la para, por exemplo, autenticação [61].

- **Deteção Anomalias** → A deteção de anomalias é muito utilizada para identificar *outliers* nos dados. Um exemplo muito comum na atualidade são os modelos capazes de detetar tráfego irregular na rede informática [62].

#### 2.1.3.2 Aprendizagem Não Supervisionada

Quando se trata de lidar com problemas reais, na maioria das ocasiões, os dados não estão classificados e é imperativo desenvolver modelos de ML capazes de realizar corretamente este *labeling* [63].

Para satisfazer esta necessidade surge a aprendizagem não supervisionada que, ao contrário da aprendizagem supervisionada, é caracterizada por dados de treino não classificados, ou seja, este paradigma pesquisa padrões em conjuntos de dados que não tem intervenção humana para a realização de classificação, por outras palavras, um supervisor [64].

Também conhecida como *self-organization*, o processo geral deste paradigma consiste em realizar a *feature selection/extraction* dos dados, aplicar um algoritmo, com que agrupará este mediante as *features* selecionadas e, de seguida, interpretar os resultados obtidos com a segmentação. Por fim, é possível extrair conhecimento dos dados classificados através de um processo, como por exemplo, está ilustrado na Figura 2.4 [65].

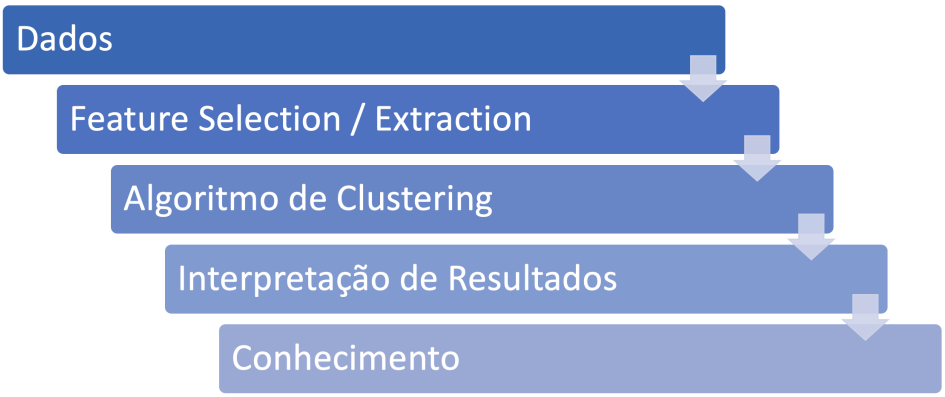



Figura 2.4: Exemplo de processo de aprendizagem não supervisionada

Assim sendo, para alcançarmos a extração de conhecimento dos conjuntos de dados, são necessários algoritmos capazes de agrupar e segmentar os dados. Nesta função, alguns desses algoritmos são [66]:

- *Clustering* Hierárquico
- *k-Means*
- *Apriori*
- *Eclat*

Com estes algoritmos, são gerados modelos que focam-se essencialmente à resolução de dois principais problemas, nomeadamente, agrupamento e redução de dimensões. Dentro destes dois problemas existem várias aplicações, tais como:

- **Segmentação** → com a segmentação é possível agrupar, por exemplo, clientes de um determinado serviço em função dos atributos comuns. Isto permite construir campanhas de *marketing* e estratégias de negócio [67].
- **Genética** → na área da genética, é realizado *clustering* de padrões de DNA de forma a analisar evoluções biológicas e com isto retirar conclusões e conhecimento importante para desenvolvimento científico [68].
- **Deteção de anomalias** → a deteção de anomalias visa identificar registos raros que podem ser considerados como sub-segmentos que não se enquadram em nenhum dos segmentos, como por exemplo, defeitos em componentes mecânicas [69].
- **Sistemas de recomendação** → através dos sistemas de recomendação, os utilizadores são agrupados mediante características e, encontrados os grupos, é possível recomendar conteúdo similar para utilizadores pertencentes ao mesmo aglomerado [70].

#### 2.1.3.3 Aprendizagem por Reforço

A aprendizagem por reforço é um paradigma muito distinto dos anteriores, os seus métodos envolvem uma estratégia de aprendizagem através de interatividade com o ambiente, seja esta, através de sequências de ações ou obtenção de recompensas [71]. Tal como na aprendizagem não supervisionada (2.1.3.2), neste paradigma não existe um supervisor, apenas números reais ou recompensas.

No contexto deste paradigma, o sistema é denominado de agente, e este, interage num determinado ambiente para realizar uma certa ação ou atingir um objetivo. Nestes contextos a variável tempo tem um papel crucial para a definição do problema. Como conseguimos observar através da Figura 2.5 , o agente realiza uma determinada ação no ambiente, e o ambiente, por sua vez, em função da ação realizada, retorna uma recompensa ou penalização para o agente [72].

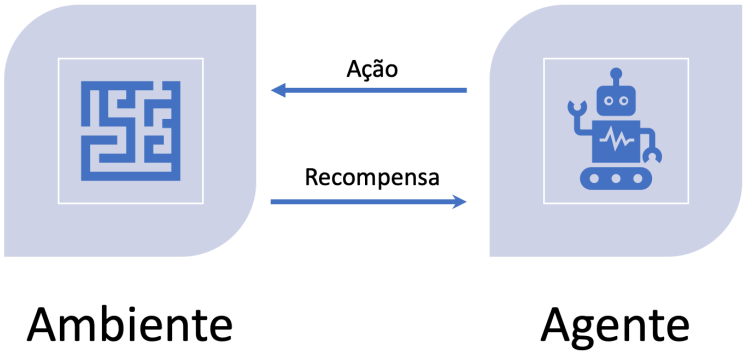



Figura 2.5: Processo geral aprendizagem por reforço

Desta forma, é possível definir dois métodos de aprendizagem por reforço, o método positivo, que ocorre por causa de uma determinada ação e impacta positivamente na ação tomada pelo agente. Por outro lado, o método negativo é definido como o fortalecimento do comportamento que ocorre devido a uma ação negativa que devia ter sido interrompida ou evitada. Com a junção destes dois métodos, o agente aprende quais as ações que ele deve levar a cabo para atingir o seu objetivo final [73].

Para alcançar os objetivos/recompensas com estes modelos, são necessários algoritmos que o tornem possível. Dois exemplos de modelos de aprendizagem por reforço são [74]:

- Processo Decisão *Markov*
- Q learning

Por fim, este paradigma é frequentemente usado para solucionar vários problemas, de destacar:

- **Área Robótica** → o uso de aprendizagem por reforço permite treinar a interação de *robots* com objetos, em movimentos como, por exemplo, agarrar, mesmo que eles nunca os tenham conhecido

na fase de treino. Esta aplicabilidade pode ser útil em várias vertentes, nomeadamente, numa linha de montagem fabril [75].

- **Mercado Financeiro** → a utilização de um agente de aprendizagem por reforço no mercado financeiro permite que o mesmo decida sobre uma tarefa, nomeadamente, se deve comprar, manter ou vender o bem. Os modelos de aprendizagem por reforço, neste caso, utilizam os padrões do mercado de forma a garantir que estão a atuar de forma correta [76].

- **Carros Autónomos** → também na área de veículos autónomos a aprendizagem por reforço pode ser utilizada para tarefas como, otimização de trajetória, planeamento dos movimentos e otimização do controlo [77].

## 2.2 Machine Learning para Deteção de Anomalias

A deteção de anomalias é o processo de identificar registos ou eventos em conjuntos de dados que diferem do normal. Esta vertente de ML é, normalmente, aplicada em dados não classificados (deteção de anomalias não supervisionada)[78].

Esta deteção assenta em dois princípios base, nomeadamente:

- As anomalias ocorrem muito raramente, ou seja, num *dataset* a existência de anomalias tem que ser invulgar.

- Os atributos das anomalias diferem significativamente dos registos normais.

Este processo, também conhecido como análise de *outliers*, pode ser aplicado em vários campos, como por exemplo:

- ***Performance* de Aplicações** → no campo das aplicações, sistemas de deteção de anomalias podem ser usados para detetar falhas antes que as mesmas afetem os utilizadores. Um grande exemplo desta aplicabilidade é a *Waze* que com mais de 100 milhões de utilizador por mês tira proveito de modelos de deteção de anomalias para detetar problemas nas estradas em tempo real. Tal como a *Waze*, outras empresas como *Telco* e *Adtech* implementam estes sistemas [79][80].

- **Qualidade de produto** → como os produtos estão em constante evolução, novas *releases*, novas funcionalidades ou alterações podem resultar em anomalias. Posto isto, qualquer negócio centralizado em produtos, beneficia com sistemas capazes de detetar anomalias prevenindo custos para estas empresas. Um exemplo disso, é em infraestruturas fabris, onde estes sistemas são implementados para detetar unidades defeituosas o mais rápido possível [81].

No que diz respeito ao processo de deteção de *outliers*, existem modelos de ML capazes de agilizar esta tarefa. Estes modelos estão na Tabela 2.1 divididos em duas categorias, aprendizagem supervisionada e aprendizagem não supervisionada onde, relativamente a aprendizagem supervisionada temos modelos

como SVM e *K-nearest neighbors* e, por outro lado, na aprendizagem não supervisionado, modelos como *K-means* e One-Class support vector machines (OCSVM) [82].

| Modelo | Aprendizagem Supervisionada | Aprendizagem Não Supervisionada |
|---|---|---|
| Isolation Forest | | |
| Support Vector Machines | | |
| Self-organizing maps | | |
| K-means | | |
| K-nearest neighbors | | |
| C-means | | |
| One-class support vector machine | | |
| Bayesian networks | | |
| Árvores decisão | | |
| LSTM AutoEncoders | | |

Tabela 2.1: Modelos para deteção de anomalias

### 2.2.1 Modelos Machine Learning

Como previamente mencionado, existem diversos modelos de ML para deteção de anomalias. Com isto, de forma a existirem termos de comparação mais interessantes, foram selecionais dois modelos de ML tradicional, e um modelo de Deep Learning (DL) [83]. Com isto, da tabela 2.1, foram selecionados dois modelos, nomeadamente, Isolation Forest (iF) e OCSVM que preenchem o grupo de modelos de ML tradicional. Por outro lado, o modelo de DL selecionado para deteção de anomalias foi a Long-Short Term Memory Auto Encoders (LSTM-AE).

#### 2.2.1.1 Isolation Forests

As iF são um algoritmo utilizado para deteção de anomalias que tem como base o algoritmo de classificação e regressão Random Forests (RF). Este algoritmo é uma extensão das Decision Trees (DT) e utiliza um método denominado de isolamento. Ao contrário dos outros algoritmos de deteção de anomalias que criam um perfil do que são instâncias normais, as iF isolam as anomalias. Este processo de isolamento é executado com partições iterativas do espaço de entrada de forma a separar uma nova observação do resto dos dados. Tal como nos princípios da deteção de anomalias, as iF tem como principais ideais que as anomalias são um pequena percentagem da totalidade dos dados e que os seus atributos diferem bastante dos restantes dados sendo assim suscetíveis ao processo de isolamento [84].

Este processo de isolamento, como é possível verificar na Figura 2.6, consiste na criação de uma floresta de várias Isolation Tree (iT) onde os ramos correspondem a divisões de uma determinada variável (que é escolhida aleatoriamente) e as folhas representam uma observação. A indicação da probabilidade de uma observação ser uma anomalia é atribuída através de um *score* que está correlacionado com o cumprimento dos caminhos necessários para isolar essa observação [85].

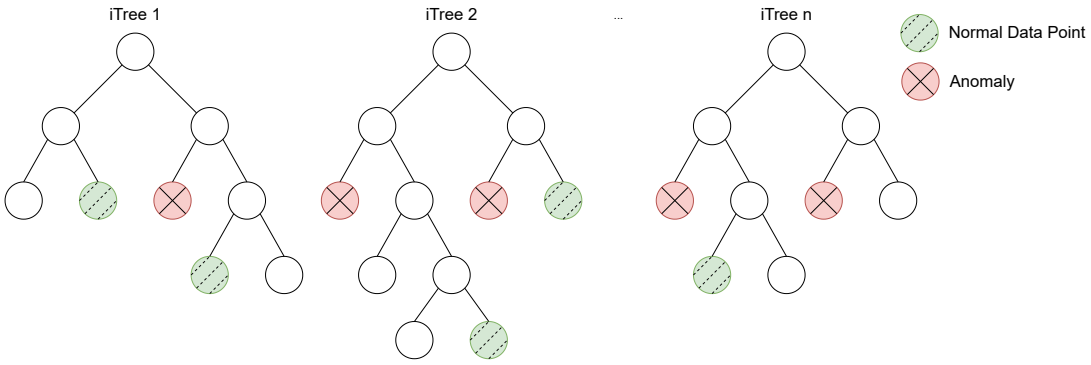



Figura 2.6: Processo de isolamento

Com isto, as iF consistem na construção de um conjunto de iT para um determinado conjunto de dados, definindo como anomalia as instâncias que possuem média de cumprimento de caminhos menores nas iT [85].

No que diz respeito ao algoritmo, as iF podem ser representadas por um conjunto de $t$ árvores binárias. Por sua vez, as anomalias produzem cumprimentos médios dos caminhos (da raiz até ás folhas) que são menores que observações normais [85]. Assim sendo, dado um dataset $X = \{x_1, ..., x_n\}, x \in \mathbb{R}^p$ para construir uma iT, $X$ é dividido recursivamente através da seleção aleatória de um atributo $q$ e um valor de divisão $p$ até que se verifique uma das seguintes condições:

1. a árvore atinge o limite de altura;
2. $|X| = 1$;
3. todos os valores de $X$ tem o mesmo valor;

Com isto, a tarefa de deteção de anomalias corresponde à classificação que reflete o grau de anomalia das observações, ou seja, com a ordenação dos registos em função do cumprimento dos caminhos / $anomaly\ score$, é possível classificar os registos que ficam no topo da lista de ordenação como anómalos.

#### 2.2.1.2 One-Class Support Vector Machines

As SVM foram introduzidas inicialmente por Vapnik no final do século 20 e, desde esse momento, vários investigadores criaram extensões das mesmas. Como principais características, destacam-se a boa performance em conjuntos de dados com ruído e poucos registos, o que torna este algoritmo extremamente interessante e útil para várias aplicações [86].

Na raiz deste modelo classificativo está um processo de construção de hiperplanos num espaço multidimensional que separa os casos de diferentes classes. De forma a alcançar um hiperplano ótimo, é aplicado um método de treino iterativo que é utilizado para minimizar a função de erro. Para além de tarefas de classificação, também é capaz de realizar regressão com variáveis categóricas ou contínuas [87].

Por sua vez, as OCSVM surgiram em 1999 da sugestão de Scholkopf et al. como uma extensão das SVM que, de forma a identificar observações anómalas, estimam a distribuição que encapsula a maioria dos registos e rotula como anormal os que estão longe dessa mesma distribuição de acordo com uma métrica adequada. Assim sendo, esta solução é construída estimando uma função de distribuição de probabilidade, que torna a maioria dos dados mais prováveis de ocorrerem que os restantes, e uma regra de decisão que separa estas observações pela maior margem possível [88].

Supondo um conjunto de dados extraído de uma distribuição de probabilidade $P$, é necessário estimar um subconjunto simples $S$ de modo que a probabilidade de um ponto de teste de $P$ esteja fora de $S$ e seja limitado por algum $v \in (0, 1)$ previamente definido. A solução para este problema centra-se em estimar a função $f$ que é positiva em $S$ e negativa no complemento $\overline{S}$. Ou seja, por outras palavras, Scholkopf desenvolveu um algoritmo que retorna uma função $f$ que torna o valor +1 numa pequena região capturando a maior parte dos dados, caso contrário, -1, como demonstra a seguinte equação:

$$f(x) = \begin{cases} +1 \; if \; x \in S \\ -1 \; if \; x \in \overline{S} \end{cases}$$

Tal como consta na Figura 2.7, este processo pode ser resumido como o mapeamento dos dados no espaço dos atributos usando um função de *kernel* apropriada e, após isso, tentar separar os vetores mapeados desde a origem com a margem máxima definida [88].

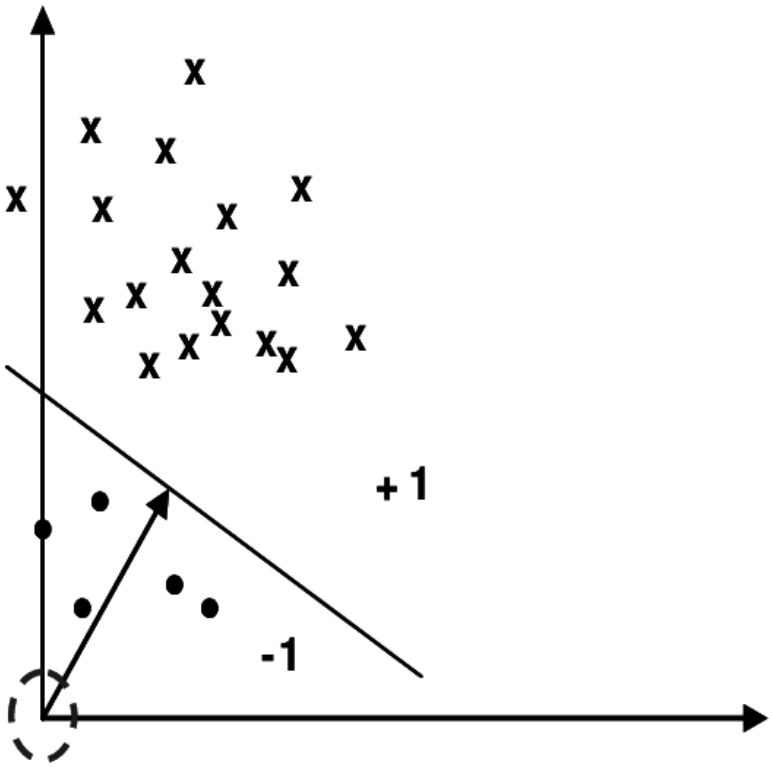



Figura 2.7: Processo de classificação do OCSVM

Embora todo este processo exija uma complexidade computacional na fase de aprendizagem mais elevada devido a um problema de programação quadrática, uma vez que a função de decisão esteja

definida, esta pode ser usada para prever a classe de novos dados sem grande esforço ou demora.

#### 2.2.1.3 Long Short-Term Memory Auto Encoder

As Recurrent Neural Networks (RNN) e a sua variante Long-Short Term Memory Networks (LSTM) são redes neuronais que fornecem uma capacidade de classificação e regressão com o domínio do tempo a ser contabilizado, ou seja, as entradas anteriores, influenciam o *output* no momento atual [89].

Uma RNN é formada com uma rede neuronal *feed-forward* de forma a conter *feeback* cíclico sobre si mesma. A parte recorrente de uma RNN consiste em *hidden layers* que tem como entrada os *timesteps* anteriores incorporados com o *timestep* atual, tal como ilustrado na Figura 2.8 [90].

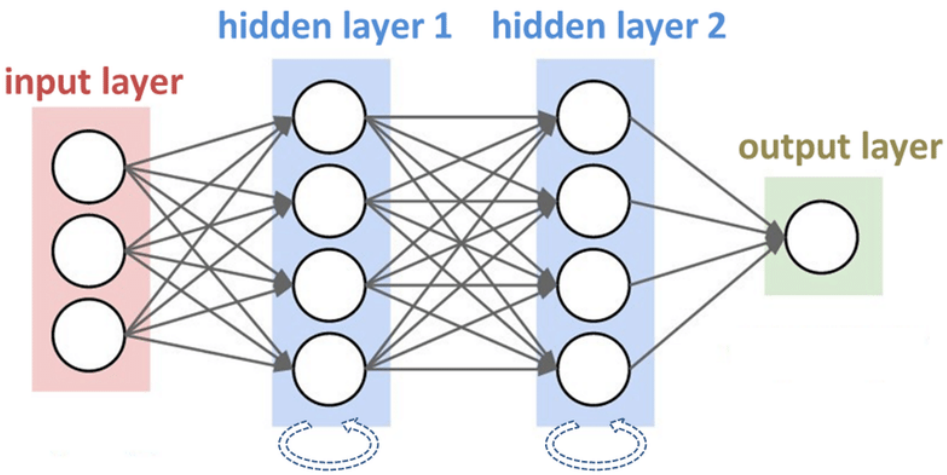



Figura 2.8: Exemplo de modelo RNN (extraído do artigo Diabetes Prediction: A Deep Learning Approach [1])

As tradicionais RNN são suscetíveis a um problema muito popular denominado de *vanishing gradients*, que surge em métodos de aprendizagem baseados em gradientes e *backpropagation*. Nestes métodos, cada peso da rede neuronal recebe uma atualização proporcional à derivada parcial da função de erro em relação ao peso atual em cada iteração de treino. Posto isto, o problema surge em alguns casos com o gradiente a tornar-se extremamente pequeno causando uma impossibilidade na alteração do peso. Nos piores casos, isto pode causar uma paragem na fase de treino da rede neuronal [91].

Por sua vez, as LSTM tem uma estrutura única que efetivamente aborda e resolve o problema previamente referido das RNN tradicionais. As LSTM utilizam uma estrutura de gradiente aditiva que inclui acesso direto ás ativações do *forget gate*, permitindo que a rede encoraje o comportamento desejado do gradiente de erro com recurso a atualizações frequentes de *gates* [92][93].

Assim sendo, ao conceito previamente mencionado dos modelos de LSTM, foi adicionado o modelo *Autoencoder* que é um tipo de rede neuronal que tem como principal objetivo reproduzir o *input*. É possível pensar nesta rede como um funil, que força a representação dos dados num espaço mínimo, também

[1] https://www.researchgate.net/figure/Deep-Neural-Network-architecture-In-deep-neural-network-activation-function-performs-a_fig1_332298424

conhecido como *encoder*, ilustrado na Figura 2.9. O mecanismo de *decoding* deve, por sua vez, reconstruir os dados iniciais através da sua representação num espaço mínimo. Através deste processo, é possível obter a *loss* que representa o erro entre os dados iniciais (*input*) e os dados reconstruídos (*output*). Posto isto, é possível identificar anomalias com base num determinado *threshold* que é calculado através dos erros de reconstrução, ou seja, se o valor real for bastante diferente do valor reconstruído (expectável), podemos estar perante uma anomalia [94].

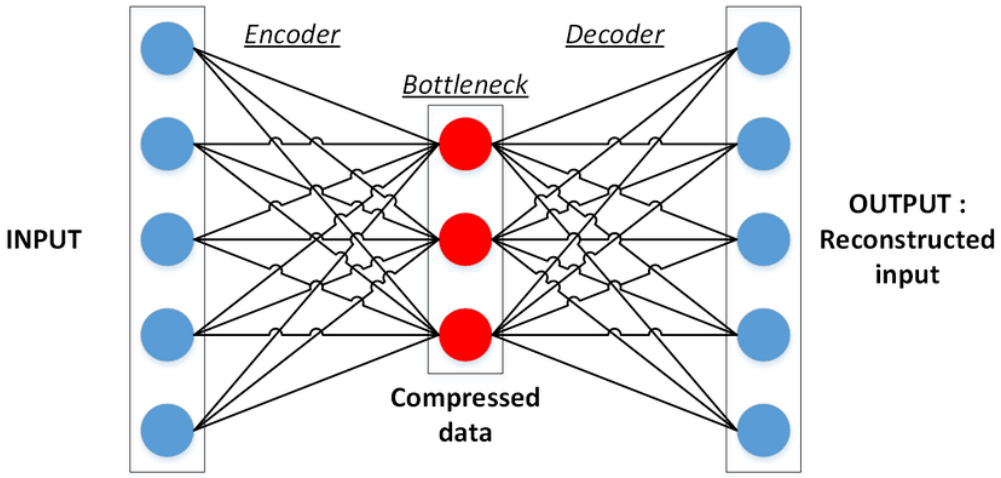



Figura 2.9: Arquitetura exemplo de modelo *Autoencoder* (extraído da publicação AutoEncoders with Tensorflow[2])

Posto isto, a abordagem da LSTM-AE, que é o modelo a ser utilizado nesta dissertação, está representada na Figura 2.10. Para a implementação do *encoder* e *decoder*, são utilizadas LSTM *cells* que são capazes de extrair dependências temporais entre instâncias. De forma a resolver problemas de deteção de anomalias, o modelo é treinado com observações não anómalas suficientes de forma que a arquitetura consiga reconstruir os registos normais com um erro de reconstrução pequeno aquando comparado com os registos anómalos [94].

[2]https://medium.com/analytics-vidhya/autoencoders-with-tensorflow-2f0a7315d161

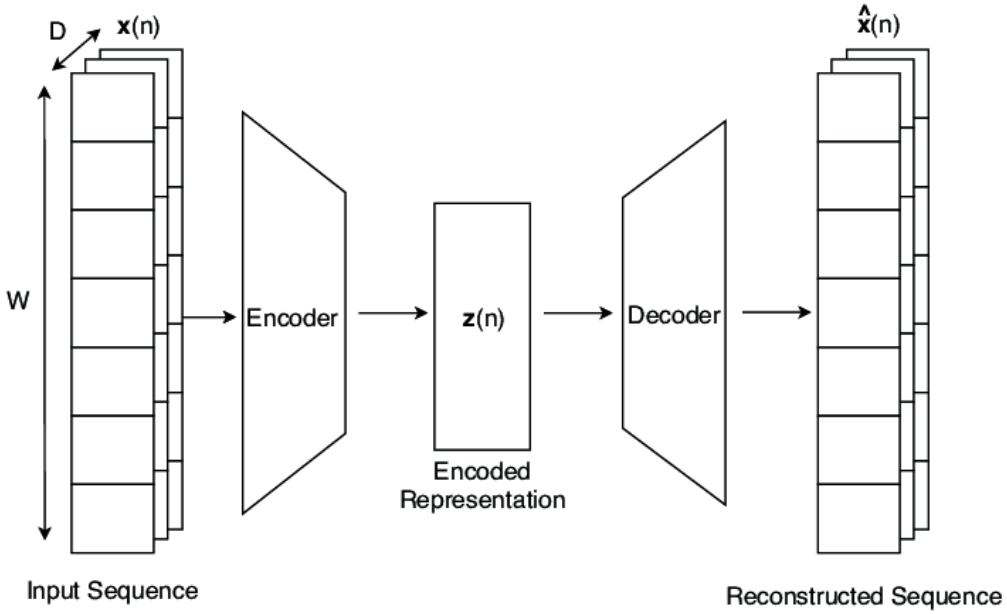



Figura 2.10: Exemplo da estrutura de um modelo da LSTM-AE (Extraído do artigo Detecting Mobile Traffic Anomalies[3])

## 2.3 Deteção de Anomalias em Estações de Tratamento de Águas Residuais

Uma ETAR é uma instalação que engloba vários processos com o intuito de remover substâncias poluentes das AR de modo a criar um efluente que esteja com a qualidade pretendida [33]. O controlo de qualidade é sustentado pela licença de descarga da ETAR que é emitida pela autoridade competente do Ministério do Ambiente [95].

### 2.3.1 Caracterização Qualitativa das Águas Residuais numa ETAR

Segundo o Decreto-Lei nº 152/97 as Águas Residuais Urbanas (ARU) podem ser definidas como sendo as Águas Residuais Domésticas (ARD) ou como a mistura das ARD com as Águas Residuais Industriais (ARI). A composição destas ARU numa determinada comunidade varia em função dos poluentes domésticos, comerciais e industriais e é condicionada pelo clima englobante [96].

As AR podem ser caracterizadas qualitativamente em três grandes grupos com o intuito de conceptualizar a fileira de tratamento, nomeadamente, características físicas, químicas e microbiológicas [95].

#### 2.3.1.1 Características Físicas em Águas Residuais

No que diz respeito ás características físicas nas AR, as principais são os sólidos, a cor, o cheiro, a viscosidade, a densidade e a temperatura [97]:

[3] https://www.researchgate.net/figure/LSTM-Autoencoder-for-Anomaly-Detection_fig2_336594630

- **Odor** → o cheiro exalado pelas AR resulta de substâncias dissolvidas na água que advém de atividade bioquímica e de, tal como na turvação, substâncias em suspensão. Para estes odores desagradáveis, as substâncias que mais contribuem são as aminas, o amoníaco, as diaminas e os sulfuretos orgânicos.

- **Temperatura** → a temperatura das AR está muito relacionada com a localização do aglomerado e com a descarga de águas quentes nos coletores. Esta temperatura é uma característica importante porque, para além de interferir nas características da água, influencia a velocidade dos processos bioquímicos, aumenta a solubilidade de solutos sólidos e líquidos e diminui a solubilidade de solutos gasosos. Esta característica surge, essencialmente de ARD e ARI.

- **Cor** → a cor é definida como a coloração com que a água fica após a remoção das partículas em suspensão. Esta cor pode ser originada por substâncias inorgânicas, substâncias orgânicas e por descargas industriais.

- **Sólidos** → os Sólidos Totais (ST) são das características físicas mais importantes nas AR e constituem as substâncias orgânicas e inorgânicas presentes em solução e em suspensão. Os sólidos são essenciais para o controlo dos processos de tratamento físicos e biológicos das AR e para garantir a conformidade com a legislação em vigor. Destes ST, a parte orgânica é denominada de Sólidos Voláteis Totais (SVT) e a parte inorgânica é designada de Sólidos Fixos Totais (SFT).

  Um outro tipo de sólidos são os Sólidos Suspensos Totais (SST) que são substâncias que ficam retidas por filtração das AR através de um filtro. Os Sólidos Suspensos (SS) podem ser classificados em dois tipos, fixos (Sólidos Suspensos Fixos (SSF)) e voláteis (Sólidos Suspensos Voláteis (SSV)) mediante a natureza orgânica ou inorgânica.

  Quanto à sua origem, os SS surgem das ARD e ARI, através de erosão do solo e infiltração nos coletores.

- **Viscosidade** → esta característica física representa a facilidade de escoamento dos fluídos e diminui com o aumento da temperatura das AR.

- **Turvação** → esta característica representa a dificuldade de penetração da luz na água. Esta dificuldade é provocada por partículas em suspensão que interferem com a penetração da luz.

  A origem destas partículas advém de diversas naturezas, desde microrganismos a partículas minerais.

- **Densidade** → a densidade de uma substância é definida pela razão entre a massa de determinado volume dessa substância e a massa de igual volume de água pura a 4°C, sendo que, a massa volúmica é caracterizada pela quantidade de matéria que existe numa unidade de volume de uma substância e, o peso volúmico é o peso da unidade de volume da substância.

#### 2.3.1.2 Características Químicas em Águas Residuais

Em termos químicos, as AR podem ser caracterizadas por [97]:

- **Oxigénio Dissolvido** → o Oxigénio Dissolvido (OD) representa o teor de moléculas de oxigénio dissolvidos na água. Este parâmetro representa um indicador de extrema relevância para analisar o estado de poluição da água. Quando existe pouca concentração de OD, significa que a água não está poluída, ao invés da elevada concentração de OD que indica uma água muito poluída.

- **Alcalinidade** → esta característica das águas reflete a eficácia da água em neutralizar os ácidos. Este parâmetro é essencial no tratamento das AR porque condiciona alguns processos de tratamento, tais como, tratamentos químicos, processos de remoção biológica de nutrientes e na remoção de amoníaco por extração em fase gasosa.

- **Potencial Hidrogeniônico (pH)** → este parâmetro avalia a concentração de iões de hidrogénio na água representando, dessa forma, o carácter ácido, neutro ou básico das AR. O valor de pH varia entre 0 e 14, sendo o 7 o valor da neutralidade. As soluções básicas apresentam valores de pH superiores a 7 sendo que as soluções ácidas apresentam valores de pH inferiores a 7.

  Este parâmetro é muito importante porque influencia o desenvolvimento de vida aquática e muitas reações importantes no tratamento de AR, bem como os processos biológicos e químicos.

- **Potencial redox** → o potencial de oxidação-redução, ou potencial redox, é um parâmetro que avalia a capacidade de uma substância ganhar ou perder eletrões, por outras palavras, de ser reduzida ou oxidada.

- **Cloretos** → esta característica quando presente nas AR pode condicionar a reutilização das águas após tratamento para rega agrícola e paisagística devido ao seu teor venenoso. Os cloretos advém essencialmente das atividades humanas, industriais, agrícolas e industriais.

- **Matéria orgânica** → os compostos orgânicos são os compostos que contem átomos de carbono. Nas AR existem diversos compostos, sendo fundamental destacar os seguintes:

  - **Proteínas** → o grupo de substâncias proteicas é o constituinte mais comum de todos os organismos. As proteínas possuem estruturas complexas e pesos moleculares muito variados. Tal como referido previamente, todos os compostos orgânicos contém carbono, e a proteína não é exceção. Quando a presença de proteínas nas AR é abundante, é muito provável que a sua decomposição gere odores.
  - **Glúcidos** → também conhecidos como glícidos, os glúcidos são compostos ternários de Carbono (C), Hidrogénio (H) e Oxigénio (O). Estes compostos incluem açúcares, celulose, amidos e fibras.
  - **Óleos e gorduras** → tal como os glúcidos, os óleos e as gorduras são compostos ternários de C, H e O resultantes da reação de álcoois.

- **CBO** → a quantidade de oxigénio utilizada na oxidação bioquímica da matéria orgânica é designada por Carência Bioquímica de Oxigénio (CBO). O período de incubação por referência costuma ser 5 dias, porém, por vezes este número de dias pode ser aumentado. É de notar que nem toda a matéria orgânica é metabolizada no período de 5 a 7 dias, assim sendo, para uma avaliação mais exata da matéria orgânica biodegradável, a incubação deve processar-se durante 20 dias de forma a oxidar cerca de 95 a 99 por cento da matéria orgânica carbonada.

  A CBO é bastante utilizada para, por exemplo, indicar a presença de grandes quantidades de matéria orgânica, porém, existem algumas limitações que podem pôr em causa esta assunção. Como a CBO representa a quantidade de oxigénio utilizada na oxidação bioquímica de matéria orgânica, um valor elevado de CBO representa, na teoria, uma quantidade elevada de matéria orgânica, no entanto, este oxigénio, pode ser metabolicamente consumido não apenas pela oxidação, mas por bactérias nitrificantes que utilizam oxigénio na oxidação bioquímica de compostos inorgânicos azotados, como a amónia, o que resulta, numa indução em erro quanto ao valor de matéria orgânica presente nas AR.

- **CQO** → enquanto a CBO está relacionada com a oxidação bioquímica, a Carência Química de Oxigénio (CQO) representa a oxidação química, ou seja, a CQO avalia a matéria orgânica presente na água, o que traduz o valor necessário de oxigénio para a oxidação química dessa matéria orgânica com dicromato de potássio.

  Em condições normais, os valores de CQO são, geralmente, mais elevados que os valores de CBO, isto porque o dicromato oxida quimicamente a matéria orgânica biodegradável e não biodegradável, aumentando os valores de oxigénio necessário.

- **COT** → de forma a calcular a concentração total de matéria orgânica presente nas AR é utilizado o parâmetro Carbono Orgânico Total (COT). Para determinar este parâmetro, a matéria orgânica é oxidada cataliticamente. A determinação deste valor em AR é pouco habitual.

- **Nutrientes** → nas AR existem diversos nutrientes, que podem dar origem a diversas consequências, de destacar os seguintes:

  - **Eutrofização** → a eutrofização da água é um processo em que elementos importantes para o desenvolvimento de plantas presentes nos poluentes das AR são libertados através da descarga das AR tratadas no meios aquáticos, proliferando excessivamente plantas microscópicas e macroscópicas.

  - **Azoto e compostos azotados** → o azoto, também conhecido como Nitrogénio (N), é um nutriente fundamental nas estruturas tecidulares proteicas. Este composto está presente nas AR de várias formas, nomeadamente, através de azoto molecular dissolvido, de forma inorgânica com iões de nitrato, nitrito e azoto amoniacal, ou de forma orgânica com aminas e proteínas.

    A presença de azoto na água é, geralmente, sinónimo de poluição aquática.

  - **Fósforo e compostos fosforados** → o Fósforo (F) é um composto que circula do solo para as plantas, destas, passa para os animais e completa o seu ciclo na água. A presença de elevados valores de compostos fosforados nas AR advém dos compostos orgânicos proteicos, detergentes e fertilizantes.

    Da mesma forma que no azoto, o fósforo pode originar a eutrofização das águas superficiais, pelo que, a sua concentração nas efluentes deve ser controlada.

- **Compostos orgânicos perigosos** → a abundância de compostos orgânicos sintetizados constitui um grande perigo para o meio ambiente, sendo que grande parte destas substâncias não é biodegradável. Um bom exemplo são os pesticidas utilizados na agricultura, nos tratamentos de madeiras, exterminação de roedores e outras pestes.

- **Enxofre e compostos sulfurados** → o Enxofre (S) é um componente do protoplasma dos seres vivos e faz parte dos aminoácidos das proteínas. A presença deste composto escurece a água e provoca maus odores.

- **Substâncias tensioativas** → as substâncias tensioativas são compostos orgânicos sintéticos constituídos por moléculas orgânicas que reduzem a tensão superficial das soluções aquosas.

  Também conhecidas como detergentes, as substâncias tensioativas tem um impacto negativo nos meios recetores pois dificulta a dissolução do oxigénio na água.

- **Metais pesados** → esta substância inclui, na sua definição, metais e semimetais e a sua presença nas AR deve-se a poluentes de origem industrial. Metais pesados, tais como, o bário, berílio, cádmio, chumbo, crómio, mercúrio e prata são extremamente tóxicos para a vida humana e animal.

#### 2.3.1.3 Características Microbiológicas em Águas Residuais

No que a características microbiológicas diz respeito, as águas residuais podem conter microrganismos. Estes microrganismos patogénicos, podem criar doenças porém, estes compostos tem um papel imprescindível na oxidação bioquímica dos poluentes orgânicos biodegradáveis.

Com a caracterização dos microrganismos das AR, é possível determinar o nível de contaminação por microrganismos patogénicos e, consequentemente, a possibilidade de transmissão de doenças. Estes organismos advém de fezes e urina humana e animal proveniente das ARD e ARI.

#### 2.3.1.4 Valores limite de emissão na descarga de Águas Residuais

Para garantir o controlo da qualidade das efluentes, no Decreto-Lei nº 236/98 [98] estão definidos os Valores limite de emissão (VLE) para que as descargas sejam seguras, nomeadamente:

| Parâmetros | Unidade de medida | Valores limite de emissão |
|---|---|---|
| pH | Escala de Sorensen | 6,0-9,0 |
| Temperatura | °C | Aumento de 3°C |
| $CBO_5$, 20°C | mg/l $O_2$ | 40 |
| CQO | mg/l $O_2$ | 150 |
| SST | mg/l | 60 |
| Alumínio | mg/l $Al$ | 10 |
| Ferro total | mg/l $Fe$ | 2,0 |
| Manganés total | mg/l $Mn$ | 2,0 |
| Cheiro | ——— | Não detectável |
| Cor | ——— | Não visível |
| Cloro residual disponível livre | mg/l $Cl_2$ | 0,5 |
| Cloro residual disponível total | mg/l $Cl_2$ | 1,0 |
| Fenóis | mg/l $Cl_6H_5OH$ | 1,0 |
| Óleos e gorduras | mg/l | 15 |
| Sulfuretos | mg/l $S$ | 1,0 |
| Sulfitos | mg/l $SO_3$ | 1,0 |
| Sulfatos | mg/l $SO_4$ | 2000 |
| Fósforo total | mg/l $P$ | 10 |
| Azoto amoniacal | mg/l $NH_4$ | 10 |
| Azoto total | mg/l $N$ | 15 |
| Nitratos | mg/l $NO_3$ | 25 |
| Aldeidos | mg/l | 1,0 |
| Arsénio total | mg/l $As$ | 1,0 |
| Chumbo total | mg/l $Pb$ | 1,0 |
| Cádmio total | mg/l $Cd$ | 0,2 |
| Crómio total | mg/l $Cr$ | 2,0 |
| Crómio hexavalente | mg/l $Cr$ (VI) | 0,1 |
| Cobre total | mg/l $Cu$ | 1,0 |
| Níquel total | mg/l $Ni$ | 2,0 |
| Mercúrio total | mg/l $Hg$ | 0,05 |
| Cianetos totais | mg/l $CN$ | 0,5 |
| Óleos minerais | mg/l | 1,0 |
| Detergentes | mg/l | 2,0 |

Tabela 2.2: Valores limite de emissão

### 2.3.2 Anomalias numa Estação de Tratamento de Águas Residuais

Uma anomalia é algo que se afasta do que é normal e expectável. No contexto de uma ETAR, as anomalias surgem, essencialmente, nos parâmetros das AR. Como indicado na Tabela 2.2, existem valores mínimos de emissão para cada parâmetro, porém, quando existe uma anomalia nestes indicadores, existem consequências associadas, nomeadamente [97]:

- **pH** → O pH deve estar próximo de 7, porém, quando tal não acontece, as AR tornam-se mais corrosivas e impossibilitam um ambiente favorável para o crescimento de microrganismos.
- **Temperatura** → Devido à receção de águas quentes, a temperatura da água nunca coincide com a temperatura ambiente. Quando existe um grande aumento da temperatura da água, existem consequências, tais como:
  - Diminui a densidade, viscosidade e tensão superficial.
  - Aumento da solubilidade de solutos sólidos e líquidos.
  - Diminuição da solubilidade de solutos gasosos.
  - Aceleração dos processos bioquímicos.
- **CBO** → O CBO é um indicador de quantidade de oxigénio utilizado na oxidação bioquímica da matéria orgânica. Quando o VLE é superior ao mínimo, é um indicador que existe uma maior quantidade de matéria orgânica nas AR que o expectável.
- **CQO** → Como previamente referido, de forma similar ao CBO, o CQO é um indicador de quantidade de oxigénio utilizado na oxidação da matéria orgânica, porém, neste caso, a oxidação é química e não bioquímica. Os valores de CQO são, geralmente, mais altos, o que pode ser comprovado com o seu VLE, que corresponde a 150 mg/l $O_2$. Tal como no CBO, quando este valor é superior ao mínimo, é indicativo da existência de uma maior quantidade de matéria orgânica nas AR.
- **Metais** → Os metais, tais como, alumínio, cádmio, chumbo, cobre, crómio e mercúrio tem VLE de mg/l bastante reduzidos, isto porque, caso os valores ultrapassem os limites de emissão, põem em causa a saúde humana e a vida aquática devido à sua toxidade [99].
- **Sulfuretos** → Os sulfuretos, são considerados registos anómalos quando existe uma ou mais miligramas por litro. Quando esta anomalia se verifica são causadas duas consequências notórias, os maus cheiros e a cor da água, que escurece. Para além destas duas consequências, também provoca a exalação de sulfureto de hidrogénio que é um ácido extremamente corrosivo e perigoso.
- **Fósforo** → É fundamental controlar os valores do fósforo pois, quando o seu VLE é ultrapassado, existem sérios riscos de originar a eutrofização das águas.
- **Azoto** → Tal com o fósforo, o azoto é uma substância que pode originar a eutrofização das águas quando os VLE são superiores ao expectável.

- **Arsénio** → O arsénio é um elemento químico que caso ultrapasse o seu limite imposto por lei, pode tornar-se muito perigoso devido à sua elevada toxidade.
- **Óleos** → Quando existe uma abundância de óleos e gorduras nas AR, é suscetível de causar problemas na ETAR bem como o entupimento de canalizações das fossas séticas.

## 2.4 Revisão da Literatura

Nesta secção é realizada a revisão da literatura na área de deteção de anomalias para o contexto das ETARs. Para a área em questão foram identificados alguns problemas e realizada uma análise crítica à literatura selecionada.

Num estudo desenvolvido por Mamandipoor et al. [100], os autores visam utilizar todos os dados provenientes dos sensores da ETAR da província de Treviso em Itália de forma a criar um modelo capaz de detetar falhas no processo de oxidação e nitrificação. Para realizar esta deteção de anomalias foi estudada uma proposta de modelo de ML com o recurso ao algoritmo da RNN, mais especificamente, LSTM devido à sua capacidade de capturar dependências a longo prazo. O conjunto de dados foi recolhido entre 20 de Janeiro de 2017 e 20 de Dezembro de 2017, resultando em mais de 5.1 milhões de pontos de dados provenientes de 12 sensores químicos e operacionais, sendo que, 11.5% (585792 registos) são falhas e 88.5% (4514280 registos) são normais. Estas falhas, são caracterizadas por regras de classificação, isto é, normalmente com o aumento do nível de amónia, o oxigénio é libertado, consequentemente, os níveis de amónia decrescem e o fluxo de oxigénio é interrompido, porém, a falha ocorre quando o nível de amónia não diminui com a libertação do oxigénio. Como referido, sendo os dados oriundos de sensores, são extremamente susceptíveis a valores nulos e valores em falta devido a, por exemplo, interrupções na conexão e falhas nos sensores. Os autores, nestes casos, optaram por ignorar as *features* cujos valores em falta eram superiores a 90% e por preencher os restantes com o último valor conhecido. Posto isto, são realizadas quatro experiências, comparando métodos tradicionais (análise estatística, Autoregressive Integrated Moving Average (ARIMA), modelo utilizando Principal Components Analysis (PCA) e SVM) à proposta apresentada das quais, destaca-se a LSTM que nas métricas de *accuracy*, *f1-score*, *precision* e *recall* atingiu resultados de 0.965, 0.927, 0.904 e 0.927, respetivamente. Por fim, os autores concluíram, com base nos resultados, que existe um vasto potencial no uso de *Deep Neural Networks* para a gestão de ETARs isto porque, não só ultrapassou os resultados dos métodos tradicionais como a *performance* atingiu valores superiores a 92%.

Num outro artigo desenvolvido por Haimi et al. [101] são analisados dados da ETAR Viikinmaki localizada na Finlândia. Estes, focam-se na conceção de um sistema inteligente de deteção de anomalias para o processo de lodo ativado, com o intuito de tornar mais eficaz e útil o uso dos sensores nesta operação. A ETAR Viikinmaki trata em média uma taxa de fluxo afluente de 250000$m^3$ por dia, da qual, cerca de 15% é industrial e os restantes 85% são domésticos. Os dados foram recolhidos em intervalos de uma hora durante um espaço temporal de dois anos (1 de Janeiro de 2009 até 31 de Dezembro de 2010).

No pré-processamento destes dados, apenas os *outliers* que evidentemente violavam os limites tecnológicos dos sensores ou continham valores nulos foram descartados. No que diz respeito aos indicadores selecionados, no processo de lodo ativado foram consideradas as seguintes variáveis, nomeadamente, amónio-nitrogénio na afluente para o biorreator, sólidos suspensos na afluente para o biorreator, taxa de fluxo de afluente para o biorreator, sólidos suspensos no biorreator, amónio-nitrogénio na efluente do biorreator, nitrato-nitrogénio na efluente do biorreator e, por último, pH na efluente do biorreator. Quanto aos métodos para deteção de anomalias, foram utilizadas quatro abordagens baseadas em extensões de *moving-window PCA* com comprimentos de janela adaptáveis e fixos, nomeadamente, *general procedure*, *moving-window procedure*, *adaptive window-lenght procedure* e *anomaly monitoring algorithm*. Os resultados experimentais demonstraram que quando as metodologias são aplicadas em conjunto com os parâmetros corretos, é possível detetar picos e desvios nas medidas, tal como anomalias processuais levando à conclusão que as técnicas propostas poderiam ser instaladas num *software* económico para monitorizar os sensores e anormalidades do processo de lodo ativado.

Os autores Harrou et al. [102] realizaram um caso de estudo numa ETAR localizada em Golden, cidade do estado do Colorado nos Estados Unidos da América com o intuito de apresentar uma solução inovadora e eficiente para a deteção de falhas com recurso a modelos de *unsupervised* DL. A ETAR de Golden tem como objetivo produzir um efluente adequado para irrigação sendo que, para tal, necessita de cumprir com determinados padrões regulamentados por agências locais. Os dados extraídos da mesma, correspondem a um espaço temporal de um mês (10 de Abril de 2010 até 10 de Maio de 2010) resultando em 4464 observações com 28 variáveis. Durante este período é conhecida a presença de uma falha que afetou o pH e a salinidade. Dos registos extraídos, foram selecionadas 7 variáveis com a ajuda de recomendações de especialistas, nomeadamente, pressão de permear da membrana do biorreator, oxigénio dissolvido da membrana do biorreator, permear turvação, condutividade do tanque de permeado, retornar o conteúdo de oxigénio dissolvido do lodo ativado, pH e total de sólidos suspensos do lodo ativado de retorno. Para atingir os objetivos, foi realizada uma abordagem híbrida Deep Belief Networks (DBN)-OCSVM. Esta abordagem combina DBN e OCSVM onde são usados dados sem falhas para construir o modelo de DBN e, de seguida, as *features* de saída desse modelo são usadas pelo OCSVM para monitorizar eventos anómalos, o que permitiu identificar a anomalia 5 dias antes do alerta que foi despoletado para os operadores. Foi possível concluir que caso existisse um sistema implementado com a capacidade e *performance* similares à abordagem presente neste artigo, os operadores seriam avisados antecipadamente da anomalia e, consequentemente, seria possível evitar a degradação causada, bem como os dois meses de reparações que foram necessários para normalizar a situação.

Num artigo elaborado por Aguado et al. [103] são apresentadas diversas abordagens de estatística multi-variável para analisar os dados do processo de tratamento de AR. Os dados utilizados neste artigo foram gerados num sistema que simulou, em intervalos de 15 minutos com um espaço temporal de 609 dias, um conjunto de dados no qual, estão incluídas as variáveis de, por exemplo, concentração de azoto na afluente, fluxo de azoto na afluente, temperatura da afluente, concentração de azoto, concentração de nitratos, concentração de sólidos suspensos, concentração de sólidos suspensos na efluente, entre

outras. Antes de aplicar qualquer algoritmo foram removidos dados não representativos provenientes de sensores não confiáveis, permitindo a deteção de períodos de calibração e valores em falta de forma a ser possível corrigir-los. Para não distorcer os dados, os dados corrompidos foram substituídos por estimativas de um modelo PCA baseado em dados dos cinco dias anteriores. Através desta correlação dos valores, a estrutura de correlação foi preservada resultando em estimativas mais precisas do que aquando da substituição por valores médios. Posto isto, foi então aplicado um modelo PCA baseado em 20 dias de operação que resultou num modelo com bons resultados para monitorizar o processo na maior parte do tempo. A segunda abordagem consistiu também num modelo PCA, porém, neste caso, um modelo adaptável no qual o fator de esquecimento tem um papel crucial na sua precisão. Adaptações mais rápidas, resultaram em tempos de deteção mais velozes, porém, geraram mais falsos alarmes. Uma outra tentativa, centrou-se no conceito da *batch technology* onde foi utilizada a implementação *full batch* que não acrescentou qualquer vantagem quando comparada à primeira abordagem (simples modelo PCA). Em suma, foi concluído que as informações geradas por estes modelos podem ajudar o operador a focar-se nas causas mais prováveis de distúrbios nos processos.

O último artigo revisto foi elaborado por Garcia-Alvarez [104] no qual foram utilizados modelos PCA para detetar falhas numa ETAR simulada. Esta técnica estatística multi-variável reduz a dimensão dos dados históricos originais projectando-os num espaço dimensional menor. Os dados da ETAR foram constituídos por 13 variáveis tais como, por exemplo, alcalinidade, nitrato e oxigénio dissolvido e as falhas consistem em três tipos de falhas processuais. Para detetar estas falhas foi construído o modelo PCA seguido de um processo de *cross-validation* de forma a avaliar a sua capacidade de generalização. Quanto à escolha das variáveis a utilizar na modelação foram utilizados dois métodos, com recurso à Cumulative Percent Variance (CPV), com máximo de 95% de nível de variância, foram identificas 5 principais componentes, por outro lado, quando este cálculo foi realizado com recurso ao gráfico SCREE, o número de componentes identificadas como melhor opção é 7, porque captura uma maior variedade do processo. Com esta abordagem, quando as condições meteorológicas estão chuvosas, os distúrbios detetados por vezes são falsos positivos devido à grande taxa de fluxo afluente. Para estes casos a melhor opção passa por construir um modelo PCA para cada modo de operação, ou seja, dois modelos, um para tempo seco e outra para tempo chuvoso onde, mediante o fluxo de afluente, é aplicado o modelo de PCA adequado (técnica *Switch-PCA*). Em suma, foi concluído no artigo que a abordagem com modelo PCA foi bem sucedida e identificou com sucesso as falhas críticas nos processos.

### 2.4.1 Análise Crítica

É transversal a todos os artigos a menção da elevada importância do papel das ETARs para o tratamento e reutilização das AR bem como, a não aproveitação da potencialidade dos sensores instalados nas várias etapas destas infraestruturas.

Existem diversas falhas a decorrer anualmente nas ETARs e a implementação de modelos de ML num sistema de monitorização adequado ás mesmas, pode alertar os operadores antecipadamente de forma

a prevenir custos elevados em extensas e demoradas reparações.

Quanto ao estudo realizado em [100], os resultados finais foram extremamente satisfatórios porém, no tratamento dos dados, todos os parâmetros cuja percentagem de valores nulos seja inferior a 90%, são preenchidos com o último valor conhecido. Para este caso em específico, seria interessante o recurso a outras técnicas para preencher estes valores, como por exemplo, através de um valor médio dos últimos N registos podendo, desta forma, beneficiar o modelo final. Ainda neste artigo, os melhores resultados foram atingidos com o modelo LSTM, porém, não existe referência à utilização de *cross validation* para compreender o quão genérico e flexível o modelo é. Por fim, foi aplicada a técnica de *random search* na procura dos melhores hiperparâmetros.

No artigo [101] foram realizadas abordagens baseadas em PCA com comprimentos de janelas adaptáveis e fixos, sendo detetadas as falhas e desvios com sucesso. Porém, tal como no primeiro artigo, teria sido interessante a aplicação de *cross validation*. No que diz respeito ao tratamento de dados, apenas os valores que claramente violavam os limites do *hardware* foram descartados. A utilização de outra abordagem para tratar os restantes valores seria interessante para detetar outros *outliers* e, consequentemente, beneficiar a *performance* do modelo final.

No terceiro artigo [102] os autores aplicaram uma abordagem híbrida em dados provenientes de uma ETAR no Colorado, no entanto, não indicaram qualquer tipo de tratamento de valores nulos e de *outliers*. Para além do tratamento de dados, tal como sucedeu nos artigos anteriores, seria interessante a implementação de *cross validation*. No entanto, os resultados foram positivos e todos os objetivos atingidos.

Os autores no artigo [103] utilizaram dados fictícios gerados através de um sistema. Quanto aos dados corrompidos, foram substituídos por estimativas de um modelo PCA, o que, certamente contribuiu na precisão do modelo final. No que diz respeito a modelos, foram utilizadas três abordagens com modelos baseados em PCA porém, não recorreram à técnica de *cross validation* e, não foi mencionado qualquer tratamento ou limpeza aos dados utilizados no processo de aprendizagem e avaliação do modelo.

No último artigo elaborado por Garcia-Alvarez [104], tal como nos artigos [101][103], são utilizados modelos PCA em dados provenientes de uma ETAR simulada. Nestes modelos PCA é utilizada uma técnica *cross validation* e, é utilizada a técnica *Switch*-PCA de forma a, mediante o fluxo do afluente, selecionar o modelo PCA mais adequado. Fica por mencionar o tratamento de dados em falta, *outliers* e valores nulos, no entanto, os resultados atingidos foram de encontro aos objetivos delineados.

De forma geral, é de notar que em nenhum dos artigos existiu quaisquer análise à possível existência de *overfitting* ou *underfitting* bem como, de salientar que apenas o artigo [104] utilizou dados meteorológicos e *cross validation* para refinar os modelos criados e apenas o artigo [100] realizou *tunning* de hiperparâmetros. Outro aspeto importante que poderia ser considerado pelos autores dos artigos, prende-se com verificação de uma possível sazonalidade influenciar os dados, por outras palavras, apenas dois artigos continham dados com um espaço temporal igual ou superior a dois anos e, assim sendo, seria vantajoso verificar a possível existência de uma correlação entre a época do ano e os respetivos valores dos parâmetros do conjunto de dados. Todas estas possíveis melhorias serão consideradas

e implementadas na presente dissertação. A tabela 2.3 resume a análise crítica aos pontos que foram considerados importantes.

| Artigo | *Cross Validation* | Tratamento dados completo | *Overfitting / Underfitting* | *Tunning* | *Dados meteorológicos* |
|---|---|---|---|---|---|
| Artigo [100] | × | × | × | √ | × |
| Artigo [101] | × | × | × | × | × |
| Artigo [102] | × | × | × | × | × |
| Artigo [103] | × | √ | × | × | × |
| Artigo [104] | √ | × | × | × | √ |

Tabela 2.3: Resumo análise crítica

Capítulo 3

# Materiais e Métodos

Neste capítulo são descritos os materiais e métodos utilizados para a conceção desta dissertação. Na primeira secção, secção 3.1, é descrito o processo de recolha e armazenamento dos dados utilizados, onde são enumerados os diversos ficheiros que foram inicialmente recebidos em formato *Excel*, bem como a implementação de uma *script* que preencheu um modelo de dados relacional. Na secção 3.2, estão presentes as descrições dos dados meteorológicos e do controlo analítico com uma análise estatística dos três indicadores a ser considerados no âmbito das fases seguintes. Por último, a manipulação dos dados está presente na secção 3.3 onde para cada um dos conjuntos de dados dos indicadores são realizadas operações de manipulação e tratamento de forma a preparar os mesmos para as experiências com os modelos de ML.

## 3.1 Recolha e Armazenamento dos Dados

Todos os projetos de ML têm como fase inicial a recolha dos dados seguida de uma análise do formato dos mesmos de forma a estruturar uma arquitetura de armazenamento que permita facilitar o acesso aos dados a partir de várias plataformas.

Os dados do controlo analítico, inicialmente, encontravam-se divididos por múltiplas pastas conforme a fase de tratamento, como por exemplo, Afluente Bruto, Entrada Reator Biológico e Lamas Mista. Dentro destas pastas continha múltiplos ficheiros *Excel* com os nomes do indicador em questão, tais como, amónia, fosforo total e alcalinidade. Esta estrutura inicial, encontra-se representada na Tabela 3.1.

| Pasta | Ficheiro |
|---|---|
| *Controlo Analítico* | |
| Afluente Bruto | azoto_total.csv |
| | cbo.csv |
| | cqo.csv |

| | |
|---|---|
| | fosforo_total.csv |
| | ph.csv |
| | sst.csv |
| Camara Degaseificacao | amonia.csv |
| | nitratos(no3).csv |
| Efluente Primario | sst.csv |
| Efluente Tratado | amonia.csv |
| | azoto_total.csv |
| | cbo.csv |
| | cqo.csv |
| | fosforo_total.csv |
| | nitrato(no3).csv |
| | ortofosfatos.csv |
| | ph.csv |
| | sst.csv |
| Entrada Reator Biologico | azoto_total.csv |
| | cbo.csv |
| | fosforo_total.csv |
| | ph.csv |
| | sst.csv |
| Lamas Biologicas 1 | amonia.csv |
| | ivl.csv |
| | nitratos(co3).csv |
| | oxigenio.csv |
| | ph.csv |
| | sst.csv |
| | ssv.csv |
| Lamas Biologicas 2 | amonia.csv |
| | ivl.csv |
| | nitratos(co3).csv |
| | oxigenio.csv |
| | ph.csv |
| | sst.csv |
| | ssv.csv |
| Lamas Biologicas Espessadas | st.csv |
| Lamas Biologicas Recirculadas | st.csv |
| Lamas Desidratadas | ph.csv |

| | |
|---|---|
| Lamas Digeridas 1 | acidos_gordos_volateis.csv |
| | alcalinidade.csv |
| | amonia.csv |
| | fosforo_total.csv |
| | ph.csv |
| | st.csv |
| | sv.csv |
| Lamas Digeridas 2 | acidos_gordos_volateis.csv |
| | alcalinidade.csv |
| | amonia.csv |
| | fosforo_total.csv |
| | ph.csv |
| | st.csv |
| | sv.csv |
| Lamas Espessadas | sst.csv |
| Lamas Mistas | st.csv |
| | sv.csv |
| Lamas Primarias 1 | st.csv |
| | sv.csv |
| Lamas Primarias 2 | st.csv |
| | sv.csv |
| Poco Escorrencias | azoto_total.csv |
| | cqo.csv |
| | fosforo_total.csv |
| | sst.csv |
| Tanque Anoxico 1 | nitratos(no3).csv |
| | oxigenio.csv |
| Tanque Anoxico 2 | nitratos(no3).csv |
| | oxigenio.csv |
| *Meteorologicos* | |
| Outros | meteo.csv |

Tabela 3.1: Estrutura inicial dos conjuntos de dados

Com o intuito de organizar os dados e armazená-los de forma relacional, foi criado um *script* (Listagem A.1 do Apêndice A) que dinamicamente acedia aos ficheiros nas sub-pastas e armazenava os valores dos vários indicadores numa base de dados *MySQL*.

Concluída a inserção dos dados na base de dados relacional, é possível aceder facilmente aos dados

dos vários indicadores a partir de uma *query* à base de dados que está ilustrada na Figura 3.1. Este modelo consiste em quatro tabelas, nomeadamente, a *city_table*, *weather_table*, *indicador_table* e *indicador_value_table*. A *city_table* armazena o nome da ETAR bem como a sua latitude e longitude, a *weather_table* contém todos os dados relacionados com a meteorologia desta cidade, tais como, temperature, humidade e velocidade do vento. Por fim, as últimas duas tabelas, *indicador_table* e *indicador_value_table* armazenam os registos relacionados com as características do indicador, como o nome, descrição e unidades, e os valores correspondentes ás várias observações recolhidas através dos sensores da ETAR.

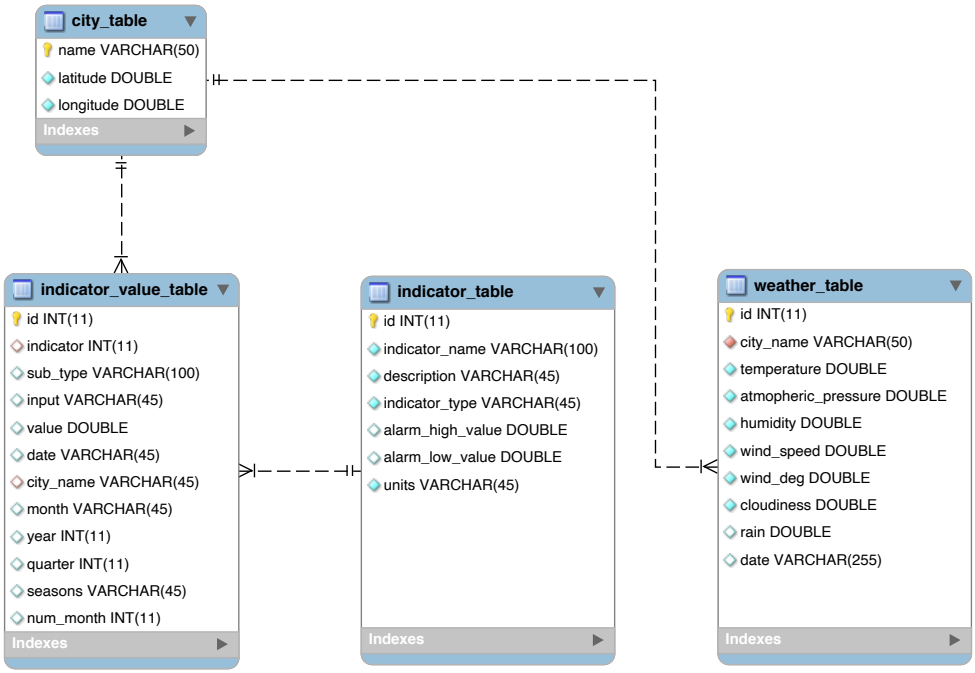



Figura 3.1: Modelo Relacional da Base de Dados utilizado para armazenar os conjuntos de dados

## 3.2 Exploração dos Dados

Concluída a recolha e armazenamento inicial dos dados, a seguinte fase centra-se na exploração inicial dos dados. Em todos os projetos de ML e DS, o processo de visualização e interpretação dos dados é crucial no sucesso das etapas que se seguem de forma a atingir os melhores resultados. Posto isto, este capítulo retrata as etapas que foram levadas a cabo no que diz respeito à exploração dos dados de controlo analítico da ETAR bem como os dados meteorológicos.

### 3.2.1 Dados do Controlo Analítico

O conjunto de dados do Controlo Analítico da ETAR contém registos sobre os indicadores que são extraídos através de sensores presentes localmente nas várias fases de tratamento. Com estes dados foi necessário, numa primeira fase, realizar uma análise estatística de forma a calcular a média, moda, mediana e desvio padrão, bem como, analisar o número de registos, valores mínimos e máximos e a periodicidade dos registos.

Na Tabela 3.2 é possível verificar a análise estatística previamente referida a todos os indicadores por fase de tratamento da ETAR bem como as unidades de medida dos mesmos.

| Fase Tratamento | Indicador | U. Medida | Registos | Mínimo | Máximo | Desv. Padrão | Média | Moda | Mediana | Period. |
|---|---|---|---|---|---|---|---|---|---|---|
| Afluente Bruto | Azoto Total | mg/l | 95 | 0 | 145 | 28.050 | 66.045 | 0 | 71.3 | Semanal |
| | CBO | mg/l | 48 | 0 | 1050 | 270.389 | 438.979 | 0 | 405.5 | Quinzenal |
| | CQO | mg/l | 95 | 0 | 1568 | 340.661 | 699.547 | 0 | 719 | Semanal |
| | Fósforo Total | mg/l | 97 | 0 | 14.5 | 3.333 | 7.118 | 0 | 7.64 | Semanal |
| | pH | Esc. Sorensen | 205 | 0 | 8.94 | 1.471 | 7.337 | 7.6 | 7.6 | Semanal |
| | SST | mg/l | 95 | 0 | 820 | 172.686 | 264.916 | 0 | 245 | Semanal |
| Câmara | Amónia | mg/l | 109 | 0 | 58.5 | 11.994 | 7.415 | 0.5 | 2.6 | Semanal |
| Degaseificação | Nitratos (NO3) | mg/l | 90 | 0 | 19.8 | 3.403 | 4.729 | 0 | 4.905 | Semanal |
| | SST | mg/l | 95 | 0 | 1380 | 264.051 | 317.842 | 0 | 265 | Semanal |
| Efluente Tratado | Amónia | mg/l | 214 | 0 | 47.6 | 10.997 | 11.379 | 0 | 7.4 | 2 dias |
| | Azoto Total | mg/l | 111 | 0 | 54 | 9.504 | 15.859 | 11.4 | 13.4 | Semanal |
| | CBO | mg/l | 49 | 0 | 12 | 3.023 | 4.327 | 2 | 3 | Quinzenal |
| | CQO | mg/l | 98 | 0 | 53.9 | 10.724 | 28.311 | 0 | 28.4 | Semanal |
| | Fósforo Total | mg/l | 104 | 0 | 6.13 | 1.183 | 1.315 | 0.5 | 0.825 | Semanal |
| | Nitratos (NO3) | mg/l | 209 | 0 | 26.3 | 3.199 | 4.284 | 0 | 4 | 2 dias |
| | Ortofosfatos | mg/l | 110 | 0 | 7.99 | 1.701 | 1.538 | 0.5 | 0.886 | Semanal |
| | pH | Esc. Sorensen | 273 | 0 | 8.49 | 1.20 | 6.832 | 6.9 | 6.99 | Semanal |
| | SST | mg/l | 98 | 0 | 33 | 5.501 | 5.541 | 3 | 4 | Semanal |
| Entrada Reator | Azoto Total | mg/l | 96 | 0 | 126 | 25.878 | 58.879 | 0 | 63.7 | Semanal |
| Biológico | CBO | mg/l | 47 | 0 | 900 | 183.586 | 274.894 | 240 | 240 | Quinzenal |
| | Fósforo Total | mg/l | 96 | 0 | 14.1 | 3.141 | 6.844 | 0 | 7.375 | Semanal |
| | pH | Esc. Sorensen | 201 | 0 | 9.56 | 1.162 | 7.404 | 7.5 | 7.52 | Semanal |
| | SST | mg/l | 96 | 0 | 480 | 73.688 | 96.343 | 0 | 78 | Semanal |
| Lamas Biológicas 1 | Amónia | mg/l | 192 | 0 | 49.9 | 8.835 | 4.607 | 0 | 0.5 | 2 dias |
| | IVL | mg/l | 96 | 0 | 258.3 | 80.969 | 129.169 | 0 | 145.15 | Semanal |
| | Nitratos (NO3) | mg/l | 192 | 0 | 21.7 | 3.118 | 3.429 | 0 | 3.38 | 2 dias |
| | Oxigénio | mg/l | 167 | 0 | 6.65 | 1.459 | 1.502 | 0 | 0.9 | 2 dias |
| | pH | Esc. Sorensen | 298 | 0 | 8.73 | 1.045 | 6.554 | 6.7 | 6.655 | Semanal |
| | SST | mg/l | 74 | 0 | 8100 | 1678.577 | 4914.851 | 0 | 5030 | Semanal |
| | SSV | mg/l | 41 | 0 | 6570 | 1576.244 | 3691.244 | 0 | 4060 | Quinzenal |
| Lamas Biológicas 2 | Amónia | mg/l | 196 | 0 | 35.4 | 8.129 | 4.629 | 0 | 0.5 | 2 dias |
| | IVL | mg/l | 96 | 0 | 312.4 | 90.789 | 129.44 | 0 | 142 | Semanal |
| | Nitratos (NO3) | mg/l | 193 | 0 | 23.1 | 3.749 | 3.517 | 0 | 2.78 | 2 dias |
| | Oxigénio | mg/l | 172 | 0 | 8.7 | 1.819 | 2.062 | 0 | 1.32 | 2 dias |
| | pH | Esc. Sorensen | 296 | 0 | 9.09 | 0.896 | 6.588 | 6.4 | 6.64 | Semanal |
| | SST | mg/l | 78 | 0 | 8210 | 2464.846 | 3565.586 | 0 | 4235 | Semanal |
| | SSV | mg/l | 40 | 0 | 6710 | 2131.129 | 2681.825 | 0 | 3250 | Quinzenal |
| Lamas Bio. Espessadas | ST | mg/l | 53 | 0 | 48.134 | 10.853 | 25.720 | 0 | 26.634 | Quinzenal |
| Lamas Bio. Recirculadas | ST | mg/l | 94 | 0 | 28.954 | 3.781 | 8.671 | 0 | 8.848 | Semanal |
| Lamas Digeridas 1 | AGV | mg/l | 94 | 0 | 67900 | 14213.697 | 36613.404 | 0 | 200 | Semanal |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Alcalinidade | mg/l | 94 | 0 | 13850 | 2046.617 | 1928.617 | 0 | 2000 | Semanal |
| | Amónia | mg/l | 94 | 0 | 1055 | 347.627 | 520.535 | 0 | 660 | Semanal |
| | Fósforo Total | mg/l | 94 | 0 | 765 | 196.218 | 287.824 | 0 | 355.5 | Semanal |
| | pH | Esc. Sorensen | 124 | 0 | 8.4 | 1.318 | 6.764 | 7.15 | 7.15 | Semanal |
| | ST | mg/l | 94 | 0 | 26.71 | 7.432 | 10.774 | 0 | 13.947 | Semanal |
| | SV | mg/l | 94 | 0 | 17.424 | 5.225 | 7.511 | 0 | 9.506 | Semanal |
| Lamas Digeridas 2 | AGV | mg/l | 95 | 0 | 66800 | 13892.099 | 3593.158 | 240 | 240 | Semanal |
| | Alcalinidade | mg/l | 95 | 0 | 8250 | 1456.578 | 2275.579 | 2080 | 2080 | Semanal |
| | Amónia | mg/l | 95 | 0 | 1100 | 283.885 | 631.546 | 0 | 705 | Semanal |
| | Fósforo Total | mg/l | 95 | 0 | 630 | 149.825 | 348.929 | 0 | 370 | Semanal |
| | pH | Esc. Sorensen | 130 | 0 | 8.5 | 1.169 | 6.756 | 7.18 | 7.05 | Semanal |
| | ST | mg/l | 95 | 0 | 36.684 | 4.766 | 15.673 | 0 | 16.12 | Semanal |
| | SV | mg/l | 95 | 0 | 25.862 | 4.239 | 10.486 | 0 | 11.148 | Semanal |
| Lamas Mistas | ST | mg/l | 95 | 0 | 52.792 | 12.051 | 25.695 | 0 | 25.523 | Semanal |
| | SV | mg/l | 52 | 0 | 41.658 | 10.348 | 19.921 | 0 | 19.794 | Semanal |
| Lamas Primárias 1 | ST | mg/l | 96 | 0 | 121.428 | 17.715 | 31.633 | 0 | 31.776 | Semanal |
| | SV | mg/l | 74 | 0 | 103.810 | 16.050 | 26.186 | 0 | 27.297 | Semanal |
| Lamas Primárias 2 | ST | mg/l | 96 | 0 | 111.592 | 17.904 | 28.412 | 0 | 28.891 | Semanal |
| | SV | mg/l | 74 | 0 | 99.821 | 16.483 | 23.125 | 0 | 24.29 | Semanal |
| Poço Escorrências | Azoto Total | mg/l | 48 | 0 | 315 | 86.940 | 170.896 | 0 | 172.5 | Quinzenal |
| | CQO | mg/l | 48 | 0 | 5710 | 1336.944 | 1625.833 | 0 | 1336.944 | Quinzenal |
| | Fósforo Total | mg/l | 48 | 0 | 138 | 20.573 | 29.469 | 0 | 28.26 | Quinzenal |
| | SST | mg/l | 48 | 0 | 3550 | 974.804 | 1040.452 | 0 | 750.5 | Quinzenal |
| Tanque Anóxico 1 | Nitratos (NO3) | mg/l | 95 | 0 | 13 | 1.618 | 0.707 | 0 | 0.443 | Semanal |
| | Oxigénio | mg/l | 52 | 0 | 3.560 | 0.484 | 0.310 | 0.2 | 0.21 | Semanal |
| Tanque Anóxico 2 | Nitratos (NO3) | mg/l | 95 | 0 | 12.9 | 1.492 | 0.663 | 0 | 0.483 | Semanal |
| | Oxigénio | mg/l | 61 | 0 | 3.27 | 0.413 | 0.349 | 0.38 | 0.29 | Semanal |

Tabela 3.2: Análise Estatística dos Dados de Controlo Analítico

Concluída a análise estatística, é necessário escolher a fase de tratamento da qual serão selecionados os indicadores a serem considerados. Posto isto, a fase do Efluente Tratado, por ser esta a última fase antes do despejo das Águas Residuais Tratadas (ART) para o meio ambiente, onde existe um grande perigo para os seres vivos em caso de presença de anomalias e representando assim uma maior importância na deteção de anomalias, foi a escolha tomada.

No total, existem 16 indicadores distintos que estão presentes nas 16 fases de tratamento da ETAR perfazendo um total de 7379 registos, ou seja, para cada um destes indicadores existe, em média, por fase de tratamento, 110 registos. No que à fase do Efluente Tratado diz respeito, em termos temporais, o pH contém registos entre Janeiro de 2016 e Maio de 2020 enquanto os restantes 8 indicadores apenas aglomeram dados entre Agosto de 2018 e Maio de 2020. Em relação ás unidades de medida, à exceção do pH, todos os indicadores apresentam registos em mg/l. Sobre as periodicidades, na grande maioria dos indicadores verifica-se periodicidade semanal com exclusão, por exemplo, da Amónia e CBO que apresentam periodicidade de 2 em 2 dias e quinzenal, respetivamente. De ressalvar que nesta análise inicial, não foram detetados quaisquer valores nulos no conjunto de dados, no entanto, existem diversos *timesteps* em falta que serão tratados à posteriori. No que diz respeito aos valores máximos e mínimos, constata-se que grande parte dos valores mínimos são 0 e que muitos deles, contém também o valor de moda 0 ou seja, significa que o valor com maior número de ocorrências é 0, porém, para grande parte dos indicadores, é fisicamente inconcebível no ambiente em questão, sendo passível de conclusão

que estes valores poderão estar incorretos e possivelmente serão futuramente considerados como nulos e corretamente substituídos ou removidos. Sobre a media e desvio padrão, é possível validar que, por exemplo, a CQO na fase do Efluente Tratado neste espaço temporal foi, em média, de 28.311 mg/l com um desvio padrão de 10.724 mg/l.

Posto isto, o passo seguinte à escolha da fase de tratamento, centrou-se na escolha de três indicadores para realizar a futura deteção de anomalias porque seria inimaginável fazê-lo para todos os 16 indicadores existentes. Esta seleção centrou-se na análise técnica concebida e, foi utilizado como métrica de escolha a quantidade de registos e a periodicidade de forma a ter um número aceitável de observações para aplicações de técnicas, tais como, DL. Os indicadores Azoto Total, Nitratos e pH possuem na fase do Efluente Tratado 111, 209 e 273 registos respetivamente, o que, em conjugação com a sua periodicidade determinou como os 3 indicadores selecionados para aplicação de técnicas de ML para deteção de anomalias. Alguns dos gráficos não presentes nas seguintes subsecções, encontram-se no Apêndice B.

#### 3.2.1.1 Azoto Total

O Azoto Total é a soma do Nitrato-Nitrogénio ($NO_3$-$N$), Nitrito-Nitrogénio ($NO_2$-$N$), Amónia-Nitrogénio ($NH_3$-$N$) e Nitrogénio organicamente ligado. Como nas ETARs o Azoto pode ser encontrado de várias formas, cada variação é analisada individualmente e o Azoto Total é calculado com a soma das quatro variações [105].

O controlo da concentração de Azoto Total no efluente tratado da ETAR é crucial de forma a evitar danos nos seres vivos que, no caso dos seres humanos são, por exemplo, sintomas de insuficiência mental, inconsciência, lábios azulados, náuseas e vómitos [106].

No que diz respeito aos registos deste indicador, existem no total 111 registos que estão inseridos no intervalo de tempo de 1 de Agosto de 2018 e 27 de Maio de 2020. Quanto à periodicidade, em média, o Azoto Total contém observações numa periodicidade semanal, porém, existem algumas semanas que não contém registos, nomeadamente:

- 2018-10-03
- 2018-10-17
- 2019-04-29
- 2019-07-26

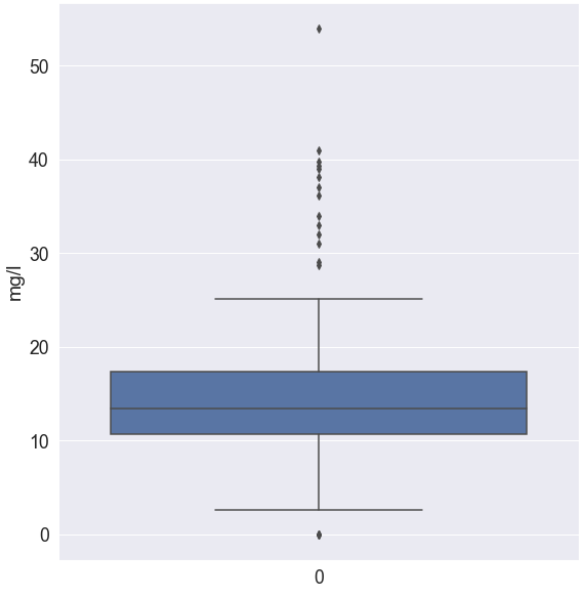
No contexto do Azoto Total no efluente tratado, a Tabela 3.3 apresenta a análise estatística de onde é possível retirar algumas ilações. A média do valor das observações é 15.59, o que, por si só, representa um valor ligeiramente acima dos limites permitidos por lei, que neste caso, é 15 (Tabela 2.2). Em termos de intervalo de valores, o mínimo é 0 e o máximo é 54, o que representa valores bastante fora dos padrões, por um lado, o zero que representa muito provavelmente valores incorretamente inseridos, por outro lado, o valor máximo é relativamente mais elevado que o limite permitido por lei. Por fim, quanto à moda, o

valor mais recorrente é 11.4, o que, à partida, está dentro dos valores espetáveis, a mediana é de 13.4 e o desvio padrão de 9.504 mg/l.

| Métrica | Valor |
|---|---|
| Mínimo | 0 |
| Máximo | 54 |
| Desvio Padrão | 9.504 |
| Média | 15.859 |
| Moda | 11.4 |
| Mediana | 13.4 |

Tabela 3.3: Análise Estatística do Azoto Total no Efluente Tratado

Posto isto, o próximo passo na exploração deste indicador passa pela análise do gráfico *box plot* e tabela com os respetivos *outliers* que estão representados na Figura 3.2. O intervalo dos quartis localiza-se entre, sensivelmente, 10.5 e 17.5, o que é um intervalo que contém valores ligeiramente fora do intervalo de limites regulamentados. Tal como referido na Tabela 3.3 a mediana encontra-se em 13.4. Ao analisar a tabela e o gráfico, conseguimos facilmente detetar *outliers* abaixo do valor mínimo, que correspondem a valores iguais a zero registados nos meses de Abril e Maio de 2020. Os restantes *outliers*, que são superiores ao máximo, são valores na sua maioria superiores a 30, dos quais, grande parte foram observados entre Setembro e Outubro de 2018 bem como Maio e Julho de 2019.




| Data | Valor |
|---|---|
| 2018-09-07 | 39.0 |
| 2018-09-14 | 54.0 |
| 2018-09-20 | 37.0 |
| 2018-09-28 | 31.0 |
| 2018-10-10 | 32.0 |
| 2018-10-12 | 33.0 |
| 2019-04-18 | 34.0 |
| 2019-05-06 | 36.2 |
| 2019-05-10 | 39.8 |
| 2019-05-17 | 41.0 |
| 2019-07-05 | 28.7 |
| 2019-07-12 | 38.1 |
| 2019-07-29 | 39.3 |
| 2019-08-30 | 29.0 |
| 2020-03-27 | 0.0 |
| 2020-04-24 | 0.0 |
| 2020-05-08 | 0.0 |
| 2020-05-22 | 0.0 |

Figura 3.2: *Box plot* Azoto Total e tabela de *outliers*

De seguida, na Figura 3.3, o histograma representa a média dos valores de Azoto total por ano e

estação do ano. É notório o decréscimo dos valores de Azoto do Verão do ano 2018 para 2019, uma queda de cerca de 60% dos valores da Primavera de 2019 para 2020, bem como do Outono de 2018 para 2019. Em sentido contrário, verifica-se uma subida do valor médio de Azoto no Inverno, subindo de 2018 para 2019, bem como, de 2019 para 2020.

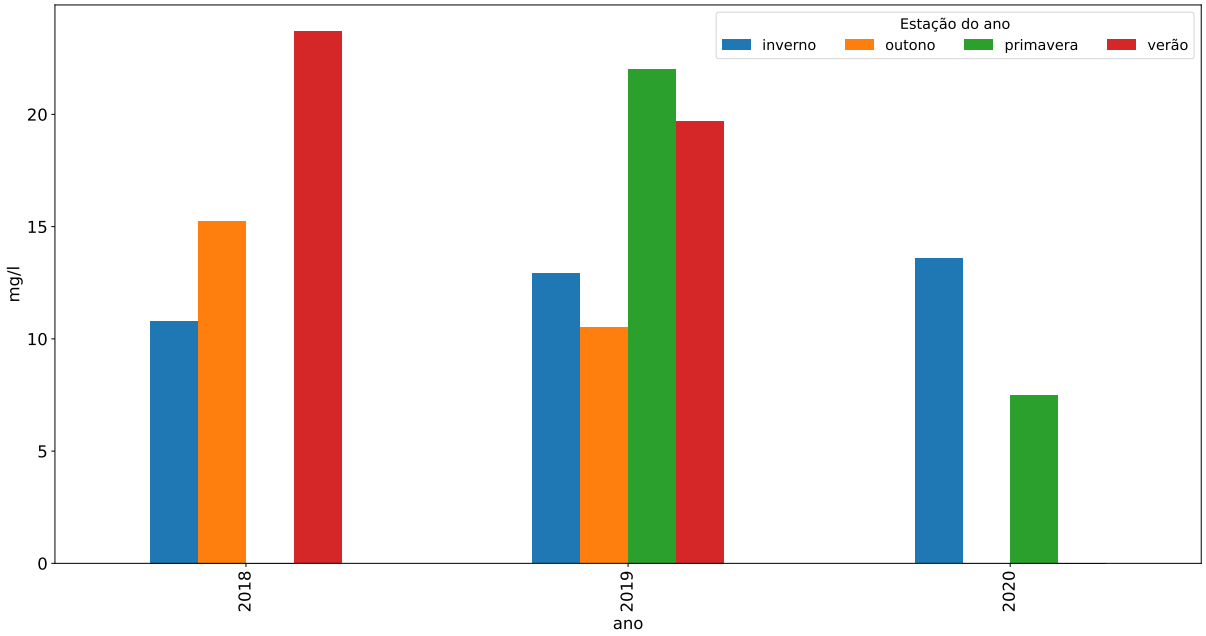



Figura 3.3: Histograma do Azoto Total por ano e estação do ano

Por fim, para perceber se existe relação entre os dados relativos à meteorologia e o indicador Azoto Total, é necessário validar a correlação. De forma a perceber qual o método de correção a utilizar, realizou-se o teste de *Kolmogorov-Smirnov* para validar se os dados seguem uma distribuição Gaussiana. Como o resultado do *p-value* foi inferior a 0.05 ($3.52e{-}9$), verificou-se que o conjunto de dados não segue uma distribuição Gaussiana e assim sendo, construiu-se um *heatmap* através dos valores obtidos pelo coeficiente de correlação *Spearman*. Posto isto, na Figura 3.4 é possível analisar que a chuva apresenta um coeficiente de -0.42 que, embora seja significativo, não é suficientemente forte para justificar a sua utilização. Posto isto, nenhum indicador meteorológico revelou um coeficiente de correlação elevado que justificasse a sua implementação nas etapas seguintes [107].

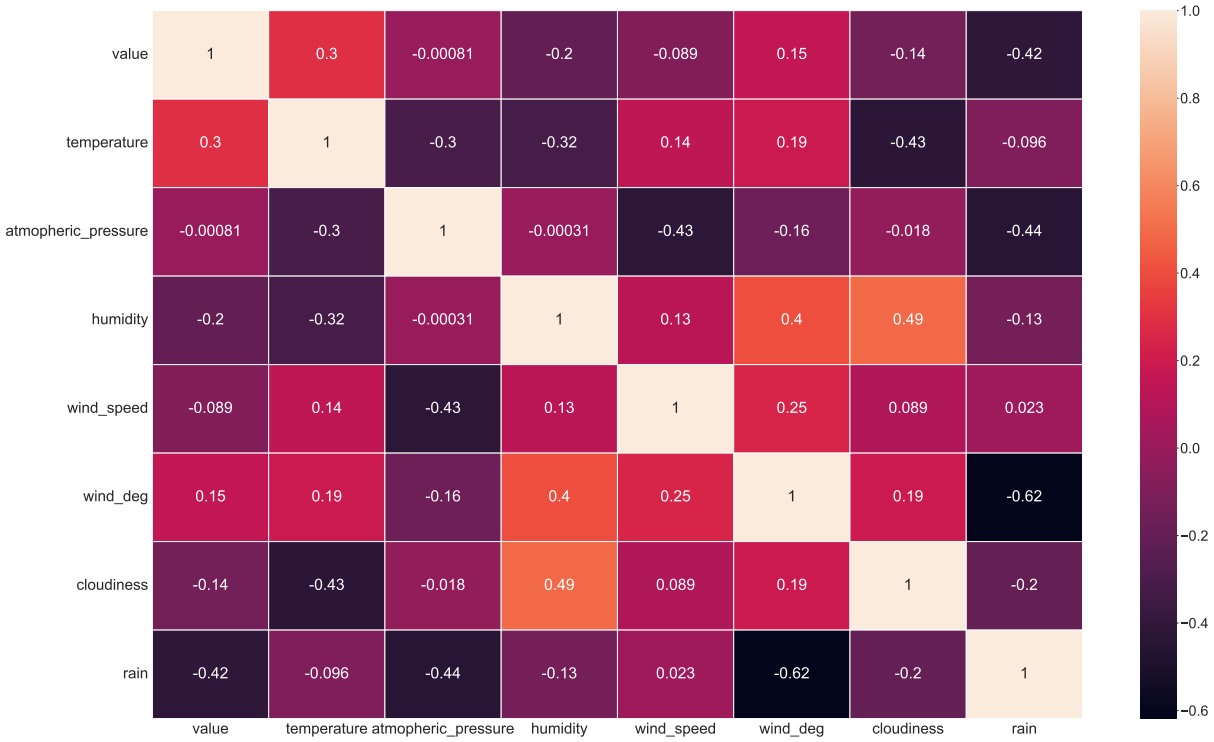



Figura 3.4: *Heatmap* com coeficiente de correlação *Spearman* entre Azoto Total e dados meteorológicos

#### 3.2.1.2 Nitratos

Os Nitratos são um conjunto de compostos que envolvem moléculas de nitrogénio e oxigénio que contém a fórmula $NO_3$. O seu controlo é extremamente importante devido ao perigo que representa para o meio ambiente aquando em concentrações elevadas sendo que esta substância é, imensas vezes, o que limita o crescimento de algas e plantas. Ou seja, valores elevados de Nitratos, poderão afetar a produção e qualidade das culturas sensíveis e causar um crescimento excessivo das plantas, levando a diversos problemas no meio ambiente [108].

Sobre este indicador existem no total 209 registos que estão compreendidos entre o intervalo de tempo de 6 de Agosto de 2018 e 27 de Maio de 2020.

Em média, este indicador contém observações numa periodicidade de 2 em 2 dias, no entanto, existem algumas dias sem registos, tais como:

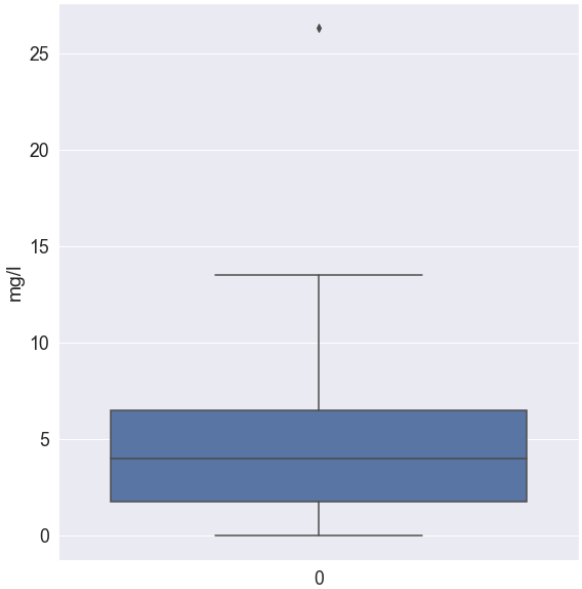
- 2018-08-10
- 2019-01-18
- 2019-12-28
- 2020-03-20
- 2020-05-22

Com foco nos Nitratos no efluente tratado, na Tabela 3.4 está presente a análise estatística e é possível verificar que em média os valores são de 4.284 com um mínimo de 0 e máximo de 26.3. Com uma moda de 0, é possível concluir que este é o número com maior ocorrências o que, por si só, indica que existem várias observações com valores incorretos que necessitam de tratamento. Por fim, a mediana apresenta um valor de 4 e o desvio padrão de 3.199.

| Métrica | Valor |
|---|---|
| Mínimo | 0 |
| Máximo | 26.3 |
| Desvio Padrão | 3.199 |
| Média | 4.284 |
| Moda | 0 |
| Mediana | 4 |

Tabela 3.4: Análise Estatística dos Nitratos no Efluente Tratado

Na Figura 3.5 temos um gráfico *box plot* e uma tabela com o respetivo *outlier*. Ao analisar detalhadamente o gráfico é possível ver que o intervalo do primeiro e do terceiro quartil situam-se entre sensivelmente 2 e 7 respetivamente, e a mediana, tal como mencionado na Tabela 3.4 é de 4. No que diz respeito aos *outliers*, apenas existem um *outlier* localizado a partir do máximo, ou seja, valores superiores a cerca de 14. A partir da Tabela, conseguimos ainda concluir que um *outlier* com o valor de 26.3 foi registado no dia 15 de Abril de 2019.




| Data | Valor |
|---|---|
| 2019-04-15 | 26.3 |

Figura 3.5: *Box plot* Nitratos e tabela de *outliers*

Seguidamente, no histograma da Figura 3.6 é notório o crescimento dos valores de Nitratos de ano para ano por estação de ano. No Verão e Outono de 2018 para 2019 existiu um aumento substancial

do valor de Nitratos bem como no caso do Inverno onde esse aumento prolongou-se ainda para o ano de 2020. Em sentido inverso, na Primavera de 2019 para 2020 registou-se o único decréscimo no valor médio do indicador em questão.

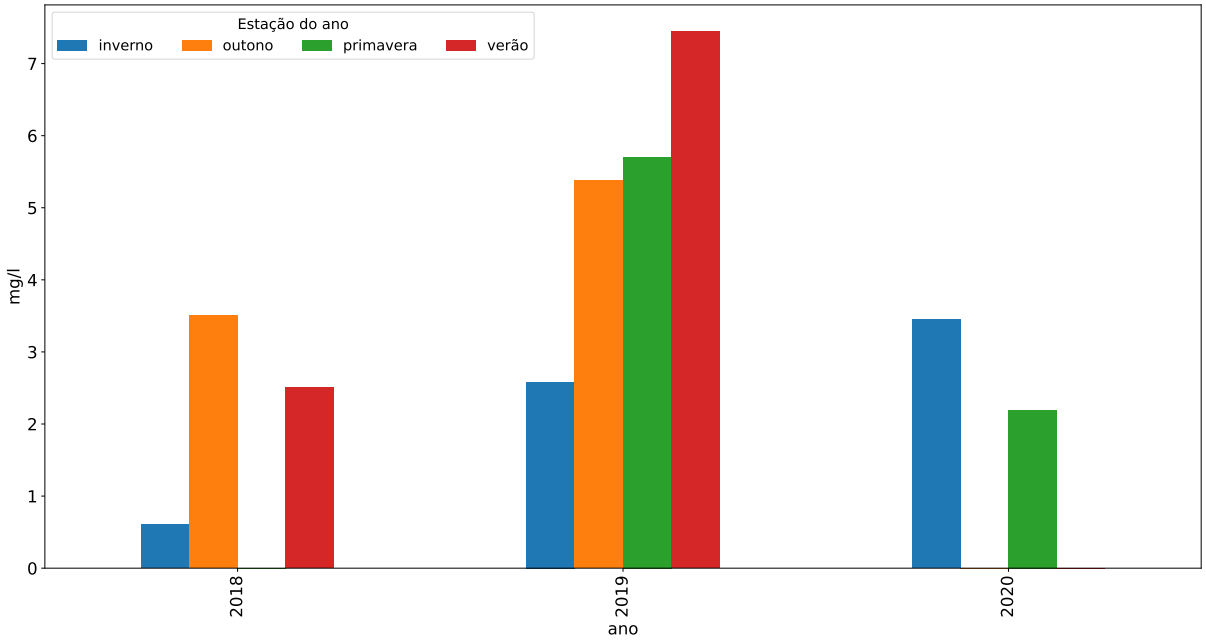



Figura 3.6: Histograma dos Nitratos por ano e estação do ano

Por último, de forma a analisar a existência de dados meteorológicos que possam ter impacto nos valores dos Nitratos, verificou-se a correlação entre os mesmos. Posto isto, para selecionar o método a utilizar na correlação, é necessário validar se o conjunto de dados dos Nitratos seguem uma distribuição Gaussiana. Assim sendo, após executar o teste de *Kolmogorov-Smirnov*, obteve-se um *p-value* de $1.13e{-}11$, o que indica que este conjunto não segue a distribuição previamente mencionada. Com isto, construiu-se um *heatmap* através dos valores obtidos pelo coeficiente de correlação com o método *Spearman* e os resultados estão representados na Figura 3.7. Neste gráfico é possível validar que nenhum indicador meteorológico revelou um coeficiente de correlação elevado o suficiente que justificasse a sua utilização nas etapas seguintes [107].

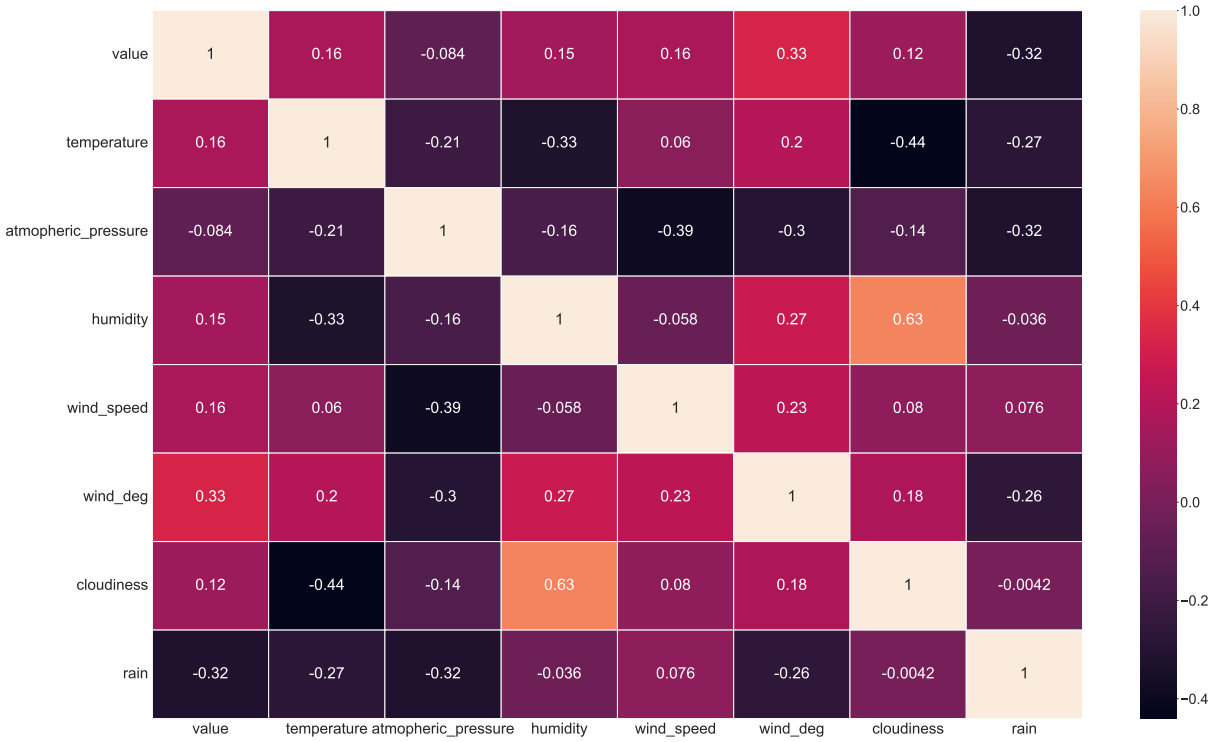



Figura 3.7: *Heatmap* com coeficiente de correlação *Spearman* entre Nitratos e dados meteorológicos

#### 3.2.1.3 pH

O último indicador selecionado foi o pH, que representa a medição de atividade do íon de hidrogénio e, tem por base, uma escala de 0 a 14 onde 7 é considerado neutro, valores menores são considerados ácidos e valores mais elevados considerados alcalinos [109].

Tal como com os outros dois indicadores, o pH carece de tratamento adequado de forma a controlar o seu valor no momento de saída da ETAR, no efluente tratado, para o meio ambiente [110]. A presença de valores de pH demasiado baixos pode indicar a presença de metais pesados o que ameaça a saúde humana, animal e vegetal.

No conjunto de dados, existem no total 273 observações deste indicador que estão compreendidas entre o intervalo de tempo de 8 de Janeiro de 2016 e 20 de Maio de 2020. Nesses registos, o pH contém, em média, registos numa periodicidade semanal, no entanto, existem algumas semanas sem registos, como por exemplo:

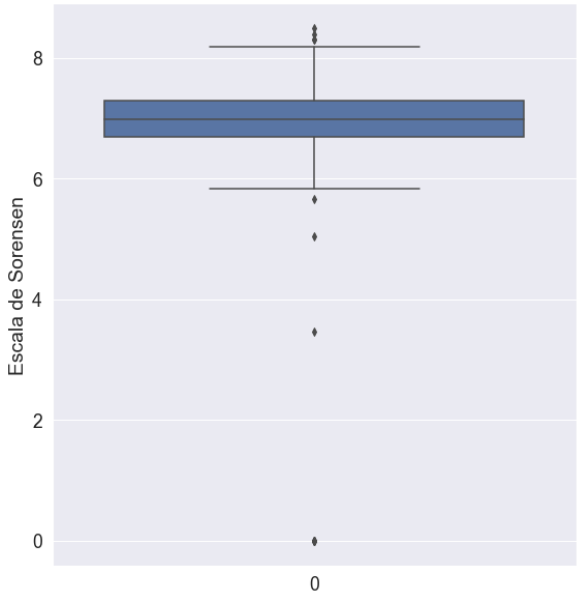
- 2016-11-27
- 2017-01-03
- 2018-10-05
- 2019-03-29
- 2020-03-27

Com foco no pH no efluente tratado, a Tabela 3.5 representa a análise estatística onde é visível que a média de pH é de 6.832 no efluente tratado, o mínimo é 0 e o máximo 8.49 ou seja, apenas o valor mínimo encontra-se fora dos limites de emissão mencionados na Tabela 2.2. Por mim, a moda de 6.9 permite assumir que este é o valor com maior ocorrências, a mediana é de 6.99 e o desvio padrão 1.2.

| Métrica | Valor |
|---|---|
| Mínimo | 0 |
| Máximo | 8.49 |
| Desvio Padrão | 1.2 |
| Média | 6.832 |
| Moda | 6.9 |
| Mediana | 6.99 |

Tabela 3.5: Análise Estatística do pH no Efluente Tratado

No gráfico *box plot* e tabela de *outliers*, representados na Figura 3.8, é visível que o intervalo dos quartis localiza-se entre, aproximadamente, 6.8 e 7.2, o que é um intervalo que contém apenas valores dentro dos valores padrão. Tal como referido na Tabela 3.5 a mediana encontra-se em 6.99 e, ao analisar a tabela e o gráfico, conseguimos facilmente detetar *outliers* abaixo do valor mínimo, que correspondem, na sua maioria, a valores iguais a zero registados em 2016 e localizam-se fora dos limites de emissão aconselháveis. Os restantes *outliers*, que são superiores ao máximo, são valores superiores a 8.1, dos quais, grande parte foram observados no mês de Fevereiro.




| Data | Valor |
|---|---|
| 2016-02-12 | 5.05 |
| 2016-02-19 | 0.0 |
| 2016-03-04 | 0.0 |
| 2016-04-07 | 0.0 |
| 2016-04-08 | 0.0 |
| 2016-08-10 | 0.0 |
| 2016-08-26 | 0.0 |
| 2016-09-20 | 3.455 |
| 2017-03-15 | 5.67 |
| 2017-06-16 | 8.3 |
| 2018-02-16 | 8.4 |
| 2019-02-20 | 8.3 |
| 2020-02-07 | 8.49 |

Figura 3.8: *Box plot* pH e tabela de *outliers*

De seguida, na Figura B.3 está presente um histograma com a média dos valores de pH por ano e estação do ano. Conseguimos destacar que no ano de 2016 o valor médio do pH foi claramente inferior

aos restantes anos. Em relação ao verão, é interessante validar que os valores médios aumentaram constantemente desde 2016 até 2019, verificando a subida mais acentuada na transição de 2016 para 2017. Para além disso, a estação do ano que apresenta uma média de valor de pH mais elevada ao longo dos anos registados, é o outono, podendo este fator eventualmente estar relacionado com esta época do ano tendencialmente apresentar índices de precipitação mais elevados.

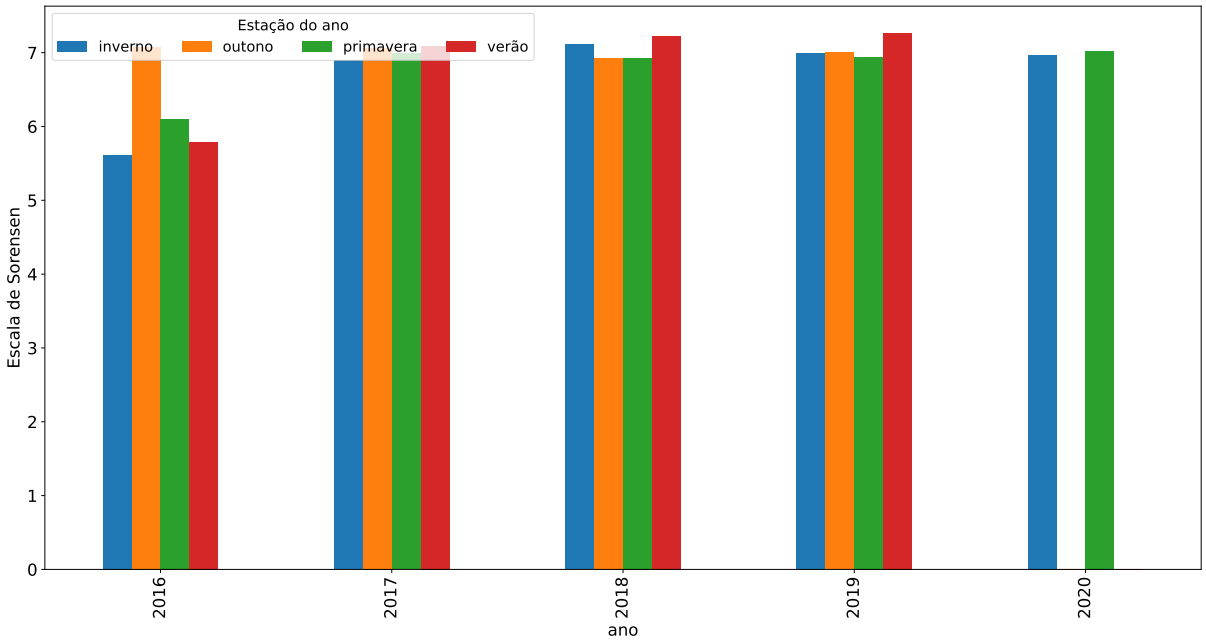



Figura 3.9: Histograma do pH por ano e estação do ano

Por fim, no que diz respeito aos dados meteorológicos, é necessário validar se estes influenciam o valor do pH validando a correlação entre ambos. Assim sendo, é necessário verificar se este conjunto de dados seguem uma distribuição Gaussiana de forma a escolher o método de cálculo de correção mais adequado para o problema. Com o teste *Kolmogorov-Smirnov*, foi possível calcular o *p-value*, e verificar que o mesmo, é inferior a 0.05 ($2.05\mathrm{e}{-27}$), ou seja, o conjunto de dados não segue a distribuição anteriormente referida. Posto isto, foi construido um *heatmap* com os valores obtidos pelo coeficiente de correlação que foi calculado com o método *Spearman* que se encontra ilustrado na Figura 3.10. Nesta Figura 3.10 é possível analisar a maior correlação verificada é com a temperatura mas, mesmo nesse caso, o valor é de apenas 0.2. Por outras palavras, nenhum indicador meteorológico revelou um coeficiente de correlação elevado que justificasse a sua implementação nas etapas seguintes [107].

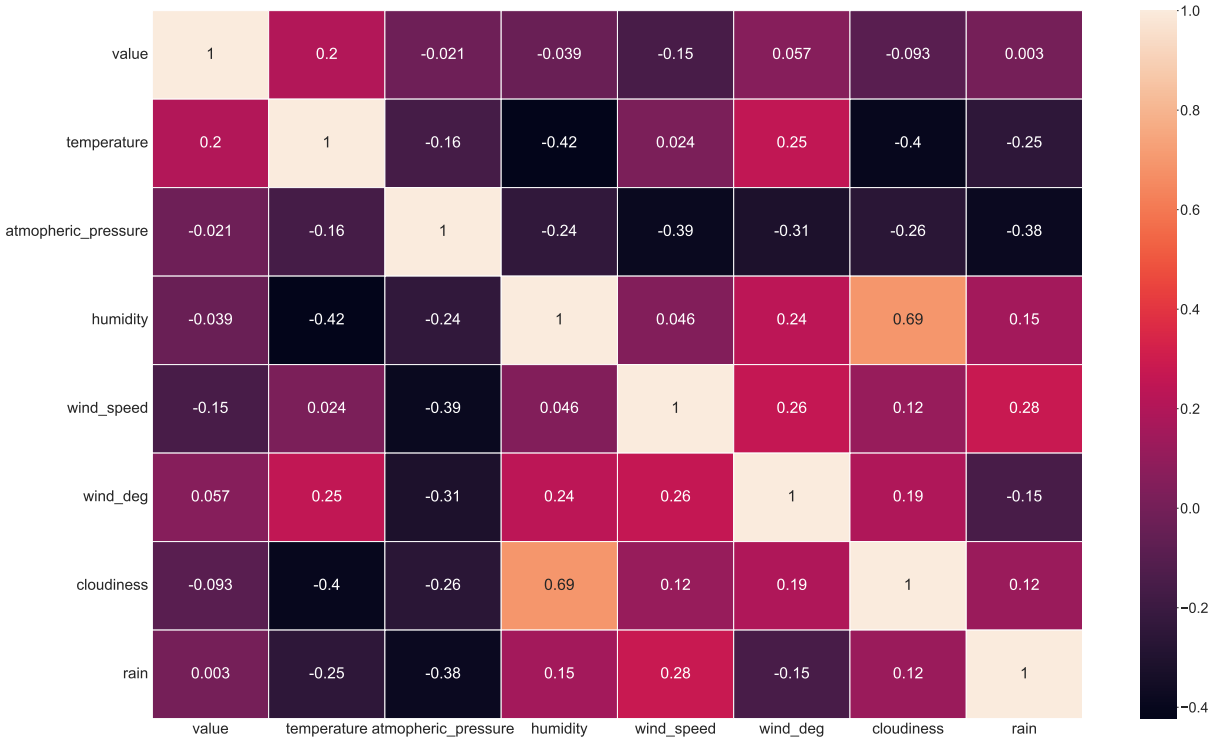



Figura 3.10: *Heatmap* com coeficiente de correlação *Spearman* entre pH e dados meteorológicos

#### 3.2.1.4 Relação entre indicadores do Efluente tratado

Concluída a análise dos três indicadores que serão estudados com mais detalhes nas etapas seguintes, é imperativo nesta fase compreender se existem possíveis correlações entre os indicadores presentes na fase do Efluente Tratado que justifiquem a sua consideração nos próximos passos.

Na Figura 3.11 estão representados os coeficientes de correlação *Spearman* entre os indicadores do Efluente tratado. No que diz respeito ao Azoto Total, conseguimos identificar um coeficiente de correlação de 0.87 com a Amónia o que motiva a sua utilização nas próximas etapas. Por outro lado, tanto com os Nitratos como com o pH não existem quaisquer valores suficientemente elevados de correlação que justifiquem consideração.

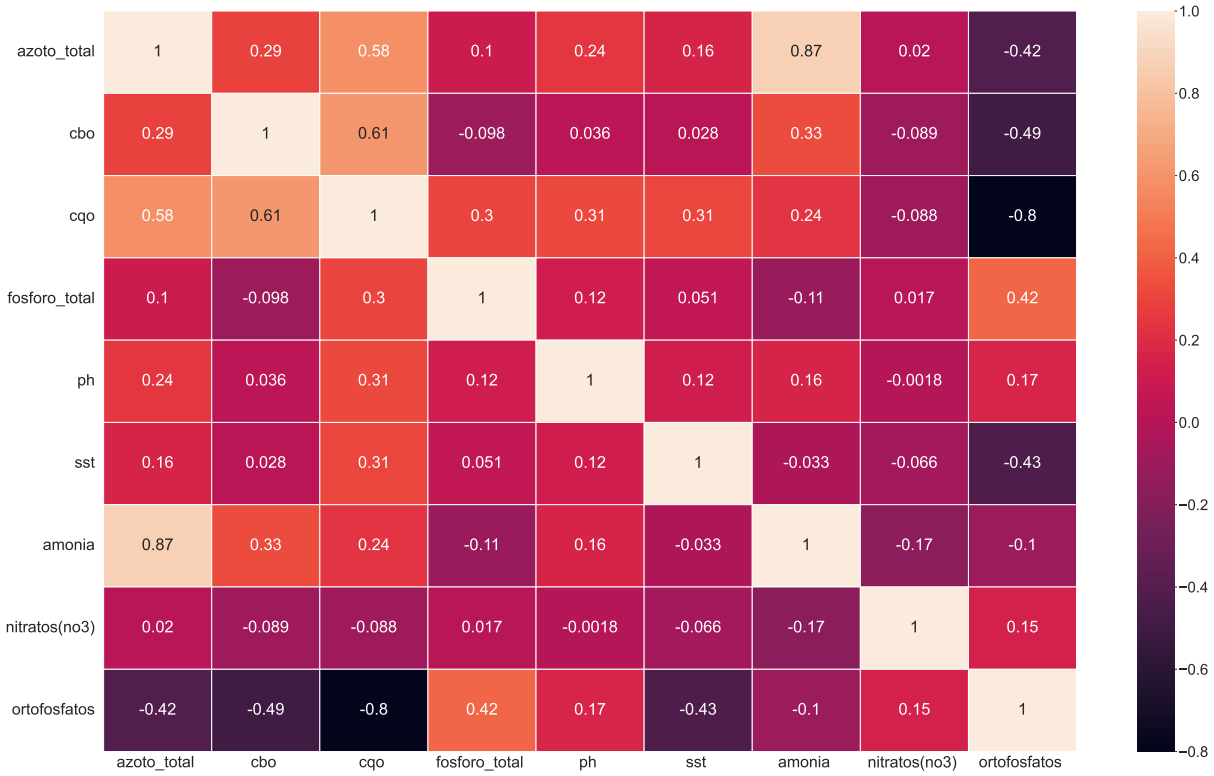



Figura 3.11: *Heatmap* com coeficiente de correlação *Spearman* entre os indicadores do Efluente tratado

### 3.2.2 Dados Meteorológicos

O conjunto de dados Meteorológicos contém registos sobre a cidade da ETAR, Vila Real. Com estes dados foi necessário, numa primeira fase, realizar uma análise estatística de forma a calcular a média, moda, mediana e desvio padrão, bem como, analisar o número de registos, valores mínimos e máximos.

Na Tabela 3.6 é possível verificar a análise estatística previamente referida a todos os indicadores meteorológicos bem como as unidades de medida dos mesmos.

| Indicador | Un. Medida | Registos | Mínimo | Máximo | Desv. Padrão | Média | Moda | Mediana |
|---|---|---|---|---|---|---|---|---|
| Chuva | mm | 450 | 0.11 | 4.062 | 0.519 | 0.638 | 0.12 | 0.498 |
| Direção do Vento | Graus | 1363 | 48.958 | 360 | 53.026 | 240.643 | 259.125 | 252.333 |
| Humidade | % | 1363 | 6.542 | 97.917 | 21.265 | 63.713 | 66.417 | 67.25 |
| Nuvens | % | 1363 | 0 | 100 | 32.394 | 47.106 | 0 | 46.75 |
| Pressão Atmosférica | °C | 1363 | 984.583 | 1038.71 | 6.678 | 1018.803 | 1016.458 | 1018.375 |
| Temperatura | °C | 1363 | -0.099 | 30.373 | 6.141 | 14.247 | 14.856 | 13.729 |
| Velocidade do Vento | m/s | 1363 | 0.755 | 7.765 | 0.809 | 2.035 | 2.375 | 1.878 |

Tabela 3.6: Análise Estatística dos Dados Meteorológicos

Com a análise estatística realizada, é interessante retirar algumas ilações. Existem, no total, 7 indicadores meteorológicos que perfazem cerca de 8628 registos, ou seja, em média, cada indicador, contém cerca de 1232 registos onde, o único indicador com registos inferiores à média, é o indicador da chuva, que contém apenas 450 registos.

O espaço temporal das observações foca-se entre Janeiro de 2016 e Maio de 2020. Quanto ás unidades de medida, existem cinco tipos de unidade, nomeadamente, mm, graus, percentagem, graus *celsius* e metros por segundo. Todas estas observações estão predispostas numa periodicidade de uma em uma hora não sendo detetados, nesta análise inicial, quaisquer valores nulos ou *missing timesteps*. A respeito de valores mínimos e máximos, apenas existem valores negativos no indicador da temperatura sendo que os restantes, à exceção da direção do vento e pressão atmosférica, encontram-se num intervalo de valores de 0 a 100. No que diz respeito à media e desvio padrão, conseguimos verificar que, por exemplo, a temperatura em Vila Real neste espaço temporal foi, em média, de 14.247 graus com um desvio padrão de 6.141 graus.

## 3.3 Manipulação dos Dados

A manipulação e tratamento dos dados é uma tarefa crucial para a futura utilização dos conjuntos de dados nos modelos de ML tradicional e DL nas experiências de forma a atingir os melhores resultados possíveis. Posto isto, nesta secção, são mencionadas as várias etapas realizadas neste processo de manipulação e tratamento.

Assim sendo, nas secções 3.3.1, 3.3.2 e 3.3.3, estão presentes o tratamento efetuado nos indicadores Azoto Total, Nitratos e pH, respetivamente. Para cada um destes indicadores, existiram 5 fases essenciais para a obtenção do conjunto de dados final, nomeadamente, a junção dos dados com outros indicadores com os quais possa existir uma correlação de *spearman* suficientemente relevante. De seguida, aplicou-se uma técnica de *feature engineering*, onde foram gerados novos atributos para os indicadores, de forma a compreender se acrescentariam valor ao conjunto de dados. Em terceiro lugar, devido à baixa periodicidade dos conjuntos de dados inicial, nesta fase, todos os registos foram agrupados por semana para reduzir a percentagem de *missing timesteps*. Posto isto, e ainda sobre *missing timesteps*, as semanas em falta após o agrupamento de dados, foram inseridas nesta fase. Já na última fase, foram tratados os *missing values* dos *timesteps* para os quais não existia valor do indicador em questão.

Por fim, é importante referir que os dados provenientes da ETAR, não continham qualquer informação sobre quais os registos que potencialmente representavam eventos anómalos. Assim sendo, houve a necessidade de realizar a labelização dos registos com a ajuda de especialistas, mais concretamente, com o apoio de investigadores do Centro de Engenharia Biológica.

### 3.3.1 Azoto Total

O primeiro indicador sujeito ao tratamento de dados foi o Azoto Total, onde foi realizada numa primeira fase a junção de dados, seguida de *feature engineering* com um tratamento e agrupamento de dados no final.

#### 3.3.1.1 Junção dos dados

Tal como foi mencionado previamente na Figura 3.11 da exploração dos dados, no Efluente tratado o Azoto Total e a Amónia têm um coeficiente de correlação *spearman* de 0.87 o que corresponde a uma valor bastante elevado e justifica a sua utilização em conjunto com o Azoto Total.

Assim sendo, inicialmente uniu-se os dados da Amónia aos dados do Azoto Total através das datas de observações do indicador em estudo.

#### 3.3.1.2 *Feature engineering*

Nesta fase, foi realizado *feature engineering* com intuito do verificar se existem novos atributos que possam ser relevantes para o problema. Posto isto, a partir da data das observações, foram criadas novas *features*, nomeadamente: dia, dia do ano, dia da semana, semana, mês, ano, estação do ano, trimestre e semestre.

Após a criação destes novos atributos, foi verificado o coeficiente de correlação *spearman* para verificar se alguma seria relevante, como ilustrado na Figura 3.12.

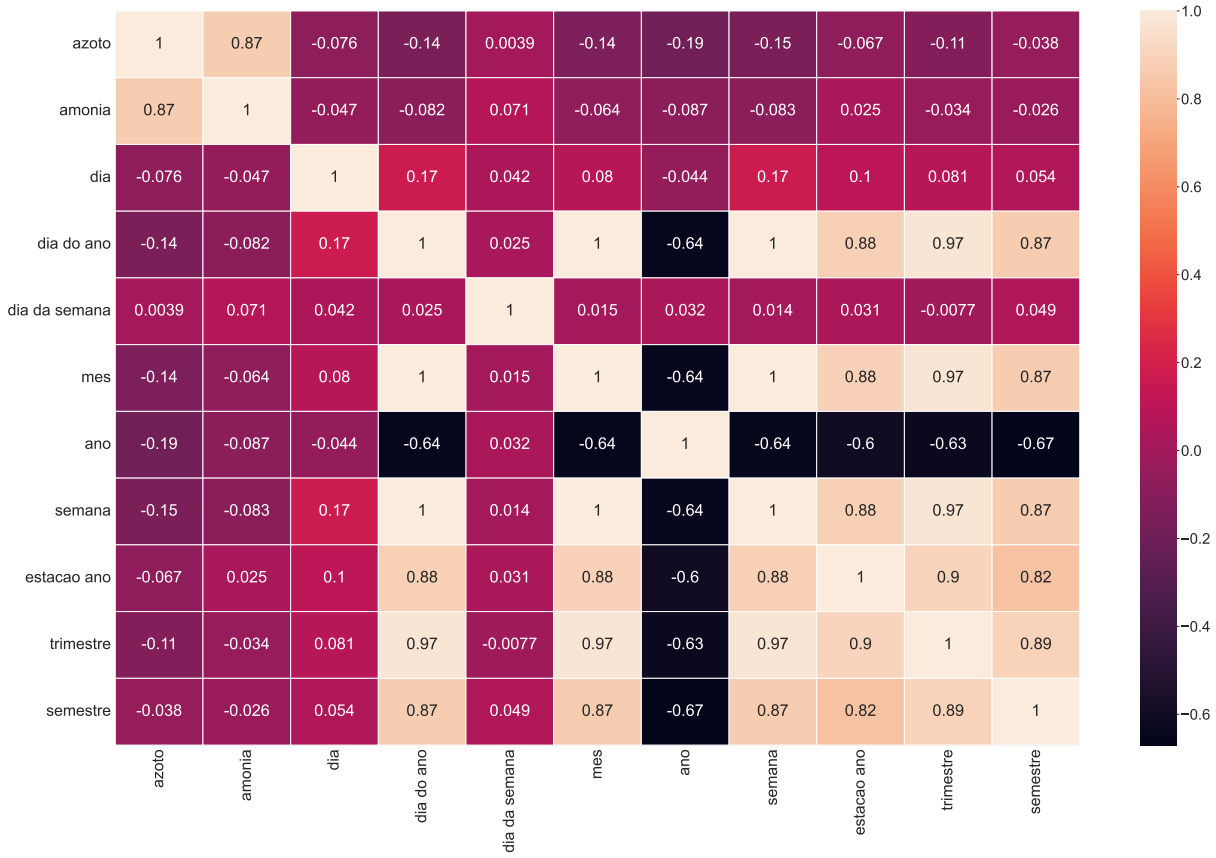



Figura 3.12: *Heatmap* com coeficiente de correlação *Spearman* com novos atributos do Azoto Total

É imediatamente percetível que nenhum atributo gerado pelo método de *feature engineering* obteve um coeficiente suficientemente elevado para justificar a sua utilização nos passos seguintes, sendo que o maior coeficiente de correlação com os indicadores foi de apenas -0.187.

#### 3.3.1.3 Agrupar por semana

De seguida, o próximo passo centrava-se em agrupar as observações por semana, devido à mesma ser a periodicidade mais frequente. Assim sendo, todas as datas foram convertidas no primeiro dia da semana correspondente e, após isso, agruparam-se os dados por data utilizando como método de agregação a média.

#### 3.3.1.4 *Missing timesteps*

Na quarta etapa, realizou-se o estudo e tratamento dos *missing timesteps*. Foi criada uma lista temporária com as datas de todos os primeiros dias da semana existentes entre a primeira e última data existente do conjunto de dados. Após isto, com a comparação das datas presentes nesta lista com as datas do conjunto de dados, verificou-se que existiam 4 *missing timesteps*, nomeadamente:

- 2018-10-01
- 2018-10-15
- 2019-04-29
- 2019-07-22

Com as datas em falta identificadas, foram inseridas estas observações no conjunto de dados com os valores de Azoto Total e Amónia correspondentes a nulo.

#### 3.3.1.5 *Missing values* e conjunto de dados final

Após inserir os *timesteps* em falta é necessário validar os *missing values*. Foram identificados nesta fase, 4 valores em falta no Azoto Total e 71 valores em falta na Amónia. Tendo em consideração que existem no total 96 registos no conjunto de dados, 71 valores em falta corresponde a uma taxa de cerca de 74% de valores nulos. Desta forma, foi decidido remover o indicador de Amónia porque seria arriscado preencher uma percentagem tão elevada de *missing values*.

No caso do Azoto Total, os 4 valores nulos correspondem aos 4 *missing timesteps* que foram introduzidos na etapa anterior. Estes mesmos valores, foram iterativamente preenchidos através da média dos três registos anteriores.

Após estas 5 etapas de tratamento, o conjunto de dados final consiste num *dataset univariate* com 96 registos de Azoto Total que constam entre 30 de Julho de 2018 e 25 de Maio de 2020, com uma periodicidade semanal.

O valor mínimo de Azoto Total é 0 e o valor máximo é 54. A média de valores é de 15.985, a moda é 11.4 e a mediana de 13.35. Por fim, o desvio padrão é de 9.74.

### 3.3.2 Nitratos

Nos três indicadores selecionados, o segundo ao qual foi aplicada a manipulação e tratamento de dados foi os Nitratos onde, inicialmente, concebeu-se o agrupamento dos dados com uma aplicação da técnica de *feature engineering* para gerar novos atributos. Por fim, realizou-se um tratamento aos dados e *timesteps* em falta, resultando no conjunto de dados final.

#### 3.3.2.1 Junção dos dados

Tal como previamente descrito na Figura 3.11 da exploração dos dados, no Efluente tratado os Nitratos não obtiveram quaisquer coeficiente de correlação *spearman* que justificasse a utilização de outro indicador. Assim sendo, nesta primeira fase, o conjunto de dados é *univariate*.

#### 3.3.2.2 *Feature engineering*

Na segunda etapa foi aplicada a técnica de *feature engineering* de forma a criar novos atributos e verificar se os mesmos podem ter alguma relação com os Nitratos. Posto isto, a partir da data de cada um dos registos, foram criados novos atributos, nomeadamente: dia, dia do ano, dia da semana, semana, mês, ano, estação do ano, trimestre e semestre.

Geradas estas *features*, foi verificado o coeficiente de correlação *spearman* para validar se estas novas *features* são relevantes para o problema em questão, como ilustrado na Figura 3.13.


Figura 3.13: *Heatmap* com coeficiente de correlação *Spearman* com novos atributos dos Nitratos

Na Tabela anterior é notório que nenhum dos atributos gerados através da técnica de *feature engineering* possui um coeficiente de correlação *spearman* significativamente alto que permita a sua inclusão nas próximas etapas, sendo que o maior coeficiente de correlação registado entre o indicador dos Nitratos e os indicadores criados foi de 0.236 com a *feature* estacao_do_ ano.

#### 3.3.2.3 Agrupar por semana

Com a técnica de *feature engineering* aplicada, nesta etapa é necessário agrupar as observações. Anteriormente, foi referido que a periodicidade mais frequente nas observações dos Nitratos é de 2 em 2 dias, porém, tendo em conta que entre a data do primeiro registo que é 6 de Agosto de 2018 e a data do último registo que é de 27 de Maio de 2020 existem 661 dias, deveriam existir cerca de 330 (661/2) registos no conjunto de dados de forma a não existirem registos em falta. Como na realidade apenas existem 209 registos, concluiu-se que com um agrupamento de 2 em 2 dias existiram bastantes *missing timesteps* e consequentemente imensos *missing values*. Assim sendo, a opção passou por uma abordagem com periodicidade semanal.

Posto isto, todas as datas foram convertidas no primeiro dia da semana correspondente a essa mesma data e, feito isso, os dados foram agrupados por semana utilizando como recurso de agregação a média dos valores do indicador Nitratos.

#### 3.3.2.4 *Missing timesteps*

Nesta fase, realizou-se uma análise dos *missing timesteps*. Inicialmente, foi criada uma lista temporária com todos os primeiros dias da semana que constam entre a primeira e última data do conjunto de dados dos Nitratos. Feito isso, através da comparação das datas presentes nesta lista com as datas das observações, verificou-se que nos 209 registos iniciais existiam não existiam quaisquer *missing timesteps*

#### 3.3.2.5 *Missing values* e conjunto de dados final

Concluída a etapa de análise dos *timesteps* em falta, o quinto passo consistiu na análise dos *missing values*. Após verificar os valores de Nitratos de todas as observações, concluiu-se que, tal como na etapa anterior, não existem quaisquer *missing values* no conjunto de dados em análise.

Concluídas todas as fases de tratamento, o conjunto de dados resultante contém 95 observações de Nitrato que estão inseridas entre 6 de Agosto de 2018 e 25 de Maio de 2020, com uma periodicidade semanal.

O valor mínimo de Nitratos é 0.477 e o valor máximo é de 13.945. A média de valores é de 4.2, a moda é 0.5 e a mediana de 3.893. Por fim, o desvio padrão é de 2.723.

### 3.3.3 pH

O último indicador ao qual o seu conjunto de dados foi tratado e manipulado é o pH. O conjunto de dados final, que foi posteriormente utilizado nas experiências com os modelos de ML, resultou do *feature engineering* dos dados agrupados, com um tratamento dos valores e *timesteps* em falta no *dataset* inicial.

#### 3.3.3.1 Junção dos dados

Tal como foi mencionado previamente na Figura 3.11 da exploração dos dados, no Efluente tratado o pH não demonstrou ter quaisquer coeficiente de correlação *spearman* suficientemente elevado com outro indicador o que, resultou num conjunto de dados *univariate* nesta fase.

#### 3.3.3.2 *Feature engineering*

Em segundo lugar, foi aplicada a técnica de *feature engineering* de forma a verificar se existem novos atributos que possam ter impacto no pH. Posto isto, a partir da data das observações, foram criadas novas *features*, nomeadamente: dia, dia do ano, dia da semana, semana, mês, ano, estação do ano, trimestre e semestre.

Após gerar estes novos atributos, foi verificado o coeficiente de correlação *spearman*, que está representado na Figura 3.14, para analisar a relevância destas novas *features*.


Figura 3.14: *Heatmap* com coeficiente de correlação *Spearman* com novos atributos do pH

Na Tabela previamente mencionada, é visível que nenhuma *feature* gerada através da técnica de *feature engineering* obteve um coeficiente de correlação *spearman* elevado ao ponto de ser plausível a sua consideração nos passos seguintes, sendo que o maior coeficiente de correlação entre o pH e os indicadores criados foi de apenas 0.143 (semestre).

#### 3.3.3.3 Agrupar por semana

Concluída a etapa de *feature engineering*, o próximo passo focava-se em agrupar as observações. Tendo em conta que a periodicidade semanal era a periodicidade mais frequente, todas as datas foram convertidas no primeiro dia da semana correspondente a essa mesma data e, feito isso, os dados foram agrupados por semana utilizando como recurso de agregação a média dos valores de pH.

#### 3.3.3.4 *Missing timesteps*

Nesta etapa, foi efetuada uma análise e tratamento dos *missing timesteps*. Primeiramente, foi criada uma lista temporária com todos os primeiros dias da semana que constam entre a primeira e última data do conjunto de dados do pH. Após isto, através da comparação das datas presentes nesta lista com as datas do conjunto de dados, verificou-se que nos 198 registos existiam, no total, 31 *missing timesteps*, como por exemplo:

- 2016-09-12
- 2016-11-28
- 2017-05-29
- 2018-08-06
- 2019-03-25
- 2020-05-11

Com as datas correspondentes aos *missing timesteps* definidas, as mesmas foram inseridas no conjunto de dados existente com o valor de pH definido como nulo. Posto isto, o número de observações passou de 198 para 229.

#### 3.3.3.5 *Missing values* e conjunto de dados final

Concluída a inserção dos *timesteps* em falta, o quinto passo consistiu na análise e tratamento dos *missing values*. Foram identificados nesta fase, 31 valores de pH em falta que correspondem aos 31 *missing timesteps* identificados e inseridos na etapa anterior. De forma a preencher estes valores nulos, foi utilizado o valor de pH médio dos três registos anteriores à observação em questão.

Concluída a fase de tratamento, o conjunto de dados resultante contém 229 observações de pH que se encontram compreendidas entre 4 de Janeiro de 2016 até 18 de Maio de 2020, com uma periodicidade semanal.

O valor mínimo de pH é 0 e o valor máximo é de 8.49. A média de valores é de 6.85, a moda é 6.9 e a mediana de 7.0. Por fim, o desvio padrão é de 1.05.

Capítulo 4

# Experiências

Este capítulo descreve as experiências que foram consideradas no âmbito desta dissertação. Como previamente referido, foram selecionados três modelos para executar as experiências, nomeadamente iF, OCSVM e LSTM-AE. Estas escolhas foram realizadas com base na simplicidade, adaptabilidade e robustez dos modelos, formando desta forma uma base de comparação e discussão interessante. A secção 4.1 apresenta os três *thresholds* que foram aplicados sobre os modelos de LSTM-AE, de forma a poder identificar as observações como normais ou anómalas, dos quais, dois *thresholds* utilizam um valor constante para todas as observações, nomeadamente *threshold* 1 e 2 (secções 4.1.1 e 4.1.2), e um *threshold* com um valor dinâmico (*threshold* 3, secção 4.1.3).

De seguida, na secção 4.2 foram retratadas as métricas de avaliação utilizadas para avaliar os modelos que foram posteriormente concebidos, mais concretamente, a Area Under The Curve - Receiver Operating Characteristics (AUC-ROC) (secção 4.2.1) e a *f1-score* (secção 4.2.2). Posto isto, a terceira secção, secção 4.3, compreende todos os parâmetros que foram considerados para cada um dos modelos concebidos e o respetivo conjunto de valores considerados na fase de otimização dos hiperparâmetros para os modelos da iF, OCSVM e LSTM-AE. Por fim, a secção 4.4, contém todas as bibliotecas com as respetivas versões que foram utilizadas no decorrer desta dissertação para alcançar os resultados.

## 4.1 LSTM-AE *Thresholds*

Como foi referido em 2.2.1.3, a função da LSTM-AE centra-se em reconstruir o *input* com o menor erro possível. Quando o erro da reconstrução é significativo, podemos estar perante uma anomalia. Posto isto, é necessário definir o valor limite de erro a partir do qual todas as observações reconstruídas com um erro superior, serão consideradas anómalas. Para esse efeito, existem na literatura algumas formulas matemáticas capazes de calcular *thresholds* [111].

O valor do *threshold* determina a sensibilidade com que o modelo reagirá a condições anormais, ou

seja, enquanto que em alguns contextos um pequeno desvio dos padrões normais pode ter um grande impacto e deve ser considerado como um evento anómalo, por outro lado, uma sensibilidade demasiado alta em ambiente de produção pode ser perturbador devido à grande quantidade de alertas devido a falsas anomalias [112]. É então, para cada caso de uso, necessária uma pré avaliação das consequências que uma má classificação podem ter.

Com isto, foram selecionadas e adaptadas três fórmulas de calcular *thresholds*, denominados de *threshold* 1, 2 e 3 (Figura 4.1).

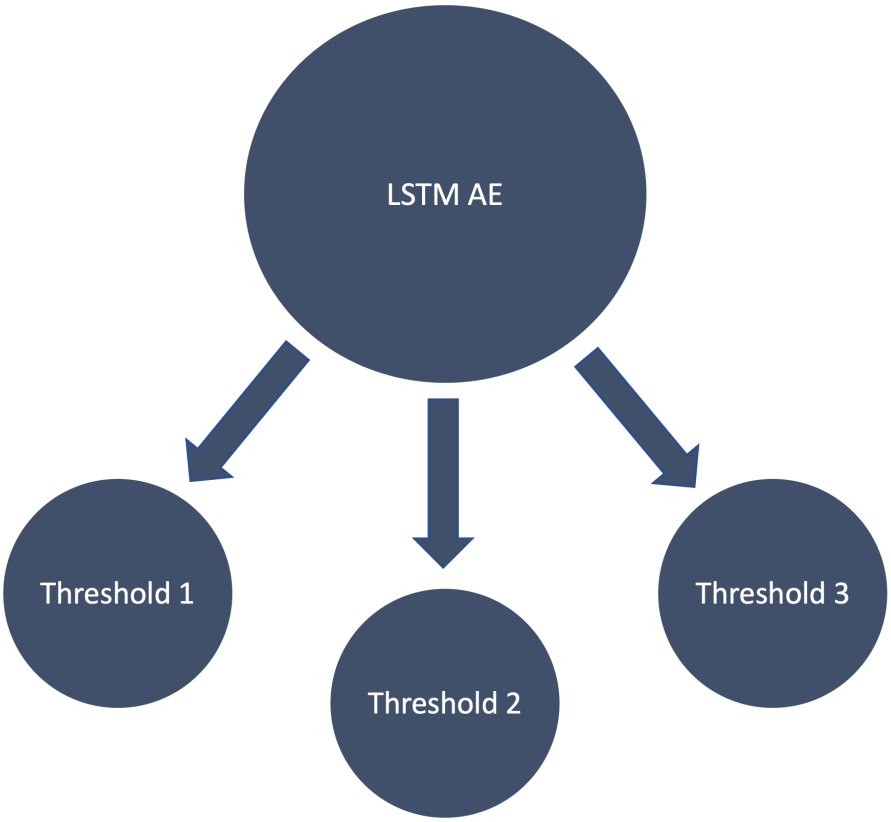



Figura 4.1: *Thresholds* considerados

### 4.1.1 Threshold 1

O primeiro *threshold* considerado consiste no valor máximo da Mean Absolute Error (MAE), ou seja, durante a fase de treino, para cada observação, é calculado a média do valor absoluto do erro de reconstrução e, após esta fase, é definido como *threshold* o maior erro obtido durante o treino. Posto isto, caso na previsão algum valor obtenha um erro de reconstrução superior ao erro máximo obtido na fase treino, esse registo será identificado como anómalo.

```
1  train_mae_loss = np.mean(np.abs(x_train_pred - x_train), axis=1)
2  threshold1 = np.max(train_mae_loss)
```

Listagem 4.1: Cálculo do *threshold* 1

### 4.1.2 Threshold 2

Em segundo lugar, o *threshold* 2 já não utiliza valores da fase de treino como no caso anterior. Este *threshold* tem por base a MAE dos dados de teste, isto é, durante a fase de teste, são calculados os valores médios absolutos da diferença entre o valor previsto e o valor real da observação. Posto isto, com este valor, é definido um valor de *std_coef* que corresponde ao coeficiente utilizado com o desvio padrão, e é calculada a média destes erros e é somado a multiplicação do *std_coef* com o desvio de padrão desses mesmos erros. O valor resultante desta operação representa o *threshold* 2.

```
1  test_mae_loss = np.mean(np.abs(x_test_pred - x_test), axis=1)
2  threshold1 = np.mean(test_mae_loss) + std_coef * np.std(test_mae_loss)
```

Listagem 4.2: Cálculo do *threshold* 2

### 4.1.3 Threshold 3

Por fim, o último *threshold*, é o mais complexo dos três. Tanto o valor do *threshold* 1 como o do *threshold* 2, são valores estáticos, ou seja, o valor definido é utilizado para definir todas as observações de teste como anómalas ou não anómalas. No caso do *threshold* 3, o valor varia, ou seja, para uma determinada observação de teste *S1*, será utilizado um valor de *threshold X*, porém, para uma observação de teste *S2*, será utilizado um valor de *threshold Y*.

Tal como no *threshold* 2, é considerado o valor de MAE resultante da previsão dos dados de teste e, calculados esses valores, é definido um valor de *window* correspondente à janela temporal que será considerada para calcular o *threshold*. Com isto, são gerados, para cada observação, a janela de valores que serão considerados para calcular o *threshold* no passo seguinte. Para terminar, com a janela de valores definida para todas as observações de teste, é calculada a média de cada uma delas e somado a multiplicação do *std_coef* (valor definido que corresponde ao coeficiente utilizado com o desvio padrão) com o desvio padrão de cada uma desta janela de valores.

```
1  test_mae_loss = np.mean(np.abs(x_test_pred - x_test), axis=1)
2  test_pred_erros_windowed = pd.Series(test_mae_loss).rolling(window=window, min_periods=1)
3  test_dynamic_threshold = test_pred_erros_windowed.mean() + std_coef *
       ↪ test_pred_erros_windowed.std()
```

Listagem 4.3: Cálculo do *threshold* 3

## 4.2 Métricas de Avaliação

Concluída a classificação das observações como anómalas ou não anómalas por parte dos modelos, com recurso aos *thresholds* mencionados no passo anterior no caso da LSTM-AE, de forma a avaliar e comparar a *performance* dos modelos de ML concebidos, existem diversas métricas de avaliação, das quais, foram escolhidas as duas que são mais indicadas para problemas de deteção de anomalias.

Os modelos de deteção de anomalias identificam observações como anómalas ou não anómalas (classificação binária), existindo apenas quatro possibilidades para cada observação prevista, nomeadamente, True Positive (TP), True Negative (TN), False Positive (FP) e False Negative (FN) [113].

- TP é quando a observação inicial está identificada como anómala e é prevista corretamente como anómala.
- TN é quando a observação inicial está identificada como não anómala e é prevista corretamente como não anómala.
- FP é quando a observação inicial está identificada como não anómala e é prevista incorretamente como anómala.
- FN é quando a observação inicial está identificada como anómala e é prevista incorretamente como não anómala.

Com base nestes resultados, a matriz de confusão é construída representando um sumário de todos os resultados dos modelos. Todas as previsões corretas e incorretas são contabilizadas e distribuídas por cada classe, como ilustrado na Figura 4.2 [113].

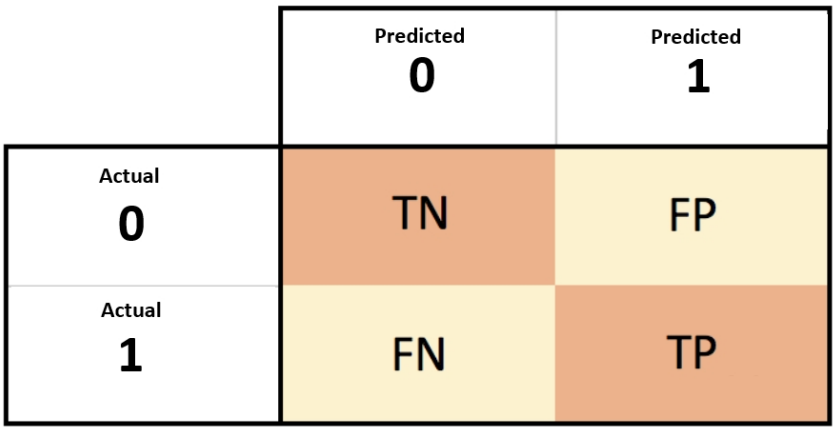



Figura 4.2: Matriz de confusão[1]

### 4.2.1 AUC-ROC

A primeira métrica a ser utilizada é a AUC-ROC, esta é uma medida de desempenho para problemas de classificação. ROC é a curva probabilística e AUC representa a capacidade de separabilidade. Ou seja, esta métrica indica a capacidade de um determinado modelo distinguir classes [114].

Um valor de AUC mais elevado (perto de 1), significa uma melhor capacidade do modelo distinguir as observações anómalas das normais, um valor. Por outro lado, um valor de AUC perto de 0 indica que o modelo está a identificar as classes de forma inversa, ou seja, uma observação anómala é identificada

[1] https://towardsdatascience.com/demystifying-confusion-matrix-29f3037b0cfa

como normal, e vice versa. Por sua vez, um valor de 0.5, denota que o modelo não tem quaisquer capacidade de distinção entre classes.

### 4.2.2 *F1-Score*

A outra métrica escolhida é o *F-Score* que representa a média ponderada da $Precision$ e da $Recall$, ou seja, esta métrica tem em consideração tanto os FP como os FN [115]. Esta função é útil quando o problema em questão tem uma distribuição de classes não balanceada. Posto isto, a *F-Score* consiste em:

$$F - Score = \frac{(1 + \beta^2) * Recall * Precision}{\beta^2 * Recall + Precision}$$

Onde o $\beta$ presente no coeficiente corresponde ao balanço entre $Recall$ e $Precision$, com valores mais elevados a favorecer $Recall$. Neste caso, *F-Score* é utilizada com $\beta = 1$, também conhecido como *F1-Score*:

$$F1 - Score = \frac{2 * Recall * Precision}{Recall + Precision}$$

Um valor de *F1-Score* é considerado perfeito quando o seu valor se aproxima de 1, o que significa que a presença de FP e FN é baixa. Por outro lado, se o seu valor for próximo de 0, o valor é considerado uma falha na tarefa de deteção de eventos anómalos [116].

## 4.3 Modelação e otimização dos hiperparâmentros

Como referido no Capítulo 2.2.1, foram selecionados três modelos do paradigma de deteção de anomalias, nomeadamente, iF, OCSVM e LSTM-AE, e para cada indicador, estes modelos selecionados foram construídos, treinados e otimizados com recurso à linguagem de programação *python*.

Na construção destes modelos foram utilizados os três conjuntos de dados resultantes do tratamento referido no Capítulo 3.3 e, os dados de treino, validação e teste foram divididos utilizando a técnica de *k-fold cross validation* e *time series split*.

### 4.3.1 iF

Na conceção dos modelos iF, foram apenas considerados quatro parâmetros devido aos restantes não acrescentarem ou impactarem nos resultados obtidos de forma direta. Assim sendo, os parâmetros considerados foram:

- *n_estimators*: Número de estimadores, ou seja, número de árvores que serão consideradas.
- *max_samples*: Número de registos utilizados do *input* para treinar cada um dos *estimators*.

- *contamination*: A quantidade de contaminação do conjunto de dados, por exemplo, a proporção de *outliers* no *dataset*.
- *bootstrap*: Se estiver definido como *true*, as árvores são treinadas em sub conjuntos aleatórios dos dados de treino com substituição. Caso o valor esteja definido como *false*, a amostragem é realizada sem substituição.

Para cada um destes hiperparâmetros foram considerados vários possíveis valores. Tais valores estão representados na Tabela 4.1 onde é possível verificar que, no caso do *bootstrap*, apenas foram considerados dois valores, true ou *false*. Nos restantes hiperparâmetros, foram considerados intervalos de valores numéricos, nomeadamente, no caso do *n_estimators*, por exemplo, foram testados valores de 100 a 450, com intervalos de 50.

| Hiperparâmetro | Valores considerados |
|---|---|
| *n_estimators* | (100, 450, 50) |
| *max_samples* | (40, 200, 20) |
| *contamination* | (0.02, 0.20, 0.02) |
| *bootstrap* | [True, False] |

Tabela 4.1: Valores considerados para os hiperparâmetros dos modelos iF

### 4.3.2 OCSVM

No caso dos modelos OCSVM, os parâmetros considerados para a otimização foram baseados na literatura, mais propriamente [117]:

- *kernel*: Tipo de *kernel* utilizado no algoritmo.
- *gamma*: Coeficiente do *kernel*. Apenas é utilizado com os *kernels rbf*, *poly* e *sigmoid*.
- *nu*: Valor que controla os erros dos treinos, bem como o número de *Support Vectors*.

Na Tabela 4.2 estão descritos os valores considerados para cada um dos hiperparâmetros dos modelos do OCSVM. No caso do *kernel*, foram considerados quatro alternativas, no entanto, no caso dos hiperparâmetros *gamma* e *nu*, foram avaliados intervalos de valores decimais, mais concretamente, no caso do *gamma*, foram considerados todos os valores entre 0.002 e 0.2, com um intervalo de 0.002.

| Hiperparâmetro | Valores considerados |
|---|---|
| *kernel* | [*linear*, *poly*, *rbf*, *sigmoid*] |
| *gamma* | (0.002, 0.2, 0.002) |
| *nu* | (0.02, 0.20, 0.02) |

Tabela 4.2: Valores considerados para os hiperparâmetros dos modelos OCSVM

### 4.3.3 LSTM-AE

Por fim, nos modelos construídos através algoritmo LSTM-AE foram considerados seis hiperparâmetros de entre os disponíveis. Esta escolha foi realizada com base no que é normalmente recomendado na literatura [118]:

- camadas: Número de camadas LSTM do modelo.
- neurónios: Número de neurónios por camada LSTM.
- função de ativação: Função de ativação a utilizar em cada camada LSTM.
- *dropout_rate*: Valor decimal entre 0 e 1 que representa a fração de valores a descartar.
- *timesteps*: Número de observações que são passadas ao modelo.
- *epochs*: Número de épocas que são utilizadas no modelo. Os valores considerados para este hiperparâmetro advém da análise das *learning curves* de forma a prevenir o *overfitting*.
- *batch_size*: Número pelo qual o número de observações é divisível. Define após quantas amostras os pesos da LSTM serão atualizados.

Os valores que foram considerados para cada um dos hiperparâmetros do modelo LSTM-AE estão descritos na tabela 4.3, onde é observável que no caso das *epochs*, foram testados vários possíveis valores, mais concretamente, todos os valores entre 100 e 400 com um intervalo de 100. Por outro lado, os restantes hiperparâmetros, foram testados com intervalos de valores mais curtos, tais como, 32, 64 e 128 no caso dos neurónios.

| Hiperparâmetro | Valores considerados |
|---|---|
| camadas | [1, 2, 3] |
| neurónios | [32, 64, 128] |
| função de ativação | [*relu*, *tanh*] |
| *dropout_rate* | [0.0, 0.1, 0.2] |
| *timesteps* | [1] |
| *epochs* | (100, 400, 50) |
| *batch_size* | [10, 20, 30] |

Tabela 4.3: Valores considerados para os hiperparâmetros dos modelos LSTM-AE

## 4.4 Tecnologias utilizadas

Para o desenvolvimento de praticamente toda a dissertação, foi utilizada a linguagem de programação *Python* (versão 3.7.13) com recurso a diversas bibliotecas, tais como, *Numpy*, *Pandas*, *Matplotlib*

e *Scikit-learn*. No que diz respeito à análise e tratamento dos dados brutos, isto foi conseguido com recurso ao *software Jupyter Notebook* sendo que, a conceção dos modelos foi essencialmente com ao *software Spyder*. Por fim, no que diz respeito ao processo de otimização dos hiperparâmetros, que por si só é uma tarefa exigente em termos de recursos, foi alcançada mais facilmente e eficazmente com o serviço *cloud* da *Google* denominado de *Google Colaboratory* onde existe a possibilidade de utilizar recursos de *hardware* com bastante capacidade de processamento [119].

Capítulo 5

# Resultados e discussão

Neste capítulo, estão presentes os resultados obtidos através das experiências bem como a discussão dos mesmos. De forma a obter os melhores resultados, centenas de experiências foram executadas com o intuito de avaliar vários modelos candidatos. Esta avaliação foi realizada com base nas duas métricas de avaliação referiadas na secção 4.2, nomeadamente, AUC-ROC e *f1-score*.

Posto isto, nas secções 5.1, 5.2 e 5.3 estão presentes os resultados e discussão dos indicadores Azoto Total, Nitratos e pH, respetivamente. Para cada um destes indicadores, são apresentados os resultados da iF, OCSVM, bem como os resultados da aplicação dos três *thresholds* referenciados na secção 4.1 ao modelo da LSTM-AE. Por fim, para cada um dos indicadores, é realizada uma análise comparativa, onde são comparados os resultados de todos os modelos, bem como uma comparação mais detalhada entre o melhor modelo de ML tradicional com o melhor modelo de DL.

## 5.1 Azoto Total

O primeiro indicador a ser utilizado foi o Azoto Total. No caso do Azoto Total, após a identificação dos registos anómalos com a ajuda de especialistas, verificou-se que existiam, à priori, uma grande percentagem de anomalias nos dados que foram utilizados para conceber e consequentemente avaliar os modelos classificativos. Este facto por si só, invalida a própria definição de anomalia que foi mencionada na secção 2.2 onde é descrito que um evento, para ser considerado anómalo, para além de ser significativamente diferente dos restantes, a sua existência, tem que ser invulgar. Posto isto, alguns dos resultados podem ser diretamente afetados por este fator provocando *performances* menos satisfatórias para o indicador em questão.

Assim sendo, nas secções seguintes, estão presentes os resultados dos modelos de iF, OCSVM e LSTM-AE concebidos, bem como uma análise comparativa dos resultados obtidos e das anomalias detetadas.

### 5.1.1 iF

Os primeiros resultados registados para o Azoto Total foram obtidos através da conceção e avaliação de modelos de classificação de iF com a AUC-ROC e *f1-score*. Estes mesmos resultados estão ilustrados na Tabela 5.1.

| # | a. | b. | c. | d. | e. | f. | g. |
|---|---|---|---|---|---|---|---|
| 8 | 100 | 40 | 0.1 | True | 0.975 | 0.941 | 0.226 |
| 30 | 100 | 60 | 0.14 | True | 0.950 | 0.889 | 0.234 |
| 103 | 150 | 80 | 0.14 | False | 0.925 | 0.842 | 0.338 |

Tabela 5.1: Melhores resultados obtidos nos modelos iF. As letras representam o seguinte: a. *n_estimators*; b. *max_samples*; c. *contamination*; d. *bootstrap*; e. AUC-ROC; f. *f1-score*; g. tempo(segundos).

É possível verificar na Tabela 5.1 que o melhor modelo candidato conseguiu obter uma AUC-ROC de 0.975 e *f1-score* de 0.941. Neste *top-3*, é possível confirmar que nenhum atributo foi homogéneo nos três modelos, no entanto, tanto o *n_estimators* com valor 100 como o *bootstrap* a *True* foram os mesmos nos dois melhores modelos. Em sentido inverso, o número de *max_samples* foi diferente nos três modelos concebidos.

Por fim, os resultados são possíveis de uma outra conclusão, os modelos com maior complexidade, ou seja, maior número de *n_estimators* e *max_samples*, embora demorassem mais tempo na fase de treino, acabaram por obter resultados menos bons, enquanto que o modelo mais simples de entre os três, obteve os melhores resultados.

### 5.1.2 OCSVM

O segundo conjunto de resultados foi obtido através de modelos concebidos a partir do algoritmo OCSVM. Os três melhores resultados relativos ao Azoto Total resultantes do processo de otimização de hiperparâmetros estão descritos na Tabela 5.2.

| # | a. | b. | c. | d. | e. | f. |
|---|---|---|---|---|---|---|
| 9 | rbf | 0.004 | 0.02 | 0.971 | 0.928 | 0.002 |
| 307 | rbf | 0.072 | 0.04 | 0.965 | 0.911 | 0.002 |
| 136 | rbf | 0.032 | 0.04 | 0.958 | 0.896 | 0.002 |

Tabela 5.2: Melhores resultados obtidos nos modelos OCSVM. As letras representam o seguinte: a. *kernel*; b. *gamma*; c. *nu*; d. AUC-ROC; e. *f1-score*; f. tempo(segundos).

Tendo por referência os resultados representados na Tabela 5.2, conseguimos constatar que o melhor modelo obteve 0.971 de AUC-ROC e 0.928 de *f1-score*. Nestes três modelos, é notória a a utilização do

*kernel rfb*. Em sentido contrário, o valor de *gamma* é inomogêneo em todos os modelos. No que diz respeito aos valores de *nu*, sendo este o único caso onde a existe influência direta nos resultados, é plausível assumir que, neste cenário, valores mais elevados causam resultados de AUC-ROC e *f1-score* mais reduzidos.

Numa outra vertente, é interessante denotar que, embora os três parâmetros dos modelos difiram, a duração da fase de treino é sempre coincidente (0.002 segundos), o que indica uma boa *performance* deste algoritmo com o conjunto de dados em questão.

### 5.1.3 LSTM-AE

Os últimos modelos concebidos e avaliados com os dados do Azoto Total foram com recurso ao algoritmo da LSTM-AE. Nestes modelos, a abordagem é diferente da previamente utilizada com as iF e OCSVM. Inicialmente, é realizada uma otimização de hiperparâmetros onde é calculada a *loss* do *ouput* dos modelos face à reconstrução do valor de *input*. A partir deste momento, é possível verificar quais os conjuntos de hiperparâmetros que garantem uma reconstrução do valor de entrada com menor erro possível.

Com isto, são aplicados aos três melhores modelos os *thresholds* referidos na secção 4.1 onde são identificadas as anomalias que permitem posteriormente calcular as métricas de AUC-ROC e *f1-score* para cada um dos modelos concebidos.

Assim sendo, os três melhores modelos candidatos que obtiveram menor *loss* na reconstrução dos *inputs* do Azoto Total, estão representados na Tabela 5.3.

| # | a. | b. | c. | d. | e. | f. | g. | h. |
|---|---|---|---|---|---|---|---|---|
| 16 | 1 | 128 | 0.0 | tanh | 100 | 10 | 0.0066 | 15.55 |
| 68 | 3 | 128 | 0.0 | relu | 100 | 10 | 0.0140 | 43.61 |
| 32 | 2 | 64 | 0.0 | relu | 100 | 10 | 0.0159 | 25.99 |

Tabela 5.3: Melhores resultados obtidos na reconstrução dos *inputs* com modelos LSTM-AE. As letras representam o seguinte: a. camadas; b. neurónios; c. *dropout rate*; d. função de ativação; e. *epochs*; f. *batch size*; g. *loss*; h. tempo(segundos).

É imediatamente percetível que os valores de *dropout rate*, *epochs* e *batch size* são comuns aos três melhores modelos, o que por si só poderá indicar que para o problema em questão, estes valores poderão a melhor opção. Em sentido inverso, tanto os valores de camadas, neurónios e de função de ativação foram divergentes.

No que diz respeito à *loss* obtida, não deixa de ser interessante apontar que a melhor *loss* (0.0066), foi obtida com apenas uma camada, o que representa um modelo menos complexo, resultando no menor tempo de treino de entre os três modelos. Posto isto, é possível com base nestes resultados, assinalar que neste cenário em concreto, modelos com complexidade relativamente menor, obtiveram melhores

resultados.

Posto isto, na seguintes secções, serão aplicados os três *thresholds* previamente referidos a cada um dos três melhores modelos referidos na Tabela 5.3 de forma a identificar as possíveis anomalias e calcular as métricas de avaliação.

#### 5.1.3.1 *Threshold 1*

No caso do primeiro *threshold*, não foram identificadas anomalias, ou seja, isto significa que durante a fase de avaliação, não houve nenhum erro de reconstrução dos *inputs* que fosse superior ao maior erro obtido na reconstrução dos *inputs* na fase de treino. Com isto, todas as observações foram identificadas como não anómalas, resultando na inexistência de duas classes distintas para calcular as métricas de avaliação (AUC-ROC e *f1-score*).

#### 5.1.3.2 *Threshold 2*

O segundo *threshold* a ser aplicado aos três melhores modelos da LSTM-AE foi o *threshold* 2. Os resultados inerentes a este cenário encontram-se ilustrados na Tabela 5.4.

| a. | b. | c. | d. | e. | f. | g. | h. | i. |
|---|---|---|---|---|---|---|---|---|
| 1 | 128 | 0.0 | tanh | 100 | 10 | 0.707 | 0.767 | 15.55 |
| 2 | 64 | 0.0 | relu | 100 | 10 | 0.697 | 0.763 | 25.99 |
| 3 | 128 | 0.0 | relu | 100 | 10 | 0.658 | 0.745 | 43.61 |

Tabela 5.4: Melhores resultados obtidos nos modelos LSTM-AE com o *threshold* 2. As letras representam o seguinte: a. camadas; b. neurónios; c. *dropout rate*; d. função de ativação; e. *epochs*; f. *batch size*; g. AUC-ROC; h. *f1-score*; i. tempo(segundos).

Nesta Tabela é visível que o melhor modelo obteve 0.707 de AUC-ROC e 0.767 de *f1-score*. Os três modelos candidatos tiveram em comum a utilização do valor 0 para a *dropout rate*, 100 *epochs* e 10 *batch size* sendo que os valores de neurónios e função de ativação foram distintos.

É possível também constatar que o modelo que obteve os melhores resultados relativamente a *loss*, que também é o modelo que obteve menor tempo de duração na fase de treino devido a sua simplicidade, foi o que atingiu melhores resultados neste *threshold*. Por outro lado, o modelo com três camadas que tinha sido o segundo melhor modelo em termos de *loss*, acabou por obter piores resultados que o modelo com duas camadas.

No que à duração de fase de treino diz respeito, os modelos com maior duração da fase de treino, obtiveram resultados menos satisfatórios.

#### 5.1.3.3 *Threshold 3*

O último *threshold* utilizado com os três melhores modelos, concebidos com o algoritmo da LSTM-AE, foi o *threshold* 3. Os resultados alcançados com o Azoto Total neste *threshold* estão representados na Tabela 5.5.

| a. | b. | c. | d. | e. | f. | g. | h. | i. |
|---|---|---|---|---|---|---|---|---|
| 1 | 128 | 0.0 | tanh | 100 | 10 | 0.723 | 0.786 | 15.55 |
| 2 | 64 | 0.0 | relu | 100 | 10 | 0.695 | 0.770 | 25.99 |
| 3 | 128 | 0.0 | relu | 100 | 10 | 0.661 | 0.752 | 43.61 |

Tabela 5.5: Melhores resultados obtidos nos modelos LSTM-AE com o *threshold* 3. As letras representam o seguinte: a. camadas; b. neurónios; c. *dropout rate*; d. função de ativação; e. *epochs*; f. *batch size*; g. AUC-ROC; h. *f1-score*; i. tempo(segundos).

Nos resultados apresentados na Tabela referida é visível que dos três modelos candidatos, o melhor resultado foi 0.723 de AUC-ROC e 0.786 de *f1-score* através da combinação de 1 camada, 128 neurónios, 0.0 *dropout rate*, função de ativação *tanh*, 100 *epochs* e 10 *batch size*.

No top-3 dos modelos concebidos, foi comum a utilização do valor 0.0 para a *dropout rate*, 100 *epochs* e 10 *batch size*, por outro lado, os valores dos neurónios e função de ativação não foram homogéneos.

Em termos de resultados, existe uma correlação negativa com a duração da fase de treino, ou seja, o aumento da duração de fase de treino resultou em piores *performances* por parte dos modelos na classificação de anomalias com o *threshold* 3.

Para além disso, o melhor modelo candidato que obteve menor *loss* na fase de treino, foi também o modelo que obteve melhores resultados neste cenário.

### 5.1.4 Análise Comparativa

Com a obtenção de todos os resultados relativos ao Azoto Total, é de extrema relevância realizar uma comparação entre os vários modelos e a sua *performance*.

Como é possível ver na Figura 5.1, os modelos tradicionais, obtiveram resultados significativamente superiores aos modelos de DL. No caso da iF, que obteve os melhores resultados, foi possível atingir uma AUC-ROC de 0.975 e uma *f1-score* de 0.941, seguida do OCSVM que obteve 0.971 de AUC-ROC e 0.928 de *f1-score*. Por outro lado, com o *threshold* 2 foi possível obter 0.707 de AUC-ROC e 0.767 de *f1-score* sendo que, o *threshold* 3 superou ligeiramente estes resultados obtendo 0.723 e 0.786, respetivamente. No que diz respeito aos resultados do *threshold* 1, não foram obtidos quaisquer valores, pelos motivos previamente mencionados na secção 5.1.3.1.

Sobre os *thresholds* aplicados aos três melhores modelos de LSTM-AE concebidos, é interessante apontar que o *threshold* que obteve melhores resultados, foi o que teve uma abordagem de cálculo

dinâmico do valor de *threshold*. Ainda sobre os *thresholds*, conseguimos validar que os melhores modelos em termos de *loss* de reconstrução (Tabela 5.3), não mantiveram a mesma ordem em termos de resultados obtidos com as métricas de avaliação, ou seja, embora tenham apresentado uma melhor *performance* com os dados de treino, com os dados de *teste* que desconheciam, os resultados foram diferentes.

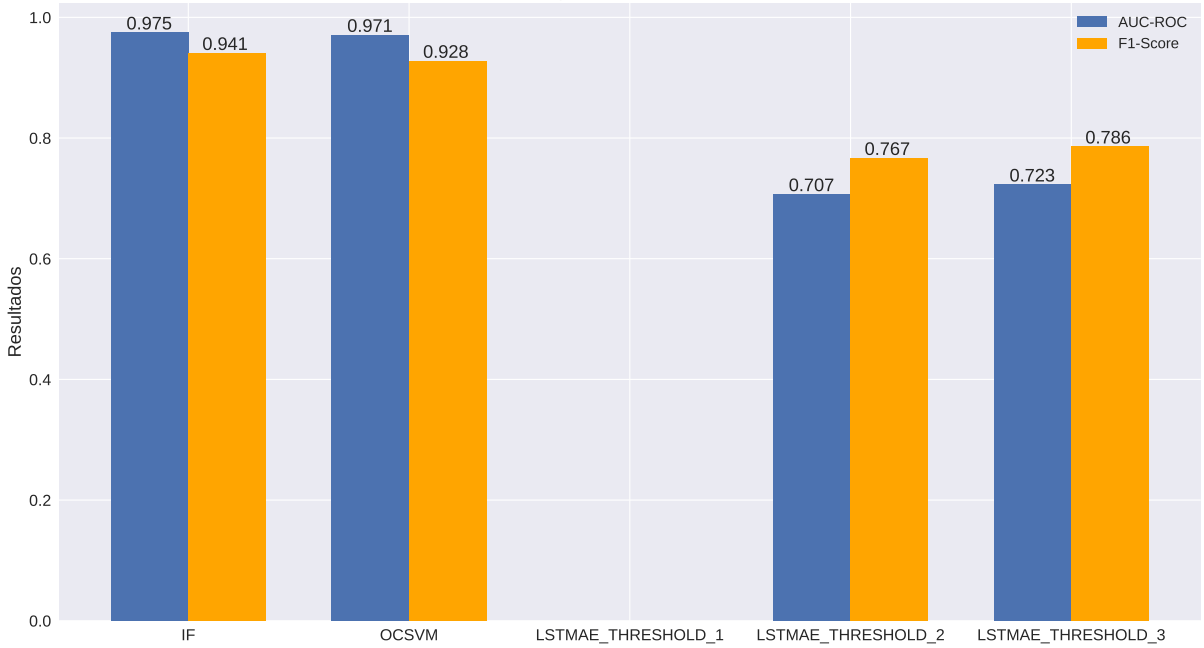
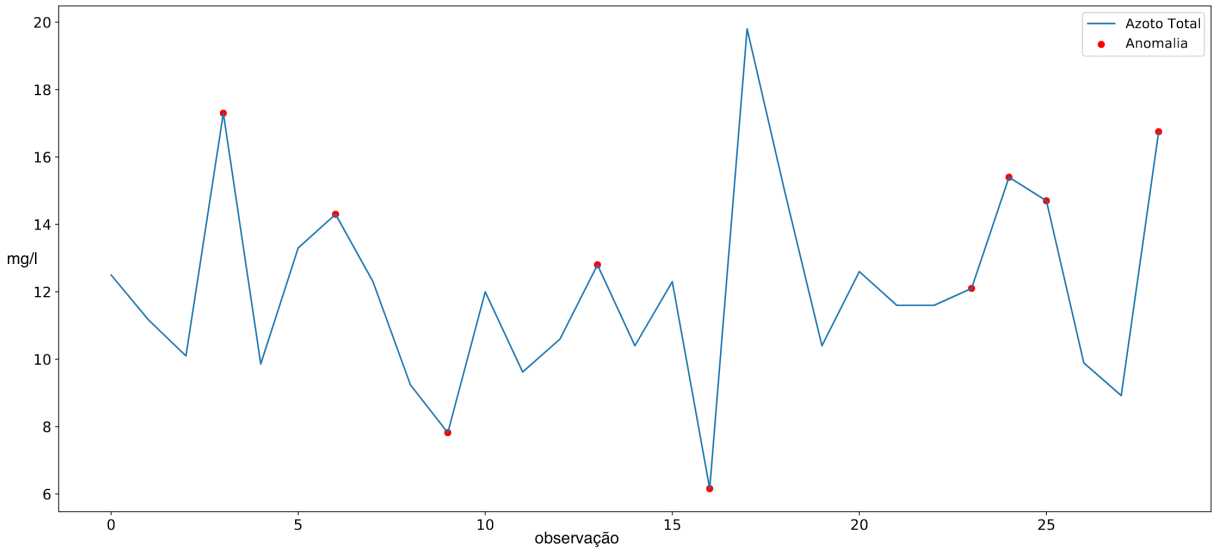



Figura 5.1: Comparação dos melhores resultados obtidos em cada modelo do Azoto Total

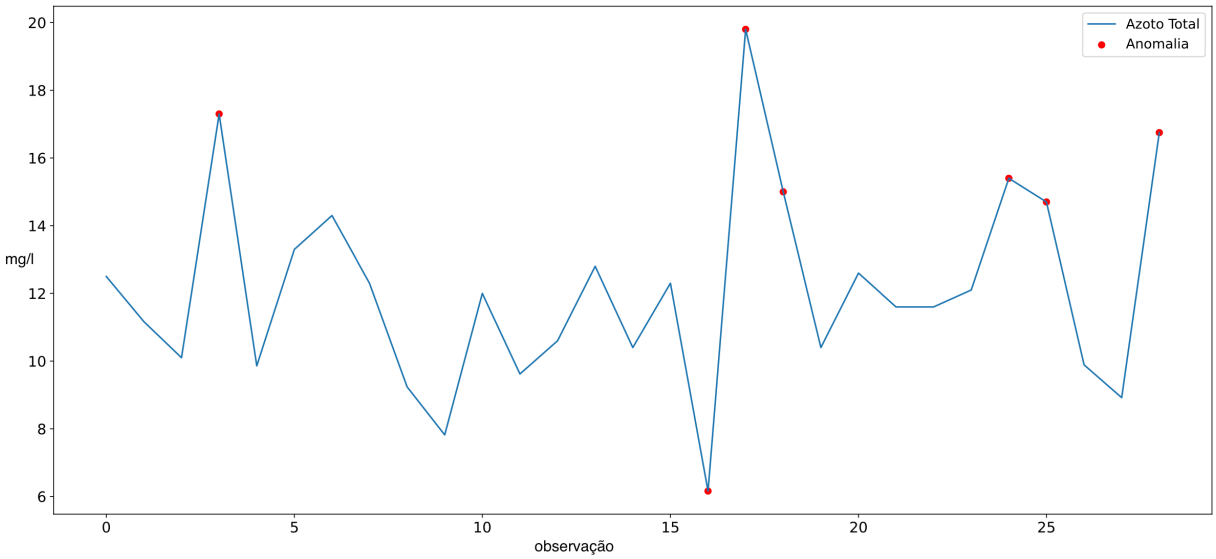
Com isto, o passo seguinte da análise comparativa, foca-se na comparação da deteção de anomalias por parte do melhor modelo candidato de DL com o melhor modelo de ML tradicional. Ou seja, no que diz respeito ao melhor modelo de ML tradicional, a iF foi a que obteve melhores resultados, por outro lado, os melhores resultados de DL foram obtidos com o *threshold* 3 aplicado à LSTM-AE.

Assim sendo, na Figura 5.2, encontram-se dois gráficos, onde o gráfico (a), representa as classificações de anomalias por parte do modelo de iF e no gráfico (b), as classificações de anomalias por parte do modelo LSTM-AE com o *threshold* 3. Nestas duas sub-figuras, é possível perceber, à partida, que a iF identificou 6 anomalias e a LSTM-AE com o *threshold* 3, detetou 9 anomalias. Posto isto, é importante perceber quantas foram bem identificadas e mal identificadas, ou seja, quais são TP e quais são FP. Com base na labelização realizada pelos especialistas que foi previamente mencionada, o Azoto Total tem um limite superior de 15 mg/l *N*, ou seja, no caso da iF, foram corretamente identificados 5 anomalias (TP) e incorretamente identificadas 2 observações como anómalas (FP). Do outro lado da moeda, a LSTM-AE, detetou corretamente 4 anomalias, porém, detetou incorretamente 5 registos como anomalia. Por último, os 2 registos que foram mal identificados pelo modelo da iF (registos 18 e 25), que também foram incorretamente identificados pela LSTM-AE, trata-se de um valor extremamente próximo do limite no caso do registo 25 (14.7 mg/l *N*), e de um valor bastante reduzido no caso do registo 18 (6.16 mg/l *N*), ou seja, por outras palavras, embora sejam FP, tratam-se de valores que, num contexto real, seriam relevantes de

ser notificados.




(a) iF




(b) LSTM-AE *Threshold* 3

Figura 5.2: Comparação melhor modelo ML tradicional com melhor modelo DL do Azoto Total

## 5.2 Nitratos

Os Nitratos ($NO_3$) foram o segundo indicador ao qual foram aplicados os modelos de ML. Tal como no Azoto Total (secção 5.1), os dados foram utilizados em modelos de DL (LSTM-AE) e modelos de ML convencionais (iF e OCSVM).

Posto isto, de seguida é possível verificar os resultados obtidos para os modelos mencionados bem como uma análise comparativa dos mesmos.

### 5.2.1 iF

O primeiro algoritmo que foi utilizado para conceber modelos com o conjunto de dados dos Nitratos foi a iF. Os resultados da avaliação destes modelos com as métricas de avaliação AUC-ROC e *f1-score* estão ilustrados na Tabela 5.6.

| # | a. | b. | c. | d. | e. | f. | g. |
|---|---|---|---|---|---|---|---|
| 36 | 100 | 80 | 0.02 | True | 0.979 | 0.808 | 0.199 |
| 56 | 100 | 100 | 0.04 | True | 0.974 | 0.778 | 0.183 |
| 78 | 100 | 120 | 0.08 | True | 0.968 | 0.733 | 0.196 |

Tabela 5.6: Melhores resultados obtidos nos modelos iF. As letras representam o seguinte: a. *n_estimators*; b. *max_samples*; c. *contamination*; d. *bootstrap*; e. AUC-ROC; f. *f1-score*; g. tempo(segundos).

Nesta Tabela 5.6 é visível que o melhor modelo candidato conseguiu obter uma AUC-ROC de 0.979 e *f1-score* de 0.808. No *top-3* listado, é possível averiguar que os hiperparâmetros *n_estimators* e *bootstrap* são homogéneos nos três melhores modelos candidatos. Por outro lado, os atributos *max_samples* e *contamination* são sempre diferentes em todos os modelos. Focando nos valores do hiperparâmetro *max_samples*, é passível de concluir que o modelo utilizar mais registos, não é inerente a uma melhor *performance*, por outro lado, tal como é expectável, o aumento dos valores do atributo *contamination* levam a resultados menos satisfatórios.

Por fim, os resultados permitem uma outra conclusão, no top-3 a duração da fase de treino do modelo, nem sempre está diretamente relacionada com uma melhor *performance*, tal como é possível verificar no segundo e terceiro melhor modelo, onde o terceiro modelo, embora tenha demorado mais segundos na fase de treino, obteve resultados ligeiramente piores que o segundo melhor modelo.

### 5.2.2 OCSVM

A segunda etapa no que diz respeito à conceção e avaliação de modelos com o indicador dos Nitratos, foi realizada com o algoritmo do OCSVM. Os três modelos que apresentaram os melhores resultados relativos ás duas métricas de avaliação consideradas, estão ilustrados na Tabela 5.7.

| # | a. | b. | c. | d. | e. | f. |
|---|---|---|---|---|---|---|
| 162 | rbf | 0.038 | 0.02 | 0.983 | 0.800 | 0.002 |
| 100 | rbf | 0.024 | 0.04 | 0.966 | 0.667 | 0.003 |
| 2 | rbf | 0.002 | 0.06 | 0.948 | 0.571 | 0.002 |

Tabela 5.7: Melhores resultados obtidos nos modelos OCSVM. As letras representam o seguinte: a. *kernel*; b. *gamma*; c. *nu*; d. AUC-ROC; e. *f1-score*; f. tempo(segundos).

Assim sendo, na Tabela 5.7 é possível verificar que o melhor modelo candidato obteve 0.983 de AUC-ROC e 0.800 de *f1-score*. Nestes resultados é possível comprovar a utilização do *kernel rfb* em todos os modelos. Em sentido contrário, os valores de *gamma* e *nu* são diferentes em todos os modelos. Ainda sobre os valores de *nu*, é notória uma relação proporcional inversa entre a subida do valor de *nu* e a descida dos valores de AUC-ROC e *f1-score* neste top-3.

Por fim, é importante denotar que embora os parâmetros dos modelos contenham valores diferentes, a duração da fase de treino é sempre bastante similar, diferenciando apenas no segundo modelo em 0.001 segundos. Tal como previamente referido na secção 5.1.2, isto indica uma boa *performance* do algoritmo de OCSVM neste cenário em específico.

### 5.2.3 LSTM-AE

Os últimos modelos concebidos com os dados dos Nitratos foi realizada com recurso ao algoritmo da LSTM-AE. Como mencionado previamente na secção 5.1.3, nestes modelos, a abordagem é mais complexa que a abordagem utilizada com os modelos das iF e OCSVM. Como previamente referido, numa primeira fase, é realizada uma otimização de hiperparâmetros onde é calculada a *loss* dos modelos na fase de reconstrução do valor de *input*. A partir deste momento, é possível verificar quais os conjuntos de hiperparâmetros que garantem uma reconstrução do valor de entrada com menor erro possível.

Com isto, são aplicados aos três melhores modelos candidatos os *thresholds* referidos na secção 4.1 onde são identificadas as anomalias que permitem posteriormente calcular as métricas de AUC-ROC e *f1-score* para cada um dos modelos concebidos. Estes três modelos estão representados na Tabela 5.8.

| # | a. | b. | c. | d. | e. | f. | g. | h. |
|---|---|---|---|---|---|---|---|---|
| 82 | 3 | 128 | 0.0 | relu | 300 | 10 | 0.0565 | 49.53 |
| 16 | 1 | 128 | 0.0 | relu | 100 | 10 | 0.0670 | 12.79 |
| 4 | 1 | 64 | 0.0 | tanh | 100 | 10 | 0.0913 | 9.52 |

Tabela 5.8: Melhores resultados obtidos na reconstrução dos *inputs* com modelos LSTM-AE. As letras representam o seguinte: a. camadas; b. neurónios; c. *dropout rate*; d. função de ativação; e. *epochs*; f. *batch size*; g. *loss*; h. tempo(segundos).

No que aos parâmetros dos modelos candidatos diz respeito, é factual que os valores de *dropout rate* (0.0) e *batch size* (10) são iguais em todos os três modelos candidatos. Em sentido oposto, tanto os valores de camadas, neurónios, função de ativação e *epochs* foram díspares nos melhores modelos.

Quando a análise se foca nos valores de *loss* obtidos, conseguimos verificar uma relação proporcional inversa entre a *loss* e a duração/complexidade dos modelos, ou seja, modelos um pouco mais complexos, como por exemplo, com mais camadas, conseguiram obter uma menor *loss*. Posto isto, neste caso em específico, é possível afirmar que nos três melhores modelos, os mais complexos obtiveram resultados

ligeiramente melhores nos valores da *loss* de reconstrução.

#### 5.2.3.1 *Threshold 1*

O primeiro *threshold* que foi utilizado foi o *threshold* 1. Os resultados relativos a este cenário estão enumerados na Tabela 5.9 com ordenação descendente pelas métricas (AUC-ROC e *f1-score*).

| a. | b. | c. | d. | e. | f. | g. | h. | i. |
|---|---|---|---|---|---|---|---|---|
| 1 | 128 | 0.0 | relu | 100 | 10 | 0.990 | 0.989 | 12.79 |
| 1 | 64 | 0.0 | tanh | 100 | 10 | 0.989 | 0.988 | 9.52 |
| 3 | 128 | 0.0 | relu | 300 | 10 | 0.884 | 0.984 | 49.53 |

Tabela 5.9: Melhores resultados obtidos nos modelos LSTM-AE com o *threshold* 1. As letras representam o seguinte: a. camadas; b. neurónios; c. *dropout rate*; d. função de ativação; e. *epochs*; f. *batch size*; g. AUC-ROC; h. *f1-score*; i. tempo(segundos).

Com base nos resultados listados na Tabela, é visível que o melhor modelo de LSTM-AE com o *threshold* 1 obteve 0.990 de AUC-ROC e 0.989 de *f1-score*. Em comum nos três melhores modelos candidatos está a utilização do valor 0.0 para a *dropout rate* bem como o valor de 10 em *batch size*. Em sentido inverso, tanto os valores de neurónios como de função de ativação não são homogéneos.

No que diz respeito à duração da fase de treino, é relevante mencionar que o modelo candidato com resultados menos satisfatórios é o modelo que necessitou de mais segundos para concluir a aprendizagem com os dados de treino, ou seja, mais complexo. Por fim, é também possível analisar que o modelo que obteve menor *loss* na fase de reconstrução, não foi o modelo que obter melhores resultados na fase de deteção de anomalias com o *threshold* 1.

#### 5.2.3.2 *Threshold 2*

Concluída a aplicação do primeiro *threshold*, foram realizados os testes com o *threshold* 2. Assim sendo, os resultados relativos a este cenário são apresentados na Tabela 5.10.

| a. | b. | c. | d. | e. | f. | g. | h. | i. |
|---|---|---|---|---|---|---|---|---|
| 1 | 128 | 0.0 | relu | 100 | 10 | 0.995 | 0.995 | 12.79 |
| 3 | 128 | 0.0 | relu | 300 | 10 | 0.994 | 0.994 | 49.53 |
| 1 | 64 | 0.0 | tanh | 100 | 10 | 0.993 | 0.993 | 9.52 |

Tabela 5.10: Melhores resultados obtidos nos modelos LSTM-AE com o *threshold* 2. As letras representam o seguinte: a. camadas; b. neurónios; c. *dropout rate*; d. função de ativação; e. *epochs*; f. *batch size*; g. AUC-ROC; h. *f1-score*; i. tempo(segundos).

Em análise à Tabela previamente referida, é constatável que o melhor modelo concebido obteve 0.995 de AUC-ROC e 0.995 de *f1-score*. No que diz respeito a homogeneidade, todos os modelos tiveram em

comum a utilização do valor 0.0 para a *dropout rate* e do valor 10 para o atributo *batch size*. Por outro lado, os valores dos atributos neurónios e função de ativação foram distintos.

Em último lugar, sobre a duração das fases de treino, tal como se sucedeu na secção 5.2.3.1, o modelo candidato com melhores resultados não é o modelo que demorou mais tempo na fase de treino, no entanto, ao contrário do verificado na secção 5.2.3.1, o terceiro melhor modelo candidato não corresponde ao primeiro modelo com menor *loss* da fase de treino. Para além disso, é verificável que o terceiro melhor modelo candidato em termos de resultados de *loss* é também o terceiro melhor modelo candidato nos resultados obtidos com o *threshold* 2.

#### 5.2.3.3 *Threshold 3*

Por fim, o último *threshold* utilizado em conjunto com o top-3 de modelos da LSTM-AE foi o *threshold* 3. Os valores das métricas de avaliação bem como os hiperparâmetros dos modelos estão ilustrados na Tabela 5.11.

| a. | b. | c. | d. | e. | f. | g. | h. | i. |
|---|---|---|---|---|---|---|---|---|
| 1 | 64 | 0.0 | tanh | 100 | 10 | 0.817 | 0.938 | 9.52 |
| 1 | 128 | 0.0 | relu | 100 | 10 | 0.656 | 0.959 | 12.79 |
| 3 | 128 | 0.0 | relu | 300 | 10 | 0.646 | 0.948 | 49.53 |

Tabela 5.11: Melhores resultados obtidos nos modelos LSTM-AE com o *threshold* 3. As letras representam o seguinte: a. camadas; b. neurónios; c. *dropout rate*; d. função de ativação; e. *epochs*; f. *batch size*; g. AUC-ROC; h. *f1-score*; i. tempo(segundos).

Nos resultados presentes na Tabela anterior, é visível que o melhor modelo candidato obteve 0.817 de AUC-ROC e 0.938 de *f1-score*. No que diz respeito aos atributos dos modelos, os três melhores modelos candidatos utilizaram o valor 0.0 para o atributo *dropout rate* e o valor 10 para o atributo *batch size*. No entanto, os valores dos atributos neurónios e função de ativação foram diferentes nos melhores modelos concebidos.

Em último lugar, sobre a duração das fases de treino, o modelo candidato com melhores resultados é também o modelo que demorou um menor tempo na fase de treino, ou seja, é verificada uma relação de proporcionalidade inversa entre duração da fase de treino e resultados obtidos, onde modelos com menor duração da fase de treino, obtém valores de AUC-ROC e *f1-score* superiores. No que diz respeito aos valores de *loss* dos modelos candidatos, verifica-se a mesma situação que na duração da fase de treino, onde os modelos com menor *loss*, obtiveram piores resultados com o último *threshold*.

### 5.2.4 Análise Comparativa

Após analisar individualmente cada um dos resultados dos vários modelos com os dados dos Nitratos, é interessante comparar de um ponto de vista global todos estes valores.

Posto isto, na Figura 5.3, conseguimos visualizar que grande parte dos resultados foram bastante satisfatórios, chegando a atingir valores na ordem dos 0.99. No que diz respeito aos valores dos modelos tradicionais, iF e OCSVM, os valores de AUC-ROC foram de 0.979 e 0.983, no que diz respeito à *f1-score* foram de 0.808 e 0.8, respetivamente. Por outro lado, os resultados dos modelos de DL foram consideravelmente superiores, com os dois primeiros *thresholds* (1 e 2), a obter tanto os valores de AUC-ROC como de *f1-score* na ordem de 0.99. Por fim, o *threshold* 3 obteve 0.817 de AUC-ROC e 0.938 de *f1-score* no seu melhor modelo. É também relevante assinalar a diferença de valores de *f1-score* dos modelos da iF e OCSVM para os modelos baseados em LSTM-AE. Esta diferença surge essencialmente pela forma de cálculo desta métrica, onde esta situação nos leva a concluir que os modelos da iF e OCSVM estariam a detetar falsas anomalias (FP), o que baixa causa um valor de *recall* mais baixo, diminuindo significativamente desta forma o valor da *f1-score* [120].

Ao contrário do que foi verificado na análise comparativa do Azoto (secção 5.1.4), no caso dos Nitratos, o pior resultado foi obtido com o *threshold* dinâmico (*threshold* 3), sendo que os *thresholds* com valores constantes, obtiverem os melhores resultados. De outro prisma, no caso da relação entre a *loss* de reconstrução dos dados de entrada (Tabela 5.8) e os resultados obtidos pelos modelos nas métricas de avaliação, é visível que o *threshold* 3, que tinha obtido os melhores resultados no campo da *loss*, foi o que obteve os piores resultados quando aplicados os *thresholds*.

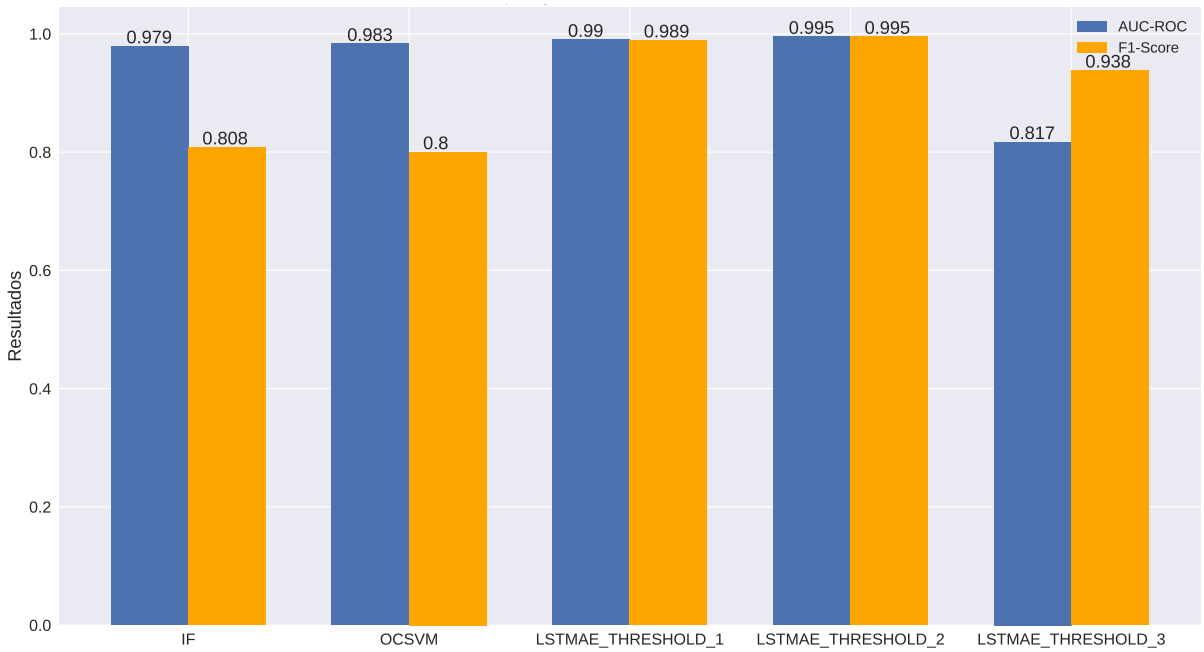


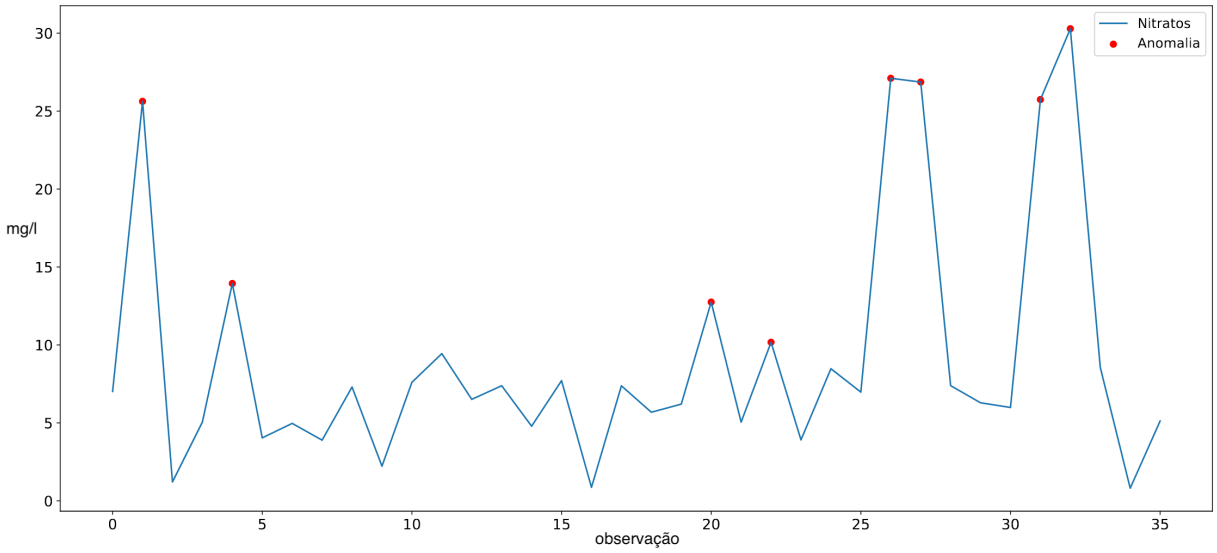

Figura 5.3: Comparação dos melhores resultados obtidos em cada modelo dos Nitratos

Seguidamente, foi realizada uma comparação da deteção de anomalias entre o melhor modelo de DL e o melhor modelo de ML tradicional, ou seja, a comparação foi realizada entre a iF (ML tradicional), que obteve os melhores resultados. No caso de DL, os melhores resultados foram obtidos com o *threshold* 2 quando aplicado à LSTM-AE.

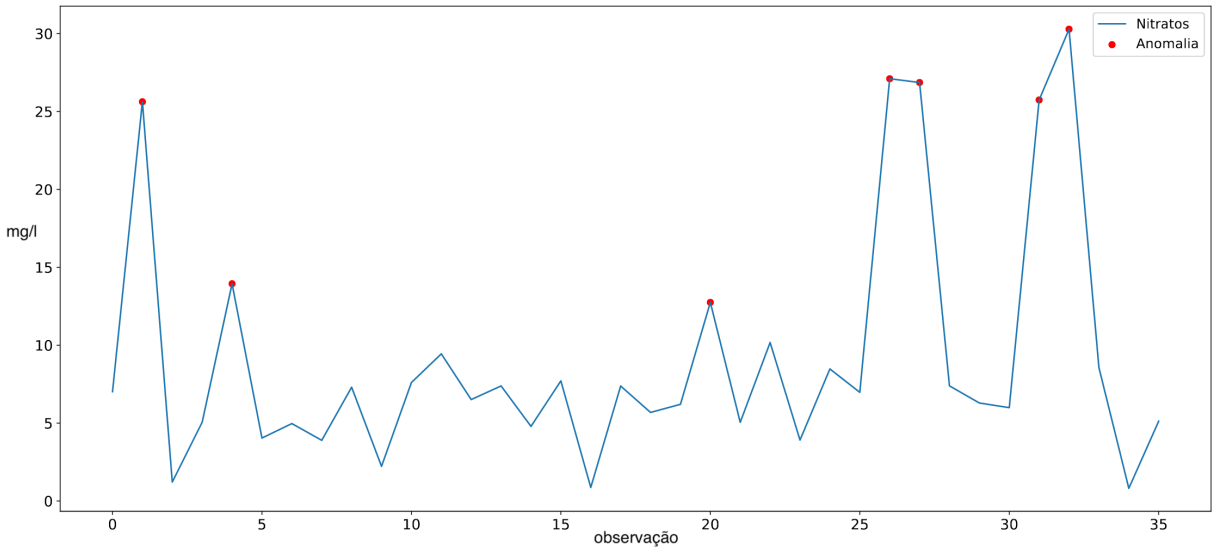
Com isto, na Figura 5.4, estão presentes duas sub-figuras onde o gráfico (a) demonstra as observações

identificas como anomalias por parte do modelo iF e, no gráfico (b), estão representadas as observações identificadas como anomalias pelo modelo LSTM-AE com o *threshold* 2 aplicado. Assim sendo, nestas duas sub-figuras, é percetível que as anomalias identificadas são bastante similares, diferindo apenas numa observação. Ou seja, no caso do modelo da iF, foram identificadas 8 observações como anomalias e, por outro lado, no caso do modelo da LSTM-AE, identificaram-se 7 registos como anomalias. Destas anomalias, e tendo como base a labelização realizada pelos especialistas, é visível que existem em ambos os modelos, 5 anomalias corretamente identificadas (TP), porém, no caso dos registos incorretamente identificados como anomalia, a iF identificou 3, e a LSTM-AE identificou apenas 2. No caso destes registos indevidamente identificados como anomalias (4, 20, e 22), é visível que, embora estejam abaixo do valor limite de emissão, são valores consideravelmente elevados, para os quais, num contexto de uma ETAR, faria sentido alertar os colaboradores.

Por fim, de forma conclusiva, devido ao número de TP e FP identificados, é visível que, tal como na Figura 5.3, o modelo tradicional da iF, obteve uma *performance* ligeiramente inferior quando comparado com os resultados de DL obtidos pelo *threshold* 2 aplicado ao modelo da LSTM-AE com o conjunto de dados dos Nitratos.




(a) iF




(b) LSTM-AE Threshold 2

Figura 5.4: Comparação melhor modelo ML tradicional com melhor modelo DL dos Nitratos

## 5.3 pH

Os dados do pH foram o último conjunto de dados a ser utilizado na aplicação dos modelos de ML. Como foi previamente mencionado no Azoto Total e Nitratos (secções 5.1 e 5.2), recorreu-se a estes registos para a conceção de modelos de DL, nomeadamente, LSTM-AE, bem como, modelos convencionais, mais concretamente, iF e OCSVM.

Por fim, de seguida, estão listados os resultados atingidos para os modelos supramencionados bem como comparação entre as *performances* atingidas.

### 5.3.1 iF

O algoritmo da iF foi o primeiro a ser utilizado para conceber modelos com o conjunto de dados do pH. Os resultados obtidos nestes modelos candidatos foram ordenados de forma descendente pelas métricas de avaliação e estão representados na Tabela 5.12 os três melhores modelos candidatos.

| # | a. | b. | c. | d. | e. | f. | g. |
|---|---|---|---|---|---|---|---|
| 20 | 100 | 60 | 0.04 | True | 0.985 | 0.889 | 0.217 |
| 58 | 100 | 100 | 0.06 | True | 0.977 | 0.842 | 0.201 |
| 63 | 100 | 100 | 0.1 | False | 0.970 | 0.800 | 0.213 |

Tabela 5.12: Melhores resultados obtidos nos modelos iF. As letras representam o seguinte: a. *n_estimators*; b. *max_samples*; c. *contamination*; d. *bootstrap*; e. AUC-ROC; f. *f1-score*; g. tempo(segundos).

Na Tabela previamente referida, é notório que o modelo candidato que obteve melhores resultados atingiu um valor de AUC-ROC de 0.985 e *f1-score* de 0.889. Nos três melhores modelos candidatos enumerados é possível constatar que o atributo *n_estimators* foi o único a obter valores iguais nos três modelos, por outro lado, os atributos *max_samples*, *contamination* e *bootstrap* apresentam valores diferentes em todos os modelos. No que diz respeito aos valores do atributo *max_samples*, é visível que o modelo candidato que utilizou menos registos, obteve os melhores resultados, por outro lado, tal como é expectável, o aumento dos valores do atributo *contamination*, tem como consequência direta, resultados menos satisfatórios.

Por último, os resultados apresentados na Tabela permitem interpretar que a duração da fase de treino do modelo, nem sempre está diretamente relacionada com uma melhor *performance*. Tal como é possível verificar, embora o segundo modelo candidato tenha demorado mais tempo na fase de treino, obteve resultados mais satisfatórios que o terceiro melhor modelo candidato.

### 5.3.2 OCSVM

Concluída a conceção dos modelos baseados no algoritmo da iF, o segundo algoritmo utilizado foi o OCSVM. Todos os modelos construídos com este algoritmo foram avaliados com a AUC-ROC e *f1-score* sendo que, o top-3 de modelos candidatos, está representado na Tabela 5.13.

| # | a. | b. | c. | d. | e. | f. |
|---|---|---|---|---|---|---|
| 712 | linear | 0.16 | 0.04 | 0.992 | 0.941 | 0.002 |
| 381 | linear | 0.086 | 0.08 | 0.985 | 0.889 | 0.002 |
| 186 | linear | 0.042 | 0.14 | 0.977 | 0.842 | 0.003 |

Tabela 5.13: Melhores resultados obtidos nos modelos OCSVM. As letras representam o seguinte: a. *kernel*; b. *gamma*; c. *nu*; d. AUC-ROC; e. *f1-score*; f. tempo(segundos).

Nos resultados descritos na Tabela referida, verifica-se que o modelo candidato com melhores resultados conseguiu obter 0.992 de AUC-ROC e 0.941 de *f1-score*. Com estes resultados é visível que a utilização do *kernel linear* no indicador pH, ao contrário do sucedido na secção 5.2.2 com os Nitratos, é o mais indicado para atingir os melhores resultados neste cenário específico, verificando-se este valor nos três melhores modelos candidatos. Em sentido contrário, os valores de *gamma* e *nu* variam em todos os modelos. Para além disso, no que diz respeito aos valores de *gamma*, verifica-se uma relação de proporcionalidade, onde valores mais elevados impactam em melhores resultados. No caso dos valores de *nu*, a situação é oposta, verifica-se uma relação de proporcionalidade inversa onde a subida do valor de *nu* causa uma descida dos valores de AUC-ROC e *f1-score*.

Posto isto, a Tabela é passível de uma outra conclusão, é visível que embora os parâmetros dos três melhores modelos candidatos contenham valores distintos, a duração da fase de treino é sempre bastante idêntica com apenas o último modelo candidato a demorar mais 0.001 segundos.

### 5.3.3 LSTM-AE

Os modelos candidatos de DL do pH foram concebidos com a utilização do algoritmo da LSTM-AE. Como previamente referido na secção 5.1.3. Para estes modelos , a abordagem seguida é bastante distinta da abordagem utilizada com os modelos "tradicionais"(iF e OCSVM). O primeiro passo, é otimizar os seis hiperparâmetros do algoritmo de forma a conseguir obter a menor *loss* possível no *output* da reconstrução do valor de *input*. Realizado este passo, são extraídos os três modelos que conseguem reconstruir os dados com o menor erro possível.

Posto isto, com os melhores modelos obtidos, são aplicados os três *thresholds* mencionados na secção 4.1 de forma a realizar a labelização dos registos de *input* como anómalos ou normais com base na *loss* da reconstrução dos dados. Com esta *label* e com a *label* prevista, são calculadas as métricas de AUC-ROC e *f1-score* descritas na secção 4.2 para cada um dos modelos.

Assim sendo, os três modelos que obtiveram melhor *performance* no que diz respeito a *loss* na fase de reconstrução dos valores de entrada de pH estão representados na Tabela 5.14.

| # | a. | b. | c. | d. | e. | f. | g. | h. |
|---|---|---|---|---|---|---|---|---|
| 19 | 1 | 128 | 0.0 | relu | 100 | 10 | 0.0072 | 16.20 |
| 35 | 2 | 64 | 0.0 | relu | 100 | 10 | 0.0086 | 23.78 |
| 7 | 1 | 64 | 0.0 | relu | 100 | 10 | 0.0098 | 19.64 |

Tabela 5.14: Melhores resultados obtidos na reconstrução dos *inputs* com modelos LSTM-AE. As letras representam o seguinte: a. camadas; b. neurónios; c. *dropout rate*; d. função de ativação; e. *epochs*; f. *batch size*; g. *loss*; h. tempo(segundos).

Nestes modelos, é notório que os valores de *dropout rate*, função de ativação, *epochs* e *batch size* são homogéneos nos três melhores modelos, respetivamente, 0.0, 100 e 10. Por outro lado, tanto o valor

das camadas como dos neurónios foram dispares nos três melhores modelos.

Com foco na *loss* obtida, é possível validar que o melhor resultado foi de 0.0072 com uma duração de 16.20 segundos, ou seja, o modelo com menor duração na fase de treino, é o modelo que obtém melhores resultados.

#### 5.3.3.1 *Threshold 1*

O *threshold* 1 foi o primeiro a ser aplicado ao top-3 de modelos do algoritmo da LSTM-AE previamente referidos. Os resultados obtidos com esta operação, encontram-se descritos na Tabela 5.15 ordenados de forma descendente pelas métricas de avaliação.

| a. | b. | c. | d. | e. | f. | g. | h. | i. |
|---|---|---|---|---|---|---|---|---|
| 1 | 64 | 0.0 | relu | 100 | 10 | 0.9 | 0.993 | 19.64 |
| 2 | 64 | 0.0 | relu | 100 | 10 | 0.8 | 0.986 | 23.78 |
| 1 | 128 | 0.0 | relu | 100 | 10 | 0.7 | 0.979 | 16.20 |

Tabela 5.15: Melhores resultados obtidos nos modelos LSTM-AE com o *threshold* 1. As letras representam o seguinte: a. camadas; b. neurónios; c. *dropout rate*; d. função de ativação; e. *epochs*; f. *batch size*; g. AUC-ROC; h. *f1-score*; i. tempo(segundos).

Através dos resultados que foram obtidos com o *threshold* 1, que estão enumerados na Tabela antecedente, é visível que o melhor modelo conseguiu atingir 0.9 de AUC-ROC e 0.993 de *f1-score*, em sentido contrário, o pior modelo obteve 0.7 de AUC-ROC e 0.979 de *f1-score*.

Quanto aos parâmetros, é notório que os dois melhores modelos, são os modelos com 64 neurónios, sendo que, o modelo com resultados menos positivos utilizou 128 neurónios, ou seja, a utilização de maior número de neurónios não está relacionada com melhores resultados neste top-3.

Por fim, relativamente à duração da fase de treino, o modelo com menor duração na fase de treino, foi também o modelo que obteve os piores resultados. No que diz respeito à *performance* dos modelos em termos de *loss*, existe uma inversão nos melhores modelos, ou seja, o melhor modelo em termos de *loss*, foi o modelo que obteve resultados menos satisfatórios nas métricas de avaliação com o *threshold* 1. Por outro lado, o modelo que obteve resultados menos positivos de *loss*, foi o modelo que obteve melhores resultados nas métricas de avaliação com o *threshold* 1.

#### 5.3.3.2 *Threshold 2*

O segundo *threshold* aplicado à LSTM-AE do conjunto de dados do pH, foi o *threshold* 2. Os resultados inerentes a esta experiência encontram-se ilustrados na Tabela 5.16.

| a. | b. | c. | d. | e. | f. | g. | h. | i. |
|---|---|---|---|---|---|---|---|---|
| 1 | 128 | 0.0 | relu | 100 | 10 | 0.853 | 0.982 | 16.20 |
| 2 | 64 | 0.0 | relu | 100 | 10 | 0.829 | 0.980 | 23.78 |
| 1 | 64 | 0.0 | relu | 100 | 10 | 0.829 | 0.980 | 19.64 |

Tabela 5.16: Melhores resultados obtidos nos modelos LSTM-AE com o *threshold* 2. As letras representam o seguinte: a. camadas; b. neurónios; c. *dropout rate*; d. função de ativação; e. *epochs*; f. *batch size*; g. AUC-ROC; h. *f1-score*; i. tempo(segundos).

Através dos resultados da Tabela 5.16 é possível verificar que existiram dois modelos com resultados de métricas de avaliação estritamente iguais, ou seja, a quantidade de observações anómalas e não anómalas corretamente identificadas foram sempre as mesmas. Estes dois modelos, obtiveram assim uma AUC-ROC de 0.829 e *f1-score* de 0.980. Por outro lado, o melhor modelo, que utilizou 128 de neurónios, com uma camada, obteve 0.853 de AUC-ROC e 0.982 de *f1-score*.

No que diz respeito aos parâmetros dos modelos, os dois modelos que obtiveram o pior resultado que utilizaram ambos, 64 neurónios, no entanto, o melhor modelo, foi atingido com 128 neurónios, ou seja, verifica-se um impacto direto do número de neurónios nos resultados finais neste cenário em concreto. Por outro lado, o número de camadas que é o único parâmetro, além dos neurónios, não homogéneo entre os três modelos, não revela qualquer relação direta com os resultados obtidos.

Em último lugar, é notório que, no que diz respeito à duração da fase de treino, o modelo que obteve melhores resultados, foi também o modelo que demorou menos tempo na conclusão da fase de treino (16.20 segundos). Ou seja, mais uma vez, neste caso em específico, modelos com menor duração, não têm necessariamente piores resultados, mas sim, o inverso. É também interessante denotar que, a ordem de melhores resultados dos modelos com a *threshold* 2, assimilou-se à ordem de resultados dos modelos no que diz respeito à *loss* de reconstrução.

#### 5.3.3.3 *Threshold 3*

Por fim, o último *threshold* considerado para este indicador, foi o *threshold* 3. O resultado da aplicação deste *threshold* ao top-3 de modelos da LSTM-AE está enumerada na Tabela 5.17.

| a. | b. | c. | d. | e. | f. | g. | h. | i. |
|---|---|---|---|---|---|---|---|---|
| 1 | 128 | 0.0 | relu | 100 | 10 | 0.786 | 0.955 | 16.20 |
| 1 | 64 | 0.0 | relu | 100 | 10 | 0.778 | 0.953 | 19.64 |
| 2 | 64 | 0.0 | relu | 100 | 10 | 0.771 | 0.943 | 23.78 |

Tabela 5.17: Melhores resultados obtidos nos modelos LSTM-AE com o *threshold* 3. As letras representam o seguinte: a. camadas; b. neurónios; c. *dropout rate*; d. função de ativação; e. *epochs*; f. *batch size*; g. AUC-ROC; h. *f1-score*; i. tempo(segundos).

Nesta Tabela o melhor modelo conseguiu alcançar 0.786 de AUC-ROC e 0.955 de *f1-score*. Por outro lado, os resultados menos positivos foram de 0.771 de AUC-ROC e uma *f1-score* de 0.943.

Sobre os hiperparâmetros dos modelos, é possível observar duas relações entre os resultados e os valores dos hiperparâmetros. Em primeiro lugar, o maior número de neurónios (128), resultou em melhores resultados. Por outro lado, um maior número de camadas (2), resultou em piores resultados no que a métricas de avaliação diz respeito neste cenário em particular.

Quanto à duração das fases de treino, é factual que existe uma relação inversa entre a mesma e os resultados das métricas de avaliação neste top-3 de resultados, ou seja, quando a duração da fase de treino é maior, os resultados dos modelos com o *threshold* 3 são piores, e vice-versa. Por outras palavras, os modelos com menor duração na reconstrução dos *inputs*, obtiveram melhores resultados neste cenário em concreto. Por fim, é também relevante verificar que o modelo que obteve melhor *loss* de reconstrução, foi o modelo que também obteve melhores resultados neste cenário.

### 5.3.4 Análise Comparativa

Concluída a análise dos resultados de todos os cenários, é fundamental comparar o melhor resultado obtido em cada um dos modelos com os dados do pH.

Na Figura 5.5 estão representadas as barras que indicam o valor das métricas para os melhores modelos de cada um dos algoritmos e, é possível verificar que os modelos tradicionais obtiveram melhores resultados, onde, com destaque, o OCSVM obteve os melhores resultados. Por outro lado, nos modelos de DL, o pior resultado pertenceu ao *threshold* 3, sendo que os melhores resultados foram atingidos pelo *threshold* 1.

Tal como se sucedeu com os dados dos Nitratos, o pior resultado da aplicação dos *thresholds*, aos modelos concebidos com o algoritmo da LSTM-AE, foi com o *threshold* 3, ou seja, o *threshold* dinâmico. Por fim, na relação entre a *loss* de reconstrução dos *inputs* de pH com os valores das métricas de avaliação após aplicação dos três *thresholds*, verifica-se que no caso do *threshold* 2, existe precisamente a mesma ordem em termos de *ranking* dos melhores modelos, com os modelos com melhor *loss* a atingir melhores resultados nas métricas de avaliação.

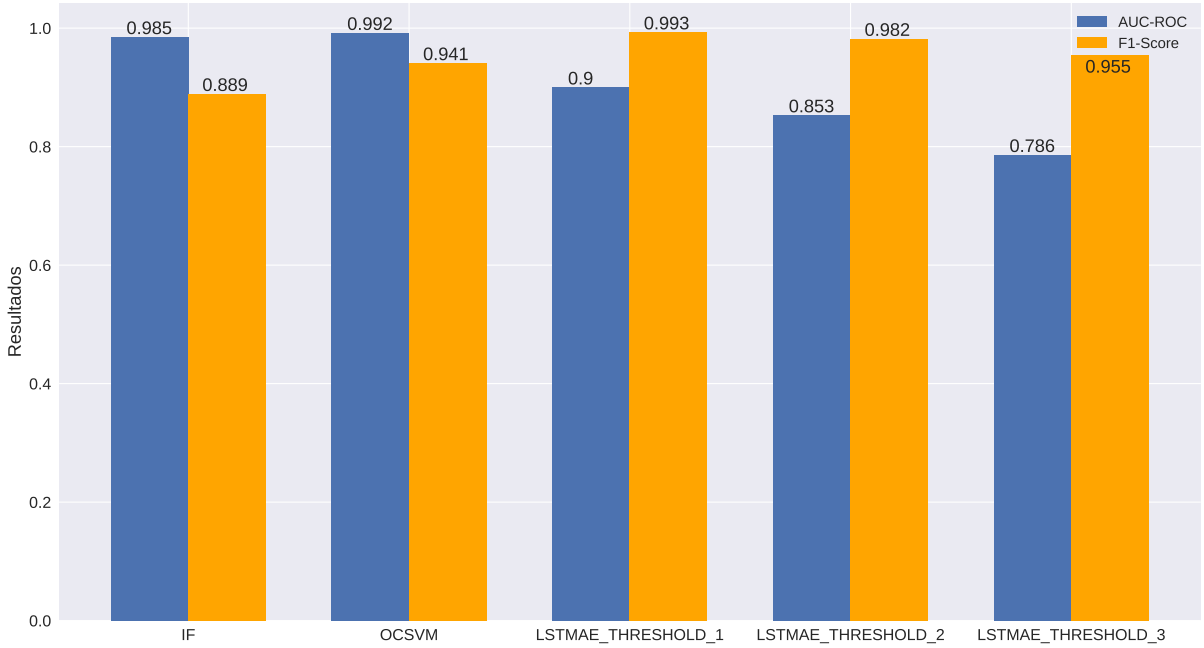
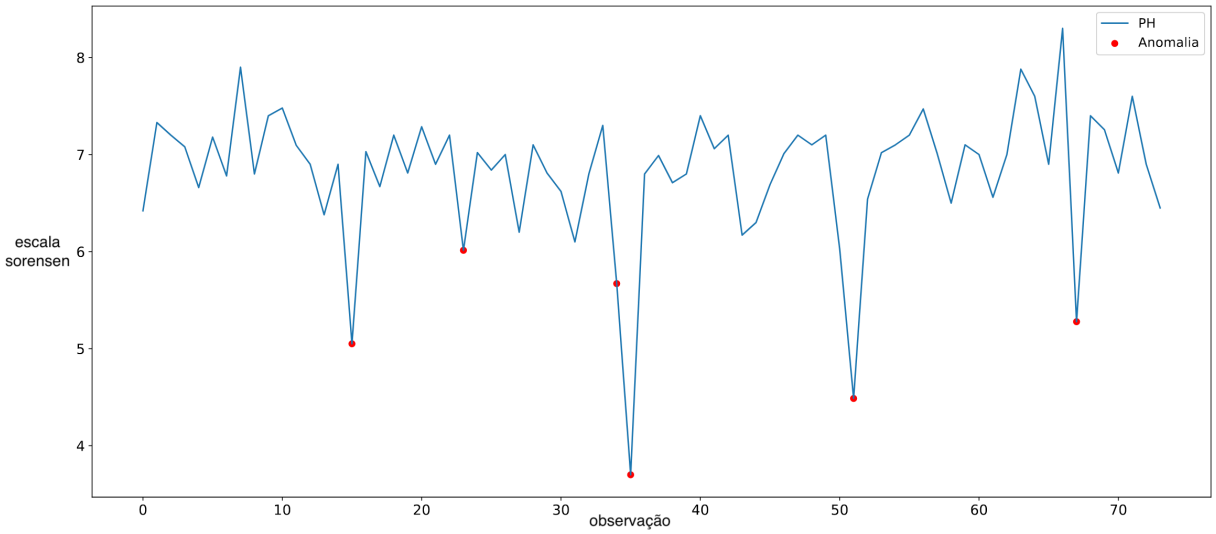



Figura 5.5: Comparação dos melhores resultados obtidos em cada modelo do pH

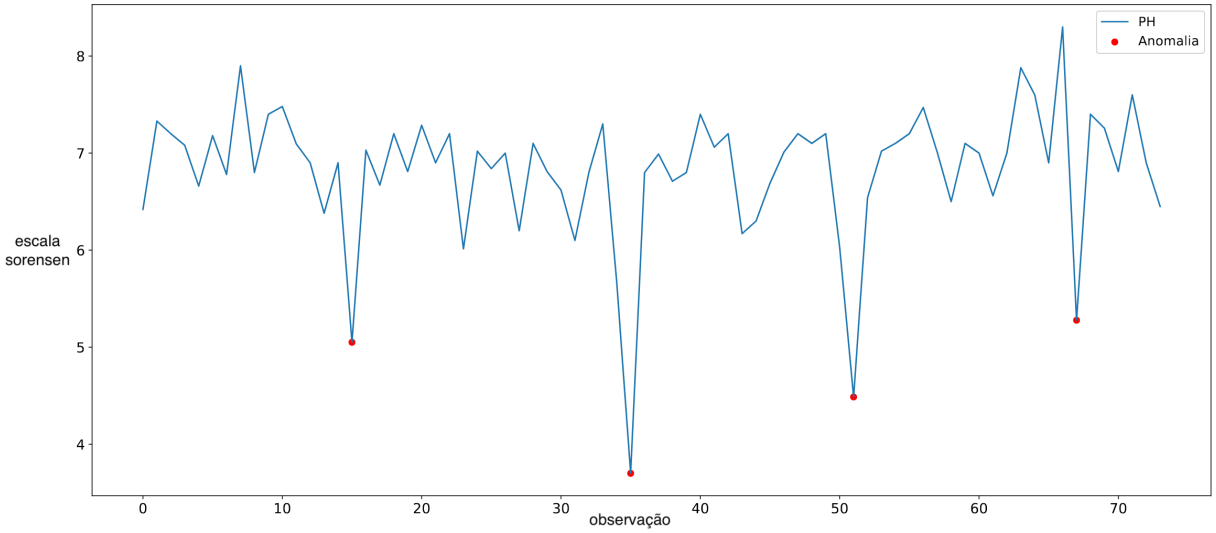
Realizada a comparação dos melhores resultados obtidos entre todos os modelos, o passo seguinte, centra-se na comparação do melhor modelo de ML tradicional com o melhor modelo de DL onde, como é possível confirmar na Figura 5.5, a comparação é entre o OCSVM e a LSTM-AE com a aplicação do *threshold* 1.

Assim sendo, na Figura 5.6 são apresentadas duas sub-figuras onde, o gráfico (a), representa os registos que foram identificados como anómalos por parte do modelo do OCSVM e, no gráfico (b), os registos identificados como anómalos pelo modelo da LSTM-AE com a aplicação do *threshold* 1. Posto isto, conseguimos reiterar algumas conclusões acerca destas sub-figuras, começando, à partida, com a quantidade de anomalias identificadas, onde, no modelo de ML tradicional, foram identificadas 6 anomalias e, no modelo de DL, foram identificadas apenas 4. Com base no processo de labelização que foi realizado com ajuda de especialistas, existe um limite superior de 9, e um limite inferior de 6, ou seja, das 6 anomalias identificadas pelo OCSVM, 5 foram corretamente identificadas (TP), e 1 foi incorretamente identificada como anómala (FP). No caso da LSTM-AE com a aplicação do *threshold* 1, foram identificadas 4 anomalias e todas elas estão corretamente identificadas (TP), ou seja, não existiram FP. É ainda interessante mencionar que, no caso FP identificado com o modelo do OCSVM, é uma observação com o valor de 6.01, ou seja, embora esteja dentro dos padrões definidos pelos especialistas, está extremamente perto de se tornar um evento anómalo, e a sua notificação a quem de direito, será recomendada e benéfica.

Em suma, embora o modelo da LSTM-AE como a aplicação do *threshold* 1 não tenha identificado qualquer FP, é de notar que identificou menos TP e, neste contexto, é mais importante a eventualidade de um alerta que, na realidade, poderá não ser um evento anómalo, que um evento anómalo para o qual não houve qualquer alerta, assim sendo, tal como nos resultados presentes na Figura 5.5, o modelo tradicional obteve resultados ligeiramente superiores ao modelo de DL com o conjunto de dados do pH.




(a) OCSVM




(b) LSTM-AE Threshold 1

Figura 5.6: Comparação melhor modelo ML tradicional com melhor modelo DL do pH

Capítulo 6

# Conclusão e Trabalho Futuro

Com uma projeção da população mundial de 9.8 mil milhões para 2050, a emergência com a preservação e proteção dos recursos naturais urge. Através deste crescimento populacional, e consequente diminuição dos recursos naturais, existem diariamente notícias que relatam a escassez dos bens essenciais nas mais variadas zonas do planeta. Um destes recursos indispensáveis, é a água, e os níveis de seca continuam a atingir níveis recorde ano após ano. Com isto, o trabalho das ETARs, é mais importante que nunca, permitindo que as AR sejam reutilizadas, prevenindo o desperdício de água e reduzindo a poluição deste bem tão valioso para todos nós. Assim sendo, esta dissertação visou contribuir positivamente no trabalho realizado por estas infraestruturas através da deteção de anomalias nas AR. Esta deteção de anomalias, permite que os colaboradores destas instituições consigam realizar o processo de tomada de decisão antecipadamente, prevenindo a possível libertação de substâncias com valores indesejados nas ART para o meio ambiente.

Numa primeira fase, foi realizada uma investigação sobre informação relativa ao tema de forma a aprofundar o conhecimento nesta área. Foi crucial a compreensão do problema global que enfrentamos com a poluição ambiental, obter conhecimento sobre o CUA, bem como a importância que as ETARs têm no combate desta adversidade. Além disso, o estudo da estrutura e fases de tratamento presentes nas ETARs, revelou-se essencial para depreender o tratamento das substâncias dentro destas infraestruturas. Com isto, existiu à priori uma investigação exaustiva dos estudos existentes relacionados com deteção de anomalias nas ETARs, o que permitiu percecionar as soluções implementadas e resultados obtidos, porém, foram também identificadas algumas lacunas, tais como, implementação de *cross validation* e otimização de hiperparâmetros, que foram concretizadas no decorrer desta dissertação.

De seguida, em relação aos dados do Controlo Analítico que foram obtidos, de entre as diversas fases de tratamento existentes, foi selecionada a última fase de tratamento antes da libertação das ART para o meio ambiente, o Efluente Tratado, devido à importância da deteção de anomalias presentes antes das ART saírem das instalações. Nesta etapa da ETAR, foi realizada uma exploração dos dados que se

demonstrou crucial no decorrer da manipulação dos dados e consecutiva elaboração de experiências. Nesta análise, foi detetada a forte correlação entre a Amónia e o Azoto Total, o que permite concluir que a alteração de valor de um destes indicadores, têm impacto direto no valor do outro. Porém, neste caso em específico, a Amónia não foi utilizada na classificação de valores do Azoto Total devido à elevada percentagem de *missing values*. No caso do Azoto Total, verificou-se uma relação sazonal onde os valores deste indicador decresceram ao longo dos anos na Primavera, Outono e Verão. No entanto, no Inverno a tendência foi inversa, verificando-se um aumento constante ao longo dos anos. Este cenário, foi encontrado também nos Nitratos onde os valores aumentaram em todas as estações ao longo dos anos, exceto na Primavera onde ocorreu o oposto.

De forma a aplicar diferentes conjuntos de modelos de deteção de anomalias, foram utilizadas três substâncias para submeter ás experiências, nomeadamente, Azoto Total, Nitratos e pH. Para estes indicadores, foi realizada a manipulação e consequente preparação dos conjuntos de dados que se revelou bastante importante para a obtenção de bons resultados no processo de ML. Com isto, foram concebidos e implementados diversos cenários experimentais com os modelos de ML elegidos, em particular, iF, OCSVM e LSTM-AE. No caso da LSTM-AE, de forma a detetar anomalias é necessária a utilização de *thresholds*. Posto isto, foram utilizadas três abordagens de *thresholds*, das quais, duas consistiam num valor estático e uma num valor dinâmico. A utilização de *thresholds* dinâmicos objetiva a evitar valores manuais bem como remover ruído e alertas desnecessários.

Relativamente à deteção de anomalias no Azoto Total, verificou-se que os modelos candidatos de ML tradicional (iF e OCSVM) obtiveram resultados significativamente melhores que os modelos candidatos de LSTM-AE, com o melhor modelo candidato de iF a atingir 0.975 de AUC-ROC e 0.941 de *f1-score*. No caso dos *thresholds*, o *threshold* 1 não obteve resultados devido a não existirem erros de reconstrução com os dados de teste superiores aos erros obtidos com o conjunto de dados de treino. Já o *threshold* dinâmico, neste contexto em concreto, obteve o melhor resultado de entre os modelos de LSTM-AE. Ainda sobre este indicador, foi possível validar que o melhor modelo (iF) quando submetido a um conjunto de dados de teste, conseguiu identificar todas as anomalias presentes.

No que diz respeito aos Nitratos, a situação foi inversa ao indicador anterior. Os melhores modelos candidatos baseados em LSTM-AE obtiveram melhores performances quando comparados com os modelos de iF e OCSVM. Neste cenário, o melhor resultado pertenceu à aplicação do *threshold* estático 2 onde foi obtido 0.995 tanto de AUC-ROC como de *f1-score*. Foi ainda percetível que, neste caso em específico, o *threshold* dinâmico (3), obteve resultados significativamente inferiores aos dois *thresholds* estáticos. Uma outra conclusão retirada deve-se à diferença considerável do valor de *f1-score* entre os modelos de iF e OCSVM com os modelos de LSTM-AE, que se deveu à classificação de FP em maior quantidade, reduzindo o valor final da métrica. Por fim, em relação ás anomalias detetadas no conjunto de dados de teste, ambos os melhores modelos candidatos conseguiram detetar todas as anomalias existentes.

O pH foi o último indicador sujeito ás diversas experiências levadas a cabo nesta dissertação. Nesta substância repetiu-se o sucedido com o Azoto Total, onde os modelos baseados em iF e OCSVM atingiram resultados significativamente mais elevados. O modelo candidato com melhor *performance* foi o

OCSVM onde a AUC-ROC foi de 0.992 e a *f1-score* de 0.941. Mais concretamente sobre os *thresholds*, verificou-se que os *thresholds* estáticos obtiveram melhores resultados que o *threshold* dinâmico. Por último, no caso deste indicador em específico, foi elaborado um caso de estudo que fez parte da *International Conference on Hybrid Artificial Intelligence Systems* de 2021, onde os resultados foram apresentados a todos os membros desta conferência [116].

Assim sendo, existiam algumas lacunas relacionadas com a periodicidade e escassez dos dados que, por exemplo, no caso da Amónia, afetaram a utilização de abordagens *multivariate*. No entanto, os resultados foram bastante satisfatórios com, de forma global, os modelos baseados nos algoritmos de iF e OCSVM a serem relativamente mais consistentes. No caso específico dos *thresholds*, é interessante assinalar que o *threshold* dinâmico obteve uma vez os melhores resultados de entre os restantes *thresholds*, o que indica que, neste cenário em específico, a remoção de ruído beneficiou o resultado final.

Em síntese, através de todos os resultados obtidos, é possível concluir que um sistema que incorpore os modelos previamente mencionados, é capaz de detetar com sucesso eventos anómalos que noutras circunstâncias poderiam passar despercebidos. Com estes processos, é possível alertar antecipadamente os responsáveis das ETARs de forma a que a tomada de decisão seja mais célere, prevenindo eventuais danos para a sociedade e meio ambiente.

Consequentemente, o trabalho futuro centrar-se-á em:

- Com um aumento nos conjuntos de dados, bem como a frequência em que são recolhidos, poderia ser possível a identificação de novos padrões bem como a melhoria dos modelos concebidos.

- Aplicar a manipulação de dados e consequentes experiências a mais indicadores presentes nestas infraestruturas, permitindo desta forma que o sistema consiga alertar os colaboradores das ETARs sobre uma maior variedade de substâncias.

- Implementação de um sistema híbrido de previsão e deteção de anomalias, mais concretamente, conceção de modelos de regressão que calculariam os valores futuros dos indicadores. Estes mesmos valores seriam identificados como anómalos ou normais pelos modelos de deteção de anomalias. Isto permitiria aumentar significativamente a janela temporal para a tomada de decisão de forma a agir antes do evento anómalo efetivamente se suceder.

- Execução de um sistema de deteção de anomalias híbrido que une modelos como o OCSVM com, por exemplo, *autoencoders* ou *Naive Bayes*. Estas abordagens unem as previsões dos vários modelos atribuindo-lhes pesos. Com isto, a classificação é ponderada e reduz as ocorrências de falsos alertas.

# Bibliografia


[1] T. Kuhlman e J. Farrington. "What is sustainability?" Em: *Sustainability* 2.11 (2010), pp. 3436–3448. doi: 10.3390/su2113436.

[2] B. Ki-moon. "Sustainability—engaging future generations now". Em: *The Lancet* 387.10036 (2016), pp. 2356–2358. doi: 10.1016/S0140-6736(16)30271-9.

[3] A. Mukherjee, N. H. Kamarulzaman, G. Vijayan e S. Vaiappuri. "Sustainability: A Comprehensive Literature". Em: jan. de 2016, pp. 248–268. isbn: 9781466696396. doi: 10.4018/978-1-4666-9639-6.ch015.

[4] R Carr, U Blumenthal e D. Mara. "Guidelines for the Safe Use of Wastewater in Agriculture: Revisiting WHO Guidelines". Em: *Water science and technology : a journal of the International Association on Water Pollution Research* 50 (fev. de 2004), pp. 31–8. doi: 10.2166/wst.2004.0081.

[5] M. Kummu, J. H. Guillaume, H. de Moel, S. Eisner, M. Flörke, M. Porkka, S. Siebert, T. I. Veldkamp e P. Ward. "The world's road to water scarcity: shortage and stress in the 20th century and pathways towards sustainability". Em: *Scientific reports* 6.1 (2016), pp. 1–16. doi: 10.1038/srep38495.

[6] A. Fernandes, R Ribeiro, S Rodrigues e A. Fernandes. "Relatório do Estado do Ambiente Portugal". Em: *Agência Portuguesa Do Ambiente* (2018).

[7] Z. Aghalari, H.-U. Dahms, M. Sillanpää, J. E. Sosa-Hernandez e R. Parra-Saldívar. "Effectiveness of wastewater treatment systems in removing microbial agents: a systematic review". Em: *Globalization and health* 16.1 (2020), pp. 1–11. doi: 10.1186/s12992-020-0546-y.

[8] S. C. Jhansi e S. K. Mishra. "Wastewater treatment and reuse: sustainability options". Em: *Consilience* 10 (2013), pp. 1–15.

[9] V. Jatana. "Machine Learning Algorithms". Em: (2019).

[10] C. Rao e B. Yan. "Study on the interactive influence between economic growth and environmental pollution". Em: *Environmental Science and Pollution Research* 27.31 (2020), pp. 39442–39465. doi: 10.1007/s11356-020-10017-6.

[11] S. Bećirović, S. Ibro e B. Kalač. "Environmental pollution and waste management". Em: *Balkan Journal of Health Science* 03 (jan. de 2015), pp. 2–10.

[12] R. R. Appannagari. “Environmental pollution causes and consequences: a study”. Em: *North Asian International Research Journal of Social Science and Humanities* 3.8 (2017), pp. 151–161.

[13] J. Cleland. “World Population Growth; Past, Present and Future”. Em: *Environmental and Resource Economics* 55 (ago. de 2013). doi: 10.1007/s10640-013-9675-6.

[14] H. Eren. “Impact of Technology on Environment”. Em: jan. de 2002. isbn: 978-0471139461.

[15] E. Oladipo. “Global impact of environmental sustainability on deforestation”. Em: *International Journal of Scientific and Engineering Research* 6.9 (2015), pp. 103–115.

[16] M Kabir, U. Habiba, M. Zafar, M Shafiq e Z. Farooqi. “Industrial pollution and its impacts on ecosystem: A Review. Bioscience Research”. Em: *Journal by Innovative Scientific Information & Services Network* 17.2 (2020), pp. 1364–1372.

[17] S Uttara, N. Bhuvandas, V. Aggarwal et al. “Impacts of urbanization on environment”. Em: *International Journal of Research in Engineering and Applied Sciences* 2.2 (2012), pp. 1637–1645.

[18] J. Barnes, J. Bender, L. TM e B. AM. “Natural and man-made selection for air pollution resistance”. Em: *Journal of Experimental Botany* 50 (set. de 1999). doi: 10.1093/jexbot/50.338.1423.

[19] N. M. Aljamali, A. M. Jawad e A. Jawad. “A Literature Review on Types of Contamination (Biological, Chemical, Medical)”. Em: *International Journal of International Journal of Green Chemistry* 5.1 (2019), pp. 7–14.

[20] M. Sierra-Vargas e L. Teran. “Air pollution: Impact and prevention”. Em: *Respirology (Carlton, Vic.)* 17 (jun. de 2012), pp. 1031–8. doi: 10.1111/j.1440-1843.2012.02213.x.

[21] A. Inyinbor, B. Adebesin, A. Oluyori, T. Adelani-Akande, A. O. Dada e O. A. “Water Pollution: Effects, Prevention, and Climatic Impact”. Em: mar. de 2018. isbn: 978-953-51-3893-8. doi: 10.5772/intechopen.72018.

[22] O. Shaltami, N. Hamed, F. Fares, H. Errishi, F. El Oshebi e E. Maceda. “Soil pollution – A review”. Em: out. de 2020.

[23] M. Elliott. “Biological pollutants and biological pollution - An increasing cause for concern”. Em: *Marine pollution bulletin* 46 (abr. de 2003), pp. 275–80. doi: 10.1016/S0025-326X(02)00423-X.

[24] V. Egorov, S. Gulin, N. Y. Mirzoyeva, G. Polikarpov, N. Stokozov, G. Laptev, O. Voitsekhovych e A. Nikitin. “The state of radioactive pollution”. Em: *Book: State of the Environment of the Black Sea (2001-2006/7). Edited by Temal Oguz. Publications of the Commission on the Protection of the Black Sea Against Pollution (BSC)* 3 (2008), pp. 163–172.

[25] K. Aayush, D. Vishal, N. Hammad e M. Ks. “Application of Artificial Intelligence in Curbing Air Pollution: The Case of India”. Em: *Asian Journal of Management* 11 (jan. de 2020), p. 285. doi: 10.5958/2321-5763.2020.00044.X.

[26] *Computer scientists develop novel artificial intelligence system that predicts air pollution levels*. https://www.lboro.ac.uk/news-events/news/2020/march/artificial-intelligence-system-air-pollution/. Acedido: 2020-12-27.

[27] R. Prasad, A. Bhattacharyya e Q. D. Nguyen. “Nanotechnology in sustainable agriculture: recent developments, challenges, and perspectives”. Em: *Frontiers in microbiology* 8 (2017), p. 1014. doi: 10.3389/fmicb.2017.01014.

[28] *https://www.forbes.com/sites/cognitiveworld/2019/09/04/how-iot-and-ai-can-enable-environmental-sustainability/?sh=2b8e5e7568df*. https://www.lboro.ac.uk/news-events/news/2020/march/artificial-intelligence-system-air-pollution/. Acedido: 2020-12-28.

[29] P. Wang, J. Yao, G. Wang, F. Hao, S. Shrestha, B. Xue, G. Xie e Y. Peng. “Exploring the application of artificial intelligence technology for identification of water pollution characteristics and tracing the source of water quality pollutants”. Em: *Science of the Total Environment* 693 (2019), p. 133440. doi: 10.1016/j.scitotenv.2019.07.246.

[30] S. B. Aher. “WATER POLLUTION: CAUSES, EFFECTS AND MANAGEMENT”. Em: jan. de 2019, pp. 194–200. isbn: 978-81-923937-8-0.

[31] R. I. M. N. Leitao. “Sustentabilidade na gestão do ciclo urbano da água: Simulação e análise de cenários”. Tese de doutoramento. Universidade de Coimbra, 2014.

[32] R. I. M. N. Leitao. “Sustentabilidade na gestão do ciclo urbano da água: Simulação e análise de cenários”. Tese de doutoramento. Universidade de Coimbra, 2014.

[33] M. Oliveira, I. Serrano, S. Harten, L. Bessa, B. Fernando e P. Costa. “Fecal contamination of wastewater treatment plants in Portugal”. Em: *Environmental Science and Pollution Research* 23 (jul. de 2016). doi: 10.1007/s11356-016-6962-0.

[34] P. Oliveira, B. Fernandes, C. Analide e P. Novais. “Forecasting energy consumption of wastewater treatment plants with a transfer learning approach for sustainable cities”. Em: *Electronics* 10.10 (2021), p. 1149. doi: 10.3390/electronics10101149.

[35] C. Simões, I. Rosmaninho e A. G. Henriques. “Guia para a Avaliação de Impacte Ambiental de Estações de Tratamento de Águas Residuais”. Em: *Agência Portuguesa do Ambiente. Instituto regulador de Águas e Resíduos. Lisboa* (2008).

[36] A. de Waal. “Specializing CRISP-DM for Evidence Mining”. Em: nov. de 2007. doi: 10.1007/978-0-387-73742-3_21.

[37] Z. Bosnjak, O. Grljevic e S. Bošnjak. “CRISP-DM as a framework for discovering knowledge in small and medium sized enterprises’ data”. Em: jun. de 2009, pp. 509 –514. doi: 10.1109/SACI.2009.5136302.

[38] H. Wang, C. Ma e L. Zhou. “A brief review of machine learning and its application”. Em: *2009 international conference on information engineering and computer science*. IEEE. 2009, pp. 1–4. doi: 10.1109/ICIECS.2009.5362936.

[39] T. M. Mitchell e T. M. Mitchell. *Machine learning*. Vol. 1. 9. McGraw-hill New York, 1997.

[40] J. P. Monteiro, D. Ramos, D. Carneiro, F. Duarte, J. M. Fernandes e P. Novais. “Meta-learning and the new challenges of machine learning”. Em: *International Journal of Intelligent Systems* 36.11 (2021), pp. 6240–6272. doi: 10.1002/int.22549.

[41] J. Carneiro, P. Saraiva, D. Martinho, G. Marreiros e P. Novais. “Representing decision-makers using styles of behavior: An approach designed for group decision support systems”. Em: *Cognitive Systems Research* 47 (2018), pp. 109–132. doi: 10.1016/j.cogsys.2017.09.002.

[42] I. H. Sarker. “Machine learning: Algorithms, real-world applications and research directions”. Em: *SN Computer Science* 2.3 (2021), pp. 1–21. doi: 10.1007/s42979-021-00592-x.

[43] J. Alzubi, A. Nayyar e A. Kumar. “Machine learning from theory to algorithms: an overview”. Em: *Journal of physics: conference series*. Vol. 1142. 1. IOP Publishing. 2018, p. 012012. doi: 10.1088/1742-6596/1142/1/012012.

[44] N. Washani e S. Sharma. “Speech Recognition System: A Review”. Em: *International Journal of Computer Applications* 115 (abr. de 2015), pp. 7–10. doi: 10.5120/20249-2617.

[45] J. Anderson, N. Kalra, K. Stanley, P. Sorensen, C. Samaras e T. Oluwatola. *Autonomous Vehicle Technology: A Guide for Policymakers*. Jan. de 2014. isbn: 9780833083982.

[46] V Christina, S Karpagavalli e G Suganya. “A Study on Email Spam Filtering Techniques”. Em: *International Journal of Computer Applications* 12 (dez. de 2010). doi: 10.5120/1645-2213.

[47] R. Domingues C. e Coimbra. “New Trauma and Injury Severity Score (TRISS) adjustments for survival prediction”. Em: *World J Emerg Surg* 13 (mar. de 2018). doi: 10.1186/s13017-018-0171-8.

[48] G. D. Magoulas e A. Prentza. “Machine learning in medical applications”. Em: *Advanced course on artificial intelligence*. Springer. 1999, pp. 300–307. doi: 10.1007/3-540-44673-7_19.

[49] J.-W. Dam e M. Velden. “Online profiling and clustering of Facebook users”. Em: *Decision Support Systems* 70 (fev. de 2015), pp. 60 –72. doi: 10.1016/j.dss.2014.12.001.

[50] C. Gómez-Uribe e N. Hunt. “The Netflix Recommender System”. Em: *ACM Transactions on Management Information Systems* 6 (dez. de 2015), pp. 1–19. doi: 10.1145/2843948.

[51] M Kaiser, L Camarinha-Matos, A Giordana, V Klingspor, J. d. R. Millan, F. De Natale, M Nuttin, R Suarez e I. R. Dillmann. “Robot learning-three case studies in robotics and machine learning”. Em: *networks* 17.63 (1994), p. 50.

[52] D. K. Bhattacharyya e J. Kalita. *Network Anomaly Detection: A Machine Learning Perspective*. Abr. de 2013. isbn: 9781466582088-K18917. doi: 10.1201/b15088.

[53] T. Ayodele. “Introduction to Machine Learning”. Em: fev. de 2010. isbn: 978-953-307-034-6. doi: 10.5772/9394.

[54] T. Ayodele. “Types of Machine Learning Algorithms”. Em: fev. de 2010. isbn: 978-953-307-034-6. doi: 10.5772/9385.

[55] P. Cunningham, M. Cord e S. Delany. “Supervised Learning”. Em: jan. de 2008, pp. 21–49. isbn: 978-3-540-75170-0. doi: 10.1007/978-3-540-75171-7_2.

[56] Q. Liu e Y. Wu. “Supervised Learning”. Em: (jan. de 2012). doi: 10.1007/978-1-4419-1428-6_451.

[57] R. Caruana e A. Niculescu-Mizil. “An empirical comparison of supervised learning algorithms”. Em: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 161–168. doi: 10.1145/1143844.1143865.

[58] N. Sebe, I. Cohen e A. Garg. *Machine Learning in Computer Vision*. Vol. 29. Jan. de 2005. isbn: 978-1-4020-3274-5. doi: 10.1007/1-4020-3275-7.

[59] M. Mohammed, M. B. Khan e E. B. M. Bashier. *Machine learning: algorithms and applications*. Crc Press, 2016. doi: 10.1201/9781315371658.

[60] E. G. Dada, J. S. Bassi, H. Chiroma, A. O. Adetunmbi, O. E. Ajibuwa et al. “Machine learning for email spam filtering: review, approaches and open research problems”. Em: *Heliyon* 5.6 (2019), e01802. doi: 10.1016/j.heliyon.2019.e01802.

[61] P. Larranaga. “Machine learning in bioinformatics”. Em: *Briefings in Bioinformatics* 7 (fev. de 2006), pp. 86–112. doi: 10.1093/bib/bbk007.

[62] S. Eltanbouly, M. Bashendy, N. AlNaimi, Z. Chkirbene e A. Erbad. “Machine learning techniques for network anomaly detection: A survey”. Em: *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. IEEE. 2020, pp. 156–162. doi: 10.1109/ICIoT48696.2020.9089465.

[63] H. B. Barlow. “Unsupervised learning”. Em: *Neural computation* 1.3 (1989), pp. 295–311. doi: 10.1162/neco.1989.1.3.295.

[64] D. Greene, P. Cunningham e R. Mayer. “Unsupervised Learning and Clustering”. Em: jan. de 2008, pp. 51–90. isbn: 9783540751700. doi: 10.1007/978-3-540-75171-7-3.

[65] P. H. Braga e H. F. Bassani. “A semi-supervised self-organizing map for clustering and classification”. Em: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pp. 1–8. doi: 10.1109/IJCNN.2018.8489675.

[66] M. Z. Rodriguez, C. H. Comin, D. Casanova, O. M. Bruno, D. R. Amancio, L. d. F. Costa e F. A. Rodrigues. “Clustering algorithms: A comparative approach”. Em: *PloS one* 14.1 (2019), e0210236. doi: 10.1371/journal.pone.0210236.

[67] M. Camilleri. “Market Segmentation, Targeting and Positioning”. Em: dez. de 2017. isbn: ISBN 978-3-319-49849-2. doi: 10.1007/978-3-319-49849-2_4.

[68] D. Jiang, C. Tang e A. Zhang. “Cluster Analysis for Gene Expression Data: A Survey”. Em: *Knowledge and Data Engineering, IEEE Transactions on* 16 (dez. de 2004), pp. 1370–1386. doi: 10.1109/TKDE.2004.68.

[69] R. A. Ariyaluran Habeeb, F. Nasaruddin, A. Gani, M. A. Amanullah, I. Hashem, E. Ahmed e M. Imran. “Clustering□based real□time anomaly detection—A breakthrough in big data technologies”. Em: *Transactions on Emerging Telecommunications Technologies* (jun. de 2019), e3647. doi: 10.1002/ett.3647.

[70] P. K. Singh, P. K. D. Pramanik, A. K. Dey e P. Choudhury. “Recommender systems: an overview, research trends, and future directions”. Em: *International Journal of Business and Systems Research* 15.1 (2021), pp. 14–52.

[71] C. Szepesvári. “Algorithms for reinforcement learning”. Em: *Synthesis lectures on artificial intelligence and machine learning* 4.1 (2010), pp. 1–103. doi: 10.2200/S00268ED1V01Y.201005AIM009.

[72] K.-L. Du e M. Swamy. “Reinforcement learning”. Em: *Neural Networks and Statistical Learning*. Springer, 2019, pp. 503–523. doi: 10.1007/978-1-4471-5571-3_18.

[73] S. Mousavi, M. Schukat e E. Howley. “Deep Reinforcement Learning: An Overview”. Em: jun. de 2018, pp. 426–440. isbn: 978-3-319-56990-1. doi: 10.1007/978-3-319-56991-8_32.

[74] H. Van Hasselt, A. Guez e D. Silver. “Deep reinforcement learning with double q-learning”. Em: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1. 2016. doi: 10.1609/aaai.v30i1.10295.

[75] J. Kober, J. Bagnell e J. Peters. “Reinforcement Learning in Robotics: A Survey”. Em: *The International Journal of Robotics Research* 32 (set. de 2013), pp. 1238–1274. doi: 10.1177/0278364913495721.

[76] S. Chakraborty. “Capturing financial markets to apply deep reinforcement learning”. Em: *arXiv preprint arXiv:1907.04373* (2019). doi: 10.48550/arXiv.1907.04373.

[77] A. Folkers, M. Rick e C. Büskens. “Controlling an Autonomous Vehicle with Deep Reinforcement Learning”. Em: jun. de 2019. doi: 10.1109/IVS.2019.8814124.

[78] S. R. Krishnan, A. Arul, S Sivakumari e R. Scholar. “Study on Machine Learning Techniques for Anomaly Detection”. Em: abr. de 2020.

[79] T. Veasey e S. Dodson. “Anomaly Detection in Application Performance Monitoring Data”. Em: *International Journal of Machine Learning and Computing* 4 (abr. de 2014), pp. 120–126. doi: 10.7763/IJMLC.2014.V4.398.

[80] H. Hameed, S. Mazhar e N. Hassan. “Real-time road anomaly detection, using an on-board data logger”. Em: *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE. 2018, pp. 1–5. doi: 10.1109/VTCSpring.2018.8417780.

[81] M. Dinç, S. Ertekin, H. Özkan, S. Meydanlı e M. V. Atalay. “Forecasting of Product Quality Through Anomaly Detection”. Em: mar. de 2020, pp. 357–366. isbn: 978-3-030-43886-9. doi: 10.1007/978-3-030-43887-6_29.

[82] S. Omar, M. Ngadi, H. Jebur e S. Benqdara. “Machine Learning Techniques for Anomaly Detection: An Overview”. Em: *International Journal of Computer Applications* 79 (out. de 2013). doi: 10.5120/13715-1478.

[83] J. P. Pinto, A. Pimenta e P. Novais. “Deep learning and multivariate time series for cheat detection in video games”. Em: *Machine Learning* 110.11 (2021), pp. 3037–3057. doi: 10.1007/s10994-021-06055-x.

[84] G. A. Susto, A. Beghi e S. McLoone. “Anomaly detection through on-line isolation forest: An application to plasma etching”. Em: *2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*. IEEE. 2017, pp. 89–94. doi: 10.1109/ASMC.2017.7969205.

[85] F. T. Liu, K. M. Ting e Z.-H. Zhou. “Isolation forest”. Em: *2008 eighth ieee international conference on data mining*. IEEE. 2008, pp. 413–422. doi: 10.1109/ICDM.2008.17.

[86] O. Chapelle, P. Haffner e V. N. Vapnik. “Support vector machines for histogram-based image classification”. Em: *IEEE transactions on Neural Networks* 10.5 (1999), pp. 1055–1064.

[87] Y.-S. Choi. “Least squares one-class support vector machine”. Em: *Pattern Recognition Letters* 30.13 (2009), pp. 1236–1240. doi: 10.1016/j.patrec.2009.05.007.

[88] L. M. Manevitz e M. Yousef. “One-class SVMs for document classification”. Em: *Journal of machine Learning research* 2.Dec (2001), pp. 139–154. doi: 10.1162/15324430260185574.

[89] Y. Ma, A. Maqsood, K. Corzine e D. Oslebo. “Long short-term memory autoencoder neural networks based dc pulsed load monitoring using short-time fourier transform feature extraction”. Em: *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*. IEEE. 2020, pp. 912–917. doi: 10.1109/ISIE45063.2020.9152477.

[90] Y. Chu, J. Fei e S. Hou. “Adaptive global sliding-mode control for dynamic systems using double hidden layer recurrent neural network structure”. Em: *IEEE transactions on neural networks and learning systems* 31.4 (2019), pp. 1297–1309. doi: 10.1109/TNNLS.2019.2919676.

[91] R. Pascanu, T. Mikolov e Y. Bengio. "On the difficulty of training recurrent neural networks". Em: *International conference on machine learning*. PMLR. 2013, pp. 1310–1318. doi: 10.48550/arXiv.1211.5063.

[92] A. Graves. "Long short-term memory". Em: *Supervised sequence labelling with recurrent neural networks* (2012), pp. 37–45. doi: 10.1007/978-3-642-24797-2.

[93] P. Oliveira, B. Fernandes, F. Aguiar, M. A. Pereira, C. Analide e P. Novais. "A deep learning approach to forecast the influent flow in wastewater treatment plants". Em: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer. 2020, pp. 362–373. doi: 10.1007/978-3-030-62362-3_32.

[94] H. D. Trinh, E. Zeydan, L. Giupponi e P. Dini. "Detecting mobile traffic anomalies through physical control channel fingerprinting: A deep semi-supervised approach". Em: *IEEE Access* 7 (2019), pp. 152187–152201. doi: 10.1109/ACCESS.2019.2947742.

[95] 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.

[96] *Diário da República n.º 139/1997, Série I-A de 1997-06-19*. https://data.dre.pt/eli/dec-lei/152/1997/06/19/p/dre/pt/html. Acedido: 2021-01-07.

[97] 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.

[98] *Diário da República n.º 176/1998, Série I-A de 1998-08-01*. https://data.dre.pt/eli/dec-lei/236/1998/08/01/p/dre/pt/html. Acedido: 2021-01-08.

[99] E. Godwill, P. Ferdinand, N. Nwalo e M. Unachukwu. "Mechanism and Health Effects of Heavy Metal Toxicity in Humans". Em: jun. de 2019, pp. 1–23. isbn: 978-1-83880-785-6. doi: 10.5772/intechopen.82511.

[100] B. Mamandipoor, M. Majd, M. Sheikhalishahi, C. Modena e V. Osmani. "Monitoring and detecting faults in wastewater treatment plants using deep learning". Em: *Environmental Monitoring and Assessment* 192 (fev. de 2020). doi: 10.1007/s10661-020-8064-1.

[101] H. Haimi, M. Mulas, F. Corona, S. Marsili-Libelli, P. Lindell, M. Heinonen e R. Vahala. "Adaptive data-derived anomaly detection in the activated sludge process of a large-scale wastewater treatment plant". Em: *Engineering Applications of Artificial Intelligence* 52 (mar. de 2016), pp. 65–80. doi: 10.1016/j.engappai.2016.02.003.

[102] F. Harrou, A. Dairi, Y. Sun e M. Senouci. "Statistical monitoring of a wastewater treatment plant: A case study". Em: *Journal of Environmental Management* 223 (2018), pp. 807 –814. issn: 0301-4797. doi: https://doi.org/10.1016/j.jenvman.2018.06.087. url: http://www.sciencedirect.com/science/article/pii/S0301479718307394.

[103] D. Aguado e C. Rosen. “Multivariate statistical monitoring of continuous wastewater treatment plants”. Em: *Engineering Applications of Artificial Intelligence* 21.7 (2008), pp. 1080 –1091. issn: 0952-1976. doi: https://doi.org/10.1016/j.engappai.2007.08.004. url: http://www.sciencedirect.com/science/article/pii/S0952197607001133.

[104] D GarcÇa-Alvarez. “Fault detection using principal component analysis (PCA) in a wastewater treatment plant (WWTP)”. Em: *Proceedings of the International Student's Scientific Conference*. Vol. 2009. 2009.

[105] M. A. Camargo Valero e D. D. Mara. “Nitrogen removal via ammonia volatilization in maturation ponds”. Em: *Water Science and Technology* 55.11 (2007), pp. 87–92. doi: 10.2166/wst.2007.349.

[106] V. J. Harding e C. T. Potter. “The Excretion of “Acetone” and Nitrogen in Nausea and Vomiting of Pregnancy”. Em: *British journal of experimental pathology* 4.3 (1923), p. 105.

[107] D. G. Bonett e T. A. Wright. “Sample size requirements for estimating Pearson, Kendall and Spearman correlations”. Em: *Psychometrika* 65.1 (2000), pp. 23–28. doi: 10.1007/BF02294183.

[108] V. Goldberg. “Groundwater pollution by nitrates from livestock wastes”. Em: *Environmental Health Perspectives* 83 (1989), pp. 25–29. doi: 10.2307/3430646.

[109] I. S. Kim, M. H. Hwang, N. J. Jang, S. H. Hyun e S. T. Lee. “Effect of low pH on the activity of hydrogen utilizing methanogen in bio-hydrogen process”. Em: *International Journal of Hydrogen Energy* 29.11 (2004), pp. 1133–1140. doi: 10.1016/j.ijhydene.2003.08.017.

[110] H.-E. Gäbler. “Mobility of heavy metals as a function of pH of samples from an overbank sediment profile contaminated by mining activities”. Em: *Journal of Geochemical Exploration* 58.2-3 (1997), pp. 185–194. doi: 10.1016/S0375-6742(96)00061-1.

[111] A. H. Mirza e S. Cosan. “Computer network intrusion detection using sequential LSTM neural networks autoencoders”. Em: *2018 26th signal processing and communications applications conference (SIU)*. IEEE. 2018, pp. 1–4. doi: 10.1109/SIU.2018.8404689.

[112] K. Hundman, V. Constantinou, C. Laporte, I. Colwell e T. Soderstrom. “Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding”. Em: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018, pp. 387–395. doi: 10.1145/3219819.3219845.

[113] A. Santra e C. J. Christy. “Genetic algorithm and confusion matrix for document clustering”. Em: *International Journal of Computer Science Issues (IJCSI)* 9.1 (2012), p. 322.

[114] S. A. Khan e Z. A. Rana. “Evaluating performance of software defect prediction models using area under precision-Recall curve (AUC-PR)”. Em: *2019 2nd International Conference on Advancements in Computational Sciences (ICACS)*. IEEE. 2019, pp. 1–6. doi: 10.23919/ICACS.2019.8689135.

[115] Z. C. Lipton, C. Elkan e B. Narayanaswamy. “Thresholding classifiers to maximize F1 score”. Em: *arXiv preprint arXiv:1402.1892* (2014). doi: 10.48550/arXiv.1402.1892.

[116] D. Gigante, P. Oliveira, B. Fernandes, F. Lopes e P. Novais. “Unsupervised Learning Approach for pH Anomaly Detection in Wastewater Treatment Plants”. Em: *International Conference on Hybrid Artificial Intelligence Systems*. Springer. 2021, pp. 588–599. doi: 10.1007/978-3-030-86271-8_49.

[117] A. Al Shorman, H. Faris e I. Aljarah. “Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection”. Em: *Journal of Ambient Intelligence and Humanized Computing* 11.7 (2020), pp. 2809–2825. doi: 10.1007/s12652-019-01387-y.

[118] H. Nguyen, K. P. Tran, S. Thomassey e M. Hamad. “Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management”. Em: *International Journal of Information Management* 57 (2021), p. 102282. doi: 10.1016/j.ijinfomgt.2020.102282.

[119] T. Carneiro, R. V. M. Da Nóbrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque e P. P. Reboucas Filho. “Performance analysis of google colaboratory as a tool for accelerating deep learning applications”. Em: *IEEE Access* 6 (2018), pp. 61677–61685. doi: 10.1109/ACCESS.2018.2874767.

[120] L. A. Jeni, J. F. Cohn e F. De La Torre. “Facing imbalanced data–recommendations for the use of performance metrics”. Em: *2013 Humaine association conference on affective computing and intelligent interaction*. IEEE. 2013, pp. 245–251. doi: 10.1109/ACII.2013.47.

Apêndice A

# Recolha e Armazenamento dos Dados

Este capítulo contém a *script* utilizada para inserir os dados contidos nos ficheiros *Excel* provenientes da ETAR de Vila Real num modelo de Base de Dados relacional.

```
db = mysql.connector.connect(**config)

def getIndicatorId(indicator_name):
   mycursor = db.cursor()
   sql = "SELECT indicator_table.id FROM indicator_table WHERE indicator_table.
       ↪ indicator_name = '" + indicator_name + "';"
   mycursor.execute(sql)

   myresult = mycursor.fetchall()
   return myresult

def folder_files_to_df(url):
   directory = os.path.join(url)
   for root,dirs,files in os.walk(directory):
      for file in sorted(files):
         if file.endswith(".csv"):

            folder = directory.split('/')[-2] + '/'
            parent_folder = directory.split('/')[-4]

            if parent_folder == 'Funcionamento':
               indicator = directory.split('/')[-3]
               myresult = getIndicatorId(indicator)
               if len(myresult) > 0:
                  id_indicator = myresult[0][0]
```

```
25                        print('Already inserted, has the id:', id_indicator, 'and name',
                              ↪ indicator)
26                    else:
27                        mycursor = db.cursor()
28                        sql = "INSERT INTO indicator_table (indicator_name, description,
                              ↪ indicator_type, alarm_high_value, alarm_low_value, units)
                              ↪ VALUES (%s, %s, %s, %s, %s, %s)"
29                        val = [
30                           (indicator, "", "Funcionamento", "-99", "-99", "h")
31                        ]
32
33                        mycursor.executemany(sql, val)
34                        db.commit()
35                        print(mycursor.rowcount, "record(s) inserted. Inserted the indicator",
                              ↪  indicator)
36
37                        myresult = getIndicatorId(indicator)
38
39                        if len(myresult) > 0:
40                           id_indicator = myresult[0][0]
41                           print('The indicator got the id:', id_indicator, 'and name',
                                 ↪ indicator)
42                else:
43                    indicator = file.split('.csv',1)[0]
44                    myresult = getIndicatorId(indicator)
45
46                    if len(myresult) > 0:
47                        id_indicator = myresult[0][0]
48                        print('Already inserted, has the id:', id_indicator, 'and name',
                              ↪ indicator)
49                    else:
50                        mycursor = db.cursor()
51                        sql = "INSERT INTO indicator_table (indicator_name, description,
                              ↪ indicator_type, alarm_high_value, alarm_low_value, units)
                              ↪ VALUES (%s, %s, %s, %s, %s, %s)"
52                        val = [
53                           (indicator, "", detalhes[folder + file][0], "-99", "-99", detalhes[
                                 ↪ folder + file][4])
54                        ]
55
56                        mycursor.executemany(sql, val)
57                        db.commit()
58                        print(mycursor.rowcount, "record(s) inserted. Inserted the indicator",
                              ↪  indicator)
59
```

```
60                      myresult = getIndicatorId(indicator)
61
62                      if len(myresult) > 0:
63                          id_indicator = myresult[0][0]
64                          print('The indicator got the id:', id_indicator, 'and name',
                                ↪ indicator)
65
66                  temp_data = pd.read_csv(url + file, delimiter=";")
67
68                  val = []
69                  for index, row in temp_data.iterrows():
70                      val.append((id_indicator, detalhes[folder + file][1], detalhes[folder +
                            ↪ file][2], str(row[1]).replace(',','.'), row[0], detalhes[folder +
                            ↪ file][3]))
71
72                  print('Number of values', len(val))
73                  print('Example of value:', val[18])
74
75                  sql = "INSERT INTO indicator_value_table (indicator, sub_type, input, value,
                        ↪  date, city_name) VALUES (%s, %s, %s, %s, %s, %s)"
76
77                  mycursor = db.cursor()
78                  mycursor.executemany(sql, val)
79
80                  db.commit()
81
82                  print(mycursor.rowcount, "record(s) of values for indicator", indicator, "
                        ↪ inserted.\n\n")
```

Listagem A.1: *Script* para Inserção de Dados na Base de Dados

Apêndice B

# Dados Controlo Analítico

Neste capítulo estão presentes Figuras sobre os três indicadores selecionados no decorrer da dissertação. Estes gráficos contem a representação da relação entre os valores do indicador, e os trimestres.

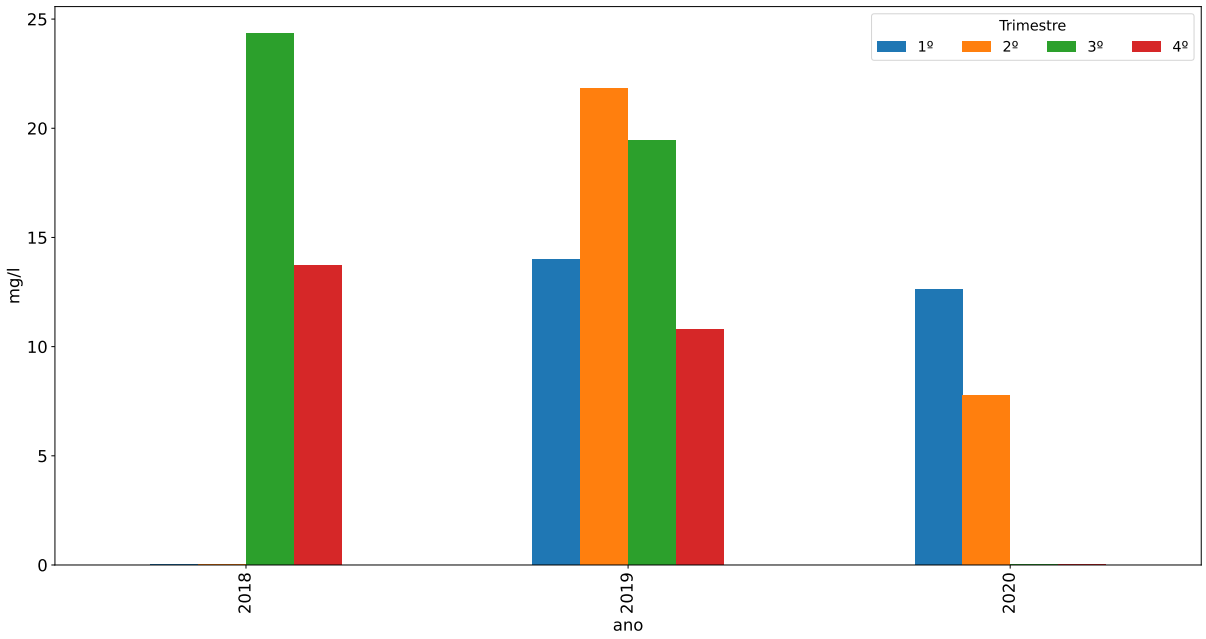



Figura B.1: Histograma do Azoto Total por ano e trimestre

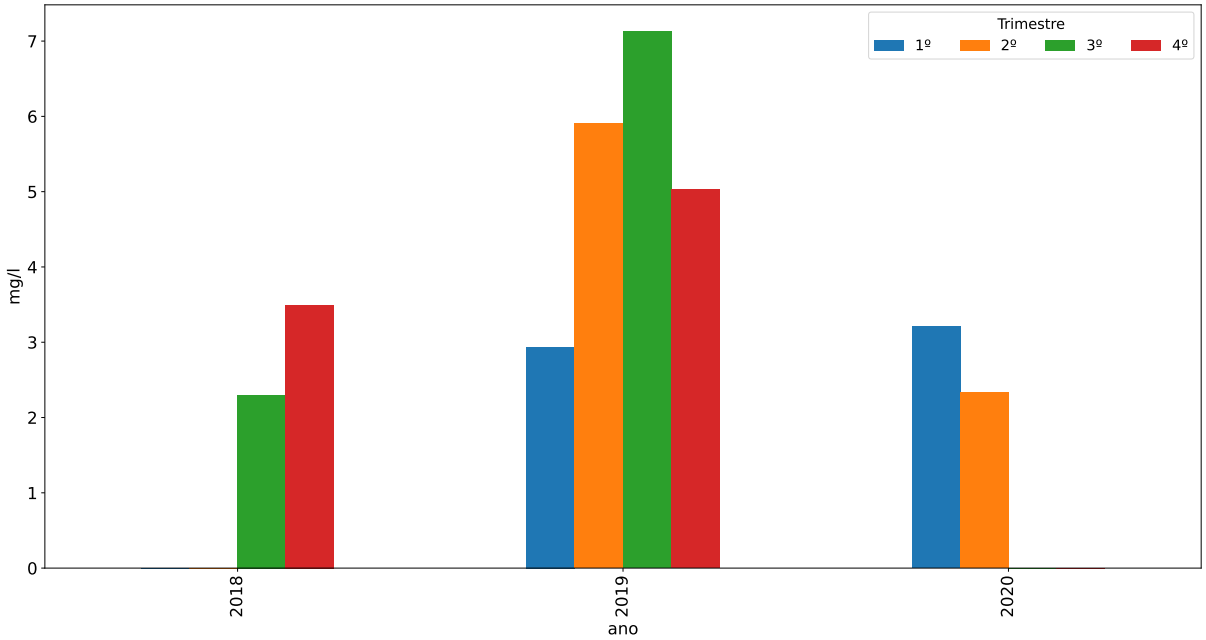



Figura B.2: Histograma dos Nitratos por ano e trimestre

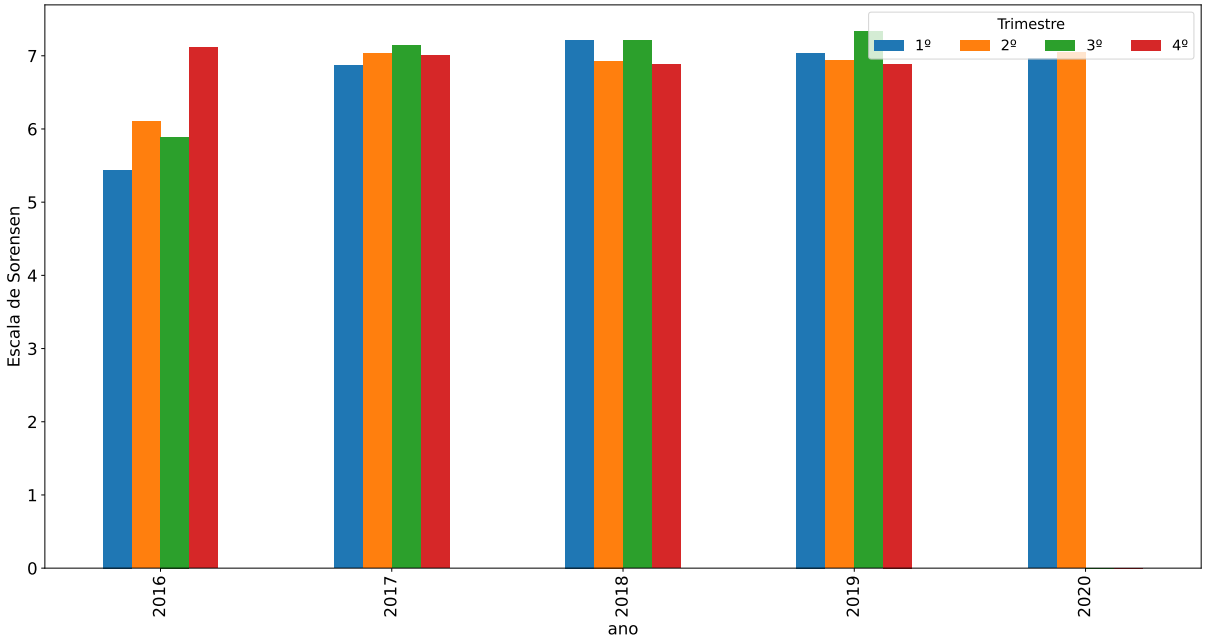



Figura B.3: Histograma do pH por ano e trimestre