

TOMÁS LOPES DE BRITO TEIXEIRA MENDES

**SISTEMA INTELIGENTE PARA DETEÇÃO DE
ANOMALIAS EM CONSUMOS HOTELEIROS**



UNIVERSIDADE DO ALGARVE
Instituto Superior de Engenharia
2022

TOMÁS LOPES DE BRITO TEIXEIRA MENDES

**SISTEMA INTELIGENTE PARA DETEÇÃO DE
ANOMALIAS EM CONSUMOS HOTELEIROS**

**Mestrado em Engenharia Eletrotécnica e de Computadores
Especialização em Sistemas de Energia e Controlo**

**Trabalho efetuado sob a orientação de:
Professor Doutor Jânio Miguel Evangelista Ferreira Monteiro
Professor Doutor Pedro Jorge Sequeira Cardoso**



**UNIVERSIDADE DO ALGARVE
Instituto Superior de Engenharia**

2022

SISTEMA INTELIGENTE PARA DETEÇÃO DE ANOMALIAS EM CONSUMOS HOTELEIROS

Declaração de autoria de trabalho

Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

I hereby declare to be the author of this work, which is original and unpublished. Authors and works consulted are properly cited in the text and included in the reference list.

(Tomás Lopes de Brito Teixeira Mendes)

©2022, TOMÁS LOPES DE BRITO TEIXEIRA MENDES

A Universidade do Algarve reserva para si o direito, em conformidade com o disposto no Código do Direito de Autor e dos Direitos Conexos, de arquivar, reproduzir e publicar a obra, independentemente do meio utilizado, bem como de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição para fins meramente educacionais ou de investigação e não comerciais, conquanto seja dado o devido crédito ao autor e editor respetivos.

The University of the Algarve reserves the right, in accordance with the terms of the Copyright and Related Rights Code, to file, reproduce and publish the work, regardless of the methods used, as well as to publish it through scientific repositories and to allow it to be copied and distributed for purely educational or research purposes and never for commercial purposes, provided that due credit is given to the respective author and publisher.

Resumo

Esta dissertação de mestrado tem como objetivo o desenvolvimento de um sistema inteligente para deteção de consumos anómalos em hotéis. O sistema em si, detetará precocemente anomalias de consumo, quer sejam de eletricidade, água ou gás, para poderem ser analisados e eventualmente corrigidos, refletindo-se em menores consumos, numa maior poupança e numa maior segurança.

Para ser possível o desenvolvimento deste sistema, como em qualquer aplicação que envolva modelos de aprendizagem máquina, primeiramente efetuou-se uma análise exploratória a um conjunto de dados de um hotel, que de alguma forma seriam representativos dos dados que a plataforma irá processar futuramente. Em seguida, é efetuado um estudo comparativo entre dois algoritmos de aprendizagem máquina, *Isolation forest* e *Variational autoencoder*, para decidir qual deles a ser implementado no sistema. Para tomar esta decisão, desenvolveu-se uma métrica de desempenho de algoritmos para deteção de anomalias em séries temporais não supervisionadas. Depois é apresentada a arquitetura do sistema e alguns resultados de operação em condições reais. Para concluir, apresentam-se as conclusões do trabalho e são feitas algumas sugestões para possíveis trabalhos futuros.

Palavras chave: Sistema Inteligente, Deteção de Anomalias, Aprendizagem Máquina, *Isolation Forest*, *Variational Autoencoder*.

Abstract

This master's dissertation aims to develop an intelligent system for anomalous consumption detections in hotels. The system itself will early detect consumption defects, whether electricity, water, or gas, so that they can be analyzed and eventually corrected, resulting in lower consumption, greater savings and higher security.

In order to develop the system, like in other applications that involve machine learning models, an initial exploratory data analysis was performed on hotel datasets, that would somehow be representative of the data that will be processed in the future by the system. Then, a comparative study was carried out between two machine learning algorithms, *Isolation forest* and *Variational autoencoder*, to decide which one to implement in the system. To make this decision, a performance metric for anomaly detection algorithms in unsupervised time series was developed. Afterward, the system architecture and some operational results under real conditions are presented. To finalize, the conclusions of the work are presented and some suggestions are made for possible future works.

Keywords: Intelligent System, Anomaly Detection, Machine Learning, *Isolation Forest*, *Variational Autoencoder*.

Dedico aos meus pais Zé-zé (in memoriam) e Jeni.

Agradecimentos

Primeiramente quero agradecer aos meus orientadores de tese, Professor Pedro Cardoso e Professor Jânio Monteiro, por terem sido incansáveis na realização deste trabalho, sem eles, com toda a certeza, tudo teria sido mais difícil.

Depois agradecer a colaboração da empresa Âmago Unipessoal Lda, na pessoa do Eng. João Raposo e do Alexandre Campos por todo o apoio prestado.

Agradecer também aos meus colegas de laboratório Ricardo Veiga e Pedro Vaz, pela partilha de conhecimentos, pontos de vista e a sempre disponibilidade para ajudar. De referir também os colegas do laboratório "SmartGrids" pelas boas conversas, boa disposição e amizade.

Para terminar queria agradecer o apoio da minha mãe, família e amigos que me acompanham nesta caminhada que é a vida.

Conteúdo

Lista de Tabelas	xv
Lista de Figuras	xvi
Lista de Abreviaturas	xix
Lista de Símbolos	xxi
Capítulo 1 Introdução	1
1.1 Contextualização do trabalho	1
1.2 Objetivos	4
1.3 Estrutura da dissertação	5
Capítulo 2 Enquadramento Teórico	7
2.1 Detecção de anomalias	8
2.2 Métodos estatísticos	9
2.2.1 Intervalo interquartil	10
2.2.2 Método do desvio padrão	11
2.3 Aprendizagem máquina e profunda	13
2.3.1 <i>Isolation forest</i>	13
2.3.2 Redes neuronais	16
2.3.3 <i>Long short term memory networks</i>	18
2.3.4 <i>Autoencoders</i>	20
2.3.5 <i>Variational autoencoder</i>	21
Capítulo 3 Análise e tratamento de dados	23
3.1 Análise exploratória dos dados	23
3.1.1 Representação gráfica das séries temporais	24
3.1.2 Análise estatística	27
3.1.3 Histogramas	28
3.1.4 Conclusões da análise exploratória	28
3.2 Correção dos dados	29
3.2.1 Detecção e remoção de <i>outliers</i>	30
3.2.2 Imputação de valores em falta	31
Capítulo 4 Resultados experimentais	35
4.1 Configuração dos testes experimentais	36
4.1.1 Conjunto de dados	36
4.1.2 Métrica de avaliação de desempenho	38
4.2 Resultados computacionais	41

4.2.1	<i>Isolation forest</i>	41
4.2.2	<i>Variational autoencoder</i>	43
4.3	Conclusões	45
Capítulo 5	Plataforma para detecção de anomalias	47
5.1	Módulo de análise de dados	49
5.2	Módulo de detecção de anomalias	50
5.3	Base de dados	51
5.3.1	Base de dados de leituras	52
5.3.2	Base de dados de configuração e estado do sistema	55
5.4	Análise dos resultados obtidos na plataforma	56
Capítulo 6	Conclusões e trabalho futuro	63

Lista de Tabelas

3.1	Descrição estatística de alguns contadores.	28
3.2	Valores das correlações existentes entre leituras às 12:00 e as leituras ocorridas no mesmo instante em dias anteriores.	33
3.3	Valores das correlações existentes entre as leituras às 00:00 horas e as leituras ocorridas no mesmo instante em dias anteriores.	33
4.1	Diferentes conjuntos de variáveis de entrada dos algoritmos, utilizados nos testes.	38
4.2	Melhores resultados para cada contador com o algoritmo IF, segundo a métrica proposta, entre todos os cenários e considerando a referida <i>grid-search</i>	43
4.3	Melhores resultados para cada contador com o algoritmo VAE, segundo a métrica proposta, entre todos os cenários e considerando a referida <i>grid-search</i>	45

Lista de Figuras

2.1	Possíveis formas de distribuição de dados. Adaptado de (Woo, 2020).	9
2.2	Distribuição normal e as suas propriedades probabilísticas.	12
2.3	Exemplo demonstrativo do processo que ocorre numa <i>Isolation tree</i> para isolar os pontos de um espaço amostral de duas dimensões.	14
2.4	Deteção de anomalias com <i>Isolation forest</i> . Adaptado de (Jung et al., 2020).	16
2.5	Analogia entre um neurónio de um cérebro e o neurónio artificial (cima); e estrutura de ligações de uma rede neuronal com o seu equivalente biológico, sinapses (baixo). Adaptado de (Meng et al., 2020).	17
2.6	Estrutura de uma célula LSTM. Adaptado de (Qiu et al., 2020).	19
2.7	Arquitetura base de um AE.	21
3.1	Representação das séries de leituras, associadas a diferentes contadores do hotel, onde se apresentam de forma clara picos de consumos ocasionais.	25
3.2	Pormenor de falha técnica na receção de leituras do contador "Eletricidade 1".	25
3.3	Análise anual para os contadores "Eletricidade 1" e "Gás".	26
3.4	Representações dos consumos diários em períodos semanais, para os 4 contadores em estudo.	27
3.5	Histogramas dos contadores "Água", "Eletricidade 1", "Eletricidade 2" e "Gás" do hotel.	29
3.6	Representação das séries de leituras com as devidas correções, associadas a diferentes contadores do hotel.	31
3.7	Pormenor de uma falha na série de leituras, associada ao contador "Eletricidade 1". O antes e depois da remoção de leituras.	32
3.8	Pormenor do que antes foi uma falha na série de leituras, associada ao consumo de energia do contador "Eletricidade 1", e agora corrigida segundo os métodos propostos: dados originais (topo), após remoção de picos (meio) e após introdução de valores estimados para os valores em falta (em baixo).	34
4.1	Representação gráfica do conjunto de dados usado nos testes computacionais. A vermelho as anomalias introduzidas no ano de teste.	36
4.2	Exemplo ilustrativo da evolução da métrica P_1 em função do número de pontos desconhecidos classificados como anómalos (PDC), com o TPD=8856.	40
4.3	Exemplo ilustrativo da evolução do valor de P_2 em função do valor de $1PAC$, para uma anomalia constituída por 24 leituras.	41
4.4	Arquitetura do VAE utilizado.	44

5.1	Arquitetura do sistema computacional desenvolvido para a deteção de anomalias em consumos hoteleiros.	48
5.2	Arquitetura do módulo "Analisador de dados".	50
5.3	Arquitetura do módulo "Detetor de anomalias".	52
5.4	Configuração da tabela "Dados por analisar" na base de dados SQL Server.	53
5.5	Configuração da tabela "Dados Analisados" na base de dados SQL Server.	54
5.6	Configuração da tabela "Registos" na base de dados SQL Server.	54
5.7	Configuração da tabela "Analisador de dados" na base de dados do tipo SQLite.	55
5.8	Configuração da tabela "Detetor de anomalias" na base de dados do tipo SQLITE.	56
5.9	Resultados da alarmística num período de 10 dias, sobre o contador "Eletricidade A" do Hotel I.	58
5.10	Resultados da alarmística num período de 10 dias, sobre o contador "Eletricidade B" do Hotel I.	58
5.11	Resultados da alarmística num período de 10 dias, sobre o contador "Eletricidade C" do Hotel I.	59
5.12	Resultados da alarmística num período de 10 dias, sobre o contador "Eletricidade D" do Hotel I.	59
5.13	Resultados da alarmística num período de 10 dias, sobre o contador "Eletricidade E" do Hotel I.	60
5.14	Resultados da alarmística num período de 10 dias, sobre o contador "Eletricidade F" do Hotel II.	60
5.15	Resultados da alarmística num período de 10 dias, sobre o contador "Eletricidade G" do Hotel II.	61

Lista de Abreviaturas

APREN	<i>Associação Portuguesa de Energias Renováveis</i>
ANN	<i>Artificial Neural Networks</i>
AE	<i>Autoencoders</i>
AM	Aprendizagem Máquina
AP	Aprendizagem Profunda
BD	Base de Dados
CNN	<i>Convolutional Neural Networks</i>
GTC	Gestão Técnica Centralizada
IA	Inteligência Artificial
IF	<i>Isolation Forest</i>
IIQ	Intervalo Interquartil
IoT	<i>Internet of Things</i>
IGBT	<i>Insulated Gate Bipolar Transistor</i>
LSTM	<i>Long Short Term Memory</i>
MDP	Método do Desvio Padrão
MMPL	Média Móvel Ponderada Linear
RNN	<i>Recurrent Neural Networks</i>
RN	Redes Neurais
SGBD	Sistemas de Gestão de Bases de Dados
VAE	<i>Variational Autoencoder</i>

Lista de Símbolos

MtCO₂eq Megatonelada de dióxido de carbono equivalente

kWh Kilowatt hora

m³ Metros cúbicos

CO₂ Dióxido de carbono

1

Introdução

1.1 Contextualização do trabalho

Com o avançar das alterações climáticas, existe um despertar geral das pessoas e dos governos para os problemas daí resultantes e uma efetiva aposta na procura de soluções para o seu combate. Nas sociedades do mundo ocidental, onde se inclui Portugal, o modelo económico de consumo adotado é altamente dependente dos recursos energéticos para a sua prosperidade, colocando uma elevada pressão na exploração dos mesmos.

O relatório global da Agência Internacional para a Energia, de 2021, apontava para um aumento da procura global de energia em 4,6 % face ao ano de 2020, superando os níveis pré-pandemia COVID-19. Também o crescimento económico global foi revisto

em alta e apontado para os 6 %, com os níveis de emissões de CO₂ a acompanharem a subida, em cerca de 4,8 %, decorrente da procura de recursos energéticos primários, como o carvão, petróleo e gás natural (International Energy Agency, 2021). Com estes dados entende-se que é inerente ao crescimento económico o consumo energético e, nos dias de hoje, a consequente emissão de gases de efeito estufa para a atmosfera de alguma forma parece inevitável.

É aqui que se coloca o dilema da energia, isto é, se será possível satisfazer a crescente procura, mantermos os nossos estilos de vida e ainda conciliá-la com as metas ambientais que se pretendem, nomeadamente, a descarbonização da sociedade até 2050. A resposta parece ser positiva e a solução passa por uma transição para uma sociedade mais verde, assente em 3 pilares: produção elétrica renovável, mobilidade elétrica e gestão energética (United Nations, 2021).

A produção renovável existe há vários anos, mas as produções fotovoltaicas e eólicas só ganharam relevo nos últimos 10 com a evolução tecnológica dos painéis e turbinas. Por exemplo, em Portugal, segundo os dados da Associação Portuguesa de Energias Renováveis (APREN), durante o ano de 2021 a geração renovável foi responsável pelo fornecimento de 65,4 % da energia consumida, evitando a emissão de 11,6 MtCO₂eq (Associação Portuguesa de Energias Renováveis, 2022).

A mobilidade elétrica também já é uma realidade possível nos dias de hoje, com um crescimento substancial da cota de mercado a cada ano que passa (International Energy Agency, 2022). A aceitação das pessoas por estes veículos em parte deve-se ao desenvolvimento de infraestruturas de carregamento, maior fiabilidade das baterias e pelo incremento da autonomia, que começa a estar equiparável, e até em certos modelos superior, à dos veículos movidos por motores de combustão interna (Holmberg and Erdemir, 2019; Klöckner et al., 2013).

Por último, na nossa lista, a gestão de energia, sendo o objetivo otimizar a utilização da mesma, produzindo mais com menos. Neste âmbito tem-se apostado muito na melhoria da eficiência energética dos edifícios, que representam 30 – 45 % do consumo de energia global (Gul and Patidar, 2015), e na integração de sistemas de gestão

de edifícios.

Neste contexto, a redução de consumos através da deteção de anomalias é uma das áreas de estudo com bastante interesse na gestão energética, isto porque muitas das vezes um consumo anómalo representa perdas adicionais, refletindo-se em maiores custos e poluição ambiental. Para combater este problema têm sido propostas várias soluções de sistemas inteligentes, que visam a deteção precoce dos defeitos e assim mitigar os seus efeitos (Himeur et al., 2021; Fenza et al., 2019; Araya et al., 2017). É neste âmbito que a dissertação aqui apresentada foi desenvolvida, especificamente na produção de um sistema inteligente para deteção de anomalias em consumos hoteleiros. Na realidade, no setor do turismo, os hotéis são das instalações que representam maiores gastos energéticos, pelo que, sendo o Algarve uma zona fortemente turística, a exploração e desenvolvimento deste tema pode vir a gerar poupanças significativas nos consumos e correspondentes gastos destas empresas.

A proposta desenvolvida, resulta de um problema real, colocado pela empresa Âmago, Unipessoal Lda. Esta empresa da região do Algarve, procurou o desenvolvimento de um sistema que permita detetar consumos anómalos. Assim sendo, este trabalho faz parte de um projeto de prestação de serviços à referida empresa, os quais são essenciais para uma aproximação do tecido empresarial da região com a academia e transferência de conhecimentos.

Um dos maiores desafios no desenvolvimento destas soluções para hotéis é a difícil caracterização dos padrões de consumo do edifício. Por vezes, a dependência de fatores como sejam o nível de ocupação, a prestação de serviços externos (e.g., eventos, conferências), ocupações não planeadas, entre outros, provocam de certo modo uma aleatoriedade e uma imprevisibilidade à forma como se processam os consumos, dificultando em muito a tarefa dos algoritmos de deteção de anomalias (Eras et al., 2016; Deng and Burnett, 2000). Outros problemas que podem surgir no desenvolvimento destas soluções, cabem-se com a necessidade de um sistema para recolha de dados fiável ou a necessidade de uma solução adaptável a diferentes contadores e hotéis, criando-se um problema de generalização dos algoritmos. No sentido de dar uma

resposta efetiva aos problemas identificados, a próxima secção apresenta os objetivos gerais propostos para o desenvolvimento do referido sistema.

1.2 Objetivos

A presente dissertação teve como principal objetivo explorar o desenvolvimento de um sistema inteligente suportado, de modo geral, em algoritmos de Aprendizagem Máquina (AM) que:

1. Integre dados provenientes de vários sensores, que recolhem dados nas instalações hoteleiras (ou outras);
2. Faça uso de metodologias da ciência dos dados para processar e extrair conhecimento dos dados disponíveis;
3. Faça a deteção de anomalias associadas a consumos (e.g., energéticos ou água) em ambiente hoteleiro;
4. Seja uma solução modular, personalizável e projetada para as necessidades da Âmago, Unipessoal Lda.

Assim perante os objetivos definidos, alguns dos tópicos alvos de estudo e desenvolvimento ao longo desta dissertação foram:

- Deteção de anomalias — neste contexto foram estudados diferentes abordagens e algoritmos para a resolução do problema de deteção de anomalias, tendo a escolha recaído sobre a análise dos modelos: *IsolationForest* e *Variational Autoencoder* (Capítulo 2);
- Análise e tratamento de dados — o que incluiu o desenvolvimento de um módulo do sistema, dedicado à remoção de *outliers* e imputação de valores em falta (Capítulo 3);

- Avaliação do desempenho dos algoritmos — averiguação de metodologias para avaliação do desempenho dos algoritmos na classificação de anomalias em séries-temporais não supervisionadas, o que motivou a criação de uma métrica de performance para o efeito (Capítulo 4).

1.3 Estrutura da dissertação

Esta dissertação está estruturada do seguinte modo:

- **Capítulo 1** — Apresentou as motivações e objetivos da dissertação;
- **Capítulo 2** — Começa por introduzir o problema da deteção de anomalias e, de seguida, efetua a análise do estado de arte dos algoritmos estatísticos e de AM considerados relevantes para o problema em causa;
- **Capítulo 3** — Apresenta os procedimentos efetuados para analisar os dados e os métodos utilizados na correção de problemas identificados;
- **Capítulo 4** - Estabelece a métrica de performance e o estudo desenvolvido para avaliação dos algoritmos de AM;
- **Capítulo 5** — Apresenta a plataforma desenvolvida para deteção de anomalias em consumos hoteleiros e resultados provenientes de testes reais;
- **Capítulo 6** — Apresenta as conclusões e possíveis passos para uma investigação futura.

2

Enquadramento Teórico

O desenvolvimento de plataformas inteligentes, hoje em dia, muito dificilmente existe sem a utilização de algoritmos de AM (Butakov, 2020; Cook et al., 2019; Dunning and Friedman, 2014). Sendo este o tema desta dissertação, neste capítulo são dados a conhecer alguns dos algoritmos que têm provado ser mais eficazes no problema em estudo e que mais tarde serão testados. Assim, primeiro é feita uma introdução ao problema da deteção de anomalias (Secção 2.1). Depois, segundo uma lógica evolutiva, é apresentado um conjunto de algoritmos estatísticos (Secção 2.2) e por fim os algoritmos de AM testados, com referências a trabalhos que sustentam as suas escolhas, Secção 2.3.

2.1 Detecção de anomalias

Um problema de detecção de anomalias pode ser definido pelo encontrar de pontos ou padrões nos dados que se distinguem dos restantes, sendo estes designados de anomalia ou *outlier*. A detecção de anomalias é uma área de tal abrangência que pode ser encontrada em vários campos do conhecimento (Abdallah et al., 2016; Chandola et al., 2009), como por exemplo: na detecção de fraudes (Zheng et al., 2020; Shen et al., 2007), no processamento de imagem (Deecke et al., 2018), no campo da saúde (Bao et al., 2019; Tang et al., 2019), na detecção de falhas em equipamentos (Zhou et al., 2020; Li et al., 2015), ou consumos (Himeur et al., 2021; Fenza et al., 2019). A detecção de anomalias está geralmente dividida em 3 categorias principais de problemas (Omar et al., 2013; Chandola et al., 2009):

1. **Detecção supervisionada** - problemas em que os dados se encontram classificados, ou seja, são conhecidas anomalias e dados normais. As abordagens de detecção costumam basear-se em algoritmos que comparam os novos dados com os existentes (Cervantes et al., 2020).
2. **Detecção semi-supervisionada** - quando apenas são conhecidos dados referentes à normalidade do sistema. Tipicamente estes algoritmos treinam com estes dados e são testados sobre um conjunto de dados nunca visto por forma a avaliar a presença de anomalias (Yan, 2020; Akcay et al., 2018).
3. **Detecção não supervisionada** - ao contrário das detecções supervisionadas e semi-supervisionadas, aqui os dados não estão rotulados, desconhecendo-se de todo, os que são anómalos ou normais. Estes algoritmos usualmente partem da premissa que existem poucas anomalias no conjunto dos dados e estas têm características diferentes dos restantes (Zong et al., 2018; Munir et al., 2018).

Tendo em atenção esta divisão de problemas, foi entendido que deveríamos enquadrar o nosso problema de detecção de anomalias na categoria dos algoritmos não supervisionados, pois os dados disponíveis para a realização do estudo não eram classificados, como será visto mais adiante no Capítulo 3, e também por ser necessário

desenvolver uma solução aplicável a diferentes hotéis e contadores, sobre condições desconhecidas, e de forma dinâmica, i.e., novos hotéis/contadores poderão ser adicionados à plataforma, devendo esta funcionar sobre "pequenas" configurações da equipa de gestão.

Assim, como referido, na próxima secção começaremos por analisar algumas metodologias estatísticas.

2.2 Métodos estatísticos

No contexto da deteção de anomalias, os algoritmos estatísticos remetem para os primórdios da área. Como o nome indica, baseiam-se em propriedades estatísticas dos dados, como sejam, a média ou a mediana, os percentis, o desvio padrão, etc. Dependendo do tipo de distribuição dos dados (ver Figura 2.1), as estratégias também podem ser diferentes. Por exemplo, se os dados sofrerem de algum tipo de enviesamento ou tiverem uma distribuição que não a normal, a aplicação do Intervalo Interquartil (IIQ) é usual (Secção 2.2.1), enquanto que se os dados se distribuem de forma normal pode ser aplicado o Método do Desvio Padrão (MDP) (Secção 2.2.2) ou a pontuação Z (Rousseeuw and Hubert, 2018).

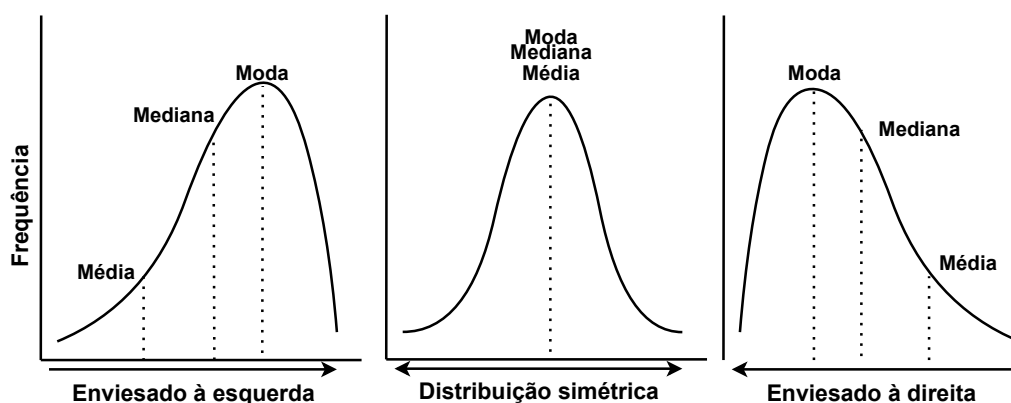


Figura 2.1: Possíveis formas de distribuição de dados. Adaptado de (Woo, 2020).

Com o desenvolver de algoritmos de AM, estes métodos perderam alguma relevância dentro do campo da deteção de anomalias no sentido de serem o elemento central, encontrando muitas das vezes utilizações durante a fase de tratamento dos da-

dos (Brownlee, 2020; Ilyas and Chu, 2019; Xu et al., 2015) ou como parte dos sistemas de detecção de anomalias no cálculo dos limites de detecção (Hamamoto et al., 2018; Luo and Zhong, 2017; Haque et al., 2015). Estas duas aplicações fundamentam o estudo de alguns métodos estatísticos que são apresentados de seguida, e mais adiante, nas Secções 3.2 e 4.2.2, poderão ser encontradas as respetivas aplicações nesta dissertação.

2.2.1 Intervalo interquartil

Um dos muitos métodos estatísticos existentes para detecção de anomalias baseia-se no Intervalo Interquartil. Esta opção, normalmente tem aplicação, quando os dados não apresentam uma distribuição normal ou aproximadamente normal. Por ser um método baseado em valores estatísticos resilientes à presença de anomalias (quartis), a sua utilização acaba por não apresentar grandes restrições (Kwak and Kim, 2017; Walfish, 2006; Hogg et al., 1977).

Como referido anteriormente, este método tem por base o valor da amplitude, IIQ, que é dada pela diferença entre o terceiro quartil (Q_3) e o primeiro quartil (Q_1),

$$IIQ = Q_3 - Q_1. \quad (2.1)$$

O IIQ representa 50% do volume de dados em volta da mediana, sendo que por esta razão acaba por ser um valor de confiança para utilizar na detecção de valores suspeitos. Para o cálculo dos limites superior ($L_{superior}$) e inferior ($L_{inferior}$) do método, que servem de termo comparativo com as leituras, são utilizadas as seguintes equações:

$$L_{superior} = Q_3 + K \times IIQ \quad (2.2)$$

$$L_{inferior} = Q_1 - K \times IIQ, \quad (2.3)$$

onde a constante K possibilita o ajuste da amplitude do intervalo de detecção, de forma a que seja mais ou menos restritivo.

O método baseado no IIQ tem sido utilizado em alguns estudos. Por exemplo, Saradjian and Akhoondzadeh (2011), compararam o desempenho de alguns métodos so-

bre dados de temperaturas, com o objetivo de identificarem alterações que antecedem a ocorrência de sismos de elevada magnitude. Os autores verificaram que o método baseado no IIQ, mesmo não tendo sido o melhor, com a sua simplicidade, teve um bom desempenho nos três cenários estudados, permitindo em dois deles antecipar os sismos, tal como o melhor algoritmo no estudo. Moustafa et al. (2017) abordaram o desenvolvimento de um sistema de deteção de ataques a dados na nuvem, no qual também utilizaram este método no cálculo dos limites para classificação dos possíveis ataques. Utilizaram um conjunto de dados, designado por UNSW-NB15, na realização dos testes e mostraram uma elevada taxa de acerto na deteção e baixa incidência de falsos-positivos, com um $K = 3$ para os melhores resultados alcançados.

2.2.2 Método do desvio padrão

O Método do Desvio Padrão costuma ser aplicado segundo dados que têm uma distribuição normal ou aproximadamente normal. A distribuição normal é uma das formas de distribuição de probabilidades contínua mais importante e amplamente utilizada em estatística e noutros campos científicos (Ahsanullah et al., 2014). Uma das suas características é possuir os valores da média, mediana e moda iguais, o que lhe confere um gráfico centrado em torno deste ponto, com 50 % dos dados para cada lado e um decréscimo da concentração de valores em ambas as direções à medida que nos afastamos do centro, o que resulta no gráfico em forma de "sino" como é vulgarmente conhecido (Figura 2.2). A utilização deste método na identificação de anomalias é bastante frequente, devido à relação que existe entre a média (μ), o desvio padrão (σ) e a probabilidade dos valores serem encontrados em determinadas zonas da curva de distribuição. Por exemplo a probabilidade de uma observação ser inferior a $\mu - \sigma$ é de 15,9% ou a de estar fora do intervalo $[\mu - 2\sigma, \mu + 2\sigma]$ é de 4,6% (Weisstein, 2002; Hogg et al., 1977).

Como exemplos da utilização deste método temos um estudo desenvolvido por Partil et al. (2015), onde os autores abordam a deteção de falhas em *Insolated Gate Bipolar Transistors* (IGBTs). Neste trabalho são estudadas duas hipóteses de deteção, uma com

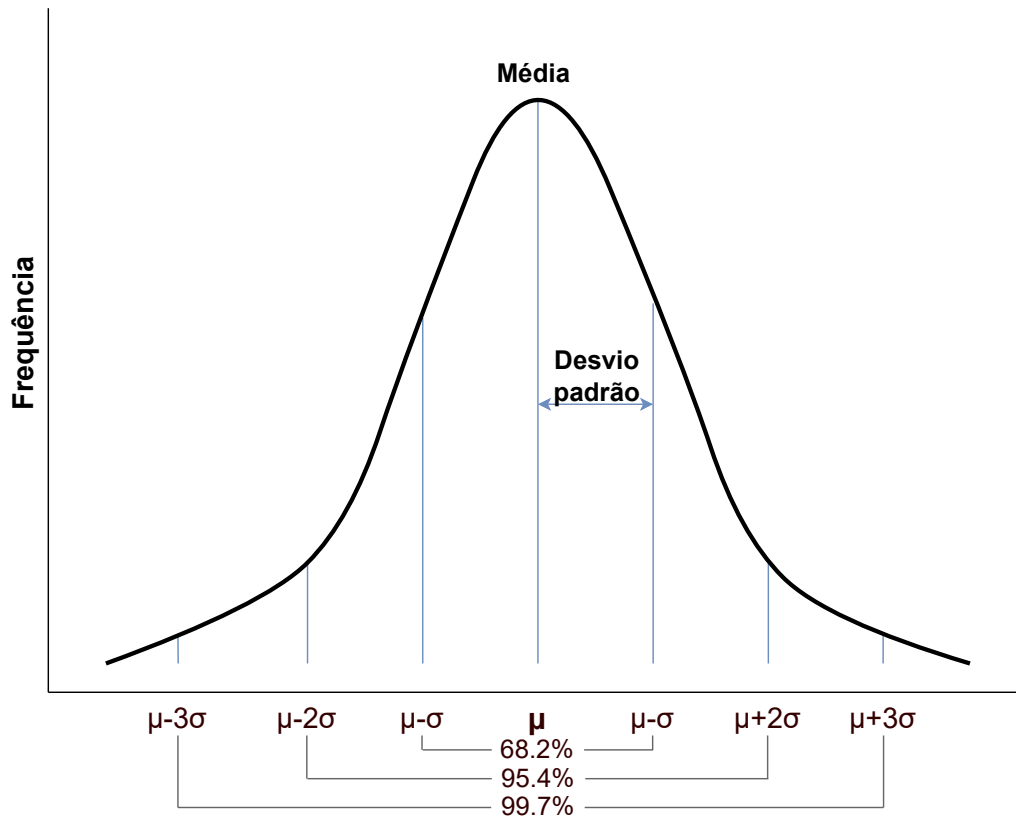


Figura 2.2: Distribuição normal e as suas propriedades probabilísticas.

base num limite fixo e outra com a distância de Mahalanobis, onde utilizam o MDP para calcular o limite que define a possível anomalia. Com os testes realizados, provaram que a abordagem que envolvia o MDP obteve os melhores resultados do estudo, permitindo até mesmo, antecipar a ocorrência das falhas dos transístores. Outro estudo, este elaborado por Giannoni et al. (2018) envolvendo séries de valores provenientes de sensores *Internet of Things* (IoT) comparou vários métodos para detecção de anomalias. Um dos métodos consistiu em calcular a média de uma janela deslizante com n amostras e calcular os limites de detecção segundo a média e o desvio padrão, para assim fazer uso, mais uma vez, das propriedades da distribuição normal. Os autores concluíram que uma junção dos vários métodos, poderia vir a ser melhor, pois cada algoritmo tinha vantagens e desvantagens nos diferentes cenários. Assim como é aplicado na detecção de anomalias, este método também é aplicado durante a fase de preparação de dados. Por exemplo, Brownlee (2020), no livro "Data Preparation for Machine Learning", apresenta um exemplo demonstrativo com o intuito de identificar

pontos que comprometem a restante série e que afetam o desempenho dos algoritmos de AM analisados.

2.3 Aprendizagem máquina e profunda

A aprendizagem máquina e aprendizagem profunda são áreas da Inteligência Artificial (IA) em pleno crescimento nos últimos anos, que abriram novos horizontes no avanço tecnológico, ao permitirem que os computadores adquirissem conhecimento sobre dados existentes e assim desenvolvessem tarefas autonomamente. Estes algoritmos, têm tido impacto prático em diversos setores, como por exemplo nos setores da saúde (Bullock et al., 2020; Nichols et al., 2019; Yu et al., 2018), das finanças (Buchanan, 2019; Dunis et al., 2016) ou da energia (Farzaneh et al., 2021; Zhao et al., 2020). Algumas das aplicações têm sido no reconhecimento de imagens (p.e., para deteção de doenças ou na condução autónoma), previsão da evolução dos valores de ações ou na deteção de anomalias em consumos (Mohammed et al., 2016; Rudin and Wagstaff, 2014).

Nas próximas secções são introduzidos os algoritmos de AM adotados nesta dissertação.

2.3.1 *Isolation forest*

No campo da AM optou-se por analisar o comportamento do algoritmo *Isolation Forest* (IF) (Liu et al., 2008), pois dentro dos algoritmos de aprendizagem máquina não-supervisionados, pareceu ser um pouco diferente dos típicos métodos de *clustering*, onde os pontos estão agrupados e as anomalias são calculadas, por exemplo, com base em distâncias ou densidades (Sinaga and Yang, 2020; Xu and Tian, 2015).

Este método é formado por um conjunto de elementos designados por *Isolation trees*. As *Isolation trees* têm como princípio de funcionamento, isolar os pontos de um determinado conjunto de dados $X = [X_1, X_2, \dots, X_j]$. Para que isto aconteça, existe uma lógica de divisão recursiva do espaço amostral, na qual vão sendo escolhidas aleatoriamente variáveis pertencentes aos elementos do espaço amostral X e depois selecio-

nado, também aleatoriamente, um valor de divisão (p) compreendido no intervalo de valores dessa mesma variável, tal como é possível verificar no exemplo da Figura 2.3. A teoria diz-nos que observações anómalas normalmente são diferentes, existem em pouca quantidade e, por isso, é esperado que neste algoritmo essas observações necessitem de um menor número de divisões para serem isoladas, isto porque deverão encontrar-se mais distantes dos restantes pontos. No final, o processo equivale à lógica de uma coleção de árvores binárias, i.e., obtém-se uma floresta.

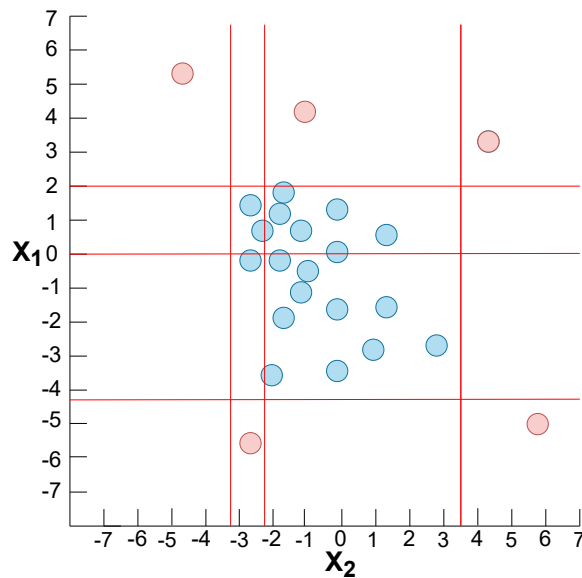


Figura 2.3: Exemplo demonstrativo do processo que ocorre numa *Isolation tree* para isolar os pontos de um espaço amostral de duas dimensões.

Para se classificar as instâncias como anomalias ou normais, existe um sistema de pontuação associado ao algoritmo, dado pela equação

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}},$$

onde, para a instância x , $h(x)$ corresponde ao comprimento do caminho desde a raiz até ao nó externo que isola x numa determinada árvore da floresta, $E(h(x))$ é o comprimento médio desses caminhos na floresta e $c(n)$ é o comprimento médio das *Isolation trees*, para n amostras, i.e., o número de instâncias em X . Este valor é calculado segundo a equação

$$c(n) = 2H(n-1) - (2(n-1)/n),$$

onde H é um número harmónico e n o número de amostras do conjunto X (Liu et al., 2008).

A IF tem como fundamento utilizar várias árvores por forma a aumentar a eficácia da deteção de anomalias pois, como o processo de divisão é independente entre árvores, quando determinados pontos produzirem coletivamente uma resposta, quer no sentido de normalidade ou anomalias, a probabilidade efetiva de o serem é maior do que apenas com uma árvore. Por exemplo, na Figura 2.4 podemos encontrar um exemplo bastante elucidativo do que anteriormente foi dito em relação à IF.

O algoritmo foi utilizado na resolução de vários problemas de deteção de anomalias. Por exemplo, Puggini and McLoone (2018) desenvolveram um estudo, no qual comparam quatro métodos para redução de variáveis e utilizaram o algoritmo IF na deteção de anomalias, isto sobre dados da indústria de semi-condutores. Para entenderem qual a melhor abordagem do estudo, realizaram dois casos de estudo, dos quais concluíram que existiriam algumas vantagens em utilizarem determinados métodos de redução em relação a outros. Propuseram também um sistema de deteção de anomalias baseado no método IF. Ding and Fei (2013) apresentaram igualmente uma arquitetura geral para deteção de anomalias em dados de *streaming* com o método IF, e sugerem uma abordagem onde relacionam a IF com a abordagem da janela deslizante, abordagem conhecida para séries temporais. Para avaliarem o método, realizaram vários testes segundo quatro conjuntos de dados, os quais permitiram validar a eficácia do algoritmo. Outro trabalho interessante, este realizado por de Santis and Costa (2020), consistiu em comparar diferentes algoritmos na identificação de falhas em pequenas centrais hidroelétricas. Dos resultados obtidos, concluíram que o algoritmo IF e a sua variante *Extended Isolation Forest* foram os algoritmos que mais se destacaram, que têm boas características para serem adotados num sistema de deteção online e são modelos, que têm pouco peso computacional.

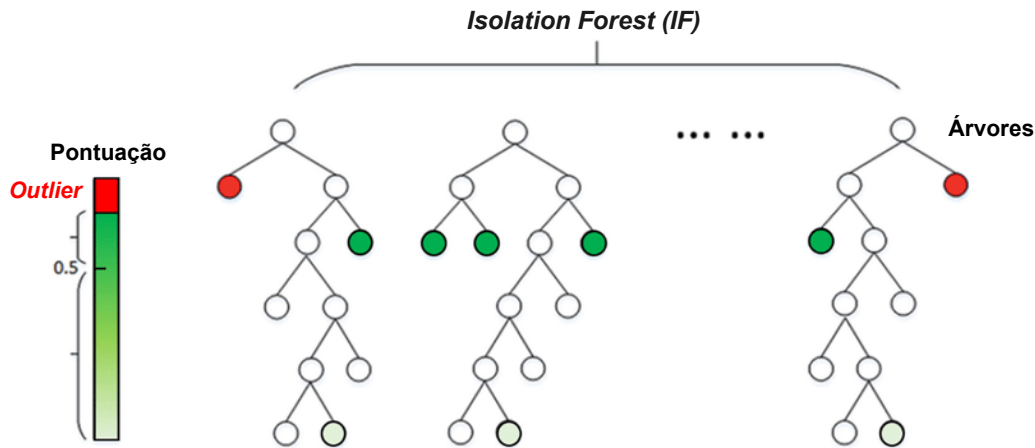


Figura 2.4: Detecção de anomalias com *Isolation forest*. Adaptado de (Jung et al., 2020).

2.3.2 Redes neurais

Algoritmos baseados em Redes Neurais (RN), devido à sua diversidade de tipologias, arquiteturas e aplicações em problemas, são por si só, uma área dentro AM, designada igualmente por Aprendizagem Profunda (AP).

O princípio desta área de investigação e aplicação remonta os anos 40 do século XX, e desde então, tem sofrido vários altos e baixos no seu desenvolvimento, investigação e popularização. Nos anos 80 é quando se dá um grande desenvolvimento, com o aparecimento do algoritmo de aprendizagem *back-propagation* para efetuar o treino das RN e mais recentemente com o desenvolvimento das capacidades computacionais. As primeiras redes a serem criadas foram as *Artificial Neural Networks* (ANN) à semelhança do cérebro humano. Todas as competências inigualáveis do cérebro, como a capacidade de aprender, adaptar-se a diferentes cenários e processar informação contextual dos mesmos, p.e. localizar uma pessoa no meio de uma multidão apenas pela cara ou reconhecer uma pessoa pela voz, eram características interessantes para se tentar criar num algoritmo (Da Silva et al., 2017; Jain et al., 1996). Assim como o cérebro humano tem por base o neurónio biológico (Figura 2.5 (a)), as RN têm como unidade primária o neurónio artificial (Figura 2.5 (b)). O cérebro em termos de organização, possui milhões de neurónios, os quais se encontram interligados entre si nas zonas de sinapse (Figura 2.5 (c)), permitindo a propagação e processamento da informação. Nas

RN o conceito é semelhante, onde os neurónios se estabelecem em camadas, que por sua vez se vão ligando entre si (Figura 2.5 (d)).

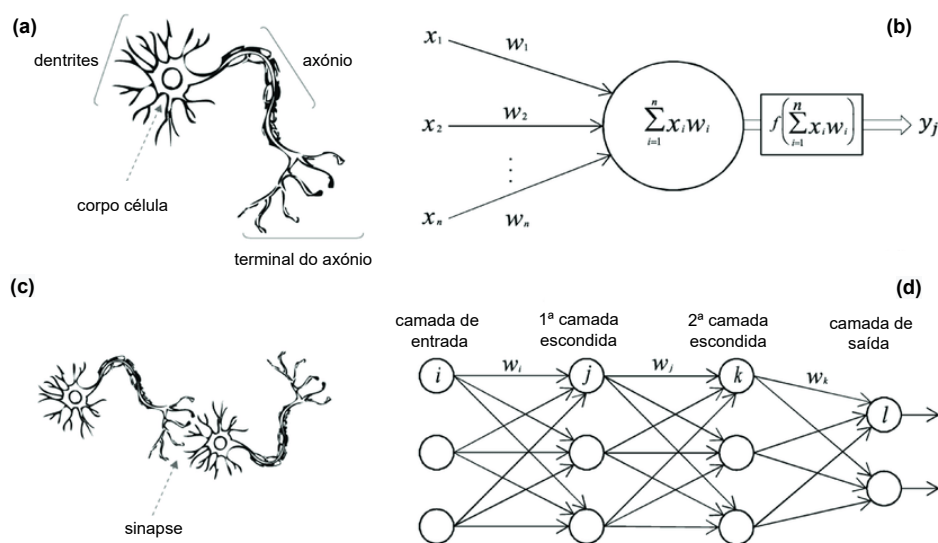


Figura 2.5: Analogia entre um neurónio de um cérebro e o neurónio artificial (cima); e estrutura de ligações de uma rede neuronal com o seu equivalente biológico, sinapses (baixo). Adaptado de (Meng et al., 2020).

Para entender melhor o princípio geral de funcionamento das RN como um todo, conhecer as ANN poderá ser bastante elucidativo, ressaltando apenas que cada tipologia de rede tem algumas particularidades. Tudo começa com o neurónio, onde a informação chega e sofre influência dos pesos associados às diferentes ligações estabelecidas entre os mesmos ($w_i, i = 1, 2, \dots, n$). O resultado desta operação ($\sum_{i=1}^n x_i w_i$) segue para uma função ativação ($f(\sum_{i=1}^n x_i w_i)$), a qual é responsável por ditar o estímulo de saída (y_j), sendo este depois propagado para os neurónios seguintes (Ng et al., 2011; Yegnanarayana, 2009; Drew and Monson, 2000). As funções ativação concebidas são inúmeras, podendo-se destacar a *SoftMax*, a *ReLU* ou a *tanh*. Para um melhor entendimento da diversidade e diferenças destas funções, podem ser consultados os trabalhos de Sharma et al. (2017) ou Ramachandran et al. (2017). Os pesos nas ligações entre os vários neurónios (w_i), são ajustados durante a fase de treino da rede de forma a preservar o máximo de informação. O algoritmo *back-propagation*, foi um grande passo no treino de RN complexas, por ser um processo de baixa complexidade computacional no cálculo de gradientes, pois muitos dos algoritmos utilizados na otimização dos pe-

os da rede dependem destes, como p.e. o gradiente descendente, gradiente descende estocástico, entre outras variantes destes (Adam ou RMSprop p.e.) (Soydaner, 2020; Goodfellow et al., 2016).

As aplicações das RN vão desde problemas que envolvem classificação até problemas de regressão (Rocha et al., 2007), com aplicações em diversas áreas científicas. As RN são também toda uma diversidade, cada qual com as suas características e especificidades. Por exemplo, as *Convolutional Neural Networks* (CNN) (Yamashita et al., 2018) são muito utilizadas em problemas de visão, as ANN ou as *Recurrent Neural Networks* (RNN) (Lipton et al., 2015; Hochreiter, 1998) para análise de séries temporais, etc. Relativamente a séries temporais, as RN têm sido muito aplicadas na resolução de problemas que requerem previsão de valores (Smyl, 2020; Tealab, 2018) e classificação, quer sejam para deteção de anomalias, reconhecimento de padrões ou previsão de falhas (Ismail Fawaz et al., 2019). Na secção seguinte é feita a introdução a um caso especial de RN designadas *Long Short Term Memory* (LSTM), que pertencem à família das RNN, por serem fundamentais na compreensão do modelo *Variational Autoencoder* (VAE), apresentado na Secção 2.3.5.

2.3.3 *Long short term memory networks*

As redes *Long Short Term Memory* são RN da família das RNN, especialmente desenvolvidas para análise de sequências. As LSTM foram criadas com o intuito de suprimir o problema de captação de dependências de longo prazo existente nas RNN convencionais. Este fenómeno ocorre durante a fase de treino, mais propriamente durante o *back-propagation*, com o desvanecimento do gradiente. O desvanecimento do gradiente (quando a norma do vetor se torna muito pequena) implica que os algoritmos de otimização não consigam encontrar os melhores parâmetros para a rede, levando a que as dependências de longo prazo sejam perdidas na rede (Hochreiter, 1998).

Estes neurónios, ou simplesmente células LSTM, representadas na Figura 2.6, têm certas características especiais como, a capacidade de guardar informação internamente sobre eventos passados, conhecido como estado da célula (c_t), e possuem um

conjunto de portas que regulam o fluxo de informação na célula. Normalmente existem 3 portas: a do esquecimento (f_t) que faz uma primeira filtragem da informação que é excluída; a de entrada ($i_t \cdot \tilde{c}_t$) que define a nova informação a ser incluída no estado da célula; e a de saída (o_t), que decide qual a informação a ser partilhada com as restantes células, com base no estado da célula. O modo como operam, permite que a saída da célula (h_t) e o estado da célula (c_t) tenham influência nos resultados da célula seguinte e assim sucessivamente. Desta forma, a análise feita relaciona a série de valores como um todo e não elemento a elemento, dando-lhes efetivas propriedades para a análise de dados sequenciais (Yu et al., 2019; Hochreiter and Schmidhuber, 1997). O seguinte conjunto de equações representam, matematicamente, a dinâmica interna da célula LSTM para cada intervalo de tempo t , como explicado anteriormente:

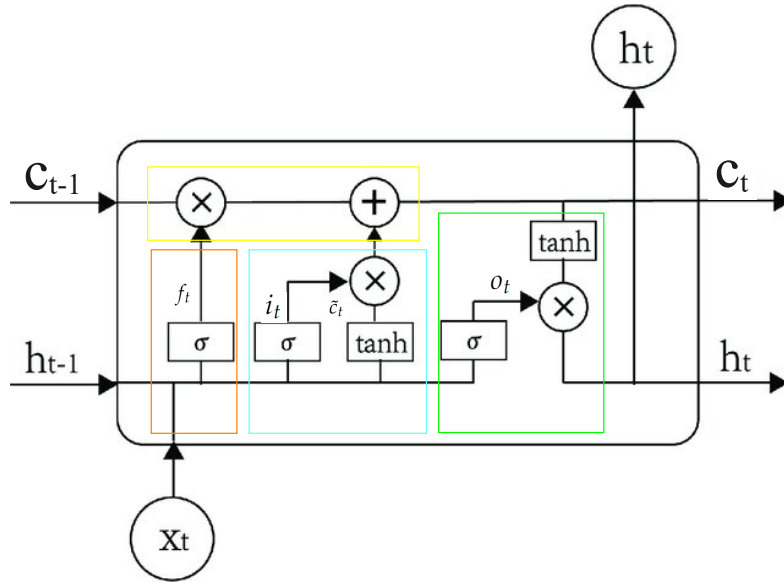


Figura 2.6: Estrutura de uma célula LSTM. Adaptado de (Qiu et al., 2020).

$$\begin{aligned}
 f_t &= \sigma(U^f x_t + W^f h_{t-1}) \\
 i_t &= \sigma(U^i x_t + W^i h_{t-1}) \\
 \tilde{c}_t &= \tanh(U^g x_t + W^g h_{t-1}) \\
 c_t &= f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \\
 o_t &= \sigma(U^o x_t + W^o h_{t-1}) \\
 h_t &= o_t \cdot \tanh(c_t)
 \end{aligned} \tag{2.4}$$

Nas equações apresentadas, W é um parâmetro referente à conexão recorrente que existe entre h_t e h_{t-1} , U traduz a matriz de pesos que conecta x_t com h_t e σ corresponde à função ativação escolhida. Para mais detalhes consultar Yu et al. (2019).

Em certas situações, ter uma percepção futura relativamente ao presente pode ser interessante. Assim, as RNN no seu conjunto, permitem uma configuração do tipo Bidirecional onde 2 camadas são associadas por forma a analisar a sequência em ambos os sentidos, dando um contexto de passado e um contexto de futuro (Yu et al., 2019; Schuster and Paliwal, 1997).

Algumas das muitas aplicações existentes das LSTM têm versado a análise de séries temporais e processamento de texto. Na área do processamento de texto têm sido empregues em modelos de classificação, que podem estar afetos a sistemas de recomendação ou sistemas anti-spam (Liu and Guo, 2019), por exemplo. Também têm sido utilizadas para tradução de texto (Qing-dao-er ji et al., 2020; Guo et al., 2018). Ao nível das séries temporais algumas das aplicações incluem as previsões de valores (Livieris et al., 2020; Cao et al., 2019) ou deteção de anomalias (Niu et al., 2020; Malhotra et al., 2015).

Uma das arquiteturas de RN em destaque na deteção de anomalias são os *Autoencoder*, os quais são apresentados em seguida, com destaque para o *Variational Auto-Encoder* (VAE), o modelo de aprendizagem profunda considerado para testes nesta dissertação.

2.3.4 *Autoencoders*

Os *Autoencoder* (AE), como afirmado por Lindemann et al. (2021), têm mostrado excelentes resultados nos problemas de deteção de anomalias em conjuntos de dados não supervisionados. Por este motivo fazia todo o sentido a investigação e avaliação dos mesmos.

Na sua base, os AE são uma arquitetura de RN (ver Secção 2.3.2), composta por duas partes: (i) o codificador, que tem a função de aprender/comprimir os dados numa dimensão reduzida, espaço latente, preservando a informação mais relevante;

e (ii) o decodificador, o outro elemento da arquitetura que tem a função de reconstruir os dados de entrada com a menor perda possível (Bank et al., 2020; Baldi, 2012). Na Figura 2.7 é possível observar um esquema genérico, que ilustra a arquitetura dos AE, conforme definido anteriormente.

Dentro dos AE existem várias tipologias motivadas por diferentes objetivos, como p.e. lidar com ruído nos dados de entrada, o *Denoising Autoencoder* (Majumdar, 2018; Gondara, 2016), ou para ter grandes capacidades de representação dos dados o caso do *Variational Autoencoder* (Kingma et al., 2019). Para maior detalhe e conhecimento de outras classes, podem ser consultados os trabalhos de Bank et al. (2020); Doersch (2016); Ng et al. (2011).

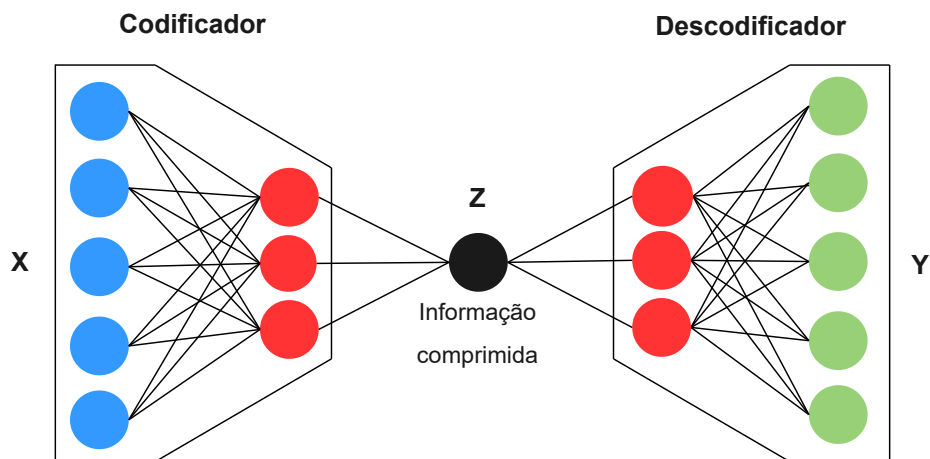


Figura 2.7: Arquitetura base de um AE.

2.3.5 Variational autoencoder

Existem vários trabalhos recentes onde têm sido utilizados os *Variational Autoencoder* para detecção de anomalias não supervisionadas, tais como nos trabalhos de Yao et al. (2019); Pereira and Silveira (2018); Xu et al. (2018). Desses trabalhos, destaca-se o de Pereira and Silveira (2018), onde um VAE foi aplicado na detecção de anomalias em dados de produção fotovoltaica, tendo representado um passo bastante importante para o entendimento do modelo, de como o criar e de como aplicar à detecção de anomalias sobre os dados disponíveis para esta dissertação.

Os VAE têm como principal característica diferenciadora, o facto de modelarem o espaço latente Z , em função de uma variável aleatória com determinado tipo de distribuição. Usualmente a distribuição Normal Isotrópica é a aplicada, porque se considera que as relações existentes entre variáveis numa dimensão reduzida, serão menos complexas que na dimensão de entrada (An and Cho, 2015).

A deteção de anomalias com este modelo pode ser efetuada com base no erro de reconstrução ou então, por se tratar de um modelo probabilístico em que se reconstroem os parâmetros de uma distribuição, é possível utilizar também medidas probabilísticas.

No próximo capítulo é apresentada a análise exploratória efetuada aos dados de quatro contadores de um hotel, representativos dos dados para o qual o sistema a ser desenvolvido deve ser preparado.

3

Análise e tratamento de dados

Neste capítulo é explicada a análise efetuada aos dados (Secção 3.1), por forma a entender potenciais problemas na qualidade dos dados, e, posteriormente, dadas a conhecer as soluções adotadas para o respetivo tratamento (Secção 3.2).

3.1 Análise exploratória dos dados

Como deve ser de conhecimento geral, antes de se iniciarem testes com modelos de AM deve ser efetuado um estudo prévio dos dados, isto é, uma análise exploratória com o intuito de se entender a sua natureza, as suas características e identificar possíveis problemas que possam futuramente prejudicar o desempenho dos modelos de AM (Brownlee, 2020; Velleman, 2004; Tukey, 1977).

Existem quatro metodologias principais, para a realização de uma análise explora-

tória dos dados:

1. **Univariável não gráfica** - são as mais simples e visam perceber padrões de uma variável, por exemplo os valores da média e do desvio padrão;
2. **Univariável gráfica** - para efeitos da visualização da própria variável, ou seja, formas gráficas como histogramas e *boxplots*;
3. **Multivariável não gráfica** - pretendem mostrar relações entre as diversas variáveis, como por exemplo, usando tabelas de correlação;
4. **Multivariável gráfica** - a representação dessas mesmas relações segundo mapas de calor, gráficos de dispersão, etc.

Ainda existem outras ferramentas mais avançadas que envolvem a própria AM para efetuar esta análise. Exemplo disso é o algoritmo *K-means clustering* (Cloud Education IBM, 2020).

As abordagens aqui utilizadas consistiram em representações gráficas, sendo também analisados alguns dos principais dados estatísticos, como a média, mediana ou quartis.

O conjunto de dados utilizados para a análise são referentes a um contador de água, um contador de gás e dois contadores de eletricidade de um hotel no Algarve, considerados representativos dos dados que irão ser analisados pelo sistema de deteção de consumos anómalos. As leituras encontravam-se amostradas num intervalo horário, referentes ao período compreendido entre janeiro de 2014 e outubro de 2021.

3.1.1 Representação gráfica das séries temporais

A primeira abordagem consistiu na representação gráfica das diferentes séries temporais. Nos vários contadores analisados, ver Figura 3.1, um dos problemas bem identificado nos gráficos foram picos ocasionais nas leituras de amplitude variável. Entendeu-se depois que estes valores, estavam associados a falhas de transmissão nos dados por parte da Gestão Técnica Centralizada (GTC) do hotel. Quando ocorre a referida falha, as leituras são acumuladas do lado do emissor até ser retomada a transmissão,

3.1. ANÁLISE EXPLORATÓRIA DOS DADOS

sendo nessa altura devolvido o valor acumulado, o que produz um aparente pico no consumo, que efetivamente não se verificou, e gera-se um intervalo de tempo sem leituras, tal como apresentado com maior definição temporal na Figura 3.2.

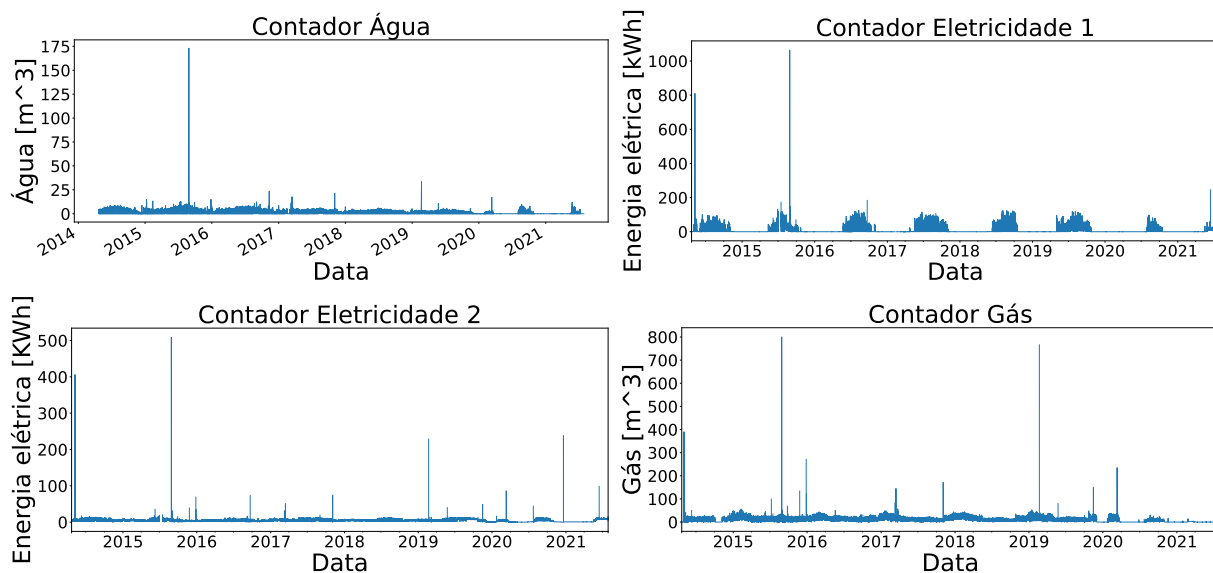


Figura 3.1: Representação das séries de leituras, associadas a diferentes contadores do hotel, onde se apresentam de forma clara picos de consumos ocasionais.

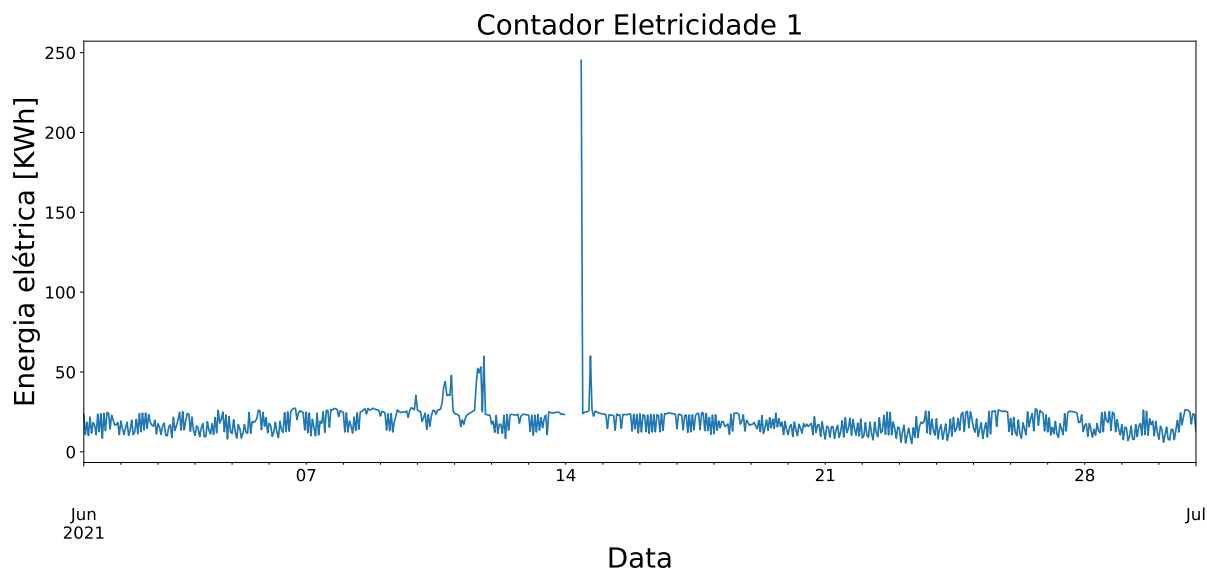


Figura 3.2: Pormenor de falha técnica na receção de leituras do contador "Eletricidade 1".

Efetuuou-se também uma análise aos dados com o intuito de entender o comportamento dos mesmos com o decorrer dos anos. Na Figura 3.3 são apresentados os consumos semanais por ano dos contadores "Eletricidade 1" e "Gás", evidenciando-se

a existência de uma sazonalidade anual, ou seja, em anos típicos os consumos tendem a seguir um padrão. O ano 2020 acaba por sair um pouco da regra devido à pandemia Covid-19 e o ano 2021 não está representado nesta análise por não existirem as leituras completas. Numa perspetiva diária, analisando algumas semanas aleatórias dos diferentes contadores, verificou-se também uma periodicidade nos consumos, concluindo-se que tipicamente de segunda a domingo, os consumos tendem a manter um padrão horário, conforme apresentado na Figura 3.4.

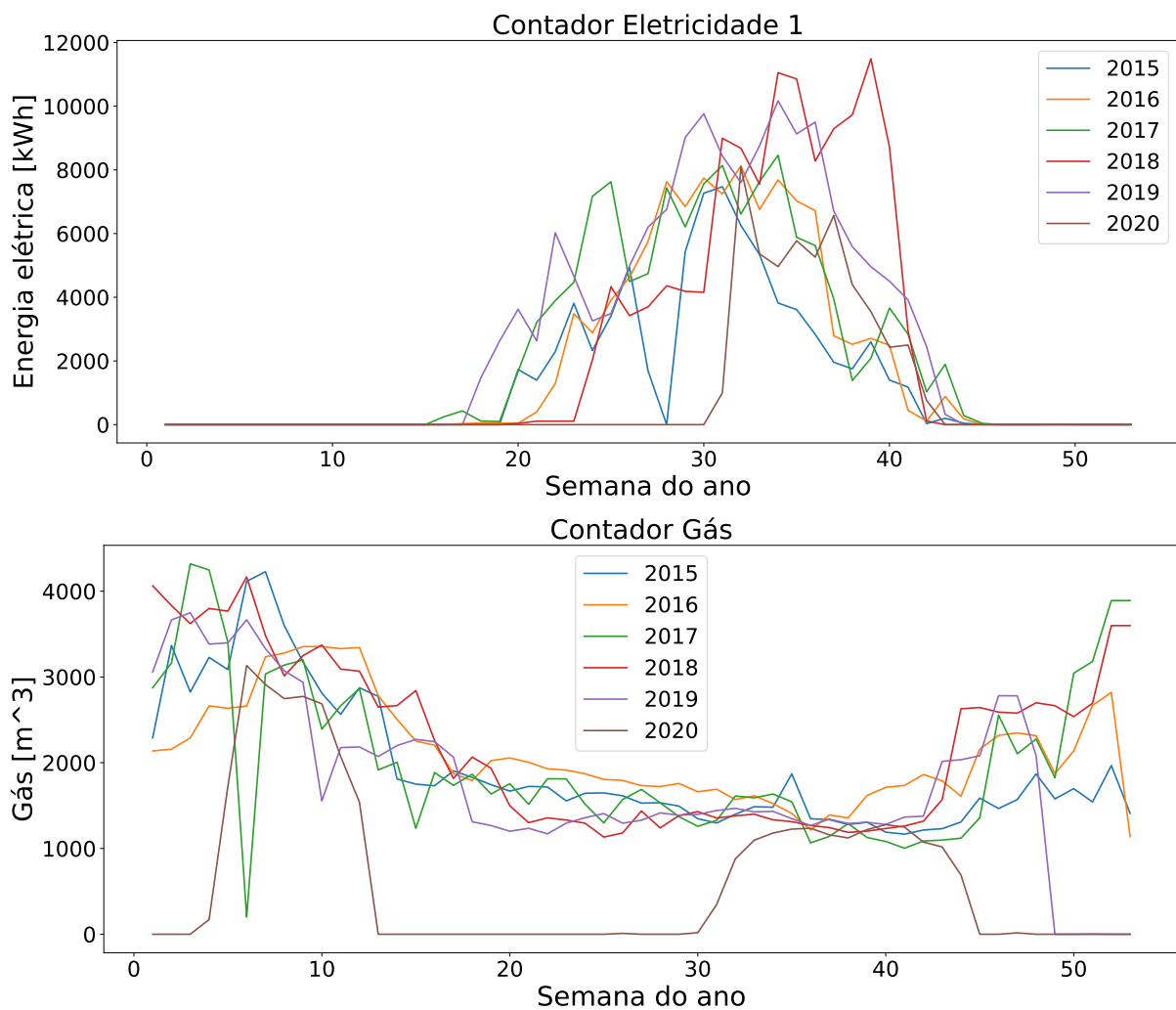


Figura 3.3: Análise anual para os contadores "Eletricidade 1" e "Gás".

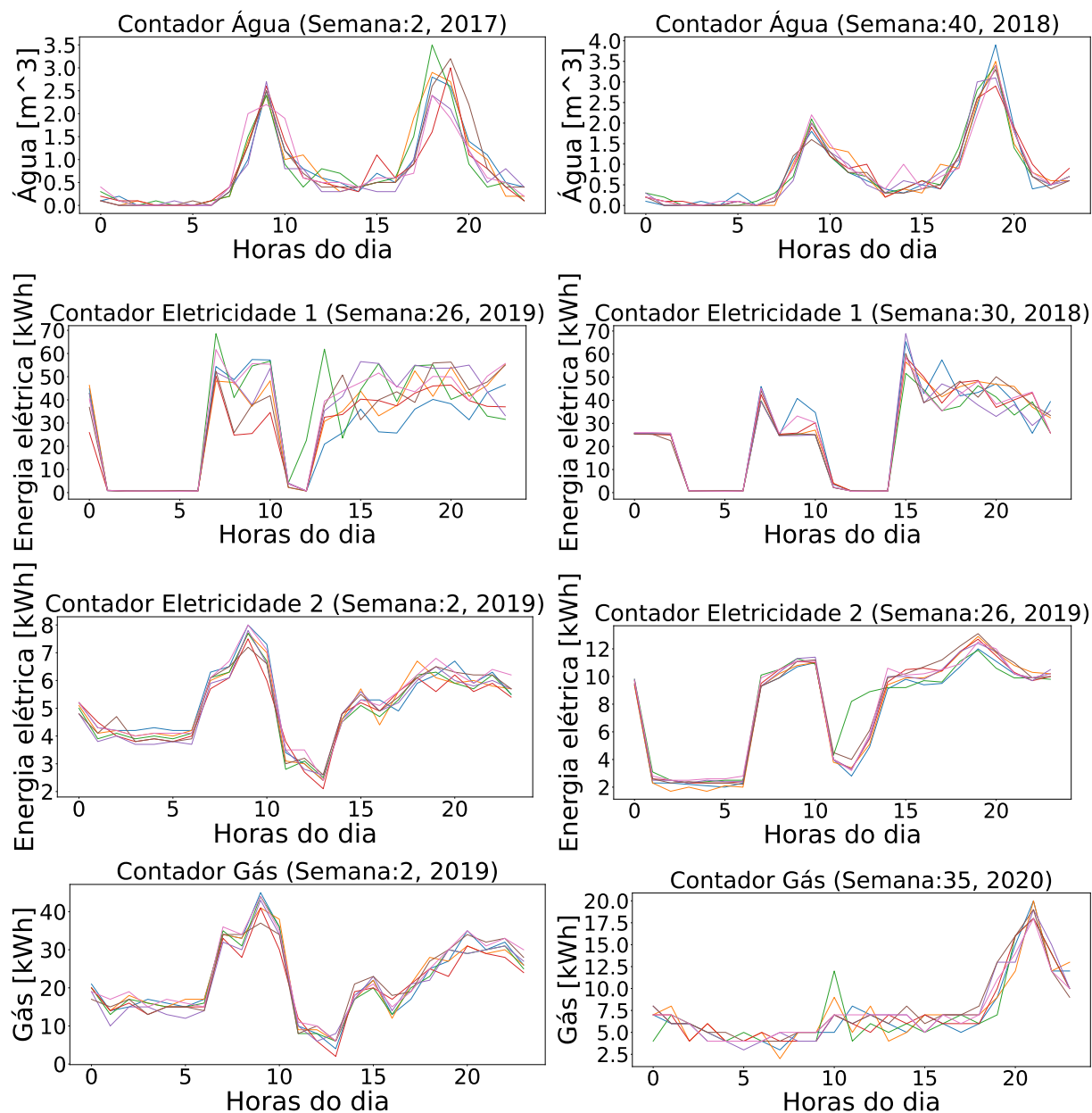


Figura 3.4: Representações dos consumos diários em períodos semanais, para os 4 contadores em estudo.

3.1.2 Análise estatística

Numa segunda fase procedeu-se a uma análise estatística dos 4 contadores com o cálculo da média, desvio padrão, quartis (Q_1 , Q_2 e Q_3) e os valores mínimos e máximos das leituras, tal como apresentado na Tabela 3.1. Destes dados puderam retirar-se algumas conclusões. Por exemplo, o valor máximo identificado nos quatro contadores encontra-se muito distante dos valores registados no terceiro quartil (Q_3). Em concreto, no caso do contador "Eletricidade 1", existe pelo menos uma leitura com 1064

Contadores	Água (m ³)	Eletricidade 1 (kWh)	Eletricidade 2 (kWh)	Gás (m ³)
média	0,83	4,73	11,61	9,95
desvio padrão	1,39	4,13	21,89	10,66
min	0,00	0,00	0,00	0,00
25%	0,00	2,70	0,00	4,00
50%	0,40	4,30	0,00	9,00
75%	1,10	6,90	18,80	14,00
máx	173,10	508,60	1064,60	800,00

Tabela 3.1: Descrição estatística de alguns contadores.

kWh de consumo elétrico horário o que é muito distante dos pelo menos 75 % dos dados, verificando-se uma situação semelhante para o contador "Água", com uma leitura horária de 173,10 m³. Qualquer um dos casos sugere, pelo menos, a possibilidade de existirem dados anómalos nas leituras disponíveis. Outro valor que sugere erros nos dados é o valor do desvio padrão, evidenciando uma grande amplitude de variação em algumas leituras em relação ao valor médio, o que mais uma vez sugere a presença de leituras problemáticas.

3.1.3 Histogramas

Numa última fase da análise, a utilização de histogramas foi importante no sentido de se confirmarem as suspeitas retiradas até então. Assim, conforme se pode ver na Figura 3.5 (note-se que a escala das ordenadas é logarítmica), é possível perceber quais os intervalos de valores mais recorrentes e perceber se os dados poderiam obedecer a alguma distribuição de dados conhecida ou não. As leituras previamente destacadas na análise estatística como possíveis anomalias, são leituras com frequências 1, o que, anexo à sua distância das restantes e conhecendo a física associada aos respetivos contadores, as garante como sendo problemáticas.

3.1.4 Conclusões da análise exploratória

Com a análise exploratória, da qual aqui se fez um resumo, conclui-se que os dados apresentam uma sazonalidades (p.e., anual e semanal); e que os maiores problemas a

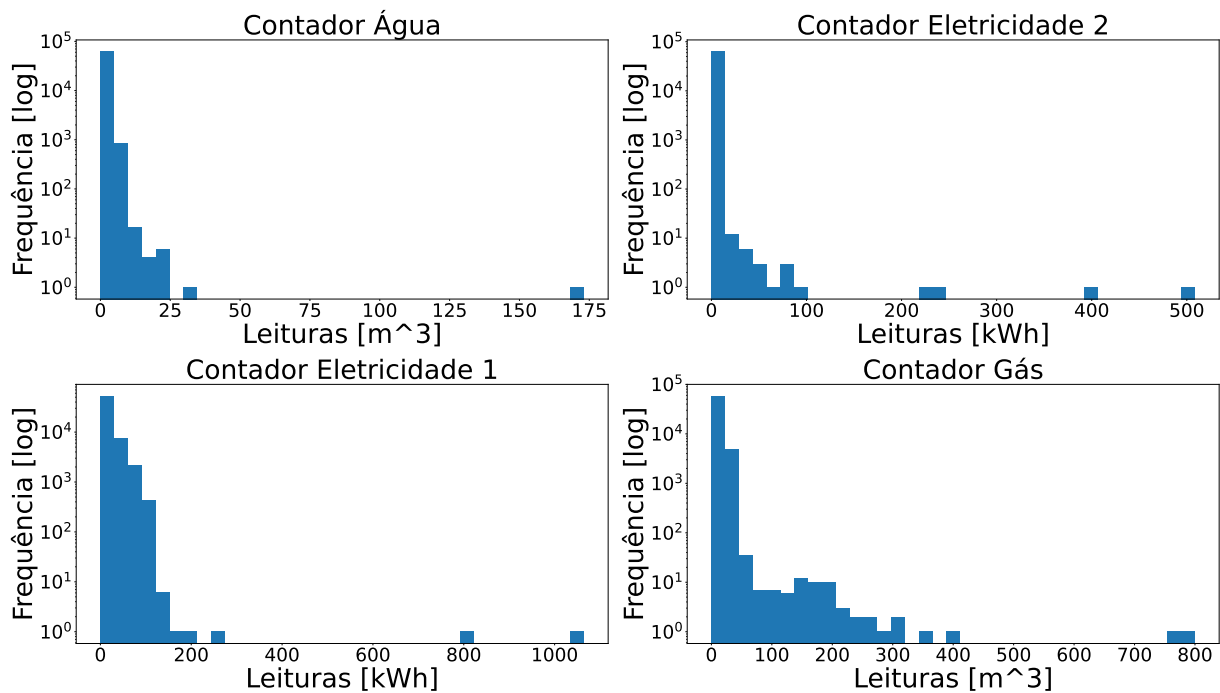


Figura 3.5: Histogramas dos contadores "Água", "Eletricidade 1", "Eletricidade 2" e "Gás" do hotel.

afetarem a qualidade dos dados são: (i) os valores de *outliers* e (ii) leituras em falta. Mais tarde foi possível concluir que estes problemas têm origem em falhas técnicas associadas à transmissão de dados do hotel. Para tentar resolver os problemas identificados, optou-se por uma abordagem assente em métodos estatísticos que substitui e/ou introduz valores estimados nas leituras anómalas ou em falta, respetivamente, tal como será proposto nas secções seguintes.

3.2 Correção dos dados

Após a realização da análise exploratória e identificação dos problemas que afetavam a qualidade dos dados, iniciou-se um estudo de soluções para corrigi-los e, assim, viabilizá-los para os algoritmos de AM. Como apresentado nas próximas secções, primeiro encontrou-se uma solução para fazer a deteção e remoção de *outliers* e depois uma para imputação de valores em falta nas séries de leituras.

3.2.1 Detecção e remoção de *outliers*

O primeiro problema que se tentou resolver foi a deteção e remoção de *outliers*, i.e., valores problemáticos que interferem diretamente com a qualidade dos dados e passíveis de gerar falsos alarmes ou mais tarde reduzir a identificação de anomalias reais, no sistema de deteção de consumos anómalos já que o algoritmo irá "aprender" com esses valores. Na escolha da abordagem a implementar, a necessidade de encontrar uma solução que fosse o mais genérica possível (i.e., se adaptasse a diferentes contadores) e fosse simples em ambiente de produção era essencial (e.g., considerando que alguns dos futuros utilizadores da aplicação possam ter conhecimentos limitados em termos de ciência dos dados e AM). Por estas razões, ponderou-se entre dois métodos estatísticos, para desempenhar estas funções: o IIQ (Secção 2.2.1) e o MDP (Secção 2.2.2).

A utilização do MDP foi a primeira a ser ponderada. Com a análise efetuada aos dados, explicada no início deste capítulo, esta solução mostrou ser pouco robusta para o que pretendíamos, pois não seria garantido que todos os contadores que viessem a ser analisados pelo nosso programa fossem respeitar uma forma de distribuição normal, para o método poder ser aplicado corretamente. De forma natural, então a solução adotada acabou por ser o método do IIQ, por ter menos limitações associadas às características dos dados.

Por exemplo, como resultado do tratamento dos dados com o método IIQ, quando comparado com os dados apresentados na Figura 3.1, é possível verificar na Figura 3.6 que a qualidade dos dados melhorou significativamente, tendo sido removidas praticamente todas as leituras que visualmente eram passíveis de ser consideradas problemáticas. Neste contexto, constatou-se que o método deve ser ajustado em função de cada contador (o parâmetro K do método IIQ), de forma a preservar o máximo de leituras corretas. Na Figura 3.7 apresenta-se para o contador "Eletricidade 1" um intervalo de tempo de um mês, com um antes e um depois da remoção dos *outliers* (de acordo com o método, considerando $K = 5$). Depois de realizado este procedimento, efetuou-se a imputação de valores em falta, tal como apresentado na próxima secção.

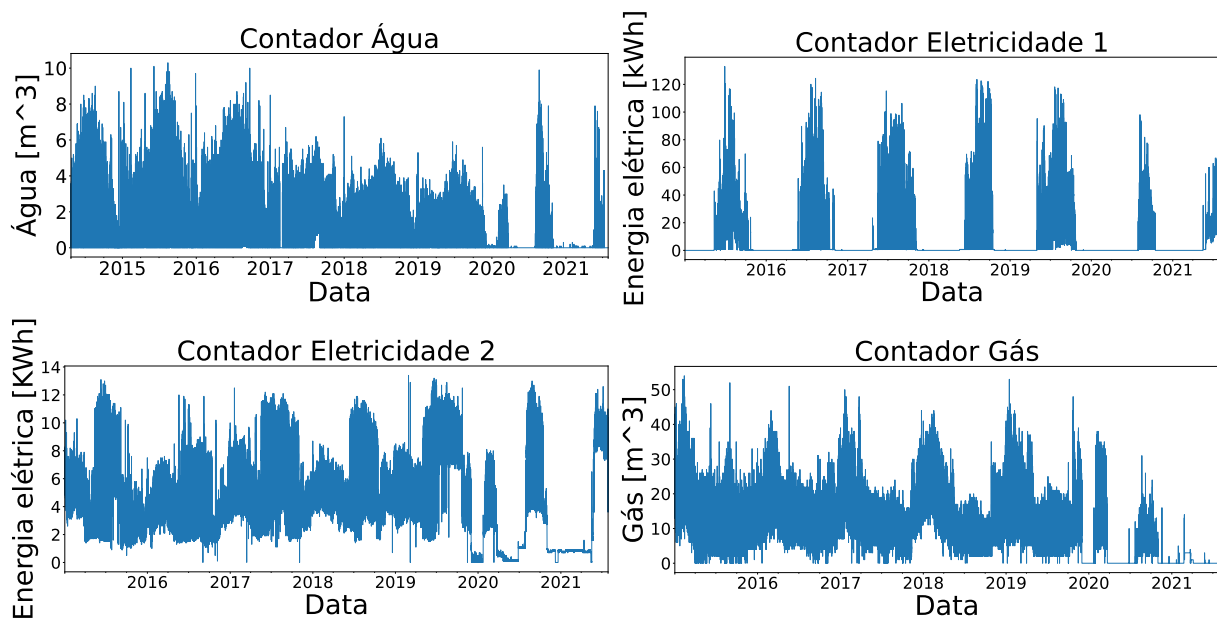


Figura 3.6: Representação das séries de leituras com as devidas correções, associadas a diferentes contadores do hotel.

3.2.2 Imputação de valores em falta

A imputação de valores em falta numa série temporal pode ser essencial para garantir a viabilidade da sua utilização em certos algoritmos de AM. Por esta razão, e para futuramente existir um histórico de leituras com melhor qualidade e fiabilidade na empresa, fazia sentido encontrar uma solução que permitisse estimar valores em falta por falha de transmissão ou por remoção após a primeira análise de valores anómalos.

A utilização de soluções mais complexas baseadas em AM, face às nossas necessidades de implementação, não eram as mais apelativas, pois acrescentavam toda uma complexidade na sua utilização, nomeadamente, e como já foi referido, por utilizadores que não são especialistas na área de AM e ciência dos dados. Assim considerou-se que a necessidade de um modelo por cada contador, seleção dos melhores parâmetros, treinos constantes e o garantir que no final os resultados justificassem a sua implementação, seria bastante difícil de colocar em prática. Por este motivo, o método que acabou por ser escolhido foi o algoritmo da Média Móvel Ponderada Linear (MMPL), formulado de acordo com a seguinte equação:

$$L_{ponderada} = \sum_{i=1}^n L_i w_i, \quad (3.1)$$

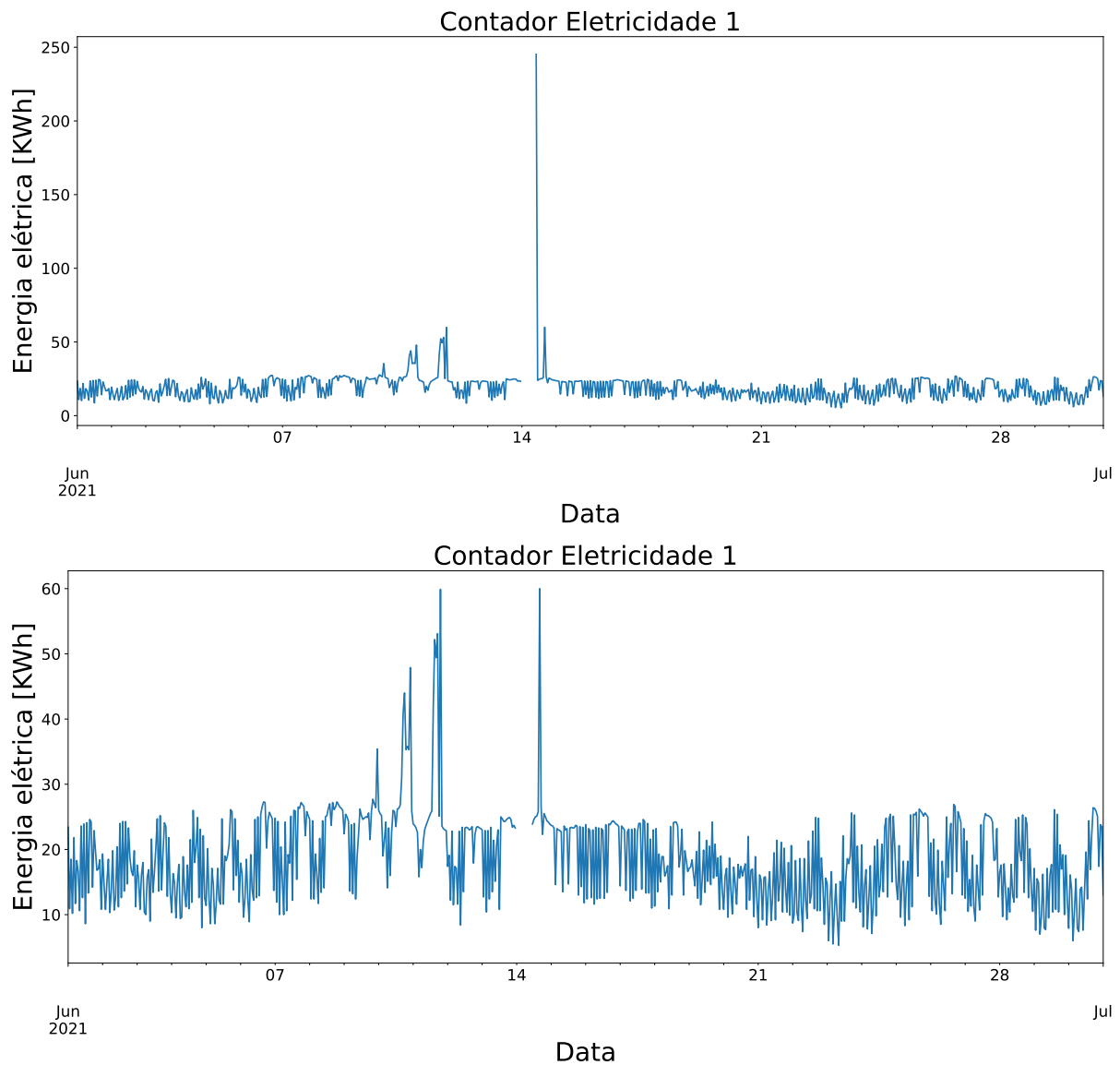


Figura 3.7: Pormenor de uma falha na série de leituras, associada ao contador "Eletricidade 1". O antes e depois da remoção de leituras.

onde a leitura ponderada, $L_{ponderada}$, é calculada com base em n leituras de dias imediatamente anteriores (L_i , $i = 1, 2, \dots, n$) ocorridas no mesmo instante da leitura ponderada. Esta solução é suportada na verificação que a generalidade dos contadores apresentam uma periodicidade semanal (ver Secção 3.1.1). Mais, cada leitura passada é afetada por um peso (w_i), dado por $w_i = \frac{2(n-i+1)}{n(n+1)}$, $i = 1, 2, \dots, n$, pois conforme se pode constatar nas Tabelas 3.2 e 3.3, existe uma maior correlação entre as leituras mais recentes nos diferentes contadores analisados.

De referir que a MMPL tem sido utilizada noutros trabalhos onde foi necessário imputar valores, concluindo-se nos mesmos, que o algoritmo foi efetivo na resposta

3.2. CORREÇÃO DOS DADOS

Nº de dias anteriores	Água	Eletricidade 1	Eletricidade 2	Gás
0	1,00	1,00	1,00	1,00
1	0,66	0,86	0,91	0,86
2	0,65	0,83	0,88	0,83
3	0,63	0,82	0,86	0,81
4	0,62	0,79	0,84	0,80
5	0,61	0,78	0,83	0,79

Tabela 3.2: Valores das correlações existentes entre leituras às 12:00 e as leituras ocorridas no mesmo instante em dias anteriores.

Nº de dias anteriores	Água	Eletricidade 1	Eletricidade 2	Gás
0	1,00	1,00	1,00	1,00
1	0,63	0,96	0,96	0,92
2	0,59	0,95	0,95	0,90
3	0,55	0,93	0,93	0,89
4	0,52	0,91	0,91	0,88
5	0,50	0,88	0,90	0,87

Tabela 3.3: Valores das correlações existentes entre as leituras às 00:00 horas e as leituras ocorridas no mesmo instante em dias anteriores.

dada (Hassani et al., 2019; Demirhan and Renwick, 2018).

A título de exemplo, é possível verificar na Figura 3.8 que as leituras em falta no contador "Eletricidade 1" foram introduzidas (utilizando a MMPL), o que levou a que fosse impercetível que tenha existido este problema depois da correção dos dados.

Partindo do princípio que os dados estavam prontos para ser aplicados nos algoritmos de deteção de anomalias selecionados deu-se início à fase experimental, conforme será apresentado no próximo capítulo.

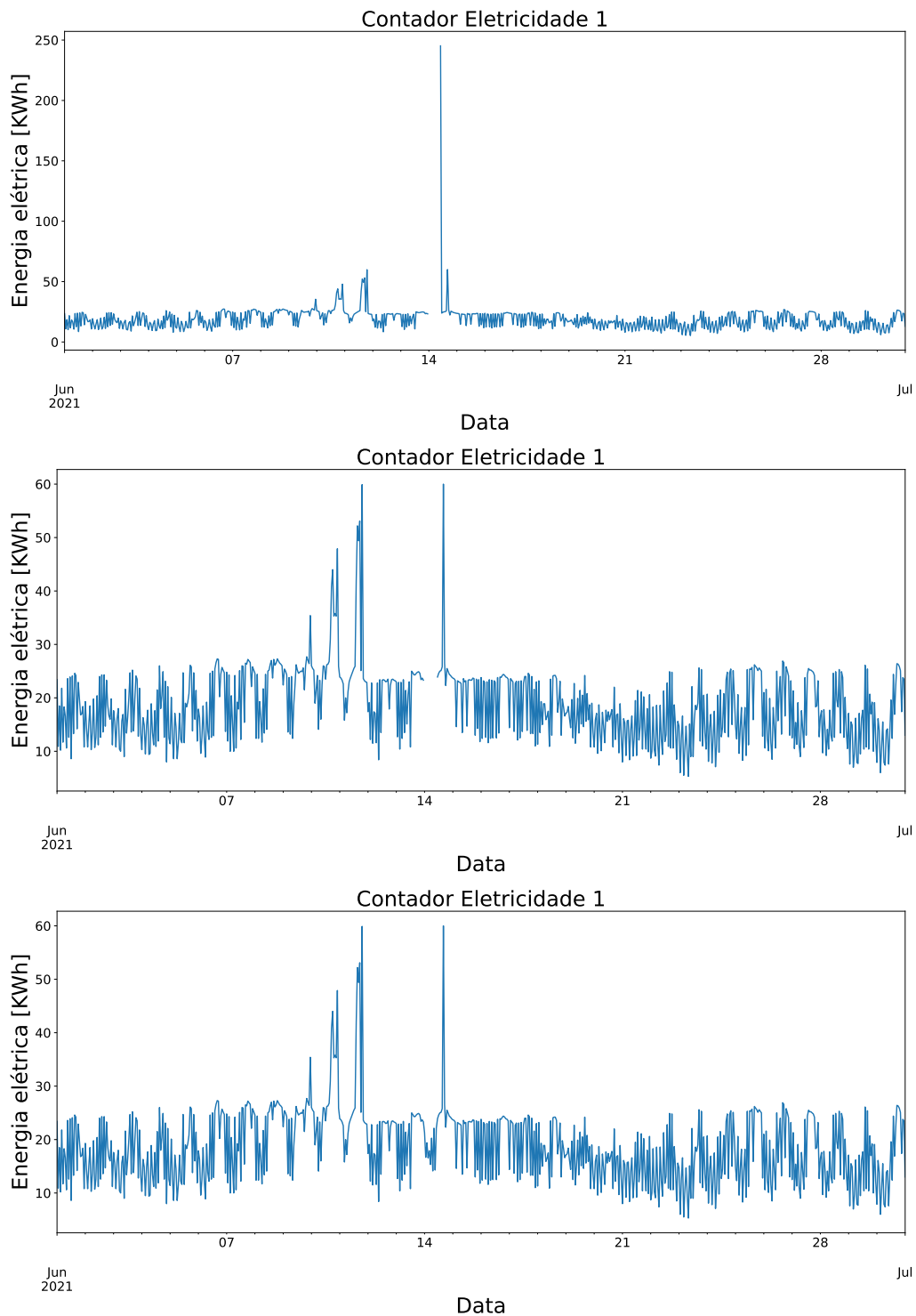


Figura 3.8: Pormenor do que antes foi uma falha na série de leituras, associada ao consumo de energia do contador "Eletricidade 1", e agora corrigida segundo os métodos propostos: dados originais (topo), após remoção de picos (meio) e após introdução de valores estimados para os valores em falta (em baixo).

4

Resultados experimentais

Este capítulo é dedicado à exploração dos resultados dos testes efetuados com os modelos de deteção de anomalias selecionados. Na primeira fase é apresentada a bateria de testes efetuada e, com maior detalhe, explicada a métrica de performance proposta para avaliar o desempenho dos algoritmos na classificação de anomalias em séries temporais não-supervionadas. No restante capítulo são apresentados os resultados do estudo realizado usando os modelos IF e VAE, por forma a decidir-se qual o modelo a ser integrado no sistema de deteção de consumos anómalos para hotéis.

De referir que numa fase precoce do estudo foram ainda ponderados outros métodos, tendo-se optado, por questão de simplicidade, não os apresentar aqui.

4.1 Configuração dos testes experimentais

4.1.1 Conjunto de dados

De modo a testar e afinar os algoritmos sobre dados semelhantes aos que serão processados no futuro, constitui-se um conjunto de testes computacionais. Neste conjunto de testes foram analisadas diferentes configurações de hiper parâmetros dos algoritmos selecionados (i.e., IF e VAE). Relativamente aos dados, usaram-se como base 2 anos (2015 e 2016) de dados de um contador para treino e 1 ano para testes (2017), sendo que foram introduzidas anomalias no ano de teste, tal como representado na Figura 4.1, onde as anomalias introduzidas estão marcadas a vermelho.

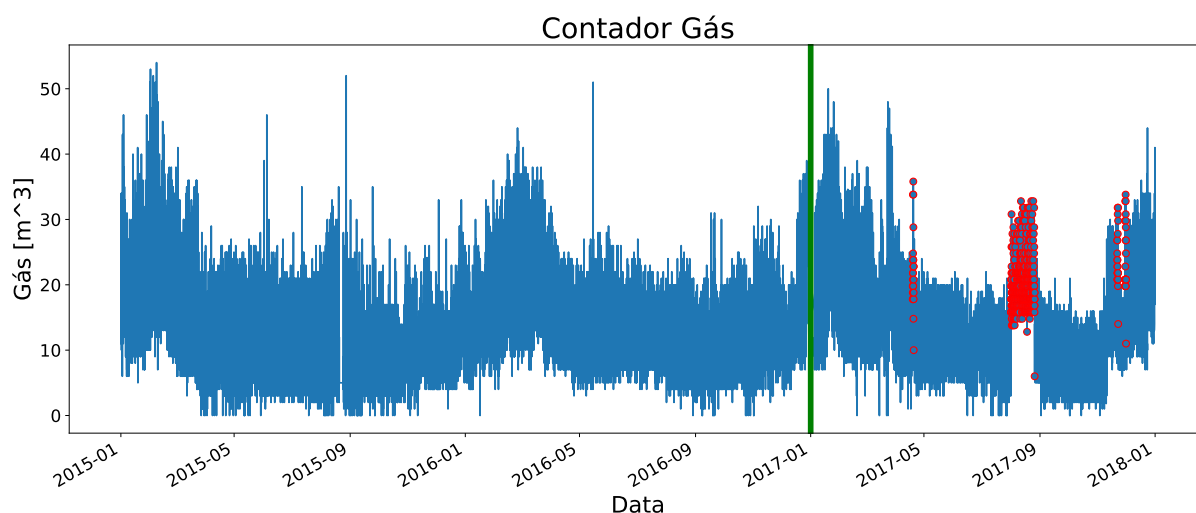


Figura 4.1: Representação gráfica do conjunto de dados usado nos testes computacionais. A vermelho as anomalias introduzidas no ano de teste.

Os diferentes cenários testados (combinação dos dados com as parametrizações dos algoritmos) tiveram como principais objetivos:

1. Avaliar o desempenho dos algoritmos na classificação de anomalias em diferentes contadores;
2. Descobrir um possível conjunto de variáveis e hiperparâmetros para *baseline* no treino dos modelos;
3. Estudar o comportamento dos algoritmos quando expostos a diferentes combinações de variáveis de entrada, pois, o sistema de deteção a ser criado tem a

necessidade de ser bastante adaptável a este tipo de situações.

Para perceber o nível de precisão na deteção de *outliers* pelos algoritmos, procedeu-se à introdução de anomalias. Neste sentido, foram introduzidas quatro anomalias no ano de teste, com o intuito de simularem consumos excessivos durante esse período. Os dados escolhidos para a realização de testes, tiveram em atenção a análise realizada na Secção 3.1, a qual mostrou serem anos típicos do hotel.

Foram considerados 4 cenários de variáveis de entrada, conforme apresentados na Tabela 4.1.1. Com base nas leituras dos contadores, utilizou-se a própria leitura e criaram-se variáveis como:

Δ_X - conjunto de seis *features*, em que X pode tomar valores em $[1h, 2h, 3h, 24h, 48h, 72h]$, que refletem a diferença entre a leitura atual e a leitura com desfasamento de X horas (p.e., Δ_{2h} é a diferença entre a leitura atual e a leitura ocorrida 2 horas antes). Pretendem auxiliar na deteção de variações anormais;

$\Delta_{média_{24h}}$ - diferença entre a leitura atual e a média das últimas 24 horas. Tal como as *features* anteriores pretendem auxiliar na deteção de variação anormal nos consumos;

\min_{24h} - identifica o consumo mínimo das últimas 24 horas e pretende fornecer uma sensibilidade à magnitude das leituras.

Testaram-se ainda variantes destas, identificadas com um b na Tabela 4.1.1, cujo objetivo foi transformar as diferenças em 0 ou 1, conforme se verificasse um aumento ou decréscimo do consumo. Utilizaram-se também variáveis temporais e variáveis associadas ao hotel como p.e. "temperaturas", "ocupação", "Grau Dia" (medida das necessidades de aquecimento ou arrefecimento do edifício) como sugerido no artigo realizado por Eras et al. (2016). De referir ainda que outras variáveis de entrada (p.e., o máximo de 24 horas) foram testadas, mas por questão de simplicidade de apresentação e por não existirem limites para a constituição de novas variáveis, optou-se, por mostrar os resultados apenas para as variáveis mais promissoras, que são aqui detalhadas.

Cenários	Variáveis Contador	Variáveis Temporais	Variáveis Hotéis
1	Leitura, Δ_{1h} , $\Delta_{2h}, \Delta_{3h}, \Delta_{24h}, \Delta_{48h}, \Delta_{72h}$, \min_{24h} , $\Delta_{m\u00e9dia_{24h}}$	Hora, Dia da Semana, M\u00eas	Nenhuma
2	Leitura, Δ_{1h}^b , $\Delta_{2h}^b, \Delta_{3h}^b, \Delta_{24h}^b, \Delta_{48h}^b, \Delta_{72h}^b$, \min_{24h}^b , $\Delta_{m\u00e9dia_{24h}^b}$	Hora, Dia da Semana, M\u00eas	Nenhuma
3	Leitura, Δ_{1h} , $\Delta_{2h}, \Delta_{3h}, \Delta_{24h}, \Delta_{48h}, \Delta_{72h}$, \min_{24h} , $\Delta_{m\u00e9dia_{24h}}$	Hora, Dia da Semana, M\u00eas	Temperatura, Ocupa\u00e7\u00e3o, Grau Dia, N\u00b0Refei\u00e7\u00f5es Di\u00e1rias, Quartos Ocupados
4	Leitura, Δ_{1h}^b , $\Delta_{2h}^b, \Delta_{3h}^b, \Delta_{24h}^b, \Delta_{48h}^b, \Delta_{72h}^b$, \min_{24h}^b , $\Delta_{m\u00e9dia_{24h}^b}$	Hora, Dia da Semana, M\u00eas	Temperatura, Ocupa\u00e7\u00e3o, Grau Dia, N\u00b0Refei\u00e7\u00f5es Di\u00e1rias, Quartos Ocupados

Tabela 4.1: Diferentes conjuntos de vari\u00e1veis de entrada dos algoritmos, utilizados nos testes.

Avaliar a qualidade dos algoritmos na detec\u00e7\u00e3o de anomalias em s\u00e9ries temporais n\u00e3o supervisionadas, pode n\u00e3o ser t\u00e3o exato no sentido de se considerar apenas as m\u00e9tricas simples de classifica\u00e7\u00e3o, como a taxa de falsos positivos ou falsos negativos, pois n\u00e3o se conhecendo os dados torna-se dif\u00edcil confirmar a classifica\u00e7\u00e3o efetuada pelos algoritmos. Por esta raz\u00e3o, criou-se uma m\u00e9trica de avalia\u00e7\u00e3o de desempenho, para casos em que as anomalias s\u00e3o conhecidas ou imputadas, que \u00e9 apresentada de seguida.

4.1.2 M\u00e9trica de avalia\u00e7\u00e3o de desempenho

A m\u00e9trica de avalia\u00e7\u00e3o de desempenho desenvolvida teve como objetivo ser uma solu\u00e7\u00e3o simples, na avalia\u00e7\u00e3o e compara\u00e7\u00e3o dos algoritmos. Assim, esta tem por base algumas das ideias do sistema da empresa Numenta (Lavin and Ahmad, 2015), que visa avaliar o desempenho dos algoritmos sobre um determinado conjunto de dados de s\u00e9ries temporais, tendo em considera\u00e7\u00e3o fatores como sejam a rapidez de detec\u00e7\u00e3o ou incid\u00eancia de falsos positivos.

O sistema criado \u00e9 formulado pela combina\u00e7\u00e3o de 2 pontua\u00e7\u00f5es: P_1 \u00e9 a pontua\u00e7\u00e3o que avalia a incid\u00eancia de falsos positivos, e P_2 a pontua\u00e7\u00e3o referente \u00e0 classifica\u00e7\u00e3o de

anomalias. A pontuação final, P_{final} , combina as duas anteriores e é responsável por atribuir a pontuação de desempenho do algoritmo para um determinado cenário de teste. As pontuações variam no intervalo $[0;1]$, e quanto mais perto de 1 for o valor da métrica melhor será a performance.

Mais detalhadamente, P_1 é obtida a partir da expressão:

$$P_1 = \frac{1}{1 + \exp\left(\frac{PDC - TPD \times k_1}{TPD \times k_2}\right)}, \quad (4.1)$$

onde PDC traduz o número de pontos detetados como anomalia pelo algoritmo, mas que tais resultados não são possível validar (i.e., não pertencem às anomalias introduzidas), TPD reflete o número total de pontos desconhecidos na amostra de teste e k_1 e k_2 são valores que permitem decidir o nível de penalização associado. Com a pontuação P_1 permite-se avaliar o algoritmo na componente dos falsos positivos. No nosso caso considerou-se $k_1 = 0,1$ e $k_2 = 0,01$, por forma a garantir que existisse uma tolerância na classificação de 5 % das leituras dentro do grupo TPD como anomalias e que quando classificadas 10 % destas leituras P_1 fosse igual a 0,5.

A Figura 4.2 representa graficamente a evolução da pontuação P_1 em função do PDC para um $TPD = 8856$, que foi o cenário estudado durante os testes realizados. Assim, o que acontece é que se o algoritmo classificar 300 leituras como anomalias, ou seja, $PDC = 300$ dentro do universo das 8856, o desempenho do algoritmo não irá sofrer penalizações significativas, pois 5 % dos 8856 são 442,8. No entanto, se o valor do PDC for superior, começa a existir uma penalização que é refletida na pontuação P_1 , e.g, com $PDC = 750$ o valor de P_1 é 0,82.

O valor de P_2 que é obtido a partir da expressão:

$$P_2 = \frac{2}{1 + \exp\left(10 \frac{1PAC - LSA}{LSA - LIA}\right)} - 1 \quad (4.2)$$

onde os índices LIA e LSA traduzem, respetivamente, o primeiro e último índice do intervalo de anomalia e $1PAC$ a primeira leitura detetada pelo algoritmo dentro da janela da anomalia. P_2 pontua o desempenho na classificação das anomalias e a celeridade do

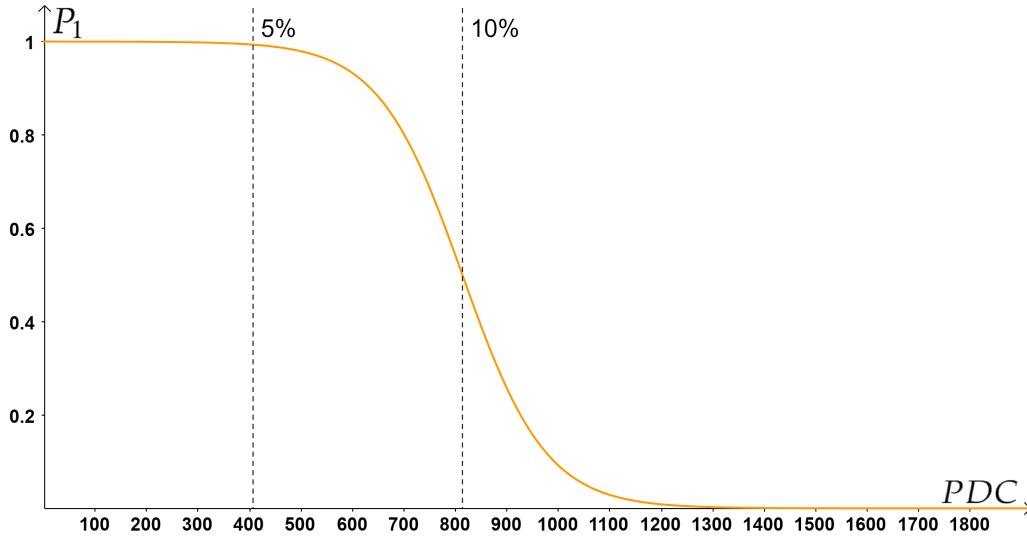


Figura 4.2: Exemplo ilustrativo da evolução da métrica P_1 em função do número de pontos desconhecidos classificados como anómalos (PDC), com o TPD=8856.

algoritmo a detetar as mesmas. Os restantes parâmetros foram seleccionados por forma a que a penalização dependa da extensão da anomalia e apenas se torna mais evidente a meio da mesma.

Na Figura 4.3 está representada a evolução de P_2 em função do 1PAC, para uma anomalia constituída por 24 pontos, o que, para os testes realizados, corresponderia a uma anomalia com duração de 24 horas. Quando o 1PAC é inferior a 12, ou seja, a deteção da anomalia é feita nas primeiras 12 horas não existe penalização, apenas quando o atraso na deteção começa a ser significativo é que o desempenho do algoritmo também começa a ser penalizado.

Por fim o valor final da métrica de desempenho do algoritmo é dado por

$$P_{final} = P_1 \frac{\sum_{i=1}^n P_{2,i}}{n}, \quad (4.3)$$

onde $P_{2,i}$ é o valor de P_2 para o intervalo anómalo i . A fórmula consiste na média de pontuação da deteção de anomalias (P_2), por forma a normalizar a pontuação quando consideradas anomalias com diferentes extensões, penalizada com a classificação dos pontos desconhecidos (P_1). No final, a intenção é que no valor final da métrica, P_{final} , seja refletido um balanço entre a deteção de anomalias e incidência de falsos positivos. Por exemplo $P_{final} = 1$ significa que todas as anomalias foram detatadas rapidamente

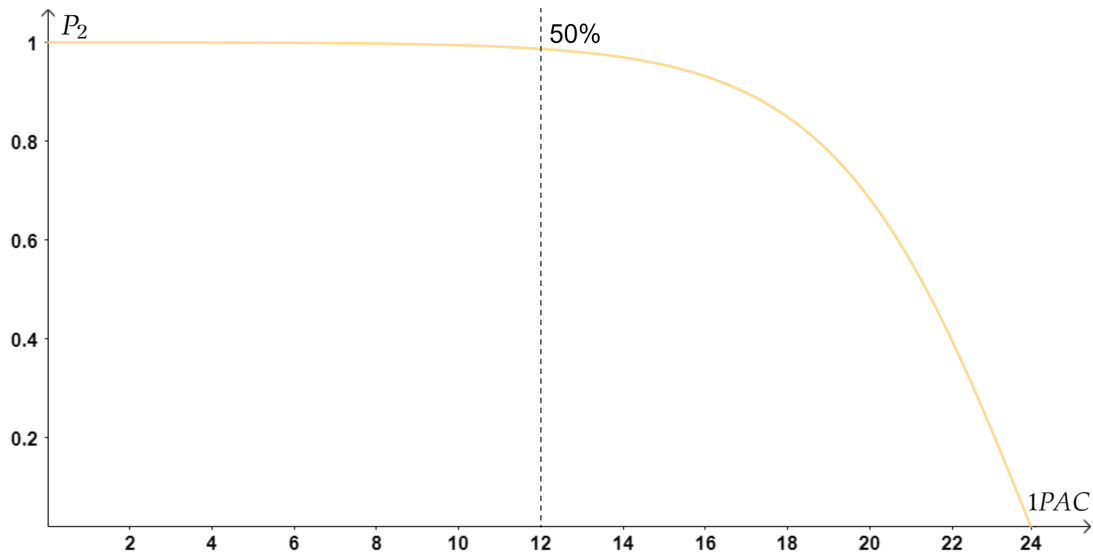


Figura 4.3: Exemplo ilustrativo da evolução do valor de P_2 em função do valor de $1PAC$, para uma anomalia constituída por 24 leituras.

e o número de pontos dos quais não se consegue aferir a classificação do algoritmo, manteve-se baixo.

4.2 Resultados computacionais

Nesta secção serão apresentados os melhores resultados computacionais obtidos com os algoritmos IF e VAE, em função da métrica de avaliação de desempenho apresentado na Secção 4.1. Os resultados serão comparados no sentido de decidir qual o algoritmo a adotar para o sistema de deteção de anomalias em consumos, acentando não só no valor da métrica, mas também dos recursos computacionais requeridos.

4.2.1 Isolation forest

Os testes computacionais realizados com o algoritmo IF utilizaram a implementação do mesmo da biblioteca Sklearn (Pedregosa et al., 2011). Em termos de configurações do algoritmo, foi feita uma *grid search* sobre os hiperparâmetros `contamination` $\in \{0.008, 0.01, 0.03, 0.05, 0.1\}$, `bootstrap` $\in \{True, False\}$ e `max_samples` $\in \{'auto', 0.05, 0.1, 0.3, 0.5\}$. Para os restantes hiperparâmetros (consultar a documentação da

classe IF¹) foram considerados os valores predefinidos da biblioteca. Os melhores resultados obtidos pelo algoritmo IF estão sumariados na Tabela 4.2. Dos resultados conclui-se que, de acordo com a métrica definida na Secção 4.1.2, o algoritmo teve uma boa prestação em todos os contadores na detecção de pontos desconhecidos ($P_1 = 0.99$), ou seja, o número de leituras não identificadas classificadas como anómalas manteve-se dentro da tolerância, e também na detecção de anomalias em fase inicial (P_2). Uma ressalva é a anomalia " P_2 Anomalia 2" no contador "Eletricidade 1", que não foi possível identificar, o que penalizou a P_{final} deste, alcançando apenas 0.73 pontos na escala.

Analisando os resultados confere-se que, se for este o modelo escolhido, é necessário efetuar um ajuste de parâmetros e variáveis de entrada em função de cada contador. Ao nível das variáveis, constatou-se que 3 dos melhores resultados encontrados estão associados a cenários onde variáveis do hotel estavam presentes, o que indica a sua relevância e que a empresa deve considerar introduzir as mesmas sempre que possível. Na escolha do conjunto de variáveis para base de treino dos modelos, ficou evidente que as leituras do contador, variáveis temporais e min_{24h} parecem melhorar a performance do algoritmo. Os resultados foram pouco claros relativamente à integração das variáveis Δ_X e $\Delta_{média_{24h}}$, em particular se as mesmas devem ser consideradas com a diferença calculada ou na perspectiva binária, de aumento/diminuição de consumo. Por isso, para tomar uma decisão fez-se uma análise crítica à situação e entendeu-se que uma abordagem binária poderia trazer alguma perda de sensibilidade, em concreto colocando na mesma proporção grandes e pequenas variações. Relativamente aos hiperparâmetros não é possível tirar grandes conclusões dada, possivelmente, a diversidade de tipos de dados considerados pelo que manter parâmetros pré-estabelecidos parece uma solução viável. No contexto de utilização o parâmetro `contamination` deverá ser ajustado em função dos resultados pelo gestor do sistema. Salienta-se ainda que as várias simulações efetuadas demoraram em média cerca de 25 segundos (19056 amostras para treino e 9528 amostras para teste), o que foi bastante rápido e demonstra ser um algoritmo com pouco peso computacional, ratificando os

¹<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html>

Contador	Cenário	<i>contamination</i>	<i>max_samples</i>	<i>bootstrap</i>	P_1	P_2 Anomalia 1	P_2 Anomalia 2	P_2 Anomalia 3	P_2 Anomalia 4	P_{final}
Água	4	0,008	'auto'	True	0,99	0,99	0,78	0,99	0,99	0,94
Eletricidade 1	2	0,05	0.3	False	0,98	0,99	0,00	0,97	0,96	0,73
Eletricidade 2	3	0,1	'auto'	False	0,94	0,99	0,99	0,99	0,99	0,94
Gás	3	0,05	0.1	True	0,96	0,99	0,90	0,98	0,99	0,93

Tabela 4.2: Melhores resultados para cada contador com o algoritmo IF, segundo a métrica proposta, entre todos os cenários e considerando a referida *grid-search*.

resultados obtidos por de Santis and Costa (2020).

4.2.2 Variational autoencoder

Os testes realizados com o modelo VAE, utilizaram a arquitetura apresentada na Figura 4.4. Teve por base o trabalho de Pereira and Silveira (2018), ao nível da conceção da rede e parametrização da mesma, como o número de neurónios, funções ativação e camadas utilizadas. O codificador (elemento laranja da figura) e o decodificador (elemento verde da figura) foram ambos concebidos com uma camada Bi-direcional (Secção 2.4) de 128 células LSTM em cada direção, perfazendo um total de 256 células, com a função ativação tanh e como resultado a concatenação das duas camadas. O espaço latente Z foi modelado segundo uma distribuição Normal Isotrópica, sendo os parâmetros μ_z e σ_z obtidos através de duas camadas ANN *fully connected* que se ligam ao último estado (h_t) do codificador, com ativação *Linear* e *SoftPlus*, respetivamente. O espaço latente Z é calculado através da aproximação $Z = \mu_z + \sigma_z \epsilon$, onde ϵ é um valor aleatório da distribuição escolhida que atua como um ruído auxiliar. O vetor do espaço latente Z é então repetido t vezes, para então ser efetuada a reconstrução dos dados no decodificador.

Para efetuar a deteção de anomalias, utilizou-se a deteção por erro de reconstrução, onde são calculados os erros de reconstrução dos dados e a utilização de uma janela deslizante com o IIQ (ver Secção 2.2.1) para identificação das anomalias.

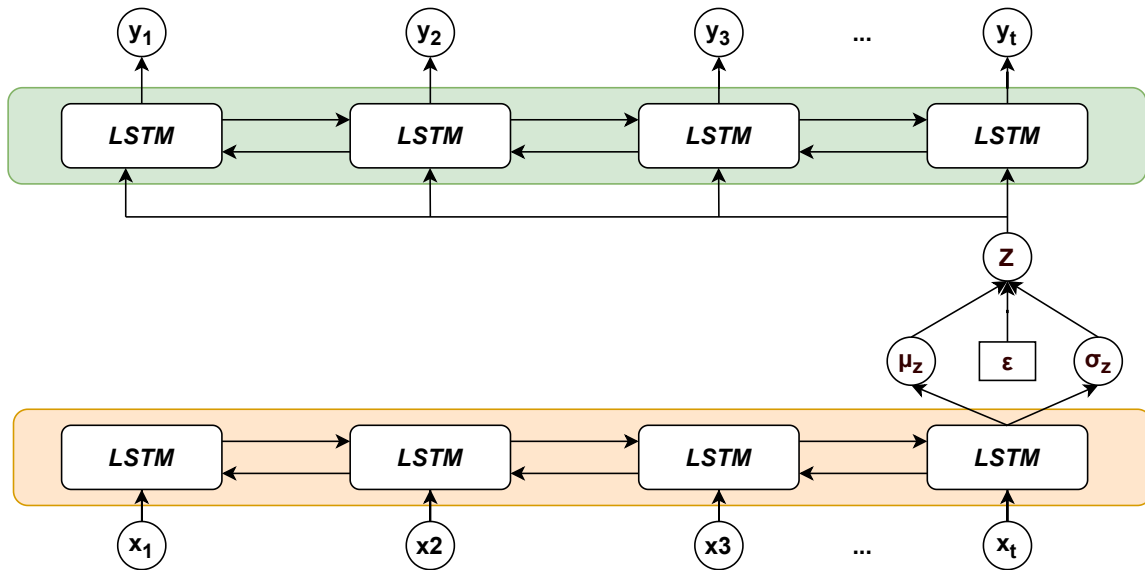


Figura 4.4: Arquitetura do VAE utilizado.

O modelo em si, foi programado com base na biblioteca computacional Keras² (Chollet et al., 2015). Os dados de entrada foram agrupados em sequências com dimensão $t = 24$, e a detecção de anomalias foi efetuada com base no erro de reconstrução, com recurso a uma janela deslizante (Aminikhanghahi and Cook, 2017; Hota et al., 2017). Dadas as restrições computacionais nos testes deste modelo, apenas se variou a largura da janela deslizante (24, 48 e 72), pois logo se verificou que as simulações podiam demorar cerca de 8h para 1500 ciclos de treino.

O modelo teve um bom desempenho em 3 dos contadores, sendo a exceção o contador "Água". Para este último, o desempenho foi mau em ambas as vertentes, i.e., tanto na detecção de pontos desconhecidos ($P_1 = 0,79$) como na detecção das anomalias, onde falhou uma delas e outra detetou com atraso, levando a uma pontuação final de 0,57. Também se verificaram problemas de convergência do modelo durante a realização dos testes.

Os cenários, com melhor desempenho neste modelo foram o 1 e o 2, ambos os que não possuíam as variáveis do hotel. Os resultados não foram propriamente conclusivos sobre a largura da janela deslizante a utilizar. Apesar de dois dos melhores resultados utilizaram uma janela com 24 erros, sendo que, um desses contadores foi o

²<https://keras.io/api/>

Contador	Cenário	<i>janela deslizante</i>	P_1	P_2 Anomalia 1	P_2 Anomalia 2	P_2 Anomalia 3	P_2 Anomalia 4	P_{final}
Água	1	24	0,79	0,99	0,00	0,99	0,90	0,57
Eletricidade 1	1	24	0,85	0,99	0,99	0,99	0,99	0,85
Eletricidade 2	2	48	0,96	0,99	0,99	0,99	0,99	0,96
Gás	2	72	0,99	0,99	0,99	0,85	0,99	0,95

Tabela 4.3: Melhores resultados para cada contador com o algoritmo VAE, segundo a métrica proposta, entre todos os cenários e considerando a referida grid-search.

da "Água" o qual não obteve um desempenho brilhante e os restantes contadores utilizaram uma largura de 48 e 72. Desta forma, conclui-se que este parâmetro deveria ser ajustável pelo utilizador caso este viesse a ser o modelo escolhido.

4.3 Conclusões

Comparando os resultados do VAE com os resultados da IF, chegou-se à conclusão de que o último seria uma melhor solução para o problema estudado, isto porque, segundo os contadores avaliados, apresentou resultados mais estáveis em termos de pontuação final. Por exemplo, a média do valor P_{final} para o algoritmo IF foi de 0,89, enquanto que para o VAE foi de 0,83. Verificou-se, no entanto, que a solução baseada no VAE, quando foi eficaz na deteção, ou seja, detetou todas as anomalias propostas com baixo índice de classificação fora das mesmas, conseguiu superar o desempenho da IF, nos contadores "Eletricidade 2" e "Gás". Não obstante, num dos contadores em avaliação, o VAE teve a pior pontuação dos resultados apresentados, tendo também verificado-se problemas de divergência, mencionados anteriormente, indicando que poderia vir a ser pouco robusto quando aplicado a uma solução como a nossa, que necessitava de ser ao máximo generalizável sobre um ambiente não controlado. Verificou-se também durante os testes que o treino dos modelos VAE são demorados e computacionalmente pesados. Esse custo computacional individual, associado a um

elevado número de hotéis, contadores e mais ou menos frequentes (re-)treinos dos algoritmos, poderia levantar problemas adicionais relativamente aos requisitos de hardware do sistema global. Outra razão que não abonou a favor do VAE foi a grande quantidade de histórico que deve existir para estes modelos serem funcionais e robustos, o que poderia ser impedimento para fazer deteção em contadores recentes.

Finalmente, futuramente existirão algumas medidas que permitirão aumentar a eficácia do sistema. Por exemplo, está previsto fazer o acompanhamento dos alarmes, corrigindo os falsos negativos e falsos positivos, para que as leituras associadas possam entrar ou não no treino do algoritmo, e com o passar do tempo, se restrinjam estas falhas do sistema. Efetuar treinos frequentes dos modelos com os dados mais recentes, é outra das formas, pois permite que o algoritmo esteja sempre atualizado, sendo apenas possível na simplicidade do treino da IF. Para finalizar, a última recomendação está relacionada com o ajuste do parâmetro `contamination` do modelo, que se reflete numa maior ou menor atividade na classificação de anomalias, e por esta razão, muito intuitivo de ser entendido e ajustado pela própria empresa em ambiente de produção.

Neste contexto, no próximo capítulo é apresentado o sistema de deteção de consumos anómalos desenvolvido, resultante dos estudos até aqui apresentados.

5

Plataforma para deteção de anomalias

Como referido no início da dissertação, o principal objetivo da mesma é o desenvolvimento de uma plataforma de alarmística avançada para deteção de consumos anómalos de hotéis. Tendo por base os estudos iniciais, nomeadamente da análise exploratória efetuada aos dados (Capítulo 3) e os testes efetuados aos algoritmos de AM considerados (Capítulo 4), a arquitetura proposta para o sistema desenvolvido foi a representada na Figura 5.1.

Neste contexto destacam-se os dois módulos lógicos principais, que serão detalhados nas secções seguintes, nomeadamente: o “Analisador de dados” e o “Detetor de anomalias”. Na arquitetura representada é possível verificar as interações existentes entre as partes constituintes do sistema p.e., tabelas das bases das dados ou ficheiros com os modelos treinados. É de salientar que o sistema foi desenvolvido na linguagem

de programação Python, com recurso a bibliotecas como “Pandas” (Wes McKinney, 2010) e “Sklearn” (Pedregosa et al., 2011), utilizando ainda os Sistemas de Gestão de Bases de Dados (SGBD) Microsoft SQL Server (Microsoft, 2019) e SQLite (Hipp, 2020). Assim, nas próximas secções são descritos com maior detalhe os módulos lógicos da aplicação e a restante estrutura, e apresentados alguns resultados da mesma aplicação em contexto real.

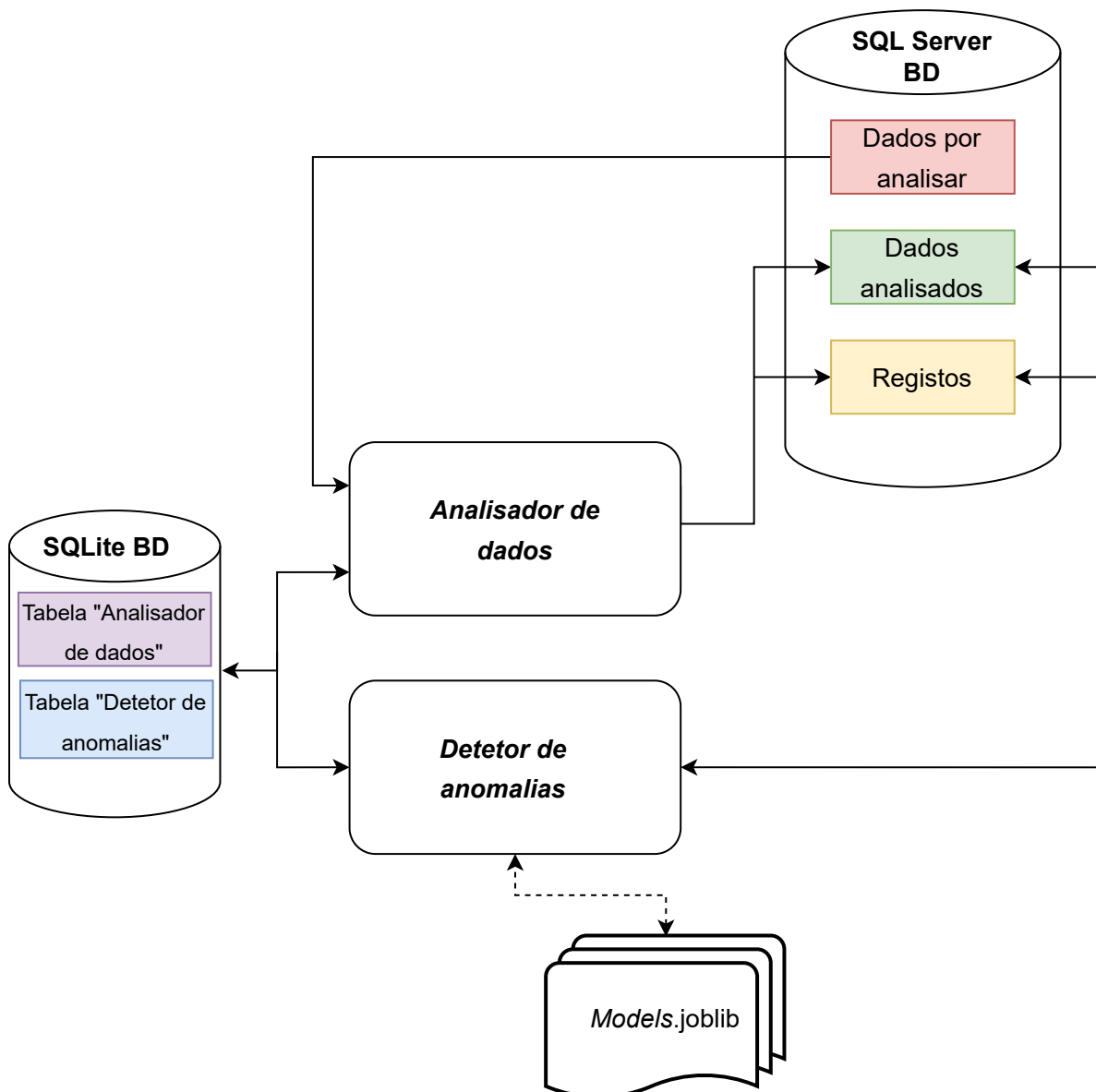


Figura 5.1: Arquitetura do sistema computacional desenvolvido para a deteção de anomalias em consumos hoteleiros.

5.1 Módulo de análise de dados

O módulo de análise de dados, representado na Figura 5.2, tem como principal objetivo garantir que as leituras que são guardadas na base de dados, que alimenta o treino e classificação de dados, não sofrem dos problemas identificados durante a fase exploratória (ver Secção 3.1). Desses problemas destacamos valores de leitura errados e/ou de leituras em falta, que são associadas de modo geral, a perdas de transmissão por parte das GTC dos hotéis. Assim, as leituras dos diversos contadores são inicialmente armazenadas (sem qualquer processamento) numa tabela designada “Dados por analisar”, sendo de seguida processados pelo módulo “Analisador de dados”. O módulo “Analisador de dados” corre periodicamente e, sempre que inicia um ciclo, verifica se há leituras ausentes no período em análise, para que no fim seja garantida a continuidade da série de leituras (*time_series_integrity*). Quando identificada uma situação deste tipo as leituras em falta são ponderadas utilizando a MMPL, conforme detalhado na Secção 3.2.2. Uma vez que se verificaram casos em que os valores das leituras estavam errados, procedeu-se também a uma validação dos mesmos, verificando se estes se encontram ou não dentro de um intervalo delimitado pelos valores mínimo e máximo de leituras expectáveis para aquele contador (*raw_data_analysis*). Estes limites (Lim_{inf} e Lim_{sup}) podem ser definidos pelo utilizador, caso sejam conhecidos, ou então calculados através do método do IIQ, já utilizado no tratamento dos dados e apresentado na Secção 2.2.1. Uma vez que o intuito não é a deteção de anomalias de consumos mas sim garantir a qualidade das leituras para que posteriormente possam ser analisadas, os limites calculados pretendem apenas delimitar uma fronteira entre leituras reais e leituras garantidamente problemáticas. A variável K presente nas equações para o cálculo dos limites (Equações 2.2 e 2.3), de acordo com os testes prévios, deve variar entre 2 e 5, permitindo regular o tamanho do intervalo de leituras consideradas aceitáveis.

Após o passo anterior, as leituras processadas são colocadas na tabela “Dados analisados”, sendo que, quando não são validadas nos passos anteriores, são estimadas novas leituras (usando de novo a MMPL). Estas correções aos dados são registadas na tabela “Registos”, permitindo ao utilizador reverter as mesmas se tal se justificar.

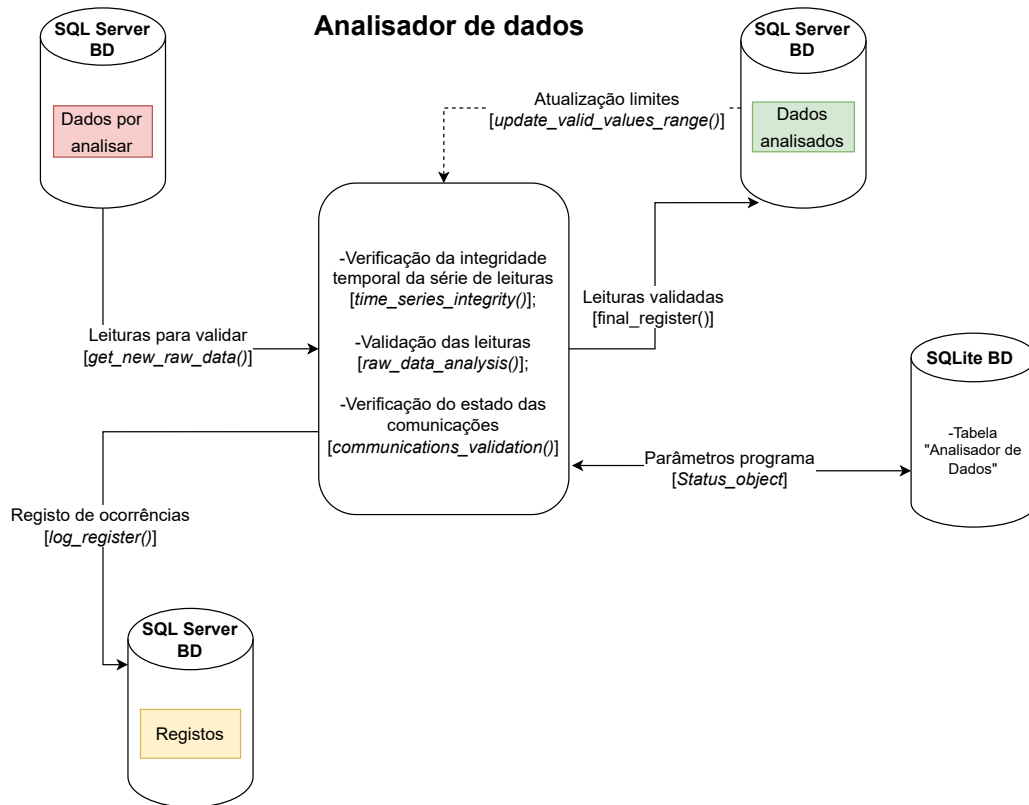


Figura 5.2: Arquitetura do módulo "Analisador de dados".

Existe também um método presente no módulo (*communications_validation*) que verifica, constantemente, se a última leitura validada ultrapassou o tempo esperado de chegada de novos dados. Se tal se verificar é identificada uma possível perda de transmissão de leituras que é colocada na tabela de registos. Por fim, também são guardadas algumas variáveis de estado, numa tabela no formato SQLite, que permitem ao módulo recuperar o ponto anterior de funcionamento (e.g., após reiniciar o módulo).

5.2 Módulo de deteção de anomalias

O módulo de deteção de anomalias é o núcleo do programa de alarmística. É através dele que são gerados os alarmes referentes a possíveis consumos anómalos com recurso ao modelo de AM (e.g., *Isolation Forest*).

Este módulo encontra-se dividido em 2 sub-módulos, conforme, apresentado a Figura 5.3: um responsável por toda a gestão de treinos de modelos, designado por "Modelo", e o outro designado por "Classificador", responsável pela análise dos con-

sumos. O sub-módulo “Modelo”, de forma cíclica, executa a rotina *train_model*, onde são verificadas as condições de (re)treino do modelo, i.e., o modelo é treinado quando no referido ciclo se verifica que este ainda não existe ou quando é ultrapassado determinado tempo desde a execução do último treino. Quando tal ocorre, é dado início ao processo de procura de dados na tabela “Dados validados”, que irão integrar o treino do novo modelo. Depois, a partir das leituras selecionadas é produzida uma *dataframe* de treino, onde são associadas as respetivas leituras, algumas variáveis temporais (e.g., horas, dia da semana) e outras desenvolvidas na fase de pesquisa e desenvolvimento, que fizeram parte do estudo de avaliação dos algoritmos (ver Secção 4.1). Depois de efetuado o treino do modelo é executado o método *save_model*, que o guarda num ficheiro em formato “joblib”, numa pasta designada para esse efeito, onde mais tarde o “Classificador” poderá carregá-lo em memória e utilizá-lo (através do método *load_model*). De salientar que o módulo foi projetado de forma, a ser possível introduzir variáveis externas no treino do modelo como a temperatura ou ocupação do hotel, que são variáveis que podem existir apenas em alguns hotéis.

O “Classificador” executa a rotina *run_anomaly_classifier*, que, se necessário, carrega o modelo em memória com o método já referido, para depois classificar as leituras e com isto detetar possíveis anomalias. Leituras que são classificadas como anomalias são registadas na tabela “Registos” e é atualizada a coluna *anomaly* na tabela “Dados validados” de forma que estas leituras não entrem nos próximos treinos. Assim, tal como no módulo “Analisador de dados” é feito um registo de recuperação numa tabela SQLite.

5.3 Base de dados

Nas arquiteturas apresentadas são visíveis 2 bases de dados: uma sobre um SGBD Microsoft SQL Server (base de dados da empresa onde estão guardadas as leituras) e outra sobre um SGBD SQLITE (base de dados criada para guardar as variáveis de estado do programa, que permitirão recomeçar o programa no estado anterior). Nas

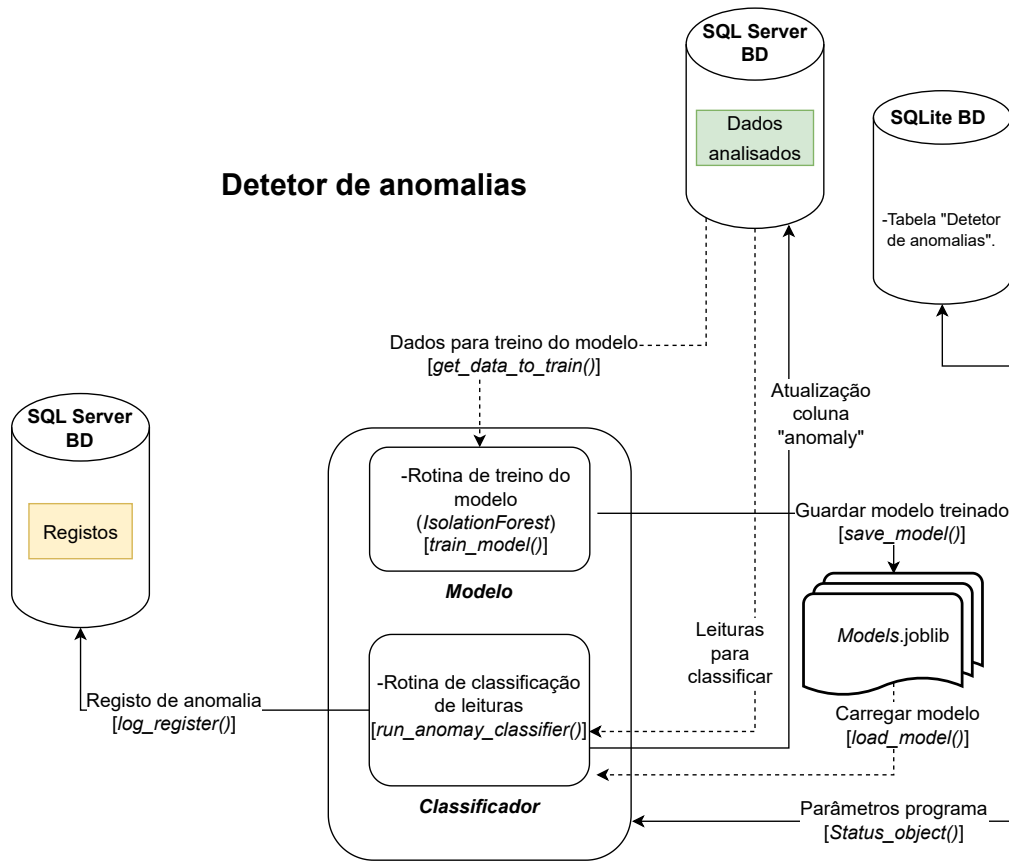


Figura 5.3: Arquitetura do módulo "Detetor de anomalias".

próximas secções são apresentados os modelos de dados associados às bases de dados mencionadas.

5.3.1 Base de dados de leituras

Dados por analisar

A tabela na Figura 5.4 apresenta a estrutura e um exemplo dos dados provenientes das GTC dos hotéis, onde se encontram as leituras por validar. A tabela tem 4 atributos, nomeadamente:

1. **ts** - data-hora da leitura;
2. **tagName** – nome do contador;
3. **tagValue** – leitura do contador;
4. **quality** – campo utilizado pela empresa, mas sem utilidade para o sistema.

<i>Dados por analisar</i>			
ts	tagName	tagValue	quality
2022-07-15 21:59:59.000	Contador 1	10,7	0
2022-07-15 21:59:59.000	Contador 2	4,8	0
2022-07-15 21:59:59.000	Contador 3	10	0
2022-07-15 22:59:59.000	Contador 1	7,2	0
2022-07-15 22:59:59.000	Contador 2	2,4	0
...

Figura 5.4: Configuração da tabela "Dados por analisar" na base de dados SQL Server.

Dados analisados

A tabela na Figura 5.5 apresenta a estrutura e um exemplo das leituras que foram validadas pelo "Analisador de dados". A tabela tem 5 atributos, nomeadamente:

- **ts** — data-hora da leitura;
- **tagName** — nome do contador;
- **tagValue** — leitura do contador;
- **quality** — identifica o tipo de leitura:
 0. leitura real;
 1. leitura ponderada.
- **anomaly** — identifica se a leitura foi considerada uma anomalia de consumo:
 0. leitura normal;
 1. leitura anómala.

Registos

A tabela "Registos" recebe os eventos registados por todo o programa tendo a estrutura apresentada na Figura 5.6. Esta tabela tem 6 atributos, nomeadamente:

<i>Dados analisados</i>				
ts	tagName	tagValue	quality	anomaly
2022-07-15 21:59:59.000	Contador 1	10,7	0	1
2022-07-15 21:59:59.000	Contador 2	4,8	0	0
2022-07-15 21:59:59.000	Contador 3	10	0	0
2022-07-15 22:59:59.000	Contador 1	7,2	0	0
2022-07-15 22:59:59.000	Contador 2	2,4	0	0
...

Figura 5.5: Configuração da tabela "Dados Analisados" na base de dados SQL Server.

<i>Registos</i>					
ts	tagName	msg	old read	new read	code
2022-07-15 21:59:59.000	Contador 1	Possivel anomalia: 31.2	0	1	4
2022-07-15 21:59:59.000	Contador 2	Perda de transmissão: passou 1 dia, desde a última leitura validada.	0	0	3
2022-07-15 21:59:59.000	Contador 3	Leitura em falta	0	0	2
2022-07-15 22:59:59.000	Contador 1	Leitura fora dos limites: 0.0 <= -2.9 <= 28.85	0	0	1
...

Figura 5.6: Configuração da tabela "Registos" na base de dados SQL Server.

- **ts** — data-hora do registo;
- **tagName** — nome do contador;
- **msg** — mensagem gerada pela aplicação que descreve o problema;
- **old read** — leitura substituída;
- **new read** — leitura que substitui a leitura antiga;
- **code** — código de registo de ocorrências:
 1. Leitura fora do intervalo;
 2. Leitura em falta;
 3. Possível perda de transmissão;
 4. Possível anomalia;
 5. Desempenho do modelo pode ser precário por leituras insuficientes.

5.3.2 Base de dados de configuração e estado do sistema

Tabela “Analisador de dados”

A tabela "Analisador de dados" guarda os parâmetros necessários para recuperação do ponto de funcionamento do módulo “Analisador de Dados” e apresenta a estrutura da Figura 5.7.

Tabela “Analisador de dados”				
meter name	tsQ	lower value	upper value	last ts validated
Contador 1	1657921197.83614	0	14,0	2022-07-15 22:59:59.000000
Contador 2	1657921198.4972	0	4,8	2022-07-10 21:59:59.000000
Contador 3	1657921195.31677	0	14,7	2022-07-15 22:59:59.000000
Contador 4	1657921197.34282	0	159,9	2022-07-15 22:59:59.000000
...

Figura 5.7: Configuração da tabela “Analisador de dados” na base de dados do tipo SQLite.

É composta por 5 atributos, nomeadamente:

- **meter name** — nome do contador;
- **tsQ** — *epoch time* referente à última atualização de limites;
- **lower value** — limite inferior;
- **upper value** — limite superior;
- **last ts validated** — timestamp correspondente à última leitura validada.

Tabela “Detetor de anomalias”

A tabela "Detetor de anomalias" apresenta a mesma funcionalidade que a tabela anterior, mas para o módulo da deteção de anomalias. A sua estrutura é a respresenta na Figura 5.8.

Esta tabela é composta por 8 atributos, que são:

- **meter name** — nome do contador;

Tabela "Detetor de anomalias"							
meter name	model file	ts train	load model	ts newest reading	ts last reading to train	last reading classified	features
Contador 1	./models/Contador1.joblib	1657925801.05858	0	2022-07-15 22:59:59.000000	2022-07-14 20:59:59.000000	2022-07-15 22:59:59.000000	['Contador 2','Temperatura']
Contador 2	./models/Contador2.joblib	1657877309.64152	1	2022-07-15 22:59:59.000000	2022-07-14 20:59:59.000000	2022-07-15 22:59:59.000000	[]
Contador 3	./models/Contador3.joblib	1657921195.31677	0	2022-07-15 22:59:59.000000	2022-07-14 20:59:59.000000	2022-07-15 22:59:59.000000	['Temperatura','Ocupação','Grau dia']
Contador 4	./models/Contador4.joblib	1657881564.21351	0	2022-07-15 22:59:59.000000	2022-07-14 20:59:59.000000	2022-07-15 22:59:59.000000	['Contador 1','Contador 2']
...	

Figura 5.8: Configuração da tabela "Detetor de anomalias" na base de dados do tipo SQLITE.

- **model file** — caminho do ficheiro com o modelo treinado;
- **ts train** — *epoch time* referente ao último treino efetuado;
- **load model** — flag que informa a existência de um novo modelo para ser carregado;
- **ts newest reading** — data-hora da leitura mais recente na base de dados SQL Server;
- **ts last reading to train** — data-hora da última leitura utilizada no treino do modelo;
- **ts last reading classified** — data-hora da última leitura analisada pelo detetor;
- **features** — variáveis indicadas pelo utilizador, utilizados como parte da análise do modelo.

Depois de apresentada toda a plataforma desenvolvida, desde a sua arquitetura às tabelas e respetivas bases de dados, seguem-se alguns dos resultados obtidos com a plataforma em condições reais de teste na secção seguinte.

5.4 Análise dos resultados obtidos na plataforma

Após a implementação do sistema de alarmística, este foi colocado a funcionar num ambiente controlado pela empresa, mas exposto a condições reais de funcionamento.

Passado um mês de execução, começaram-se a obter alguns resultados, os quais são objeto de análise nesta secção. Durante os testes em ambiente real, a plataforma monitorizou continuamente 2 hotéis (que designamos de Hotel I e Hotel II). Ambos os hotéis, são da região do Algarve e apresentam contadores de diferentes tipologias: água, gás e eletricidade.

No Hotel I foi perceptível que inicialmente o nível de contamination estaria demasiado elevado, tendo-se no dia 06-07-2022, procedido a uma redução deste parâmetro. Nas Figuras 5.9, 5.10 e 5.11, estão representados períodos de análise de 10 dias para 3 contadores distintos (a vermelho estão marcadas as leituras identificadas como possível anomalia e a linha verde, como o dia da mudança do parâmetro contamination). Nas figuras referentes aos resultados dos contadores "Eletricidade A" e "Eletricidade B" (Figuras 5.9 e 5.10), é possível verificar que antes da intervenção estaria a ocorrer uma elevada incidência de falsos alarmes, pois não parece haver alterações significativas nos padrões de consumo. Após os intervalos existe um período de pouca atividade da alarmística, até aparecerem alterações significativas que são detetadas pelo sistema. Na Figura 5.11, mesmo antes da mudança, existe uma situação anormal detetada até ser restabelecido o padrão de consumo e já com o consumo normalizado, é identificado um pico de consumo. Em qualquer dos casos verificou-se uma melhoria no desempenho do sistema com esta intervenção, ao diminuir a incidência de alarmes, mas continuando a reportar situações mais suspeitas. Posteriormente voltou-se a efetuar uma nova análise onde se obtiveram deteções de situações bastante interessantes. Na Figura 5.12 o sistema identificou uma situação onde existe um aumento de consumo que se estabelece no tempo, o que poderia ser sugestivo de uma anomalia ou ,p.e., da ligação de um novo equipamento, tendo detetado também dois picos de consumo. Relativamente à Figura 5.13, do contador "Eletricidade E", observa-se uma quebra de consumo que foi prontamente detetada assim como, uma variação abrupta do consumo.

Em relação ao Hotel II verificou-se um elevado índice de disparos do algoritmo na grande maioria dos contadores, tal como é verificável nas Figuras 5.14 e 5.15. Neste

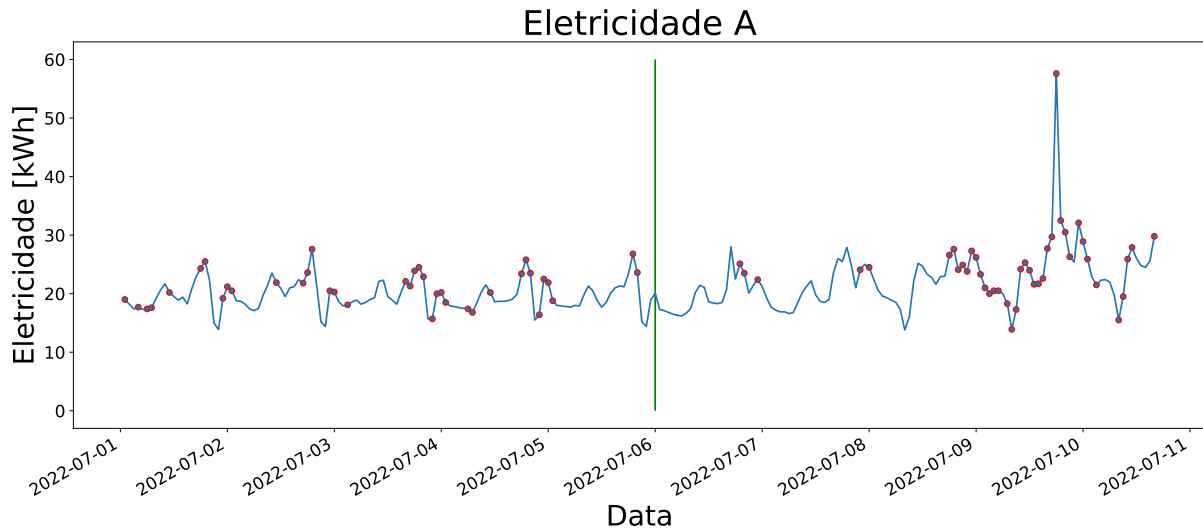


Figura 5.9: Resultados da alarmística num período de 10 dias, sobre o contador “Eletricidade A” do Hotel I.

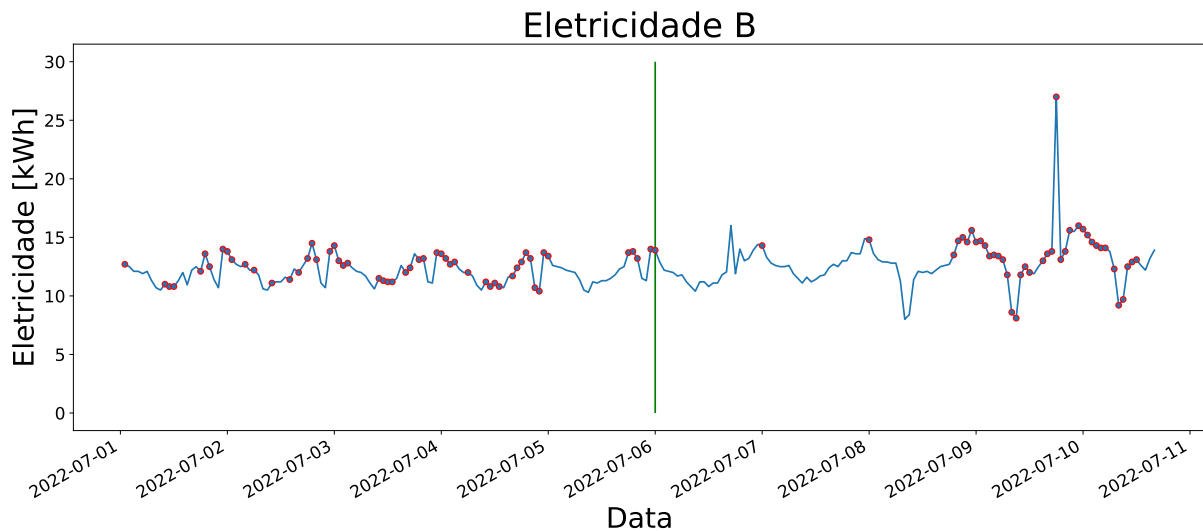


Figura 5.10: Resultados da alarmística num período de 10 dias, sobre o contador “Eletricidade B” do Hotel I.

caso nem mesmo a diminuição da contaminação foi possível compensar o sucedido, o que de alguma forma levantou preocupações. Para entender a razão deste problema, investigou-se os dados que estariam a ser utilizados para treinar os modelos, tendo-se constatado que muitos destes contadores teriam estado parados nos anos de pandemia e que por sua vez os modelos estariam a ser treinados apenas com leituras deste período.

Assim para ser possível fazer alguma monitorização nestes contadores, é necessário que o estado “anomaly” na tabela de armazenamento destas leituras seja reposto a 0

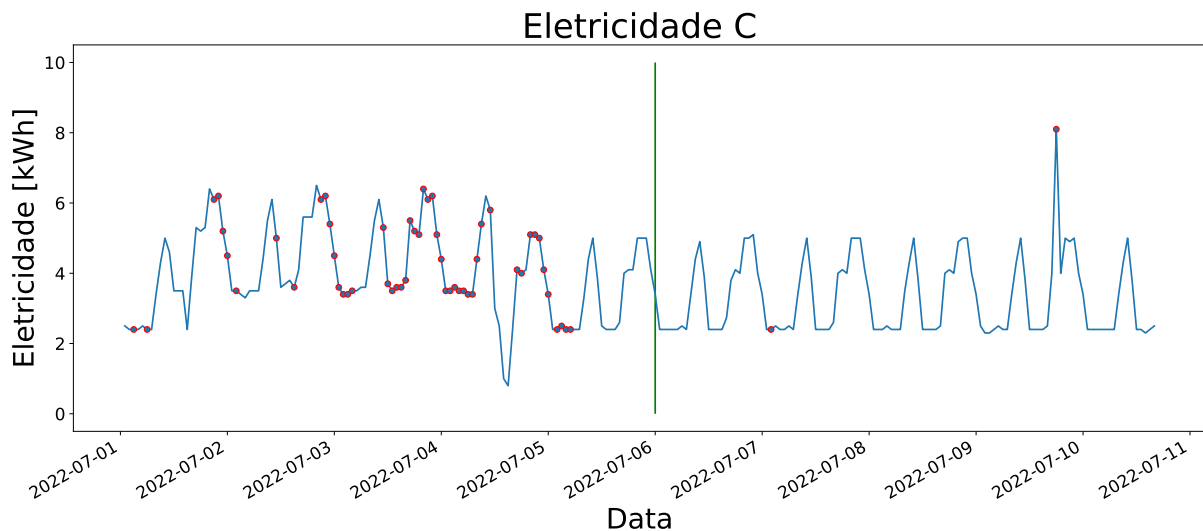


Figura 5.11: Resultados da alarmística num período de 10 dias, sobre o contador “Eletricidade C” do Hotel I.

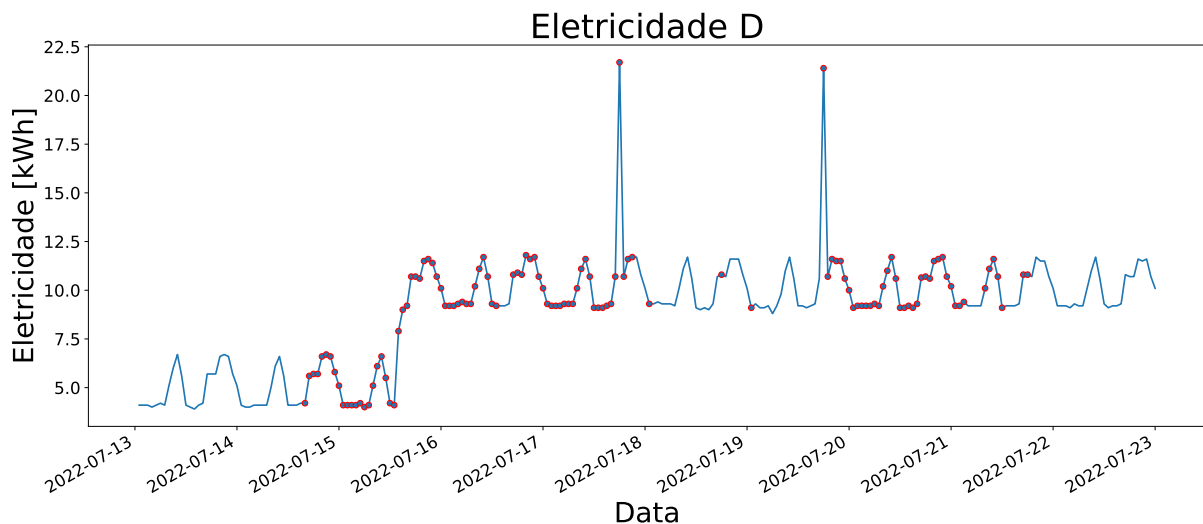


Figura 5.12: Resultados da alarmística num período de 10 dias, sobre o contador “Eletricidade D” do Hotel I.

ou até mesmo desligada a detecção de anomalias nestes contadores por algum tempo, para que as novas leituras possam começar a integrar os treinos do algoritmo, levando a que no curto, médio prazo, o modelo ganhe o entendimento da nova realidade. Com estes resultados, provou-se também, que as recomendações deixadas poderão fazer diferença no desempenho do programa e deverão ser tidas em consideração no futuro.

No próximo capítulo são feitas as conclusões desta dissertação e apresentadas algumas possibilidades para trabalhos futuros.

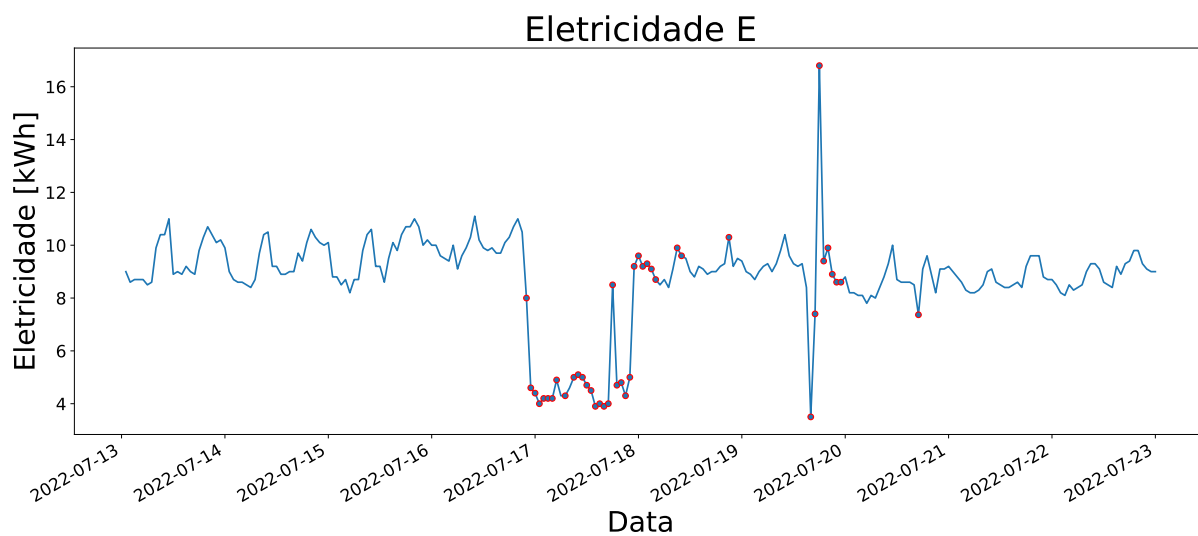


Figura 5.13: Resultados da alarmística num período de 10 dias, sobre o contador “Eletricidade E” do Hotel I.

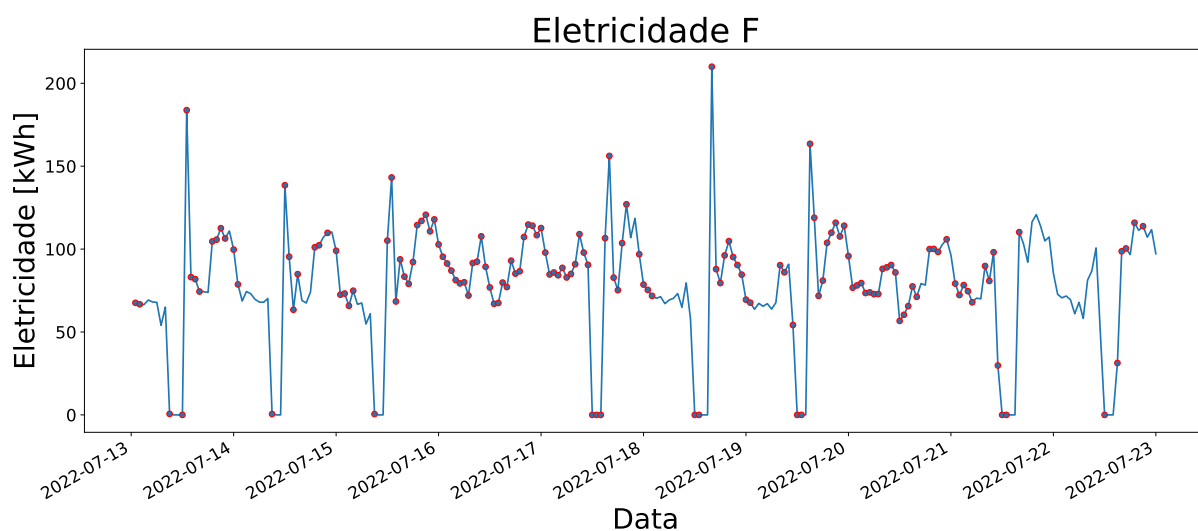


Figura 5.14: Resultados da alarmística num período de 10 dias, sobre o contador “Eletricidade F” do Hotel II.

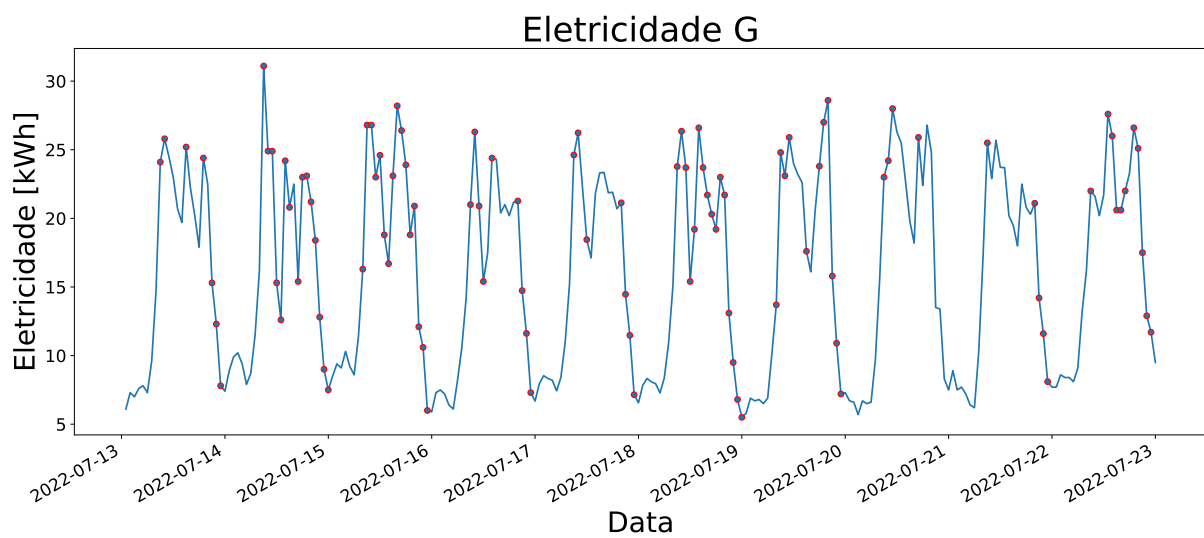


Figura 5.15: Resultados da alarmística num período de 10 dias, sobre o contador “Eletricidade G” do Hotel II.

6

Conclusões e trabalho futuro

Esta dissertação teve como principal objetivo desenvolver e implementar uma plataforma de detecção de anomalias para consumos hoteleiros. Com este sistema pretende-se evitar consumos que existam desnecessariamente ou até mesmo detetar avarias nas instalações.

Durante o estudo desenvolvido sobre os dados, verificaram-se algumas problemáticas nas leituras fornecidas, que poderiam interferir com o desempenho da plataforma, tais como: (i) os valores outliers e (ii) leituras em falta. A solução encontrada foi o desenvolvimento de um módulo designado de “Analisador de dados” para responder efetivamente aos problemas identificados, garantindo-se que o histórico de leituras apresentará futuramente uma melhor qualidade.

Para se avaliar os algoritmos IF e VAE, desenvolveu-se uma métrica de avaliação de

desempenho de algoritmos para séries temporais não supervisionadas. Comparando os resultados obtidos entre os 2 algoritmos, chegou-se à conclusão que o algoritmo *IsolationForest* daria uma melhor resposta ao problema, por ter obtido um maior equilíbrio nos resultados de teste por comparação com o VAE. Exemplo disso, a média da P_{final} do algoritmo IF foi de 0,89, enquanto que a do VAE foi de 0,83. O VAE, por outro lado, mostrou ser capaz de fornecer bons resultados de deteção, inclusive melhores resultados do que a IF em duas situações, o que pode indicar ser uma boa solução para um problema, onde a generalização e fácil adaptabilidade do algoritmo em situações não controladas, não seja um critério. Em conclusão, o algoritmo IF pertence aos algoritmos de AM do tipo não supervisionados, o que quer dizer, que faz uma análise de dados não classificados. Este algoritmo conseguiu responder às necessidades específicas deste problema, demonstrando na fase de testes, ser simples e adaptável a diferentes dados. Demonstrou ainda conseguir detetar as anomalias introduzidas, possuir um treino pouco exigente computacionalmente e ser muito adaptável às diferentes variáveis disponíveis para análise.

Futuramente, para melhoria do desempenho da plataforma, concluiu-se que será importante manter um acompanhamento dos alarmes por forma a seleccionar os dados que integrarão os treinos do algoritmo e ajustar o valor do parâmetro *contamination* sempre que se verificar necessidade de reduzir ou aumentar a incidência de alarmes.

Também para trabalho futuro, poderá ser interessante o desenvolvimento de uma ferramenta de realidade aumentada, suportada nos resultados deste sistema, que permitisse aos técnicos de um hotel, facilmente identificar os locais e aparelhos com anomalias detetadas. Uma funcionalidade também importante que poderia ser explorada passa pela capacidade de interpretar cada anomalia detetada, explicando a razão por detrás da mesma. Apesar do algoritmo proposto e testado ter mostrado ser uma boa solução, em termos futuros poderia ser interessante também testar novos algoritmos e até combiná-los num mecanismo de *Ensemble Learning* para produzir uma solução mais robusta de deteção. De referir que a plataforma foi desenhada com uma arquitetura modular, precisamente para permitir a adoção de outros métodos de forma global ou

especificamente para determinado ou determinados contadores.

Este trabalho foi realizado no âmbito do projeto 10PS00107 - PS UALG/Âmago, Unipessoal Lda, financiado pela União Europeia através do CRESC ALGARVE 2020, e do Portugal 2020.

Bibliografia

- Abdallah, A., Maarof, M. A., and Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68:90–113.
- Ahsanullah, M., Kibria, B., and Shakil, M. (2014). Normal distribution. In *Normal and Student t Distributions and Their Applications*, pages 7–50. Springer.
- Akçay, S., Atapour-Abarghouei, A., and Breckon, T. P. (2018). Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*, pages 622–637. Springer.
- Aminikhanghahi, S. and Cook, D. J. (2017). A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367.
- An, J. and Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18.
- Araya, D. B., Grolinger, K., Elyamany, H. F., Capretz, M. A., and Bitsuamlak, G. (2017). An ensemble learning framework for anomaly detection in building energy consumption. *Energy and Buildings*, 144:191–206.
- Associação Portuguesa de Energias Renováveis (2022). Apren - boletim dezembro 2021. <https://www.apren.pt/contents/publicationsreportcarditems/boletim-renovaveis-dezembro-2021.pdf>. Accessed: 2022-01-16.
- Baldi, P. (2012). Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 37–49. JMLR Workshop and Conference Proceedings.
- Bank, D., Koenigstein, N., and Giryas, R. (2020). Autoencoders. *arXiv preprint arXiv:2003.05991*.
- Bao, Y., Tang, Z., Li, H., and Zhang, Y. (2019). Computer vision and deep learning-based data anomaly detection method for structural health monitoring. *Structural Health Monitoring*, 18(2):401–421.
- Brownlee, J. (2020). *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python*. Machine Learning Mastery.
- Buchanan, B. (2019). Artificial intelligence in finance.
- Bullock, J., Luccioni, A., Pham, K. H., Lam, C. S. N., and Luengo-Oroz, M. (2020). Mapping the landscape of artificial intelligence applications against covid-19. *Journal of Artificial Intelligence Research*, 69:807–845.

-
- Butakov, N. (2020). How to build robust anomaly detectors with machine learning. <https://www.ericsson.com/en/blog/2020/4/anomaly-detection-with-machine-learning>. Accessed: 2022-03-23.
- Cao, J., Li, Z., and Li, J. (2019). Financial time series forecasting model based on ceemdan and lstm. *Physica A: Statistical mechanics and its applications*, 519:127–139.
- Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., and Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408:189–215.
- Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58.
- Chollet, F. et al. (2015). Keras. <https://github.com/fchollet/keras>.
- Cloud Education IBM (2020). Exploratory data analysis. <https://www.ibm.com/cloud/learn/exploratory-data-analysis>. Acedido: 2022-03-15.
- Cook, A. A., Mısırlı, G., and Fan, Z. (2019). Anomaly detection for iot time-series data: A survey. *IEEE Internet of Things Journal*, 7(7):6481–6494.
- Da Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H. B., and dos Reis Alves, S. F. (2017). Artificial neural networks. *Cham: Springer International Publishing*, 39.
- de Santis, R. B. and Costa, M. A. (2020). Extended isolation forests for fault detection in small hydroelectric plants. *Sustainability*, 12(16):6421.
- Deecke, L., Vandermeulen, R., Ruff, L., Mandt, S., and Kloft, M. (2018). Image anomaly detection with generative adversarial networks. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 3–17. Springer.
- Demirhan, H. and Renwick, Z. (2018). Missing value imputation for short to mid-term horizontal solar irradiance data. *Applied Energy*, 225:998–1012.
- Deng, S.-M. and Burnett, J. (2000). A study of energy performance of hotel buildings in hong kong. *Energy and Buildings*, 31:7–12.
- Ding, Z. and Fei, M. (2013). An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes*, 46:12–17.
- Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.
- Drew, P. J. and Monson, J. R. (2000). Artificial neural networks. *Surgery*, 127(1):3–11.
- Dunis, C., Middleton, P., Karathanasopolous, A., and Theofilatos, K. (2016). *Artificial intelligence in financial markets*. Springer.
- Dunning, T. and Friedman, E. (2014). *Practical machine learning: a new look at anomaly detection*. "O'Reilly Media, Inc."

-
- Eras, J. J. C., Santos, V. S., Gutiérrez, A. S., Álvarez Guerra Plasencia, M., Haeseldonckx, D., and Vandecasteele, C. (2016). Tools to improve forecasting and control of the electricity consumption in hotels. *Journal of Cleaner Production*, 137:803–812.
- Farzaneh, H., Malehmirchegini, L., Bejan, A., Afolabi, T., Mulumba, A., and Daka, P. P. (2021). Artificial intelligence evolution in smart buildings for energy efficiency. *Applied Sciences*, 11(2):763.
- Fenza, G., Gallo, M., and Loia, V. (2019). Drift-aware methodology for anomaly detection in smart grid. *IEEE Access*, 7:9645–9657.
- Giannoni, F., Mancini, M., and Marinelli, F. (2018). Anomaly detection models for iot time series data. *arXiv preprint arXiv:1812.00890*.
- Gondara, L. (2016). Medical image denoising using convolutional denoising autoencoders. In *2016 IEEE 16th international conference on data mining workshops (ICDMW)*, pages 241–246. IEEE.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.
- Gul, M. S. and Patidar, S. (2015). Understanding the energy consumption and occupancy of a multi-purpose academic building. *Energy and Buildings*, 87:155–165.
- Guo, D., Zhou, W., Li, H., and Wang, M. (2018). Hierarchical lstm for sign language translation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Hamamoto, A. H., Carvalho, L. F., Sampaio, L. D. H., Abrão, T., and Proença Jr, M. L. (2018). Network anomaly detection system using genetic algorithm and fuzzy logic. *Expert Systems with Applications*, 92:390–402.
- Haque, S. A., Rahman, M., and Aziz, S. M. (2015). Sensor anomaly detection in wireless sensor networks for healthcare. *Sensors*, 15(4):8764–8786.
- Hassani, H., Kalantari, M., and Ghodsi, Z. (2019). Evaluating the performance of multiple imputation methods for handling missing values in time series data: A study focused on east africa, soil-carbonate-stable isotope data. *Stats*, 2:457–467.
- Himeur, Y., Ghanem, K., Alsalemi, A., Bensaali, F., and Amira, A. (2021). Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. *Applied Energy*, 287:116601.
- Hipp, R. D. (2020). SQLite.
- Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Hogg, R. V., Tanis, E. A., and Zimmerman, D. L. (1977). *Probability and statistical inference*, volume 993. Macmillan New York.

-
- Holmberg, K. and Erdemir, A. (2019). The impact of tribology on energy use and co2 emission globally and in combustion engine and electric cars. *Tribology International*, 135:389–396.
- Hota, H., Handa, R., and Shrivasa, A. (2017). Time series data prediction using sliding window based rbf neural network. *International Journal of Computational Intelligence Research*, 13(5):1145–1156.
- Ilyas, I. F. and Chu, X. (2019). *Data cleaning*. Morgan & Claypool.
- International Energy Agency (2021). Review 2021 - assessing the effects of economic recoveries on global energy demand and co2 emissions in 2021 global energy. <https://www.iea.org/reports/global-energy-review-2021>. Accessed: 2021-12-15.
- International Energy Agency (2022). Electric cars fend off supply challenges to more than double global sales. <https://www.iea.org/commentaries/electric-cars-fend-off-supply-challenges-to-more-than-double-global-sales>. Accessed: 2022-03-23.
- Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963.
- Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3):31–44.
- Jung, C., Lee, Y., Lee, J., and Kim, S. (2020). Performance evaluation of the multiple quantile regression model for estimating spatial soil moisture after filtering soil moisture outliers. *Remote Sensing*, 12.
- Kingma, D. P., Welling, M., et al. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.
- Klößner, C. A., Nayum, A., and Mehmetoglu, M. (2013). Positive and negative spillover effects from electric car purchase to car use. *Transportation Research Part D: Transport and Environment*, 21:32–38.
- Kwak, S. K. and Kim, J. H. (2017). Statistical data preparation: management of missing values and outliers. *Korean Journal of Anesthesiology*, 70:407.
- Lavin, A. and Ahmad, S. (2015). Evaluating real-time anomaly detection algorithms—the numanta anomaly benchmark. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pages 38–44. IEEE.
- Li, D., Ho, S.-C. M., Song, G., Ren, L., and Li, H. (2015). A review of damage detection methods for wind turbine blades. *Smart Materials and Structures*, 24(3):033001.
- Lindemann, B., Maschler, B., Sahlab, N., and Weyrich, M. (2021). A survey on anomaly detection for technical systems using lstm networks.
- Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

-
- Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth IEEE international conference on data mining*, pages 413–422. IEEE.
- Liu, G. and Guo, J. (2019). Bidirectional lstm with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337:325–338.
- Livieris, I. E., Pintelas, E., and Pintelas, P. (2020). A cnn-lstm model for gold price time-series forecasting. *Neural computing and applications*, 32(23):17351–17360.
- Luo, H. and Zhong, S. (2017). Gas turbine engine gas path anomaly detection using deep learning with gaussian distribution.
- Majumdar, A. (2018). Blind denoising autoencoder. *IEEE transactions on neural networks and learning systems*, 30(1):312–317.
- Malhotra, P., Vig, L., Shroff, G., Agarwal, P., et al. (2015). Long short term memory networks for anomaly detection in time series. In *Proceedings*, volume 89, pages 89–94.
- Meng, Z., Hu, Y., and Ancey, C. (2020). Using a data driven approach to predict waves generated by gravity driven mass flows. *Water*, 12:600.
- Microsoft (2019). Sql server 2019. <https://www.microsoft.com/en-us/sql-server/sql-server-2019>. Acedido: 2022-09-25.
- Mohammed, M., Khan, M. B., and Bashier, E. B. M. (2016). *Machine learning: algorithms and applications*. Crc Press.
- Moustafa, N., Creech, G., Sitnikova, E., and Keshk, M. (2017). Collaborative anomaly detection framework for handling big data of cloud computing. In *2017 military communications and information systems conference (MilCIS)*, pages 1–6. IEEE.
- Munir, M., Siddiqui, S. A., Dengel, A., and Ahmed, S. (2018). Deepant: A deep learning approach for unsupervised anomaly detection in time series. *Ieee Access*, 7:1991–2005.
- Ng, A. et al. (2011). Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19.
- Nichols, J. A., Herbert Chan, H. W., and Baker, M. A. (2019). Machine learning: applications of artificial intelligence to imaging and diagnosis. *Biophysical reviews*, 11(1):111–118.
- Niu, Z., Yu, K., and Wu, X. (2020). Lstm-based vae-gan for time-series anomaly detection. *Sensors*, 20(13):3738.
- Omar, S., Ngadi, A., and Jebur, H. H. (2013). Machine learning techniques for anomaly detection: An overview. *International Journal of Computer Applications*, 79:33–41.
- Patil, N., Das, D., and Pecht, M. (2015). Anomaly detection for igbts using mahalanobis distance. *Microelectronics Reliability*, 55(7):1054–1059.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

-
- Pereira, J. and Silveira, M. (2018). Unsupervised anomaly detection in energy time series data using variational recurrent autoencoders with attention. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 1275–1282. IEEE.
- Puggini, L. and McLoone, S. (2018). An enhanced variable selection and isolation forest based methodology for anomaly detection with oes data. *Engineering Applications of Artificial Intelligence*, 67:126–135.
- Qing-dao-er ji, R., Su, Y. L., and Liu, W. W. (2020). Research on the lstm mongolian and chinese machine translation based on morpheme encoding. *Neural Computing and Applications*, 32(1):41–49.
- Qiu, J., Wang, B., and Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PloS one*, 15(1):e0227222.
- Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.
- Rocha, M., Cortez, P., and Neves, J. (2007). Evolution of neural networks for classification and regression. *Neurocomputing*, 70(16-18):2809–2816.
- Rousseeuw, P. J. and Hubert, M. (2018). Anomaly detection by robust statistics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(2):e1236.
- Rudin, C. and Wagstaff, K. L. (2014). Machine learning for science and society.
- Saradjian, M. and Akhoondzadeh, M. (2011). Thermal anomalies detection before strong earthquakes ($m > 6.0$) using interquartile, wavelet and kalman filter methods. *Natural Hazards and Earth System Sciences*, 11(4):1099–1108.
- Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.
- Sharma, S., Sharma, S., and Athaiya, A. (2017). Activation functions in neural networks. *towards data science*, 6(12):310–316.
- Shen, A., Tong, R., and Deng, Y. (2007). Application of classification models on credit card fraud detection. In *2007 International conference on service systems and service management*, pages 1–4. IEEE.
- Sinaga, K. P. and Yang, M.-S. (2020). Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727.
- Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85.
- Soydaner, D. (2020). A comparison of optimization algorithms for deep learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(13):2052013.
- Tang, Z., Chen, Z., Bao, Y., and Li, H. (2019). Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. *Structural Control and Health Monitoring*, 26(1):e2296.

-
- Tealab, A. (2018). Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, 3(2):334–340.
- Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley Pub. Co.
- United Nations (2021). Theme report on energy transition towards the achievement of sdg 7 and net-zero emissions. https://www.un.org/sites/un2.un.org/files/2021-twg_2-062321.pdf. Accessed: 2022-03-23.
- Velleman, P. F. (2004). Applications, basics, and computing of exploratory data analysis the internet-first university press.
- Walfish, S. (2006). A review of statistical outlier methods. *Pharmaceutical technology*, 30(11):82.
- Weisstein, E. W. (2002). Normal distribution. <https://mathworld.wolfram.com/>.
- Wes McKinney (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.
- Woo, S. (2020). Modern definitions in reliability engineering. In *Reliability Design of Mechanical Systems*, pages 53–99. Springer.
- Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193.
- Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., et al. (2018). Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 world wide web conference*, pages 187–196.
- Xu, S., Lu, B., Baldea, M., Edgar, T. F., Wojsznis, W., Blevins, T., and Nixon, M. (2015). Data cleaning in the process industries. *Reviews in Chemical Engineering*, 31(5):453–490.
- Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4):611–629.
- Yan, W. (2020). Detecting gas turbine combustor anomalies using semi-supervised anomaly detection with deep representation learning. *Cognitive Computation*, 12(2):398–411.
- Yao, R., Liu, C., Zhang, L., and Peng, P. (2019). Unsupervised anomaly detection using variational auto-encoder based feature extraction. In *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–7. IEEE.
- Yegnanarayana, B. (2009). *Artificial neural networks*. PHI Learning Pvt. Ltd.
- Yu, K.-H., Beam, A. L., and Kohane, I. S. (2018). Artificial intelligence in healthcare. *Nature biomedical engineering*, 2(10):719–731.

-
- Yu, Y., Si, X., Hu, C., and Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270.
- Zhao, S., Blaabjerg, F., and Wang, H. (2020). An overview of artificial intelligence applications for power electronics. *IEEE Transactions on Power Electronics*, 36(4):4633–4658.
- Zheng, L., Liu, G., Yan, C., Jiang, C., Zhou, M., and Li, M. (2020). Improved tradaboost and its application to transaction fraud detection. *IEEE Transactions on Computational Social Systems*, 7(5):1304–1316.
- Zhou, P., Zhou, G., Wang, H., Wang, D., and He, Z. (2020). Automatic detection of industrial wire rope surface damage using deep learning-based visual perception technology. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11.
- Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. (2018). Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*.