Universidade de Aveiro
2021

Nelson Kévin das Neves Bastos

Análise de Modelos Machine Learning para previsão e otimização do comportamento de redes de abastecimento de água

Universidade de Aveiro
2021

Nelson Kévin das Neves Bastos

Análise de Modelos Machine Learning para previsão e otimização do comportamento de redes de abastecimento de água

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob a orientação científica do Doutor António Gil d'Orey de Andrade Campos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro

Este trabalho teve o apoio do programa COMPETE 2020, Aviso 17/SI/2019, e do programa operacional regional do centro (CENTRO2020) através do projeto I-RETIS-WATER (CENTRO-01-0247-FEDER-069857).

Dedico este trabalho à minha família.

o júri

presidente

Professor Doutor Joaquim Alexandre Mendes de Pinho da Cruz
Professor Auxiliar da Universidade de Aveiro

vogais

Professor Doutor Sérgio Guilherme Aleixo de Matos
Professor Auxiliar em Regime Laboral da Universidade de Aveiro

Professor Doutor António Gil D'Orey de Andrade Campos
Professor Auxiliar com Agregação da Universidade de Aveiro

agradecimentos

Um sincero agradecimento ao meu orientador Gil Campos por toda a dedicação, paciência e orientação durante todo o processo. Um agradecimento também aos membros da SCUBIC por todos os conselhos e dados fornecidos.

palavras-chave

Sistemas de abastecimento de água, machine learning, modelação, otimização, redução custos operacionais

resumo

Fornecer água em quantidade, qualidade e pressão adequada a toda a população tem sido um dos grandes desafios da história da humanidade. Por serem sistemas vitais ao funcionamento da sociedade, os sistemas de abastecimento de água (SAA) focaram-se na eficácia do transporte de água para todas as casas e indústria, não dando a devida importância aos custos energéticos resultantes desse processo. Porém, com o contínuo crescimento populacional registado nos últimos anos e consequente aumento dos consumos, tornou-se essencial otimizar o sistema. Atualmente, as bombas são colocadas em funcionamento quando os depósitos responsáveis pelo abastecimento atingem um valor mínimo, sendo desligadas posteriormente quando estes atingem um determinado valor máximo. Este funcionamento é pouco eficiente, pois não tem em conta as variações do custo das tarifas de energia ao longo do dia.

Ao longo das últimas décadas várias técnicas de simulação e simuladores têm sido desenvolvidos de modo a otimizar os SAA. De todos, o simulador mais comum é o software de simulação hidráulica EPANET. Apesar de ser amplamente utilizado e de obter excelentes resultados, o EPANET apresenta um processo de calibração extremamente complexo e pouco funcional.

Nesta dissertação é apresentada uma solução para tornar os SAA mais eficientes, eliminando todo o processo de calibração associado ao EPANET. São utilizados métodos de aprendizagem automática *(Machine Learning) n*a simulação e otimização de SAA de modo a garantir um padrão de funcionamento das bombas ideal, resultando no menor custo do consumo de energia possível. Para isso, são utilizados dois algoritmos com arquiteturas diferentes: ANN e XGBoost. São também testados dois modelos diferentes: um modelo diferencial e um modelo com os valores obtidos no final de cada variação temporal. Os resultados indicam que todos os modelos utilizados são capazes de prever com precisão o comportamento do sistema, principalmente o XGBoost diferencial que apresentou constantemente os melhores resultados. Assim, os modelos *ML* utilizado*s* apresentam-se como uma excelente alternativa na modelação e otimização de sistemas de abastecimento de água reais.

.

keywords

Water distribution systems, machine learning, modelling, optimization, reduce operational costs.

abstract

Providing water in adequate quantity, quality, and pressure to the entire population has been one of the challenges of human history. Being critical systems to the society, water supply systems (WSS) have focused on the effectiveness of transporting water to all homes and industries, not giving due importance to energy costs resulting from this process. However, with the continuous population growth in recent years and the consequent increase in water demand, it has become essential to optimize the system [1]. Currently, in the majority of the WSS, pumps start when the tanks reach the minimum, and switch off when the maximum value is reached. This operation is inefficient, as it does not consider the energy cost tariffs variations throughout the day nor any water demand forecasting. However, to predict the most efficient operation of WSS and considering that these are critical systems, it is necessary to model and calibrate these systems. However, even using well-recognized hydraulic simulators, the calibration of these models can be cumbersome. Much of this difficulty is due to pre-established formulations, which do not present enough flexibility to real data.

Over the last few decades several simulation techniques and simulators have been developed to optimize WSS. Of all, the most common hydraulic simulation software is EPANET. Despite being widely used and obtaining excellent results, EPANET presents an extremely complex calibration process.

In this work, a possible solution to the problem is presented using machine learning methods for simulating water supply systems and subsequently optimizing the system to ensure an optimal operation of the pumps, resulting in the lowest possible energy consumption and subsequent cost reduction. For this, two algorithms with different architectures are used: artificial neural networks and a decision tree-based algorithm (XGBOOST). Two different models are also tested: a differential and total time-updated model.

Índice

Lista de Figuras

Lista de Tabelas

1. Introdução

1.1 Enquadramento geral

A água e a energia representam recursos fundamentais para o bem-estar e desenvolvimento socioeconómico da humanidade. Nas próximas décadas é esperado um contínuo crescimento da procura por água doce e por energia, devido ao aumento da população, desenvolvimento das economias e às variações no estilo de vida e padrões de consumo [1].

Os sistemas de abastecimento de água (SAA) são responsáveis pelo tratamento, armazenamento, transporte e distribuição de água para casas, indústrias e necessidades públicas (combate a incêndios) [2]. A água é inicialmente bombeada numa fonte superficial (barragens, lagos) ou subterrânea (poços), sendo de seguida conduzida por uma rede de tubagens até reservatórios ou redes de distribuição. De modo a satisfazerem as necessidades dos consumidores em quantidade e qualidade, os SAA consomem elevadas quantidades de energia elétrica, representando um gasto mundial anual de aproximadamente €12 mil milhões (cerca de 35% de todas as despesas com a produção de água) [3].

Os reservatórios são responsáveis pelo armazenamento de água em períodos de menor procura, garantindo que esta se encontra disponível em períodos de grande procura ou em casos de emergência (fogos). Quando o reservatório atinge um determinado mínimo, as válvulas e bombas são ativadas, bombeando água até ser atingido um valor máximo predeterminado, sem ter em conta as diferentes variações diárias do preço da energia, tornando o sistema economicamente ineficiente [4].

Um dos maiores obstáculos para uma maior eficiência em redes de distribuição de água está diretamente relacionada com a complexidade dos sistemas, quer em termos de configuração das redes quer no número de variáveis a controlar e aos baixos níveis de flexibilidade de operação deste tipo de sistemas [4]. Sendo os SAA sistemas críticos, não é possível testar os sistemas para que estes se tornem mais eficientes, uma vez que qualquer falha poderia levar a prejuízos imensuráveis. De modo a solucionar este problema, várias técnicas de simulação e simuladores têm sido desenvolvidos. De todos, o simulador mais comum é o software de simulação hidráulica EPANET [5]. Apesar de ser amplamente utilizado na indústria e pela comunidade académica, este apresenta problemas relacionados com a calibração do software [6]. Adicionalmente, o processo de calibração necessita de um software adicional, de modo a ajustar diversas características (perdas de água, rugosidade dos canos), resultando num sistema extremamente complexo e pouco funcional.

Com o desenvolvimento de algoritmos cada vez mais eficazes, flexíveis e robustos as técnicas de inteligência artificial (mais concretamente *Machine Learning*[1]) têm vindo a ganhar cada vez maior popularidade na resolução de problemas relacionados com SAA, demonstrando serem capazes de reproduzir sistemas complexos, eliminando a necessidade da utilização de sistemas como o EPANET.

[1] Em português denomina-se aprendizagem automática, porém, por ser um termo universalmente reconhecido, foi utilizado o termo em inglês durante todo o documento.

1.2 Objetivos do trabalho

O principal objetivo deste trabalho será a implementação de técnicas *Machine Learning* (ML) em redes de abastecimento de água de forma a modelar o seu comportamento e aumentar a sua eficiência energético-financeira.

Para isso, é necessário dividir o trabalho em duas etapas. A primeira etapa passa pela modelação do sistema através de técnicas de ML de modo a prever o comportamento do sistema e a energia consumida nessa operação. O sistema recebe o estado inicial da rede (níveis de depósitos e estado das bombas e/ou válvulas) e os consumos, prevendo o nível final dos depósitos e a energia consumida pelas bombas para um determinado espaço temporal. É analisada a robustez e a eficiência computacional das técnicas e dos diferentes modelos utilizados.

O segundo objetivo passa pela otimização da operação do sistema. Este objetivo define-se como a resolução do problema do horário das bombas (*pump scheduling problem)*. Através dos modelos treinados anteriormente, é utilizado um algoritmo de otimização de modo a garantir a melhor configuração das bombas, minimizando os custos associados da operação, tendo em conta as variações dos tarifários ao longo do dia. Os resultados são validados utilizando duas redes com valores obtidos sinteticamente e testados numa terceira rede com dados reais.

1.3 Guia de leitura

Este trabalho encontra-se estruturado em 5 capítulos:

- Capítulo 1: Introdução ao tema, abordando a sua importância, os problemas a resolver, bem como uma possível solução, objetivo do trabalho e estrutura da dissertação;

- Capítulo 2: Revisão da literatura. Inclui as diferentes técnicas observadas para a simulação e otimização de sistemas de abastecimento de água. Comparação entre métodos tradicionais (EPANET) e *machine learning*;

- O Capítulo 3 será dividido em 2 secções. Metodologia utilizada para a simulação e otimização do sistema. Descrição das técnicas utilizadas, descrição e justificação dos *inputs* (*ou features)* fornecidos ao sistema, procedimento adotado para gerar os dados de treino, teste e validação do sistema, e técnicas de avaliação da performance dos modelos;

- Capítulo 4 – Implementação das técnicas e interpretação dos resultados obtidos em cada um dos casos de estudo observados;

- Capítulo 5 – Apresentação de conclusões que sintetizam todo o trabalho desenvolvido e sugestões para trabalhos futuros.

2. Estado de arte

2.1 Modelação de sistemas de abastecimento de água

Simuladores hidráulicos são programas numéricos onde é possível implementar modelos de transporte e distribuição de água. Estes modelos tentam replicar redes complexas resolvendo um conjunto de equações hidráulicas, incluindo a conservação da energia e de massa. Vários modelos diferentes têm sido desenvolvidos ao logo do tempo, como por exemplo modelos de balanço de massa, modelos de regressão, modelos de redes hidráulicas simplificados e modelos de simulação de redes hidráulicas [7].

Os modelos de balanço de massa são os modelos mais simples e consistem em relações funcionais ponderadas entre os níveis dos tanques de armazenamento, descargas da bomba e fluxos [6]. Os pesos associados às relações funcionais podem ser determinados através de regressões lineares [8].

Os modelos de regressão também são modelos empíricos, porém são mais precisos que os modelos de balanço de massa. Estes modelos são baseados num conjunto de equações não-lineares. Os modelos de regressão têm como principal vantagem poderem incorporar algum grau de não-linearidade, fornecendo um mecanismo de tempo eficiente para estimar a resposta da rede. Apesar desta vantagem, são modelos sensíveis a variações de dados, podendo levar a resultados pouco precisos [6].

Os modelos de redes hidráulicas simplificados podem ser considerados um passo intermédio entre os modelos empíricos e os modelos de simulação de redes hidráulicas. Em casos particulares, a aplicação de um conjunto de equações lineares é suficiente para representar sistemas hidráulicos [9-10]. Porém, em modelos complexos e reais a sua utilização não é aconselhada [6].

Ao contrário dos modelos empíricos e dos modelos de redes hidráulicas simplificados, os modelos de simulação de redes hidráulicas são capazes de modelar as dinâmicas não-lineares de uma rede de distribuição de água resolvendo equações hidráulicas, incluindo equações da conservação de massa e da conservação de energia. São modelos que se adaptam a mudanças físicas e variações espaciais, sendo por isso extremamente robustos. Contudo, necessitam de um esforço muito maior para serem calibrados corretamente [6].

Nos últimos 30 anos [11], tem ocorrido um avultado investimento no desenvolvimento de softwares de simulação hidráulica, nomeadamente o WATNET [12] e EPANET 2.0, sendo este último o mais utilizado tanto pela indústria como pela comunidade académica [13]. Apesar da sua vasta utilização e de ser capaz de apresentar resultados precisos, a sua utilização em tempo real pode ser impraticável em extensas redes de distribuição de água por causa do esforço computacional exigido na sua otimização. Adicionalmente, o processo de calibração requer que um operador, ou um software adicional, ajuste diversas características dos elementos da rede de distribuição, como a rugosidade das tubagens e perdas de água. Caso fosse necessário correr a simulação para cada mudança das bombas ou configuração de válvulas, o mais provável seria a configuração mais eficiente não ser encontrada antes de serem necessárias efetuar mais alterações.

2.2 *Machine Learning* em Simulação Hidráulica

Machine learning é o campo de estudo que, por meio de algoritmos, dá aos computadores a habilidade de aprenderem sem serem explicitamente programados [14]. Os algoritmos de *machine learning* podem ser divididos em três diferentes categorias: aprendizagem supervisionada, não-supervisionada e por reforço [15-16]. Na aprendizagem supervisionada são fornecidos ao sistema dados de entrada e de saída. O algoritmo é treinado sobre um conjunto de dados pré-definidos tentando identificar características dos dados de entrada para obter os dados de saída fornecidos, sendo depois testados em novos dados de entrada não treinados anteriormente. São divididos em problemas de classificação e de regressão. Na aprendizagem não-supervisionada apenas são fornecidos dados de entrada, tendo o algoritmo de identificar padrões nos dados fornecidos. É uma técnica usada em deteção de anomalias ou problemas onde é necessário aglomerar dados em diferentes categorias. Por fim, na aprendizagem por reforço (*reinforcement learning*) não são fornecidos dados nem de entrada nem de saída. O agente utiliza tentativa e erro para encontrar a solução ideal para o problema. São oferecidas recompensas ou penalidades ao programa por cada ação tomada, sendo o seu principal objetivo maximizar a recompensa total.

Apesar da crescente utilização de técnicas de *machine learning* em sistema de abastecimento de água, nomeadamente na previsão de consumos [17], na otimização de bombas de velocidade variável [18] e controlo de válvulas [19], poucos artigos existem com a utilização de *machine learning* na simulação e consequente minimização dos custos em sistemas de abastecimento de água [20].

Um dos primeiros projetos utilizando *machine learning* em redes de distribuição de água focado na otimização e no controlo em tempo real de um sistema de abastecimento de água foi o projeto POWADIMA [6,21–25]. O POWADIMA é um projeto financiado pela Comissão Europeia, tendo como principal objetivo auxiliar na redução de custos operacionais de um sistema de abastecimento de água em tempo real. O modelo proposto tem como base a substituição dos simuladores hidráulicos convencionalmente utilizados pelo uso de redes neuronais artificiais. Ao contrário dos simuladores hidráulicos, as redes neuronais artificiais (ANN) são computacionalmente muito mais eficientes e robustas, porém requerem uma grande quantidade de dados de treino para serem utilizadas. Para isso, é necessário primeiramente usar um simulador hidráulico para produzir dados de entrada e de saída de modo a serem utilizados no treino das redes neuronais artificiais. Depois de treinadas as ANN, foi introduzido um processo de otimização baseado em algoritmos genéticos [26]. Este processo é responsável por selecionar a melhor combinação de controlo das bombas/válvulas para satisfazer os requerimentos atuais de modo a minimizar os custos de energia. Este projeto consistiu num modelo de teste e dois casos de estudo diferentes.

Para o modelo de teste, Rao e Alvarruiz [6] escolheram o sistema “Any Town” por ser um modelo de teste simples e amplamente estudado na literatura (Figura 1). O sistema possui 41 tubos e 19 nós, com 3 bombas de velocidade fixa e 3 reservatórios. A camada de entrada é constituída por 5 variáveis: uma para o número de bombas em uso, um para os consumos agregados dos 19 nós e um para cada reservatório disponível. A camada de saída é definida por 7 neurónios: potência total consumida pelas bombas, pressão nos 3 nós representativos e

valor final de cada reservatório. Os resultados obtidos foram idênticos aos verificados pelo EPANET, sendo registada uma redução média de 10 vezes do tempo computacional.

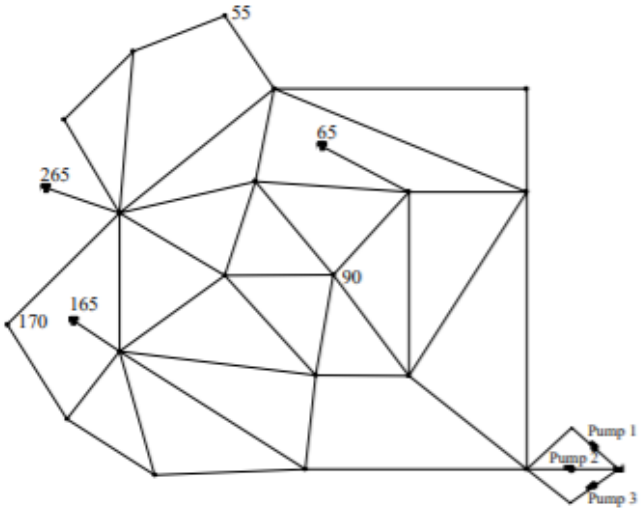


Figura 1 - Rede de distribuição de água "Any Town".

O primeiro caso de estudo do projeto é explicado por Salomons et al [24]. Neste caso de estudo foram utilizados dados reais da cidade de Haifa, Israel. Esta cidade é constituída por uma população de 60 mil habitantes. A rede é composta por 126 tubos, 112 nós, 9 reservatórios, 1 válvula redutora de pressão e 17 bombas em 5 estações de bombeamento diferentes. A operação do sistema apenas tem em conta o nível da água nos tanques, sem ser dada atenção aos diferentes preços da energia ao longo do dia. Foram utilizados ciclos temporais diários, divididos em intervalos de 1 hora. De modo a evitar que o tanque ficasse vazio no final do ciclo, impossibilitando a sua utilização no ciclo seguinte, foi fixado um valor mínimo nos tanques no final das 24 horas. A rede neuronal artificial apresentava 29 neurónios de entrada, 80 neurónios intermédios e 15 neurónios de saída (Figura 2). Os neurónios de entrada representavam o estado das bombas (on/off), os níveis dos reservatórios, o controlo da válvula e a procura em cada zona de pressão. Os neurónios de saída eram compostos pelos níveis dos tanques no momento final, a potência consumida por cada bomba e a pressão no nó crítico do sistema. Como não estavam disponíveis os valores dos consumos de água, estes foram obtidos numa combinação das séries de Fourier com séries de análise temporais. Salomons et al. aplicaram o sistema para todo o ano de 2000 obtendo uma redução de custos de 25,4%.

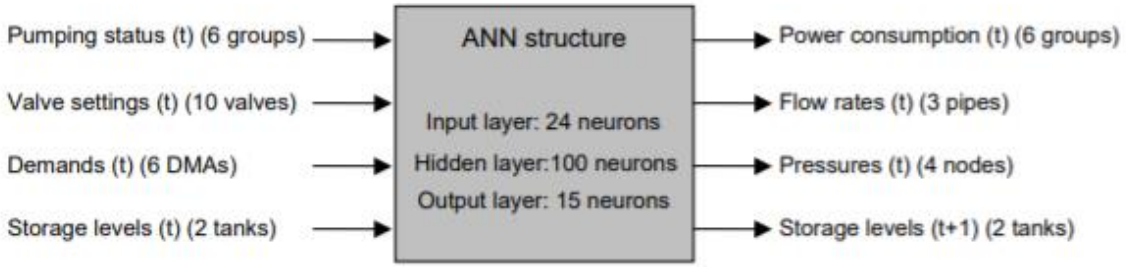


Figura 2 – Estrutura da ANN da rede de Haifa. Figura retirada de [24].

O segundo caso de estudo foi desenvolvido por Martinez et al. [25] e foi realizado na cidade de Valência, Espanha. Este sistema de abastecimento de água abastece aproximadamente 1,2 milhões de habitantes. É uma rede composta por 725 nós, 10 válvulas operacionais, 17 bombas em 2 estações de bombeamento e 2 reservatórios. A rede é composta por 24 neurónios de entrada: 6 grupos de bombas, 10 válvulas, a procura em cada zona das 6 zonas de pressão e o nível inicial dos dois reservatórios. A rede também é composta por 15 neurónios de saída: energia consumida por cada um dos grupos de bombas, pressão nos 4 nós críticos, níveis dos tanques no momento final e foram utilizados 100 neurónios na camada oculta (Figura 3). Martinez et al. analisaram os dados relativos ao ano de 2001, obtendo uma redução de custos de 17.6% e uma velocidade 94 vezes maior na simulação com este modelo quando comparado com o EPANET.

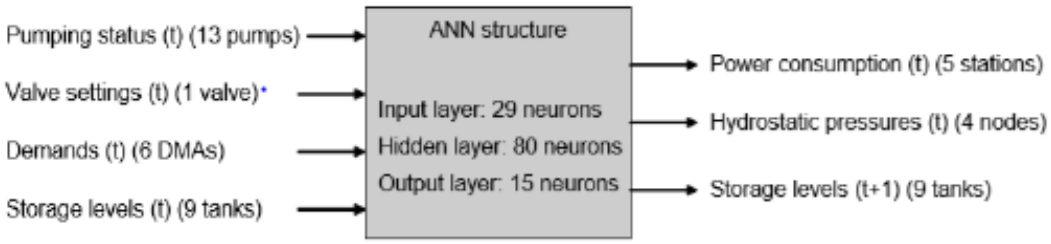


Figura 3 - Estrutura da ANN da rede de Valência usada em [25].

Odan et al. [27] utilizaram redes neuronais artificiais na cidade de Araraquara, Brasil. A estação é composta por doze tubos, um reservatório e duas bombas. Foi utilizada uma rede neuronal artificial utilizando a técnica AMGA [28], evitando o processo de determinar o número de neurónios ideal na camada oculta. A rede neuronal artificial tinha 3 neurónios de entrada: consumo total, estado da bomba e nível inicial do tanque e 3 neurónios de saída: energia da bomba, nível final do reservatório e pressão no nó crítico.

Behandish and Wu [29] também aplicaram redes neuronais artificiais a um sistema de abastecimento de água de dimensões superiores aos utilizados anteriormente. Este estudo foi realizado em Oldham, Inglaterra. Este sistema é composto por 3273 tubos, 12 reservatórios, 19 bombas e 420 válvulas. Várias restrições são assinaladas neste projeto, incluindo o nível dos tanques só poder variar entre os 0.3 e os 0.95 da sua capacidade total; o nível dos tanques no final e no início do ciclo não podiam ser superiores a 2.5% e cada bomba não poderia variar o seu estado (on/off) mais de 4 vezes durante o dia. Contrariamente aos trabalhos anteriores, Behandish and Wu utilizaram 12 sub-redes neuronais artificiais para simular o sistema (1 para cada tanque). Foram utilizados 24 neurónios de entrada, sendo posteriormente analisados individualmente (Figura 4). Cada variável de entrada foi testada utilizando o máximo, mínimo e média. Esta análise de sensibilidade resultou no desenvolvimento de sub-redes neuronais artificiais mais reduzidas, reduzindo consideravelmente o esforço computacional necessário.

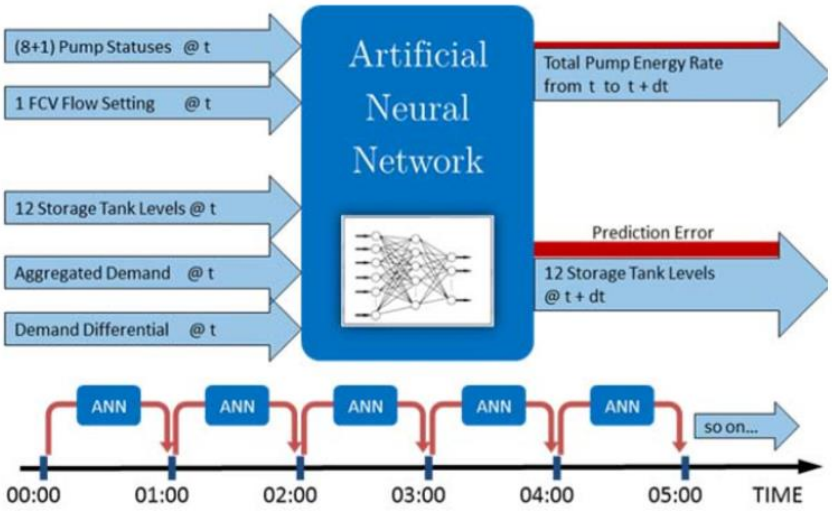


Figura 4 - Estrutura da rede ANN. Figura retirada de [30].

Wu et al. [30] por sua vez propuseram um método baseado em *deep learning* [31] (ou aprendizagem profunda). Ao contrário das redes neuronais artificiais simples, *deep learning* é composto por várias camadas ocultas, tentando imitar as várias camadas de neurónios do cérebro humano. A sua principal vantagem é a sua capacidade de automaticamente extrair *features*, treinando o sistema com dados não rotulados. Ao contrário dos estudos anteriores onde era necessário efetuar uma análise de sensibilidade antes de serem treinadas as ANN, sendo necessário identificar os dados de entrada sensíveis aos dados de saída desejados, com a técnica representada neste artigo deixa de ser necessário essa análise. A arquitetura de *deep learning* utilizada denomina-se *Deep Belief Network* (DBN) e é composta por *Restricted Boltzmann Machines* (Máquinas de Boltzmann restritas) empilhadas [32-33]. O DBN foi comparado com redes neuronais artificiais convencionais e com as suas sub-redes neuronais artificiais apresentadas no artigo anterior, apresentando melhores resultados.

A Tabela 1 apresenta de forma sistemática uma comparação dos modelos de *ML* utilizados, restrições e resultados obtidos em cada estudo mencionado anteriormente.

Autores	Data	Casos de estudo	Modelo	Restrições	Entrada	Saída	Resultados
Rao & Alvarruiz	2007	"Any Town"	DRAGA-ANN	- Pressão mínima nos nós - Nível mínimo e máximo dos depósitos - Limite máximo de energia em cada estação de bombeamento	- Controlo bombas (on/off) -Controlo válvulas - Nível inicial dos depósitos - Consumos nós	- Energia consumida por cada grupo de bombas - Pressão nós críticos - Nível final dos depósitos	- 10x mais rápido que EPANET
Salomons et al.	2007	Haifa, Israel					- Redução do custo 25% - 25x mais rápido que EPANET
Martinez et al.	2007	Valência, Espanha					- Redução do custo (17.6%) - 94x mais rápido que EPANET
Behandish & Wu	2012	Oldham, Inglaterra	Múltiplas-ANN-GA	- Nível mínimo e máximo nos depósitos - Pressão mínima e máxima nos nós - Fluxo máximo e mínimo nos tubos - Limite variações estado da bomba - Limite variação depósitos final e inicial	- Controlo bombas (on/off) - Controlo válvulas - Nível inicial dos depósitos - Consumo agregado - Diferencial do consumo	- Energia total das bombas - Nível final dos depósitos	- Redução do custo entre 10 e 15%
Wu et al.	2017		DBN-RBM				- Maior precisão que ANN
Odan et al.	2014	São Paulo, Brasil	Adaptive ANN-AMALGAM	- Pressão mínima - Nível final superior a inicial nos depósitos - 3 variações estado da bomba	- Nível Inicial dos depósitos - Controlo bombas - Consumo total	-Pressão nos nós -Consumo bombas - Nível final depósitos	- Redução 16% dos custos

Tabela 1 – Resumo artigos machine Learning na modelação de SAA.

3. Metodologia

Neste capítulo é explicada a metodologia utilizada na modelação e otimização de sistemas de abastecimento de água. Primeiramente, são discutidos os diferentes modelos implementados e a escolha e obtenção dos diferentes *inputs*/*outputs*. De seguida, são explicadas as diferentes técnicas *machine learning* utilizadas e a metodologia utilizada na otimização do sistema.

3.1 Modelos de simulação

Os SAA são sistemas extremamente grandes e complexos, estando sujeitos a um grande número de variáveis sobre as quais existe pouco controlo. Como o objetivo do modelo é prever o comportamento do sistema de modo a minimizar o custo associado ao bombeamento, é obrigatório saber dois valores: o nível de cada depósito e a energia das bombas no fim de cada intervalo temporal. Este tipo de problemas em que os modelos tentam prever variáveis contínuas são denominados de problemas de regressão. Duas formas diferentes são encontradas para a simulação da energia das bombas: energia total das bombas ou energia das bombas calculadas individualmente. Inicialmente, para todos os casos de estudo foram utilizados os dois processos, não se encontrando diferenças significativas entre os dois valores. Por isso, e porque o tarifário foi considerado igual para todas as bombas, foi decidido utilizar o processo mais simples e eficiente (a energia/potência total). Para além destas duas variáveis, outro *output* encontrado na literatura é a pressão em nós críticos. Todavia, neste trabalho esse valor não foi considerado.

Apesar de na literatura se utilizar variações temporais constantes de 1h, de modo a tornar o sistema mais robusto e eficiente, foi decidido utilizar variações temporais com valores diferentes, sempre iguais ou inferiores a 1h. Deste modo e ao contrário dos artigos revistos anteriormente, foi adicionado a variação temporal como dado de entrada no sistema (*feature*).

Outro aspeto a ter em conta é a utilização dos consumos como entrada. Como observado anteriormente, vários artigos utilizaram a soma dos consumos ao invés dos consumos individuais. Apesar de serem obtidos bons resultados, foi considerado mais indicado a utilização dos consumos individuais para este trabalho. Isto deve-se principalmente aos diferentes layouts que o sistema pode apresentar, podendo ter depósitos que apenas forneçam água a pontos de consumo isolados ou a diferenças de elevação em cada elemento, resultando em comportamentos da rede completamente distintos, principalmente o nível dos depósitos. Com pontos de consumo individuais essas diferentes variações do comportamento da rede são mais fáceis de serem assimiladas, resultando em modelos mais precisos.

Tal como verificado nas Figuras 5 e 6, foram utilizados dois modelos distintos. Para o primeiro modelo foram utilizados os valores de saída mencionados anteriormente: níveis finais do depósito e potência consumida no final de cada intervalo temporal. O segundo modelo é um modelo diferencial. Neste, é calculada a variação da potência das bombas ao longo do tempo (energia) e a variação dos níveis dos depósitos (nível depósito final-nível depósito inicial) em função do tempo, deixando de ser necessário a utilização do tempo como *feature* do sistema.

Assim, temos como valores de input do sistema:

- $[B_t^1, \dots, B_t^n]$ - Vetor com o estado de cada uma das $\mathrm{n}_{\mathrm{bombas}}$. Este valor pode ter os valores de 1 quando a bomba está ligada e 0 quando a bomba está desligada durante o intervalo de tempo;
- $[D_t^1, \dots, D_t^m]$ - Vetor com o nível inicial, em metros, de cada um dos $m_{\mathrm{depósitos}}$ presentes no sistema no instante t;
- $[C_t^1, \dots, C_t^o]$ - Vetor com o consumo médio, em m^3/h, de cada nó durante o intervalo $t + \Delta t$
- Δt – Variação temporal, em horas, de cada amostra. Apenas presente no modelo 1.

E como valores de output:

Modelo 1:

- $[D_{t+\Delta t}^1, \dots, D_{t+\Delta t}^m]$ - Vetor com o nível dos depósitos, em metros, de cada um dos depósitos no instante $t + \Delta t$;
- *E* – Energia total consumida pelas bombas, em kWh.

Modelo 2:

- $[\Delta D^1, \dots, \Delta D^m]$ - Variação temporal (velocidade) do nível de cada um dos depósitos, em m/h;
- *P* – Somatório das potências das bombas, em kW.

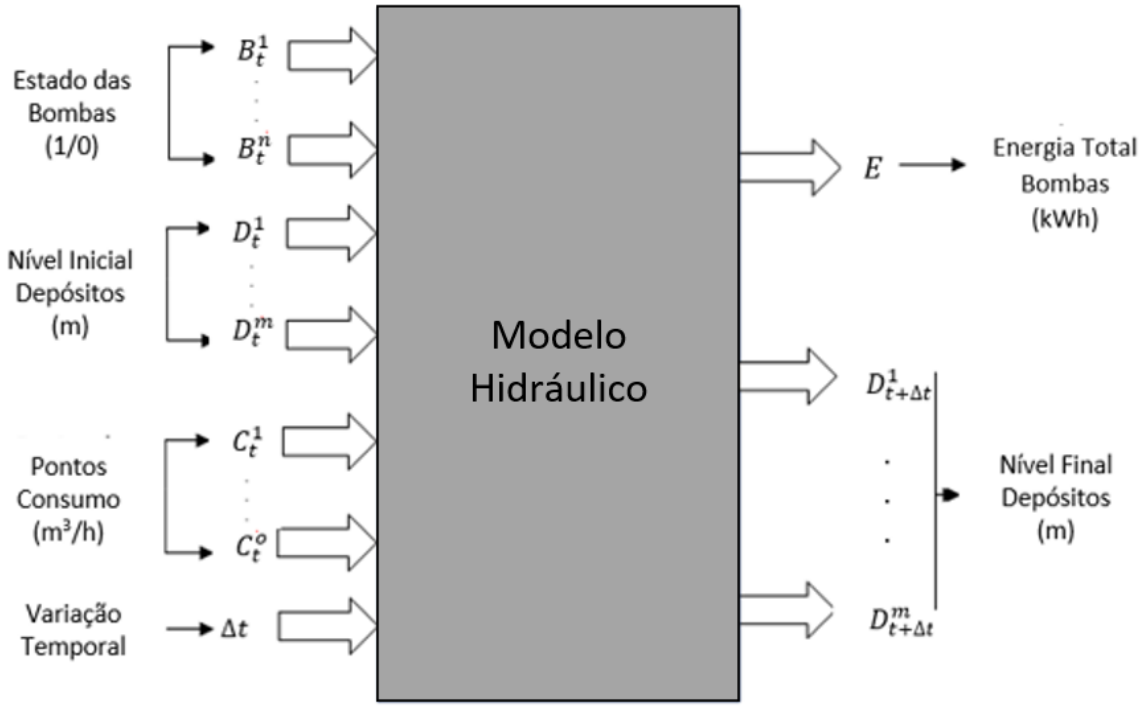


Figura 5 – Representação dos input/output do modelo 1.

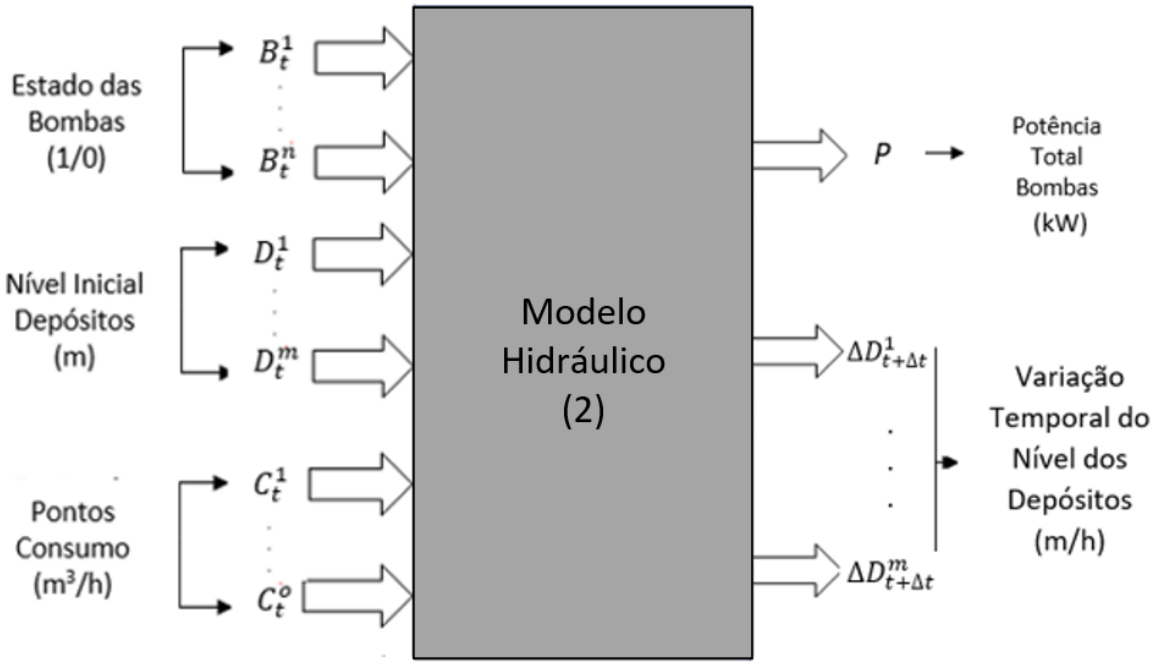


Figura 6 – Representação dos input/output do modelo 2.

3.2 Criação e treino do modelo

Neste trabalho são analisados dois algoritmos diferentes de *Machine Learning*: ANN e XGBOOST. Como analisado no capítulo 2, os algoritmos baseados em redes neuronais artificiais apresentaram excelentes resultados na simulação de sistemas de abastecimento de água. Embora redes neurais mais complexas sejam encontradas na literatura com melhores resultados (tal como *Deep Belief Networks*), estas diferenças foram reduzidas, não compensando o esforço computacional consideravelmente superior e, por isso, foi decidido utilizar simples ANN. Apesar do XGBOOST [37] não ser encontrado na literatura na modelação de sistemas de abastecimento de água, este tem apresentado excelentes resultados em competições de *Machine Learning* na plataforma *Kaggle* [34], sendo considerado um dos melhores algoritmos para problemas de regressão e classificação.

3.2.1 ANN

As redes neuronais artificiais [35] são um dos algoritmos mais populares em *Machine Learning*. O seu nome e estrutura são inspirados no cérebro humano, tentando replicar o seu funcionamento. Estas são compostas por uma camada de entrada, uma camada de saída e por uma ou mais camadas ocultas (Figura 7). A camada de entrada é a primeira camada do sistema e o seu tamanho depende do número de *features* utilizadas no sistema. A camada de

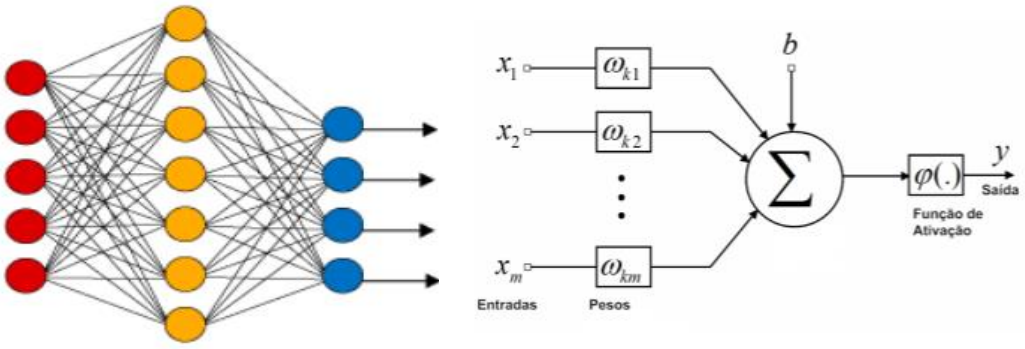
saída é a última camada da rede e o seu número de neurónios é igual ao número de outputs do sistema. As camadas ocultas são camadas intermédias e o seu número de neurónios dependem da complexidade do sistema. Independentemente do seu tamanho ou complexidade, todas as redes neurais são constituídas por neurónios (*perceptrons*) (Figura 7) e por conexões (sinapses) que definem como os neurónios se associam uns aos outros. Para cada *perceptron* inicialmente é calculada a soma de todos os inputs (**x**) multiplicada pelo peso (**w**) associado a cada conexão e de seguida é adicionada um valor *bias*, *b*, que permite mover a função de ativação:

$$\sum_{i=0}^{m} w_i x_i + b. \tag{1}$$

No final deste processo, é aplicada uma função de ativação no resultado, de modo a introduzir não-linearidade, sendo esse resultado a saída do *perceptron*. Uma das funções de ativação mais utilizadas em problemas de regressão é a função *ReLu* [35]. Esta função de ativação utiliza um simples cálculo e garante não-linearidade com um reduzido esforço computacional. Esta retorna o próprio valor caso este seja superior a 0, retornando 0 caso este seja inferior ou igual a 0. A função *ReLu* é definida por:

$$\sigma(x) = \max(0, x), \tag{2}$$

com $x = \sum_{i=0}^{m} w_i x_i + b$, sendo esse o resultado do output. Para a rede aprender, é necessário aplicar um algoritmo de treino. O algoritmo de otimização utilizado para treino (*SGD*, *Adam* ou *BFGS*) tem como objetivo minimizar uma função de custo, ajustando cada peso tendo em conta a diferença entre o output de cada iteração e o valor esperado. Este método, para calcular o gradiente, é conhecido como *Backpropagation*.



(a) (b)

Figura 7 - Representação de a) ANN e b) perceptron.

Para a implementação das redes neuronais foi utilizada a biblioteca Keras [36]. O Keras é uma biblioteca bastante popular pela sua simplicidade, flexibilidade e facilidade de aprender e trabalhar. Como verificado anteriormente, a construção das ANNs é um processo iterativo, não havendo uma solução ideal. As ANNs são essencialmente definidas pelo seu tamanho

(número de neurónios e número de camadas), função de ativação e algoritmo de otimização utilizado para o treino. Apesar de não haver consenso em relação ao tamanho e número de neurónios (necessário otimizar no treino das redes), a utilização da função de ativação ReLu (nas camadas ocultas) e o algoritmo de otimização Adam são os mais aconselhados pelos anteriores trabalhos na simulação de SAA e por isso são os utilizados neste trabalho. Um dos hiperparâmetros mais importantes de ajustar é a taxa de aprendizagem (*learning rate*) do algoritmo de otimização do treino. Encontrar um valor adequado para a taxa de aprendizagem é um processo de tentativa e erro, pois valores demasiado grandes levam à divergência dos resultados e valores demasiado pequenos levam a tempos de convergência excessivamente elevados.

3.2.2 XGBOOST

O *XGBoost* (*Extreme Gradient Boosting*) foi originalmente proposto por Chen e Guestrin [37] e é o algoritmo mais utilizados pela comunidade em problemas de regressão e classificação. É um método baseado em árvores de decisão, mas com uma componente de treino.

Em *Machine Learning*, *boosting* [38] é uma técnica de conjunto (*ensemble*) que tenta criar um modelo robusto (*strong learner*) de múltiplos modelos mais fracos (*weak learners*). O uso de um conjunto de modelos permite uma melhor performance do que a utilização de modelos individuais, ajudando a reduzir a variância e reduzindo desta forma o erro [39]. Inicialmente, é criado um modelo com os dados de treino, sendo de seguida criado um segundo modelo que tenta corrigir os erros do primeiro modelo. São sucessivamente criados modelos até se atingirem erros inferiores a um determinado valor ou a serem obtidos o número máximo de modelos previamente estipulados. No final, é realizada a média ponderada de todos os modelos de acordo com a sua performance (um peso maior é dado aos modelos mais fracos).

Embora o *XGBoost* seja baseado no modelo *Gradient Boosting* [40], apresenta algumas melhorias que ajudam a melhorar a performance. Enquanto o *Gradient Boosting* original adiciona os *weak learners* em sequência, o XGBOOST utiliza processamento paralelo, resultando num modelo muito mais rápido. Para além disso, também permite a regularização dos modelos, ajudando a diminuir *overfitting*.

O *XGBoost* disponibiliza uma grande variedade de hiperparâmetros que dependem da complexidade do sistema e por isso são necessários ajustar:

- *learning_rate* – define o quão rapidamente o modelo aprende. Quanto menor o seu valor maior a precisão do modelo e o custo computacional. O valor pode variar entre]0; 1];
- *max_depth* – indica a altura máxima de cada árvore utilizada no modelo. Quanto maior o seu número, mais complexo será o modelo e maior será a tendência de ocorrer *overfitting*. Porém, um número pequeno leva a um modelo demasiado simples (*underfitting*). O valor pode variar entre [0,∞[;
- *Min_child_weight* – o peso mínimo requerido para criar um novo nó na árvore. Um valor baixo leva a um modelo mais complexo, mas mais propenso a ocorrer *overfitting*. O valor pode variar entre [0,∞[;
- *n_estimators* - número de árvores utilizadas]0,∞[.

3.2.3 Avaliação do modelo

A avaliação do desempenho de um modelo é uma das fases mais importantes no desenvolvimento de modelos de *machine learning*, uma vez que ajuda a analisar a qualidade dos modelos. Em problemas de regressão, a missão dos modelos é prever valores numéricos e analisar o quão próximos estes se encontram dos valores esperados.

3.2.3.1 Erro Absoluto Médio (MAE)

É uma métrica bastante utilizada em problemas de regressão. Como o próprio nome indica, calcula a média dos erros absolutos entre o valor esperado e o valor obtido:

$$MAE = \frac{\sum_{i=0}^{z} |y_i - \hat{y}_i|}{n}, \tag{3}$$

onde y_i é o valor obtido de cada amostra pelo modelo, $\hat{y}_i$ o valor esperado de cada amostra e *n* o número total de amostras. O MAE apenas retorna a magnitude dos erros e não a sua direção.

3.2.3.2 Raiz quadrada do erro quadrático médio (RMSE)

O RMSE é a métrica mais indicada em problemas de regressão e a utilizada em competições de machine learning. A sua fórmula é semelhante à MAE, porém, ao contrário do MAE que apresenta erros lineares, os erros do RMSE são quadráticos, ou seja, erros mais elevados apresentam um maior peso no resultado final. Devido a esta particularidade, é necessário ter particular atenção à sua utilização, pois é particularmente sensível a *outliers* (valores completamente distintos dos restos dos valores, podendo existir devido a erro de medida ou de entrada dos dados, ou por serem pontos obtidos por um processo diferente). O RMSE é dado pela seguinte expressão:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n} (y_i - \hat{y}_i)^2}{n}}. \tag{4}$$

3.2.3.3 Coeficiente de determinação R^2

Outra métrica frequentemente utilizada em problemas de regressão é o coeficiente de determinação R^2. O R^2 é uma medida estatística que reflete o quão próximo estão os dados da linha de regressão ajustada, podendo ser entendida como a percentagem total de variância explicada pelo modelo. O R^2 é dado pela seguinte fórmula:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{amostras}-1} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{amostras}-1} (y_i - \bar{y})^2}, \tag{5}$$

onde $\bar{y}$ é a média de *y*.

3.2.4 Criação de dados sintéticos

Com o propósito de avaliar o comportamento e robustez dos modelos em SAA, são analisadas três redes com tamanhos, número de elementos e complexidade distintas. Nos dois primeiros casos de estudo são utilizados dados sintéticos obtidos através do simulador hidráulico EPANET. Estes dois casos de estudo, embora apresentem redes mais simples que a rede do último caso de estudo, têm como principal objetivo aprimorar técnicas e ganhar competências necessárias para problemas de grau de dificuldade maior.

Em casos reais, é extremamente comum as empresas responsáveis pelos SAA receberem dados com erros devido a problemas nos sensores ou na aquisição e transferência dos dados. Mesmo que não ocorra nenhum erro, os sensores operam com uma margem de erro, diminuindo a precisão dos dados. A maioria dos erros (devido a outliers) são facilmente identificados, porém o ruído constitui um grave problema. Como para os casos de estudo 1 e 2 os dados são obtidos sinteticamente (pelo simulador hidráulico EPANET), estes não apresentam ruído, facilitando o comportamento dos modelos machine learning. Deste modo, foi adicionado ruído artificial à base de dados destes dois casos de estudo e analisado o comportamento de cada modelo. O processo de adição do ruído é explicado no primeiro caso de estudo.

Outro aspeto importante que é explorado neste capítulo é a quantidade de dados que são necessários fornecer ao treino dos modelos. Ao contrário dos dados obtidos sinteticamente, em casos de estudo reais a quantidade de dados disponíveis pode ser reduzida. Isto pode dever-se a alterações provocadas no sistema (tal como a adição de um novo ponto de consumo, alteração dos níveis dos depósitos permitidos, alteração das bombas) e por isso é observada como a variação da quantidade de dados altera o desempenho dos modelo. É analisada a quantidade mínima de dados necessária a fornecer aos modelos sem que a sua precisão seja consideravelmente prejudicada.

3.2.5 Treino, Teste e Validação do modelo

Para os dois primeiros casos de estudo foram obtidos dados sintéticos através de modelos hidráulicos (EPANET) de modo a substituir dados reais. Apesar dos modelos *machine learning* conseguirem obter bons resultados em sistemas complexos, estes necessitam de uma grande quantidade de dados. Embora na literatura não haja um consenso em relação ao número de amostras necessárias a fornecer ao sistema, uma vez que este valor depende maioritariamente da complexidade dos sistema, foi decidido utilizar 50000 amostras para os dois primeiros casos de estudo. De modo a conseguir um sistema o mais robusto possível e que consiga representar com precisão os variados comportamentos da rede e apresentar uma maior generalização, foi tentado criar o maior número diferente de combinações de entrada possíveis. Deste modo os valores dos depósitos iniciais e os estados das bombas e consumos em cada instante foram gerados aleatoriamente.

De forma a evitar *overfitting* (o modelo apresenta excelentes resultados nos dados de treino, porém não generaliza, obtendo resultados consideravelmente inferiores quando confrontado com dados não treinados anteriormente) e perceber o desempenho real do modelo, é necessário dividir a base de dados em treino, teste e validação. Para o treino do

sistema serão utilizados aleatoriamente 85% dos dados obtidos anteriormente, sendo os outros 15% utilizados na validação do modelo.

Como o objetivo do sistema é funcionar de forma contínua, para a validação do sistema é utilizado um novo dia de teste (24 horas). Os dados foram adquiridos com o mesmo processo realizado na aquisição dos dados de treino e teste. São disponibilizados aos modelos os valores dos níveis dos depósitos no início do dia, sendo posteriormente utilizados como níveis iniciais dos depósitos os valores previstos no espaço temporal anterior.

3.3 Otimização do sistema

Hoje em dia, todos os sistemas de engenharia devem ser eficientes tanto energeticamente como economicamente. Os sistemas de bombagem, particularmente as estações elevatórias de sistemas de abastecimento de água, devem ser geridos de forma a minimizar os custos energéticos. Para isso, e com auxílio de depósitos, as bombas devem ser operadas nos períodos onde o custo energético é mais reduzido. Contudo, devido a limitações nas dimensões dos depósitos e na capacidade de bombagem, não é possível bombear a totalidade de água que é necessária fornecer aos consumidores no período mais económico. Deste modo, é essencial haver um planeamento tendo em conta as dimensões dos depósitos, as necessidades de consumo de água e as capacidades de bombagem. Este planeamento deve conduzir ao menor custo possível de energia total. Assim, o objetivo passa por encontrar o estado de operação das bombas de forma a

$$\text{minimizar } C(\boldsymbol{x}) = \sum_{i=1}^{n} \sum_{l=1}^{m_{inc}} P_i \, x_{i,l} \, t_{\text{intervalo},l} \, Tarif_l \, , \tag{6}$$

onde *n* é o número total de bombas, m_{inc} o número total de incrementos temporais, P_i a potência de cada bomba, $x_{i,l}$ o tempo de funcionamento de cada bomba em cada incremento e $Tarif_l$ o custo da energia no período i. Sendo x a variável que define o tempo de funcionamento das bombas em cada incremento temporal, esta varia continuamente entre 0 e 1. Contudo, esta variável é posteriormente transformada como variável binária 0 ou 1, onde 1 corresponde ao funcionamento da bomba, dividindo o intervalo de tempo em dois subintervalos temporais

$$t_{\text{func}} = x \, t_{\text{intervalo}}, com \, x \in [0,1]. \tag{7}$$

No subintervalo inicial, a bomba encontra-se em funcionamento, e no subintervalo final ($t_{\text{intervalo}} - t_{\text{func}}$), a bomba encontra-se em repouso. Para além disso, é necessário restringir o sistema de forma a garantir que o nível máximo e mínimo dos depósitos *d* não é ultrapassado:

$$g_i = \begin{bmatrix} g_{1,i}^1 \\ g_{2,i}^1 \\ \vdots \\ g_{1,i}^d \\ g_{1,i}^d \end{bmatrix} = \begin{bmatrix} {h^1}_{min} - {h^1}_{F,i} \\ {h^1}_{F,i} - {h^1}_{max} \\ \vdots \\ {h^d}_{min} - {h^d}_{F,i} \\ {h^d}_{F,i} - {h^d}_{max} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \; i = 1, \dots, m_{inc}. \tag{8}$$

3.3.5 Tarifa

Uma das principais considerações em qualquer problema de otimização que envolva minimizar o custo de energia está relacionado com a tarifa elétrica. Como as tarifas elétricas dependem das entidades reguladores locais (variando com o local, estação do ano, dias da semana, feriados), serão utilizadas diferentes tarifas para os diferentes casos de estudo. As tarifas também são diferentes ao longo do dia sendo, na sua maioria, tetra-horárias ou penta-horárias.

3.3.6 Intervalos temporais de operação ótima

Embora seja possível utilizar ciclos diferentes de 24h, a maioria dos SAA's opera neste intervalo de tempo. Em relação aos intervalos de tempo, tal como aconselhado na literatura foi decidido utilizar intervalos de 1h. Apesar de ser algo conservativo, intervalos de tempo mais pequenos poderiam levar a cargas computacionais demasiado elevadas, impossibilitando a sua utilização em SAA em tempo real.

3.3.7 Implementação

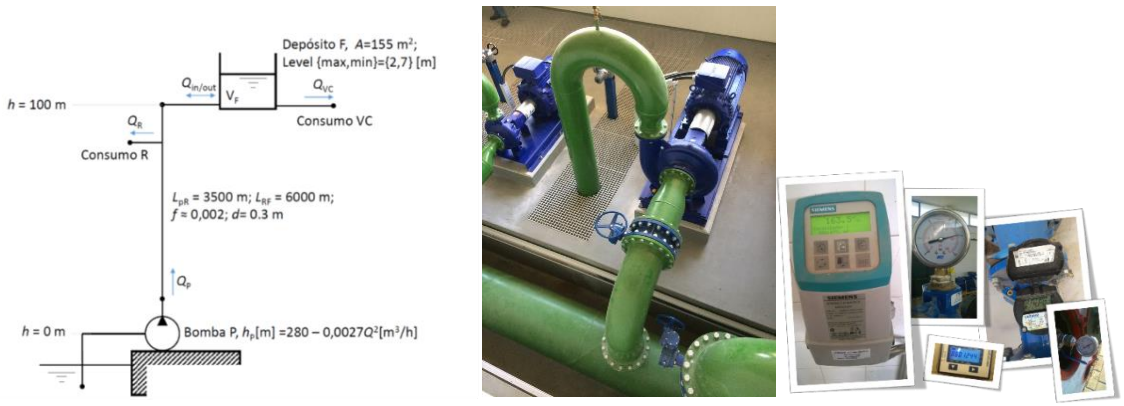
Para a implementação de procedimentos de otimização foi utilizada a biblioteca *optimize* do scipy [41]. O scipy é composto por vários algoritmos de otimização para problemas não lineares, cálculo de raízes e ajustes de curva. Para este trabalho é utilizado um método de otimização do tipo de região de confiança e método de ponto interior para as restrições não lineares de desigualdade (*trust-constraint)* [41]. É o algoritmo mais versátil dos implementados no scipy e preparados para restrições. É também o mais apropriado para problemas de larga-escala. Este método calcula o gradiente através do método das diferenças finitas e utiliza métodos quase-Newton (método BFGS) [42] para aproximar a Hessiana. Para encontrar as soluções das novas variáveis, utiliza o gradiente tanto da função objetivo como das restrições, usando uma região de confiança.

4. Resultados

Neste capítulo são apresentados os resultados obtidos utilizando a metodologia descrita no capítulo anterior. Cada modelo é avaliado de acordo com as métricas referidas anteriormente, utilizando um ou mais dias de teste, sendo também efetuada uma comparação dos modelos. É feita a otimização do sistema para o caso de estudo do subsistema de água da Fontinha.

4.1 Caso de estudo 1: sub-rede da Fontinha

A rede da fontinha (Figura 8) é um subsistema de abastecimento de água e faz parte de uma rede maior localizada na zona centro de Portugal. É constituída por uma única bomba (75% de eficiência) que se encontra a uma cota de 0 m. Esta bomba, quando está em funcionamento, é responsável por abastecer a região R e fornecer água para o depósito. O único depósito tem 155 m^2 de área, está a uma cota de 100 m e tem capacidade para um nível máximo de 9 m. Contudo, por razões de segurança, apenas opera entre os níveis 2 e 7 m. Este depósito é responsável pelo abastecimento dos consumidores da região VC, podendo também abastecer os consumidores da região R sem qualquer custo de energia caso a bomba não esteja em funcionamento.



(a) (b) (c)

Figura 8 - (a) Esquema e (b-c) equipamentos/sensores do caso de estudo da Fontinha.

4.1.1 Obtenção dos dados

Como referido anteriormente, para treinar os modelos *ML* é necessário fornecer ao modelo um conjunto de dados de entrada/saída. Para a rede da fontinha foram utilizadas 5 variáveis de entrada para o modelo 1 e 4 para o modelo 2. As *features* de entrada são:

- B_t - Estado da única bomba;
- D_t- Nível inicial do único depósito (m);
- $[C_t^1, C_t^2]$ – 2 Pontos de consumo relativos aos pontos R e VC (m^3/h) ;
- Δt – Variação temporal (apenas para o 1º modelo) (h),

e 2 variáveis de saída para cada modelo. Para o modelo 1 estas são:

- $D_{t+\Delta t}$ - Nível final do depósito (m);
- *E* – Energia utilizada pela bomba (kWh).

Para o modelo 2 estas são:

- ΔD - Variação nível do depósito por hora (m/h);
- *P* – Potência de funcionamento da bomba.

Os dados foram obtidos através do software EPANET. Depois de adquiridos, os dados foram normalizados (todos os valores se encontram entre 0 e 1) e divididos em treino, teste e validação. Para normalizar os dados foi utilizada a seguinte expressão:

$$x_{\text{norma}} = \frac{x_i - x_{min}}{x_{max} - x_{min}}. \quad (9)$$

4.1.2 Criação e treino dos modelos

O processo de escolha manual dos hiperparâmetros dos modelos é um processo muito demorado e pouco eficiente. Este processo pode ser melhorado através da técnica *GridsearchCV* [43]. O *GridSearchCV* é um módulo presente na biblioteca *scikit-learn* e é uma ferramenta utilizada para automatizar o processo de escolha dos hiperparâmetros dos modelos. Inicialmente é necessário fornecer uma gama de valores para os quais se quer testar cada hiper-paramêtro. De seguida, o *GridSearchCV* faz a combinação de todos os modelos fornecidos e, através de validação cruzada avalia a média do erro de cada uma das combinações (para este caso foi utilizado o RMSE), fornecendo os hiperparâmetros que melhor generalizam.

A Tabela 2 apresenta a gama de valores utilizados para cada hiperparâmetro. De modo a encontrar a configuração ideal, foram testadas inúmeras redes com 1, 2 e 3 camadas ocultas. De modo a limitar o número de opções para o número de neurónios por camada, foi decidido utilizar múltiplos do número de neurónios da camada de entrada (5). Alguns dos valores obtidos podem ser encontrados na Figura 9. Foi utilizado o RMSE como métrica de avaliação. Para os modelos com as ANN (Tabela 3) foi observado que o melhor modelo apresentava 2 camadas ocultas com 60 e 15 neurónios. A função de ativação dos neurónios das duas camadas ocultas foi a *ReLu*, sendo utilizada a função de identidade ($f(x) = x$) como função de ativação da última camada. A função de otimização para o treino utilizada foi o *Adam* com

um learning rate constante de 0.00025 e um *batch size*[2] de 25. A Figura 11 mostra a curva da função de perda por *epoch*, utilizando o MAE como métrica.

Por sua vez, para o XGBoost foram utilizados os valores que se encontram na Tabela 2, perfazendo um total de 240 combinações. A Figura 10 mostra os erros obtidos de cada valor dos diferentes hiperparâmetros utilizados. Tal como anteriormente, para calcular o erro, todos os hiperparâmetros foram fixos (utilizando os melhores valores encontrados no *gridsearch*), variando apenas o hiperparâmetro a testar. Nos modelos utilizando o *XGBoost* (Tabela 4) foi utilizado uma taxa de aprendizagem (*learning_rate*) de 0.05 e um número mínimo de 5 amostras para ocorrer a separação dos nós das árvores (*min_child_weight*). O tamanho máximo para cada decisão de árvore foi de 15 (*max_depth*). A métrica de avaliação utilizada foi o RMSE (*eval_metric*) e foram utilizadas 750 árvores (*n_estimators*). Na Figura 12 é possível observar a curva da função de perda (*MAE*) com o aumento do número de árvores (*n_estimators*) do XGBoost.

ANN	
Hiperparâmetros	**Valores**
Nº camadas ocultas	1, 2, 3
Nº Neurónios/ camada	[10, 15, 20, … , 90]
Batch_size	[25, 50, 75, 100]
Learning_rate	[0.0001, 0.00025, 0.0005, 0.00075]
Função de ativação	ReLu
Algoritmo de otimização	Adam
XGBoost	
Hiperparâmetros	**Valores**
learning_rate	[0.05, 0.1, 0.15, 0.2]
min_child_weight	[1, 3, 5]
max_depth	[5, 10, 15, 20]
n_estimators	[100, 250, 500, 750, 1000]

Tabela 2 – Hiperparâmetros ANN e XGBoost.

[2] número de amostras utilizadas para estimar o gradiente. Um grande número de amostras pode levar a resultados mais precisos, necessitando de menos iterações para convergir, contudo leva a tempos mais elevados por iteração

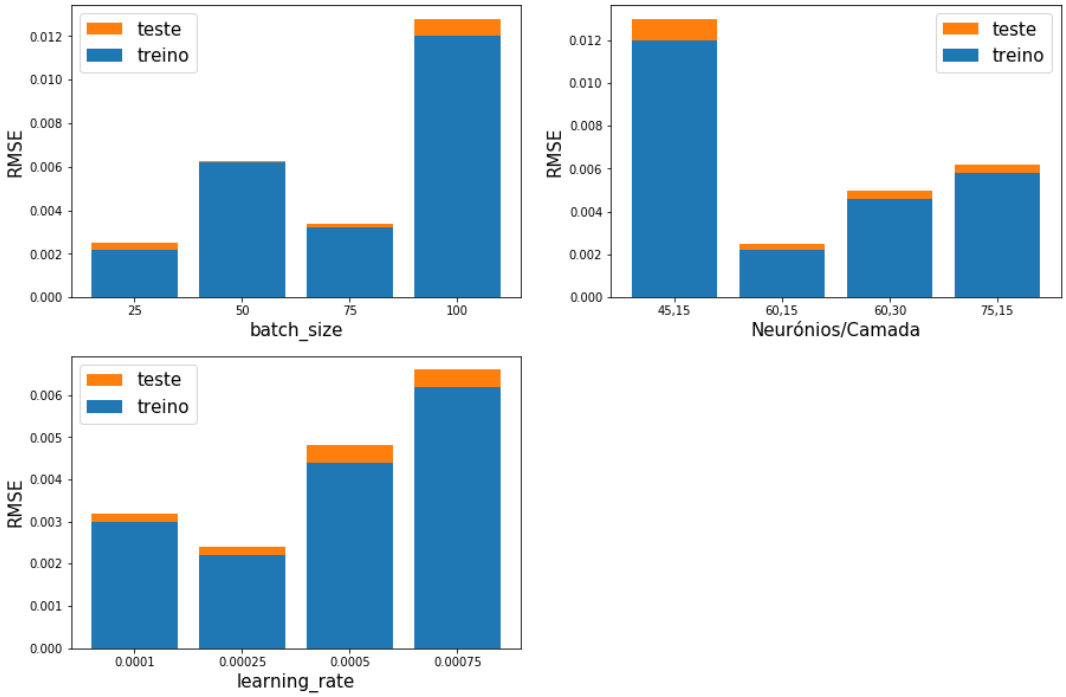


Figura 9 – Erro treino/teste dos diferentes hiperparâmetros ANN - Fontinha.

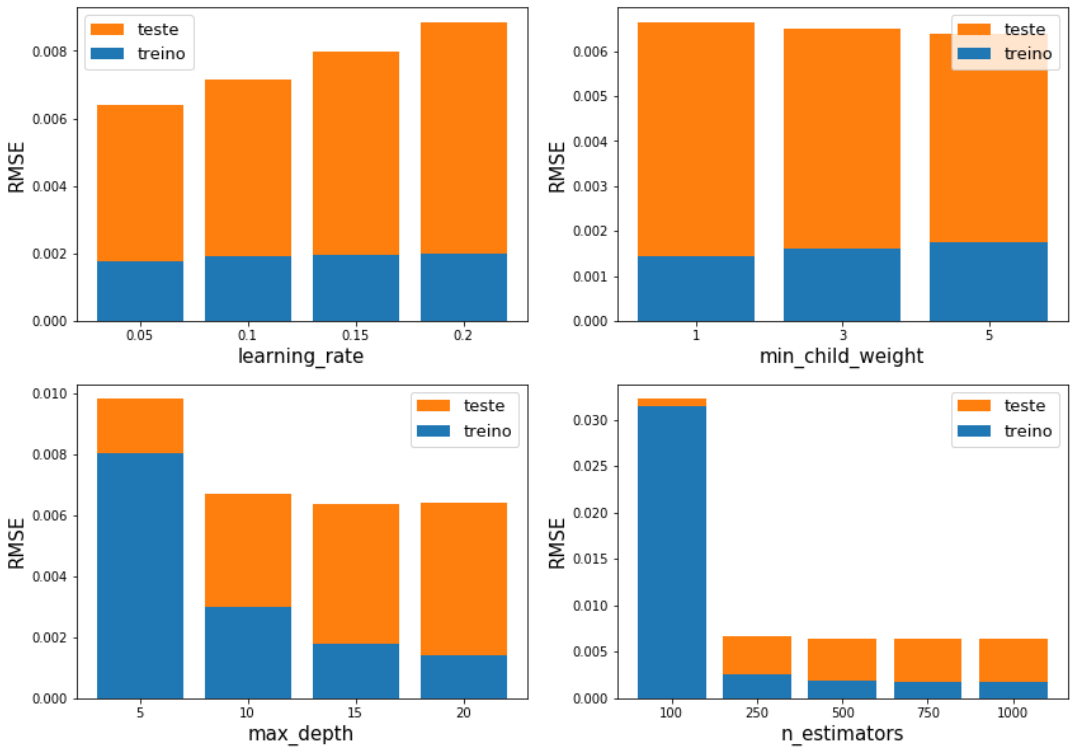


Figura 10 – Erro treino/teste dos diferentes hiperparâmetros XGBoost - Fontinha.

Nº camadas ocultas	Nº de neurónios/camada	batch_size	learning_rate	Função de Ativação	Algoritmo de Otimização
2	60,15	25	0.00025	ReLu	Adam

Tabela 3 - Configuração ANN – Fontinha.

learning_rate	min_child_weight	max_depth	n_estimators
0.05	5	15	750

Tabela 4 – Configuração XGBoost – Fontinha.

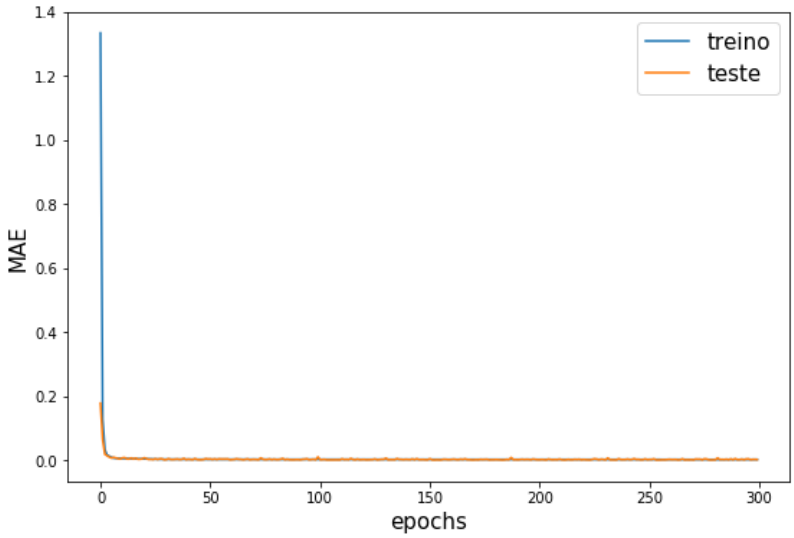


Figura 11 - Curva da função de perda ANN - Fontinha.

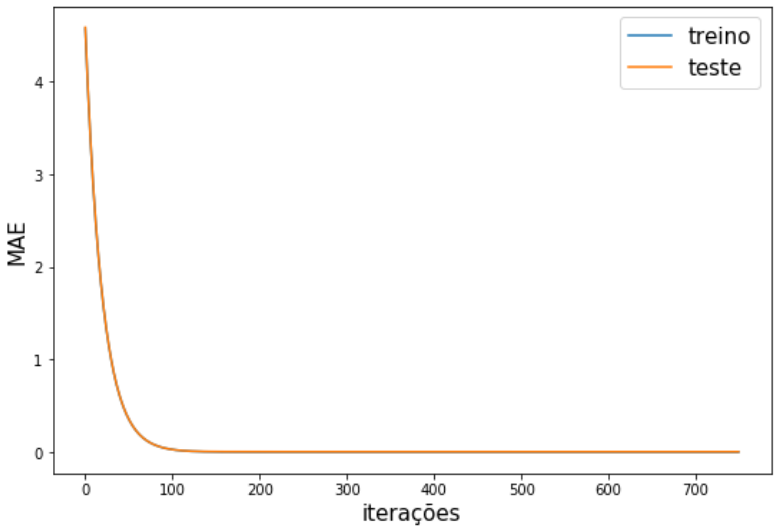


Figura 12 - Curva da função de perda XGBoost – Fontinha.

4.1.3 Validação do modelo

Concluído o processo de treino e teste dos modelos é necessário proceder à sua validação. Como os SAA funcionam num sistema contínuo, não bastando prever o comportamento num determinado momento, mas também os próximos num ciclo de 24h, foi utilizado um dia de teste obtido aleatoriamente como explicado no capítulo anterior. A Figura 13 demonstra o comportamento de cada um dos constituintes da rede ao longo do dia. A Figura 14 mostra os valores previstos e os valores obtidos por cada um dos modelos. Para o caso dos modelos diferenciais (ANN2 e XGBOOST2), como estes previam a variação dos níveis dos depósitos, foi necessário multiplicar cada valor previsto pelo respetivo intervalo temporal e posteriormente adicionado ao nível inicial dos depósitos. Para o caso da potência da bomba no modelo 1, como referido anteriormente, inicialmente foi calculada a energia resultante, sendo de seguida dividida por cada espaço temporal, obtendo assim a potência. Os resultados demonstram um comportamento praticamente perfeito de todos os modelos tanto na previsão do nível do depósito como da potência da bomba. Um resumo dos resultados obtidos pode ser encontrado na Tabela 5. No caso dos depósitos todos os modelos apresentaram valores do RMSE inferiores a 0.01 m e RMSE inferior a 0.05 kW na previsão dos valores de potência da bomba. Todos os modelos apresentaram um R^2 muito próximo de 1.

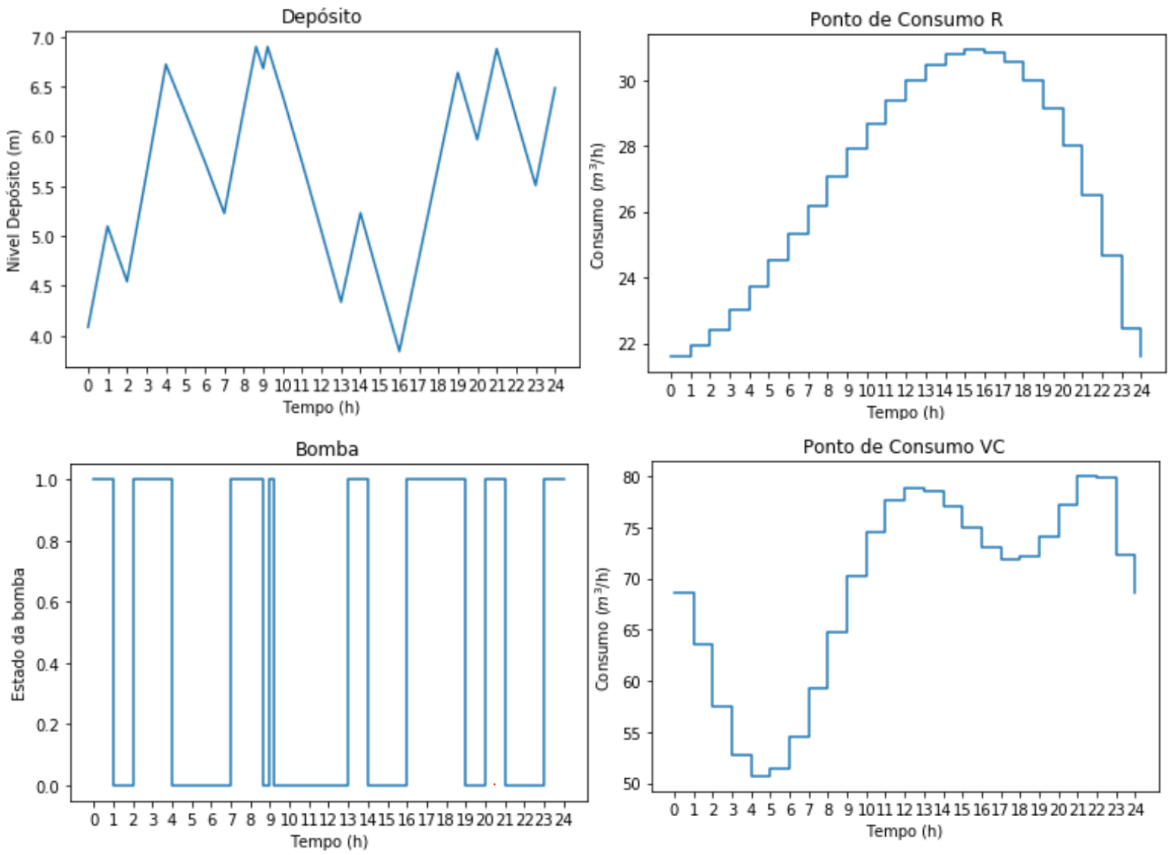


Figura 13 - Comportamento dos inputs do modelo no dia de teste da rede da Fontinha.

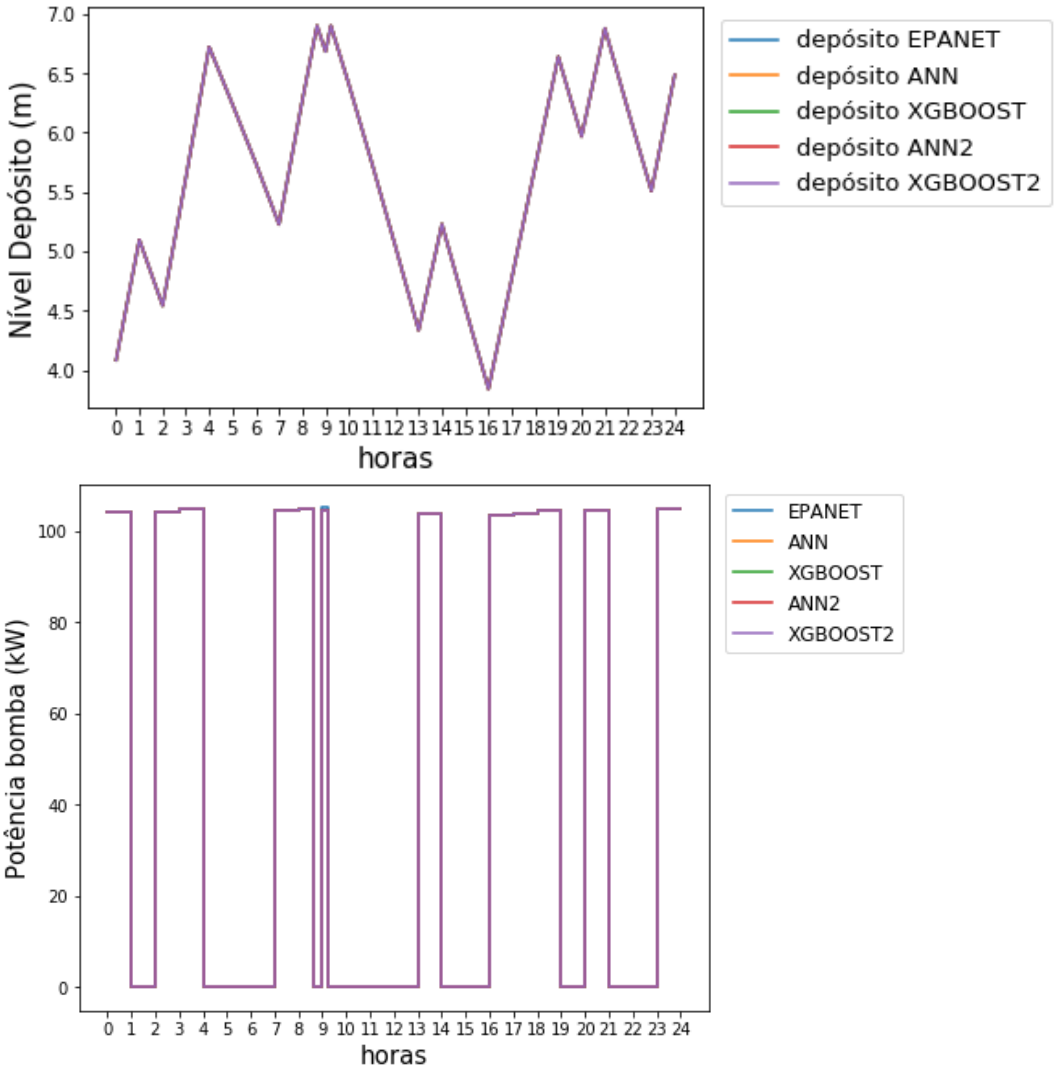


Figura 14 - Simulação (a-b) do depósito e da energia para o dia de teste.

		ANN	XGBoost	ANN2	XGBoost2
Nível do Depósito (m)	**RMSE**	0.0059	0.0063	0.0051	0.0017
	MAE	0.0051	0.0053	0.0043	0.0014
	R^2	0.9999	0.9999	0.9999	0.9999
Potência bombas (kW)	**RMSE**	0.0390	0.0176	0.0085	0.0080
	MAE	0.0238	0.0067	0.0036	0.0031
	R^2	0.9999	0.9999	0.9999	0.9999

Tabela 5 - Resultados obtidos para cada modelo para o caso de estudo da rede da Fontinha. Comparação com os dados obtidos pelo simulador EPANET.

4.1.4 Análise ao ruído

Foram colocados ruídos de 5, 10 e 15% no nível do depósito, nos dois pontos de consumo e na potência da bomba. O ruído foi gerado aleatoriamente seguindo uma distribuição uniforme (ruído gaussiano). Os resultados são avaliados com as mesmas métricas e o mesmo dia de teste utilizado anteriormente. Tal como anteriormente, de modo a realizar uma comparação mais justa dos modelos, a energia do modelo 1 foi transformada em potência, bem como a variação do nível do depósito para o modelo 2 em nível final.

As Figuras 15, 16, 17 e 18 mostram a variação do *RMSE*, *MAE* e R^2 com o aumento do ruído para cada um dos modelos. Como é possível pelas figuras, os níveis de ruído colocado não parecem apresentar uma grande influência no desempenho dos modelos na previsão da potência da bomba. Contudo, na previsão dos níveis dos depósitos é possível verificar diferenças bastante significativas com 15% de ruído para os dois modelos 1, apresentando RMSE superiores a 30 cm tanto para o XGBoost, como para a ANN.

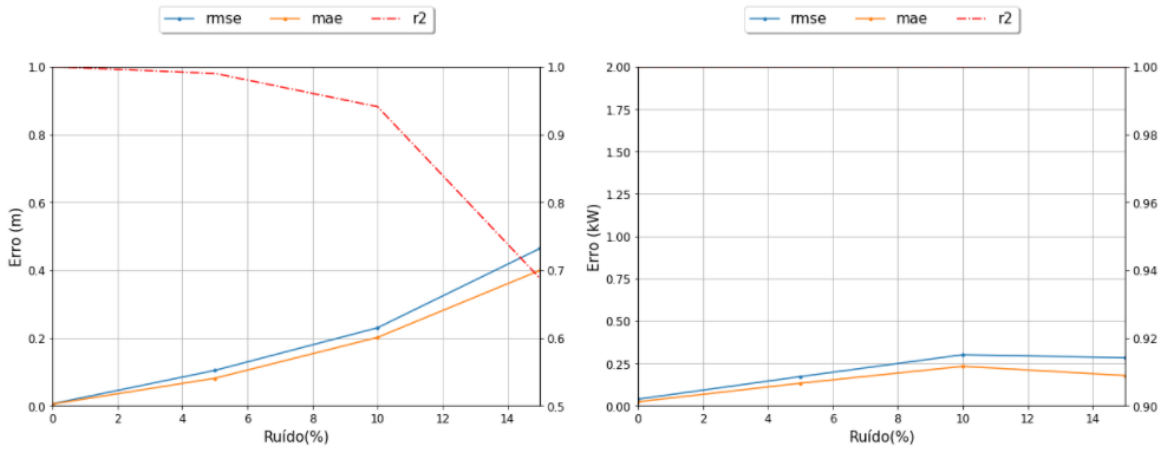


Figura 15 - Variação do RMSE, MAE e R^2 com o aumento do ruído – Modelo 1 ANN.

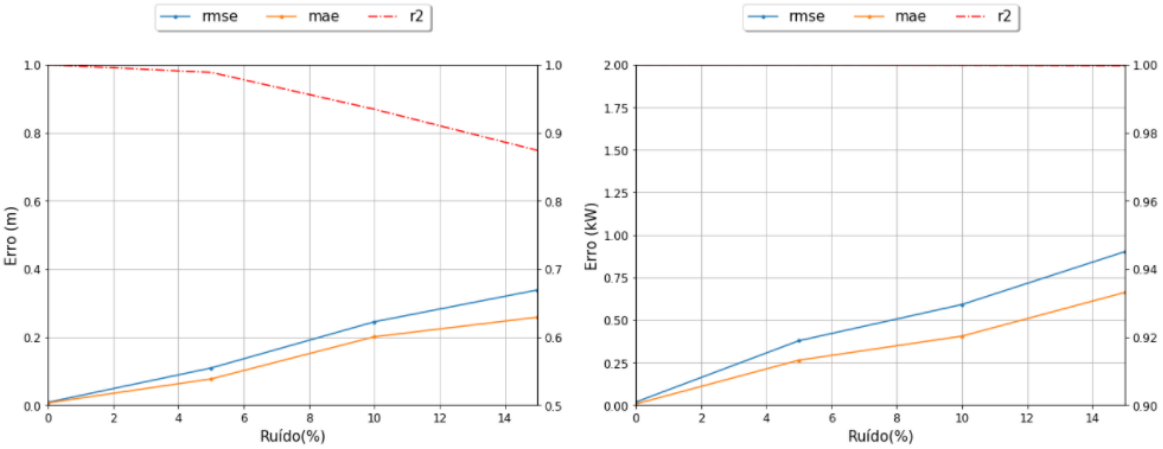


Figura 16 - Variação do RMSE, MAE e R^2 com o aumento do ruído – Modelo 1 XGBoost.

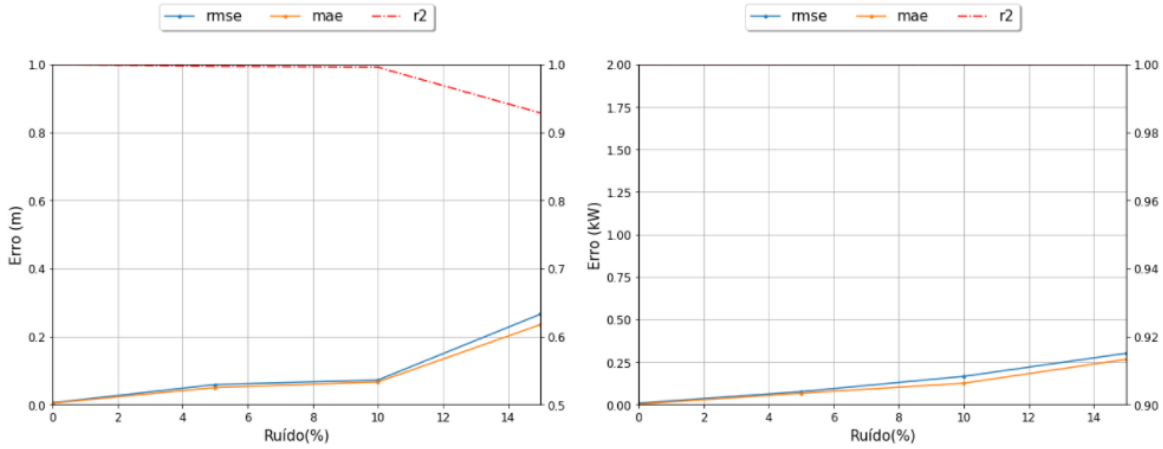


Figura 17 - Variação do RMSE, MAE e R^2 com o aumento do ruído – Modelo 2 ANN.

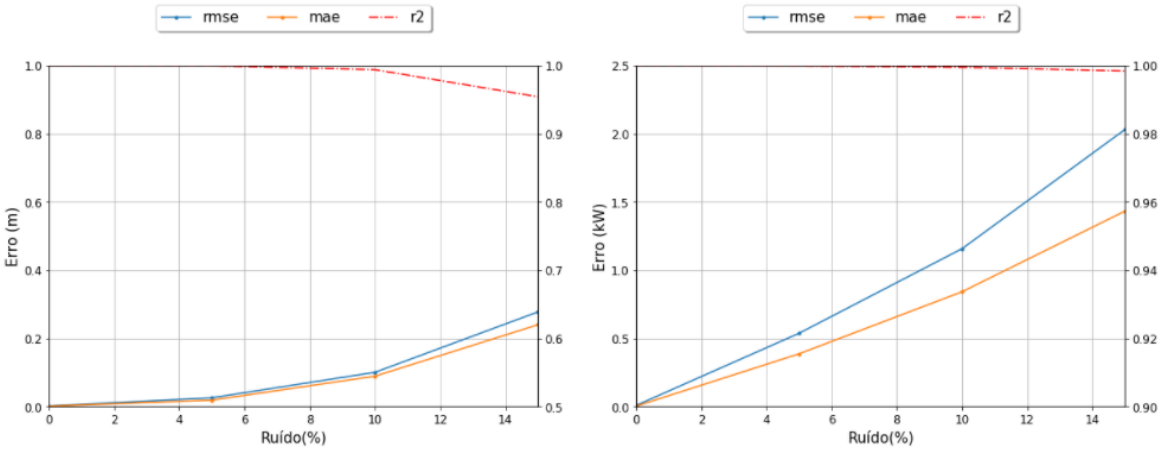


Figura 18 - Variação do RMSE, MAE e R^2 com o aumento do ruído – Modelo 2 XGBoost.

4.1.5 Análise à variação do número de amostras

As Figuras 20, 16, 21 e 22 mostram a variação do *RMSE*, *MAE* e R^2 com a diminuição do número de amostras. As amostras utilizadas em cada teste foram obtidas aleatoriamente. De modo a evitar que testes com menos amostras apresentassem casos favoráveis e fossem beneficiados pela divisão, o processo foi repetido 5 vezes. Nos testes foram utilizados 50000, 25000, 10000, 5000, 2500 e 1000 amostras. Os resultados foram obtidos com o mesmo dia de teste utilizado anteriormente. Tal como observado na adição de ruído, a diminuição do número de amostras não apresentou grandes diferenças na previsão da potência da bomba, com todos os modelos a apresentar RMSE inferiores a 2 kW. Em relação ao nível do depósito, apesar de ser possível observar um decréscimo de precisão com 1000 amostras, todos os modelos obtiveram R^2 superior a 0.95 e RMSE inferiores a 25 cm.

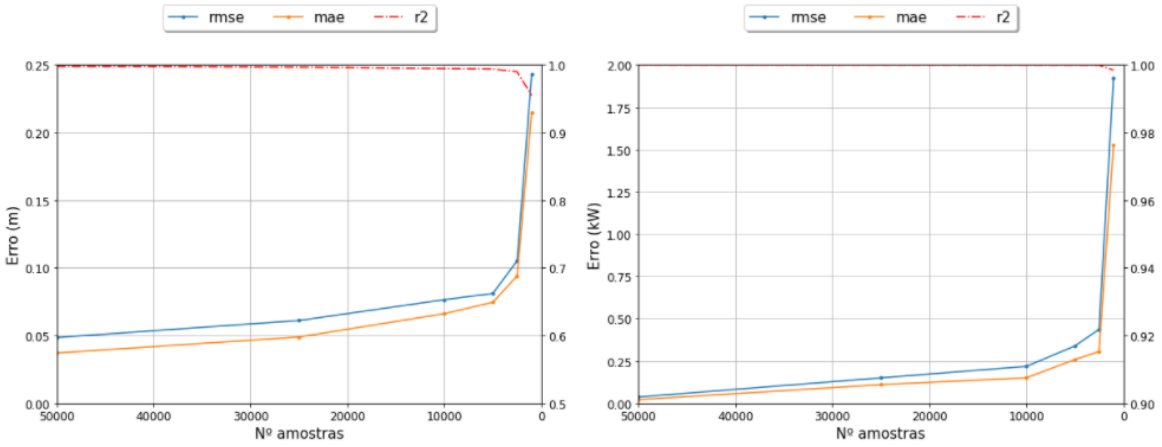


Figura 20 - Variação do RMSE, MAE e R^2 com a diminuição das amostras – Modelo 1 ANN.

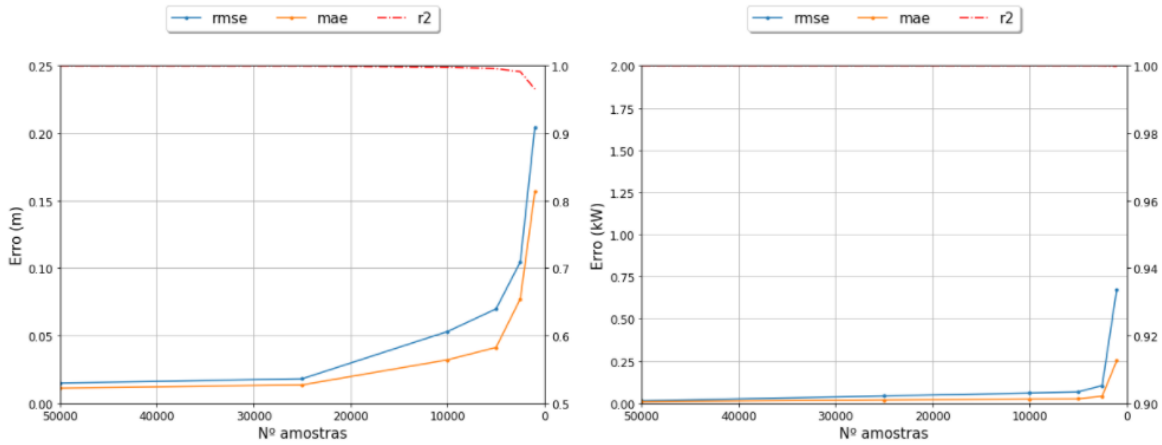


Figura 19 - Variação do RMSE, MAE e R^2 com a diminuição das amostras – Modelo 1 XGBoost.

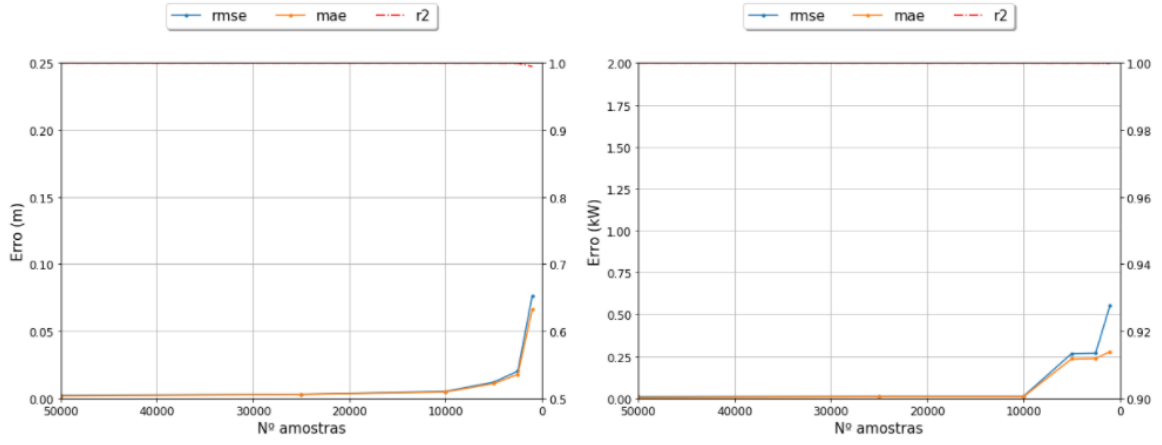


Figura 21 - Variação do RMSE, MAE e R^2 com a diminuição das amostras – Modelo 1 XGBoost.

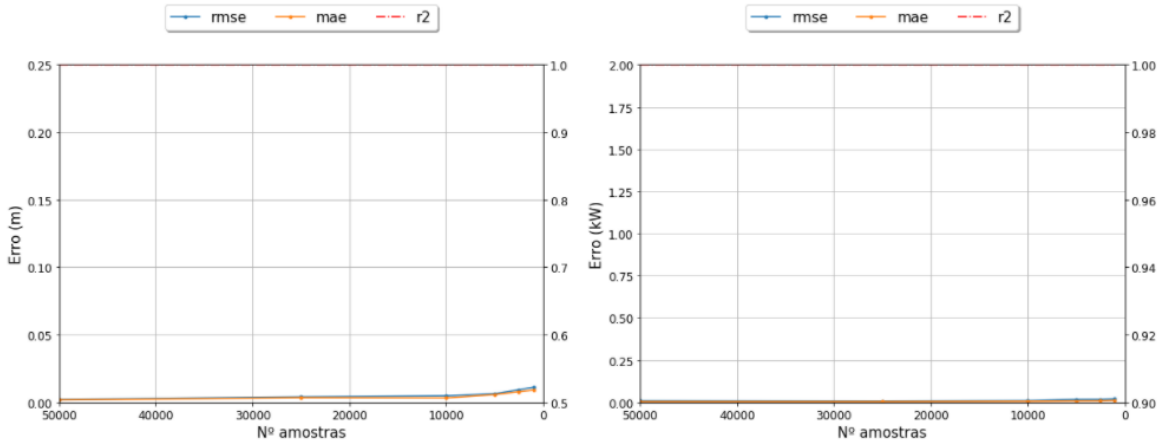


Figura 22 - Variação do RMSE, MAE e R^2 com a diminuição das amostras – Modelo 2 XGBoost.

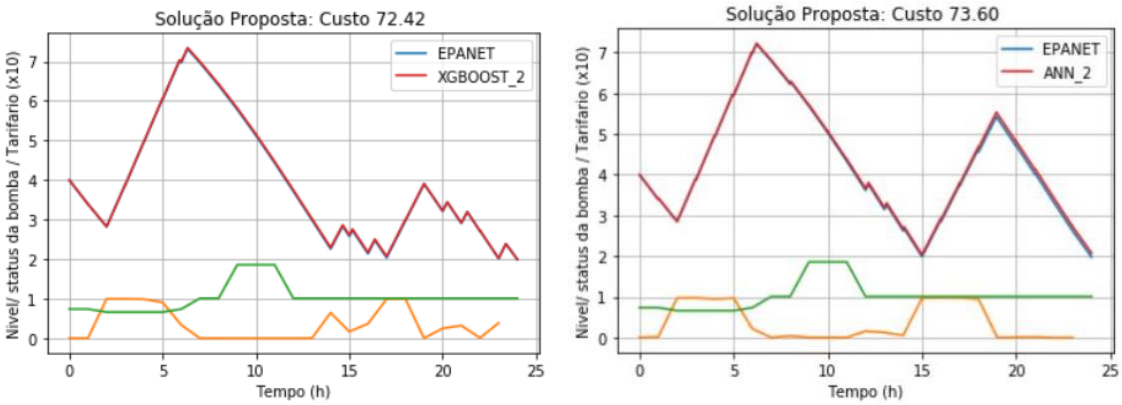
4.1.6 Otimização

Finalizado o processo de modulação dos modelos, procedeu-se à sua otimização. Tal como referido anteriormente, foi necessário fornecer como restrição o nível máximo (7 m) e mínimo (2 m) do depósito ao algoritmo. Como é possível verificar na Tabela 6, a tarifa era composta por 6 intervalos temporais diferentes. Esta apresentava um custo mais reduzido entre as 0 e as 7 horas, tendo valores consideravelmente mais elevados durante as 9 e as 12h. Devido ao valor das bombas no input apenas apresentarem os valores de 0 (bomba desligada) ou 1 (bomba ligada) e o algoritmo de otimização variar este valor entre 0 e 1, foi necessário calcular 2 valores. Por exemplo, caso o valor da bomba num determinado momento fosse (0.23) era considerado que o bomba estava inicialmente ligada 23% do tempo e desligada o restante tempo (77%). Os valores do consumo foram retirados da curva obtida em [44].

A Figura 23 mostra os resultados obtidos para os modelos diferenciais. Para o primeiro modelo não foi possível obter resultados. Tal como explicado anteriormente, o XGBoost é baseado em árvores de decisão e por isso não apresenta um valor de saída contínuo. Tendo em conta que o algoritmo de otimização utiliza o método das diferenças finitas e utilizando o tempo como variável de entrada, torna-se impossível ao algoritmo encontrar os valores ótimos das bombas (o valor nunca muda do valor inicial). Em relação às ANN também não foi possível obter resultados aceitáveis. É possível observar uma simulação quase perfeita dos dois modelos diferenciais em relação aos valores obtidos com o EPANET. Apesar da ANN ter sido capaz de otimizar o sistema, foi necessário alterar os valores iniciais da bomba. As bombas foram colocadas a 1 nos intervalos de tempo onde o tarifário era mais barato e a 0 nos outros intervalos. Para o XGBoost não foi necessário, obtendo o gráfico com valores de 0.5 nas bombas em todos os intervalos temporais.

Intervalo [h]	Período horário	Custo [€/kWh]
[0,2[	Vazio	0,0737
[2,6[	Super Vazio	0,06618
[6,7[	Vazio	0,0737
[7,9[	Cheia	0,100094
[9,12[	Ponta	0,18581
[12,24[	Cheia	0,100094

Tabela 6 – Tarifário energético ao longo do dia.



(a) (b)

Figura 23 – Otimização da operação do sistema da Fontinha com (a) modelo 2 ANN e (b) modelo 2 XGBoost.

4.1.7 Conclusão

Todos os modelos conseguiram reproduzir perfeitamente o comportamento do sistema.

Os modelos foram capazes de prever a potência da bomba mesmo com níveis de ruído nos 15%, porém para os níveis de depósitos houve alguma perda de precisão. Esta maior quebra é expectável, pois os níveis dos depósitos são duplamente prejudicados (no input e no output). Para além disso, como no dia de teste o nível do depósito depende dos níveis anteriores e o erro acumula ao longo do tempo, pequenas alterações podem levar a grandes diferenças no final do ciclo.

Por se tratar de uma rede simples, foi possível observar um excelente comportamento dos modelos mesmo com uma quantidade de dados bastante reduzida.

Na otimização foi comprovado que os modelos diferenciais são capazes de otimizar o sistema, apresentando resultados idênticos aos obtidos pelo EPANET. Contudo, o modelo 1 não foi capaz de apresentar resultados com nenhum dos algoritmos utilizando um algoritmo de otimização com restrições baseado no gradiente e por isso não é aconselhada a sua utilização em otimização de sistemas com este algoritmo de otimização.

4.2 Caso de estudo 2: rede de Richmond

A rede esqueletizada de Richmond (Figura 24) representa um modelo simplificado da rede real de Richmond, sendo esta um subsistema da rede de distribuição de água de Yorkshire, Reino Unido. Foi inicialmente analisada por Jakobus van Zyl [44], tornando-se um dos casos mais analisados e documentados na literatura e servindo como marca de referência na otimização operacional de sistemas de abastecimento de água. A rede de Richmond é composta por 6 depósitos (localizados entre os 187 e os 260 metros de altura), 7 bombas, 1 reservatório e 41 nós (10 nós de consumo). Os níveis dos depósitos iniciais correspondem a 95% da sua capacidade e a simulação começa às 7 horas da manhã.

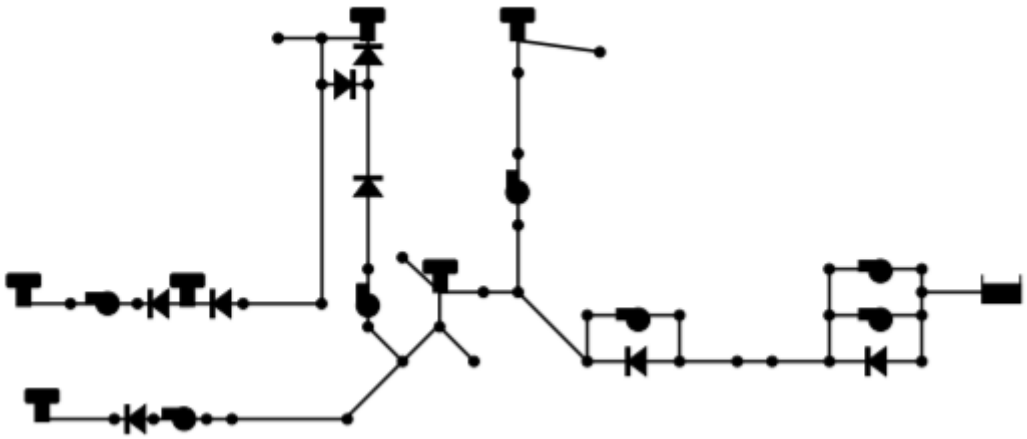


Figura 24 – Esquema simplificado da rede de Richmond [46].

4.2.1 Obtenção de dados

De modo a obter os dados necessários para o treino dos modelos, foram simulados 1500 dias da rede de Richmond com o software EPANET. Esta rede de Richmond pode ser encontrada online [44]. Como explicado anteriormente, de modo a garantir um modelo mais robusto, foi alterado aleatoriamente os estados iniciais das bombas, os padrões de consumo e o nível inicial de cada um dos depósitos. Os dados obtidos representavam variações temporais de cada estado do sistema, tendo como intervalos temporais máximos de 1 hora.

Para este caso de estudo, foram obtidas 24 variáveis de entrada:

- $[B_t^1, \dots, B_t^7]$- Estado das 7 bombas;
- $[D_t^1, \dots, D_t^6]$ - Nível inicial 6 depósitos (m);
- $[C_t^1, \dots, C_t^{10}]$ - 10 pontos de consumo (m^3/h);
- Δt - Variação temporal (h);

e 7 variáveis de saída para cada modelo. Para o modelo 1 estas são:

- E - Energia total das bombas (kWh);
- $[D^1_{t+\Delta t}, \dots, D^6_{t+\Delta t}]$ - Nível final dos 6 depósitos (m)

e para o modelo 2:

- P - Potência total das bombas (kW);
- $[\Delta D^1, \dots, \Delta D^6]$ - Variação (velocidade) do nível dos 6 depósitos (m/h).

4.2.2 Criação e treino dos modelos

Tal como no caso de estudo anterior, foi utilizado o *GridSearchCV* para encontrar os hiperparâmetros ideais de cada um dos modelos.

Para a escolha do número de neurónios de cada camada oculta, foi utilizado o mesmo procedimento, escolhendo apenas múltiplos do número de neurónios da camada de entrada (24). A Tabela 7 mostra os diferentes valores utilizados para cada hiperparâmetro e Figura 26 os valores dos erros para cada hiperparâmetro. Para os modelos com as ANN (Tabela 8), foi utilizada a mesma função de ativação (*ReLu*), a mesma função de otimização do treino (Adam) e o mesmo número de camadas ocultas. Apesar de se tratar de uma rede maior e mais complexa que a anterior, não foi observada uma grande diferença no número de neurónios nas camadas ocultas (72 e 24 neurónios). O *learning rate* (0.00025) e o *batch size* (25) foram os mesmo aos utilizados no caso de estudo anterior. Na Figura 27 é possível observar a variação do erro (*MAE*) treino/teste das ANN ao longo das *epochs*.

Em relação ao XGBoost, foi utilizada a mesma gama de valores do caso de estudo anterior (Tabela 7). Os resultados obtidos para cada hiperparâmetro encontram – se na Figura 25. Foi utilizada a mesma taxa de aprendizagem de 0.05 e um número mínimo de amostras para separação dos nós das árvores de 5. Ao contrário do caso de estudo anterior, foi necessário utilizar um maior número de árvores (1000), mas uma menor profundidade máxima (10) (Tabela 9).

Hiper-paramêtros	**Valores**
Nº camadas ocultas	1, 2, 3
Nº Neurónios/ camada	[24,48,72,96,120]
Batch_size	[25, 50, 75, 100]
Learning_rate	[0.0001, 0.00025, 0.0005, 0.00065]
Função de ativação	ReLu
Algoritmo de otimização	Adam
XGBoost	
Hiper-paramêtros	**Valores**
learning_rate	[0.05, 0.1, 0.15, 0.2]
min_child_weight	[1, 3, 5]
max_depth	[5, 10, 15, 20]
n_estimators	[100, 250, 500, 750, 1000]

Tabela 7 - Hiperparâmetros ANN e XGBoost - Richmond.

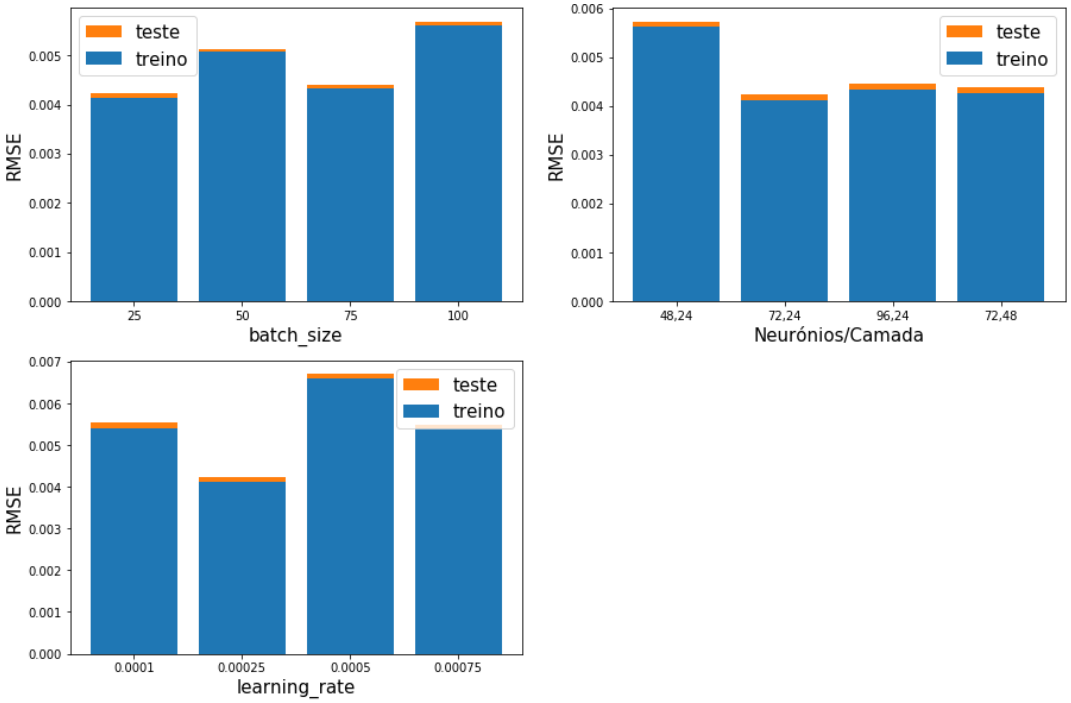


Figura 26 - Erro treino/teste dos diferentes hiperparâmetros ANN - Richmond.

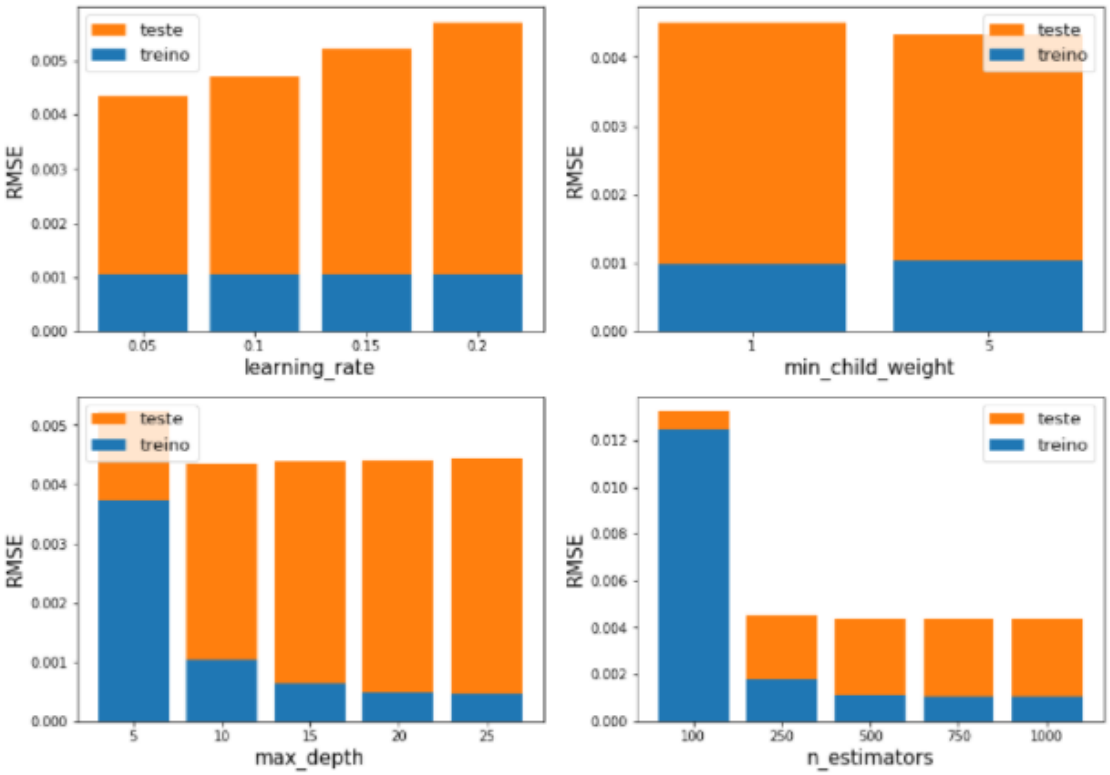


Figura 25 - Erro treino/teste dos diferentes hiperparâmetros XGBoost - Richmond.

Nº camadas ocultas	Número de neurónios/camada	batch_size	learning_rate	Função de Ativação	Algoritmo de Otimização
2	72,24	25	0.00025	ReLu	Adam

Tabela 8 – Configuração ANN – Richmond.

learning_rate	min_child_weight	max_depth	n_estimators
0.05	5	10	1000

Tabela 9 – Configuração XGBoost – Richmond.

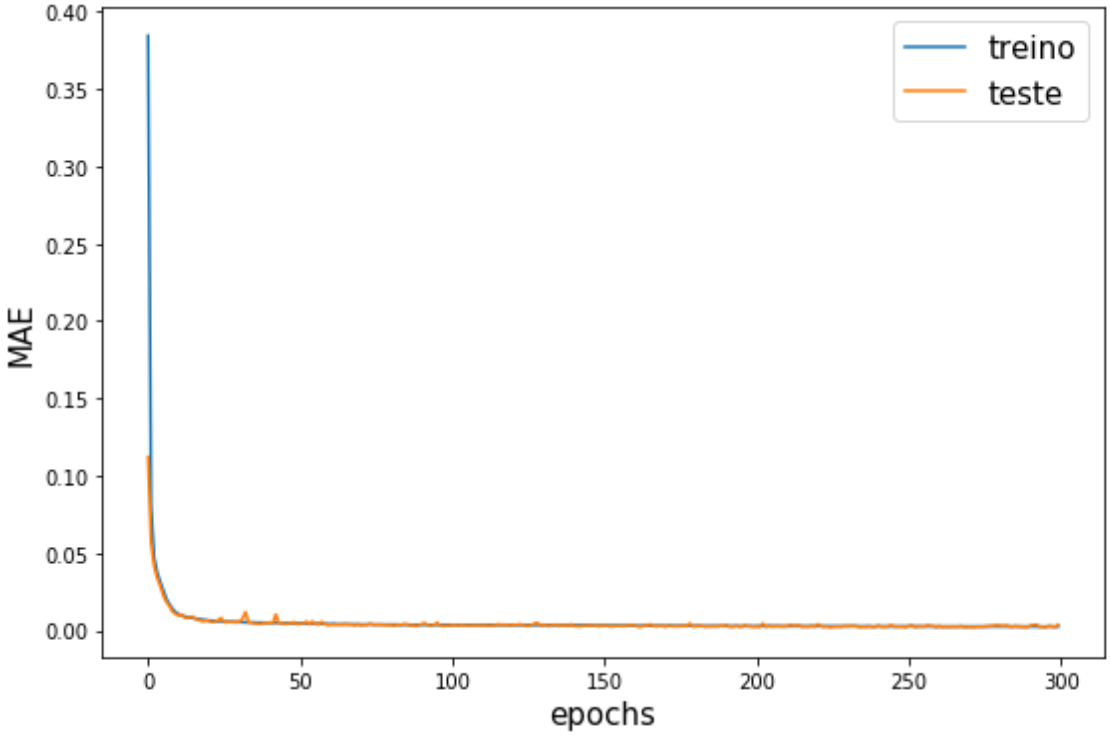


Figura 27 - Curva da função de perda ANN - Fontinha.

4.2.3 Validação do modelo

A Figura 28 demonstra o comportamento das bombas e do padrão de consumo no dia de teste utilizado (composto por 59 intervalos temporais). Apesar de haver 10 pontos de consumo diferentes, todos apresentam o mesmo padrão de consumo, apenas variando o seu volume. Embora a rede de Richmond apresente mais depósitos, estes são todos consideravelmente mais pequenos do que o depósito da Fontinha (o maior depósito tem um nível máximo de 3.6 m e menor tem nível máximo de 1.9 m).

Nas Figuras 29 e 30 encontram-se os valores obtidos para os dois modelos. É possível observar diferenças significativas nos valores dos depósitos 1 e 3 no modelo 1 ANN. Apesar dessa diferença, todos os modelos mostraram-se bastante competentes na simulação de um sistema hidráulico, particularmente o modelo XGBoost diferencial. Na Tabela 10 estão representados os erros de cada um dos modelos. Os modelos apresentaram excelentes resultados na previsão da potência das bombas, com o R^2 praticamente igual a 1. Em relação ao nível do depósito, os modelos são menos precisos, contudo apresentam todos RMSE inferiores a 7 cm.

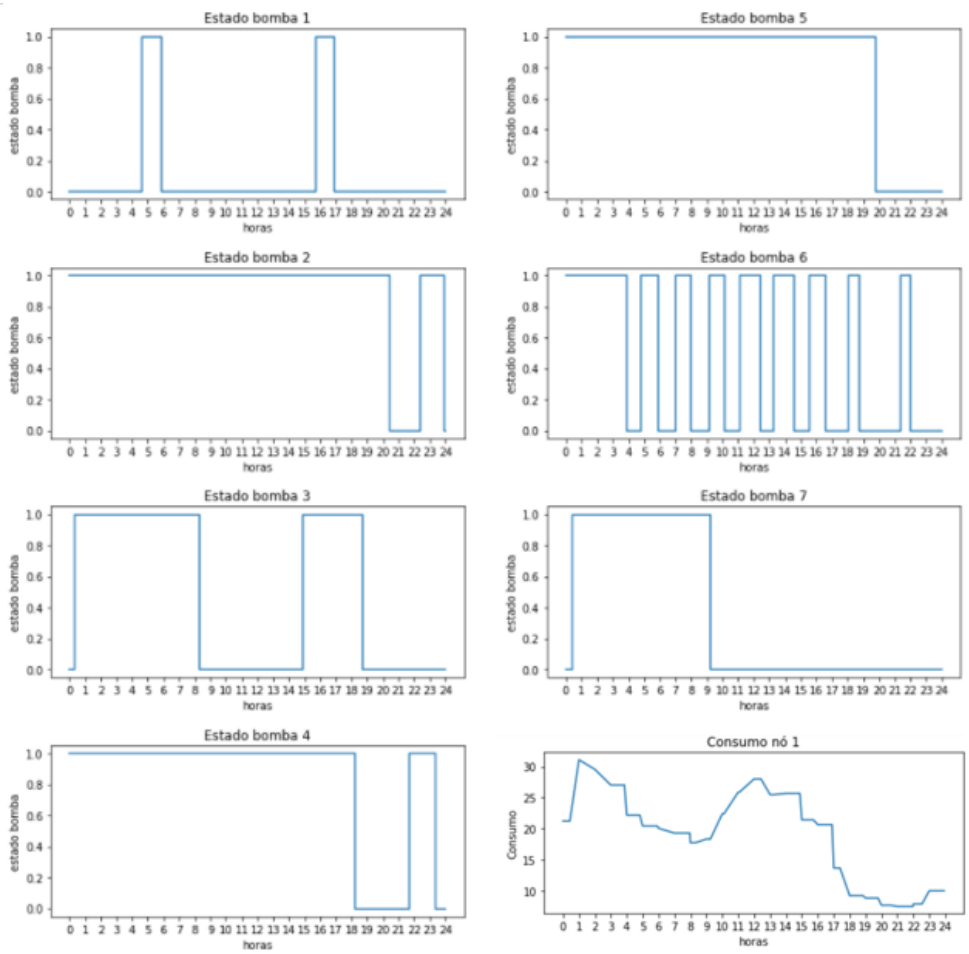


Figura 28 - Comportamento dos inputs do modelo no dia de teste da rede de Richmond.

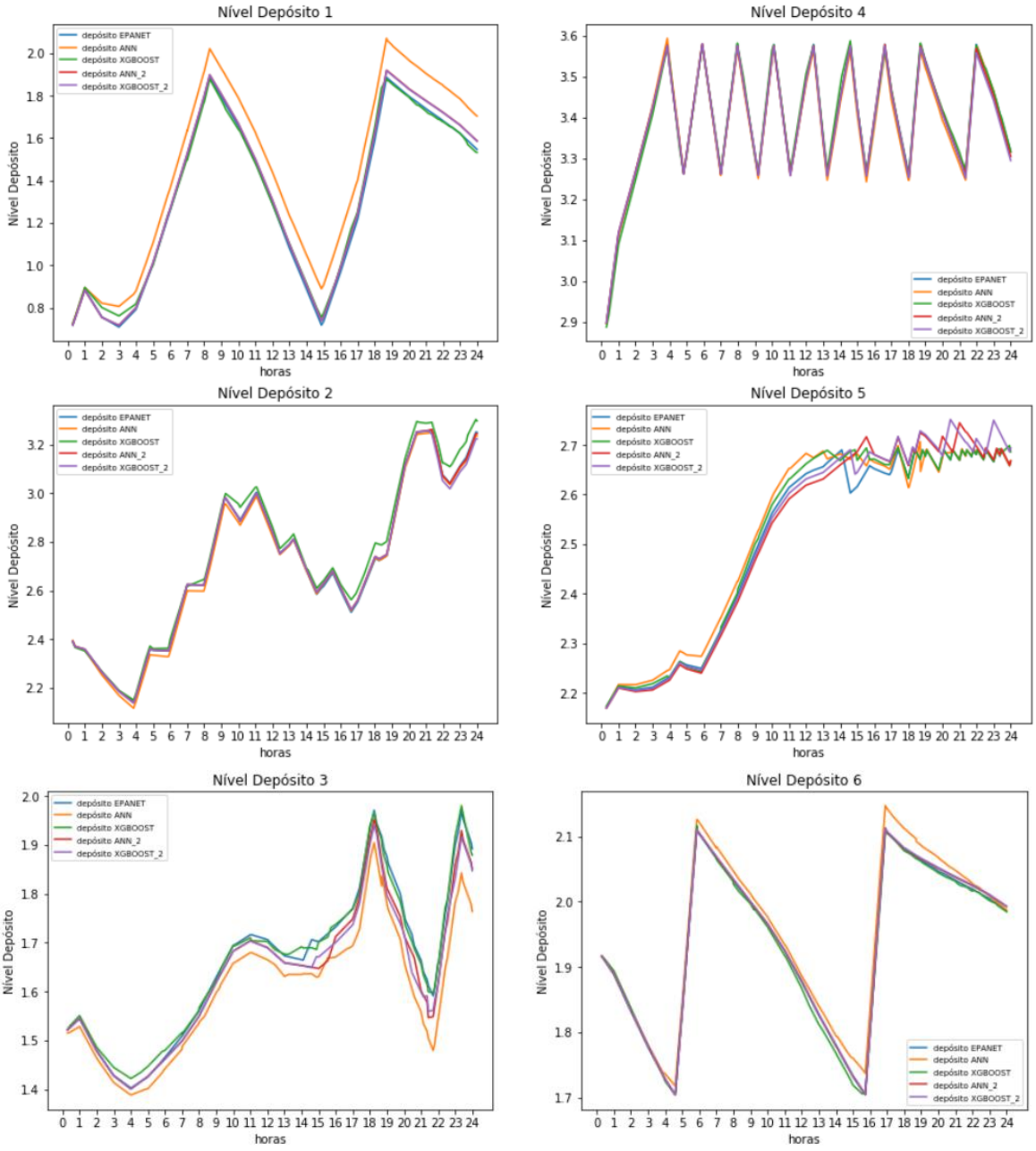


Figura 29 – Simulação depósitos para o dia de teste – Richmond.

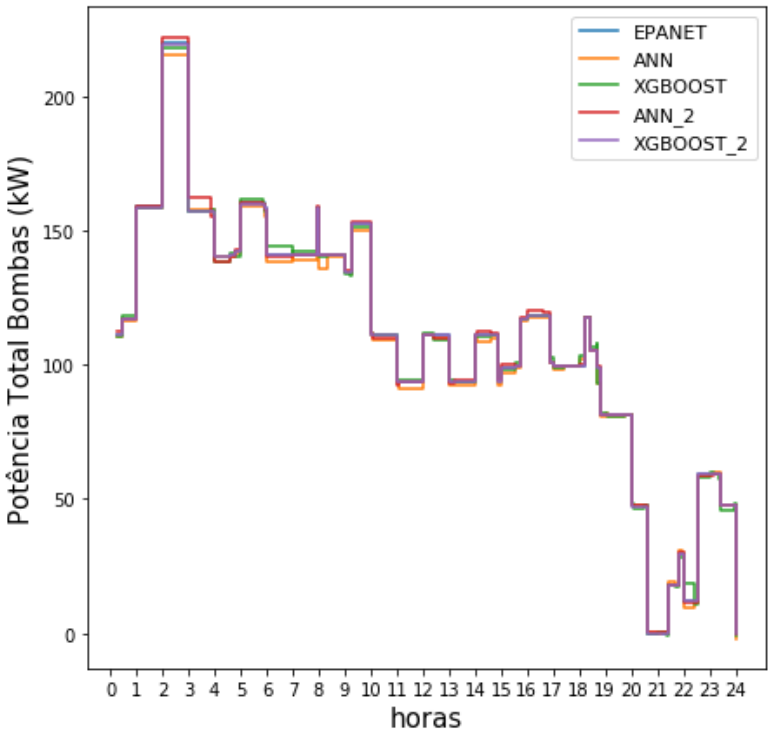


Figura 30 – Simulação da potência para o dia de teste – Richmond.

		ANN	XGBoost	ANN2	XGBoost2
Nível Depósitos (m)	**RMSE**	0.0634	0.0217	0.0312	0.0163
	MAE	0.0473	0.0158	0.0238	0.0117
	R^2	0.9356	0.9858	0.9765	0.9970
Potência bombas (kW)	**RMSE**	2.9486	1.6474	0.9747	0.4683
	MAE	1.5795	0.9118	0.4254	0.1892
	R^2	0.9873	0.9921	0.9921	0.9948

Tabela 10 - Resultados obtidos para cada modelo para o caso de estudo da rede de Richmond. Comparação com os resultados simulados pelo EPANET.

4.2.4 Análise ao ruido

Tal como explicado no caso de estudo anterior, foi colocado ruído de 5, 10 e 15% nos depósitos, níveis de consumo e potências das bombas.

As Figuras 31, 32, 33 e 34 mostram a variação do *RMSE*, *MAE* e R^2 com o aumento do ruído para cada um dos modelos. Tal como na rede da Fontinha, mesmo com níveis de ruído de 15% os modelos não perdem precisão mantendo um R^2 superior a 0.99 e apresentando RMSE inferiores a 2 kW na previsão da potência total das bombas. Em relação aos níveis dos depósitos, os modelos com o *XGBoost* não foram muito afetados com o aumento do ruído. Porém, os modelos com ANN sofreram uma grande diminuição na precisão com ruídos de 15%, apresentando RMSE inferiores a 0.85.

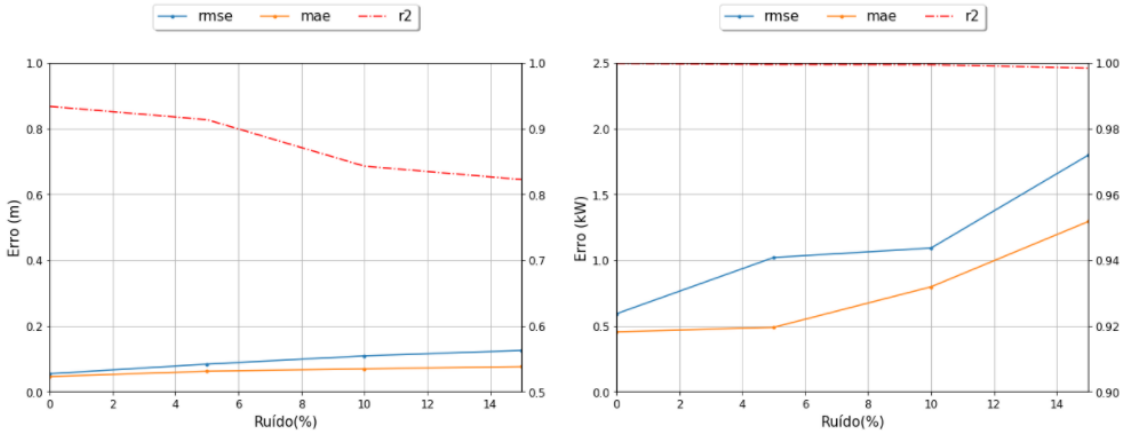


Figura 31 - Variação do RMSE, MAE e R^2 com o aumento do ruído – Modelo 1 ANN Richmond.

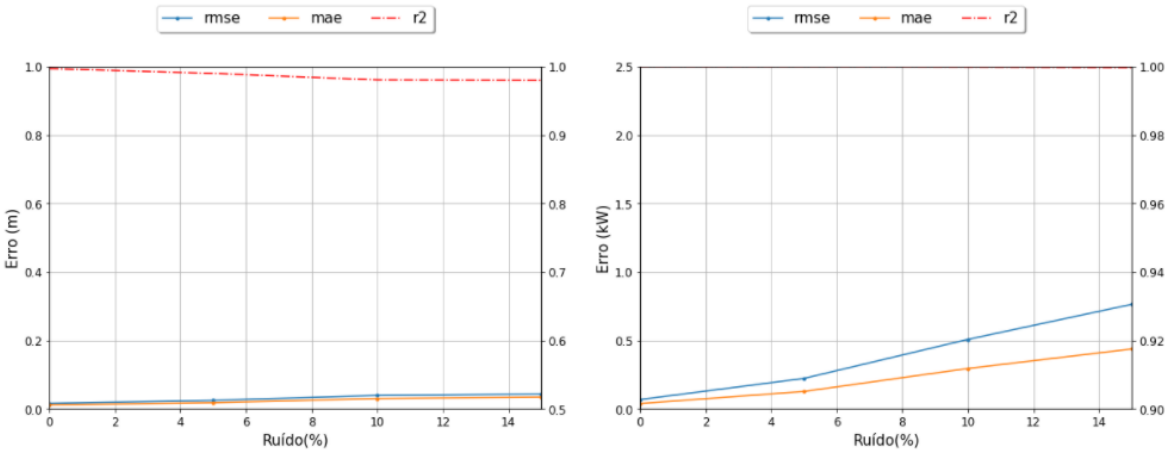


Figura 32 - Variação do RMSE, MAE e R^2 com o aumento do ruído – Modelo 1 XGBoost Richmond.

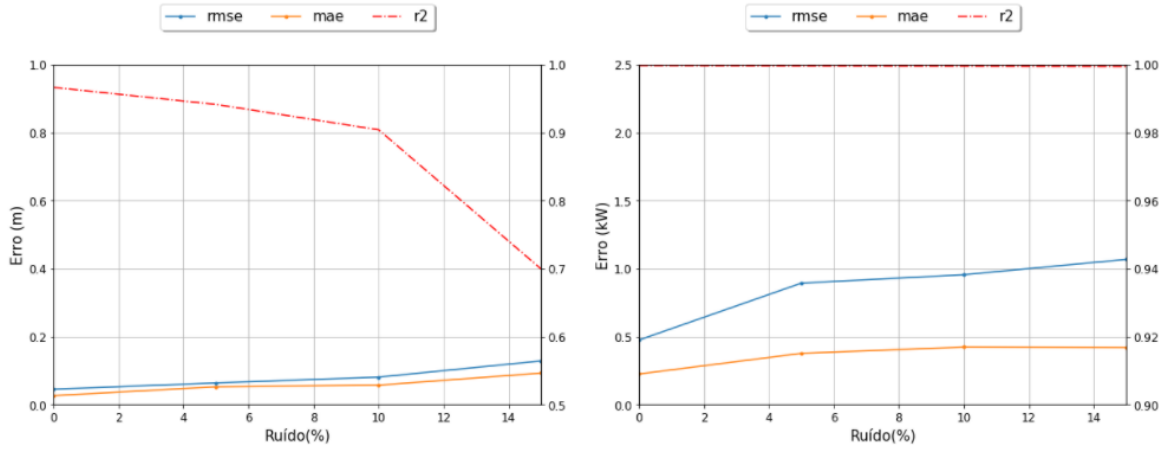


Figura 33 - Variação do RMSE, MAE e R^2 com o aumento do ruído – Modelo 2 ANN Richmond.

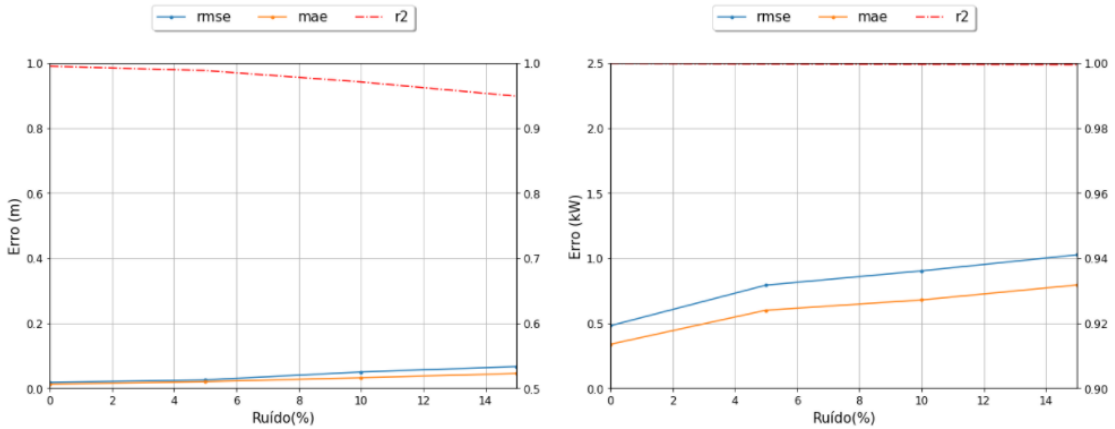


Figura 34 - Variação do RMSE, MAE e R^2 com o aumento do ruído – Modelo 2 XGBoost Richmond.

4.2.5 Análise à variação de amostras

As Figuras 35, 36, 37 e 38 mostram a variação do RMSE, MAE e R^2 com a diminuição do número de amostras. Devido à maior complexidade e tamanho desta rede, o número de amostras mínima utilizadas foi de 2500. Em comparação com a sub-rede da Fontinha, nesta rede foram necessárias mais amostras para o modelo conseguir simular o comportamento dos depósitos. A rede apresenta uma grande perda de precisão a partir das 5000 amostras, principalmente nos modelos com ANN. Na simulação da potência das bombas o sistema mantém o R^2 superior a 0.99 e RMSE inferiores a 3 kW mesmo com apenas 2500 amostras.

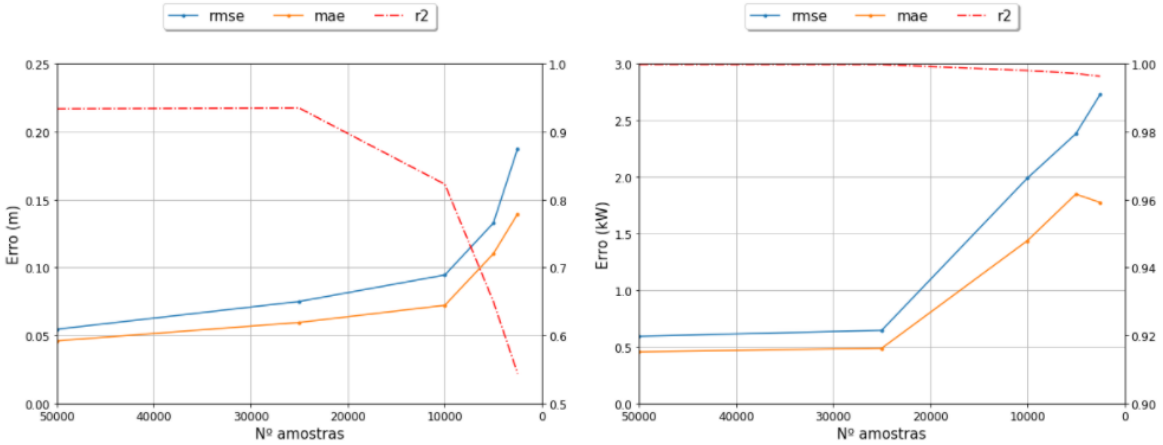


Figura 35 - Variação do RMSE, MAE e R2 com a diminuição das amostras – Modelo 1 ANN Richmond.

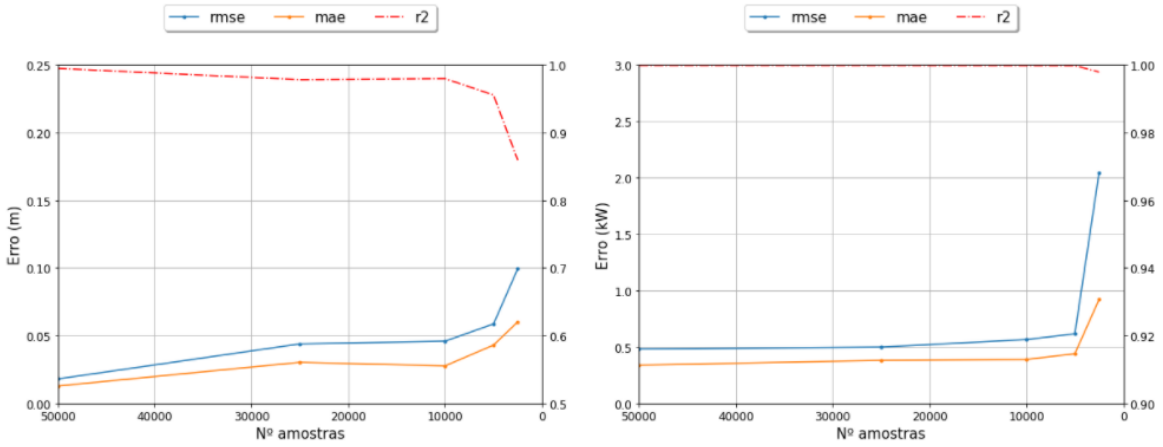


Figura 36 - Variação do RMSE, MAE e R2 com a diminuição das amostras – Modelo 1 XGBoost Richmond.

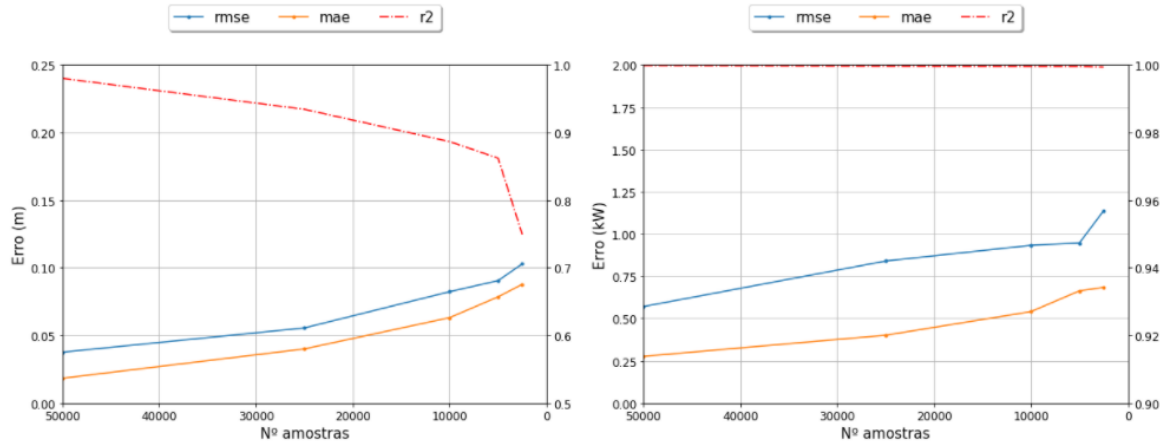


Figura 37 – Variação do RMSE, MAE e R2 com a diminuição das amostras – Modelo 2 ANN Richmond.

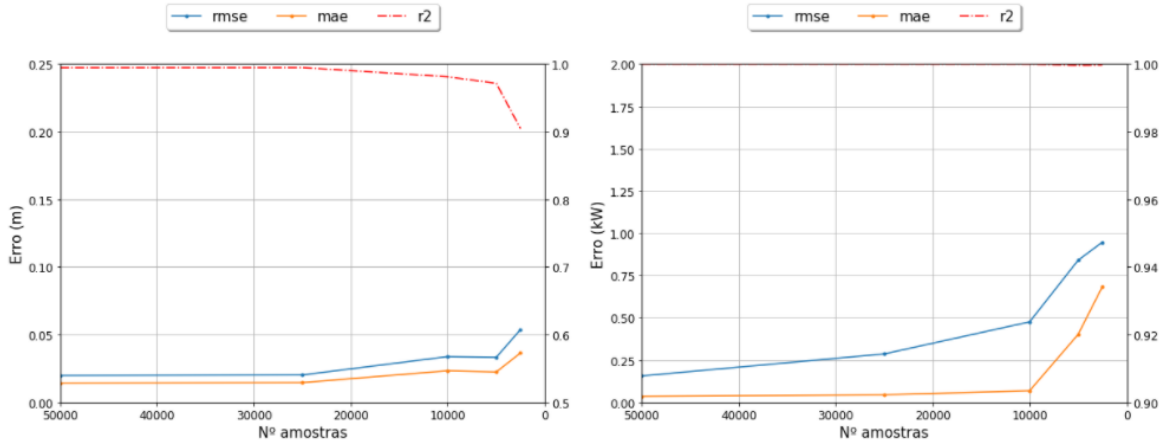


Figura 38 - Variação do RMSE, MAE e R2 com a diminuição das amostras – Modelo 2 XGBoost Richmond.

4.2.6 Conclusão

Este caso de estudo demonstrou que os algoritmos de *Machine Learning* são capazes de reproduzir o comportamento de redes mais complexas e de maiores dimensões. Todos os modelos apresentaram excelentes resultados na previsão das potências totais das bombas, apresentando R^2 superiores a 0.98 e RMSE inferiores a 5 kW. Em relação à previsão dos níveis dos depósitos, O modelo ANN 1 apresentou algumas dificuldades na previsão dos níveis dos depósitos apresentando um R^2 ligeiramente inferior a 0.95 e um RMSE superior a 6 cm. Apesar destes resultados menos favoráveis, todos os outros modelos apresentaram excelentes resultados, conseguindo reproduzir com precisão o comportamento da rede.

Em relação ao ruido, os modelos ANN apresentaram um declínio da precisão na previsão dos níveis dos depósitos com 15% de ruido. Contudo, os modelos mostraram-se mais resistentes ao ruído neste caso de estudo quando comparado aos valores do caso de estudo anterior.

Em relação ao número de amostras necessárias para simular a rede foi observado um aumento significativo quando comparado com a rede da Fontinha. Este aumento era expetável devido à grande diferença no tamanho e complexidade das duas redes. Contudo, os modelos XGBoost apresentaram excelentes resultados mesmo com apenas 5000 amostras. Apesar do excelente resultado dos modelos com um número reduzido de amostras, estes valores foram obtidos com dados sintéticos que não apresentavam ruído nem *outliers*. Assim, em casos reais com ruído, é expetável que o número de amostras necessárias para reproduzir o sistema seja consideravelmente superior.

Apesar de não ter sido efetuado um estudo, pela análise do estudo espera-se que os modelos XGBoost apresentem resultados satisfatórios com apenas 10000 amostras com níveis de ruído de 10%. O mesmo não é esperado para os modelos ANN, que apresentam grandes quedas de precisão mesmo com 50000 amostras e ruído de 10%.

4.3 Caso de estudo 3: rede da Ronqueira

A rede da Ronqueira (Figura 39) é um subsistema de abastecimento de água e fornece água a cerca de 25 mil habitantes nas freguesias de Penacova e Vila Nova de Poiares, Coimbra**.** Este subsistema é constituído por 2 estações de bombeamento, 3 depósitos, 3 pontos de consumo (Aveleira, Alto da Espinheira e Albarqueira). Ao contrário dos casos de estudo anteriores é constituída por uma válvula que está localizada antes do depósito do Alto da Espinheira e abre sempre que o depósito atinge um valor mínimo, voltando a fechar quando o depósito atinge um valor máximo estipulado.

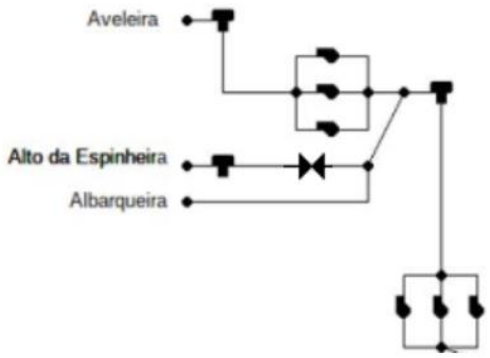


Figura 39 – Esquema do caso de estudo da rede da Ronqueira.

4.3.1 Obtenção dos dados

Ao contrário dos casos de estudo anteriores, os dados deste caso de estudo eram dados reais. Os dados foram fornecidos pela empresa SCUBIC [45]. Um dos aspetos mais importantes de qualquer problema de Machine Learning é ter dados de qualidade. Como explicado anteriormente, é extremamente comum em dados reais estes virem com imprevisões devido a erros na sua obtenção ao à margem de erro em que os sensores trabalham.

Em casos reais, é extremamente comum as empresas responsáveis pelos SAA receberem dados com erros devido a problemas nos sensores ou na aquisição e transferência dos dados. Mesmo que não ocorra nenhum erro, os sensores operam com uma margem de erro, diminuindo a precisão dos dados. Devido a isso, inicialmente foi necessário proceder à limpeza da base de dados. Para a eliminação do ruido nos dados é necessário utilizar técnicas de suavização. Uma das técnicas mais populares de suavização é a suavização exponencial (*Exponential Smoothing)*. Para além de ser uma técnica flexível e fácil de calcular também apresenta bons resultados. A *Exponential Smoothing* calcula a média ponderada de todas as observações, dando um peso superior às novas observações. Esta técnica é dada pela seguinte fórmula:

$$y_{t+1} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \cdots, \text{ com } \ 0 \leq \alpha \leq 1 \qquad (10)$$

onde α é o parâmetro de suavização e *y* o valor das observações.
Esta técnica foi utilizada recorrendo à biblioteca *statsmodel* e foi utilizada nos valores dos depósitos de Espinheira e Albarqueira, sendo utilizado um parâmetro de suavização de 0.5 para o depósito de Espinheira e de 0.8 para o depósito de Albarqueira.
Neste caso de estudo não foram fornecidos os valores da potência das bombas, sendo apenas fornecido o caudal e a curva hidráulica de cada uma delas. Também foi necessário calcular a curva de eficiência de cada uma das bombas. As curvas da eficiência e hidráulica da bomba da Aveleira são respetivamente:

$$\eta = 0{,}0004Q^3 - 0{,}0741Q^2 + 4{,}1025Q + 3{,}3658, \quad (11)$$

$$h_p[m] = 245.18 - 0.001285Q^{2,76}, \quad (12)$$

com Q em m^3/h. Por sua vez, as curvas da bomba da Albarqueira são as seguintes:

$$\eta = 1E^{-5}Q^3 - 0{,}0075Q^2 + 1{,}301Q + 8{,}7409, \quad (13)$$

$$h_p[m] = 417{,}52 - 0.00242Q^{2,15}, \quad (14)$$

Calculando por fim a potência através da seguinte fórmula:

$$P\ [kW] = \frac{Q\rho h_p g}{3{,}6*10^6\eta}, \quad (15)$$

onde ρ (kg/m^3) é a densidade do fluído, *g* (m/s^2).

Ao contrário dos casos de estudo anteriores, os dados encontravam-se todos com o mesmo intervalo temporal e por isso foi decidido utilizar apenas o modelo diferencial pelas suas vantagens na otimização. Assim, foram utilizadas 9 *features* de entrada para o caso de estudo da Ronqueira:

- $[B_t^1, B_t^2]$ - Estado das duas bombas;
- $[D_t^1, D_t^2, D_t^3]$ Nível inicial do único depósito (m);
- $[C_t^1, C_t^2, C_t^3]$ – 2 Pontos de consumo (m^3/h);

e 4 variáveis de saída:

- $[\Delta D^1, \Delta D^2, \Delta D^3]$ - Variação nível do depósito por hora (m/h);
- *P* – Potência total das bombas (kW).

Em relação à base de dados, esta apresentava valores de 1 ano, divididas em intervalos de 15 minutos. Devido a alterações no comportamento da rede, com um dos depósitos a aumentar a sua capacidade máxima e por isso alterando o seu funcionamento, não foi possível utilizar os 6000 dados iniciais. Para além disso, foram retiradas da base de dados 288 amostras (3 dias) para validação do sistema. Assim, foram utilizadas 30690 amostras para o treino e teste do sistema.

4.3.2 Criação e treino do modelo

Tal como nos casos de estudo anteriores, procedeu-se à utilização do GridSearchCV para encontrar os melhores hiperparâmetros para este caso de estudo.

Tal como nos casos anteriores, foram utilizados múltiplos do número de neurónios da camada de entrada (9) para o número de neurónios das diferentes camadas ocultas. Na Tabela 11 encontra-se a gama de valores utilizada para cada hiperparâmetro testado. A Figura 40 mostra os erros de cada hiperparâmetro. Tal como nos casos de estudo anteriores, para as (ANNs) foi utilizada a função de ativação (*ReLu*) e a função de otimização do treino (Adam). O número de neurónios das duas camadas ocultas (72,27) foi muito semelhante ao utilizado no caso de estudo da rede de Richmond. O learning_*rate* utilizado na função de ativação foi de 0.001 e foi utilizado um batch_size de 25. A Figura 42 mostra a curva da função de perda (utilizando o MAE) no treino/teste das ANN ao longo das *epochs*. Ao contrário dos casos de estudo anteriores, foi necessário um maior número de *epochs* para o erro estabilizar. Para além disso, é possível observar uma oscilação constante do erro nos dados de teste. Esta oscilação poderá estar relacionada com o menor número de amostras utilizadas neste caso de estudo, ou com algum ruído presente nas amostras.

Para os modelos XGBoost foram testados o mesmo conjunto de valores utilizados nos casos de estudo anteriores (Tabela 11). A Figura 41 mostra os resultados obtidos de cada hiperparâmetro. Foi utiliza a mesma taxa de aprendizagem (0.05) e o mesmo número mínimo de amostras para separação dos nós (5) dos casos de estudo anteriores. Para este caso de estudo (Tabela 13), foi necessário utilizar um valor inferior da profundidade máxima de cada árvore (5), pois o aumento conduzia a *overfitting* (erros muitos inferiores no treino quando comparados com o teste), como se pode verificar na Figura 41. Foram utilizadas 1000 árvores.

Hiper-paramêtros	**Valores**
Nº camadas ocultas	1, 2, 3
Nº Neurónios/ camada	[18,27,36, … , 108]
Batch_size	[25, 50, 75, 100]
Learning_rate	[0.0005, 0.00075, 0.001, 0.002]
Função de ativação	ReLu
Algoritmo de otimização	Adam
XGBoost	
Hiper-paramêtros	**Valores**
learning_rate	[0.05, 0.1, 0.15, 0.2]
min_child_weight	[1, 3, 5]
max_depth	[5, 10, 15, 20, 25]
n_estimators	[100, 250, 500, 750, 1000]

Tabela 11 – Hiperparâmetros ANN e XGBoost - Ronqueira.

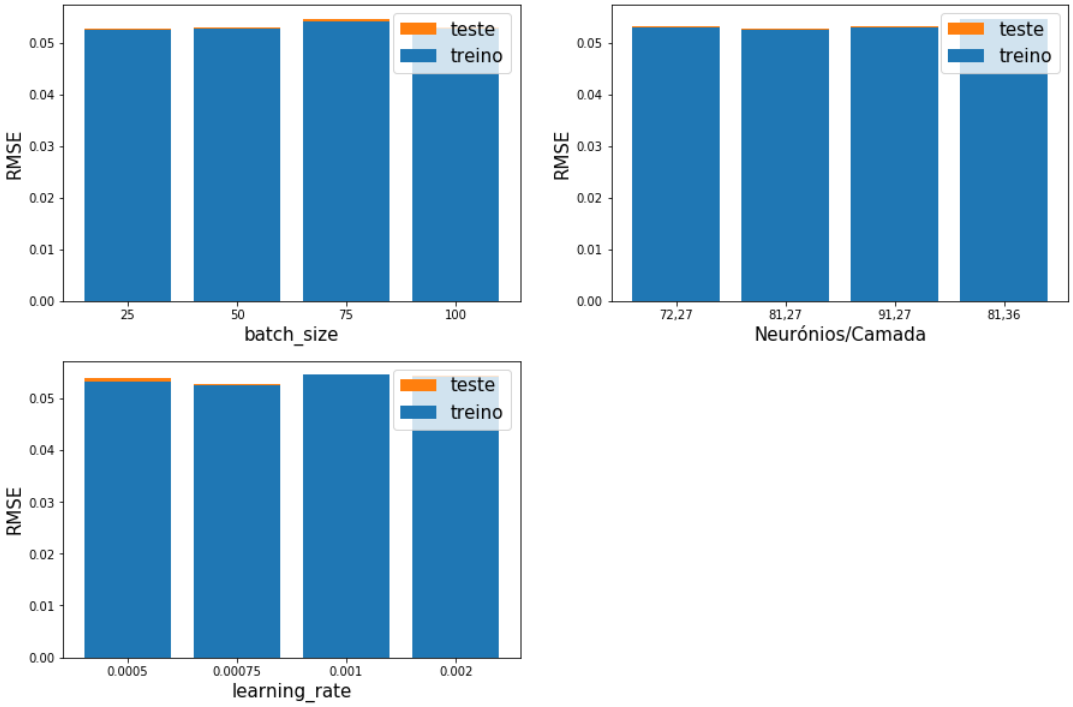


Figura 40 - Erro treino/teste dos diferentes hiperparâmetros ANN - Ronqueira.

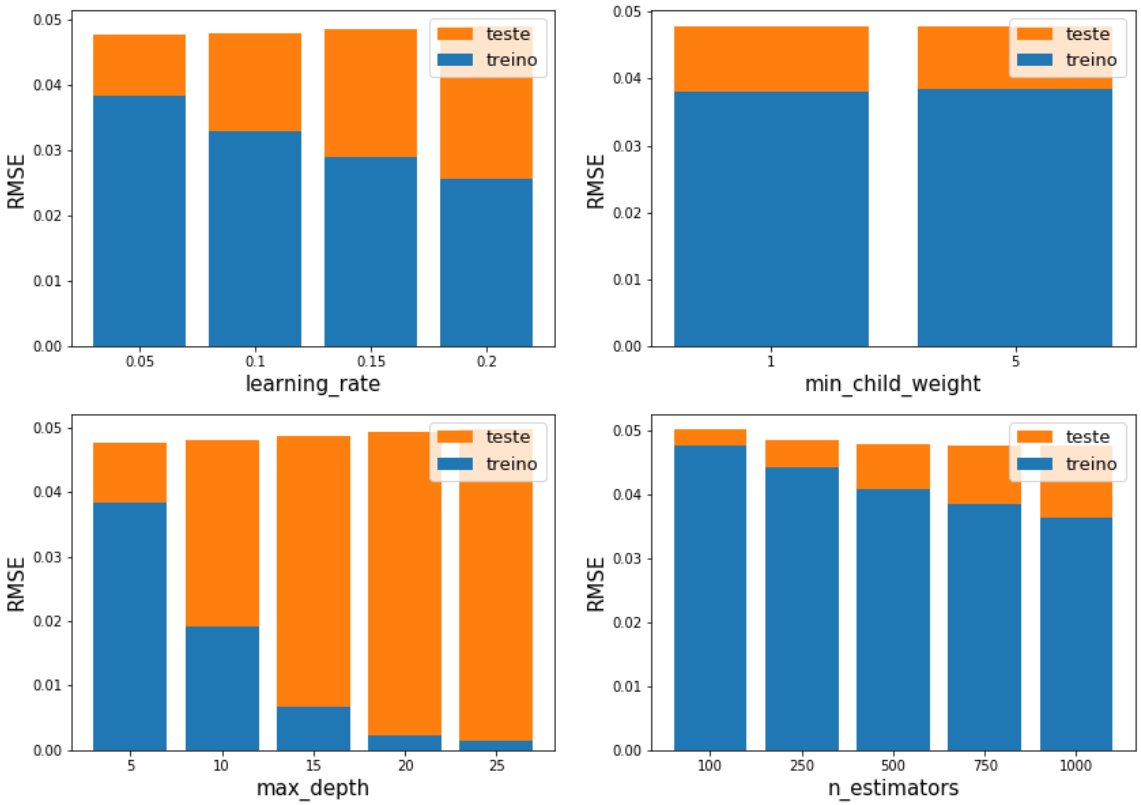


Figura 41 - Erro treino/teste dos diferentes hiperparâmetros XGBoost - Ronqueira.

Nº camadas ocultas	Número de neurónios/camada	Batch_size	learning_rate	Função de Ativação	Função de Otimização
2	72,27	25	0.001	ReLu	Adam

Tabela 12 - Configuração ANN Ronqueira.

learning_rate	min_child_weight	max_depth	n_estimators
0.05	5	5	1000

Tabela 13 – Configuração XGBoost Ronqueira.

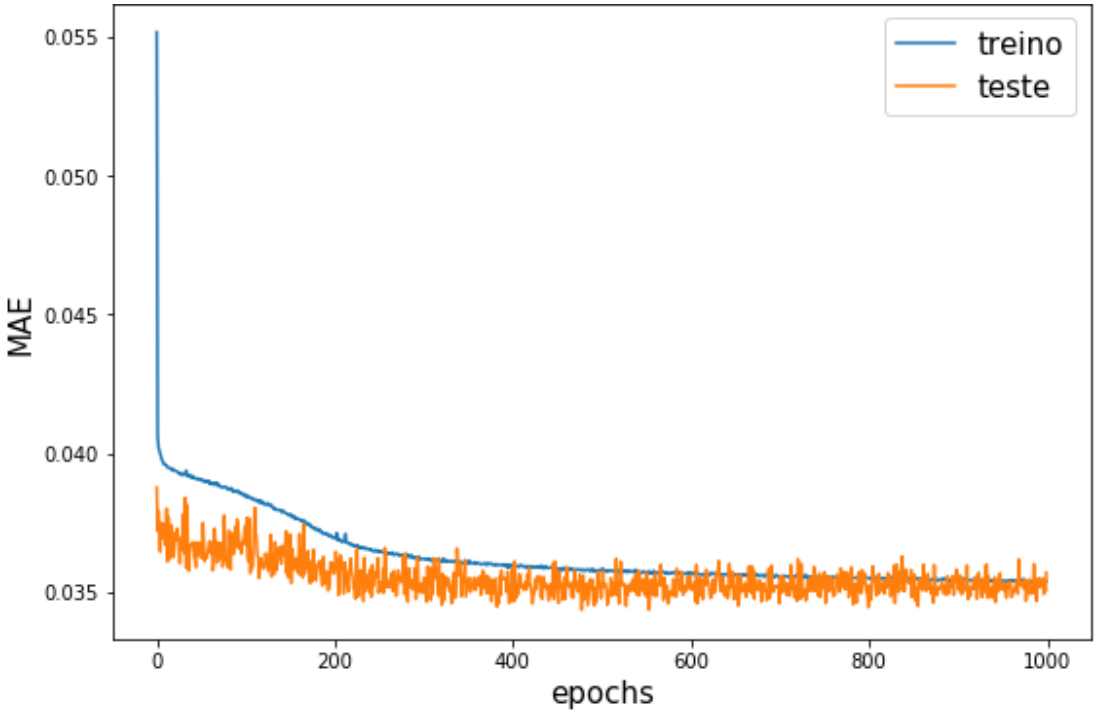


Figura 42 - Curva da função de perda ANN - Ronqueira.

4.3.3 Validação dos modelos

Mais do que apresentarem resultados excelentes num determinado dia, é necessário que os modelos apresentem um resultado consistente durante todo o tempo. Por esse motivo, foram utilizados 5 dias para validar o sistema. Tal como referido anteriormente, 3 dias foram retirados da base de dados (sendo estes os dias de teste 1, 3 e 5) e os outros 2 (dias de teste 2 e 4) foram utilizados no treino dos modelos. Estes dias foram escolhidos aleatoriamente. Porém, para conseguir obter uma maior generalização, foram utilizadas diferentes alturas do ano. Para uma maior comparação, foram utilizados também valores obtidos pelo EPANET fornecidos pela SCUBIC.

As figuras 43, 45, 44, 47 e 46 mostram os valores dos depósitos e potência previstos para os diferentes dias de teste. A Tabela 11 mostra os valores do *RMSE*, *MAE* e R^2 obtidos para cada dia de teste, bem como a média dos seus erros. Apesar dos dois modelos apresentarem erros na previsão da potência das bombas superiores aos observados nos casos de estudo anteriores, estes apresentam resultados bastante positivos, com uma média de R^2 próximo dos 0.97 e um RMSE próximo dos 9 kW. Em relação à previsão dos níveis dos depósitos, é possível observar uma diferença significativa dos dois modelos, com o XGBoost a apresentar melhores resultados. Embora os dois modelos apresentem erros superiores aos apresentados nos casos de estudo anteriores, ambos apresentam bons resultados, com uma média de R^2 superior a 0.85 e um *RMSE* inferior a 10 cm.

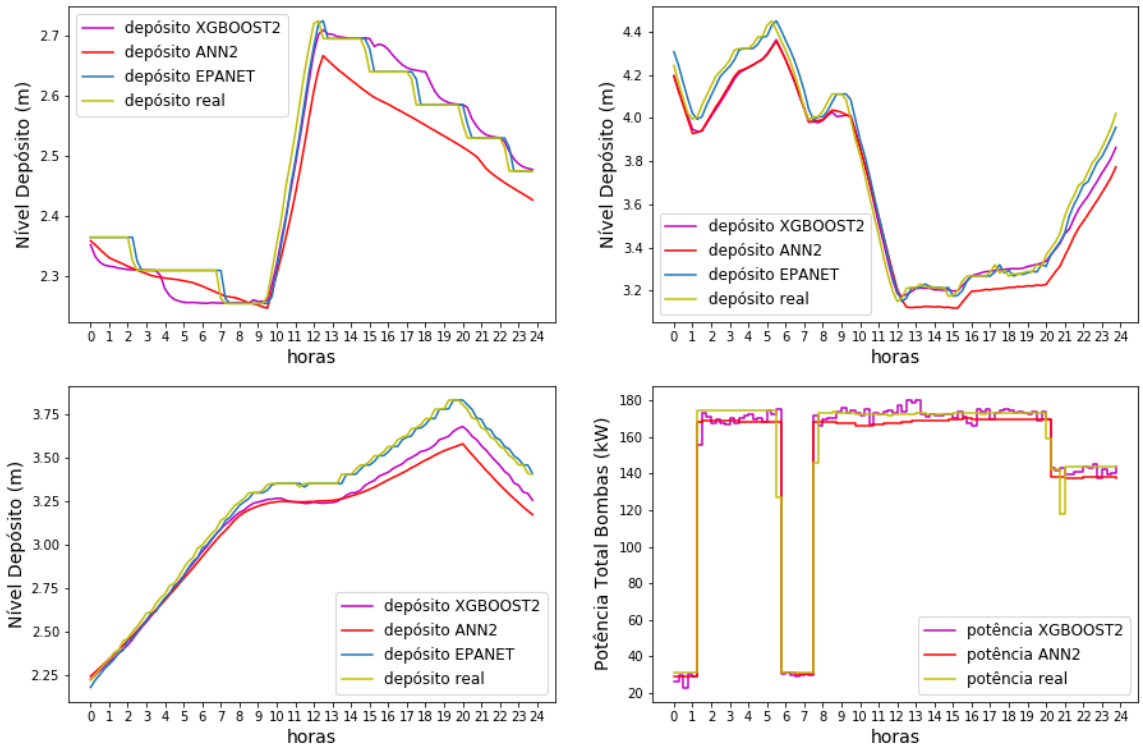


Figura 43 – Simulação dos depósitos e da energia para o dia de teste 1

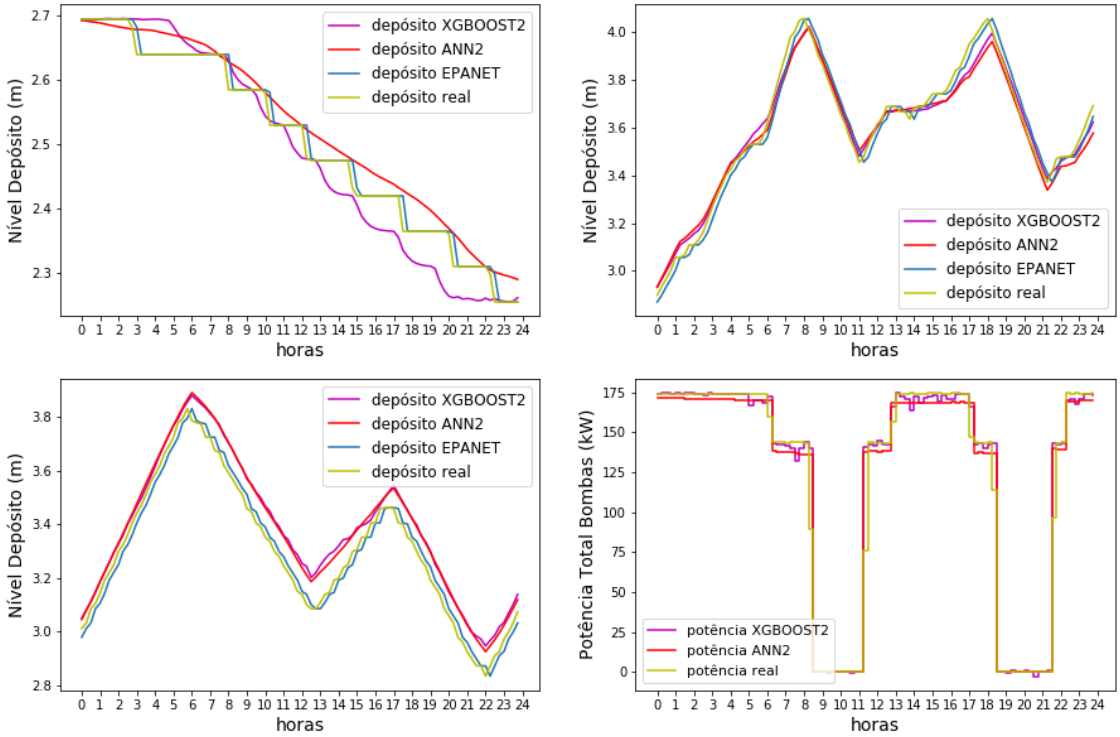


Figura 45 - Simulação dos depósitos e da energia para o dia de teste 2

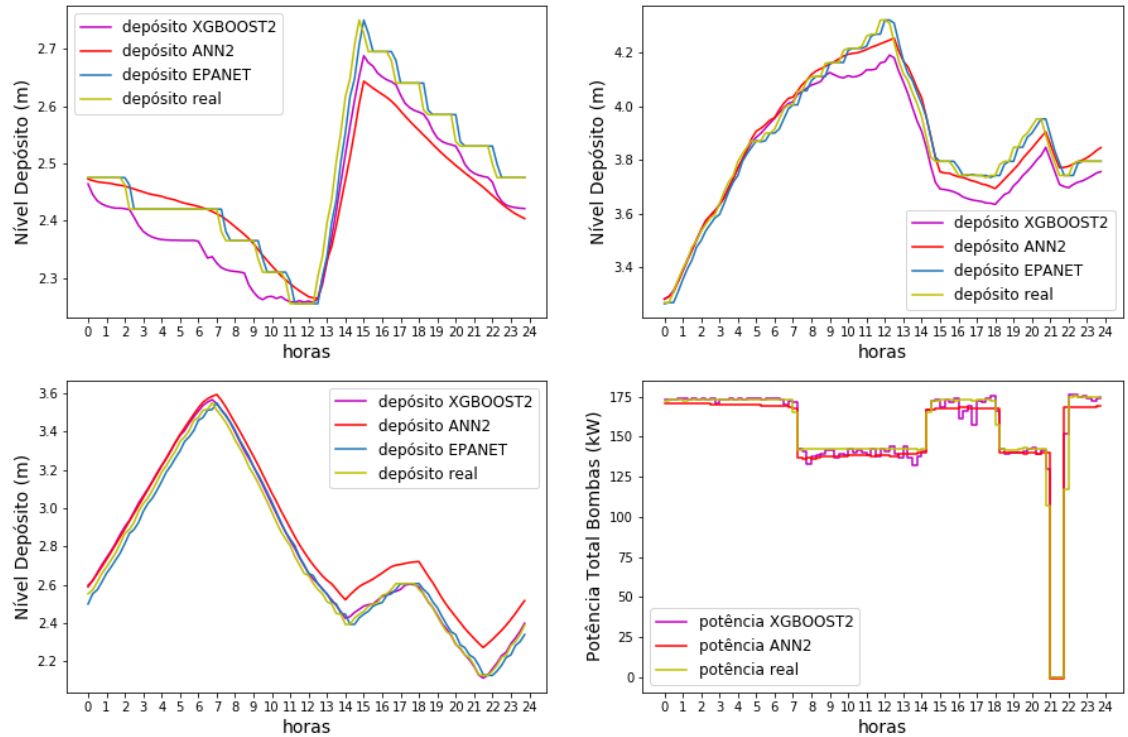


Figura 44 - Simulação dos depósitos e da energia para o dia de teste 3

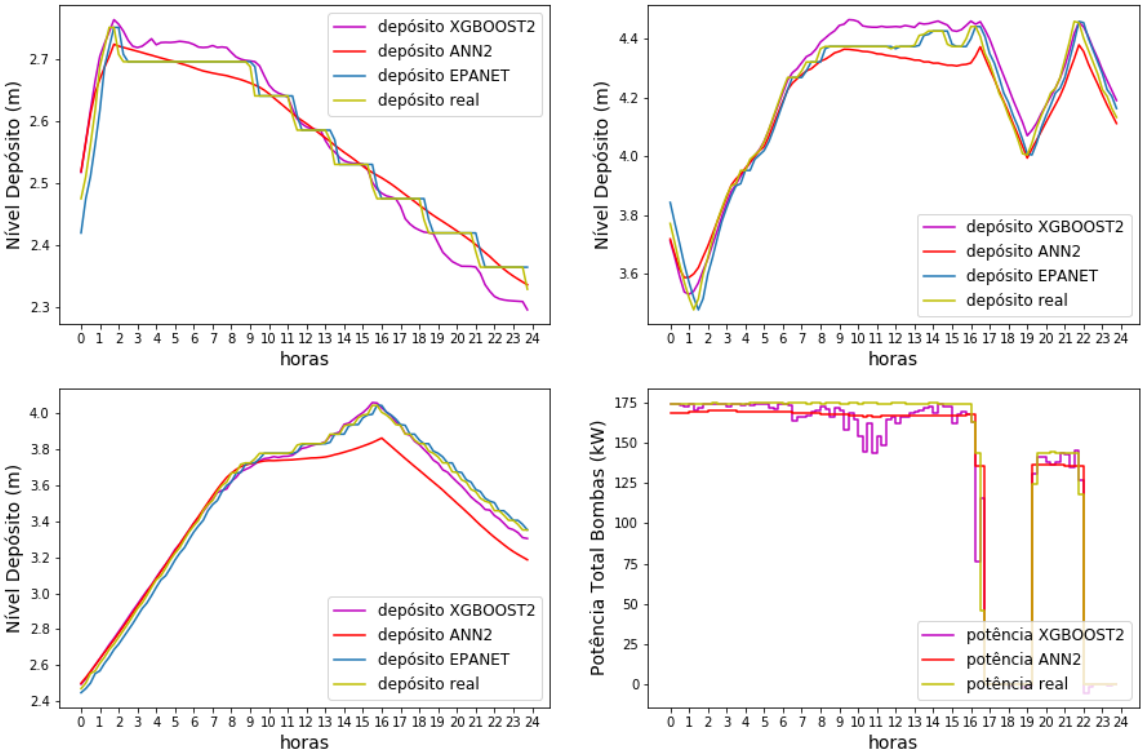


Figura 47 - Simulação dos depósitos e da energia para o dia de teste 4

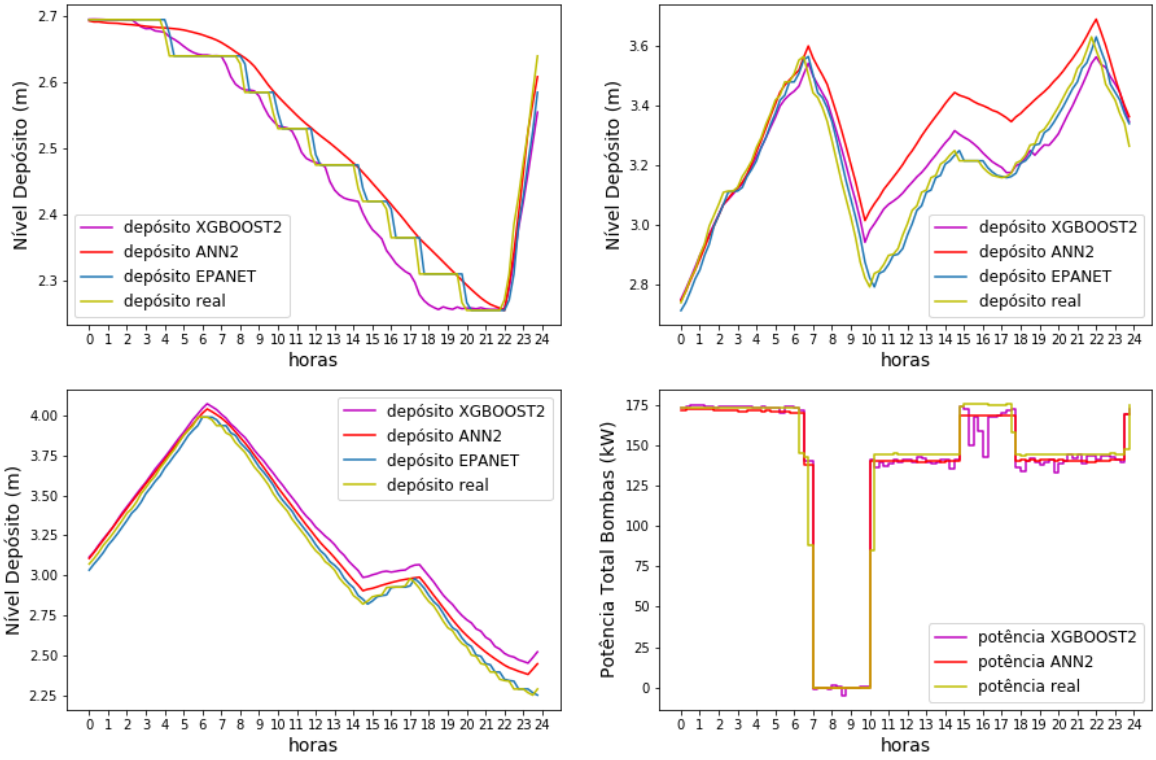


Figura 46 - Simulação dos depósitos e da energia para o dia de teste 5

		ANN			XGBoost		
		RMSE	MAE	R^2	RMSE	MAE	R^2
Potência Bombas (kW)	Dia 1	10,62	5,55	0,9767	10,94	4,24	0,9759
	Dia 2	7,46	4,69	0,9422	6,27	2,93	0,9606
	Dia 3	7,04	5,16	0,9760	7,62	4,18	0,9741
	Dia 4	11,15	6,39	0,9700	11,02	4,93	0,9715
	Dia 5	9,17	4,80	0,9701	10,06	4,30	0,9646
	Média	8,94	5,17	0,9743	9,54	4,24	0,9715
Nível Depósitos (m)	Dia 1	0,0590	0,0479	0,9323	0,0603	0,0472	0,9306
	Dia 2	0,0720	0,0567	0,8528	0,0561	0,0444	0,9029
	Dia 3	0,1109	0,0872	0,8762	0,0768	0,0585	0,9515
	Dia 4	0,0702	0,0473	0,9444	0,0383	0,0308	0,9723
	Dia 5	0,0946	0,0684	0,8204	0,0894	0,0678	0,8984
	Média	0,0882	0,0678	0,8763	0,0755	0,0578	0,9268

Tabela 14 - Resultados obtidos para cada dia de teste das ANN e XGBoost Comparação com dados reais. Média dos dias de teste 1,3 e 5.

4.3.4 Conclusão

Tal como seria esperado, a presença de ruído na base de dados criou dificuldades acrescidas no treino tanto da ANN como do XGBoost. Para ultrapassar esse problema, recorreu-se a técnicas de suavização. Apesar das melhorias significativas com a sua utilização, não foi possível eliminar todo o ruido presente, como é possível verificar, por exemplo, nas oscilações do modelo XGBoost na simulação da potência das bombas. Isto poderá dever-se às limitações do Exponential Smoothing. Tal como referido anteriormente, o Exponential Smoothing dá um maior destaque às amostras mais pequenas, não conseguindo identificar sazonalidades ou outro tipo de padrões. Assim, uma das formas de tentar reduzir o ruido e melhorar os resultados seria utilizar técnicas de suavização mais avançadas.

Apesar de nenhum dos modelos ter conseguido imitar na perfeição o comportamento da rede, os dois apresentaram resultados bastante positivos com valores de R^2 superiores a 0,85 e RMSE inferiores a 10 cm na simulação dos depósitos. Estes resultados demonstram que os modelos utilizados são capazes de replicar o comportamento de redes reais de elevada complexidade, mesmo com elevados valores de ruido.

Embora os modelos *ML* não tenham conseguido obter resultados melhores que o EPANET na previsão dos níveis dos depósitos, estes apresentam resultados promissores. Para além da utilização de técnicas de suavização mais avançadas, um conhecimento mais profundo sobre o comportamento da rede pode ajudar a obter melhores resultados. Por se tratar de um erro cumulativo, pequenos pormenores no funcionamento do sistema (como ligeira variação do nível máximo ou mínimo dos depósitos em diferentes dias da semana ou alturas do ano) podem levar a resultados menos precisos. Assim, um grande conhecimento da rede pode levar a resultados dos modelos mais precisos.

5. Conclusão

Neste trabalho, foram desenvolvidos diferentes modelos de *Machine Learning* para modelar e otimizar SAA. Para isso, foram utilizados diferentes casos de estudo de complexidades e tamanhos diferentes de modo a verificar o comportamento destes modelos em diferentes situações. Ao contrário dos trabalhos encontrados, os modelos foram treinados com intervalos temporais variáveis. Isto provou-se uma grande vantagem na generalização e precisão dos modelos, principalmente o modelo diferencial. Esta caraterística pode ser bastante útil para casos reais, permitindo às empresas a possibilidade de adaptarem os modelos para as suas necessidades.

Para o treino dos modelos é necessário fornecer dados de qualidade que refletem o comportamento do sistema. Caso os dados apresentem elevados valores de ruido, os modelos vão tentar encontrar padrões para esses valores e analisar erradamente esses dados, resultando num modelo bastante mais complexo e menos preciso. Em casos reais, devido às limitações dos sensores, é impossível garantir dados sem ruído. Contudo, como demonstrado no caso de estudo da rede da Ronqueira, com a ajuda de técnicas de suavização eficientes, é possível melhorar a qualidade dos dados obtidos e garantir assim modelos com elevada precisão.

Devido às limitações de tempo e do enorme número de variáveis a controlar resultantes dos intervalos de 15 minutos (96 intervalos de tempo para cada uma das bombas) utilizados no treino do caso de estudo da Ronqueira, não foi possível realizar a otimização desse sistema, ficando adiado para um trabalho futuro.

Embora todos os modelos tenham apresentado excelentes resultados na simulação dos SAA, apenas os modelos diferenciais foram capazes de otimizar as redes. Isto deve-se essencialmente devido à utilização de intervalos temporais como *input,* não conseguindo os algoritmos ajustar o seu funcionamento às variações temporais extremamente reduzidas introduzidas pelos algoritmos de otimização. Como os modelos diferenciais não dependiam do tempo, estes foram capazes de otimizar o primeiro caso de estudo. Apesar disso, a utilização destes modelos, com este algoritmo de otimização, em casos de estudo com mais do que duas bombas não é viável. Como para o modelo de otimização as bombas se encontram entre os valores 0 e 1, mas são fornecidos aos modelos como 0 ou 1, é extremamente complexo fornecer a informação ao sistema no caso de as bombas apresentarem tempos de funcionamento diferentes. Assim, é necessário encontrar uma melhor alternativa.

Apesar disso, por terem apresentado excelentes resultados para todos os casos de estudo e por serem computacionalmente mais eficientes que os simuladores hidráulicos e não necessitarem de processos de calibração extremamente longos, estes modelos de *Machine Learning* apresentam-se como uma excelente alternativa na modelação e otimização de sistemas de abastecimento de água reais.

Referências Bibliográficas

[1] J. Schleich, T. Hillenbrand. "Determinants of residential water demand in Germany". *Ecol. Econ.* vol. *68*, pp .1756–1769, 2009.

[2] J. Nathanson, "Water supply system." [Online]. Available: https://www.britannica.com/technology/water-supply-system %0A [Accessed: 01-April-2021].

[3] B. Coelho, "Energy efficiency of water supply systems using optimisation techniques and microhydroturbines," PhD Thesis, Universidade de Aveiro, 2016.

[4] A. Antunes, A. Andrade-Campos, A. Sardinha-Lourenço, and M. S. Oliveira, "Short-term water demand forecasting using machine learning techniques," *J. Hydroinformatics*, vol. 20, no. 6, pp. 1343–1366, Aug. 2018.

[5] U. S. E. P. Agency, "EPANET." [Online]. Available: https://www.epa.gov/water-research/epanet [Accessed: 04-April-2021].

[6] Z. Rao and F. Alvarruiz, "Use of an artificial neural network to capture the domain knowledge of a conventional hydraulic simulation model," *J. Hydroinformatics - J HYDROINFORM*, vol. 9, 2007.

[7] M. López-Ibáñez, "Operational Optimisation of Water Distribution Networks.," School of Engineering and the Built Environment, Edinburgh Napier University, 2009.

[8] M. J. H. STERLING and B. COULBECK, "TECHNICAL NOTE. A DYNAMIC PROGRAMMING SOLUTION TO OPTIMIZATION OF PUMPING COSTS.," *Proc. Inst. Civ. Eng.*, vol. 59, no. 4, pp. 813–818, 1975.

[9] P. Jowitt and G. Germanopoulos, "Optimal Pump Scheduling in Water_Supply Networks," *J. Water Resour. Plan. Manag.*, vol. 118, pp. 406–422, 1992.

[10] L. K. W. and M. B. J., "Minimization of Raw Water Pumping Costs Using MILP," *J. Water Resour. Plan. Manag.*, vol. 115, no. 4, pp. 511–522, Jul. 1989.

[11] W. R. Centrer, "Analysis and Simulation of Water Networks and a Guide to WATNET 3," 1987.

[12] I. Nouiri,"WATNET."Online].Available:https://sites.google.com/site/drissamnouiri/watnet-software (visited on 05/31/2019).

[13] A. Saad and I. Saad, "Hydraulic analysis and cost optimization of water network by using the EPANET software." 2018.

[14] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM J. Res. Dev.*, vol. 3, no. 3, pp. 210–229, 1959.

[15] A. Giron, *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly, 2019.

[16] C. M. Bishop, *Pattern Recognition and Machine Learing*. New York: Springer, 2006.

[17] S. Mouatadid and J. Adamowski, "Using extreme learning machines for short-term urban water demand forecasting," *Urban Water J.*, vol. 14, no. 6, pp. 630–638, 2017.

[18] H. Gergely, P. György, and G.-T. Bálint, “Deep Reinforcement Learning for Real-Time Optimization of Pumps in Water Distribution Systems,” *J. Water Resour. Plan. Manag.*, vol. 146, no. 11, p. 4020079, Nov. 2020.

[19] C. Y. Hu, J. Y. Cai, D. Z. Zeng, X. S. Yan, W. Y. Gong, and L. Wang, “Deep reinforcement learning based valve scheduling for pollution isolation in water distribution network.,” *Math. Biosci. Eng.*, vol. 17, no. 1, pp. 105–121, Sep. 2019.

[20] H. Mala-Jetmarova, N. Sultanova, and D. Savic, “Lost in Optimisation of Water Distribution Systems? A Literature Review of System Design,” *Water*, vol. 10, p. 307, Mar. 2018.

[21] D. Jamieson, U. Shamir, F. Martínez Alzamora, and M. Franchini, “Conceptual design of a generic, real-time, near-optimal control system for water-distribution networks,” *J. Hydroinformatics - J HYDROINFORM*, vol. 9, Jan. 2007.

[22] Z. Rao and E. Salomons, “Development of a real-time, near-optimal control process for water-distribution networks,” *J. Hydroinformatics*, vol. 9, pp. 25–37, 2006.

[23] S. Alvisi, M. Franchini, and A. Marinelli, “A short-term, pattern-based model for water-demand forecasting,” *J. Hydroinformatics*, vol. 9, no. 1, pp. 39–50, Jan. 2007.

[24] E. Salomons, A. Goryashko, U. Shamir, Z. Rao, and S. Alvisi, “Optimizing the Operation of the Haifa-A Water System,” *J. Hydroinformatics*, vol. 9, pp. 51–64, Jan. 2007.

[25] F. Martínez Alzamora, V. Hernández, J. Alonso, Z. Rao, and S. Alvisi, “Optimizing the Operation of the Valencia Water Distribution System,” *J. Hydroinformatics*, vol. 9, pp. 65–78, Jan. 2007.

[26] S. P. Beckwith and K. Wong, “A genetic algorithm approach for electric pump scheduling in water supply systems,” *Proc. 1995 IEEE Int. Conf. Evol. Comput.*, vol. 1, pp. 21-, 1995.

[27] F. Odan, L. Reis, and Z. Kapelan, “Use of Metamodels in Real-Time Operation of Water Distribution Systems,” *Procedia Eng.*, vol. 89, pp. 449–456, Dec. 2014.

[28] M. M. Islam, M. A. Sattar, M. F. Amin, X. Yao, and K. Murase, “A New Adaptive Merging and Growing Algorithm for Designing Artificial Neural Networks,” *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 39, no. 3, pp. 705–722, 2009.

[29] B. M. and W. Z. Y., “GPU-Based Artificial Neural Network Configuration and Training for Water Distribution System Analysis,” *World Environmental and Water Resources Congress 2012*. pp. 3108–3121, 24-Jun-2021.

[30] Z. Y. Wu, M. El-Maghraby, and S. Pathak, “Applications of Deep Learning for Smart Water Networks,” *Procedia Eng.*, vol. 119, pp. 479–485, 2015.

[31] A. C. Ian Goodfellow, Yoshua Bengio, *Deep Learning*. MIT Press, 2016.

[32] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets.,” *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[33] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science (80-.).*, vol. 313, no. 5786, pp. 504 LP – 507, Jul. 2006.

[34] Kaggle, “Kaggle.com/competitions.” [Accessed: 10-Jun-2021].

[35] F. Chollet, *Deep learning with python*. Manning Publications.

[36] “Home - Keras Documentation.” [Online]. Available: https://keras.io/. [Accessed:

12-May-2021].

[37] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[38] “Introduction to Boosted Trees,” 2015. [Online]. Available: http://xgboost.readthedocs.io/en/latest/model.html.

[39] P. Bühlmann, “Bagging, Boosting and Ensemble Methods BT - Handbook of Computational Statistics: Concepts and Methods,” J. E. Gentle, W. K. Härdle, and Y. Mori, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 985–1022.

[40] J. Friedman, “Greedy function approximation: A gradient boosting machine.,” *Ann. Stat.*, vol. 29, pp. 1189–1232, 2001.

[41] D. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.,” 2020. [Online]. Available: https://doi.org/10.1038/s41592-019-0686-2.

[42] R. Fletcher, *Practical Methods of Optimization*, vol. 2nd ed. New York: John Wiley & Sons, 1987.

[43] “scikit-learn: machine learning in Python — scikit-learn 0.20.3 documentation.” [Online].Available:https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. [Accessed: 12-Jun-2021].

[44] “Richmond.”[Online].Available:https://emps.exeter.ac.uk/engineering/research/cws/research/distribution/benchmarks/operation/richmond.html. [Accessed: 01-Jun-2021].

[45] “Scubic.” [Online]. Available: https://scubic.tech/pt/. [Accessed: 04-Jun-2021].