



**Previsão de consumos de energia associados à
climatização em edifícios de serviços com base em
estratégias de *Machine Learning***

Dissertação

Mestrado em Engenharia da Energia e do Ambiente

Xavier Dinis Valeiro Godinho

Leiria, novembro de 2020

Esta página foi intencionalmente deixada em branco



**Previsão de consumos de energia associados à
climatização em edifícios de serviços com base em
estratégias de *Machine Learning***

Mestrado em Engenharia da Energia e do Ambiente

Xavier Dinis Valeiro Godinho

Dissertação realizada sob a orientação do Professor Hermano Joaquim dos Santos Bernardo
e coorientado pelo Professor João Miguel Charrua de Sousa

Leiria, novembro de 2020

Esta página foi intencionalmente deixada em branco

Este trabalho foi apoiado por fundos FEDER e fundos Nacionais através da FCT no âmbito dos projetos ESRIDS (POCI-01-0145-FEDER-016434) e MAnAGER (POCI-01- 0145-FEDER- 028040).

Trabalho desenvolvido no âmbito da Bolsa de Investigação UI0308-ML2Building.1/2019, promovida pelo INESC Coimbra – Instituto de Engenharia de Sistemas e Computadores de Coimbra, com a duração total de 12 meses.

Esta página foi intencionalmente deixada em branco

Originalidade e direitos de autor

A presente dissertação é original, elaborada unicamente para este fim, tendo sido devidamente citados todos os autores cujos estudos e publicações contribuíram para a elaborar.

Reproduções parciais deste documento serão autorizadas na condição de que seja mencionado o Autor e feita referência ao ciclo de estudos no âmbito do qual a mesma foi realizada, a saber, Curso de Mestrado em Engenharia da Energia e do Ambiente, no ano letivo 2019/2020, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, Portugal, e, bem assim, à data das provas públicas que visaram a avaliação destes trabalhos.

Esta página foi intencionalmente deixada em branco

Agradecimentos

Queria aproveitar para deixar um agradecimento a todas as pessoas que, de alguma forma, contribuíram para a realização desta dissertação.

Em primeiro lugar, ao Professor Hermano Joaquim dos Santos Bernardo e ao Professor João Miguel Charrua de Sousa por me terem dado a oportunidade de ingressar neste projeto, por todo o apoio prestado e por tudo o que me ensinaram ao longo de todo o meu percurso académico.

Em segundo lugar, à minha família por terem sido o meu maior suporte durante todos estes anos, especialmente a minha mãe, o meu irmão e a minha avó.

Por fim, a todos os meus amigos por toda a ajuda que me deram, por todas as vivências que partilhei com eles e pelas palavras de incentivo, que me ajudaram a superar esta etapa da minha vida.

Esta página foi intencionalmente deixada em branco

Resumo

A previsão de consumos energéticos tem vindo a ganhar uma grande notoriedade nos últimos anos pelo seu papel na gestão e monitorização de consumos dos edifícios. Além disso, através da análise do padrão de consumos energéticos, é possível estudar algumas potenciais medidas de poupança de consumos, bem como detetar algumas falhas operacionais, seja de consumos excessivos ou inferiores ao esperado.

Com o aumento da tecnologia que tem sido desenvolvida e o número de sensores que têm sido instalados nos edifícios, a quantidade e qualidade da informação associada aos edifícios tem melhorado, o que permite uma melhor *performance* dos modelos associados ao *machine learning*.

Com base nisso, foram desenvolvidos nesta dissertação modelos de previsão baseados em diferentes técnicas de *machine learning* para prever as necessidades de aquecimento e arrefecimento para o dia seguinte de um edifício de escritórios localizado no Parque das Nações, Lisboa. Os modelos selecionados para este fim foram as redes neuronais artificiais e os modelos *baseados em máquinas de vetores de suporte*. Além disso, foi ainda estudada uma variação dos modelos SVM aplicando o *simulated annealing* como meta-heurística para otimizar os parâmetros destes modelos. Os históricos de potência e algumas variáveis exógenas (tais como a temperatura exterior, o perfil de ocupação, a humidade relativa, entre outras...) foram utilizados como variáveis de entrada para alimentar os modelos de forma a perceber a sua influência na previsão das necessidades de aquecimento e arrefecimento.

As métricas de erro utilizadas para comparar os valores previstos pelos modelos com os valores utilizados como referência foram o erro médio absoluto e o erro médio quadrático.

As variáveis mais influentes nas previsões das necessidades de aquecimento foram os históricos de potência, a temperatura exterior e a radiação solar direta. Já no período de arrefecimento, além dos históricos de potência e da temperatura exterior, também o perfil de ocupação e a humidade relativa permitiram aos modelos prever com precisão as necessidades de arrefecimento do edifício.

Palavras-chave: *Machine Learning*, máquinas de vetores de suporte, previsão de consumos, redes neuronais artificiais, *simulated annealing*

Abstract

Energy load forecast in buildings has been increasing its importance over the last few years in managing and monitoring building's energy consumptions. Besides, by analyzing the building's energy consumption pattern, it is possible to study some energy efficiency measures and also detect some irregular patterns or even alarm the facility manager to remotely control a specific equipment.

With the increasing development of the technology associated to building's energy management and the amount of sensors being installed, the quality and quantity of information being retrieved from the buildings has also been increasing, allowing the machine learning models to have a higher performance in forecasting energy demands.

Based on that factor, in this study some machine learning models were developed to forecast heating and cooling needs for the next day of an office building located in Parque das Nações, Lisboa, Portugal. The machine learning techniques selected to achieve the proposed goal were artificial neural network and support vector machine. Complementing that, it was also considered a meta-heuristic method (simulated annealing) as a variation of the SVM models, to optimize the SVM parameters and increase its efficiency in forecasting heating and cooling needs of the case study building. Historical records and some exogenous variables (such as outdoor air temperature, the estimated occupancy profile, relative humidity, etc.) were considered as inputs to feed the models and study their influence in forecasting heating and cooling needs.

The error metrics used to compare the predicted values obtained through the use of machine learning models with the reference values were the mean absolute error and the root mean squared error.

Historical thermal power records, outdoor air temperature and direct solar radiation were the most influent variables in forecasting heating energy demand. As for the cooling period, besides historical records and outdoor air temperature, also the occupancy profile and relative humidity provided the best forecasts for cooling energy demand.

Keywords: Machine Learning, Heating and Cooling load forecasting, Artificial Neural Network, Simulated Annealing, Support Vector Machines

Índice

Originalidade e direitos de autor	vii
Agradecimentos	ix
Resumo	xi
Abstract	xiii
Lista de figuras	xviii
Lista de tabelas	xx
Lista de siglas e acrónimos.....	xxi
1. Introdução	1
1.1. Enquadramento	1
1.2. Objetivos.....	2
1.3. Estrutura do trabalho	3
2. Previsão de consumos energéticos.....	5
2.1. Redes neuronais artificiais.....	8
2.2. Máquinas de vetores de suporte.....	11
3. Metodologia.....	13
3.1. Descrição do caso de estudo.....	13
3.2. Descrição do conjunto de dados	14
3.3. Adaptação dos modelos de previsão	16
3.3.1. Redes neuronais artificiais.....	16
3.3.2. Máquinas de vetores de suporte.....	18
3.3.3. Utilização de <i>simulated annealing</i> para otimização dos parâmetros SVM.....	18
3.4. Métricas de Erro.....	22
3.5. Análise de correlação	24
3.5.1. Período de aquecimento.....	24
3.5.2. Período de arrefecimento.....	25
3.5.3. Influência da temperatura exterior nas necessidades de aquecimento e arrefecimento	26

4.	Apresentação e discussão de resultados	29
4.1.	Período de aquecimento.....	29
4.1.1.	Caracterização dos modelos testados	29
4.1.2.	ANN - Seleção de variáveis de entrada.....	30
4.1.3.	ANN - Comparação dos valores previstos com a referência	31
4.1.4.	ANN - Análise da distribuição dos erros	32
4.1.5.	SVM - Seleção de variáveis de entrada.....	33
4.1.6.	SVM - Comparação dos valores previstos com a referência	34
4.1.7.	SVM - Análise da distribuição dos erros	35
4.1.8.	SVM + SA - Seleção de variáveis de entrada	36
4.1.9.	SVM + SA - Comparação dos valores previstos com a referência	37
4.1.10.	SVM + SA - Análise da distribuição dos erros	38
4.1.11.	Análise comparativa	39
4.1.12.	Previsão vs Referência	40
4.1.13.	Distribuição dos erros para os modelos de previsão mais precisos.....	41
4.2.	Período de arrefecimento	42
4.2.1.	Caracterização dos modelos testados	42
4.2.2.	ANN – Seleção de variáveis de entrada	43
4.2.3.	ANN - Comparação dos valores previstos com a referência	44
4.2.4.	ANN - Análise da distribuição dos erros	45
4.2.5.	SVM - Seleção de variáveis de entrada.....	46
4.2.6.	SVM - Comparação dos valores previstos com a referência	47
4.2.7.	SVM - Análise da distribuição dos erros	48
4.2.8.	SVM+SA - Seleção de variáveis de entrada	49
4.2.9.	SVM + SA - Comparação dos valores previstos com a referência	50
4.2.10.	SVM + SA - Análise da distribuição dos erros	51
4.2.11.	Análise comparativa	52
4.2.12.	Previsão vs referência.....	53
4.2.13.	Distribuição dos erros para os modelos de previsão mais precisos.....	54
5.	Considerações finais.....	55
	Referências bibliográficas.....	57
	Anexos	60
	Anexo A	61
	Anexo B	64
	Anexo C	67

Anexo D	74
----------------------	-----------

Lista de figuras

Figura 1.1. Energia final consumida por setor de atividade (1971-2018) em Mtep	1
Figura 2.1. Comparação entre a utilização de simulação dinâmica e modelos baseados em machine learning para previsão de consumos energéticos em edifícios	6
Figura 2.2. Influência da variação do γ na função RBF	11
Figura 2.3. Parâmetro ϵ como influente na margem de erro para os modelos SVM [26]	12
Figura 3.1. Modelo geométrico do edifício do caso de estudo [22]	13
Figura 3.2. Estrutura de uma rede neuronal artificial utilizada neste estudo baseada em históricos de consumo e temperatura exterior	17
Figura 3.3. Fluxograma que descreve a abordagem utilizada para aplicar o SA aos modelos SVM	20
Figura 3.4. Evolução da função objetivo ao longo da otimização dos parâmetros SVM durante o processo do SA	21
Figura 3.5. Representação de um gráfico <i>boxplot</i>	23
Figura 3.6. Relação entre a temperatura exterior e a potência de aquecimento de referência	26
Figura 3.7. Relação entre a temperatura exterior e a potência de arrefecimento de referência	27
Figura 4.1. Métricas de erro para as combinações testadas no período de aquecimento utilizando os modelos ANN	30
Figura 4.2. Comparação entre as previsões do modelo ANN H4 e os valores de referência	31
Figura 4.3. Boxplot de distribuição dos erros entre o modelo ANN H4 e a referência no subconjunto de teste	32
Figura 4.4. Métricas de erro para as combinações testadas no período de aquecimento utilizando os modelos SVM	33
Figura 4.5. Comparação entre as previsões do modelo SVM H4 e os valores de referência	34
Figura 4.6. <i>Boxplot</i> de distribuição dos erros entre o modelo SVM H4 e a referência no subconjunto de teste	35
Figura 4.7. Métricas de erro para as combinações testadas no período de aquecimento utilizando os modelos SVM+SA	36
Figura 4.8. Comparação entre as previsões do modelo SVM+SA H4 e os valores de referência	37
Figura 4.9. Boxplot de distribuição dos erros entre o modelo SVM+SA H4 e a referência no subconjunto de teste	38
Figura 4.10. Diagrama de carga semanal para os melhores modelos de previsão de consumos no período de aquecimento	39
Figura 4.11. Gráficos de dispersão das previsões dos modelos ANN H4, SVM H4 e SVM+SA H4 comparadas com os valores de referência no subconjunto de teste do período de aquecimento	40

Figura 4.12. Distribuição dos erros dos melhores modelos de previsão para o subconjunto de teste do período de aquecimento.....	41
Figura 4.13. Métricas de erro para as combinações testadas no período de arrefecimento utilizando os modelos ANN.....	43
Figura 4.14. Comparação entre as previsões do modelo ANN C2 e os valores de referência.....	44
Figura 4.15. <i>Boxplot</i> de distribuição dos erros entre o modelo ANN C2 e a referência no subconjunto de teste	45
Figura 4.16. Métricas de erro para as combinações testadas no período de arrefecimento utilizando os modelos SVM.....	46
Figura 4.17. Comparação entre as previsões do modelo SVM C5 e os valores de referência	47
Figura 4.18. <i>Boxplot</i> de distribuição dos erros entre o modelo SVM C5 e a referência no subconjunto de teste	48
Figura 4.19. Métricas de erro para as combinações testadas no período de arrefecimento utilizando os modelos SVM+SA	49
Figura 4.20. Comparação entre as previsões do modelo SVM+SA C5 e os valores de referência.....	50
Figura 4.21. <i>Boxplot</i> de distribuição dos erros entre o modelo SVM+SA C5 e a referência no subconjunto de teste	51
Figura 4.22. Diagrama de carga semanal para os melhores modelos de previsão de consumos no período de arrefecimento.....	52
Figura 4.23. Gráficos de dispersão das previsões dos modelos ANN C2, SVM C5 e SVM+SA C5 comparadas com os valores de referência no subconjunto de teste do período de arrefecimento	53
Figura 4.24. Distribuição dos erros dos melhores modelos de previsão para o subconjunto de teste do período de arrefecimento	54

Lista de tabelas

Tabela 2.1. Variáveis de entrada consideradas por outros autores	8
Tabela 3.1. Descrição do conjunto de dados	14
Tabela 3.2. Análise de correlação dos dados no período de aquecimento	24
Tabela 3.3. Análise de correlação dos dados associados ao período de arrefecimento	25
Tabela 4.1. Combinações de parâmetros testadas para o período de aquecimento	29
Tabela 4.2. Combinações de parâmetros testadas para o período de arrefecimento	42

Lista de siglas e acrónimos

ADALINE	<i>Adaptive Linear Neuron</i>
ANFIS	<i>Adaptive Neuro Fuzzy Interface System</i>
ANN	<i>Artificial Neural Network</i>
C	Parâmetro SVM Custo
DNN	<i>Deep Neural Network</i>
ENN	<i>Elman Neural Network</i>
ESTG	Escola Superior de Tecnologia e Gestão
GRNN	<i>General Regression Neural Network</i>
IPL	Instituto Politécnico de Leiria
MAE	<i>Mean Absolute Error</i>
MAPE	<i>Mean Absolute Percentage Error</i>
MLP	<i>Multi-Layer Perceptron</i>
OCDE	Organização para a Cooperação e Desenvolvimento Económico
PLS	<i>Partial Least Squares</i>
RBF	<i>Radial Basis Function</i>
RBFN	<i>Radial Basis Function Network</i>
RL	Regressão Linear
RMSE	<i>Root Mean Squared Error</i>
RP	Regressão Polinomial
SA	<i>Simulated Annealing</i>
SGT	Sistemas de Gestão Técnica
SVM	<i>Support Vector Machines</i>
UTA	Unidade de Tratamento de Ar

1. Introdução

1.1. Enquadramento

A gestão de consumos energéticos em edifícios tem sido uma temática cada vez mais abordada devido à sua contribuição para a adoção de medidas de eficiência energética tanto em edifícios já existentes como em novos edifícios que venham a ser construídos. Como se observa no gráfico da Figura 1.1, de 1971 até 2018, o aumento do consumo de energia nos países da OCDE associado ao setor dos edifícios (linha verde) foi bastante notório, pelo que a necessidade de investimento em medidas de poupança energética e também na utilização mais racional da energia final disponível para os edifícios tem sido cada vez mais investigada e focada.

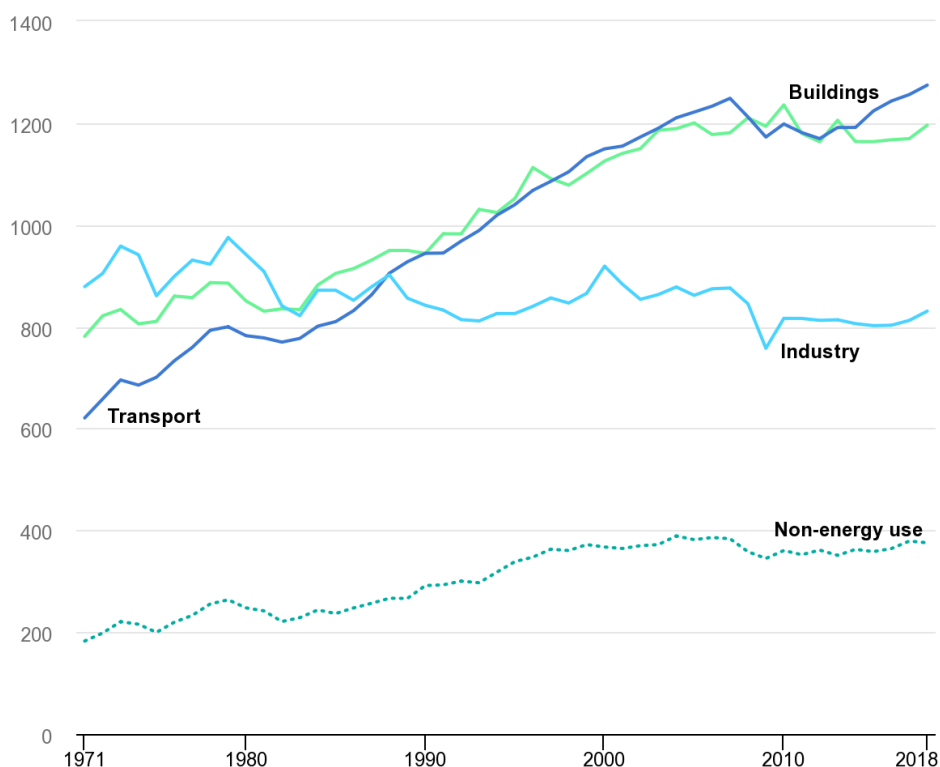


Figura 1.1. Energia final consumida por setor de atividade (1971-2018) em Mtep

Fonte: IEA, Total final consumption by source, OECD, 1971-2018, IEA, Paris <https://www.iea.org/data-and-statistics/charts/total-final-consumption-by-source-oecd-1971-2018>

Na Europa, 40% do consumo da energia final corresponde ao setor dos edifícios enquanto que em Portugal essa percentagem baixa para 30% [1].

Para uma gestão e monitorização mais facilitadas e completas dos consumos energéticos em edifícios de serviços, foram criados os sistemas de gestão técnica (SGT), que têm vindo a crescer nos últimos anos para fazer face, não só à legislação que tem vindo a ser desenvolvida no que diz respeito ao desempenho energético em edifícios de comércio e serviços, como também à necessidade de reduzir os consumos de energia e aumentar a familiarização com as novas tecnologias que têm vindo a ser desenvolvidas neste âmbito. [2]

Um SGT pode ser definido como o conjunto de técnicas e métodos aplicados a um edifício de modo a promover a utilização racional da energia disponível e a melhorar o seu desempenho energético. A utilização destes sistemas tem como objetivo ajudar a reduzir a emissão de gases de efeito de estufa bem como diminuir os encargos associados ao uso de energia. [3]

Uma das formas de otimizar os consumos energéticos de um edifício é através da previsão desses consumos. Existem duas principais metodologias para previsão de consumos energéticos: simulação dinâmica e *machine learning*. No caso do *machine learning*, apenas pode ser aplicado em edifícios já existentes ou em edifícios similares aos que pretendem ser estudados, uma vez que necessita de históricos de consumo desse edifício para treinar o modelo.

1.2. Objetivos

O presente trabalho consiste no desenvolvimento de modelos de previsão baseados em *machine learning* para prever as necessidades de aquecimento e arrefecimento para o dia seguinte de um edifício de serviços. Além disso, pretende-se também avaliar a influência de algumas variáveis exógenas nos consumos associados ao aquecimento e arrefecimento no edifício do caso de estudo.

Numa primeira fase, houve um processo de adaptação e aprendizagem em relação à linguagem *Python* (que foi a linguagem de programação utilizada para construir os modelos de previsão) e também à temática do *machine learning*. Nesta fase foram efetuados alguns testes, utilizando os consumos de eletricidade do edifício do caso de estudo, que apresentam um perfil mais fácil de prever.

Numa segunda fase, ocorreu a transição para os consumos associados ao aquecimento e arrefecimento, com todo o pré-tratamento dos dados que foi necessário fazer, bem como as adaptações dos modelos selecionados para efetuar as previsões.

Numa terceira fase, de modo a aumentar a eficiência dos modelos SVM, recorreu-se a uma meta-heurística (*Simulated Annealing*) de forma a otimizar a escolha dos parâmetros SVM mais adequados para as previsões.

1.3. Estrutura do trabalho

A presente dissertação está organizada em 5 capítulos distintos.

No capítulo 2 são explicados e desenvolvidos alguns conceitos tais como o conceito de *machine learning*, que foi a base desta dissertação, e também uma breve abordagem ao conceito de simulação dinâmica, que pode ser considerado como alternativa aos modelos *machine learning*. Também neste capítulo é feita uma revisão de trabalhos já desenvolvidos neste âmbito.

No capítulo 3 é feita a descrição do edifício utilizado no caso de estudo abordado nesta dissertação, do *dataset* utilizado e de quais as variáveis disponíveis e consideradas para este estudo. É ainda caracterizado o pré-processamento dos dados presentes no *dataset* e a forma como os períodos de aquecimento e arrefecimento foram definidos. É ainda descrita a adaptação dos modelos ANN e SVM para a previsão das necessidades de aquecimento e arrefecimento do edifício do caso de estudo, bem como a aplicação do SA de forma a otimizar os parâmetros dos modelos SVM. Por fim, é efetuada uma análise de correlação entre as variáveis consideradas como variáveis de entrada e os valores de referência para cada período e é estudada a influência da temperatura nas necessidades de aquecimento e arrefecimento dentro do edifício em estudo.

De seguida, no capítulo 4 são apresentados os resultados dos modelos desenvolvidos em cada período (aquecimento e arrefecimento). Numa primeira análise é feita a caracterização das várias combinações de variáveis de entrada consideradas, seguida dos resultados apresentados por cada modelo individualmente e, por fim, é feita uma análise comparativa entre as melhores combinações de cada modelo. Esta estrutura é aplicada primeiro para o período de aquecimento e depois também para o período de arrefecimento.

Previsão do consumo de energia associado à climatização em edifícios de serviços com base em estratégias de *Machine Learning*

Por fim, o capítulo 5 apresenta as conclusões obtidas neste estudo e algumas propostas para trabalhos futuros.

2. Previsão de consumos energéticos

Os modelos de simulação dinâmica e *machine learning* são utilizados muito frequentemente em estudos de previsão de consumos energéticos, seja de consumos de energia elétrica ou consumos associados à climatização dos edifícios.

Os modelos de simulação dinâmica dimensionados para este fim são mais utilizados de forma a obter uma estimativa dos consumos do edifício ainda na fase em que o projeto do edifício está a ser desenvolvido. Assim, através de uma caracterização e parametrização bastante precisas do edifício, é possível prever através de leis físicas os consumos energéticos futuros do edifício e de que forma os diferentes elementos constituintes do edifício (paredes, vãos envidraçados, cobertura, pontos de ventilação natural, etc.) influenciam esses consumos. Após os edifícios estarem operacionais, estes modelos têm como função monitorizar os consumos do edifício, comparar consumos com outros edifícios destinados ao mesmo fim e também avaliar medidas de eficiência energética que sejam aplicadas e aferir de que forma essas medidas afetam efetivamente os consumos do edifício. De forma a dimensionar estes modelos, é necessário passar por um processo de calibração que tem de ser bastante cuidado e extenso de forma a replicar da forma mais precisa possível todas as condições e todos os parâmetros associados ao edifício em questão.

Quanto aos modelos baseados em *machine learning*, tendo em conta o objetivo da previsão de consumos energéticos, só podem ser aplicados em edifícios já operacionais, uma vez que necessitam de dados históricos dos consumos do edifício de forma a “treinar” os modelos para prever os consumos energéticos futuros. Um dos melhores exemplos de aplicação dos modelos *machine learning* é nas pesquisas do Google. Com base nas pesquisas já efetuadas em relação a um determinado termo, os modelos sugerem várias hipóteses de pesquisa com base na sua relevância em relação ao termo a ser pesquisado. Quanto mais vezes esse termo for pesquisado, mais informação é adicionada aos modelos, que vão processar essa informação (processo de treino) e melhoram as sugestões quando esse termo é pesquisado novamente. [4]

A Figura 2.1 apresenta a comparação entre os métodos de previsão baseados em simulação dinâmica e em modelos de *machine learning*.

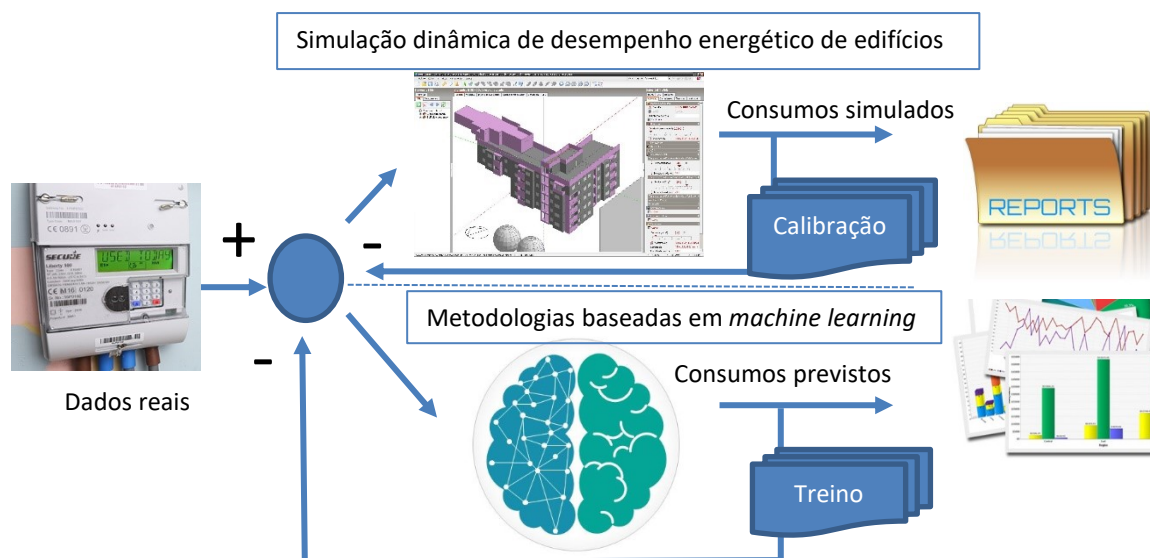


Figura 2.1. Comparação entre a utilização de simulação dinâmica e modelos baseados em machine learning para previsão de consumos energéticos em edifícios

No que concerne à utilização dos modelos de *machine learning* para previsão de consumos energéticos, há muitas abordagens distintas propostas por vários autores.

Bourdeau et.al [5] efetuaram uma revisão de vários artigos publicados tendo em conta os diferentes tipos de modelos utilizados na previsão de consumos energéticos de edifícios. Estes artigos foram seleccionados tendo em conta não só a data da sua publicação, considerando apenas os artigos publicados na última década, como também o foco dos artigos ser a previsão de consumos energéticos (elétricos ou necessidades de aquecimento e/ou arrefecimento) em edifícios singulares, descartando assim artigos focados em previsão de consumos em bairros, cidades, regiões ou países.

Com base na sua pesquisa em [5], os autores concluíram que os modelos autorregressivos, regressões estatísticas, o modelo dos “vizinhos mais próximos” (*k nearest neighbours*), árvores de decisão, SVM e ANN eram os modelos mais recorrentemente utilizados na previsão de consumos energéticos em edifícios residenciais ou de serviços.

Neto e Fiorelli [6] fizeram uma comparação entre um modelo de simulação dinâmica utilizando o *software EnergyPlus* e um modelo de *machine learning* (ANN) para prever os consumos energéticos de um edifício administrativo da universidade de S. Paulo no Brasil.

Katsatos e Moustris [7] recorreram também a redes neuronais artificiais para prever o consumo energético, o consumo de gás e as necessidades de arrefecimento para as 24h

seguintes do edifício da Autoridade Reguladora para a Energia, situado em Atenas, na Grécia.

Deb et. al [8] desenvolveram um estudo para prever consumos associados às necessidades de arrefecimento de três edifícios de serviços num campus de uma universidade em Singapura com base em modelos ANN e modelos ANFIS (*Adaptive Neuro Fuzzy Inference System*), que são uma variante dos modelos ANN. Este tipo de modelos faz a junção entre as redes neuronais e a lógica *fuzzy* proposta por Takagi e Sugeno.

Zhao e Liu [9] utilizaram modelos baseados em *wavelet transform*, PLS (*Partial Least Square Regression*) e SVM para prever as necessidades de aquecimento e arrefecimento de um edifício de escritórios situado em Tianjin, China, com 1h a 24h de antecedência.

Saloux e Candanedo [10], como parte de um estudo em maior escala que tem como objetivo prever as necessidades de aquecimento de uma comunidade composta por 52 casas, investigaram a capacidade de 3 modelos de *machine learning* (árvores de decisão, SVM e ANN) para prever as necessidades de aquecimento dessa comunidade.

Considerando as variáveis de entrada utilizadas pelos diferentes autores na previsão de consumos energéticos em edifícios, as abordagens também diferem muito. A Tabela 2.1 apresenta as variáveis consideradas pelos diferentes autores nos seus respetivos estudos.

Tabela 2.1. Variáveis de entrada consideradas por outros autores

Referências	[6]	[7]	[8]	[9]	[10]	[11]
Variáveis						
Históricos de potência (eletricidade)		X				
Históricos de potência (aquecimento e/ou arrefecimento)		X	X	X		X
Hora do dia				X	X	X
Tipo de dia (semana, fim-de-semana...)	X	X	X			
Mês / estação do ano		X				X
Temperatura	X	X		X	X	X
Perfil de ocupação	X	X				X
Humidade relativa	X	X		X		X
Radiação direta ou difusa	X			X	X	X
Vento (velocidade, orientação...)	X	X		X		X

2.1.Redes neuronais artificiais

As redes neuronais artificiais são sistemas computacionais programados que tentam replicar a microestrutura do sistema nervoso biológico. Tal como no ramo da biologia, as redes neuronais artificiais são compostas por unidades interligadas entre si chamadas de neurónios artificiais [12]. A estrutura das ANN está dividida em três seções distintas: camada de entrada (*input layer*), na qual são inseridos todos os valores relativos às variáveis de entrada, que depois são enviados para as camadas posteriores, a camada de saída (*output layer*), da qual são retirados os valores resultantes do modelo e das interligações criadas entre os neurónios da(s) camada(s) oculta(s) e dos neurónios existentes nesta própria camada de saída e a(s) camada(s) oculta(s), onde os dados provenientes da camada de entrada são processados através de uma função de ativação associada aos neurónios constituintes desta camada na procura de um padrão/correlação entre os diferentes neurónios das variáveis de entrada e os neurónios da camada de saída, no processo de treino. Esse padrão é representado por pesos (*weights*) e constantes (*biases*) que são atribuídos a cada neurónio na camada oculta. Depois de descobertos esses parâmetros, são introduzidos novos dados no processo de teste que utilizam esses valores para calcular os dados de saída.

Outro ponto muito importante a considerar no desenvolvimento de uma rede neuronal é a escolha da(s) função(ões) de ativação na(s) camada(s) oculta(s) da rede. A função de ativação confere a não-linearidade ao modelo, uma vez que os neurónios, por si só, funcionam de forma linear. A escolha da função de ativação mais adequada ao objetivo que se pretende é fundamental para o desempenho destes modelos.

Com base no tipo de ligações criadas entre os neurónios das diferentes camadas constituintes da ANN, estas podem ser divididas em seis tipos [13]:

- Adaptive Linear Neuron (ADALINE) – é a versão mais simples de uma rede neuronal artificial, uma vez que é composta apenas pela camada de entrada (que pode depender de várias entradas) e a camada de saída que tem apenas um neurónio, ou seja, este tipo de redes neuronais tem apenas um valor como *output*;
- Multi-Layer Perceptron (MLP) – apresenta uma estrutura *feed-forward*, ou seja, procura correlações apenas no sentido dos *inputs* para os *outputs* e não no sentido contrário. Este tipo de redes neuronais foca o seu processo de aprendizagem na adaptação dos pesos das ligações entre neurónios de modo a otimizar a função para a qual são desenvolvidos. No caso de a função de ativação selecionada ser não linear (por exemplo, sigmoide ou tangente hiperbólica), este tipo de redes têm a capacidade de identificar relações não-lineares entre os *inputs* e os *outputs*;
- Elman Neural Network (ENN) – também tem uma estrutura *feed-forward* e conta com uma camada adicional, designada por camada de contexto. Esta camada é paralela à camada oculta e tem a função de receber parte dos dados que saem da camada oculta e realimentar essa mesma camada, criando um desfaseamento nos dados a ser processados [14];
- Radial Basis Function Network (RBFN) – caracterizada por apresentar um grau de convergência bastante rápido, tem apenas uma camada oculta na sua estrutura e os neurónios constituintes dessa camada têm a mesma função de ativação radial. Este tipo de modelos é bastante apropriado para controlar não-linearidades e sistemas nos quais o maior foco seja nas incertezas e as variações de parâmetros associadas;

- General Regression Neural Network (GRNN) – é uma variante da RBFN. Tem uma camada de base radial e uma camada linear. Tem como principal vantagem não precisar de treino, uma vez que as previsões são uma média pesada dos exemplos de treino mais próximos e conta com um parâmetro (*sigma*) para determinar a precisão das previsões;
- Deep Neural Network (DNN) – utilizada para representar variáveis complexas a partir de outras mais simples, o que torna este tipo de modelos mais indicado para tratar grandes quantidades de dados, como é o caso do processamento de imagem. Este processamento é possível devido à existência de várias camadas ocultas na estrutura deste tipo de redes. Tal como em outras estruturas, tem uma fase de otimização associada, que consiste em encontrar parâmetros das DNN que reduzam significativamente a função custo.

A fase de treino dos modelos ANN é também fulcral para a sua eficácia em cumprir o objetivo para o qual foram desenvolvidos. Para isso, é selecionada uma métrica de erro como função objetivo que deve ser minimizada à medida que o número de iterações aumenta. Esse erro é depois enviado para as camadas precedentes, levando a que os pesos associados a cada neurónio sejam otimizados. Este processo é designado como retro-propagação do erro.

2.2. Máquinas de vetores de suporte

Os modelos SVM são algoritmos de *machine learning* baseados na aprendizagem estatística e no princípio da minimização estrutural do erro, apresentados pela primeira vez por Cortes e Vapnik em 1995 [15]. Apesar dos modelos SVM serem maioritariamente aplicados em problemas de classificação, têm sido muito estudados também em problemas de regressão [16]–[19]. A principal ideia subjacente a este tipo de modelos em problemas de classificação passa por encontrar um hiperplano otimizado para separar duas classes distintas. No caso dos problemas de regressão, o objetivo do hiperplano é aproximar-se do maior número de pontos possível, ou seja, minimizar a distância total entre os pontos pertencentes ao *dataset* e o hiperplano criado pelo modelo [20].

De forma a otimizar o modelo SVM, alguns parâmetros, tais como a função *kernel*, o parâmetro *kernel* (*gamma*), o parâmetro custo (*C*) e o *Epsilon* precisam de ser ajustados. A função *kernel* mais adequada para lidar com a não-linearidade dos dados, o que acontece frequentemente em casos de consumos energéticos, é a *Radial Basis Function* (RBF) [21]. Podem ainda ser aplicadas as funções linear, polinomial ou *sigmoide* (na qual se enquadra a função da tangente hiperbólica). O parâmetro *gamma* indica o nível de não-linearidade associado ao modelo SVM, de acordo com a largura do sino formado pela função RBF Figura 2.2.

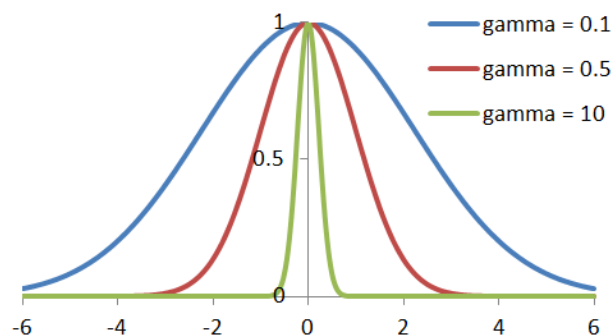


Figura 2.2. Influência da variação do *gamma* na função RBF

Fonte: <https://tomaszkacmajor.pl/index.php/2016/04/24/svm-model-selection/>

O parâmetro custo (C) tem como função ajustar o modelo SVM de forma a penalizar erros individuais e também evitar que o modelo se torne demasiado dependente dos dados do subconjunto de treino, perdendo a capacidade de previsão a partir de novos dados que lhe sejam introduzidos. A variação do *Epsilon* e a sua influência em problemas de regressão está representada na Figura 2.3.

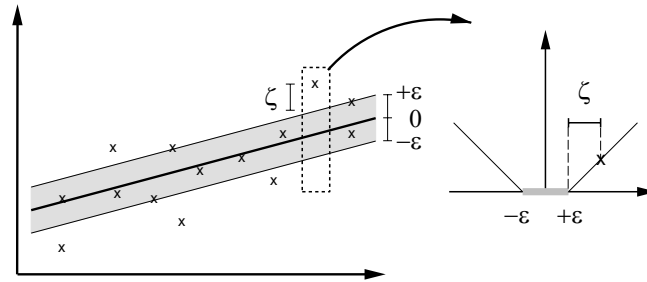


Figura 2.3. Parâmetro *Epsilon* como influente na margem de erro para os modelos SVM [26]

3. Metodologia

3.1. Descrição do caso de estudo

O caso de estudo desta dissertação foi baseado num edifício de escritórios localizado no Parque das Nações, em Lisboa, construído em 2007 e que está equipado com um sistema de controlo e automação avançado. Possui uma área total de 13 000 m², com cerca de 7 000 m² de área para escritórios distribuídos por 8 andares acima do solo e cerca de 5 000 m² de estacionamento subterrâneo. A envolvente do edifício é maioritariamente envidraçada, com as restantes paredes exteriores a serem constituídas por betão com espessura de 0,30m. A cobertura do edifício é composta por uma placa horizontal em betão armado, XPS e uma camada de seixo rolado. [22]

A climatização do edifício é assegurada a partir de uma central de trigeração, sendo a energia distribuída sob a forma de água arrefecida e água aquecida. A transferência da energia entre a rede de distribuição e o edifício é efetuada na subestação existente no piso - 2, por meio de permutadores de calor, do tipo placas, para o circuito de distribuição do edifício a quatro tubos. A ventilação é assegurada por duas unidades de tratamento de ar (UTAs) com possibilidade de aquecimento e arrefecimento do ar. [22]

O modelo geométrico do edifício do caso de estudo, obtido através do *software DesignBuilder*, está representado de seguida na Figura 3.1. [22]

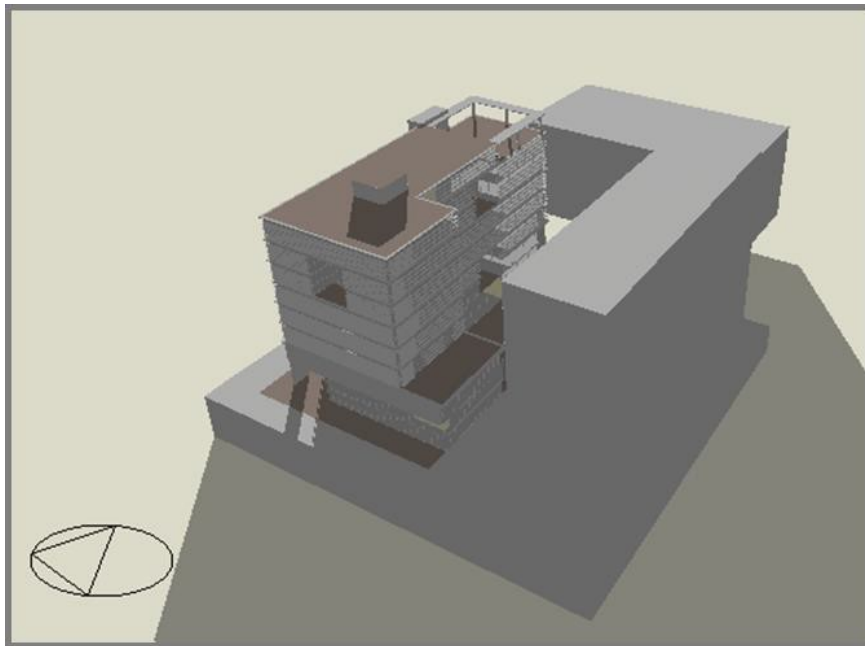


Figura 3.1. Modelo geométrico do edifício do caso de estudo [22]

3.2. Descrição do conjunto de dados

Na Tabela 3.1 estão representadas todas as variáveis que constituem o conjunto de dados utilizado neste estudo.

Tabela 3.1. Descrição do conjunto de dados

Variável	Tipo	Unidade	Valor Máximo	Valor Mínimo	Resolução
Data	<i>Date</i>	-	01/01/2002 a 31/12/2002		-
Hora	<i>Time</i>	h	-	-	-
Potência de aquecimento ¹	<i>Float</i>	kW	360,82	0	Horária
Potência de arrefecimento ¹	<i>Float</i>	kW	817,24	0	
Perfil de ocupação do edifício	<i>Float</i>	kW	88,85	0,08	
Temperatura de bolbo seco no exterior	<i>Float</i>	°C	35,90	4,20	
Ganhos solares pela envolvente ²	<i>Float</i>	kW	99,37	0	
Humidade relativa	<i>Float</i>	%	85,13	31,32	
Radiação solar direta	<i>Float</i>	kW	1,19	0	
Radiação solar horizontal difusa	<i>Float</i>	kW	0,40	0	
Velocidade do vento	<i>Float</i>	m/s	11,40	0	

¹ Tanto os dados associados à potência de aquecimento como à potência de arrefecimento foram obtidos a partir de um modelo de simulação dinâmica dimensionado e calibrado para o edifício em estudo.

² como os ganhos solares foram dimensionados através de um modelo de simulação dinâmica criado a partir do *software DesignBuilder*, e não havendo forma de medir este valor no próprio edifício, esta variável não foi contabilizada para os resultados deste estudo.

O efeito dos históricos de consumo para aquecimento e arrefecimento foi considerado tendo em conta os registos obtidos 7 e 14 dias antes do dia a ser previsto.

Este dataset foi depois dividido em dois períodos distintos: período de aquecimento e período de arrefecimento. O período de aquecimento foi definido começando no dia 1 de janeiro até 31 de março e de 1 de outubro a 31 de dezembro com as primeiras duas semanas (1 a 14 de janeiro) a entrarem apenas como *inputs* na variável dos históricos de consumos para aquecimento 14 dias antes do primeiro dia a ser previsto (15 de janeiro). Os primeiros 80% do total de dados a ser previstos respeitantes ao período de aquecimento (de 15 de janeiro a 31 de março e de 1 de outubro a 27 de novembro) foram utilizados para treinar os modelos considerados neste estudo e os restantes 20% (de 28 de novembro a 31 de dezembro) foram utilizados para testar a eficácia dos modelos na previsão das necessidades de aquecimento. Relativamente ao período de arrefecimento, foi aplicada a mesma proporção, ou seja, cerca de 80% do total de dados (15 de abril a 28 de agosto) foram utilizados para treinar os modelos e os restantes 20% (29 de agosto a 30 de setembro) foram utilizados para testar a eficácia dos modelos em prever as necessidades de arrefecimento, com o período de 1 a 14 de abril a integrar apenas os históricos de consumo para arrefecimento. O método de normalização utilizado neste estudo em ambos os períodos foi o método min-max, utilizando como fatores de normalização (0 e 1) o valor mais baixo e mais alto de cada variável estudada no subconjunto de treino [11]. Foi também estudado o método de normalização *standard*, baseado na média e desvio-padrão dos dados a ser normalizados. No entanto, os testes efetuados nas mesmas condições entre os dois métodos mostraram que o método min-max obteve melhores resultados, pelo que foi este o método selecionado para este estudo.

3.3. Adaptação dos modelos de previsão

3.3.1. Redes neuronais artificiais

Para gerar estes modelos em Python, foi utilizada a função *MLPRegressor*, importada da biblioteca *sklearn.neural-network*, na qual são definidos como argumentos:

- o número de neurónios a inserir na camada oculta;
- a função de ativação para essa camada;
- o número máximo de iterações;
- a função de otimização dos pesos de cada neurónio;
- a tolerância para a otimização do erro de validação das redes neuronais, forçando a paragem do processo caso a diminuição do erro de validação fosse inferior a essa tolerância;
- a percentagem dos dados de treino que seria selecionada como validação dos modelos

É ainda necessário ativar também o argumento *early_stopping* (ver Anexo A).

Assim, foram feitos alguns testes utilizando a função da tangente hiperbólica e a função logística como funções de ativação, por permitirem encontrar eventuais relações não-lineares entre entradas e saídas. A função de ativação selecionada após esses testes foi a função da tangente hiperbólica. A otimização dos pesos atribuídos a cada neurónio da rede neuronal foi feita através da função “*adam*” por ser a mais adequada (entre as funções disponíveis) para tratar grandes quantidades de dados. Foi considerada apenas uma camada oculta, a tolerância para a otimização considerada foi de 1^{-10} e 20% dos dados foram reservados para terminar a fase de treino das redes se o fator de validação parasse de melhorar com o número de iterações a serem feitas. De forma a selecionar o número mais adequado de neurónios a ser considerados na camada oculta, foram efetuados testes ao treinar os modelos por 10 vezes, considerando 10, 20, 30, 40 e 50 neurónios na camada oculta em cada bateria de testes. Através das métricas de erro associadas aos modelos gerados em cada bateria de testes, aquela que obteve melhores resultados foi a estrutura com 40 neurónios.

De um modo geral, os modelos ANN com um número reduzido de neurónios levam a uma maior incapacidade dos modelos em relacionar os *inputs* entre eles de modo a gerar previsões de necessidades de aquecimento e arrefecimento mais precisas. Por outro lado, quanto maior for o número de neurónios, maior é a probabilidade dos modelos ANN ficarem

demasiado dependentes do subconjunto de dados utilizado para treinar os modelos, fazendo com que a sua capacidade de se adaptarem a um novo conjunto de dados que lhe seja introduzido seja muito menor.

A Figura 3.2 descreve a estrutura de uma rede neuronal artificial utilizando os históricos (7 e 14 dias antes do dia a ser previsto) e a temperatura exterior como variáveis de entrada.

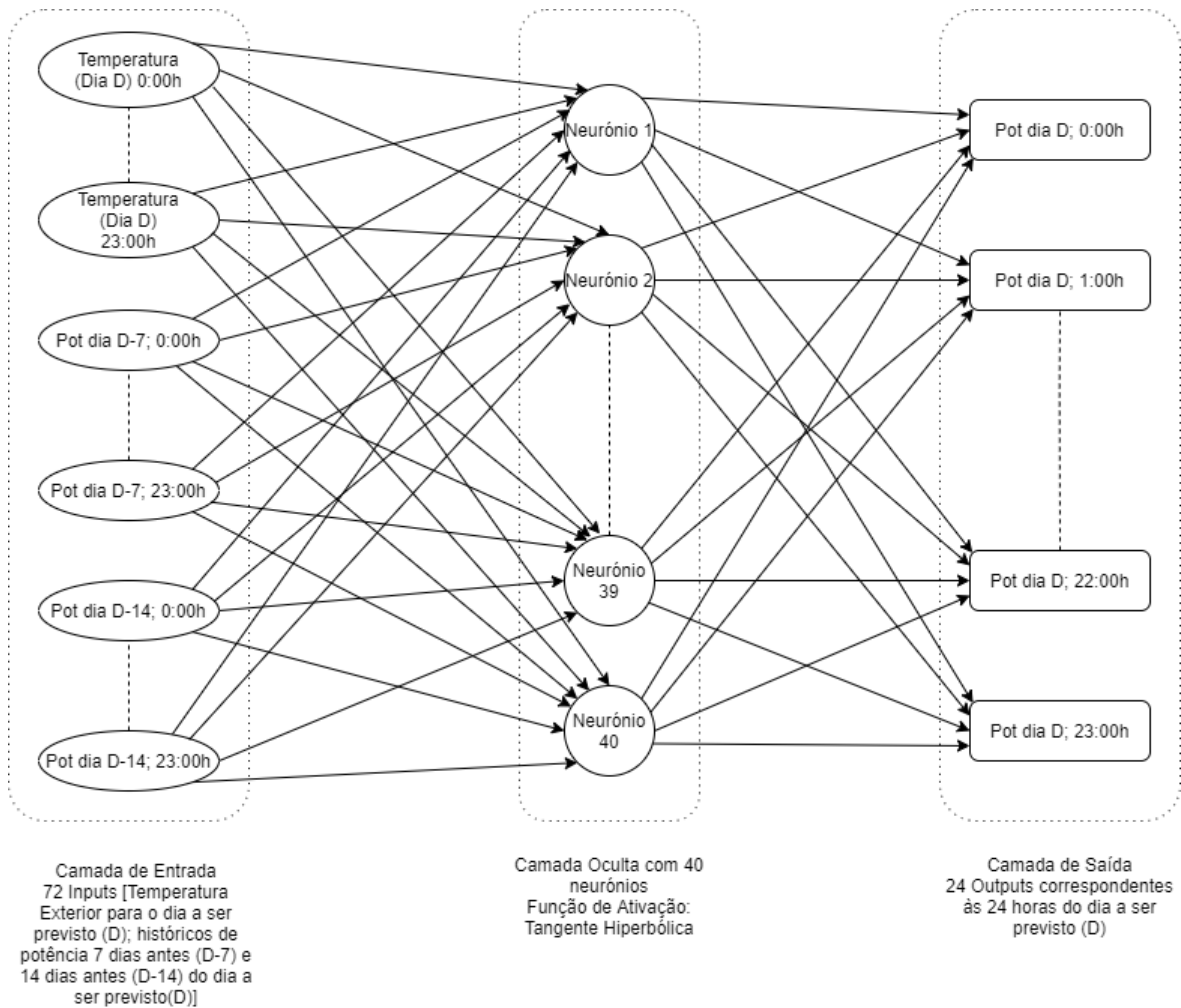


Figura 3.2. Estrutura de uma rede neuronal artificial utilizada neste estudo baseada em históricos de consumo e temperatura exterior

3.3.2. Máquinas de vetores de suporte

Em *Python*, estes modelos foram criados recorrendo à função SVR, importada da biblioteca *sklearn.svm* onde além dos parâmetros C , γ e ϵ , foi também definida a função de ativação RBF (ver Anexo B).

Para este estudo, foram testadas algumas combinações de parâmetros de forma empírica para determinar qual a mais eficaz para prever as necessidades de aquecimento e arrefecimento a partir deste *dataset*. Os parâmetros utilizados nestes testes foram os seguintes: parâmetro custo (100; 500), γ (0,1; 0,5) e ϵ (0,5; 1). A combinação que obteve as métricas de erro mais baixas para o subconjunto de teste foi usando os parâmetros: $C=500$, $\gamma=0,5$ e $\epsilon=0,5$.

3.3.3. Utilização de *simulated annealing* para otimização dos parâmetros SVM

O uso de métodos combinatórios é frequentemente sugerido para otimizar a seleção dos parâmetros SVM [17], [18], [23]–[25]. Entre várias meta-heurísticas que podiam ser utilizadas para otimizar os parâmetros SVM, neste estudo optou-se pelo *simulated annealing* (SA).

Esta técnica é uma das meta-heurísticas mais antigas e uma das suas principais características é a sua capacidade de conseguir escapar a mínimos ou máximos locais (consoante a função objetivo seja respetivamente de minimização ou de maximização).

A ideia subjacente a esta característica do *simulated annealing* é a existência da probabilidade de aceitar soluções que eventualmente possam deteriorar a função objetivo de modo a afastar-se de possíveis mínimos (ou máximos) locais. A solução atual é, assim, substituída por uma nova solução gerada de forma aleatória na vizinhança, sendo que a extensão da busca é determinada por uma probabilidade de distribuição proporcional à “temperatura”. À medida que o algoritmo vai avançando e novas soluções vão sendo testadas, a temperatura vai diminuindo, ou seja, o espaço de soluções possíveis vai sendo cada vez menor.

No entanto, de forma a evitar que o algoritmo se fixe num mínimo (máximo) local, existe sempre uma probabilidade de aumentar a temperatura (espaço de soluções) de forma a procurar novas soluções em outras regiões, procurando garantir que a função objetivo evite um mínimo (máximo) local e que se aproxime mais de um mínimo (máximo) absoluto.

Para este estudo, a abordagem aos modelos SVM com otimização do SA contou com algumas variantes em relação aos modelos SVM sem otimização.

Em primeiro lugar, apenas foram consideradas para efeitos de previsão as horas em que os sistemas de aquecimento e arrefecimento estavam a funcionar nos respetivos períodos.

Em segundo lugar, enquanto que nos modelos SVM sem otimização foi considerada a mesma combinação de parâmetros SVM para todas as horas a ser previstas, com a utilização do SA cada hora a ser prevista considera a sua própria combinação de parâmetros de forma a permitir aos modelos um maior grau de adaptação consoante a hora que esteja a ser prevista.

Por último, e associado ao ponto anteriormente abordado, os dados associados a cada hora a ser prevista foram normalizados utilizando como fatores de normalização os valores máximo e mínimo registados no subconjunto de treino para essa hora específica, enquanto que nos modelos SVM sem otimização os fatores de normalização considerados foram os valores máximo e mínimo globais registados para cada variável no subconjunto de treino.

A Figura 3.3 mostra o fluxograma que representa o algoritmo seguido neste estudo para aplicação do SA aos modelos SVM. No Anexo C são apresentados mais detalhadamente todos os passos associados à implementação do simulated annealing em Python.

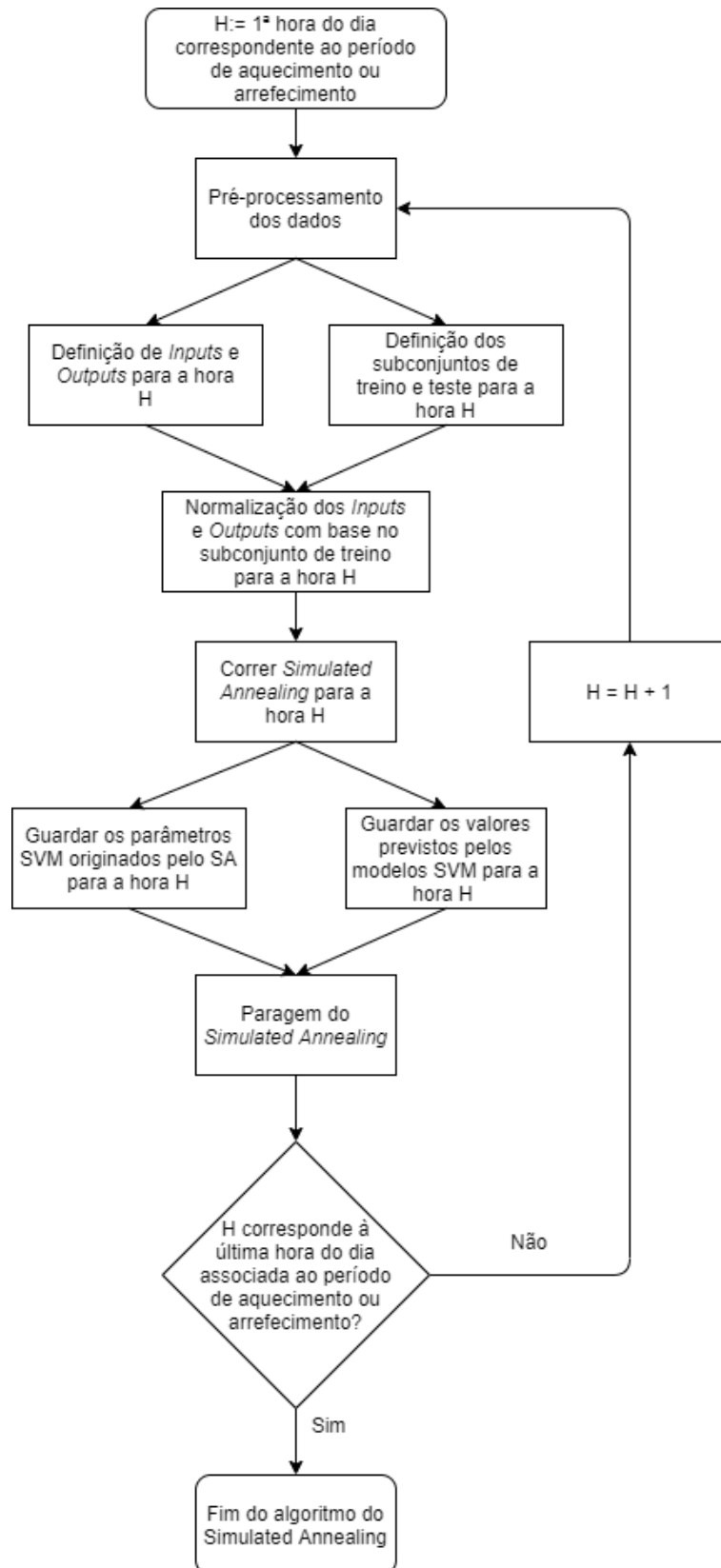


Figura 3.3. Fluxograma que descreve a abordagem utilizada para aplicar o SA aos modelos SVM

Durante o processo do SA, em cada hora foram efetuados 30 ciclos (iterações). Em cada ciclo, 15 combinações aleatórias de parâmetros SVM foram testadas sendo que, no final de cada ciclo, a “temperatura” baixava de maneira a estreitar o intervalo de opções disponíveis para cada parâmetro. O parâmetro C foi testado entre 0 e 700, o $Epsilon$ foi testado entre 0.0001 e 3 e o $gamma$ foi testado entre 0.0001 e 10. A função objetivo definida para este processo foi baseada no RMSE gerado para o subconjunto de validação. Na Figura 3.4 está representado um exemplo da otimização de parâmetros SVM recorrendo ao SA durante as 30 iterações referidas para as 14h no período de arrefecimento, utilizando históricos de potência de arrefecimento 7 e 14 dias antes do dia a ser previsto, temperatura exterior, perfil de ocupação e humidade relativa como variáveis de entrada. Nesta figura, são notórias as variações dos parâmetros a ser testados e as oscilações na função objetivo, comprovando a probabilidade de aceitar piores soluções durante o processo. No Anexo D são apresentados mais alguns gráficos de otimização dos parâmetros SVM em outras horas do período de arrefecimento, utilizando este modelo.

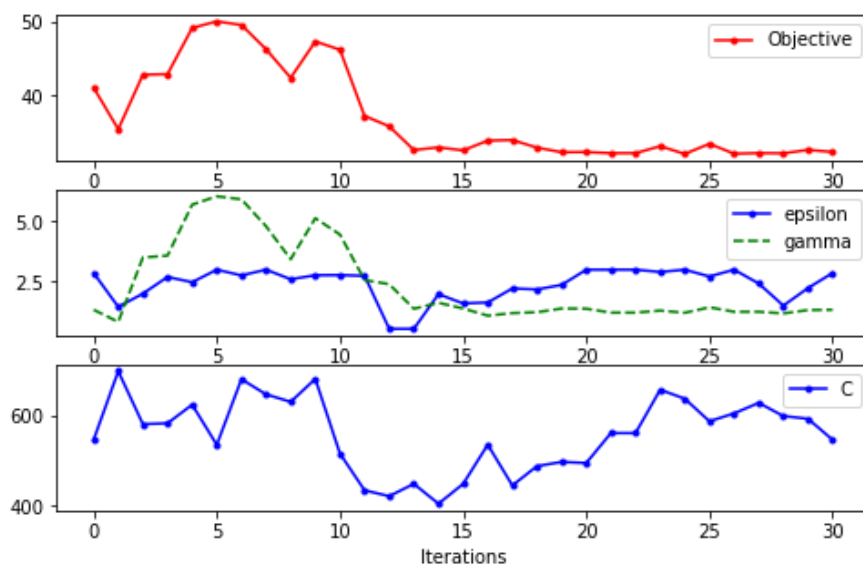


Figura 3.4. Evolução da função objetivo ao longo da otimização dos parâmetros SVM durante o processo do SA

O computador utilizado neste estudo conta com um processador Intel(R) Core(TM) i3-5005U @ 2GHz e memória RAM de 4,0 GB.

3.4. Métricas de Erro

Nesta seção estão descritas as métricas de erro utilizadas para avaliar os resultados obtidos pelos modelos de previsão estudados. A primeira métrica de erro utilizada foi o erro médio absoluto (MAE) que representa a média dos erros absolutos entre os valores de referência (y_i) e os valores previstos (\hat{y}_i) em n observações (1):

$$MAE = \frac{1}{n} \cdot \sum_{i=1}^n |y_i - \hat{y}_i|. \quad (1)$$

O erro médio quadrático (RMSE) também foi utilizado neste estudo, representando o desvio-padrão médio dos valores previstos pelos modelos em relação aos valores de referência (2):

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

Em muitos estudos ([7], [9], [16]–[18], [24]) é ainda aplicado o erro médio percentual absoluto (MAPE) que traduz os erros entre os valores previstos e os valores de referência sob a forma de percentagem, de acordo com a equação (3).

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \quad (3)$$

No entanto, como os sistemas de aquecimento e arrefecimento do edifício do caso de estudo apenas se encontravam ligados em algumas horas do dia durante a semana e durante o fim-de-semana eram muito pouco utilizados, havia muitos registos dos valores de referência que ficavam a 0, levando a que não fosse possível utilizar esta métrica.

Para analisar a distribuição dos erros obtidos entre os valores previstos e os valores de referência, foram utilizados os gráficos *boxplot* (também designados por diagramas de extremos e quartis). Este tipo de gráficos apresenta a distribuição dos erros sob a forma de quartis: o primeiro quartil (Q1) indica que 25% dos erros são inferiores a Q1, o segundo quartil (Q2) indica que 50% dos erros são inferiores a Q2 (também designado por mediana), e o terceiro quartil (Q3) indica que 75% dos erros são inferiores a Q3. Além dos quartis, neste tipo de gráficos são ainda representados os valores atípicos (*outliers*), caracterizados por

Previsão do consumo de energia associado à climatização em edifícios de serviços com base em estratégias de *Machine Learning*

apresentarem um grande desvio em relação aos valores restantes. Na Figura 3.5 está ilustrada uma representação de um gráfico *boxplot*, indicando os diferentes quartis do gráfico.

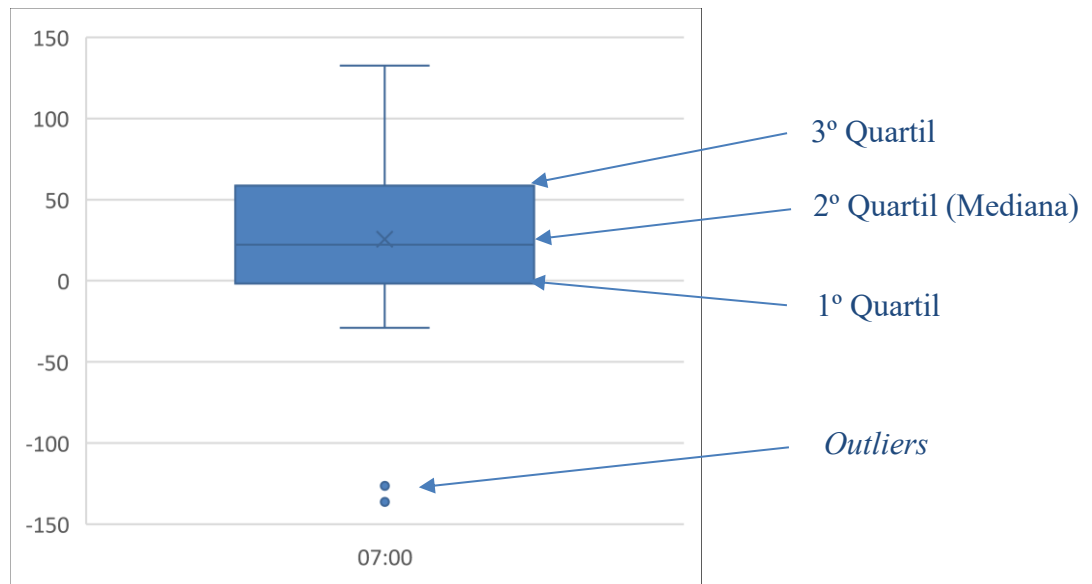


Figura 3.5. Representação de um gráfico *boxplot*

3.5. Análise de correlação

De modo a tentar perceber *a priori* quais as variáveis mais influentes na previsão das necessidades de aquecimento e arrefecimento do edifício em estudo, foi efetuada uma análise de correlação entre as variáveis consideradas e os dados de saída em cada período, considerados como *outputs*.³

3.5.1. Período de aquecimento

A Tabela 3.2 apresenta os fatores de correlação entre as várias variáveis e os dados de saída associados ao período de aquecimento.

Tabela 3.2. Análise de correlação dos dados no período de aquecimento

	Outputs [kW]	Input 1 [kW] ⁴	Input 2 [kW] ⁵	Temperatura [°C]	Ocupação [kW]	Vento [m/s]	Radiação Direta [kW]	Radiação Difusa [kW]	Humidade Relativa [%]
Outputs [kW]	1	0,716	0,749	-0,326	0,251	0,190	-0,040	0,003	-0,314
Input 1 [kW] ⁴	0,716	1	0,749	-0,184	0,246	0,161	0,067	0,061	-0,258
Input 2 [kW] ⁵	0,749	0,749	1	-0,188	0,234	0,167	0,059	0,043	-0,178
Temperatura [°C]	-0,326	-0,184	-0,188	1	0,350	0,078	0,352	0,397	0,379
Ocupação [kW]	0,251	0,246	0,234	0,350	1	0,275	0,478	0,582	-0,097
Vento [m/s]	0,190	0,161	0,167	0,078	0,275	1	0,180	0,211	-0,148
Radiação Direta [kW]	-0,040	0,067	0,059	0,352	0,478	0,180	1	0,685	-0,170
Radiação Difusa [kW]	0,003	0,061	0,043	0,397	0,582	0,211	0,685	1	-0,099
Humidade Relativa [%]	-0,314	-0,258	-0,178	0,379	-0,097	-0,148	-0,170	-0,099	1

As cores mais alaranjadas indicam uma correlação positiva entre as variáveis, ou seja, se uma das variáveis aumenta, a outra também aumenta e vice-versa, sendo que a diagonal principal da matriz da Tabela 3.2 indica a correlação de cada variável com ela própria, daí ter o fator de correlação 1. Analisando a correlação entre as variáveis consideradas como variáveis de entrada e os *outputs*, os históricos de consumo apresentam uma correlação positiva bastante alta com os valores a ser previstos, como era expectável. As células preenchidas a azul indicam uma correlação negativa, significando que se uma aumenta, a outra tende a diminuir e vice-versa. A temperatura exterior e a humidade relativa são as

³ A análise de correlação apenas permite identificar relações lineares entre as diferentes variáveis e os valores a prever podendo, no entanto, existir relações não-lineares que este tipo de análises não consegue detetar.

⁴ Variável correspondente aos históricos de potência de aquecimento 14 dias antes do dia a ser previsto.

⁵ Variável correspondente aos históricos de potência de aquecimento 7 dias antes do dia a ser previsto.

variáveis que apresentam uma correlação negativa maior com os valores a ser previstos, apesar de, ainda assim, não ser uma correlação muito forte.

3.5.2. Período de arrefecimento

A Tabela 3.3 apresenta os fatores de correlação entre as variáveis consideradas para o período de arrefecimento e os valores a ser previstos também nesse período.

Tabela 3.3. Análise de correlação dos dados associados ao período de arrefecimento

	Outputs [kW]	Input 1 [kW] ⁶	Input 2 [kW] ⁷	Temperatura [°C]	Ocupação [kW]	Vento [m/s]	Radiação Direta [kW]	Radiação Difusa [kW]	Humidade Relativa [%]
Outputs [kW]	1	0,889	0,891	0,672	0,747	0,211	0,582	0,567	0,066
Input 1 [kW] ⁶	0,889	1	0,897	0,612	0,683	0,202	0,538	0,536	-0,074
Input 2 [kW] ⁷	0,891	0,897	1	0,602	0,728	0,219	0,544	0,554	-0,049
Temperatura [°C]	0,672	0,612	0,602	1	0,392	0,301	0,520	0,495	0,003
Ocupação [kW]	0,747	0,683	0,728	0,392	1	0,156	0,540	0,651	-0,029
Vento [m/s]	0,211	0,202	0,219	0,301	0,156	1	0,230	0,245	0,004
Radiação Direta [kW]	0,582	0,538	0,544	0,520	0,540	0,230	1	0,648	-0,236
Radiação Difusa [kW]	0,567	0,536	0,554	0,495	0,651	0,245	0,648	1	-0,151
Humidade Relativa [%]	0,066	-0,074	-0,049	0,003	-0,029	0,004	-0,236	-0,151	1

Também no período de arrefecimento se verificou uma correlação positiva bastante alta entre os históricos de potência de arrefecimento 7 e 14 dias antes do dia a ser previsto e os *outputs*. O perfil de ocupação do edifício, a temperatura registada no exterior e as radiações direta e difusa também apresentaram uma correlação positiva alta com os valores a ser previstos no período de arrefecimento.

⁶ Variável correspondente aos históricos de potência de arrefecimento 14 dias antes do dia a ser previsto.

⁷ Variável correspondente aos históricos de potência de arrefecimento 7 dias antes do dia a ser previsto.

3.5.3. Influência da temperatura exterior nas necessidades de aquecimento e arrefecimento

As potências de aquecimento e arrefecimento de um edifício são condicionadas pela variação de temperatura no exterior do edifício, sendo o principal objetivo a manutenção do conforto térmico dos ocupantes no seu interior.

Assim, foi estudada a influência da variação da temperatura exterior nos consumos das potências de aquecimento e arrefecimento no edifício do caso de estudo através de gráficos de dispersão. A relação entre a variação da temperatura exterior e as necessidades de aquecimento do edifício consideradas neste estudo como valores de referência está representada no gráfico de dispersão da Figura 3.6 ⁸.

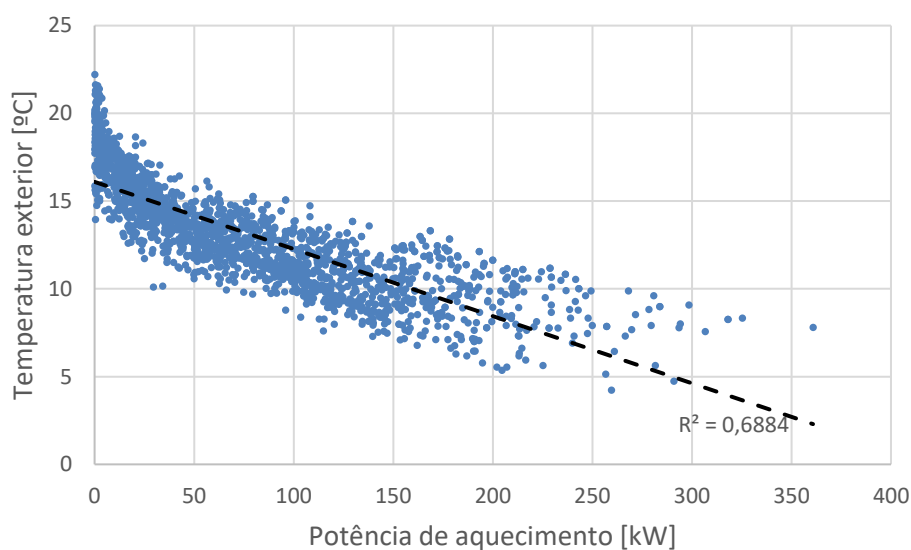


Figura 3.6. Relação entre a temperatura exterior e a potência de aquecimento de referência

Como verificado na Tabela 3.2, os resultados apresentados na Figura 3.6 corroboram a correlação negativa existente entre a temperatura exterior e a potência de aquecimento, ou seja, à medida que a temperatura no exterior do edifício diminui, as necessidades de aquecimento no interior do edifício tendem a aumentar.

A Figura 3.7 apresenta o gráfico de dispersão que relaciona a variação da temperatura exterior com as necessidades de arrefecimento do edifício em estudo.⁸

⁸ Apenas foram considerados os registos de potência de aquecimento e arrefecimento não-nulos.

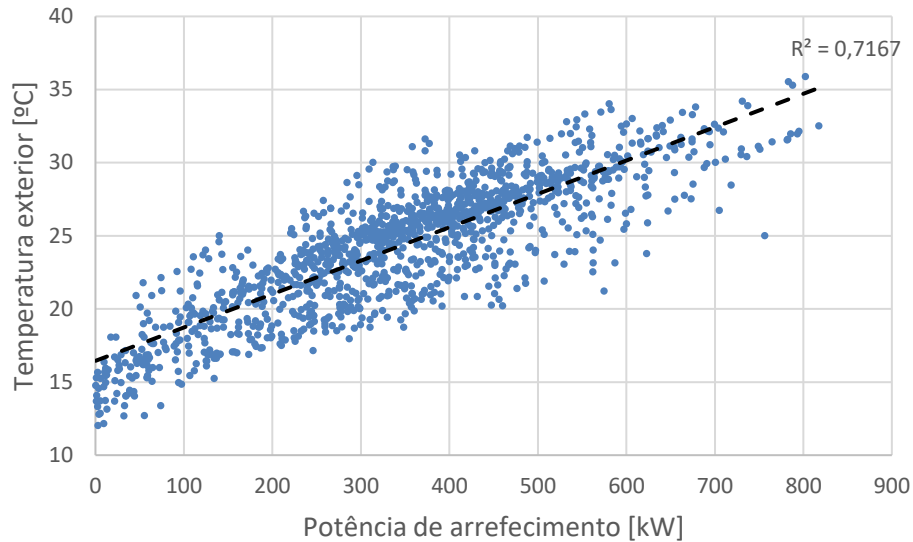


Figura 3.7. Relação entre a temperatura exterior e a potência de arrefecimento de referência

No período de arrefecimento, a temperatura exterior também assume um papel importante nas necessidades de arrefecimento do edifício, como se comprova não só no coeficiente de correlação apresentado na Tabela 3.3, como também na Figura 3.7. À medida que a temperatura aumenta, as necessidades de arrefecimento também tendem a aumentar, confirmando a correlação positiva bastante alta existente entre estas duas variáveis.

Esta página foi intencionalmente deixada em branco

4. Apresentação e discussão de resultados

Nesta seção são apresentados os resultados dos modelos de previsão considerados de forma a avaliar a eficácia destes na previsão das necessidades de aquecimento e arrefecimento do edifício para o dia seguinte. Dado o elevado número de variáveis consideradas para alimentar os modelos, foi necessário proceder a um processo de triagem inicial de forma a estudar quais as variáveis mais influentes para a previsão das necessidades de aquecimento e arrefecimento do edifício. Após essa primeira fase, os valores previstos a partir da melhor combinação de parâmetros de entrada para cada modelo estudado são apresentados e comparados com os valores tidos como referência. De seguida, é avaliada a distribuição dos erros obtidos para cada modelo. Por fim, as melhores combinações de cada modelo são comparadas para perceber qual o modelo mais eficaz para prever as necessidades de aquecimento e arrefecimento.

4.1. Período de aquecimento

4.1.1. Caracterização dos modelos testados

Na Tabela 4.1, estão caracterizadas as diferentes combinações de variáveis testadas nos modelos de previsão para o período de aquecimento.

Tabela 4.1. Combinações de parâmetros testadas para o período de aquecimento

	H1	H2	H3	H4	H5	H6
Potência de aquecimento 7 dias antes [kW]	X		X	X	X	X
Potência de aquecimento 14 dias antes [kW]	X		X	X	X	X
Temperatura exterior [°C]		X	X	X	X	X
Velocidade do vento [m/s]		X				
Radiação solar direta [kW]				X		
Radiação solar difusa [kW]					X	
Humidade relativa [%]		X				X

4.1.2. ANN - Seleção de variáveis de entrada

Nesta primeira fase é efetuada uma “triagem” das variáveis consideradas neste estudo de modo a avaliar quais as variáveis mais influentes na previsão das necessidades de aquecimento. Para isso, foram testadas várias combinações de variáveis (representadas na Tabela 4.1) para alimentar os modelos de previsão e, através das métricas de erro resultantes (MAE e RMSE) para o subconjunto de teste, é selecionada a melhor combinação, i.e., a combinação com as métricas de erro mais baixas no subconjunto de teste.

Na Figura 4.1, estão representadas as métricas de erro associadas às diferentes combinações de modelos testadas, utilizando os modelos ANN.

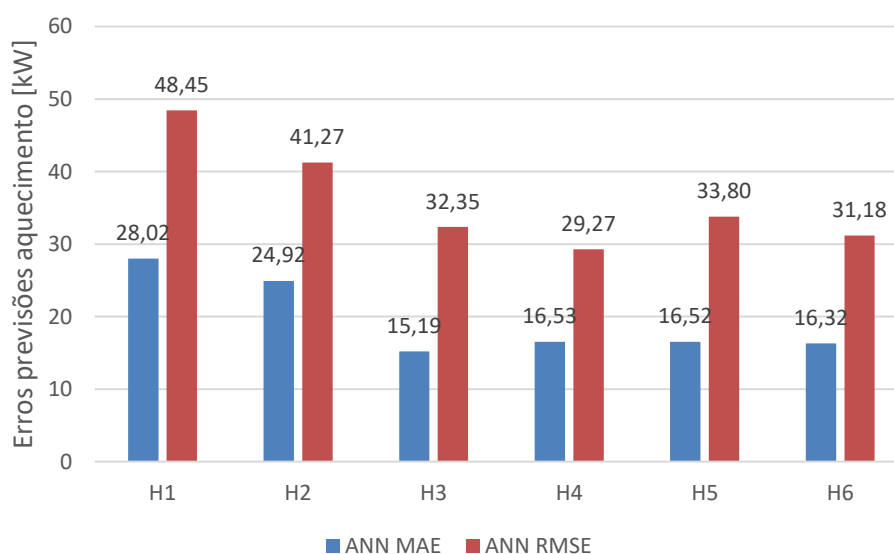


Figura 4.1. Métricas de erro para as combinações testadas no período de aquecimento utilizando os modelos ANN

Analisando a Figura 4.1, observa-se que, com base no MAE, a combinação mais eficaz é a combinação H3, baseada em históricos de potência (7 e 14 dias antes do dia a ser previsto) e temperatura exterior. No entanto, uma vez que o RMSE dá mais ênfase aos erros mais elevados, este foi o mais preponderante na escolha da combinação mais eficaz para previsão das necessidades de aquecimento do edifício em estudo. Assim, a combinação [H4](#), que inclui históricos de potência (7 e 14 dias antes do dia a prever), temperatura exterior e radiação direta, foi a selecionada com base no RMSE. As combinações [H5](#) e [H6](#), além da combinação [H3](#) (já referida acima), proporcionam também métricas de erro baixas.

4.1.3. ANN - Comparação dos valores previstos com a referência

Após ser selecionada a combinação mais eficaz utilizando os modelos ANN, é efetuada uma comparação entre os valores previstos resultantes do modelo ANN [H4](#) com os valores de referência, tendo como base os respetivos diagramas de carga para um dia completo do subconjunto de teste ⁹ (Figura 4.2).

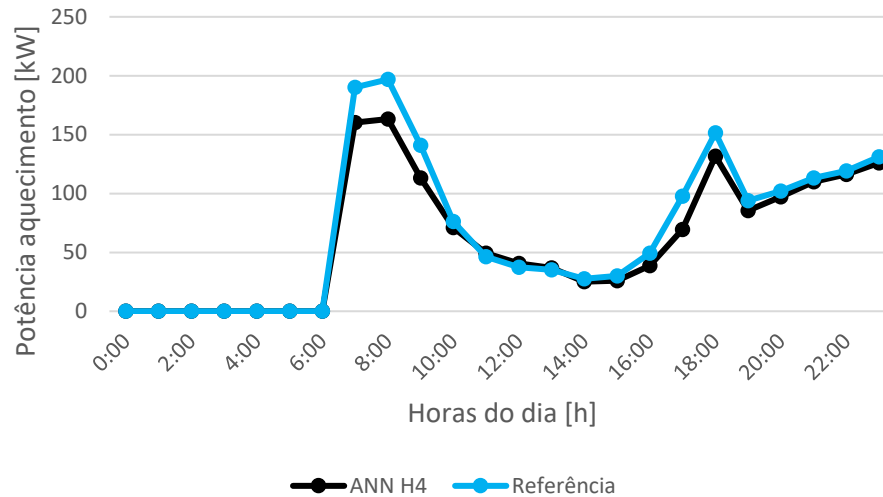


Figura 4.2. Comparação entre as previsões do modelo ANN H4 e os valores de referência

Pode-se observar na Figura 4.2 que o perfil de consumo ao longo do dia dos valores previstos pelo modelo ANN [H4](#) acompanha muito bem o perfil dos valores tidos como referência neste estudo. As variações mais significativas ocorrem nas horas de maior consumo. No geral, a tendência verificada pelo modelo ANN [H4](#) é de prever os valores abaixo do referenciado.

⁹ O diagrama de carga apresentado no período de aquecimento é referente ao dia 19 de dezembro (quinta-feira).

4.1.4. ANN - Análise da distribuição dos erros

De forma a entender melhor as diferenças entre as previsões obtidas pelo modelo ANN [H4](#) e os valores de referência, recorre-se a um *boxplot* para avaliar a distribuição dos erros em cada hora do dia no subconjunto de teste, representado na Figura 4.3.

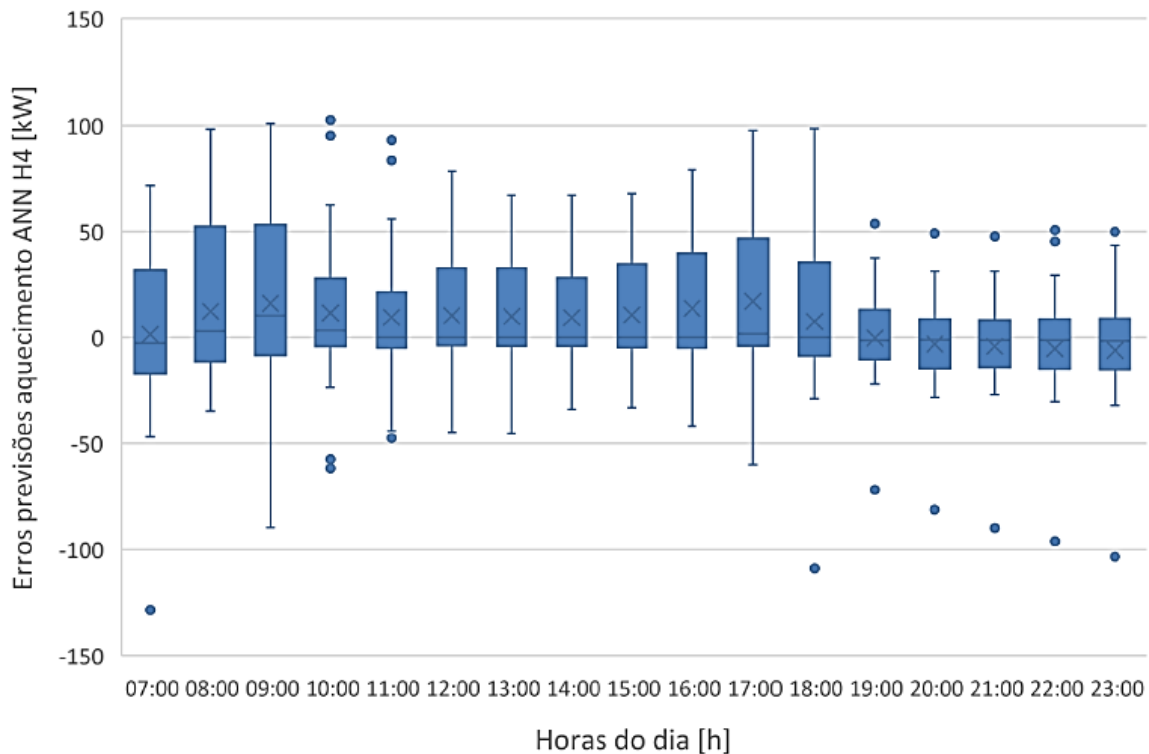


Figura 4.3. Boxplot de distribuição dos erros entre o modelo ANN H4 e a referência no subconjunto de teste

O gráfico *boxplot* da Figura 4.3 indica que, no geral, a mediana dos erros em todas as horas se encontra próxima de 0, com as maiores variações a ocorrerem nas horas coincidentes com os picos de consumo, evidenciados também na Figura 4.2. A tendência dos *boxplots* em cada hora se localizarem maioritariamente acima de 0 indicia que os valores de referência tendem a ser superiores aos valores previstos pelo modelo ANN [H4](#).

4.1.5. SVM - Seleção de variáveis de entrada

Na Figura 4.4 estão representadas as métricas de erro relativas às combinações de variáveis consideradas no período de aquecimento, utilizando os modelos SVM.

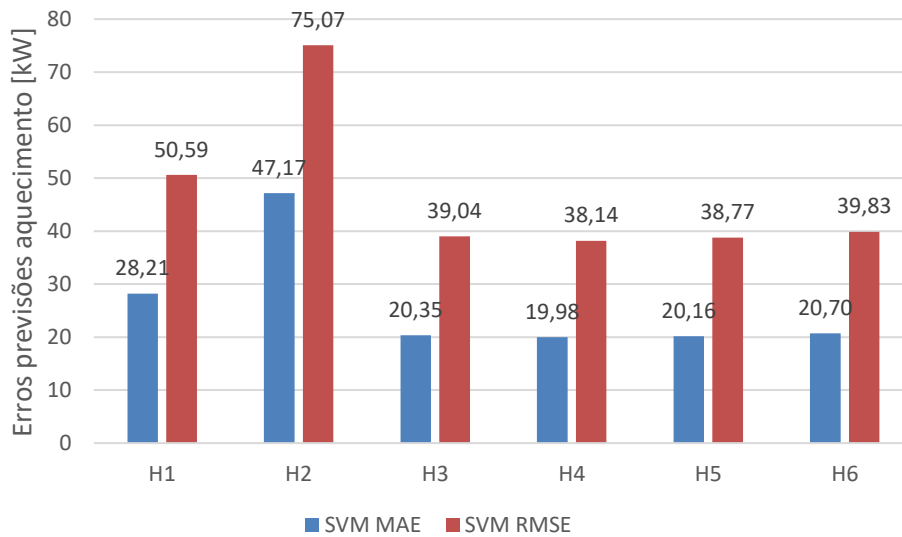


Figura 4.4. Métricas de erro para as combinações testadas no período de aquecimento utilizando os modelos SVM

A Figura 4.4 indica que, utilizando os modelos SVM, todas as combinações testadas apresentam erros semelhantes, excetuando apenas os modelos SVM [H1](#) e [H2](#), que se baseiam apenas nos históricos de consumo e apenas em variáveis exógenas, respectivamente. Ainda assim, e apesar da diferença ser muito residual, a combinação com as métricas de erro mais baixas voltou a ser a combinação [H4](#), tal como tinha ocorrido para os modelos ANN.

4.1.6. SVM - Comparação dos valores previstos com a referência

A Figura 4.5 representa os diagramas de carga dos valores previstos pelo modelo SVM [H4](#) para um dia completo no subconjunto de teste, comparando com os valores de referência para esse mesmo dia. ⁹

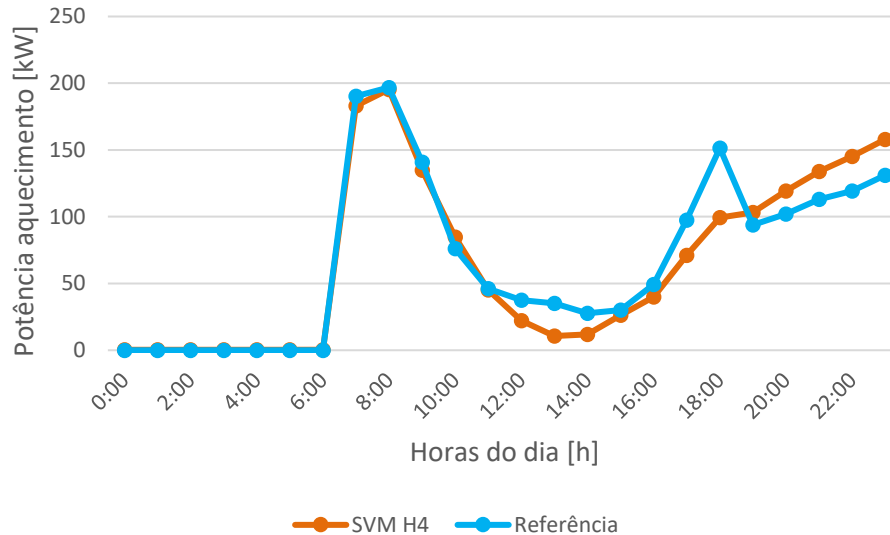


Figura 4.5. Comparação entre as previsões do modelo SVM H4 e os valores de referência

Observa-se na Figura 4.5 que, no período da manhã, o diagrama de carga referente aos valores previstos pelo modelo SVM [H4](#) é bastante semelhante ao diagrama de carga de referência, sendo que as maiores diferenças ocorrem mais para o final do dia. Tal como no modelo ANN [H4](#), também aqui se verifica uma tendência para os modelos em preverem os valores abaixo da referência, com a exceção das últimas horas do dia, em que essa tendência se inverte.

4.1.7. SVM - Análise da distribuição dos erros

A Figura 4.6 mostra o *boxplot* referente à distribuição dos erros entre os valores de referência e os valores previstos pelo modelo SVM [H4](#) no subconjunto de teste.

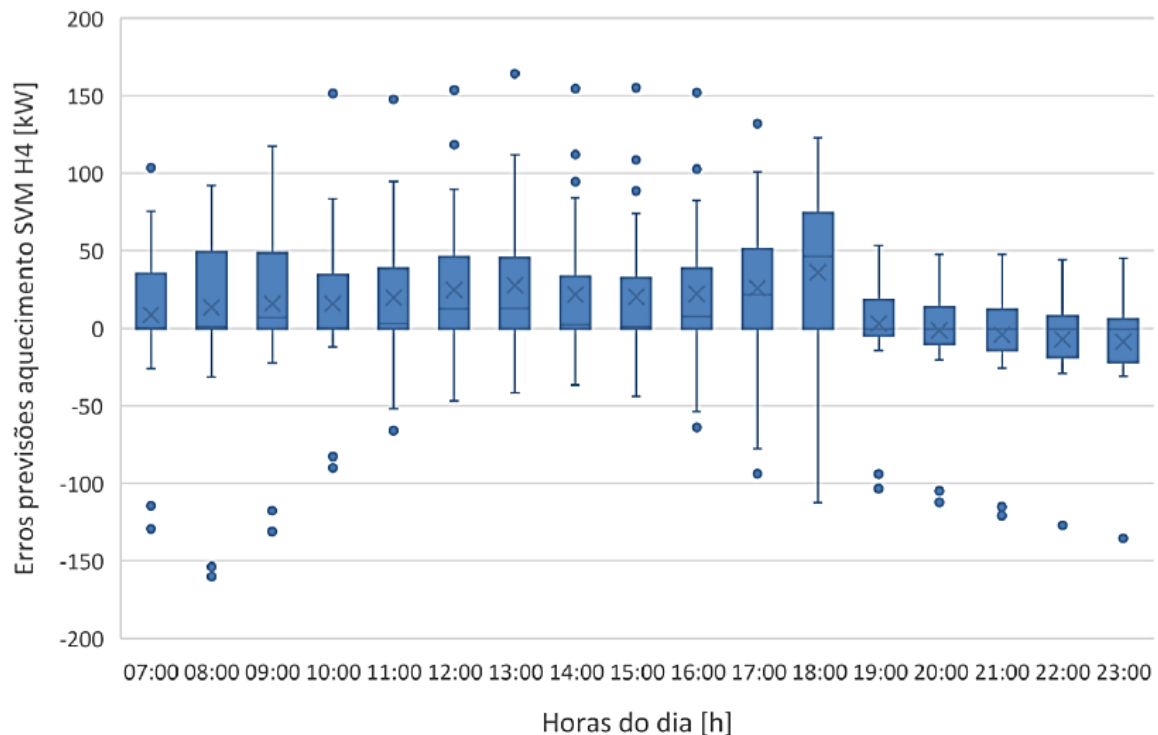


Figura 4.6. *Boxplot* de distribuição dos erros entre o modelo SVM H4 e a referência no subconjunto de teste

Tal como evidenciado na Figura 4.5 com a análise do diagrama de carga do modelo SVM [H4](#), o *boxplot* representado na Figura 4.6 indicia também a tendência para os valores de referência serem superiores aos valores previstos por este modelo, tendência essa que se inverte ligeiramente nas últimas horas do dia. As maiores amplitudes dos erros voltam a acontecer nas horas em que o consumo aumenta, ou seja, da parte da manhã e depois entre as 17h e as 19h.

4.1.8. SVM + SA - Seleção de variáveis de entrada

A Figura 4.7 indica as métricas de erro referentes às combinações apresentadas na Tabela 4.1, utilizando os modelos SVM com otimização possibilitada a partir da meta-heurística SA.

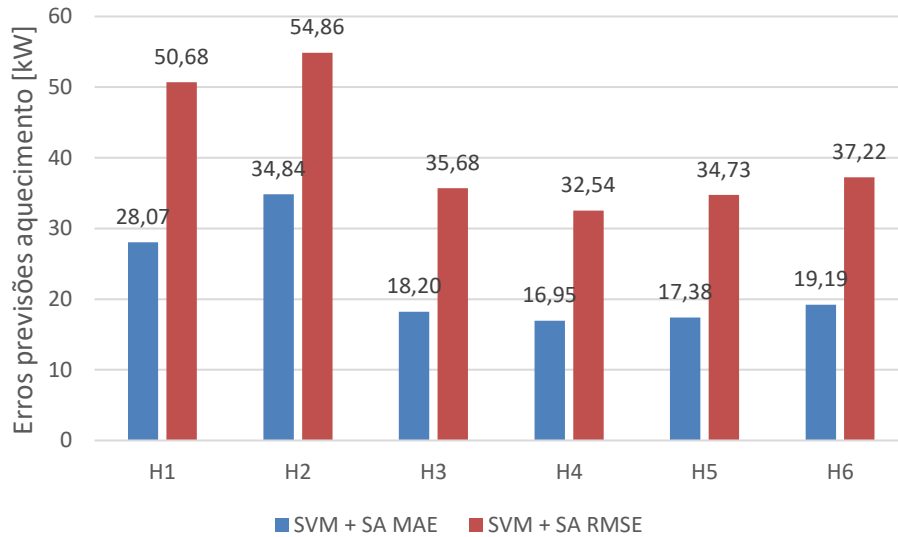


Figura 4.7. Métricas de erro para as combinações testadas no período de aquecimento utilizando os modelos SVM+SA

A otimização dos parâmetros SVM através da meta-heurística SA está refletida nos resultados das métricas de erro apresentados na Figura 4.7. Registrou-se uma melhoria nas métricas de erro de todas as combinações testadas no período de aquecimento, quando comparadas com as métricas de erro no modelo SVM sem a utilização do SA (Figura 4.4). Uma vez mais, a combinação [H4](#) registou as métricas de erro mais baixas, utilizando o modelo SVM + SA.

4.1.9. SVM + SA - Comparação dos valores previstos com a referência

O diagrama de carga resultante dos valores previstos pelo modelo SVM+SA [H4](#) está representado na Figura 4.8, juntamente com o diagrama de carga dos valores de referência para um dia completo do subconjunto de teste.⁹

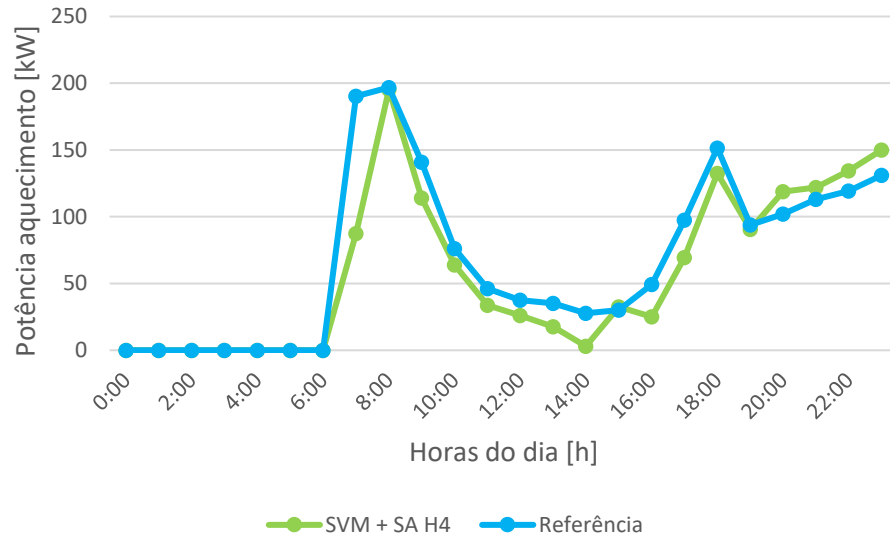


Figura 4.8. Comparação entre as previsões do modelo SVM+SA H4 e os valores de referência

O diagrama de carga originado pelo modelo SVM+SA [H4](#), representado na Figura 4.8, apresenta um perfil bastante semelhante aos valores de referência, apenas com uma variação mais evidente às 07:00h. Tal como no modelo SVM [H4](#), também com a otimização do SA se verifica que os valores previstos tendem a ser inferiores à referência, com a inversão dessa tendência nas últimas horas do dia.

4.1.10. SVM + SA - Análise da distribuição dos erros

A Figura 4.9 apresenta a distribuição dos erros associados às previsões do modelo SVM+SA [H4](#) em relação aos valores de referência.

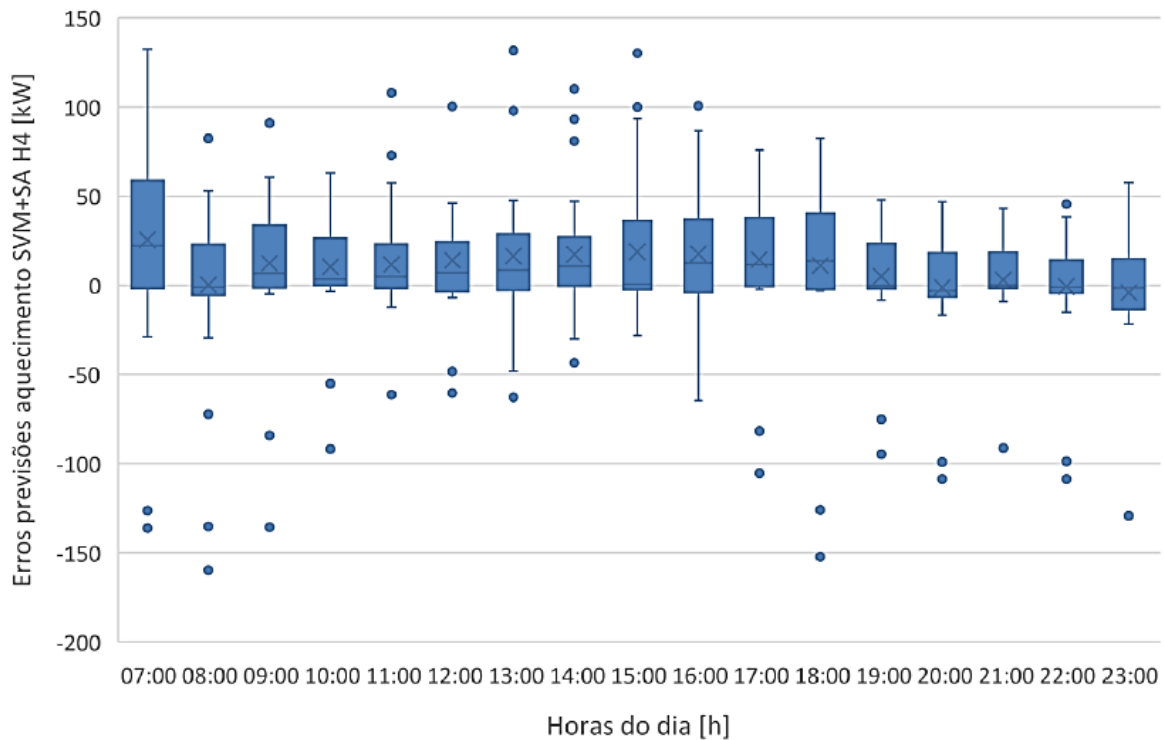


Figura 4.9. Boxplot de distribuição dos erros entre o modelo SVM+SA H4 e a referência no subconjunto de teste

A Figura 4.9 mostra uma melhoria global dos erros em relação ao *boxplot* da Figura 4.6, sem a otimização do SA. Tal como o verificado na Figura 4.8, a maior amplitude de erros ocorre às 07:00h. Nas outras horas, a amplitude de erros é mais baixa e confirma a tendência também do modelo em prever os consumos abaixo dos valores de referência.

4.1.11. Análise comparativa

Após a apresentação dos resultados para cada modelo considerado, segue-se uma análise comparativa de modo a perceber qual dos modelos considerados é mais eficaz em prever os consumos energéticos associados ao período de aquecimento.

A Figura 4.10 apresenta o diagrama de carga das melhores combinações de cada modelo (ANN [H4](#), SVM [H4](#) e SVM+SA [H4](#)) para uma semana completa do subconjunto de teste.¹⁰

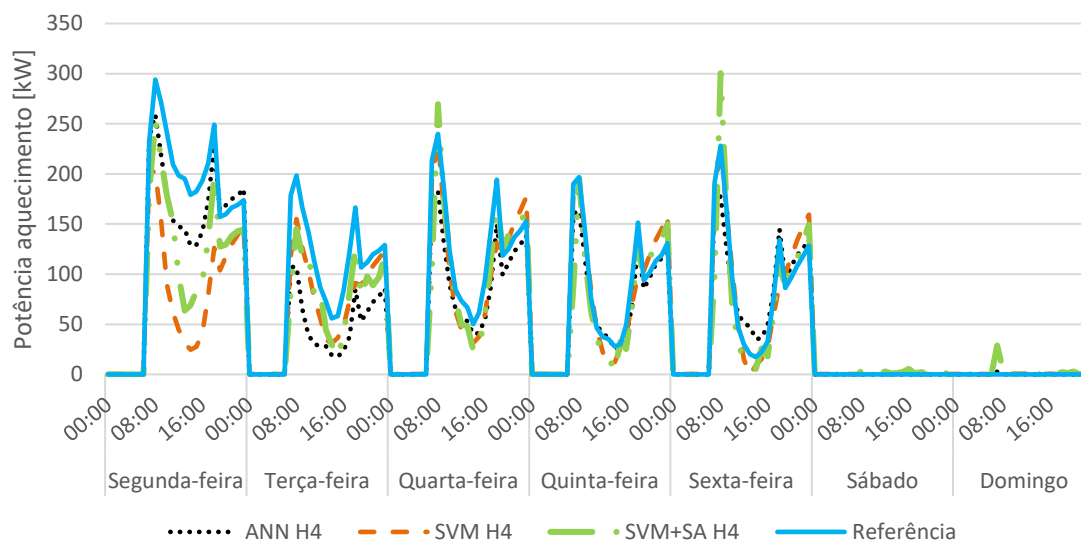


Figura 4.10. Diagrama de carga semanal para os melhores modelos de previsão de consumos no período de aquecimento

Analisando os perfis de consumo de cada modelo, aquele que globalmente se aproxima mais dos valores de referência é o modelo ANN [H4](#). Com este diagrama de carga semanal, também se podem observar as variações de consumo de um dia para o outro ao longo de uma típica semana de trabalho nos valores de referência e como as previsões dos modelos conseguem acompanhar essas variações. Isto deve-se, em parte, à contribuição dos históricos de consumo (7 e 14 dias antes do dia a ser previsto), o que permite aos modelos captar as tendências dos consumos em cada dia da semana.

¹⁰ O diagrama de carga semanal no período de aquecimento é referente à semana de 15 a 22 de dezembro (segunda-feira a domingo).

4.1.12. Previsão vs Referência

De forma a avaliar todos os dados previstos pelos três melhores modelos, é utilizado um gráfico de dispersão para comparar os valores previstos por cada modelo com os respetivos valores de referência (Figura 4.11).

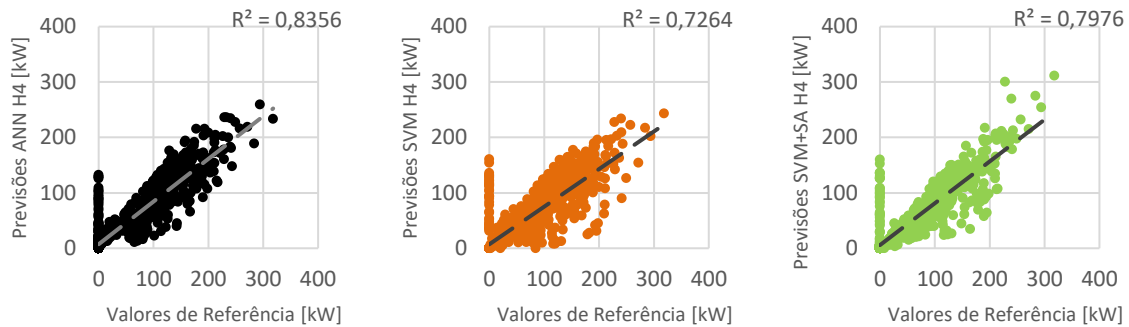


Figura 4.11. Gráficos de dispersão das previsões dos modelos ANN H4, SVM H4 e SVM+SA H4 comparadas com os valores de referência no subconjunto de teste do período de aquecimento

Os gráficos de dispersão da Figura 4.11 revelam que os modelos de previsão no período de aquecimento não têm um fator de correlação muito alto, o que pode ser explicado em parte pelo desvio da linha de tendência nos três gráficos em relação à bissetriz, indicando que os valores das previsões são tendencialmente inferiores aos valores de referência. Outro fator que pode contribuir para este desvio é o facto de o período de Natal coincidir com o período de teste, com os modelos a considerarem esse período como dias de consumo normais. Ainda assim, confirma-se que o modelo ANN [H4](#) é o modelo que consegue prever melhor os consumos associados à climatização no período de aquecimento.

4.1.13. Distribuição dos erros para os modelos de previsão mais precisos

A Figura 4.12 apresenta a comparação da distribuição dos erros dos modelos ANN [H4](#), SVM [H4](#) e SVM+SA [H4](#).

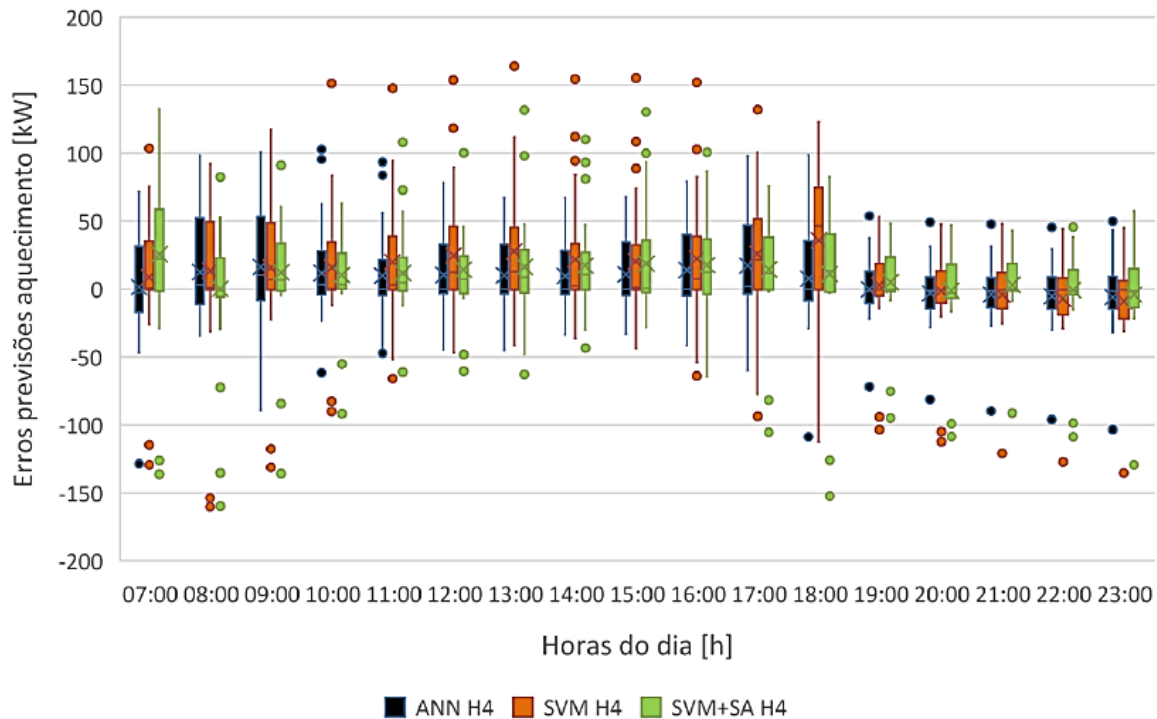


Figura 4.12. Distribuição dos erros dos melhores modelos de previsão para o subconjunto de teste do período de aquecimento

A Figura 4.12 indica que, no geral, a distribuição dos erros é bastante uniforme em cada hora para os três modelos considerados. O modelo ANN [H4](#), além de ter a menor amplitude de erros na maioria das horas, é o modelo que apresenta menos *outliers*. Outro ponto a considerar é o efeito da aplicação do *simulated annealing* nos modelos SVM, resultando numa diminuição considerável na amplitude de erros do modelo SVM [H4](#) para o modelo SVM+SA [H4](#).

4.2. Período de arrefecimento

4.2.1. Caracterização dos modelos testados

Na Tabela 4.2 estão caracterizadas as diferentes combinações de parâmetros testadas para o período de arrefecimento.

Tabela 4.2. Combinações de parâmetros testadas para o período de arrefecimento

	C1	C2	C3	C4	C5	C6
Potências de arrefecimento 7 dias antes [kW]	X		X	X	X	X
Potências de arrefecimento 14 dias antes [kW]	X		X	X	X	X
Temperatura Exterior [°C]		X	X	X	X	X
Perfil de Ocupação [kW]		X		X	X	
Humidade Relativa [kW]		X			X	X

4.2.2. ANN – Seleção de variáveis de entrada

Tal como para o período de aquecimento, a seleção de variáveis de entrada é a primeira fase a ser implementada de modo a verificar qual a combinação de parâmetros que apresenta as métricas de erro mais baixas. A Figura 4.13 apresenta os resultados das métricas de erro associadas às diferentes combinações testadas (Tabela 4.2) no subconjunto de teste do período de arrefecimento utilizando os modelos ANN.

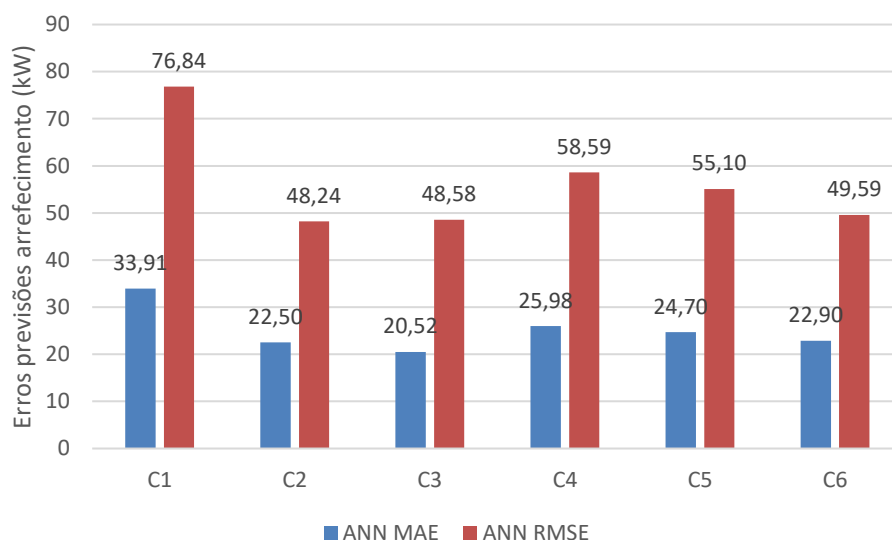


Figura 4.13. Métricas de erro para as combinações testadas no período de arrefecimento utilizando os modelos ANN

A Figura 4.13 indica que, analisando apenas o MAE, a combinação que melhor prevê as necessidades de arrefecimento através dos modelos ANN é a [C3](#), baseada apenas em históricos de potência (7 e 14 dias antes do dia a ser previsto) e temperatura exterior. No entanto, uma vez que o RMSE penaliza mais os erros mais elevados, é também muito importante analisar essa métrica de erro. Pelo RMSE, a combinação mais eficaz na previsão de necessidades de arrefecimento é a combinação [C2](#) (baseada apenas em variáveis exógenas), apesar de estar muito perto também das combinações [C3](#) e [C6](#).

4.2.3. ANN - Comparação dos valores previstos com a referência

Tendo sido selecionado o modelo ANN [C2](#) como o mais preciso dos modelos ANN na previsão das necessidades de arrefecimento, segue-se a comparação entre os valores previstos por este modelo com os valores de referência, analisando os respetivos diagramas de carga para um dia completo pertencente ao subconjunto de teste, representado na Figura 4.14.¹¹

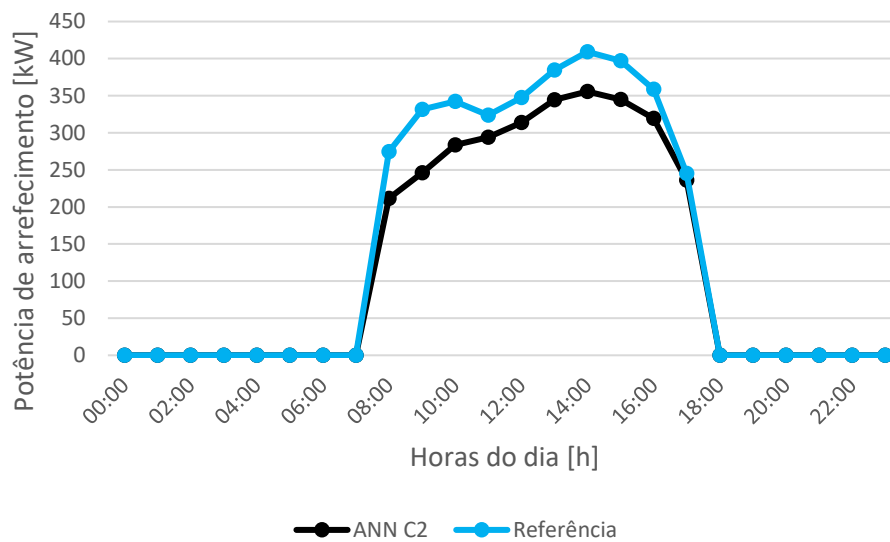


Figura 4.14. Comparação entre as previsões do modelo ANN C2 e os valores de referência

A Figura 4.14 mostra que os valores previstos pelo modelo ANN [C2](#) se situam sempre abaixo dos valores de referência. O perfil de consumo apresenta também algumas diferenças, principalmente nas horas de maior consumo, o que pode em parte ser explicado pelo facto de o modelo ANN [C2](#) não ter em conta os históricos de consumo.

¹¹ O diagrama de carga apresentado no período de arrefecimento é referente ao dia 13 de setembro (sexta-feira).

4.2.4. ANN - Análise da distribuição dos erros

Após a comparação das previsões do modelo ANN [C2](#) com os valores tidos como referência para um dia normal de trabalho do subconjunto de teste, segue-se a análise da distribuição dos erros em todo o subconjunto de teste de acordo com as horas em que os sistemas de arrefecimento estão a trabalhar (desde as 8h até às 18h). O *boxplot* referente à distribuição dos erros do modelo ANN [C2](#) para o subconjunto de teste está representado de seguida na Figura 4.15.

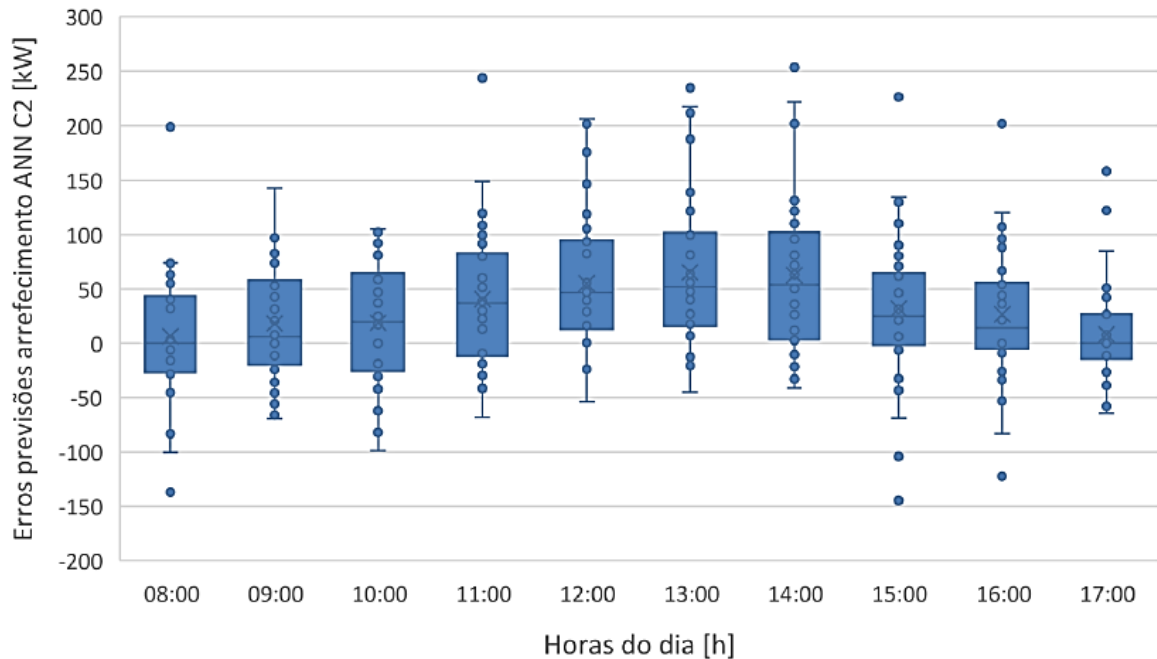


Figura 4.15. *Boxplot* de distribuição dos erros entre o modelo ANN C2 e a referência no subconjunto de teste

O *boxplot* da Figura 4.15 mostra que as horas onde se registam necessidades de arrefecimento mais elevadas correspondem também às maiores variações nos erros e que coincide com os registos de temperatura mais elevados no exterior, o que já é esperado. Tal como verificado na Figura 4.14, confirma-se a tendência de os valores de referência serem superiores aos valores previstos pelo modelo ANN [C2](#), comprovado pelo facto de a maior parte dos erros serem positivos em todas as horas.

4.2.5. SVM - Seleção de variáveis de entrada

Para os modelos SVM sem otimização no período de arrefecimento, estão representadas na Figura 4.16 as métricas de erro associadas às combinações testadas no subconjunto de teste, utilizando estes modelos.

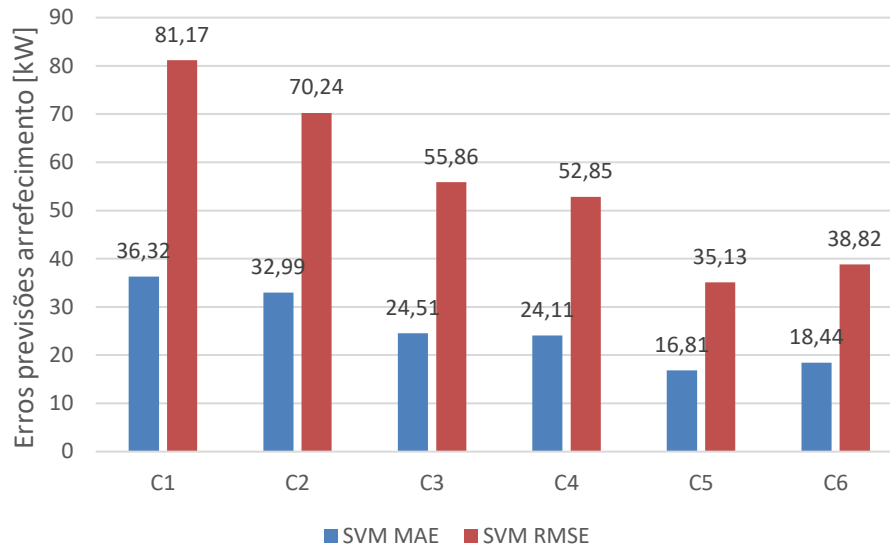


Figura 4.16. Métricas de erro para as combinações testadas no período de arrefecimento utilizando os modelos SVM

Um dos principais pontos a destacar na análise da Figura 4.16 é o facto de, quando comparada com a Figura 4.13, as métricas de erro associadas às combinações [C5](#) e [C6](#) terem baixado muito, mesmo sem a otimização dos parâmetros SVM. A combinação [C4](#) também sofreu uma ligeira melhoria, mas ainda longe das métricas de erro mais baixas, enquanto que as outras combinações ([C1](#), [C2](#) e [C3](#)) pioraram. O modelo SVM [C5](#) foi o que registou as métricas de erro mais baixas, ou seja, o que melhor prevê as necessidades de arrefecimento.

4.2.6. SVM - Comparação dos valores previstos com a referência

Tendo sido selecionado o modelo SVM [C5](#), segue-se a análise do diagrama de carga dos valores previstos por este modelo, em comparação com os valores de referência para um dia completo do subconjunto de teste (Figura 4.17).¹¹

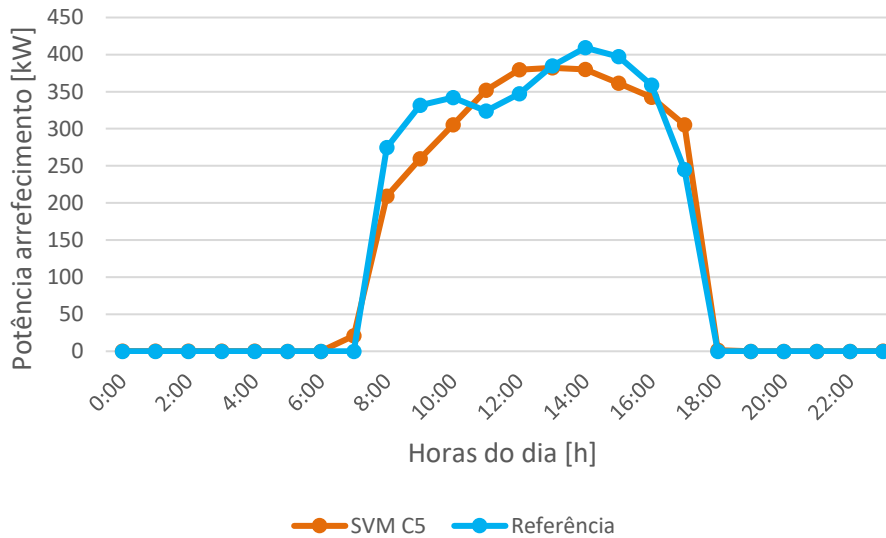


Figura 4.17. Comparação entre as previsões do modelo SVM C5 e os valores de referência

A Figura 4.17 indica que os valores previstos pelo modelo SVM [C5](#) nunca se afastam muito dos valores de referência, apesar do perfil de consumo ter algumas diferenças, estando em algumas horas acima da referência e em outras abaixo. Ainda assim, comparando com o diagrama de carga do modelo ANN [C2](#) representado na Figura 4.14, o modelo SVM [C5](#) aproxima-se mais dos valores de referência para este dia do subconjunto de teste.

4.2.7. SVM - Análise da distribuição dos erros

Na Figura 4.18 está representado o *boxplot* que mostra a distribuição dos erros associados aos valores previstos pelo modelo SVM [C5](#) no subconjunto de teste.

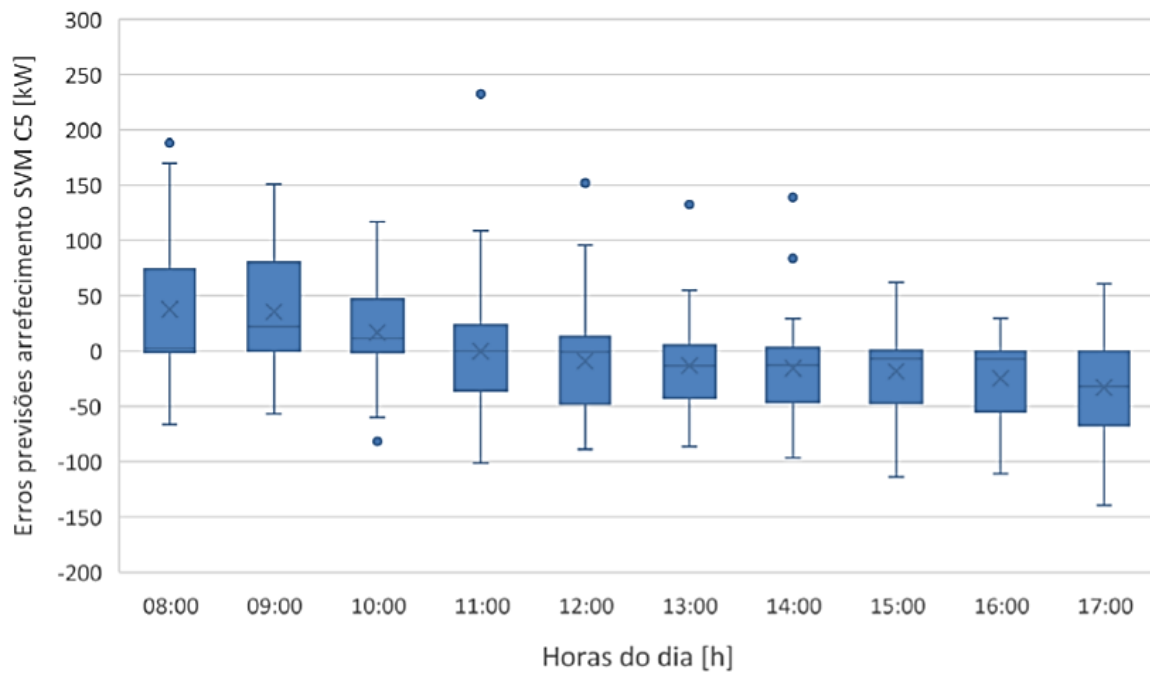


Figura 4.18. *Boxplot* de distribuição dos erros entre o modelo SVM C5 e a referência no subconjunto de teste

O *boxplot* da Figura 4.18 indica que a tendência nas primeiras horas em que os sistemas de arrefecimento estão a trabalhar é de os valores previstos serem inferiores aos valores de referência enquanto que nas restantes horas os valores previstos tendem a ser superiores.

4.2.8. SVM+SA - Seleção de variáveis de entrada

As métricas de erro associadas aos modelos SVM com a otimização através da meta heurística *simulated annealing* estão representadas de seguida na Figura 4.19.

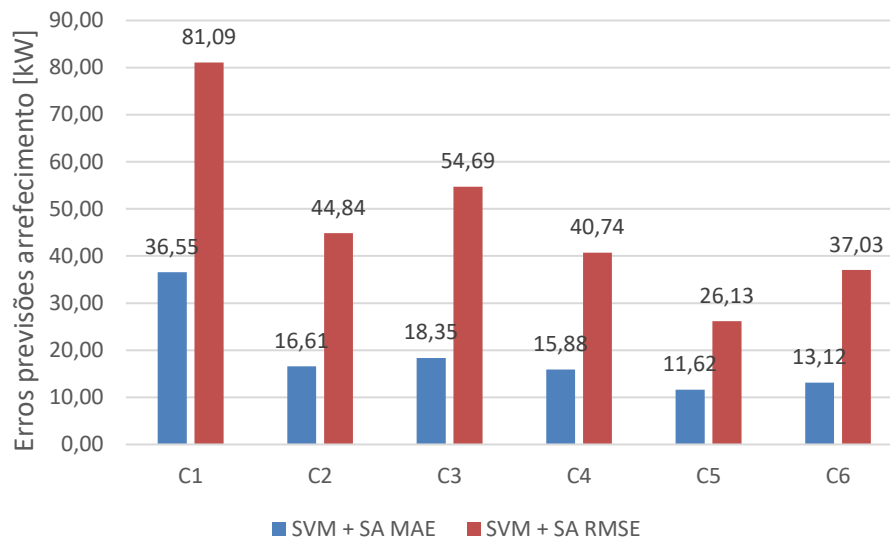


Figura 4.19. Métricas de erro para as combinações testadas no período de arrefecimento utilizando os modelos SVM+SA

Como se esperava, recorrendo ao *simulated annealing* para otimizar os parâmetros SVM, registou-se uma melhoria considerável nas métricas de erro relativas às combinações [C2](#), [C4](#), [C5](#) e [C6](#), comparando com a Figura 4.16 sem a utilização dessa meta-heurística. Tal como nos modelos SVM sem otimização, também a combinação [C5](#) registou as métricas de erro mais baixas, mas com um MAE e um RMSE consideravelmente mais baixo devido à utilização do *simulated annealing*.

4.2.9. SVM + SA - Comparação dos valores previstos com a referência

Como referido na secção 4.2.8, o modelo SVM+SA [C5](#) foi o mais preciso na previsão das necessidades de arrefecimento. A Figura 4.20 representa o diagrama de carga resultante dos valores previstos pelo modelo SVM+SA [C5](#) confrontado com o diagrama de carga dos valores de referência para o mesmo dia no subconjunto de teste.¹¹

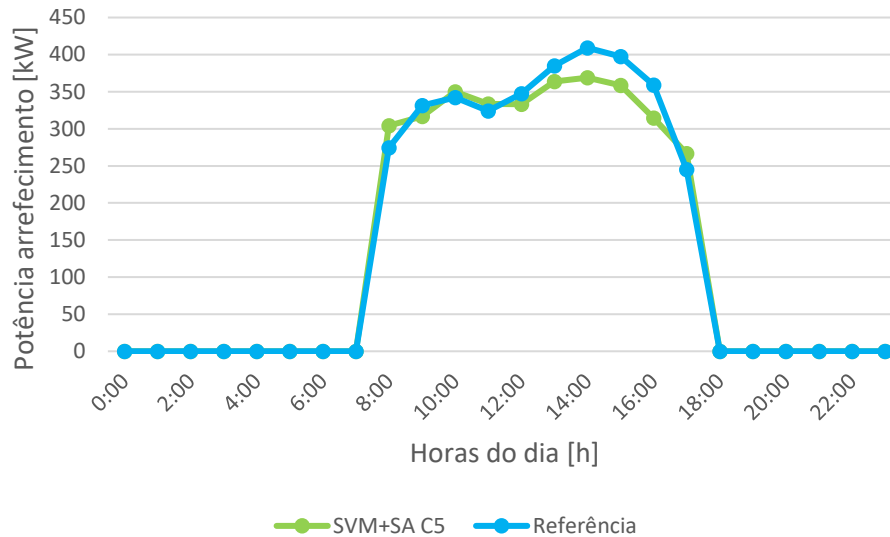


Figura 4.20. Comparação entre as previsões do modelo SVM+SA C5 e os valores de referência

Analisando a Figura 4.20, pode-se constatar a influência que o *simulated annealing* teve nos modelos SVM, ao apresentar um perfil de consumo bastante semelhante à referência. As maiores variações ocorreram entre as 14:00h e as 16:00h, apesar das diferenças não serem muito significativas, comprovando que o modelo SVM+SA [C5](#) é bastante preciso na previsão das necessidades de arrefecimento.

4.2.10. SVM + SA - Análise da distribuição dos erros

A distribuição dos erros associados ao modelo SVM+SA [C5](#) no subconjunto de teste está representada de seguida no boxplot da Figura 4.21.

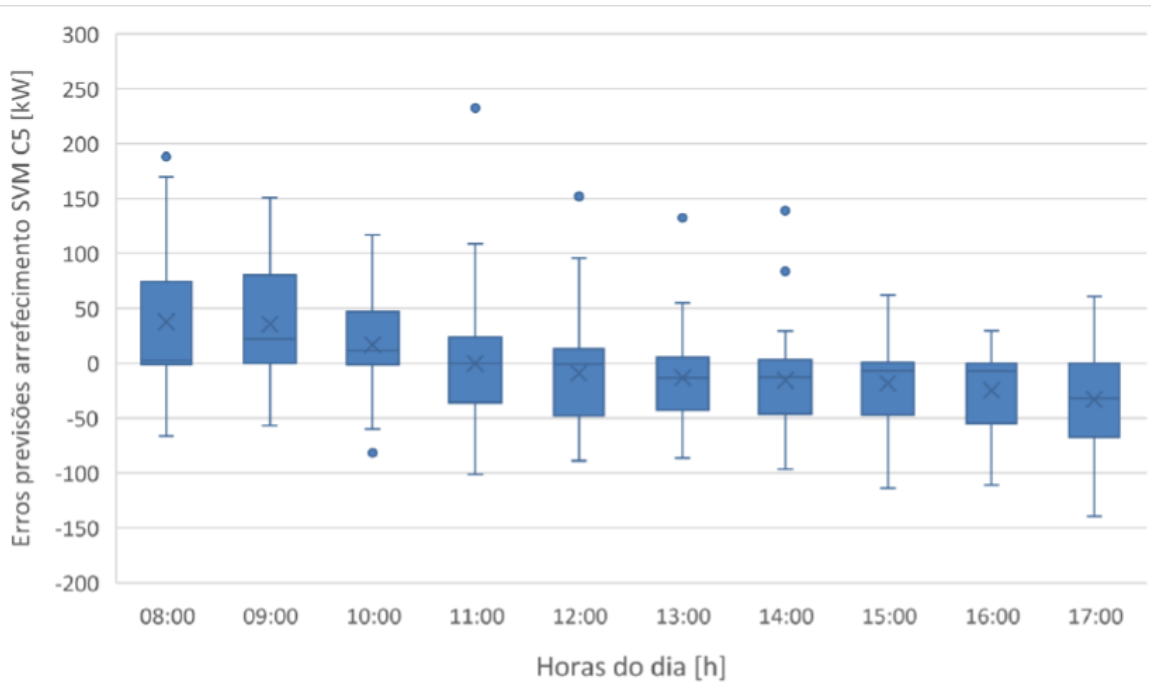


Figura 4.21. *Boxplot* de distribuição dos erros entre o modelo SVM+SA C5 e a referência no subconjunto de teste

A Figura 4.21 evidencia uma distribuição muito mais baixa dos erros para todas as horas em que os sistemas de arrefecimento estão a trabalhar nos dias pertencentes ao subconjunto de teste. No entanto, ao contrário do modelo SVM [C5](#) que, na Figura 4.18, evidencia uma tendência para que as suas previsões das necessidades de arrefecimento sejam superiores aos valores de referência, o modelo SVM+SA [C5](#) tende a prever valores inferiores à referência, exceto às 8:00h e às 17:00h.

4.2.11. Análise comparativa

Após ser selecionada a melhor combinação de variáveis de entrada para cada modelo (ANN, SVM e SVM+SA), segue-se uma análise comparativa desses modelos, através dos diagramas de carga resultantes das previsões de cada modelo selecionado para uma semana completa do subconjunto de teste e comparado com o respetivo diagrama de carga de referência para essa semana (Figura 4.22).¹²

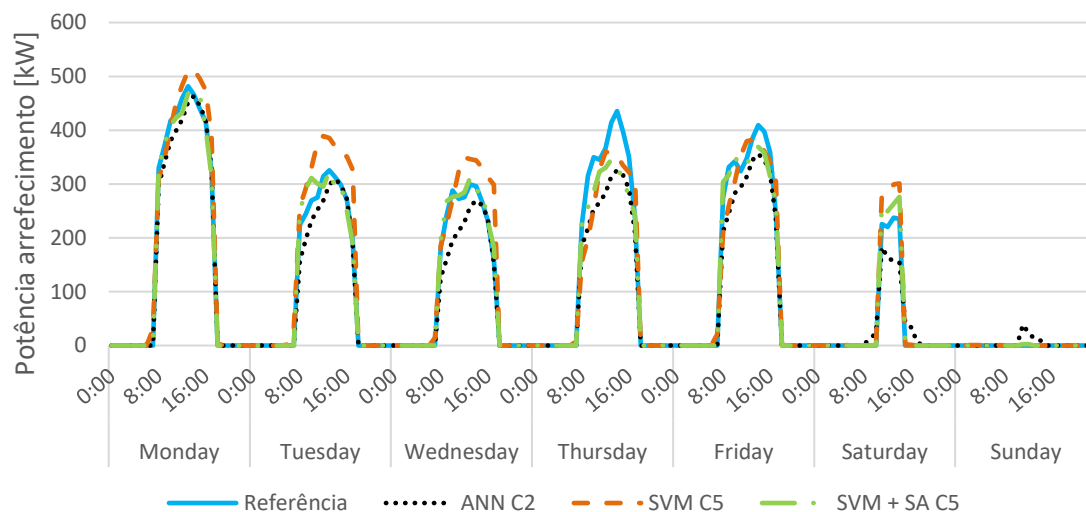


Figura 4.22. Diagrama de carga semanal para os melhores modelos de previsão de consumos no período de arrefecimento

Apesar de todos os modelos apresentarem poucas variações no que diz respeito ao perfil de consumo semanal, o diagrama de carga originado pelo modelo SVM+SA [C5](#) é aquele que melhor se assemelha ao diagrama de carga dos valores de referência em todos os dias da semana. Tal como referido na seção 4.2.3, o facto do modelo ANN [C2](#) não ter em conta os históricos de consumo de 7 e 14 dias antes do dia a ser previsto, leva a que haja alguma disparidade entre os valores previstos por este modelo e os valores de referência.

¹² O diagrama de carga semanal no período de aquecimento é referente à semana de 9 a 15 de setembro (segunda-feira a domingo).

4.2.12. Previsão vs referência

De forma a analisar a relação entre os valores previstos por cada modelo e os valores de referência em todo o subconjunto de teste, estão representados na Figura 4.23 os gráficos de dispersão dos valores previstos por cada modelo relacionados com os valores de referência.

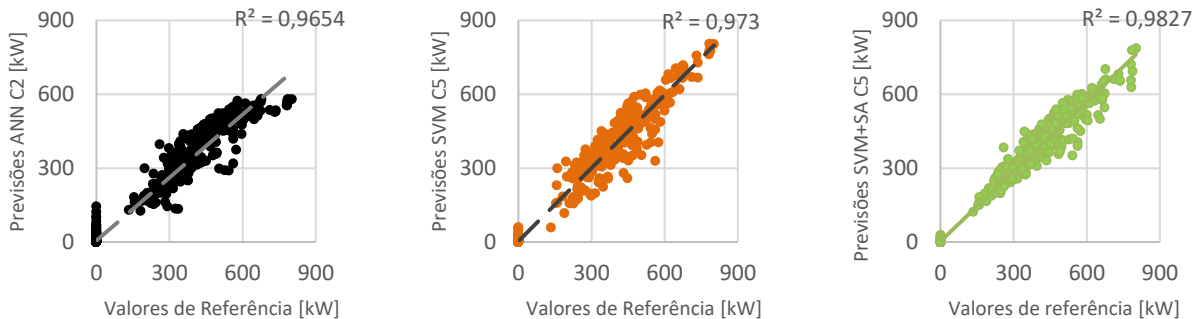


Figura 4.23. Gráficos de dispersão das previsões dos modelos ANN C2, SVM C5 e SVM+SA C5 comparadas com os valores de referência no subconjunto de teste do período de arrefecimento

Os gráficos de dispersão indicam que os 3 modelos seleccionados apresentam um coeficiente de correlação (R^2) muito alto entre os valores previstos e os valores de referência, indicando que as previsões dos modelos estão bastante próximas da referência. No gráfico de dispersão relativo ao modelo ANN [C2](#) é possível observar uma ligeira inclinação da linha de tendência na direção dos valores de referência, corroborando a ideia dos valores previstos por este modelo serem tendencialmente inferiores à referência. Esta tendência pode ser explicada pelo facto deste modelo não ter os históricos de potência como variáveis de entrada para as suas previsões, não permitindo assim distinguir dias de semana dos fins-de-semana. Nos gráficos relativos aos modelos SVM [C5](#) e SVM+SA [C5](#), ambas as linhas de tendência são praticamente coincidentes com a bissetriz dos respetivos gráficos, indicando uma correlação bastante próxima de 1 entre os valores previstos e a referência. No gráfico de dispersão afeto ao modelo SVM+SA [C5](#) a concentração dos pontos está mais próxima da linha de tendência, enquanto que no gráfico relativo ao modelo SVM [C5](#), a dispersão é ligeiramente maior, tornando assim o modelo SVM+SA [C5](#) mais preciso na previsão das necessidades de arrefecimento.

4.2.13. Distribuição dos erros para os modelos de previsão mais precisos

A comparação entre a distribuição dos erros de cada modelo (ANN C2, SVM C5 e SVM+SA C5) para o subconjunto de teste do período de arrefecimento está representada na Figura 4.24.

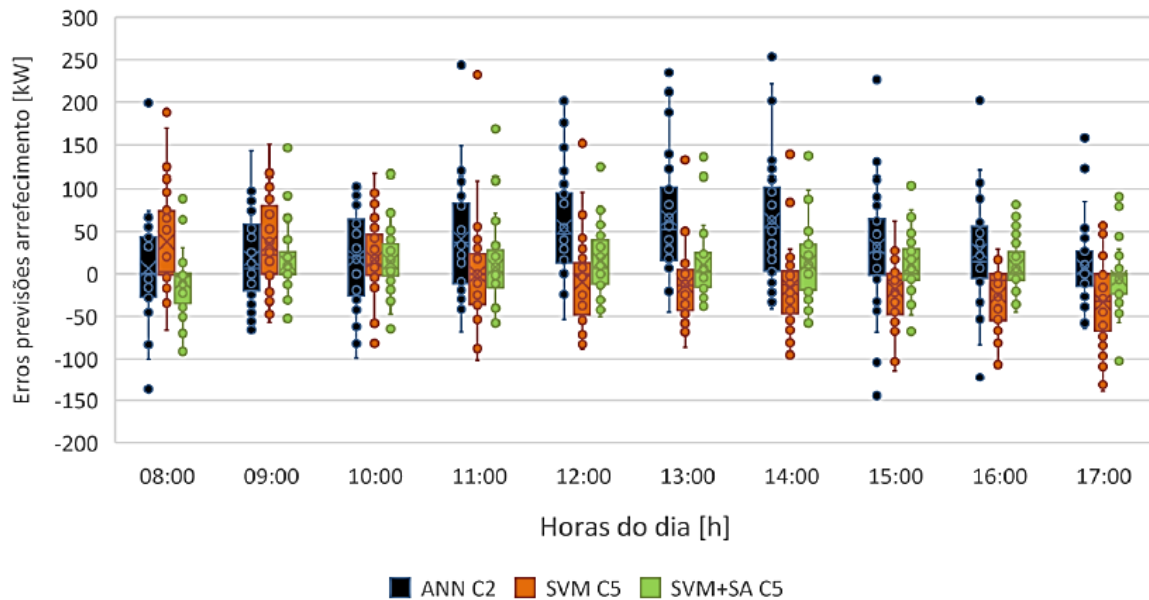


Figura 4.24. Distribuição dos erros dos melhores modelos de previsão para o subconjunto de teste do período de arrefecimento

Na Figura 4.24, principalmente a partir das 12:00h, a tendência do modelo ANN [C2](#) em prever os valores para as necessidades de arrefecimento abaixo dos valores de referência mostra-se bastante evidente, quando comparada com os outros dois modelos. Além disso, a amplitude de distribuição dos erros deste modelo é também maior. Tal como evidenciado no período de arrefecimento (Figura 4.12), também aqui é notório o efeito do *simulated annealing* na redução da amplitude dos erros associados ao modelo SVM+SA [C5](#), comparando com o modelo SVM [C5](#), sem essa otimização.

5. Considerações finais

Com este estudo, pretendeu-se desenvolver e testar modelos baseados em estratégias de *machine learning* de forma a prever as necessidades de aquecimento e arrefecimento de um edifício de serviços e estudar a influência de algumas variáveis exógenas nessas previsões.

Como os dados utilizados neste estudo se baseavam num modelo de simulação dinâmica dimensionado e calibrado para o edifício do caso de estudo (principalmente os valores relativos às necessidades de aquecimento e arrefecimento, que foram utilizados como valores de referência), existe sempre alguma margem de erro associada aos resultados deste estudo, uma vez que não se está a trabalhar com os valores realmente registados naquele edifício.

Ainda assim, com os modelos construídos neste estudo, foi possível obter uma previsão bastante precisa não só das necessidades de aquecimento, mas principalmente das necessidades de arrefecimento do edifício do caso de estudo.

No que diz respeito às variáveis que mais influenciaram as previsões de aquecimento, o modelo que registou as métricas de erro mais baixas foi o modelo ANN [H4](#), que tinha como variáveis de entrada os históricos de consumo para aquecimento 7 e 14 dias antes do dia a ser previsto, a temperatura exterior e ainda a radiação solar direta. Quanto ao período de arrefecimento, o modelo que melhor previu as necessidades de arrefecimento do edifício do caso de estudo foi o modelo SVM+SA [C5](#), que considerava como variáveis de entrada os históricos de consumo para arrefecimento 7 e 14 dias antes do dia a ser previsto, a temperatura exterior, o perfil de ocupação estimado e a humidade relativa.

A meta-heurística selecionada para o processo de otimização dos parâmetros SVM (*simulated annealing*) mostrou ser um complemento muito valioso de modo a melhorar as previsões pelos modelos SVM e tornando estes modelos bastante mais fiáveis na previsão das necessidades de aquecimento e arrefecimento do edifício.

Uma das formas de aumentar a fiabilidade deste estudo seria confrontar os modelos desenvolvidos com os valores reais registados no edifício e aferir quais seriam os erros associados às previsões dos modelos.

Outra forma de melhorar os resultados obtidos neste estudo, desta vez mais focada nas redes neuronais artificiais, seria procurar o número ideal de neurónios a definir na camada oculta consoante o número de registos a ser introduzidos nestes modelos. Esta otimização iria permitir às redes neuronais artificiais uma melhor adaptação aos dados que lhes são introduzidos, melhorando as suas previsões.

Na fase de pré-processamento dos dados, poderiam ser ainda aplicadas algumas técnicas, como a transformada *wavelet*, que permitiria a deteção de padrões de consumo ou de algum ruído existente nos dados, contribuindo assim para uma melhoria nos resultados obtidos.

Referências bibliográficas

- [1] Direção Geral da Energia e Geologia, “Eficiência Energética - Edifícios.” [Online]. Available: <https://www.dgeg.gov.pt/pt/areas-setoriais/energia/eficiencia-energetica/edificios/>. [Accessed: 21-Oct-2020].
- [2] A. Vieira, “Os sistemas de Gestão Técnica Centralizada de Edifícios,” *Revista O Instalador*, 2018. [Online]. Available: <https://oinstalador.com/Artigos/249966-Os-sistemas-de-Gestao-Tecnica-Centralizada-de-Edificios.html>. [Accessed: 21-Oct-2020].
- [3] D. Bonilla, M. G. Samaniego, R. Ramos, and H. Campbell, “Practical and low-cost monitoring tool for building energy management systems using virtual instrumentation,” *Sustain. Cities Soc.*, vol. 39, pp. 155–162, 2018.
- [4] A. Smola and S. V. N. Vishwanathan, *Introduction to Machine Learning*. Cambridge: Cambridge University Press, 2008.
- [5] M. Bourdeau, X. qiang Zhai, E. Nefzaoui, X. Guo, and P. Chatellier, “Modeling and forecasting building energy consumption: A review of data-driven techniques,” *Sustain. Cities Soc.*, vol. 48, p. 101533, Jul. 2019.
- [6] A. H. Neto and F. A. S. Fiorelli, “Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption,” *Energy Build.*, vol. 40, no. 12, pp. 2169–2176, 2008.
- [7] A. L. Katsatos and K. P. Moustris, “Application of artificial neuron networks as energy consumption forecasting tool in the building of regulatory authority of energy, Athens, Greece,” in *Energy Procedia*, 2019, pp. 851–861.
- [8] C. Deb, L. S. Eang, J. Yang, and M. Santamouris, “Forecasting Energy Consumption of Institutional Buildings in Singapore,” in *Procedia Engineering*, 2015, pp. 1734–1740.
- [9] J. Zhao and X. Liu, “A hybrid method of dynamic cooling and heating load forecasting for office buildings based on artificial intelligence and regression analysis,” *Energy Build.*, vol. 174, pp. 293–308, 2018.

- [10] E. Saloux and J. A. Candanedo, “Forecasting District Heating Demand using Machine Learning Algorithms,” in *Energy Procedia*, 2018, pp. 59–68.
- [11] X. Godinho, H. Bernardo, F. T. Oliveira, and J. C. Sousa, “Forecasting heating and cooling energy demand in an office building using machine learning methods,” in *Proceedings - 2020 International Young Engineers Forum, YEF-ECE 2020*, 2020, pp. 1–6.
- [12] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, “Forecasting with artificial neural networks: The state of the art,” *Int. J. Forecast.*, vol. 14, no. 1, pp. 35–62, 1998.
- [13] T. B. Lopez-Garcia, A. Coronado-Mendoza, and J. A. Domínguez-Navarro, “Artificial neural networks in microgrids: A review,” *Eng. Appl. Artif. Intell.*, vol. 95, p. 103894, 2020.
- [14] D. T. Gomes, “Redes Neurais Recorrentes Para Previsão de Séries Temporais de Memórias Curta e Longa,” Universidade Estadual de Campinas, 2005.
- [15] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, pp. 273–397, 1995.
- [16] S. Seyedzadeh, F. P. Rahimian, I. Glesk, and M. Roper, “Machine learning for estimation of building energy consumption and performance: a review,” *Vis. Eng.*, vol. 6, no. 1, 2018.
- [17] J. C. Sousa, H. M. Jorge, and L. P. Neves, “Short-term load forecasting based on support vector regression and load profiling,” *Int. J. Energy Res.*, vol. 38, no. 3, pp. 350–362, 2014.
- [18] L. Xuemei, D. Lixing, and L. Lanlan, “A novel building cooling load prediction based on SVR and SAPSO,” in *2010 International Symposium on Computer, Communication, Control and Automation (3CA)*, 2010, pp. 528–532.
- [19] F. Zhang, C. Deb, S. E. Lee, J. Yang, and K. W. Shah, “Time series forecasting for building energy consumption using weighted Support Vector Regression with differential evolution optimization technique,” *Energy Build.*, vol. 126, pp. 94–103, 2016.
- [20] T. B. Trafalis and H. Ince, “Support vector machine for regression and applications

- to financial forecasting,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 6, pp. 348–353, 2000.
- [21] “RBF SVM parameters.” [Online]. Available: https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html. [Accessed: 20-Feb-2020].
- [22] J. J. G. Tiago, “Utilização de simulação dinâmica para estudar o efeito de ações de Demand Response num edifício de escritórios,” Polytechnic Institute of Leiria, 2019.
- [23] P. F. Pai and W. C. Hong, “Support vector machines with simulated annealing algorithms in electricity load forecasting,” *Energy Convers. Manag.*, vol. 46, no. 17, pp. 2669–2688, 2005.
- [24] S. R. Abbas and M. Arif, “Electric load forecasting using support vector machines optimized by genetic algorithm,” *10th IEEE Int. Multitopic Conf. 2006, INMIC*, pp. 395–399, 2006.
- [25] S. W. Lin, Z. J. Lee, S. C. Chen, and T. Y. Tseng, “Parameter determination of support vector machine and feature selection using simulated annealing approach,” *Appl. Soft Comput. J.*, vol. 8, no. 4, pp. 1505–1512, 2008.
- [26] A. J. Smola and B. Scholkopf, “A tutorial on support vector regression,” *Stat. Comput.*, vol. 14, pp. 199–222, 2004.

Anexos

Anexo A

Ficheiro Python para as redes neuronais artificiais no período de arrefecimento

Separação dos dados em *inputs* e *outputs* para os subconjuntos de treino e teste

```
import numpy as np
from numpy import array
import pandas as pd

#Importação do Ficheiro Excel (dados normalizados)
dataset=pd.read_excel(r'C:\Users\Xavi2\Desktop\Escola\Dados de entrada\Dados de entrada Python.xlsx',sheet_name='24 Horas Arrefecimento Norm')
dataset=array(dataset)

#Definição de Inputs e Outputs para os subconjuntos de treino e teste

Outputs_Train=(dataset[1:136,12:22]).astype(float)
Outputs_Test=(dataset[136:170,12:22]).astype(float)

Input1_Train=(dataset[1:136,39:49]).astype(float)
Input1_Test=(dataset[136:170,39:49]).astype(float)

Input2_Train=(dataset[1:136,66:76]).astype(float)
Input2_Test=(dataset[136:170,66:76]).astype(float)

Temperatura_Train=(dataset[1:136,93:103]).astype(float)
Temperatura_Test=(dataset[136:170,93:103]).astype(float)

Ganhos_Train=(dataset[1:136,119:129]).astype(float)
Ganhos_Test=(dataset[136:170,119:129]).astype(float)

Ocupacao_Train=(dataset[1:136,145:155]).astype(float)
Ocupacao_Test=(dataset[136:170,145:155]).astype(float)

Vento_Train=(dataset[1:136,172:182]).astype(float)
Vento_Test=(dataset[136:170,172:182]).astype(float)

Direct_Rad_Train=(dataset[1:136,199:209]).astype(float)
Direct_Rad_Test=(dataset[136:170,199:209]).astype(float)

Diffuse_Rad_Train=(dataset[1:136,226:236]).astype(float)
Diffuse_Rad_Test=(dataset[136:170,226:236]).astype(float)

HumidadeRelativa_Train=(dataset[1:136,253:263]).astype(float)
HumidadeRelativa_Test=(dataset[136:170,253:263]).astype(float)
```

Implementação dos modelos ANN, cálculo das métricas de erro associadas e gravação dos modelos gerados em cada ciclo

```
'Artificial Neural Network'  
  
from sklearn.neural_network import MLPRegressor  
from sklearn.metrics import mean_absolute_error  
from sklearn.metrics import mean_squared_error  
import pickle  
  
#ciclo "for" para correr os modelos ANN por 10 vezes  
  
for simulacao in range (0,10):  
    print ('Artificial Neural Network')  
  
    ANN_md1=MLPRegressor (hidden_layer_sizes = (40), activation = 'tanh', max_iter=50000, tol=1e-10, early_stopping=True, verbose='True', validation_fraction=0.2)  
    ANN_md1.fit(Inputs_Train,Outputs_Train) # para calibrar o modelo de acordo com as pares de entradas/saída  
    Y_pred_ANN=ANN_md1.predict(Inputs_Train) # para sujeitar o modelo criado a estimar saídas baseadas nas entradas do subconjunto de dados de treino  
    Y_pred_Test_ANN=ANN_md1.predict(Inputs_Test) # para sujeitar o modelo criado a estimar saídas baseadas nas entradas do subconjunto de dados de teste  
  
    #converter valores negativos gerados pelo modelo ANN  
  
    for i in range (0,len(Y_pred_ANN)):  
        for j in range (0,len(Y_pred_ANN[0])):  
            if Y_pred_ANN[i,j]<0:  
                Y_pred_ANN[i,j]=0.0001  
  
    for i in range (0,len(Y_pred_Test_ANN)):  
        for j in range (0,len(Y_pred_Test_ANN[0])):  
            if Y_pred_Test_ANN[i,j]<0:  
                Y_pred_Test_ANN[i,j]=0.0001  
  
    #MAE ANN  
  
    MAE_ANN_Train=mean_absolute_error(Outputs_Train_8a24,Previsoes_Train)  
    MAE_ANN_Test=mean_absolute_error(Outputs_Test_8a24,Previsoes_Test)  
  
    #MSE Artificial Neural Network  
  
    MSE_ANN_Train=mean_squared_error(Outputs_Train_8a24,Previsoes_Train)  
    MSE_ANN_Test=mean_squared_error(Outputs_Test_8a24,Previsoes_Test)  
  
    #RMSE Artificial Neural Network  
  
    RMSE_ANN_Train=np.sqrt(MSE_ANN_Train)  
    RMSE_ANN_Test=np.sqrt(MSE_ANN_Test)  
  
    #guardar as pesos atribuidas a cada modelo para replicar no futuro  
  
    Modelo_ANN = 'Modelo ANN Arref Temp+Hrel+Ocup.sav'  
    pickle.dump(ANN_md1, open(Modelo_ANN + str(simulacao), 'wb'))
```

Replicação do melhor dos 10 modelos gerados para as redes neuronais

```
'Best-of'  
del ANN_md1  
loaded_model = pickle.load(open('Modelo ANN Arref Temp+HRel+Ocup.sav1', 'rb'))  
Y_pred_ANN_Best_Of=loaded_model.predict(Inputs_Train) # para sujeitar o modelo criado a estimar saídas baseadas nas entradas do subconjunto de dados de treino  
Y_pred_Test_ANN_Best_Of=loaded_model.predict(Inputs_Test) # para sujeitar o modelo criado a estimar saídas baseadas nas entradas do subconjunto de dados de teste  
  
for i in range (0,len(Y_pred_ANN_Best_Of)):  
    for j in range (0,len(Y_pred_ANN_Best_Of[0])):  
        if Y_pred_ANN_Best_Of[i,j]<0:  
            Y_pred_ANN_Best_Of[i,j]=0.0001  
  
for i in range (0,len(Y_pred_Test_ANN_Best_Of)):  
    for j in range (0,len(Y_pred_ANN_Best_Of[0])):  
        if Y_pred_Test_ANN_Best_Of[i,j]<0:  
            Y_pred_Test_ANN_Best_Of[i,j]=0.0001
```

Anexo B

Ficheiro Python para as máquinas de vetores de suporte no período de arrefecimento

Separação dos dados em *inputs* e *outputs* para os subconjuntos de treino e teste

```
import numpy as np
from numpy import array
import pandas as pd

#Importação do Ficheiro Excel

dataset=pd.read_excel(r'C:\Users\Xavi2\Desktop\Escola\Dados de entrada\Dados de entrada Python.xlsx')
dataset=array(dataset)

#Definição de Inputs e Outputs para os subconjuntos de treino e teste

Input1_Train=(dataset[2160:5736-336,16]).astype(float)
Input1_Test=(dataset[5736-336:6552-336,16]).astype(float)

Input2_Train=(dataset[2328:5736-168,16]).astype(float)
Input2_Test=(dataset[5736-168:6552-168,16]).astype(float)

Ocupacao_Train=(dataset[2496:5736,1]).astype(float)
Ocupacao_Test=(dataset[5736:6552,1]).astype(float)

Ganhos_Train=(dataset[2496:5736,2]).astype(float)
Ganhos_Test=(dataset[5736:6552,2]).astype(float)

Temperatura_Train=(dataset[2496:5736,6]).astype(float)
Temperatura_Test=(dataset[5736:6552,6]).astype(float)

Vento_Train=(dataset[2496:5736,7]).astype(float)
Vento_Test=(dataset[5736:6552,7]).astype(float)

Direct_Rad_Train=(dataset[2496:5736,18]).astype(float)
Direct_Rad_Test=(dataset[5736:6552,18]).astype(float)

Diffuse_Rad_Train=(dataset[2496:5736,19]).astype(float)
Diffuse_Rad_Test=(dataset[5736:6552,19]).astype(float)

HumidadeRelativa_Train=(dataset[2496:5736,20]).astype(float)
HumidadeRelativa_Test=(dataset[5736:6552,20]).astype(float)

Outputs_Train=(dataset[2496:5736,16]).astype(float)
Outputs_Test=(dataset[5736:6552,16]).astype(float)
```

Escolha de variáveis de entrada e respetiva normalização

```
#Escolha de variáveis de entrada

Inputs_Train1=np.vstack((Input1_Train,Input2_Train, Temperatura_Train,Ocupacao_Train,HumidadeRelativa_Train))
Inputs_Train=np.transpose(Inputs_Train1)
Inputs_Test1=np.vstack((Input1_Test,Input2_Test, Temperatura_Test,Ocupacao_Test,HumidadeRelativa_Test))
Inputs_Test=np.transpose(Inputs_Test1)

#Normalização dos dados

from sklearn.preprocessing import MinMaxScaler

scaler=MinMaxScaler()
scaler.fit(Inputs_Train)
Inputs_Train_Norm=scaler.transform(Inputs_Train)
Inputs_Test_Norm=scaler.transform(Inputs_Test)
```

Implementação dos modelos SVM e cálculo das métricas de erro associadas

```
'Support Vector Regression'  
  
from sklearn.svm import SVR  
  
SVR_md1= SVR (C=500, kernel='rbf', epsilon=0.5,gamma=0.5)  
SVR_md1.fit(Inputs_Train_Norm,Outputs_Train) # para calibrar o modelo de acordo com os pares de entradas/saída  
Y_pred_SVR=SVR_md1.predict(Inputs_Train_Norm) # para sujeitar o modelo criado a estimar saídas baseadas nas entradas do subconjunto de dados de treino  
Y_pred_Test_SVR=SVR_md1.predict(Inputs_Test_Norm) # para sujeitar o modelo criado a estimar saídas baseadas nas entradas do subconjunto de dados de teste  
  
for i in range (0,len(Y_pred_SVR)):  
    if Y_pred_SVR[i]<0:  
        Y_pred_SVR[i]=0.0001  
  
for j in range (0,len(Y_pred_Test_SVR)):  
    if Y_pred_Test_SVR[j]<0:  
        Y_pred_Test_SVR[j]=0.0001  
  
from sklearn.metrics import mean_absolute_error  
from sklearn.metrics import mean_squared_error  
  
#MAE Support Vector Regression  
  
MAE_SVR_Train=mean_absolute_error(Outputs_Train,Y_pred_SVR)  
MAE_SVR_Test=mean_absolute_error(Outputs_Test,Y_pred_Test_SVR)  
  
#MSE Support Vector Regression  
  
MSE_SVR_Train=mean_squared_error(Outputs_Train,Y_pred_SVR)  
MSE_SVR_Test=mean_squared_error(Outputs_Test,Y_pred_Test_SVR)  
  
#RMSE Support Vector Regression  
  
RMSE_SVR_Train=np.sqrt(MSE_SVR_Train)  
RMSE_SVR_Test=np.sqrt(MSE_SVR_Test)
```

Anexo C

Ficheiro Python para a otimização dos parâmetros SVM através do *simulated annealing* no período de arrefecimento

Separação dos dados em *inputs* e *outputs* para os subconjuntos de treino e teste

```
import pandas as pd
import numpy as np
from numpy import array

#Importação do Ficheiro Excel

dataset=pd.read_excel(r'C:\Users\Xavi2\Desktop\Escola\Dados de entrada\Dados de entrada Python.xlsx',sheet_name='24 Horas Arrefecimento')
dataset=array(dataset)

#Definição de Inputs e Outputs para os subconjuntos de treino e teste

Outputs_Train=(dataset[1:136,12:22]).astype(float)
Outputs_Test=(dataset[136:170,12:22]).astype(float)

Input1_Train=(dataset[1:136,38:48]).astype(float)
Input1_Test=(dataset[136:170,38:48]).astype(float)

Input2_Train=(dataset[1:136,66:76]).astype(float)
Input2_Test=(dataset[136:170,66:76]).astype(float)

Temperatura_Train=(dataset[1:136,94:104]).astype(float)
Temperatura_Test=(dataset[136:170,94:104]).astype(float)

Ganhos_Train=(dataset[1:136,121:131]).astype(float)
Ganhos_Test=(dataset[136:170,121:131]).astype(float)

Ocupacao_Train=(dataset[1:136,147:157]).astype(float)
Ocupacao_Test=(dataset[136:170,147:157]).astype(float)

Vento_Train=(dataset[1:136,174:184]).astype(float)
Vento_Test=(dataset[136:170,174:184]).astype(float)

Direct_Rad_Train=(dataset[1:136,201:211]).astype(float)
Direct_Rad_Test=(dataset[136:170,201:211]).astype(float)

Diffuse_Rad_Train=(dataset[1:136,228:238]).astype(float)
Diffuse_Rad_Test=(dataset[136:170,228:238]).astype(float)

HumidadeRelativa_Train=(dataset[1:136,255:265]).astype(float)
HumidadeRelativa_Test=(dataset[136:170,255:265]).astype(float)
```

Abertura do ciclo “for” com seleção de *inputs* e *outputs*, normalização dos dados e separação entre dados de treino e validação para cada hora

```
#ciclo "for" para cada hora de funcionamento dos sistemas de arrefecimento
for h in range (0,10):
    #Definição de Inputs e Outputs para cada hora "h"
    Inputs_Train=np.transpose(np.vstack((Input1_Train[:,h],Input2_Train[:,h],Temperatura_Train[:,h],Ocupacao_Train[:,h],HumidadeRelativa_Train[:,h])))
    Inputs_Test=np.transpose(np.vstack((Input1_Test[:,h],Input2_Test[:,h],Temperatura_Test[:,h],Ocupacao_Test[:,h],HumidadeRelativa_Test[:,h])))

    Hour_Outputs_Train=Outputs_Train[:,h]
    Hour_Outputs_Test=Outputs_Test[:,h]

    #Normalização de dados
    from sklearn.preprocessing import MinMaxScaler

    scaler=MinMaxScaler()
    scaler.fit(Inputs_Train)
    Inputs_Train_Norm=scaler.transform(Inputs_Train)
    Inputs_Test_Norm=scaler.transform(Inputs_Test)

    #Separação de 20% dos dados de treino para validação
    from sklearn.model_selection import train_test_split

    Inputs1=np.transpose(Inputs_Train_Norm)
    Train=np.transpose(np.vstack((Inputs1,Hour_Outputs_Train)))
    Train_Valid=train_test_split(Train,test_size=0.2)

    Dataset_Train_Valid=Train_Valid[0]
    Inputs_Train_Valid=Dataset_Train_Valid[:,0:len(Dataset_Train_Valid[0])-1]
    Outputs_Train_Valid=Dataset_Train_Valid[:,len(Dataset_Train_Valid[0])-1]

    Dataset_Valid=Train_Valid[1]
    Inputs_Valid=Dataset_Valid[:,0:len(Dataset_Valid[0])-1]
    Outputs_Valid=Dataset_Valid[:,len(Dataset_Valid[0])-1]
```

Definição da função objetivo

```
#Função objetivo para aplicar no Simulated Annealing

from sklearn.svm import SVR
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error

def f(x):
    SVR_md1= SVR (C = x[0], epsilon = x[1], gamma = x[2], kernel='rbf')
    SVR_md1.fit(Inputs_Train_Valid,Outputs_Train_Valid)
    Y_pred_SVR=SVR_md1.predict(Inputs_Train_Valid)
    Y_pred_Valid_SVR=SVR_md1.predict(Inputs_Valid)
    MSE_SVR_Valid=mean_squared_error(Outputs_Valid,Y_pred_Valid_SVR)
    RMSE_SVR_Valid=np.sqrt(MSE_SVR_Valid)
    obj=RMSE_SVR_Valid
    return obj
```

Definição de parâmetros iniciais para o *simulated annealing*

```
import math

# Combinação inicial
x_start = [500, 0.5, 0.5]

# Simulated Annealing
#####
# Número de ciclos
n = 30
# Número de combinações a testar em cada ciclo
m = 15
# Número de soluções aceites
na = 0.0
# Probabilidade de aceitar a pior solução no início
p1 = 0.5
# Probabilidade de aceitar a pior solução no fim
p50 = 0.001
# Temperatura inicial
t1 = -1.0/math.log(p1)
# Temperatura final
t50 = -1.0/math.log(p50)
# Fração de redução em cada ciclo
frac = (t50/t1)**(1.0/(n-1.0))
# Inicializar x
x = np.zeros((n+1,3))
x[0] = x_start
xi = np.zeros(3)
xi = x_start
na = na + 1.0
# Melhor combinação até agora
xc = np.zeros(3)
xc = x[0]
fc = f(xi)
fs = np.zeros(n+1)
fs[0] = fc
# Temperatura atual
t = t1
# Média DeltaE
DeltaE_avg = 0.0
```

Processo de *simulated annealing*

```
for i in range(n):
    print('Cycle: ' + str(i) + ' with Temperature: ' + str(t) + ' Objective: ' + str(fc))
    for j in range(m):
        import random
        # Gerar novos pontos
        xi[0] = xc[0] + random.random()*100 - 50
        xi[1] = xc[1] + random.random() - 0.5
        xi[2] = xc[2] + random.random() - 0.5
        # Definir limites máximo e mínimo para cada parâmetro SVM
        xi[0] = max(min(xi[0],700.0),1.0)
        xi[1] = max(min(xi[1],3.0),0.00001)
        xi[2] = max(min(xi[2],10.0),0.0001)
        DeltaE = abs(f(xi)-fc)
        if (f(xi)>fc):
            # Inicializar DeltaE se uma pior solução for encontrada
            # na primeira iteração
            if (i==0 and j==0): DeltaE_avg = DeltaE
            # função objetivo é pior
            # gerar probabilidade de aceitação
            p = math.exp(-DeltaE/(DeltaE_avg * t))
            # determinar se aceita ou não a pior solução
            if (random.random()<p):
                # aceitar a pior solução
                accept = True
            else:
                # não aceitar a pior solução
                accept = False
        else:
            # a função objetivo é menor, aceita a solução automaticamente
            accept = True
        if (accept==True):
            # atualizar a combinação atualmente aceite
            xc[0] = xi[0]
            xc[1] = xi[1]
            xc[2] = xi[2]
            fc = f(xc)
            # incrementar o número de soluções aceites
            na = na + 1.0
            # atualizar DeltaE_avg
            DeltaE_avg = (DeltaE_avg * (na-1.0) + DeltaE) / na
            # Guardar os melhores valores de cada parâmetro no fim de cada ciclo
            x[i+1][0] = xc[0]
            x[i+1][1] = xc[1]
            x[i+1][2] = xc[2]
            fs[i+1] = fc
            # Baixar a temperatura para o próximo ciclo
            t = frac * t

        # Mostrar solução
    print('Best solution: ' + str(xc))
    print('Best objective: ' + str(fc))
```

Criação de gráficos para a evolução da otimização dos parâmetros SVM, obtenção de valores previstos para cada hora, gravação dos respectivos parâmetros obtidos e fecho do ciclo “for”

```
#Gerar plots para a evolução da função objetivo e parâmetros SVM testados

import matplotlib.pyplot as plt

fig = plt.figure(figsize=(8.0,5.0))
ax1 = fig.add_subplot(311)
ax1.plot(fs, 'r.-')
ax1.legend(['Objective'])
ax2 = fig.add_subplot(312)
ax2.plot(x[:,1], 'b.-')
ax2.plot(x[:,2], 'g.-')
ax2.legend(['epsilon', 'gamma'])
ax3 = fig.add_subplot(313)
ax3.plot(x[:,0], 'b.-')
ax3.legend(['C'])
plt.xlabel('Iterations')

#gerar valores previstos para os dados de treino, validação e teste com base na
#melhor solução retirada do simulated annealing em cada hora

SVR_md1= SVR (C = xc[0], epsilon = xc[1], gamma = xc[2], kernel='rbf')
SVR_md1.fit(Inputs_Train_Valid,Outputs_Train_Valid)
Y_pred_Train_SVR=SVR_md1.predict(Inputs_Train_Valid)
Y_pred_Valid_SVR=SVR_md1.predict(Inputs_Valid)
Y_pred_Test_SVR=SVR_md1.predict(Inputs_Test_Norm)

#salvar valores previstos em cada subconjunto e parâmetros para cada hora

Predicted_Train_Random=np.hstack((Y_pred_Train_SVR,Y_pred_Valid_SVR))
Predicted_Train[:,h]=np.hstack((Y_pred_Train_SVR,Y_pred_Valid_SVR))
Predicted_Test[:,h]=Y_pred_Test_SVR
Parameters=np.hstack((xc[0],xc[1],xc[2]))
Hour_Parameters[:,h]=Parameters
Outputs_Train_Random[:,h]=np.hstack((Outputs_Train_Valid,Outputs_Valid))
```

Conversão de valores negativos gerados pelo modelo SVM+SA e cálculo das métricas de erro associadas

```
#converter valores negativos gerados pelo modelo SVM

for i in range (0,len(Previsoes_Train)):
    for j in range (0,len(Previsoes_Train[0])):
        if Previsoes_Train[i,j]<0:
            Previsoes_Train[i,j]=0.0001

for i in range (0,len(Previsoes_Test)):
    for j in range (0,len(Previsoes_Test[0])):
        if Previsoes_Test[i,j]<0:
            Previsoes_Test[i,j]=0.0001

Outputs_Train=np.hstack((Horas0_8_Train,Outputs_Train_Random,Horas19_24_Train))
Outputs_Test=(dataset[136:170,4:28]).astype(float)

#MAE Support Vector Machine

MAE_SVR_Train=mean_absolute_error(Outputs_Train,Previsoes_Train)
MAE_SVR_Test=mean_absolute_error(Outputs_Test,Previsoes_Test)

#MSE Artificial Neural Network

MSE_SVR_Train=mean_squared_error(Outputs_Train,Previsoes_Train)
MSE_SVR_Test=mean_squared_error(Outputs_Test,Previsoes_Test)

#RMSE Artificial Neural Network

RMSE_SVR_Train=np.sqrt(MSE_SVR_Train)
RMSE_SVR_Test=np.sqrt(MSE_SVR_Test)
```

Anexo D

Gráficos de otimização dos parâmetros SVM no período de arrefecimento

