

Universidade Federal de Ouro Preto
ICEA - Instituto de Ciências Exatas e Aplicadas
Graduação em Engenharia Elétrica

Aline do Espirito Santo Romão

**PREDIÇÃO DA EFICIÊNCIA ENERGÉTICA DE EDIFÍCIOS
RESIDENCIAIS UTILIZANDO REDES NEURAS ARTIFICIAIS**

João Monlevade

2022

Aline do Espirito Santo Romão

**PREDIÇÃO DA EFICIÊNCIA ENERGÉTICA DE EDIFÍCIOS
RESIDENCIAIS UTILIZANDO REDES NEURAIS ARTIFICIAIS**

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Título de Bacharela em Engenharia Elétrica pelo Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto.

Orientadora: Profa. Dra. Sarah Negreiros de Carvalho Leite

João Monlevade

2022

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

R766p Romão, Aline do Espírito Santo.
Predição da eficiência energética de edifícios residenciais utilizando
redes neurais artificiais. [manuscrito] / Aline do Espírito Santo Romão. -
2022.
44 f.: il.: color., gráf., tab..

Orientadora: Profa. Dra. Sarah Negreiros de Carvalho Leite.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia
Elétrica .

1. Aprendizado do computador. 2. Carga e distribuição elétrica -
Previsão. 3. Edifícios - Energia elétrica - Distribuição. 4. Energia -
Conservação. 5. Redes neurais (Computação). 6. Telecomunicações. I.
Leite, Sarah Negreiros de Carvalho. II. Universidade Federal de Ouro
Preto. III. Título.

CDU 621.31

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



FOLHA DE APROVAÇÃO

Aline do Espírito Santo Romão

Predição da eficiência energética de edifícios residenciais utilizando redes neurais artificiais

Monografia apresentada ao Curso de Engenharia Elétrica da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharela em Engenharia Elétrica.

Aprovada em 01 de novembro de 2022.

Membros da banca

Profa. Dra. Sarah Negreiros de Carvalho Leite - Orientadora - Universidade Federal de Ouro Preto
Prof. Dr. Wilingthon Guerra Zvietcovich - Universidade Federal de Ouro Preto
Me. Guilherme Vettorazzi Vargas – Universidade Estadual de Campinas

Profa. Dra. Sarah Negreiros de Carvalho Leite, orientadora do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 03/02/2023.



Documento assinado eletronicamente por **Sarah Negreiros de Carvalho Leite, PROFESSOR DE MAGISTERIO SUPERIOR**, em 03/02/2023, às 16:13, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0469082** e o código CRC **4FC105CC**.

RESUMO

Métodos de aprendizado de máquina podem ser usados para ajudar a projetar edifícios, reduzindo as cargas de energia enquanto se mantem a temperatura interna em níveis desejados. O presente trabalho visa estudar o efeito de oito variáveis referentes a construção de edifícios, a saber: compacidade relativa, área de superfície, área da parede, área do telhado, altura total, orientação, área envidraçada e distribuição da área envidraçada, para prever duas variáveis de saída (carga de aquecimento e carga de refrigeração) utilizando redes neurais artificiais. A base de dados empregada considera dados de 768 edifícios. Foi realizada um estudo estatístico entre as variáveis e foram testados os parâmetros de modelos preditivos do tipo *MultiLayer Perceptron* (MLP) e Redes Convolucionais (CNN). Dentre os cenários testados, o método MLP forneceu um erro absoluto médio de 0.28 na predição da carga de aquecimento e de 0.36 para a predição da carga de aquecimento. O erro cometido pelo modelo proposto é menor que os relatados na literatura que também empregam a mesma base de dados.

Palavras-chave:

CNN, MLP, Redes Neurais, Aprendizado de Máquina, Eficiência Energética.

ABSTRACT

Machine learning can help design buildings, reducing energy loads while maintaining internal temperature at desired levels. This study aims to evaluate the effect of eight variables referring to the construction of buildings, namely: relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area and glazing area distribution, to predict two output variables (heating load and cooling load) using artificial neural networks. The database has data from 768 buildings. A statistical study was carried out considering different parameters of predictive models of the Multilayer Perceptron (MLP) and Convolutional Networks (CNN). Among the tested scenarios, the MLP provided a mean absolute error of 0.28 to predict the heating load and 0.36 for heating load prediction. In the best configuration, the error of our proposed model is smaller than those reported in the literature, considering the same database.

Keywords: CNN, MLP, Neural Networks, Machine Learning, Energy Efficiency.

LISTA DE ILUSTRAÇÕES

Figura 1 – Geração de formas com base em dezoito elementos cúbicos.	13
Figura 2 – Variação do coeficiente de compacidade relativa.	14
Figura 3 – Ilustração das cinco primeiras e cinco últimas amostras das variáveis de entrada e saída da base de dados.	14
Figura 4 – Compacidade relativa.	16
Figura 5 – Área de superfície.	16
Figura 6 – Área da parede.	16
Figura 7 – Área do telhado.	16
Figura 8 – Altura total.	16
Figura 9 – Orientação.	16
Figura 10 – Área envidraçada.	16
Figura 11 – Distribuição da área envidraçada.	16
Figura 12 – Boxplot das variáveis de entrada.	16
Figura 13 – Matrix de correlação dos dados.	17
Figura 14 – Rede Neural Rasa.	23
Figura 15 – Exemplo de rede CNN.	25
Figura 16 – RMSE ao longo das épocas para Y_1 no método CNN sem seleção. . . .	28
Figura 17 – RMSE ao longo das épocas para Y_2 no método CNN sem seleção. . . .	28
Figura 18 – Diferença entre valores preditos e reais para Y_1 no método CNN sem seleção.	29
Figura 19 – Diferença entre valores preditos e reais para Y_2 no método CNN sem seleção.	29
Figura 20 – RMSE para Y_1 no método CNN com seleção	31
Figura 21 – RMSE para Y_2 no método CNN com seleção.	31
Figura 22 – Diferença entre valores preditos e reais para Y_1 no método CNN com seleção.	32
Figura 23 – Diferença entre valores preditos e reais para Y_2 no método CNN com seleção.	32
Figura 24 – RMSE para Y_1 no método MLP sem seleção.	34
Figura 25 – RMSE para Y_2 no método MLP sem seleção.	34
Figura 26 – Diferença entre valores preditos e reais para Y_1 no método MLP sem seleção.	35
Figura 27 – Diferença entre valores preditos e reais para Y_2 no método MLP sem seleção.	35
Figura 28 – RMSE para Y_1 no método MLP com seleção.	37
Figura 29 – RMSE para Y_2 no método MLP com seleção.	37
Figura 30 – Diferença entre valores preditos e reais para Y_1 no método MLP com seleção.	38

Figura 31 – Diferença entre valores preditos e reais para Y_2 no método MLP com seleção.	38
--	----

LISTA DE TABELAS

Tabela 1 – Parâmetros e configurações testadas.	26
Tabela 2 – Melhor seleção de parâmetros para rede CNN.	27
Tabela 3 – Resultados obtidos sem seleção de variáveis para rede CNN.	27
Tabela 4 – Resultados obtidos com seleção de variáveis para rede CNN.	30
Tabela 5 – Melhor seleção de parâmetros para a rede MLP.	33
Tabela 6 – Resultados obtidos sem seleção de variáveis para a rede MLP.	33
Tabela 7 – Resultados obtidos com seleção de variáveis para a rede MLP.	36
Tabela 8 – Comparação de resultados obtidos.	40

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Objetivos	9
2	REVISÃO BIBLIOGRÁFICA	10
3	METODOLOGIA	12
3.1	Base de dados	12
3.1.1	Padronização da base de dados	18
3.2	Redes neurais e aprendizado supervisionado	19
3.2.1	Redes neurais	19
3.2.2	Aprendizado supervisionado	19
3.3	Seleção de características	20
3.4	Modelos preditivos	21
3.4.1	Rede MLP	22
3.4.2	Rede CNN	23
4	RESULTADOS	26
4.1	Cenários testados	26
4.2	Implementação da rede CNN	26
4.2.1	Modelo sem seleção de variáveis	26
4.2.2	Modelo sem as variáveis X_6 e X_8	30
4.3	Implementação da rede MLP	33
4.3.1	Modelo sem seleção de variáveis	33
4.3.2	Modelo sem a variável X_6	36
5	CONCLUSÃO	40
	REFERÊNCIAS	42

1 INTRODUÇÃO

O consumo energético tem aumentado constantemente nas últimas décadas em todo o mundo [1]. As condições climáticas de edifícios residenciais podem ser controladas por meio de equipamentos de ar condicionado e aquecedores. Porém, o uso constante destes dispositivos pode gerar elevado consumo de energia, tornando necessário a existência de alternativas para reduzir o uso de equipamentos de refrigeração e aquecimento e manter as condições climáticas internas desejadas.

A regulação do clima interno nos edifícios, por meio do aquecimento, ventilação e ar condicionado, são responsáveis por grande parte do consumo energético [2]. Desta forma, a existência de projetos de construção energeticamente mais eficientes, com base em características construtivas e com propriedades de conservação de energia, são essenciais. Esta abordagem permite uma climatização interna agradável ao mesmo tempo que evita o aumento desnecessário do consumo energético.

Ferramentas de simulação energética de edifícios são amplamente utilizadas a fim de analisar ou prever o consumo energético do edifício, facilitando o projeto e a operação de edifícios energeticamente eficientes. Existem diversos softwares, como o Ecotect e o Autodesk Revit Architecture, que permitem realizar essas simulações. Resultados práticos mostram que as simulações muitas vezes conseguem refletir com precisão as medições reais [3]. As ferramentas de simulação permitem a experimentação com parâmetros que seriam inviáveis, ou de difícil controle na prática da projeção energética dos edifícios, podendo, por exemplo, facilitar a comparação de edifícios idênticos onde apenas um único parâmetro é modificado, a fim de avaliar os efeitos no projeto final.

Desta forma, a utilização de softwares de simulação de energia pode fornecer soluções confiáveis na etapa do projeto de construção dos edifícios. Entretanto, segundo Yezioro , este processo pode ser muito demorado e requer experiência do usuário em um determinado programa, além da precisão dos resultados estimados sofrer variação entre os diferentes pacotes de software de simulação disponíveis.

Uma abordagem que vem sendo cada vez mais conveniente se baseia em aplicar ferramentas de aprendizado de máquina para estudar o efeito de vários parâmetros de construção sobre algumas variáveis de interesse, tendo como base um banco de dados das variáveis do projeto. Uma vez que um modelo foi adequadamente treinado, o uso de conceitos estatísticos e de aprendizado de máquina aceleram o processo de obtenção de respostas e facilitam a variação dos parâmetros. As simulações também têm a função de fornecer bons indicadores da provável mudança percentual e de qualquer tendência dos dados reais, permitindo comparações do consumo energético de edifícios [4].

1.1 Objetivos

Tendo como base o contexto abordado anteriormente, o objetivo geral deste trabalho é desenvolver um modelo preditivo utilizando técnicas de aprendizado de máquinas, mais especificamente, redes neurais artificiais, para estimar, com base em parâmetros de construção dos edifícios, a carga de aquecimento e resfriamento exigida. Serão aplicadas as técnicas CNN e MLP com alteração dos parâmetros, a fim de se obter uma predição mais condizente com os valores reais.

Para atingir tal objetivo geral, foram definidos os seguintes objetivos específicos:

- Compreender a influência das variáveis construtivas dos edifícios sobre a predição dos valores de carga de aquecimento e resfriamento.
- Implementar sistemas preditivos utilizando redes neurais artificiais do tipo *Multilayer Perceptron* e Convolutacional para previsão do desempenho energético de edifícios.
- Investigar o desempenho do modelo com diferentes topologias de redes neurais e realizar o tunelamento dos parâmetros.
- Comparar o desempenho dos modelos implementados entre si e com as análises encontradas na literatura.

Todo o embasamento teórico e desenvolvimento desta monografia está organizada da seguinte maneira: no Capítulo 1, é apresentada a introdução do tema do trabalho, a descrição do problema de pesquisa e os objetivos. No Capítulo 2, é apresentada a revisão de literatura, com comentários sobre trabalhos que utilizam a mesma base de dados e técnicas semelhantes às que foram adotadas no presente trabalho. No Capítulo 3, é realizada a descrição dos modelos e técnicas aplicadas. No Capítulo 4, são apresentados e discutidos os resultados. No Capítulo 5, são feitas as conclusões e sugeridos trabalhos futuros sobre o tema.

2 REVISÃO BIBLIOGRÁFICA

Usando extensas simulações térmicas paramétricas, Pessenlehner e Mahdavi examinaram a influência de parâmetros morfológicos que definem formas de construção residencial para cargas de aquecimento. Usando os mesmos dados coletados em 2003 por Pessenlehner e Mahdavi, Tsanas e Xifara forneceram uma análise estatística meticulosa para obter uma visão importante das propriedades subjacentes de variáveis de entrada e saída, o que levou a estimar a carga de aquecimento com apenas 0,5 pontos de desvio e a de resfriamento com desvio de 1,5 pontos dos resultados simulados.

Em 2014, Chou e Bui implementaram técnicas de inteligência artificial para prever o desempenho energético dos edifícios, utilizando um modelo conjunto que consiste na combinação linear de dois ou mais modelos. Após as análises, os autores sugeriram um aprofundamento nos estudos com foco na otimização dos parâmetros do modelo, a fim de obter melhorias em sua precisão na previsão das variáveis de saída.

Em 2011, Kwok et al. utilizaram um modelo de rede neural MLP para simular a carga total de resfriamento de um prédio de escritórios em Hong Kong, utilizando três simulações e considerando: apenas fatores externos; uma combinação de fatores internos e externos da área de ocupação e no terceiro modelo também foi considerado o consumo energético em PAU (unidades de tratamento de ar primário). Os melhores resultados obtidos concluíram que a utilização de dados de ocupação dos prédios geraram significativo aumento na precisão dos resultados, principalmente da carga de resfriamento.

Navarro-Gonzalez e Villacampa (2019) desenvolveu uma técnica de regressão octaédrica para estimar as necessidades energéticas de diferentes edifícios, de acordo com suas características principais. Segundo os autores, essa metodologia consiste em uma média do kernel que tenta simplificar todo o processo, desde a aquisição de dados até a geração de modelo. A técnica desenvolvida pelo autor apresentou como vantagem a utilização de características geométricas advindas das propriedades do octaedro, permitindo utilizar as distribuições espaciais dos eixos de simetria para analisar o efeito de diferentes orientações sobre o cálculo parcial da estimativa.

Seyedzadeh et al. (2019) utilizou cinco técnicas de aprendizado de máquina para realizar a estimativa energética de edifícios, sendo elas: redes neurais artificiais (RNA), máquinas de vetores de suporte (SVM), processos gaussianos, *Random Forest* e *Gradient Boosted Regression Trees* (GBRT). Após as análises, os autores chegaram a conclusão de que a técnica GBRT foi capaz de fornecer as previsões mais precisas. Porém, quando os dados (em termos de variáveis de entrada e tamanho) eram simples, o SVM provou ser a melhor escolha devido a simplicidade e rapidez dos cálculos, enquanto que, com dados complexos, as redes neurais artificiais multicamadas se mostraram mais apropriadas, principalmente diante de uma demanda massiva, que exigia muitas simulações de energia,

uma vez que essa técnica era mais rápida que as outras analisadas.

Sadeghi et al. (2020) aplicou redes neurais artificiais e *deep learning* na base de dados criada por Tsanas e Xifara, e concluiu que a técnica de *deep learning* alcançou resultados preditivos mais precisos. Além disso, os autores observaram que separar as cargas de aquecimento e resfriamento em modelos individuais melhora a previsão das cargas de resfriamento, mas diminuiu a precisão na previsão das cargas de aquecimento. Os autores indicaram que a randomização e a média móvel são capazes de melhorar a previsibilidade de ambas as variáveis de saída.

Moayedi et al. (2019) utilizaram as redes neurais do tipo *Multilayer Perceptron* (MLP), algoritmos genéticos e algoritmos competitivos imperialistas (ICA) na mesma base de dados utilizada neste presente trabalho. Para isso, os autores selecionaram aleatoriamente 80% (614 registros) dos dados para fase de treinamento, e os 20% restantes (154 registros) foram usados para avaliar a precisão dos modelos gerados, através da análise do coeficiente de determinação (R^2), erro médio absoluto (MAE) e raiz quadrada do erro quadrático médio (RMSE). Eles apontam que a técnica ICA auxilia as redes neurais artificiais a aumentar a correlação dos resultados e também supera os algoritmos genéticos na melhoria dos resultados preditivos. A análise dos coeficientes determinou que a técnica de *random forest* gerou os melhores resultados na base de dados, sendo estes 0.9997, 0.19 e 0.2399 para R^2 , MAE e RMSE, respectivamente, que podem ser considerados resultados satisfatórios.

No trabalho de Duarte et al. (2017), foram utilizadas quatro técnicas para analisar a base de dados: árvore de decisão, máquina de vetores de suporte, *random forest* e MLP. Eles indicam que as cargas de aquecimento podem ser estimadas com mais precisão do que as cargas de resfriamento devido ao fato de que as variáveis interagem de forma mais eficiente para fornecer uma estimativa das cargas de aquecimento. A pesquisa dos autores conclui que, apesar de ambas as técnicas fornecerem resultados similares para as medidas estatísticas, o método de *random forest* apresentou os melhores resultados com coeficiente de determinação (R^2) de 0,991 na predição.

3 METODOLOGIA

No presente Capítulo, é apresentada a base de dados utilizada e é realizada uma breve introdução sobre os conceitos referentes a teoria de aprendizado de máquina e das redes neurais artificiais do tipo *Multilayer Perceptron* e Redes Convolucionais 1D.

3.1 Base de dados

A base de dados utilizada neste trabalho foi gerada e disponibilizada por Tsanas e Xifara em 2012 e contém oito variáveis de entrada que caracterizam a construção dos edifícios e duas variáveis de saída, conforme segue:

- Variáveis de entrada:

1. X_1 - Compacidade relativa: É um índice utilizado para determinar o grau de compactação de um material em porcentagem. Neste estudo foram consideradas doze formas de construção dos edifícios, todos com o mesmo volume de $771,75 m^3$, mas com superfícies e dimensões diferentes (ver Figura 2).
2. X_2 - Área de superfície: quantidade de espaço que cobre o exterior de uma forma tridimensional, dado em metros quadrados.
3. X_3 - Área da parede: área medida em metros quadrados dada pela multiplicação do comprimento e da altura da parede.
4. X_4 - Área do telhado: área medida em metros quadrados considerando as dimensões do telhado.
5. X_5 - Altura total: altura total do edifício, medida em metros.
6. X_6 - Orientação: posição da construção com relação aos pontos cardeais, as amostras da base de dados são giradas de forma a considerar os 4 pontos cardeais. Para a amostragem atribuiu-se números de 2 a 5, com cada um representando um ponto cardinal.
7. X_7 - Área envidraçada: proporção da área que é coberta por janelas, paredes de vidro, telhados de vidro, etc. Na base de dados há três tipos de áreas envidraçadas expressas como porcentagens da área útil: 10%, 25% e 40%.
8. X_8 - Distribuição da área envidraçada: há cinco diferentes cenários de distribuição para cada percentual de área envidraçada: (1) uniforme: com 25% de envidramento em cada lado, (2) norte: 55% no lado norte e 15% em cada um dos outros lados, (3) leste: 55% no lado leste e 15% em cada um dos outros lados, (4) sul: 55% no lado sul e 15% em cada um dos outros lados, e (5) oeste: 55% no lado oeste e 15% em cada um dos outros lados. Também foram consideradas amostras sem áreas envidraçadas (0).

Esta base de dados foi gerada considerando 12 diferentes estruturas de edifícios residenciais localizados na cidade de Atenas (Grécia), com cada bloco sendo ocupado por sete pessoas realizando atividades sedentárias, com uma média de consumo de 70 W. As condições de projeto na ocasião da coleta dos dados foram definidas da seguinte forma:

- umidade: 60%,
- velocidade do ar: 0,30 m/s,
- nível de iluminação: 300 lux.

Os edifícios simulados são compostos por 18 elementos de $3,5 \times 3,5 \times 3,5$ m, cada um, totalizando um volume igual a 771,75 metros cúbicos, conforme mostra a Figura 1.

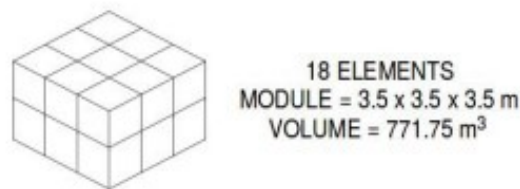


Figura 1 – Geração de formas com base em dezoito elementos cúbicos.

Fonte:[6].

- Variáveis de saída: Para as propriedades térmicas, foi considerada uma eficiência de 95% dos equipamentos de refrigeração/aquecimento para manter a temperatura na faixa de 19–24°C. Sendo a utilização dos equipamentos de 15–20h em dias úteis e de 10–20h aos finais de semana.
 1. Y_1 - Carga de aquecimento: a quantidade de energia térmica que precisaria ser adicionada a um espaço para manter a temperatura em uma faixa aceitável [7].
 2. Y_2 - Carga de resfriamento: quantidade de energia térmica que precisaria ser removida de um espaço a fim de manter a temperatura em uma faixa aceitável [7].

A base de dados contem 768 registros referentes as doze formas de construção (X_1), três valores de área envidraçada (X_7), com cinco valores de distribuição da área envidraçada (X_8) e quatro orientações (X_6), obtendo $12 \times 3 \times 5 \times 4 = 720$ amostras de construção. Além disso, foram consideradas as doze formas de construção para as quatro orientações sem vidros. Portanto, no total foram analisadas $720 + 12 \times 4 = 768$ configurações de construção de edifícios. Na Figura 2, são apresentados os 12 modelos de formatos de edifícios empregados, com os seus respectivos valores de compactidade relativa.

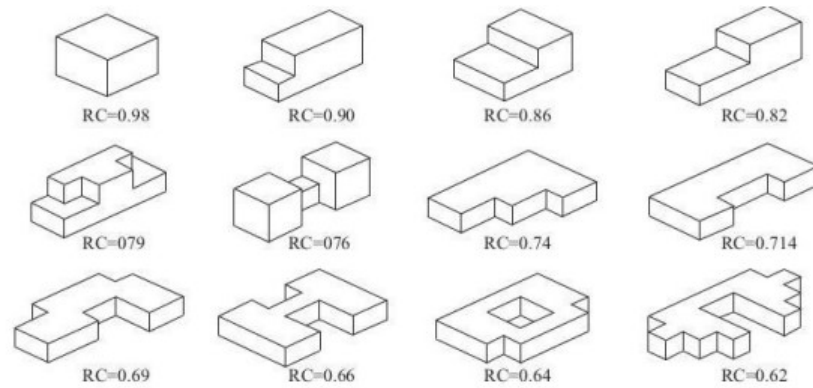


Figura 2 – Variação do coeficiente de compacidade relativa.

Fonte: [8].

A Figura 3 apresenta uma amostra das cinco primeiras e das cinco últimas entradas da base de dados para cada uma das variáveis. Através da observação da base de dados, se nota que para $X6$ e $X8$ ocorre a atribuição de números para identificar cada caso de orientação e distribuição da área envidraçada, como foi dito anteriormente. As medidas de $X1$ e $X7$ são dadas em porcentagem, $X2$, $X3$ e $X4$ em metros quadrados e $X5$ é dada em metros.

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2
0	0.98	514.5	294.0	110.25	7.0	2	0.0	0	15.55	21.33
1	0.98	514.5	294.0	110.25	7.0	3	0.0	0	15.55	21.33
2	0.98	514.5	294.0	110.25	7.0	4	0.0	0	15.55	21.33
3	0.98	514.5	294.0	110.25	7.0	5	0.0	0	15.55	21.33
4	0.90	563.5	318.5	122.50	7.0	2	0.0	0	20.84	28.28
...
763	0.64	784.0	343.0	220.50	3.5	5	0.4	5	17.88	21.40
764	0.62	808.5	367.5	220.50	3.5	2	0.4	5	16.54	16.88
765	0.62	808.5	367.5	220.50	3.5	3	0.4	5	16.44	17.11
766	0.62	808.5	367.5	220.50	3.5	4	0.4	5	16.48	16.61
767	0.62	808.5	367.5	220.50	3.5	5	0.4	5	16.64	16.03

Figura 3 – Ilustração das cinco primeiras e cinco últimas amostras das variáveis de entrada e saída da base de dados.

As simulações foram feitas utilizando a ferramenta Ecotect. Este software realiza análise ambiental e foi usado para realizar uma análise preliminar abrangente do desempenho energético dos edifícios. O Ecotect possui uma interface de modelagem 3D relativamente

simples e intuitiva, com uma variedade de funções de análise e uma exibição visual e interativa que permite análises de resultados e sua apresentação no contexto da modelagem de construção [9]. Ele é utilizado principalmente por arquitetos, durante os estágios iniciais e conceituais do projeto. Uma descrição mais detalhada da geração da base de dados está disponível em [10].

As Figuras 4 a 11 apresentam os boxplots das variáveis de entrada do modelo e permitem visualizar o comportamento e a distribuição estatística destas variáveis. O centro da distribuição mostra a mediana e é indicado pela linha que está dentro do centro do quadrado. Pela amplitude do gráfico, se observam os valores máximo (onde a haste termina) e mínimo (onde a haste começa). Quanto maior for a amplitude, maior a variação nos dados.

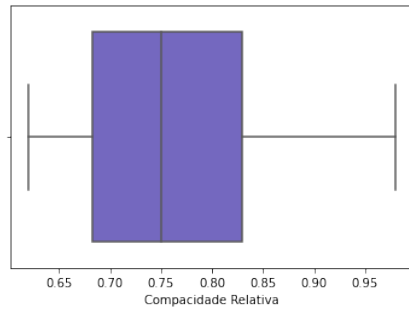


Figura 4 – Compacidade relativa.

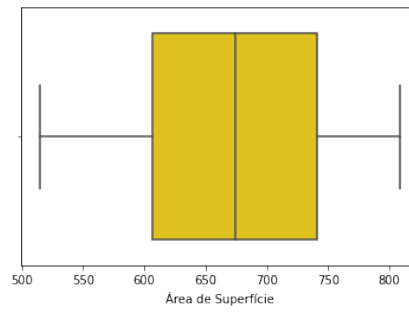


Figura 5 – Área de superfície.

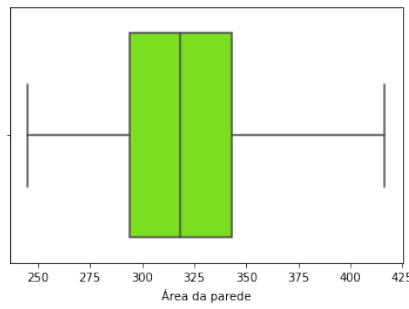


Figura 6 – Área da parede.

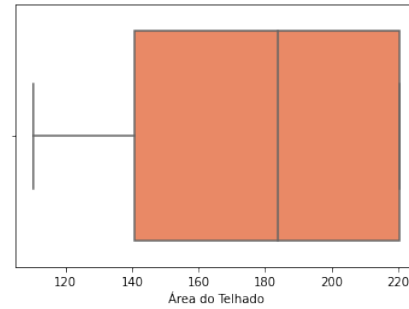


Figura 7 – Área do telhado.

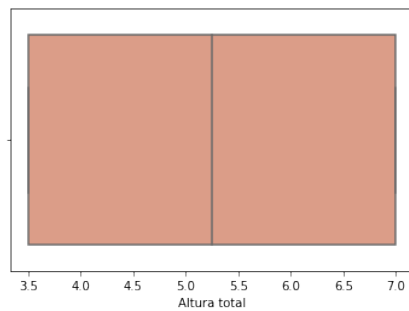


Figura 8 – Altura total.

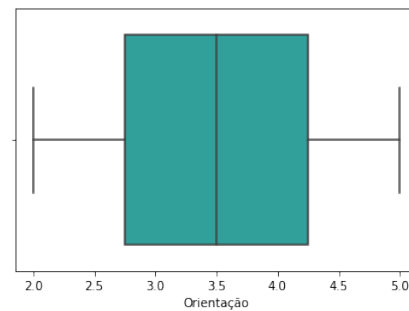


Figura 9 – Orientação.

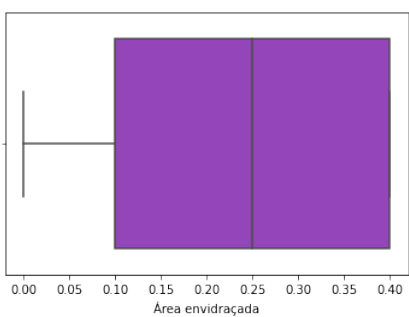


Figura 10 – Área envidraçada.

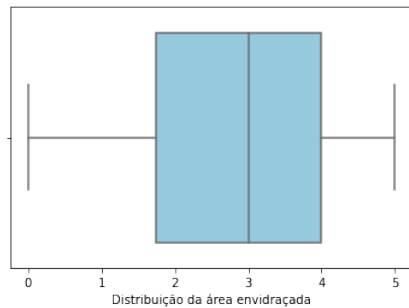


Figura 11 – Distribuição da área envidraçada.

Figura 12 – Boxplot das variáveis de entrada.

Com base nos boxplots, é possível notar que as variáveis possuem desvio padrão pequeno e se encontram concentradas em torno da média. As variáveis "Orientação" e "Altura Total" se encontram uniformemente distribuídas, com média de 3.5 e 5.25, respectivamente. Na Figura 13 apresenta a matriz de correlação dos dados de entrada e saída.

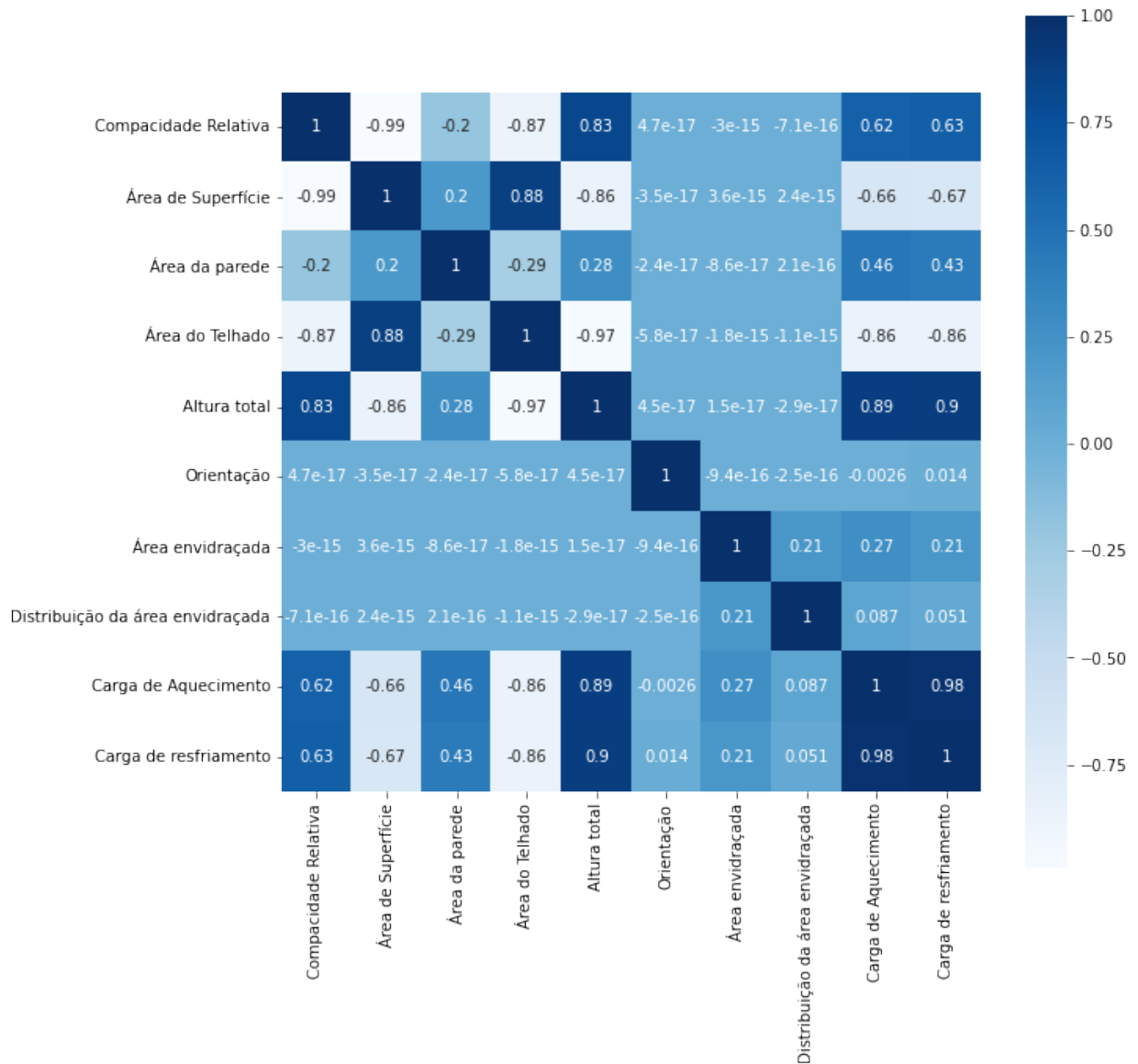


Figura 13 – Matrix de correlação dos dados.

O coeficiente de correlação fornece a relação entre diferentes variáveis e o comportamento de cada variável em um cenário onde a outra está variando, quantificando essa relação. Esse coeficiente pode variar entre -1 (correlação negativa) e 1 (correlação positiva) [11]. Se o coeficiente for igual a 0, significa que as variáveis são descorrelacionadas, ou seja, uma não implica na outra. O coeficiente de correlação ρ pode ser calculado da seguinte

maneira :

$$\rho = \frac{cov(X,Y)}{\sigma_X\sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X\sigma_Y} \quad (3.1)$$

sendo X e Y duas variáveis aleatórias, $cov(X,Y)$ é a covariância entre elas, dada pelo cálculo do operador esperança $E(.)$. Enquanto que μ_X e μ_Y são as médias das variáveis e σ_X e σ_Y são seus desvios padrões [11].

A matriz de correlação apresentada na Figura 13, mostra que praticamente todas as variáveis de entrada impactam de maneira simultânea e equivalente sobre as duas variáveis de saída (cargas de aquecimento (Y_1) e resfriamento (Y_2)). As variáveis com maior índice de correlação são (em ordem decrescente): X_5 (Altura total), X_4 (Área do Telhado), X_2 (Área de superfície), X_1 (Compacidade relativa), X_3 (Área da parede) e X_7 (Área envidraçada). As variáveis X_8 (Distribuição da área envidraçada) e X_6 (Orientação) apresentam baixa correlação ($\rho < 0.1$) com as variáveis de saída. As variáveis de saída são diretamente correlacionadas entre si com ($\rho = 0.98$).

3.1.1 Padronização da base de dados

Para muitos estimadores de aprendizado de máquina, é necessário realizar a padronização de um conjunto de dados, visto que eles podem se comportar mal se os recursos individuais não se parecem mais ou menos com dados padrão normalmente distribuídos[12].

A Figura 3 apresenta as cinco primeiras e cinco últimas amostras, há grande diferença na ordem de grandeza, quando se comparam os dados no início e final da base de dados, o que também pode ser observado através da análise dos boxplots das variáveis, na Figura 12.

Através dessa análise, conclui-se que há a necessidade de normalização dos dados para um melhor condicionamento das variáveis de entrada na rede neural. Dessa forma, a base de dados foi normalizada para assumir valores no intervalo de 0 a 1. Este procedimento foi executado com a função *MinMaxScaler()* da biblioteca Sklearn do Python. Essa transformação é frequentemente usada como uma alternativa ao escalonamento de média zero e variância unitária e é calculada através da expressão [12]:

$$X_{std} = \frac{(X - X.min(axis = 0))}{(X.max(axis = 0) - X.min(axis = 0))} \quad (3.2)$$

$$X_{scaled} = X_{std} * (max - min) + min \quad (3.3)$$

onde X_{std} transforma os dados e os coloca na mesma escala.

X_{min} e X_{max} são respectivamente os valores mínimos e máximos e min , max corresponde ao intervalo desejado de dados em escala, sendo que o intervalo padrão para o recurso retornado por *MinMaxScaler* é de 0 a 1.

3.2 Redes neurais e aprendizado supervisionado

3.2.1 Redes neurais

Segundo Haykin (2001), uma rede neural é uma máquina projetada para modelar a maneira como o cérebro realiza uma função de interesse ou tarefa particular. A rede neural é implementada através da utilização de componentes eletrônicos ou de simulações. A fim de alcançar um desempenho satisfatório, as redes neurais são compostas de uma interligação maciça de células computacionais, conhecidos como "neurônios" ou "unidades de processamento", que são utilizados para armazenar o conhecimento adquirido [13].

Os algoritmos de aprendizagem tem como função modificar os pesos sinápticos da rede de forma ordenada, a fim de alcançar um objetivo de projeto desejado. A principal característica de uma rede neural é sua capacidade de aprender utilizando os dados que tem à sua disposição e, dessa forma, melhorar o desempenho do modelo. A rede se torna mais instruída sobre o seu ambiente após cada iteração do processo de aprendizagem [13]. A rede neural é estimulada pelo ambiente e, devido a isso, sofre modificações em seus parâmetros livres, respondendo de uma maneira nova ao ambiente como consequência das modificações ocorridas em sua estrutura interna.

As redes neurais artificiais são estruturas bem simplificadas do sistema biológico e não se aproximam da complexidade do cérebro, mas podem ser treinadas para solucionar diversos problemas reais que envolvam, por exemplo, predição e classificação de dados [14].

3.2.2 Aprendizado supervisionado

Nesta abordagem se dispõe de uma base de dados que possui entradas rotuladas com a correspondente saída esperada. A aprendizagem consiste na criação de um modelo que se ajuste a estes dados. O modelo tem a finalidade de aprender a relação que existe entre os dados de entrada/saída e determinar as saídas que correspondem a novos dados de entrada ao invés de programar uma resposta para todas as possíveis entradas [14].

As técnicas de aprendizado supervisionado de máquina são compostas por [14]:

1. Um conjunto de dados, que pode ser particionado para treinamento, validação e teste.
2. Um modelo matemático que expressa o comportamento de entrada/saída.
3. Um critério para medir o desempenho do modelo, como o erro quadrático médio e a taxa de erro de classificação.

O algoritmo de aprendizado aprende pela saída da função de custo, a qual auxilia o algoritmo a encontrar um método que aprenda de maneira eficiente e precisa, permitindo

medir a diferença entre a resposta correta e a retornada pelo algoritmo [13]. Algumas das formas de se calcular a função de custo correta, são:

1. Erro quadrático médio (MSE): consiste na soma dos valores absolutos da diferença entre os valores reais e os valores previstos pelo algoritmo.

$$C(w) = \frac{1}{N} \sum_x^N \|y(x) - r\|^2 \quad (3.4)$$

onde $y(x)$ corresponde a saída fornecida pelo algoritmo quando recebeu a entrada x e, r corresponde ao valor desejado.

2. Erro médio absoluto (MAE): fornece a média de diferença absoluta entre a predição do modelo e o valor de destino.

$$C(w) = \frac{1}{N} \sum_x^N |y(x) - r| \quad (3.5)$$

3. Raiz do erro quadrático médio (RMSE): é calculada através da raiz quadrada da Equação 3.4 e têm o benefício de penalizar os erros de maior magnitude. O RMSE mensura o quão bem o modelo é capaz de prever a resposta, constituindo o critério mais importante para o ajuste quando o principal objetivo é usar o modelo para fins preditivos.

A diferença de valores entre as métricas descritas auxilia no diagnóstico de predições não acuradas ou outliers, o que pode ser observado se o valor de RMSE for muito maior que o de MAE.

Nas redes neurais do tipo feedforward, a informação se move dos nós de entrada em direção aos nós de saída, passando pelas camadas ocultas (se houver). Esse tipo de rede não possui conexões entre os neurônios que formem ciclos ou loops. Quanto maior o número de neurônios na camada intermediária, maior a flexibilidade do modelo matemático resultante e maior a complexidade da rede, podendo haver o risco de sobreajuste dos dados, o que gera modelos que generalizam mal para dados novos [14].

3.3 Seleção de características

Os métodos de seleção de características têm por objetivo determinar quais variáveis da entrada são mais relevantes para a determinação da saída da rede neural, auxiliando na redução do número de entradas [15]. Ao analisar a base de dados, foi observado que as variáveis X_8 (Distribuição da área envidraçada) e X_6 (Orientação) apresentavam baixa correlação em relação às variáveis de saída. Isso levou à suposição de que essas variáveis não teriam tanta relevância na obtenção da saída, sendo redundantes e não contribuindo significativamente na obtenção da saída, o que pode tornar o treinamento das RNAs mais difícil e demorado.

Com base nisso, foram testados quatro cenários:

1. Cenário 1: Remoção das variáveis X_6 e X_8
2. Cenário 2: Remoção de X_6
3. Cenário 3: Remoção de X_8
4. Cenário 4: Remoção de X_6 , X_7 e X_8

A variável X_7 (Área envidraçada) apresentou correlação 0,27 e 0,21 quando comparada à Y_1 e Y_2 , respectivamente, o que não é considerado um valor alto, justificando a testagem de sua remoção do modelo.

Após a testagem dos cenários mencionados acima, verificou-se que para a rede CNN, a melhor opção foi o Cenário 1 e, para a rede MLP, a melhor opção foi o Cenário 2. Em ambos os casos, os resultados dos Cenários 1, 2 e 3 foram bem similares.

A suposição de que X_7 não contribua significativamente com as saídas se mostrou infundada, visto que os resultados no Cenário 4 foram muito piores do que os demais.

3.4 Modelos preditivos

Os modelos preditivos, em geral, servem para identificar padrões e prever a saída de acordo com os dados analisados. A fim de identificar padrões e oferecer uma previsão do modelo, a rede neural é testada e pode ser usada com dados de teste ou novos dados. Algumas das funções que permitem encontrar a saída da rede neural artificial projetada são: [14]

1. *model.evaluate()*: Fornece os valores de perda e métricas para os dados de entrada considerando uma etapa de teste.
2. *model.predict()*: Gera a saída da rede neural para os dados de entrada.

Foi utilizada a função *model.predict()* nas redes testadas.

Antes de iniciar o treinamento do modelo de rede neural, é necessário configurar o processo de aprendizagem, compilando o modelo. Para isso é necessário definir a técnica de otimização, que é responsável por definir o modo como o algoritmo vai encontrar os ajustes certos para os pesos [14]. Para isso, são usados otimizadores. Na construção do modelo, foram testados os otimizadores:

- ADAM: É o otimizador mais utilizado para o aprendizado de máquina, sua sigla significa "Adaptive Moment Estimation" e baseia-se no gradiente descendente de funções objetivas estocásticas, empregando a média e variância de gradientes consecutivos para obter a direção de atualização dos coeficientes [14].

- RMSprop: Usa a média móvel de gradientes quadrados para normalizar o gradiente e equilibrar o tamanho do passo (taxa de aprendizado), usando uma taxa adequada a cada iteração, diminuindo o passo para gradientes grandes a fim de evitar a explosão e aumentando o passo para gradientes pequenos para evitar o desaparecimento [14].
- Adamax: É uma variação do Adam baseada na norma do infinito, sendo às vezes superior ao Adam, principalmente em modelos com embeddings [16]. Este otimizador adapta automaticamente um tamanho de passo separado (taxa de aprendizado) para cada parâmetro no problema de otimização.
- Nadam: O otimizador Nadam pode ser descrito como o Adam com impulso Nesterov aplicado [16]. Funciona adicionando impulso à descida do gradiente que incorpora alguma inércia às atualizações, esse processo é realizado na nova posição projetada, o que é chamado de Gradiente Acelerado de Nesterov (NAG) ou momento de Nesterov [17].

Com todos os testes realizados, observou-se que o otimizador Adam apresentou melhor desempenho, o que já era esperado, visto que é o mais utilizado em problemas de aprendizado de máquina.

3.4.1 Rede MLP

Segundo [18], uma rede neural MLP consiste em unidades dispostas em camadas. Cada camada é composta por nós que se conectam ao nó das camadas subsequentes. Cada MLP é composto por um mínimo de três camadas, sendo elas uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Os nós de entrada têm funções de ativação lineares e sem limites. Cada nó de unidade oculto, assim como cada nó de saída possuem limites associados a eles, além dos pesos. Os nós de unidade ocultos têm funções de ativação não lineares e as saídas têm funções lineares de ativação. Dessa forma, cada sinal que alimenta um nó em uma camada subsequente tem a entrada original multiplicada por um peso com um limite adicionado e então é passado uma função de ativação que pode ser linear ou não linear (unidades ocultas).

As redes neurais do tipo MLP podem ser divididas em três partes, sendo elas:

- Camada de entrada: recebe as informações do bando de dados
- Camada oculta ou intermediária: extrai as informações necessárias dos dados
- Camada de saída: responsável por fornecer o resultado final da rede

A Figura 14 apresenta uma rede neural rasa, ou seja, ela possui apenas uma camada oculta de neurônios entre os dados de entrada e os neurônios de saída da rede.

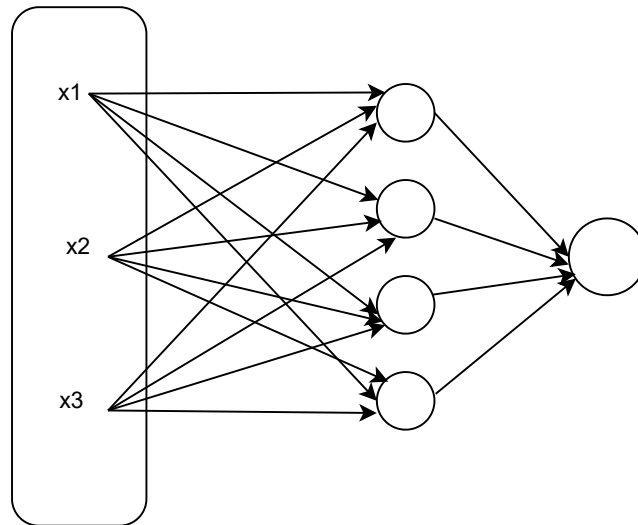


Figura 14 – Rede Neural Rasa.

3.4.2 Rede CNN

Conhecida como Rede Neural Convolutiva, consiste em uma rede neural do tipo *feedforward* que contém muitas camadas e é explorada no contexto de deep learning. Essa rede consiste na alternância entre dois tipos diferentes de camadas - convolutiva e pooling (agrupamento ou subamostragem) - e seu último estágio é feito de uma ou mais camadas densas (totalmente conectadas). Todas as camadas realizam operações não-lineares e o ajuste dos pesos sinápticos é realizado por técnicas de otimização não-linear, empregando a retropropagação do erro[14].

A rede CNN é composta pelas camadas:

- Camada convolutiva: nessa camada, os neurônios funcionam como filtros que são aplicados aos dados de entrada, sendo um filtro uma matriz de pesos. Segundo Goodfellow (2016) , as camadas convolucionais incluem os seguintes hiper parâmetros:
 - Tamanho do *kernel*: tamanho da janela usada para a convolução.
 - *Stride*: tamanho do passo do *kernel* ao deslizar a janela.
 - *Padding*: define como a borda da amostra é tratada.
- Camada de *pooling*: reduz a dimensionalidade de cada mapa de *features*, retendo as informações mais importantes. Dessa forma é possível criar representações de entrada menores e mais gerenciáveis, fazendo assim, o número de parâmetros e cálculos na rede reduzirem. Após a aplicação da ReLU (operação que substitui todos os valores de pixels negativos no mapa por zero), as redes convolucionais fazem o max pooling

que sintetiza a informação copiando o valor máximo observado na região analisada [19]. Também é utilizado average pooling, que calcula o valor médio para patches de um mapa e o usa para criar um mapa com amostragem reduzida.

- Camada completamente conectada: Geralmente encontrada ao final da arquitetura de uma CNN, nessa camada, todas as entradas se conectam a todos os neurônios. Logo após essa camada, é comum a presença da operação de ‘flatten’, que realiza a conversão da matriz da saída da última camada de pooling em um vetor de dados com uma única coluna. Este vetor de dados serve como entrada para a camada totalmente conectada da rede neural [14].
- Camada de *Dropout*: Devido à enorme flexibilidade associada aos mapeamentos de entrada-saída que podem ser realizados por uma CNN profunda, o *overfitting* se torna relevante. Para atenuá-lo emprega-se a camada de *dropout*, que é usada apenas na fase de treinamento e é retirada na fase de teste. O *dropout* consiste na eliminação aleatória de neurônios da rede neural, juntamente com suas conexões sinápticas, forçando a saída a zero com uma probabilidade fixa. Dessa forma, a rede neural é forçada a ser redundante e conseguir classificar corretamente mesmo na ausência de alguns neurônios, aprendendo a não depender de nenhum neurônio, dessa forma, evitando o sobreajuste [14].

A função de ativação ReLU, produz um erro de treinamento menor se comparado com as outras funções, devido a não ser tão suscetível ao problema da dissipação dos gradientes. Além disso, essa função de ativação reduz o tempo de convergência dos parâmetros, pois resulta na função identidade para valores positivos.

Além da camada ReLU, são comumente utilizadas as camadas Sigmoide e TanH.

- Sigmoide: É útil para executar classificações com múltiplas saídas e seu intervalo é limitado entre 0 e 1. A função sigmoide sofre da dissipação do gradiente, o que estagna o aprendizado da rede em algumas circunstâncias. [14]
- TanH: É uma função do tipo sigmoide que varia de -1 a 1 . O gradiente da função tanH permite detectar pequenas diferenças, tornando a classificação mais sensível. A função tanH também sofre com o problema de dissipação do gradiente.[14]

A Figura 15 mostra uma CNN de reconhecimento de dígitos, que tem como entrada uma imagem com dimensões 32×32 e faz uma convolução com matrizes quadradas de ordem 5, que resulta no C_1 , que é o primeiro mapa de características com dimensões 28×28 . Logo após, é feito o Max-Pooling que tem a função de pegar os valores mais altos, resultando em S_1 com dimensões 14×14 . Com o resultado de S_1 , a rede faz uma nova convolução, que resulta em C_2 e um novo Max-Pooling que resulta em S_2 . A última etapa

é a camada totalmente conectada que recebe os neurônios de saída representados pelos círculos verdes e os classifica de acordo com as classes estabelecidas no treinamento [19].

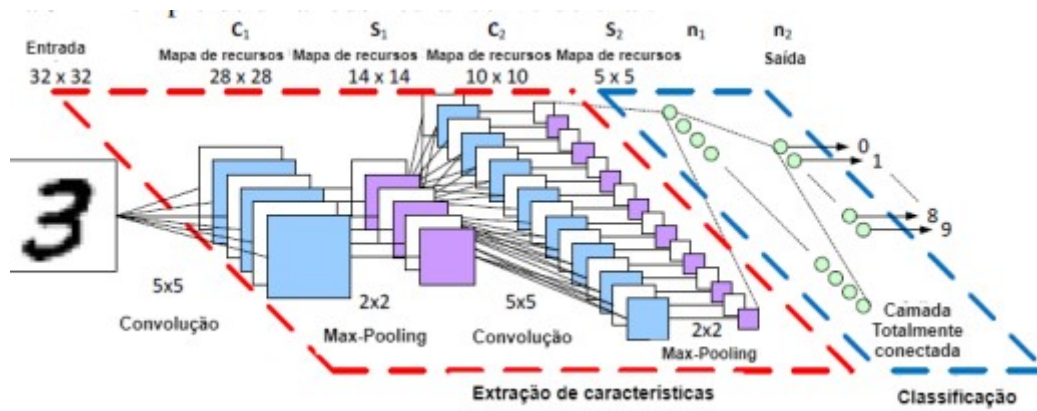


Figura 15 – Exemplo de rede CNN.

Fonte: [19]

4 RESULTADOS

4.1 Cenários testados

A fim de realizar comparações, em ambas as análises, foram realizados 5 testes com cada conjunto de parâmetros, com 500 épocas em cada execução do código, a fim de obter resultados mais precisos.

Na Tabela 1, são apresentados alguns dos cenários testados sem a seleção das variáveis, que será tratada posteriormente.

Tabela 1 – Parâmetros e configurações testadas.

Parâmetros	CNN	MLP
Nº de Camadas densas	3, 4	3
Nº de Camadas ocultas	1, 2	-
Otimizador	relu, sigmoid, tanh	relu, sigmoid, tanh
Nº de neurônios	120, 200, 220, 240	120, 200, 220, 240
Nº do kernel	4, 6, 8	4, 6, 8
Função de ativação	Adamax, Adam Nadam, RMSProp	Adamax, Adam Nadam, RMSProp
Métrica	MAE, MSE, MAPE	MAE, MSE, MAPE

4.2 Implementação da rede CNN

4.2.1 Modelo sem seleção de variáveis

Foram realizados diversos testes em dois modelos:

- Modelo 1: Sem camada de *pooling*
- Modelo 2: Foram testadas *Maxpooling* e *AveragePooling*, com tamanho de agrupamento (*pool size*) de 2 e 3.

Como mencionado anteriormente, a camada de *pooling* tem como função reduzir a dimensionalidade da matriz de *feature*, retendo as informações mais importantes e criando representações de entrada menores e mais gerenciáveis, diminuindo o número de parâmetros e cálculos na rede. Os testes feitos com o Modelo 2 obtiveram melhor resultado com a utilização de *Maxpooling* do que com *AveragePooling*. Entretanto, os resultados do Modelo 1 foram mais satisfatórios, o que levou à conclusão de que o agrupamento causou perda de dados, reduzindo o desempenho do modelo.

A Tabela 2 apresenta a configuração da rede CNN no melhor cenário testado. Para a obtenção do melhor resultado, foi utilizado o Modelo 1 com a função de ativação "Adam" e

o otimizador "MAE". Considerando os casos testados apresentados na Tabela 1, optou-se por empregar apenas uma camada oculta, visto que a diferença de resultados foi mínima, não tornando justificável a inserção de uma nova camada no modelo.

Tabela 2 – Melhor seleção de parâmetros para rede CNN.

Camada	Função de Ativação	Nº de neurônios/filtros	Tamanho do Kernel
Convolutacional 1	sigmoid	200	8
Convolutacional 2	sigmoid	200	8
Convolutacional 3	sigmoid	200	8
Densa 1	relu	120	
Densa 2	linear	120	
Densa 3	tanh	120	
Densa 4	linear	120	

O cenário de teste da Tabela 2 apresenta um modelo composto por 683.302 parâmetros. A Tabela 3 apresenta a média de cinco resultados obtidos para o modelo representado.

Tabela 3 – Resultados obtidos sem seleção de variáveis para rede CNN.

Perda	$Y_1 - MAE$	$Y_1 - RMSE$	$Y_2 - MAE$	$Y_2 - RMSE$
0.7227	0.2759	0.3784	0.4468	0.5867

As Figuras 16 a 19 apresentam o desempenho do modelo através de gráficos que mensuram a diferença entre o RMSE, que foi a métrica usada no treinamento do modelo, e se encontra em azul, e na validação do modelo, e se encontra em verde.

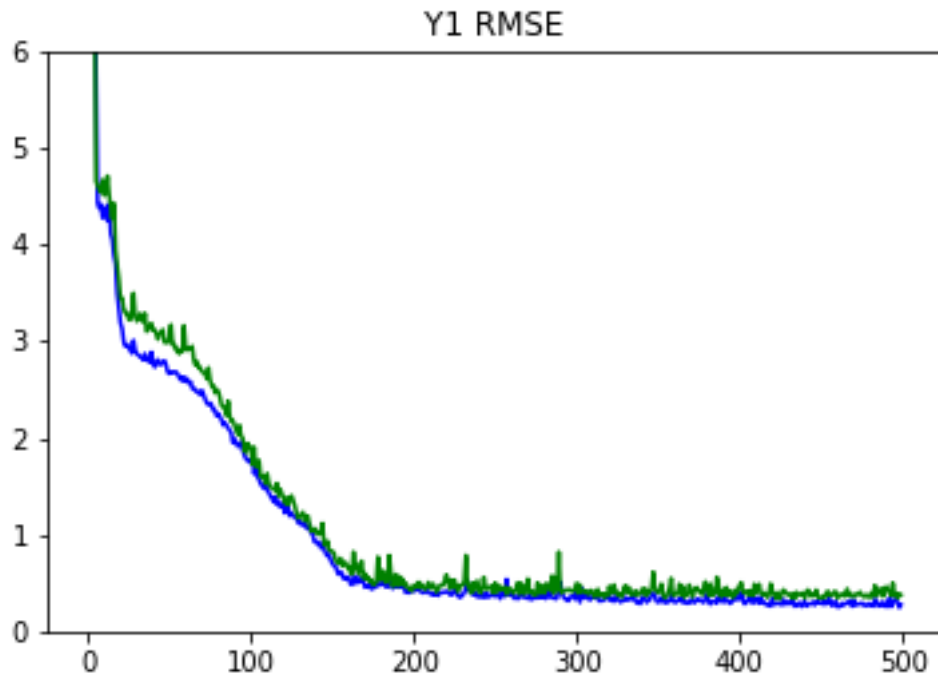


Figura 16 – RMSE ao longo das épocas para Y_1 no método CNN sem seleção.

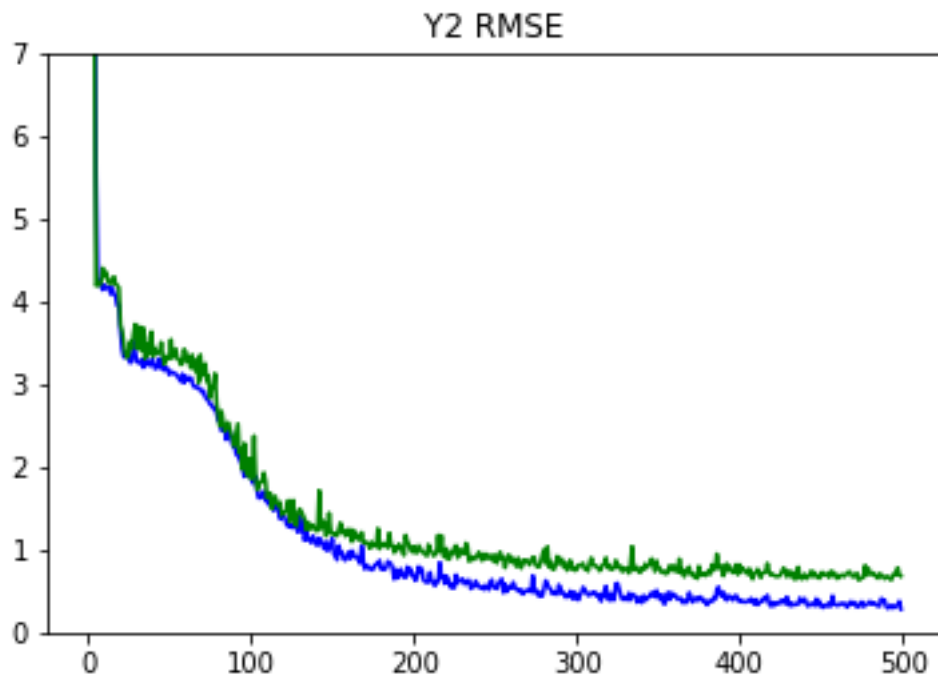


Figura 17 – RMSE ao longo das épocas para Y_2 no método CNN sem seleção.

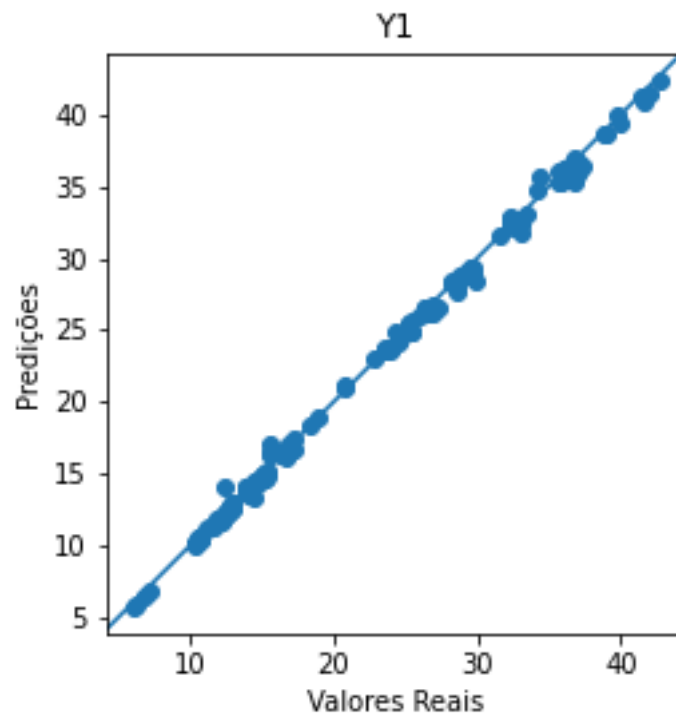


Figura 18 – Diferença entre valores preditos e reais para Y_1 no método CNN sem seleção.

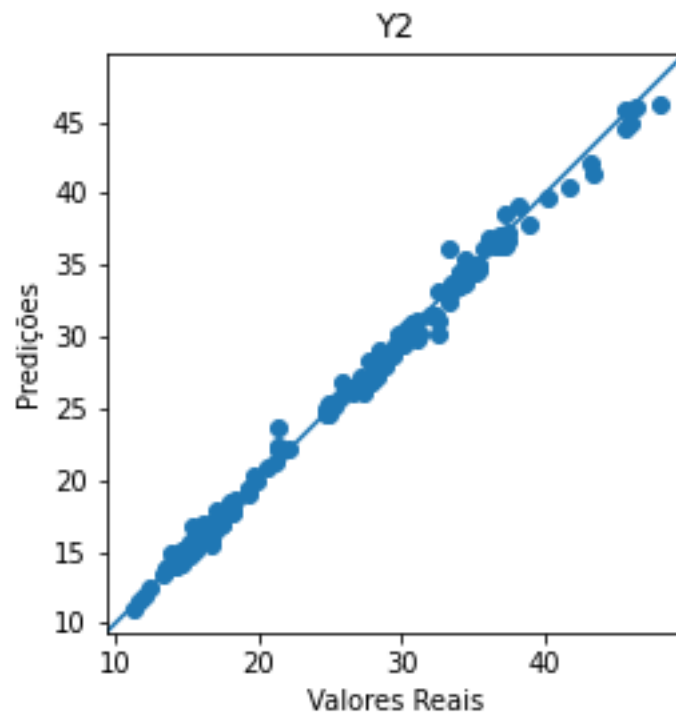


Figura 19 – Diferença entre valores preditos e reais para Y_2 no método CNN sem seleção.

As Figuras 16 e 17 apresentam os valores de RMSE para as variáveis de saída,

indicando que o modelo resultou em um bom ajuste de Y_1 , mas que a capacidade de prever a resposta de Y_2 não foi tão satisfatória, o que pode ser observado na diferença entre as curvas do modelo e dos valores reais apresentados na Figura 17.

Através da análise das Figuras 18 e 19 se observa uma proximidade entre os valores preditos e o valor real, com alta correlação, demonstrando que foi feito um bom ajuste do modelo, principalmente para Y_1 , onde há poucos pontos fora da curva preditiva.

4.2.2 Modelo sem as variáveis X_6 e X_8

A seleção de variáveis foi implementada, conforme descrito na Seção 3.3, com a eliminação das variáveis X_8 (Distribuição da área envidraçada) e X_6 (Orientação), devido à baixa correlação em relação às variáveis de saída. Para a execução dos testes, utilizou-se a melhor configuração encontrada, que é apresentada na tabela 2.

Os testes realizados com a eliminação da variável X_7 (Área envidraçada), geraram resultados inferiores e devido a isso os resultados desse teste não foram apresentados no presente trabalho.

A Tabela 4 apresenta a média de cinco resultados obtidos para o modelo com eliminação das variáveis X_6 e X_8 .

Tabela 4 – Resultados obtidos com seleção de variáveis para rede CNN.

Perda	$Y_1 - MAE$	$Y_1 - RMSE$	$Y_2 - MAE$	$Y_2 - RMSE$
1.6318	0.3378	0.4716	1.2939	2.1240

As Figuras 20 a 23 apresentam o desempenho do modelo através de gráficos que mensuram a diferença entre o RMSE, que foi a métrica usada no treinamento do modelo, e se encontra em azul, e na validação do modelo, e se encontra em verde

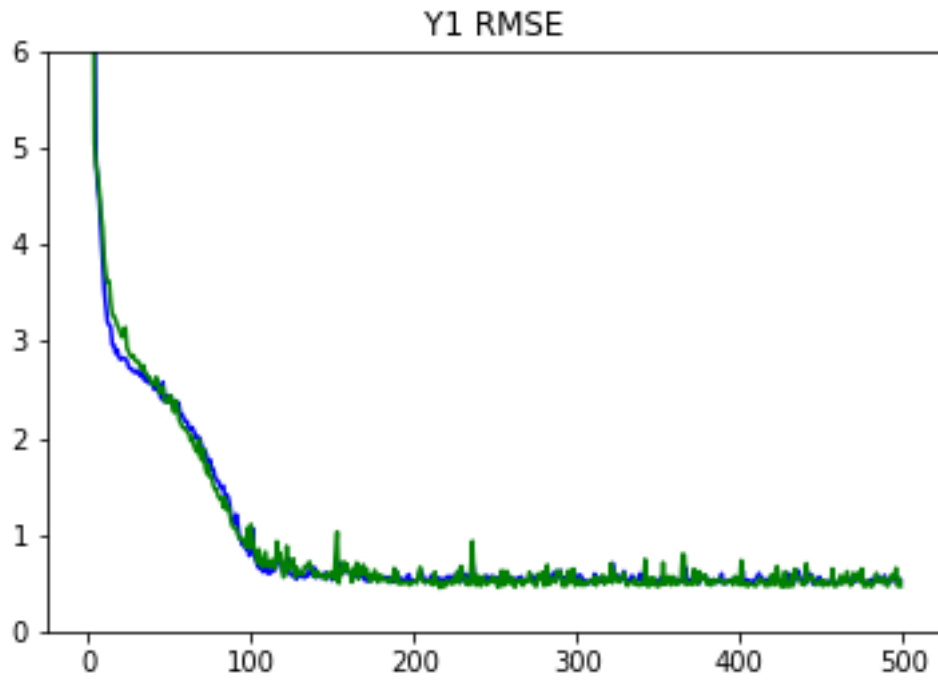


Figura 20 – RMSE para Y_1 no método CNN com seleção

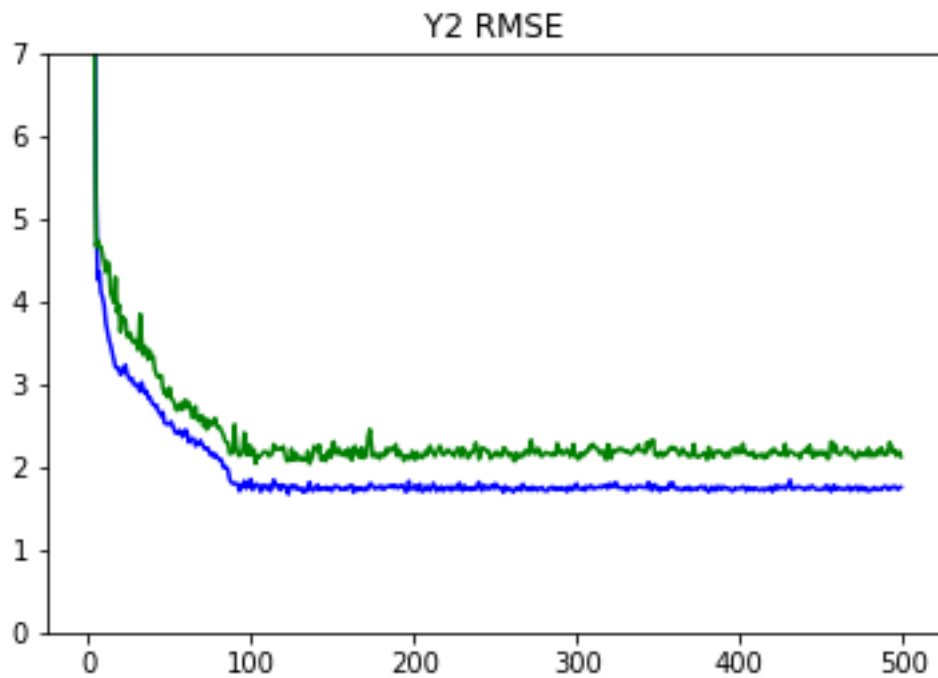


Figura 21 – RMSE para Y_2 no método CNN com seleção.

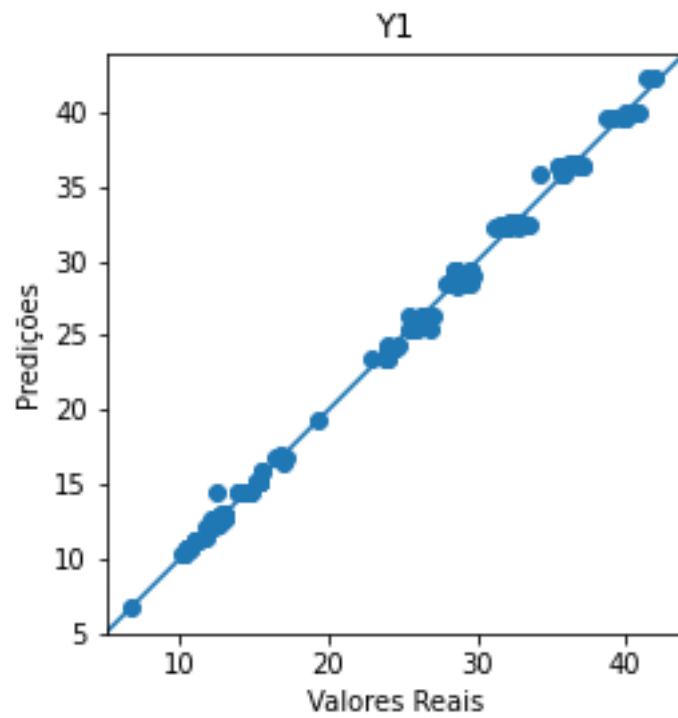


Figura 22 – Diferença entre valores preditos e reais para Y_1 no método CNN com seleção.

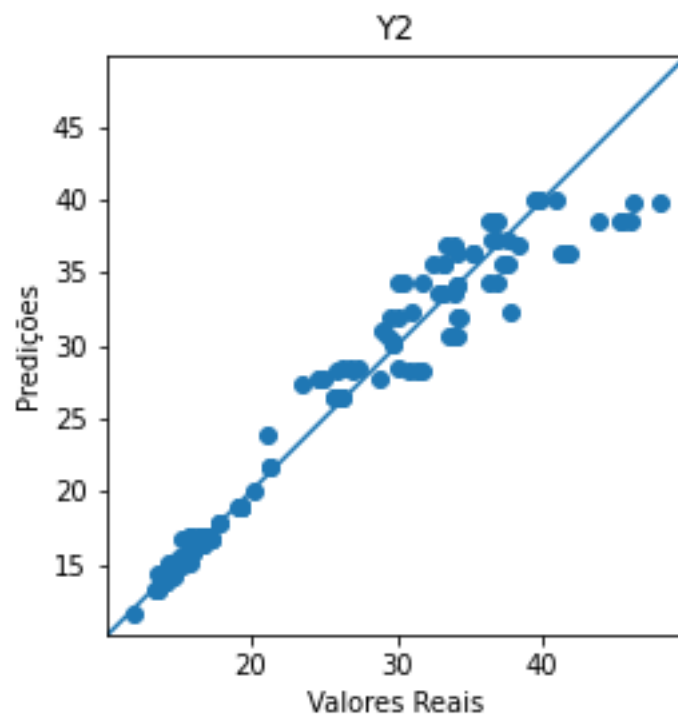


Figura 23 – Diferença entre valores preditos e reais para Y_2 no método CNN com seleção.

Comparando as Figuras 20 e 22 as as Figuras 16 e 18 se observa que o modelo com

seleção apresentou resultados similares ao do modelo sem seleção para a saída Y_1 (carga de aquecimento), enquanto que para a saída Y_2 (cargas de resfriamento) o modelo com todas as variáveis apresentou um desempenho melhor, visto que as Figuras 20, 23 e a Tabela 4 indicam um erro maior para a predição quando houve seleção de características.

4.3 Implementação da rede MLP

4.3.1 Modelo sem seleção de variáveis

Após vários testes, concluiu-se que a melhor configuração para o modelo é a que segue na Tabela 5.

Tabela 5 – Melhor seleção de parâmetros para a rede MLP.

Camada	Função de Ativação	Nº de neurônios/filtros	Tamanho do Kernel
Densa 1	relu	280	8
Densa 2	relu	280	8
Densa 3	relu	280	8

A configuração da Tabela 5 gerou um modelo com 160.442 parâmetros, cujo resultado da execução pode ser visto na Tabela 6.

Tabela 6 – Resultados obtidos sem seleção de variáveis para a rede MLP.

Perda	$Y_1 - MAE$	$Y_1 - RMSE$	$Y_2 - MAE$	$Y_2 - RMSE$
0.6455	0.2845	0.4707	0.3610	0.4950

Nas Figuras 24 a 27 é possível observar o desempenho do modelo através da análise do RMSE e da diferença entre o valor real e a predição realizada pelo modelo.

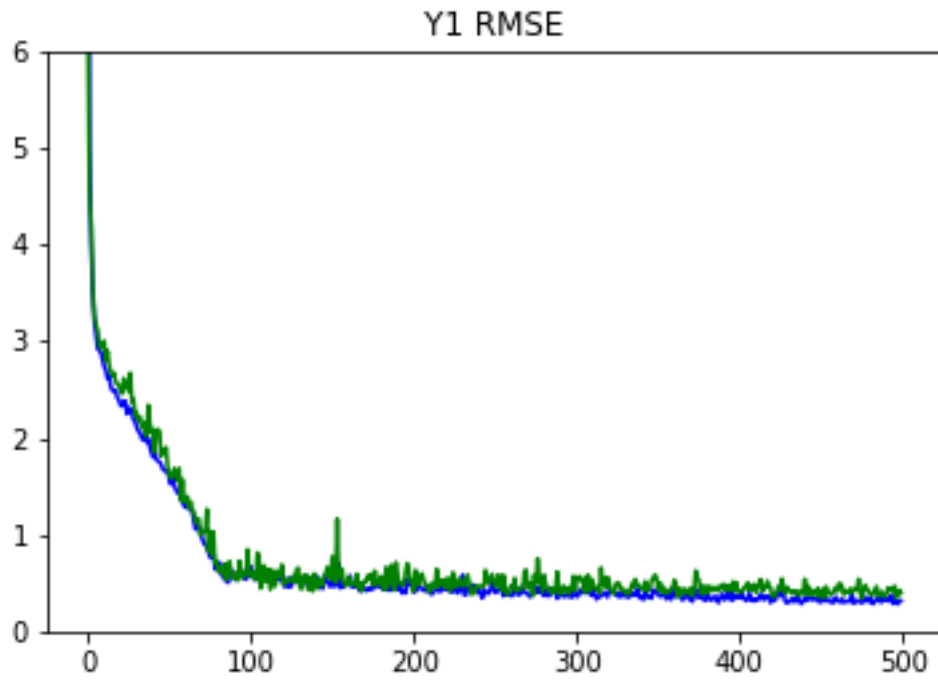


Figura 24 – RMSE para Y_1 no método MLP sem seleção.

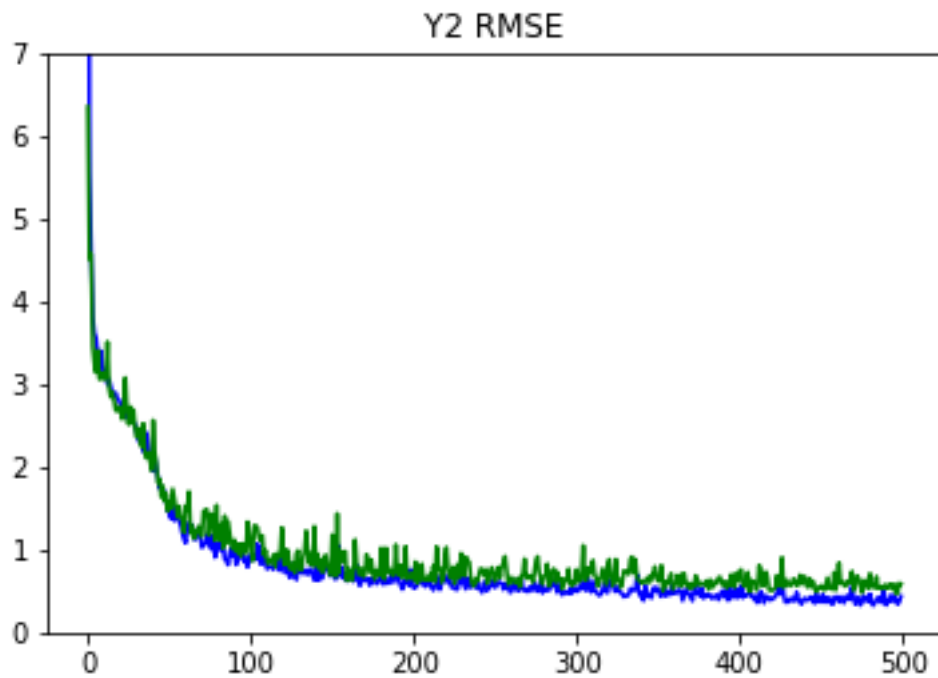


Figura 25 – RMSE para Y_2 no método MLP sem seleção.

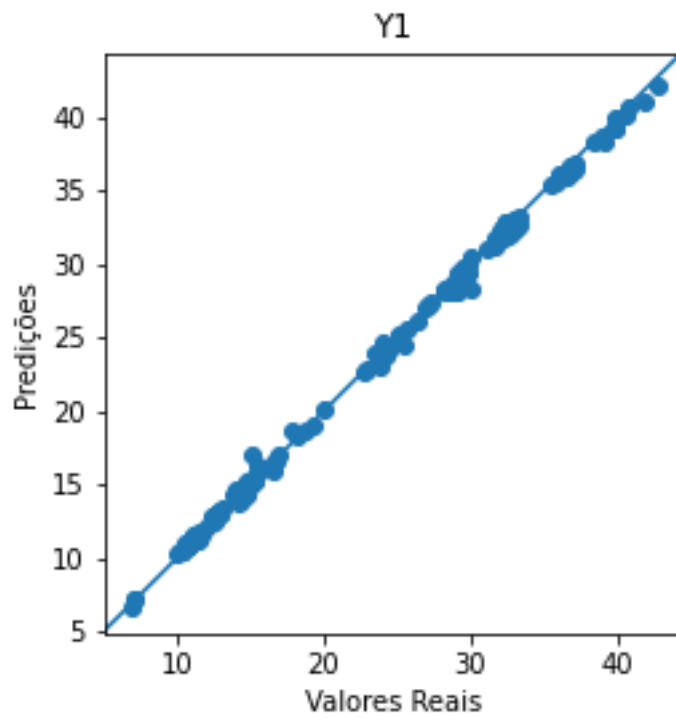


Figura 26 – Diferença entre valores preditos e reais para Y_1 no método MLP sem seleção.

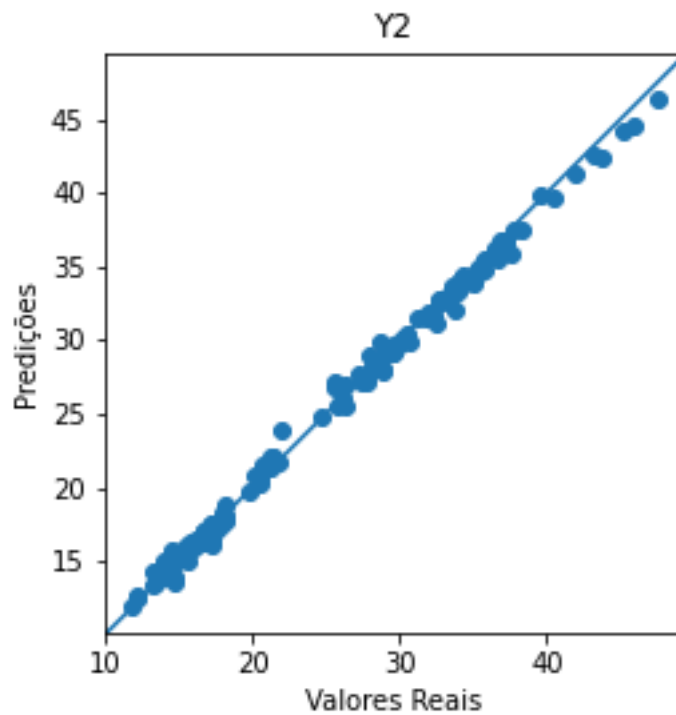


Figura 27 – Diferença entre valores preditos e reais para Y_2 no método MLP sem seleção.

Comparando o desempenho da CNN com a MLP, se nota que a rede MLP apresenta

um melhor desempenho, sendo capaz de realizar uma predição mais acurada para a variável Y_2 . Em relação à variável Y_1 , os dados foram bastante similares entre os dois métodos.

Levando em consideração que a quantidade de parâmetros empregada na execução da técnica CNN foi 4.2 vezes maior do que a MLP, e de que foram usadas menos camadas e, por isso, menos parâmetros no método com MLP, conclui-se que a rede MLP apresenta melhor resultado. Além disso, a obtenção de um bom resultado com o método MLP foi obtido de maneira mais rápida, necessitando de apenas um modelo, ao contrário da rede CNN, que só alcançou um resultado satisfatório após vários testes com dois modelos diferentes.

4.3.2 Modelo sem a variável X_6

Conforme descrito na Seção 3.3, os testes realizados com seleção de variáveis resultou em um modelo com erros menores quando apenas a variável X_6 foi eliminada. Devido a isso, foram realizados testes com a configuração descrita em 5 considerando a exclusão da variável X_6 . A média dos resultados obtidos na execução desse caso segue na Tabela 7.

Tabela 7 – Resultados obtidos com seleção de variáveis para a rede MLP.

Perda	$Y_1 - MAE$	$Y_1 - RMSE$	$Y_2 - MAE$	$Y_2 - RMSE$
1.7118	0.3986	0.5930	1.3132	2.3443

As Figuras 28a 31 apresentam o desempenho do modelo através de gráficos que mensuram a diferença entre o RMSE, que foi a métrica usada no treinamento do modelo, e se encontra em azul, e na validação do modelo, e se encontra em verde.

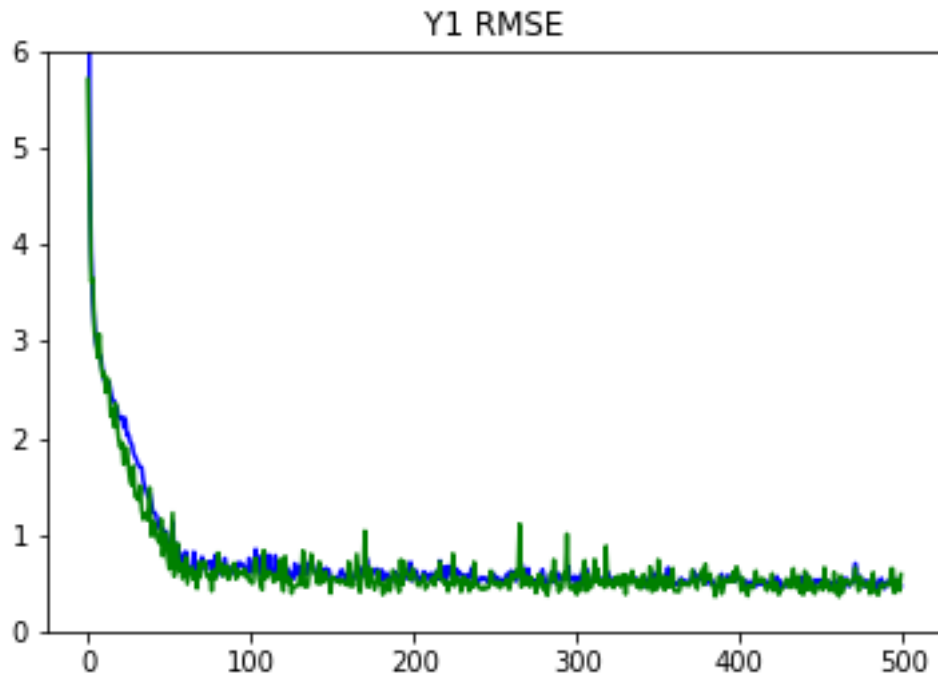


Figura 28 – RMSE para Y_1 no método MLP com seleção.

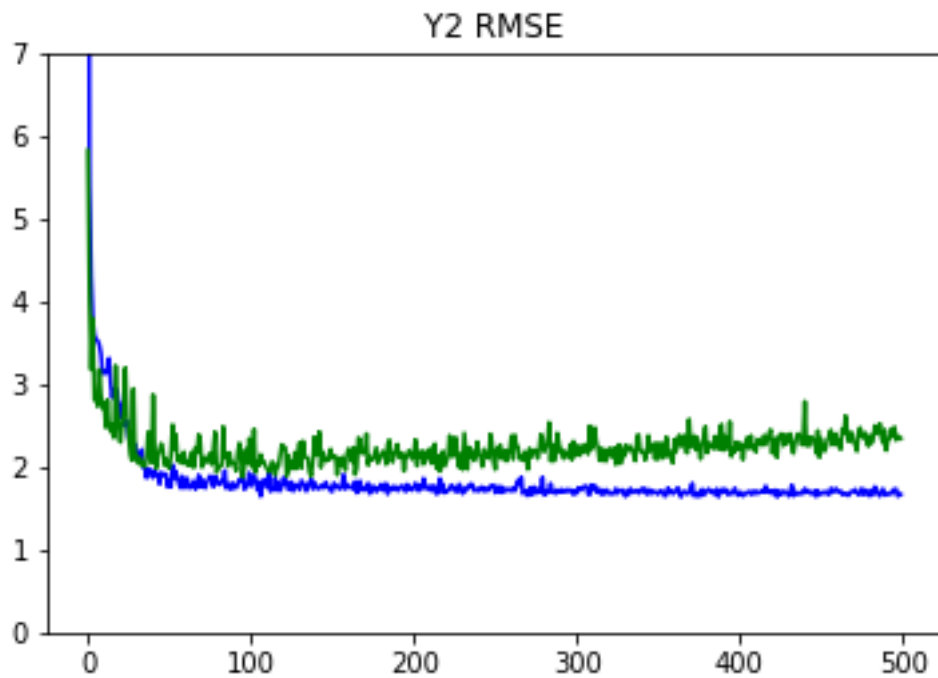


Figura 29 – RMSE para Y_2 no método MLP com seleção.

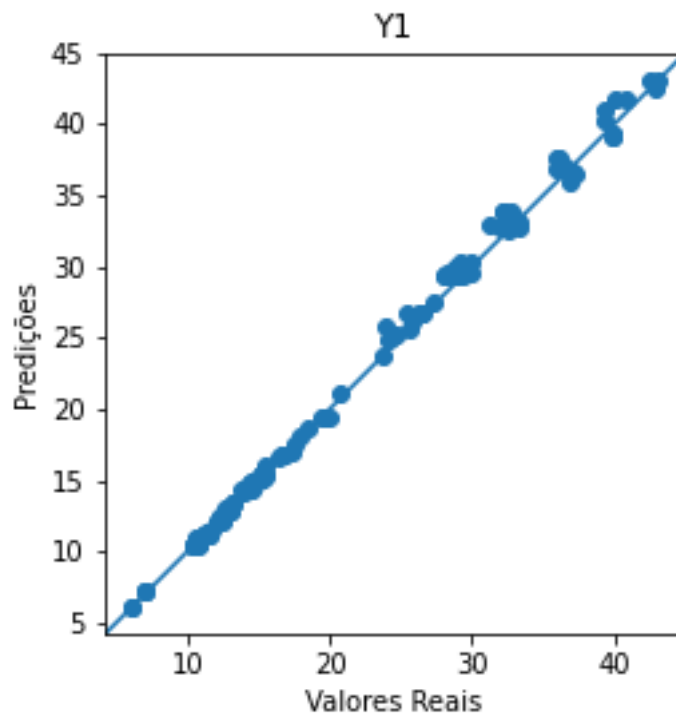


Figura 30 – Diferença entre valores preditos e reais para Y_1 no método MLP com seleção.

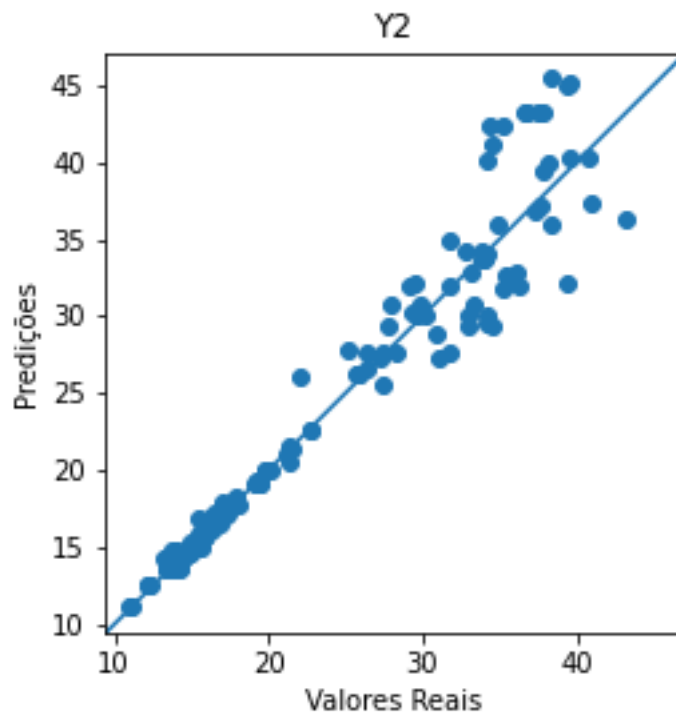


Figura 31 – Diferença entre valores preditos e reais para Y_2 no método MLP com seleção.

Conforme análise das Figuras 28, 30 e da Tabela 7 a eliminação da variável X_6

resultou em resultados satisfatórios para a saída Y_1 , visto que os valores de MAE e RMSE foram pequenos, o que pode ser observado pela maneira como a curva dos gráficos dos valores reais e preditos estão próximos. O mesmo não pode ser dito em relação à Y_2 , visto que os gráficos 31 e 29 mostram vários pontos fora da curva preditiva, além de o valor de RMSE encontrado ser acima do esperado, e muito acima do encontrado no modelo sem seleção.

A análise realizada indica que a retirada da variável X_6 do modelo não resultou em um bom ajuste e a variável eliminada contribui de forma significativa para a predição das saídas, principalmente de Y_2 , não podendo ser eliminada.

5 CONCLUSÃO

Após vários testes, foi encontrada a melhor configuração de parâmetros das redes MLP e CNN, sendo que o processo de ajuste do modelo foi realizado de maneira mais fácil com a rede MLP, contando também com um menor número de parâmetros. Nesse cenário, considerando a mesma base de dados, o valor de MAE obtido para Y_1 (carga de aquecimento) foi de 0.2845 para os testes realizados com o método MLP, apresentando menor valor do que o obtido por Duarte et al. (2017), que foi de 0.315, sendo o menor valor encontrado por [9] dentre todos os métodos testados. O valor de RMSE para Y_1 , no modelo MLP, apresentou valor 0.4707, levemente superior ao valor encontrado em [9] que foi de 0.420 como pode ser observado na Tabela 8.

Tabela 8 – Comparação de resultados obtidos.

Referência	Modelo	$Y_1 - MAE$	$Y_1 - RMSE$	$Y_2 - MAE$	$Y_2 - RMSE$
Duarte et al. (2017)	MLP	0.315	0.420	0.565	0.837
Duarte et al. (2017)	Random forests	0.315	0.223	0.941	2.118
Tsanas e Xifara (2012)	Random forests	0.510	-	1.420	-
Este estudo	MLP	0.2845	0.4707	0.3610	0.4950
Este estudo	CNN	0.2759	0.3784	0.4468	0.5867

Se tratando de Y_2 (carga de resfriamento), o modelo MLP alcançou os valores de 0.3610 e 0.4950 para MAE e RMSE respectivamente, e Duarte et al. (2017) encontrou 0.565 e 0.837 para os mesmos índices.

Não foi encontrado na literatura trabalhos que utilizaram o método CNN na base de dados utilizada, sendo essa a razão da escolha de sua utilização no presente trabalho. Os resultados obtidos através da análise desse método foram muito similares aos resultados encontrados na implementação da técnica MLP para a variável Y_1 e inferiores se tratando de Y_2 .

Apesar da rede CNN não ter alcançado um desempenho tão preciso como o modelo proposto com a rede MLP, ainda assim, é uma abordagem interessante, uma vez que apresentou valores de MAE e RMSE de 0.2759 e 0.3784 para Y_1 e de 0.4468 e 0.5867 para Y_2 . Estes valores de erros são inferiores aos encontrados pelo método usando a MLP proposta por Duarte et al. (2017), para ambas as saídas.

Dentre os modelos testados, a rede MLP forneceu a melhor acurácia na previsão das cargas de resfriamento e aquecimento dos edifícios, em comparação com o modelo CNN. Além disso, para ambas as arquiteturas de redes testadas (MLP e CNN), o melhor desempenho foi obtido empregando todas as oito variáveis de entrada, ou seja, sem a seleção de variáveis.

Mais testes com a rede CNN podem ser realizados em trabalho futuros, objetivando aumentar a acurácia da predição das cargas de aquecimento e, principalmente de resfriamento para a base de dados analisada, visto que por possuir muitos parâmetros, seu ajuste se torna complexo e são necessários uma quantidade maior de testes para obtenção de melhor desempenho.

REFERÊNCIAS

- [1] L. Pérez-Lombard, J. Ortiz, and C. Pout, “A review on buildings energy consumption information,” *Energy and buildings*, vol. 40, no. 3, pp. 394–398, 2008.
- [2] R. Yao, B. Li, and K. Steemers, “Energy policy and standard for built environment in china,” *Renewable Energy*, vol. 30, no. 13, pp. 1973–1988, 2005.
- [3] A. Yezioro, B. Dong, and F. Leite, “An applied artificial intelligence approach towards assessing building performance simulation tools,” *Energy and Buildings*, vol. 40, no. 4, pp. 612–620, 2008.
- [4] K. K. Wan, D. H. Li, D. Liu, and J. C. Lam, “Future trends of building heating and cooling loads and energy consumption in different climates,” *Building and Environment*, vol. 46, no. 1, pp. 223–234, 2011.
- [5] S. S. Kwok, R. K. Yuen, and E. W. Lee, “An intelligent approach to assessing the effect of building occupancy on building cooling load prediction,” *Building and Environment*, vol. 46, no. 8, pp. 1681–1690, 2011.
- [6] W. Pessenlehner and A. Mahdavi, *Building morphology, transparency, and energy performance*. na, 2003.
- [7] A. desconhecido. (2022, Outubro) Heating and cooling loads. [Online]. Available: <https://basix.nsw.gov.au/iframe/thermal-help/heating-and-cooling-loads.html#:~:text=The%20heating%20load%20is%20the,temperature%20in%20an%20acceptable%20range>.
- [8] J.-S. Chou and D.-K. Bui, “Modeling heating and cooling loads by artificial intelligence for energy-efficient building design,” *Energy and Buildings*, vol. 82, pp. 437–446, 2014.
- [9] G. R. Duarte, L. G. da Fonseca, P. Goliatt, and A. C. de Castro Lemonge, “Comparison of machine learning techniques for predicting energy loads in buildings,” *Ambiente Construído*, vol. 17, no. 3, pp. 103–115, 2017.
- [10] A. Tsanas and A. Xifara, “Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools,” *Energy and buildings*, vol. 49, pp. 560–567, 2012.
- [11] M. E. G. Martins. (2022, Outubro) Coeficiente de correlação amostral. [Online]. Available: https://www.fc.up.pt/pessoas/jfgomes/pdf/vol_2_num_2_69_art_coeficienteCorrelacaoAmostrai.pdf
- [12] A. Desconhecido. (2022, Setembro) Preprocessing data. [Online]. Available: <https://scikit-learn.org/stable/modules/preprocessing.html>

- [13] S. Haykin, *Redes neurais: princípios e prática*. Bookman Editora, 2001.
- [14] S. N. de Carvalho Leite, “Redes neurais artificiais e deep learning,” in *Ciência de dados*, 2020.
- [15] Álvaro Lima, W. Lacerda, and H. Neto, “Seleção de características de dados utilizando redes neurais artificiais,” in *Anais do XIII Simpósio Brasileiro de Sistemas de Informação*. Porto Alegre, RS, Brasil: SBC, 2017, pp. 135–142. [Online]. Available: <https://sol.sbc.org.br/index.php/sbsi/article/view/6035>
- [16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers
- [17] J. Brownlee. (2022, Outubro) Gradient descent optimization with nadam from scratch. [Online]. Available: <https://machinelearningmastery.com/gradient-descent-optimization-with-nadam-from-scratch/>
- [18] W. H. Delashmit, M. T. Manry *et al.*, “Recent developments in multilayer perceptron neural networks,” in *Proceedings of the seventh Annual Memphis Area Engineering and Science Conference, MAESC*, 2005.
- [19] R. E. V. d. Silva, “Um estudo comparativo entre redes neurais convolucionais para a classificação de imagens,” 2018.