

O uso de *Machine Learning* para a classificação da resistência à compressão do concreto

David Hapner Barzotto

Universidade Estadual do Oeste do Paraná (UNIOESTE)

Rua Universitária, 1619

dhbarzotto@hotmail.com

Simone Aparecida Miloca

Universidade Estadual do Oeste do Paraná (UNIOESTE)

Rua Universitária, 1619

smiloca@gmail.com

RESUMO

O concreto é o material mais usado no Brasil para edificações de pequeno e grande porte, sendo seu estudo de grande importância para o avanço tecnológico e sustentável da engenharia civil. Uma das principais métricas para a confiabilidade estrutural é a resistência à compressão do concreto. Este trabalho propõe a construção de modelos de *Machine Learning* (ML) envolvendo algoritmos de *Support Vector Machines* (SVM) e Redes Neurais Artificiais (RNAs) para uma análise classificatória da resistência à compressão do concreto com base na quantidade de materiais presentes em seu traço. Os resultados se mostraram promissores para este tipo de problema, alguns com acurácia em torno de 98%.

PALAVRAS CHAVE. *Support Vector Machines*. Redes Neurais Artificiais. Concreto.

Tópicos: Análise teórica do SVM. Análise teórica de RNAs. Estudo sobre a viabilidade da classificação de dados do concreto via *Machine Learning*.

ABSTRACT

Concrete is the most used material in Brazil for small and large buildings, and its study is of great importance for the technological and sustainable advancement of civil engineering. Thus, one of the main metrics for structural reliability is the compressive strength of concrete. This work proposes a machine learning (ML) models known as Support Vector Machines (SVM) and Artificial Neural Networks (ANN) for a classificatory of the compressive strength of concrete based on the number of materials present in its mix. The results were promising for this type of problem, some with accuracy around 98%.

KEYWORDS. Support Vector Machines. Artificial Neural Networks. Concrete.

Paper topics: Theoretical analysis of SVM. Theoretical Analysis of ANN. Study of viability of concrete data classification using Machine Learning.

1. Introdução

O concreto é o material mais utilizado no ambiente de construção civil brasileiro que se estende desde casas e edifícios residenciais, até pontes e usinas geradoras de energia [Tavares et al. 2019]. Dessa forma, a definição do traço do concreto, ou seja, a forma de determinar a proporção de cada material com base na quantidade de cimento, é alvo de vários estudos de centros de pesquisa em todo o mundo. Após a obtenção do traço, é necessário preparar uma mistura experimental e avaliar as suas características para verificar se há necessidade de ajustes e/ou correções. Segundo [Oliveira et al. 2007] os métodos utilizados, chamados de dosagem experimental, são estudos empíricos e não apresentam um modelo matemático por conta da heterogeneidade do concreto. Dessa forma, a classificação das características é de importância fundamental para compreender a resistência à compressão do concreto.

O *Support Vector Machines* (SVM) e as Redes Neurais Artificiais (RNAs), envolvem algoritmos que apresentam uma boa capacidade de generalização para problemas de classificação. Embora funcionem com princípios diferentes, sendo o SVM através da aprendizagem estatística e alguns modelos de RNAs pelo aprendizado de correção de erro, mostram grandes similaridades em seus resultados e aplicações [Haykin 2001].

Como apresentado por [Tavares et al. 2019], para concretos autoadensáveis existem estudos que implementam RNAs, tanto para misturas autoadensáveis tradicionais, quanto para outras que utilizaram cinzas volantes, cinzas de incineração de resíduos industriais e sílica ativa para a substituição parcial do concreto. Tais modelos obtiveram excelentes resultados com a implementação de RNAs para capturar interações não lineares entre os vários parâmetros existentes que são acrescentados [Tavares et al. 2019], [Behnood et al. 2018], [Yaman et al. 2017] e [Siddique et al. 2011].

Dessa forma, este trabalho utiliza SVM e Redes Neurais Artificiais do tipo *Multi-Layer Perceptron* (MLP) como uma forma alternativa de análise classificatória da resistência, portanto, sem a utilização de métodos empíricos. Serão propostos e testados modelos capazes de realizar a classificação, avaliados por diferentes métricas.

2. Objetivos e Contribuições

O principal objetivo do estudo é utilizar modelos de *Machine Learning* envolvendo o SVM e a MLP para o problema de classificação da resistência característica à compressão do concreto. Alguns estudos a respeito do uso de MLP na análise classificativa e preditiva da resistência do concreto foram feitos, como [Arora et al. 2019], [Tavares et al. 2019], [Felix et al. 2017], [Akpınar e Khashman 2017], [Oliveira et al. 2007], sendo influenciados por [Yeh 1998]. Dessa forma, os trabalhos supracitados foram utilizados como referência técnica para o desenvolvimento da RNA, a fim de produzir um modelo eficaz.

Por mais que o concreto seja o material mais usado e mais importante na construção civil brasileira, o seu modo de preparo e avaliação em canteiro de obras é pouco eficiente. Entretanto, com o surgimento do concreto usinado, os estudos e precisões com relação ao material evoluíram [Moretti 2010].

Todavia, os métodos de avaliação tratam de retirar uma grande quantidade de corpos de prova das obras e das usinas para avaliação das características que é de fundamental importância para um controle da obra eficiente conforme o projeto estrutural. Neste sentido, a contribuição deste trabalho reside na utilização de uma abordagem não tradicional para a análise da resistência do concreto, fazendo uso da inteligência computacional (IC), com a principal ideia de diminuir os custos e o tempo de análise da obra, trazendo maior eficiência, sustentabilidade e precisão à construção civil.

3. Teoria e processos de classificação

As técnicas que envolvem classificação estão inseridas em uma área denominada Aprendizado de Máquina (ML, do inglês *Machine Learning*) e buscam a construção de modelos que permitem alocar um novo objeto a uma determinada classe (grupo), dentre um conjunto finito de classes definidas a priori. Segundo [Bramer 2016], a classificação é uma forma de previsão,

onde o valor a ser previsto é um rótulo. É um pouco diferente da previsão numérica (por exemplo, chamada de regressão), cujo objetivo é prever um valor numérico, como os lucros de uma empresa ou o preço de uma ação. A literatura apresenta diferentes algoritmos para resolver problemas de classificação tais como o MLP e SVM [Awad e Khanna 2015].

Segundo [Awad e Khanna 2015], algoritmos de classificação funcionam bem para limites bem definidos, em que a entrada (*input*) segue pares específicos de atributos, na qual as saídas (*output*) são categóricas. Geralmente, a classificação é um processo supervisionado, ou seja, baseia-se nos dados de entrada lidos e dimensionados em dados de saída, relacionando-os com os dados esperados.

Um classificador envolvendo o estado da arte atual em *Machine Learning* é o SVM. Tal método é uma técnica complexa baseada na teoria do aprendizado estatístico [Steinwart e Christmann 2008]. Um outro muito utilizado também envolve a construção de modelos de redes neurais do tipo MLP, que, nesse caso, é fundamentada no algoritmo de *Back-propagation* [Haykin 2001], [Rumelhart et al. 1986].

Segundo [Aggarwal 2018] os dados devem ser utilizados de modo a atender à expectativa do propósito do modelo, de uma maneira que seja eficiente em seu processamento. O procedimento envolve a separação do conjunto de dados originais em outros dois conjuntos nomeados por conjuntos de treinamento e teste, como ilustrado na Figura 1, podendo ser utilizadas diferentes técnicas como *hold-out validation*, *cross validation*, e *bootstrapping*. Depois de definido o algoritmo de *Machine Learning* e o modelo de classificação gerado, é essencial fazer uma análise de desempenho. As principais métricas de avaliação são a acurácia, o erro e a matriz de confusão [Ertel 2017].

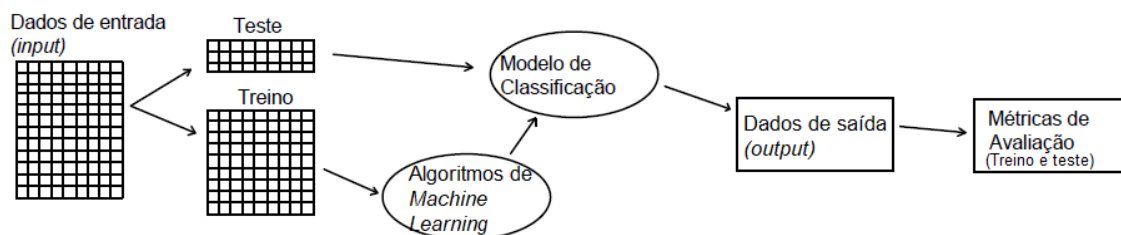


Figura 1: Processo de classificação [O Autor].

3.1. Support Vector Machines

Esta seção apresenta aspectos teóricos da técnica denominada *Support Vector Machines* (SVM) para problemas de classificação e sua construção teve por principais referências [Cortes e Vapnik 1995], [Haykin 2001], [Steinwart e Christmann 2008], [Awad e Khanna 2015] e [Deisenroth et al. 2020].

A técnica SVM consiste em separar os dados por meio da utilização de um hiperplano cuja equação é,

$$\mathbf{f}(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = 0 \quad (1)$$

em que $\mathbf{x}$ é o vetor de entrada, $\mathbf{w}$ é o vetor peso ajustável e b é o coeficiente linear também chamado de bias ou limiar linear. Tal equação divide o espaço dos dados, que por hipótese deve ser linearmente separável, em duas regiões de modo que,

$$\mathbf{w}^T \mathbf{x}_i + b \geq 1 \text{ se } y_i = +1$$

$$\mathbf{w}^T \mathbf{x}_i + b \leq -1 \text{ se } y_i = -1$$

em que, y_i é o resultado esperado.

Segundo [Deisenroth et al. 2020] e [Steinwart e Christmann 2008], pode-se escrever a formulação matemática para a geração do hiperplano através de um problema de otimização:

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

$$\text{sujeito a } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \forall i = 1, 2, \dots, N.$$

Esse problema de minimização implica na maximização da margem p_0 , em que esse é definido pela maior distância possível entre os pontos mais próximos do hiperplano separador ótimo, que é encontrado com base nos vetores suporte exemplificada na Figura 2. As restrições devem garantir que não haja dados entre as margens de separação de classes. Esse modelo de minimização é chamado de *Hard Margin SVM*.

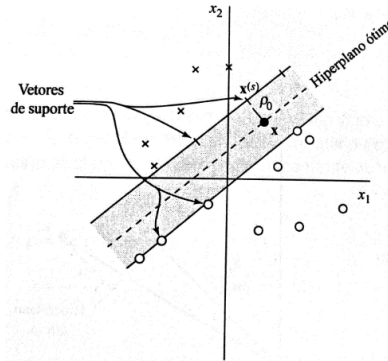


Figura 2: Hiperplano ótimo com margem maximizada, [Haykin 2001].

O SVM envolve, portanto, a resolução de um problema de otimização quadrático, cujos conceitos teóricos estão bem estabelecidos [Luenberger e Ye 2016]. Assim, dadas as características do problema que envolve uma função objetivo quadrática e conjuntos de restrições convexo, pode-se olhar sob duas formulações, o SVM Primal e o SVM Dual [Deisenroth et al. 2020]. Dependendo da natureza do problema, pode ser que os dados de entrada sejam não linearmente separáveis. Para estes casos pode-se empregar o modelo Primal e Dual para padrões não lineares, visto que tais modelos permitem dados entre as margens de separação das classes [Haykin 2001] e [Awad e Khanna 2015].

Assim, dado um conjunto de dados de entrada não linearmente separáveis, pode-se construir um modelo em que ao encontrar o hiperplano, o erro de classificação seja minimizado, calculado como a média do conjunto de treinamento [Haykin 2001]. Para isso, são introduzidas variáveis de folga ξ_i , com objetivo de relaxar as restrições impostas ao problema de otimização primal, permitindo que haja erros de classificação, além de um parâmetro C que controla a variável de folga [Awad e Khanna 2015].

Assim, a formulação do problema de otimização para encontrar o hiperplano ótimo que condiciona o SVM Primal para padrões não lineares é:

$$\begin{aligned} \min_{\mathbf{w}, \xi} \Phi(\mathbf{w}, \xi) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i \\ \text{sujeito a } y_i(\mathbf{w}^T \mathbf{x}_i + b) &\geq 1 - \xi_i, \quad \forall i = 1, 2, \dots, N \\ e \xi_i &\geq 0, \forall i = 1, 2, \dots, N \end{aligned}$$

onde C é um parâmetro de regularização que pode variar conforme a meta de otimização, cujo é um valor definido pelo usuário. À medida que C aumenta, uma margem mais estreita é obtida e mais ênfase é colocada para minimizar os erros de classificação. À medida que C diminui, mais violações são permitidas, porque maximizar a margem entre as duas classes torna-se o objetivo do SVM [Awad e Khanna 2015].

O problema primal lida com uma função de custo convexa e com restrições lineares. Dado um problema de otimização restrito como este, é possível construir seu Dual, que em termos de Lagrange, pode ser formalizado como:

$$\max_{\lambda} Q(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

sujeito a $0 \leq \lambda_i \leq C \quad \forall i = 1, 2, \dots, N$

$$e \sum_{i=1}^N \lambda_i y_i = 0$$

em que, o λ_i é o multiplicador de Lagrange associado, y_i e y_j são os resultados esperados e $\mathbf{x}_i$ e $\mathbf{x}_j$ são os vetores dos dados de entrada.

Segundo [Awad e Khanna 2015] e [Haykin 2001], quando os dados de entrada não são linearmente separáveis, o SVM pode usar a teoria do kernel para mapear tais dados para um novo espaço de dimensão superior, chamado de espaço de características. O uso da função do kernel, mapeia os dados em um novo espaço, de dimensão superior, em que um separador linear seria capaz de discriminar entre as diferentes classes.

O kernel é definido como uma transformação dos vetores de entrada que estão contidos no espaço de características (espaço imagem) da transformação não linear. Como descrito por [Awad e Khanna 2015], a formulação do SVM Dual permite uma maior facilidade de aplicações de kernels, assim, podemos formalizar a otimização quadrática com o uso do kernel como:

$$\max_{\lambda} Q(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

sujeito a $0 \leq \lambda_i \leq C \quad \forall i = 1, 2, \dots, N$

$$e \sum_{i=1}^N \lambda_i y_i = 0$$

em que, $K(\mathbf{x}_i, \mathbf{x}_j)$ é a função do kernel, sendo apresentado na Equação 2 o kernel Radial Basis Function (RBF).

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2 \right) \quad (2)$$

onde, γ é o valor da largura comum a todos os núcleos, sendo tal parâmetro definido pelo usuário [Haykin 2001]. Para o caso de utilização do kernel linear, que é definido pelo produto interno dos dados de entrada, tem-se a formulação do SVM Dual.

3.2. Redes Neurais Multi-Layer Perceptron

Formalmente descrito em [Aggarwal 2018], as RNAs surgiram logo após o advento dos computadores nas décadas de 1950 e 1960. Foram desenvolvidas para simular o sistema nervoso humano para tarefas de aprendizado de máquina, tratando as unidades computacionais em um modelo de aprendizado de maneira semelhante aos neurônios humanos.

As redes neurais artificiais foram pensadas e construídas usando como base as redes neurais biológicas (neurônios). Com isso, os principais filamentos usados como modelos são, o dendrito (*dendrite*), axônio (*axon*) e corpo celular (*cell body*) ou soma, como exemplificado na Figura 3.a. Com tais modelos, foram projetadas e modeladas as primeiras redes neurais artificiais que trabalham apenas com um impulso por vez, da mesma maneira que o neurônio biológico. Através de métodos e avanços científicos, o campo de estudo e suas aplicações aumentaram gradativamente com o passar dos anos.

Neste trabalho, a RNA de interesse é o *Multi-Layer Perceptron* (MLP), por possuir uma grande capacidade classificativa de dados [Fausett 1994], [Haykin 2001], [Aggarwal 2018]. Tem sido aplicada com sucesso para resolver diversos problemas difíceis, através do seu treinamento de forma supervisionada com um algoritmo muito popular conhecido como *algoritmo de retropropagação de erro* (*error back-propagation*). Este algoritmo é baseado na regra de aprendizagem por correção de erro [Haykin 2001]. Possui três características distintas:

1. O modelo de cada neurônio da rede inclui uma função de ativação geralmente não linear, sendo as principais a função de ativação sigmóide e ReLU.
2. A rede contém uma ou mais camadas de neurônios ocultos, que não são parte da entrada ou da saída da rede. Estes neurônios ocultos capacitam a rede a aprender tarefas complexas extraindo progressivamente as características mais significativas dos padrões (vetores) de entrada.
3. A rede exibe alto grau de conectividade, determinado pelas sinapses. Uma modificação na conectividade requer uma mudança na população das conexões sinápticas ou de seus pesos.

É a partir dessas três características, junto com a habilidade de aprender através do treinamento, que a MLP deriva seu poder computacional [Haykin 2001]. Um modelo de MLP cuja arquitetura envolve uma camada de entrada, uma camada oculta e uma camada de saída é exemplificada na Figura 3.b. A Figura 3.c mostra a formulação para cada neurônio de uma RNA, que utiliza uma função de ativação para processar os sinais de entrada e fornecer uma saída.

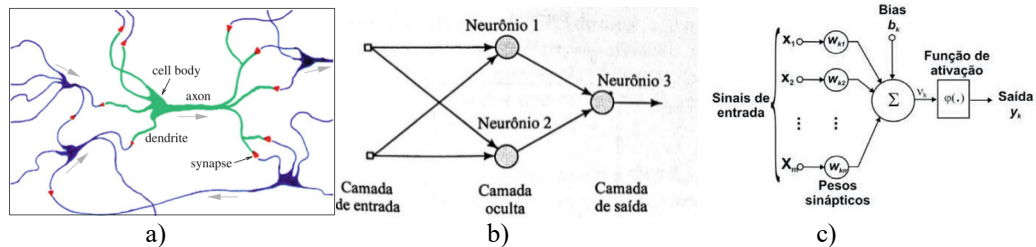


Figura 3: a) Modelo de Rede Neural Biológica. b) Topologia de uma RNA multicamadas. c) Formulação de um neurônio de uma RNA, adaptado de [Ertel 2017] e [Haykin 2001].

Assim, a formulação matemática por trás de cada neurônio da MLP pode ser escrita como

$$y_k = \varphi(x) = \varphi\left(\sum_{j=1}^m W_{kj}X_j + b_k\right) \quad (3)$$

onde, y_k é a resposta obtida pela função de ativação $\varphi(x)$, W_{kj} é o peso sináptico da rede, X_j é o dado de entrada e b_k é o bias. A função de ativação (4) é chamada ReLU, a Função (5) Sigmóide e a Função (6) é a Linear.

$$ReLU(x) = \max\{0, x\} \quad (4); \quad h(x) = \frac{1 - \exp(-\sigma x)}{1 + \exp(-\sigma x)} \quad (5); \quad \varphi(x) = x \quad (6).$$

Para treinar a MLP, utiliza-se o gradiente descendente estocástico (GDE), associado a uma taxa de aprendizado, em que tal taxa é o *passo* dado pelo GDE para convergir a um peso sináptico mínimo correspondente [Kovács 2006]. Em tal modelo, atualizam-se os parâmetros calculando uma aproximação estocástica da função de perda do gradiente total, podendo ser baseado tanto no grupo dos dados de treinamento, quanto em um subgrupo dos dados de treinamento conhecido como mini batch [Qian e Klabjan 2020]. O Algoritmo 1 apresenta o funcionamento da MLP com base no *back-propagation* para separar duas classes hipotéticas, C_1 e C_2 .

Entrada: Inicializar o vetor de pesos W_{kj} com valores aleatórios;

Repetir

Para (X_j, d_k) **faça**

1. Aplicar o vetor de entrada X_j à camada de entrada e indique o valor da taxa de aprendizado (η) do GDE;
2. Propagação para frente:

Para todas as camadas, desde a primeira camada oculta

Para cada neurônio, calcule a ativação:

$$y_k = \varphi(\sum_{j=1}^m W_{kj}X_j + b_k);$$

3. Calcule o erro $E(W)$ pelo *Mean Squared Error (MSE)*, entre o resultado esperado e o calculado;

$$E(W) = \frac{1}{m} \left(\sum_{k=1}^m d_k - y_k \right)^2$$

4. Propagação para trás (*Back-propagation*):

Para todos os níveis de pesos, do primeiro ao último

Para cada peso W_{kj} :

$$W_{kj}^{Novo} = W_{kj}^{Velho} - \eta \frac{\partial}{\partial w} (E(w))$$

5. Separar o problema em duas classes C_1 e C_2 :

Se $\varphi(\sum_{j=1}^m W_{kj}^{Novo} X_j + b_k) > 0,5$; então pertence a C_1 ,

Se $\varphi(\sum_{j=1}^m W_{kj}^{Novo} X_j + b_k) \leq 0,5$; então pertence a C_2 .

Fim

Até: W_{kj} convergir para o ponto local ou global ótimo, ou a variação do erro estipulado ser atingida, ou o algoritmo atingir o número máximo de iterações estabelecido pelo usuário.

Algoritmo 1: Algoritmo de funcionamento do *Multi-Layer Perceptron* [O Autor].

As principais vantagens do uso de redes neurais sobre outras técnicas incluem a adaptabilidade que consiste na habilidade de se ajustar a novas informações, velocidade através de paralelismo massivo e tolerância a falhas pois possui capacidade de oferecer boas respostas mesmo com falta, confusão ou dados ruidosos.

4. Concreto

Como apresenta [Moretti 2010], o concreto é o material mais utilizado na construção civil, sendo assim, há mais de meio século estudado e aperfeiçoado. É confeccionado utilizando diversas matérias primas regionais, como o cimento, a brita e a areia. Desse modo, quantificar adequadamente esses materiais é o objetivo da dosagem do concreto, de tal modo a se obter um concreto que atenda às necessidades estruturais exigidas, sendo a principal delas a resistência característica à compressão. A Figura 4 apresenta o ensaio de resistência à compressão do concreto. Note que à esquerda o corpo de concreto está inteiro, já a direita ele está rompido. Com isso, a resistência à compressão do concreto é o quanto o corpo de prova aguenta de compressão axial até romper.

Assim, para o projetista estrutural, a resistência característica à compressão do concreto endurecido é o principal parâmetro a ser considerado [Helene e Terzian 1993]. De forma sucinta, o concreto é um material heterogêneo constituído por uma vasta gama de partículas granulares, variando de dimensões menores que um micron (sílica ativa) até a centímetros (agregados graúdos/britas) [Moretti 2010]. Dessa maneira, todos os componentes presentes no concreto influenciam a sua resistência à compressão, sendo neste estudo, avaliados a quantidade de cimento, de escória de alto forno, de cinza volante, de água, de superplastificante, de agregados graúdos e de agregados finos [Yeh 1998].



Figura 4: Ensaio de resistência à compressão do concreto [O Autor].

5. Materiais e Métodos

A metodologia de pesquisa consistiu em estudos teóricos das técnicas utilizadas, descritas nas Seções 3 e 4 e também na análise de resultados obtidos via modelagem e testes computacionais.

A base de dados utilizada foi “Concrete Compressive Strength Data Set” [Yeh 1998] extraída do UCI Machine Learning Repository. É multivariada com características reais, tendo um total de oito variáveis (*input*) (componentes do traço do concreto e idade de cura) utilizados para classificar a resistência característica à compressão. Para a definição de saída (*output*), tomou-se por base informações da [ABNT NBR 8953 2015], em que, tal documento, distingue o concreto em duas classes devido a resistência à compressão (em Mega Pascal (MPa)), sendo a classe I de 20 MPa a 50 MPa e a classe II de 55 MPa a 100 MPa. Neste trabalho foi considerado pertencente a classe I, o elemento que apresentar 1 MPa a 50 MPa, de modo a permitir a utilização dos 1030 resultados presentes na base de dados, além de ser possível a compreensão de concretos de baixa resistência. As informações são detalhadas na Tabela 1.

Para os conjuntos de treinamento e teste, optou-se por separar em 70% para treino e 30% para teste, sendo realizado 10 ensaios com a técnica *hold-out* em cada um dos 8 experimentos.

Tabela 1: Atributos da base de dados. Valores máximos e mínimos, adaptado de [Akpınar e Khashman 2017].

Entrada	Parâmetro	Máximo	Mínimo
X1	Água (kg/m ³)	247	121,8
X2	Cimento (kg/m ³)	540	102
X3	Cinza Volante (kg/m ³)	200,1	0
X4	Escória (kg/m ³)	359,4	0
X5	Agregados Finos (kg/m ³)	992,6	594
X6	Agregados Graúdos (kg/m ³)	1145	801
X7	Superplastificante (kg/m ³)	32,2	0
X8	Idade (dias)	365	1
Saída	Parâmetros	Máximo	Mínimo
Y1	Resistência à compressão (MPa)	82,6	2,3

O pré-processamento dos dados foi efetuado e consistiu na normalização e aleatorização para que cada ensaio feito seja diferente do anterior, proporcionando ambientes adversos aos algoritmos, esperando que, consigam, assim, um melhor aprendizado.

Com relação as implementações dos códigos, utilizou-se a linguagem de programação Python com os pacotes *sklearn* e *keras*, para o SVM e MLP, respectivamente. Para o SVM foram utilizadas duas bibliotecas o *LibSVM* e o *LibLINEAR*. Ambos utilizam o SVM, todavia, o *LibLINEAR* é baseado no modelo do SVM Primal para padrões não lineares descrito na Seção 3.1. Já o *LibSVM* possui a capacidade de utilização de diferentes kernels em seu modelo.

Foram definidos quatro experimentos com relação ao SVM, sendo dois deles utilizando modelos lineares como o *LibLINEAR* e o *LibSVM* com kernel linear, e dois utilizando o kernel Radial Basis Function (RBF) (Equação 2, Seção 3.1). Para o SVM, conforme a implementação do *sklearn*, tanto o parâmetro γ , quanto o C , devem ser positivos, entretanto, caso não seja definido um valor para γ , o *sklearn* calcula-o como $1/(a \text{ quantidade dos dados de entrada vezes a variância dos dados de entrada})$, e, caso não seja estipulado o parâmetro C , o *sklearn* adota-o como 1. Para os três primeiros experimentos, fixou-se os parâmetros para todos os ensaios feitos. Já para o último experimento com o kernel RBF, foi utilizado o pacote *GridSearchCV* para obter os melhores parâmetros à base de dados utilizada no treinamento.

Com relação a MLP, foram considerados também quatro experimentos. Para a definição de tais experimentos foram realizados alguns pré-testes para auxiliar na definição dos parâmetros do modelo. Os melhores resultados indicaram o número máximo 1000 iterações e a escolha da rede cuja arquitetura envolve 8 neurônios na camada de entrada, 16 neurônios na camada oculta e 1

neurônio na camada de saída, como na Figura 5. Para dois desses experimentos, utilizou-se nos 24 primeiros neurônios, a função de ativação ReLU (Equação 4, Seção 3.2) e no neurônio de saída a função de ativação sigmóide (Equação 5, Seção 3.2), ou seja, formulou-se um modelo misto. Com relação aos outros dois experimentos, considerou-se a mesma topologia (arquitetura) da RNA, porém, foram utilizadas funções de ativações lineares (Equação 6, Seção 3.2). Foram utilizadas taxas de aprendizado diferentes, sendo dois experimentos com 0,01 e os outros com 0,05, e o tamanho do mini batch de 10, ambos os parâmetros foram os melhores escolhidos.

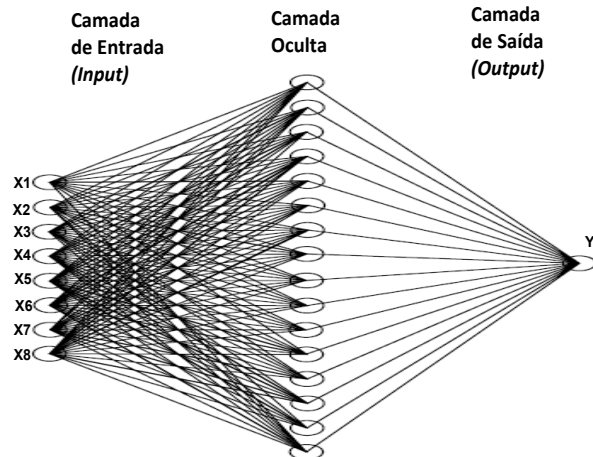


Figura 5: Topologia da MLP implementada [O Autor].

Deve-se ressaltar que, foi utilizado para avaliar os erros de treinamento o *Mean Squared Error* (MSE), pois além de ter sido usado pela maior parte dos trabalhos na área [Arora et al. 2019], [Tavares et al. 2019], [Akpınar e Khashman 2017], apresenta mais clareza ao expor os resultados.

Por fim, o computador utilizado tem as configurações de AMD Ryzen 4800H with Radeon Graphics 2.90 GHz, 16 GB RAM com sistema operacional de 64 bits e processador baseado em x64.

6. Resultados e Discussão

A Tabela 2 apresenta os resultados obtidos via SVM. Percebe-se que para o LibLINEAR1 e para o LibSVM2, o que foi utilizado a biblioteca LibLINEAR, foi o melhor pelo fato de apresentar menor número de iterações, maior acurácia de treino, menor MSE de treino e melhor tempo de treinamento. Com relação aos experimentos usando o kernel RBF, o LibSVM4 apresentou melhores resultados no geral, sendo pior apenas no número de iterações e no tempo. Destaca-se que, para todos os modelos, a acurácia ficou acima de 80% com intervalo de confiança de 0,95. Dessa forma, pode-se interpretar que para o SVM, o melhor resultado obtido foi o LibSVM4, pelo fato de obter as melhores métricas de acurácia e de erro. Deve-se ressaltar que, os quatro modelos utilizaram em média 2,0 GB de memória RAM.

Tabela 2: Resultados obtidos pelo SVM [O Autor].

Modelo	Parâmetro C	Parâmetro γ	Kernel	Iterações	Acurácia Treino	Acurácia Teste	MSE Treino	MSE Teste	Tempo (s)
LibLINEAR1	1	-	-	1188±85	85,35%±0,266	83,91%±0,276	0,143±0,007	0,161±0,016	0,040±0,004
LibSVM2	1	-	Linear	2374±459	84,35%±0,273	84,15%±0,275	0,155±0,005	0,159±0,011	0,043±0,027
LibSVM3	1	scale	RBF	336±30	92,50%±0,198	88,25%±0,242	0,139±0,003	0,117±0,007	0,048±0,024
LibSVM4	100	scale	RBF	2254±320	98,21%±0,099	92,28%±0,201	0,017±0,004	0,077±0,014	0,086±0,107

Nota: Para acurácia, MSE, iteração e tempo, o intervalo de confiança foi de 0,95.

A Tabela 3 mostra os resultados obtidos pelos experimentos da MLP. É possível observar que, para os modelos com função de ativação lineares, a MLP Linear1 apresentou ligeira melhoria nos resultados em todas as métricas avaliadas. Já para os modelos de MLP mista, a MLP Mista4, que utilizou taxa de aprendizado 0,05, obteve-se melhores resultados de acurácia de treino e de teste, de MSE de treino e de tempo. Mesmo obtendo piores resultados em MSE de teste, os demais

resultados são bem superiores o que demonstram que a MLP Mista4 foi o melhor experimento das redes neurais. Note-se que, em tais modelos, a acurácia ficou acima de 80% considerando um intervalo de confiança de 0,95. Além do mais, mesmo para experimentos diferentes feitos para a MLP o custo de memória foi em média 2,0 GB RAM.

Tabela 3: Resultados obtidos pela MLP [O Autor].

Modelo	Taxa de Aprendizado	Mini Batch	Função de ativação	Iterações	Acurácia Treino	Acurácia Teste	MSE Treino	MSE Teste	Tempo (s)
MLP Linear1	0,01	10	Linear	1000	85,48%±0,265	84,75%±0,270	0,102±0,001	0,103±0,003	228±22
MLP Linear2	0,05	10	Linear	1000	85,19%±0,267	84,29%±0,274	0,105±0,005	0,105±0,006	197±03
MLP Mista3	0,01	10	ReLU - Sigmóide	1000	97,09%±0,126	91,42%±0,210	0,027±0,002	0,065±0,006	208±14
MLP Mista4	0,05	10	ReLU - Sigmóide	1000	98,11%±0,102	92,07%±0,203	0,017±0,005	0,071±0,014	206±15

Nota: Para acurácia, MSE e tempo, o intervalo de confiança foi de 0,95.

Ao avaliar os melhores experimentos de ambos os modelos, SVM e MLP, observa-se que a MLP Mista4 apresentou melhores resultados de MSE de teste, com um número menor de iterações. Já LibSVM4, apresentou ligeira melhoria na acurácia de treino e teste, e foi consideravelmente superior no tempo de treino. Dessa maneira, pode-se perceber que ambos os resultados foram muito similares nas acurácias e nos MSEs, mostrando a boa capacidade de classificação para ambos os métodos. A Figura 6 apresenta o histórico obtido pela MLP Mista4, em que o gráfico mostra a relação entre as acurácias de treino e de teste, juntamente com os erros de treino e de teste.

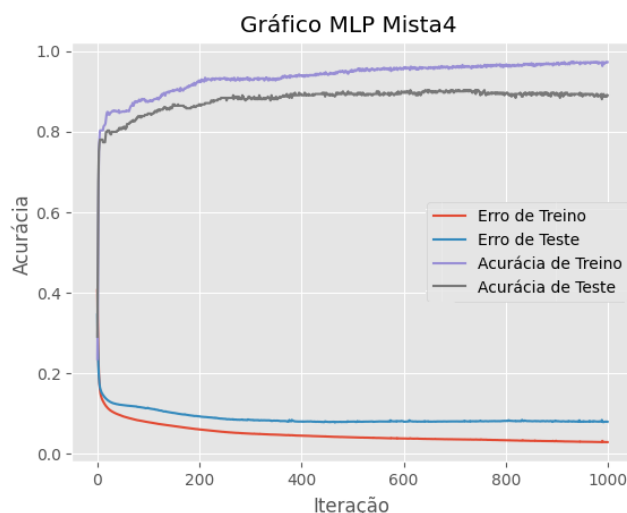


Figura 6: Acurácia e Erro obtidos pelo melhor modelo de MLP [O Autor].

Ao comparar os resultados de classificação obtidos com os de [Akpinar e Khashman 2017], o modelo de RNA desse trabalho obteve resultados significativamente melhores de ambas as acurácias e de MSE, utilizando um número significativamente menor de iterações. Deve-se ressaltar que o estudo proposto por [Akpinar e Khashman 2017] classifica três classes de concreto e os dados foram separados em 50% treino e 50% teste, o que neste estudo considerou-se duas classes e separados em 70% treino e 30% teste. Nesse quesito, o modelo apresentado neste trabalho apresenta ótimos resultados, mostrando a capacidade ótima de avaliar materiais heterogêneos como o concreto.

Conclusão

Os modelos apresentados nesse trabalho demonstram a evolução existente entre os trabalhos da área feitos anteriormente, assim aplicando uma nova proposta de pesquisa. Além disso, mostra resultados promissores para o uso de SVMs, RNAs e MLPs na área de engenharia no sentido de melhorar a eficiência dos projetos e obras. Dessa forma, os resultados angariam

melhores análises de futuros estudos e pesquisas na área da tecnologia dos materiais e da engenharia civil, fazendo uso de técnicas presentes na Inteligência Artificial.

Agradecimentos

Este estudo foi financiado pela Fundação Araucária de Apoio ao Desenvolvimento Científico e Tecnológico do Estado do Paraná.

Referências

ABNT NBR 8953, (2015). Concreto para fins estruturais – Classificação pela massa específica, por grupos de resistência e consistência. Rio de Janeiro, 7p.

Aggarwal, C. C. (2018). Neural Networks and Deep Learning. Springer, New York.

Arora, S., Balraj, S. e Bhardwaj, B. (2019). Strength performance of recycled aggregate concretes containing mineral admixtures and their performance prediction through various modeling techniques. Journal of Building Engineering 24 100741.

Akpınar, P. e Khashman A. (2017). Intelligent classification system for concrete compressive strength. 9th International Conference on Theory and Application of Soft Computing, Computing with Words and Perception, ICSCCW 2017, 24-25 August 2017, Budapest, Hungary.

Awad, M. e Kanna, R. (2015). Efficient Learning Machines. Apress Open, London.

Behnood, A. e Golafshani E. M. (2018). Predicting the compressive strength of sílica fume concrete using hybrid artificial neural network with multi-objective grey wolves. Journal of Cleaner Production, v. 202, p. 54–64.

Bramer, M. (2016). Principles of Data Mining. Springer, London.

Cortes, C. e Vapnik, V. (1995). Support-Vector Networks. Boston, Kluwer Academic Publishers.

Deisenroth, M. P., Faisal, A. A. e Ong., C. S. (2020). Mathematics for machine learning. Cambridge University Press, Cambridge.

Ertel, W. (2017). Introduction to Artificial Intelligence. Springer, London.

Fausett, L. (1994). Fundamentals of Neural Network. Prentice Hall, Hoboken.

Félix, E. F., Carrazedo, R., Possan E. (2017). Análise paramétrica da carbonatação em estruturas de concreto armado via Redes Neurais Artificiais. Revista ALCONPAT, 7(3), pp. 302-316.

Haykin, S. (2001). Redes Neurais: Princípios e Prática. Bookman, São Paulo.

He, Z., Zhang, M., e Zhang, H. (2016). Data-driven research on chemical features of Jingdezhen and Longquan celadon by energy dispersive X-ray fluorescence. Ceramics International, 42(4), 5123-5129.

Helene, P. e Terzian, P. (1993). Manual de dosagem e controle do concreto. PINI, 349p, São Paulo.

Kovács, Z. L. (2006). Redes Neurais Artificiais: Fundamentos e Aplicações, 4^a Edição Revista. Editora Livraria da Física, São Paulo.

Moretti, J. F. (2010), Sistema Inteligente Baseado na Redes Neurais Artificiais para Dosagem do Concreto. Tese de Doutorado (Programa de Pós-Graduação em Engenharia Elétrica), Universidade Estadual Paulista “Júlio de Mesquita Filho”, Ilha Solteira.

Luenberger, D. G. e Ye, Y. (2016). Linear and Nonlinear Programming, fourth edition. Springer, Switzerland.

Oliveira, H., Biondi Neto, L., Tavares, M. (2007). Redes Neurais Artificiais na obtenção do traço e na definição da resistência à compressão de concreto de alta resistência. SPOLM 2007.

Qian, X., Klabjan, D. (2020). The Impact of the Mini-batch Size on the Variance of Gradients in Stochastic Gradient Descent. arXiv:2004.13146v1.

Rummelhart, D. E., Hinton, G. E. e Williams R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533-536.

Siddique, R., Aggarwal, P. e Aggarwal, Y. (2011). Prediction of compressive strength of self-compacting concrete containing bottom ash using artificial neural networks. *Advances in Engineering Software*, v. 42, n. 10, p. 780–786.

Sklearn (2023). Web page. <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>. Acessado: 2023-26-05

Steinwart, I. e Christmann, A. (2008). Support Vector Machines. Springer, New York.

Tavares, D., Yanagi Junior, T., Lacerda, W. e Ribeiro, D. (2019). Uso de Redes Neurais Artificiais na Predição da Resistência à Compressão do Concreto. XXVIII Congresso de Pós-Graduação UFLA (2019).

Yaman, M. A., Elaty, M. A., Taman, M. (2017). Predicting the ingredients of self compacting concrete using artificial neural network. *Alexandria Engineering Journal*, v. 56, n. 4, p. 523–532.

Yeh, I.-C. (1998). Modeling of Strength of High-Performance Concrete Using Artificial Neural Networks. *Cement and Concrete Research*, Vol. 28. No.12, pp. 1797-1808.