

DETECÇÃO DE DANOS ESTRUTURAIS UTILIZANDO ANÁLISE ESTATÍSTICA E AUTO-CODIFICADOR ESPARSO

STRUCTURAL DAMAGE DETECTION USING STATISTICAL ANALYSIS AND SPARSE AUTOENCODERS

Rafaelle P. Finotti^a, Aldemon L. Bonifácio^a, Flávio S. Barbosa^{a,b}, Alexandre A. Cury^b e
Leonardo G. Fonseca^{a,b}

^a*Programa de Pós-Graduação em Modelagem Computacional, Universidade Federal de Juiz de Fora, Campus Universitário, Juiz de Fora-MG, 36036-330, Brasil, rafaelle.finotti@engenharia.ufjf.br, aldemon.bonifacio@engenharia.ufjf.br, flavio.barbosa@ufjf.edu.br, <http://www.ufjf.br/pgmc>*

^b*Departamento de Mecânica Aplicada e Computacional, Faculdade de Engenharia, Universidade Federal de Juiz de Fora, Campus Universitário, Juiz de Fora-MG, 36036-330, Brasil, alexandre.cury@ufjf.edu.br, leonardo.golliat@engenharia.ufjf.br, <http://www.ufjf.br/mac/>*

Palavras-chave: Dinâmica Estrutural, Detecção de Danos, Inteligência Computacional.

Resumo. Recentemente, novos métodos computacionais denominados de aprendizado de máquinas profundo, do inglês “*deep learning*”, vêm sendo propostos e aplicados aos mais diversos tipos de problemas de reconhecimento de padrões. Esses novos métodos buscam modelar de forma robusta abstrações complexas dos dados através do mapeamento linear e não-linear utilizando várias camadas de processamento. Um dos algoritmos deep learning mais conhecidos é o Auto-Codificador Esparso (em inglês, “*Sparse Autoencoder*” – SAE), que realiza o aprendizado das características dos dados de forma não-supervisionada. Diante do exposto, objetiva-se no presente trabalho avaliar o uso do SAE para o problema de detecção de danos estruturais em uma simulação numérica de viga, desenvolvendo-se modelos de classificação baseados em indicadores estatísticos extraídos das respostas dinâmicas no domínio do tempo. Os bons resultados incentivam o uso do Auto-Codificador Esparso e o desenvolvimento de modelos computacionais de aprendizado profundo na avaliação da integridade das estruturas.

Keywords: Structural Dynamics, Damage Identification, Computational Intelligence.

Abstract. More recently, new computational methods known as Deep Learning Machines have been proposed and applied to several types of pattern recognition problems. These new methods try to create robust models by complex data abstractions through linear and non-linear mapping using multiple processing layers. One of the best-known deep learning algorithms is the Sparse Autoencoder (SAE), which attempts to learn the characteristics of the data in an unsupervised way. Thus, the objective of the present study is to evaluate the application of SAE for structural damage detection problem. To this end, a numerical simulation of a beam is employed to develop classification models based on statistical indicators extracted from the dynamic responses in the time domain. The good results encourage the use of the Sparse Autoencoder and the development of computational models employing deep learning techniques for structural integrity assessment.

1 INTRODUÇÃO

O monitoramento contínuo estrutural permite a extração, processamento e acompanhamento de características que possibilitam avaliar a integridade da estrutura e traçar planos de manutenção e reparo, auxiliando na prevenção de danos e acidentes e, conseqüentemente, aumentando a sua vida útil. Desta forma, o diagnóstico eficaz de danos em estruturas tem sido tema de estudo nos mais diversos segmentos da engenharia, tais como: construção civil, indústria aeroespacial, indústria metalúrgica/mecânica, usinas de energia, entre outros (Ge et al., 2010; Wickramasinghe et al., 2016; Sakaris et al., 2017). É fato conhecido que o processo de degradação altera as propriedades físicas da estrutura, tais como massa e rigidez. Essas alterações influenciam o comportamento dinâmico estrutural e podem ser refletidas diretamente nos dados vibracionais. Em vista disso, grande parte dos métodos de detecção é desenvolvida com base em informações extraídas das respostas de vibração (Doebeling et al., 1998).

Dentre as várias abordagens possíveis, destacam-se aqui as que trabalham com as respostas dinâmicas no domínio do tempo. Normalmente, nesses tipos de abordagem os sinais são processados através de técnicas de tratamento de dados e, posteriormente, são comparados e classificados por algoritmos de inteligência computacional, como por exemplo Redes Neurais, Máquinas de Vetores Suporte, Árvores de Decisão, Agrupamento, etc (Cury et al., 2011; Alves et al., 2015; Amezcua-Sanchez e Adeli, 2016; Cardoso et al., 2017). De uma forma geral, as estratégias de detecção supracitadas apresentam bom desempenho, salvo em raras exceções. No entanto, devido à necessidade de manipulação de grande volume de dados, muitas vezes com recursos computacionais limitados, o desenvolvimento de métodos mais robustos para a solução de problemas de identificação de danos ainda é um desafio. Além disso, eventualmente, tem-se a dificuldade em lidar com a alta dimensionalidade das variáveis envolvidas, o que pode comprometer a generalização do modelo, uma vez que o número de dimensões de interesse pode crescer exponencialmente e se tornar maior do que o número de exemplos de treinamento disponíveis.

Recentemente, visando superar as dificuldades expostas, novos métodos computacionais denominados de Aprendizado de Máquinas Profundo, do inglês “*Deep Learning*”, vêm sendo propostos e aplicados a diversos problemas de reconhecimento de padrões (Sun et al., 2016; Abdeljaber et al., 2017; Liu et al., 2018). O aprendizado profundo consiste em modelar de forma robusta informações representativas dos dados por meio de aproximações mais complexas de funções e de transformações lineares e não-lineares utilizando várias camadas de processamento (Goodfellow et al., 2016). Devido à sua capacidade de representar os dados em alto nível, de forma concisa e satisfatória, um dos algoritmos mais conhecidos e empregados é o Auto-Codificador Esparso (em inglês, “*Sparse Autoencoder*” – SAE). O SAE é uma rede neural profunda de aprendizado não-supervisionado cujo objetivo é retornar, como saída da rede, a reprodução dos seus dados de entrada, através de uma codificação interna que transforma essas variáveis de entrada (teoricamente de alta dimensão) em “*novas*” variáveis de menor dimensão.

Diante do exposto, objetiva-se no presente trabalho avaliar o uso do Auto-Codificador Esparso aplicado ao problema de detecção de danos estruturais em uma simulação numérica de viga, desenvolvendo-se modelos de classificação baseados em indicadores estatísticos extraídos diretamente das medições dinâmicas no domínio do tempo. Os parâmetros estatísticos são inicialmente utilizados como informações de entrada de um modelo SAE que fornece, como variáveis de saída, vetores de menor dimensão capazes de representar adequadamente esses dados iniciais. Em uma etapa posterior, os vetores de menor dimensão são repassados como entradas de um algoritmo de inteligência computacional, que classifica cada um desses dados

como provenientes de um modelo de viga com ou sem a presença de danos. Utiliza-se como classificadores os algoritmos de Rede Neural Artificial e Máquina de Vetor Suporte, uma vez que foram esses os que forneceram os melhores resultados em estudo anterior (Finotti et al., 2016). Os resultados alcançados com a metodologia descrita, empregando a transformação e redução de parâmetros via SAE, são comparados com resultados de modelos de classificação que utilizam diretamente os parâmetros estatísticos dos sinais como entrada para os mesmos algoritmos de Rede Neural e Vetor Suporte.

2 ESTATÍSTICA APLICADA A DADOS DE VIBRAÇÃO

Os métodos de detecção de alterações estruturais baseados em algoritmos de inteligência computacional são compostos, basicamente, por duas etapas: caracterização dos sinais dinâmicos e classificação. Assim, o uso de características capazes de fornecer informações suficientes e representativas dos dados de vibração é crucial para que as diferentes condições da estrutura sejam reconhecidas e classificadas com sucesso.

Segundo Farrar e Worden (2012), a análise de séries temporais das respostas estruturais permite a extração de parâmetros dos sinais de forma simples, já que possibilitam trabalhar diretamente com as medidas dinâmicas no domínio do tempo. Assim, faz-se neste trabalho o uso de dez diferentes indicadores estatísticos sugeridos por Farrar e Worden (2012) para caracterizar as respostas vibracionais, conforme consta na Tab. 1.

Valor de pico: $x_{pico} = \max \mathbf{x} $	Raiz da média quadrática: $rms = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i)^2}$	Assimetria: $s = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\sigma^4}$	Fator-K: $Kf = x_{pico} \cdot rms$
Média: $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$	Variância: $\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$	Curtose: $k = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\sigma^4}$	
Média quadrática: $x_q = \frac{1}{n} \sum_{i=1}^n (x_i)^2$	Desvio padrão: $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$	Fator de Crista: $Cf = \frac{x_{pico}}{rms}$	

Tabela 1: Indicadores estatísticos.

3 INTELIGÊNCIA COMPUTACIONAL NA DETECÇÃO DE DANOS ESTRUTURAIS

Como já explicitado anteriormente, o presente trabalho utiliza algoritmos baseados em Redes Neurais Artificiais, Máquinas de Vetor Suporte e Auto-Codificadores Esparsos, cujos conceitos serão abordados de forma resumida nos próximos itens.

3.1 Rede Neural Artificial (ANN - Artificial Neural Network)

Rede Neural Artificial é um algoritmo de inteligência computacional de sistema adaptativo que possui a capacidade de aprender e generalizar determinado conjunto de dados a partir da associação de informações previamente conhecidas. Segundo Principe et al. (2000), a classificação feita por uma Rede Neural Artificial é de natureza estatística, onde as classes são representadas por pontos em um espaço de decisão multidimensional dividido em regiões que estão associadas a cada tipo de saída. Os limites de decisão são estimados por um processo adaptativo de treinamento e suas construções são feitas pela variabilidade estatística existente entre as classes, através de um esquema de otimização baseado na avaliação do erro de saída da rede. Um dos tipos de Rede Neural mais comuns é o *perceptron* de múltiplas camadas (MLP - *Multilayer Perceptron*), que é uma rede neural *feedforward* composta por uma camada

de entrada, uma ou mais camadas intermediárias e a camada de saída, onde os elementos de processamento, também conhecidos como neurônios, trabalham com funções cujo objetivo é inserir uma não-linearidade no modelo, chamadas de funções de ativação. Dentre as várias técnicas de aprendizagem existentes para as redes MLP, a mais popular é a *backpropagation*. Nesta técnica, a saída fornecida pela rede é comparada com a resposta correta a fim de se obter um valor de erro por uma métrica predefinida. A partir disto, a técnica propaga o valor do erro obtido na camada de saída para as demais camadas e, posteriormente, atualiza os pesos relacionados a cada conexão da MLP (Bishop, 2006). Esse processo é repetido até que a diferença entre as respostas, alcançada pela rede e o alvo, convirja para um valor de erro previamente especificado. Nesta situação, diz-se que a rede está treinada. O processo de ajuste de pesos se dá através de métodos para otimização não-linear, tais como o método do gradiente descendente, método de Newton, Levenberg-Marquardt, entre outros (Principe et al., 2000). Ao término do treino a rede deverá ter adquirido a capacidade de generalizar resultados para o problema proposto com base nos parâmetros extraídos. Para avaliar o desempenho final do modelo, dados que não foram utilizados na fase de treinamento são classificados a fim de testar a capacidade de generalização da rede. Esta última etapa é denominada fase de teste. Cabe ressaltar que a fase de teste não tem efeito na construção do modelo de classificação, já que fornece apenas uma medida do desempenho da rede para dados ainda não “vistos”.

3.2 Máquina de Vetor Suporte (SVM - Support Vector Machine)

Outra técnica de inteligência computacional popular para o problema de reconhecimento de padrões é a Máquina de Vetor Suporte. As SVMs são desenvolvidas a partir de um algoritmo de aprendizado estatístico treinado para determinar o limite entre duas classes de dados em um espaço, onde um hiperplano ótimo de separação é construído com o intuito de maximizar a margem entre as regiões de cada classe e minimizar as classificações erradas (Vapnik, 1995). No entanto, a teoria da SVM pode ser estendida de problemas de classificação binários para problemas de classificação multiclases, através de estratégias baseadas na combinação de classificadores de duas classes, como as estratégias um contra um e um contra todos, por exemplo (Bishop, 2006). O aprendizado do algoritmo SVM é interpretado como a minimização de uma função de otimização, o que é equivalente a maximizar a distância normal ao hiperplano que separa as duas classes (margem). Os dados de treinamento localizados nas margens são chamados de vetores de suporte. Para classificações não-lineares, as entradas são mapeadas em um espaço de características de dimensão maior que a original através de alguma função de kernel. No presente trabalho, a função utilizada como kernel é a Função Gaussiana, denominada também por Função de Base Radial (RBF - Radial Basis Function). Neste caso, os modelos tem dois parâmetros livres que precisam ser especificados: σ da função do kernel RBF; e C, um parâmetro de regularização a partir da formulação da maximização da margem. Estes parâmetros são selecionados por um procedimento de busca exaustiva conhecido como *Grid Search* (Hsu et al., 2004). Neste método, dois conjuntos finitos de valores (discretos) para σ e C são previamente determinados e empregados no treinamento de modelos SVM, onde todas as combinações são avaliadas através de uma validação cruzada *10-fold* (Kohavi et al., 1995). Por fim, o par de parâmetros (σ , C) que obtiver o modelo com o menor erro de generalização é escolhido.

3.3 Auto-Codificador Esparso (SAE - Sparse Autoencoder)

No que tange à metodologia de Aprendizado de Máquinas Profundo (“*Deep Learning*”), um Auto-Codificador (“*Autoencoder*”) pode ser interpretado como uma Rede Neural Artificial de

treinamento não-supervisionado, construída para retornar como saída uma aproximação da sua respectiva entrada (Rumelhart et al., 1986). Internamente, esse tipo de rede é constituído basicamente por uma camada de codificação descrita por uma função $\mathbf{h} = f(\mathbf{x})$, que “aprende” as características da entrada, e uma camada de decodificação descrita por $\mathbf{y} = g(\mathbf{h})$, que reconstrói a entrada \mathbf{x} (Goodfellow et al., 2016). Em resumo, os Auto-Codificadores são normalmente projetados para produzirem $\mathbf{y} \approx \mathbf{x}$, obtendo-se \mathbf{h} com uma dimensão inferior a de \mathbf{x} . Quando os Auto-Codificadores são empregados somente para o mapeamento dos dados, sem reconstrução das características, são chamados de Auto-Codificadores Incompletos. A grande vantagem de se reduzir a dimensão de \mathbf{x} para \mathbf{h} é a possível identificação de parâmetros relevantes da entrada em alto-nível de abstração, que podem ser bastante úteis para o reconhecimento de padrões do problema em questão. Neste caso, o processo de aprendizado da Rede Neural que o constitui pode ser realizado através da minimização de uma função de penalização das diferenças entre $g(f(\mathbf{x}))$ e \mathbf{x} . Entretanto, caso o codificador e o decodificador tenham excessiva habilidade para aproximar $\mathbf{y} \approx \mathbf{x}$, podem ocorrer situações onde \mathbf{h} tenha uma dimensionalidade alta, não sendo capaz de fornecer parâmetros interessantes para a modelagem do problema que se aborda. A fim de melhorar o desempenho dessa técnica computacional, surgiu o Auto-Codificador Esperso, um Auto-Codificador Incompleto onde uma penalidade denominada esparsa ($\Gamma(\mathbf{h})$) é incorporada à função de penalização no processo de treinamento. Dessa forma, o SAE pode produzir um modelo apto a identificar informações representativas e exclusivas do conjunto de dados para o qual foi treinado.

4 APLICAÇÃO

A abordagem proposta é analisada através de dados sintéticos obtidos por um modelo de elementos finitos de uma viga biapoiada de 6m de comprimento e seção transversal perfil I, desenvolvido no trabalho de Alves (2012), cujas propriedades mecânicas são: módulo de elasticidade longitudinal (E) de 210 GPa; massa específica de 7850 kg/m³; área de seção transversal de 2.81×10^{-3} m² e momento de inércia de 1.845×10^{-8} m⁴.

O modelo é constituído por uma malha de 200 elementos de viga de Bernoulli formados por dois nós com dois graus de liberdade cada um (rotação e translação vertical), excitado por uma força com frequências e amplitudes aleatórias aplicada a 0,69m do apoio direito. As respostas dinâmicas foram consideradas como os deslocamentos verticais medidos em 10 pontos equidistantes da viga durante 100s com frequência de amostragem de 100Hz, simulando uma instrumentação real feita por meio de acelerômetros. A representação da viga modelada é apresentada na Figura 1(a).

Três configurações estruturais foram consideradas: Viga sem danos; Viga com dano, modelada através da redução de 20% do módulo de elasticidade longitudinal no meio do vão, representado pelo trecho em cinza no esquema da viga da Figura 1(a) (dano 1) e; Viga com dano, modelada através da redução de 10% do módulo de elasticidade no quarto de vão da viga, correspondente ao trecho em preto da viga da Figura 1(a), somada a redução anterior de 20% (dano 2). Para cada uma das situações de dano, admitiu-se três níveis de ruído às respostas dinâmicas: Sem ruído, 5% de ruído (ruído 1) e 10% de ruído (ruído 2). A maneira como os ruídos foram gerados e adicionados ao sinal dinâmico é definida pela Eq.(1) abaixo:

$$x_{i,ruído} = x_i + n_{ruído} \cdot \sigma_{x_i} \cdot V \sim N(0, 1) \quad (1)$$

onde $x_{i,ruído}$ é o vetor do sinal com ruído, x_i é o vetor do sinal sem ruído, $n_{ruído}$ é o nível de ruído, σ_{x_i} é o desvio-padrão e $V \sim N(0, 1)$ é um vetor gaussiano de média nula e desvio-padrão unitário.

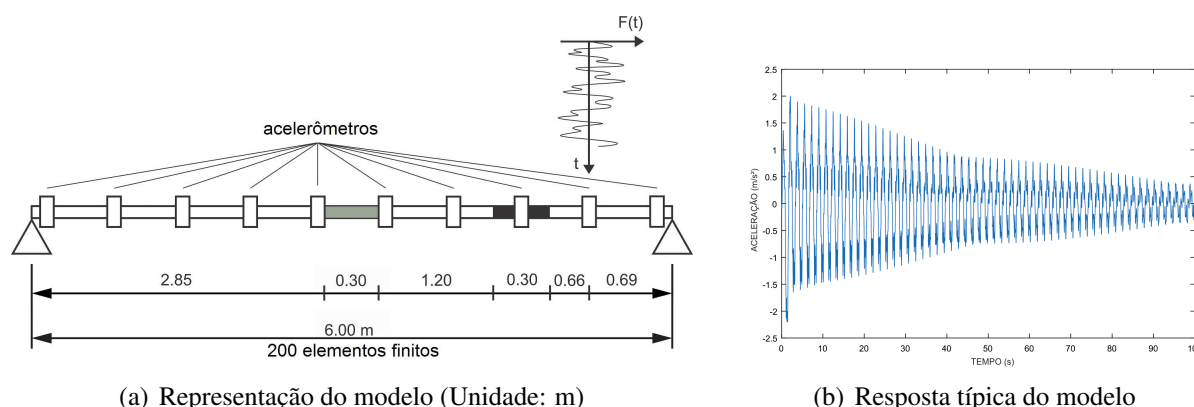


Figura 1: Modelo numérico de viga biapoada.

Dez ensaios foram simulados para cada caso de dano e ruído, totalizando 90 ensaios (3 configurações estruturais × 3 níveis de ruído × 10 ensaios = 90 ensaios). A Figura 1(b) mostra uma resposta típica da estrutura. Posteriormente, os dez indicadores apresentados na seção 2 foram aplicados aos históricos das acelerações estruturais. Considerou-se o ensaio ao longo de todo o comprimento da viga, desta forma os dados de entrada estão organizados em uma matriz [90×100], onde as linhas representam o sinal em questão e as colunas são os indicadores estatísticos (10 indicadores × 10 pontos de medição simulada computacionalmente = 100 indicadores).

Como a classificação em si é realizada por ANN e SVM (aprendizado supervisionado), é necessário também apontar a classe de dano para cada ensaio. Por esse motivo, para os algoritmos ANN é gerada uma matriz alvo [90×3], em que as linhas indicam a categoria do sinal através da seguinte codificação binária: (1 0 0) - Sem dano; (0 1 0) - Dano 1 e, (0 0 1) - Dano 2. Enquanto que, para os classificadores SVM, as três diferentes condições estruturais são representadas por um vetor alvo codificado como: 1 - Sem dano; 2 - Dano 1 e, 3 - Dano 2.

5 RESULTADOS

Duas estratégias diferentes de classificação são realizadas neste trabalho: uma utilizando diretamente a matriz de dados [90×100] como parâmetro de entrada dos algoritmos ANN e SVM (abordagem tradicional); e outra utilizando como entrada dos algoritmos de classificação as características “aprendidas” pelo SAE a partir da matriz de dados [90×100], conforme exemplificado na Figura 2. Em cada um desses casos os modelos de classificação ANN e SVM

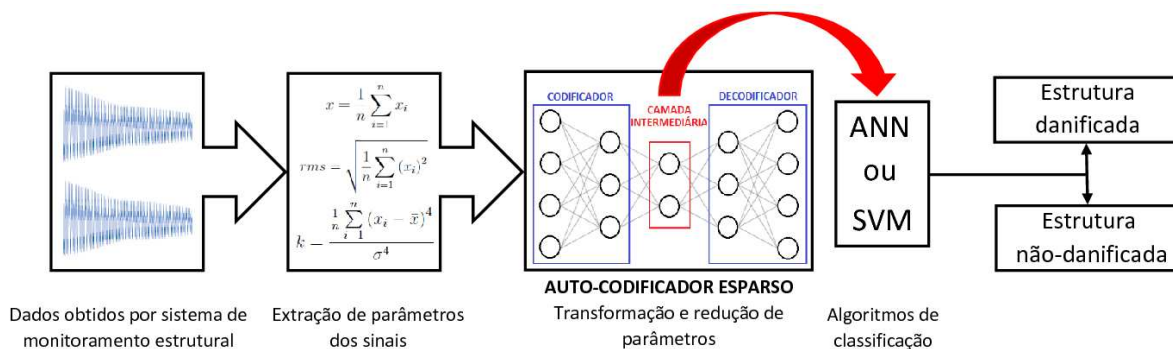


Figura 2: Procedimento de identificação de alterações estruturais utilizando SAE.

foram executados e treinados 30 vezes aplicando-se o método de validação cruzada *10-fold*. As Tabelas 2 e 3 apresentam os resultados obtidos pelos modelos de detecção na fase de teste, em que a taxa de classificação é representada pela média das 30 repetições (nº de acertos dividido pelo nº total de ensaios classificados).

	Média	Desvio-padrão	Melhor classificação	Pior classificação
ANN	86,78%	6,25%	95,56%	73,33%
SVM	99,41%	1,04%	100,00%	96,67%

Tabela 2: Resultados de classificação sem o uso do Auto-Codificador Esparso (melhores parâmetros SVM: $\sigma = 3,0000$ e $C = 8,5000$).

	Média	Desvio-padrão	Melhor classificação	Pior classificação
ANN	87,93%	3,49%	94,44%	81,11%
SVM	92,70%	3,24%	98,89%	83,33%

Tabela 3: Resultados de classificação com o uso Auto-Codificador Esparso (melhores parâmetros SVM: $\sigma = 0,0316$ e $C = 316,2278$).

Os modelos de classificação ANN implementados são do tipo MLP criados com 1 camada oculta composta por 10 neurônios para ambas as estratégias (entrada $[90 \times 100]$ e características fornecidas pelo SAE). O número de neurônios na camada oculta foi definido através de experimentos prévios, considerando-se o melhor desempenho nas classificações ao se treinar as redes para diferentes quantidades de neurônios. Por ter obtido melhores resultados de classificação, o método de otimização de Levenberg-Marquardt foi escolhido como função de treinamento, utilizando a média quadrática como métrica de erro e uma sigmóide logística como função de ativação.

Por se tratar de um problema de classificação não-linear, o modelo SVM foi elaborado utilizando como kernel uma função de base radial (RBF - *Radial Basis Function*), onde os melhores parâmetros σ e C são selecionados através de um treinamento com diferentes valores deles em uma validação cruzada *10-fold* (procedimento de busca exaustiva). Como a aplicação é multiclasse, optou-se pela estratégia de classificação "um contra um", que é a mais utilizada na literatura.

Na segunda estratégia, o modelo SAE foi configurado com duas camadas intermediárias (codificadoras), ambas utilizando 10 neurônios, ou seja, redução de 100 variáveis do conjunto de dados de entrada para apenas 10. Salienta-se que, para este caso, a quantidade de neurônios adotados é o que determina o tamanho do conjunto de variáveis que irão formar os dados de entrada do SVM e da ANN. A taxa de aprendizado do SAE foi fixada em 0,001 e os parâmetros de regularização e proporção relativos à esparsidade foram definidos como sendo 4,00 e 0,05 respectivamente (escolhidos por apresentarem melhor desempenho em testes anteriores). Cabe ressaltar que o treinamento do modelo SAE é realizado antes do treinamento dos classificadores ANN e SVM acima descritos, uma vez que as características fornecidas pelo SAE são utilizadas para inicializar a rede MLP e o SVM neste tipo de abordagem.

Os resultados mostram o bom desempenho alcançado pelo SAE no aprendizado e redução das características dos indicadores estatísticos dos sinais dinâmicos. Para o caso da rede neural

sem o uso do SAE, observa-se um desvio-padrão alto (maior que 6%) quando comparado aos outros obtidos (em torno de 3%). Este resultado pode ser atribuído à grande quantidade de parâmetros (100) comparada ao tamanho do banco de dados (90), o que dificulta o processo de aprendizado do algoritmo. Percebe-se também que o algoritmo SVM sem a aplicação do SAE forneceu taxas de classificação ligeiramente superiores do que o SVM cujas entradas advinham do SAE. No entanto, por trabalhar com um número menor de parâmetros de entrada, o classificador SVM que usa o SAE tem vantagens quanto ao processamento computacional do modelo de detecção.

Devido aos resultados atingidos pelo classificador SVM, uma análise considerando a taxa de classificação em função do número de neurônios presentes nas camadas intermediárias do SAE é realizada para esse algoritmo. A Figura 3 apresenta o gráfico com os valores obtidos, onde se pode verificar que o aumento na porcentagem de classificações corretas não é proporcional à quantidade de neurônios. Avaliando-se as taxas de identificação, o desempenho do classificador é melhor quando 20 neurônios são utilizados nas camadas do SAE, atingindo valores em torno de 97%. Embora os resultados para 30 e 40 neurônios apresentem um decréscimo (classificação em torno de 95%), observa-se uma tendência de crescimento com 50 neurônios, onde tem-se novamente uma taxa de acerto em torno de 97%. Isto provavelmente se deve ao fato de que o uso de uma quantidade maior de neurônios pode “aprender” características muito específicas e peculiares do problema, comprometendo a capacidade de generalização do modelo SVM (*overfitting*).

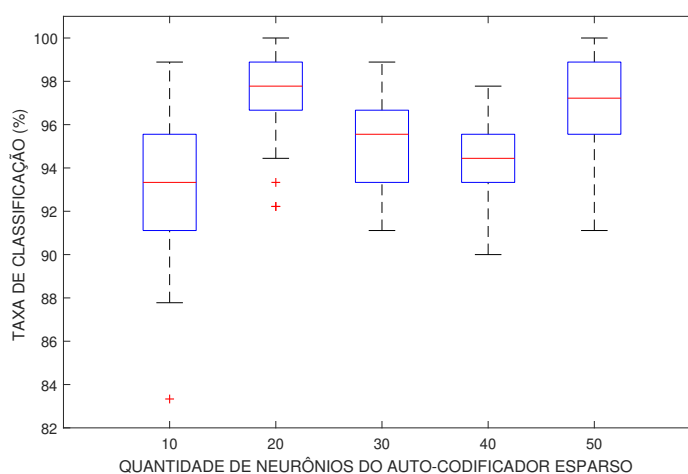


Figura 3: Taxa de classificação em função do número de neurônios presentes nas camadas intermediárias do Auto-Codificador Esperso.

Os modelos computacionais desse trabalho, assim como os indicadores estatísticos, foram desenvolvidos utilizando *toolboxes* e funções built-in disponíveis no *Matlab*.

6 CONCLUSÕES

O presente estudo investigou o uso do Auto-Codificador Esperso aplicado ao problema de identificação de danos em um modelo numérico de viga bi-apoiada através de classificadores baseados em algoritmos de Rede Neural Artificial e Máquina de Vetor Suporte. Indicadores estatísticos extraídos das respostas dinâmicas no domínio do tempo foram utilizados como parâmetros de entrada do SAE, que retornava como saída, informações representativas dessa mesma

entrada. As saídas fornecidas pelo algoritmo de auto-codificação eram posteriormente repassadas como variáveis para o treinamento dos modelos ANN e SVM. Duas estratégias diferentes de classificação foram realizadas, uma utilizando diretamente as estatísticas como matriz de entrada dos classificadores e outra fornecendo como entrada dos algoritmos de classificação as características “aprendidas” pelo SAE a partir dos dados estatísticos.

Ambos os algoritmos, ANN e SVM, alcançaram taxas satisfatórias de classificação para as estratégias propostas, com médias de acerto acima de 85%. Embora o desempenho tenha sido similar para as duas estratégias, destacam-se aqui os resultados obtidos ao se utilizar o algoritmo de aprendizado SAE, já que os modelos de classificação trabalharam com um volume menor de dados quando comparados aos modelos da primeira estratégia.

A ideia do Auto-Codificador é que cada camada seja responsável pelo aprendizado de uma característica específica do conjunto de dados a ela apresentado. Assim, por ser capaz de captar mais detalhes desse conjunto, o uso de mais camadas pode melhorar o desempenho do modelo SAE. Levando em consideração que os dados desse trabalho provêm de uma simulação numérica de uma viga bi-apoiada e que por isso, as respostas estruturais são mais bem definidas e comportadas, a arquitetura definida para o SAE utilizando apenas duas camadas intermediárias foi suficiente. Entretanto para o caso de estruturas reais, geralmente mais complexas e com um número maior de graus de liberdade, provavelmente haja a necessidade de se incluir mais camadas no Auto-Codificador, uma vez que os dados se tornam mais difíceis de interpretar e apresentam dimensionalidade muito alta.

É importante ressaltar que esse é um estudo inicial com resultados parciais, porém a análise aqui apresentada aponta que o método SAE tem potencial a ser explorado na área de determinação de características dos sinais de vibração para resolver os problemas de detecção de alterações estruturais. Como o sucesso de sistemas de identificação de danos baseados em técnicas de inteligência computacional depende, em grande parte, da escolha adequada dos parâmetros de entrada desses algoritmos, as considerações anteriores encorajam o desenvolvimento de ferramentas computacionais de aprendizado profundo para a avaliação de danos. Estudos desse tipo possuem grande utilidade dentro da área de monitoramento estrutural por fornecer informações da integridade da estrutura. Assim, pretende-se ainda em trabalhos futuros verificar o desempenho do SAE para outros tipos de parâmetros de entrada e, também, em outras estruturas. Destaca-se também a necessidade de uma análise mais detalhada de melhores parâmetros para a ANN e SVM, pois nesta pesquisa foi usada a mesma arquitetura de rede e parâmetros σ e C , tanto para uma maior quanto para uma menor quantidade de dados. Assim, ao se modificar o tamanho da entrada desses algoritmos, talvez seja necessário utilizar outras configurações para que os modelos se adequem melhor ao problema em questão.

AGRADECIMENTOS

Os autores agradecem à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), à FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais), ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) e à UFJF (Universidade Federal de Juiz de Fora) pelo apoio financeiro.

REFERÊNCIAS

Abdeljaber O., Avci O., Kiranyaz S., Gabbouj M., e Inman D.J. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, 388:154–170, 2017.

- Alves V., Cury A., Roitman N., Magluta C., e Cremona C. Structural modification assessment using supervised learning methods applied to vibration data. *Engineering Structures*, 99:439–448, 2015.
- Alves V.N. *Estudo de novas estratégias para identificação de danos estruturais a partir de dados vibracionais*. Dissertação de Mestrado, Universidade Federal de Ouro Preto, 2012.
- Amezquita-Sanchez J.P. e Adeli H. Signal processing techniques for vibration-based health monitoring of smart structures. *Archives of Computational Methods in Engineering*, 23(1):1–15, 2016.
- Bishop C.M. *Pattern recognition and machine learning*. Springer-Verlag, 2006.
- Cardoso R., Cury A., e Barbosa F. A robust methodology for modal parameters estimation applied to shm. *Mechanical Systems and Signal Processing*, 95:24–41, 2017.
- Cury A.A., Borges C.C., e Barbosa F.S. A two-step technique for damage assessment using numerical and experimental vibration data. *Structural Health Monitoring*, 10(4):417–428, 2011.
- Doebling S.W., Farrar C.R., e Prime M.B. A summary review of vibration-based damage identification methods. *Shock and Vibration Digest*, 30(2):91–105, 1998.
- Farrar C.R. e Worden K. *Structural health monitoring: a machine learning perspective*. John Wiley & Sons, 2012.
- Finotti R.P., Bonifácio A.L., Barbosa F.S., e Cury A.A. Evaluation of computational intelligence methods using statistical analysis to detect structural damage. *Mecánica Computacional*, 34(22):1389–1397, 2016.
- Ge Z., Kruger U., Lamont L., Xie L., e Song Z. Fault detection in non-gaussian vibration systems using dynamic statistical-based approaches. *Mechanical Systems and Signal Processing*, 24(8):2972–2984, 2010.
- Goodfellow I., Bengio Y., Courville A., e Bengio Y. *Deep learning*. MIT press Cambridge, 2016.
- Hsu C.W., Chang C.C., Lin C.J., et al. *A practical guide to support vector classification*. Technical Report, Department of Computer Science and Information Engineering, National Taiwan University, 2004.
- Kohavi R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. Em *Ijcai*, volume 14, páginas 1137–1145. Montreal, Canada, 1995.
- Liu H., Zhou J., Zheng Y., Jiang W., e Zhang Y. Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders. *ISA transactions*, 77:167–178, 2018.
- Principe J.C., Euliano N.R., e Lefebvre W.C. *Neural and adaptive systems: fundamentals through simulations*. Wiley New York, 2000.
- Rumelhart D.E., Hinton G.E., e Williams R.J. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.
- Sakaris C.S., Sakellariou J.S., e Fassois S.D. Random-vibration-based damage detection and precise localization on a lab-scale aircraft stabilizer structure via the generalized functional model based method. *Structural Health Monitoring*, 16(5):594–610, 2017.
- Sun W., Shao S., Zhao R., Yan R., Zhang X., e Chen X. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement*, 89:171–178, 2016.
- Vapnik V. *The nature of statistical learning theory*. Springer-Verlag, 1995.
- Wickramasinghe W.R., Thambiratnam D.P., Chan T.H., e Nguyen T. Vibration characteristics and damage detection in a suspension bridge. *Journal of Sound and Vibration*, 375:254–274, 2016.