

Identificação do Estágio de Propagação de Descontinuidade em um Corpo Rígido Tubular utilizando Emissão Acústica e Redes Neurais*

Luiz Rennó Costa¹
Luiza Ribeiro Marnet²
Carlos Fernando Carlim Pinto³
Francesco Nosedà⁴
Luiz Pereira Calôba⁶

Resumo

Testes destrutivos e não-destrutivos são a base para a compreensão das propriedades físicas da falha do material, esses testes são feitos durante a fabricação de gasodutos ou líquidos, para que se possa identificar quais níveis ou padrões de pressão devem ser evitados. Além disso, uma vez que o sistema é submetido a alta pressão, qualquer pequeno defeito pode se propagar e ter sérias consequências, portanto, uma identificação e análise adequadas de um defeito é de extrema importância e deve ser resolvido rapidamente. Métodos na literatura usam o fato de que tal sistema emite ondas acústicas e tentam determinar o grau de perigo com a ajuda de um especialista. Este artigo propõe então usar um modelo de rede neural *feedforward* parcialmente autônomo para determinar quanto perigo esse defeito representa ao tentar separá-lo em três classes: Sem Propagação, Propagação Estável e Propagação Instável. A obtenção de uma possível taxa de classificação de 82.9% indica que essa ferramenta pode ser usada para ajudar especialistas com facilidade.

Palavras-chave: Redes Neurais, Emissão Acústica, Indústria 4.0.

IDENTIFYING THE PROPAGATION STAGE OF A DISCONTINUITY ON A RIGID TUBULAR BODY USING ACOUSTIC EMISSION AND NEURAL NETWORKS

Abstract

Destructive and non-destructive tests are the base to understanding the physical properties of material failure, these tests are done throughout the making of gasoducts or liquiduct so one can identify which levels or patterns of pressure should be avoided. Moreover, since the system is subjected to high pressure, any small defect can propagate and burst, therefore, a proper identification and analysis of said defect is of utmost importance and must be done with haste. Methods in the literature use the fact that such system emits acoustic waves, and attempt to analyze their data with the help of a specialist. This paper then proposes to use a feedforward neural network model to identify how much danger this defect represents by trying to separate it into three classes, No Propagation, Stable Propagation and Unstable Propagation. Achieving a possible classification rate of 82.9% indicates that such a tool can be used to aid specialists with ease.

Keywords: Neural Network, Acoustic Emission, 4.0 Industry.

- ¹ Engenharia Eletrônica e de Computação, M.Sc., Programa de Engenharia Elétrica, UFRJ, Rio de Janeiro, RJ, Brasil.
- ² Engenharia de Controle e Automação, M.Sc., Programa de Engenharia Elétrica, UFRJ, Rio de Janeiro, RJ, Brasil.
- ³ Engenharia Eletrônica e de Computação, D.Sc., Programa de Engenharia Elétrica, UFRJ, Rio de Janeiro, RJ, Brasil.
- ⁴ Ph.D., Instituto de Matemática, UFRJ, Rio de Janeiro, RJ, Brasil.
- ⁵ Engenharia Eletrônica, Dr.Ing., Programa de Engenharia Elétrica, UFRJ, Rio de Janeiro, RJ, Brasil.

1 INTRODUÇÃO

Grandes desastres da engenharia costumam envolver uma má manutenção preventiva aliada à alguma espécie de erro humano [8]. Rompimentos, por exemplo, de tubulações de abastecimento de gás veicular natural (GNV) podem causar explosões de proporções catastróficas. Considerando também que estas malhas de distribuição podem ter centenas de quilômetros de extensão, caso haja alguma fratura física potencialmente perigosa, a sua localização não é simples de determinar, além disto, para efetuar eventuais reparos é necessário que o sistema não esteja em operação.

Ou seja, não identificar o problema (falso negativo) pode causar desastres das mais variadas proporções, por outro lado, caso o sistema tenha que ser parado e não haja necessidade (falso positivo) este vai causar perdas monetárias ao prestador do serviço que utiliza a infraestrutura.

Neste contexto, diferentes técnicas de avaliação, identificação e manutenção de problemas físicos em infraestrutura foram desenvolvidos, entre eles, a técnica de análise de emissão acústica (EA). Esta costuma ser feita como auxiliar à uma inspeção baseada em risco [15], [16], [4].

A análise dos dados de EA é feita de maneira subjetiva, onde um especialista é provido de um conjunto extenso de medições de diversos parâmetros da emissão acústica (discutidos em mais detalhes nas Seções 2 e 4) e ao analisar o perfil destes, ele determina se há necessidade de reparos imediatos ou não.

É neste ambiente que este artigo se insere, propondo uma técnica muito menos subjetiva baseada em redes neurais artificiais para a determinação de manutenção ou não de um corpo rígido em operação. São determinadas 3 classes nas quais a rede deve classificar os dados de EA, Sem Propagação (SP), Propagação Estável (PE) e Propagação Instável (PI), correspondendo aos 3 estágios possíveis da propagação de uma trinca, o primeiro sem deformação (ou com deformação desprezível), deformação elástica e deformação plástica [14]. O estágio de PI é o mais perigoso, como o próprio nome sugere, quando a trinca entra nesse estágio o desastre é praticamente impossível de se evitar.

2 Emissão Acústica

Emissão acústica (Figura 2) é uma onda física resultante da rápida liberação de energia de um material quando este sofre mudanças micro ou macroestruturais [7]. Dentre as várias possíveis causas de tal fenômeno, este artigo se concentrará na EA proveniente da propagação de descontinuidades (trincas) no material (Figura 1).

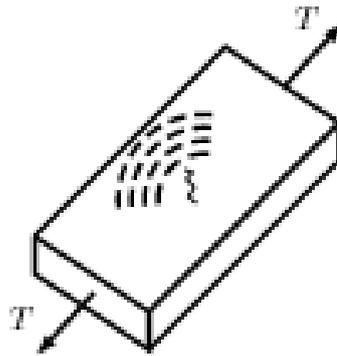


Figure 1: Um bloco sólido com uma descontinuidade em seu centro sendo tracionado emitindo uma onda de emissão acústica

A EA é bastante utilizada como ferramenta de auxílio em procedimentos de inspeção e manutenção de infraestrutura [10], [2], [10]. Esta técnica se baseia em dados e parâmetros coletados utilizando sensores piezoelétricos na estrutura (ou em alguma capa de proteção que ela possa vir a possuir) que convertem a onda física em impulsos elétricos. O cálculo e definição dos parâmetros da onda são feitos em cima de ondas capturadas denominadas de *hits*. Para se iniciar a captura de um *hit* (Figura 2), basta que o sinal lido do sensor ultrapasse um *threshold* (THR) 1 vez. Diversos parâmetros são calculados em cima de um *hit*. O detalhamento dos parâmetros utilizados neste artigo se encontram na Seção 4.

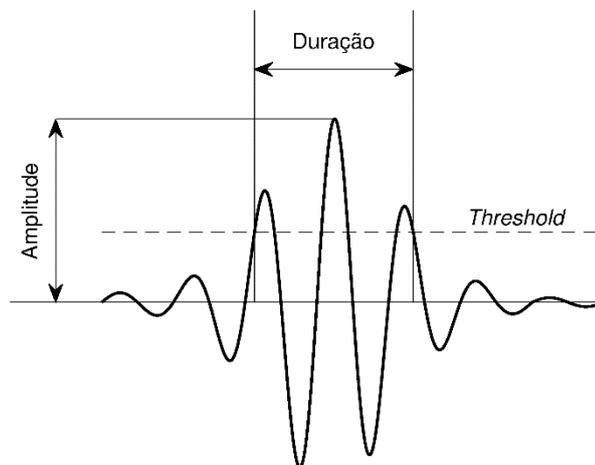


Figure 2: Um sinal teórico de emissão acústica, com *threshold*, amplitude e duração ilustrados.

3 Rede Neural *Feedforward*

A rede neural artificial, como o nome sugere, é um modelo matemático inspirado na estrutura neural humana. O modelo mais simplificado da rede é o *perceptron* [6], a unidade mais básica de uma rede neural artificial. O *perceptron* pode ser entendido como o "neurônio" da rede, recebendo as entradas x ponderadas por um vetor de pesos (sinapses) w e efetuando uma função $\Phi(\cdot)$ em cima do somatório destas entradas ponderadas (Figura 3).

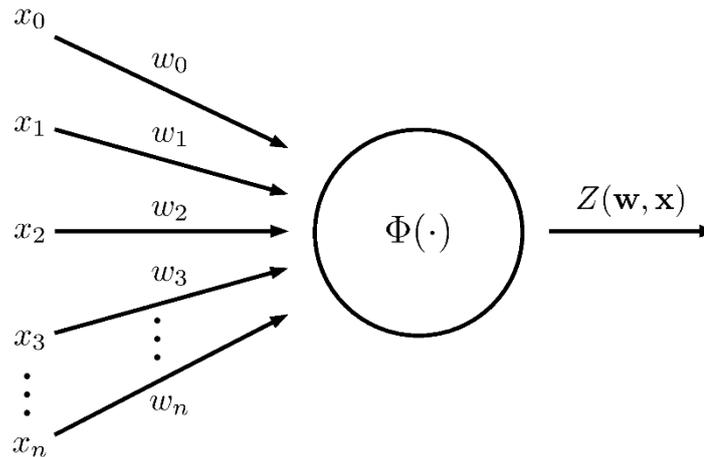


Figure 3: Modelo matemático do *perceptron*, onde $Z(w, x) = \Phi(wx)$.

Uma rede neural usual é composta de diversos *perceptrons* espalhados em pelo menos uma camada onde as camadas anteriores estão completamente ligadas nas posteriores (Figura 4), esta estrutura é denominada de *Multilayer Perceptron* (MLP) e como as conexões seguem apenas um sentido entre as camadas, este tipo de rede é denominada *feedforward*.

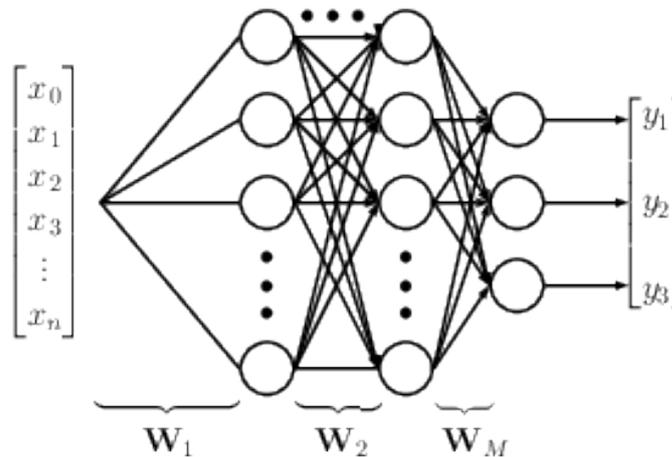


Figure 4: Estrutura de uma rede neural *feedforward* de M camadas e 3 saídas.

Esta estrutura é bastante utilizada devido a sua alta capacidade de mapeamento entre o conjunto de entrada x e o de saída y , podendo ser utilizada como um classificador se y for uma variável categórica, ou como um aproximador de uma função $y = f(x)$ qualquer. Por ser um modelo relativamente simples de implementar, também é comum encontrar modificações implementando por exemplo, um algoritmo de Colônia de Formigas [3] dentro da rede neural [17].

O primeiro algoritmo desenvolvido para determinar os valores das matrizes de peso W é chamado de *backpropagation*, detalhado na Seção 3.1. Este algoritmo é uma descida de gradiente aplicado à uma função custo definida pelo usuário do modelo, ou seja, determinar os pesos é um problema de otimização. Com essa interpretação, se tornou possível implementar diversos otimizadores de ordens maiores ou estocásticos como *Levenberg-Marquardt* [12], *ADAM* [9], *SGD* [1], *L-BFGS* [18], etc.

3.1 Backpropagation

Supondo que a rede faz um mapeamento aproximado da função $f(\cdot)$ e chamando y^* a saída real e y a saída do modelo, é possível definir uma função de erro tal que:

$$J(e) = E \left[\frac{e^2}{2} \right] = E \left[\frac{\|y^* - y\|^2}{2} \right] \quad (1)$$

Onde $E[\cdot]$ é o operador de valor esperado. Se existem N amostras de saída y^* e entrada x , a Equação 1 se torna:

$$J(e) = E \left[\frac{e^T e}{2} \right] = \frac{1}{2N} \sum_{i=0}^{N-1} \|y^{*(i)} - y^{(i)}\|^2 \quad (2)$$

Tendo $y = \Phi(W_M z_{(M-1)})$ e sendo $z_{(M-1)}$ a saída da penúltima ($M - 1$) camada :

$$\frac{\partial J(e)}{\partial W_M} = \frac{\overbrace{\partial \Phi(W_M z_{(M-1)})}^{\alpha_M}}{\partial z_{(M-1)}} \circ e z_{(M-1)}^T \quad (3)$$

Onde o operador \circ representa o produto de Hadamard, um produto elemento-a-elemento. Calculando agora a derivada parcial em função dos pesos da penúltima ($M - 1$) camada, sendo $u = z_{(M-2)}$ a saída da antepenúltima ($M - 2$) camada:

$$\frac{\partial J(e)}{\partial W_{(M-1)}} = \frac{\overbrace{\partial \Phi(W_{(M-1)} u)}^{\alpha_{M-1}}}{\partial u} \circ W_M^T \alpha_M u^T, \quad (4)$$

Definindo $\alpha_{(M-1)}$, α_M nas Equações 3, 4 e fazendo (sem perda de generalidade):

$$W_{(M+1)} = I \quad (5)$$

$$\alpha_{(M+1)} = \frac{\partial J(e)}{\partial e} \quad (6)$$

$$z_i = \text{Saída da Camada } i \quad (7)$$

Utilizando as Equações 5, 6 e 7 é possível então definir que o gradiente da função custo em função da matriz de pesos da camada i , W_i é:

$$\frac{\partial J(e)}{\partial W_i} = \alpha_{(i+1)} z_{(i-1)}^T \quad (8)$$

O termo α_i pode então ser interpretado como a proporção do erro da camada $i+1$ que foi propagado para a camada i . Este algoritmo portanto ganha o nome de *backpropagation*.

Com a informação do gradiente $\partial J/\partial W$ é possível utilizar diversos métodos de otimização mencionados na Seção 3.

4 Metodologia

Os dados utilizados foram capturados em um ensaio destrutivo realizado pelo Laboratório de Metalurgia Física (LAMEF) na Universidade Federal do Rio Grande do Sul (UFRGS). O ensaio foi feito em um corpo de prova cilíndrico, fechado em uma extremidade, com 40m de comprimento, 20 polegadas de diâmetro e 1.45cm de espessura com uma trinca semi-elíptica ao longo de seu comprimento com profundidade aproximadamente igual à metade da espessura do tubo. Importante ressaltar que este corpo de prova também possui uma capa protetora de borracha utilizada para amortecer eventuais oscilações. Foram colocados ao longo do tubo 16 sensores (Figura 4) relevantes ao artigo (representados por quadrados com números em seu interior) e mais 12 sensores cujos dados não foram utilizados.

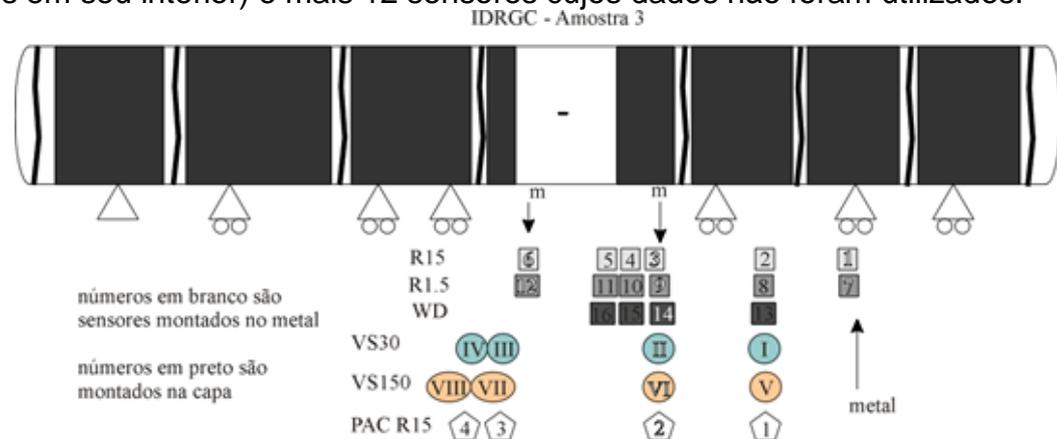


Figura 5: Posicionamento dos sensores ao longo do corpo de prova.

O ensaio consiste em pressurizar o tubo com água à 10% da pressão máxima (300 bar, determinada através do limite de escoamento do material) a cada 15 minutos, a atingindo portanto em pouco mais de 2 horas, 9000 segundos. Existe uma fase de calibração dos sensores que consiste na quebra de um cilindro de grafite com 5mm de diâmetro próximo aos sensores e uma de enchimento do tubo que precedem o ensaio. Durante todo o teste são realizadas medições para determinar as dimensões da trinca tanto radial quanto longitudinalmente através de *strain-gauges* colocados em suas extremidades e sinais de ultrassom (denominados de TOFD).

Os dados utilizados neste artigo estão contemplados em arquivos criados através de uma rotina escrita no LabVIEW, programa da National Instruments (NI), feita pelos engenheiros do LAMEF. Esta salva os dados brutos dos sensores no formato TDMS (específico da NI) e totalizam 900 Gigabytes (GB) de dados. Este ensaio foi dividido em 1500 arquivos onde cada um contém o sinal dos 16 canais discretizados à uma taxa de 2.5MHz com 2^{24} pontos por canal por arquivo, uma matriz $2^{24} \times 16$.

Estes dados brutos foram processados por um *script* feito no MATLAB que identifica e captura eventos de emissão acústica provenientes da trinca para serem usadas posteriormente no modelo. Devido ao grande volume de dados e arquivos, o primeiro passo do *script* consiste em determinar quais arquivos possuem (potencialmente) informações relevantes.

Portanto, se estipulou empiricamente que o sinal não poderia ter uma resolução com mais de (aproximadamente) 1% do fundo de escala, o equivalente a 7 bits. Sabendo que o sinal é bifásico, também deve existir 1 bit de sinal, totalizando um mínimo de 8 bits para a codificação. Para cada um dos 1500 arquivos e 16 canais foi calculado quantos bits foram utilizados no processo de quantização, todos os arquivos com menos de 8 bits foram ignorados.

Após este procedimento de escolha de arquivos, a captura das formas de onda foi feita como especificado na documentação do aparelho de referência da *Pacific Acoustic System (PAC)*, o DiSP 16BT [11] junto com as definições da Equação 9.

Assim que qualquer sinal ultrapassar o THR, a janela de captura é iniciada começando PDT (do inglês, *Peak Definition Time*) segundos antes do disparo, até que o sinal volte a ficar abaixo do THR por HDT (do inglês, *Hit Definition Time*) segundos. Para evitar sinais espúrios e garantir que a próxima captura será uma de EA, a forma de onda seguinte só poderá ser registrada após HLT (do inglês, *Hit Lockout Time*) segundos. Foram utilizados:

$$\begin{aligned} HDT &= 1000\mu\text{s}, \\ HLT &= 1000\mu\text{s}, \\ PDT &= 800\mu\text{s} \end{aligned} \quad (9)$$

O THR é calculado de maneira adaptativa estimando o nível de ruído por canal por arquivo da seguinte forma:

$$THR_{i,k} = (3\sigma_{X_{i,k}}) \cdot \varphi \quad (10)$$

Onde φ é um valor empiricamente definido, $\varphi = 3$, $X_{i,k}$ é o vetor de dados (contendo 2^{24} pontos) do arquivo i , canal k e $\sigma_{X_{i,k}}$ o seu desvio padrão (Equação 11).

$$\sigma_x = \sqrt{\frac{1}{N} \sum_1^N (X - \bar{X})^2} \quad (11)$$

Com isso, 130 formas de onda foram capturadas. Analisando o tempo de captura das ondas junto com o desenvolvimento da trinca (Figura 5) é possível separá-las nas 3 classes de interesse, SP, PE e PI. Desta maneira, o conjunto é separado em 83 amostras da classe SP, 29 PE e 18 PI. Além disto, diversos parâmetros foram calculados a partir de cada forma de onda, os utilizados como entradas são:

- Amplitude Máxima (dB)
 - Maior amplitude da EA (Figura 2) medida em decibéis.
- Rise Time (s)
 - Tempo entre o início da EA e o tempo da Amplitude Máxima.
- Count
 - Quantidade de picos que ultrapassam positivamente o *Threshold*, a Figura 2 representa uma EA com *Count* = 3.
- Energia
 - Energia clássica da emissão acústica.
- Duração (s)
 - Duração da emissão acústica. RMS Valor RMS da energia da forma de onda.
- Count-To-Peak
 - Análogo à variável *Count*, mas se contam apenas os picos entre o início do *hit* até o tempo da amplitude máxima. Figura 2 representa uma EA com *Count - To - Peak* = 2.
- Frequência de Reverberação
 - $(Count - Count-To-Peak) / (Duração - Rise Time)$.
- Threshold
 - Calculado através da Equação 10.

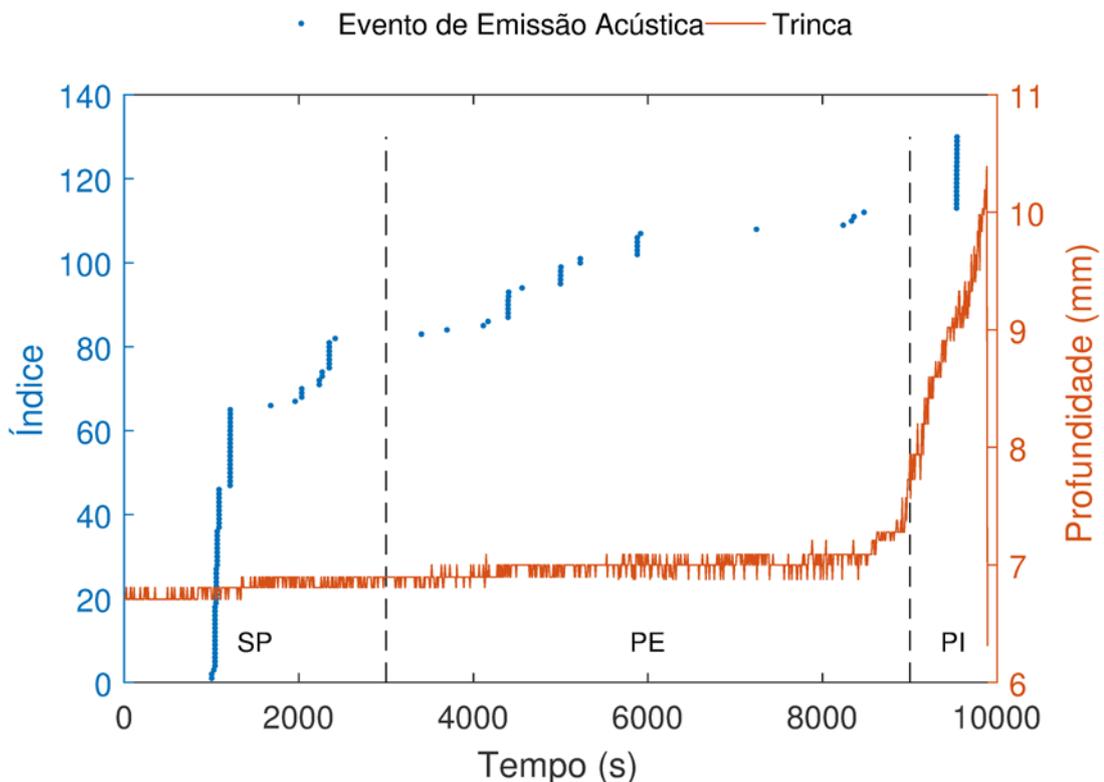


Figure 6: Distribuição de capturas de eventos de emissão acústica no decorrer do ensaio junto da propagação da trinca.

Além destas entradas calculadas diretamente no domínio do tempo, outras seis foram incluídas a partir do espectro de frequência da AE. Estas são definidas como a potência espectral relativa de uma faixa de frequência (Equação 12).

$$I = \frac{\sum_{\omega_0}^{\omega_f} |F(x)|^2}{\sum_{-\infty}^{\infty} |F(x)|^2} \quad (12)$$

Onde $|F(x)|$ é o módulo da transformada de Fourier da EA x , ω_0 e ω_f são as frequências inicial e final (em Hertz) respectivamente. Estas faixas de frequência foram determinadas através de uma análise visual do gráfico de correlação entre a potência espectral normalizada com a saída. A região de interesse é definida tal que sua correlação para com uma classe de saída possui um sinal diferente quando comparada com outra classe.

Por exemplo, comparando a correlação do espectro com a classe SP e com a classe PE (Figura 6) é possível determinar cinco regiões marcadas com asteriscos onde o sinal da correlação é diferente entre as classes e também maiores que o mínimo aceitável (linha pontilhada).

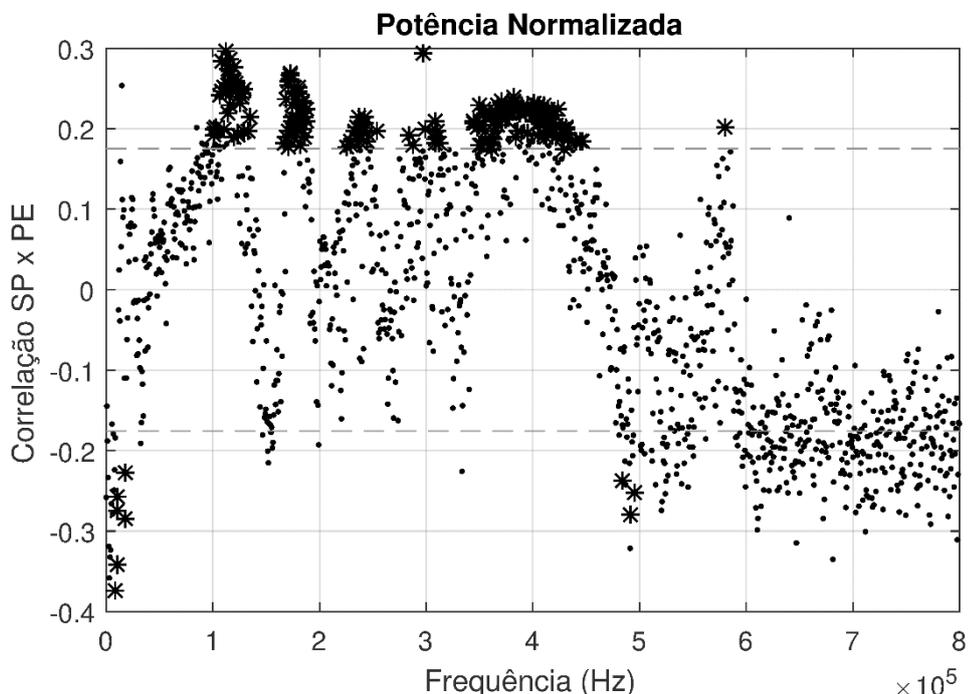


Figure 7: Exemplo de gráfico de correlação do espectro de potência com a saída da classe SP comparado com a classe PE.

Portanto, as faixas Ω escolhidas foram as cinco explicitadas anteriormente e mais uma não incluída no gráfico contido na Figura 6, totalizando 6 regiões:

$$\begin{aligned}
 \Omega(1) &= [0 - 22.6] \text{ KHz} \\
 \Omega(2) &= [99.5 - 138.1] \text{ KHz} \\
 \Omega(3) &= [168.8 - 190.1] \text{ KHz} \\
 \Omega(4) &= [226.5 - 251.8] \text{ KHz} \\
 \Omega(5) &= [283.5 - 315.3] \text{ KHz} \\
 \Omega(6) &= [347.5 - 442.3] \text{ KHz}
 \end{aligned} \quad (13)$$

O modelo utilizado foi uma rede MLP com 1 camada escondida de 10 neurônios. A função de ativação foi $\Phi(\cdot) = \tanh(\cdot)$ e a saída das classes foi codificada de maneira maximamente esparsa (Equação 14), as entradas foram normalizadas utilizando normalização z-score (Equação 15) e o conjunto de dados foi separado em Treino e Validação utilizando 65% e 35% dos dados respectivamente. O baixo número de amostras (130) faz com que a inclusão do conjunto de Teste implicasse em resultados muito voláteis.

$$\begin{aligned}
 SP &= [1 \ 0 \ 0]^T \\
 PE &= [0 \ 1 \ 0]^T \\
 PI &= [0 \ 0 \ 1]^T
 \end{aligned} \quad (14)$$

$$X' = \frac{X - \bar{X}}{\sigma_x} \quad (15)$$

Também foi implementado um balanceamento de classes utilizando *resampling*, fazendo com que as classes no conjunto de treinamento tenham sempre o mesmo número de amostras repetindo as amostras das classes menores.

O treinamento foi feito utilizando a *Neural Network Toolbox* do MATLAB, com a função de treinamento *trainbr*, que implementa *Levenberg-Marquardt* em conjunto com Regularização Bayesiana [5] para controle da distribuição dos pesos \mathbf{W} . A rede foi inicializada e treinada 100 vezes, cada uma com seus pesos escolhidos de acordo com o algoritmo de Nguyen-Widrow [13].

As métricas utilizadas consistem na tabela de confusão e acurácia (Equação 16) média das 100 iterações.

$$Acc = \frac{Acertos}{Acertos + Erros} \quad (16)$$

5 Resultados

Utilizando o modelo descrito na Seção 4, foi possível encontrar uma matriz de confusão média das 100 rodadas (Tabela 0) junto com seu respectivo desvio padrão (Tabela 0) e uma acurácia média de 84.22% no conjunto de Validação.

Tabela 2: Média da matriz de confusão do conjunto de Validação em percentagem (%).

	SP	PE	PI
SP	88.02	9.40	2.58
PE	27.04	71.74	1.22
PI	5.21	4.77	90.02

Tabela 2: Desvio Padrão da matriz de confusão do conjunto de Validação em percentagem (%).

	SP	PE	PI
SP	5.64	5.13	2.78
PE	17.43	17.45	3.08
PI	9.54	9.72	12.28

O alto teor de confusão entre as classes SP e PE já era algo esperado [14] e é um problema que ainda não foi possível solucionar. Entretanto, as classes SP e PI estão representadas com mais de 80% de acerto em média. Infelizmente o desvio padrão das classes PE e PI se mostraram relevantes, o que indica necessidade de implementação de melhorias no modelo, pois o mesmo não conseguiu generalizar completamente para este conjunto de dados.

6 Conclusões

Os resultados apresentados indicam que é certamente possível utilizar um modelo baseado em redes neurais artificiais como mais uma ferramenta no auxílio para uma determinação da integridade física de uma infraestrutura sujeita à fraturas e imperfeições através da emissão acústica.

Estes resultados também indicam que esta pode ser uma solução ainda mais impactante, talvez retirando completamente a necessidade de uma pessoa especialista na avaliação. Esta torna o problema de identificação de fraturas um mais objetivo, eventualmente reduzindo (ou até eliminando por completo) o risco de erro humano.

Não distante, todas as técnicas utilizadas neste artigo não envolvem participação humana, salvo pela definição das faixas de frequência para o cálculo de algumas entradas do modelo. Eventualmente, desenvolvendo um método de definir estas faixas automaticamente ou as fixando, o sistema se torna completamente autônomo.

Agradecimentos

Os autores agradecem à Petrobras, CNPq, CAPES, COPPE e o LAMEF pela cooperação e apoio.

Referências

[1] L. Bottou, “Large-Scale Machine Learning with Stochastic Gradient Descent,” in *Proceedings of COMPSTAT’2010*, pp. 177–186, Physica-Verlag HD, 2010.

[2] A. Carpinteri, G. Lacidogna, and N. Pugno, “Structural damage diagnosis and life-time assessment by acoustic emission monitoring,” *Engineering Fracture Mechanics*, vol. 74, pp. 273–289, Jan. 2007.

[3] M. Dorigo and M. Birattari, “Ant Colony Optimization,” in *Encyclopedia of Machine Learning*, pp. 36–39, Springer, Boston, MA, 2011.

[4] A. Farhidzadeh, S. Salamone, B. Luna, and A. Whittaker, “Acoustic emission monitoring of a reinforced concrete shear wall by b-value-based outlier analysis,” *Structural Health Monitoring*, vol. 12, pp. 3–13, Jan. 2013.

[5] F. D. Foresee and M. T. Hagan, “Gauss-Newton approximation to Bayesian learning,” in *International Conference on Neural Networks, 1997*, vol. 3, pp. 1930–1935 vol.3, June 1997.

[6] Frank Rosenblatt, “The Perceptron, A Perceiving and Recognizing Automaton.,” Tech. Rep. 85-460-1, Cornell Aeronautical Laboratory, INC, Buffalo, N. Y., Jan. 1957.

[7] C. R. Heiple and S. H. Carpenter, “Acoustic emission produced by deformation of metals and alloys - A review.,” *Journal of Acoustic Emission*, vol. 6, pp. 177–204, Sept. 1987.

[8] S. G. Kariuki and K. Löwe, “Integrating human factors into process hazard analysis,” *Reliability Engineering & System Safety*, vol. 92, pp. 1764–1773, Dec. 2007.

[9] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv:1412.6980 [cs]*, Dec. 2014.

[10] T. C. Lindley, I. G. Palmer, and C. E. Richards, “Acoustic emission monitoring of fatigue crack growth,” *Materials Science and Engineering*, vol. 32, pp. 1–15, Jan. 1978.

[11] Mistras Group, “DiSP with AEwin USER’S MANUAL,” July 2011.

[12] J. J. Moré, “The Levenberg-Marquardt algorithm: Implementation and theory,” in *Numerical Analysis, Lecture Notes in Mathematics*, pp. 105–116, Springer, Berlin, Heidelberg, 1978.

[13] D. Nguyen and B. Widrow, “Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights,” in *1990 IJCNN International Joint Conference on Neural Networks*, pp. 21–26 vol.3, June 1990.

[14] C. F. C. Pinto, *MONITORAÇÃO DE DEFEITOS EM DUTOS RÍGIDOS POR ANÁLISE DOS PARÂMETROS DE EMISSÃO ACÚSTICA UTILIZANDO REDES NEURAIS*. PhD thesis, Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Dec. 2014.

[15] J. Shuai, K. Han, and X. Xu, “Risk-based inspection for large-scale crude oil tanks,” *Journal of Loss Prevention in the Process Industries*, vol. 25, pp. 166–175, Jan. 2012.

[16] M. Singh and T. Markeset, “A methodology for risk-based inspection planning of oil and gas pipes based on fuzzy logic framework,” *Engineering Failure Analysis*, vol. 16, pp. 2098–2113, Oct. 2009.

[17] K. Socha and C. Blum, “An ant colony optimization algorithm for continuous optimization: Application to feed-forward neural network training,” *Neural Computing and Applications*, vol. 16, pp. 235–247, May 2007.

[18] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, “Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-scale Bound-constrained Optimization,” *ACM Trans. Math. Softw.*, vol. 23, pp. 550–560, Dec. 1997.