

V Seminário de Iniciação Científica

Talentos da Ciência e Tecnologia em ação

☰ Dias 26 e 27 de setembro de 2019

📍 Auditório e Pátio - Unidade II



APRENDIZAGEM PROFUNDA APLICADA NA IDENTIFICAÇÃO DE DANO NO MONITORAMENTO DE INTEGRIDADE ESTRUTURAL

Raimundo Matos de Barros Neto¹ – Unifesspa

netobarros@unifesspa.edu.br

Adam Dreyton Ferreira dos Santos² - Unifesspa

Adamdreyton@unifesspa.edu.br

Agência Financiadora: PBIC/Unifesspa

Eixo Temático/Área de Conhecimento: Engenharias e Computação

1. INTRODUÇÃO

Em todos os centros urbanos é importante que as construções civis (prédios, casas, estações de metrô, etc.) proporcionem confiabilidade de segurança estrutural para as pessoas. Logo, é cabível a importância de monitoramento da integridade dessas estruturas. Porém, existem diversos fatores que incidem sobre as mesmas, como vento, temperatura, tempo de construção, materiais usados etc., portanto, é difícil mensurar quais são os fatores que afetam diretamente ou indiretamente a performance estrutural dessas construções.

Dado esses fatos, com intuito de colaborar com o devido monitoramento, o Monitoramento da Integridade Estrutural (*Structural Health Monitoring* - SHM), contribui para avaliação estrutural presente e futura. De acordo com Silva et al. (2018), a SHM depende da medida contínua de dados de resposta estrutural, que caracterizam aspectos de sensibilidade a diferentes tipos de danos estruturais. Como foi dito anteriormente, é difícil mensurar especificamente fatores operacionais e ambientais que incidem sob determinadas estruturas. Portanto, essa é uma tarefa bastante complexa que a SHM tenta realizar.

Através da utilização de Inteligência Artificial (IA) em SHM, os sistemas de monitoramento estrutural têm obtido muito destaque no meio de pesquisas, a respeito da avaliação da performance estrutural. Para Maio 2010 (apud ARAUJO, 2017, p. 1) as pesquisas em IA representam uma vantagem se comparadas aos sistemas baseados em inspeções visuais. Mediante isto, a SHM trabalha com intuito de substituir manutenções programadas por inspeções necessárias. É importante frisar de que forma o SHM trabalha para o devido monitoramento, levando em consideração as variabilidades citadas anteriormente. Para tal, são classificadas essas variabilidades como ambientais e operacionais (ARAUJO 2017), condições de medição que geram respostas para a caracterização de dano estrutural.

2. MATERIAS E MÉTODOS

Foi utilizado para o entendimento do algoritmo Autoencoder, uma biblioteca de alto nível, escrita em Python, linguagem de programação utilizada por esta pesquisa para elaboração dos códigos, intitulada de Keras. Juntamente com esta biblioteca, foi usado como mecanismo de desenvolvimento uma outra biblioteca mais específica chamada de TensorFlow. Essas bibliotecas são

¹Raimundo Matos de Barros Neto - Universidade Federal do Sul e Sudeste do Pará

²Adam Dreyton Ferreira dos Santos - Professor Adjunto da Universidade Federal do Sul e Sudeste do Pará (FACEEL/IGE/Unifesspa).

de código aberto, também compatível com Python. A biblioteca Keras pode treinar, validar e testar redes neurais para problemas de classificação, reconhecimento de imagens, redes neurais recorrentes, simulações baseadas em equações diferenciais parciais, etc. (YEGULALP, 2018).

O algoritmo Autoencoder é baseado em uma rede neural treinada com o objetivo de a partir de uma dada entrada (input) copiá-la para uma saída (output), ou seja, o algoritmo tem o objetivo de aprender representações (Encodings - Codificações) de dados, que tem potencial para uma redução de dimensionalidade (SILVA et al., 2018). O algoritmo pode ser dividido em duas partes: 1º - Codificador (Encoder), que é uma função $f(x)$ para transforma sua entrada (Input) para outra representação que será chamada de h ; 2º - Decodificador (Decoder), que é uma função $g(x)$ que irá transformar sua representação h em sua reconstrução r (DANIEL, 2017).

O processo de detecção de dano será feito a partir dos dados medidos de vibrações da Ponte Z-24. Para a extração das frequências naturais da ponte, a partir dos dados de vibração, foi utilizado o método de identificação de sub-espaco estocástico por covariância (ARAUJO 2017). Através deste método às amostras obtidas formam um conjunto de $Z^{3932} \times 4$, sendo 3932 amostras com quatro frequências naturais extraídas dos sinais (ARAUJO 2017). De acordo com as amostras do conjunto de dados, as 3476 primeiras amostras são dados relacionados a condição normal da estrutura e as demais amostras são referentes à condição anormal da ponte.

Os resultados obtidos após o processo de teste com o referido algoritmo serão categorizados de duas formas: Erros Tipo I e Erros Tipo II. Essa categorização é utilizada para avaliação de desempenho de uma classificação binária, o método considera que uma indicação falso-positiva pode ter consequências diferentes de uma indicação falso-negativa (ARAUJO, 2017). Os Erros do Tipo I serão indicados pelos pontos acima da linha do *threshold* (baseado no limiar de 95% de confiança dos dados de treino, onde o algoritmo possui até 5% de possibilidade de erros) na faixa de comportamento normal da estrutura, enquanto os pontos abaixo do *threshold* e na faixa de comportamento anormal da estrutura indicam Erros Tipo II.

3. RESULTADOS E DISCUSSÃO

A partir das configurações supracitadas, os resultados obtidos com a rede Autoencoder foram os seguintes, para a 1º configuração (Figura 1 e Tabela 1).

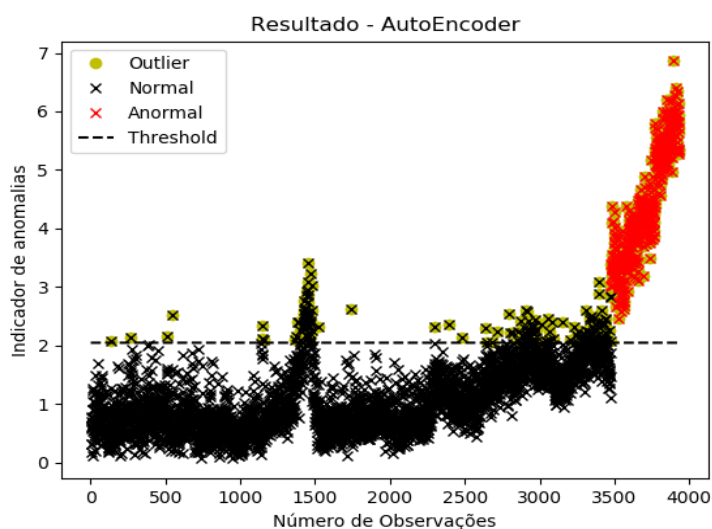


Figura 1 – Resultado para a 1º configuração testada.

Tabela 1 – Resultado de Erros Tipo I e II para a 1ª configuração.

<i>Nº da Configuração</i>	<i>Erro do Tipo I</i>	<i>Erro do Tipo II</i>	<i>Total de Erros</i>
1ª	174 (5.00%)	0.0 (0.0%)	174 (4.42%)

Para a configuração 1, o algoritmo respeitou a condição do limiar e se manteve em seu percentual de erros dentro do aceitável. Desta forma, o algoritmo obteve um excelente resultado quanto ao resultado do Tipo II, zerando o seu valor e atenuando os dados dos erros Tipo I. Mais a frente, vemos os resultados obtidos da rede de acordo com os parâmetros da 2ª e 3ª configuração nas Figuras 2 e 3, e Tabelas 3 e 4.

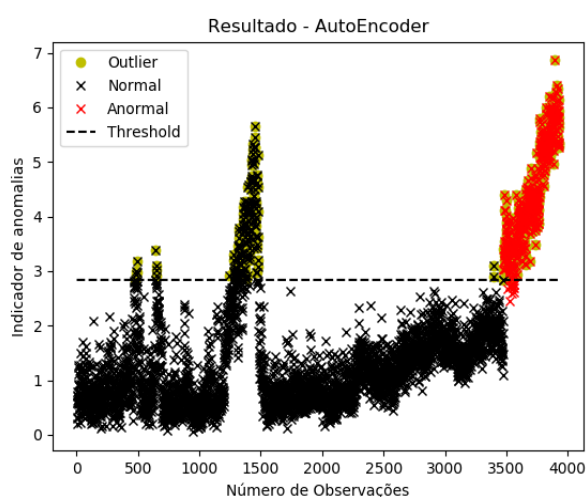


Figura 2 – Resultado para a 2ª configuração testada.

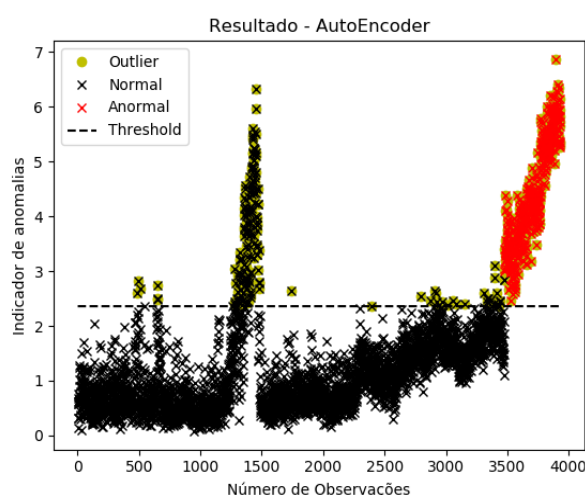


Figura 3 – Resultado para a 3ª configuração testada.

Tabela 2 – Resultado de Erros Tipo I e II para a 2ª configuração.

<i>Nº da Configuração</i>	<i>Erro do Tipo I</i>	<i>Erro do Tipo II</i>	<i>Total de Erros</i>
2ª	174 (5.00%)	12 (2.63%)	186 (4.73%)

Tabela 3 – Resultado de Erros Tipo I e II para a 3ª configuração.

<i>Nº da Configuração</i>	<i>Erro do Tipo I</i>	<i>Erro do Tipo II</i>	<i>Total de Erros</i>
3ª	174 (5.00%)	0.0 (0.0%)	174 (4.42%)

Na configuração 2, o algoritmo não obteve um melhor resultado se comparado a 1ª e 3ª configuração, devido ao aparecimento dos erros do Tipo II (12), elevando seu percentual total de erros, denotando um mal desempenho da rede com a devida configuração. Cabe ressaltar que essa configuração teve os mesmos parâmetros da 1ª configuração, diferindo apenas pela função de otimização, que no caso, foi utilizada a função Adam, no lugar da Adadelta utilizada nas demais configurações. Para a última configuração, os resultados são exemplificados na Figura 3 e Tabela 3.

A 3ª configuração obteve um resultado satisfatório, assim como a 1ª. É importante observar que os parâmetros dessa configuração se equiparam a 1ª configuração apenas pelo fato da função de

ativação da última cama ser a função Relu, pois, as camadas anteriores estão com a função Tangente (Tanh). Mesmo com o resultado satisfatório da configuração, já que não obteve nenhum erro do Tipo II, ocorreu assim como na 2ª configuração, uma discrepância na distribuição dos seus dados, referentes aos erros do Tipo I. Vale ressaltar que, todas as configurações neste trabalho foram inicializadas com 500 iterações.

4. CONSIDERAÇÕES FINAIS

O resultado demonstrado pela 1ª configuração conseguiu obter o melhor resultado se comparado com a 2ª e 3ª configurações, pois além de atenuar os erros do Tipo I, que são ilustrados na Figura 2, não identificou nenhum erro do Tipo II. No entanto, a 3ª configuração não conseguiu atenuar os erros do Tipo I. Esta discrepância identificada pelo algoritmo na 3ª configuração, é explicada pelo fato da ponte congelar em períodos de extremo frio, na época de inverno. Estas duas configurações (1ª e 3ª) tiveram como parâmetro a mesma função de ativação na última camada da rede neural, a função não linear Relu.

Já a 2ª configuração teve como resultado o mesmo número de erros do Tipo I como as demais, assim como a mesma discrepância nos dados apresentado pela 3ª configuração, porém, referente aos erros do Tipo II, o algoritmo não repetiu o bom resultado, encontrando 12 erros. Cabe citar que é extremamente importante que se evite ao máximo esses erros, pois eles se referem a condição danificada da estrutura. Logo seu resultado não foi satisfatório se comparado com os demais resultados, pelo seguinte fato, a função Adam se adequa melhor a problemas de classificação de dados, o que não é o caso para este trabalho, já a função Adadelta se encaixa na remoção de ruídos (variações), que é exatamente o objetivo desta pesquisa.

REFERÊNCIAS

- SILVA, Moisés et al., **Deep principal component analysis: An enhanced approach for structural damage identification**. In: Structural Health Monitoring (2018).
- ARAUJO, Alessandra P. **Aplicação de algoritmo de clusterização na detecção de danos estruturais**. Monografia (Bacharel em Ciências da Computação). 59p. Belém. UFPA, 2017.
- Silva, M, Santos, A, Figueiredo, E. **A novel unsupervised approach based on a genetic algorithm for structural damage detection in bridges**. Eng Appl Artif Intel 2016; 52: 168–180.
- Yegulalp, Serdar. **O que é o TensorFlow? A biblioteca de machine learning explicada**. Blog.hdstore, disponível em: <<https://blog.hdstore.com.br>>. Acesso em: 22 Novembro de 2018.
- Daniel, José. **Construindo Autoencoders**. Curso-r, disponível em: <<https://www.curso-r.com>>. Acesso em: 22 de Novembro de 2018.